BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi yang pesat telah mendorong penerapan *artificial intelligence* (AI) dalam berbagai bidang, termasuk kesehatan. Deteksi dini penyakit kanker merupakan salah satu bidang yang mendapat perhatian khusus, di mana pemanfaatan *machine learning* (ML) dan *neural network* telah terbukti membantu proses diagnosis yang lebih cepat dan akurat [1][2]. Namun, integrasi antara model AI dengan sistem aplikasi berbasis *web* yang digunakan pengguna akhir masih menjadi tantangan [3]. Model AI biasanya dikembangkan secara terpisah menggunakan *framework* seperti TensorFlow atau PyTorch tanpa arsitektur *backend* yang terstruktur untuk mengatur komunikasi dengan sistem *frontend* [4].

Permasalahan *deployment* model AI ke lingkungan *production* masih signifikan. Studi menunjukkan bahwa hanya 34% model AI eksperimental yang berhasil mencapai tahap *production*, dengan 55% organisasi menyebutkan kurangnya praktik MLOps sebagai hambatan utama [5][6]. Dalam konteks *healthcare*, evaluasi performa dari sisi arsitektur *backend* masih terbatas meskipun banyak penelitian membahas pengembangan model AI untuk diagnosis medis [7]. Penelitian *existing* cenderung fokus pada akurasi model dan mengabaikan aspek *system-level performance* seperti *response time*, *throughput*, dan *scalability* yang kritis untuk implementasi klinis [8].

Evaluasi performa *backend framework* seperti Node.js dan FastAPI umumnya dilakukan untuk kasus penggunaan umum seperti operasi CRUD dengan *payload* kecil, bukan untuk *workload* spesifik seperti ML *inference* yang bersifat *compute-intensive* dan memiliki *response time* bervariasi [9]. *Survey* tentang REST API *testing* juga menunjukkan bahwa hanya 15% studi yang membahas *performance testing*, dengan mayoritas fokus pada *functional testing* [10]. Studi tentang arsitektur *microservices* pun masih menyisakan gap dalam evaluasi performa pola API *gateway* untuk skenario ML *serving* di bawah beban tinggi [11].

Sistem AIRA (*AI Research for Oncology*) merupakan sistem berbasis AI yang dirancang untuk membantu dokter dalam melakukan deteksi dini kanker prostat dan payudara menggunakan data genomik. Sistem ini menggunakan arsitektur *backend* yang terdiri dari AI *Backend* berbasis FastAPI untuk inferensi

multiple machine learning models, dan *Cancer Gateway* berbasis Node.js (Express) yang bertindak sebagai API *gateway* antara sistem *frontend* dengan AI *Backend*. Arsitektur dirancang dengan pendekatan *hybrid* di mana kedua *service* di-*containerize* menggunakan Docker namun *co-located* dalam satu *server* untuk mengoptimalkan *latency* komunikasi antar-*service*.

Berbeda dengan penelitian *existing* yang umumnya berfokus pada perancangan arsitektur atau peningkatan akurasi model tanpa mengkaji kesiapan sistem dari sisi performa, penelitian ini mengangkat masalah belum adanya evaluasi empiris terhadap performa arsitektur *backend* dan *API gateway* pada sistem AI diagnosis kanker yang ditujukan untuk penggunaan dengan beban tinggi. Untuk menjawab permasalahan tersebut, sistem AIRA dilengkapi dengan evaluasi performa melalui *load testing* dan *stress testing*. Evaluasi difokuskan pada metrik *response time*, *throughput*, *error rate*, dan *resource utilization* di bawah beban tinggi untuk mengidentifikasi *bottleneck* dalam arsitektur, memahami dampak keberadaan *intermediary gateway layer* terhadap *end-to-end latency*, serta menentukan *acceptable performance thresholds* bagi sistem *machine learning* di bidang kesehatan. Dengan demikian, penelitian ini tidak hanya berkontribusi pada perancangan arsitektur, tetapi juga menyediakan data performa empiris yang relevan sebagai dasar pengambilan keputusan bagi praktisi yang ingin mengimplementasikan sistem serupa.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang arsitektur *backend* yang efisien untuk mengintegrasikan *multiple machine learning models* dengan sistem *web* menggunakan API *gateway pattern*?
2. Bagaimana performa arsitektur *backend* yang dirancang ketika diuji dengan *stress testing* dan *load testing* untuk menangani *concurrent prediction requests* pada ML *inference*?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Merancang arsitektur *backend* yang modular dengan *separation of concerns* antara API *gateway* (*Cancer Gateway*) dan ML *inference service* (AI *Backend*) untuk sistem deteksi kanker.

2. Mengevaluasi performa arsitektur *backend* melalui *stress testing* menggunakan *load testing tools* untuk mengukur *response time*, *throughput*, *error rate*, dan *resource utilization* di bawah beban tinggi.

1.4 Batasan Masalah

Agar penelitian ini tetap fokus dan terarah, maka ditetapkan batasan masalah sebagai berikut:

1. Penelitian fokus pada evaluasi performa arsitektur *backend*, bukan pada pengembangan atau optimalisasi model *machine learning*. Model yang digunakan adalah *pre-trained models* untuk deteksi kanker prostat dan payudara dengan data genomik yang sudah tersedia.

2. Evaluasi performa dilakukan melalui *load testing* dengan fokus pada skenario *stress testing* dalam *controlled environment* menggunakan *tools* seperti Apache JMeter atau Locust. Metrik yang diukur meliputi *response time*, *throughput*, *error rate*, dan *resource utilization*. Evaluasi tidak mencakup *security testing* atau *compliance testing*.

3. *Deployment* dilakukan dalam *single server* dengan *co-located containers* menggunakan Docker. Evaluasi untuk *distributed deployment* atau arsitektur *multi-server* tidak termasuk dalam *scope* penelitian.

1.5 Urgensi Penelitian

Sistem AIRA ini memiliki urgensi tinggi karena gap antara pengembangan model ML dan *deployment* ke *production* masih menjadi hambatan signifikan dalam implementasi AI di bidang kesehatan. Studi menunjukkan bahwa hanya sepertiga model ML berhasil di-*deploy* ke *production*, dan 55% organisasi menyebutkan kurangnya praktik MLOps sebagai hambatan utama [5][6]. Dalam konteks *healthcare*, *system responsiveness* merupakan faktor kritis yang mempengaruhi *clinical decision-making*, namun evaluasi performa arsitektur *backend* secara sistematis masih jarang dilakukan [7][8]. Penelitian *existing* lebih banyak berfokus

pada akurasi model tanpa menguji bagaimana sistem berperforma di bawah beban tinggi. Melalui implementasi dan *stress testing* pada sistem AIRA, penelitian ini diharapkan dapat memberikan data empiris dan *insight* mengenai karakteristik performa arsitektur *backend* untuk sistem ML di bidang kesehatan.

1.6 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah:

1. Bagi akademisi: menyediakan data performa empiris dan metodologi evaluasi untuk arsitektur *backend* ML *inference systems* yang dapat dijadikan referensi dalam penelitian lanjutan tentang MLOps dan sistem AI di bidang kesehatan.
2. Bagi industri kesehatan: menyediakan *benchmark performance metrics* untuk sistem ML di bidang kesehatan yang dapat dijadikan panduan dalam menentukan *acceptable response time* dan persyaratan *throughput* untuk *clinical deployment*, khususnya pada skenario beban tinggi.
3. Bagi penelitian lanjutan: membuka peluang penelitian lebih lanjut mengenai optimalisasi performa *backend*, *scalability testing* dengan peningkatan *workload*, dan perbandingan dengan pola arsitektur alternatif.

1.7 Luaran Penelitian

Luaran yang diharapkan dari penelitian ini meliputi:

1. Prototipe sistem *backend* terintegrasi yang terdiri dari *Cancer Gateway* dan AI *Backend* dengan *containerization* Docker yang dapat menangani *multiple cancer detection models* dengan berbagai tipe data.
2. *Dataset* hasil evaluasi performa yang mencakup metrik *response time*, pengukuran *throughput*, analisis *error rate*, data utilisasi *resource*, dan hasil identifikasi *bottleneck* dari skenario *stress testing*.
3. Artikel ilmiah yang membahas perancangan, implementasi, dan evaluasi performa arsitektur *backend* untuk sistem ML *inference* di bidang kesehatan yang dapat dipublikasikan di konferensi atau jurnal ilmiah.

4. Rekomendasi *best practices* dan *lessons learned* dalam mengimplementasikan pola API *gateway* untuk ML *serving* dalam konteks kesehatan yang dapat dijadikan panduan untuk praktisi.