BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan pesat teknologi Artificial Intelligence (AI) telah marak digunakan di berbagai sektor industri dan menjadi faktor utama transformasi industri global, dengan prediksi dampak ekonomi mencapai $15,7 triliun pada tahun 2030 dan peningkatan produktivitas hingga 40% [1]. Dalam ekosistem ini, cloud computing berperan sebagai fondasi elastis yang memungkinkan implementasi sistem AI modern untuk menangani beban kerja komputasi. Google Cloud Platform (GCP) muncul sebagai salah satu platform utama dengan tingkat pengguna mencapai 36,8%, didukung oleh ekosistem terpadu yang memfasilitasi pemrosesan data skala besar secara fleksibel [2]. Bagi perusahaan teknologi besar seperti Alibaba, optimalisasi dan efisiensi manajemen infrastruktur komputasi AI krusial untuk menjaga kestabilan serta kinerja layanan [3]. Relevansi topik ini menjadi semakin penting seiring dengan meningkatnya kebutuhan industri akan sumber daya yang mampu mengimbangi pesatnya evolusi model AI saat ini.

Namun, fakta di lapangan menunjukkan bahwa infrastruktur *single-node* sering kali tidak sanggup menangani model AI yang semakin masif, terutama pada fase *inference* yang menuntut skalabilitas tinggi dan menyerap lebih dari 90% total biaya infrastruktur komputasi AI di industri [3]. Tingginya risiko saturasi komputasi vCPU pada tingkat beban ekstrem serta keterbatasan memori fisik pada satu mesin memaksa sistem bertransisi ke arsitektur terdistribusi [4], [5]. Masalah utama yang muncul adalah ketidakefisienan akibat *communication bottleneck* dan latensi jaringan yang sering kali lebih dominan daripada kecepatan komputasi itu sendiri [6]. Tanpa adanya pemahaman arsitektur yang baik, penambahan sumber daya justru berisiko menurunkan efisiensi sistem secara keseluruhan dan memicu fenomena *over-provisioning* yang menyebabkan pemborosan biaya [7].

Fenomena beban ini bukan sekadar isu teoretis yang terbatas pada skala industri teknologi besar, melainkan juga tercermin nyata pada layanan digital institusi pendidikan. Sistem layanan akademik kampus seperti pengisian Kartu Rencana Studi (KRS) hingga pembayaran UKT, terbukti rentan mengalami lonjakan trafik ekstrem pada periode-periode tertentu, khususnya di awal semester

[8]. Kondisi ini sejalan dengan hasil wawancara bersama staf IT kampus (Lampiran 1) yang telah menerapkan pengujian beban (*stress test*) pra-rilis pada sistem akademik berbasis *hybrid cloud* untuk menguji lonjakan trafik saat *event* besar seperti ujian masuk, namun masih menghadapi tantangan untuk menentukan estimasi alokasi *resource* yang benar-benar presisi. Meskipun pengujian beban (*stress test*) manual yang diterapkan dinilai efektif untuk kebutuhan jangka pendek, pendekatan ini masih mengandalkan estimasi dan pengujian berulang tiap *event*, belum berdasarkan acuan yang terstandardisasi secara empiris. Keterbatasan acuan ini berpotensi menimbulkan risiko *over-provisioning* yang berdampak pada biaya sewa *cloud*. Karakteristik beban kerja ini merepresentasikan tantangan skalabilitas horizontal yang sejalan dengan dinamika *inference* AI berskala industri.

Penelitian sebelumnya telah banyak mengeksplorasi optimalisasi sistem, seperti penggunaan *Machine Learning* untuk manajemen sumber daya dan strategi *multi-cloud* yang terbukti menekan biaya operasional hingga 25% [7], [9]. Studi lain juga telah mengembangkan metode penjadwalan *hybrid* untuk meningkatkan skalabilitas beban kerja AI [10]. Dari sisi penyedia layanan, GCP tercatat unggul dalam menangani model ML [11], dan dalam ekosistem GCP arsitektur terdistribusi tersebut diwujudkan melalui Google Kubernetes Engine (GKE) selaku standar orkestrasi container. Sebagai instrumen uji, arsitektur MobileNetV2 efektif diadopsi sebagai representasi karakteristik beban kerja [12], yang telah divalidasi secara akademis oleh standar industri MLPerf [13] serta terbukti sukses diterapkan pada lingkungan kontainer [14]. Meski demikian, sejumlah literatur turut mengisyaratkan adanya celah kajian pada variabilitas kinerja infrastruktur cloud publik dan penempatan kontainer yang optimal [15]

Terlepas dari berbagai temuan tersebut, sebagian besar riset terdahulu berfokus pada pengembangan algoritma optimasi kustom dan *custom scheduler*, sehingga performa murni bawaan dari ekosistem cloud belum dipetakan secara mendalam. Di sisi lain, dokumentasi resmi Kubernetes/GKE sendiri hanya menjelaskan mekanisme kerja komponen orkestrasi secara desain seperti algoritma penskalaan dan ambang pemicunya tanpa menjabarkan secara rinci karakteristik atau perilaku empirisnya ketika benar-benar dihadapkan pada beban kerja AI yang bersifat intensif dan fluktuatif [16]. Ketiadaan data dasar ini berisiko menimbulkan

ketidakpresisian dalam alokasi sumber daya, yang berujung pada *over-provisioning* dan pemborosan biaya operasional. Selain itu, studi-studi tersebut umumnya belum menyertakan analisis dinamika performa pada level kontainer secara empiris, terutama dalam mengaitkan variabilitas kinerja dengan batas skalabilitas horizontal yang terukur. Kesenjangan ini sejalan dengan rekomendasi riset terdahulu yang meminta pengujian kestabilan sistem di bawah tekanan beban fluktuatif ekstrem [15], [17]. Untuk mengisi kesenjangan tersebut, penelitian ini menawarkan analisis empiris atas pola perubahan parameter teknis pada GKE, melalui metode *benchmarking* beban kerja AI MobileNetV2 dengan dataset CIFAR-10 untuk menghasilkan skema penyesuaian parameter teknis yang divalidasi secara kuantitatif dan dibandingkan langsung dengan mekanisme resmi platform menjembatani kesenjangan antara pemahaman berbasis desain dan kebutuhan pengambilan keputusan berbasis bukti empiris.

Berdasarkan seluruh urutan komparasi tersebut, masalah utama dalam penelitian ini diidentifikasi ke dalam dua poin. Pertama, kompleksitas arsitektur terdistribusi pada GKE berpotensi menimbulkan *overhead* komunikasi dan sinkronisasi antar-node yang dapat menghambat efisiensi komputasi paralel. Kedua, ketiadaan validasi empiris terhadap perilaku bawaan GKE saat menghadapi lonjakan beban fluktuatif, meskipun mekanismenya telah terdokumentasi secara resmi dari sisi desain. Kedua hambatan tersebut menegaskan urgensi dilakukannya pemetaan batas skalabilitas horizontal secara empiris untuk menemukan titik jenuh operasional klaster.

Berangkat dari identifikasi masalah tersebut, penelitian ini bertujuan untuk menganalisis kinerja arsitektur *cloud* terdistribusi dalam menangani beban kerja AI pada Google Cloud Platform menggunakan Google Kubernetes Engine (GKE), sekaligus menyusun dan memvalidasi secara analitis skema penyesuaian parameter teknis sebagai solusi mitigasi atas karakteristik performa yang teridentifikasi. Analisis dilakukan dengan mengukur variabel latensi, *throughput*, serta utilisasi CPU pada berbagai tingkat intensitas beban kerja. Hasil dari penelitian ini diharapkan dapat memberikan pemahaman mendalam mengenai karakteristik respon sistem terhadap beban kerja AI, serta menjadi dasar teknis dalam

menentukan konfigurasi infrastruktur yang paling optimal, stabil, dan memiliki performa efisiensi yang tinggi bagi arsitek sistem.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana karakteristik performa cloud terdistribusi GKE yang meliputi throughput, latensi, dan utilisasi sumber daya CPU dalam menangani variasi beban kerja inferensi AI MobileNetV2?
2. Bagaimana efektivitas sistem terdistribusi GKE dalam menjaga stabilitas sistem serta meminimalkan dampak overhead (diminishing return) saat menghadapi variabilitas beban kerja inferensi AI MobileNetV2?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dikemukakan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisis karakteristik performa cloud terdistribusi GKE yang meliputi throughput, latensi, dan utilisasi sumber daya CPU dalam menangani variasi beban kerja inferensi AI MobileNetV2.
2. Mengevaluasi efektivitas sistem terdistribusi GKE dalam menjaga stabilitas sistem serta meminimalkan dampak *overhead* (*diminishing return*) saat menghadapi variabilitas beban kerja inferensi AI MobileNetV2.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan beberapa manfaat, antara lain:

1.4.1 Manfaat Teoritis

Penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang cloud computing dan artificial intelligence. Hasil penelitian ini diharapkan dapat menjadi referensi ilmiah bagi penelitian selanjutnya yang berfokus pada:

1. Pengembangan ilmu komputasi khususnya ilmu sistem terdistribusi dan komputasi awan melalui pemodelan beban kerja AI.

2. Memberikan pemahaman mendalam mengenai pola perilaku, batas jenuh, dan respons nyata arsitektur cloud ketika menangani proses inferensi deep learning yang berat (resource-intensive).

1.4.2 Manfaat Praktis

Secara praktis, hasil penelitian ini diharapkan dapat memberikan acuan teknis dalam penerapan sistem AI berbasis cloud terdistribusi yiatu sebagai berikut:

1. Menjadi acuan teknis bagi Cloud Engineer dan DevOps dalam mengonfigurasi sumber daya klaster di lingkungan Google Kubernetes Engine secara efisien, serta menyajikan data komparatif untuk memandu pemilihan arsitektur statis maupun autoscaling.
2. Memberikan panduan taktis untuk memitigasi penurunan performa akibat CPU throttling sekaligus menyediakan solusi untuk mencegah eror kehabisan memori.
3. Menyediakan skema penyesuaian parameter *Horizontal Pod Autoscaler* (*stabilization window*) yang selaras dengan dokumentasi resmi Kubernetes, sebagai acuan mitigasi fenomena *pod thrashing* pada beban kerja AI yang bersifat fluktuatif.

1.5 Batasan Masalah

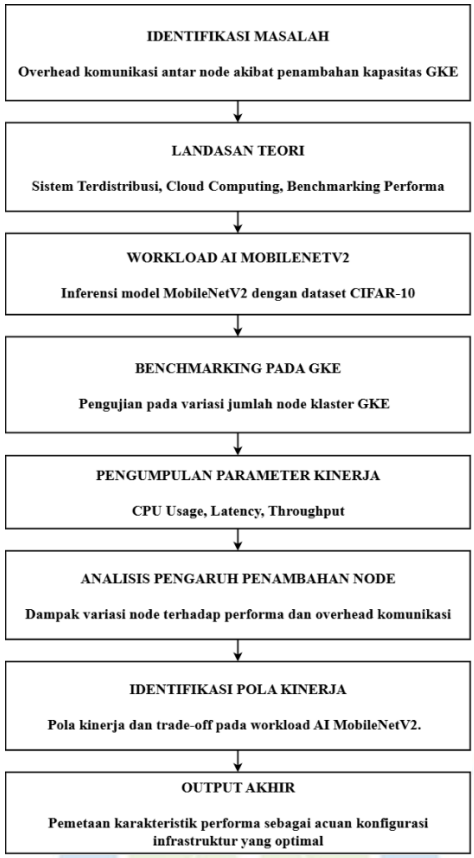
Agar penelitian ini memiliki ruang lingkup yang terarah dan fokus, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Pengujian dibatasi pada Google Cloud Platform (GCP) dengan layanan utama GKE yang dijalankan pada satu region saja.
2. Spesifikasi cluster GKE dibatasi pada tipe mesin tertentu yaitu E2-standard-2 untuk membentuk arsitektur terdistribusi dan baseline pembanding, tanpa mengeksplorasi seluruh variasi tipe mesin di GCP.
3. Model deep learning yang digunakan secara eksklusif hanya MobileNetV2 (CNN), tanpa melakukan komparasi dengan arsitektur model AI lainnya.
4. Model dibatasi hanya pada proses inferensi (*prediction*) menggunakan dataset publik yang bersifat statis, sehingga tidak mencakup fase pelatihan (*training*) model.

5. Penelitian ini sama sekali tidak mengevaluasi atau mengukur tingkat akurasi, presisi, atau kualitas hasil prediksi dari model AI yang digunakan.
6. Variasi beban kerja dibatasi pada manipulasi jumlah pengguna simultan secara bertahap, yang ditetapkan secara kaku pada tingkat 10, 50, dan 100 pengguna.
7. *Benchmarking* difokuskan pada tiga metrik performa: *throughput*, latensi, dan utilisasi CPU.
8. Pengujian fitur autoscaling pada cluster hanya menggunakan Horizontal Pod Autoscaler (HPA), tanpa melibatkan Vertical Pod Autoscaler (VPA) atau custom metric autoscaler eksternal..
9. Pengujian mengabaikan faktor jaringan eksternal seperti internet latency atau intervensi traffic public serta tidak menggunakan konfigurasi load balancing di luar standar bawaan GKE.
10. Penelitian tidak membahas aspek keamanan, privasi data, *fault tolerance*, serta tidak melakukan analisis biaya finansial secara langsung (*cloud billing*). Fokus penelitian murni dibatasi pada pemetaan batas teknis efisiensi komputasi terdistribusi.

1.6 Kerangka Pemikiran

Untuk memberikan gambaran logis mengenai alur penyelesaian masalah dalam penelitian ini, disusun sebuah kerangka pemikiran sistematis yang ditunjukkan pada Gambar 1.1 Kerangka pemikiran ini bergerak secara kronologis mulai dari tahap identifikasi masalah di lapisan infrastruktur hingga menghasilkan output berupa acuan optimasi sistem.



Gambar 1.1 Diagram Alur Kerangka Pemikiran Penelitian

Secara rinci, penjelasan dari setiap tahapan pada Gambar 1.1 adalah sebagai berikut:

1. Identifikasi Masalah

Alur kerangka pemikiran ini diawali dari adanya sebuah fenomena masalah arsitektural pada ekosistem *public cloud*. Peningkatan jumlah *node* pada arsitektur GKE secara teoretis bertujuan untuk meningkatkan kapasitas komputasi, namun pada realitasnya berpotensi menimbulkan *overhead* yang memicu fenomena *diminishing return* [18]. Penambahan *node* ini memicu munculnya *overhead* komunikasi dan sinkronisasi jaringan antar-*node* yang tinggi [19], yang berdampak pada pemborosan biaya.

2. Landasan Teori

Untuk membedah masalah secara objektif, penelitian ini berdiri di atas tiga pilar landasan teoretis utama, yaitu:

a. Arsitektur Sistem Terdistribusi: Untuk memahami karakteristik interkoneksi, pembagian beban kerja, dan koordinasi antar-node komputasi.

b. Cloud Computing: Sebagai fondasi pemahaman lingkungan *Cloud multi-tenancy* tempat klaster GKE berjalan.
c. Benchmarking & Performance Evaluation: Sebagai panduan ketat dalam merancang prosedur eksperimen yang valid, terukur, dan dapat direplikasi secara ilmiah.

3. Spesifikasi Beban Kerja

Berlandaskan kerangka teori tersebut, penelitian ini membutuhkan instrumen uji yang representatif terhadap karakteristik beban kerja industri modern, yaitu beban kerja kecerdasan buatan (*AI workload*) MobilenetV2 berbasis komputasi murni CPU yang intensif dan menuntut sinkronisasi data cepat.

4. Eksekusi Pengujian

Instrumen uji yang telah dirumuskan kemudian diimplementasikan melalui serangkaian empirical benchmarking secara nyata pada lingkungan Google Cloud Platform (GCP), dengan merancang variasi skenario yang merepresentasikan perbedaan kapasitas node dan intensitas beban pengguna.

5. Pengumpulan Parameter Kinesis

Selama proses pengujian berlangsung, sistem dipantau untuk menangkap parameter performa secara *multi-layer* yang mencakup metrik dari sisi klien maupun sisi infrastruktur internal *node* guna merekam respons sistem secara komprehensif terhadap setiap skenario beban yang diujikan

6. Analisis Pengaruh Penambahan Node

Data performa yang telah dikumpulkan kemudian dianalisis melalui dua analisis utama:

a. Analisis Sebab-Akibat Statistik: Untuk membuktikan hubungan sebab-akibat antara penambahan node dengan kemunculan *overhead* komunikasi jaringan dan penurunan efisiensi sistem (*diminishing return*).
b. Analisis Kontainer: untuk mengetahui penyebab penurunan performa aplikasi saat menghadapi lonjakan beban kerja ekstrem, dengan menelusuri gangguan operasional pada level kontainer yang muncul akibat keterbatasan sumber daya fisik *node*.

7. Identifikasi Pola Kinerja

Hasil analisis dari tahap sebelumnya digunakan untuk mengidentifikasi pola respons sistem secara empiris, termasuk titik jenuh sistem serta pola *trade-off* antara kestabilan sistem dan kecepatan waktu respons di bawah tekanan berbagai skenario beban pengguna simultan.

8. Output Akhir

Ujung dari seluruh rangkaian kerangka pemikiran ini adalah menghasilkan output berupa *blueprint*/*action plan* pemetaan batas karakteristik respons performa GKE terhadap beban kerja AI. *Blueprint* ini berperan sebagai dasar acuan teknis yang valid bagi para arsitek sistem terdistribusi dalam menentukan konfigurasi infrastruktur klaster yang paling optimal, stabil, dan memiliki performa efisiensi tinggi tanpa membuang anggaran biaya cloud secara berlebih.

1.7 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini disusun untuk memberikan gambaran mengenai isi setiap bab secara sistematis. Adapun sistematika penulisan yang digunakan adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan dasar pelaksanaan penelitian yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, kerangka pemikiran, serta sistematika penulisan sebagai gambaran umum isi laporan.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan landasan teori, penelitian terdahulu, konsep, model, dan referensi yang berkaitan dengan topik penelitian sebagai dasar dalam penyusunan dan analisis penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan, meliputi pendekatan penelitian, teknik pengumpulan data, metode analisis data, serta tahapan penelitian yang dilakukan untuk mencapai tujuan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penelitian yang diperoleh, disertai analisis dan pembahasan berdasarkan teori maupun penelitian terdahulu untuk menjawab rumusan masalah yang telah ditetapkan.

BAB V SIMPULAN DAN SARAN

Bab ini memuat simpulan yang diperoleh dari hasil penelitian serta saran yang dapat dijadikan masukan bagi penelitian selanjutnya maupun pihak-pihak yang berkepentingan.