

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Dalam penulisan pengembangan ini, beberapa penelitian terkait digunakan sebagai referensi. Jurnal dan penelitian yang memiliki keterkaitan dengan topik yang dibahas dan memiliki tujuan serupa dalam mengembangkan sistem monitoring.

Tabel 1 Penelitian Terkait

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
1	Dede Rahman, Hydra Amnur, Indri Rahmayuni (2020)	Judul : Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram Tools : Prometheus, Grafana, Telegram, Apache Exporter, MySQL Exporter Pokok Bahasan : Penelitian ini membahas monitoring kondisi server dan layanan berbasis Linux menggunakan Prometheus untuk mengumpulkan <i>metrics</i> dan visualisasi dengan Grafana, serta notifikasi melalui Telegram jika ada masalah pada layanan atau <i>resource server</i> . Hasil Penelitian : Sistem berhasil mengirimkan notifikasi <i>real-time</i> ke Telegram saat layanan server bermasalah dan membantu administrator dalam memantau performa server secara lebih efisien
2	Saleh Dwiyatno, Edy Rakhmat, Oki Gustiawan (2020)	Judul: Implementasi Virtualisasi Server Berbasis Docker Container Tools: Docker, Prometheus, Grafana, Apache JMeter, Portainer, Putty Pokok Bahasan: Penelitian ini membahas implementasi virtualisasi server berbasis Docker Container untuk efisiensi penggunaan resource CPU dan memori. Sistem diuji untuk menilai performa container saat menerima beban pengguna yang bervariasi, serta dilakukan monitoring resource menggunakan Prometheus dan Grafana. Hasil Penelitian: Hasil pengujian menunjukkan bahwa Docker Container meningkatkan efisiensi server dengan

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		penggunaan CPU di bawah 15% dan memori di bawah 7 MiB, meski dalam kondisi hingga 1000 user request. Sistem berhasil dijalankan dan dimonitor melalui Grafana.
3	Muhammad Aliyul Husna, Perani Rosyani (2021)	Judul : Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Zabbix yang Terintegrasi dengan Grafana dan Telegram Tools : Zabbix, Grafana, Telegram, SNMP, Mikrotik Pokok Bahasan : Penelitian ini membahas monitoring jaringan dan server menggunakan Zabbix untuk mengumpulkan data performa, dengan visualisasi di Grafana dan notifikasi melalui Telegram untuk gangguan atau anomali. Hasil Penelitian : Penelitian tersebut menunjukkan bahwa sistem berhasil memberikan pemantauan real-time pada perangkat jaringan. Dengan integrasi Telegram, notifikasi gangguan dapat dikirimkan secara otomatis sehingga mempercepat respons tim teknis.
4	Ach Izalul Haq, Banu Santoso (2021)	Judul: Analisis Perbandingan Performa Metode ELK Stack dan Grafana Loki Pada Honeypot Server. Tools: Grafana Loki (Promtail, Loki, Grafana), ELK Stack (Filebeat, Logstash, Elasticsearch, Kibana), Glances Pokok Bahasan: Penelitian ini membandingkan performa dua metode visualisasi log Grafana Loki dan ELK Stack dalam menganalisis dari honeypot server saat terjadi serangan. Hasil Penelitian: Grafana Loki terbukti lebih efisien dari segi penggunaan CPU dan memori dibandingkan ELK Stack. Namun, fitur ELK Stack lebih lengkap. Penggunaan CPU dan memori ELK Stack jauh lebih tinggi. Visualisasi data berhasil dilakukan dengan dashboard di Grafana dan Kibana.

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
5	Ramadoni, Mahmud Z. Amirudin, Rifki Fahmi, Ema Utami, M. Syukri Mustafa (2021)	Judul: Evaluasi Penggunaan Prometheus dan Grafana Untuk Monitoring Database MongoDB Tools: Prometheus, Grafana, mongodb_exporter, node_exporter, Alertmanager Pokok Bahasan: Penelitian ini menganalisis penggunaan Prometheus dan Grafana untuk memantau performa database MongoDB secara <i>real-time</i> melalui metrik seperti CPU, memory, koneksi, query, dokumen dan replikasi. Hasil Penelitian: Prometheus & Grafana terbukti mudah dikonfigurasi, open source, dan dapat menampilkan visualisasi yang informatif dan praktis. Namun, alerting belum diimplementasikan, hanya disarankan untuk penelitian lanjutan.
6	Mohit Bajpai (2022)	Judul : <i>Automating Monitoring and Incident Management with Prometheus, Grafana, and Google Cloud Pub/Sub.</i> Tools : Prometheus, Grafana, Google Cloud Pub/Sub, Jira Pokok Bahasan : Penelitian ini membahas otomatisasi monitoring dan manajemen insiden dengan integrasi alerting. Prometheus ke sistem tiket seperti Jira Service Management, menggunakan Google Cloud Pub/Sub untuk menghubungkan <i>alert</i> dan pembuatan tiket. Hasil Penelitian : Sistem dapat mengotomatiskan pembuatan tiket insiden berdasarkan alert dari Prometheus, meningkatkan efisiensi manajemen insiden, dan mengurangi waktu respons terhadap kegagalan sistem.
7	Guntoro Yudhy Kusuma, Unan Yusmaniar Oktiawati (2022)	Judul : Perancangan Sistem Monitoring Performa Aplikasi Menggunakan Opentelemetry dan Grafana Stack Tools : OpenTelemetry, Prometheus, Jaeger, Grafana

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		Pokok Bahasan : Penelitian ini bertujuan membuat sistem monitoring performa aplikasi menggunakan OpenTelemetry, Prometheus, Jaeger, dan Grafana, serta notifikasi melalui Slack untuk deteksi anomali. Hasil Penelitian : Sistem monitoring dapat mengumpulkan data secara <i>real-time</i> dengan rata-rata delay 13,8 detik. Sistem juga berhasil mengirimkan peringatan ke Slack ketika terjadi anomali. Namun, implementasi OpenTelemetry menurunkan throughput aplikasi rata-rata 23,32% dan meningkatkan request latency rata-rata 22,80%.
8	Favaz Mohammed Rawoof, Mohd Tajammul, Fakhrun Jamal (2022)	Judul : <i>On-Premise Server Monitoring with Prometheus and Telegram Bot</i> Tools : Prometheus, Telegram Bot, Python Agent, MongoDB Pokok Bahasan : Penelitian ini bertujuan membangun sistem monitoring server <i>on-premise</i> yang dapat diakses secara jarak jauh melalui Telegram Bot. Prometheus digunakan untuk mengumpulkan metrik server, sedangkan Python agent mengelola pengiriman data ke database dan mengintegrasikannya dengan Telegram untuk keperluan notifikasi. Hasil Penelitian : Sistem berhasil memungkinkan administrator mengakses metrik server dari mana saja melalui Telegram, tanpa perlu berada di lokasi fisik server. Data dikumpulkan oleh Prometheus, disimpan di MongoDB, lalu dikirim ke Telegram Bot sesuai permintaan. Sistem ini menawarkan solusi praktis, ringan, dan user-friendly untuk monitoring jarak jauh server on-premise.
9	Renaldi Yulvianda, Muhammad Ismail (2023)	Judul : Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix dan Grafana Tools : Zabbix, Grafana, Telegram

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		Pokok Bahasan : Penelitian ini bertujuan untuk memantau sumber daya server (CPU, RAM, hard disk, jaringan) secara <i>real-time</i> menggunakan Zabbix untuk pengumpulan <i>metrics</i>, visualisasi di Grafana, dan pengiriman notifikasi melalui Telegram jika terdeteksi anomali. Hasil Penelitian : Sistem berhasil mengirimkan notifikasi <i>real-time</i> ke Telegram saat layanan server bermasalah dan membantu administrator dalam memantau performa server secara lebih efisien.
10	G. P. Kalyan Chakravarthi, M. Surya Prakash, C. H. Mohan, I. Chandu, R. Meena (2023)	Judul : <i>Automated Monitoring and Analysis of Cron Jobs on Virtual Machines with Grafana Visualization</i> Tools : Grafana, Google Cloud Platform, JSON, Bitbucket, SSH, Python, Bash, Infinity Plugin Pokok Bahasan : Penelitian ini membahas monitoring dan visualisasi cron jobs pada virtual machines di GCP menggunakan script Python & Bash untuk pengumpulan data, disimpan dalam JSON, dan divisualisasikan di Grafana dengan Infinity Plugin. Hasil Penelitian : Sistem berhasil mengumpulkan dan menampilkan data cron jobs dari beberapa VM secara terpusat, mengurangi proses manual dalam pemantauan tugas terjadwal.
11	Marchenko dan Shchemelinin (2023)	Judul : <i>Development of an Accessibility Testing System for the Virtual Machine Deployment Service in the Cloud</i> Tools : Prometheus, Selenium, Kubernetes, Python, Grafana, Prometheus Pushgateway, Alertmanager, VictorOps, NumPy, SciPy Pokok Bahasan : Penelitian ini mengembangkan sistem untuk menguji ketersediaan layanan deployment virtual machine di lingkungan cloud internal milik Intel.

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		Sistem ini mengintegrasikan alat <i>open-source</i> seperti Prometheus, Selenium, dan Kubernetes. Tujuannya adalah memastikan keandalan dan performa layanan cloud melalui pemantauan otomatis berbasis skenario pengguna. Hasil Penelitian: Sistem ini dapat membuat dan menghapus VM, mencatat metrik seperti waktu login, waktu peluncuran VM, dan waktu eksekusi total. Berdasarkan data Prometheus, waktu peluncuran VM dianalisis menggunakan metode statistik untuk menetapkan ambang batas performa. Sistem ini berhasil mendeteksi dan mencegah masalah performa lebih awal, meningkatkan pengalaman pengguna, dan mengurangi biaya serta waktu respons terhadap insiden pada layanan cloud.
12	Rizki Tian Darmawan (2023)	Judul: Implementasi Grafana untuk Auto Scaling pada NGINX Web-Server di Lingkungan Docker Container Tools: Docker, Grafana, Prometheus, cAdvisor, Webhook, Shell Script Pokok Bahasan: Penelitian ini mengimplementasikan sistem auto-scaling berbasis container NGINX dengan pemantauan metrik CPU menggunakan Prometheus dan visualisasi serta alerting oleh Grafana. Saat batas CPU tertentu tercapai, Grafana men-trigger eksekusi script shell untuk scale in/out container secara otomatis. Hasil Penelitian: Sistem berhasil men-trigger auto-scaling: scale-out saat CPU > 70% dan scale in saat CPU <20%, sesuai batas konfigurasi. Grafana digunakan untuk memantau metrik dan mengirim alert berbasis kondisi anomali CPU. Skalabilitas dilakukan secara horizontal dengan menambah/menghapus replika container.

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
13	Kadek Agus Permana Putra, Gede Sukadarmika, Dewa Made Wiharta (2024)	Judul : Model Utilisasi Dan Visualisasi Resource Menggunakan Prometheus Dan Grafana Untuk Pengelolaan Server Tools : Prometheus, Grafana, Ubuntu, VMware Pokok Bahasan : Penelitian ini membahas monitoring utilisasi server (CPU, RAM, storage) untuk pengelolaan infrastruktur TI di Universitas Udayana menggunakan Prometheus untuk pengumpulan <i>metrics</i> dan visualisasi di Grafana. Hasil Penelitian : Sistem berhasil menampilkan visualisasi resource server dalam dashboard Grafana, sehingga administrator jaringan dapat memantau kondisi server secara <i>real-time</i> tanpa perlu memeriksa langsung.
14	Pragathi B.C., Hrithik Maddirala, Sneha M. (2024)	Judul : <i>Implementing an Effective Infrastructure Monitoring Solution with Prometheus and Grafana.</i> Tools : Prometheus, Grafana, Node Exporter, Kubernetes, Helm, PromQL, Alertmanager, NetData (sebagai pembanding) Pokok Bahasan : Penelitian ini membahas penerapan sistem monitoring infrastruktur yang efektif dalam lingkungan Kubernetes menggunakan kombinasi Prometheus, Grafana, dan Node Exporter. Fokus utama adalah pada pemantauan performa sistem secara <i>real-time</i> dan otomatisasi deteksi anomali. Hasil Penelitian : Hasil menunjukkan sistem ini mampu memberikan visibilitas menyeluruh terhadap performa cluster Kubernetes, seperti pemantauan CPU, memori, disk I/O, dan aktivitas jaringan. Dengan integrasi alert dan dashboard di Grafana, administrator dapat merespons masalah dengan cepat. Pengujian juga menunjukkan bahwa sistem ini efektif untuk menganalisis tren historis dan membantu perencanaan kapasitas. Sistem berhasil mendeteksi beban tinggi, memory leak, dan

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		konfigurasi disk yang kurang optimal melalui metrik dan alert otomatis.
15	Dhane A., Joshi A., Borgaonkar C., Deshpande M., Patil P. (2024)	Judul : <i>Application Monitoring using Prometheus and Grafana</i> Tools : Prometheus, Grafana, Node Exporter, Alertmanager, PromQL, prom-client (Node.js), SMTP, MySQL. Pokok Bahasan : Penelitian ini membahas penerapan sistem pemantauan aplikasi menggunakan Prometheus dan Grafana, berfokus pada bagaimana kedua tools open-source ini digunakan untuk mengukur performa, mendeteksi anomali, dan mengelola infrastruktur aplikasi modern berbasis Node.js dan MySQL. Hasil Penelitian : Sistem berhasil mengumpulkan berbagai metrik seperti total permintaan pengguna, tingkat keberhasilan pembayaran, performa MySQL, durasi permintaan HTTP, serta event loop lag pada Node.js. Visualisasi data di Grafana dan sistem alerting yang dibangun dengan Alertmanager membantu mendeteksi permasalahan lebih cepat dan mengoptimalkan alokasi sumber daya. Studi kasus perusahaan seperti Amadeus, DigitalOcean, dan JP Morgan Chase menunjukkan efektivitas penerapan Prometheus dan Grafana dalam lingkungan produksi nyata.
16	Akhyar Lubis, Muhammad Amal Giffari, Sri Wahyuni (2024)	Judul : <i>Enhancing Network Performance Visibility with Grafana and Prometheus: A Case Study at P.T. Nata Digital Solution</i> Tools : Prometheus, Grafana, Node Exporter, SNMP Exporter, Blackbox Exporter, Telegram Pokok Bahasan : Penelitian ini bertujuan membangun sistem monitoring jaringan berbasis Prometheus dan Grafana untuk meningkatkan visibilitas performa jaringan serta memberikan notifikasi <i>real-time</i>

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		melalui Telegram dalam mendeteksi anomali jaringan. Hasil Penelitian : Sistem berhasil memantau metrik penting seperti beban CPU, penggunaan memori, disk, bandwidth, dan <i>latency</i> secara <i>real-time</i>. Alert dikirimkan ke Telegram saat terjadi anomali seperti lonjakan CPU dan <i>packet loss</i>. Sistem ini meningkatkan efisiensi pemantauan jaringan dan mempercepat respon tim I.T. terhadap gangguan.
17	Norma Ningsih, Karimatun Nisa, Endryco Farel R., Bintang D. R., Afifah D.R. (2024)	Judul: Optimalisasi Monitoring Tag dengan AWS Cloud: Studi Kasus Aplikasi Tag Tracker pada PT. XYZ. Tools: AWS Lambda, AWS SQS, AWS EventBridge, AWS DynamoDB, AWS CloudFormation, HTML/CSS/JS, Golang Pokok Bahasan: Penelitian ini mengembangkan sistem monitoring tak views menggunakan arsitektur serverless AWS untuk perusahaan. Terdiri dari dua layanan: dashboard untuk visualisasi statistik mingguan dan aggregator untuk menghitung total views tiap tag. Hasil Penelitian: Sistem menunjukkan respons cepat (1-2 menit) untuk update data mingguan ke DynamoDB dan dashboard. Pengujian blackbox menunjukkan keberhasilan 100%. Sistem efektif digunakan tim internal untuk analisis performa tag dalam menentukan strategi bisnis.
18	Banu Rasyidi, Firman Pratama (2024)	Judul: Sistem Monitoring Server di PT. XYZ Media Indonesia Berbasis Grafana dan Prometheus Tools: Prometheus, Grafana, Node Exporter, PromQL, Telegram, Ubuntu Server, Xshell Pokok Bahasan: Penelitian ini merancang sistem monitoring server menggunakan Prometheus dan Grafana untuk memantau utilisasi CPU, memori, storage dan jaringan. Sistem ini terintegrasi dengan

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		Telegram untuk mengirimkan notifikasi otomatis saat terjadi malfungsi server. Hasil Penelitian: Sistem berhasil memantau server secara <i>real-time</i> dan mengirimkan notifikasi alert ke Telegram. Pengujian blackbox menunjukkan semua fitur berfungsi, termasuk registrasi, login, visualisasi dashboard, dan pengiriman alert saat threshold terlampaui.
19	Fajar Kurnia Ramadhan, Garno, Arip Solehudin (2024)	Judul: Comparative Study of Web Server Performance Testing with and without Docker Based on Virtual Machine Tools: Prometheus, Grafana, Node Exporter, Apache JMeter, Docker, VMWare ESXi, Joomla CMS. Pokok Bahasan: Penelitian membandingkan performa dua web server yang berjalan di atas VM dengan dan tanpa Docker. Sistem dimonitor menggunakan Prometheus & Grafana untuk mengukur penggunaan CPU dan RAM saat dilakukan load test menggunakan JMeter. Hasil Penelitian: Hasil menunjukkan VM tanpa Docker lebih efisien dalam menangani HTTP request dan penggunaan CPU. VM Dockerized gagal di atas 2790 user. Data monitoring berhasil dikumpulkan via Grafana dengan Node Exporter Full, tanpa implementasi alert.
20	Taiwo Joseph Akinbolaji, Olamide Durojaye, dan Oluwaseun Adeniyi (2024)	Judul: Proactive Monitoring and Security in Cloud Infrastructure Leveraging Tools like Prometheus, Grafana, and Loki Tools: Prometheus, Grafana, Loki, Alertmanager dan Docker Compose Pokok Bahasan: Penelitian ini membahas penerapan sistem monitoring keamanan cloud secara proaktif dengan memanfaatkan metrik dan log secara real-time. Prometheus digunakan untuk mengumpulkan data performa server, Grafana

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		untuk menampilkan visualisasi dan alerting, serta Loki untuk mengelola log sistem. Sistem di-deploy menggunakan Docker Compose agar mudah direplikasi di berbagai lingkungan. Hasil Penelitian : Hasil implementasi menunjukkan sistem mampu memantau performa server secara real-time serta mengirimkan notifikasi ketika anomali terdeteksi. Integrasi dengan Loki memungkinkan identifikasi cepat terhadap penyebab masalah keamanan.
21	Fotios Voutsas, Apostolos Leivadeas, dan Mohamed Cheriet (2024)	Judul: Mitigating Alert Fatigue in Cloud Monitoring Systems: A Machine Learning Perspective Tools: Netdata Monitoring Platform, dua dataset operasional (ND2022 & ND2023), dan algoritma Machine Learning. Pokok Bahasan: Penelitian ini membahas masalah alert fatigue dalam sistem monitoring cloud, yaitu kondisi ketika jumlah notifikasi terlalu banyak sehingga operator mengabaikan alert penting. Penulis menggunakan data operasional dari Netdata dan menerapkan beberapa algoritma machine learning untuk mengklasifikasi alert yang relevan, sehingga hanya peringatan prioritas tinggi yang diteruskan ke operator. Hasil Penelitian: Sistem machine learning yang dikembangkan mampu menyaring alert tidak penting sehingga jumlah notifikasi yang diterima operator berkurang signifikan. Random forest menjadi model paling konsisten dalam mengidentifikasi alert prioritas tinggi dari dataset operasional. Berdasarkan evaluasi pakar, sistem ini membantu meningkatkan fokus operator, mempercepat respons insiden, dan terbukti efektif dalam memitigasi alert fatigue pada lingkungan monitoring cloud.

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
22	Yash Jani (2024)	Judul: Unified Monitoring for Microservices: Implementing Prometheus and Grafana for Scalable Solutions Tools: Prometheus, Grafana Pokok Bahasan: Penelitian ini membahas penerapan unified monitoring pada arsitektur microservices dengan mengintegrasikan Prometheus sebagai sistem pengumpul metrik dan Grafana sebagai platform visualisasi. Fokus penelitian ini meliputi konfigurasi Prometheus untuk memantau metrik utama layanan, penggunaan service discovery untuk otomatisasi penambahan instance baru, serta penerapan alert rules untuk deteksi dini gangguan. Penelitian juga menyoroti penggunaan dashboard dinamis di Grafana untuk memudahkan pemantauan banyak layanan dalam skala besar. Hasil Penelitian: Hasil implementasi menunjukkan bahwa integrasi Prometheus dengan Grafana mampu memberikan monitoring real-time yang stabil pada lingkungan microservices. Sistem dapat mendeteksi gangguan lebih cepat melalui mekanisme alerting dan tetap mempertahankan performa monitoring meskipun jumlah layanan meningkat. Pendekatan unified monitoring ini terbukti meningkatkan efisiensi operasional dan keandalan dalam pengelolaan microservices.
23	Mohammed Daffala Elradi (2025)	Judul: Prometheus & Grafana: A Metrics-focused Monitoring Stack Tools: Prometheus, Grafana, Node Exporter, dan Docker Pokok Bahasan: Penelitian ini membahas penerapan open-source monitoring stack berbasis Prometheus dan Grafana untuk memvisualisasikan metrik dari server Linux secara real-time. Docker digunakan untuk mempercepat proses deployment dan memastikan layanan berjalan konsisten.

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		Hasil Penelitian: Sitem berhasil menampilkan data metrik seperti CPU load, penggunaan RAM, aktivitas disk, dan trafik jaringan secara real-time melalui dashboard Node Exporter di Grafana. Hasil visualisasi memberikan pandangan menyeluruh terhadap performa sever serta membantu identifikasi bottleneck dan optimasi sumber daya.
24	Raden Mohamad Rishwan, E Haodudin Nurkifli, Arip Solehudin (2025)	Judul: Analisis Perbandingan Performa Server Websocket dengan Menggunakan Payload JSON, Binary Serialization, dan Protobuf dengan Metode Load Testing. Tools: Golang, Flutter, Prometheus, Grafana, Node Exporter, K6 Pokok Bahasan: Pengujian performa server WebSocket dengan berbagai jenis payload (JSON, Binary, Protobuf). Sistem diuji menggunakan load testing K6 dan dimonitor dengan Prometheus dan Grafana. Hasil Penelitian: Protobuf dan Binary Serialization lebih efisien dibandingkan JSON (CPU & memori). Monitoring berjalan dengan Grafana, namun fitur alerting tidak aktif.
25	Rim Doukha dan Abderrahmane Ez-zahout (2025)	Judul: Enhanced Virtual Machine Resource Optimization in Cloud Computing Using Real-Time Monitoring and Predictive Modeling Tools: Proxmox VE, Prometheus, Grafana, Amazon S3, Random Forest Regression, dan LSTM Neural Network. Pokok Bahasan: Penelitian ini membahas optimasi sumber daya virtual machine di lingkungan cloud computing melalui sistem monitoring real-time yang menganalisis metrik performa seperti CPU, memori, dan jaringan. Penelitian ini menggunakan pendekatan prediksi beban untuk mengidentifikasi potensi bottleneck dan melakukan penyesuaian alokasi resource secara

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
		dinamis agar performa VM tetap stabil dan efisien. Hasil Penelitian: Hasil menunjukkan bahwa integrasi monitoring real-time dan prediksi beban mampu meningkatkan efisiensi alokasi sumber daya VM serta mengurangi terjadinya kemacetan resource. Sistem yang diusulkan dapat mendeteksi kondisi tidak optimal lebih cepat dan memberikan rekomendasi penyesuaian resource yang meningkatkan stabilitas performa VM secara signifikan dibandingkan metode statis konvensional.

Berbagai studi telah membahas implementasi sistem monitoring menggunakan Prometheus dan Grafana, termasuk integrasi notifikasi otomatis melalui platform seperti Telegram. Namun, sebagian besar penelitian tersebut berfokus pada server fisik, jaringan atau lingkungan container, sementara kajian yang secara khusus membahas monitoring VM di lingkungan cloud masih relatif terbatas. Selain itu, efektivitas sistem dalam mendeteksi lonjakan beban CPU secara *real-time* serta mekanisme pengiriman peringatan otomatis umumnya belum menjadi fokus utama penelitian sebelumnya.

Kondisi tersebut menunjukkan adanya celah penelitian terkait pengembangan dan pengujian sistem monitoring CPU pada VM di lingkungan cloud yang mampu merespons lonjakan beban secara cepat dan akurat. Oleh karena itu, penelitian ini difokuskan untuk mengisi celah tersebut dengan mengembangkan sistem monitoring CPU Virtual Machine pada lingkungan cloud yang terintegrasi dengan notifikasi otomatis melalui Telegram, serta melakukan pengujian kinerja sistem saat terjadi lonjakan beban menggunakan Apache JMeter. Fokus ini diharapkan dapat memberikan kontribusi tambahan terhadap kajian sistem monitoring pada infrastruktur cloud.

B. Landasan Teori

1. Cloud Computing

Cloud computing merupakan teknologi jaringan berbasis internet yang menyediakan layanan sumber daya komputasi secara *online*, termasuk server, penyimpanan, database dan perangkat lunak[6]. Teknologi ini memungkinkan pengguna untuk mengakses layanan komputasi sesuai kebutuhan tanpa harus memiliki infrastruktur fisik sendiri. Cloud computing merupakan model

komputasi dengan tingkat skalabilitas yang sangat tinggi, yang memungkinkan pengguna untuk mengakses dan menggunakan sumber daya seperti jaringan, server, penyimpanan, aplikasi, dan layanan lainnya yang ada dalam lingkungan cloud yang dapat dibagi dan digunakan bersama[7].

2. Virtualisasi

Virtualisasi adalah metode untuk menyembunyikan sifat fisik sumber daya komputer atau bahkan menciptakan perangkat yang secara fisik tidak ada menjadi tampak seolah ada[8]. Melalui virtualisasi, sebuah perangkat keras yang dapat menjalankan beberapa sistem operasi secara bersamaan melalui lingkungan terisolasi disebut sebagai mesin virtual. Virtualisasi berperan dalam mengoptimalkan penggunaan sumber daya, meningkatkan kapasitas infrastruktur teknologi informasi serta mendukung proses pengembangan dan pengujian tanpa mengganggu sistem operasi yang ada. Virtual Machine adalah mesin virtual yang menjalankan OS dan aplikasi seperti komputer fisik[9].

3. Google Cloud Platform (GCP)

Google Cloud Platform adalah kumpulan layanan cloud computing yang dikelola oleh Google untuk memenuhi kebutuhan komputasi secara daring[10]. Google Cloud Platform (GCP) memiliki banyak layanan produk yang fungsinya sebagai cloud computing dan dapat digunakan untuk merancang infrastruktur server. Setiap layanan memiliki fungsi dan keunggulan yang berbeda dengan fitur yang ditawarkan[11]. Salah satu layanan utamanya adalah Compute Engine, yang memungkinkan pengguna untuk membuat dan mengelola instance dengan berbagai konfigurasi. Compute Engine menjadi inti dalam layanan komputasi GCP karena kemampuannya dalam menjalankan berbagai aplikasi di lingkungan cloud[12]. Penelitian ini memanfaatkan GCP sebagai platform implementasi, dengan layanan Compute Engine digunakan untuk membuat dan menjalankan instance yang berfungsi sebagai server yang monitoring dan target monitoring.

4. Monitoring

Monitoring adalah proses pengumpulan dan analisis data secara rutin untuk mengetahui status dan performa suatu sistem, serta mendeteksi masalah lebih awal agar dapat segera ditangani[13]. Dalam konteks teknologi informasi, monitoring mencakup pengamatan dan analisis terhadap kondisi sistem melalui data performa yang dikumpulkan. Proses ini mencakup pengumpulan, agregasi, dan analisis metrik yang berkaitan dengan performa dan kesehatan sistem. Sistem monitoring memungkinkan visualisasi data, penyimpanan log, serta pemantauan metrik secara *real-time*. Monitoring ini berperan penting dalam pengelolaan sumber daya sistem, seperti CPU, memori, penyimpanan, dan bandwidth jaringan, termasuk dalam proses alokasi, penjadwalan, dan migrasi sumber daya[14].

5. Prometheus

Prometheus merupakan alat *open source* yang umum dirancang untuk pencatatan dan penyimpanan data dalam format *time-series*[5]. Alat ini bekerja dengan mekanisme *pull-based*, di mana Prometheus secara aktif menarik data metrik dari berbagai target atau *endpoint* monitoring sesuai dengan interval yang telah ditentukan. Metrik yang dikumpulkan dapat berasal dari berbagai sumber, seperti aplikasi, sistem operasi, atau layanan lainnya yang menyediakan data performa dan kesehatan sistem secara *real-time*. Dalam ekosistem Prometheus, exporter merupakan komponen penting yang berfungsi menyediakan data metrik dari sumber tertentu agar dapat diambil dan diproses oleh Prometheus. Salah satu exporter yang populer adalah Node Exporter, yang khusus mengumpulkan metrik terkait performa sistem operasi pada mesin fisik maupun virtual.

6. Grafana

Grafana merupakan platform *open source* yang digunakan untuk menampilkan data metrik dari berbagai sumber dalam bentuk visualisasi[15]. Platform ini mendukung integrasi dengan berbagai basis data, serta memungkinkan pengguna membuat *dashboard* yang dapat dimodifikasi. Visualisasi data dalam Grafana digunakan untuk memahami data metrik yang dikumpulkan. Selain visualisasi data, Grafana juga menyediakan fitur notifikasi yang dapat mengirimkan peringatan secara otomatis jika metrik mencapai ambang batas yang telah ditentukan. Notifikasi ini bisa dikonfigurasi untuk dikirim melalui berbagai media komunikasi seperti email, Slack, PagerDuty, dan Telegram.

7. Telegram

Telegram adalah platform komunikasi digital yang menyediakan layanan pesan teks, panggilan suara, video, serta mendukung penggunaan bot melalui Bot API[16]. Telegram Bot memungkinkan integrasi dengan sistem eksternal, seperti server, untuk mengirimkan notifikasi secara *real-time* berdasarkan peristiwa atau ambang batas metrik yang telah ditentukan. Telegram merupakan aplikasi layanan pengiriman pesan instan multiplatform berbasis cloud yang fokus pada kecepatan dan keamanan yang bersifat *open source* nirlaba[17]. Telegram juga memiliki fitur Bot yang dapat diotomatisasi untuk berbagai keperluan, termasuk mengirim notifikasi secara otomatis dalam sistem monitoring[18]. Telegram Bot dapat mengirimkan pesan secara *real-time* melalui platform mobile, desktop, maupun web. Bot ini memiliki fungsi khusus dan berjalan otomatis sesuai instruksi atau permintaan yang telah diatur.

8. Stress Testing

Stress testing adalah metode pengujian yang digunakan untuk mengevaluasi batas kemampuan suatu sistem saat menghadapi beban yang ekstrem[19]. Pengujian ini bertujuan untuk melihat bagaimana sistem bereaksi terhadap tekanan tinggi dalam waktu tertentu, seperti lonjakan permintaan atau volume data. Dalam penelitian ini, *stress testing* dilakukan untuk menciptakan kondisi

abnormal berupa lonjakan penggunaan CPU. Tujuan dari langkah ini adalah untuk memverifikasi apakah sistem monitoring yang dirancang mampu mendeteksi lonjakan tersebut dan mengirimkan notifikasi secara *real-time* melalui aplikasi Telegram.

9. Apache JMeter

Apache JMeter adalah aplikasi *open source* berbasis Java yang dikembangkan oleh Apache Software Foundation untuk melakukan pengujian *performance testing*, *load testing*, dan *stress testing* terhadap berbagai jenis layanan, khususnya aplikasi berbasis web. Aplikasi ini mampu mensimulasikan ribuan permintaan pengguna secara paralel, sehingga efektif dalam mengevaluasi respons, *throughput*, serta kestabilan sistem di bawah tekanan tinggi[20]. Pengujian dilakukan dengan memantau metrik seperti waktu respons, jumlah permintaan yang dapat diproses dalam interval waktu tertentu, serta deteksi *bottleneck* ketika sumber daya sistem mencapai batas.