PENGEMBANGAN *SYSTEM MONITORING* DAN VISUALISASI DATA PENGGUNAAN SUMBER DAYA PADA SERVER MENGGUNAKAN *PROMETHEUS* DAN GRAFANA PADA DATABASE TERINTEGRASI DI PT. DIRGANTARA

1st Bagus Bintang Fauzi
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
justreindstr@student.telkomuniversity.ac.id

2st Muhammad Iqbal
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
miqbal.staff.telkomuniversity.ac.id

3st Hendar Rusman
PT. Dirgantara Indonesia(persero) Tbk
Bandung, Indonesia
rusman@indonesian-aerospace.com

***Abstrak* - Dalam Pengembangan *system* infrastruktur jaringan dan server, *Engineer Development* membutuhkan sebuah *system* untuk menjaga kualitas dan kestabilan *performa* server secara berkelanjutan, dalam hal ini *system monitoring dan alerting* sangat diperlukan untuk mendeteksi dan merespon gangguan pada server dengan cepat. Proses pengembangan akan meliputi beberapa tahapan instalasi dan konfigurasi yaitu *Prometheus*, Node Exporter, Grafana sebagai dashboard visualisasi data penggunaan CPU, disk, memory dan network, lalu terakhir adalah mengintegrasikan *Prometheus* dengan *system* alertingnya. Tujuan dari pembuatan *system monitoring and alerting* adalah memudahkan tim IT di PT. Dirgantara Indonesia untuk memantau performa server sebanyak 128 Vmware dalam satu dashboard yang dijalankan secara real-time dan menganalisis beban kerja *system* agar dapat meningkatkan efisiensi dan mencegah potensi gangguan pada server sejak dini. Hasil akhir dari pengembangan ini adalah dapat merancang *system* yang mampu memberikan informasi data *source* pengunaan server secara akurat dan *real-time*, serta memberikan notifikasi pada saat terjadi masalah pada server. *system* ini diharapkan dapat mendukung pengelolaan infrastruktur server yang lebih efisien di lingkungan PT. Dirgantara Indonesia.**

Kata Kunci* - Dashboard, *Monitoring, Alerting, Prometheus, Node Exporter, Grafana.

I. PENDAHULUAN

PT. Dirgantara Indonesia adalah perusahaan manufaktur pesawat terbang terbesar di Asia Tenggara dan bagian dari holding BUMN *DEFEND.ID*. Untuk mendukung operasional sistem informasi dan infrastruktur *server*, perusahaan memiliki tim IT, termasuk divisi *Engineering Development* yang bertugas menjaga kestabilan *server* dan mencegah anomali.

Saat ini, sistem *monitoring server* dan jaringan masih menggunakan *software* terpisah, seperti *HP Intelligent Management Center (IMC)* untuk jaringan, serta *VCenter* dan *Netapp Management* untuk penggunaan sumber daya di *VMware*. Sistem ini kurang efisien dan cukup mahal, sehingga dibutuhkan solusi *monitoring* baru yang lebih terintegrasi, akurat, dan mampu menampilkan data secara *real-time* dalam satu *dashboard*. Selain itu, sistem juga perlu memberikan notifikasi jika terjadi gangguan. *Server* berperan penting dalam menyediakan akses data, dan pemantauan manual sering kesulitan mendeteksi masalah dengan cepat [1][2].

Untuk mengatasi kendala tersebut, dikembangkan sistem *monitoring* dan visualisasi yang efisien dan terintegrasi menggunakan *Prometheus*, *Grafana*, *Node Exporter*, dan *Telegram*. *Prometheus* mengumpulkan data dari *Node Exporter* dan menyimpannya dalam format *time-series*, sedangkan *Grafana* menampilkan visualisasi data tersebut. *Telegram* digunakan untuk mengirim notifikasi otomatis 30–60 detik setelah anomali terdeteksi, tergantung pada jaringan. Sistem ini terbukti efektif dalam mendeteksi dan merespons gangguan secara cepat [3], dan *Grafana* mendukung integrasi *plugin* dengan berbagai *basis data* [4].

Dengan keempat *tools* tersebut, tim IT dapat memantau penggunaan CPU, *disk*, *memory*, dan *network* secara *real-time*, serta mendapatkan notifikasi otomatis jika terjadi gangguan. Sistem juga memungkinkan analisis historis untuk mendukung pengambilan keputusan teknis yang lebih baik. Pengembangan ini diharapkan mampu menekan biaya, mengurangi ketergantungan pada *platform* berbayar, serta meningkatkan efisiensi dan keandalan infrastruktur *server* di PT. Dirgantara Indonesia.

II. TINJAUAN PUSTAKA

A. Tinjauan Umum/ Deksripsi dan Skenario dan Perancangan Sistem

Seiring meningkatnya ketergantungan perusahaan pada *sistem informasi* dan layanan digital, kebutuhan akan *monitoring* dan *alerting* performa server yang andal dan terintegrasi menjadi krusial. *Monitoring* server adalah proses pengawasan komponen dan kinerja sistem, seperti penggunaan CPU, *memory* (RAM), aktivitas aplikasi (*disk*), dan *bandwidth* jaringan (*network*). Sistem ini memberikan *alert* saat terjadi masalah pada protokol, serta memberi informasi spesifik secara otomatis kepada administrator jaringan [5].

Komponen penting dari *monitoring* adalah *alerting*, yaitu mekanisme peringatan otomatis ketika parameter melebihi *threshold* atau terjadi kondisi abnormal. Hal ini memungkinkan tim IT untuk merespons sebelum gangguan meluas.

Integrasi *monitoring* dan *alerting* bertujuan menciptakan pengawasan proaktif. Dalam implementasinya, digunakan *tools* seperti *Prometheus Alert Manager* yang mengelola aturan *alert* dan mengirim notifikasi melalui *email*, *Telegram*,

atau sistem internal.

Salah satu teknologi yang umum digunakan adalah *Prometheus*, *tool open-source* untuk *monitoring* dan *alerting* yang dikembangkan oleh SoundCloud. *Prometheus* menggunakan metode *time-series database* untuk menyimpan *metrics* dalam interval waktu tertentu dan mendukung bahasa kueri *PromQL*. Ia bekerja berdasarkan *pull model*, mengumpulkan data dari *target* melalui *HTTP endpoint*.

Untuk visualisasi, digunakan *Grafana*, yang menyajikan data dari *Prometheus* dalam bentuk *dashboard* interaktif. Kombinasi keduanya banyak diadopsi industri karena mampu menyajikan data informatif secara real-time maupun historis.

Dalam penelitian ini, *monitoring* difokuskan pada empat parameter utama:

- Penggunaan CPU (beban kerja prosesor)
- Penggunaan *memory* (efisiensi RAM)
- Konsumsi *bandwidth* jaringan (aktivitas masuk dan keluar)
- Aktivitas aplikasi (*disk* dan sumber daya)

Data diperoleh dari *Node Exporter*, agen *Prometheus* yang berjalan di server target dan mengekspor metrik sistem operasi. Data disimpan oleh *Prometheus* dan divisualisasikan melalui *Grafana*. Untuk analisis lebih lanjut, data historis juga dicatat dalam *database* seperti *InfluxDB*, termasuk untuk keperluan *forecasting*.

Dengan pendekatan ini, sistem tidak hanya memantau kondisi saat ini, tetapi juga memungkinkan analisis jangka panjang, evaluasi beban kerja, dan prediksi kebutuhan sumber daya di lingkungan server PT Dirgantara Indonesia.

B. *Tinjauan Khusus 1 (Konsep Monitoring System)*

Salah satu komponen penting dalam *system monitoring* adalah *visualisasi data* yang informatif dan *real-time*. *Visualisasi* ini bertujuan untuk mempermudah pemahaman terhadap kondisi *system* melalui representasi grafis dari data yang dikumpulkan. Dalam Karya Ilmiah ini, digunakan *Grafana* sebagai *platform visualisasi* karena kemampuannya yang fleksibel, mendukung berbagai jenis *datasource*, dan tampilannya yang interaktif serta mudah di *customize*.

C. *Tinjauan Khusus 2 (Konsep Alerting)*

Alerting merupakan komponen penting dalam sistem monitoring yang memungkinkan pengiriman notifikasi otomatis saat suatu metrik melebihi ambang batas tertentu. Dalam sistem ini, Prometheus bekerja sama dengan Alertmanager untuk menangani notifikasi tersebut. Salah satu media yang digunakan dalam penelitian ini adalah Telegram, karena sifatnya yang cepat, mudah diakses, dan mendukung bot otomatis [3]. Dengan mengintegrasikan Alertmanager dan Telegram Bot, sistem dapat mengirim pesan peringatan secara real-time kepada administrator ketika terjadi anomali seperti penggunaan CPU yang tinggi, memori hampir penuh, atau aplikasi yang tidak merespons. Hal ini memungkinkan tim IT di PT. Dirgantara Indonesia untuk melakukan tindakan mitigasi dengan lebih cepat dan efisien.

D. *Tinjauan Khusus 3 (Grafana)*

Grafana merupakan sebuah *platform open-source* terkemuka yang dirancang khusus untuk *visualisasi* dan pemantauan data berbasis *web* [6]. *Grafana* merupakan *platform open-source* yang digunakan untuk mem*visualisasikan* dan memantau data berbasis *web*. *Grafana* mempunyai dukungan untuk menggunakan berbagai macam tipe *datasource* data seperti *Graphite*, *Prometheus*, *Elasticsearch*, *OpenTSDB*, dan *InfluxDB*. Seperti halnya *Prometheus*, *Grafana* juga merupakan salah satu proyek berbasis *sumber terbuka* yang sangat populer saat ini. *Prometheus server* akan mengambil data *target* pada *interval waktu* yang telah ditentukan pada konfigurasi dari *target* tertentu dan menyimpannya dalam bentuk basis data *time series*. *Kode sumber* dari *perangkat lunak Prometheus* bisa diakses dan diunduh di halaman *GitHub*-nya [7].

Selain *visualisasi data real-time*, *Grafana* juga mendukung analisis historis yang memungkinkan *administrator* melacak tren penggunaan sumber daya dalam periode tertentu. Fitur ini sangat membantu dalam mendeteksi pola penggunaan yang dapat dijadikan dasar dalam optimasi kinerja *server* [13]. Dengan demikian, *Grafana* bukan hanya alat *visualisasi*, tetapi juga alat analisis yang membantu dalam pengambilan keputusan berbasis data.

E. *Tinjauan Khusus 4 (Prometheus)*

Prometheus adalah *toolkit* pemantauan dan peringatan *system sumber terbuka* yang awalnya dibangun di *SoundCloud*. Sejak awal pembuatannya pada tahun 2012, banyak perusahaan dan organisasi yang mengadopsi *Prometheus*, dan proyek ini memiliki komunitas pengembang dan pengguna yang sangat aktif [7]. *Prometheus* mengumpulkan *metrik* dari data *resource*, baik secara langsung atau melalui *gateway push*. *Prometheus* menggunakan *metrik* untuk pekerjaan yang berjangka pendek. *Metrik* mengumpulkan data yang diambil dari *exporter* yang telah di-*install* dan dapat digunakan untuk memberi peringatan [9]. File konfigurasi utama yang digunakan oleh *Prometheus* untuk menentukan bagaimana *system* melakukan pengambilan (*scraping*) data dari berbagai *target monitoring* adalah *prometheus.yml.*

Dalam *Prometheus.yml*, pengguna dapat menentukan *job_name* yang merepresentasikan kelompok target tertentu, seperti *node-exporter*, kemudian menetapkan alamat *IP* atau *hostname* dari server yang akan dimonitor melalui bagian *targets*. Selain itu, konfigurasi ini juga dapat mencakup parameter tambahan seperti *label*, *timeouts*, dan *rule evaluation* untuk kebutuhan *alerting* (jika digunakan). *File* ini bersifat *fleksibel* dan mudah disesuaikan, sehingga sangat penting dalam *arsitektur Prometheus* untuk memastikan bahwa *metrik* yang relevan dapat dikumpulkan secara teratur dan akurat dari berbagai sumber data[8].

Contoh isi dari *file Prometheus.yml* sebagai konfigurasi penargetan *client*, sebagai berikut:

```
“global:

  scrape_interval: 15s  # Interval pengambilan data dari target

scrape_configs:

  - job_name: 'node_exporter'

    static_configs:

      - targets: ['localhost:9100'] # Ganti dengan IP VM atau server
```

F. Tinjauan Khusus 5 (PrompQL)

PromQL (*Prometheus Query Language*) adalah bahasa *query* yang digunakan untuk mengambil dan memproses data dari *time-series database Prometheus*. *PromQL* memungkinkan pengguna untuk mengeksekusi berbagai jenis *query*, baik itu untuk melihat nilai saat ini, menghitung rata-rata, tren, maupun untuk membuat kondisi *alert*. Contoh *query* sederhana yang dapat dilihat pada Tabel 2.1, sebagai berikut:

Tabel 2. 1 Query Sederhana

Nama Data	Query Dasar
CPU	node_cpu_seconds_total
Disk	node_filesistem_size_bytes
Memory	node_memory_MemTotal_bytes
Network	node_network_receive_bytes_total

Query di atas digunakan untuk menghitung laju penggunaan *CPU*, *Disk*, *Memory*, dan *Network*. *PromQL* memiliki fungsi-fungsi seperti *rate()*, *avg()*, *sum()*, dan *max()* yang sangat berguna dalam analisis performa *system*.

G. Tinjauan Khusus 5 (Node Exporter)

Node Exporter adalah komponen tambahan yang digunakan oleh *Prometheus*. Alat ini berfungsi untuk mengumpulkan data seperti penggunaan *CPU*, *memori*, *disk I/O*, *jaringan*, dan berbagai *metrik* lainnya yang berkaitan dengan kinerja infrastruktur [6]. Alat ini biasanya digunakan untuk mengumpulkan *metrik* seperti penggunaan *CPU*, *memory*, *I/O*, *network traffic* [12]. *Node Exporter* bekerja dengan mengekspos *metrik* melalui *HTTP endpoint*, biasanya di *port* 9100, yang kemudian di-*scrape* oleh *Prometheus*.

Dalam implementasi ini, *Node Exporter* di*instal* pada *server* yang ingin dimonitor. Setelah aktif, *Prometheus* dikonfigurasi untuk mengambil data dari alamat *IP* dan *port Node Exporter* melalui file *prometheus.yml*. Data yang dikumpulkan oleh *Node Exporter* bersifat *granular* dan sangat berguna untuk analisis performa *system* secara *real-time* melalui *dashboard Grafana*. Contoh *endpoint metrik Node Exporter*: http://localhost:9100.

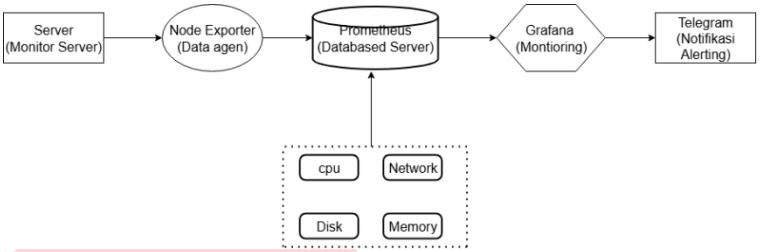
H. Tinjauan Khusus 5 (DDoS)

Distributed Denial of Service (*DDoS*) adalah salah satu bentuk serangan siber yang bertujuan untuk melumpuhkan layanan atau *system* dengan cara membanjiri server, jaringan, atau aplikasi dengan lalu lintas (*traffic*) yang sangat tinggi. Serangan ini dilakukan dari banyak sumber (terdistribusi), sehingga lebih sulit untuk ditangkal dibandingkan dengan serangan *DoS* (Denial of Service) biasa yang hanya berasal dari satu sumber. Serangan Distributed Denial of Service (*DDoS*) merupakan salah satu jenis serangan paling berbahaya yang dapat membuat layanan daring tidak dapat diakses oleh pengguna yang sah [15].

Dalam konteks *system monitoring* dan *alerting*, simulasi serangan *DDoS* digunakan sebagai metode pengujian untuk mengukur kemampuan *system* dalam mendeteksi lonjakan *traffic* atau penggunaan sumber daya secara abnormal [10]. Hal ini penting sebagai bentuk evaluasi performa dan efektivitas *system* dalam menghadapi potensi ancaman nyata di lingkungan server perusahaan.

III. PEMODELAN DAN PERANCANGAN

A. Arsitektur Sistem



Gambar 3. 1 Blok System Monitoring

Arsitektur *system monitoring* ini dirancang dengan pendekatan *client-server* yang terintegrasi menggunakan *Prometheus* sebagai *database time-series* dan *Grafana* sebagai *platform* visualisasi. *System* ini bertujuan untuk memantau performa server secara *real-time* terhadap *metrik* penting seperti CPU, *Memory*, *Disk*, dan *Network usage* agar memudahkan *admin* dalam pengelolaan infrastruktur server. *System* ini terdiri dari empat komponen utama, yaitu:

1. ***Server (Monitoring Host)***
   Server adalah perangkat utama yang dipantau dalam *system* ini. Di dalam *server* ini berjalan berbagai proses *system* yang mengonsumsi sumber daya seperti CPU, memori, *disk*, dan jaringan. Performa dari *server* ini menjadi objek utama dalam proses monitoring. *Server* ini juga menjadi tempat di mana *Node Exporter* diinstal untuk mengambil data *metrik system*.
2. ***Node Exporter***
   Node Exporter merupakan agen ringan yang berjalan di dalam *server* dan bertugas untuk mengumpulkan *metrik* performa seperti penggunaan CPU, memori, *disk I/O*, dan lalu lintas jaringan. *Node Exporter* menyediakan data dalam format yang kompatibel dengan *Prometheus* melalui *HTTP endpoint*. *Node Exporter* bertindak sebagai jembatan antara *system operasi server* dengan *Prometheus*, sehingga memungkinkan *Prometheus* untuk menarik (*scrape*) data secara periodik.
3. ***Prometheus (Database Server)***
   Prometheus berfungsi sebagai *server database time-series* yang mengambil dan menyimpan data dari *Node Exporter* dalam bentuk *time-series*. *Prometheus* melakukan proses *scraping* terhadap data yang disediakan oleh *Node Exporter* sesuai interval waktu yang telah ditentukan.
   Selain itu, *Prometheus* juga mendukung *query* bahasa *PromQL* untuk melakukan agregasi, *filter*, dan manipulasi data *metrik*. Data yang sudah dikumpulkan akan digunakan untuk keperluan pemantauan maupun analisis lebih lanjut, termasuk peramalan (*forecasting*). data merupakan catatan atas kumpulan fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, konsep, peristiwa dan lain sebagainya yang diwujudkan dalam bentuk huruf, angka, simbol, gambar, teks, bunyi atau kombinasi lainnya [11].

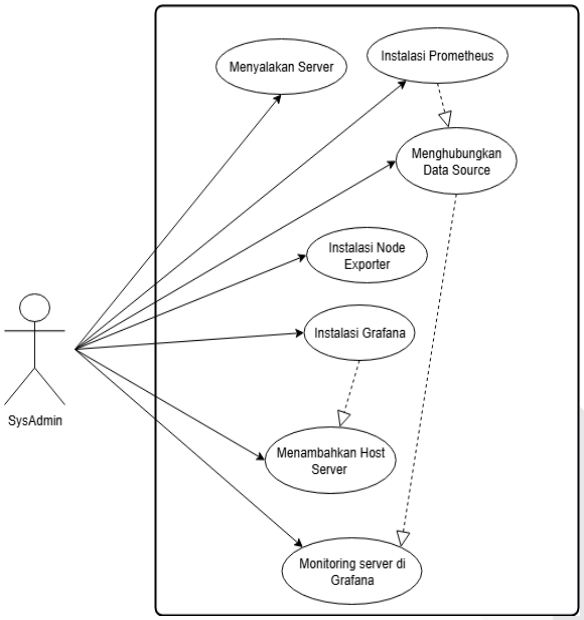
4. ***Grafana (Monitoring)***

   Grafana adalah antarmuka visualisasi yang terhubung ke *Prometheus* sebagai sumber datanya. Melalui *Grafana*, administrator dapat membuat *dashboard* interaktif yang menampilkan grafik dan indikator performa dari *server* secara *real-time*. *Grafana* juga dapat dikonfigurasi untuk menampilkan hasil *forecasting* terhadap data *metrik*, seperti prediksi penggunaan CPU dalam 7 hari ke depan, dengan memanfaatkan integrasi *Python script* atau *plugin forecasting*. Ini memberikan nilai tambah dalam pengambilan keputusan dan perencanaan kapasitas *server*.

5. **Telegram (*Notifikasi Alerting*)**

   Telegram Berfungsi sebagai platform penerima *alert* dari *Grafana* pada saat sumber daya server sudah mencapai batas *threshold* yang sudah ditentukan oleh *admininstrator* berdasarkan kebutuhan yang diinginkan. *System alerting* ini sangat penting dalam menjaga performa server dan dapat mengetehaui aktivitas di dalam server ketika ada anomali yang terjadi seperti lonjakan data atau terjadinya *downtime*, sehingga administrator dapat mengatasi masalah tersebut.

B. *Use Case System Monitoring*

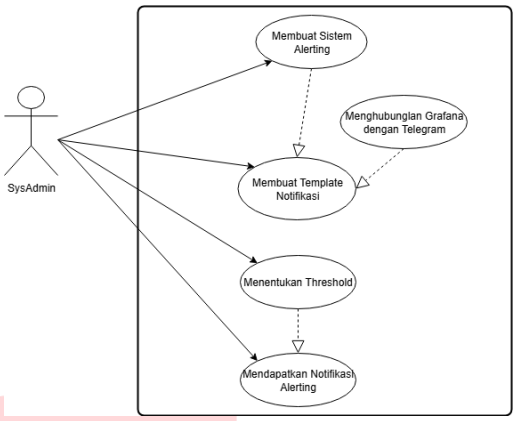


Gambar 3. 2 Use Case Monitoring

Gambar *use case* monitoring menggambarkan alur kerja *system monitoring server* menggunakan *Prometheus* dan *Grafana* yang dilakukan oleh seorang *System*. Proses dimulai dari menyalakan *server*, menginstal *Prometheus*, *Grafana*, dan *Node Exporter*, hingga menghubungkan *data source* dan menambahkan host ke *Grafana*. Setelah itu, *System* dapat melakukan *monitoring server* secara visual melalui *Grafana*.

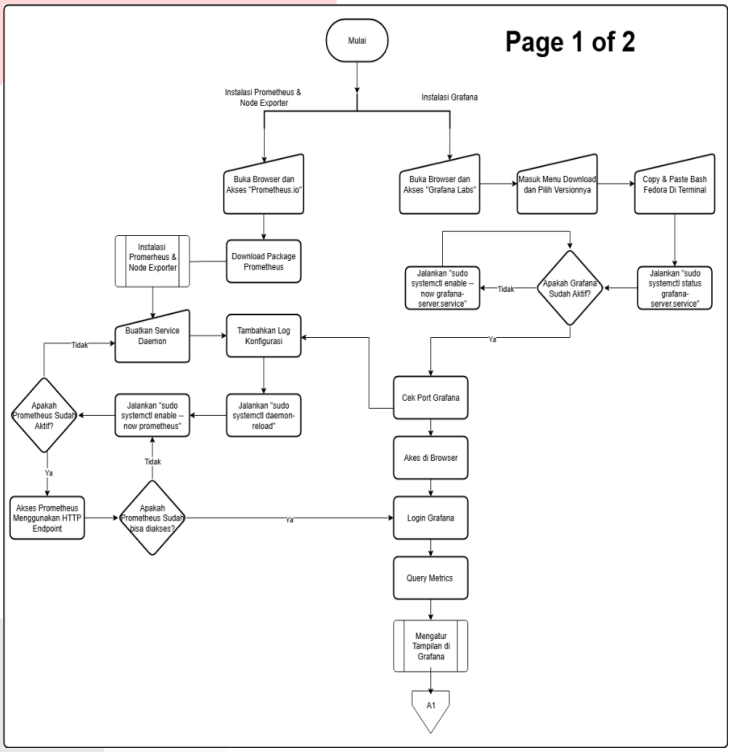
C. *Use Case System Alerting*

Gambar *use case* monitoring di bawah ini menggambarkan alur kerja membangun *system alerting* untuk mendeteksi anomali atau kondisi tertentu pada *server*. *System* menghubungkan *Grafana* dengan *Telegram*, membuat *template notifikasi*, dan menentukan *threshold*. Jika kondisi yang ditentukan terpenuhi, *system* akan mengirimkan notifikasi secara otomatis melalui *Telegram*. Diagram ini menunjukkan pentingnya integrasi antar komponen dan otomatisasi untuk meningkatkan responsivitas dan efisiensi dalam pengelolaan infrastruktur TI.



Gambar 3. 3 Use Case Alerting

D. Proses Pengerjaan (*System Monitoring*)

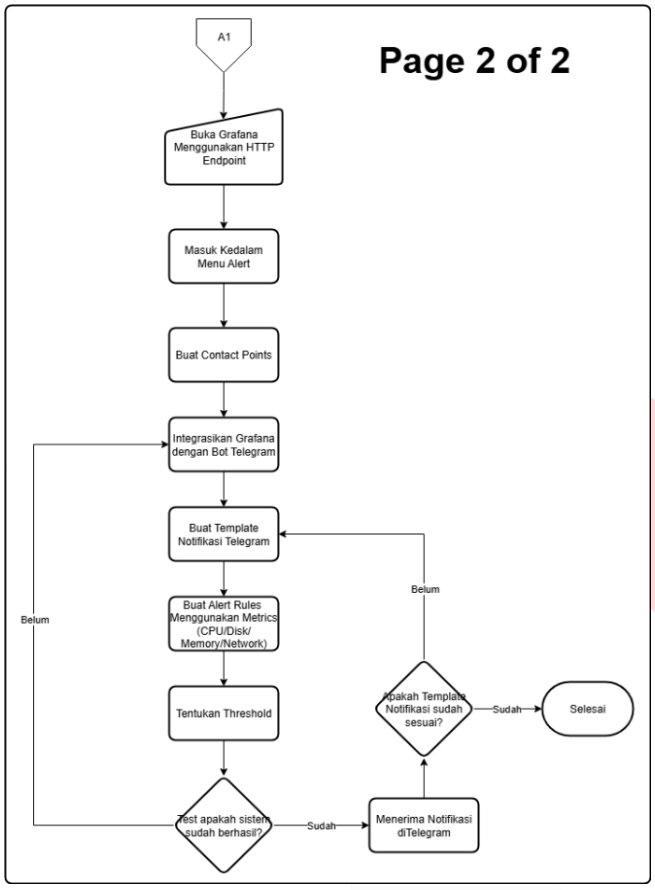


Gambar 3. 4 Instalasi System Monitoring Page 1 of 2

Flowchart "Page 1 of 2" menggambarkan tahapan instalasi dan konfigurasi sistem *monitoring* berbasis *Prometheus*, *Node Exporter*, dan *Grafana* untuk server di PT Dirgantara Indonesia. Proses dimulai dengan mengakses situs resmi masing-masing *tools*, mengunduh dan menginstal paket sesuai sistem operasi Fedora melalui *terminal*, lalu mengatur *service daemon* agar berjalan otomatis saat sistem menyala. Setelah *Prometheus* aktif dan dapat diakses melalui *HTTP endpoint*, proses dilanjutkan dengan instalasi *Grafana*, pengecekan status layanan, hingga akses ke *dashboard* melalui *browser*. Tahap akhir meliputi penambahan *data source* dari *Prometheus* ke *Grafana* dan visualisasi metrik seperti CPU, *memory*, *disk*, dan *network* dalam bentuk grafik interaktif.

E. Proses Pengerjaan (*System Alertig*)

Flowchart "Page 2 of 2" ini menjelaskan tahapan konfigurasi *alerting* di *Grafana*, dimulai dari membuka *Grafana* melalui *HTTP endpoint*, masuk ke menu *Alert*, dan membuat *contact points*. Selanjutnya, *Grafana* diintegrasikan dengan *bot Telegram* untuk mengirim notifikasi. Pengguna kemudian membuat *template notifikasi Telegram*, menentukan *alert rules* berdasarkan metrik (CPU, *disk*, *memory*, *network*), serta menetapkan *threshold*. Setelah diuji, sistem akan mengirim notifikasi ke Telegram. Jika *template* dan notifikasi sudah sesuai, proses dinyatakan selesai; jika belum, dilakukan penyesuaian dan pengujian ulang.
Ask ChatGPT



Gambar 3. 5 Instalasi System alerting Page 2 of 2

F. Standarisasi Spesifikasi *Server*

Spesifikasi *server* berikut merupakan standarisasi internal yang dibutuhkan untuk mendukung *system monitoring* dan pengolahan data secara optimal, khususnya dalam mengelola *metrik* performa dari banyak *server* secara *real-time*:

Tabel 3. 1 Tabel Standarisasi spesifikasi Server

Komponen	Spesifikasi
Storage	500 GB
CPU	8 Core
Ram	8 GB
Layanan Tambahan	*Prometheus,* Node Exporter, Grafana, dan *Alertmanager.*

PT. Dirgantara Indonesia memiliki total 128 *server*, di mana sekitar 50 *server* aktif digunakan untuk kebutuhan aplikasi, dan sekitar 4 hingga 6 *server* dalam kondisi tidak aktif. Sisanya difungsikan untuk pengolahan data internal perusahaan, termasuk proses komputasi, analisis, serta penyimpanan data penting. *Server* yang digunakan untuk *system monitoring* ini harus mampu menangani volume data *metrik* dari seluruh *server* yang aktif dan terhubung. Oleh karena itu, spesifikasi minimum seperti CPU 8 core dan RAM 8 GB diperlukan untuk menjamin proses pengumpulan dan pengolahan data berjalan lancar. Penyimpanan 500 GB digunakan untuk menyimpan log *system*, konfigurasi, serta data historis dari *metrik server*. Selain itu, koneksi jaringan yang stabil dan cepat diperlukan agar proses *monitoring* dan *alerting* dapat dilakukan secara *real-time* dan tanpa hambatan.

IV. HASIL DAN PEMBAHASAN

Implementasi *system monitoring* dan *alerting* ini dilakukan secara bertahap, dimulai dari proses instalasi *Prometheus* sebagai data *scraper* utama, pemasangan *Node Exporter* pada setiap *server* target, serta penyusunan *dashboard Grafana* sebagai media visualisasi data performa *server* [13]. *System* ini memungkinkan pemantauan *metrik* penting seperti penggunaan CPU, *memory*, *disk*, dan bandwidth jaringan secara *real-time*. Selain itu, *system* juga terintegrasi dengan fitur *alerting* melalui *Telegram* yang akan mengirimkan notifikasi otomatis apabila terjadi lonjakan performa, khususnya pada penggunaan CPU yang melebihi *Threshold* 75%.

A. PENGUJIAN *SYSTEM* FUNGSIONAL

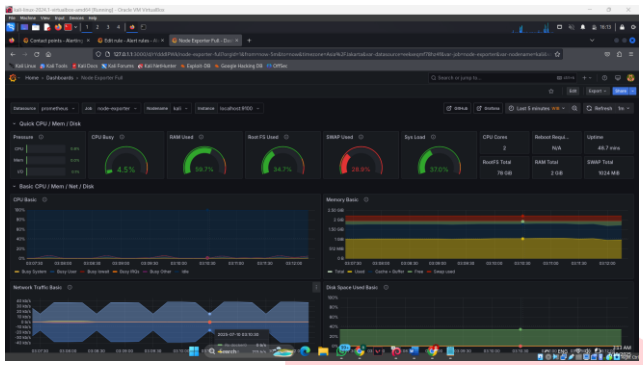
Adapun hasil dari pengujian *system* fungsional, hasil lengkapnya disajikan pada tabel 4.1 sebagai berikut :

Tabel 4. 1 Pengujian Funsgional

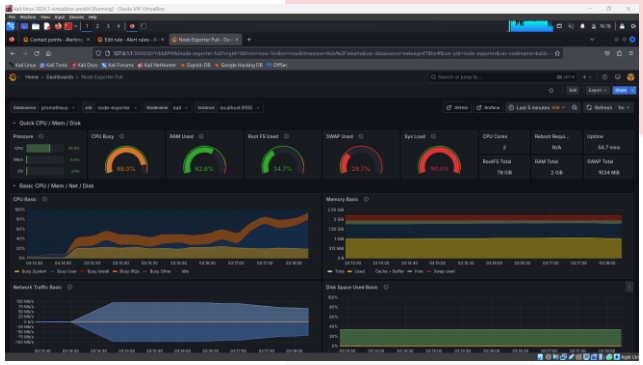
No	Nama Pengujian	Tujuan Pengujian	Hasil Pengujian	Kesimpulan
1	Login ke Dashboard Grafana	Memastikan pengguna dapat login dengan kredensial default	Pengguna berhasil login ke Grafana	*System* login berfungsi dengan baik
2	Eksplorasi Panel CPU & Memory	Melihat reaksi pengguna terhadap visualisasi performa	Pengguna dapat membaca grafik, mengganti waktu, melihat warna indikator	UI mudah dipahami, navigasi user-friendly
3	mulasi Serangan DDoS	Menguji apakah alert dikirim saat CPU melebihi threshold	Telegram menerima alert saat CPU > 75%, Grafana menampilkan warna merah	Fitur alert dan visualisasi threshold bekerja sempurna
4	Pengujian Notifikasi Telegram	Memastikan notifikasi diterima oleh pengguna ketika kondisi abnormal	Semua pengguna yang tergabung di grup Telegram menerima alert dari Prometheus	Integrasi Telegram dan alert *system* berjalan dengan baik

B. Metode Pengujian Stress Testing

Metode pengujian yang digunakan dalam pengembangan *system* ini mencakup pengujian simulasi serangan DDoS (*Distributed Denial of Service*) ringan terhadap *server* target. Tujuan dari pengujian ini adalah untuk mengetahui sejauh mana *system monitoring* dan *alerting* yang dibangun mampu mendeteksi lonjakan performa, terutama pada penggunaan CPU, sebagai akibat dari serangan tidak normal yang membebani *system* secara berlebihan dalam waktu singkat [15]. Jika penggunaan CPU melebihi nilai ambang batas (*threshold*) yang telah ditentukan, yaitu 75%, maka *system alerting* yang terhubung dengan *Telegram Bot API* akan secara otomatis mengirimkan notifikasi kepada administrator [14].



Gambar 4. 1 Data Sebelum Pengujian



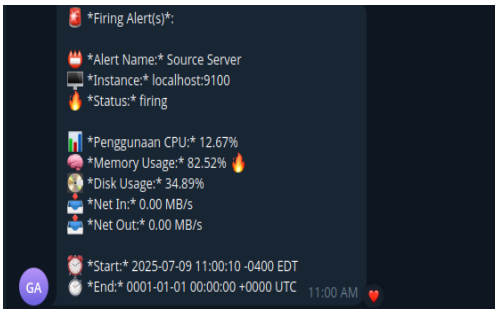
Gambar 4. 2 Data Sesudah Pengujian

Berikut adalah tabel pengujian stabilitas *system monitoring* selama 5 menit dengan refresh setiap 1 menit sekali. Pengujian ini dilakukan untuk mengukur ketahanan *system* saat menerima beban tinggi, seperti saat terjadi serangan DDoS:

Tabel 4. 2 Hasil pengujian system

Menit Ke	CPU Usage (%)	Memory Usage (%)	Disk Usage (%)	Jaringan (*Throughput*)
0 (Awal)	4.5	59.7	34.7	37 KB/s
1	37.4	61.4	34.7	30 MB/s
2	53.3	62.1	34.7	92 MB/s
3	55.7	63.4	34.7	87 MB/s
4	72.3	63.6	34.7	82 MB/s

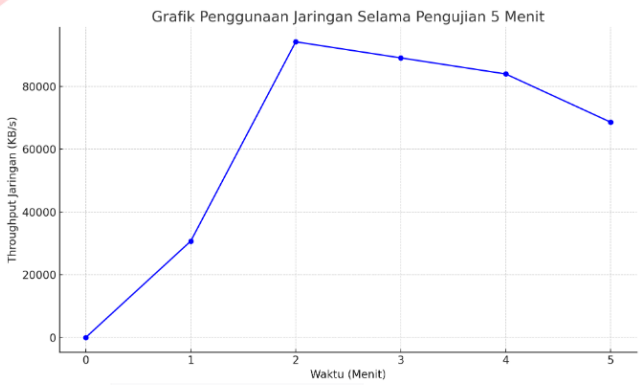
Selama pengujian berlangsung, *Prometheus* terus melakukan *scraping* data, dan *dashboard Grafana* tetap menampilkan data secara akurat tanpa *error*. *Alert* dari *Grafana* juga tetap aktif dan dikirim ke *Telegram* secara konsisten saat kondisi *threshold* tercapai. Hal ini menunjukkan bahwa sistem *alerting* berfungsi dengan baik dan responsif terhadap perubahan parameter sistem. Hasil pada Gambar 4.3 memperlihatkan notifikasi *alert* dari aktivitas pada server yang berhasil diterima melalui *Telegram*, menandakan integrasi berjalan lancar dan dapat diandalkan dalam mendeteksi serta memberi peringatan dini terhadap anomali sistem.



Gambar 4.1 Notifikasi Telegram

Hasil pengujian menunjukkan bahwa *system monitoring* dan *alerting* ini berjalan stabil, responsif, dan dapat memberikan peringatan secara tepat waktu, sehingga layak diterapkan dalam lingkungan operasional yang membutuhkan pengawasan *system server* secara *real-time*.

Adapun grafik *traffic network* selama waktu pengujian untuk memudahkan dalam membaca tren, disajikan dalam bentuk visualisasi berikut. Grafik ini merepresentasikan perubahan *traffic* jaringan (dalam satuan MB/s) selama durasi pengujian lima menit, dengan interval pemantauan setiap satu menit sekali. Visualisasi ini membantu dalam mengidentifikasi pola lonjakan atau anomali *traffic* yang dapat mengindikasikan aktivitas mencurigakan seperti serangan DDoS, sebagai berikut:



Gambar 4. 3 *Traffic Throughput*

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian, sistem *monitoring* dan visualisasi performa server yang dikembangkan berhasil memenuhi seluruh tujuan yang ditetapkan. Sistem ini mampu memantau lebih dari 128 *Virtual Machine* di lingkungan PT Dirgantara Indonesia dengan metrik penting seperti CPU, *memory*, *disk*, dan *network* yang ditampilkan secara *real-time* melalui *dashboard Grafana*. Fitur *alerting* otomatis juga telah berfungsi dengan baik, mengirimkan notifikasi ke *Telegram* saat terjadi anomali melebihi *threshold*, memungkinkan tim IT melakukan mitigasi lebih cepat. Dalam pengembangannya, sistem memanfaatkan *tools open-source* seperti *Prometheus*, *Node Exporter*, dan *Grafana* pada sistem operasi Linux Fedora untuk proses pengambilan data, penyimpanan *time series*, visualisasi, serta

notifikasi. Secara keseluruhan, sistem ini mendukung operasional IT secara efisien, mudah diawasi, hemat biaya, dan sangat layak untuk diimplementasikan serta dikembangkan lebih lanjut.

B. Saran

Berdasarkan kesimpulan yang telah disampaikan, pengembangan sistem *monitoring* ke depan dapat mencakup integrasi dengan sistem manajemen IT perusahaan seperti *IT Service Management* atau *ticketing system* untuk mempercepat penanganan insiden. Selain itu, peningkatan aspek keamanan juga penting dilakukan, misalnya dengan menerapkan enkripsi data dan autentikasi multi-faktor guna melindungi akses ke *dashboard monitoring*.

.

REFERENSI

[1] Harsono, H. (2022). Faktor-Faktor Yang Mempengaruhi Sistem Informasi Berbasis Komputer: Sistem Operasi, Server, Dan Programmer (Literature Review Executive Support Sistem for Business). *Jurnal Manajemen Pendidikan Dan Ilmu Sosial*, *3*(2).

[2] Tasrif, E., Huda, A., Saputra, H. K., & Mubai, A. (2019, November). Design Of Server Performance Monitoring Application Integrated Administration Service System In Electronic Engineering Department. In *Journal Of Physics: Conference Series* (Vol. 1387, No. 1, p. 012029). IOP Publishing.

[3] Saputra, M. Y. E., Noprianto, Arief, S. N., Wijayaningrum, V. N., & Syaifudin, Y. W. (2024). *Real-time server monitoring and notification system with Prometheus, Grafana, and Telegram integration*. In Proceedings of the 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS) (pp. 1808–1813). IEEE. https://doi.org/10.1109/ICETSIS61505.2024.10459488

[4] Febriana, R. M. (2020). Implementasi Sistem Monitoring Menggunakan Prometheus Dan Grafana. *Semin. Nas. Telekomun dan Inform*, *13*, 164-169.

[5] Sulaeman, Y. J. (2022). Implementasi Network Monitoring dan Notifikasi Sistem di PT XYZ Menggunakan Zabbix. *Jurnal Instrumentasi dan Teknologi Informasi (JITI)*, *4*(1), 1-7.

[6] Rasyidi, B., & Pratama, F. (2024). Sistem Monitoring Server di PT. XYZ Media Indonesia Berbasis Grafana dan Prometheus: Server Monitoring System at PT. XYZ Media Indonesia Based on Grafana and Prometheus. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(4), 1456-1465.

[7] Amirudin, M. Z., Fahmi, R., Utami, E., & Mustafa, M. S. (2021). Evaluasi Penggunaan Prometheus dan Grafana Untuk Monitoring Database Mongodb. *Jurnal Informatika Polinema*, *7*(2), 43-50.

[8] Nurrohman, I. (2024). *PERANCANGAN DAN IMPELENTASI SISTEM PEMANTAUAN JARINGAN BERBASIS PROMETHEUS DAN GRAFANA DI SMK HARAPAN BANGSA* (Doctoral dissertation, Sekolah Tinggi Teknologi Terpadu Nurul Fikri).

[9] Rahman, D., Amnur, H., & Rahmayuni, I. (2020). Monitoring server dengan prometheus dan grafana serta notifikasi telegram. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, *1*(4), 133-138.

[10] Ahmim, A., Maazouzi, F., Ahmim, M., Namane, S., & Ben Dhaou, I. (2023). *Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model*. IEEE Access, 11, 119862–119875.

[11] Rachmadi, T., & Kom, S. (2020). *Sistem basis data* (Vol. 1). Tiga Ebook.

[12] Rishwan, R. M., Nurkifli, E. H., & Solehudin, A. (2025). ANALISIS PERBANDINGAN PERFORMA SERVER WEBSOCKET DENGAN MENGGUNAKAN PAYLOAD JSON, BINARY SERIALIZATION, DAN PROTOBUF DENGAN MENGGUNAKAN METODE LOAD TESTING. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(1), 706-712.

[13] Siddiqui, I., Pandey, A., Jain, S., & Kothadia, H. (2023). Comprehensive monitoring and observability with Jenkins and Grafana: A review of integration strategies, best practices, and emerging trends. 2023 7th International Conference on Computing Methodologies and Communication (ICCMC). IEEE.

[14] Putra, M. A., & Sari, R. (2022). Sistem Monitoring Serangan pada Mikrotik Berbasis Bot Telegram. *Jurnal Ilmiah Manajemen dan Informatika*, 8(1), 45-52. Diakses

[15] Nugroho, A. (2021). Analisis Keamanan Website Terhadap Serangan DDoS. *Jurnal Keamanan Informasi*, 6(2), 78-85.