Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 4 Iss. 4 October 2024, pp: 1456-1465
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Server Monitoring System at PT. XYZ Media Indonesia Based on Grafana and Prometheus

Sistem *Monitoring Server* di PT. XYZ Media Indonesia Berbasis Grafana dan Prometheus

Banu Rasyidi[1*], Firman Pratama[2]

[1,2]Program Studi Teknik Informatika , Fakultas Ilmu Komputer, Universitas Pamulang, Indonesia

E-Mail: [1]banurasyidi17@gmail.com, [2]dosen02407@unpam.ac.id

Received Jul 16th 2024; Revised Aug 30th 2024; Accepted Sept 10th 2024
Corresponding Author: Banu Rasyidi

Abstract

The server is one of the main components of a computer network system, functioning to provide services to its users. The activities and operational services of a server to its clients involve several types of processes to fulfill all client requests sent by the server. Therefore, a monitoring system is required to oversee all activities within the server, enabling users to monitor and receive alerts if any issues arise with the server. This system is designed using Unified Modeling Language (and employs a node exporter to collect metrics, Prometheus Query Language to access metric data, and Grafana for visualization.This research aims to design a system capable of monitoring and collecting information from all server devices at PT. XYZ Media Indonesia. This monitoring system can provide information related to memory utilization, CPU utilization, storage utilization, and network utilization on the server. Additionally, it can send alert messages in the form of Telegram notifications to users when errors or malfunctions occur on the server, thereby enabling faster escalation and resolution of server issues.

Keyword: Monitoring, Node Exporter, Prometeheus, Server, Telegram

Abstrak

Server merupakan salah satu komponen utama dari sistem jaringan komputer yang memiliki fungsi untuk memberikan suatu service terhadap penggunanya. Setiap aktifitas dan operasional pelayanan suatu server terhadap client dalam penerapannya terdiri dari beberapa jenis proses untuk memenuhi segala permintaan atau request client yang dikirimkan oleh server. Oleh sebab itu, diperlukan suatu sistem monitoring yang dapat melakukan pemantauan segala macam aktifitas di dalam server, dengan demikian pengguna dapat memantau dan memberikan peringatan apabila terjadi permasalahan pada server yang digunakan. Pada sistem ini dirancang dengan menggunakan perancangan Unified Modeling Language, serta menggunakan node exporter untuk mengambil metrik dan bahasa kueri berbasis Prometheus Query Language untuk mengakses data metrik dan grafana untuk memvisualisasi. Penelitian ini bertujuan rancang sistem yang mampu melakukan pemantauan dan pengumpulan informasi dari seluruh perangkat server yang berada di PT. XYZ Media Indonesia. Sistem monitoring ini juga, mampu memberikan beberapa informasi terkait utilisasi memori, utilisasi CPU, utilisasi storage dan utilisasi jaringan pada server, serta pada sistem ini juga mampu mengirimkan pesan peringatan dalam bentuk notifikasi telegram kepada pengguna saat terjadi kesalahan atau malfunction pada server dengan demikian proses eskalasi penanganan server menjadi lebih cepat dilakukan.

Kata Kunci: Monitoring, Node Exporter, Prometeheus, Server, Telegram

1. PENDAHULUAN

Perkembangan teknologi saat ini semakin cepat dan pesat dari waktu ke waktu, khususnya teknologi informasi dan komunikasi[1]. Teknologi merupakan salah satu bagian alat bantu yang sangat sering digunakan dalam aktifitas pekerjaan sehari-hari, peran teknologi dalam melakukan pengolahan informasi menjadi sangat penting, karena dapat membantu menghasilkan informasi yang akurat dan valid bagi penggunanya [2]. Dengan demikian, penulis *server* dan membangun sistem monitoring *server* dengan menggunakan grafana, prometheus dan node exporter agar mempermudah memonitoring *server* pada Perusahaan PT. XYZ Media Indonesia.

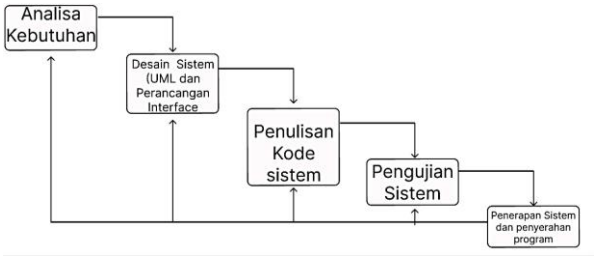
Server Merupakan suatu sistem komputer besar yang terintegrasi dan tersusun pada suatu jaringan komputer besar yang menyediakan suatu layanan bagi para pengguna yang biasa disebut sebagai *client*. Suatu *server* umumnya telah menjalankan banyak proses untuk membantu dalam memenuhi permintaan dari *client* [3]. Oleh sebab itu, sering sekali *server* mengalami beberapa kendala yang disebabkan karena tidak adanya sumber daya *(resource)* yang mumpuni atau sesuai untuk memenuhi *resource* perbaikan *server* Hal tersebut dapat menyebabkan layanan atau *service server* mati secara mendadak dikarenakan *file system kernel* memutuskan untuk mematikan layanan pada *server* yang membutuhkan *resource* yang sangat besar. *File system kernel* adalah komponen inti atau penting dari suatu sistem operasi. *Kernel* memiliki tanggung jawab untuk menyelesaikan tugas-tugas tingkat rendah/bawah seperti untuk manajemen storage, manajemen CPU dan manajemen memori [4].

Permasalahan pada perusahaan masih menggunakan metode manual, seperti perintah *command line top* untuk memantau penggunaan CPU dan df -h untuk melihat penggunaan *disk*. Metode ini memiliki beberapa keterbatasan yang signifikan. Oleh karena itu, penelitian ini penting untuk merancang dan membangun sebuah sistem monitoring otomatis yang dapat memberikan data yang lebih akurat, *real-time*, dan mudah diakses. Sistem yang dirancang dengan baik tidak hanya akan mengurangi beban kerja tim , tetapi juga meningkatkan efisiensi operasional dan membantu dalam pengambilan keputusan yang lebih baik.Seperti dapat mengumpulkan beberapa informasi *server* seperti utilisasi CPU, utilisasi memori, utilisasi storage dan utilisasi network. Adapun kebutuhan yang dibutuhkan dalam membangun sistem monitoring ini adalah. dengan menggunakan perancangan *Unified Modeling Language* (UML), serta menggunakan node exporter untuk mengambil metrik dan bahasa kueri berbasis *Prometheus Query Language* (PromQl) untuk mengakses data metrik dan grafana untuk memvisualisasi. Promteheus juga banyak digunakan untuk menyimpan data baik untuk jaringan internet maupun *Local Area Network* (LAN) atau biasa disebut intranet.

Terdapat penelitian sebelumnya yang pernah melakukan sistem monitoring server menggunakan grafana dan prometheus , seperti penelitian implementasi sistem monitoring menggunakan grafana dan prometheus [5], pada penelitian tersebut sistem hanya memiliki visual utilisasi data *resource server* dan tidak memiliki pemberitahuan kepada pengguna. Kemudian monitoring server dengan prometheus dan grafana serta notifikasi telegram[6], pada penelitian tersebut sistem hanya memiliki satu ultilisasi resources server yang digabung dan memiliki mobilitas yang cepat tetapi sistem ini kurang efisien untuk perusahaan yang memiliki banyak *server*. selanjutnya pada penelitian evaluasi penggunaan prometheus dan grafana untuk monitoring *database* Mongodb [7], pada penelitian tersebut sistem memiliki monitoring ultilisasi server yang baik dan efisien tetapi tidak memiliki pemberitahuan *alert* jika server mengalami kendala. Pada penelitian monitoring Kubernetes cluster menggunakan prometheus dan grafana[8], pada penelitian tersebut sistem memiliki data ultilisasi server secara lengkap tetapi tidak memiliki tampilan *user friendly* untuk pengguna .dikarenakan selalu adanya kekurangan sistem pada penelitian terdahulu maka penelitian ini memiliki sistem monitoring server ini memiliki data ultilisasi *resource server* secara rinci dan dibagikan berdasarkan grup seperti CPU, memori, disk, *network traffic* dan lain- lain. untuk mobilitas data yang secara efisien. Sistem ini memiliki *user interface* yang mudah dipahami oleh pengguna dan sistem ini memiliki pemberitahuan *alert server* jika mengalami kendala.

Penelitian ini bertujuan untuk membangun penerapan suatu Sistem *monitoring server* yang sebelumnya dilakukan secara manual . dengan adanya sistem menggunakan grafana,prometeheus dan node exporter agar mendukung pemantauan *server* secara real time dan otomatis memberikan *alert* kepada pengguna. Selain itu, pada sistem ini juga akan mengumpulkan data *log server* untuk beberapa parameter yang bersumber dari ultilisasi CPU, *storage*, memori dan *network* yang berasal dari *server*. Dan jika mengalami kendala *threshold* pada server ,sistem ini akan melakukan pemberitahuan *alert* kepada pengguna melalui *platform telegram*.

2. DASAR TEORI PENUNJANG DAN METODE PENELITIAN



Gambar 1. Metode Penelitian

Metode *waterfall* merupakan kerangka kerja yang mengatur proses pembuatan secara linear dan berurutan [9]. Proses nya terdiri dari serangkaian tahapan yang dijalankan secara berurutan dan terpisah satu

sama yang lain tahap pertama melibatkan analisis mendalam terhadap sistem yang akan dikembangkan di ikuti tahap Perancangan sistem, penulisan kueri sistem ,pengujian sistem dan penyerahan [10] .

Dalam penelitian ini seperti yang dilihat pada gambar 1, terdapat lima tahap utama . jadi pada tahapan penelitian ini yaitu Indentifikasi Masalah kebutuhan perusahaan seperti perusahaan untuk monitoring server dilakukan secara manual dan fungsionalitas sistem yang diinginkan oleh perusahaan adalah sistem monitoring sistem dengan notifikasi alert jika mengalami kendala pada server perusahaan dengan pengumpulan data seperti wawancara terhadap karyawan perusahaan seperti penggunaan *software xshell* untuk melakukan remote dan penggunaan grafana ,prometheus dan node- exporter, Desain Unified Modeling Language (UML) dan Perancangan *Interface* sistem seperti *flowchart sistem*¸ arsiterktur sistem, *use care diagram, activity diagram dan sequence diagram* .

Pengkodean sistem menggunakan bahasa bash untuk mengkonfigurasi service grafana, prometheus, node-exporter dan untuk *service* untuk notifikasi telegram dan bahasa kueri prometheus *query language* digunakan untuk mengambil metriks dari node exporter. selanjutnya Pengujian Sistem Dengan Menggunakan *Black box* melakukan pengujian terhadap sistem dan notifikasi pada sistem. Setelah sistem diuji dan disetujui, program siap untuk diimplementasikan. Pengguna kemudian dilatih untuk menggunakan sistem, dan dokumentasi sistem biasanya diserahkan sebagai panduan untuk pemeliharaan dan pengembangan lebih lanjut

2.1 Xshell

Xshell adalah sebuah emulator terminal yang dirancang untuk pengguna windows agar dapat mengakses server unix atau linux. Ini memungkinkan pengguna untuk berinteraksi dengan shell dari sistem unix/linux dari komputer windows mereka [11]. Fitur utamanya meliputi emulasi terminal yang kuat, manajemen sesi, keamanan dengan enkripsi SSH , pengeditan teks lanjutan, skrip dan otomatisasi, xshell memiliki antarmuka yang menggunakan sistem tab, memungkinkan pengguna untuk membuka beberapa sesi terminal dalam satu jendela . Ini mempermudah pengelolaan dan navigasi antara berbagai koneksi tanpa harus membuka jendela terminal terpisah untuk setiap sesi.

2.2 Ubuntu Server

Ubuntu *server* adalah sebuah distribusi linux yang dikembangkan secara terbuka dan didesain khusus untuk digunakan sebagai sistem operasi pada *server*. Mirip dengan sistem operasi ubuntu pada umumnya, ubuntu *Server* juga bersifat open-source, artinya kode sumbernya dapat diakses dan dimodifikasi secara bebas oleh pengguna [12] , Ubuntu *server* juga memiliki kesamaan dalam konsep *open-source* yang memungkinkan pengguna untuk mengakses dan memodifikasi kode sumbernya sesuai dengan kebutuhan perusahaan.

2.3 Grafana

Grafana adalah perangkat lunak sumber terbuka yang digunakan untuk memvisualisasikan dan menganalisis data dalam bentuk grafik dan dashboard interaktif. Secara ilmiah, grafana merupakan alat yang sering digunakan dalam bidang pemantauan sistem dan analisis data [13],Grafana memungkinkan pengguna untuk mengintegrasikan dan memvisualisasikan data dari berbagai sumber, termasuk basis data , sistem pemantauan seperti prometheus.

2.4 Prometheus

Prometheus dirancang untuk memantau kinerja dan status sistem monitoring secara *real-time* dengan mengumpulkan metrik dari berbagai komponen sistem serperti utilisasi *server*.prometheus meliputi kemampuan untuk mengumpulkan dan mempertahankan data metrik yang memungkinkan pengguna untuk menganalisis perilaku sistem [14].

2.5 Node Exporter

Node Exporter adalah alat pengumpul metrik yang digunakan dalam ekosistem prometheus untuk memantau kinerja dan status sistem operasi. Alat ini berfungsi untuk mengumpulkan data seperti penggunaan CPU, memori, disk I/O, jaringan, dan berbagai metrik lainnya yang berkaitan dengan kinerja infrastruktur. Data yang dikumpulkan oleh Node Exporter [15] .

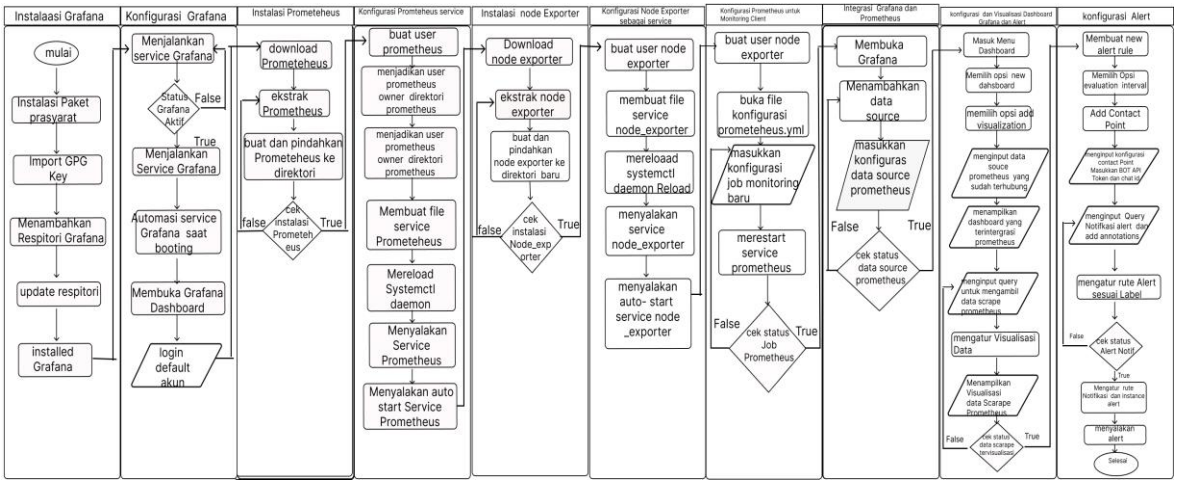
3. HASIL DAN PEMBAHASAN

Tahap ini merupakan tahapan yang berguna untuk mengetahui sistem. Hasil *scarape* metrik *resource server* di PT. XYZ Media Indonesia merupakan kebutuhan sistem. *Unified Modeling Language* (UML) merupakan model yang digunakan untuk merancang dan mendokumentasikan sistem Monitoring terstruktur dan pengujian fungsional dengan metode *black box*.

3.1 Flowchart Sistem

Flowchart atau diagram alur adalah representasi grafis dari serangkaian langkah atau proses [16]. Dalam penelitian ini. Dimulai dengan instalasi yang dibutuhkan oleh sistem. Tahap selanjutnya adalah

konfigurasi sistem dengan pembuatan file service yang berisi konfigurasi grafana, prometheus dan node exporter. tahap selanjutnya adalah tahap integrasi data *source* ke grafana untuk menampilkan visualisasi dan tahap terakhir adalah konfigurasi alert rule sesuai kebutuhan kemudian alert dikirimkan melalui platform telegram kepada pengguna.



Gambar 2. *Flowchart* Sistem

Gambar 2 adalah *flowchart* instalasi dan intergrasi sistem monitoring berkerja. Pertama admin *install* paket prasyarat yang dibutuhkan dan impor GPG *key* .Setelah instalasi selesai proses dimulai dengan mengupdate respitori dan menjalankan *service* grafana , selanjutnya prometheus di unduh ,diinstall dan di ekstrak diikuti dengan pembuatan user khusus untuk prometheus dan penetapan kepemilikan direktori prometheus terkait . Setelah itu menyalakan *service* promtheus dan automasi *service* nya. Hal yang sama dilakukan untuk node_exporter.

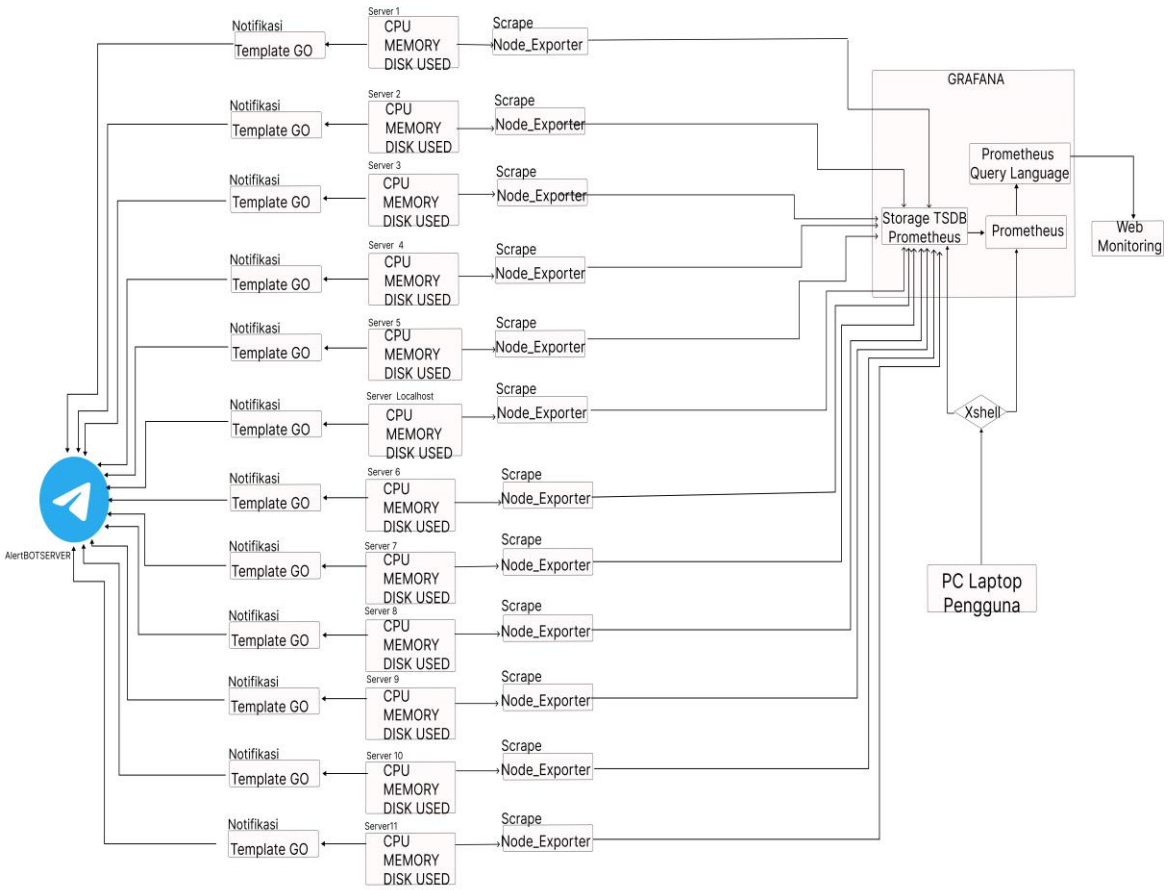
Setelah mengekstrak promtehus dan node exporter , langkah berikutnya penambahan respitori grafana dan memastikan otomatisasi service grafana saat *booting* . kemudian dilakukan cek instalasi dan status prometheus serta node exporter. *Service* prometheus dan node exporter dijalankan ,diikuti pembuatan *file service* prometheus dan konfigurasi *auto-start* untuk kedua *service* tersebut.konfigurasi sistem berlanjut dengan pembuatan *file service* untuk node exporter dan memasukkan konfigurasi kedalam file prometheus ,yml. Konfigurasi tambahan mencakup pengaturan *job monitoring* baru serta data *source* prometheus. Data *source* ini kemudian di input ke grafana untuk memulai integrarsi data . Pada Grafana Dashboard baru dibuat dan data *source* ditambahkan .

Selanjutnya *alert rule* diatur ,*query* di input untuk mengambil data scrape dari prometheus , dan rute *alert* disesuaikan label ditetapkan . Konfigurasi visualisasi data serta notifikasi *alert* dan *instance alert* dilakukan untuk memastikan sistem monitoring berjalan sesuai harapan. Proses akhir , *service* prometheus di restart dan status data source promtheus dicek untuk integrasi berjalan dengan baik . Evaluasi *service* dan *autostart service* diverifikasi untuk kelancaran sistem.

3.2 Arsitektur Sistem Monitoring Server

Pada gambar 3 di bahwa grafana merupakan *platform open-source* yang digunakan untuk memvisuasualisasikan dan memantau data berbasis web . pada grafana yaitu prometheus merupakan sistem pemantauan yang digunakan untuk mengambil metrik *server* dan *storage TSDB* dalam prometheus merupakan *database binery* yang digunkan untuk menampung data-data *Node*_exporter yang *scrape* data dari *server* kemudian Prometheus *query language* digunakan untuk menampilkan hasil *scrape* dalam bentuk panel yang ada di web monitoring yang dapat diakses secara jaringan intranet.

Pada gambar 3 diatas sistem ini mengadopsi node_exporter untuk mengumpulkan data metrik seperti penggunaan cpu, memori, dan disk dari beberapa server, yang kemudian di-*scrape* oleh node exporter kemudia prometheus mengscarpe dari node exporter selanjutnya disimpan dalam *time series database* (TSDB). Untuk visualisasi data, Grafana pengguna dapat menggunakan dengan Prometheus Query Language (PromQL) untuk menampilkan dashboard monitoring real-time, memungkinkan pemantauan kondisi server secara efisien. Selain itu, sistem ini dilengkapi dengan mekanisme peringatan otomatis melalui AlertBotServer yang mengirimkan notifikasi ke Telegram berdasarkan template yang telah ditentukan.



Gambar 3. Arsitektur Sistem Monitoring

3.3 *Use Case Diagram*

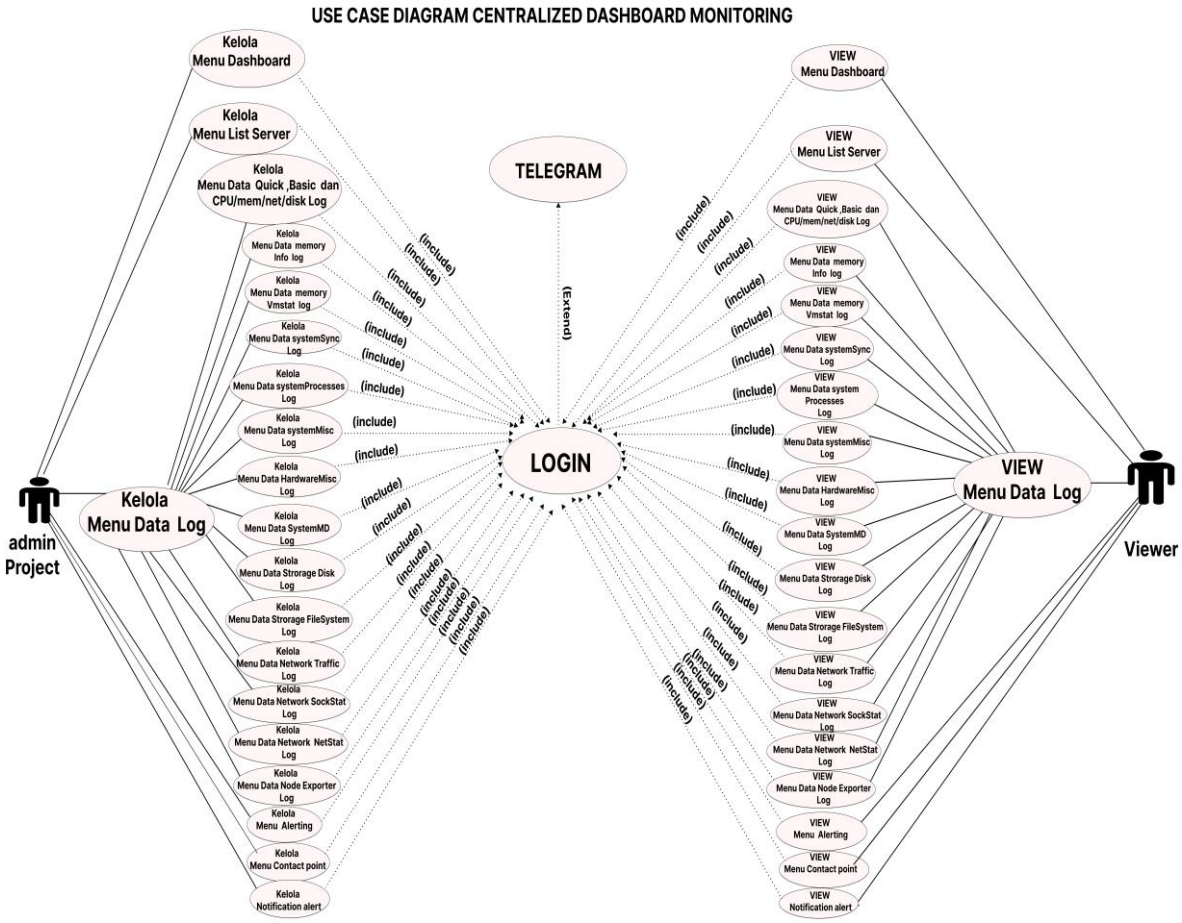
Dari *use case diagram* dijelaskan bahwa pada saat pengguna login sebagai admin , maka pengguna bisa melakukan pengelolaan menu dari sistem tersebut.pada gambar tersebut, <<*include*>> berarti bahwa admin dan *user* harus terlebih dahulu login agar dapat menjalankan fungsi sistem tersebut , dan pada gambar <<*extend*>> berarti bahwa admin dan user baru bisa mengelola menu *contact point* , menu *alerting* dan menu notifikasi telegram apabila pesan dikirim kan secara otomatis.

Diagram ini menampilkan interaksi antara dua aktor utama, yaitu admin project dan *viewer*, dengan sistem. Admin Project bertanggung jawab untuk mengelola berbagai aspek sistem, termasuk menu dashboard, daftar server, dan data log yang mencakup informasi penting seperti penggunaan CPU, memori, jaringan, dan disk. Admin dapat melakukan manajemen data secara mendetail, seperti mengelola data sistem, log jaringan, log penyimpanan, serta konfigurasi alerting dan notifikasi. Proses pengelolaan ini mencakup berbagai sub-menu yang memungkinkan admin untuk memantau performa server secara komprehensif.

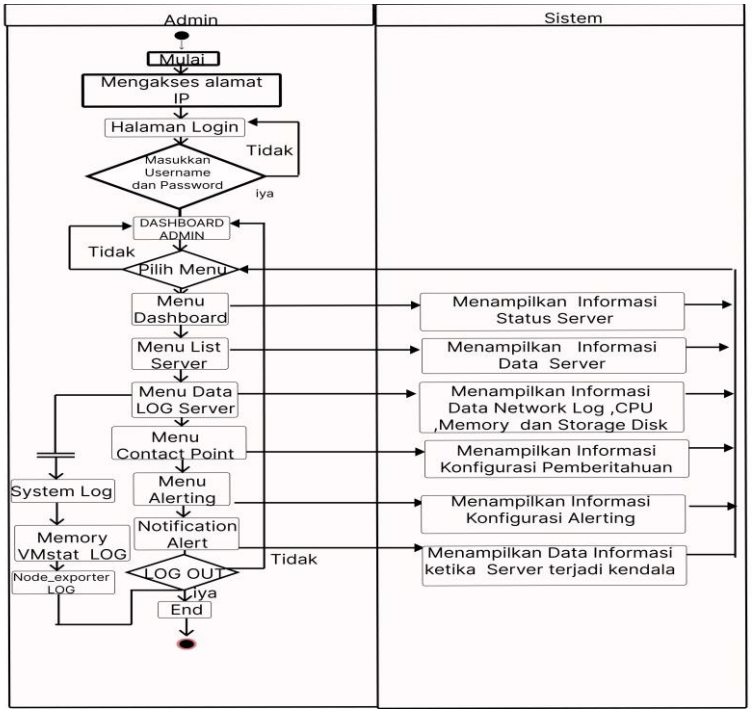
Sementara itu, *viewer* memiliki akses yang terbatas pada fungsi melihat atau *view* menu dashboard, daftar server, dan data log. Viewer dapat melihat data yang telah dikelola oleh Admin, tetapi tidak memiliki hak untuk melakukan perubahan atau pengelolaan. Fungsi login adalah langkah awal bagi kedua aktor untuk masuk ke dalam sistem dan mengakses fitur-fitur yang relevan sesuai peran mereka. Setelah login, pengguna, baik Admin maupun *viewer*, dapat mengakses berbagai menu sesuai dengan hak akses yang telah diberikan. Selain itu, diagram ini juga menunjukkan adanya ekstensi ke *telegram*, yang mengindikasikan bahwa notifikasi atau peringatan dari sistem monitoring ini dapat dikirim melalui platform tersebut.

3.4 *Activity Diagram*

Pada *activity* diagram yang terlihat pada gambar . Setelah proses login sebagai admin maka sistem akan menampilkan semua menu dimana admin dapat melakukan tambah ,ubah ,dan hapus dari semua data yang ada. pada halaman utama *dashboard admin* akan menampilkan informasi data *list server, data log, contact point , alerting dan* notifkasi alert.



Gambar 4. *Use Case* Diagram



Gambar 5. *Activity* Diagram

Proses dimulai dengan langkah awal di mana admin mengakses alamat ip untuk membuka halaman *login.* pada halaman login, admin harus memasukkan *username* dan *password* untuk mengautentikasi. Jika

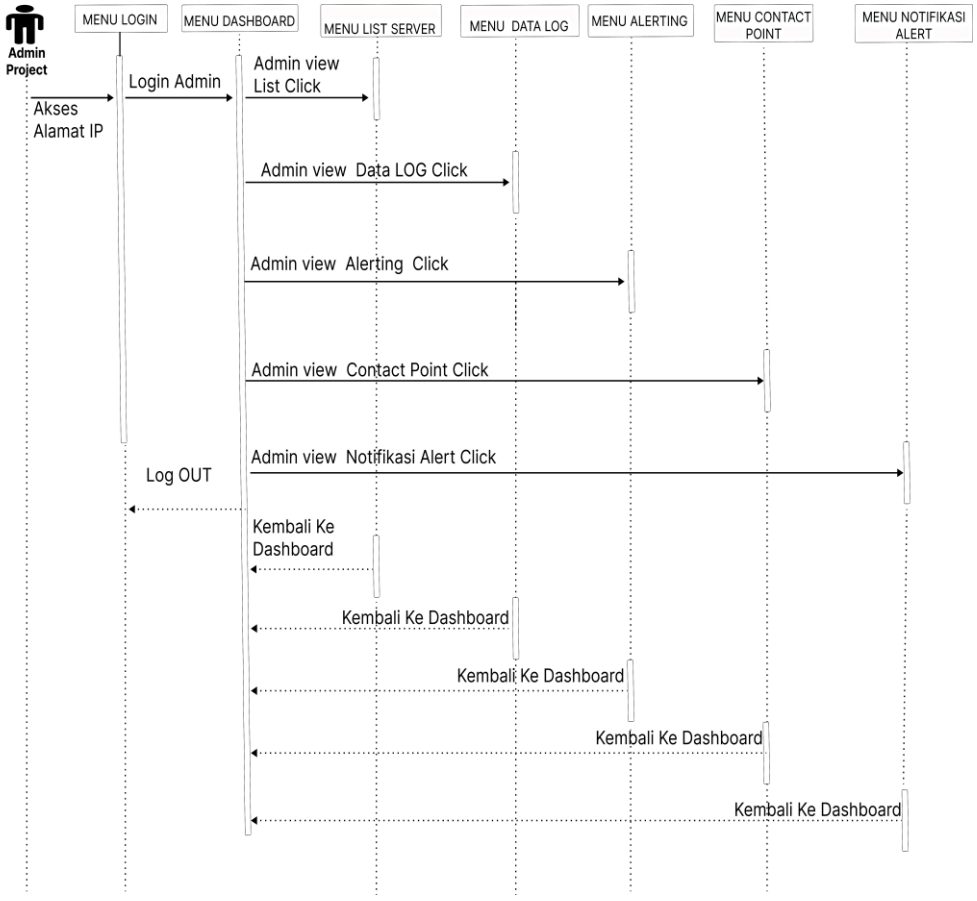
informasi login yang dimasukkan tidak benar, sistem akan mengarahkan kembali ke halaman *login*. namun, jika informasi login benar, admin akan diberikan akses ke dashboard admin.

Setelah berhasil masuk ke dashboard admin, admin diberikan pilihan untuk mengakses berbagai menu. menu yang tersedia meliputi menu dashboard, yang menampilkan informasi status server, dan menu list server, yang memberikan informasi data server secara rinci. Admin juga dapat memilih menu data log server untuk mengakses informasi data log jaringan, CPU, memori, dan disk penyimpanan. jika admin memilih menu *contact point*, sistem akan menampilkan informasi konfigurasi pemberitahuan, yang mencakup detail tentang bagaimana notifikasi dikirim ketika kondisi tertentu terpenuhi.

Selanjutnya, ada opsi menu *alerting* yang memungkinkan admin untuk melihat informasi tentang konfigurasi alerting yang ada di sistem. admin dapat memantau notifikasi yang masuk melalui *notification* alert, yang berfungsi memberikan peringatan apabila ada masalah dengan server. Selain itu, admin memiliki akses untuk melihat log sistem dan log performa memori melalui system log dan memory vmstat log. admin juga dapat mengakses log yang dihasilkan oleh node_exporter, yang memberikan metrik tambahan tentang kinerja server. Jika pada suatu titik admin memutuskan untuk keluar, ada pilihan *log out* untuk mengakhiri sesi, dan proses ini akan membawa kembali admin ke titik akhir, menutup akses terhadap sistem.

3.5 *Sequence Diagram*

Pada diagram *sequence* bisa dilihat yang menjadi *actor* adalah admin. *activation boxes* biasanya memiliki garis yang memberitahu aktifitas yang terjadi ketika *actor* atau objek berinteraksi ke objek lain .ketika *actor login* sebagai admin ,maka aktifitas yang terjadi adalah menu *dashboard* admin, menu *list server*, menu *data log*, menu *alerting*, menu *contact point* dan menu notifikasi alert .



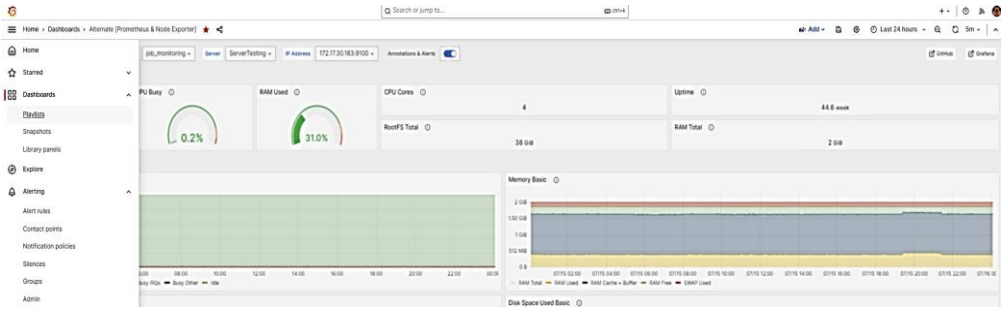
Gambar 6. *Sequence* Diagram

Pada gambar diatas admin mengakses alamat ip kemudian admin login menggunakan username dan password. sistem akan menampilkan menu dashboard selanjut nya admin memilih menu list server akan menampilkan list server yang di monitoring , admin memilih menu data log sistem akan menampilkan dala log ultilisasi resource server kepada admin , selanjutnya admin memilih menu contact point sistem akan menampilkan konfigurasi alert ,kemudian admin memilih notifikasi alert sistem akan menampilkan konfigurasi template notifikasi alert dan admin ingin mengakhiri sistem admin memilih *log out* sistem akan kembali dashboard login.

3.6 Rancangan Antar Muka

Rancangan *interface* untuk memberikan representasi visual dari tampilan atau antarmuka pengguna dan juga dapat dijadikan sebagai acuan dalam pengembangan desain antarmuka aplikasi [17]. Berikut di bawah ini hasil dari rancangan *interface* sebagai berikut.

1. Tampilan *Dashboard*



Gambar 7. Tampilan Menu *Dashboard*

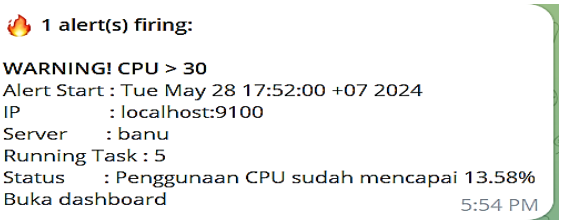
Menu dashboard adalah tampilan utama saat pengguna masuk [18] . Menu dashboard akan menampilkan kepada pengguna tentang informasi list server, menu alerting, menu contact point dan menu notifikasi alert.

2. Tampilan *Data Log*

Gambar 8. Tampilan Menu Data Log

Tampilan menu data log dalam sistem monitoring server dirancang untuk memberikan akses yang jelas dan terorganisir terhadap informasi log yang dihasilkan oleh server[19]. Pada bagian atas menu, biasanya terdapat header yang mencakup nama menu dan opsi navigasi untuk berpindah ke bagian lain dari sistem. Menu ini sering dilengkapi dengan fitur filter dan pencarian, memungkinkan pengguna untuk menyaring log berdasarkan kriteria seperti tanggal, jenis log, atau kata kunci tertentu.

3. Notifikasi *Alert*



Gambar 9. Notifikasi Alert

Notifikasi Alert adalah pemberitahuan real-time kepada administrator melalui aplikasi Telegram. Ketika terjadi peristiwa penting atau masalah pada server, sistem secara otomatis mengirimkan pesan

alert ke saluran Telegram yang telah ditentukan[20]. Notifikasi ini biasanya mencakup informasi kritis seperti jenis masalah, waktu kejadian, dan rincian lainnya yang relevan, memungkinkan administrator untuk segera merespons dan menangani isu tersebut.

3.7 Hasil Pengujian

Pengujian sistem monitoring server berdasarkan resouce server di PT. XYZ Media Indonesia, penulis menggunakan pengujian *blackbox*. Berikut merupakan hasil pengujian sistem:

Tabel 2. Pengujian Blackbox sistem monitoring *server* di PT. Xyz Media Indonesia

No	Menu yang di uji	Item Pengujian	Cara yang Diuji	Hasil	Keterangan
1	Registrasi dan Login	Tombol Registrasi akun	Akun berhasil dibuat	Akun siap digunakan	Selesai
		Tombol Login	Memasukkan *username* dan *password*	Menampilkan menu *dashboard*	Selesai
2	Utama	Tombol *list server*	Menekan tombol list server	Menampilkan list server	Selesai
		Tombol *data log*	Menekan tombol *data log*	Menampilkan *data log*	Selesai
		Tombol *alert rules*	Menekan tombol *alert rules*	Menampilkan konfigurasi *alert rules*	Selesai
		Tombol *contact point*	Menekan tombol *contact point*	Menampilkan menu *contact point*	Selesai
3	*Alert*	Pengiriman notifikasi *alert* dalam keadaan normal	Ultilisasi *server* dalam keadaan normal	Notifikasi *alert* terkirim	Selesai
		Pengiriman notifikasi *alert* dalam keadaan ambang batas	Ultilisasi *server* dalam keadaan melebihi ambang batas	Notifikasi *alert* terkirim	Selesai

Pada tabel 2 diatas didapatkan hasil pengujian terhadapa sistem pada registrasi dan login akun dengan pengujian berupa tombol registrasi akun dan login akun mendapatkan hasil berupa akun berhasil dibuat pada registrasi akun dan masukkan username dan password pada tombol login dengan menampilkan menu dashboard. Selanjutnya pada menu utama dilakukan pengujian pada tombol list server dengan cara menekan tombol list server mendapatkan hasil menampilkan list server , pengujian pada tombol data log dengan cara menekan tombol data log akan mendapatkan hasil menampilkan data resource server dan pengujian tombol contact point didapatkan hasil menampilkan menu konfigurasi contact point . Selanjutnya pengujian pada alert mempunyai 2 keadaan yaitu dalam keadaan normal dan keadaaan darurat hasil dari pengujian ini adalah notifkasi alert yang terkirim ke pengguna .

4. KESIMPULAN

Berdasarkan pembahasan dan hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem monitoring server berbasis prometheus dan grafana telah berhasil dibangun dan diimplementasikan dengan baik. Pengujian langsung oleh karyawan divisi IT menunjukkan bahwa sistem ini mampu mempermudah proses monitoring server secara remote, yang sebelumnya dilakukan secara manual. Sistem ini efektif dalam memberikan informasi terkait aktivitas server yang dipantau, termasuk status penggunaan CPU, memori, koneksi jaringan, dan utilisasi storage. Selain itu, pelaporan melalui telegram terbukti memberikan respon cepat, memungkinkan administrator untuk menangani permasalahan dengan segera. Metode pengujian blackbox yang diterapkan pada sistem menunjukkan bahwa semua fungsi sistem beroperasi dengan baik, menandakan tingkat fungsionalitas yang memuaskan.

REFERENSI

[1] H. Basri, T. S. Wibowo, and S. Saputra, “Sistem Informasi Dan Monitoring Data Berbasis Web Menggunakan Model Waterfall Untuk Meningkatkan Efisiensi Pengolahan Data Di Pt Perseroda PITS BUMD,” *BIN: Bulletin Of Informatics*, vol. 2, no. 2, pp. 175–177, 2024.

[2] M. Rusli, C. M. Usman, M. F. Mulya, and T. W. Widyaningsih, “Aplikasi Sistem Monitoring Server Menggunakan Device Orange Pi Berbasis Web Service Studi Kasus PT. MNC Televisi Indonesia–

MNC Group," *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 5, no. 2, pp. 24–35, 2022.

[3] L. O. Sari, H. A. Suri, E. Safrianti, and F. Jalil, "Rancang Bangun Sistem Monitoring Bandwidth Server pada PT. Industri Kreatif Digital: Design and Build of Server Bandwidth Monitoring System at PT. Industri Kreatif Digital," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 168–179, 2023.

[4] E. A. Pradana, A. A. Putry, and S. Mursidayanti, "Rancang Bangun Media Praktikum Mata Kuliah Sistem Operasi Dengan Kernel Virtual Machine Server Terintegrasi Dengan Sistem Akademik," *Indo-MathEdu Intellectuals Journal*, vol. 4, no. 2, pp. 1237–1248, 2023.

[5] R. M. Febriana, "Implementasi Sistem Monitoring Menggunakan Prometheus Dan Grafana," *Semin. Nas. Telekomun dan Inform*, vol. 13, pp. 164–169, 2020.

[6] D. Rahman, H. Amnur, and I. Rahmayuni, "Monitoring server dengan prometheus dan grafana serta notifikasi telegram," *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 1, no. 4, pp. 133–138, 2020.

[7] M. Z. Amirudin, R. Fahmi, E. Utami, and M. S. Mustafa, "Evaluasi Penggunaan Prometheus dan Grafana Untuk Monitoring Database Mongodb," *Jurnal Informatika Polinema*, vol. 7, no. 2, pp. 43–50, 2021.

[8] S. R. Dira and M. A. F. Ridha, "Monitoring Kubernetes Cluster MenggunakanPrometheus dan Grafana," *ABEC Indonesia*, pp. 345–351, 2023.

[9] S. Wijayanto, F. Fahrullah, D. Mirwansyah, and R. Riyayatsyah, "Implementasi Metode Waterfall Pada Aplikasi Monitoring Cuti Pegawai Dealer Toyota Auto2000 Samarinda," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 2, no. 2, pp. 73–89, 2022.

[10] M. Badrul, "Penerapan Metode Waterfall Untuk Perancangan Sistem Informasi Inventory Pada Toko Keramik Bintang Terang," *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, vol. 8, no. 2, pp. 52–57, 2021.

[11] R. M. Febriana, "Implementasi Sistem Monitoring Menggunakan Prometheus Dan Grafana," *Semin. Nas. Telekomun dan Inform*, vol. 13, pp. 164–169, 2020.

[12] Z. Husen and M. S. Surbakti, *Membangun Server dan Jaringan Komputer dengan Linux Ubuntu*. Syiah Kuala University Press, 2020.

[13] S. R. Dira and M. A. F. Ridha, "Monitoring Kubernetes Cluster MenggunakanPrometheus dan Grafana," *ABEC Indonesia*, pp. 345–351, 2023.

[14] R. M. Febriana, "Implementasi Sistem Monitoring Menggunakan Prometheus Dan Grafana," *Semin. Nas. Telekomun dan Inform*, vol. 13, pp. 164–169, 2020.

[15] M. Y. E. Saputra, S. N. Arief, V. N. Wijayaningrum, and Y. W. Syaifudin, "Real-Time Server Monitoring and Notification System with Prometheus, Grafana, and Telegram Integration," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, IEEE, 2024, pp. 1808–1813.

[16] S. P. Simbolon and R. Maulany, "Perancangan Aplikasi Pendeteksi Jenis-jenis Sampah Berbasis Android: Development of an Android-Based Waste Type Detection," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 926–935, 2024.

[17] N. Fadila, D. Setiawati, A. Wahyono, and M. A. Aziz, "Perancangan Prototype User Interface Sistem Informasi Pusat Data Dokumen Menggunakan Figma," *Jurnal Komputer, Informasi dan Teknologi*, vol. 4, no. 1, p. 22, 2024.

[18] A. Syihabuddin and Z. Abidin, "Sistem Monitoring Dan Evaluasi Nilai Siswa Berbasis Dashboard Berdasarkan Key Performance Indicator (Studi Kasus: Smp Kartika Ii-2 Bandarlampung)," *Jurnal Teknologi Dan Sistem Informasi*, vol. 1, no. 2, pp. 17–25, 2020.

[19] R. Yulvianda and M. Ismail, "Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix dan Grafana," *Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)*, vol. 3, no. 1, pp. 322–329, 2023.

[20] B. Alfiansyah, S. Syaifuddin, and D. Risqiwati, "Pengelompokan Notifikasi Alert Intrusion Detection System Snort Pada Bot Telegram Menggunakan Algoritma K-Means," *Jurnal Repositor*, vol. 2, no. 3, 2020.