**BAB II**

**DESKRIPSI TEORITIK DAN KERANGKA BERFIKIR**

1. **Deskripsi Teoritik**
2. **Definisi Optimalisasi**

Optimalisasi berasal dari kata “optimal” yang berarti terbaik, tertinggi, paling menguntungkan. Optimalilasasi secara sempit adalah tindakan/kegiatan perbaikan dan optimalisasi. Meskipun dalam arti luas, optimalisasi adalah proses pelaksanaan program yang direncanakan untuk mencapai tujuan/sasaran dan mengoptimalkan kinerja (Lestari 2020) dikutip dalam (Zulkifli, 2022).

Optimalisasi merupakan upaya dalam peningkatan kinerja pada suatu sistem maupun program kerja yang berkaitan dengan kepentingan publik sehingga tercapainya hasil dari tujuan dari penyelenggaraan aktivitas tersebut (Febriant Rizaldy et al., 2024).

Menurut Poerdwadarminta (Ali, 2014) optimalisasi adalah hasil yang dicapai sesuai dengan keinginan, jadi optimalisasi merupakan pencapaian hasil sesuai harapan secara efektif dan efisien. Optimalisasi banyak juga diartikan sebagai ukuran dimana semua kebutuhan dapat dipenuhi dari kegiatan-kegiatan yang dilaksanakan (Lucky, 2021).

Pengertian optimalisasi mengacu pada elemen-elemen dari beberapa set alternatif yang bersedia. Dalam kasus paling sederhana, optimalisasi yakni memecahkan masalah-masalah dimana seseorang berusaha meminimalkan atau memaksimalkan fungsi dengan sistematis, memilih menilai variabel. Secara umum, pengertian optimalisasi adalah pencarian nilai terbaik dari yang tersedia dari beberapa fungsi yang diberikan pada suatu konteks (Anjani, 2022).

Berdasarkan definisi tersebut bisa disimpulkan bahwa bahwa Optimalisasi adalah suatu kinerja yang terbaik dalam pemecahan masalah agar mendapatkan hasil yang maksimal.

1. **Definisi *Monitoring***

*Monitoring* merupakan siklus kegiatan yang didalamnya terdapat pengumpulan, peninjauan ulang suatu proses yang sedang diimplementasikan (Fietri Setiawati Sulaeman, 2021).

*Monitoring* (pemenatauan) merupakan sebuah proses penilaian kualitas kinerja sistem dari waktu ke waktu (siklus). Pemantauan ini dilakukan secara berkelanjutan sejalan dengan kegiatan usaha yang mencakup kegiatan sehari-hari. Pengawasan adalah pengendalian yang dilakukan dengan melaksanakan pemeriksaan, penilaian kemampuan, meningkatkan dan menyempurnakan, baik manajemen maupun bidang operasional (Orlando Isakh et al., 2020).

*Monitoring* adalah proses pengumpulan informasi secara berkelanjutan yang bertujuan untuk dapat mengawasi kegiatan yang telah dilakukan guna meningkatkan penyempurnaan tujuan yang ingin di capai (Putri & Zakaria, 2023) dikutip dalam (Misbullah et al., 2023).

*Monitoring* juga didefinisikan sebagai pemantauan yang dapat dijelaskan sebagai kesadaran *(awareness)* tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan kearah tujuan atau menjauh dari itu. Sebuah sistem *monitoring* melakukan proses pengumpulan data mengenai dirinya sendiri dan melakukan analisis terhadap data-data tersebut dengan tujuan untuk memaksimalkan seluruh sumber daya yang dimiliki. Data yang dikumpulkan pada umumnya merupakan data yang *real-time*, baik data yang diperoleh dari sistem yang *hard* *real-time* maupun sistem yang *soft* *real-time* (Rahmat et al., 2022) dikutip dalam (Misbullah et al., 2023).

*Monitoring* menurut (Megawaty & Putra, 2020) merupakan pemantauan dari proses pengumpulan dan analisis informasi yang bertujuan untuk mengetahui dan mengevaluasi atas apa yang telah dihasilkan (Bastomi et al., 2024).

Dari beberapa definisi di atas dapat disimpulkan bahwa *monitoring* merupakan pantauan terhadap suatu kinerja yang nanti nya digunakan sebagai bahan evaluasi kearah yang lebih baik.

1. **Definisi *Server***

*Server* adalah sebuah sistem komputer yang menyediakan jenis layanan *(service)* tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan *(network operating system).* *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat didalamnya, seperti halnya berkas atau alat pencetak *(printer)* dan memberikan akses kepada *workstation* anggota jaringan (ilham efendi, n.d.).

Server atau dalam bahasa Indonesia biasa disebut peladen merupakan suatu sistem komputer yang memiliki layanan khusus berupa penyimpanan data. Data yang disimpan melalui Server berupa informasi dan beragam jenis dokumen yang kompleks. Layanan tersebut ditujukan khusus untuk client yang berkebutuhan dalam menyediakan informasi untuk pengguna atau pengunjungnya (Dicoding Intern, 2020).

*Server* adalah sebuah sistem komputer yang menjalankan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung oleh prosesor yang bersifat *scalable* atau dapat terskalakan dan RAM yang besar,dan juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya contoh seperti halnya berkas atau pencetak, dan memberikan akses kepada stasiun kerja anggota jaringan (Anna Tiffanie, n.d.).

*Server* adalah suatu sistem komputer yang mempunyai layanan khusus sebagai penyimpanan data. Peran *Server* sangat penting dalam mengirim atau menerima data maupun informasi yang tersedia (Amira K, n.d.).

Berdasarkan beberapa definisi di atas dapat disimpulkan bahwa *Server* merupakan suatu sistem komputer yang berfungsi untuk menyimpan data dan menjalankan suatu layanan dalam sebuah jaringan.

1. **Definisi Visualisasi Data**

Visualisasi data adalah teknik untuk mempresentasikan data dalam bentuk grafik atau gambar agar dapat memberikan gambaran yang lebih jelas tentang informasi yang terdapat dalam data (Wahjoerini et al., 2022) dikutip dalam (Ayu Aknesia, 2024).

Visualisasi data adalah proses penyajian data dalam bentuk grafik yang membuat informasi mudah dimengerti, hal ini membantu menjelaskan tentang fakta dan menentukan arah tindakan (Rizki, 2020) dikutip dalam (Septa & Alfia, 2022).

Visualisasi dapat dilakukan dengan menggunakan *dashboard*, dimana teks, pola dan korelasi yang tidak terdeteksi dapat dengan mudah divisualisasikan dengan menggunakan perangkat lunak visualisasi (Asmiatun et al., 2020) dikutip dalam (Septa & Alfia, 2022).

Visualisasi data tidak hanya mengubah data menjadi grafik visual, akan tetapi visualisasi data juga memerlukan perencanaan (Winarno, 2020) dikutip dalam (Septa & Alfia, 2022).

Visualisasi data memanfaatkan kekuatan analitik data dan menambahkan tampilan visual untuk memanfaatkan cara kerja otak kita. Anda mungkin pernah mendengar ungkapan “Sebuah gambar bernilai seribu kata”, tampilan interaktif, bagan dengan kemampuan telusuri dan analisis data geospasial melakukan hal itu dan merupakan beberapa dari banyak cara perusahaan dapat menyajikan data untuk meningkatkan pengambilan keputusan. Misalnya, peta dapat menceritakan kisah yang jauh lebih menarik daripada kata atau angka, dengan penggunaan petunjuk visual yang efektif. Pengambil keputusan organisasi mengandalkan isyarat visual untuk memahami dan memproses sejumlah besar informasi (Mardia et al., 2021) dikutip dalam (Septa & Alfia, 2022).

Berdasarkan beberapa definisi tentang visualisasi data dapat disimpulkan bahwa visualisasi data adalah cara mengolah sebuah data kedalam bentuk yang menarik untuk di lihat dan memudah kan bagi siapa saja yang membacanya, contoh menampilkan data kedalam *dashboard* yang berisi grafik-grafik.

1. **Definisi Prometheus**

Prometheus adalah perangkat lunak pemantauan dan peringatan sistem yang bersifat *open-source* yang awalnya dikembangkan oleh perusahaan bernama SoundCloud. Sejak dimulai pada 2012, banyak perusahaan dan organisasi yang telah mengadopsi Prometheus dan memiliki banyak komunitas pengembang dan pengguna yang sangat aktif. 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 *ex*port*er* yang telah di install dan dapat digunakan untuk memberi peringatan. Grafana atau konsumen *Application Programming Interface* (API) lainnya dapat digunakan untuk memvisualisasikan data yang dikumpulkan (Rahman & Amnur, 2020).

Prometheus adalah *toolkit* pemantauan dan peringatan sistem sumber terbuka yang awalnya dikembangkan oleh perusahaan bernama 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. Saat ini, Prometheus adalah proyek sumber terbuka mandiri dan dikelola secara independen dari perusahaan manapun. Untuk menekankan hal ini, dan untuk menjelaskan struktur tata kelola proyek, Prometheus bergabung dengan *Cloud native Computing foundation* pada tahun 2016 sebagai proyek yang di *host* kedua, setelah Kubernetes. Prometheus mengumpulkan dan menyimpan metriknya sebagai *statistics time series*, yaitu informasi metrik disimpan dengan *timestamp* pada saat direkam, bersamaan dengan pasangan kunci-nilai opsional yang disebut label (Nurrohman, 2024).

Prometheus adalah perangkat lunak berbasis sumber terbuka yang berguna untuk melakukan *monitoring* dan alerting. Salah satu keunggulan Prometheus dibanding perangkat lunak *monitoring* yang lain adalah memiliki banyak metrik (pengukuran) yang dibutuhkan untuk memantau system (Yudhy Kusuma & Oktiawati, 2022).

Berdasarkan beberapa definisi dari Prometheus dapat disimpulkan bahwa Prometheus merupakan suatu perangkat lunak pemantauan dan peringatan terhadap suatu sistem yang bersifat *open source*.

1. **Definisi Grafana**

Grafana adalah analitik sumber terbuka *multi-platform* serta perangkat lunak *web* visualisasi interaktif. Ini menyampaikan bagan, grafik dan peringatan buat *web* saat terhubung ke sumber data yang didukung. Pengguna akhir dapat membuat *dashboard* pemantauan yang kompleks memakai Produsen kueri interaktif. Grafana artinya aplikasi visualisasi dan analisis *open source*. Grafana memungkinkan pengguna memvisualisasikan data dari *database* deret saat menjadi grafik yang praktis dibaca (Nurrohman, 2024).

Grafana adalah perangkat lunak visualisasi dan analitik yang bersifat *open source*. Grafana memungkinkan untuk memvisualisasi- kan, mengingatkan dan menjelajahi metrik disimpan. Alat untuk mengubah data *time series database* (TSDB) menjadi grafik dan visualisasi yang indah. Grafana digunakan untuk menampilkan status *service* yang berjalan pada aplikasi maupun *Server* yang digunakan

(Rahman & Amnur, 2020).

Grafana adalah sebuah *software open source* yang membaca sebuah data *metrics* untuk dibuat menjadi sebuah grafik atau sebuah data tertulis. Grafana sering digunakan untuk melakukan analisis data dan *monitoring*. Grafana mendukung banyak *storage backends* yang berbeda untuk data *time series* (Source Data) (Febriana, 2020).

Grafana adalah perangkat lunak berbasis sumber terbuka yang berfungsi untuk memvisualisasikan data *monitoring* dalam bentuk grafik dan *chart*. Grafana mempunyai dukungan untuk menggunakan berbagai macam tipe *data source* data seperti Graphite, Prometheus, Elasticsearch, OpenTSDB and InfluxDB (Ramadoni, 2021).

Grafana adalah *stack* observabilitas lengkap yang memungkinkan untuk memantau dan menganalisis *metrics*, *logs*, dan *traces*. Grafana memungkinkan untuk melakukan query, memvisualisasikan, membuat peringatan, dan memahami data telemetri di manapun data tersebut disimpan. Grafana mendukung berbagai sumber data seperti Prometheus, Graphite, InfluxDB, ElasticSearch, MySQL, PostreSQL, dll (Yudhy Kusuma & Oktiawati, 2022).

Berdasarkan dari beberapa definisi Grafana dapat disimpulkan bahwa Grafana merupakan perangkat lunak visualisasi data yang memiliki template sehingga pembuatan visualisasi data dapat dibuat dengan cepat.

1. **Definisi Node Exporter**

Node Exporter dirancang untuk mengumpulkan data berasal lingkungan Unix. Itu menunjukkan. metrik terkait perangkat keras dan kernel asal mesin sasaran, contohnya Metrik CPU, memori dan ruang *disk*. Eksportir Node hanya dimaksudkan buat memantau mesin itu sendiri, bukan proses atau layanan individual di dalamnya (Brasil 2018). Pengekspor bisa diunduh berasal halaman unduh Prometheus, dan itu wajib diinstal ke mesin sasaran. Node Exporter berjalan pada *port* 9100 sebagai standarnya. (Github Inc.2021a) dikutip dalam (Nurrohman, 2024)

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 (Rasyidi & Pratama, 2024).

Berdasarkan dari beberapa definisi Node Exporter dapat disimpulkan bahwa Node Exporter merupakan alat pengumpul metrik yang dirancang khusus untuk ekosistem Prometheus, dengan fokus pada pemantauan kinerja dan status sistem operasi pada lingkungan berbasis Unix. Alat ini mengumpulkan data terkait metrik perangkat keras dan kernel, seperti penggunaan CPU, memori, ruang disk, jaringan dan aktivitas disk I/O. Node Exporter difokuskan untuk memantau performa mesin secara keseluruhan, bukan proses atau layanan individual di dalamnya. Sebagai komponen penting dalam manajemen infrastruktur, Node Exporter berjalan pada *port* standar 9100 dan memerlukan instalasi langsung pada mesin target, menjadikannya alat yang esensial dalam mengawasi kesehatan *Server* secara *real-time*.

1. **Definisi Telegram**

Telegram adalah Aplikasi pesan *chatting* yang memungkinkan pengguna untuk mengirimkan pesan *chatting* rahasia yang dienkripsi *end-to-end* sebagai keamanan tambahan. Dengan Telegram dimungkinkan berbagi lebih dari sekedar gambar dan video, tapi Telegram juga memungkinkan Anda mentransfer dokumen atau mengirim lokasi Anda python yang relatif lebih mudah jika dibandingkan harus berhubungan langsung dengan MTProto (Hergika, 2021).

Jadi, pada awal perkembangan dunia bot di Telegram, hampir semua bot dibuat menggunakan telegram-cli dan lua. Bot yang paling terkenal adalah telegram-bot buatan Yago Perez. Bot telegram-cli bekerja layaknya akun pribadi (karena memang ia adalah akun biasa), kita bahkan bisa juga *login* sebagai akun bot telegram-cli ini dan melakukan apa yang dapat dilakukan oleh akun normal (Hergika, 2021).

Manfaat bot ini diamini juga oleh pihak Telegram yang kemudian meluncurkan bot API agar orang banyak dapat membangun bot menggunakan bahasa pemrograman yang mereka kuasai tanpa harus berhubungan dengan telegram-cli atau MTProto. Bot API adalah akun bot, ada hal-hal tertentu yang bisa dilakukan akun normal yang tidak bisa dilakukan akun bot, misal membuat *group*, memasukkan orang ke dalam *group* dan mengeluarkan orang dari *group* (Hergika, 2021).

Berdasarkan dari beberapa definisi Telegram dapat disimpulkan bahwa Telegram adalah aplikasi pesan instan yang mendukung komunikasi aman melalui enkripsi *end-to-end* serta memungkinkan berbagi *file*, dokumen, gambar, video, dan lokasi dengan mudah. Dalam pengembangan bot, Telegram awalnya menggunakan telegram-cli dan lua, di mana bot berfungsi seperti akun pribadi. Namun, Telegram kemudian merilis Bot API untuk mempermudah pengembang membangun bot menggunakan berbagai bahasa pemrograman tanpa harus berinteraksi langsung dengan telegram-cli atau MTProto. Meskipun Bot API memiliki batasan seperti tidak dapat membuat atau mengelola anggota grup, API ini memberikan fleksibilitas tinggi dalam mengembangkan fitur bot yang inovatif dan praktis.

1. **Definisi Ubuntu**

Ubuntu Linux adalah salah satu dari beberapa varian dan juga disebut sebagai distribusi sistem operasi Linux dan merupakan perusahaan Inggris bernama Canonical Ltd. Pada tahun 1994, perusahaan ini didirikan oleh Mark Shuttleworth. (Ikhwan Nurrohman, 2024).

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, Ubuntu *Server* juga memiliki kesamaan dalam konsep *open source* yang memungkinkan pengguna untuk mengakses dan memodifikasi kode sumbernya sesuai dengan kebutuhan perusahaan (Rasyidi & Pratama, 2024).

Ubuntu adalah distribusi Linux yang terdiri dari banyak perangkat lunak bebas dan terbuka dan merupakan versi komersial dari Debian. Saat ini ada beberapa edisi Ubuntu, yang diperkenalkan pada 20 Oktober 2004, termasuk Desktop, *Server* dan *Core* untuk perangkat *Internet of Things* dan robot. Komunitas pengembang dan perusahaan British Canonical Ltd. berkolaborasi untuk membuat Ubuntu. Banyak fitur yang didukung Ubuntu, seperti pembaruan semestral dan dua tahunan dengan dukungan panjang waktu (LTS), didasarkan pada model pengembangan meritokratisnya (Hidayatullah et al., 2024).

Linux Ubuntu OS adalah sistem operasi yang tergolong dalam keluarga Linux. Ubuntu sendiri berasal dari bahasa Zulu dan Xhosa di Afrika Selatan, yang berarti “kemanusiaan terhadap sesama”. Ubuntu dirancang untuk memberikan pengalaman pengguna yang mudah, hemat biaya dan aman. Sistem operasi ini sangat populer dikalangan pengguna Linux dan dikenal karena mudah digunakan dan diinstal (it.telkomuniversity, 2023).

Linux Ubuntu OS didasarkan pada distribusi Debian Linux dan dikembangkan oleh Canonical Ltd, perusahaan yang didirikan oleh entrepreneur Afrika Selatan Mark Shuttleworth. Sistem operasi ini pertama kali dirilis pada tahun 2004 dan sejak itu telah menjadi salah satu distribusi Linux paling populer di dunia (it.telkomuniversity, 2023).

Berdasarkan dari beberapa definisi Ubuntu dapat disimpulkan bahwa Ubuntu adalah distribusi Linux berbasis Debian yang dikembangkan oleh Canonical Ltd., perusahaan yang didirikan oleh Mark Shuttleworth. Dirilis pertama kali pada tahun 2004, Ubuntu bersifat *open source*, memungkinkan pengguna mengakses dan memodifikasi kode sumbernya. Terdiri dari berbagai edisi seperti Desktop, *Server* dan *Core*, Ubuntu dirancang untuk kemudahan, keamanan dan fleksibilitas, menjadikannya salah satu sistem operasi Linux paling populer di dunia.

1. **Definisi SMTP**

SMTP adalah protokol yang berbasis teks dimana protokol ini menyebutkan satu atau lebih penerima email untuk kemudian diverifikasi. Jika penerima email valid, maka email akan segera dikirim. Sendmail adalah Mail Transfer Agent pertama yang mengimplementasikan *port* 25. Mail Transfer Agent yang mengimplementasikan SMTP baik sebagai *client* maupun sebagai *Server*, seperti Exim, IBM Postfix, Qmail dan Microsoft Exchange *Server* (Adiguna, 2020).

SMTP merupakan singkatan dari *Simple Mail Tranfer Protocol*, SMTP merupakan layanan yang dapat digunakan pada proses pengiriman email. Menurut Wawan Putra Perdana cara kerja SMTP yaitu surat elektronik yang akan dikirim berada pada email *client* (komputer pengirim email) kemudian dikirimkan ke SMTP *Server* kemudian dimasukan ke dalam jaringan internet. Kemudian email *client* yang lain (tempat yang dituju dalam pengiriman email) akan mengambil surat elektronik dari jaringan internet melalui *Server* POP3 yang merupakan penyedia layanan email penerima. SMTP muncul karena sistem email yang mengharuskan adanya *Server* untuk penampung, sebelum email tersebut diambil oleh penerima (Adiguna, 2020).

SMTP merupakan protokol yang sederhana, berbasis teks dimana dapat menerima satu atau lebih penerima email yang kemudian akan diverifikasi, SMTP dapat menggunakan *port* 25 dan dapat dihubungkan melalui telnet. Kelebihan dari layanan SMTP, yaitu proses berkirim email dapat dilakukan dengan cepat sehingga dapat dikatakan efisien dan efektif dalam berkirim surat. Selain itu penggunaan SMTP, sederhana. Adapun kekurangan dari layanan ini adalah memungkinkan terjadinya pemalsuan identitas (Adiguna, 2020).

SMTP atau *Simple Mail Transfer Protocol* adalah suatu protokol yang digunakan untuk berkomunikasi antara *Server* untuk mengirimkan email dari *Server* hosting ke *Server* email. Gmail merupakan salah satu layanan email milik Google yang menggunakan SMTP (Rachmat & Arman, 2020).

SMTP adalah protokol yang ber-fungsi sebagai layanan pesan alternatif yang menginformasikan jika terdapat *troubleshoot* atau aktivitas lain seperti penurunan *bandwidth* yang menyebabkan terputusnya koneksi internet (Vindi Eka Safitri et al., 2024).

Berdasarkan dari beberapa definisi SMTP dapat disimpulkan bahwa SMTP *(Simple Mail Transfer Protocol)* adalah protokol berbasis teks yang digunakan untuk mengirimkan email dari pengirim ke penerima melalui *Server*. Protokol ini bekerja dengan memverifikasi alamat email penerima dan meneruskan pesan melalui *Server* SMTP sebelum diambil oleh penerima melalui *Server* lain, seperti POP3. SMTP, yang menggunakan *port* 25 dan dapat diakses melalui telnet, memungkinkan komunikasi antar *Server* secara cepat, efisien dan sederhana, SMTP sering digunakan oleh layanan email populer, seperti Gmail dan juga berfungsi untuk memberikan notifikasi terkait masalah koneksi atau aktivitas jaringan lainnya.

1. **Definisi MobaXterm**

MobaXterm merupakan suatu terminal yang memiliki kinerja yang ditingkatkan pada X *Server* dan satu set perintah Unix (GNU / Cygwin) yang dikemas dalam sebuah *file* .exe tunggal dan *portable* (tanpa proses instalasi) (Kusuma & Indrati, 2024).

Aplikasi MobaXterm merupakan salah satu SSH *client* yang bersifat *open source*. Aplikasi MobaXterm mendukung berbagai tipe koneksi SSH, Rlogin, RDP, VNC, XDMCP, FTP, SFTP dan Serial Sessions. Fitur yang dimiliki lebih memudahkan untuk *remote* perangkat *Server* dari jarak jauh (Aziza, 2024).

Mobaxterm merupakan sebuah *software* untuk komputasi dengan jarak jauh dalam satu aplikasi. Mobaxterm menyediakan fungsi yang dapat disesuaikan untuk administrator programmer, *web*master, IT straff dalam menyediakan semua alat jaringan jarak jauh seperti SSH, X11, RDP, VNC, FTP, MOSH dan perintah unix (Nitiwaluyo, 2021).

Mobaxterm adalah sebuah perangkat lunak yang dirancang untuk menyediakan lingkungan kerja yang nyaman bagi profesional IT, pengembang dan administrator sistem. Perangkat lunak ini menggabungkan berbagai alat yang diperlukan untuk bekerja pada sistem *remote*, seperti terminal emulator, SSH client, dan X11 *Server*, ke dalam satu aplikasi yang intuitif dan mudah digunakan. Dengan Mobaxterm, Kamu bisa mengakses dan mengelola *Server*, transfer *file*, menjalankan aplikasi *remote* dan banyak lagi, semuanya dari satu antarmuka (Adisty C. Putri, 2024).

Berdasarkan dari beberapa definisi MobaXterm dapat disimpulkan bahwa MobaXterm adalah perangkat lunak serbaguna yang menyediakan lingkungan kerja untuk akses dan pengelolaan sistem jarak jauh. Dirancang sebagai terminal emulator dan SSH *client*, MobaXterm mendukung berbagai koneksi seperti SSH, FTP, RDP dan VNC, serta dilengkapi X11 *Server* dan perintah Unix. Dengan fitur yang intuitif dan mudah digunakan, aplikasi ini mempermudah profesional IT, pengembang dan administrator sistem dalam melakukan tugas-tugas komputasi jarak jauh dalam satu antarmuka.

1. **Penelitian Sebelumnya**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Peniliti, tahun, judul penelitian, universitas | Masalah | Metode | Hasil penelitian | Perbandingan |
| 1 | Rahayu Mutiara Febriani, 2020, Implementasi Sistem *Monitoring* Menggunakan Prometheus Dan Grafana, Politeknik Negeri Jakarta | Pemantauan jaringan komputer sangat penting dilakukan agar memudahkan seorang sistem administrator jaringan dalam mengontrol sistem jaringan yang ada. Kebutuhan jaringan yang semakin besar membuat resiko gangguan hingga kerusakan dalam suatu jaringan juga semakin besar. Karena itu seorang sistem administrator diharuskan selalu memantau seluruh sistem jaringan. Ada beberapa langkah yang dilakukan saat melakukan perancangan sistem *monitoring* yang diinginkan. Prometheus dan Grafana dapat membantu membuat sistem *monitoring* jaringan yang mudah dimengerti dan dapat membantu seorang sistem administrator. Prometheus akan mengambil data dari *protocol* SNMP dan memproses data tersebut menggunakan exporter. Untuk melengkapi *system monitoring* ini diperlukan visualisasi. Grafana dapat membuat visualisasi dari data yang telah diproses. Hasil penelitian ini menunjukan *device* yang terhubung didalam jaringan dapat ditampilkan grafana | System Development Life Cycle (SDLC). | Dari hasil penelitian yang dilakukan dapat diperoleh sebuah kesimpulan bahwa adanya sistem *monitoring* jaringan menggunakan Prometheus dan Grafana, dapat membantu sistem administrator jaringan untuk mengetahui kondisi jaringan yang ada. Obyek yang dapat dimonitor pada sistem yang berjalan seperti, interface yang aktif, penggunaan CPU, penggunaan Memori dan *network traffic* serta jumlah perangkat yang terhubung dalam jaringan. Dengan adanya sistem *monitoring* ini maka sistem administrator tidak perlu mengecek keadaan sistem secara manual satu persatu sehingga memberikan efektifitas kerja bagi admin | Penelitian sebelumnya menggunakan Prometheus dan Grafana untuk memantau jaringan secara *real-time*, dengan fitur pemantauan interface aktif, penggunaan CPU, memori, *traffic* jaringan, dan jumlah perangkat terhubung. Sistem ini meningkatkan efisiensi kerja dengan mengurangi pengecekan manual, namun tidak dilengkapi fitur notifikasi otomatis atau pengelolaan akses pengguna. Sebagai perbandingan, sistem yang akan dikembangkan menambahkan notifikasi otomatis melalui Telegram untuk peringatan *real-time*, fitur pengunduhan laporan *resource* untuk analisis dan dokumentasi, serta pembagian akses pengguna yang meningkatkan keamanan dan fleksibilitas. Hal ini menjadikan sistem yang dikembangkan lebih unggul dalam efisiensi dan fungsionalitas. |
| 2 | Dede Rahman, Hidra Amnur, Indri Rahmayuni, 2020, *Monitoring* *Server* Dengan Prometheus Dan Grafana Serta Notifikasi Telegram, Politeknik Negeri Padang. | Administrator jaringan adalah orang yang bertanggung jawab dalam mengelola jaringan komputer, Salah satu tugas dari seorang administrator jaringan adalah *monitoring* jaringan karena kesalahan pada *Server* dan *service* yang berjalan di *Server* tidak dapat diketahui kapan terjadinya. Hal ini akan menyulitkan jika administrator berada jauh dari pusat pengawasan. Situasi ini dapat berakibat fatal jika terjadi kesalahan pada *Server* atau tiba-tiba *service* dari *Server* tidak berjalan sebagaimana mestinya tanpa diketahui karena administrator jaringan sedang tidak ada di tempat pengawasan. Dengan memanfaatkan prometheus dan grafana dengan sistem notifikasi telegram yang menggunakan *Server* berbasis Linux, administrator jaringan dapat mendapatkan informasi kondisi *Server* dan *service* yang berjalan di *Server* secara cepat. Hasil yang didapatkan adalah sebuah sistem yang mampu membantu administrator jaringan dalam melakukan *monitoring* *Server* dan *service* yang sedang berjalan di *Server* kapan saja dan dimana saja untuk menjaga *Server* dan *service* yang berjalan pada *Server* agar tetap stabil walaupun administrator jaringan sedang tidak di pusat pengawasan | Prometheus dan Grafana | *Monitoring* *Server* berhasil dilakukan dengan menggunakan prometheus dan grafana terhadap *Server*. *Monitoring* *Server* bekerja dengan baik dan sesuai dengan harapan. Sistem operasi yang digunakan pada *monitoring* *Server* adalah Ubuntu *Server* dengan versi 18.04. Sistem yang dibangun memberikan notifikasi atau pemberitahuan kepada admin apabila CPU, memori ataupun *service* apache dan MYSQL ada yang mati. Alert di grafana akan selalu mengirimkan pemberitahuan pada telegram apabila kondisi *Server* melewati batas yang telah di tentukan atau apabila *service* dari apache maupun mysql mati. *Flavor* yang bisa digunakan berdasarkan spesifikasi VCPU, RAM dan Disk sesuai dengan pemilihan spesifikasi minimal untuk membuat *instance*. Diharapkan untuk pengembangan selanjutnya sistem *monitoring* *Server* dapat memberikan *alert* ke aplikasi mobile yang dibuat sendiri. untuk pengembangan selanjutnya, implementasi *monitoring* *Server* dengan prometheus dan grafana memiliki tempat implementasi khusus. Diharapkan ada pengembangan dari *monitoring* *Server* dengan prometheus dan grafana untuk *monitoring* service lainnya seperti SSH service , mail *service* , jenkins , kubernetes dan lain – lain | Pada penelitian sebelumnya, sistem *monitoring Server* menggunakan Prometheus dan Grafana telah berhasil diterapkan untuk memantau kondisi *Server*, dengan dukungan sistem operasi Ubuntu *Server* 18.04. Sistem ini mampu memberikan notifikasi melalui Telegram jika terjadi anomali seperti CPU atau memori melebihi batas, atau jika layanan Apache dan MySQL mati. Sistem ini juga menggunakan *alert* pada Grafana untuk mengirimkan pemberitahuan secara *real-time*, dengan spesifikasi minimal instance berdasarkan VCPU, RAM, dan disk. Meski demikian, pengembangan selanjutnya diarahkan pada pembuatan aplikasi mobile untuk notifikasi dan perluasan cakupan monitoring pada layanan lain seperti SSH, mail *service*, Jenkins, dan Kubernetes. Sebagai perbandingan, sistem yang akan dikembangkan memiliki beberapa keunggulan tambahan. Selain mendukung pemantauan *resource Server* serta fitur notifikasi otomatis melalui integrasi Telegram untuk memastikan administrator mendapatkan peringatan *real-time*. Sistem ini juga menyediakan fitur pengunduhan laporan *resource* untuk analisis lebih lanjut, yang tidak disediakan dalam penelitian sebelumnya. Selain itu, sistem ini menawarkan pembagian akses pengguna, memungkinkan pengelolaan hak akses sesuai peran untuk meningkatkan keamanan dan fleksibilitas. Dengan fitur-fitur ini, sistem yang dikembangkan tidak hanya mencakup kebutuhan *monitoring* *Server* dasar, tetapi juga menghadirkan efisiensi, skalabilitas, dan keamanan yang lebih baik dibandingkan sistem pada penelitian sebelumnya. |
| 3 | Banu Rasyidi, Firman Pratama, 2024, Sistem *Monitoring Server* di PT. XYZ Media Indonesia Berbasis Grafana dan Prometheus, Universitas Pamulang. | *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 | Prometheus dan Grafana | 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. | Penelitian sebelumnya berhasil mengembangkan sistem monitoring *Server* berbasis Prometheus dan Grafana yang mempermudah proses pemantauan *Server* secara *remote*, terutama melalui pelaporan cepat via Telegram. Sistem ini efektif dalam menampilkan informasi status CPU, memori, koneksi jaringan, dan storage, serta telah diuji menggunakan metode *blackbox* dengan hasil memuaskan. Namun, sistem ini belum mencakup fitur tambahan seperti pengelolaan akses pengguna, pengunduhan laporan *resource*, atau dukungan untuk analisis lebih lanjut. Sebagai perbandingan, sistem yang akan dikembangkan menawarkan fitur yang lebih lengkap dan fleksibel. Selain mendukung *monitoring resource Server* dan notifikasi otomatis melalui Telegram, sistem ini memungkinkan pengunduhan laporan *resource* untuk analisis mendalam dan dokumentasi. Fitur pembagian akses pengguna juga ditambahkan untuk meningkatkan keamanan dan fleksibilitas pengelolaan. Dengan keunggulan ini, sistem yang akan dikembangkan dirancang untuk memberikan efisiensi, skalabilitas, dan fungsionalitas yang lebih tinggi dibandingkan penelitian sebelumnya. |
| 4 | Sinatria Banyu Adil, Yos Richard Beeh, 2024, Implementasi *Monitoring* Sistem Perusahaan *On-Premises* dan *Cloud* Menggunakan Teknologi Jenkins, Universitas Kristen Satya Wacana. | Penelitian ini mengimplementasikan Jenkins untuk memantau dan mengelola proses job, serta integrasinya dengan infrastruktur *on-premises* dan *cloud* di PT. AAA. *Compute Engine* digunakan sebagai *Server* utama untuk menjalankan Jenkins. Tahapan meliputi konfigurasi *Server* *Compute Engine*, instalasi plugin penting, dan penggunaan *library* Python. Hasilnya menunjukkan Jenkins berhasil dalam menjalankan proses job antara sistem *on-premises* dan *cloud*. Email *Extension Plugin* digunakan untuk memantau proses job dan memberikan notifikasi ke PIC jika terjadi kesalahan. Pengelolaan akses diterapkan melalui *Role-based Authorization Strategy*. Pengujian dilakukan menggunakan *blackbox* testing dengan tingkat keberhasilan 100%. Diharapkan implementasi ini dapat meningkatkan efisiensi dan efektivitas manajemen proses job di PT. AAA. | Teknologi Jenkins | Berdasarkan hasil penelitian, penerapan Jenkins dalam pemantauan dan otomatisasi proses job sinkronisasi antara *database on-premises* dan *cloud* telah berhasil dilakukan oleh PT. AAA. Dengan menggunakan *Compute Engine* sebagai *Server* utama, konfigurasi *Server* yang optimal, instalasi *plugin* dan *library* yang diperlukan, serta pembuatan jenkins *file*, PT. AAA berhasil memantau dan mengelola proses job dengan efisien. | Penelitian sebelumnya menunjukkan keberhasilan penerapan Jenkins dalam pemantauan dan otomatisasi proses sinkronisasi antara database *on-premises* dan *cloud* di PT. AAA. Dengan memanfaatkan *Compute Engine* sebagai *Server* utama, optimalisasi konfigurasi *Server*, instalasi plugin, serta pembuatan jenkins *file*, sistem tersebut memungkinkan pengelolaan proses job secara efisien. Namun, sistem ini terbatas pada pemantauan dan otomatisasi sinkronisasi database tanpa menyediakan notifikasi *real-time* atau fitur pelaporan yang lebih mendalam. Sebagai pembanding, sistem yang akan dikembangkan menawarkan fungsionalitas yang lebih luas dan mendalam. Selain mendukung monitoring *resource* dan notifikasi otomatis melalui Telegram, sistem ini memungkinkan pengunduhan laporan *resource* untuk analisis lebih lanjut serta pembagian akses pengguna yang lebih terstruktur untuk meningkatkan keamanan dan fleksibilitas pengelolaan. Dengan demikian, sistem yang akan dikembangkan memberikan efisiensi, fleksibilitas, dan fungsionalitas yang lebih tinggi dibandingkan sistem yang dijelaskan pada penelitian sebelumnya. |
| 5 | Saleh Dwiyatno, Edy Rakhmat, Oki Gustiawan, 2020, Implementasi Virtualisasi *Server* Berbasis Docker Container, Universitas Banten Jaya, Universitas Serang Raya. | Docker adalah sebuah aplikasi yang berbasiskan teknologi *open source* yang memungkinkan *Developer* atau siapapun untuk membuat, menjalankan, melakukan percobaan dan meluncurkan aplikasi di dalam sebuah container. Docker membuat proses pemaketan aplikasi bersama komponennya secara cepat dalam sebuah container yang terisolasi, sehingga dapat dijalankan dalam infrastruktur lokal tanpa melakukan perbuahan konfigurasi pada container. Docker juga sangat ringan dan cepat jika dibandingkan dengan mesin *virtual* yang berbasis *hypervisor*. SMK Negeri 1 Rangkasbitung mempunyai beberapa *Server* yang mana didalamnya sudah menampung beberapa aplikasi *web*. *Server* tersebut berbasis *hypervisor* sehingga membutuhkan *resource* besar. Karena setiap VM menjalankan *guest OS* beserta kernelnya sendiri terpisah dari *host*. Oleh karena itu, dilakukan implementasi virtualisasi berbasis docker container supaya dapat meningkatkan efektifitas dalam penggunaan sumber daya CPU dan memori pada *Server*. Pengujian dilakukan dengan cara melakukan variasi banyaknya jumlah user request yang berbeda pada masing-masing container menggunakan perangkat lunak apache jmeter | Survey Literatur | Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan, yaitu: Dengan adanya penerapan virtualisasi *Server* berbasis docker container pada Ubuntu 18.04 LTS, dapat menjamin *web* menjadi stabil. Karena pada docker container memastikan aplikasi dan sumber daya yang terisolasi serta terpisah sehingga para penggunanya dapat menyesuaikan kebutuhan di setiap aplikasi tanpa perlu mempengaruhi konfigurasi pada aplikasi yang lain, Adanya pemanfaatan docker container pada perancangan *Server* dapat memanfaatkan *hardware* yang ada untuk digunakan secara maksimal. Karena pada docker container dimana kernel yang digunakan adalah dari bagian sistem operasi *host* nya sendiri, sehingga tidak membebani kinerja dari *Server* *host*. | Penelitian sebelumnya berhasil menunjukkan bahwa penerapan virtualisasi *Server* berbasis Docker container pada Ubuntu 18.04 LTS mampu meningkatkan stabilitas aplikasi *web*. Docker container memungkinkan isolasi aplikasi dan sumber daya, sehingga konfigurasi setiap aplikasi dapat dikelola tanpa saling memengaruhi. Selain itu, Docker memungkinkan pemanfaatan *hardware* secara maksimal dengan menggunakan kernel dari sistem operasi *host*, yang mengurangi beban pada *Server host*. Sebagai perbandingan, sistem yang akan dikembangkan tidak hanya mendukung pengelolaan dan stabilitas *Server* melalui *monitoring resource*, tetapi juga menawarkan fitur-fitur tambahan untuk meningkatkan efisiensi dan fleksibilitas pengelolaan. Fitur notifikasi otomatis melalui Telegram memberikan peringatan *real-time* jika terjadi masalah, sementara kemampuan untuk mengunduh laporan *resource* mendukung analisis dan dokumentasi yang lebih baik. Selain itu, sistem ini menyediakan pembagian akses pengguna untuk meningkatkan keamanan dan pengelolaan yang lebih terstruktur. Dengan demikian, sistem yang akan dikembangkan mencakup keunggulan virtualisasi sekaligus menambahkan fitur yang lebih kaya untuk memenuhi kebutuhan pengelolaan *Server* yang lebih komprehensif. |

1. **Kerangka Berfikir**

Berdasarkan permasalahan yang ada, kerangka berpikir dapat disusun sebagai berikut:

|  |
| --- |
| **Identifikasi Masalah** |
| 1. Pengelolaan Server di PT Concord Consulting Indonesia masih dilakukan secara manual menggunakan perintah Linux untuk memantau sumber daya Server. 2. Tidak adanya notifikasi otomatis yang terhubung ke tim *Developer* dan *IT Operation* sehingga memperlambat respons terhadap gangguan. 3. Data monitoring yang ada hanya dalam bentuk teks, membuatnya sulit untuk dipahami dengan cepat serta menghambat pengambilan keputusan yang tepat waktu. 4. Report server masih dilakukan secara manual menggunakan Excel, dan data report server belum bersifat real-time, karena hanya dihasilkan per hari, sehingga mempersulit analisis gangguan pada server secara cepat dan akurat. |

|  |
| --- |
| **Teknik Pemecahan Masalah** |
| 1. Menggunakan Node Exporter untuk mengumpulkan data metrik *Server* secara *real-time.* 2. Menggunakan Prometheus untuk menyimpan data metrik *Server* secara *real-time.* 3. Menggunakan Grafana untuk membuat visualisasi data metrik yang informatif. 4. Menggunakan Telegram sebagai media notifikasi otomatis untuk memberi peringatan jika terjadi gangguan. |

|  |
| --- |
| **Solusi** |
| Dari beberapa identifikasi masalah yang ada, solusi yang diusulkan dapat membantu mengatasi kendala dan kelemahan sistem *monitoring Server* yang saat ini berjalan di PT. Concord Consulting Indonesia, serta mendukung operasional perusahaan secara lebih efisien dan berkelanjutan. |

|  |
| --- |
| **Hasil** |
| Dirancangnya *dashboard monitoring* berbasis *web* yang berfungsi untuk menampilkan sumber daya *Server* secara *real-time* dengan visualisasi yang mudah dipahami. Selain itu, dilengkapi dengan fitur notifikasi otomatis yang dapat langsung terhubung ke tim *Developer* dan *IT Operation*, sehingga memungkinkan respons cepat terhadap masalah yang terdeteksi. |