

# **OPTIMALISASI *MONITORING SERVER* MELALUI VISUALISASI DATA MENGGUNAKAN PROMETHEUS DAN GRAFANA DI PT. CONCORD CONSULTING INDONESIA**

## **PROPOSAL SKRIPSI**

Diajukan Untuk Memenuhi Salah Satu Syarat Menyelesaikan Pendidikan Program  
Sarjana (S1) Dan Memperoleh Gelar Sarjana Komputer (S.Kom)



Oleh :

NAMA : NANANG WAHYUDI  
NPM : 1101211060  
PROGRAM STUDI : TEKNIK INFORMATIKA  
JENJANG PENDIDIKAN : STRATA 1 (S1)

**FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI TEKNIK INFORMATIKA  
UNIVERSITAS BANTEN JAYA  
JANUARI 2025**

## **PENGESAHAN PEMBIMBING**

NAMA : NANANG WAHYUDI  
NPM : 1101211060  
PROGRAM STUDI : TEKNIK INFORMATIKA  
JENJANG PENDIDIKAN : STRATA 1 (S1)

Judul Proposal Skripsi : **OPTIMALISASI *MONITORING SERVER*  
MELALUI VISUALISASI DATA  
MENGUNAKAN PROMETHEUS DAN  
GRAFANA DI PT. CONCORD  
CONSULTING INDONESIA**

Disetujui, untuk diuji dan dipertahankan dalam seminar proposal.

Serang , 01 Februari 2025

Dosen Pembimbing

Irma Yunita R, S.Si., M.TI

NIDN: 0421067606

## **PENGESAHAN DEKAN DAN KETUA PROGRAM STUDI**

NAMA : NANANG WAHYUDI  
NPM : 1101211060  
PROGRAM STUDI : TEKNIK INFORMATIKA  
JENJANG PENDIDIKAN : STRATA I (S-1)

Judul Proposal Skripsi : **OPTIMALISASI *MONITORING SERVER*  
MELALUI VISUALISASI DATA  
MENGUNAKAN PROMETHEUS DAN  
GRAFANA DI PT. CONCORD  
CONSULTING INDONESIA**

Disetujui, untuk diuji dan dipertahankan dalam seminar proposal

Serang, 03 Februari 2025

Dekan Fakultas

Ketua Program Studi

Sutanto, S.Kom., M.Kom

NIDN: 0014047601

Ely Nuryani, S.Kom., M.TI

NIDN: 0416078411

**NANANG WAHYUDI “OPTIMALISASI *MONITORING SERVER* MELALUI VISUALISASI DATA MENGGUNAKAN PROMETHEUS DAN GRAFANA DI PT. CONCORD CONSULTING INDONESIA”** Jurusan / Program Studi Teknik Informatika, Jenjang Pendidikan Strata 1 (S1), Proposal Skripsi Januari 2025. xii + 74 Halaman, 34 Gambar, 4 Lampiran, 4 Tabel

---

**ABSTRAK**

Perkembangan Teknologi Informasi dan Komunikasi (TIK) yang pesat telah membawa dampak signifikan pada berbagai sektor kehidupan, termasuk dunia kerja. Salah satu tantangan utama yang dihadapi perusahaan adalah pengelolaan dan pemantauan infrastruktur *server* untuk memastikan ketersediaan dan keandalan layanan. PT. Concord Consulting Indonesia, yang sangat bergantung pada infrastruktur *server* untuk mendukung layanan konsultasi risiko bisnis, masih mengandalkan metode manual dalam memantau sumber daya *server*. Hal ini menyebabkan kesulitan dalam mendeteksi masalah secara cepat dan *real-time* serta ketidakmampuan menghasilkan laporan yang informatif secara otomatis. Penelitian ini bertujuan untuk mengembangkan sistem *monitoring server* berbasis *web* dengan menggunakan Prometheus dan Grafana untuk memantau kondisi *server* secara *real-time*, memberikan visualisasi data yang mudah dipahami, serta menyediakan fitur notifikasi otomatis untuk memudahkan respons terhadap masalah. Selain itu, sistem ini dilengkapi dengan fitur pembuatan laporan kondisi *server* secara *real-time* yang dapat mempercepat proses analisis gangguan. Diharapkan, solusi ini dapat meningkatkan keandalan *server*, mengurangi risiko *downtime* dan mendukung operasional perusahaan secara optimal.

**Kata kunci:** Grafana, *Monitoring Server*, Prometheus, Visualisasi Data

**NANANG WAHYUDI “OPTIMIZATION OF SERVER MONITORING THROUGH DATA VISUALIZATION USING PROMETHEUS AND GRAFANA AT PT. CONCORD CONSULTING INDONESIA”** *Department/Study Program Informatics Engineering, Bachelor’s Degree (S1), Thesis Proposal, January 2025. xii + 74 Pages, 34 Images, 4 Appendices, 4 Tables*

---

**ABSTRACT**

*The rapid development of Information and Communication Technology (ICT) has significantly impacted various aspects of life, especially in the workplace. One of the major challenges faced by companies is the management and monitoring of server infrastructure to ensure the availability and reliability of services. PT. Concord Consulting Indonesia, which heavily relies on its server infrastructure to support business risk consulting services, still relies on manual methods to monitor server resources. This leads to difficulties in quickly and accurately detecting issues and an inability to automatically generate informative reports. This research aims to develop a web-based server monitoring system using Prometheus and Grafana to monitor server conditions in real-time, provide easy-to-understand data visualization, and offer automatic notification features to facilitate rapid responses to problems. Additionally, the system is equipped with a real-time server status reporting feature that accelerates the analysis of disruptions. It is expected that this solution will enhance server reliability, reduce downtime risks and optimally support the company's operations.*

**Keywords:** *Data Visualization, Grafana, Monitoring Server, Prometheus*

## **KATA PENGANTAR**

Puji dan syukur dipanjatkan kehadirat Allah SWT, yang telah memberikan Rahmat dan Hidayah-Nya. Sholawat serta salam semoga selalu tercurah kepada Nabi Muhammad SAW, beserta keluarga, sahabat dan umatnya, sampai akhir zaman. Aamiin. Bahwa berkat izin-Nya penyusunan proposal skripsi ini dapat diselesaikan.

Mengingat akan keterbatasan pengetahuan, kemampuan dan pengalaman yang dimiliki maka disadari sepenuhnya bahwa Proposal Skripsi ini masih jauh dari sempurna. Oleh karena itu, sangat diharapkan segala bentuk saran dan masukan bahkan kritik yang membangun dari berbagai pihak.

Pada kesempatan ini, ucapan terimakasih kepada seluruh pihak yang telah memberikan bimbingan serta dukungan dalam penyelesaian Proposal Skripsi ini terutama kepada yth:

1. Rektor Universitas Banten Jaya Bapak Prof. Drs. Muhammad Syadeli Hanafi, M.Pd, selaku Rektor Universitas Banten Jaya.
2. Dekan Fakultas Ilmu Komputer Bapak Sutanto, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer.
3. Kaprodi Teknik Informatika Ibu Ely Nuryani, S.Kom., M.TI, selaku Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Banten Jaya.
4. Dosen Pembimbing Proposal Ibu Irma Yunita R, S.Si., M.TI, selaku dosen pembimbing Proposal Skripsi Universitas Banten Jaya yang telah menyediakan waktu, pikiran dan bimbingannya. Berkat arahan yang diberikan oleh beliau maka penyusunan Laporan Proposal Skripsi ini dengan baik hingga selesai.
5. Keluarga, sahabat dan teman-teman seperjuangan yang telah memberikan doa, semangat, masukan dan motivasi untuk menyelesaikan Proposal Skripsi ini.
6. Serta kepada seluruh pihak yang tidak dapat disebutkan satu-persatu, yang telah memberikan dukungan dalam penyusunan Proposal Skripsi.

Akhir kata diharapkan Proposal Skripsi ini dapat bermanfaat bagi para pembaca terutama Mahasiswa Universitas Banten Jaya. Semoga amal dan budi baik

semua pihak mendapat pahala dari Allah SWT. Sekali lagi ucapan terimakasih sebesar-besarnya kepada semua pihak yang telah membantu.

Serang, 01 Februari 2025

Nanang Wahyudi

NPM: 1101211060

## DAFTAR GAMBAR

Gambar 3. 1 Tahapan Metode <i>Waterfall</i> .....	38
Gambar 3. 2 <i>Use Case Diagram</i> .....	40
Gambar 3. 3 Kelola <i>Dashboard</i> .....	41
Gambar 3. 4 Kelola Data CPU .....	42
Gambar 3. 5 Kelola Data <i>Memory</i> .....	43
Gambar 3. 6 Kelola Data <i>Network</i> .....	44
Gambar 3. 7 Kelola Data <i>Storage</i> .....	45
Gambar 3. 8 Kelola Data <i>Sources</i> .....	46
Gambar 3. 9 Kelola Akses Pengguna.....	47
Gambar 3. 10 Kelola <i>Alert Rules</i> .....	48
Gambar 3. 11 Kelola <i>Notification Policies</i> .....	49
Gambar 3. 12 Kelola <i>Contact Points</i> .....	50
Gambar 3. 13 <i>Notification Alert</i> .....	51
Gambar 3. 14 View Informasi <i>Server</i> .....	52
Gambar 3. 15 View <i>Alert Rules</i> .....	53
Gambar 3. 16 View <i>Notification Policies</i> .....	54
Gambar 3. 17 View <i>Contact Points</i> .....	55
Gambar 3. 18 <i>Download Report Resource Server</i> .....	56
Gambar 3. 19 Struktur Menu <i>IT Operation</i> .....	57
Gambar 3. 20 Struktur Menu Kepala <i>IT Operation</i> dan <i>Developer</i> .....	58
Gambar 3. 21 Halaman <i>Dashboards</i> .....	59
Gambar 3. 22 Halaman Informasi <i>Server</i> .....	59
Gambar 3. 23 Halaman Akses Pengguna.....	60
Gambar 3. 24 Halaman Data <i>Sources</i> .....	60
Gambar 3. 25 Halaman <i>Alert Rules</i> .....	61
Gambar 3. 26 Halaman <i>Notification Policies</i> .....	61
Gambar 3. 27 Halaman <i>Contact Points</i> .....	62
Gambar 3. 28 Halaman Tambah atau Edit <i>Contact Points</i> .....	62
Gambar 3. 29 Halaman Tambah atau Edit <i>Alert Rules</i> .....	63



Gambar 3. 30 Modal Edit <i>Notification Policies</i> .....	64
Gambar 3. 31 Halaman Tambah atau Edit Data <i>Sources</i> .....	64
Gambar 3. 32 Halaman <i>Invite Users</i> .....	65
Gambar 3. 33 Halaman Tambah dan Edit <i>Panel</i> .....	65
Gambar 3. 34 Halaman <i>Download Report Resource</i> .....	66

## DAFTAR TABEL

Tabel 1. 1 Timeline Proposal Skripsi .....	7
Tabel 2. 1 Simbol <i>Use Case Diagram</i> .....	24
Tabel 2. 2 Simbol <i>Activity Diagram</i> .....	26
Tabel 2. 3 Penelitian Sebelumnya .....	28

## DAFTAR ISI

PENGESAHAN PEMBIMBING.....	ii
PENGESAHAN DEKAN DAN KETUA PROGRAM STUDI .....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
DAFTAR GAMBAR .....	viii
DAFTAR TABEL.....	x
DAFTAR ISI.....	xi
BAB I PENDAHULUAN .....	1
A. Latar Belakang Masalah.....	1
B. Identifikasi Masalah .....	3
C. Pembatasan Masalah .....	3
D. Perumusan Masalah .....	4
E. Tujuan dan Manfaat Penelitian .....	4
a. Tujuan Penelitian .....	4
b. Manfaat Penelitian .....	5
F. Sistematika Penulisan .....	6
BAB II DESKRIPSI TEORITIK DAN KERANGKA BERFIKIR.....	8
A. Deskripsi Teoritik.....	8
A.1. Definisi Optimalisasi.....	8
A.2. Definisi <i>Monitoring</i> .....	9
A.3. Definisi <i>Server</i> .....	10
A.4. Definisi Visualisasi Data.....	11
A.5. Definisi Prometheus .....	12
A.6. Definisi Grafana .....	14
A.7. Definisi Node Exporter .....	15
A.8. Definisi Telegram.....	16
A.9. Definisi Ubuntu .....	17
A.10. Definisi SMTP .....	19

A.11. Definisi MobaXterm .....	20
A.12. Definisi <i>Unified Modeling Language</i> (UML) .....	22
B. Penelitian Sebelumnya .....	28
C. Kerangka Berfikir.....	35
BAB III METODOLOGI PENELITIAN.....	37
A. Metodologi Pengembangan Sistem.....	37
A.1. Metode Pengumpulan Data .....	37
A.2. Metode Pengembangan Sistem .....	37
B. Perancangan Sistem .....	40
B.1. Pemodelan .....	40
B.2. Desain Antarmuka Pemakai ( <i>User Interface Design</i> ).....	57
DAFTAR PUSTAKA .....	67
DAFTAR LAMPIRAN.....	71
Lampiran A : Surat Keterangan Riset .....	71
Lampiran B : Kehadiran Bimbingan .....	72
Lampiran C : Daftar Riwayat Hidup .....	73
Lampiran D : Dokumentasi Riset .....	74

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang Masalah**

Perkembangan Teknologi Informasi dan Komunikasi (TIK) yang semakin pesat telah memberikan dampak signifikan terhadap berbagai aspek kehidupan, terutama dalam dunia kerja. Teknologi menjadi alat bantu yang sangat penting untuk meningkatkan efisiensi dan produktivitas, khususnya dalam pengolahan informasi. Namun, pesatnya perkembangan ini juga diiringi dengan tantangan baru, yaitu memastikan ketersediaan dan keandalan sistem aplikasi. Ketika sebuah aplikasi mengalami gangguan atau tidak dapat diakses, hal ini dapat menyebabkan kerugian besar bagi perusahaan maupun instansi.

Salah satu komponen utama yang mendukung kelancaran aplikasi digital adalah *server*. *Server* berperan penting dalam memenuhi permintaan pengguna (*client*) dan mendukung berbagai layanan berbasis jaringan. Namun, *server* sering menghadapi gangguan yang disebabkan oleh keterbatasan sumber daya (*resource*), seperti CPU, memori, atau penyimpanan. Gangguan ini dapat menyebabkan layanan berhenti secara tiba-tiba karena *kernel* sistem operasi mengambil langkah untuk menonaktifkan layanan yang menghabiskan terlalu banyak sumber daya. Sayangnya, banyak perusahaan maupun instansi masih mengandalkan metode pemantauan manual untuk mengelola *server*, sehingga sulit mendeteksi masalah secara cepat dan *real-time*.

Kondisi ini juga dialami oleh PT. Concord Consulting Indonesia, yang sangat bergantung pada infrastruktur *server* untuk mendukung layanan konsultasi risiko bisnisnya. Pengelolaan *server* di perusahaan ini masih dilakukan secara manual menggunakan perintah Linux untuk memantau sumber daya seperti CPU, memori dan penyimpanan. Pendekatan ini kurang efisien dalam mendeteksi masalah seperti penyimpanan penuh atau CPU

*overload*. Selain itu, pembuatan laporan kondisi *server* di PT. Concord Consulting Indonesia masih dilakukan secara manual menggunakan Excel dan hanya dihasilkan per hari. Hal ini menyebabkan data laporan tidak bersifat *real-time* sehingga mempersulit analisis gangguan yang terjadi pada *server*.

Berbagai penelitian menunjukkan bahwa Prometheus dan Grafana dapat menjadi solusi *monitoring server* yang efisien. Prometheus mengumpulkan data metrik *server* secara *real-time*, sementara Grafana menyajikan visualisasi data yang informatif. Namun, banyak implementasi serupa belum dilengkapi dengan pembuatan laporan kondisi *server* secara *real-time* dan notifikasi otomatis untuk mempercepat respons terhadap masalah.

Penelitian ini bertujuan untuk mengembangkan sistem *monitoring server* berbasis *web* di PT. Concord Consulting Indonesia dengan memanfaatkan Prometheus dan Grafana. Sistem ini dirancang untuk mendukung pemantauan *server* secara *real-time*, menyediakan visualisasi data yang mudah dipahami, serta dilengkapi dengan fitur notifikasi yang terhubung langsung ke setiap *Developer* dan *IT Operation*. Selain itu, sistem ini juga diharapkan dapat menyediakan fitur pembuatan laporan kondisi *server* secara *real-time* untuk mempermudah proses analisis gangguan dan meningkatkan efisiensi pengelolaan data *server*. Solusi ini diharapkan meningkatkan keandalan *server*, mengurangi risiko *downtime* dan mendukung operasional perusahaan secara optimal. Sistem yang diusulkan juga diharapkan menjadi solusi yang efisien dan inovatif dalam mengelola infrastruktur *server* di perusahaan.

Oleh karena itu, judul yang diangkat dalam penelitian ini adalah **"OPTIMALISASI MONITORING SERVER MELALUI VISUALISASI DATA MENGGUNAKAN PROMETHEUS DAN GRAFANA DI PT. CONCORD CONSULTING INDONESIA"**.

## B. Identifikasi Masalah

Berdasarkan uraian dari latar belakang yang telah dikemukakan diatas, maka dapat diidentifikasi masalah yaitu sebagai berikut :

1. Pengelolaan *server* masih dilakukan secara manual dengan perintah Linux, yang memerlukan banyak waktu dan tidak efisien untuk mendeteksi masalah secara cepat.
2. Data *monitoring* yang ada hanya dalam bentuk teks, membuatnya sulit untuk dipahami dengan cepat serta menghambat pengambilan keputusan yang tepat waktu.
3. Tidak adanya notifikasi otomatis yang terintegrasi dengan tim *Developer* dan *IT Operation* memperlambat respons terhadap gangguan, yang memperbesar risiko kerusakan sistem.
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.

## C. Pembatasan Masalah

Pada penelitian ini memiliki batasan-batasan masalah yang perlu diperhatikan sebagai berikut :

1. Sistem ini hanya menampilkan data mengenai penggunaan beban (*load*) dan ruang kosong (*freespace*) pada *resource server*.
2. Akses ke sistem *monitoring server* ini terbatas hanya untuk *IT Operation* dan *Developer* yang terdaftar dan memiliki hak akses yang sah.
3. Sistem *monitoring server* ini hanya dapat digunakan pada *server* yang terhubung dalam jaringan lokal (*local area network*) atau intranet perusahaan.

4. Sistem *monitoring server* ini hanya akan mengirimkan notifikasi melalui Telegram menggunakan bot, yang hanya dapat diterima oleh *IT Operation* dan *Developer* terdaftar.

#### **D. Perumusan Masalah**

Berdasarkan latar belakang yang telah dikemukakan diatas, maka dapat dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana sistem *monitoring server* yang dirancang dapat memantau penggunaan *resource server* secara *real-time* untuk mendukung kinerja *server* yang optimal?
2. Bagaimana sistem notifikasi otomatis melalui Telegram yang dirancang dapat mempercepat respons *IT Operation* dan *Developer* terhadap gangguan yang terjadi pada *server*?
3. Bagaimana visualisasi data *server* yang dirancang dapat membantu *IT Operation* dalam membaca dan menginterpretasikan informasi performa *server* secara mudah dan informatif?
4. Bagaimana fitur *download report* yang dirancang dapat memuat informasi *server* secara *real-time* sehingga mempermudah proses analisis gangguan pada *server*?

#### **E. Tujuan dan Manfaat Penelitian**

##### **a. Tujuan Penelitian**

Adapun tujuan dari penelitian ini sebagai berikut :

1. Merancang sistem *monitoring server* berbasis *web* yang mampu memantau penggunaan *resource server* secara *real-time*.
2. Menerapkan fitur notifikasi otomatis melalui Telegram untuk memberikan informasi kondisi *server* kepada *IT Operation* dan *Developer* secara cepat dan efisien.
3. Memvisualisasikan data *server* agar mudah dibaca dan diinterpretasikan oleh *IT Operation*.



4. Merancang fitur *download report* yang memuat data *server* secara *real-time* untuk mempermudah proses analisis gangguan pada *server*.

## **b. Manfaat Penelitian**

Adapun manfaat yang akan dicapai dari penelitian ini, dapat diklasifikasikan sebagai berikut :

### **1. Manfaat Teoritis**

#### **a. Pengembangan Ilmu Pengetahuan**

Penelitian ini berkontribusi dalam pengembangan pengetahuan tentang integrasi Prometheus dan Grafana untuk *monitoring server*, khususnya dalam mengoptimalkan penggunaan teknologi visualisasi data dan notifikasi *real-time*.

#### **b. Referensi Akademik**

Hasil penelitian ini dapat menjadi referensi bagi penelitian selanjutnya untuk mengembangkan sistem *monitoring* serupa atau memperluas penerapan teknologi *monitoring* pada skala yang lebih luas.

### **2. Manfaat Praktis**

#### **a. Bagi Peneliti**

Penelitian ini memberikan pengalaman langsung dalam merancang dan mengimplementasikan sistem *monitoring server* berbasis *web* menggunakan Prometheus dan Grafana, serta mengintegrasikan fitur notifikasi melalui Telegram.

#### **b. Bagi Perusahaan**

Penelitian ini membantu perusahaan mengatasi permasalahan dalam *monitoring server* yang selama ini dilakukan secara manual, sehingga proses pemantauan menjadi lebih efisien, responsif dan terstruktur. Dengan demikian, perusahaan dapat meminimalkan risiko gangguan operasional akibat masalah *server*.

c. Bagi Masyarakat atau Industri Lain

Sistem *monitoring* ini dapat diadaptasi oleh organisasi atau industri lain yang memiliki kebutuhan serupa untuk meningkatkan efisiensi dalam pengelolaan *server* dan mencegah *downtime* sistem.

d. Bagi Lembaga Pendidikan

Penelitian ini dapat dijadikan bahan pembelajaran atau studi kasus dalam bidang pengembangan perangkat lunak, administrasi *server*, atau teknologi *monitoring* berbasis visualisasi data.

## F. Sistematika Penulisan

Untuk memberikan gambaran menyeluruh terhadap masalah yang dibahas, maka sistematika penulisan dibagi ke dalam beberapa bab yaitu :

### **BAB I PENDAHULUAN**

Pada bab ini menjelaskan tentang latar belakang dari hal-hal yang berhubungan dengan perusahaan, identifikasi masalah, rumusan masalah, batasan masalah, tujuan yang ingin dicapai, manfaat yang diperoleh dengan dibuatnya Proposal Skripsi ini, serta sistematika dari penulisan Laporan Proposal Skripsi.

### **BAB II DESKRIPSI TEORITIK DAN KERANGKA BERPIKIR**

Bab ini menjelaskan secara singkat teori pembuatan sistem informasi yang digunakan sebagai bahan acuan dalam pelaksanaan dan penulisan skripsi, serta pada bagian akhir bab ini akan dijelaskan kerangka pikir dalam perancangan dan pembuatan sistem informasi pengolahan nilai.

### **BAB III METODOLOGI PENELITIAN**

Bab ini diuraikan tentang langkah-langkah yang digunakan dalam penelitian, yang meliputi waktu dan tempat pelaksanaan, tinjauan organisasi, metode perancangan sistem informasi, analisis sistem berjalan, analisis data, analisis kebutuhan, analisis keluaran, analisis masukan serta analisis proses dan pemodelan.



## **BAB II**

### **DESKRIPSI TEORITIK DAN KERANGKA BERFIKIR**

#### **A. Deskripsi Teoritik**

##### **A.1. Definisi Optimalisasi**

Optimalisasi berasal dari kata “optimal” yang berarti terbaik, tertinggi, paling menguntungkan. Optimalisasi 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 Poerdwadminta (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 tersedia. 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.

## A.2. Definisi *Monitoring*

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

*Monitoring* (pemerantauan) 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 nantinya digunakan sebagai bahan evaluasi kearah yang lebih baik.

### A.3. 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.

#### **A.4. 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 memudahkan bagi siapa saja yang membacanya, contoh menampilkan data kedalam *dashboard* yang berisi grafik-grafik.

#### **A.5. 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 *exporter* 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 & Oktawati, 2022).

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

### A.6. 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 memvisualisasikan, 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, PostgreSQL, 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.

#### **A.7. 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*.

#### A.8. 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 MTPProto (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 MTPProto. 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.

#### **A.9. 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.

#### A.10. 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 Transfer 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 berfungsi 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.

#### **A.11. 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, *webmaster*, 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.

#### A.12. Definisi *Unified Modeling Language* (UML)

UML (*Unified Modeling Language*) merupakan bentuk pemodelan untuk sistem atau perangkat lunak yang sedang dirancang atau akan dibangun agar bisa tersruktur dengan rapi ketikan akan dibuat nantinya. Pemodelan digunakan untuk menyederhankan permasalahan yang rumit agar dapat dimengerti dan dipelajari dengan mudah (Sigi Irawan, 2024).

UML merupakan bahasa untuk memperjelas, menggambarkan, serta membangun dasar dari sistem perangkat lunak yang akan dirancang, termasuk didalamnya melibatkan aturan-aturan bisnis (Adi Nugroho, 2014) dikutip dalam (Sigi Irawan, 2024).

UML (*Unified Modeling Language*) merupakan bahasa pemodelan yang banyak digunakan untuk merancang sistem perangkat lunak (Ramdany et al., 2024).

Menurut Rosa A.S dan M. Shalahuddin (2015:133) menyatakan bahwa *Unified Modeling Language* (UML) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Aman & Suroso, 2021).

Menurut Nunu Nurdiana (2018:29) menyatakan bahwa *Unifiield Modeling Language* (UML) adalah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggambarkan diagram dan teks-teks pendukung (Aman & Suroso, 2021).

Berdasarkan dari beberapa definisi UML dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah metode

pemodelan berbasis objek yang digunakan untuk merancang dan membangun sistem atau perangkat lunak secara terstruktur. UML membantu menyederhanakan permasalahan yang kompleks agar lebih mudah dipahami dan dipelajari. Selain itu, UML mencakup serangkaian teknik dan struktur untuk pemodelan desain sistem dan berfungsi sebagai bahasa untuk mendefinisikan, memvisualisasikan, mengembangkan dan mendokumentasikan sistem informasi. UML juga menjadi bahasa standar yang banyak digunakan dalam industri untuk mendefinisikan kebutuhan, menganalisis, merancang dan menggambarkan arsitektur dalam pemrograman berbasis objek.

### 1) *Use Case Diagram*

*Use case diagram* merupakan deskripsi sistem dari sudut pandang pengguna, *use case diagram* bekerja dengan cara menjabarkan interaksi yang akan dilakukan antara sistem satu dengan sistem lainnya atau sistem yang ada didalam sistem itu sendiri, melalui sebuah cerita singkat tentang bagaimana sistem bekerja (Sigi Irawan, 2024).

*Use case diagram* yaitu model hasil analisis perancangan sistem yang bertujuan untuk mendeskripsikan kebutuhan sistem. Kebutuhan sistem tersebut akan diterapkan oleh pengguna sehingga perancangan sistem dapat tergambarkan (Ocha Widya Susanti & Ummami, 2022) dikutip dalam (Ramdany et al., 2024).



*Use case diagram* adalah abstraksi dari interaksi antara sistem dengan aktor. Oleh karena itu sangat penting untuk memilih abstraksi yang sesuai dengan kebutuhan. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara admin dan user dari sebuah sistem dengan melalui sebuah diagram bagaimana sebuah sistem dipakai (Ibnu Alvayet &


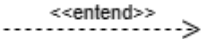

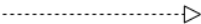
VeZRino Barrichelo, 2023) dikutip dalam (Ramdany et al., 2024).

Diagram kasus pengguna (*use case diagram*) merupakan jenis diagram yang digunakan dalam pemodelan sistem informasi untuk menggambarkan interaksi antara pengguna dan sistem. Diagram ini memanfaatkan simbol-simbol standar untuk menggambarkan aktor, kasus pengguna, dan hubungan antara keduanya (Riza et al., 2024).

Berdasarkan dari beberapa definisi *Use Case Diagram* dapat disimpulkan bahwa *Use Case Diagram* merupakan diagram yang digunakan dalam pengembangan perangkat lunak untuk mendeskripsikan hubungan antara aktor (pengguna atau sistem lain) dengan sistem yang akan dibuat. Diagram ini membantu menggambarkan secara ringkas siapa yang menggunakan sistem, apa saja fungsi yang tersedia dan merupakan langkah awal yang penting dalam pemodelan perangkat lunak berbasis objek.

Tabel 2. 1 Simbol *Use Case Diagram*

No	Simbol	Nama simbol	Deskripsi
1		<i>ACTOR</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2		<i>USE CASE</i>	Fungsionalitas yang disediakan sistem sebagai unit- unit yang saling bertukar pesan antar unit atau <i>actor</i> , biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .

3		ASOSIASI / ASSOCIATION	Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> dan memiliki interaksi dengan <i>actor</i> .
4		EKSTEND / EXTEND	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5		GENERALISA SI / GENERALIZATION	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
6		MENGGUNAK AN / INCLUDE	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsional atau sebagai syarat dijalankan <i>use case</i> ini.

## 2) Activity Diagram

*Activity diagram* merupakan bentuk khusus dari kemampuan sistem yang bertujuan memodelkan setiap aktifitas dan aliran kerja yang sedang berlangsung didalam sistem atau perangkat lunak yang sedang dikembangkan (Sigi Irawan, 2024).

*Activity diagram* menggambarkan aliran fungsionalisme dalam suatu sistem informasi. Secara lengkap, *activity diagram* mendefinisikan dimana *workflow* dimulai, dimana berakhirnya, aktivitas apa yang terjadi selama *workflow* dan bagaimana urutan kejadian aktivitas tersebut (Dewi et al., 2017) dikutip dalam (Ramdany et al., 2024).



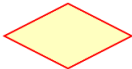
Diagram aktivitas adalah tipe diagram dalam *Unified Modeling Language* (UML) yang mengilustrasikan alur kerja dan aktivitas yang terjadi dalam suatu sistem atau proses.




Diagram ini memanfaatkan simbol-simbol grafis untuk merepresentasikan aktivitas, keputusan, objek dan aliran data dalam sistem atau proses yang sedang dimodelkan (Riza et al., 2024).

*Activity Diagram* ialah diagram yang dapat menampilkan prosedur logika dan proses bisnis dalam sebuah sistem informasi (Wulandari & Nurmiati, 2022).

Berdasarkan dari beberapa definisi *Activity Diagram* dapat disimpulkan bahwa *Activity Diagram* adalah diagram dalam *Unified Modeling Language* (UML) yang berfungsi untuk menggambarkan alur aktivitas, prosedur logika dan proses bisnis dalam suatu sistem atau proses. Diagram ini memanfaatkan simbol-simbol grafis untuk merepresentasikan aktivitas, pengambilan keputusan, objek, serta aliran data, sehingga memberikan gambaran yang jelas mengenai alur kerja dalam sistem yang dimodelkan.

Tabel 2. 2 Simbol *Activity Diagram*

No	Simbol	Nama simbol	Deskripsi
1		Status Awal / <i>Initial</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas / <i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		Percabangan / <i>Decision</i>	Asosiasi percangana dimana ada pilihan aktivitas yang lebih dari satu.

4		Penggabungan / <i>Join</i>	Asosiasi penggabungan yang mana lebih dari satu aktivitas lalu digabungkan menjadi satu
5		Status Akhir / <i>Final</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.

## B. Penelitian Sebelumnya

Tabel 2. 3 Penelitian Sebelumnya

No	Peniliti, tahun, judul penelitian, universitas	Masalah	Metode	Hasil penelitian	Perbandingan
1	Rahayu Mutiara Febriani, 2020, Implementasi Sistem <i>Monitoring</i> 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 <i>monitoring</i> yang diinginkan. Prometheus dan Grafana dapat membantu membuat sistem <i>monitoring</i> jaringan yang mudah dimengerti dan dapat membantu seorang sistem administrator. Prometheus akan mengambil data dari <i>protocol</i> SNMP dan memproses data tersebut	System Development Life Cycle (SDLC).	Dari hasil penelitian yang dilakukan dapat diperoleh sebuah kesimpulan bahwa adanya sistem <i>monitoring</i> 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 <i>network traffic</i> serta jumlah perangkat yang terhubung dalam jaringan. Dengan adanya sistem <i>monitoring</i> 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 <i>real-time</i> , dengan fitur pemantauan interface aktif, penggunaan CPU, memori, <i>traffic</i> 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 <i>real-time</i> , fitur pengunduhan laporan <i>resource</i> untuk analisis dan dokumentasi, serta pembagian akses pengguna yang meningkatkan keamanan dan fleksibilitas. Hal ini menjadikan sistem yang dikembangkan lebih unggul



		menggunakan exporter. Untuk melengkapi <i>system monitoring</i> ini diperlukan visualisasi. Grafana dapat membuat visualisasi dari data yang telah diproses. Hasil penelitian ini menunjukkan <i>device</i> yang terhubung didalam jaringan dapat ditampilkan grafana			dalam efisiensi dan fungsionalitas.
2	Dede Rahman, Hidra Amnur, Indri Rahmayuni, 2020, <i>Monitoring Server Dengan Prometheus Dan Grafana Serta Notifikasi Telegram</i> , Politeknik Negeri Padang.	Administrator jaringan adalah orang yang bertanggung jawab dalam mengelola jaringan komputer, Salah satu tugas dari seorang administrator jaringan adalah <i>monitoring</i> jaringan karena kesalahan pada <i>Server</i> dan <i>service</i> yang berjalan di <i>Server</i> 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 <i>Server</i> atau tiba-tiba <i>service</i> dari <i>Server</i> 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 <i>Server</i> berbasis	Prometheus dan Grafana	<i>Monitoring Server</i> berhasil dilakukan dengan menggunakan prometheus dan grafana terhadap <i>Server</i> . <i>Monitoring Server</i> bekerja dengan baik dan sesuai dengan harapan. Sistem operasi yang digunakan pada <i>monitoring Server</i> adalah Ubuntu <i>Server</i> dengan versi 18.04. Sistem yang dibangun memberikan notifikasi atau pemberitahuan kepada admin apabila CPU, memori ataupun <i>service</i> apache dan MYSQL ada yang mati. Alert di grafana akan selalu mengirimkan pemberitahuan pada telegram apabila kondisi <i>Server</i> melewati batas yang telah di tentukan atau apabila <i>service</i> dari apache maupun mysql mati. <i>Flavor</i> yang bisa digunakan berdasarkan spesifikasi VCPU, RAM dan Disk sesuai dengan pemilihan spesifikasi minimal	Pada penelitian sebelumnya, sistem <i>monitoring Server</i> menggunakan Prometheus dan Grafana telah berhasil diterapkan untuk memantau kondisi <i>Server</i> , dengan dukungan sistem operasi Ubuntu <i>Server</i> 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 <i>alert</i> pada Grafana untuk mengirimkan pemberitahuan secara <i>real-time</i> , 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 <i>service</i> , Jenkins, dan Kubernetes. Sebagai

		Linux, administrator jaringan dapat mendapatkan informasi kondisi <i>Server</i> dan <i>service</i> yang berjalan di <i>Server</i> secara cepat. Hasil yang didapatkan adalah sebuah sistem yang mampu membantu administrator jaringan dalam melakukan <i>monitoring Server</i> dan <i>service</i> yang sedang berjalan di <i>Server</i> kapan saja dan dimana saja untuk menjaga <i>Server</i> dan <i>service</i> yang berjalan pada <i>Server</i> agar tetap stabil walaupun administrator jaringan sedang tidak di pusat pengawasan		untuk membuat <i>instance</i> . Diharapkan untuk pengembangan selanjutnya sistem <i>monitoring Server</i> dapat memberikan <i>alert</i> ke aplikasi mobile yang dibuat sendiri. untuk pengembangan selanjutnya, implementasi <i>monitoring Server</i> dengan prometheus dan grafana memiliki tempat implementasi khusus. Diharapkan ada pengembangan dari <i>monitoring Server</i> dengan prometheus dan grafana untuk <i>monitoring service</i> lainnya seperti SSH service , mail <i>service</i> , jenkins , kubernetes dan lain – lain	perbandingan, sistem yang akan dikembangkan memiliki beberapa keunggulan tambahan. Selain mendukung pemantauan <i>resource Server</i> serta fitur notifikasi otomatis melalui integrasi Telegram untuk memastikan administrator mendapatkan peringatan <i>real-time</i> . Sistem ini juga menyediakan fitur pengunduhan laporan <i>resource</i> 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 <i>monitoring Server</i> dasar, tetapi juga menghadirkan efisiensi, skalabilitas, dan keamanan yang lebih baik dibandingkan sistem pada penelitian sebelumnya.
3	Banu Rasyidi, Firman Pratama, 2024, Sistem <i>Monitoring Server</i> di PT. XYZ Media Indonesia Berbasis Grafana dan	<i>Server</i> merupakan salah satu komponen utama dari sistem jaringan komputer yang memiliki fungsi untuk memberikan suatu <i>service</i> terhadap penggunanya. Setiap aktifitas dan operasional	Prometheus dan Grafana	Berdasarkan pembahasan dan hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem <i>monitoring Server</i> berbasis prometheus dan grafana telah berhasil dibangun dan diimplementasikan dengan baik.	Penelitian sebelumnya berhasil mengembangkan sistem <i>monitoring Server</i> berbasis Prometheus dan Grafana yang mempermudah proses pemantauan <i>Server</i> secara <i>remote</i> , terutama melalui pelaporan cepat

	Prometheus, Universitas Pamulang.	<p>pelayanan suatu <i>Server</i> terhadap <i>client</i> dalam penerapannya terdiri dari beberapa jenis proses untuk memenuhi segala permintaan atau <i>request client</i> yang dikirimkan oleh <i>Server</i>. Oleh sebab itu, diperlukan suatu sistem <i>monitoring</i> yang dapat melakukan pemantauan segala macam aktifitas di dalam <i>Server</i>, dengan demikian pengguna dapat memantau dan memberikan peringatan apabila terjadi permasalahan pada <i>Server</i> yang digunakan. Pada sistem ini dirancang dengan menggunakan perancangan <i>Unified Modeling Language</i>, serta menggunakan node exporter untuk mengambil metrik dan bahasa kueri berbasis Prometheus <i>Query Language</i> untuk mengakses data metrik dan grafana untuk memvisualisasi. Penelitian ini bertujuan rancang sistem yang mampu melakukan pemantauan dan pengumpulan informasi dari seluruh perangkat <i>Server</i> yang berada di PT. XYZ Media Indonesia. Sistem <i>monitoring</i> ini juga, mampu memberikan</p>		<p>Pengujian langsung oleh karyawan divisi IT menunjukkan bahwa sistem ini mampu mempermudah proses <i>monitoring Server</i> secara <i>remote</i>, yang sebelumnya dilakukan secara manual. Sistem ini efektif dalam memberikan informasi terkait aktivitas <i>Server</i> yang dipantau, termasuk status penggunaan CPU, memori, koneksi jaringan, dan utilisasi <i>storage</i>. Selain itu, pelaporan melalui telegram terbukti memberikan respon cepat, memungkinkan administrator untuk menangani permasalahan dengan segera. Metode pengujian <i>blackbox</i> yang diterapkan pada sistem menunjukkan bahwa semua fungsi sistem beroperasi dengan baik, menandakan tingkat fungsionalitas yang memuaskan.</p>	<p>via Telegram. Sistem ini efektif dalam menampilkan informasi status CPU, memori, koneksi jaringan, dan storage, serta telah diuji menggunakan metode <i>blackbox</i> dengan hasil memuaskan. Namun, sistem ini belum mencakup fitur tambahan seperti pengelolaan akses pengguna, pengunduhan laporan <i>resource</i>, atau dukungan untuk analisis lebih lanjut. Sebagai perbandingan, sistem yang akan dikembangkan menawarkan fitur yang lebih lengkap dan fleksibel. Selain mendukung <i>monitoring resource Server</i> dan notifikasi otomatis melalui Telegram, sistem ini memungkinkan pengunduhan laporan <i>resource</i> 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.</p>
--	-----------------------------------	--	--	---	---

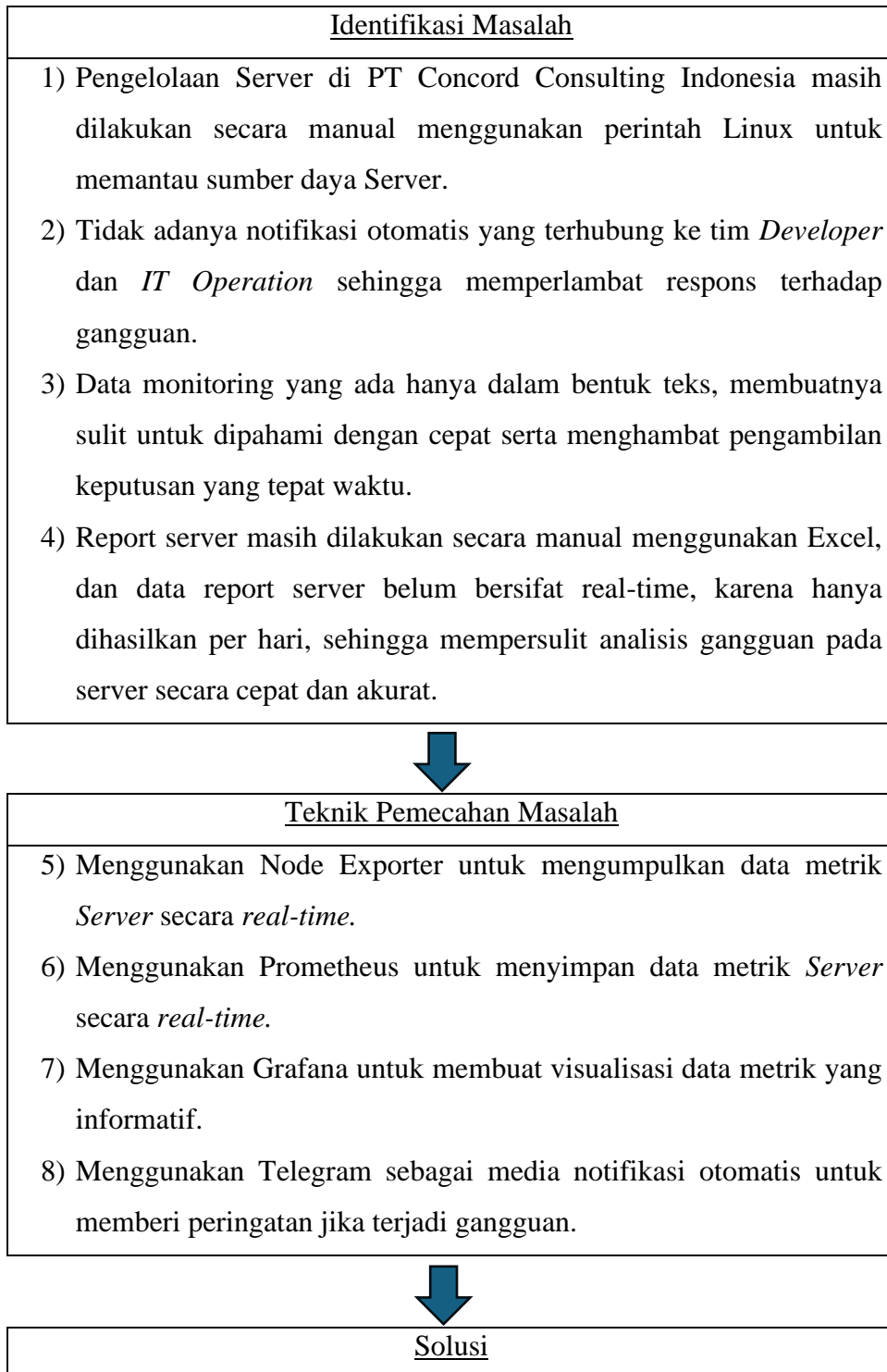
		beberapa informasi terkait utilisasi memori, utilisasi CPU, utilisasi <i>storage</i> dan utilisasi jaringan pada <i>Server</i> , serta pada sistem ini juga mampu mengirimkan pesan peringatan dalam bentuk notifikasi telegram kepada pengguna saat terjadi kesalahan atau <i>malfunction</i> pada <i>Server</i> dengan demikian proses eskalasi penanganan <i>Server</i> menjadi lebih cepat dilakukan			
4	Sinatria Banyu Adil, Yos Richard Beeh, 2024, Implementasi <i>Monitoring</i> Sistem Perusahaan <i>On-Premises</i> dan <i>Cloud</i> Menggunakan Teknologi Jenkins, Universitas Kristen Satya Wacana.	Penelitian ini mengimplementasikan Jenkins untuk memantau dan mengelola proses job, serta integrasinya dengan infrastruktur <i>on-premises</i> dan <i>cloud</i> di PT. AAA. <i>Compute Engine</i> digunakan sebagai <i>Server</i> utama untuk menjalankan Jenkins. Tahapan meliputi konfigurasi <i>Server Compute Engine</i> , instalasi plugin penting, dan penggunaan <i>library</i> Python. Hasilnya menunjukkan Jenkins berhasil dalam menjalankan proses job antara sistem <i>on-premises</i> dan <i>cloud</i> . <i>Email Extension Plugin</i> digunakan untuk memantau	Teknologi Jenkins	Berdasarkan hasil penelitian, penerapan Jenkins dalam pemantauan dan otomatisasi proses job sinkronisasi antara <i>database on-premises</i> dan <i>cloud</i> telah berhasil dilakukan oleh PT. AAA. Dengan menggunakan <i>Compute Engine</i> sebagai <i>Server</i> utama, konfigurasi <i>Server</i> yang optimal, instalasi <i>plugin</i> dan <i>library</i> yang diperlukan, serta pembuatan jenkins <i>file</i> , PT. AAA berhasil memantau dan mengelola proses job dengan efisien.	Penelitian sebelumnya menunjukkan keberhasilan penerapan Jenkins dalam pemantauan dan otomatisasi proses sinkronisasi antara <i>database on-premises</i> dan <i>cloud</i> di PT. AAA. Dengan memanfaatkan <i>Compute Engine</i> sebagai <i>Server</i> utama, optimalisasi konfigurasi <i>Server</i> , instalasi plugin, serta pembuatan jenkins <i>file</i> , sistem tersebut memungkinkan pengelolaan proses job secara efisien. Namun, sistem ini terbatas pada pemantauan dan otomatisasi sinkronisasi database tanpa menyediakan notifikasi <i>real-time</i> atau fitur pelaporan yang lebih mendalam. Sebagai pembanding, sistem yang akan

		<p>proses job dan memberikan notifikasi ke PIC jika terjadi kesalahan. Pengelolaan akses diterapkan melalui <i>Role-based Authorization Strategy</i>. Pengujian dilakukan menggunakan <i>blackbox</i> testing dengan tingkat keberhasilan 100%.</p> <p>Diharapkan implementasi ini dapat meningkatkan efisiensi dan efektivitas manajemen proses job di PT. AAA.</p>			<p>dikembangkan menawarkan fungsionalitas yang lebih luas dan mendalam. Selain mendukung monitoring <i>resource</i> dan notifikasi otomatis melalui Telegram, sistem ini memungkinkan pengunduhan laporan <i>resource</i> 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.</p>
5	<p>Saleh Dwiyatno, Edy Rakhmat, Oki Gustiawan, 2020, Implementasi Virtualisasi <i>Server</i> Berbasis Docker Container, Universitas Banten Jaya, Universitas Serang Raya.</p>	<p>Docker adalah sebuah aplikasi yang berbasiskan teknologi <i>open source</i> yang memungkinkan <i>Developer</i> 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</p>	Survey Literatur	<p>Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan, yaitu: Dengan adanya penerapan virtualisasi <i>Server</i> berbasis docker container pada Ubuntu 18.04 LTS, dapat menjamin <i>web</i> menjadi stabil. Karena pada docker container memastikan aplikasi dan sumber daya yang terisolasi serta terpisah sehingga para penggunaanya dapat menyesuaikan kebutuhan di</p>	<p>Penelitian sebelumnya berhasil menunjukkan bahwa penerapan virtualisasi <i>Server</i> berbasis Docker container pada Ubuntu 18.04 LTS mampu meningkatkan stabilitas aplikasi <i>web</i>. Docker container memungkinkan isolasi aplikasi dan sumber daya, sehingga konfigurasi setiap aplikasi dapat dikelola tanpa saling memengaruhi. Selain itu, Docker memungkinkan pemanfaatan <i>hardware</i> secara</p>

		<p>dijalankan dalam infrastruktur lokal tanpa melakukan perubahan konfigurasi pada container. Docker juga sangat ringan dan cepat jika dibandingkan dengan mesin <i>virtual</i> yang berbasis <i>hypervisor</i>. SMK Negeri 1 Rangkasbitung mempunyai beberapa <i>Server</i> yang mana didalamnya sudah menampung beberapa aplikasi <i>web</i>. <i>Server</i> tersebut berbasis <i>hypervisor</i> sehingga membutuhkan <i>resource</i> besar. Karena setiap VM menjalankan <i>guest OS</i> beserta kernelnya sendiri terpisah dari <i>host</i>. Oleh karena itu, dilakukan implementasi virtualisasi berbasis docker container supaya dapat meningkatkan efektifitas dalam penggunaan sumber daya CPU dan memori pada <i>Server</i>. Pengujian dilakukan dengan cara melakukan variasi banyaknya jumlah user request yang berbeda pada masing-masing container menggunakan perangkat lunak apache jmeter</p>		<p>setiap aplikasi tanpa perlu mempengaruhi konfigurasi pada aplikasi yang lain, Adanya pemanfaatan docker container pada perancangan <i>Server</i> dapat memanfaatkan <i>hardware</i> yang ada untuk digunakan secara maksimal. Karena pada docker container dimana kernel yang digunakan adalah dari bagian sistem operasi <i>host</i> nya sendiri, sehingga tidak membebani kinerja dari <i>Server host</i>.</p>	<p>maksimal dengan menggunakan kernel dari sistem operasi <i>host</i>, yang mengurangi beban pada <i>Server host</i>. Sebagai perbandingan, sistem yang akan dikembangkan tidak hanya mendukung pengelolaan dan stabilitas <i>Server</i> melalui <i>monitoring resource</i>, tetapi juga menawarkan fitur-fitur tambahan untuk meningkatkan efisiensi dan fleksibilitas pengelolaan. Fitur notifikasi otomatis melalui Telegram memberikan peringatan <i>real-time</i> jika terjadi masalah, sementara kemampuan untuk mengunduh laporan <i>resource</i> 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 <i>Server</i> yang lebih komprehensif.</p>
--	--	--	--	---	--

### C. Kerangka Berfikir

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



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.



## **BAB III**

### **METODOLOGI PENELITIAN**

#### **A. Metodologi Pengembangan Sistem**

##### **A.1. Metode Pengumpulan Data**

Cara yang digunakan dalam penelitian ini diantaranya:

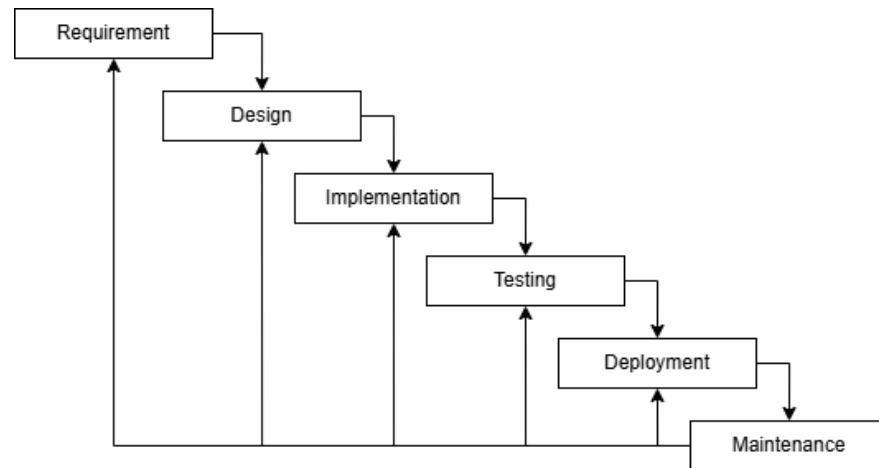
- 1) Penelitian Lapangan
  - a) Observasi, yaitu teknik pengumpulan data dengan cara mengamati secara langsung disekitar lingkungan kerja yang dibuat sebagai objek penelitian.
  - b) Wawancara, yaitu teknik pengumpulan data dengan cara wawancara langsung dengan mengajukan pertanyaan-pertanyaan tentang masalah-masalah yang akan dibahas.
- 2) Studi Literatur

Yaitu teknik pengumpulan data dengan cara mempelajari referensi, buku, jurnal, dokumen, atau sumber tertulis lainnya yang relevan dengan topik penelitian.

##### **A.2. Metode Pengembangan Sistem**

Salah satu metode pengembangan sistem yaitu metode *waterfall*. Metode *waterfall* merupakan metode air terjun atau yang sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan pengelolaan (*maintenance*) pada sistem yang diserahkan.

Tahapan metode *waterfall* dapat dilihat pada gambar dibawah ini



Gambar 3. 1 Tahapan Metode Waterfall

#### 1) Tahapan Metode *Waterfall*

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu: *requirement* (analisis kebutuhan), *design* (desain sistem), *implementation* (pengkodean), *testing* (pengujian), *deployment* (penyebaran) dan *maintenance* (pemeliharaan). Tahapan-tahapan dari metode *waterfall* adalah sebagai berikut:

##### a) *Requirement*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi, atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

*b) Design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain sistem membantu dalam menentukan perangkat keras dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

*c) Implementation*

Pada tahap ini, sistem pertama kali dikembangkan diprogram kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

*d) Integration & Testing*

Seluruh *unit* yang dikembangkan dalam tahap implementasi akan diintegrasikan kedalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

*e) Deployment*

Setelah seluruh sistem berhasil melewati tahap pengujian, perangkat lunak akan diterapkan ke lingkungan produksi. Proses *deployment* meliputi pengaturan *server*, instalasi perangkat lunak, konfigurasi sistem dan pengelolaan basis data. *Deployment* juga mencakup uji coba dalam lingkungan nyata untuk memastikan bahwa perangkat lunak berjalan sesuai dengan spesifikasi yang diharapkan.

*f) Maintenance*

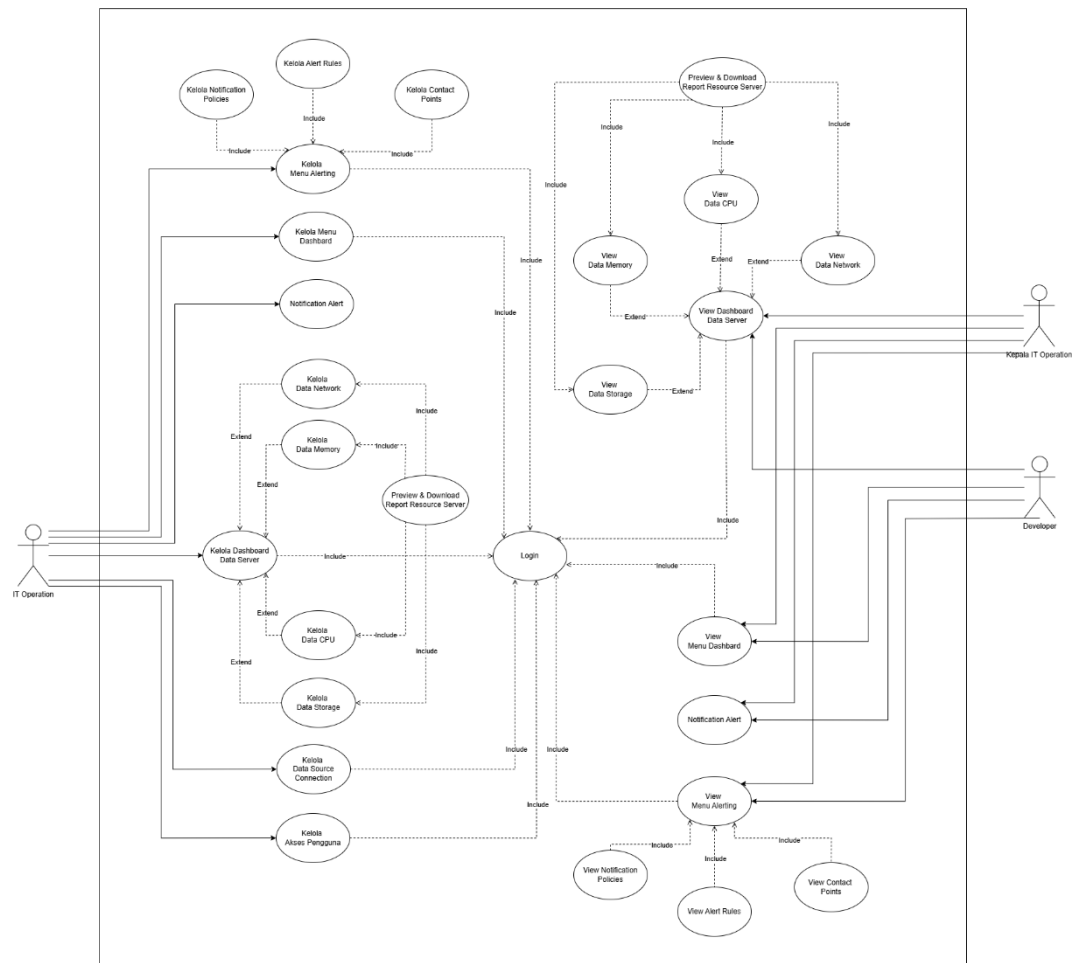
Tahap akhir dalam metode *waterfall*. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya, menyesuaikan sistem

dengan kebutuhan pengguna baru dan meningkatkan kinerja perangkat lunak jika diperlukan.

## B. Perancangan Sistem

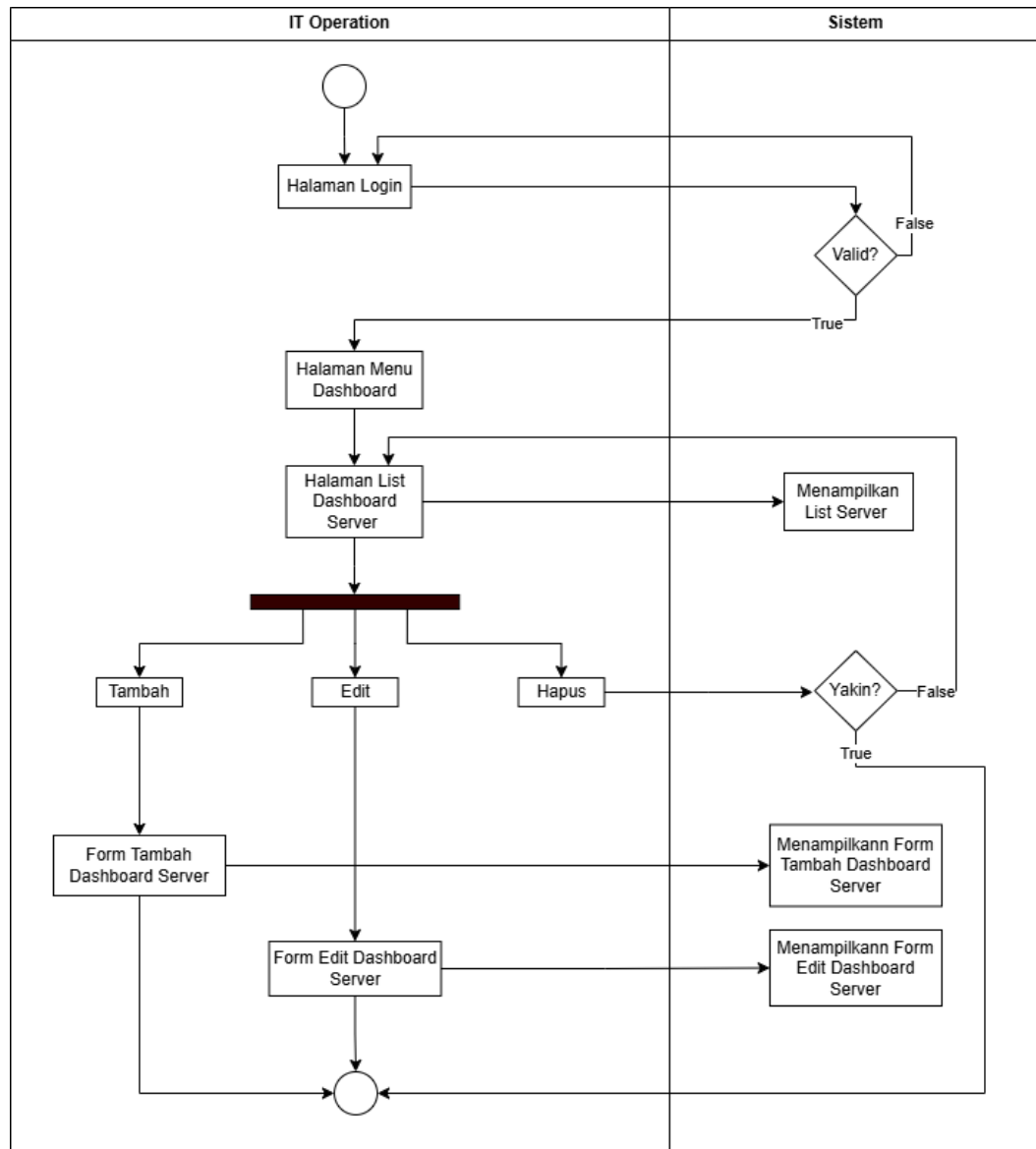
### B.1. Pemodelan

#### 1) Use Case Diagram

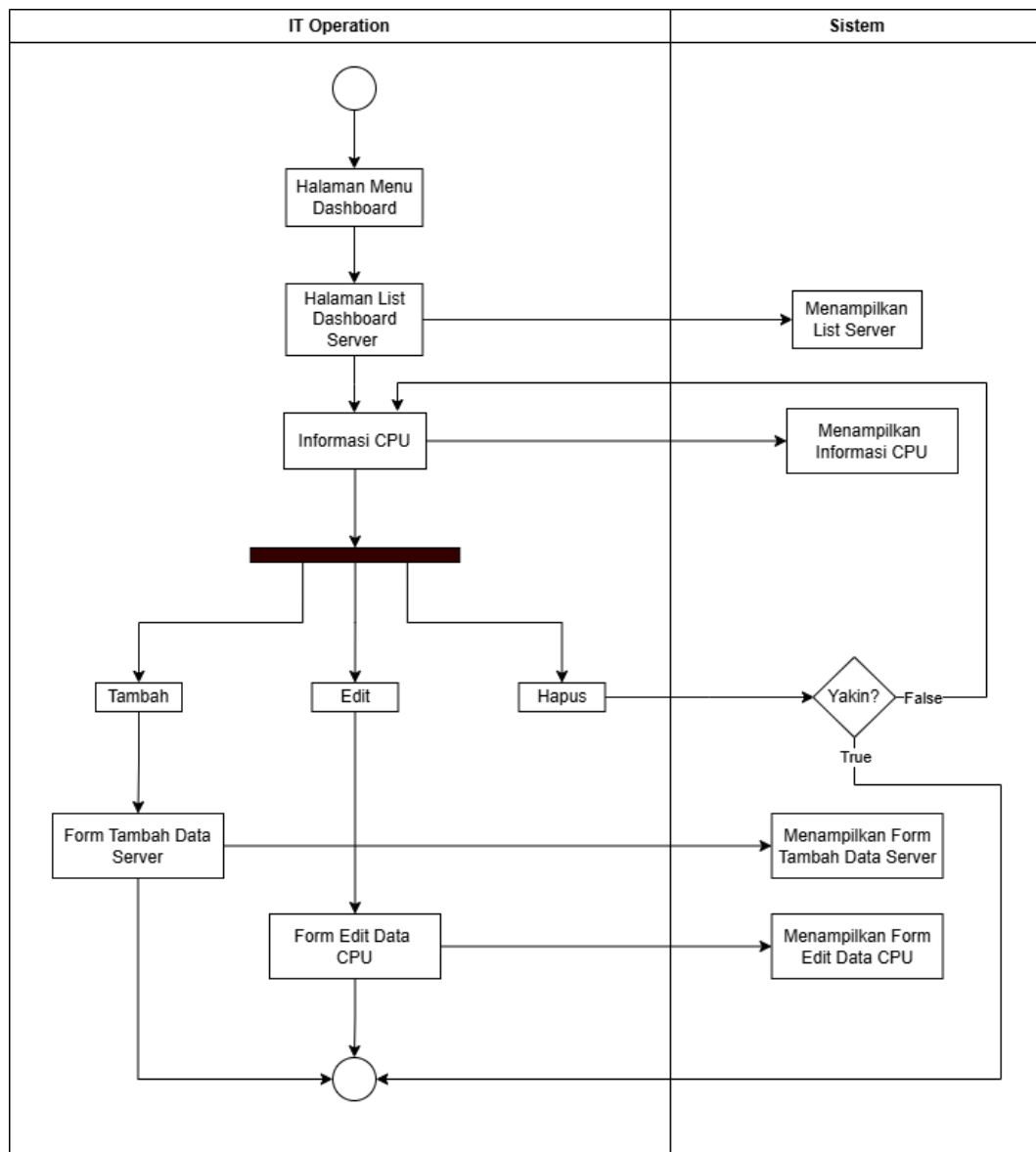


Gambar 3. 2 Use Case Diagram

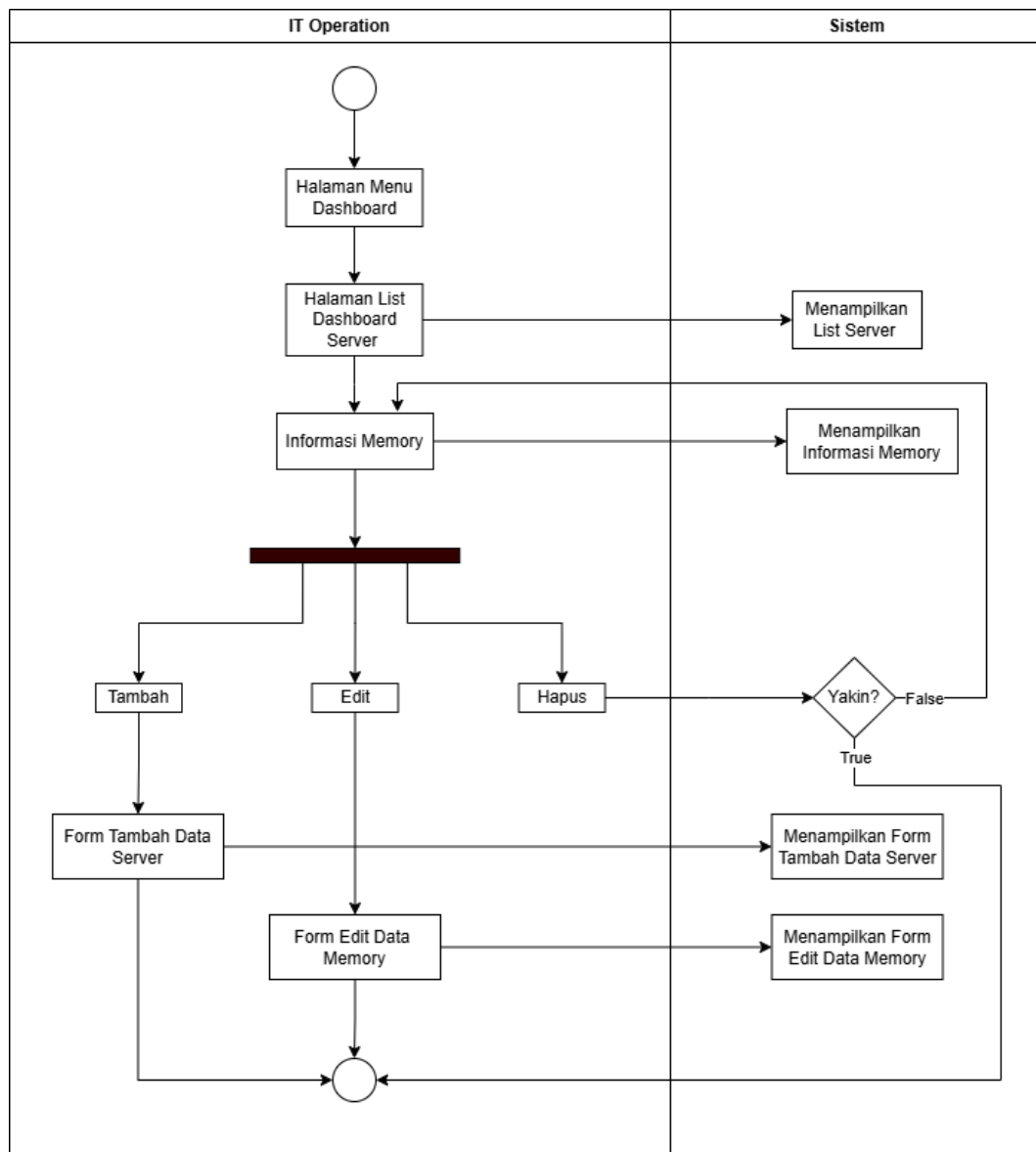
## 2) Activity Diagram

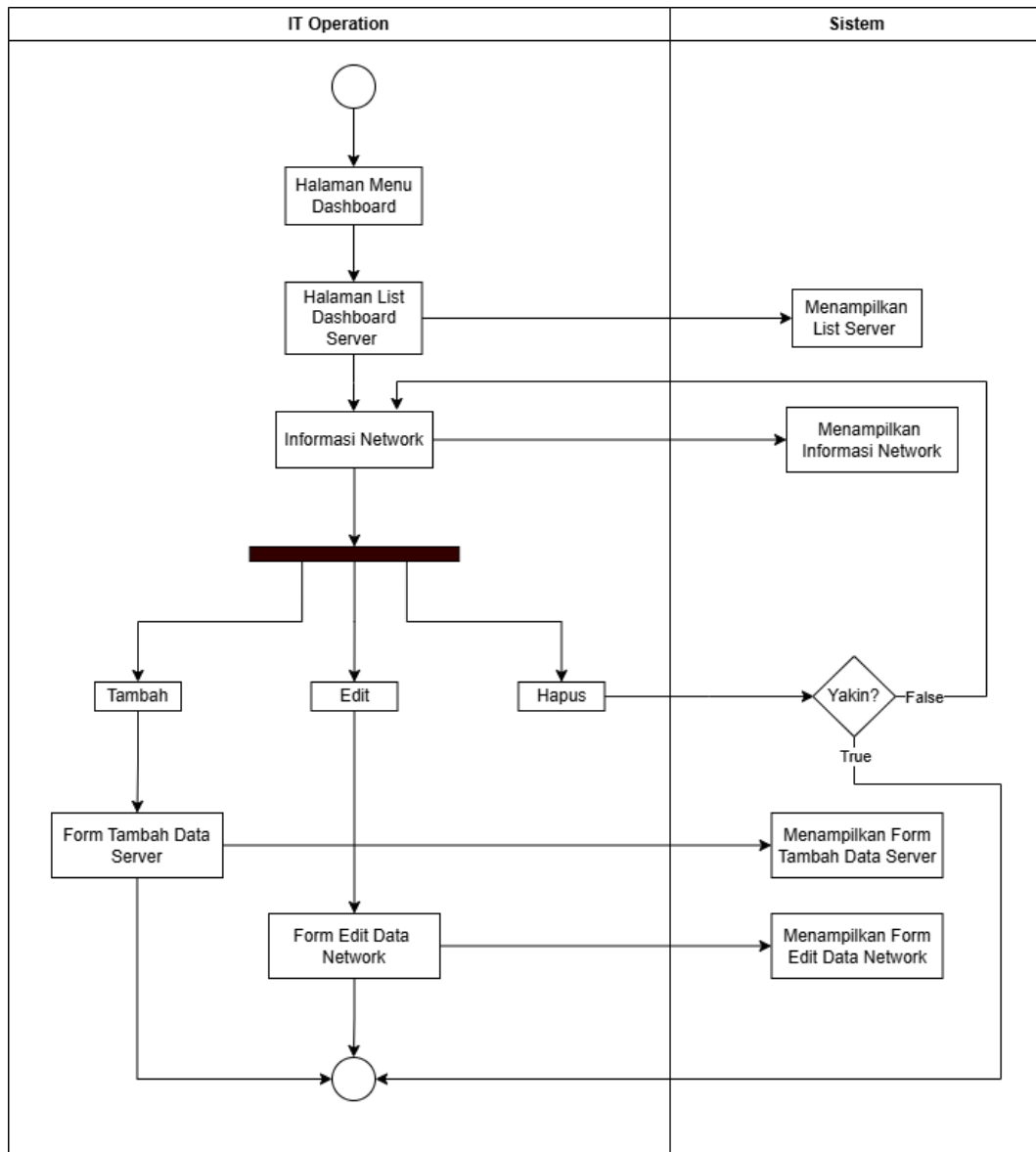


Gambar 3. 3 Kelola Dashboard

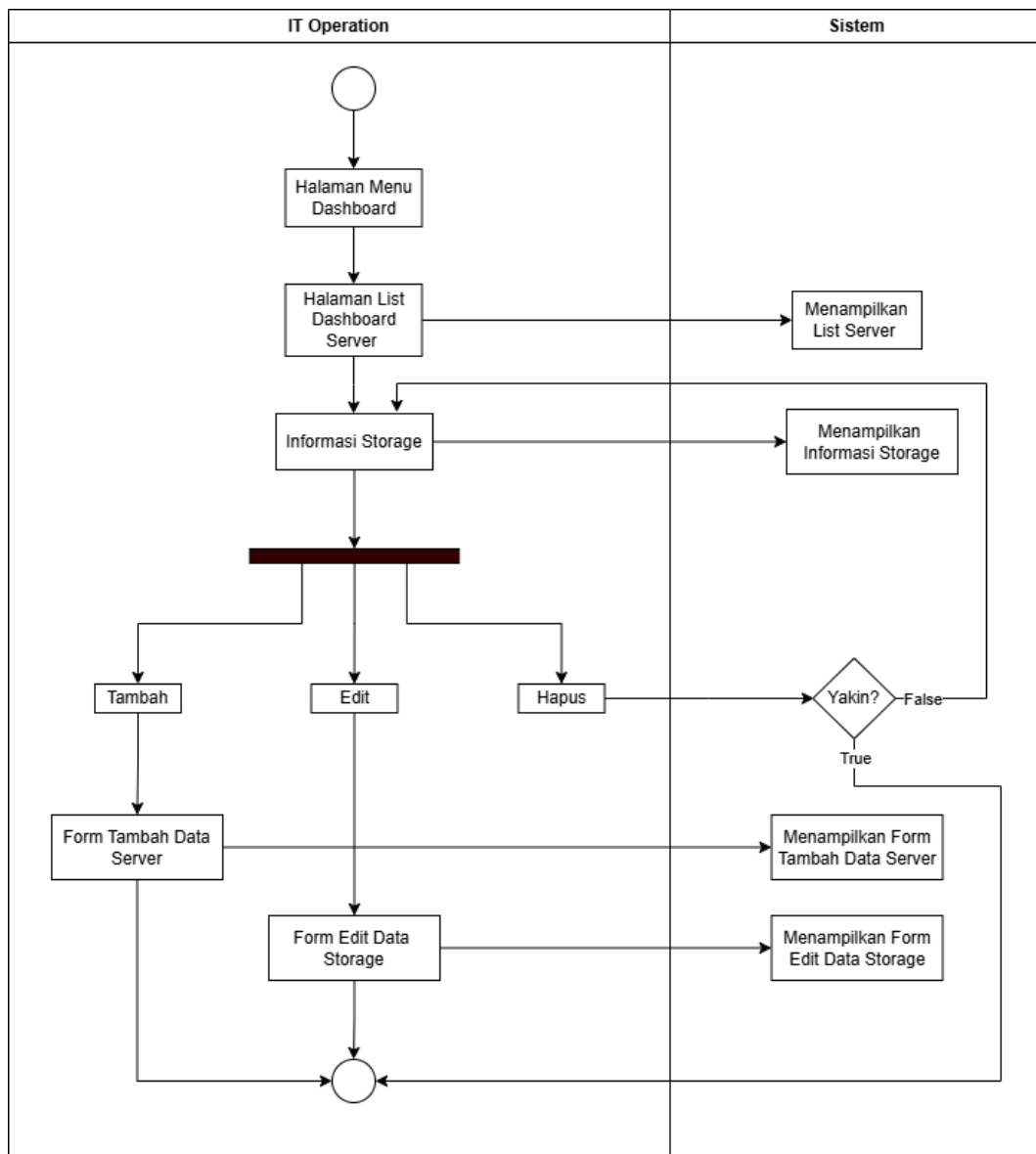


Gambar 3. 4 Kelola Data CPU

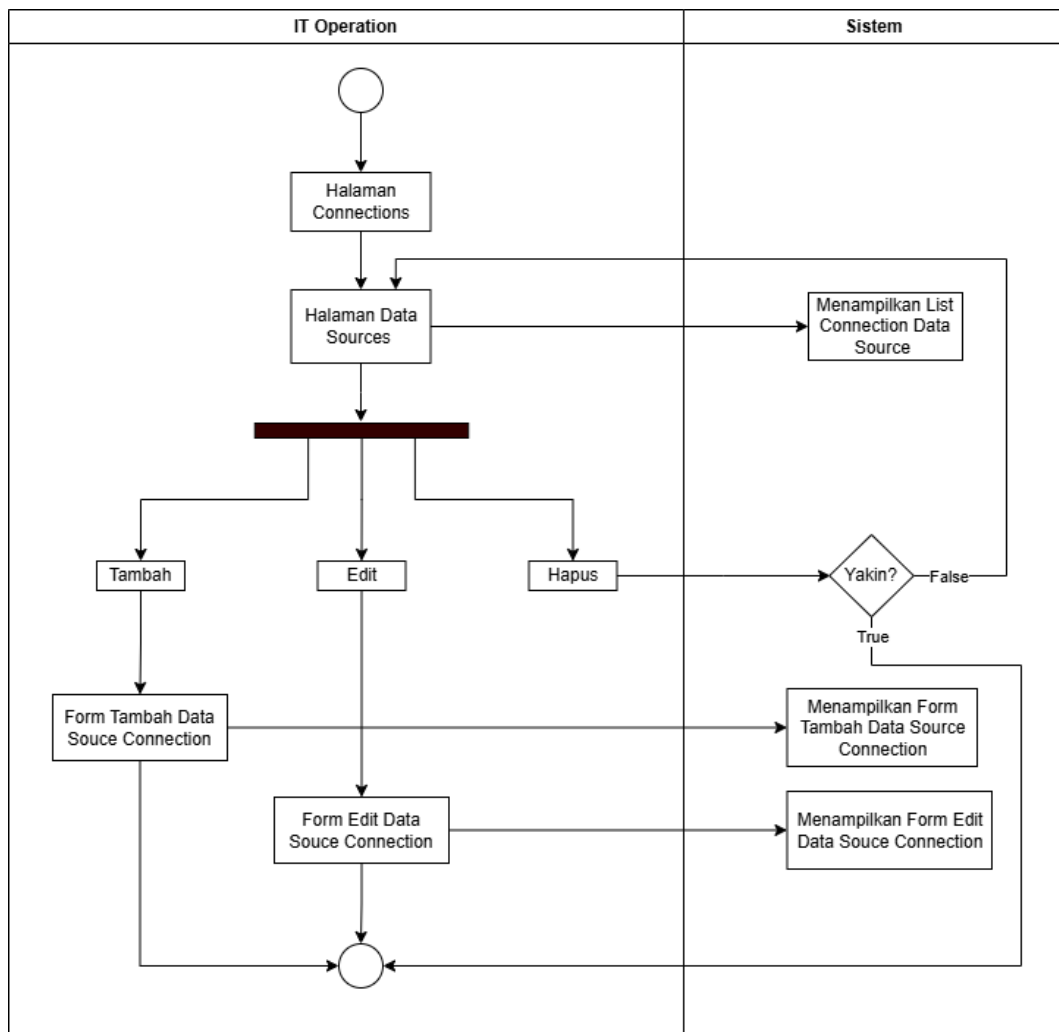
Gambar 3. 5 Kelola Data *Memory*

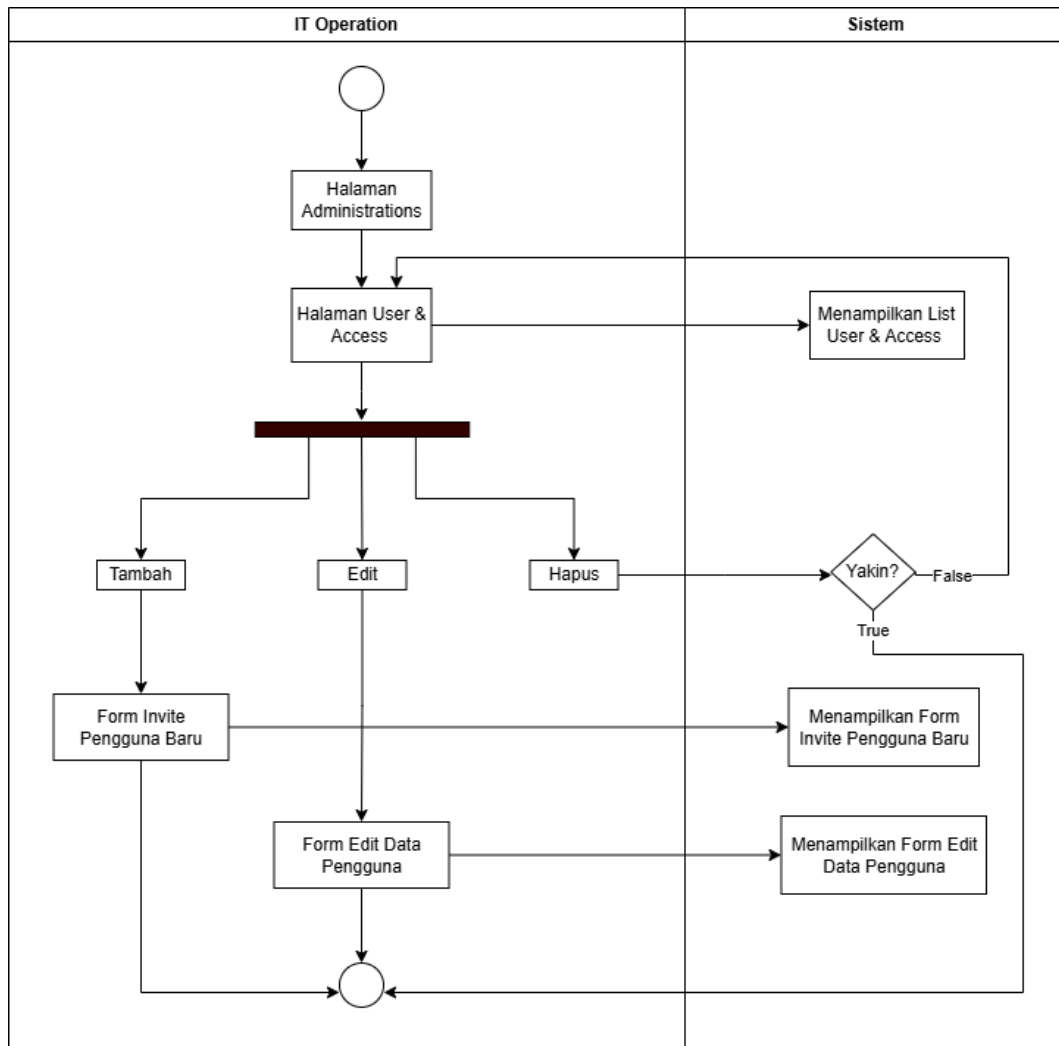
Gambar 3. 6 Kelola Data *Network*



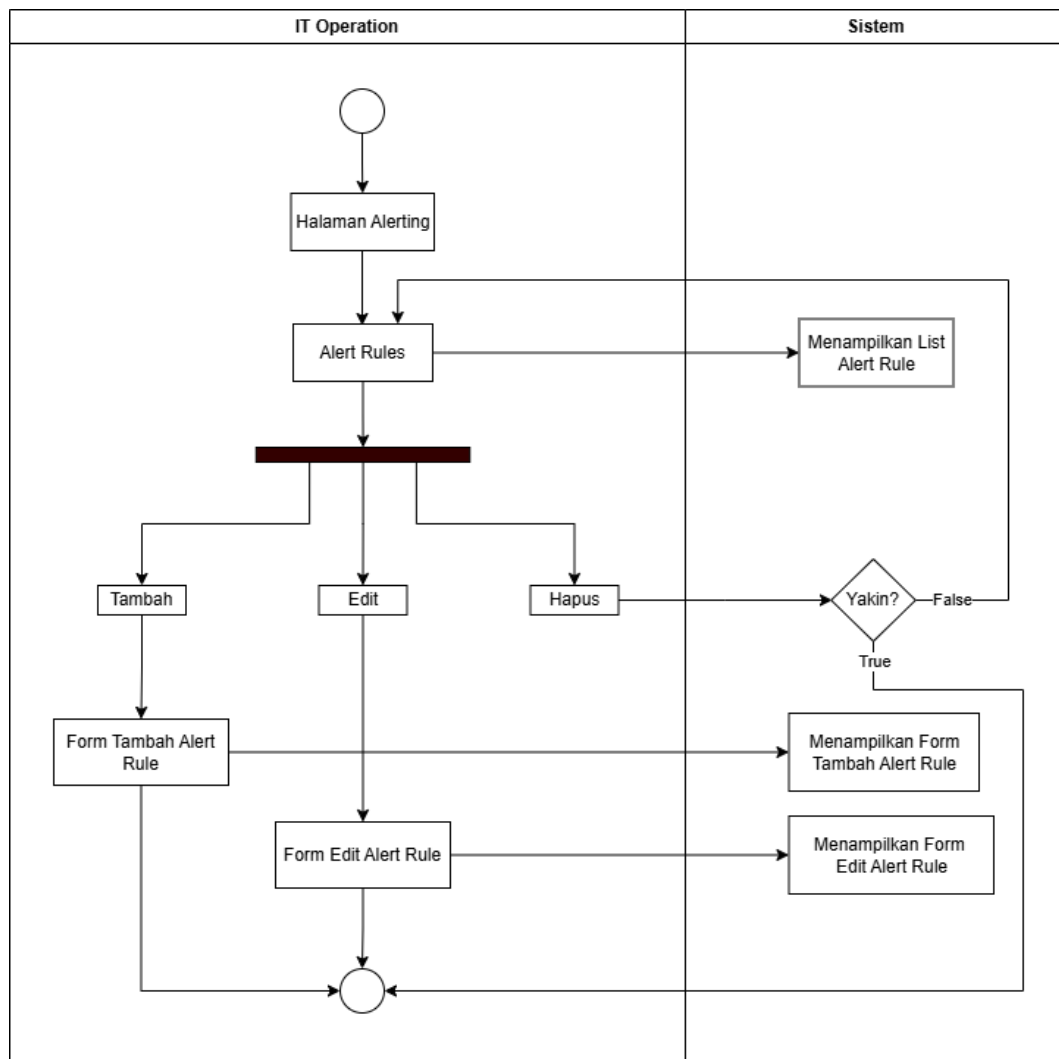


Gambar 3. 7 Kelola Data Storage

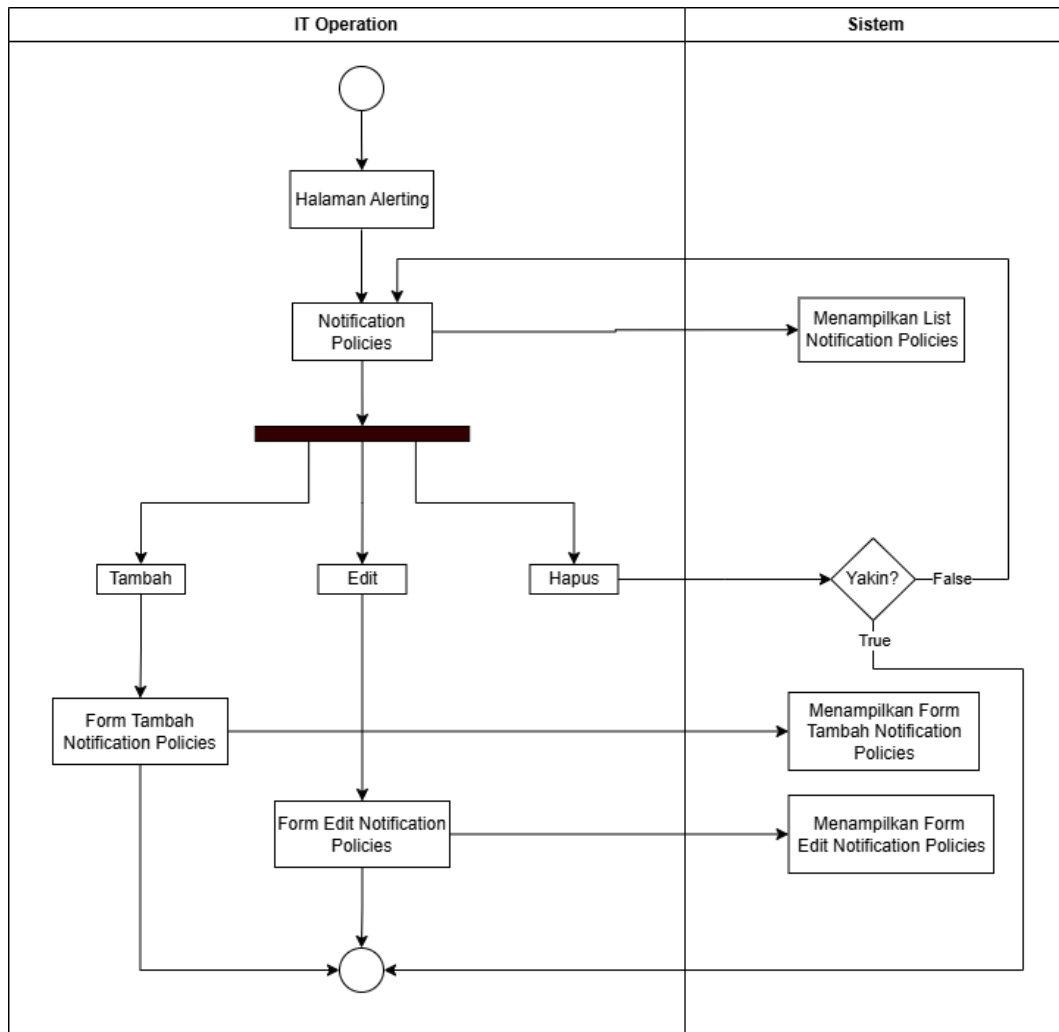
Gambar 3. 8 Kelola *Data Sources*

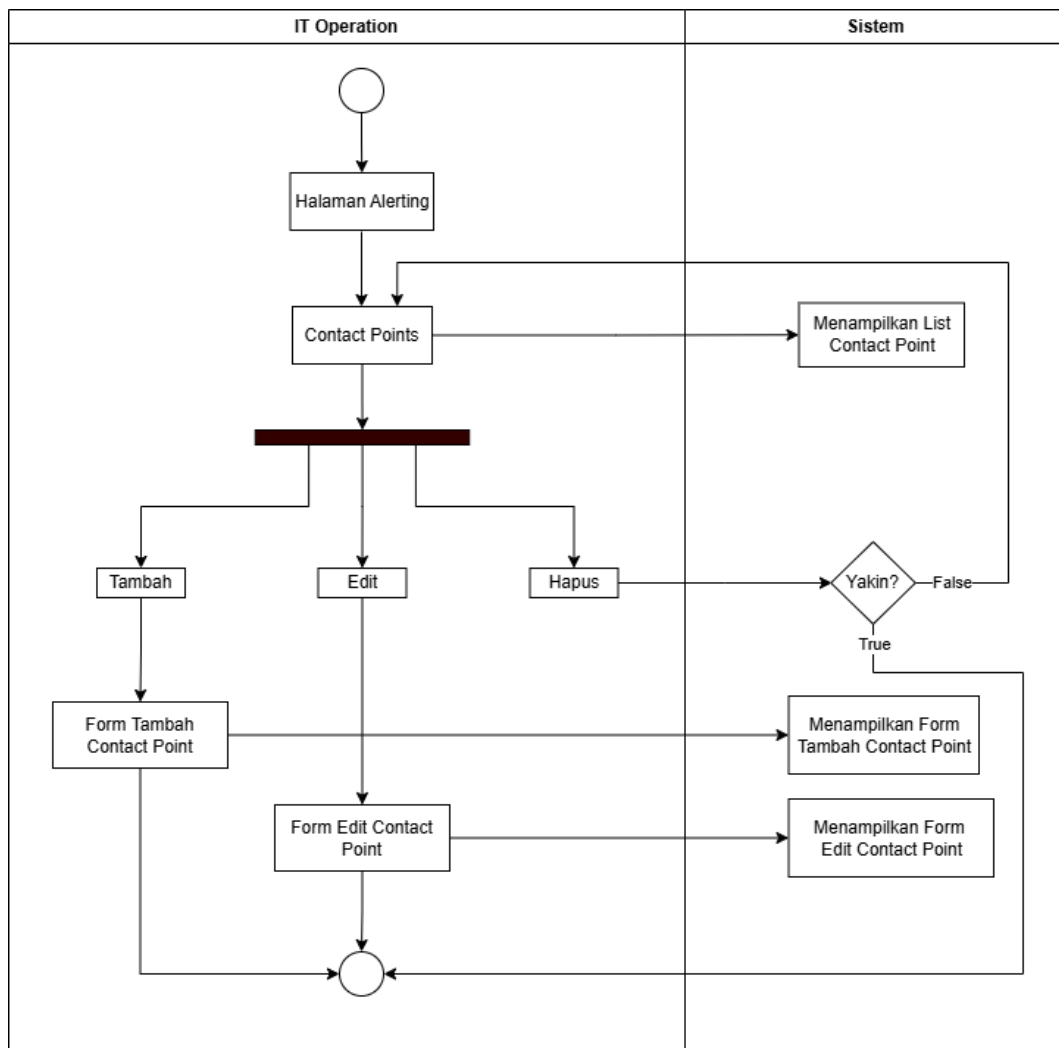


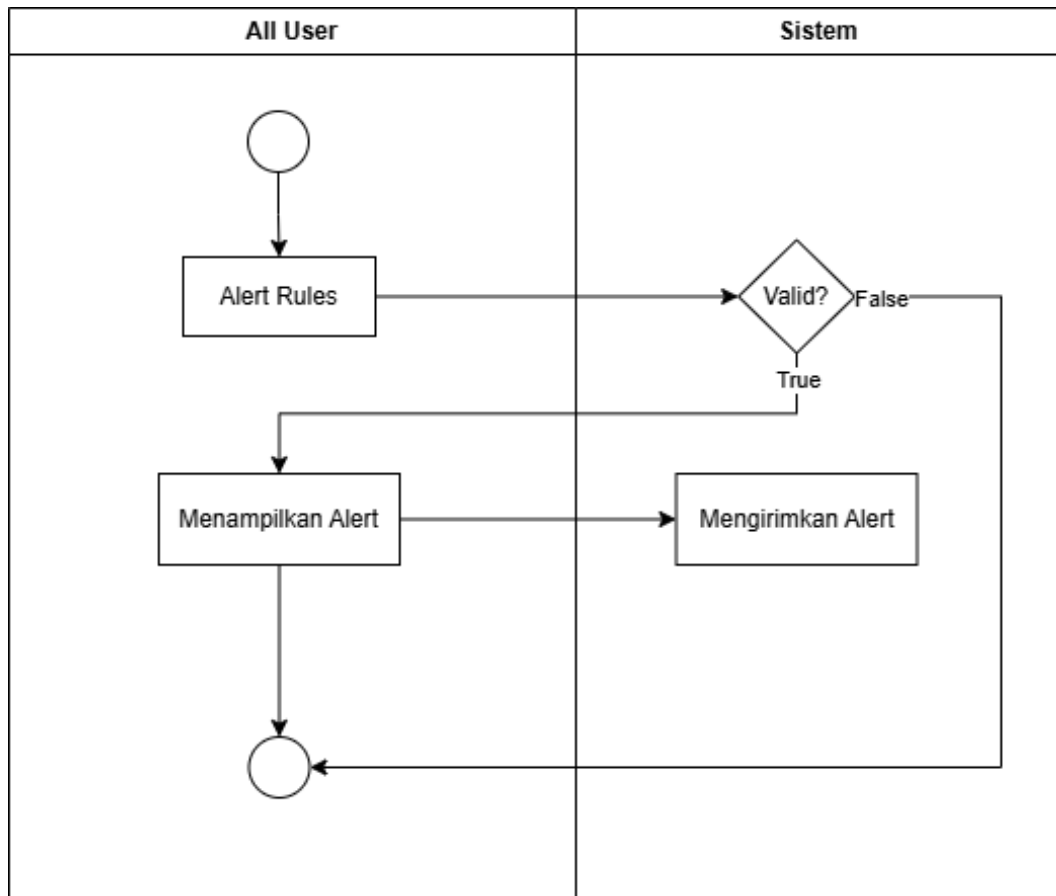
Gambar 3. 9 Kelola Akses Pengguna



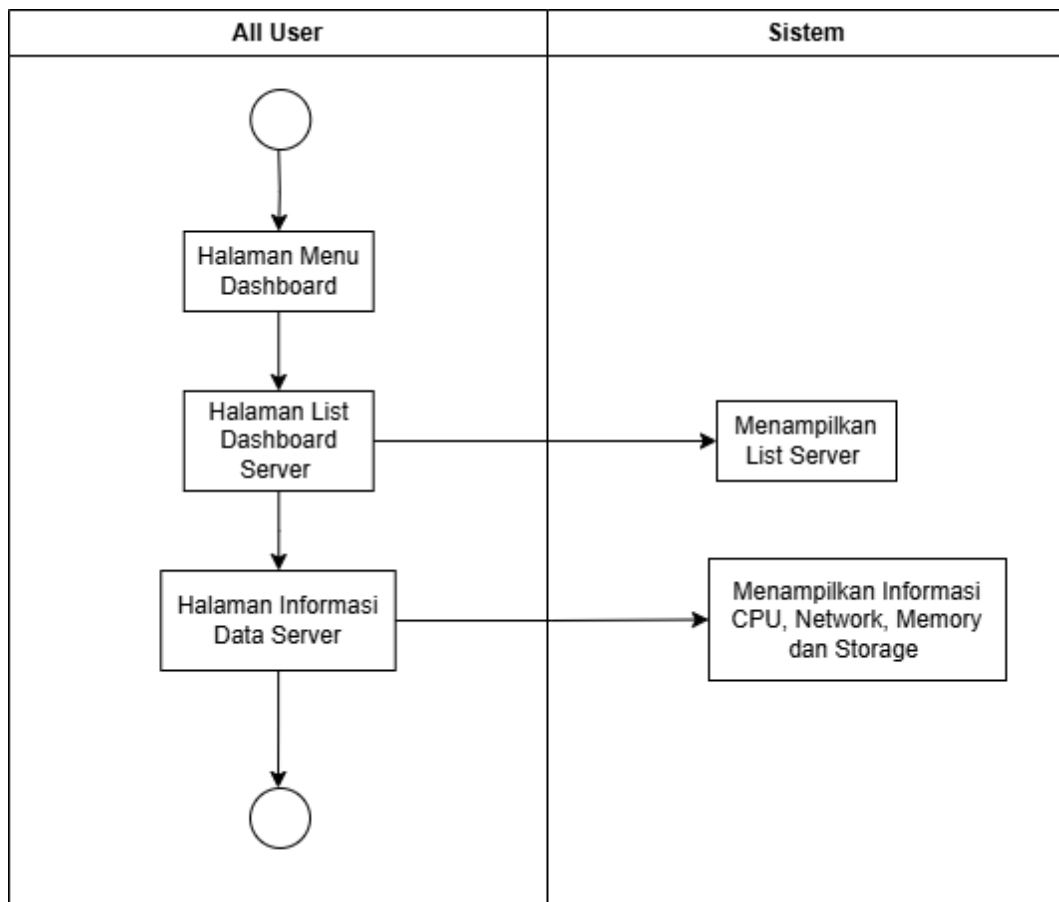
Gambar 3. 10 Kelola Alert Rules

Gambar 3. 11 Kelola *Notification Policies*

Gambar 3. 12 Kelola *Contact Points*

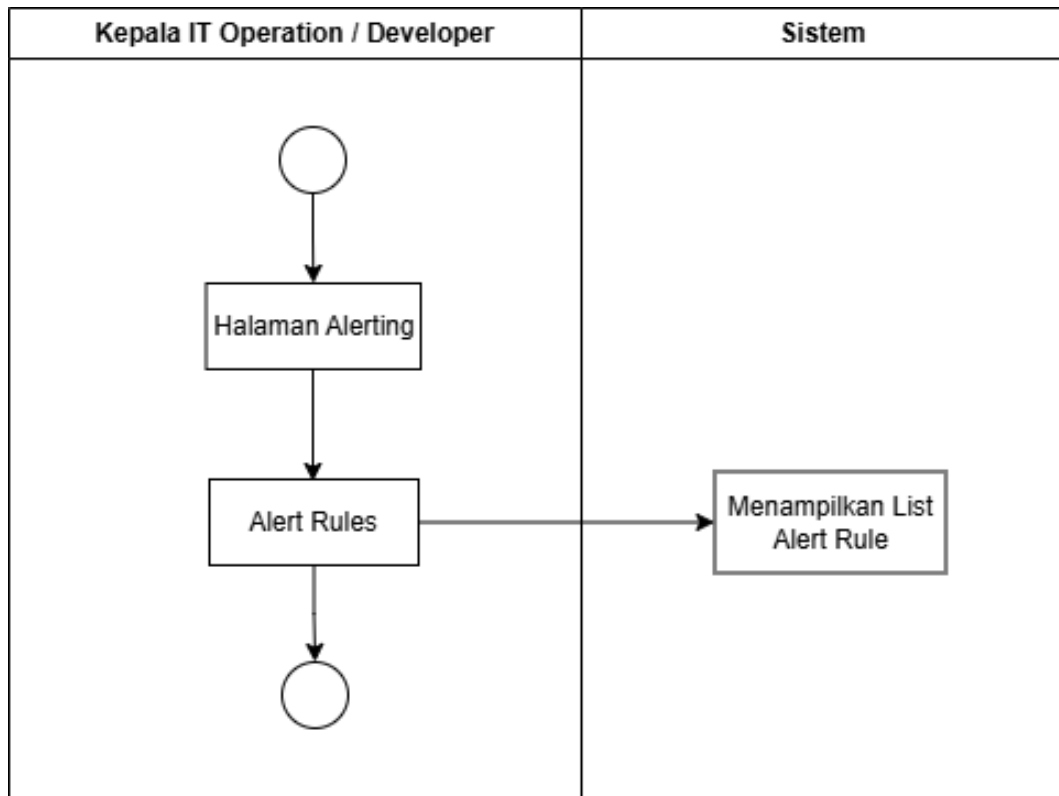


Gambar 3. 13 *Notification Alert*

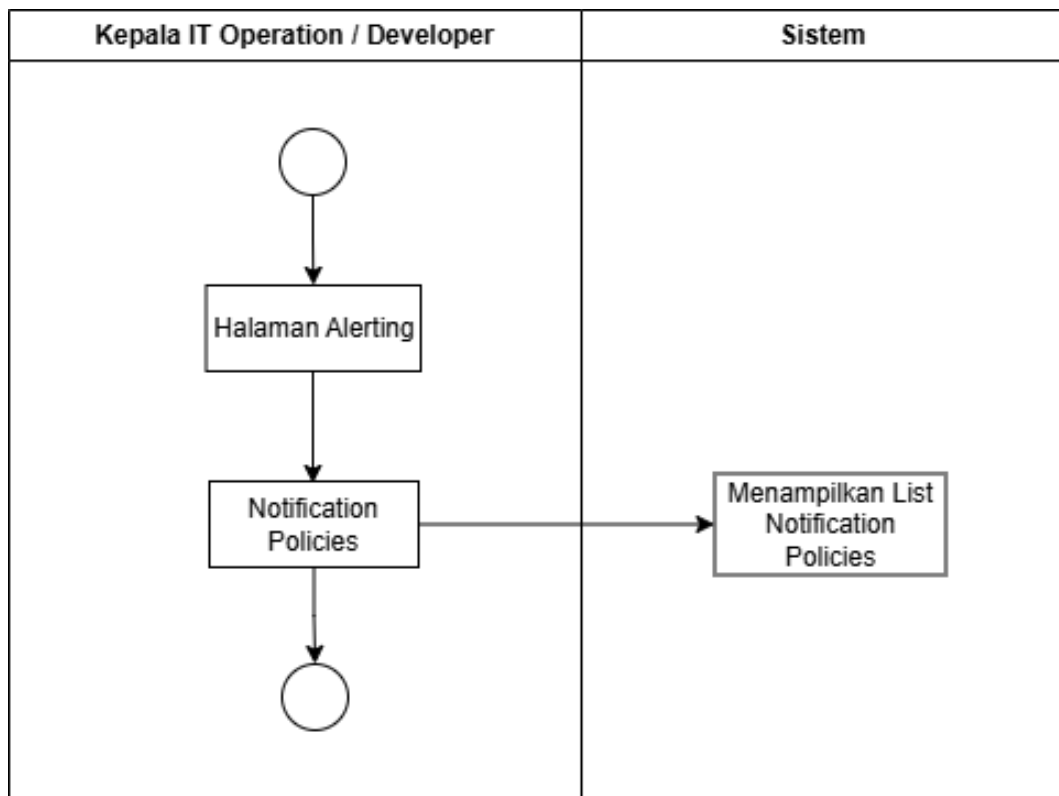


Gambar 3. 14 View Informasi Server

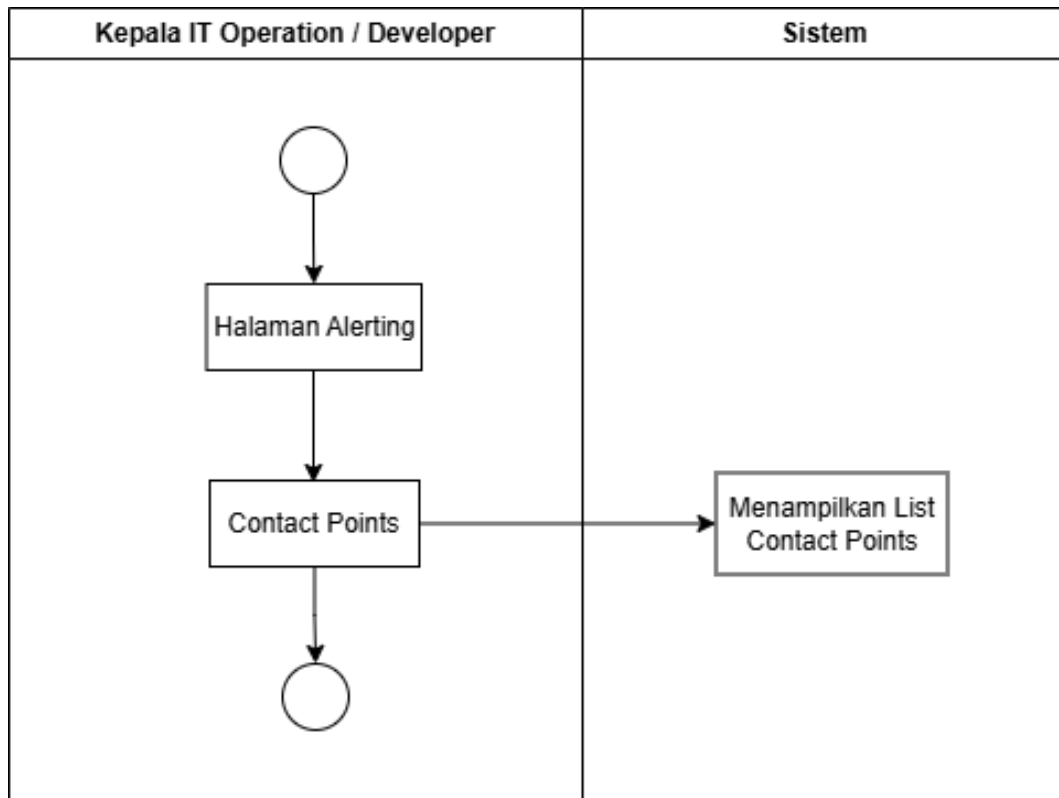




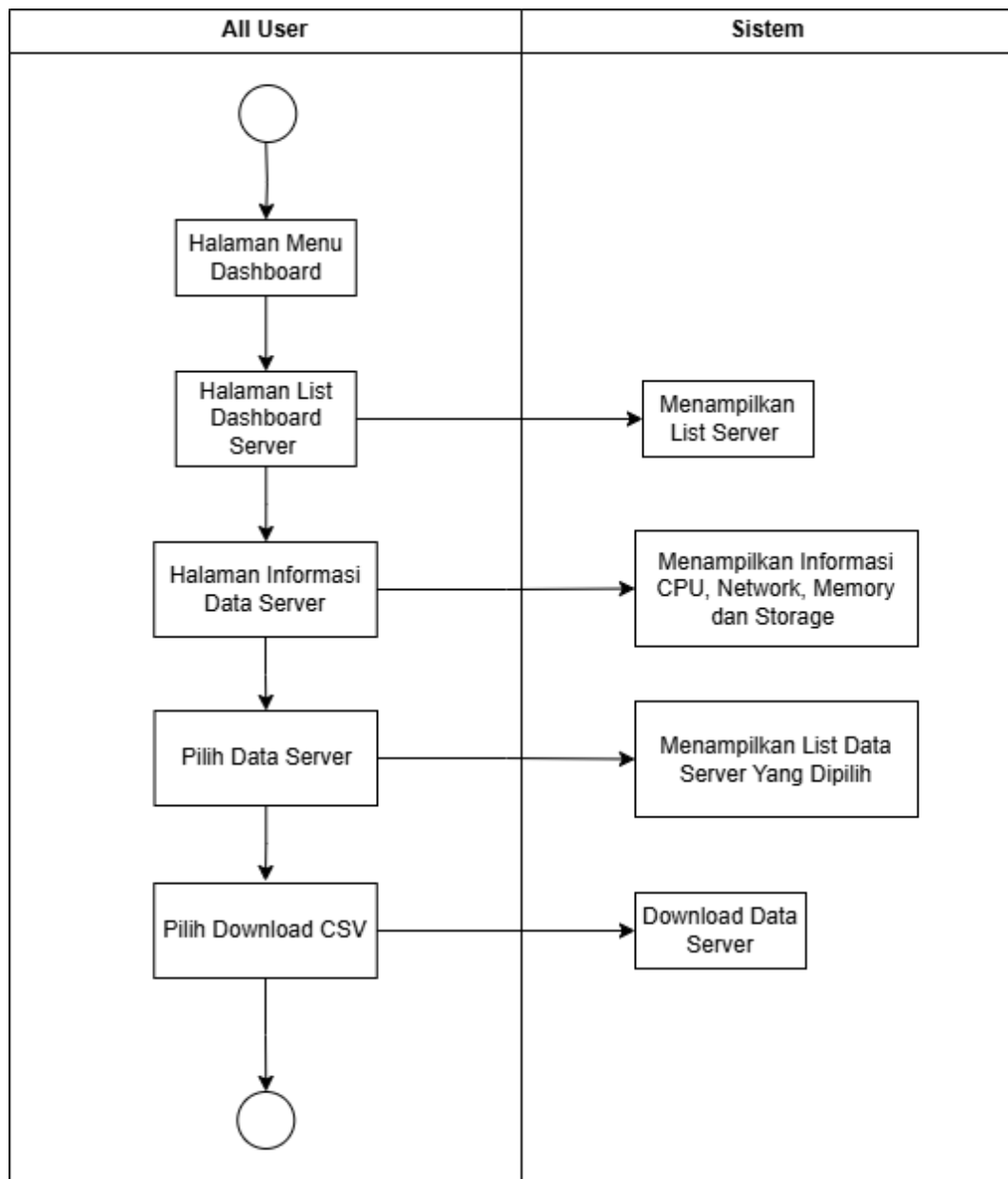
Gambar 3. 15 *View Alert Rules*



Gambar 3. 16 *View Notification Policies*



Gambar 3. 17 *View Contact Points*

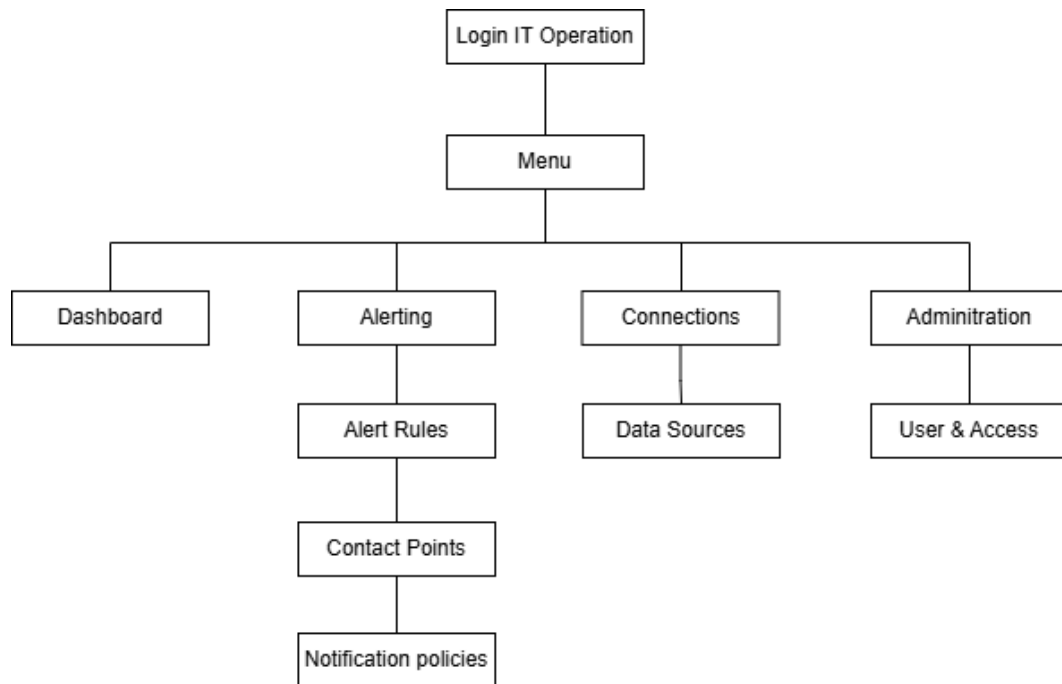


Gambar 3. 18 *Download Report Resource Server*

## B.2. Desain Antarmuka Pemakai (*User Interface Design*)

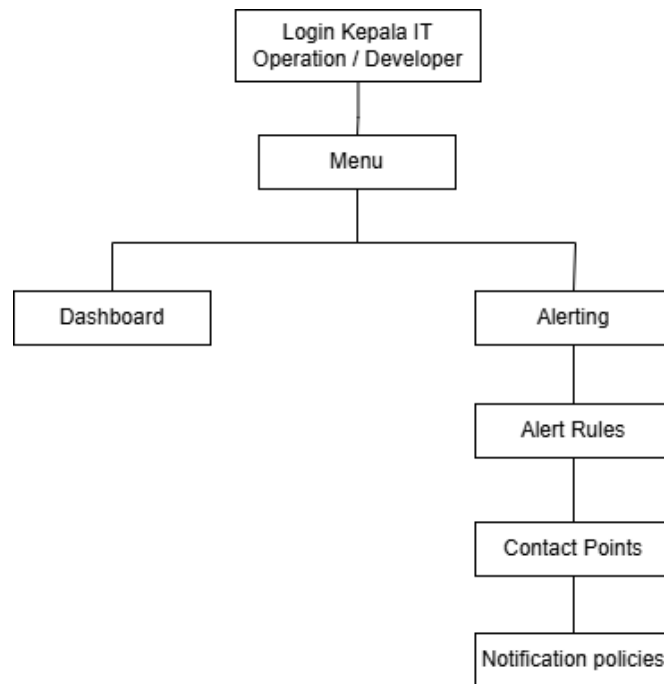
### B.2.1. Struktur Menu

#### 1) Struktur Menu *IT Operation*



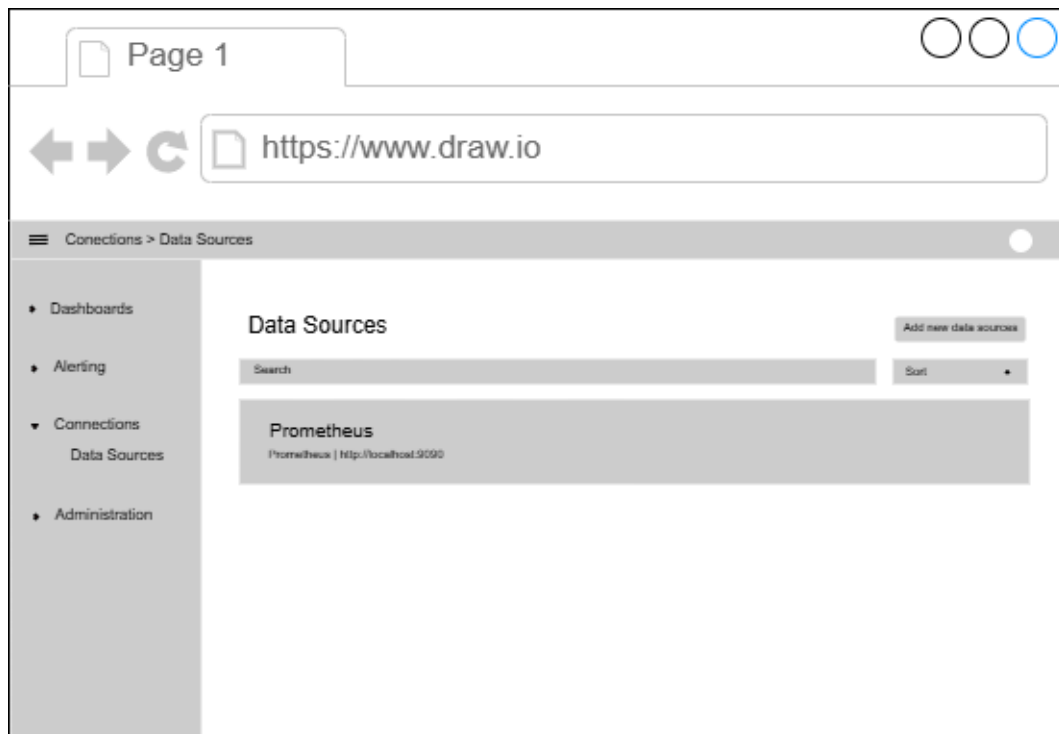
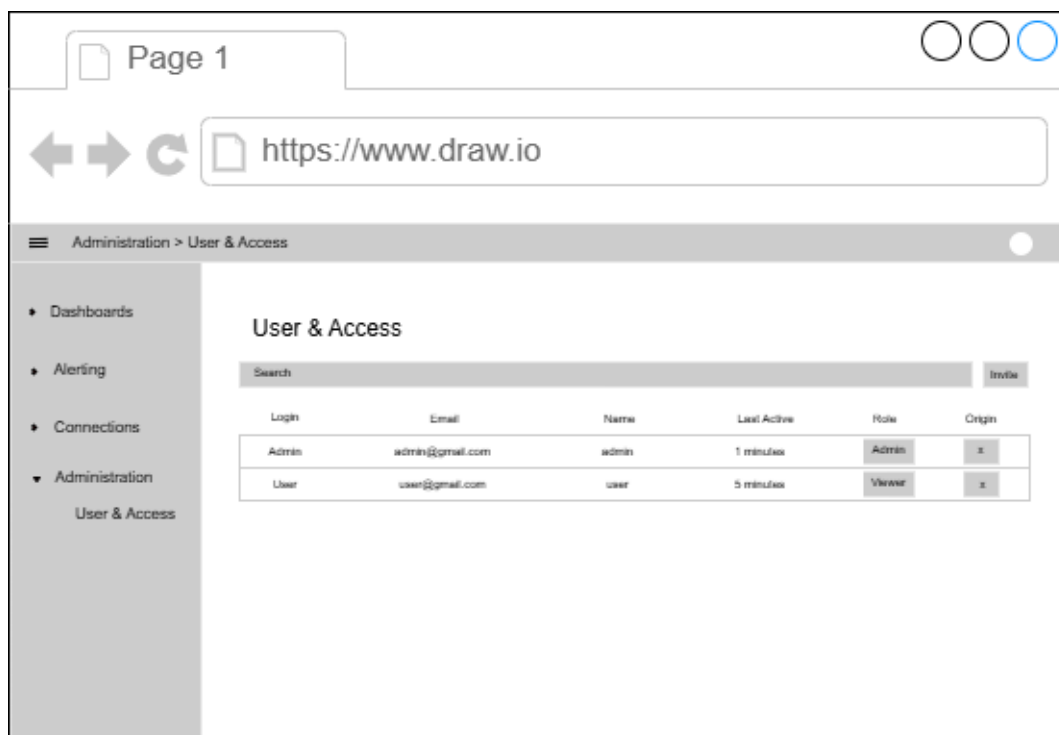
Gambar 3. 19 Struktur Menu *IT Operation*

## 2) Struktur Menu Kepala *IT Operation* dan *Developer*



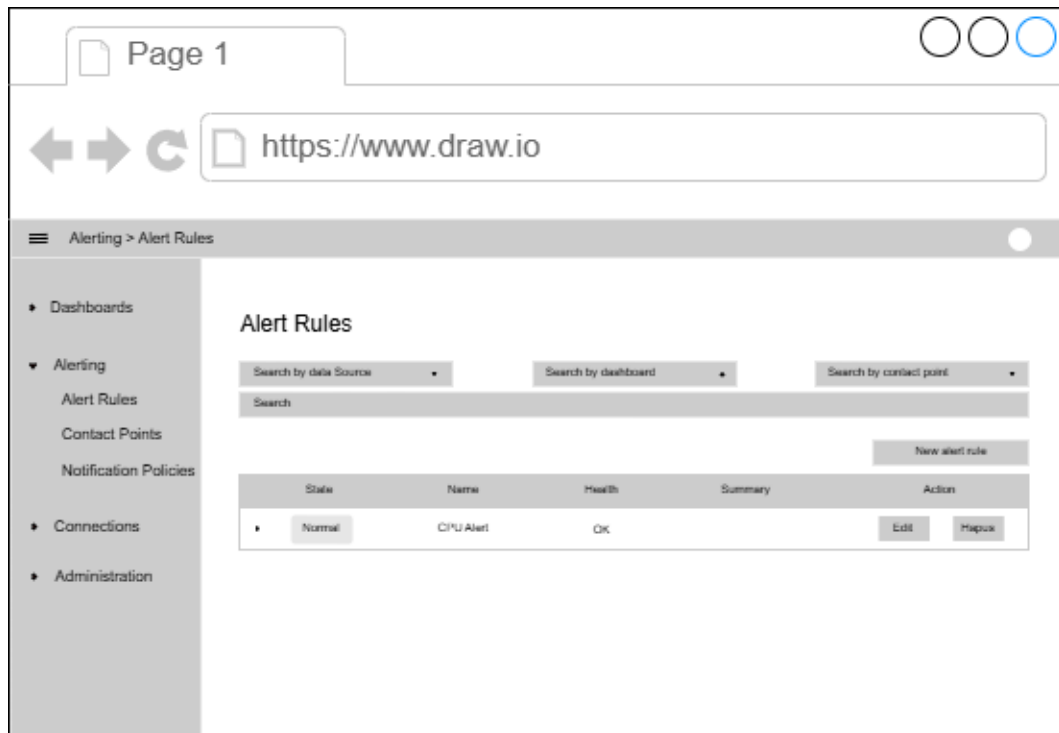
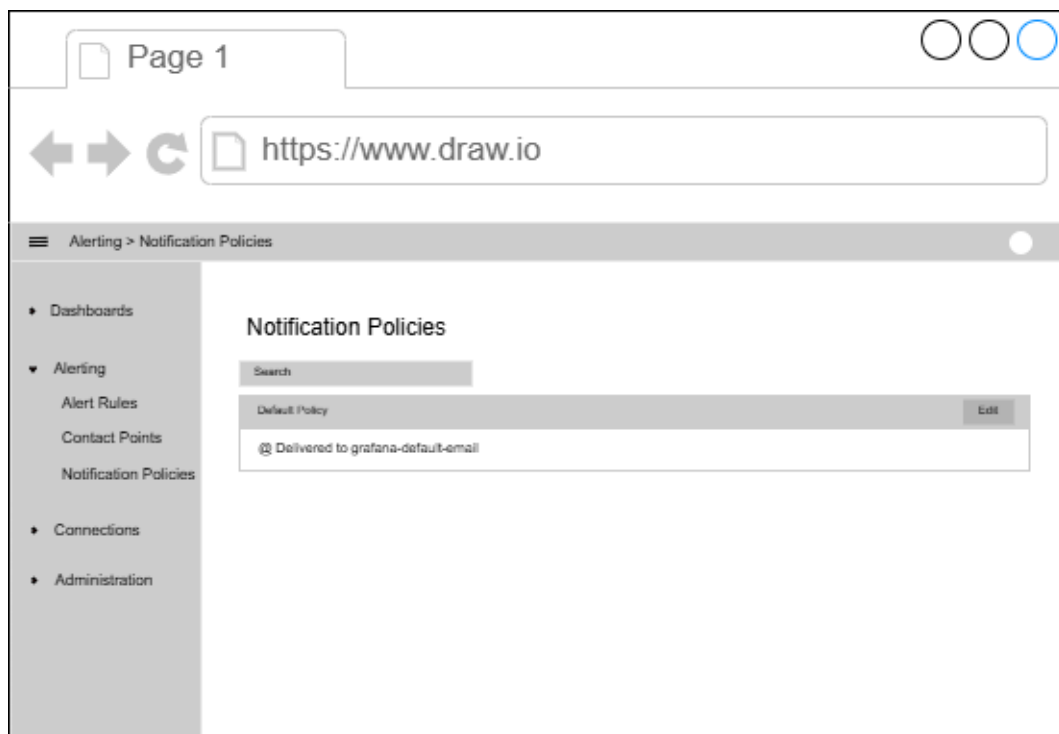
Gambar 3. 20 Struktur Menu Kepala *IT Operation* dan *Developer*

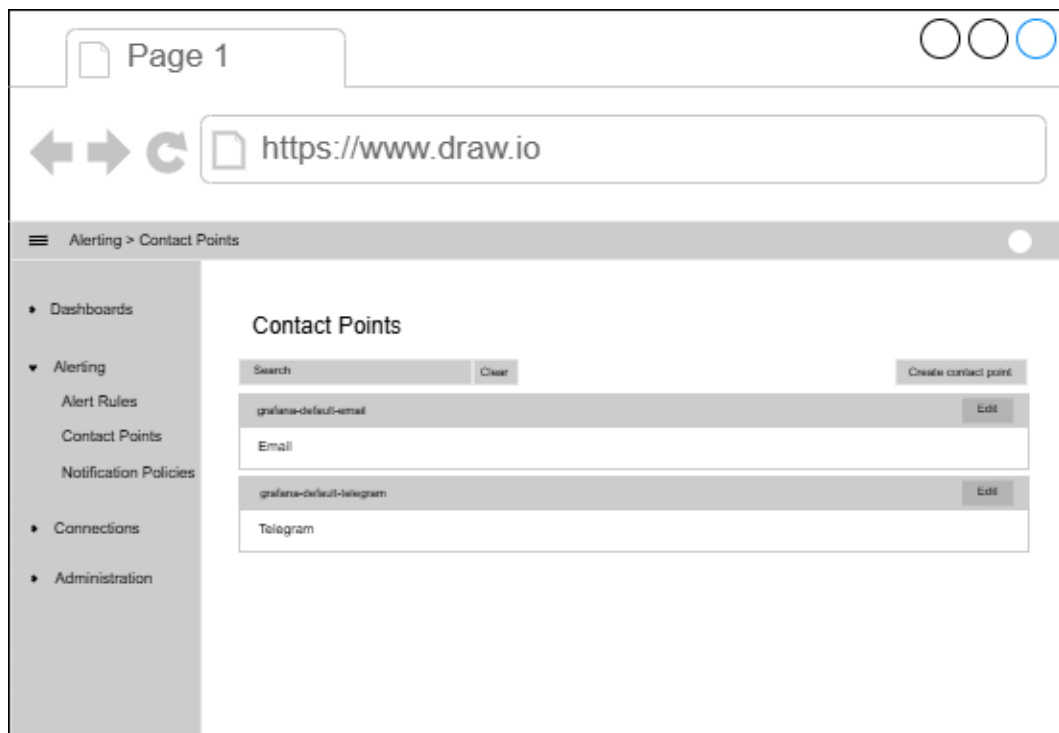
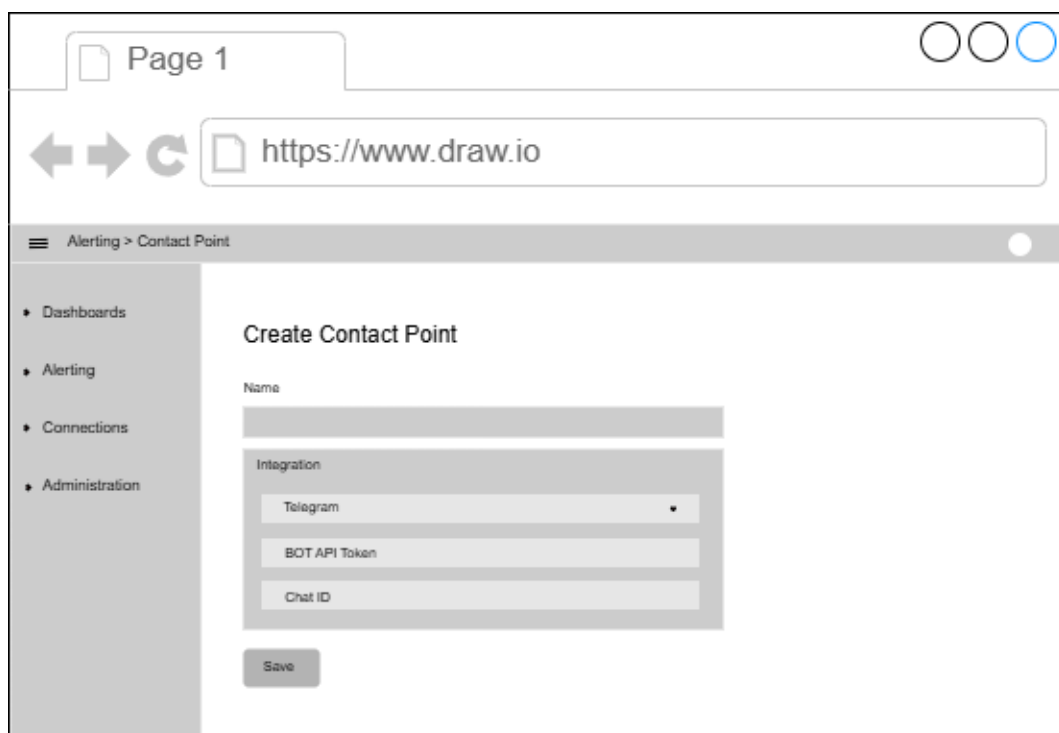


Gambar 3. 24 Halaman *Data Sources*

Gambar 3. 23 Halaman Akses Pengguna



Gambar 3. 25 Halaman *Alert Rules*Gambar 3. 26 Halaman *Notification Policies*

Gambar 3. 27 Halaman *Contact Points*Gambar 3. 28 Halaman Tambah atau Edit *Contact Points*

Page 1

https://www.draw.io

Alerting > Alert Rules > New Alert Rule

### New Alert Rule

Save and Exit

Name

A. Define Query and

Enter a PromQL Query

Add Query

Expressions

B. Reduce

Input A

Function Last Mode Strict

C. Threshold

Input A

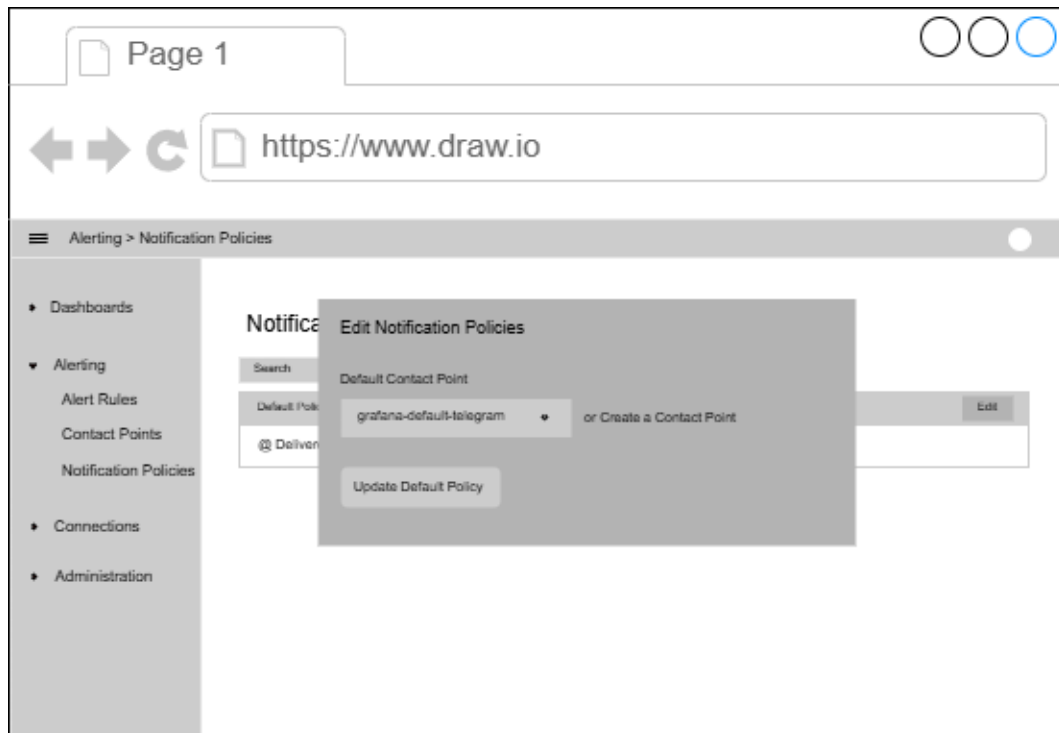
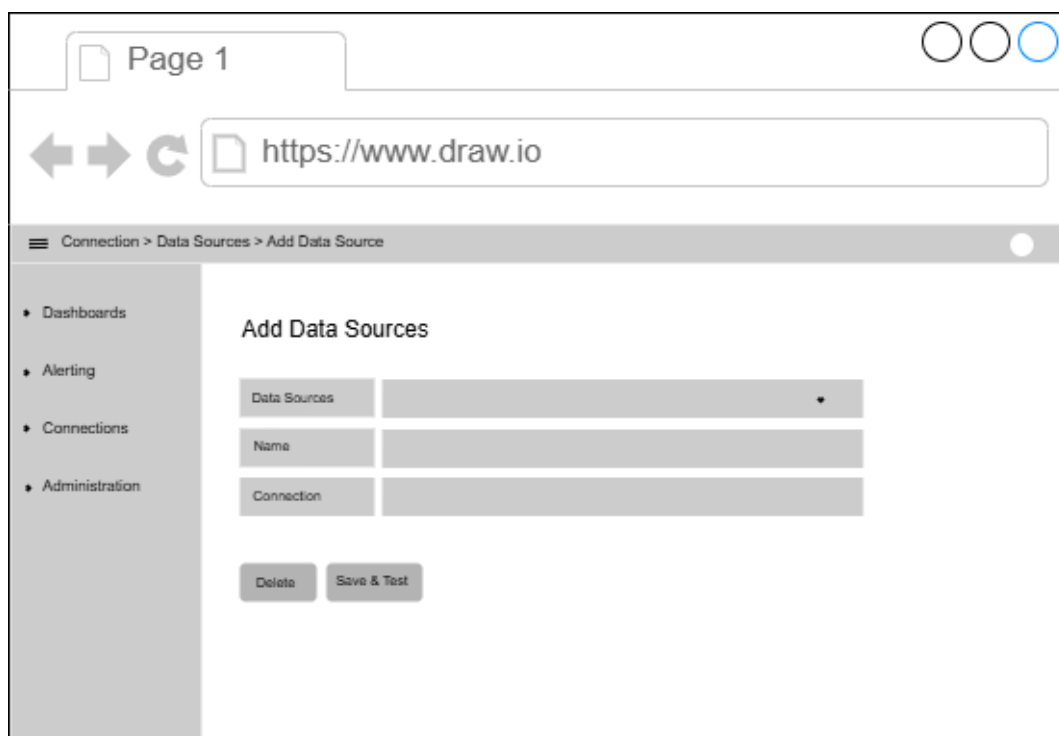
Is Below 0

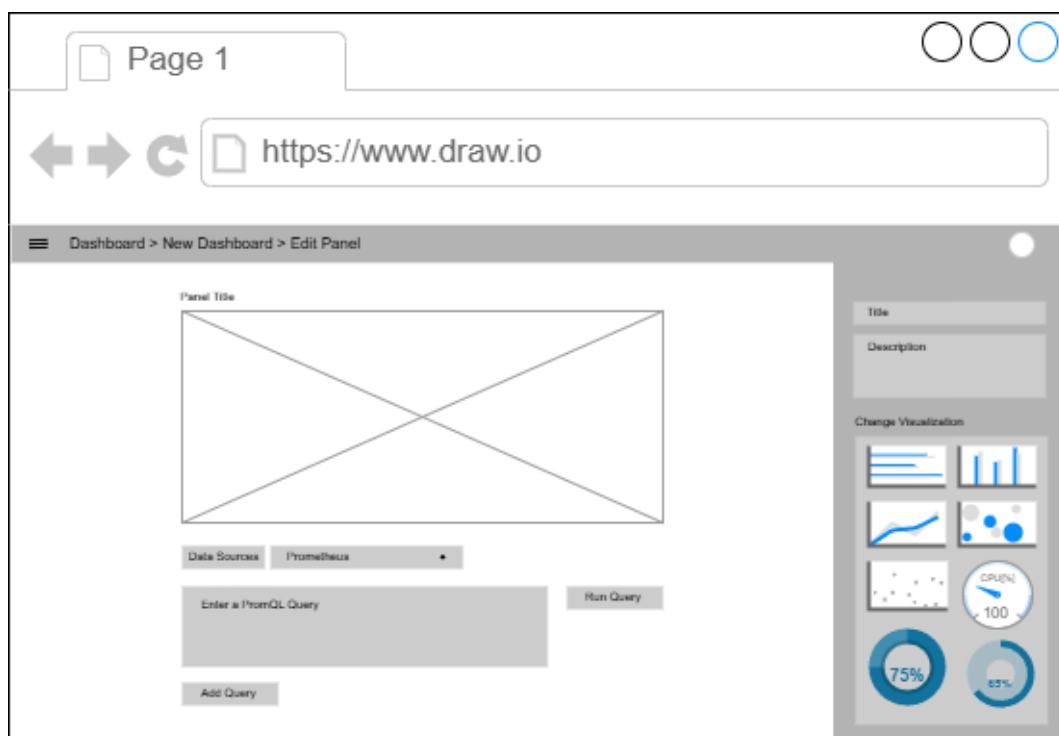
Configure notification message

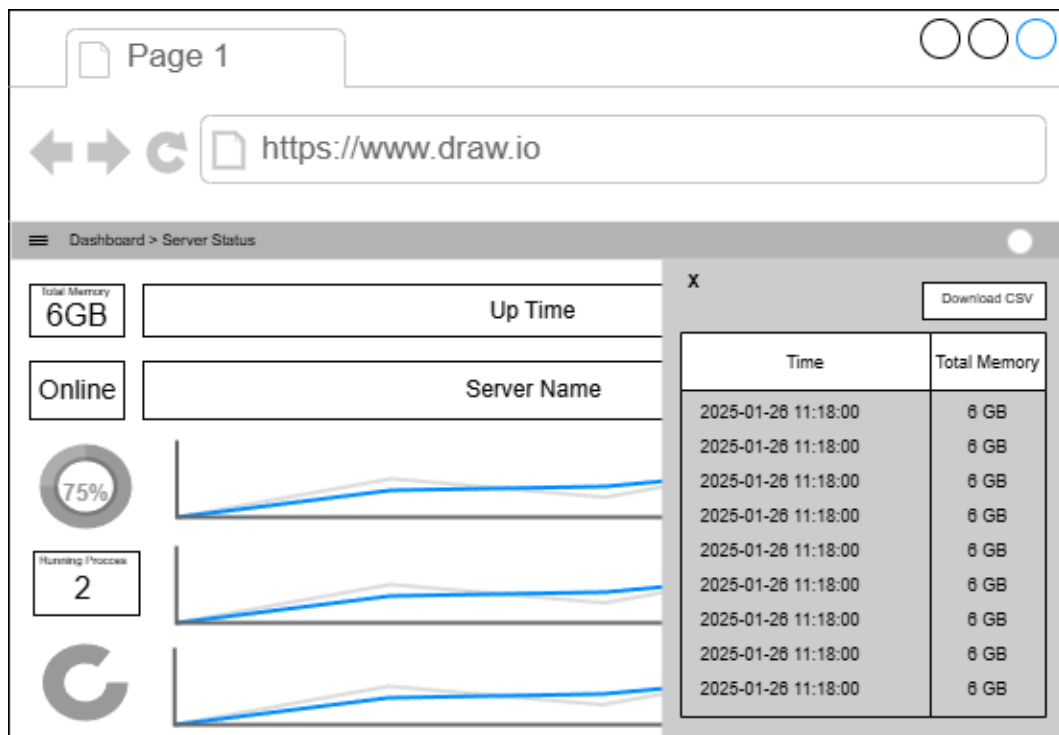
Summary

Description

Gambar 3. 29 Halaman Tambah atau Edit *Alert Rules*

Gambar 3. 30 Modal Edit *Notification Policies*Gambar 3. 31 Halaman Tambah atau Edit *Data Sources*

Gambar 3. 32 Halaman *Invite Users*Gambar 3. 33 Halaman Tambah dan Edit *Panel*



Gambar 3. 34 Halaman *Download Report Resource*

## DAFTAR PUSTAKA

- Adiguna, M. A. (2020). Pemanfaatan SMTP Client pada Sistem Absensi VB.Net. *Jurnal Teknologi Dan Informasi*. <https://doi.org/10.34010/jati.v10i2>
- Adisty C. Putri. (2024). *Apa Itu Mobaxterm dan Bagaimana Cara Menggunakannya*. <https://www.domainesia.com/Berita/Mobaxterm-Adalah/>.
- Aman, M., & Suroso. (2021). Pengembangan Sistem Informasi Wedding Organizer Menggunakan Pendekatan Sistem Berorientasi Objek Pada CV Pesta. *Jurnal Janitra Informatika Dan Sistem Informasi*, 1(1), 47–60. <https://doi.org/10.25008/janitra.v1i1.119>
- Amira K. (n.d.). *Pengertian Server: Jenis, Fungsi dan Cara Kerjanya*. [https://www.gramedia.com/Literasi/Pengertian-Server/?Srsltid=AfmBOop12uT\\_emCwDIjhIJBzWtjisC2kyrGXVMdkwOWaGNEw0N1\\_ozaO](https://www.gramedia.com/Literasi/Pengertian-Server/?Srsltid=AfmBOop12uT_emCwDIjhIJBzWtjisC2kyrGXVMdkwOWaGNEw0N1_ozaO).
- Anjani, M. P. (2022). *DI MTSN 3 Karawang Karawang*.
- Anna Tiffanie. (n.d.). *Apa Itu Server ?* [https://babelprov.go.id/artikel\\_detil/Apa-Itu-Server](https://babelprov.go.id/artikel_detil/Apa-Itu-Server).
- Ayu Aknesia, I. E. A. S. M. U. A. M. B. P. I. U. (2024). *Visualisasi Data Penduduk Desa Sungai Buaya Menggunakan Aplikasi Tableau Public*.
- Aziza, A. N. (2024). *Analisa Pelaporan Gangguan dan Tiket Gangguan Iconnet Menggunakan Aplikasi iCRM di PLN Icon Plus Unit Layanan D.I Yogyakarta*.
- Bastomi, M., Faradita, I., Fitriani, A. D., Putri, R. F., & Agustiningtyas, E. (2024). Pengembangan Bisnis Melalui Pengelolaan Laporan Keuangan UMKM “Ida Collection.” *Altafani Jurnal Pengabdian Kepada Masyarakat*, 4(1). <https://jurnal.insan.ac.id/?journal=altafani>
- Dicoding Intern. (2020). *Apa itu Server ? Berikut Pengertian, Jenis dan Fungsinya*. <https://www.dicoding.com/blog/Apa-Itu-Server/>.
- Febriana, R. M. (2020). *Implementasi Sistem Monitoring Menggunakan Prometheus Dan Grafana*. <https://www.researchgate.net/publication/342511231>

- Febriant Rizaldy, R., Adi, K., Wijaya, S., Putu, R., & Purnamaningsih, E. (2024). Optimalisasi Reformasi Birokrasi Terhadap Kinerja Pelayanan Kelurahan Airlangga Kecamatan Gubeng Kota Surabaya English Title: Optimization of Bureaucratic Reform on Service Performance of Airlangga Urban Village, Gubeng Subdistrict, Surabaya City. In *Ethics and Law Journal: Business and Notary (ELJBN)* (Vol. 2, Issue 1). <http://journals.ldpb.org/index.php/eljbn>
- Fietri Setiawati Sulaeman, I. H. P. (2021). *Sistem Monitoring Penerapan Rencana Anggaran Biaya Berbasis Web*.
- Hergika, G. (2021). *Perancangan Internet Of Things (Iot) Sebagai Kontrol Infrastruktur Dan Peralatan Toll Pada Pt. Astra Infratoll Road*. 8(2). <https://www.esp8266.com/viewtopic.php?p=68657>
- Hidayatullah, A., Hidayat, H., Zidan, M., Fauzi, I., Syafitri, N. A., Jurusan, ), & Komputer, I. (2024). *Impelementasi Sistem Operasi Server Linux Ubuntu untuk Server NAS menggunakan TRUENAS* (Vol. 3).
- Ikhwan Nurrohman. (2024). *Perancangan Dan Impelentasi Sistem Pemantauan Jaringan Berbasis Prometheus Dan Grafana Di Smk Harapan Bangsa*.
- ilham efendi. (n.d.). *Apa Yang di Maksud Dengan Server ?* <https://Www.It-Jurnal.Com/Apa-Yang-Di-Maksud-Dengan-Server/>.
- it.telkomuniversity. (2023). *Simak Pengertian, Sejarah, Kelebihan dan Kekurangan Linux Ubuntu*. <https://It.Telkomuniversity.Ac.Id/Simak-Pengertian-Sejarah-Kelebihan-Dan-Kekurangan-Linux-Ubuntu/>.
- Kusuma, D., & Indrati, A. (2024). Perbandingan Uji Performa Impala dan Hive-Hadoop. In *Syntax Admiration* (Vol. 5, Issue 11).
- Lucky, F. (2021). *Optimalisasi Pencegahan Pencemaran Minyak Dari Kapal Ke Laut Pada Mt. Jeanita*.
- Misbullah, A., Gilang, D., & Rancang, G. ". (2023). *2023 Dalila Husna Yunardi*.
- Nitiwaluyo, K. A. (2021). *Visualisasi Data Project Automatic Vehicle Classification di Cikampek Utama*.
- Nurrohman, I. (2024). *Perancangan Dan Impelentasi Sistem Pemantauan Jaringan Berbasis Prometheus Dan Grafana Di Smk Harapan Bangsa*.



- Orlando Isakh, J., Marcus Zakaria, T., & Maranatha JI Drg Surya Sumantri, K. (2020). *Sistem Informasi Pemantauan Perkembangan Mahasiswa yang Memiliki Kendala dalam Proses Belajar Mengajar* (Vol. 2).
- Rachmat, N., & Arman, M. (2020). Rancang Bangun Aplikasi Penerimaan Peserta Didik Baru Sekolah Swasta Berbasis Web. In *Jurnal Teknologi Informasi Mura Nur Rachmat, Molavi Arman* (Vol. 12).
- Rahman, D., & Amnur, H. (2020). *Indri Rahmayuni 133 Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram Jurnal Ilmiah Teknologi Sistem Informasi* (Vol. 1, Issue 4). <http://jurnal-itsi.org>
- Ramadoni, M. Z. A. R. F. E. U. M. S. M. (2021). *Evaluasi Penggunaan Prometheus dan Grafana Untuk Monitoring Database Mongodb*.
- Ramdany, S. W., Aulia Kaidar, S., Aguchino, B., Amelia, C., Putri, A., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. In *Journal of Industrial and Engineering System* (Vol. 5, Issue 1).
- Rasyidi, B., & Pratama, F. (2024). Sistem Monitoring Server di PT. XYZ Media Indonesia Berbasis Grafana dan Prometheus. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(4), 1456–1465. <https://doi.org/10.57152/malcom.v4i4.1546>
- Riza, N., Rahayu, W. I., Farhan, M. F., Ayuni, R., & Fitri, K. (2024). Sistem Informasi Kewirausahaan Mahasiswa Wau ( Wirausaha Anak Ulbi ) Menggunakan Laravel. In *Jurnal Teknik Informatika* (Vol. 16, Issue 2).
- Septa, F., & Alfia, D. (2022). *Literature Review Visualisasi Data dan Sistem Informasi Geografis Literature Review Visualisasi Data dan Sistem Informasi Geografis Literature Review Data Visualization and Geographic Information Systems 1)\* Famia Septa Dinda Alfia, 2) Agussalim*. <https://doi.org/10.36418/comserva.v2i08.493>
- Sigi Irawan, A. (2024). *Perancangan Smart Home Menggunakan Bluetooth Pada Smartphone Android Dan Arduino*. <https://journal.mediapublikasi.id/index.php/oktal>

- Vindi Eka Safitri, Iqsyahiro Kresna A, & Cahyo Prihantoro. (2024). Penerapan Network Monitoring Menggunakan The Dude Mikrotik dan Notifikasi Pesan dengan Aplikasi Telegram, WhatsApp, dan Email. *Decode: Jurnal Pendidikan Teknologi Informasi*, 4(1), 94–106. <https://doi.org/10.51454/decode.v4i1.200>
- Wulandari, T., & Nurmiati, D. S. (2022). Rancang Bangun Sistem Pemesanan Wedding Organizer Menggunakan Metode Rad Di Shofia Ahmad Wedding. *Jurnal Rekayasa Informasi*, 11(1).
- Yudhy Kusuma, G., & Oktiawati, U. Y. (2022). Perancangan Sistem Monitoring Performa Aplikasi Menggunakan Opentelemetry dan Grafana Stack. *Journal of Internet and Software Engineering*, 3(1). <http://34.128.121.13:5000/v1/campaigns>
- Zulkifli, E. (2022). *Pengaruh Optimalisasi Pembelajaran Online, Partisipasi Mahasiswa Dan Gaya Mengajar Dosen Terhadap Motivasi Belajar Pada Pembelajaran Daring Ditengah Pandemi Covid-19 (Studi Pada Kampus STIE Indonesia Jakarta)*.

## DAFTAR LAMPIRAN

### Lampiran A : Surat Keterangan Riset



#### PT. CONCORD CONSULTING INDONESIA

Gedung GTI, JL. Warung Jati Barat No. 41, Jati Padang, Pasar Minggu Jakarta 12540 Indonesia

[www.concord-consulting.com](http://www.concord-consulting.com)

---

#### SURAT KETERANGAN RISET

Nomor : CNCRD-2506

Yang bertanda tangan dibawah ini :

Nama : Diki Saputra  
Jabatan : IT Department

Menerangkan bahwa:

Nama : Nanang Wahyudi  
NPM : 1101211060  
Program Studi : Teknik Informatika  
Asal Universitas : Universitas Banten Jaya

Telah melakukan riset berupa analisis alur monitoring server di PT. Concord Consulting Indonesia pada tanggal 01 Oktober 2024 hingga 10 Oktober 2024. Riset ini dilakukan untuk mengumpulkan data guna pengembangan sistem monitoring server yang dirancang agar dapat diterapkan di perusahaan serta digunakan oleh Divisi IT Department.

Jakarta, 01 Oktober 2024

IT Department

  
**Diki Saputra**

## Lampiran B : Kehadiran Bimbingan

	<b>FAKULTAS ILMU KOMPUTER</b> <b>UNIVERSITAS BANTEN JAYA</b> Jalan Citaru II No. 73 Telp. (0254) 209583 Serang 42117	
	FORMULIR	F-PRP-03

Lampiran Surat Nomor: 059.6/FILKOM/PP/2024

Nama : Nanang Wahyudi  
 NPM : 1101211060  
 Prodi /Jenjang : Teknik Informatika S-1  
 Pembimbing : Irma Yunita R., S.Si, M.Ti.  
 Batas Waktu Bimbingan : 24 Januari 2025  
 Judul ACC : "Optimisasi Monitoring server Melalui Visualisasi Data Menggunakan Prometheus dan Grafana Di PT. Concord consulting Indonesia"

### CATATAN BIMBINGAN

PERTEMUAN KE TANGGAL	BAHASAN	TTD PEMBIMBING
(24/10/24)	Konsultasi judul	Yf
(25/11/24)	Revisi Bab I	Yf
(14/12/24)	Bab I ok ! Revisi Bab II	Yf
(16/12/24)	Revisi Bab II dan Bab III	Yf
(6/1/25)	Bab II ok ! Revisi Bab III (usecase)	Yf
(16/1/25)	Bab III ok ! Revisi abstrak, DP, KP, dll !	Yf
(27/1/25)	Cek ulang keseluruhan berkas !	Yf
(1/2/25)	Acc berkas proposal skripsi !!!	Yf

Telah melaksanakan bimbingan dan disetujui oleh Pembimbing pada hari sabtu, tanggal 1, bulan Februari, tahun 2025 untuk diseminarkan.

Serang, 1 - 2 - 2025

Pembimbing

Irma Yunita R., S.Si, M.Ti.  
 NIDN: 0421067606

Mengetahui  
 Ka. Prodi,

Ely Nuryani, S.Kom, M.Ti.  
 NIDN: 0416078411

## **Lampiran C : Daftar Riwayat Hidup**

### **DAFTAR RIWAYAT HIDUP**

Nama	: Nanang Wahyudi
NPM	: 1101211060
Tempat Tanggal Lahir	: Tanjung Harapan, 22 Mei 2003
Jenis Kelamin	: Laki-laki
Agama	: Islam
Status	: Belum Menikah
Kewarganegaraan	: Indonesia
Alamat	: Jl. Kiajorum Kp. Kembang RT/RW 002/006 Kel. Curug Kec. Curug Kota Serang Provinsi Banten

### **PENDIDIKAN**

2009 - 2015	: SD Negeri 4 Banding Agung
2015 - 2018	: SMP Negeri 1 Banding Agung
2018 - 2021	: SMA Negeri 1 Banding Agung
2021 - sekarang	: Universitas Banten Jaya

## Lampiran D : Dokumentasi Riset

