

BAB III PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Pada pelaksanaan kerja profesi ini, praktikan berada pada divisi *IT Operation* di PT. Lancar Wiguna Sejahtera dengan jabatan *Network Engineer & Messaging*. Bidang kerja ini mendukung kegiatan operasional dalam mengelola, mengontrol, dan memelihara jaringan kantor pusat, kantor cabang dan toko serta mengelola email perusahaan secara terpusat.

Praktikan bertugas melakukan monitoring dan analisis performa jaringan antar kantor pusat, kantor cabang dan juga toko. Praktikan juga melakukan pengecekan *Service Level Agreement (SLA)* yang digunakan sebagai perhitungan performa dari setiap vendor penyedia internet maupun perangkat jaringan yang sedang digunakan pada PT. Lancar Wiguna Sejahtera sesuai dengan *Standard Operating Procedure (SOP)* yang berlaku, dengan berpedoman pada parameter keamanan dan SLA yang telah ditentukan perusahaan.

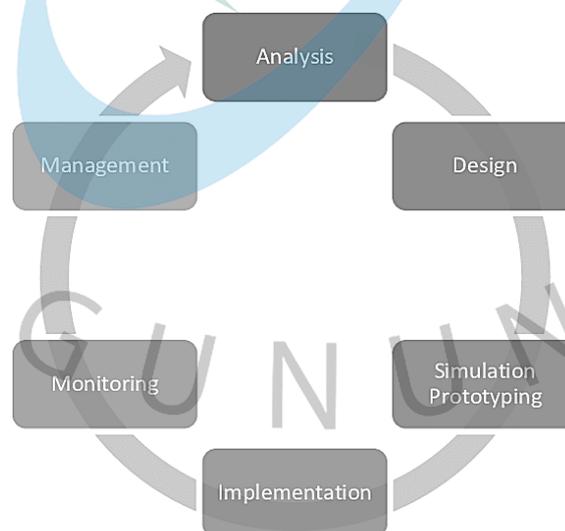
Selain sebagai pengelola jaringan PT. Lancar Wiguna Sejahtera, Praktikan bertugas sebagai pengelola layanan email perusahaan yang menggunakan *platform google workspace*. Praktikan melakukan pengecekan akun aktif, mengatur distribusi grup email antar divisi serta membantu pengguna jika terjadi masalah seperti ganti password, *suspend* akun email dan juga *recovery* akun email karyawan.

Selama masa kerja profesi ini, Praktikan juga memegang tanggung jawab atas proses maintenance berkala, seperti update firmware pada seluruh perangkat jaringan supaya performa perangkat tersebut tetap baik untuk menunjang operasional kantor setiap harinya, praktikan juga berperan dalam pengecekan *log* pada *firewall* dan perangkat jaringan lainnya guna mengetahui anomali yang menjadi penghambat kinerja perangkat.

3.2 Pelaksanaan Kerja

Pada pelaksanaan kerja profesi ini sesuai dengan tanggung jawab praktikan sebagai *Network Engineer & Messaging*, praktikan berfokus dalam peningkatan efektifitas pemantauan performa jaringan yang menghubungkan kantor pusat, kantor cabang dan toko khususnya dalam menghitung ketersediaan (*Uptime & Downtime*) koneksi antar lokasi tersebut.

Dalam kerja profesi ini praktikan mengimplementasikan *Network Monitoring System* (NMS) menggunakan Zabbix dan memvisualisasikannya pada aplikasi Grafana menggunakan metode *Network Development Life Cycle* (NDLC). NDLC merupakan sebuah metode pengembangan yang saling bergantung antara satu tahap dengan tahap sebelumnya. NDLC merupakan sebuah metode yang disebut dengan model kunci dibalik pada proses perancangan jaringan komputer. Untuk memenuhi tujuan bisnis yang strategis, pendekatan top-down harus diambil untuk informasi keseluruhan proses pengembangan sistem. NDLC memiliki beberapa tahapan yang saling terkait, yaitu:



Gambar 3.1 Metode NDLC

a. *Analysis*

Pada tahapan ini praktikan melakukan beberapa kegiatan untuk memahami kebutuhan, yaitu: analisa kebutuhan, analisa

permasalahan yang muncul, analisa keinginan pengguna dan analisa dari topologi jaringan yang sudah ada sebelumnya pada PT. Lancar Wiguna Sejahtera.

Dalam tahapan ini praktikan melakukan wawancara langsung dengan pembimbing kerja untuk memperoleh informasi teknis dan operasional yang lebih detail mengenai kebutuhan sistem monitoring jaringan yang akan diimplementasikan. Selain itu pula praktikan melakukan studi literatur dengan mengacu pada jurnal dan beberapa referensi ilmiah lainnya.

Setelah informasi dan juga data-data sudah terkumpul, praktikan melakukan analisa data yang didapat menjadi rincian spesifikasi perangkat keras dan perangkat lunak yang akan digunakan untuk membangun NMS tersebut. Diputuskan bahwa sistem akan dijalankan menggunakan Proxmox VE, dengan konfigurasi sebagai berikut:

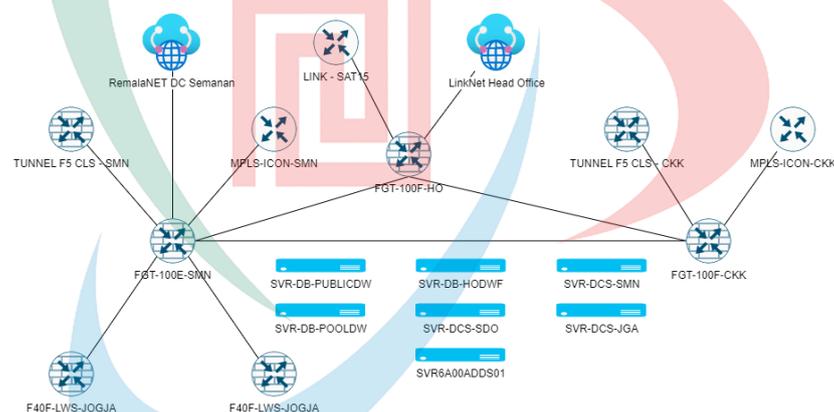
Tabel 3.1 Spesifikasi *Virtual Server* NMS

Nama Perangkat	Spesifikasi
Proxmox	Versi 7.4-18
<i>Operating System</i>	Ubuntu Server 22.04 LTS
<i>CPU Core</i>	4 Core
<i>Memory (RAM)</i>	12 GB
<i>Disk Size</i>	250 GB

Spesifikasi ini telah dipilih dan disepakati secara langsung bersama dengan pembimbing kerja praktikan guna memastikan sistem monitoring dapat berjalan dengan baik dan stabil mampu menangani beban pemantauan secara *real time* dari banyak perangkat jaringan yang ada pada lingkup perusahaan, seperti kantor pusat, kantor cabang dan pada toko lawson.

b. *Design*

Tahapan ini praktikan melakukan pembuatan gambar desain topologi jaringan inti (*Core Network*) yang akan diimplementasikan, diharapkan desain topologi ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada saat ini. Desain topologi tersebut praktikan tidak membangun topologi jaringan baru, praktikan hanya menggambarkan pada jaringan yang telah berjalan saat ini. Berikut adalah topologi jaringan inti (*Core Network*):

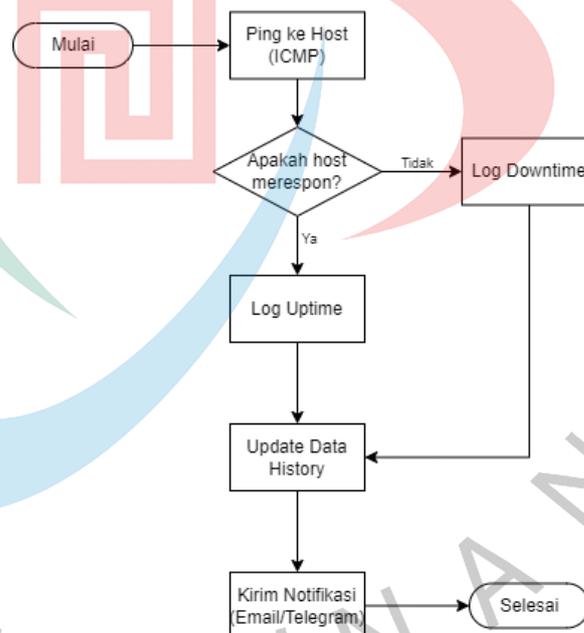


Gambar 3.2 Topologi Jaringan Inti (*Core Network*)

c. *Simulation/Prototyping*

Pada tahapan ini praktikan membuat *flowchart* yang menggambarkan alur kerja sistem secara berkonsep. *Flowchart* ini akan menjelaskan bagaimana kerja Zabbix saat melakukan identifikasi status perangkat melalui protokol ICMP, mencatat status tersebut pada *log database* kemudian memberikan notifikasi yang sudah diatur bisa menggunakan email maupun telegram. Selain itu praktikan juga menggambarkan *flowchart* terkait penambahan *host* yang akan di *monitoring*.

Untuk tahapan proses monitoring status perangkat dalam sistem ini berawal dari pengambilan data oleh Zabbix melalui metode protokol seperti ICMP (ping), SNMP, dan Zabbix Agent pada perangkat yang dipantau/dimonitor, jika sistem mengetahui adanya perangkat yang tidak merespon, maka status perangkat dianggap *down* dan sistem akan menyimpan *log* tersebut pada *database* sekaligus mengaktifkan *trigger* notifikasi. Notifikasi tersebut dikirimkan secara otomatis melalui media yang sudah di konfigurasi. Sebaliknya, jika perangkat kembali aktif maka sistem akan mencatatnya sebagai *up*, dan mengirimkan notifikasi pemulihan (*recovery*) sebagai penanda bahwa koneksi telah normal kembali.



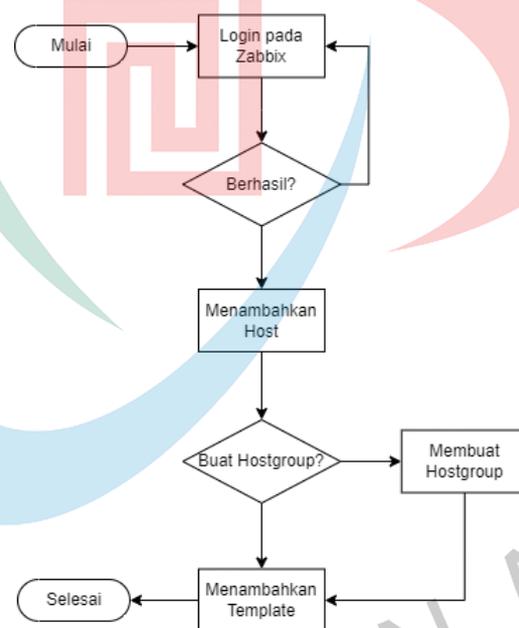
Gambar 3.3 Flowchart proses monitoring Host

Selanjutnya, praktikan akan menjelaskan proses penambahan host pada Zabbix, konfigurasi awal monitoring perangkat jaringan pada sistem monitoring ini diawali dengan masuk ke sistem Zabbix menggunakan *user* dan *password* yang sudah dibuat oleh *administrator*. Langkah berikutnya adalah menambahkan *host* yang akan dimonitor, setelah *host*

ditambahkan maka perangkat tersebut apakah akan dikelompokkan kedalam sebuah *group host* atau tidak dengan tujuan jika dikelompokan tersebut untuk mengorganisir perangkat berdasarkan lokasi atau jenis perangkat.

Setelah *host* dan *host group* sudah ditentukan dan dipilih, selanjutnya yaitu menambahkan *template* yang digunakan untuk kebutuhan monitoring. Template ini berisi kumpulan *item*, *trigger*, dan grafik yang digunakan untuk memantau metrik yaitu seperti status koneksi, *latency*, dan respon ICMP.

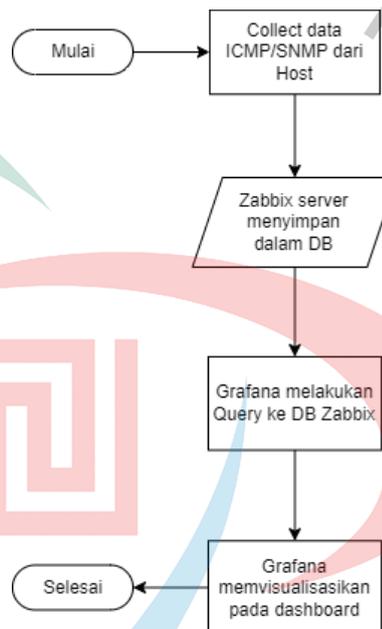
Dengan penjelasan langkah tersebut, perangkat sudah siap secara aktif dimonitor oleh zabbix



Gambar 3.4 Flowchart penambahan Host

Pada tahapan selanjutnya praktikan memberikan gambaran terkait rancangan alur integrasi antara Zabbix dengan Grafana sebagai media untuk visualisasi grafik dan perhitungan SLA yang akan diimplementasikan, proses dimulai dengan pengambilan data status perangkat jaringan melalui ICMP atau SNMP, lalu data tersebut dikirimkan ke server Zabbix dan disimpan kedalam *database*. Selanjutnya Grafana akan akan

mengambil data tersebut menggunakan *query* dari Zabbix *plugin*, dan menampilkannya dalam bentuk visual pada *dashboard* Grafana yang lebih mudah dipahami oleh tim operasional maupun manajemen.



Gambar 3.5 *Flowchart* integrasi Zabbix dengan Grafana

d. *Implementation*

Pada tahapan ini praktikan akan melakukan penerapan semua yang telah direncanakan dan didesain sebelumnya, tahapan ini juga menentukan dari berhasil atau gagalnya proyek yang akan diimplementasi.

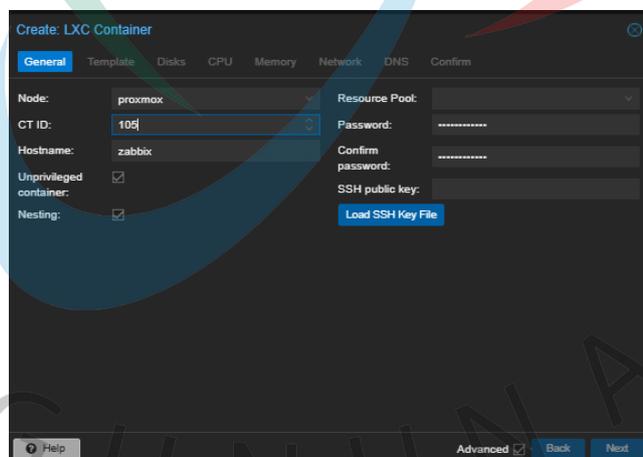
Sebelum melakukan instalasi Zabbix dan Grafana, praktikan akan menyiapkan *Container* pada Proxmox.

Pada Proxmox praktikan memilih menu “*Create CT*” untuk memulai pembuatan *container* dengan konfigurasi yang meliputi :

- Node: dipilih proxmox, yaitu host fisik tempat container akan dijalankan.

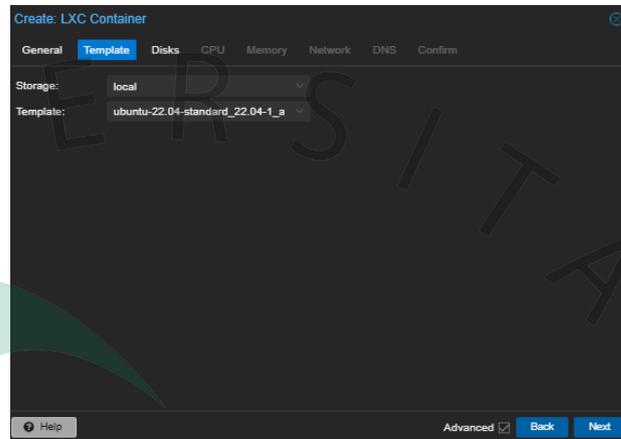
- CT ID: diisi dengan angka unik, misalnya 105, yang berfungsi sebagai identitas container di dalam sistem Proxmox.
- Hostname: diisi dengan nama zabbix untuk memudahkan identifikasi peran container di dalam infrastruktur jaringan.
- *Unprivileged Container*: dicentang untuk alasan keamanan, sehingga *container* berjalan dengan hak akses terbatas terhadap host.
- *Nesting*: juga dicentang untuk memungkinkan container menjalankan fungsi tertentu seperti *docker* atau virtualisasi terbatas di dalam CT, jika dibutuhkan.

Praktikan juga mengatur *password* akses *root* ke dalam *container*, yang akan digunakan saat *login* melalui *console* atau SSH.



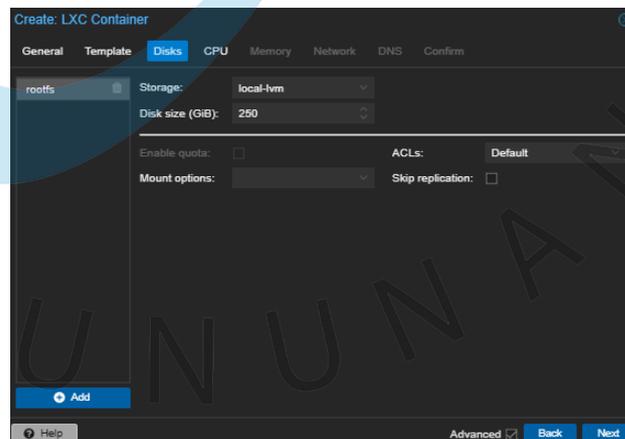
Gambar 3.6 Instalasi *Container General*

Selanjutnya, praktikan memilih *storage* dan *template* yang akan digunakan yaitu *ubuntu-22.04-standard_*.tar.gz* yang merupakan basis sistem operasi yang akan digunakan dalam *container*.



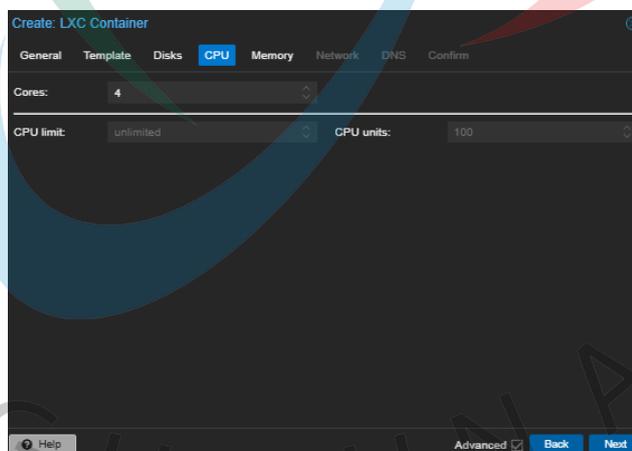
Gambar 3.7 Instalasi *Container Template*

Pada tahap ini, praktikan melakukan konfigurasi penyimpanan untuk *container* sebagai basis instalasi Ubuntu Server. Praktikan memilih jenis penyimpanan "local-lvm" yang merupakan *default* penyimpanan container pada proxmox, kemudian mengatur ukuran *disk* dengan 250GB yang diatur sesuai kebutuhan.



Gambar 3.8 Instalasi *Container Disks*

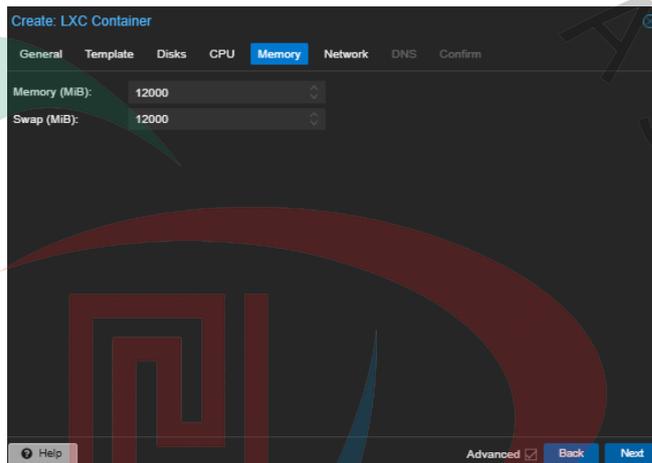
Pada tahap ini, praktikan melakukan konfigurasi penyimpanan untuk *container* sebagai basis instalasi Ubuntu Server. Praktikan memilih jenis penyimpanan "local-lvm" yang merupakan *default* penyimpanan container pada proxmox, kemudian mengatur Praktikan melakukan konfigurasi alokasi sumber daya CPU yang akan digunakan oleh *container* (CT) untuk menjalankan sistem Ubuntu Server dan aplikasi *monitoring* Zabbix. Jumlah core CPU yang dialokasikan adalah sebanyak 4 core, yang dianggap memadai untuk kebutuhan proses monitoring jaringan secara real-time, termasuk pengolahan data, query SLA, dan integrasi visualisasi dengan Grafana. Opsi CPU limit dibiarkan pada kondisi "*unlimited*", artinya container dapat menggunakan seluruh kapasitas 4 core yang diberikan tanpa pembatasan frekuensi atau load maksimum.



Gambar 3.9 Instalasi *Container CPU*

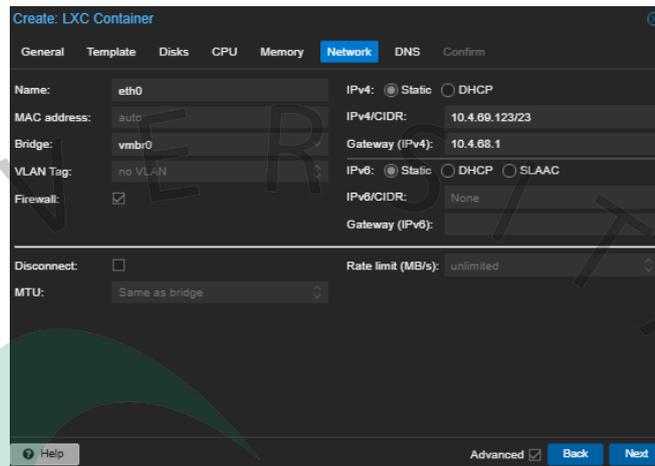
Praktikan mengalokasikan kapasitas memori untuk container yang akan digunakan sebagai server monitoring. Pengaturan dilakukan melalui menu "Memory" pada wizard pembuatan LXC Container di Proxmox VE. Jumlah Memory (RAM) yang dialokasikan sebesar 1200 MiB (sekitar 12 GB), yang dinilai cukup untuk menjalankan sistem operasi Ubuntu Server 22.04 LTS beserta layanan ringan seperti Zabbix Server.

Selain RAM, praktikan juga mengatur *Swap memory* sebesar 12000 MiB, yang bertujuan untuk memberikan ruang cadangan jika penggunaan RAM melampaui kapasitas utama. *Swap memory* berfungsi sebagai memori virtual tambahan untuk menjaga stabilitas sistem ketika terjadi beban kerja mendadak.



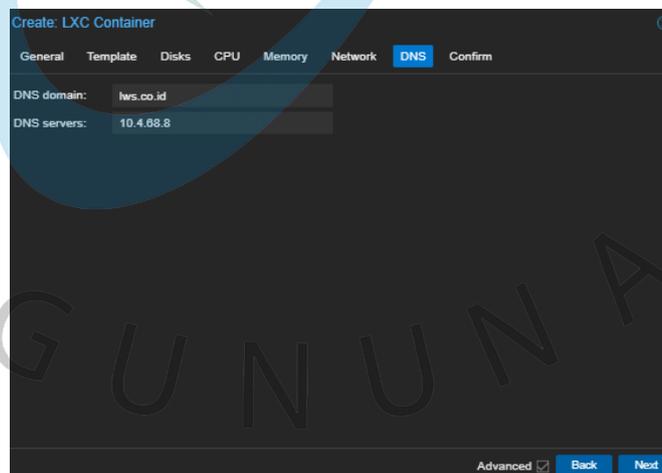
Gambar 3.10 Instalasi *Container Memory*

Pada tahap konfigurasi jaringan, praktikan mengatur parameter konektivitas container agar dapat terhubung dengan jaringan internal perusahaan dan internet. Pengaturan ini dilakukan melalui tab "*Network*" pada *wizard* pembuatan *Container* di Proxmox. Konfigurasi ini memastikan container memperoleh alamat IP tetap (*static IP*), yang penting untuk keperluan server seperti Zabbix agar dapat diakses secara konsisten oleh client atau dashboard. *Container* akan terhubung ke jaringan lokal melalui *interface* *vmbr0*, memungkinkan komunikasi dengan perangkat lain dalam *subnet* yang sama serta akses keluar ke *internet* melalui *gateway* yang telah ditentukan.



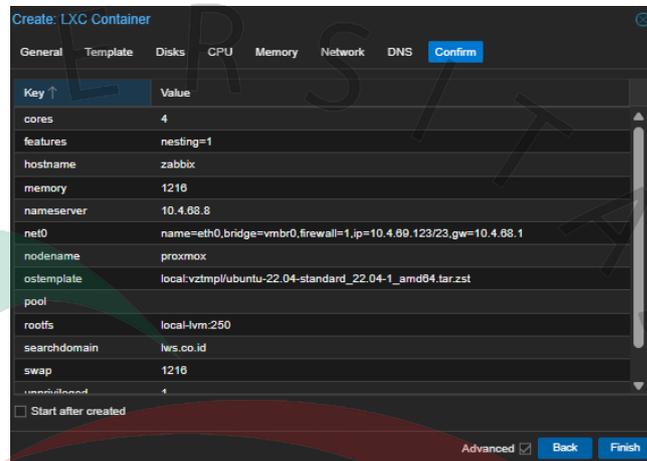
Gambar 3.11 Instalasi *Container Network*

Selanjutnya, melakukan konfigurasi *Domain Name System* (DNS) untuk container yang akan digunakan sebagai server monitoring berbasis Ubuntu. Konfigurasi ini bertujuan agar container dapat menerjemahkan nama domain menjadi alamat IP dengan benar ketika mengakses jaringan internal maupun internet.



Gambar 3.12 Instalasi *Container DNS*

Tahap akhir yaitu konfirmasi atas konfigurasi yang sudah dibuat.



Gambar 3.13 Instalasi *Container Confirm*

Setelah *container* tersebut sudah selesai terbentuk dan bisa digunakan untuk tahap selanjutnya yaitu instalasi zabbix server dan juga grafana, untuk tahap instalasi zabbix dengan langkah awal praktikan menambahkan *repository* resmi Zabbix versi 6.4 untuk sistem operasi Ubuntu 22.04 agar bisa mengakses paket terbaru.

Tabel 3.2 Instalasi *Repository* Zabbix

```
wget
https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.4-1+ubuntu22.04_all.deb
sudo dpkg -i zabbix-release_6.4-1+ubuntu22.04_all.deb
sudo apt update
```

Melakukan instalasi semua komponen yang penting dan dibutuhkan pada zabbix.

Tabel 3.3 Instalasi Komponen Utama Zabbix

```
sudo apt install zabbix-server-mysql zabbix-frontend-php
zabbix-apache-conf zabbix-sql-scripts zabbix-agent
```

Praktikan menginstal seluruh paket penting Zabbix yang meliputi server, frontend (web UI), konfigurasi Apache, script database, kebijakan SELinux, serta agent.

Tabel 3.4 Instalasi Komponen Zabbix Server dengan DNF

```
dnf install zabbix-server-mysql zabbix-web-mysql zabbix-  
apache-conf zabbix-sql-scripts zabbix-selinux-policy zabbix-  
agent
```

Untuk kebutuhan penyimpanan data monitoring yang ada pada zabbix, maka praktikan menginstal MariaDB.

Tabel 3.5 Instalasi MariaDB Server

```
sudo apt install mariadb-server
```

Selanjutnya dibuat *database* dan user khusus untuk zabbix yang akan digunakan.

Tabel 3.6 Konfigurasi Awal Database

```
mysql -u root -p  
mysql> create database zabbix character set utf8mb4 collate  
utf8mb4_bin;  
mysql> create user zabbix@localhost identified by  
'password';  
mysql> grant all privileges on zabbix.* to zabbix@localhost;  
mysql> set global log_bin_trust_function_creators = 1;  
mysql> quit;
```

Praktikan melakukan impor struktur dan data awal zabbix kedalam database yang dibuat sebelumnya.

Tabel 3.7 Impor Skema Database Zabbix

```
zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql  
--default-character-set=utf8mb4 -uzabbix -p zabbix
```

Lalu, praktikan mengubah file konfigurasi pada zabbix server dengan direktori `/etc/zabbix/zabbix_server.conf`.

Tabel 3.8 Pengaturan Password Database

```
DBPassword="password"
```

Tahap terakhir untuk praktikan menjalankan *service* pada zabbix server.

Tabel 3.9 Menjalankan dan Mengaktifkan Layanan Zabbix

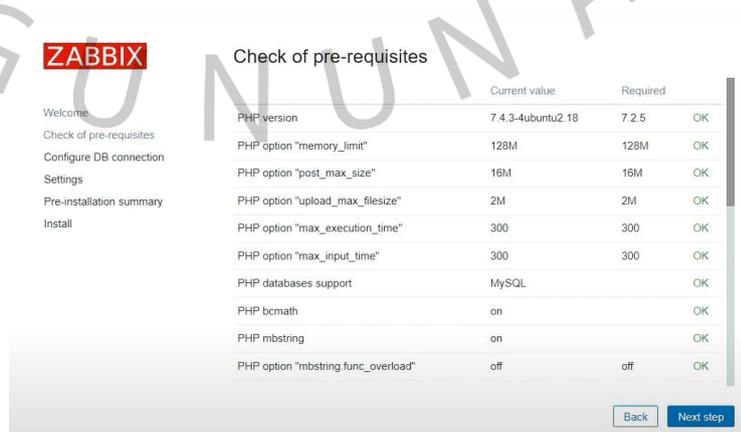
```
systemctl restart zabbix-server zabbix-agent apache2  
systemctl enable zabbix-server zabbix-agent apache2
```

Pada tahap ini praktikan mengakses web sistem monitoring tersebut menggunakan *browser* dengan URL <http://10.4.69.123/zabbix> dan akan muncul tampilan konfigurasi awal dengan tahapan-tahapan, sebagai berikut:



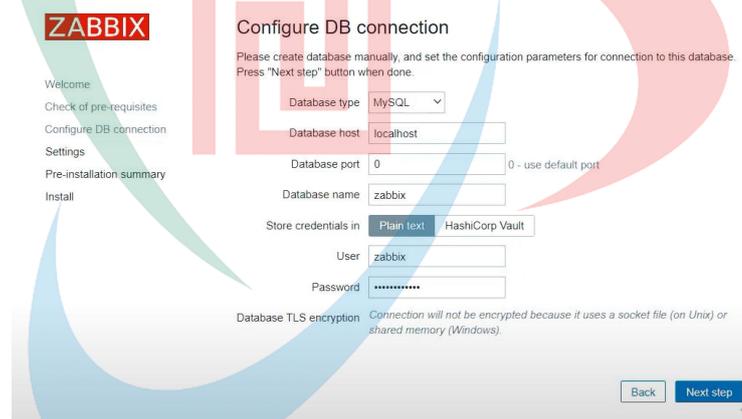
Gambar 3.14 Tampilan Awal Zabbix

Check of pre-requisites adalah pengecekan kelengkapan dan kompatibilitas zabbix server dengan elemen pendukung yang sudah dilakukan instalasi sebelumnya, seperti versi PHP dengan konfigurasi yang tertera pada gambar, database MySQL dan lain sebagainya. Dengan konfigurasi standar yang ditentukan oleh zabbix server tersebut sudah dinyatakan memenuhi syarat maka proses sudah bisa dilanjutkan.



Gambar 3.15 Pengecekan Syarat

Pada langkah *Configure DB Connection* ini, praktikan menginput informasi untuk menghubungkan antarmuka web Zabbix dengan database MySQL yang telah terinstal sebelumnya. Pengaturan dilakukan dengan memilih jenis *database* MySQL, *host localhost*, dan nama basis data zabbix. Nama pengguna dan kata sandi yang dimasukkan sesuai dengan pengguna basis data yang telah diberikan izin akses penuh. Basis data port dibiarkan 0 saja agar menggunakan pengaturan standar pada sistem yang menggunakan port 3306. Koneksi dilakukan secara lokal tanpa enkripsi TLS karena memanfaatkan soket internal. Setelah seluruh data diinput dengan benar, praktikan berlanjut ke tahap selanjutnya.



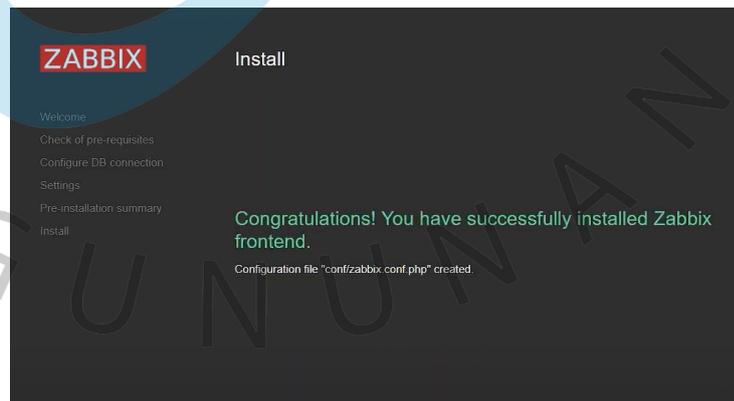
Gambar 3.16 Konfigurasi *Database*

Pada tahap *Settings* ini, praktikan mengubah nama server Zabbix menjadi "NMS-LWS" sebagai identifikasi sistem pemantauan pada *dashboard* zabbix. Zona waktu dikonfigurasi ke UTC, dan tema antarmuka yang dipilih adalah *Dark* sebagai tata letak default. Setelah pengaturan selesai, praktikan melanjutkan instalasi dengan menekan langkah berikutnya.



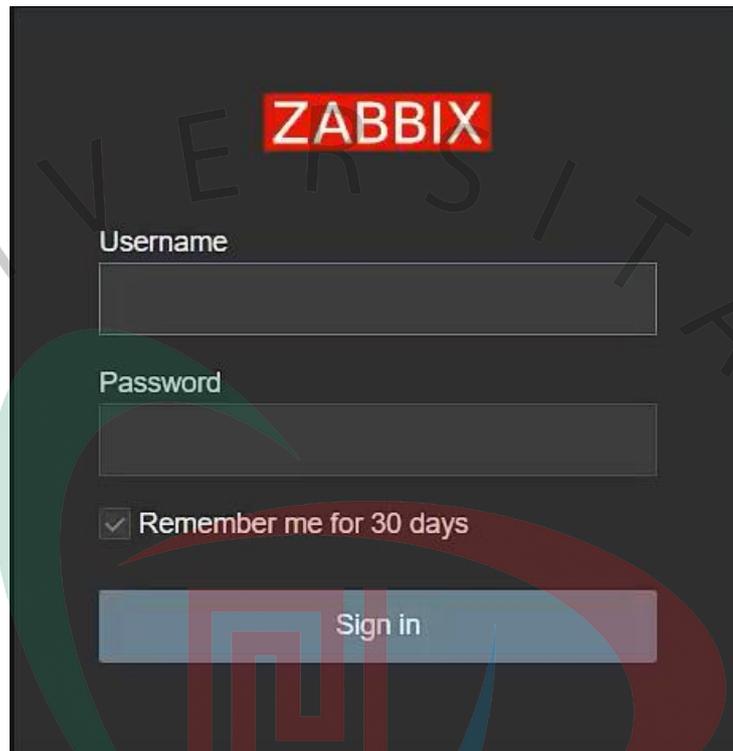
Gambar 3.17 Konfigurasi Settings

Semua proses konfigurasi instalasi zabbix server sudah berhasil tanpa kendala dengan ditandai pesan sesuai pada gambar, tahap ini menghasilkan file konfigurasi utama pada server yaitu `zabbix.conf.php` yang disimpan pada direktori *frontend*, file tersebut berisi seluruh pengaturan yang telah dikonfigurasi pada tahapan sebelumnya.



Gambar 3.18 Konfigurasi Berhasil

Selanjutnya, praktikan mengakses halaman login menggunakan *user* dan *password* yang sudah dibuat sebelumnya, lalu akan menampilkan *dashboard* awal zabbix.



Gambar 3.19 Login Zabbix



Gambar 3.20 Dashboard Awal Zabbix

Langkah selanjutnya, menambahkan *host* ke dalam sistem *monitoring*. Gambar dibawah ini menunjukkan tampilan formulir “*New Host*” yang digunakan untuk menambahkan perangkat atau sistem yang ingin dipantau oleh Zabbix. Pada form ini, praktikan perlu mengisi beberapa parameter penting seperti *hostname*, yaitu nama untuk perangkat yang dipantau, *visible name* jika ingin menampilkan nama lain di *dashboard*, serta memilih *template monitoring* yang cocok dengan jenis perangkat, seperti ICMP ping atau SNMP.

langsung kepada tim operasional saat terjadi status down, up, atau keadaan lain yang telah ditentukan melalui trigger.

Langkah awal yang diambil adalah menciptakan bot Telegram lewat layanan @BotFather. Pengguna memberi nama dan *username bot*, kemudian memperoleh *token API* yang berfungsi sebagai kredensial otentikasi antara Zabbix dan Telegram.

Setelahnya, praktikan memanfaatkan layanan @userinfobot untuk memperoleh chat ID dari akun Telegram yang menerima notifikasi. ID ini diperlukan agar sistem mengetahui tujuan pengiriman pesan.

Selanjutnya, praktikan mengakses web Zabbix dan pergi ke menu Administration > Media types, kemudian menambahkan media type baru dengan jenis Webhook dan nama Telegram. Dalam pengaturan jenis media, praktikan menginput parameter yang diperlukan, seperti:

token : Telegram bot API token

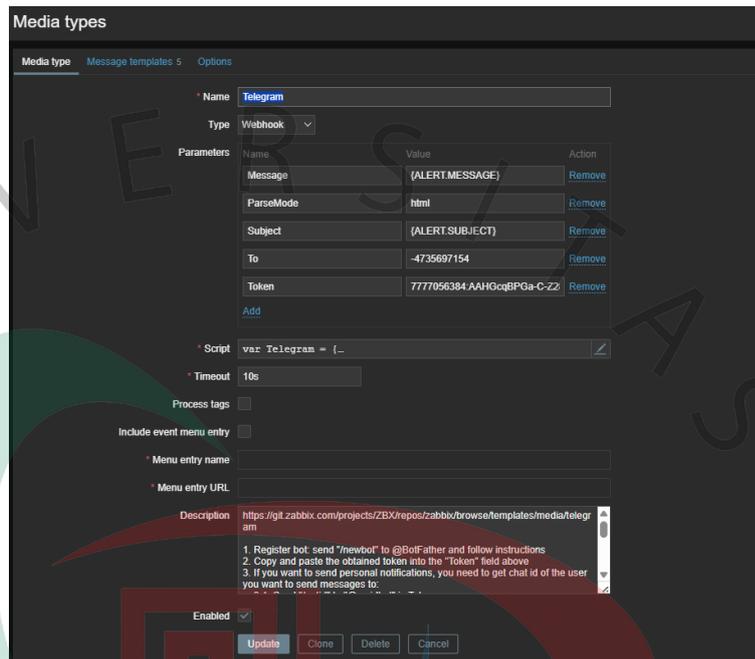
chatid : ID penerima chat

parseMode : html, supaya pesan bisa diformat dengan lebih jelas.

pesan dan subjek : memanfaatkan makro bawaan seperti {ALERT.SUBJECT} dan {ALERT.MESSAGE}

Setelah tipe media disimpan, praktikan menambahkan pengguna media melalui menu Administration > Users, dan menetapkan tipe media Telegram untuk akun tertentu, serta mengatur tingkat keparahan (severity) pesan yang akan diterima, seperti: Peringatan, Tinggi, atau Bencana.

Langkah terakhir adalah membuat aturan aksi melalui menu Konfigurasi - Aksi, yang menetapkan syarat kapan pemberitahuan dikirim, siapa yang menerimanya, serta isi pesan tersebut. Aksi ini umumnya dihubungkan dengan pemicu tertentu, seperti status host tidak aktif atau kehilangan paket melebihi batas.



Gambar 3.23 Konfigurasi Notifikasi Telegram

Setelah semua *host* pada zabbix sudah ditambahkan sesuai dengan kebutuhan, langkah selanjutnya adalah instalasi Grafana sebagai media untuk memvisualisasikan atas data monitoring *host* tersebut.

Instalasi Grafana dilakukan pada container Ubuntu Server 22.04 yang sama dengan Zabbix, untuk efisiensi dan kemudahan integrasi. Proses dimulai dengan menambahkan repositori resmi Grafana, lalu diikuti dengan instalasi paket utama melalui perintah terminal.

Tabel 3.10 Instalasi Paket Utama Grafana

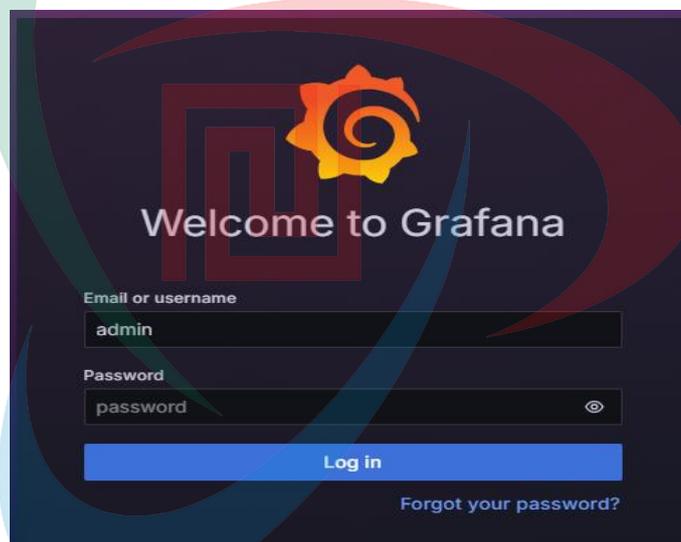
<code>sudo apt install -y software-properties-common</code>
<code>sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"</code>
<code>wget -q -O - https://packages.grafana.com/gpg.key sudo apt-key add -</code>
<code>sudo apt update</code>
<code>sudo apt install grafana</code>

Setelah instalasi selesai, praktikan mengaktifkan dan menjalankan layanan Grafana agar dapat diakses melalui web browser.

Tabel 3.11 Menjalankan dan Mengaktifkan Layanan Grafana

<code>sudo systemctl start grafana-server</code>
<code>sudo systemctl enable grafana-server</code>

Praktikan mengakses web grafana tersebut menggunakan browser dengan URL `http://10.4.69.123:3000` dan akan muncul tampilan login.

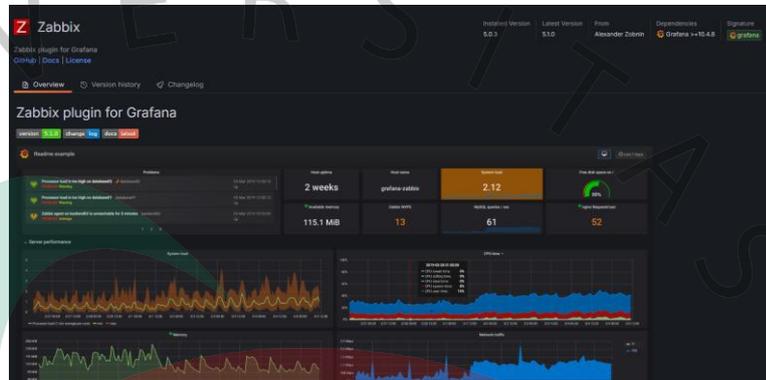


Gambar 3.24 Halaman Login Grafana

Setelah masuk menggunakan *username* dan *password* default, praktikan melanjutkan fase integrasi dengan Grafana sebagai sarana visualisasi. Tujuan dari integrasi ini adalah untuk menyajikan data hasil pemantauan dalam format visual yang lebih informatif, interaktif, dan mudah dimengerti oleh tim operasional serta manajemen.

Proses integrasi dimulai dengan pemasangan plugin Zabbix di Grafana, yang dilakukan baik melalui perintah CLI maupun melalui antarmuka web. Plugin ini memungkinkan Grafana untuk mengambil data dari Zabbix melalui API atau secara langsung dari database. Setelah plugin ditambahkan dengan sukses, praktikan mengakses halaman *Data Sources* di

dashboard Grafana dan menambahkan sumber data baru bertipe Zabbix.



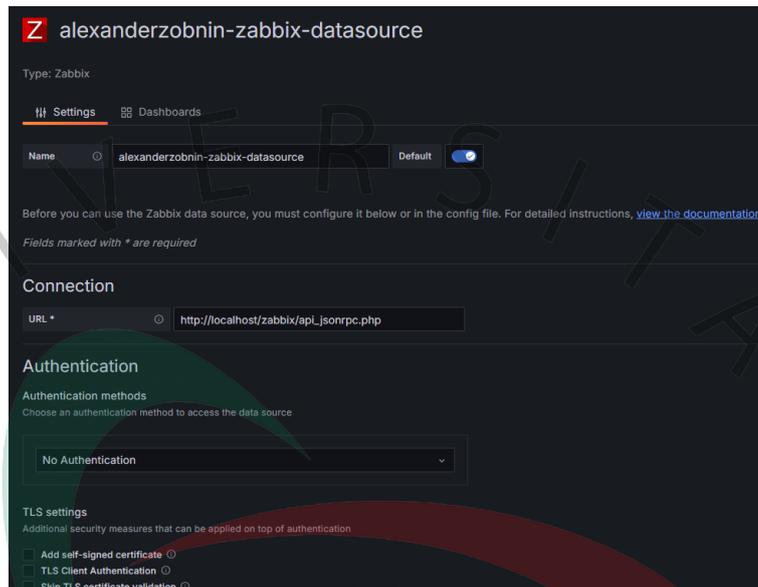
Gambar 3.25 Plugin Zabbix

Dalam pengaturan sumber data, praktikan mengisi parameter yang dibutuhkan, seperti: URL API Zabbix, yang mengarah ke lokasi frontend Zabbix (contohnya: [http://\[ip-address\]/zabbix/api_jsonrpc.php](http://[ip-address]/zabbix/api_jsonrpc.php))

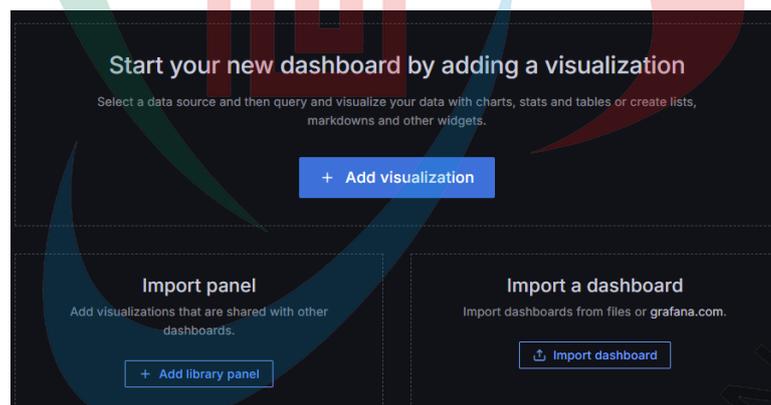
Nama pengguna dan kata sandi Zabbix, untuk otentikasi. Versi Zabbix dan dukungan tren, disesuaikan dengan versi yang terinstal.

Setelah konfigurasi disimpan dengan sukses dan koneksi teruji (status "Sukses"), praktikan melanjutkan ke pembuatan dashboard baru. Dashboard ini dibuat untuk memamerkan metrik utama seperti:

- Kondisi koneksi host (UP/DOWN)
- Rata-rata SLA setiap hari/bulan
- Grafik waktu henti dan waktu aktif



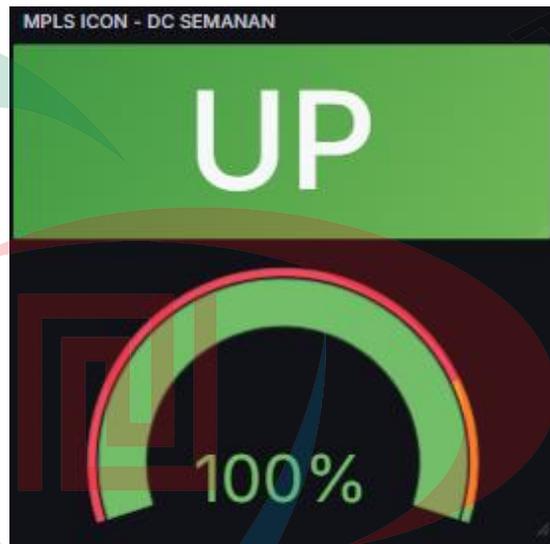
Gambar 3.26 Konfigurasi *Datasource* Grafana



Gambar 3.27 Penambahan Dashboard Grafana

Salah satu hasil utama dari sistem pemantauan jaringan yang dikembangkan adalah visualisasi keadaan koneksi host (*UP/DOWN*) dalam dasbor Grafana. Visualisasi ini menyajikan tampilan *real-time* mengenai keadaan perangkat jaringan yang diawasi, baik di kantor pusat, cabang, maupun gerai. Untuk menampilkan rata-rata SLA (*Service Level Agreement*) jaringan

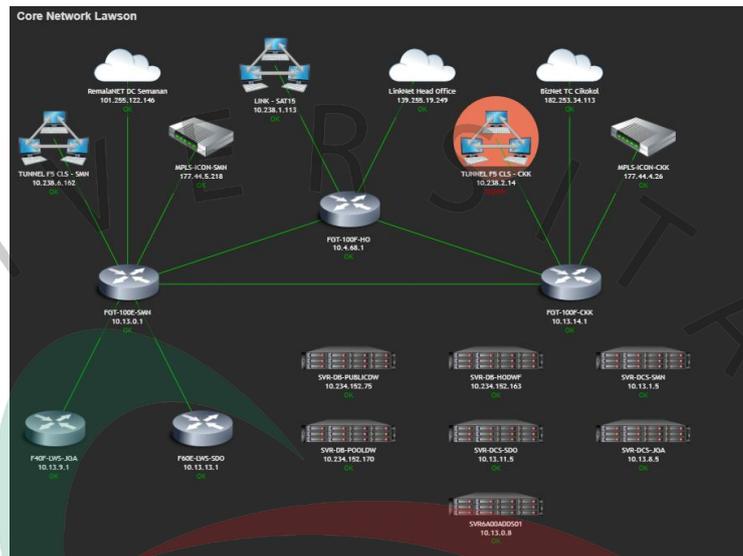
dengan cara yang lebih intuitif dan informatif, praktikan memanfaatkan gauge chart di dalam *dashboard* Grafana. Grafik gauge dipilih karena dapat menyajikan tampilan visual yang jelas dan langsung menampilkan seberapa baik performa konektivitas jaringan dalam bentuk persentase.



Gambar 3.28 Status Perangkat dan SLA

e. *Monitoring*

Tahapan ini dalam metode NDLC adalah sebagai tahapan pemantauan performa dan kestabilan perangkat jaringan secara berkelanjutan, khususnya fokus pada optimalisasi pengukuran dan evaluasi SLA.



Gambar 3.29 Map Topologi Zabbix

Sistem pemantauan dibangun dengan Zabbix sebagai platform utama untuk pengumpulan data, dan Grafana sebagai alat untuk visualisasi. Zabbix secara rutin melakukan polling terhadap perangkat jaringan dengan menggunakan ICMP (ping) dan SNMP, serta menyimpan status koneksi setiap perangkat dalam basis data internal. Data itu mencakup waktu operasional (uptime), waktu tidak berfungsi (downtime), dan status perangkat jaringan lain yang berkaitan dengan perhitungan SLA.

Data historis yang telah dikumpulkan selanjutnya diolah dan disajikan secara visual melalui dashboard Grafana. Praktisan merancang representasi metrik SLA dalam bentuk grafik batang, gauge, dan heatmap, sehingga mempermudah tim operasional dan manajemen untuk mengawasi kinerja jaringan secara keseluruhan. Melalui dashboard ini, status sambungan semua perangkat dapat terlihat dalam satu tampilan terintegrasi, termasuk deteksi cepat terhadap toko atau cabang yang mengalami masalah.



Gambar 3.30 Dashboard Grafana

Sebagai bagian dari pengoptimalan sistem, praktikan juga mengimplementasikan fitur notifikasi otomatis melalui Telegram, yang akan mengirimkan pesan peringatan setiap kali ada anomali atau status perangkat berubah menjadi tidak aktif. Notifikasi ini dikirim berdasarkan pemicu yang telah ditentukan di Zabbix, untuk memastikan respons yang cepat dari tim teknis.

f. *Management*

Tahap Management ini fase terakhir dari metode Network Development Life Cycle (NDLC) yang konsentrasi pada aktivitas pengelolaan, perawatan, dan penilaian SLA berkelanjutan terhadap sistem yang sudah diterapkan. Pada langkah ini, praktikan memastikan bahwa sistem pemantauan SLA jaringan yang didasarkan pada Zabbix dan Grafana dapat beroperasi dengan baik, stabil, serta mampu beradaptasi dengan kebutuhan operasional jangka panjang.

Setelah proses instalasi, pengaturan, dan integrasi sistem pemantauan selesai, praktikan menyusun dokumentasi teknis

sebagai panduan penggunaan sistem, baik untuk tim operasional maupun untuk manajemen. Dokumentasi ini mencakup proses konfigurasi host, template, ambang notifikasi, pembuatan dashboard Grafana, dan prosedur penambahan perangkat baru ke sistem monitoring.

Di samping itu, praktikan juga melakukan penyesuaian pada trigger dan notifikasi menurut pengamatan hasil pemantauan, untuk mencegah false alarm serta menjamin bahwa alert yang dikirimkan benar-benar mencerminkan keadaan jaringan yang bermasalah. Evaluasi dilakukan secara rutin dengan memeriksa data historis dari Zabbix dan dashboard Grafana, terutama grafik downtime dan nilai SLA untuk setiap lokasi (kantor pusat, cabang, dan toko).

Sebagai bagian dari pengelolaan yang berkelanjutan, sistem ini juga dibuat agar dapat diperluas untuk mendukung lebih banyak perangkat jika diperlukan, serta memungkinkan pengembangan ke fitur-fitur canggih seperti integrasi dengan gerbang email, sistem pelaporan otomatis, atau penambahan notifikasi berbasis Telegram atau Email.

Tahap ini juga melibatkan proses pelatihan internal dan transfer pengetahuan kepada tim IT internal perusahaan, sehingga sistem yang telah dioptimalkan tidak hanya bergantung pada pengelola awal, tetapi dapat dikelola dan dikembangkan secara bersama di masa mendatang.

3.3 Kendala Yang Dihadapi

Selama melaksanakan kerja profesi dan penerapan sistem pemantauan SLA jaringan dengan Zabbix dan Grafana, praktikan menemui berbagai masalah teknis dan non-teknis yang membutuhkan penanganan khusus agar sistem dapat berfungsi dengan baik dan mencapai tujuan yang ditetapkan.

Salah satu kendala utama yang dihadapi adalah pada pengolahan data SLA dari *database* Zabbix, khususnya dalam mengekstrak data

historis *uptime/downtime* dari tabel *history_uint*. Data yang tersimpan menggunakan format UNIX timestamp dan struktur ID item yang kompleks, sehingga memerlukan pemahaman mendalam terhadap struktur database Zabbix.

Kendala selanjutnya adalah terkait integrasi Zabbix dengan Grafana, di mana plugin Zabbix tidak langsung menyajikan metrik SLA harian atau bulanan.

Selama pelaksanaan sistem pemantauan SLA jaringan melalui Zabbix dan Grafana, praktikan mengalami berbagai tantangan yang timbul dari aspek teknis maupun pengaturan sistem.

Hambatan pertama muncul ketika mengambil data historis *uptime* dan *downtime* dari *database* Zabbix. Informasi yang disimpan dalam format UNIX *timestamp* dan ID item mengakibatkan proses pengolahan menjadi tidak langsung, sehingga menyulitkan perhitungan SLA secara otomatis.

Masalah selanjutnya timbul ketika menggabungkan Zabbix dengan Grafana. *Plugin* Zabbix secara default tidak menawarkan visualisasi SLA, sehingga perlu pendekatan manual untuk menyusun *query* dan menyesuaikan tampilan metrik yang diinginkan.

Di samping itu, praktikan juga mengalami kendala dalam pengaturan notifikasi Telegram, khususnya dalam memperoleh token bot dan chat ID yang sah, serta menyesuaikan parameter *webhook* supaya dapat berfungsi dengan baik.

Dari segi infrastruktur, kekurangan sumber daya (RAM dan CPU) di *container* Proxmox sempat membuat layanan Zabbix mengalami ketidakstabilan saat proses pemantauan berlangsung bersamaan dengan Grafana.

3.4 Cara Mengatasi Kendala

Untuk mengatasi masalah-masalah itu, praktikan melakukan pendekatan teknis dan penelitian dokumentasi sistem.

Dalam menghadapi kendala pengolahan data historis, praktikan menerapkan *query* SQL tingkat lanjut dengan memanfaatkan fungsi *FROM_UNIXTIME()* untuk mengonversi timestamp menjadi format waktu

yang dapat dibaca, serta menyaring item berdasarkan ID yang sesuai dengan metrik uptime.

Integrasi Zabbix ke Grafana dilakukan dengan menginstal *plugin* resmi Zabbix untuk Grafana dan mengonfigurasi sumber data melalui API Zabbix. Praktikan juga membuat *dashboard* kustom dengan menambahkan panel dan *query* spesifik supaya data SLA bisa divisualisasikan dengan tepat.

Untuk notifikasi Telegram, praktikan mengikuti panduan dari komunitas dan melakukan pengujian *webhook* secara bertahap, sampai berhasil mengirim pesan peringatan secara langsung menggunakan bot Telegram dan ID chat yang sah.

Untuk mengatasi kendala sumber daya, praktikan mengubah konfigurasi kontainer melalui Proxmox, dengan menambah alokasi RAM dan CPU, serta menonaktifkan beberapa fitur pemantauan yang tidak begitu penting untuk mengurangi beban pada sistem.

Dengan menggunakan metode percobaan dan kesalahan serta pemahaman tentang struktur sistem, praktikan berhasil mengatasi tantangan yang ada dan memastikan sistem pemantauan berfungsi dengan baik, stabil, dan dapat diandalkan dalam mengawasi kinerja jaringan perusahaan.

Melalui pelaksanaan kerja profesi ini, praktikan mendapatkan banyak pembelajaran, baik dari segi teknis, metodologis, maupun pengalaman kerja di lapangan yang berhubungan langsung dengan dunia industri.

Secara teknis, praktikan memperoleh pengetahuan mendalam tentang penerapan sistem pemantauan jaringan dengan menggunakan alat open-source seperti Zabbix dan Grafana. Praktikan mempelajari metode untuk melakukan instalasi, konfigurasi, serta integrasi berbagai sistem, termasuk cara memproses data sejarah untuk kebutuhan analisis dan visualisasi SLA jaringan.

Praktikan juga menyadari akan pentingnya struktur *database* Zabbix, cara item dan trigger berfungsi, serta bagaimana mengatur notifikasi otomatis dengan menggunakan *media type* seperti Telegram *Webhook*. Kemampuan dalam menyusun *query* SQL dan mengelola

dashboard di Grafana menjadi keterampilan tambahan yang sangat relevan dengan posisi praktikan di bidang infrastruktur jaringan.

Dari segi metodologi, praktikan memperoleh pengalaman langsung dalam menerapkan tahap-tahap *Network Development Life Cycle* (NDLC), mulai dari analisis kebutuhan hingga fase pengelolaan sistem. Pendekatan ini mengajarkan cara merancang, mengevaluasi, dan mengembangkan sistem jaringan secara sistematis sesuai dengan kebutuhan aktual di perusahaan.

Selain aspek teknis dan metodologis, praktikan juga mendapatkan pengalaman kerja di lingkungan profesional, mencakup kemampuan komunikasi teknis, koordinasi tim, penyusunan dokumentasi, serta pemecahan masalah secara mandiri. Menghadapi hambatan dan mencari solusi secara langsung di lokasi menjadi pengalaman berharga yang meningkatkan kesiapan praktikan dalam menghadapi tantangan di dunia profesional setelah lulus.

Secara keseluruhan, pengalaman profesi ini telah memberikan wawasan dan keterampilan praktis yang tidak hanya berkaitan dengan disiplin Sistem Informasi, tetapi juga menjadi modal penting untuk pengembangan karier di sektor teknologi informasi, terutama dalam pengelolaan dan pemantauan infrastruktur jaringan perusahaan.

3.5 Pembelajaran Yang Didapat dari Kerja Profesi

Pekerjaan-pekerjaan yang telah dijelaskan sebelumnya memberikan praktikan berbagai pengalaman baru yang tidak hanya bersifat teknis, tetapi juga memperdalam pemahaman praktikan tentang penerapan ilmu yang telah dipelajari pada perkuliahan dalam konteks dunia kerja yang sesungguhnya. Dengan terlibat langsung dalam perencanaan, pelaksanaan, hingga penilaian sistem pemantauan jaringan berbasis Zabbix dan Grafana, praktikan mendapatkan pengalaman berharga dalam menghadapi masalah nyata, melakukan analisis sistem, serta mengelola infrastruktur jaringan yang digunakan secara operasional oleh perusahaan.

Adapun beberapa bentuk pembelajaran konkret yang didapatkan praktikan selama menjalankan kerja profesi antara lain adalah:

1. Memahami penggunaan sistem informasi untuk mendukung kegiatan operasional perusahaan, terutama dalam hal infrastruktur jaringan;
2. Menerapkan metode *Network Development Life Cycle* (NDLC) dalam dunia kerja secara langsung, mulai dari tahapan analisis kebutuhan jaringan hingga proses manajemen sistem monitoring yang telah berjalan;
3. Keterampilan teknis dalam instalasi dan pengaturan Zabbix Server, mencakup penyesuaian item, trigger, template, serta administrasi database monitoring yang menyimpan data historis uptime dan keadaan perangkat jaringan;
4. Penggabungan sistem pemantauan dengan Grafana, mencakup pengaturan sumber data Zabbix, pemanfaatan plugin, dan pembuatan dashboard visual seperti grafik gauge serta panel status koneksi host secara langsung;
5. Kemampuan untuk menyelesaikan masalah teknis, seperti kesalahan dalam *query*, pengelolaan sumber daya kontainer, dan pemecahan isu integrasi di antara sistem *open-source*;
6. Keterampilan komunikasi verbal dan tulisan dengan berbagai pihak di lingkungan kerja;
7. Pengalaman kolaborasi tim dan komunikasi teknis dalam konteks profesional, yang mengasah keterampilan koordinasi, penulisan dokumentasi teknis, serta pengambilan keputusan yang didasarkan pada analisis data jaringan;
8. Kemampuan dalam pengelolaan waktu;
9. Kemampuan dalam menyelesaikan tugas di bawah tekanan dengan permintaan yang tinggi dalam waktu yang terbatas.

Selain itu, praktikan juga mendapatkan pemahaman yang lebih mendalam mengenai hubungan antara posisi sebagai Network Engineer & Messaging di departemen IT Operation dengan konsep-konsep yang telah dipelajari selama perkuliahan di program studi Sistem Informasi. Saat melaksanakan kerja profesi, Praktikan menyadari bahwa konsep-konsep

yang selama ini dipelajari secara teori semuanya saling berkaitan dan diterapkan secara langsung dalam pekerjaan. Berikut adalah pembelajaran yang didapatkan:

Tabel 3.12 Hubungan Antara Item Pekerjaan dengan Mata Kuliah

No	Pekerjaan	Mata Kuliah Terkait	Teori Mata Kuliah Terkait	Penjelasan
1	Membuat alur kerja sistem monitoring.	INS202 Analisis Proses Bisnis	Busines Process Modeling	Memahami alur kerja yang direncanakan dan mengimplementasikan sesuai rencana tersebut.
2	Implementasi topologi, konsep jaringan dan manajemen jaringan komputer.	INS209 Keamanan Informasi dan Administrasi Jaringan	Konsep Jaringan, TCP/IP, Network Management dan Konsep Jaringan	Memahami konsep jaringan, protokol jaringan dan manajemen pemantauan jaringan.
3	Optimalisasi Sistem Pemantauan Jaringan.	INS105 Pengantar Sistem Informasi	Inovasi dan Perubahan Organisasi	Meningkatkan atau mengganti proses bisnis untuk meningkatkan efisiensi dan produktivitas.
4	Instalasi <i>database server</i> untuk penyimpanan data <i>Network Monitoring System</i> .	INS102 Sistem Basis Data	Arsitektur Sistem Basis Data	Memahami penggunaan basis data dan konsep dasar basis data.

Berdasarkan tabel di atas, disimpulkan bahwa semua aktivitas yang dilaksanakan selama kerja profesi sangat berkaitan dengan materi kuliah di jurusan Sistem Informasi. Setiap tugas yang dilakukan mencerminkan penerapan langsung dari teori yang telah dipelajari di ruang kelas, baik dari segi analisis proses bisnis, administrasi jaringan, sistem basis data, maupun pengembangan dan peningkatan sistem informasi.