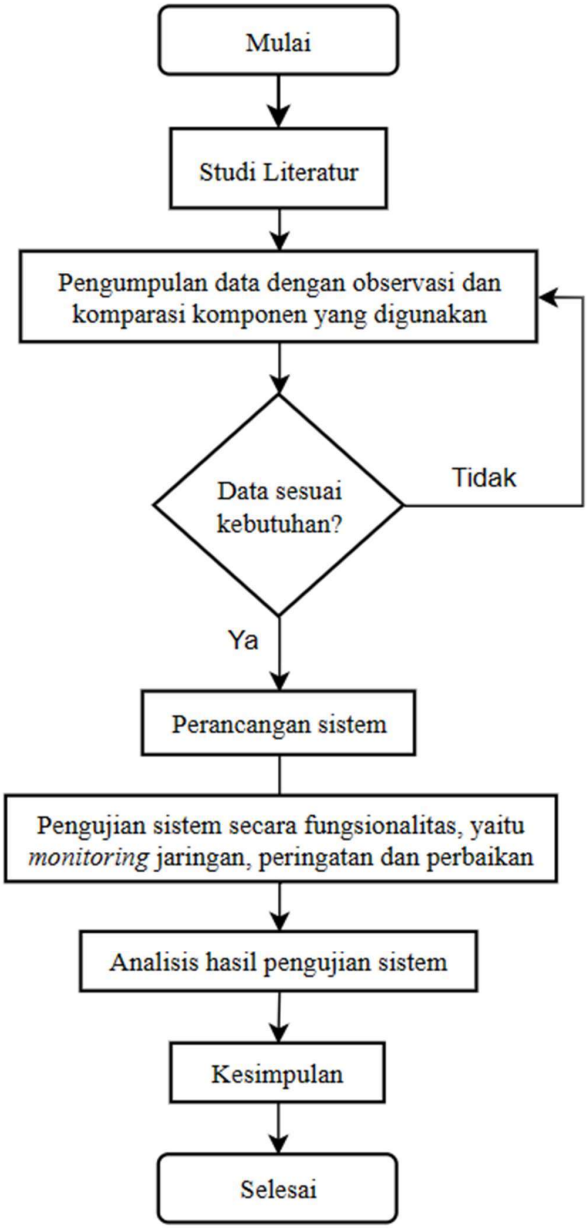
BAB III
METODE PENELITIAN

3.1 Desain Penelitian

Dalam penelitian ini terdapat sebuah *flowchart* alur penelitian yang ditunjukan pada Gambar 3.1.



Gambar 3. 1 *Flowchart* Alur Penelitian

Pada langkah awal yang dilakukan adalah studi literatur. Literatur yang berhubungan dengan sistem *monitoring* jaringan. Literatur tersebut diperoleh melalui jurnal nasional maupun internasional, dan *e-book.* Serta mencari informasi tentang kebutuhan yang diperlukan untuk menunjang penelitian ini.

Setelah mempelajari studi literatur, langkah selanjutnya adalah analisis kebutuhan. Kebutuhan yang dianalisa adalah kebutuhan *software*. *Software* mencakup pilihan aplikasi Web analitik dan visualisasi, aplikasi simulasi, *virtual machine*, dan lainnya. Selain itu juga melalukan observasi lapangan yang berlokasi di DSTI UPI untuk mendapatkan data seperti topologi jaringan dan *monitoring* yang digunakan.

Langkah selanjutnya adalah merancang sistem secara keseluruhan dari segi *software* yang meliputi instalasi dan konfigurasi yang dibutuhkan seperti *Prometheus*, SNMP *Exporter, Grafana*, dan *Telegram*. Perancangan sistem yang telah rampung, selanjutnya sistem dilakukan pengujian dengan membuat skenario gangguan yaitu putusnya kabel *ethernet* pada *router*. Serta dilakukan perbaikan sistem berupa *backup* internet.

Pengujian sistem bertujuan untuk mengetahui fungsionalitas sistem *monitoring* jaringan dan peringatan yang dimunculkan melalui notifikasi. Setelah uji sistem selesai dan data hasil uji telah didokumentasikan, data hasil uji tersebut dianalisis. Hal ini dilakukan sebagai bahan evaluasi terhadap sistem yang dikembangkan.

3.2 Lokasi Penelitian

Penelitian ini dilaksanakan di Gedung A Fakultas Pendidikan Teknik dan Kejuruan Universitas Pendidikan Indonesia yang berolakasi di Jl. Dr. Setiabudi No.229, Isola, Kec. Sukasari, Kota Bandung, Jawa Barat (40154). Gedung A FPTK UPI merupakan gedung dimana seluruh proses administrasi seperti input data depertemen jurusan yang ada di FPTK UPI terpusat di Gedung A. Gedung tersebut memiliki 5 lantai yang digunakan oleh 15 prodi pendidikan kejuruan yang menempati lantai 1 sampai lantai 5. Gedung A ditampilkan pada Gambar 3.2.

Gambar 3. 2 Gedung A FPTK UPI

3.3 Metode Pengumpulan Data

Dalam penelitian ini, penulis mengumpulkan informasi dan data akurat yang dapat mendukung penelitian. Pengumpulan data pada penelitian ini dilakukan dengan beberapa tahap yaitu sebagai berikut:

1. Studi literatur

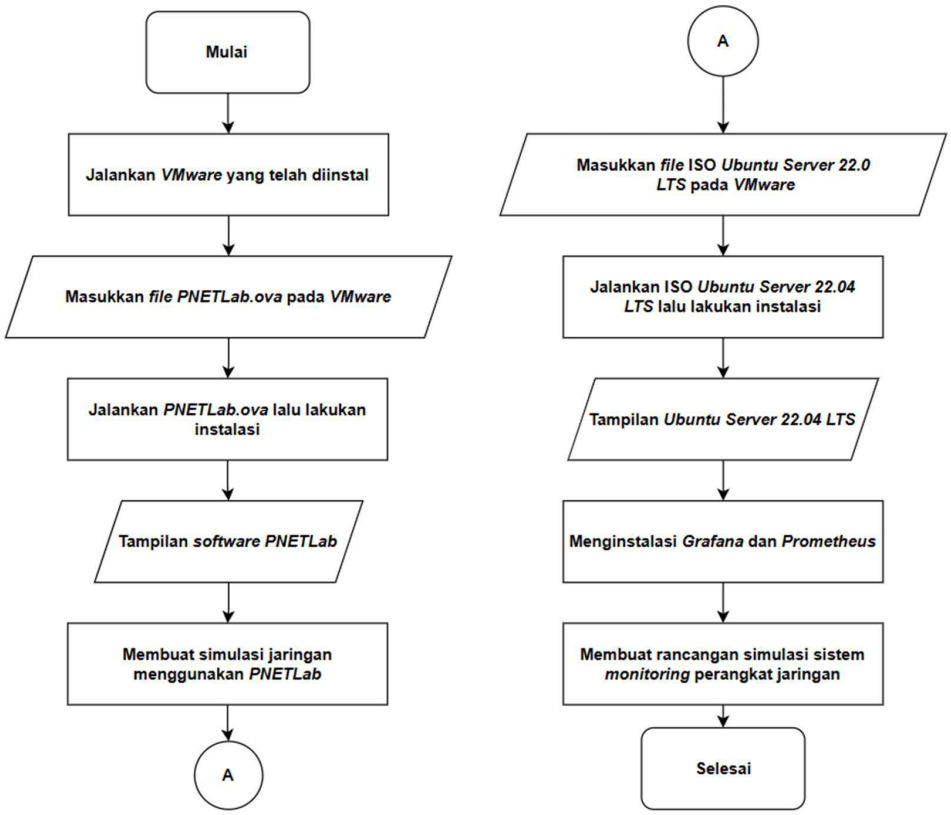
Tahap ini melakukan studi literatur terlebih dahaulu sebelum melakukan tahap observasi. Studi literatur dilakukan dengan cara mengkaji jurnal, artikel, *website*, dan *e-book* baik itu nasional atau internasional yang berkaitan dengan perancangan sistem *monitoring*. Studi literatur ini bertujuan untuk mendapatkan informasi sebagai referensi yang dapat menunjang penelitian.

2. Observasi

Tahap ini melakukan observasi lapangan yang berlokasi di Gedung A Fakultas Pendidikan Teknologi dan Kejuruan UPI sebagai objek dari penelitian ini untuk melakukan sistem *monitoring*. Serta melakukan observasi ke Direktorat Sistem dan Teknologi Informasi (DSTI) UPI untuk mendapatkan data yang menunjang penelitian yaitu topologi jaringan gedung A FPTK UPI yang meliputi konektivitas dan *hardware information*. Pada tahap ini juga melakukan diskusi dengan beberapa pihak yaitu *staff* DSTI UPI dan dosen pembimbing.

3.4 *Software* Penunjang Penelitian

Pada penelitian ini terdapat *software* yang digunakan untuk menunjang penelitian. Berikut ini merupakan *flowchart* instalasi *software* penunjang penelitian yang dapat diamati pada Gambar 3.3. Tahap awal adalah membuka *VMware* sebagai *virtual machine*. Selanjutnya adalah memasukan *file PNETLab.ova* pada *VMware* dan melakukan instalasi. Setelah instalasi selesai, langkah selanjutnya adalah melakukan simulasi jaringan pada *sofrware PNETLab*. Tahap berikutnya adalah memasukan *file image Ubuntu Server 22.04 LTS* dan melakukan instalasi. Setelah *Ubuntu Server* diinstalasi, langkah selanjutnya adalah menginstalasi *Grafana* dan *Prometheus* dan membuat rancangan sistem *monitoring* perangkat jaringan.



Gambar 3. 3 *Flowchart* Instalasi *Software* Penunjang Penelitian

3.5 Instalasi dan Konfigurasi

Pada penelitian ini dibutuhkan instalasi dan konfigurasi agar *software* yang digunakan dapat berfungsi dengan baik.

3.5.1 Instalasi dan Konfigurasi *Prometheus*

1. Langkah pertama adalah melakukan instalasi *Prometheus* pada *server* menggunakan perintah berikut.

```
# wget
https://github.com/prometheus/prometheus/releases/download/v2.45.0/pro
prometh-2.45.0.linux-amd64.tar.gz
# tar -xvf prometheus-2.45.0.linux-amd64.tar.gz
# sudo mv prometheus promtool /usr/local/bin/
# sudo groupadd --system prometheus
# sudo useradd --system -s /sbin/nologin -g prometheus prometheus
# sudo mkdir /etc/prometheus
# sudo mv consoles/ console_libraries/ prometheus.yml /etc/prometheus/
```

Gambar 3. 4 Perintah Instalasi *Prometheus*

2. Setelah proses instalasi selesai, langkah selanjutnya adalah mengkonfigurasi *Prometheus.yml* dengan perintah berikut.

```
# cd /etc/prometheus/
# sudo nano prometheus.yml

global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: "prometheus"
    static_configs:
       - targets: ["localhost:9090"]
```

Gambar 3. 5 Perintah Konfigurasi *Prometheus.yml*

3. Untuk menghindari *permission issues*, diperlukan pengaturan kepemilikan yang benar untuk *Prometheus* dengan menggunakan perintah berikut.

```
# sudo mkdir /var/lib/prometheus
# sudo chown -R prometheus:prometheus /var/lib/prometheus/
```

Gambar 3. 6 Perintah Pengaturan Kepemilikan Untuk *Prometheus*

4. Kemudian Langkah selanjutnya agar *Prometheus* dapat dijalankan di belakang layar, maka dilakukan konfigurasi pada *service Prometheus* dengan perintah berikut.

```
# sudo nano /etc/system/system/prometheus.service
(di dalam prometheus.service)
 [Unit]
Description=Prometheus
Documentation=https://prometheus.io/docs/introduction/overview/
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
--config.file /etc/prometheus/prometheus.yml \
--storage.tsdb.path /var/lib/prometheus/ \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

Gambar 3. 7 Perintah Konfigurasi *Service Prometheus*

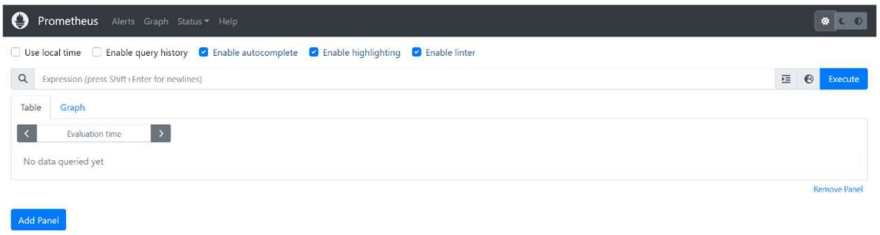
5. Setelah melakukan konfigurasi *service Prometheus*, diwajibkan untuk memuat ulang *service* tersebut dengan perintah `# sudo systemctl daemon-reload`

6. Langkah selanjutnya adalah menjalankan *service Prometheus* dengan perintah berikut.

```
# sudo systemctl enable --now prometheus.service
# sudo systemctl status prometheus.service
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled;
vendor preset: enabled)
     Active: active (running) since Wed 2023-07-12 12:13:48 UTC; 1h
28min ago
       Docs: https://prometheus.io/docs/introduction/overview/
   Main PID: 878 (prometheus)
      Tasks: 8 (limit: 4516)
     Memory: 89.0M
        CPU: 8.536s
     CGroup: /system.slice/prometheus.service
             └─878 /usr/local/bin/prometheus --config.file
/etc/prometheus/prometheus.yml --storage.tsdb.path
/var/lib/prometheus/
```

Gambar 3. 8 Perintah Untuk Menjalankan *Service Prometheus*

7. Setelah *service Prometheus* dijalankan, langkah selanjutnya adalah membuka *web browser* untuk mengakses halaman *Prometheus* menggunakan format URL *"http://192.168.131.132:9090"* dan klik *enter*.



Gambar 3. 9 Tampilan Halaman *Prometheus*

3.5.2 Instalasi dan Konfigurasi SNMP *Exporter*

1. Langkah pertama adalah melakukan instalasi SNMP *Exporter* pada *server* menggunakan perintah berikut.

```
# wget
https://github.com/prometheus/snmp_exporter/releases/download/v0.22.0/
snmp_exporter-0.22.0.linux-amd64.tar.gz
# tar xvf snmp_exporter-0.22.0.linux-amd64.tar.gz
# sudo mv snmp_exporter /usr/local/bin/
# sudo mkdir /etc/snmp_exporter
# sudo mv snmp.yml /etc/snmp_exporter/
```

Gambar 3. 10 Perintah Instalasi SNMP *Exporter*

2. Kemudian Langkah selanjutnya agar SNMP *Exporter* dapat dijalankan di belakang layar, maka dilakukan konfigurasi pada *service* SNMP *Exporter* dengan perintah berikut.

```
#sudo nano /etc/systemd/system/snmp-exporter.service
(di dalam snmp-exporter.service)
[Unit]
Description=SNMP Exporter
After=network-online.target

# This assumes you are running snmp_exporter under the user
"prometheus"

[Service]
User=prometheus
Restart=on-failure
ExecStart=/usr/local/bin/snmp_exporter --
config.file=/etc/snmp_exporter/snmp.yml

[Install]
WantedBy=multi-user.target
```

Gambar 3. 11 Perintah Konfigurasi *Service* SNMP *Exporter*

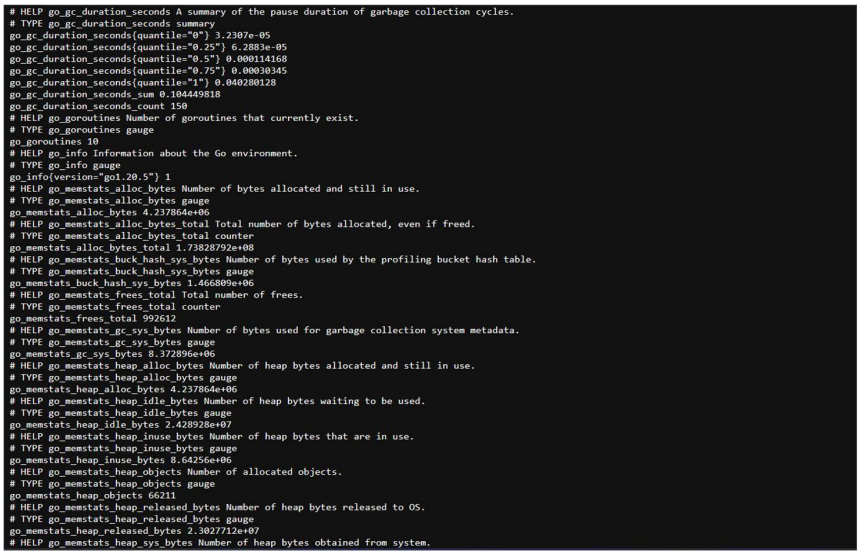
3. Setelah melakukan konfigurasi *service* SNMP *Exporter*, diwajibkan untuk memuat ulang *service* tersebut dengan perintah `# sudo systemctl daemon-reload`

4. Langkah selanjutnya adalah menjalankan service SNMP *Exporter* dengan perintah berikut.

```
# sudo systemctl enable --now snmp-exporter.service
# sudo systemctl status snmp-exporter.service
● snmp-exporter.service - SNMP Exporter
     Loaded: loaded (/etc/systemd/system/snmp-exporter.service;
enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-07-12 12:13:48 UTC; 4h
52min ago
   Main PID: 882 (snmp_exporter)
      Tasks: 8 (limit: 4516)
     Memory: 24.6M
        CPU: 7.651s
     CGroup: /system.slice/snmp-exporter.service
             └─882 /usr/local/bin/snmp_exporter --
config.file=/etc/snmp_exporter/snmp.yml
```

Gambar 3. 12 Perintah Untuk Menjalankan *Service* SNMP *Exporter*

5. Setelah *service* SNMP *Exporter* dijalankan, langkah selanjutnya adalah membuka *web browser* pada URL *"http://192.168.131.132:9116"* dan klik *enter* untuk mengakses *metrik* atau data yang dihasilkan oleh SNMP *Exporter*.



Gambar 3. 13 Tampilan Metrik Yang Dihasilkan SNMP *Exporter*

3.5.3 Instalasi dan Konfigurasi *Grafana*

1. Langkah pertama adalah melakukan instalasi *Grafana* pada *server* menggunakan perintah berikut.

```
# sudo apt-get install -y apt-transport-https
# sudo apt-get install -y software-properties-common wget
# sudo wget -q -O /usr/share/keyrings/grafana.key
https://apt.grafana.com/gpg.key
# echo "deb [signed-by=/usr/share/keyrings/grafana.key]
https://apt.grafana.com stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list
# sudo apt-get update
# sudo apt-get install grafana
```

Gambar 3. 14 Perintah Instalasi *Grafana*

2. Langkah selanjutnya adalah menjalankan *service Grafana* dengan perintah berikut.

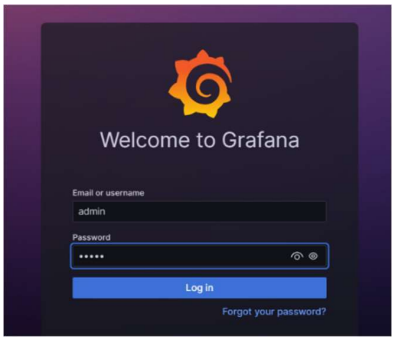
```
# sudo systemctl enable --now grafana-server.service
# sudo systemctl status grafana-server.service
● grafana-server.service - Grafana instance
     Loaded: loaded (/lib/systemd/system/grafana-server.service;
enabled; vendor preset: enable>
     Active: active (running) since Wed 2023-07-12 12:13:48 UTC; 5h
17min ago
       Docs: http://docs.grafana.org
   Main PID: 872 (grafana)
      Tasks: 23 (limit: 4516)
     Memory: 721.8M
        CPU: 4min 20.946s
     CGroup: /system.slice/grafana-server.service
             ├─ 872 /usr/share/grafana/bin/grafana server --
config=/etc/grafana/grafana.ini --p>
             └─1136 /var/lib/grafana/plugins/grafana-image-
renderer/plugin_start_linux_amd64
```

Gambar 3. 15 Perintah Untuk Menjalankan *Service Grafana*

3. Setelah *service Grafana* dijalankan, langkah selanjutnya adalah membuka *web browser* pada URL *"http://192.168.131.132:3000"* dan klik *enter* untuk mengakses halaman *log in Grafana*.

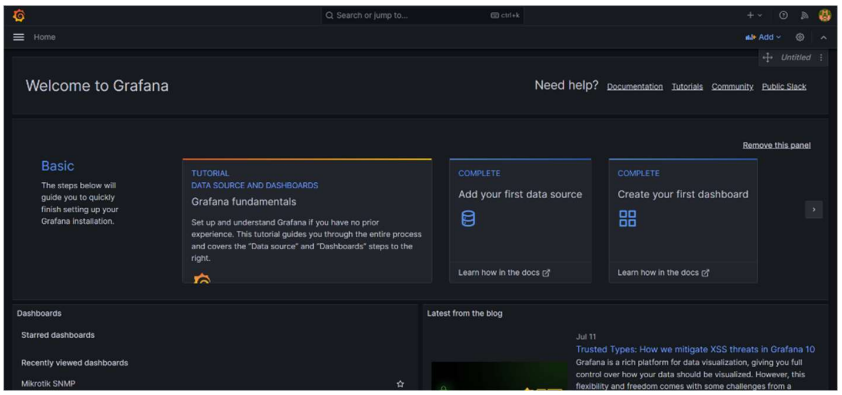
Gambar 3. 16 Halaman *Log In Grafana*

4. Pada halaman pertama, masukan ”*admin”* tanpa tanda petik sebagai *input* untuk *Email or username* dan *password.*



Gambar 3. 17 Tampilan *Log In Grafana*

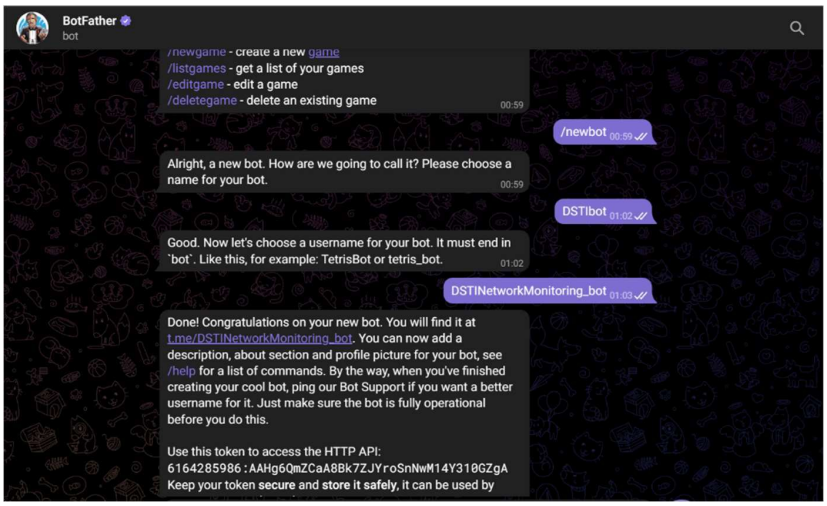
5. Selanjutnya setelah proses *log in* berhasil, *Dashboard Grafana* akan terbuka yang dimana secara keseluruhan proses instalasi *Grafana* telah berhasil dan siap digunakan untuk kegiatan *monitoring.*



Gambar 3. 18 Tampilan *Dashboard Grafana*

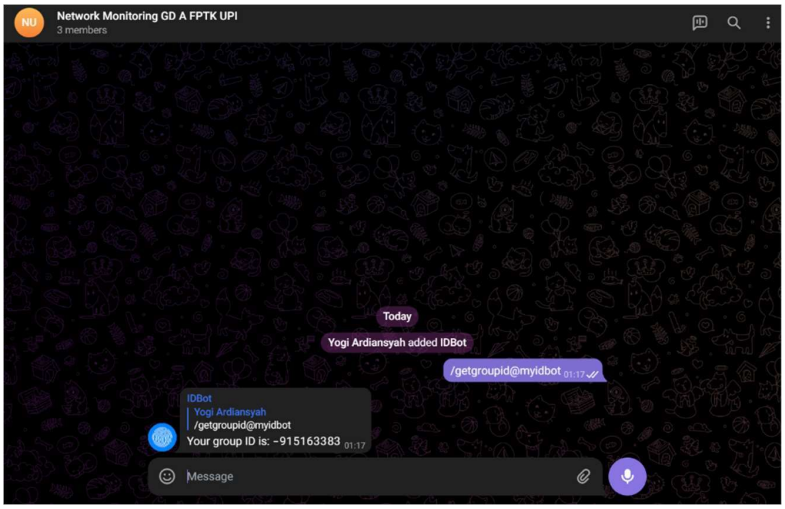
3.5.4 Konfigurasi *Telegram*

1. Langkah pertama yang ditunjukkan pada Gambar 3.19 adalah mendaftarkan *bot Telegram* baru dengan cara mengirikan pesan *text "/newbot"* ke *@BotFather* lalu mengikuti petunjuknya. *Token* yang disediakan oleh *@BotFather* pada langkah terakhir akan digunakan oleh *Grafana* untuk dikonfigurasi agar dapat terintegrasi dengan *Telegram*.



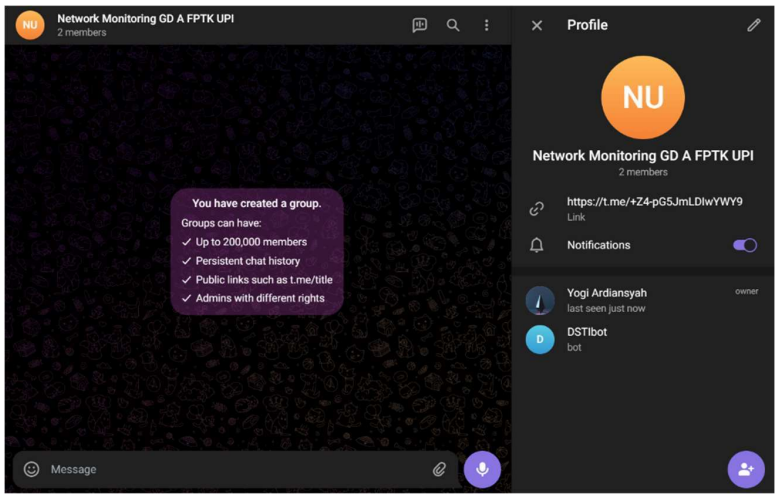
Gambar 3. 19 Pembuatan *Bot Telegram*

2. Selanjutnya buat *Group* antara *administrator* dengan *bot*.



Gambar 3. 20 Tampilan *Group Monitoring* Jaringan

3. Langkah selanjutnya adalah menambahkan *bot @myid* ke group yang telah dibuat, lalu meminta *group* ID kepada *@myid* dengan perintah berikut.



Gambar 3. 21 *Request Group* ID

3.6 Pengujian Sistem

Pengujian sistem bertujuan untuk menguji fungsionalitas dari rancangan sistem *monitoring* perangkat jaringan Gedung A FPTK UPI. Pengujian yang akan dilakukan terdiri dari pembuatan skenario permasalahan dan skenario perbaikan. Skenario permasalahan pada perangkat jaringan yaitu memutus salah satu konektivitas pada *router*, sedangkan skenario perbaikan pada perangkat jaringan yaitu membuat *backup* internet pada *router* yang mati. Skenario permasalahan dan skenario perbaikan pada perangkat jaringan yaitu sebagai berikut:

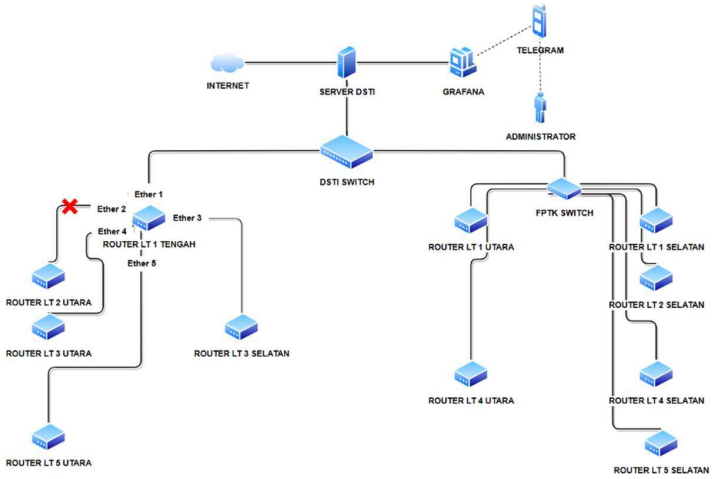
1. Skenario Permasalahan

Skenario pengujian permasalahan konektivitas perangkat jaringan Gedung A FPTK UPI terdapat topologi yang tertera pada Gambar 3.22. Topologi tersebut terdiri dari internet yang terkoneksi dengan *server* DSTI yang telah terinstalasi *software* untuk monitoring yaitu *Grafana*. *Grafana* tersebut terintegrasi dengan *Telegram* sebagai media notifikasi yang dimana dapat diakses oleh *administrator*.

Pembagian jaringan internet diatur oleh *administrator* melalui DSTI *switch* yang dimana jaringan tersebut dibagi menjadi dua jalur. Untuk jalur pertama dibagikan menuju *router* lantai 1 tengah dan untuk jalur kedua menuju FPTK *switch*. Perangkat jaringan yang terhubung dengan *router* lantai 1 tengah adalah *router* lantai 2 utara, *router* lantai 3 utara, *router* lantai 3 selatan, dan *router* lantai 5 utara. Untuk perangkat yang terhubung dengan FPTK *switch* adalah *router* lantai

1 utara, *router* lantai selatan, *router* lantai 2 selatan, *router* lantai 4 utara, *router* lantai 4 selatan, dan *router* lantai 5 selatan.

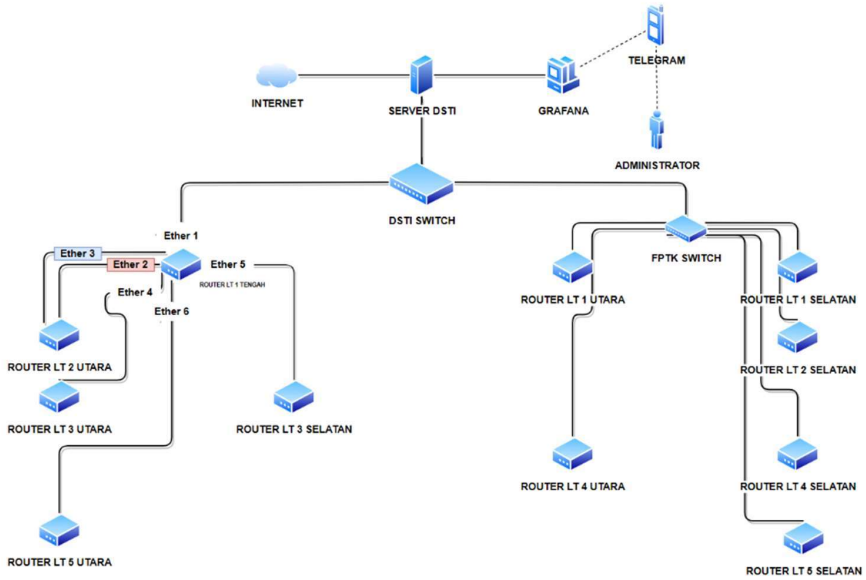
Skenario permasalahan yang dilakukan adalah memutus *ether* 2 yang terhubung dari *router* lantai 1 tengah ke *router* lantai 2 utara untuk menguji fungsionalitas *Grafana* sebagai sistem *monitoring* dan *alerting*.



Gambar 3. 22 Topologi Skenario Pengujian Permasalahan Konektivitas Perangkat Jaringan Gedung A FPTK UPI

2. Skenario Perbaikan

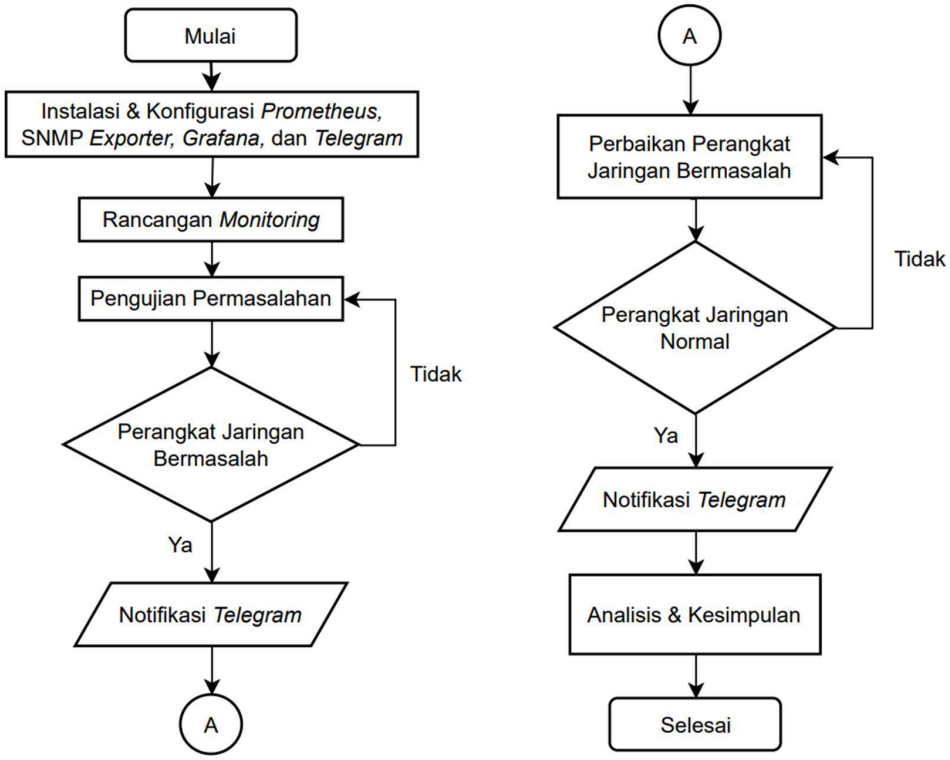
Skenario topologi perbaikan konektivitas perangkat jaringan Gedung A FPTK UPI yang tertera pada Gambar 3.23 menunjukkan perbaikan perangkat jaringan dilakukan dengan cara menambahkan satu konektivitas yaitu *ether* 3 pada router lantai 1 tengah ke *router* lantai 2 utara sebagai backup jaringan internet yang dimana *router* lantai 2 utara memiliki dua konektivitas. Ketika *ether* 2 terputus, otomatis *ether* 2 akan mengambil alih untuk menghubungkan konektivitas jaringan internet dari *router* lantai 1 tengah ke lantai 2 utara.



Gambar 3. 23 Topologi Skenario Pengujian Perbaikan Konektivitas Perangkat Jaringan Gedung A FPTK UPI

3.7 Metode Analisis Data

Penelitian ini dilakukan untuk mengetahui langkah-langkah perancangan sistem *monitoring* jaringan internet di gedung A FPTK UPI. Hal tersebut dapat diketahui pada gambar berikut:



Gambar 3. 24 *Flowchart* Analisis Data

Pada langkah awal yang dilakukan adalah instalasi dan konfigurasi *Prometheus,* SNMP *Exporter, Grafana,* dan *Telegram*. *Prometheus* yang di*install* dan dikonfigurasi pada *server* DSTI, kemudian diintegrasikan dengan SNMP *Exporter* yang telah diinstal dan dikonfigurasi pada *server* DSTI. Setelah *job* SNMP *Exporter* dapat berjalan dalam *Prometheus,* data atau informasi yang sudah dikumpulkan oleh *Prometheus* akan divisualisasikan oleh *Grafana* dan diintegrasikan dengan *Telegram* sebagai media notifikasi untuk sistem peringatan.

Visualisasi data atau informasi terkait perangkat jaringan *Grafana* perlu input *data source Prometheus* agar dapat terintegrasi dengan data atau informasi yang dimiliki oleh *Prometheus*. Kemudian dibuatkan sebuah *dashboard* untuk dapat membaca semua data atau informasi.

Apabila *dashboard* telah ditampilkan sebagai hasil rancangan sistem monitoring, selanjutnya dilakukan pengujian permasalahan pada sistem *monitoring* jaringan gedung A FPTK UPI yang bertujuan untuk mendeteksi kendala atau masalah sedini mungkin sehingga dapat memudahkan tugas *administrator* jaringan dalam mengatasi kendala yang terjadi. Pengujian tersebut dilakukan dengan memutus konektivitas *ether* 2 *router* lantai 1 tengah yang terhubung dengan *router* lantai 2 utara. Skenario permasalahan tersebut disimulasikan menggunakan *software PNETLab* dan divisualisasikan oleh *Grafana* serta memberikan informasi berupa notifikasi ke *Telegram* melalui *smartphone administrator*.

Apabila SNMP *Exporter* mendeteksi sebuah masalah seperti konektivitas perangkat jaringan terputus, maka SNMP *Exporter* secara otomatis akan mengirimkan sebuah metrik kepada *Prometheus,* kemudian *Grafana* akan menvisualisasikan metrik tersebut pada *dashboard* yang telah dibuat. Informasi permasalahan yang muncul pada *dashboard Grafana* akan mengirim pesan berupa notifikasi pada *Telegram* yang telah dikonfigurasi.

Konektivitas perangkat jaringan yang bermasalah kemudian secara otomatis akan kembali berfungsi karena *ether* 3 sebagai backup akan menggantikan *ether* 2 yang terputus. Maka secara otomatis *Grafana* akan mengvisualisasikan keadaan konektivitas jaringan antara *router* lantai 1 tengah dan *router* lantai 2 utara serta mengirimkan notifikasi ke *Telegram* untuk memberikan informasi bahwa *router* lantai 2 utara dapat berfungsi kembali.

Langkah terakhir adalah yaitu melakukan analisis skenario pengujian permasalahan dan perbaikan permasalahan pada konektivitas perangkat jaringan gedung A FPTK UPI. Serta membuat kesimpulan berdasarkan hasil analisis yang telah dilakukan.