# BAB II

# LANDASAN TEORI

## 2.1 Tinjauan Penelitian

Tinjauan Penelitian ini akan menjelaskan teori yang dijadikan sebagai acuan dari penelitian ini meliputi *dashboard, monitoring* sistem dan jaringan, Jaringan Komputer, *Cacti, RRD (Round Robin Database), Simple Network Management Protocol* (SNMP)*, Grafana, Pushgateway, Prometheus.*

### 2.1.1 Jaringan Komputer

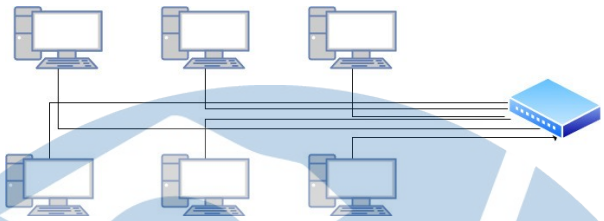
Menurut Sofana [13] "jaringan komputer adalah suatu himpunan interkoneksi sejumlah komputer, dalam bahasa popular dapat di jelaskan bahwa jaringan komputer adalah kumpulan beberapa komputer, dan perangkat lain seperti *router, switch* dan sebagainya". Alat yang bisa terhubung dengan suatu lainnya untuk memudahkan memahami jaringan komputer para ahli sudah membagi beberapa klasifikasi, diantaranya:

- Berdasarkan area atau skala.
- Berdasarkan media pengantar.
- Berdasarkan fungsi

Menurut Sofana [13] "jaringan komputer terbagi beberapa jenis jaringan, yang memisahkan berdasarkan area atau skala dan terbagi menjadi tiga bagian", yaitu :

- *Local Area Network* (LAN)

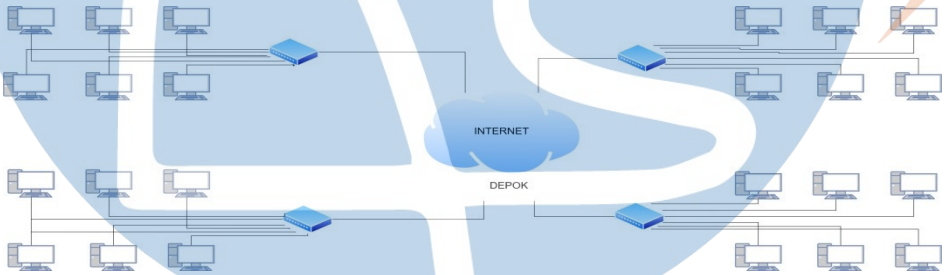
LAN adalah jaringan komputer yang terhubung dalam suatu area tertentu, misalnya bangunan hotel, mall, institusi pendidikan, dan lainnya. Jaringan LAN dapat berjarak 2 mil sesuai dengan kebutuhan penggunaannya. Jaringan LAN dapat berbentuk hanya dengan menggunakan *PC* dan sebuah printer dalam sebuah rumah.



*Gambar 2.1 Tampilan LAN*

- *Metropolitan Area Network* (MAN)

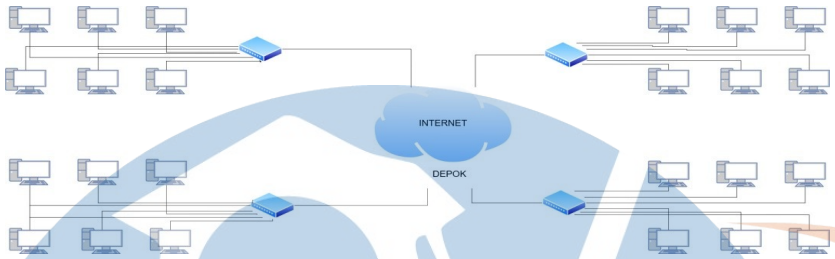
*Metropolitan Area Network* menggunakan metode yang sama dengan LAN namun daerah cangkupannya lebih luas. Daerah cangkupan MAN bisa satu RW, beberapa kantor yang berada dalam satu komplek yang sama, satu/beberapa desa. Dapat dikatakan MAN pengembangan dari LAN.




*Gambar 2.2 TampilanMAN*

- *Wide Area Network* (WAN)

*Wide Area Network* cangkupannya lebih luas dari MAN. Cangkupan MAN meliputi satu kawasan, satu Negara, satu pulau bahkan satu dunia, metode yang digunakan WAN sama seperti LAN dan MAN. Umumnya WAN dihubungkan dengan jaringan telepon digital. Namun media transmisi lain pun dapat digunakan.




*Gambar 2.3 Tampilan WAN*

### 2.1.2 Monitoring Sistem dan Jaringan

Sistem *monitoring* jaringan merupakan sistem yang berfungsi untuk memantau aktivitas pada perangkat jaringan. *Monitoring* digunakan untuk mengetahui perangkat jaringan mana yang mati dan hidup. Dengan melakukan *monitoring* diharapkan jika terjadi permasalahan pada jaringan dapat diperbaiki dengan cepat dan mudah oleh *administrator*.

Sistem *monitoring* digunakan untuk memantau, mengawasi, dan mengontrol jalan atau tidaknya suatu perangkat jaringan. Pentingnya *monitoring* adalah terpantau secara rutin perangkat yang bermasalah yang berpotensi mengganggu jaringan internet. Masalah jaringan yang sering muncul adalah kerusakan perangkat jaringan dan listrik tidak stabil, dimana kesalahannya tidak diketahui oleh pemantau jaringan secara manual dan pemeriksaan jaringan yang terlalu lama melakukannya.

Untuk menjaga jaringan dapat digunakan secara maksimal, diperlukan adanya *monitoring* perangkat jaringan seperti pada objek penelitian *monitoring* perangkat jaringan berbasis *Cacti* menggunakan *Grafana* untuk kemudian *monitoring* tersebut dan juga dapat ditampilkan dalam bentuk *website* yang bertujuan mempermudah *administrator* melakukan tugas *monitoring* jaringan.

#### 2.1.2.1 Metode Monitoring Sistem Dan Jaringan

Berikut beberapa metode yang dapat dilakukan dalam melakukan *monitoring* sistem dan jaringan:

1. **Deteksi Intrusi**

   Metode pertama adalah mendeteksi *intrusi*. Maksudnya adalah deteksi *intrusi monitoring* jaringan area lokal untuk akses yang tidak sah oleh peretas. Biasanya para profesional IT menggunakan metode ini secara otomatis untuk beberapa kegiatan. Misalnya saja untuk mendeteksi virus dan *malware*, kerentanan jaringan. Meskipun dapat digunakan secara otomatis, metode yang satu ini juga dapat diimplementasikan secara manual.

2. **Sniffing Packet**

   *Sniffing packet* merupakan program yang memberikan setiap paket informasi yang melewati jaringan. Tujuannya adalah mendeteksi perangkat lunak *monitoring* jaringan yang tidak sah yang dapat diinstal oleh peretas untuk memata-matai aktivitas bisnis dan proses informasi. Jadi, metode ini dapat melindungi aktivitas bisnis Anda dari orang yang tidak bertanggung jawab.

3 **Vulnerability Scanning**

   *Vulnerability Scanning* atau pemindaian kerentanan yang akan memindai jaringan untuk mengetahui kerentanan dan kelemahan yang membuka potensi eksploitasi. Metode ini dapat bervariasi atau berbeda dari metode pertama yaitu deteksi intrusi karena dapat mendeteksi kelemahan sebelum serangan terjadi pada jaringan.

4 **Monitoring Firewall**

   Metode *Monitoring Firewall* ini dapat digunakan untuk melacak aktifitas dari *firewall*, untuk memastikan proses penyaringan untuk koneksi masuk dan keluar dapat berfungsi dengan aman dan baik.

5 **Penetration Testing**

Metode *Peneration Testing* ini merupakan metode yang digunakan oleh peretas untuk menembus jaringan yang kemudian digunakan juga oleh IT professional. Tujuannya tentu berbeda, para IT professional menggunakan metode terakhir ini untuk membawa keamanan jaringan ke tingkat lain dengan menemukan kerentanan yang mungkin diketahui oleh peretas namun belum terdeteksi melalui metode *monitoring* lainnya.

### 2.1.2.2 Jenis Protokol

Pada jaringan di seluruh dunia, ada beberapa jenis protokol yang digunakan untuk berhubungan. Beberapa jenis protokol yaitu:

1. **TCP/IP**

*Transmission Control Protocol* (TCP) dan *Internet Protocol* (IP) merupakan standar dari komunikasi data yang dipakai oleh komunitas internet. Standar ini mengatur dalam proses tukar-menukar data atau informasi dari satu komputer ke komputer lain di dalam jaringan internet.

2. **User Datagram Protokol (UDP)**

*User Datagram Protocol* (UDP) adalah *transport* TCP/IP yang dapat mendukung komunikasi yang *unreliable*, tanpa adanya koneksi antar *host* di dalam suatu jaringan.

3. **Domain Name System (DNS)**

*Domain Name Server* (DNS) adalah *distribute database* yang dipakai dalam pencarian nama komputer di dalam jaringan menggunakan TCP/IP. DNS dapat bekerja pada jaringan dengan skala kecil sampai dengan global. Terkadang DNS juga digunakan pada aplikasi yang terhubung langsung dengan internet.

4. **HTTPS**

HTTPS berasal dar *Hypertext Transfer Protocol* (HTTP) yang merupakan protokol untuk mengatur komunikasi antara client dan server. Sedangkan HTTPS merupakan versi aman dari HTTP biasa. HTTPS merupakan kombinasi dari komunikasi HTTP biasa melalui

*Socket Secure Layer* (SSL) atau *Transport Layer Security* (TLS), jadi bukan merupakan protokol yang berbeda. Sehingga, ada dua jenis lapisan enkripsi. Kombinasi dilakukan untuk menjaga keamanan beberapa serangan pihak ketiga. Biasanya serangan yang dilakukan adalah menyadap informasi dari komunikasi yang terjadi.

**5. SSH (Secure Shell)**

SSH adalah sebuah protokol jaringan yang memungkinkan terjadinya pertukaran data antara dua komputer dengan aman. Mulai dari mengirim *file*, mengendalakan pada jarak yang jauh dan lain sebagainya. Dibanding dengan *telnet*, FT, protokol ini mempunyai tingkat keamanan yang unggul.

**6. Telnet (Telecommunication Network)**

*Telnet* dikembangkan pada 1969, *telnet* memiliki standarisasi sebagai IETF STD 8 yang merupakan standar internet pertama kali. Protokol ini berjalan pada koneksi Internet atau LAN. Namun sayangnya *telnet* mempunyai keterbatasan keamanan yang masih beresiko.

**7. OSI Layer**

*OSI Layer* merupakan standar komunikasi yang diterapkan untuk jaringan komputer. Standar ini digunakan untuk menentukan aturan sehingga seluruh alat komunikasi bisa saling terkoneksi melalui jaringan internet. *OSI Layer* dikembangkan untuk komputer agar dapat berkomunikasi pada jaringan yang berbeda secara efisien. Protokol ini digambarkan sebagai informasi dari suatu aplikasi komputer yang berpindah melalui jaringan internet ke komputer yang lainnya. *OSI Layer* secara konseptual terbagi ke dalam tujuh lapisan dimana masing-masing lapisan memiliki tugas yang spesifik. Berikut ketujuh lapisan *OSI Layer*:

- **Application Layer**

  *Layer OSI* ini paling berdekatan dengan *end user*. *Layer* ini bertanggung-jawab atas pertukaran informasi antara program komputer, seperti program *e-mail*, dan *service* lain yang jalan di jaringan, seperti *server* printer atau aplikasi komputer lainnya.

- **Presentation Layer**

  *Layer OSI* ini bertanggung jawab dalam pengkodean dan konversi data dari *application layer*. *Presentation later* bertanggung jawab untuk memastikan semua data yang berasal dari application *layer* dapat dibaca pada sistem lainnya.
- **Session Layer**

  *Layer OSI* ini mempunyai tugas untuk menentukan bagaimana dua terminal menjaga, memelihara dan mengatur koneksi. Selain itu *layer* ini berfungsi untuk membentuk,me-*manage*,dan memutuskan *session* komunikasi antara entitas presentation *layer*.
- **Transport Layer**

  *Layer OSI* ini bertanggung jawab untuk membagi data menjadi segmen, menjaga koneksi logika antar terminal, dan menyediakan penanganan *error*.
- **Network Layer**

  *Layer OSI* ini bertanggung jawab untuk menentukan alamat jaringan, menentukan rute yang harus diambil selama perjalanan, dan menhaga antrian trafik di jaringan.
- **DataLink Layer**

  *Layer OSI* ini mempunyai tugas untuk menyediakan *link* untuk data dan memaketkannya menjadi *frame* yang berhubungan dengan *hardware* kemudian di distribusikan melalui media.
- **Physical Layer**

  *Layer OSI* yang terakhir ini bertugas untuk mengirimkan dan menerima data mentah pada media fisik. Tujuan utama penggunaan *OSI Layer* adalah untuk membantu desainer jaringan memahami fungsi dari tiap-tiap *layer* yang berhubungan dengan aliran komunikasi data. Termasuk jenis-jenis protokol jaringan dan metode transmisi.

#### 2.1.2.3 Perangkat Monitoring Jaringan

*Monitoring* jaringan adalah upaya sistematis jaringan komputer untuk mendeteksi komponen jaringan yang mengalami masalah seperti

lambat maupun kegagalan. Contohnya seperti *server* yang kelebihan beban atau macet/beku, *switch* maupun *router* yang gagal, dan masalah perangkat lainnya. Upaya *monitoring* jaringan membantu untuk menghemat waktu, biaya dan melihat bagimana performanya. Ada beberapa perangkat yang dapat di *monitoring* oleh *monitoring* jaringan.

1. **Router**

   Router adalah perangkat yang menganalisis konten paket data yang dikirimkan dalam suatu jaringan ke jaringan yang lain. Monitoring jaringan bisa memonitoring router yang kita miliki.

2. **CCTV**

   Perangkat pertama adalah CCTV. *Closed-circuit television* adalah sistem TV yang mana sinyal tidak didistribusikan secara public tetapi di monitor, terutama untuk tujuan pengawasan dan keamanan. Tidak hanya perangkat jaringan, *Monitoring* jaringan juga dapat memonitor perangkat CCTV.

3. **Server**

   Perangkat kedua yang bisa di monitor oleh *Monitoring* jaringan adalah *server*. Menurut Rouse dalam artikelnya, *server* adalah program komputer atau perangkat yang menyediakan layanan ke program komputer lain dan penggunanya, juga dikenal sebagai *client*. Di pusat data, komputer fisik yang menjalankan program *server* juga sering disebut sebagai *server*.

4. **Akses Point**

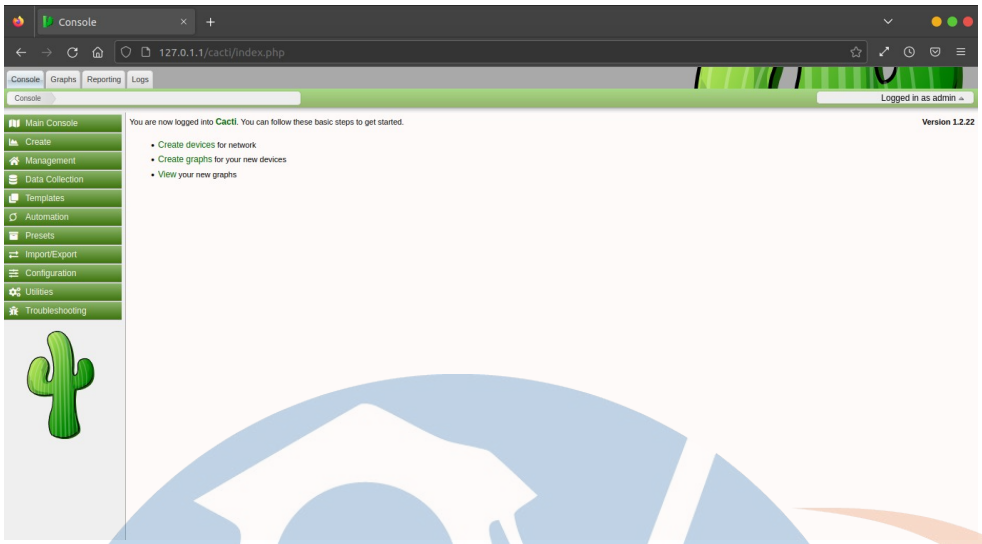
   Monitoring jaringan juga dapat memonitoring Akses Point. Menurut Rouse dalam WLAN (*Wireless Local Area Network*), akses point adalah stasiun yang mentransmisikan dan menerima data. Akses point menghubungkan pengguna ke pengguna lain dalam jaringan dan dapat berfungsi sebagai titik interkoneksi antara WLAN dan jaringan kabel tetap.

### 2.1.3 *Cacti*

[9] *Cacti* adalah frontend lengkap untuk RRDTool, ia menyimpan semua informasi yang diperlukan untuk membuat grafik dan mengisinya dengan data dalam database MySQL. Frontend sepenuhnya digerakkan oleh PHP. Seiring dengan kemampuan untuk memelihara grafik, sumber data, dan arsip *round robin* dalam *database*, kaktus menangani pengumpulan data. Ada juga dukungan SNMP bagi mereka yang terbiasa membuat grafik lalu lintas dengan MRTG. Untuk menangani pengumpulan data, pengguna dapat memberi makan kaktus jalur ke skrip / perintah eksternal bersama dengan data apa pun yang perlu "diisi" oleh pengguna, kemudian kaktus akan mengumpulkan data ini dalam tugas cron dan mengisi database MySQL / arsip *round robin*. Setelah satu atau lebih sumber data ditentukan, grafik RRDTool dapat dibuat menggunakan data. *Cacti* memungkinkan pengguna membuat enkin semua grafik RRDTool yang dapat dibayangkan menggunakan semua jenis grafik RRDTool enkinsus fungsi konsolidasi. Area pemilihan warna dan fungsi padding teks otomatis juga membantu pembuatan grafik untuk mempermudah proses.

*Cacti* adalah salah satu *software* yang digunakan untuk keperluan monitoring yang banyak digunakan saat ini. *Cacti* menyimpan semua data / informasi yang diperlukan untuk membuat grafik dan mengumpulkannya dengan database MySQL. Untuk menjalankan *Cacti* diperlukan service pendukung seperti Apache2, MySQL, PHP5, dan SNMP.

Terakhir, *Cacti* dapat melakukan penskalaan ke sejumlah besar sumber data dan grafik melalui penggunaan template. Ini memungkinkan pembuatan satu grafik atau template sumber data yang mendefinisikan grafik atau sumber data yang terkait dengannya.



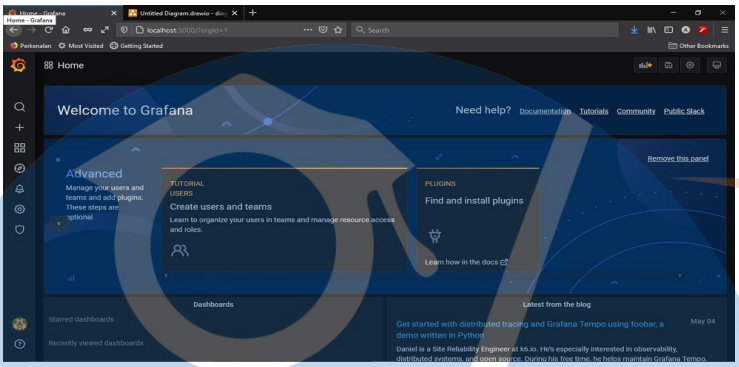

*Gambar 2.4 Tampilan Cacti*

### 2.1.4 *Grafana*

*Grafana* adalah analitik sumber terbuka *multi-platform* dan aplikasi web visualisasi interaktif . *Grafana* menyediakan bagan, grafik, dan peringatan untuk web saat terhubung ke sumber data yang didukung. Versi *Grafana Enterprise* berlisensi dengan kemampuan tambahan juga tersedia sebagai penginstalan yang dihosting sendiri atau akun di layanan *cloud Grafana labs*. *Grafana* dapat diperluas melalui sistem *plug-in* . Pengguna akhir dapat membuat *dashboard* pemantauan yang kompleks. *Grafana* pertama kali dirilis pada 2014 oleh Torkel Ödegaard sebagai cabang dari proyek di Orbitz . *Grafana* menargetkan *database* deret waktu seperti *InfluxDB , OpenTSDB*, dan *Prometheus,* tetapi berevolusi untuk mendukung *database* relasional seperti *MySQL , PostgreSQL* dan *Microsoft SQL Server* . Tahun 2019, *Grafana Labs* dijamin $24 juta di Seri A pendanaan.

Di dalam dokumentasi G*rafana labs* menjelaskan bahwa *grafana* adalah perangkat lunak analisis dan visualisasi metrik *open source*. Dimana *Grafana* ini digunakan untuk *query*, memvisualisasikan, mengingatkan, dan menjelajahi metrik dimanapun mereka disimpan. *Grafana* ini memberi alat untuk mengubah data *time-series database*(TSDB) menjadi grafik dan visualisasi yang indah.

Menurut Karolus TFhias Widagdo, Teguh Indra Bayu, dan Yeremia Alfa Susetyo [12] *Grafana* merupakan salah satu penyedia layanan monitoring dalam bentuk *dashboard* yang dikembangkan oleh *Grafana labs*. *Grafana* menyediakan berbagai macam *Application Programming Interface*(API) yang dapat mendukung sinkronisasi dengan berbagai macam data *source* sebagai sumber informasi yang akan ditampilkan di *dashboard*.




*Gambar 2.5 Tampilan Grafana*

### 2.1.5 **Dashboard**

Penelitian ini mempunyai focus utama yaitu dalam mengimplementasikan *dashboard* untuk memonitoring sistem dan jaringan. *Dashboard* adalah jenis antarmuka pengguna grafis yang sering kali memberikan tampilan sekilas tentang indikator kinerja utama yang relevan dengan tujuan atau proses bisnis tertentu. Dalam penggunaan lain, *"dashboard"* adalah nama lain untuk "laporan kemajuan" atau "laporan" dan dianggap sebagai bentuk visualisasi data [8]. Dalam banyak kasus, ini dapat dikonfigurasi, dimana hal itu memungkinkan pengguna untuk memilih data mana yang ingin pengguna lihat dan apakah pengguna ingin menyertakan bagan atau grafik untuk memvisualisasikan angka-angkanya.

#### 2.1.5.1 Tipe Dashboard

Menurut Rasmussen, ada beberapa macam tipe dari *dashboard* antara lain adalah:

1. **Startegical *Da*shboard,** yang berfungsi sebagai pendukung garis organisasi dengan tujuan yang strategis.
2. ***Tactical Dashboard,*** yang berfungsi sebagai pendukung pengukuran progress dalam kunci atau inisiatif proyek.
3. ***Operational Dashboard,*** yang berfungsi sebagai pendukung monitoring dari aktifitas proses bisnis bersifat spesifik.

Berdasarkan ketiga tipe *dashboard* yang ada, maka salah satu metode diatas penulis dalam penelitian tugas akhir ini akan menggunakan *operational dashboard* sebagai pendukung *monitoring* dari aktifitas prosesnya yang dimana nantinya penulis akan menggunakan tampilan utama dalam *monitoring* sistem dan jaringan menggunakan *Grafana*.

#### 2.1.5.2 Manfaat Dashboard

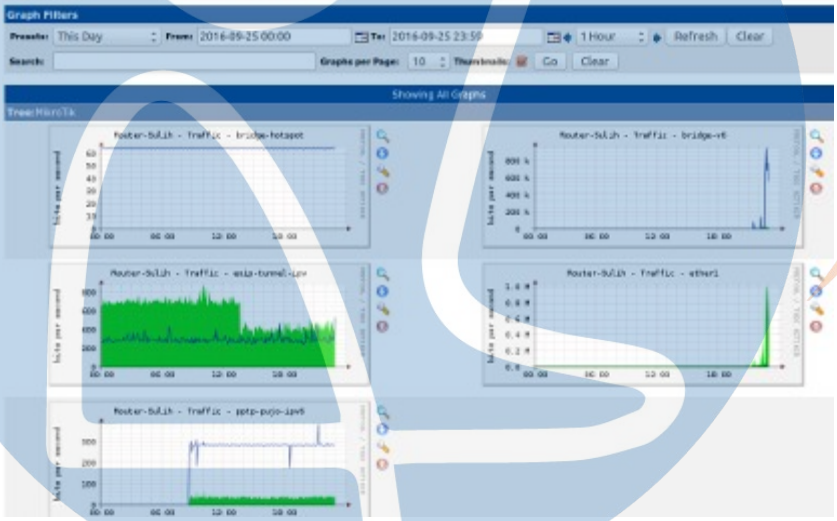
*Dashboard* juga mempunyai beberapa manfaat bagi penggunanya supaya mampu memberikan kemudahan untuk menggunakan *dashboard* tersebut, berikut beberapa manfaat yang dihasilkan dari *dashboard*:

1 Sebagai Sistem Penunjang Keputusan bagi Middle, Top Level Management sebuah instansi, organisasi ataupun perusahaan.

2 Sebagai Media Informasi yang dapat menyajikan informasi secara efisien dengan adanya grafik atapun kalimat ringkasan dari informasi yang disajikan**.**

3 Sebagai Media *Monitoring* yang dapat memantau progress atau perkembangan dari suatu kegiatan.


*Gambar 2.6 Tampilan Dashboard Grafana*




*Gambar 2.7 Tampilan Dashboard Grafik Cacti*

### 2.1.6 Round Robin Database (RRDTool)

RRD (*Round Robin Database*) adalah sebuah *database* yang menyimpan informasi dengan cara yang sangat *compact* yang tidak berkembang seiring waktu. RRDtool merujuk pada sesederetan tool yang memungkinkan anda menciptakan dan mengubah *database* RRD, dan juga menghasilkan grafik untuk merepresentasikandata. Dipakai untuk mencatat data terhadap waktu (seperti jaringan *bandwith*, temperatur ruang mesin, atau *load server* rata-rata) dan bisa menampilkan data itu sebagai rata-rata dalam selang waktu tertentu.

### 2.1.7 Prometheus

*Prometheus* adalah *opensource* sistem *monitoring* dan *alerting* sistem sumber terbuka yang awalnya dibuat di *SoundCloud.* Sejak diluncurkan pada tahun 2012, banyak perusahaan dan organisasi telah mengadopsi *Prometheus,* dan proyek ini memiliki komunitas *developer* dan *user* yang sangat aktif. *Prometheus* mengumpulkan dan menyimpan *metric* sebagai data deret waktu, yaitu informasi *metric* disimpan dengan *timestamp* saat direkam, disamping pasangan nilai kunci opsional yang disebut label. Fitur utama *Prometheus* adalah:

1. model data multidimensi dengan data deret waktu yang diidentifikasi dengan nama *metric* dan pasangan *key/value.*
2. *PromQL,* bahasa *query* yang fleksibel untuk memanfaatkan dimensi ini.
3. Tidak bergantung pada penyimpinan terdistribusi *single server node* bersifat *otonom.*
4. Pengumpulan deret waktu terjadi melalui model tarikan melalui *HTTP.*
5. *Push* deret waktu didukung melalui perantara *Pushgateway.*
6. Target ditemukan melalui penemuan layanan atau konfigurasi statis.
7. Beberapa node grafik dan dukungan *dashboard.*

*Metric* adalah pengukuran *numerik*. *Time series* berarti bahwa perubahan dicatat dari waktu ke waktu, apa yang ingin diukur oleh pengguna berbeda dari satu aplikasi ke aplikasi yang lainnya. Untuk *server* web mungkin waktu permintaan, untuk *database* mungkin jumlah *query* aktif. *Prometheus* terdiri dari banyak komponen, banyak diantaranya bersifat opsional:

1. *Server Prometheus* utama yang membuka dan menyimpan data deret waktu.
2. C*lient libraries* menginstrumensasi kode aplikasi.
3. *Pushgateway* untuk *supporting short-lived job.*

4. Eksportir tujuan khusus untuk layanan seperti *HAProxy, StatsD, Graphite.*
5. *Alert manager* untuk menangani peringatan.

Sebagian besar komponen *Prometheus* ditulis dalam *Go,* membuatnya mudah dibuat dan diterapkan sebagai binari statis[10].

### 2.1.8 Pushgateway

*Pushgateway* adalah fitur *Prometheus* yang memungkinkan pekerjaan singkat dan batch untuk mengekspos *metric* mereka ke aplikasi. Seringkali, jenis pekerjaan ini tidak memiliki siklus hidup yang cukup lama untuk program perangkat lunak mengikisnya dan menarik sejumlah besar metrik yang diperlukan untuk pemantauan jaringan yang efektif. Proses ini berbeda dari aplikasi metrik dari *Prometheus*. Dengan *Pushgateway,* rangkaian metrik pengguna terus menerus diekspos ke *Prometheus* kecuali seseorang menghapusnya secara manual menggunakan *API pushgateway*. Saat *Prometheus* menggunakan metrik model tarikan, ia hanya memiliki akses ke instans yang masih ada. Aplikasi ini memiliki pustaka klien untuk *Go, Java, Scala, Python,* dan *Ruby,* sehingga berguna untuk khalayak luas.

*Pushgateway* bukanlah jawaban untuk semua pekerjaan siklus pendek pengguna. *Prometheus* memperingatkan pengguna bahwa *Pushgateway* hanya berfungsi dengan baik dalam keadaan tertentu, terutama untuk pekerjaan batch tingkat layanan seperti menghapus data pengguna yang telah meminta untuk dihapus (hak untuk dilupakan).



### 2.1.9 Simple Network Management Protocol (SNMP)

*Simple Network Management Protocol* (SNMP) merupakan sebuah protokol yang dirancang untuk memonitor dan mengatur suatu jaringan yang berbasis TCP/IP baik dari jarak jauh (*remote*) atau dalam satu pusat kontrol saja. Protokol ini dapat memberikan informasi tentang status dan keadaan dari suatu jaringan atau perangkat jaringan seperti *server, desktop, hub, router, switch*.

Protokol ini menggunakan *transport* UDP pada *port* 161 dan *port* 162 pada kondisi pengiriman pesan trap dari *agent* ke *manager*[11]. Pengolahan informasi pada SNMP dijalankan dengan mengumpulkan data dan melakukan penetapan terhadap *variabel-variabel* dalam elemen jaringan yang dikelola.

## 2.2 Penelitian Terkait

Penelitian ini tidaklah secara keseluruhan hal yang baru, melainkan sudah ada penelitian sebelumnya terkait dengan penelitian ini, berikut diantaranya:

*Tabel 2.1 Penelitian Terkait*

| No | Nama dan Tahun | Judul | Topik | Hasil |
|---|---|---|---|---|
| 1 | (Dwi Risza Budi Raharja, Periyadi, AnangSularsa, 2015) | Implementasi monitoring jaringan menggunakan *cacti* dan web *authentication* menggunakan *Kerberos* pada MAN 1 Bojonegoro | *Authentication* | *Monitoring* sistem yang dilakukan untuk memudahkan admin dalam mengontrol *server* berjalan dengan baik. Manajemen *Bandwith* menggunakan *HTB Tools* berjalan pada server dan sudah dapat dipakai dan sudah melalui pengujian pada proyek akhir ini. |

| | | | | |
|---|---|---|---|---|
| 2 | (Prida Apriani,T.M. Diansyah,Risko Liza, 2020) | Pemanfaatan fitur *cacti* berbasis *telegram messenger* untuk notifikasi gangguan jaringan di PT.TELKOM AKSES MEDAN | Fitur | *ENACCS* bekerja berdasarkan nilai enkinsu masalah jaringan dan durasi *down* yang direkam oleh *cacti* untuk kemudian dikelola dan dikirimkan melalui grup *Telegram* berdasarkan wilayah yang telah dimasukkan pada eld notes pada basis data *cacti*, tur penentuan grup *Telegram* ini dapat diatur secara dinamis. *ENACCS* merupakan wujud nyata agar teknisi PT. Telkom Akses Medan dapat menerima notifikasi gangguan lebih dini jika terjadi gangguan jaringan. |
| 3 | (Dede Rahman,Hidra Amnur,Indri Rahmayuni, 2020) | *Monitoring server* dengan *Prometheus* dan *grafana* serta notifikasi *telegram* | *Monitoring server* | *Monitoring server* berhasil dilakukan dengan meng gunakan *enkinsu* dan *grafana* terhadap *server*. Diharapkan untuk pengembangan selanjutnya sistem *monitoring server* dapat memberikan alert ke aplikasi *mobile* yang dibuat sendiri. |

Diantara ketiga tabel penelitian terkait diatas bahwa penelitian yang memiliki kedekatan pada penelitian saya ini adalah pada penelitian yang berjudul "*Monitoring* server dengan Prometheus dan *Grafana* serta notifikasi *telegram*", karena mempunyai kemiripan tentang bagaimana cara memonitoring dengan *Grafana* dan juga nanti bisa dikembangkan juga dengan menggunakan berbasis *Cacti* dalam penelitian saya.