



## PERANCANGAN DAN IMPLEMENTASI NETWORK MONITORING SYSTEM BERBASIS NAGIOS MENGGUNAKAN NOTIFIKASI PADA LAYANAN TELEGRAM

Christine Ayu<sup>1</sup>, Henry Saptono<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
[christineayu97@gmail.com](mailto:christineayu97@gmail.com), [henry@nurulfikri.co.id](mailto:henry@nurulfikri.co.id)

### Abstract

Computer networks have become an essential requirement for organizations or institutions in supporting their business processes. Of course, that needs to be accompanied by continuous management, maintenance, and monitoring. When something unexpected happens, network managers will find it easier to analyze improvements and problem-solving. For that, we need a system or software that can lighten and make it easier for network managers to carry out their duties in maintaining and monitoring the network. Monitoring in this system includes monitoring the status of the network condition, the computer's performance, the health of the server computer, including the status of a service running. With this system, it is hoped that it can provide an alert mechanism (warning) to managers when something can cause damage or failure to the network. In this study, researchers designed and implemented a Nagios-based network monitoring system with notification support using Telegram. The use of Telegram in this research is not only as alerts and notifications but also to optimize an improvement made by researchers from the traditional network monitoring system. The improvement is implementing a monitoring system that network administrators can carry out sending commands to telegrams using telegram bots. The telegram bot will access the Nagios-based network monitoring system, which is implementing in this research. So, the network managers do not have to monitor a web-based dashboard continuously but can also continue interacting on Telegram.

**Keywords:** Network Monitoring System, Notification, Alert, Chatbot, Telegram, Nagios

### Abstrak

Jaringan komputer sudah menjadi sebuah kebutuhan yang utama bagi organisasi atau institusi dalam mendukung proses bisnisnya. Hal tersebut tentunya perlu diiringi dengan pengelolaan, pemeliharaan dan pemantauan secara terus menerus, agar ketika terjadi sesuatu hal yang tidak diharapkan, pengelola jaringan akan lebih mudah dalam melakukan analisis perbaikan dan pemecahan masalah. Untuk itu diperlukan sebuah sistem atau *software* yang dapat meringankan dan mempermudah pengelola jaringan dalam melakukan tugasnya dalam memelihara dan memantau jaringan. Pemantauan dalam sistem ini, meliputi pemantauan status dari kondisi jaringan, performa dari komputer, kesehatan dari komputer server termasuk status berjalan dan tidaknya suatu layanan. Dengan adanya sistem tersebut, diharapkan dapat memberikan mekanisme *alert* (peringatan) kepada pengelola, ketika ada yang berpotensi akan menyebabkan kerusakan atau kegagalan pada jaringan. Dalam penelitian ini, peneliti melakukan perancangan dan penerapan terhadap *network monitoring system* berbasis Nagios dengan dukungan notifikasi menggunakan Telegram. Pemanfaatan Telegram dalam penelitian ini bukan hanya sebagai *alert* dan notifikasi, namun juga untuk optimalisasi sebuah *improvement* yang dilakukan peneliti dari sistem *network monitoring system* yang tradisional. Dimana *improvement* tersebut berupa penerapan sistem pemantauan yang dapat dilakukan oleh *network administrator* melalui pengiriman *command* ke telegram menggunakan telegram bot. Kemudian, telegram bot akan mengakses informasi yang ada di *network monitoring system* berbasis Nagios yang diterapkan dalam penelitian ini. Sehingga pengelola jaringan tidak harus selalu memantau *dashboard* berbasis web, tetapi juga dapat tetap berinteraksi di Telegram.

**Kata kunci:** Network Monitoring System, Notifikasi, Alert, Chatbot, Telegram, Nagios

## 1. PENDAHULUAN

Dewasa ini jaringan komputer bukanlah sesuatu yang baru di dunia teknologi penggunaan jaringan komputer umumnya digunakan di setiap instansi, perusahaan dan institusi pendidikan, jaringan komputer digunakan untuk melakukan pertukaran data dan untuk mengetahui informasi sehingga jaringan komputer sudah menjadi sebuah kebutuhan oleh sebab itu di perlukannya pemeliharaan dan pemantauan khusus untuk jaringan komputer agar tidak terjadi sesuatu hal yang tidak di inginkan [1].

Untuk mencegah terjadinya gangguan yang terlalu lama biasanya seorang *network administrator* selalu mengawasi dan memantau kinerja jaringan komputer. *Network administrator* tidak mungkin terus menerus mengamati dan mengawasi di depan komputer secara langsung suatu perangkat ataupun *service* yang berjalan pada sebuah jaringan selama 24 jam.

Oleh sebab itu diperlukan sebuah fasilitas pendukung yaitu *network monitoring system*. *Network monitoring system* adalah sebuah *tool* yang digunakan untuk melakukan monitoring ataupun pengawasan pada hal-hal tertentu didalam jaringan komputer, monitoring dilakukan untuk mengetahui suatu masalah sedini mungkin agar dapat langsung ditangani dengan maksimal sehingga permasalahan tersebut tidak akan meluas, serta untuk mengetahui performa dari suatu perangkat ataupun jaringan yang nantinya akan dianalisa oleh seorang administrator.

Dengan adanya monitoring sistem suatu masalah dalam jaringan dapat diselesaikan sedini mungkin sehingga permasalahan tersebut tidak akan meluas dan bertambah banyak dan dengan adanya monitoring sistem juga memudahkan *network administrator* untuk selalu dapat memantau kegiatan perangkat penting dari jaringan komputer seperti router, switch, komputer *client*, performa dan lain-lain.

*Nagios* adalah aplikasi pemantauan jaringan yang bekerja untuk memantau sistem dan jaringan, sistem *nagios* ini memantau jaringan yang berjalan pada server, seperti CPU, Memory, Disk dan bandwidht, *Nagios* sendiri dapat dikonfigurasi sesuai dengan kebutuhan. Salah satu fitur penting dalam *Network Monitoring System* adalah notifikasi *alert*, yaitu sistem pemberitahuan notifikasi kepada kontak *network administrator* jika sistem mendeteksi adanya permasalahan pada host maupun *service*. Pemberitahuan notifikasi dapat melalui berbagai media seperti email, sms, dan lain-lain.

Namun hal itu dirasa kurang efisien karena terkadang seorang administrator tidak mengetahui adanya notifikasi ke email mereka sehingga terkadang hal tersebut menghambat pekerjaan administrator oleh karena itu penulis merasa akan lebih efisien jika *alert* (notifikasi) menggunakan notifikasi via telegram, *nagios* akan mengirimkan *alert* (notifikasi) jika suatu host mengalami masalah atau melebihi batas

penggunaan, *nagios* akan mengirimkan notifikasi melalui telegram bot, dan pada penelitian ini penulis menambahkan fitur *chatbot* telegram sehingga memudahkan *network administrator* untuk untuk mengetahui status host, performa ataupun *service* dengan mengetikkan *command* yang telah ditetapkan sebelumnya.

Oleh karena itu diharapkan dengan adanya penelitian ini dapat membantu mempermudah pekerjaan administrator dalam memonitoring sebuah sistem sehingga efektivitas dan efisiensi dapat terwujud dalam penelitian ini penulis berfokus pada permasalahan yang akan dijawab berdasarkan kajian dari beberapa hal berikut, antara lain:

1. Bagaimana rancangan arsitektur sistem monitoring jaringan yang akan digunakan secara logik dan fisik?
2. Bagaimana rancangan program sistem notifikasi untuk *network monitoring system* berbasis *nagios* dengan menggunakan layanan telegram?
3. Apakah efektif system monitoring system dengan menggunakan dukungan telegram bot?
4. Apa teknik atau cara agar *chatbot* telegram dapat terintegrasi dengan *nagios*?

Tujuan dan manfaat dari penelitian ini antara lain:

1. Mengimplementasikan *network monitoring system* berbasis *nagios* dengan *alerting* berupa notifikasi telegram bot.
2. Merancang sebuah sistem monitoring untuk dapat terintegrasi dengan system API Telegram Bot.
3. Menerapkan program dan skrip dalam system monitoring berbasis *nagios*.
4. Memudahkan pengelolaan jaringan dalam melakukan pemantauan sumber daya dalam jaringan komputer.
5. Mempercepat penyampaian notifikasi (pesan-pesan) *alerting* kepada pengelola administrator jaringan melalui pesan instan telegram bot.
6. Menghasilkan suatu karya tulis yang bisa menjadi masukan bagi orang lain di dalam menerapkan *network monitoring system*.

Dalam penelitian ini terdapat beberapa batasan masalah agar penelitian ini lebih terarah, antara lain:

1. *Network monitoring system* dirancang dan di implementasikan pada jaringan simulasi (virtual).
2. Hal-hal yang di monitoring hanya yang berkaitan dengan status, performa system dan beberapa jaringan umum saja.
3. Di dalam pembahasan tugas akhir ini penulis tidak memasukkan bahasan sosial media lain selain telegram.
4. Penulis tidak mengembangkan dashboard monitoring system *nagios*.
5. Pada *alerting* (peringat) penulis hanya berfokus pada notifikasi via telegram.

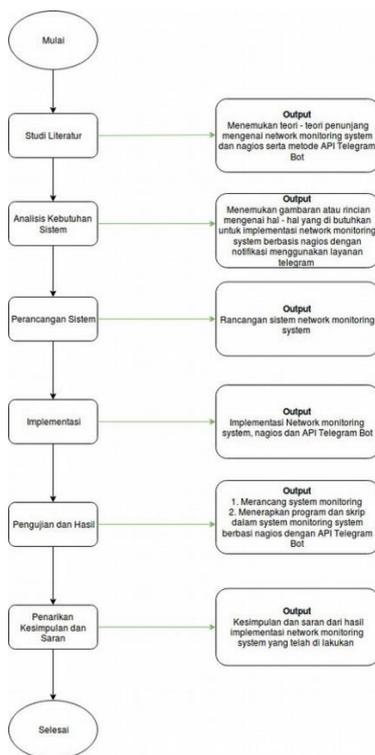
## 2. METODE PENELITIAN

### 2.1 Pengumpulan Data

Metode pengumpulan dan pengambilan data ini digunakan untuk memperoleh informasi dan data yang nantinya akan menjadi acuan untuk penulis untuk melakukan perancangan dan menyelesaikan masalah yang penulis hadapi, metode yang digunakan untuk penelitian ini yaitu metode studi pustaka dimana penulis akan melakukan pencarian data dan informasi mengenai penelitian yang terkait dengan penelitian ini dan akan mengumpulkan data dan informasi yang terkait sehingga dengan seiring berjalannya waktu dapat membantu penulis untuk menyelesaikan penelitian ini, hasil dari pencarian data dan informasi tentang penelitian terkait akan digunakan sebagai masukan yang diperlukan untuk kebutuhan penelitian pada tugas akhir ini.

Dan pada penelitian ini penulis menggunakan metode penelitian pengembangan (*Development Research*) dengan cara menggunakan sebuah *software* atau sistem yang sudah ada lalu selanjutnya dilakukan pendalaman mengenai *software* tersebut dan melakukan suatu pola atau hal yang baru yang nantinya diharapkan dapat menghasilkan suatu hal yang lebih efektif dan lebih efisien di kemudian hari.

## 2.2 Tahapan Penelitian



Gambar 1. Flowchart Tahapan Penelitian

Berikut penjelasan dari gambar di atas mengenai tahapan-tahapan yang akan dilakukan dalam penelitian ini:

### a. Studi Literatur

Pada tahapan ini penulis mencari referensi mengenai *network monitoring system*, *nagios*, Telegram Bot dan bahasa pemrograman python serta hal-hal yang berkaitan dengan penelitian yang penulis teliti dari berbagai sumber seperti jurnal terkait, artikel, buku elektronik (*e-book*), skripsi penelitian terkait maupun

dari *website* resmi untuk dapat lebih dalam lagi mengkaji mengenai *network monitoring system*.

### b. Analisis Kebutuhan Sistem

Tahapan ini dilakukan untuk menganalisis apa saja kebutuhan yang diperlukan di dalam sebuah sistem dalam pengimplementasian *network monitoring system*, apakah sudah optimal atau belum dalam penerapannya.

### c. Perancangan Sistem

Pada tahapan ini akan dilakukan perancangan sistem yang dibuat dan nantinya akan diujikan, perancangan sistemnya meliputi perancangan *nagios* yang nantinya akan diterapkan sistem alerting dengan notifikasi telegram menggunakan bahasa pemrograman python dengan Telegram Bot API di Linux ubuntu 16.04.

### d. Implementasi

Pada tahapan selanjutnya, setelah dilakukan proses pengumpulan dan pengambilan data pada *network monitoring system* tahapan selanjutnya proses implementasi, di tahapan ini akan dilakukan proses instalasi *nagios* dan konfigurasi *nagios* serta menerapkan program dan skrip dalam sistem monitoring *nagios* menggunakan Python Telegram Bot.

### e. Pengujian dan Hasil

Pada tahapan pengujian dan hasil, setelah melewati proses implementasi akan dilakukan pengujian terhadap *network monitoring system* mengenai proses monitoring menggunakan *alerting telegram*.

### f. Kesimpulan dan Saran

Pada tahapan ini berisi tentang kesimpulan mengenai semua keseluruhan proses tersebut yang telah dilakukan dan saran mengenai pengembangan selanjutnya.

## 3. URAIAN PENELITIAN

Analisis dan perancangan dalam penelitian ini dilakukan dalam beberapa tahap diantaranya analisis kebutuhan *hardware* dan *software*, perancangan sistem *logic*, perancangan sistem fisik, perancangan *alerting*, dan rancangan pengujian yang akan digunakan pada penelitian ini, berikut adalah penjelasan nya:

### I. Analisis Kebutuhan *Hardware* dan *Software*

#### a. Kebutuhan *hardware*

Dalam penelitian ini, penulis menggunakan laptop dengan spesifikasi sebagai berikut:

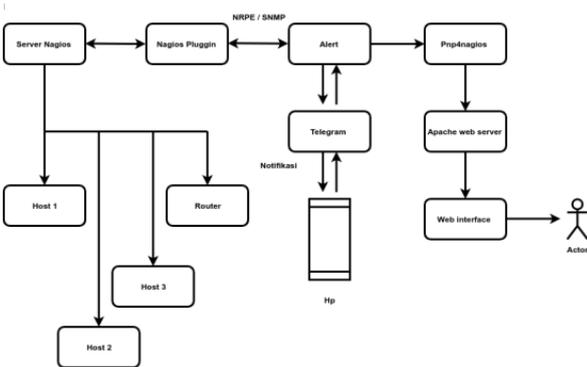
- Processor : Intel(R) Core(TM) i5-6200U  
CPU @ 2.30GHz
- RAM : 8GB
- Hardisk : 500GB

#### b. Kebutuhan *software*

Dalam penelitian ini penulis menggunakan perangkat lunak (*software*) yang dibutuhkan berdasarkan studi literatur yang berkaitan dengan penerapan atau

implementasi sistem monitoring *nagios*, serta perangkat tersebut dipilih berdasarkan hasil rekomendasi yang didapatkan penulis dari jurnal terkait dengan sedikit kemungkinan terjadinya *crack*, dan semua perangkat lunak di bawah dapat diunduh dari internet. Berikut perangkat lunak (*software*) yang akan digunakan OS Ubuntu 16.04, *Nagios Core*, *Nagios Pluggins*, NRPE, SNMP, Nsclient++, PNP4Nagios, Virtualbox, Telegram Bot, Python, *Web Browser*, *Apache Web Server*.

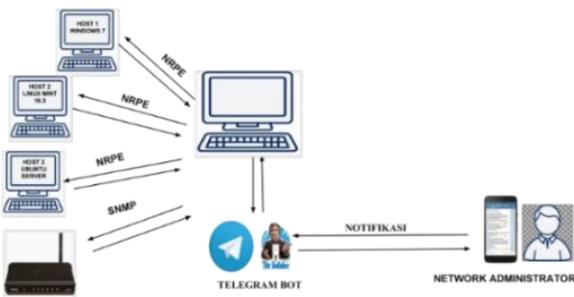
II. Perancangan Arsitektur Sistem Logik



Gambar 2. Rancangan Arsitektur Sistem Logik

Pada gambar di atas menunjukkan rancangan arsitektur sistem logik yang akan berjalan pada server *nagios*, pada penelitian ini server *nagios* akan memantau 4 host yaitu host Ubuntu, Linux Mint, Windows dan router, server *nagios* akan mendapatkan informasi terkait status host yang di pantau melalui *nagios pluggins*, NRPE akan menjalankan *nagios pluggins* dari jarak jauh di host Ubuntu, Linux Mint, Windows dan SNMP juga akan menjalankan *nagios pluggins* dari jarak jauh di host router, nantinya NRPE dan SNMP akan memantau metrik mesin jarak jauh (penggunaan *disk*, *memory usage*, *service HTTP* dan *service FTP*), jika terjadi *trouble* pada host yang di pantau *nagios* akan mengirimkan notifikasi ke group telegram administrator dan pada penelitian ini juga akan di tambahkan fitur chatbot telegram sehingga *network administrator* dapat mengetahui status host yang dipantau, dan PNP4Nagios akan menampilkan grafik dari status host yang di pantau ke dashboard *nagios*.

III. Perancangan Arsitektur Sistem Fisik

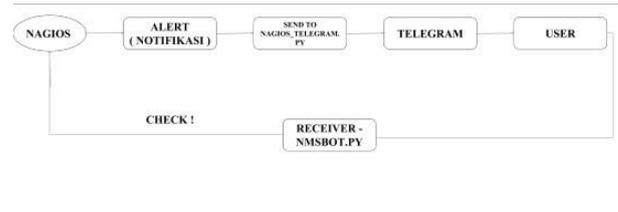


Gambar 3. Rancangan Arsitektur Sistem Fisik

Pada gambar di atas menunjukkan rancangan arsitektur sistem fisik dimana komputer yang bertindak sebagai server *nagios*

akan memantau 4 host yang terdiri dari host Ubuntu, Linux Mint, Windows dan router melalui host jarak jauh NRPE dan SNMP jika terjadi *trouble* pada host yang di pantau server *Nagios* akan mengirimkan notifikasi ke telegram bot yang akan disampaikan ke *network administrator* dan pada penelitian ini penulis menambahkan fitur chatbot telegram sehingga *nagios* dapat mengirimkan notifikasi ke telegram dan juga dapat berkomunikasi dengan *nagios* melalui *command* yang telah dibuat sebelumnya.

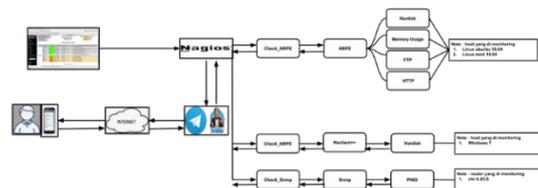
IV. Perancangan Alerting



Gambar 4. Rancangan Alerting

Pada gambar di atas menunjukkan perancangan alur *alerting* yang akan di gunakan pada penelitian kali ini, server *nagios* akan memberikan notifikasi jika terjadi *trouble* pada host yang di pantau nantinya *nagios* akan menjalankan *script nagios\_telegram.py* selanjutnya *nagios* akan memberikan notifikasi ke telegram bot dan user, karena pada penelitian ini penulis akan menambahkan fitur *chatbot* telegram sehingga akan di jalankan *script* pada server *nagios* dengan nama *nmsbot.py* setelah *script* tersebut dijalankan dan dilakukan pengecekan ke server *nagios* selanjutnya proses pertama akan diberjalan lagi hal itu dilakukan secara terus menerus.

V. Rancangan Pengujian



Gambar 5. Rancangan Pengujian

Pada rancangan pengujian sistem ini dashboard *nagios* akan menampilkan proses dari status host, service, dan performa system yang akan diuji, host yang akan dipantau adalah host Ubuntu meliputi (performa *hardisk*, *memory usage*, *service HTTP*), host Linux Mint meliputi (performa *hardisk*, *memory usage*, *service FTP*), host Windows meliputi (performa *hardisk*), router meliputi (status host), jika terjadi *trouble* pada host, performa ataupun *service* yang dipantau *nagios* akan memberikan *alert* (notifikasi) ke grup telegram dengan tambahan fitur *chatbot* telegram *nagios* dapat berinteraksi dengan telegram melalui *command* yang telah ditentukan sebelumnya.

4. PEMBAHASAN

### 4.1 Pengujian

Pengujian bertujuan untuk memastikan sistem agar dapat berjalan dengan semestinya. Pada pengujian ini akan dilakukan pengujian terhadap beberapa performa dan service yang terdapat pada sistem. Terdapat 4 komponen utama yang diuji diantaranya adalah monitoring (*hardisk, memory usage, HTTP, FTP*), notifikasi Telegram, dan *chatbot* telegram, adapun pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian pemantauan status host
2. Pengujian pemantauan status *service* host
3. Pengujian notifikasi telegram host dan *service*
4. Pengujian proses pemantauan melalui *chatbot* telegram

### 4.2 Data Hasil Pengujian

Hasil pengujian *monitoring system* terhadap efektivitas nagios dan efektivitas terhadap notifikasi telegram akan di jelaskan pada tabel di bawah ini:

**Tabel 1.** Data Analisis Checking Host dan Service “Down”

No	Host	Waktu Penon-aktifan	Waktu deteksi nagios	Durasi	Status / service
1	Host Ubuntu	15:15:17	15.17.00	0:02:17	Down
2	Hardisk Ubuntu	13.49.03	13.52.28	00:03:25	Critical
3	Memory Usage Ubuntu	14:39:36	14.43.04	0:04:40	Critical
4	Http Ubuntu	17.04.34	17.05.03	00:01:31	Critical
5	Host Linux Mint	19.07.01	19.07.17	0:00:16	Down
6	Hardisk Linux MInt	21.27.13	21.31.43	00:04:30	Critical
7	Ftp Linux Mint	17.05.20	17.05.47	00:00:27	Critical
8	Host Windows	19.28.20	19.30.40	0:02:20	Down
9	Hardisk Windows	08.09.10	08.12.18	00:03:8	Critical
10	Host Router	14.20.01	14.26.02	00:06:01	Down

**Tabel 2.** Data Analisis Checking Host dan Service “Up”

No	Host	Waktu Penonaktifan	Waktu deteksi nagios	Durasi	Status / service
1	Host Ubuntu	14.39.30	14.39.36	0:00:06	Up
2	Hardisk Ubuntu	22.15.04	22.20.48	00:05:44	Ok
3	Memory Usage Ubuntu	19.34.01	19.34.21	0:00:20	Ok
4	Http Ubuntu	13.02.47	13.04.03	00:02:44	Ok
5	Host Linux Mint	18.21.10	18.21.41	0:00:31	Up
6	Hardisk Linux MInt	11.23.00	11.26.21	00:003:21	Ok
7	Ftp Linux Mint	21.09.01	21.10.07	00:01:06	Ok
8	Host Windows	19.23.10	19.24.29	0:01:19	Up
9	Hardisk WIndows	07.50.00	07.55.10	00:05:10	Critical
10	Host Router	14.29.01	14.31.06	00:02:05	Up

Selanjutnya hasil pengujian analisis pemantauan host dan *service* dari rata rata waktu yang dibutuhkan untuk nagios memberikan notifikasi ke telegram akan di jelaskan pada tabel di bawah ini:

**Tabel 3.** Data Analisis Pemantauan *Host* dan *Service*

No	Item pengujian	Hasil pengujian	Rata-rata	Catatan
1	Hardisk Host Ubuntu	Berhasil	5 – 10 menit	Notifikasi Muncul
2	Memory Usage Host Ubuntu	Berhasil	5 – 10 menit	Notifikasi Muncul
3	Http Host Ubuntu	Berhasil	5 – 10 menit	Notifikasi Muncul
4	Hardisk Host Linux Mint	Berhasil	5 – 10 menit	Notifikasi Muncul
5	Ftp Host Linux Mint	Berhasil	5 – 10 menit	Notifikasi Muncul
6	Hardisk Host Windows	Berhasil	5 – 10 menit	Notifikasi Muncul

7	PING Host Ubuntu	Berhasil	5 menit	Notifikasi Muncul
8	PING Host Linux mint	Berhasil	5 menit	Notifikasi Muncul
9	PING Host WIndows	Berhasil	5 menit	Notifikasi Muncul
10	PING Host Router	Berhasil	5 menit	Notifikasi Muncul

1. Pengujian pemantauan status host

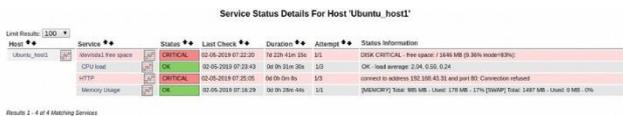
Pengajuan pemantauan *network monitoring system* berbasis *nagios* terhadap host yang dimonitoring dilakukan setelah host terinstal pada *nagios*, selanjutnya penulis melakukan pengujian efektivitas nms berbasis *nagios* terhadap host Linux Ubuntu, host Linux Mint, host Windows dan host router dengan menguji apakah host tersebut dapat dipantau. Dari hasil pengujian dapat dilihat bahwa *nagios* dapat memonitoring host Ubuntu, host Linux Mint, host Windows, dan host router, pada pengujian kali ini penulis akan menampilkan gambar dari pengujian efektivitas *nagios* terhadap host Ubuntu di bawah ini:



Gambar 6. Pengujian NMS Nagios dalam memantau host ubuntu

2. Pengujian pemantauan status host

Pengujian pemantauan *network monitoring system* berbasis *nagios* terhadap status *service* atau performa yang di monitoring dilakukan dengan mematikan service HTTP dan membuat *hardisk* melebihi kapasitas (*threshod*) yang ditentukan untuk host Ubuntu, berikut adalah pengujian efektivitas NMS terhadap status *service* atau performa host yang dipantau:



Gambar 7. Pengujian NMS Nagios dalam memantau service http dan disk pada host Ubuntu

3. Pengajuan notifikasi telegram host dan service

Pada tahapan ini penulis melakukan pengujian efektivitas *network monitoring system* berbasis *nagios* dengan notifikasi ke telegram, dilakukan dengan melihat apakah notifikasi berfungsi sebagai mana yang di harapkan pada pengujian kali ini, skenario yang akan di lakukan yaitu notifikasi terjadi jika *network monitoring system nagios* menemui kondisi host yang di monitoring berada dalam kondisi down dan kondisi saat dilakukan *recovery*, dengan pengaturan *max\_check\_attempts* sebanyak 5 kali dan interval waktu selama 5 menit, maka *nagios* akan

melakukan pengujian selama 5 kali dengan waktu *max\_check\_attempts* melakukannya selama 30 detik hal itu dilakukan sampai mencapai batas *max\_check\_attempts* jika telah melewati batasnya maka akan ada kondisi *hardstate* dimana akan memberikan notifikasi ke telegram bahwa host tersebut mengalami kondisi *down* atau *recovery*. Saat host telah memberikan notifikasi maka interval waktu untuk pengecekan host selanjutnya adalah 5 menit. Sedangkan untuk pengujian service, penulis akan melakukan skenario pengujian dimana saat kondisi service dimatikan dan saat kondisi service di *recovery nagios* akan mengirimkan notifikasi ke group telegram mengenai kondisi service saat itu, sama hal nya dengan pengujian host, service di uji dengan pengaturan *max\_check\_attempts* sebanyak 3 kali dan interval waktu selama 10 menit maka jika service mengalami kondisi *down* atau *recovery* akan dilakukan pengujian selama 3 kali,dengan waktu *max\_check\_attempts* melakukannya selama 30 detik hal itu dilakukan sampai mencapai batas *max\_check\_attempts* jika telah melewati batasnya maka akan ada kondisi *hardstate* dimana akan memberikan notifikasi ke telegram bahwa *service* tersebut mengalami kondisi *down* atau *recovery*. Saat service telah memberikan notifikasi maka interval waktu untuk pengecekan host selanjutnya adalah 10 menit.

Pemantauan hosts diuji pada host Ubuntu yang dilakukan sebanyak 5 kali dengan kondisi saat host “DOWN” dan kondisi saat host dilakukan *recovery*, Hasil pengujian pemantauan status host Ubuntu dalam keadaan “DOWN” dapat dilihat pada Tabel 4 dan “UP” pada Tabel 5.

Tabel 4. Data status notifikasi dari pemantauan host Ubuntu “Down”

Pengujian	Host	max_check_attempts	Status	Notifikasi
1	Ubuntu_host1	1	Host Down	Tidak ada notifikasi
2	Ubuntu_host1	2	Host Down	Tidak ada notifikasi
3	Ubuntu_host1	3	Host Down	Tidak ada notifikasi
4	Ubuntu_host1	4	Host Down	Tidak ada notifikasi
5	Ubuntu_host1	5	Host Down	Notifikasi Muncul

Tabel 5. Data status notifikasi dari pemantauan host Ubuntu “Up”

Pengujian	Host	max_check_attempts	Status	Notifikasi
1	Ubuntu_host1	1	Host up	Tidak ada notifikasi
2	Ubuntu_host1	2	Host up	Tidak ada notifikasi
3	Ubuntu_host1	3	Host up	Tidak ada notifikasi

4	Ubuntu_host1	4	Host up	Tidak ada notifikasi
5	Ubuntu_host1	5	Host up	Notifikasi Muncul

Pemantauan kapasitas *hardisk* host Ubuntu dilakukan sebanyak 3 kali pengujian, dengan kondisi *service* mengalami gangguan/eror dan kondisi *service* dilakukan *recovery*. Hasil pemantauan saat *hardisk* dalam keadaan melewati kapasitas penggunaan dapat dilihat pada Tabel 6. dan saat *hardisk* dilakukan *recovery*. Hasil pemantauan saat *hardisk recovery* dijelaskan pada Tabel 7.

**Tabel 6.** Data status notifikasi dari pemantauan kapasitas hardisk pada Ubuntu “Down”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	Hardisk	1	CRITICAL	Tidak ada notifikasi
2	Hardisk	2	CRITICAL	Tidak ada notifikasi
3	Hardisk	3	CRITICAL	Notifikasi Muncul

**Tabel 7.** Data status notifikasi dari pemantauan kapasitas hardisk pada Ubuntu “Recovery”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	Hardisk	1	OK	Tidak ada notifikasi
2	Hardisk	2	OK	Tidak ada notifikasi
3	Hardisk	3	OK	Notifikasi Muncul

Pemantauan *memory usage* pada host Ubuntu dilakukan sebanyak 3 kali pengujian dengan kondisi *memory usage* mengalami gangguan/eror dan kondisi saat dilakukan *recovery*. Hasil pemantauan saat *memory usage* dalam keadaan melewati kapasitas penggunaan *memory usage* dapat dilihat pada Tabel 8 dan saat *memory usage* dilakukan *recovery* di jelaskan pada tabel 9.

**Tabel 8.** Data pemantauan status notifikasi dari pemantauan penggunaan *memory usage* pada ubuntu “Down”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	Memory Usage	1	CRITICAL	Tidak ada notifikasi
2	Memory Usage	2	CRITICAL	Tidak ada notifikasi

3	Memory Usage	3	CRITICAL	Notifikasi Muncul
---	--------------	---	----------	-------------------

**Tabel 9.** Data status notifikasi dari pemantauan penggunaan *memory usage* pada Ubuntu “Recovery”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	Memory Usage	1	OK	Tidak ada notifikasi
2	Memory Usage	2	OK	Tidak ada notifikasi
3	Memory Usage	3	OK	Notifikasi Muncul

Pengujian *service* HTTP pada host Ubuntu dilakukan sebanyak 3 kali pengujian dengan kondisi *service* HTTP mengalami gangguan/eror dan kondisi *service* HTTP dilakukan *recovery*. Hasil pemantaun *service* saat HTTP dalam keadaan dinonaktifkan dapat dilihat pada Tabel 10, dan saat HTTP dilakukan *recovery* dijelaskan pada tabel 11.

**Tabel 10.** Data status notifikasi dari pemantauan *service* HTTP pada Ubuntu “Down”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	HTTP	1	CRITICAL	Tidak ada notifikasi
2	HTTP	2	CRITICAL	Tidak ada notifikasi
3	HTTP	3	CRITICAL	Notifikasi Muncul

**Tabel 11.** Data status notifikasi dari pemantauan *service* HTTP pada Ubuntu “Recovery”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	HTTP	1	OK	Tidak ada notifikasi
2	HTTP	2	OK	Tidak ada notifikasi
3	HTTP	3	OK	Notifikasi Muncul

Pemantauan hosts diuji pada host Linux Mint yang dilakukan sebanyak 5 kali dengan kondisi saat host “DOWN” dan kondisi saat host dilakukan *recovery*. Hasil pemantauan status host linux mint dalam keadaan “DOWN” dapat dilihat pada Tabel 12 dan “UP” pada Tabel 13.

**Tabel 12.** Data status notifikasi dari pemantauan host linux mint “Down”

Pengujian	Host	max_chec k_attempts	Status	Notifikasi
1	Mint_hos t3	1	Host Down	Tidak ada notifikasi
2	Mint_hos t3	2	Host Down	Tidak ada notifikasi
3	Mint_hos t3	3	Host Down	Tidak ada notifikasi
4	Mint_hos t3	4	Host Down	Tidak ada notifikasi
5	Mint_hos t3	5	Host Down	Notifikasi Muncul

**Tabel 13.** Data status notifikasi dari pemantauan host Linux Mint “Up”

Pengujian	Host	max_chec k_attempts	Status	Notifikasi
1	Mint_hos t3	1	Host up	Tidak ada notifikasi
2	Mint_hos t3	2	Host up	Tidak ada notifikasi
3	Mint_hos t3	3	Host up	Tidak ada notifikasi
4	Mint_hos t3	4	Host up	Tidak ada notifikasi
5	Mint_hos t3	5	Host up	Notifikasi Muncul

Pemantauan status *hardisk* pada host Linux Mint di lakukan sebanyak 3 kali pengujian dengan kondisi hardisk mengalami gangguan/eror dan kondisi hardisk dilakukan *recovery*. Hasil pemantaun status hardisk saat hardisk dalam keadaan melewati kapasitas penggunaan dapat dilihat pada Tabel 14, dan saat hardisk dilakukan *recovery* di jelaskan pada tabel 15.

**Tabel 14.** Data status notifikasi dari pemantauan kapasitas hardisk pada linux mint “Down”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	Hardisk	1	CRITICAL	Tidak ada notifikasi
2	Hardisk	2	CRITICAL	Tidak ada notifikasi
3	Hardisk	3	CRITICAL	Notifikasi Muncul

**Tabel 15.** Data status notifikasi dari pemantauan kapasitas hardisk linux mint “Recovery”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi

1	Hardisk	1	OK	Tidak ada notifikasi
2	Hardisk	2	OK	Tidak ada notifikasi
3	Hardisk	3	OK	Notifikasi Muncul

Pemantauan *service* FTP Linux Mint dilakukan sebanyak 3 kali pengujian dengan kondisi *service* mengalami gangguan/eror dan kondisi *service* dilakukan *recovery*. Hasil pemantauan *service* saat FTP saat dinonaktifkan dapat dilihat pada Tabel 16, dan saat FTP Linux Mint di lakukan *recovery* di jelaskan pada tabel 17.

**Tabel 16.** Data status notifikasi dari pemantauan *service* FTP pada Linux Mint “Down”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	FTP	1	CRITICAL	Tidak ada notifikasi
2	FTP	2	CRITICAL	Tidak ada notifikasi
3	FTP	3	CRITICAL	Notifikasi Muncul

**Tabel 17.** Data status notifikasi dari pemantauan *service* FTP pada Linux Mint “Recovery”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	FTP	1	OK	Tidak ada notifikasi
2	FTP	2	OK	Tidak ada notifikasi
3	FTP	3	OK	Notifikasi Muncul

Pemantauan hosts diuji pada host windows yang dilakukan sebanyak 10 kali dengan kondisi saat host “DOWN” dan kondisi saat host dilakukan *recovery*. Hasil pemantauan status saat host Windows dalam keadaan “DOWN” dapat dilihat pada Tabel 18 dan “UP” pada Tabel 19.

**Tabel 18.** Data status notifikasi dari pemantauan host Windows “Down”

Pengujian	Host	max_check _attempts	Status	Notifikasi
1	Winserver	1	Host Down	Tidak ada notifikasi
2	Winserver	2	Host Down	Tidak ada notifikasi
3	Winserver	3	Host Down	Tidak ada notifikasi

4	Winserv	4	Host Down	Tidak ada notifikasi
5	Winserv	5	Host Down	Tidak ada notifikasi
6	Winserv	6	Host Down	Tidak ada notifikasi
7	Winserv	7	Host Down	Tidak ada notifikasi
8	Winserv	8	Host Down	Tidak ada notifikasi
9	Winserv	9	Host Down	Tidak ada notifikasi
10	Winserv	10	Host Down	Notifikasi Muncul

**Tabel 19.** Data status notifikasi dari pemantauan host Windows “Up”

Pengujian	Host	max_check_attempts	Status	Notifikasi
1	Winserv	1	Host up	Tidak ada notifikasi
2	Winserv	2	Host up	Tidak ada notifikasi
3	Winserv	3	Host up	Tidak ada notifikasi
4	Winserv	4	Host up	Tidak ada notifikasi
5	Winserv	5	Host up	Tidak ada notifikasi
6	Winserv	6	Host up	Tidak ada notifikasi
7	Winserv	7	Host up	Tidak ada notifikasi
8	Winserv	8	Host up	Tidak ada notifikasi
9	Winserv	9	Host up	Tidak ada notifikasi
10	Winserv	10	Host up	Notifikasi Muncul

Pengujian status hardisk pada host Windows di lakukan sebanyak 3 kali pengujian dengan kondisi *hardisk* mengalami gangguan/eror dan kondisi *hardisk* dilakukan *recovery*. Hasil pengujian *hardisk* saat *hardisk* dalam keadaan melewati kapasitas penggunaan dapat dilihat pada Tabel 20, dan saat hardisk dalam keadaan *recovery* di jelaskan pada tabel 21.

**Tabel 20.** Data status notifikasi dari pemantauan kapasitas *hardisk* pada Windows “Down”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi

1	Hardisk	1	CRITICAL	Tidak ada notifikasi
2	Hardisk	2	CRITICAL	Tidak ada notifikasi
3	Hardisk	3	CRITICAL	Notifikasi Muncul

**Tabel 21.** Data status notifikasi dari pemantauan kapasitas *hardisk* pada Windows “Recovery”

Pengujian	Performa yang diuji	max_chec k_attempts	Status	Notifikasi
1	Hardisk	1	OK	Tidak ada notifikasi
2	Hardisk	2	OK	Tidak ada notifikasi
3	Hardisk	3	OK	Notifikasi Muncul

Pemantauan hosts diuji pada host router yang dilakukan sebanyak 10 kali dengan kondisi saat host “DOWN” dan kondisi saat host dilakukan *recovery*. Hasil pemantauan status saat host Windows dalam keadaan “DOWN” dapat dilihat pada Tabel 22 dan “UP” pada Tabel 23.

**Tabel 22.** Data status notifikasi dari pemantauan host router “Down”

Pengujian	Host	max_chec k_attempts	Status	Notifikasi
1	Router	1	Host Down	Tidak ada notifikasi
2	Router	2	Host Down	Tidak ada notifikasi
3	Router	3	Host Down	Tidak ada notifikasi
4	Router	4	Host Down	Tidak ada notifikasi
5	Router	5	Host Down	Tidak ada notifikasi
6	Router	6	Host Down	Tidak ada notifikasi
7	Router	7	Host Down	Tidak ada notifikasi
8	Router	8	Host Down	Tidak ada notifikasi
9	Router	9	Host Down	Tidak ada notifikasi
10	Router	10	Host Down	Notifikasi Muncul

**Tabel 23.** Data status notifikasi dari pemantauan host router “Up”

Pengujian	Host	max_chec k_attempts	Status	Notifikasi

1	Router	1	Host up	Tidak ada notifikasi
2	Router	2	Host up	Tidak ada notifikasi
3	Router	3	Host up	Tidak ada notifikasi
4	Router	4	Host up	Tidak ada notifikasi
5	Router	5	Host up	Tidak ada notifikasi
6	Router	6	Host up	Tidak ada notifikasi
7	Router	7	Host up	Tidak ada notifikasi
8	Router	8	Host up	Tidak ada notifikasi
9	Router	9	Host up	Tidak ada notifikasi
10	Router	10	Host up	Tidak ada notifikasi

4. Pengujian proses pemantauan melalui chatbot telegram

I. Pengujian Pemantauan Status Host dan Status Service dari Host

Bertujuan untuk mengetahui status hidup atau matinya suatu komputer melalui chatbot telegram melalui perintah /checkhost sebagaimana perintah yang telah di jelaskan di bawah ini:

- a. /checkhost winserver // mengecek status host winserver

Bertujuan untuk mengetahui status hidup atau matinya suatu service komputer dengan melalui chatbot telegram melalui perintah /checkservicehost sebagai perintah yang telah di jelaskan.

- b. /checkservice Ubuntu\_host1 Memory Usage // mengecek service memory usage host ubuntu



Gambar 8. Telegram Bot

4.3 Analisis Pengujian

a. Analisa Pengujian Notifikasi Host

Dari pengujian notifikasi Telegram terhadap hosts yang telah dilakukan, didapatkan hasil yaitu Nagios berhasil diuji karena dapat mengirimkan pesan notifikasi alert kepada admin dalam Group Telegram “Tes mon”. Durasi saat Nagios melakukan pengiriman notifikasi tergantung pada konfigurasi retry\_interval dan max\_check\_attempts.

Dari data pengujian notifikasi host Ubuntu dan Linux Mint dalam kondisi “DOWN” didapat durasi saat Nagios mendeteksi perubahan status “DOWN” sampai Nagios melakukan pengiriman notifikasi terjadi selama selang waktu kurang dari 5 menit untuk host Ubuntu dan Linux Mint. Kondisi tersebut sesuai dengan konfigurasi yang telah didefinisikan, yaitu max\_check\_attempts yang diatur 5 kali, dan retry\_interval yang diatur 1 menit. Jadi, ketika Nagios mendeteksi perubahan status pada host Ubuntu dan linux mint (dalam keadaan soft state 1), Nagios melakukan pengecekan ulang dalam selang interval 1 menit yang dilakukan selama 5 kali sampai host dalam keadaan hard state.

Dan dari data pengujian notifikasi host Windows dan router dalam kondisi “DOWN” didapat durasi saat Nagios mendeteksi perubahan status “DOWN” sampai Nagios melakukan pengiriman notifikasi terjadi selama selang waktu kurang dari 10 menit untuk host Windows dan router. Kondisi tersebut sesuai dengan konfigurasi yang telah didefinisikan, yaitu max\_check\_attempts yang diatur 10 kali, dan retry\_interval yang diatur 1 menit. Jadi, ketika Nagios mendeteksi perubahan status pada host Windows dan router (dalam keadaan soft state 1), Nagios melakukan pengecekan ulang dalam selang interval 1 menit yang dilakukan selama 10 kali sampai host dalam keadaan hard state.

Sedangkan dari data pengujian notifikasi host Ubuntu , Linux Mint, Windows, dan router ketika dilakukan recovery sampai kondisi host menjadi “UP” didapat hasil yaitu tidak ada selang waktu diantaranya. Kondisi tersebut dikarenakan ketika Nagios melakukan pengecekan pertama untuk memastikan perubahan status menjadi “UP”, Nagios langsung mendeteksi bahwa keadaan tersebut sudah dalam keadaan hard state. Notifikasi Nagios terhadap hosts melalui Telegram akan dikirim setelah Nagios mendeteksi kondisi hard state setelah dilakukan pengecekan berulang, hal ini dilakukan untuk memastikan kembali apakah kondisi hosts benar dalam keadaan bermasalah atau tidak sebelum dikirimkan kepada Group Telegram Admin. Pengiriman notifikasi Telegram pun bergantung pada koneksi internet di sisi server Nagios. Jika tidak ada koneksi internet/koneksi internet terputus, maka Nagios tidak akan mengirimkan notifikasi kepada Group Telegram Admin.

b. Analisis Pengujian Notifikasi Services

Dari pengujian notifikasi Telegram terhadap *services* yang telah dilakukan, didapatkan hasil yaitu Nagios berhasil diuji karena dapat mengirimkan pesan notifikasi *alert* kepada admin dalam Group Telegram “Tes mon”. Durasi saat Nagios melakukan pengiriman notifikasi tergantung pada konfigurasi *retry\_interval* dan *max\_check\_attempts*. Dari data pengujian notifikasi *services* FTP, HTTP, *disk* dan *memory usage* dalam kondisi “CRITICAL” didapat saat Nagios mendeteksi perubahan status “CRITICAL” sampai Nagios melakukan pengiriman notifikasi terjadi selama selang waktu kurang dari 6 menit. Kondisi tersebut sesuai dengan konfigurasi yang telah didefinisikan, yaitu *max\_check\_attempts* yang diatur 3 kali, dan *retry\_interval* yang diatur 2 menit. Jadi, ketika Nagios mendeteksi perubahan status “CRITICAL” pada *services* FTP dan HTTP (dalam keadaan *soft state* 1), Nagios melakukan pengecekan ulang dalam selang interval 2 menit yang dilakukan selama 3 kali sampai *services* dalam keadaan *hard state*. Setelah *services* dalam keadaan *hard state*, pada saat itulah Nagios mengirimkan notifikasi melalui Telegram Messenger.

Sedangkan dari data pengujian notifikasi service FTP, HTTP, *disk*, *memory usage* ketika dilakukan *recovery* sampai kondisi *services* menjadi “OK” didapat hasil yaitu tidak ada selang waktu diantaranya. Kondisi tersebut dikarenakan ketika Nagios melakukan pengecekan pertama untuk memastikan perubahan status *services* menjadi “OK”, Nagios langsung mendeteksi bahwa keadaan tersebut sudah dalam keadaan *hard state*. Notifikasi Nagios terhadap *services* melalui Telegram akan dikirim setelah Nagios mendeteksi kondisi *hard state* setelah dilakukan pengecekan berulang, hal ini dilakukan untuk memastikan kembali apakah kondisi *services* benar dalam keadaan bermasalah atau tidak sebelum dikirimkan kepada Group Telegram Admin. Pengiriman notifikasi Telegram pun bergantung pada koneksi internet di sisi server Nagios. Jika tidak ada koneksi internet/koneksi internet terputus, maka Nagios tidak akan mengirimkan notifikasi kepada Group Telegram Admin.

### c. Analisis Pengujian Chatbot Telegram

Dari pengujian *chatbot* telegram untuk mengetahui status host dan status *service* yang di-*scarping* dari *dashboard* web *nagios* yang telah dilakukan, didapatkan hasil yaitu *chatbot* telegram untuk memantau status host dan status *service* berhasil diuji karena dapat mengirimkan pesan status host dan *service* yang di *scarping* dari *dashboard* *nagios* ke telegram kepada admin dalam Group Telegram “Tes mon”. Dengan menjalankan skrip yang telah di letakkan di server *nagios*, pengiriman *chatbot* telegram pun bergantung pada koneksi internet di sisi server Nagios. Jika tidak ada koneksi internet/koneksi internet terputus, maka Nagios tidak akan mengirimkan status dari host dan *service* yang dicek serta jika skrip dari *chatbot* telegram tidak di jalankan maka *chatbot* juga tidak dapat digunakan dan tidak dapat

mengirimkan status host dan *service* kepada Group Telegram Admin.

## 5. KESIMPULAN

### 5.1 Kesimpulan

1. Hasil rancangan monitoring dilakukan secara logik dan fisik yaitu dengan menyiapkan 4 sistem komputer yang terhubung dalam jaringan, dan satu server telah terinstall *nagios* dan telah terintegrasi dengan telegram bot, untuk memantau host dan *service* dari komputer yang dimonitoring.
2. Hasil rancangan program sistem notifikasi untuk monitoring jaringan yaitu dengan membuat skrip menggunakan bahasa pemrograman python yang dijalankan dalam server monitoring dan untuk fitur *chatbot* telegram dibuat skrip yang akan memanggil *command* yang ditentukan sebelumnya pada skrip, *command* tersebut telah di *define* sebelumnya.
3. Sistem monitoring dengan menggunakan telegram bot dapat dinyatakan efektif karena waktu proses notifikasi saat proses server up atau down dapat dilakukan antara antara selang waktu 5 menit untuk host dan 10 menit untuk *service*.
4. Teknik yang digunakan agar *chatbot* telegram terintegrasi dengan sistem *nagios* yaitu dengan membuat skrip otomatisasi menggunakan metode *scarping* web dan menggunakan *libre beautiful soap* yang dijalankan oleh server *nagios*.

### 5.2 Saran

Dalam penelitian dilakukan pada lingkungan sistem komputer virtual, disarankan dapat dijalankan dalam jaringan yang sesungguhnya. Untuk skrip otomatisasi monitoring jaringan masih dituliskan dalam dua skrip yang terpisah, sebaiknya dituliskan kedalam satu skrip tunggal agar lebih efektif dan efisien. Pada penelitian selanjutnya diharapkan dapat mengukur *delay* notifikasi yang diterima dari *nagios* ke telegram bot secara lebih spesifik.

### Ucapan Terima Kasih

Terima kasih kepada semua pihak yang telah mendukung tulisan ini terutama kepada : Ketua STT NF, Kepala LPPM STT NF, Kaprodi TI, Kaprodi Si dan Teman-Teman Vixagraph sehingga penelitian ini dapat berjalan dengan baik dan lancar.

### DAFTAR PUSTAKA

- [1] F. B. Fatria, “Pengembangan Fitur Nagios untuk Pemantauan Jaringan Berbasis SMS (*Short Message Service*)”, Pekanbaru : Tugas Akhir, 2011.
- [2] H. Amnur. dkk, “Perancangan dan Implementasi Network Monitoring System Menggunakan Nagios dengan Email dan SMS Alert”, Padang : Tugas Akhir, 2014.

- [3] M. Syafrizal, "Pengantar Jaringan Komputer", Yogyakarta: Andi Offset, 2005.
- [4] E. Sutanta, "Pengantar Teknologi Informasi", Graha ilmu Yogyakarta, 2005.
- [5] A. Kurniawan, "Implementasi *Network Monitoring System* (NMS) secara Visual pada Infrastruktur Jaringan Fisik Berbasis Web", Makassar: Tugas Akhir 2012.
- [6] W. Barth, "*Nagios System and Network Monitoring*", 2nd Edition, Open Source Press GmbH, San Francisco , 2008.
- [7] G. Sastrawangsa, "Pemanfaatan Telegram Bot untuk Automatisasi Layanan dan Informasi Mahasiswa dalam Konsep *Smart Campus*", Bali, 2017.
- [8] J. Joyner, "*Python Programming For Beginners: Python Programming Language Tutorial*", First Published, United State of America, 2014.
- [9] M.P. Ninit Alfianika, "Metode Penelitian Pengajaran Bahasa Indonesia", Yogyakarta: Depublish, 2016.
- [10] M.P.Prof. Muri Yusuf, "Metode Penelitian Kuantitatif, Kualitatif Dan Penelitian Gabungan", Jakarta: Kencana, 2014.
- [11] W. Kocjan, "*Learning Nagios 4*", Birmingham: Packt Publishing Ltd, 2014.
- [12] Nagios. Nagios Core Documentation, 2016, [Online]. Available: <https://www.nagios.org> [diakses 23 Desember 2018]
- [13] E. Galstad, NRPE Documentation, 2007, [Online]. Available: <https://assets.nagios.com/downloads/nagioscore/docs/nrpe/NRPE.pdf> [diakses 23 Desember 2018]