



## ***Network Monitoring System Menggunakan Nagios dengan Event Handler Notifikasi Whatsapp***

Abdi Muttaqin<sup>1</sup>, Ferdi Chahyadi<sup>2\*</sup>, Nurul Hayati<sup>3</sup>

<sup>1,2</sup>Jurusan Teknik Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji

<sup>1,2</sup>Jl. Politeknik Senggarang, Tanjungpinang 29100

\*Corresponding Author: ferdi.chahyadi@umrah.ac.id

**Abstract**—*Network monitoring system is very important because its systems were increasingly complex. For this reason, regular maintenance of the network is needed so that it can run as it should. Therefore we need a facility that can monitor the network. Nagios is an open source-based network monitoring application that has many features such as monitoring, notification alerts, event handlers, and others. The results of the test show that Nagios can detect changes in host status down and up in less than 2 minutes, event handlers can record host status and restart problematic services. Nagios can also monitor host bandwidth usage, send host down notifications in less than 3 minutes, and send host bandwidth usage according to the time period applied by having a delay of less than 10 seconds.*

**Keywords**— *Network, Monitoring, Nagios, Event Handler, Notification Alert, Whatsapp.*

**Intisari**—Sistem *monitoring* jaringan saat ini sangat dibutuhkan dilihat karena semakin kompleksnya sistem jaringan yang digunakan, sehingga diperlukan adanya pemeliharaan teratur terhadap jaringan sehingga dapat berjalan sebagaimana mestinya. Oleh sebab itu dibutuhkan sebuah fasilitas yang bisa melakukan *monitoring* terhadap jaringan tersebut. Nagios merupakan salah satu aplikasi *monitoring* jaringan yang berbasis *open source* yang memiliki banyak fitur seperti *monitoring, notification alert, event handler*, dan lain – lain. Hasil dari pengujian didapatkan bahwa nagios dapat mendeteksi perubahan status *host down* dan *up* kurang dari 2 menit, *event handler* dapat merekam status *host* dan melakukan *restart* terhadap *service* bermasalah, melakukan *monitoring* terhadap penggunaan *bandwidth host*, mengirimkan notifikasi *host down* dengan waktu kurang dari 3 menit, dan mengirimkan penggunaan *bandwidth host* sesuai dengan jangka waktu yang diterapkan dengan memiliki waktu keterlambatan kurang dari 10 detik.

**Kata kunci**— *Network, Monitoring, Nagios, Event Handler, Notification Alert, Whatsapp.*

### **I. PENDAHULUAN**

Jaringan komputer bukanlah sesuatu yang baru dalam teknologi informasi dan komunikasi. Pada umumnya di setiap perusahaan, instansi, ataupun institusi pendidikan saat ini sudah menggunakan jaringan komputer dengan akses internet untuk memperlancar arus informasi

ataupun pertukaran datanya. Penggunaan jaringan komputer tentu tidak lepas dari penggunaan *bandwidth* untuk akses ke internet. Jaringan komputer saat ini menjadi sebuah kebutuhan, sehingga diperlukan adanya pemeliharaan dan kontrol yang teratur untuk

menjamin agar jaringan dapat berjalan sebagaimana mestinya.

*Monitoring* merupakan suatu kegiatan yang bertujuan untuk melakukan pengamatan dan pengaturan untuk sebuah jaringan. Adanya suatu jaringan pasti selalu mengalami peningkatan atau penambahan perangkat jaringan maka menjadi jaringan tersebut sangat kompleks dan membutuhkan *monitoring* secara *real-time*. *Monitoring* dapat menggambarkan bagaimana suatu *system* bekerja secara konstan seperti *router server* maupun *device* lainnya. Apabila pada suatu sistem jaringan tersebut *error/down*/tidak stabil maka secara otomatis diketahui oleh sistem administrator melalui sebuah sistem peringatan berupa email atau notifikasi lainnya. Maka dalam hal ini *monitoring* sangat diperlukan. Dengan berkembangnya IPTEK (Ilmu Pengetahuan dan Teknologi) mempengaruhi teknologi yang digunakan dalam jaringan. Contohnya aplikasi *monitoring* seperti Nagios, *Netwatch*, *Graphing*, *Whireshark* dll.

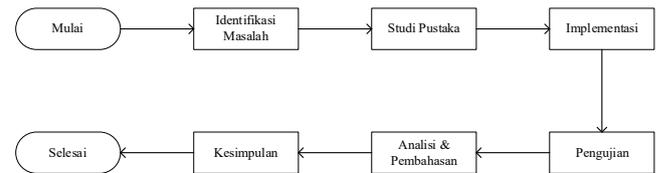
Nagios merupakan sebuah aplikasi *monitoring* yang dapat digunakan untuk *monitoring* sistem komputer, *monitoring* jaringan dan *monitoring* infrastruktur jaringan. Nagios berbasis *open source* yang dapat dijalankan pada sistem operasi Linux. Beberapa fitur yang dimiliki Nagios seperti *monitoring*, notifikasi *alert*, *report*, *event handler*, *monitoring resource* dan lain-lain. Nagios sendiri dapat dikonfigurasi sesuai dengan kebutuhan seperti menambahkan *service* untuk pengecekan status *bandwidth up/down* pada *host*.

Menurut Habibullah dan Arnaldy, (2016) 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, Whatsapp Messenger, dan lainnya. Selain fitur notifikasi *alert*, *network monitoring system* Nagios mempunyai fitur yang dapat melakukan *restart* terhadap *service* yang mengalami gangguan secara otomatis. Berdasarkan latar belakang tersebut, maka dilakukan penelitian tentang “*Network*

*Monitoring System* Nagios dengan *Event Handler* Dan Notifikasi Whatsapp”.

## II. METODE PENELITIAN

Pada metode menjelaskan tahapan-tahapan yang akan dilakukan sebagai acuan dalam penulisan penelitian ini berikut ini merupakan alur metodologi yang digunakan pada penelitian.



Gambar 1. Metodologi Penelitian

### A. Identifikasi Masalah

Permasalahan yang ada pada penelitian ini adalah bagaimana membangun sistem *monitoring* jaringan dengan Nagios yang bisa *memonitoring* kondisi pemakaian *bandwidth* oleh *host* dan memaksimal kan kinerja *event handler* dalam pemantauan kondisi dari *host* atau *service* serta dapat membuat aplikasi whatsapp sebagai penerima notifikasi kondisi dari *host*.

### B. Studi Pustaka

Studi pustakan dilakukan untuk mempelajari teori – teori yang menunjang penelitian yang berasal dari jurnal – jurnal penelitian & buku – buku yang berkaitan dengan Nagios, *Bandwidth*, *Event Handler* dan notifikasi Whatsapp.

### C. Implementasi

Implementasi ialah tahapan menerapkan semua sistem yang telah dirancang, Implementasi dilakukan pada *server* ubuntu 18.04 Lts yang diinstal pada virtual mesin VMWare. Tahap instalasi membahas penginstalan nagios, konfigurasi *service check bandwidth*, dan konfigurasi whatsapp agar bisa mengirim notifikasi.

### D. Pengujian

Pengujian bertujuan untuk memastikan sistem agar dapat berjalan dengan semestinya. Pengujian dilakukan dengan metode *Black Box*,

dimana dilakukan pengujian terhadap beberapa fungsi yang terdapat pada sistem. Terdapat 4 komponen utama yang diuji diantaranya adalah *web interface*, notifikasi Telegram, *event handler*, dan *service bandwidth Handler*.

### E. Analisis & Pembahasan

Analisis yang dilakukan ialah hasil yang didapatkan setelah pengujian. Analisis berupa keberhasilan pengujian fungsionalitas sistem dan data dari beberapa pengujian yang dilakukan.

### F. Kesimpulan

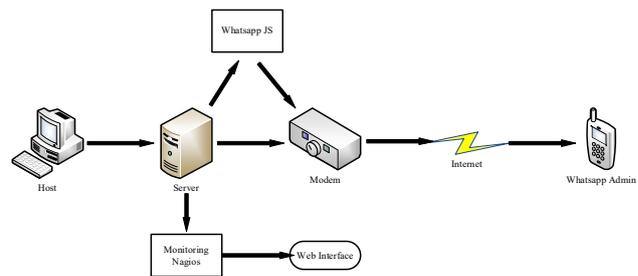
Kesimpulan merupakan tahap terakhir yang memberikan kesimpulan dari hasil analisis & pembahasan yang telah dilakukan, serta menjadi rumusan masalah yang telah didapat pada penelitian ini. Serta saran yang berhubungan dari analisis dan pembahasan, sehingga dapat memperbaiki kekurangan dan berguna untuk pengembangan lebih lanjut.

## III. PERANCANGAN

Pada bab ini menggambarkan rancangan arsitektur dari *system monitoring* yang akan dibangun. *System* yang dibangun adalah *network monitoring system* nagios dengan *event handler* dan notifikasi whatsapp dengan *system monitoring*.

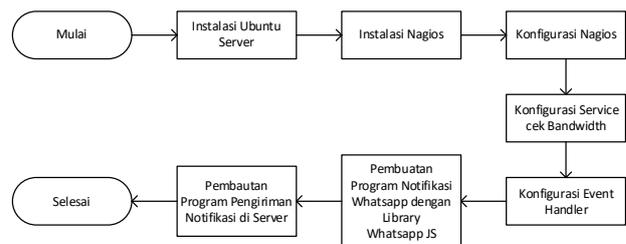
Sistem yang dibangun adalah implementasi *Network Monitoring System* Nagios dengan *Event Handler* dan Notifikasi Telegram *Messenger*. Dengan sistem *monitoring* jaringan ini, dapat memonitor objek *monitoring* di dalam jaringan secara *real-time*, dapat mendeteksi status (*up* atau *down*) serta dapat mendeteksi adanya *error* atau masalah yang terjadi pada jaringan. *Monitoring machine* akan mengirimkan notifikasi *alert* melalui Whatsapp *Messenger* apabila *host* maupun *service* yang berada pada jaringan dalam keadaan *down/error*. *Monitoring machine* pun dapat menangani *error* pada *service* secara otomatis dengan *Event Handler*.

Pada Gambar 2 menjelaskan bagaimana topologi perancangan sistem *monitoring* dan pengiriman notifikasi ke whatsapp *admin*.



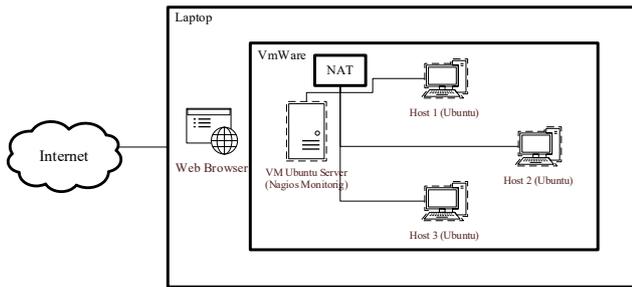
Gambar 2. Workflow Arsitektur Sistem

Pada Gambar 3 ialah alur kerja pembuatan sistem dengan tahapan instalasi Ubuntu, nagios, nagios *web interface* & nagios plugins serta membuat *user* akun nagios. Mengkonfigurasi nagios untuk mendefinisikan *hosts* dan *services*. Membuat program notifikasi whatsapp dengan *library* whatsapp web js dan membuat perintah pengiriman notifikasi di *server*.



Gambar 3. Alur Kerja Pembuatan Sistem

Topologi implementasi virtual yang dilakukan pada virtual mesin VMWare bisa dilihat pada Gambar 3, dimana pada gambar tersebut VMWare diinstal pada laptop. Dalam virtual mesin VMWare diinstal empat virtual mesin yang terdiri dari satu sebagai *server* ubuntu tempat penginstalan nagios dan tiga sebagai *host* untuk di-*monitoring*. Setiap mesin virtual yang diinstal pada VMWare mendapat ip address secara DHCP dari settingan jaringan NAT pada VMWare dan terkoneksi dengan internet. Menggunakan VMWare untuk menginstall virtual mesin sehingga membentuk suatu jaringan lokal pada laptop tersebut. Laptop bisa terhubung dengan semua virtual mesin yang terdapat pada VMWare.



Gambar 4. Topologi Implementasi Virtual

#### IV. IMPLEMENTASI

Pada bab ini membahas implementasi sistem yang akan dibangun. Tahap implementasi mencakup instalasi nagios, konfigurasi *service check bandwidth*, konfigurasi *event handler*, dan konfigurasi whatsapp untuk pengiriman notifikasi.

##### A. Instalasi Nagios

Nagios diinstal pada ubuntu server 18.04 Lts. Nagios membutuhkan paket-paket pendukung seperti apache, php, dan libc6 yang telah terinstal sebelum melakukan tahap instalasi nagios. Download file master nagios. Dipenelitian ini didownload dari [github.com](https://github.com). dan lakukan penginstalan nagios.

##### B. Konfigurasi Check Bandwidth

*Service check bandwidth* yang digunakan dipenelitian ini didapat dari website komunitas nagios yaitu [nagios-exchange.com](https://nagios-exchange.com), dengan memanfaatkan plugin tersebut agar bisa melakukan monitoring penggunaan bandwidth oleh host.

Sebelum melakukan konfigurasi *service*, download terlebih dahulu file *service*. Kemudian pindah kan file *service check bandwidth* di direktori `/usr/local/nagios/libexec/`. Setelah itu lakukan konfigurasi *command* seperti digambar 5 dan konfigurasi definisi *service* seperti pada gambar 6 agar bisa diakses dari *web interface*.

```
define command {
    command_name    check_bandwidth
    command_line    /usr/local/nagios/libexec/check_bandwidth -i=$ARG1$ -ct=10 -w 20 -c 10
}

```

Gambar 5. Define Command Check Bandwidth

```
define service {
    host_name        SERVER
    service_description    Bandwidth usage
    use              generic-service
    check_command    check_bandwidth! -i=ens33
    max_check_attempts    1
    check_interval    1
    retry_interval    1
    register          1
    event_handler     record_service_data
}

```

Gambar 6. Definisi Service Check Bandwidth

##### C. Konfigurasi Event Handler

*File event handler* yang digunakan masih dari bawaan nagios sendiri, dan dipenelitian ini dilakukan perombakan pada *event handler* bawaan dengan menambahkan beberapa fitur yang belum ada pada *event handler* bawaan.

Fitur yang ditambahkan pada *event handler* ialah seperti merekam pemanggilan *host* dan aksi terhadap *service* dan Pengubahan bentuk log *event handler* di *web interface* nagios. Pada Gambar 7 ialah konfigurasi untuk *event handler* merekam perubahan status terhadap *host* dan *service*

```
define command {
    command_name    record_service_data
    command_line    /usr/bin/printf "%b" "$LONGDATETIME$: $SERVICESTATE$ (attempt $SERVICEATTEMPT$)\n" >>/usr/local/nagios/var/status-$HOSTNAME$-$SERVICEDESC$.log }

define command {
    command_name    record_host_data
    command_line    /usr/bin/printf "%b" "$LONGDATETIME$: $HOSTSTATE$ (attempt $HOSTATTEMPT$)\n" >>/usr/local/nagios/var/status-$HOSTNAME$.log }

```

Gambar 7. Konfigurasi Command Event Handler

Pada file *switch.cfg* yang terletak di direktori `/usr/local/nagios/etc/objects/` tambahkan direktif *event\_handler* dengan nilai *record\_host\_data*. Tambah kan juga pada konfigurasi definisi *service* direktif *event\_handler* dengan nilai *record\_service\_data*. Penambahan ini bertujuan apabila terjadi perubahan status pada host atau *service* maka akan langsung di periksa oleh *event\_handler*.

```
define host {
    use                linux-server
    host_name          HOST1
    alias              delphi
    address            192.168.88.133
    event_handler      record_host_data }

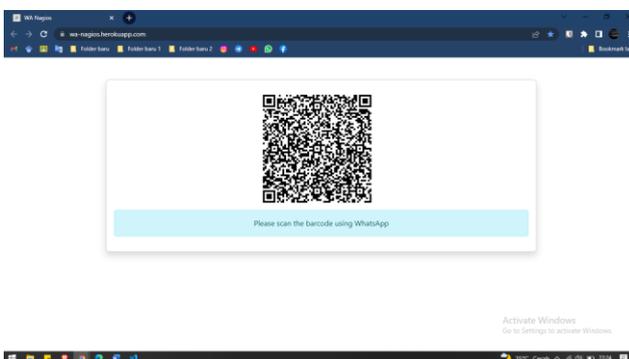
define service {
    use                generic-service
    host_name          HOST1
    service_description PING
    check_command      check_ping!100,10%!200,20%
    event_handler      record_service_data }
```

Gambar 8. Konfigurasi Definisi Event Handler

#### D. Konfigurasi Notifikasi Whatsapp

Notifikasi whatsapp dengan menggunakan Library Whatsapp JS dengan memunculkan qr code pada tampilan website dengan code yang sederhana. Dengan bantuan library whatsapp js memudahkan untuk memberikan notifikasi perubahan host pada nagios yang dihubungkan dengan socket IO. Selain itu untuk membuat library whatsapp js didukung dengan framework express js sebagai server yang menerima data dari nagios serta untuk menjalankan library whatsapp js pada hosting Heroku.

Express js meng-generate qr code untuk tampilan register whatsapp js dan mengecek apakah nomor whatsapp tersebut telah ter-register melalui qr code yang ada pada tampilan website sehingga memudahkan melakukan register terhadap notifikasi untuk whatsapp dari host nagios. Gambar 9 merupakan tampilan scan qr code untuk mengaktifkan whatsapp web yang telah dibuat menggunakan NodeJS dengan Library whatsappweb.js



Gambar 9. Web Interface QR Code Whatsapp JS

Agar server bisa terhubung dengan whatsapp js buat terlebih dahulu program pengiriman diserver dengan memakai import request agar bisa memanggil API pada

Whatsapp JS Agar status service ping down pada host bisa diketahui sistem dan dikirim pada admin maka dibutuhkan subprocess untuk meengkseskusi perintah pengecekan ping tersebut. Dalam program yang dibuat apa bila terdapat host dengan status service ping down atau terputus dari server maka program akan mengirim notifikasi kepada admin secara spam sampai host tersebut sudah terhubung kembali dengan server.

```
import subprocess
import requests

res2 =
    subprocess.getoutput('/usr/local/nagios/libexec/check_ping -H
    192.168.88.134 -w 3000.0,80% -c 5000.0,100% -p 5')

output = res2
ping_ok = output.split('-')
status = ping_ok[0].split(' ')

print(status[1])
if status[1] == 'OK':
    requests.get("-")
else:
    requests.get("https://wa-
    nagios.herokuapp.com/api/send?no=6281267186774&pesan="+ 'HOST1 =
    ' + 'PING DOWN' + ' ==> ' + 'TOLONG SEGERA DIPERIKSA')
```

Gambar 10. Code Notifikasi Host di Server

Dalam pengiriman notifikasi penggunaan bandwidth juga sama menggunakan request untuk pemanggilan link api dari whatsapp js untuk mengirim notifikasi dan subprocess untuk mengeksekusi perintah pengecekan penggunaan bandwidth.

```
import subprocess
import requests

res1 = subprocess.getoutput('/usr/local/nagios/libexec/check_bandwidth
-i=ens33 -gt=10 -w 20 -c 10')

requests.get("https://wa-
nagios.herokuapp.com/api/send?no=6281267186774&pesan="+ 'HOST1 ' +
' ==> ' + res1)
```

Gambar 11. Code Notifikasi Bandwidth di Server

## V. PENGUJIAN

Pengujian dilakukan untuk mengetahui apakah system yang dibuat berjalan sesuai fungsinya. Pengujian memiliki beberapa tahapan yaitu:

### A. Pengujian Fungsionalitas Web Interface

Pengujian terhadap fungsionalitas interface ada dua tahapan yaitu autentikasi web interface dan pengujian host. Pengujian autentikasi web

*interface* dilakukan dengan cara mengakses web *interface* nagios. Jika *input* data autentikasi benar, maka akan masuk kedalam web *interface* nagios.

**Tabel 1.** Hasil Pengujian Autentikasi Web *Interface* "Hasil uji benar"

Hasil Uji Benar			
<i>Input</i>	Hasil yang diharapkan	<i>Output</i>	Hasil
<b>Mengisi benar data login: Username: abdi Password: 123</b>	Data <i>login username</i> dan <i>password valid</i> dapat masuk kedalam web <i>interface</i> nagios	<i>Input</i> data berhasil dilakukan dan masuk kedalam web <i>interface</i>	Berhasil

Tabel 4.2 Hasil Pengujian Autentikasi Web *Interface* "Hasil uji benar"

Hasil Uji Salah			
<i>Input</i>	Hasil yang diharapkan	<i>Output</i>	Hasil
<b>Mengosongkan Username Password</b>	Tidak dapat <i>Login</i> dan kembali pada <i>window</i> autentikasi	Tidak dapat login dan diarahkan kembali pada <i>window</i> autentikasi	Berhasil

Pengujian *host* dilakukan dengan menonaktifkan *host* sehingga tidak terkoneksi dengan jaringan *server* untuk melihat respon perubahan *host* pada *monitoring interface* nagios.

**Tabel 3.** Pengujian penonaktifan Host

Peng ujian	Nama	Waktu Penonaktifan	Perubahan pada Web Nagios	Interval
1	HOST1	16:33:10	16:34:49	00:01:39
2	HOST1	16:41:35	16:42:24	00:00:49
3	HOST2	16:45:09	16:47:29	00:02:20
4	HOST2	17:11:51	17:13:01	00:01:10
5	HOST3	17:30:31	17:33:09	00:02:38
6	HOST3	17:39:19	17:40:53	00:01:34

**Tabel 4.** Pengujian pengaktifan Host

Peng ujian	Nama	Waktu Mengaktifan	Perubahan pada Web Nagios	Interval
1	HOST1	16:36:03	16:36:38	00:00:35
2	HOST1	16:42:44	16:43:43	00:00:59
3	HOST2	16:55:23	16:56:27	00:01:04
4	HOST2	17:20:31	17:20:57	00:00:26
5	HOST3	17:37:01	17:38:06	00:01:05
6	HOST3	17:46:45	17:47:24	00:00:39

### B. Pengujian Service Check Bandwidth

Pengujian *service check bandwidth* dilakukan dengan cara mengecek *host* saat terkoneksi internet dan saat tidak terkoneksi internet. *Service check bandwidth* tidak bisa *down* walaupun nilainya nol, karena *service check bandwidth* memiliki fungsi hanya untuk mengecek koneksi jaringan pada *host* saja.

**Tabel 5.** Pengujian *Service check bandwidth*

Pengujian	Nilai TX	Nilai RX
144p	138kb	14kb
480p	211kb	145kb
1080p	862kb	233kb

### C. Pengujian Event Handler

Pengujian *event handler* dilakukan dengan cara menonaktifkan *host* untuk melihat respon aksi dari *event handler* terhadap *host* dan *service* yang bermasalah.

**Tabel 6.** Pengujian Event Handler

Peng ujian	Nama	Waktu Penonaktifan	Event Handler	Interval
1	HOST1	20:13:03	20:13:58	00:00:55
2	HOST1	20:29:19	20:32:09	00:02:50
3	HOST2	20:41:55	20:42:47	00:00:52
4	HOST2	20:59:03	20:59:51	00:00:48
5	HOST3	21:17:29	21:19:02	00:01:33
6	HOST3	21:36:30	21:37:36	00:01:06

**D. Pengujian Pengiriman Notifikasi Whatsapp**

Tahap pengujian dilakukan untuk menguji fungsionalitas dari *system* apakah berjalan dan bisa mengirimkan notifikasi kepada *admin* melalui whatsapp ketika terdapat *host* yang mengalami kondisi *down*.

Pengujian pengiriman notifikasi *host* dilakukan dengan dua tahap, pertama pengujian saat *host down* dan kedua pengujian saat *host* telah terhubung kembali dengan *server*. Hasil dari pengujian mendapat kan data pada Tabel 7 dan Tabel 8.

**Tabel 7.** Pengujian Notifikasi Whatsapp “HOST DOWN”

Peng ujian	Nama	Waktu Penon aktifan	Pengiriman Notifikasi	Interval
1	HOST1	08:09:45	08:09:57	00:00:12
2	HOST1	08:21:10	08:22:03	00:00:53
3	HOST2	08:33:31	08:33:59	00:00:28
4	HOST2	08:49:47	08:52:13	00:02:26
5	HOST3	09:15:23	09:16:46	00:01:23
6	HOST3	09:28:50	09:30:22	00:01:32

**Tabel 8.** Pengujian Notifikasi Whatsapp “HOST UP”

Peng ujian	Nama	Waktu Pengaktifan	Terakhir Pengiriman Notifikasi	Interval
1	HOST1	08:14:31	08:14:18	00:00:13
2	HOST1	08:27:12	08:27:01	00:00:11
3	HOST2	08:39:56	08:39:36	00:00:20
4	HOST2	08:55:34	08:55:21	00:00:13
5	HOST3	09:23:01	09:22:49	00:00:12
6	HOST3	09:39:04	09:38:56	00:00:08

Pengujian pengiriman notifikasi penggunaan *bandwidth* oleh *host* dilakukan kepada satu *host* dengan jangka pengiriman notifikasi yang berbeda setiap pengujian. Hasil dari pengujian pengiriman notifikasi penggunaan *bandwidth* oleh *host* mendapatkan data pada table 9.

**Tabel 9.** Pengujian Notifikasi Whatsapp Cek *Bandwidth*

Peng ujian	Jangka Waktu	Pengiriman Notifikasi Pertama	Pengiriman Notifikasi Kedua	Interval
1	5 Menit	09:58:03	10:03:09	00:05:06

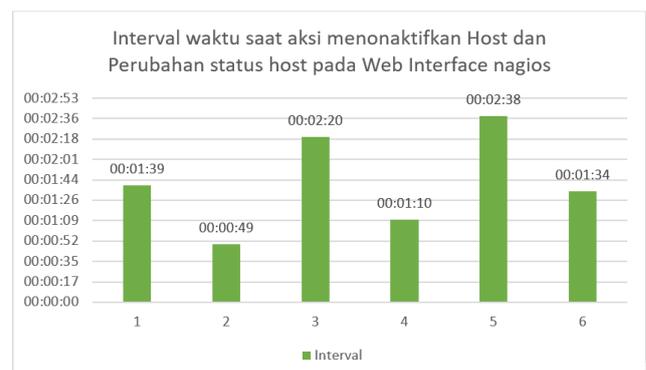
2	10 Menit	10:05:32	10:15:37	00:10:05
3	20 Menit	10:18:20	10:38:31	00:20:11
4	30 Menit	10:45:47	11:16:02	00:30:15
5	1 Jam	11:20:16	12:20:29	01:00:13
6	2 Jam	12:25:50	14:26:13	02:00:23

**VI. HASIL ANALISIS DATA**

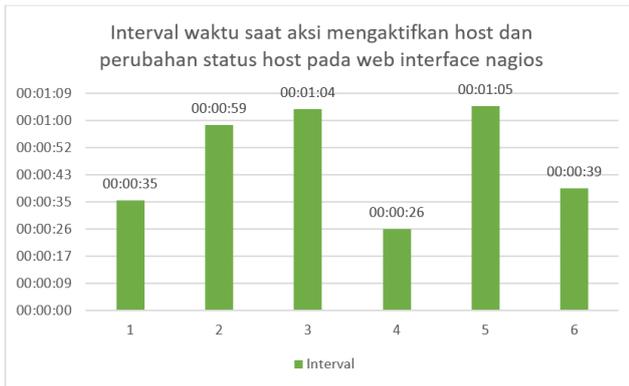
**A. Analisis Hasil Pengujian Fungsionalitas Web Interface Nagios**

Berdasarkan hasil pengujian autentikasi web *interface* nagios didapatkan hasil yaitu autentikasi berhasil diuji. Hal ini didapat Ketika menginputkan *username* dan *password* yang benar, sedangkan ketika salah *input user name* dan *password*, *user* tidak dapat masuk ke web *interface* nagios dan dialihkan kembali kehalaman autentikasi.

Dari data pengujian *host* pada Tabel 3 dan Tabel 4 didapatkan data perbandingan durasi Nagios dalam mendeteksi perubahan status *hosts* dalam kondisi “*DOWN*” dan “*UP*” pada Gambar 11 dan Gambar 12. Durasi tercepat nagios dalam mendeteksi perubahan status dalam kondisi *down* ialah 48 detik dan terlama ialah pada waktu 2 menit 38 detik sedangkan saat kondisi up nagios lebih cepat dalam mendeteksi perubahan status yaitu terlama ada pengujian 5 dengan waktu 1 menit 5 detik dan tercepat pada waktu 26 detik. Dapat disimpulkan nagios bisa mendeteksi perubahan terjadi selama selang waktu kurang dari 3 menit.



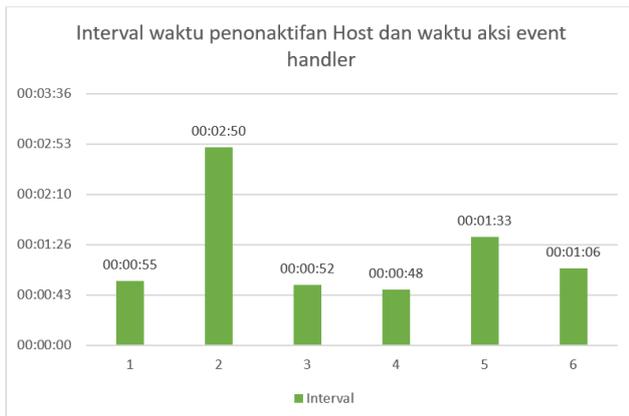
**Gambar 11.** Grafik Interval *Host Down Interface* Nagios



**Gambar 12.** Grafik Interval *Host Up Interface* Nagios

**B. Analisis Hasil Pengujian Event Handler**

Dari keenam pengujian yang telah dilakukan dimana waktu interval paling cepat ada pada pengujian empat dengan waktu 48 detik dan paling lama ada pada pengujian kedua dengan waktu 2 menit 50 detik. Dari keenam pengujian didapat lah rata – rata *interval* waktu saat *host down* dengan waktu aksi *event handler* selama 1 menit 21 detik.



**Gambar 13.** Interval *Host Down* dan aksi Event Handler

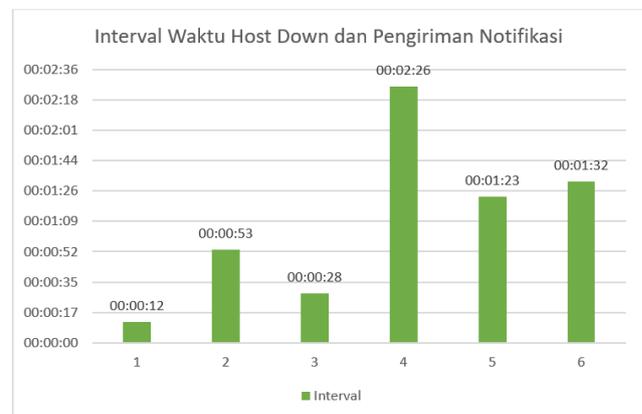
**C. Analisis Hasil Pengujian Service Check Bandwidth**

Dari hasil pengujian yang telah dilakukan *service bandwidth* berhasil menampilkan status *bandwidth* pada *host*. Dimana *service check bandwidth* menampilkan status diinterface nagios dalam bentuk dalam nilai TX dan nilai RX. Dari pengujian bisa disimpulkan bahwa saat *host* tidak terkoneksi internet maka nilai status *service bandwidth* akan nol dan jika terhubung nilai nya akan berubah sesuai dengan penggunaan oleh *host* tersebut.

Dari data pengujian pada tabel 5 didapatkan juga bahwa nagios berhasil menampilkan status *bandwidth* sesuai pemakaian *host* nya. Pengujian yang dilakukan dengan membuka video youtube dengan berbeda kualitas memperlihatkan semakin tinggi kualitas video yang dibuka semakin besar juga penggunaan *bandwidth*-nya, dan dari pengujian tersebut nagios berhasil menampilkan status *bandwidth* yang berbeda pada setiap pengujian dengan membuka video dengan berbeda kualitas.

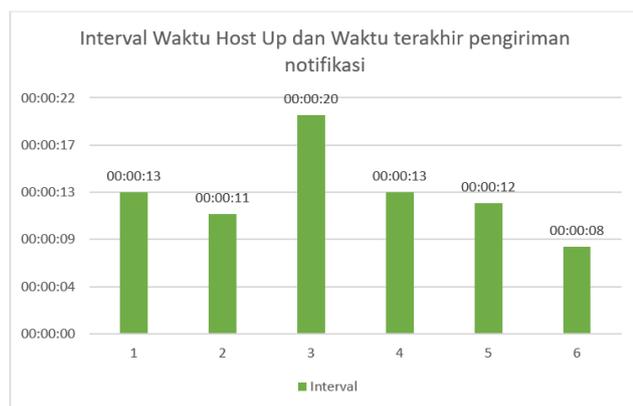
**D. Analisis Hasil Pengujian Pengiriman Notifikasi Whatsapp**

Pada grafik pada gambar 14 terlihat perbedaan interval penerimaan pesan dari keenam pengujian tersebut. Dimana interval penerimaan pesan tercepat ada pada pengujian satu yang selama 12 detik dan interval terlama terdapat ada pada pengujian empat yaitu selama 2 menit 26 detik. Dari keenam pengujian didapat rata – rata pengiriman notifikasi selama 1 menit 9 detik.



**Gambar 14.** Interval Waktu *Host Down* dan Pengiriman Notifikasi

Dari data grafik pada gambar 15 dapat dilihat perbedaan interval keenam pengujian yang telah dilakukan. Interval terpendek antara waktu pengaktifan *host* dengan penerimaan notifikasi terakhir ada pada pengujian enam dengan durasi waktu 8 detik dan terpanjang ada pengujian tiga dengan durasi waktu 20 detik. Dari tersebut didapatkan juga rata – rata interval yaitu selama 13 detik



**Gambar 15.** Interval Waktu *Host Up* dan Waktu terakhir pengiriman notifikasi

Pengujian terhadap fungsional sistem pengiriman notifikasi status *bandwidth host* yang telah dilakukan berhasil dengan pengujian terhadap skala waktu pengiriman yang berbeda. Dari data pengujian yang didapat pada Tabel 9 menunjukkan keterlambatan pengiriman dari waktu yang ditentukan kurang dari 20 detik

## VII. KESIMPULAN

Berdasarkan penelitian ini dapat disimpulkan bahwa sistem *monitoring* nagios dapat bekerja dengan baik dalam melakukan *monitoring* perangkat dan *services* yang ada dalam jaringan. Aplikasi *monitoring* nagios yang bersifat *open source* dapat memudahkan dalam penambahan *plugin* dari luar bawaan nagios itu sendiri. Dengan memanfaatkan whatsapp sebagai notifikasi *alert* sangat membantu dalam melakukan *monitoring* perangkat yang ada, karena selain dapat memantau keadaan perangkat tanpa mengecek *interface monitoring* nagios, *admin* akan diberi informasi jika ada perangkat yang mengalami gangguan atau dalam keadaan *down* melalui whatsapp *messenger* secara *real time*.

## VIII. SARAN

Saran untuk penelitian selanjutnya berdasarkan kesimpulan terhadap hasil penelitian ini sebagai berikut

1. Menambahkan target *monitoring* seperti printer, ip camera, dan lain – lain.
2. Menambahkan *notification alert* lain, seperti *audio notification alarm*.

3. Menambahkan *plugin* nagios yang lain seperti grafik *service*, *nagiosql*, dan lain – lain.
4. Melakukan perbandingan pengujian sistem *monitoring* nagios dengan sistem *monitoring* lain dalam penggunaan whatsapp sebagai *notification alert*.

## REFERENSI

- [1] Chalal, D., Kharb, L., Choudhary, D., 2019, *Performance Analytics of Network Monitoring Tools*, *International Journal of Innovative Technology and Exploring Engineering*, vol.8.
- [2] Christanto, F., W., Suprayogi, M., S., 2018, *Enhancement Network Monitoring System Functionality by Implementing an Android-based Notification System to Monitor Virtual Servers on Cloud Computing Network*, *International Journal of Applied IT*, vol. 02, no. 01.
- [3] Fahreza, Rifqi, M., 2020, *Nagios Core Optimization By Utilizing Telegram as Notification of Disturbance*, *Journal of Applied Science, Engineering, Technology, and Education* vol.2, no.2, pp.121–135.
- [4] Fernando, N., Humaira, Asri, E., 2020, *Monitoring Jaringan dan Notifikasi Dengan Telegram pada Dinas Komunikasi dan Informatika Kota Padang*, *Jurnal Ilmiah Teknologi Sistem Informasi*, vol.1, no.4, pp.121-126.
- [5] Habibullah, T., Arnaldy, D., 2016, *Implementasi Network Monitoring System Nagios dengan Event Handler dan Notifikasi Telegram Messenger*, *Jurnal Multinetics*, vol.2, no.1
- [6] Hendri, Masagus. (2012). *Pengenalan Desktop Ubuntu dan Libreoffice*. Kementerian Kehutanan Republik Indonesia. Bukudigital.net. Google Books Program
- [7] Rahman, D., Amnur, H., dan Rahmayuni, I., 2020, *Monitoring Server dengan*

- Prometheus dan Grafana serta Notifikasi Telegram, Jurnal Ilmiah Teknologi Sistem Informasi, vol.1, no.4, pp.133-138.
- [8] Manai, Syamsudin. (2013). 60 Menit Belajar *Monitoring Jaringan (Cacti)*. Bukudigital.net. Google Books Program
- [9] Oktivasari, P., Habibullah, T., 2017, Kajian *Network Monitoring System* Menggunakan Nagios Dengan Whatsapp Sebagai Notifikasi *Alert*, Jurnal Komunikasi, Media Dan Informatika, vol.6, no.3.
- [10] Pratam, R., A., (2019) Kuliah Melalui Whatsapp (KULWAP), Jurnal Pendidikan Matematika, vol.2 no.1
- [11] Rifai, B., Nuryandi, N., Ripai, A., 2019, Implementasi Telegram *Notification Alert* Pada *Network Monitoring System* Dengan Nagios, Jurnal Mantik Penusa, vol.3, no.3, pp. 54-60.
- [12] Syafrizal, Melwin. (2005) Pengantar Jaringan Komputer. Yogyakarta: Andi Offset Bukudigital.net.Google Books Program
- [13] Yanto, J., Ruswanda, M., 2017, Implementasi Sistem *Monitoring Server* Menggunakan Nagios, vol.13, no.1.