

KAJIAN NETWORK MONITORING SYSTEM MENGGUNAKAN NAGIOS DENGAN WHATSAPP SEBAGAI NOTIFIKASI ALERT

Prihatin Oktivasari¹, Tubagus Habibullah²

^{1,2}Program Studi Teknik Informatika, Konsentrasi Teknik Komputer dan Jaringan Politeknik Negeri Jakarta
Kukusan, Beji, Kota Depok, Jawa Barat 16425

Email : Prihatin.oktivasari@tik.pnj.ac.id

Diterima : 14 Oktober 2017 | Direvisi : 21 Oktober 2017 | Disetujui : 29 Oktober 2017

Abstrak

Salah satu tujuan dibuatnya Network Monitoring System adalah agar network administrator dapat selalu memantau seluruh kegiatan perangkat penting dari jaringan komputer seperti router, switch, komputer client, dan lain-lain. Nagios merupakan salah satu aplikasi open source untuk network monitoring dan memiliki banyak plugins yang dapat digunakan untuk memaksimalkan proses monitoring perangkat jaringan yang ada. Beberapa fitur yang dimiliki Nagios seperti monitoring, alert, report, memantau resource CPU load, memory, status up/down, up time, data traffic, bandwidth dari perangkat. 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 dirancang menggunakan WhatsApp messenger. Pengujian sistem dilakukan dengan menguji fungsionalitas kinerja sistem Nagios. Dari hasil pengujian didapatkan bahwa Nagios dapat mendeteksi perubahan status dengan selang waktu interval 5 menit untuk hosts dan 10 menit untuk services, notifikasi alert dapat terkirim kepada Group Telegram dengan selang waktu rata-rata 5-10 menit setelah Nagios mendeteksi adanya perubahan status down/recovery, dan Event Handler dapat melakukan restart terhadap services yang gagal fungsi dengan rata-rata waktu aksi 2 menit. Hal ini membuktikan bahwa Nagios dapat memonitoring perangkat dan services pada jaringan dengan baik sesuai dengan konfigurasi yang telah dilakukan.

Kata kunci: Network Monitoring System, Nagios, WhatsApp messenger

Study of Network Monitoring System Using Nagios with Whatsapp as Notification Alert

Abstract

One of the goals of Network Monitoring System is that network administrators can always monitor all important device activities from computer networks such as routers, switches, client computers, and others. Nagios is one of the open source applications for network monitoring and has many plugins that can be used to maximize the monitoring process of existing network devices. Some features of Nagios such as monitoring, alert, report, monitor CPU load resources, memory, up / down status, up time, data traffic, bandwidth of the device. Nagios itself can be configured as needed. One important feature in Network Monitoring System is alert notification, is notification system to network administrator contact if system detects any problem on host or service. Notifications are designed using WhatsApp messenger. System testing is done by testing the functionality of Nagios system performance. From the test results obtained that Nagios can detect status changes with intervals of 5 minutes for hosts and 10 minutes for services, notification alerts can be sent to the Telegram Group with an average interval of 5-10 minutes after Nagios detect any change in down / recovery status, and Event Handler can restart the services that fail the function with an average 2-minute action time. This proves that Nagios can monitor devices and services on the network well in accordance with the configuration that has been done.

Keywords: Network Monitoring System, Nagios, WhatsApp messenger

PENDAHULUAN

Jaringan komputer sekarang ini sudah seperti sebuah kebutuhan, sehingga diperlukan adanya pemeliharaan dan kontrol yang teratur untuk menjamin agar jaringan dapat berjalan sebagaimana mestinya. Salah satu tujuan dibuatnya *Network Monitoring System* adalah agar *network administrator* dapat selalu memantau seluruh kegiatan perangkat penting dari jaringan komputer seperti router, switch, komputer *client*, dan lain-lain. Nagios merupakan salah satu aplikasi *open source* untuk *network monitoring* dan memiliki banyak *plugins* yang dapat digunakan untuk memaksimalkan proses *monitoring* perangkat jaringan yang ada. Beberapa fitur yang dimiliki Nagios seperti *monitoring*, *alert*, *report*, memantau *resource* (*CPU load*, *memory*, *status up/down*, *up time*, *data traffic*, *bandwidth*) dari perangkat. 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, WhatsApp *messenger*, dan lain-lain.

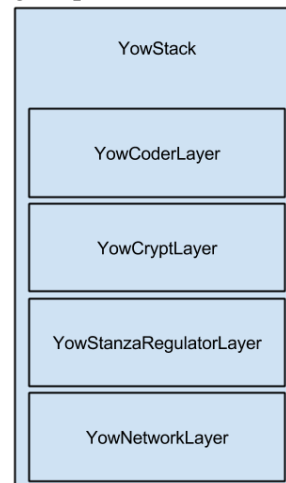
Nagios Network Monitoring System

Monitoring sistem di nagios dibedakan menjadi dua kategori, yaitu *hosts* dan *service*. *Hosts* mewakili sebuah perangkat fisik atau virtual di dalam jaringan seperti server, router, workstation, *printer*, dll. *Services* adalah sebuah fungsi partikular, sebagai contoh, sebuah Secure Shell (SSH) *server* (*sshd process* pada perangkat) bisa didefinisikan sebagai *service* yang akan dimonitor. Setiap *service* pasti berhubungan dengan sebuah *host* yang menjalankannya. Beberapa fitur-fitur yang tersedia pada Nagios, antara lain: memonitoring *network services*, yaitu SMTP, POP3, HTTP, NNTP, PING; memonitoring *host resources* seperti *processor load*, *disk usage*; desain *plugins* yang sederhana, yang memungkinkan pengguna untuk mengembangkan sendiri pemeriksaan terhadap servisnya; *service checks* yang paralel; *web interface* yang fakultatif untuk melihat status *network*, urutan masalah dan notifikasi, log file, dan sebagainya; kemampuan untuk

mendefinisikan kejadian yang ditangani selama servis atau *host* berlangsung untuk mempermudah pemecahan masalah, perputaran *log file* yang otomatis, dan notifikasi kontak ketika servis atau *host* terjadi masalah dan mendapat penanganan via email, *pager*, sms, whatsapp, atau metode yang didefinisikan *user*.

Yowsup

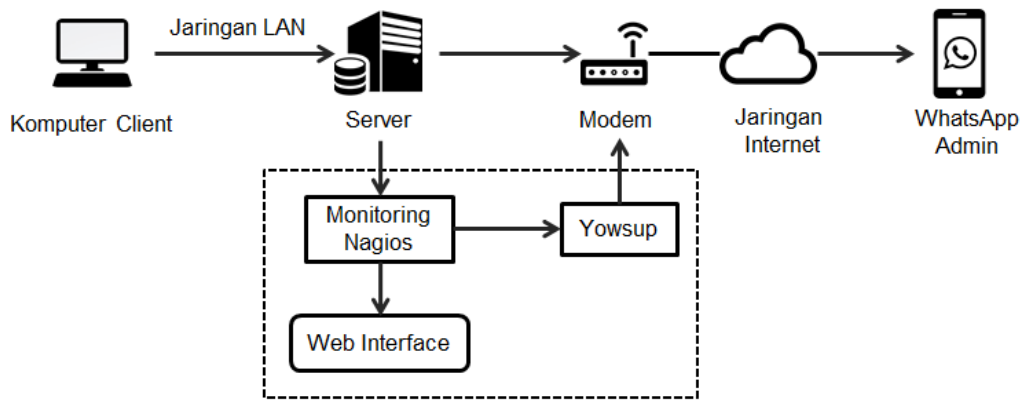
Yowsup merupakan sebuah platform *opensource* yang dikembangkan dengan *python library* untuk membuat aplikasi yang memakai servis WhatsApp. *Yowsup* telah dipakai untuk membuat klien WhatsApp unofficial Nokia N9 melalui proyek Wazapp yang dimanfaatkan lebih dari 200 ribu pengguna. (Tarek, 2015). Sebuah aplikasi *Yowsup* yang sedang berjalan adalah tumpukan dari beberapa layer. Sebuah layer menjelaskan tentang mekanisme aliran data yang berpindah.



Gambar 1. YowStack Layer

METODOLOGI PENELITIAN

Dengan sistem monitoring jaringan ini maka komputer server dapat mendeteksi status (*up* atau *down*) pada komputer *client*. Komputer server akan mengirimkan pesan notifikasi ke WhatsApp admin secara otomatis apabila komputer *client* atau *servicenya* dalam keadaan *down*. Semua perubahan dan notifikasi akan tersimpan pada *reports* di web interface.



Gambar 2. *Workflow* Arsitektur Sistem



Gambar 3. Alur Pembuatan Sistem

Metode yang digunakan menggunakan tahapan berikut: instalasi Ubuntu, nagios, *nagios web interface* & *nagios plugins* serta membuat *user* akun nagios; mengkonfigurasi nagios untuk mendefinisikan *hosts* dan *services*; menginstalasi dan konfigurasi Yowsup; registrasi nomor handphone, dan tahapan terakhir adalah mengkonfigurasi Yowsup di Nagios dengan cara meng-copy direktori yowsup ke direktori Nagios, mendefinisikan notifikasi *hosts* & *services*, serta mengedit kontak Nagios.

PEMBAHASAN

Konfigurasi service yang akan *dimonitoring* adalah dengan menambahkan service yang akan *dimonitoring*. Berikut service yang akan *dimonitoring*:
Ping

```

# Define a service to "ping" the local machine

define service{
    use                local-service           ; Name of service
    host_name          localhost, tubagush-pc, fadhli, asus-pc
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
}
  
```

Root Partition

```

# Define a service to check the disk space of the root partition
# on the local machine. Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use                local-service           ; Name of service
    host_name          localhost, tubagush-pc, fadhli, asus-pc
    service_description Root Partition
    check_command      check_local_disk!20%!10%!/
}
  
```

Current Users

```
# Define a service to check the number of currently logged in
# users on the local machine. Warning if > 20 users, critical
# if > 50 users.

define service{
    use                local-service        ; Name of service
    host_name          localhost, tubagush-pc, asus-pc, fadhli
    service_description Current Users
    check_command      check_local_users!20!50
}
```

Total Process

```
# Define a service to check the number of currently running procs
# on the local machine. Warning if > 250 processes, critical if
# > 400 users.

define service{
    use                local-service        ; Name of service
    host_name          localhost, tubagush-pc, asus-pc, fadhli
    service_description Total Processes
    check_command      check_local_procs!250!400!RSZDT
}
```

Current Load

```
# Define a service to check the load on the local machine.

define service{
    use                local-service        ; Name of service $
    host_name          localhost, tubagush-pc, asus-pc, fadhli
    service_description Current Load
    check_command      check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}
```

Swap Usage

```
# Define a service to check the swap usage the local machine.
# Critical if less than 10% of swap is free, warning if less than 20% is free

define service{
    use                local-service        ; Name of service
    host_name          localhost, tubagush-pc, asus-pc, fadhli
    service_description Swap Usage
    check_command      check_local_swap!20!10
}
```

SSH

```
# Define a service to check SSH on the local machine.
# Disable notifications for this service by default, as not all users may have !

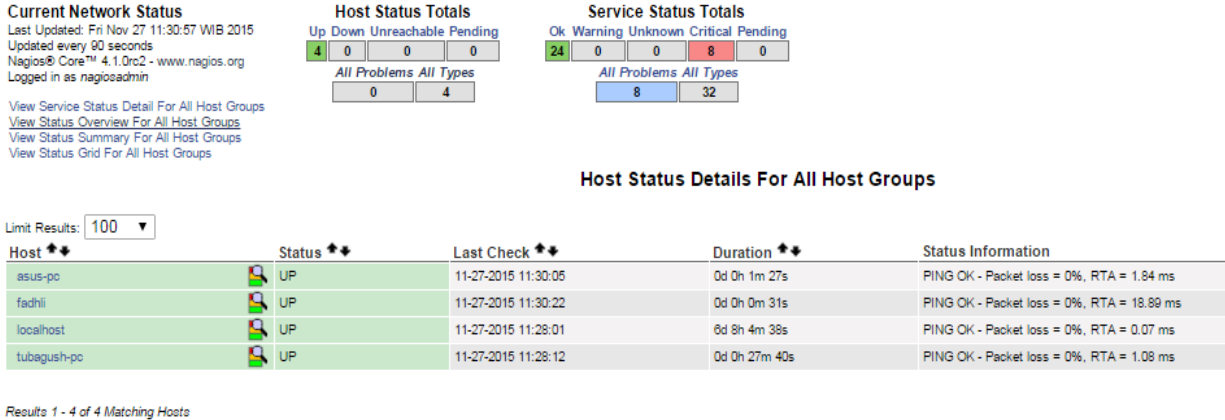
define service{
    use                local-service        ; Name of service
    host_name          localhost, tubagush-pc, asus-pc, fadhli
    service_description SSH
    check_command      check_ssh
    notifications_enabled 0
}
```

HTTP

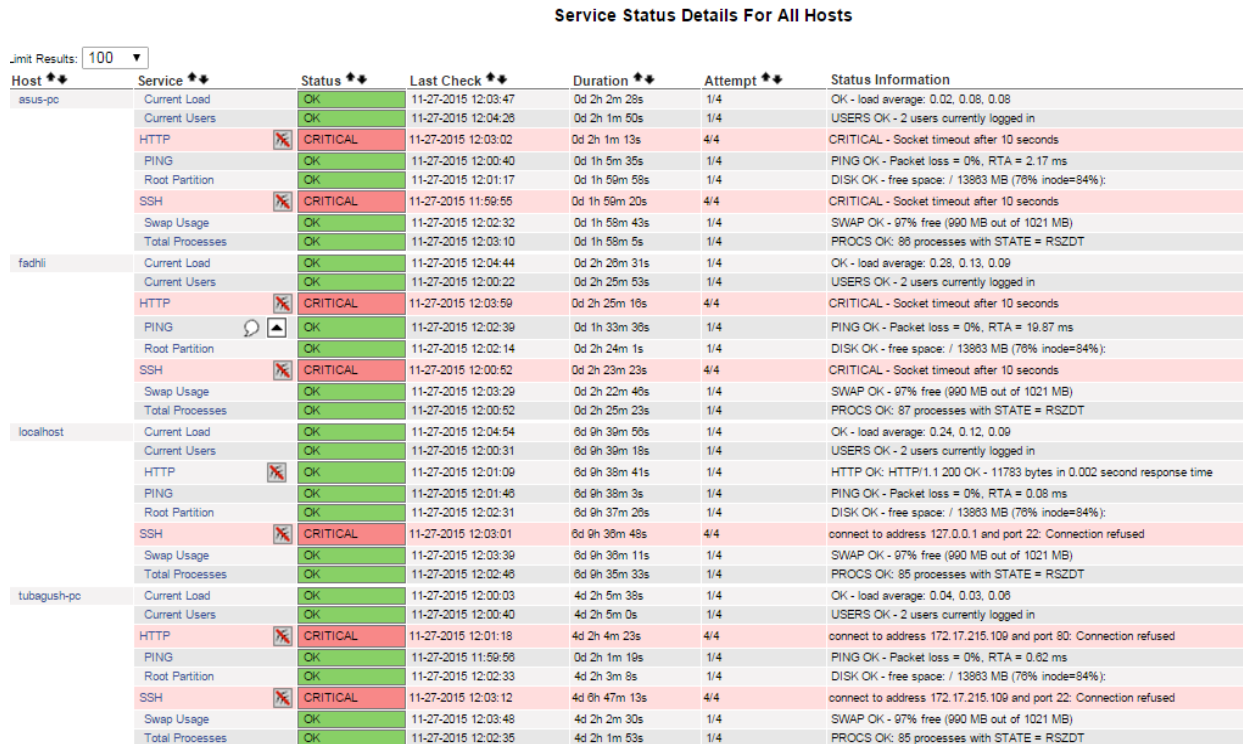
```
# Define a service to check HTTP on the local machine.
# Disable notifications for this service by default, as not all users may have !

define service{
    use                local-service        ; Name of service
    host_name          localhost, tubagush-pc, fadhli, asus-pc
    service_description HTTP
    check_command      check_http
    notifications_enabled 0
}
```

Untuk memastikan dan mengecek apakah target *monitoring* sudah ditambahkan adalah pada bagian *Hosts* di panel kiri pada *Current Status*, semua host berada dalam status *up* terlihat pada gambar 4 dan mengecek *Services*, pada bagian *Services* di panel kiri *Current Status* tampak pada gambar 5. Tampilan *service status all host* yang menguraikan keterangan dari service yang berjalan pada target *monitoring*.



Gambar 4. Tampilan *Current Status Hosts*



Gambar 5. Tampilan *Service Status* semua *Host*

Sebelum mengkonfigurasi di dalam nagios, melakukan instalasi yowsup terlebih dahulu yang akan memberi peringatan/notifikasi menggunakan WhatsApp *Service*. Tahapan penting dalam instalasi yowsup: dalam direktori yowsup dan beri *script permissions* pada **setup.py**, dengan *command*:

```
root@ubuntu:/home/ubuntu# cd yowsup
root@ubuntu:/home/ubuntu/yowsup# chmod +x setup.py
```

dan melakukan konfigurasi whatsapp dengan merubah *permissions* file *yowsup-cli*, dengan *command*:

```
root@ubuntu:/home/ubuntu/yowsup# chmod +x yowsup-cli
```

Yowsup-cli adalah *command line interface* dalam *library* Yowsup. Yowsup-cli menyediakan beberapa *options/commands* diantaranya *registration*, dan *demos* seperti *command line client*. Kemudian mendaftarkan nomor *handphone*, dengan menggunakan *command* *yowsup-cli* dengan *options* *registration*.

```
root@ubuntu:/home/ubuntu/yowsup# ./yowsup-cli registration --requestcode sms --p
hone 6285881782767 --cc 62 --mcc 510 --mnc 01
INFO:yowsup.common.http.warequest:{"status":"sent","length":6,"method":"sms","re
try_after":1805}

status: sent
retry_after: 1805
length: 6
method: sms
```

Beberapa bagian yang harus dilakukan dari Whatsapp *registration options* adalah: a. *--mcc* yaitu Mobile Country Code, digunakan dalam kombinasi dengan kode jaringan mobile (MNC/Mobile Network Code) untuk secara unik mengidentifikasi sebuah operator jaringan seluler dengan menggunakan GSM (Global System for Mobile), UMTS (Universal Mobile Telecommunications System), dan LTE (Long Term Evaluations); b. *--mnc*, yaitu Mobile Network Code, adalah dua digit identifikasi yang digunakan untuk mengidentifikasi sebuah *mobile network*; c. *--p*, yaitu *phone*, nomor telepon lengkap termasuk *country code* nya; dan *--cc*, yaitu *Country Code*, kode geografi yang merepresentasikan negara dan area nya, digunakan untuk *data processing* dan komunikasi. Setelah mendapatkan pesan dari *provider* yang berisikan kode registrasi WhatsApp, menyelesaikan proses registrasi dengan mengetikkan *command* berikut:

```
root@ubuntu:/home/ubuntu/yowsup# ./yowsup-cli registration --register 686-508 --
phone 6285881782767 --cc 62
```

Selanjutnya adalah melakukan konfigurasi yowsup, dengan membuat file *config*, menggunakan *command* **nano yowsup-cli.conf**. meng-konfigurasi di nagios dengan meng-copy file yowsup ke direktori */usr/local/nagios/libexec/*, pertama buat direktori **yowsup** terlebih dahulu ke dalam */usr/local/nagios/libexec/* dengan *command*:

```
root@ubuntu:/usr/local/nagios/libexec# mkdir yowsup
```

Dalam pengetesan *alert* notifikasi dengan *disconnect* salah satu host, dan setelah beberapa saat maka akan ada pesan notifikasi di WhatsApp *Messenger* dari nomor *handphone* yang telah didefinisikan di Nagios. Notifikasi akan dikirimkan ketika Nagios mendeteksi adanya host atau service yang mengalami masalah. Selain itu Nagios juga menampilkan interface di web yang fakultatif untuk melihat status network, urutan masalah dan notifikasi, dan log file.



Gambar 6. Tampilan Web Interface Nagios



Gambar 7. Tampilan Pesan Alert Notifikasi di WhatsApp

Log file dari notifikasi dapat dilihat di *Panel Reports*, bagian *Notifications*, terlihat pada gambar 7. Pengujian dari Implementasi *Network Monitoring System* dengan *Event Handler* dan Notifikasi *Telegram Messenger* 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 *monitoring agent* (NSClient++, NRPE, dan SNMP), *web interface*, notifikasi *Telegram*, dan *Event Handler*.

Pengujian monitoring agent terhadap fungsionalitas plugins Nagios

- a. Pengujian NSClient++ terhadap *plugin check_nt*
 Pengujian NSClient++ dilakukan dengan menguji *plugin check_nt* agar dapat mengambil informasi *service/resources* dari *host* Windows yang dimonitor. Setelah dijalankan *plugins check_nt* maka akan tampil output dari informasi *service/resource*. *Output* tersebut akan sama dengan apa yang ditampilkan pada *web interface* Nagios.
- b. Pengujian NRPE terhadap *plugin check_nrpe*
 Pengujian NRPE dilakukan dengan menguji *plugin check_nrpe* agar dapat mengambil informasi *service/resources* dari *host* Linux/Ubuntu yang dimonitor. *Output* tersebut akan sama dengan apa yang ditampilkan pada *web interface* Nagios.
- c. Pengujian SNMP terhadap *plugin check_snmp*
 Pengujian SNMP dilakukan dengan menguji *plugin check_snmp* pada *monitoring* router untuk mengambil informasi *service/resources* yang diinginkan. *Output* tersebut akan sama dengan apa yang ditampilkan pada *web interface* Nagios.

Tabel 1. Tabel Hasil Pengujian *Monitoring Agent*

Pengujian	Plugins	Service	Output	Kesimpulan
NSClient++	Check_nt	Check_load	CPU Load 4%	Berhasil diuji
NRPE	Check_nrpe	Check_load	OK - load	Berhasil diuji
SNMP	Check_snmp	Uptime	SNMP OK - Timetic 1:31:02	Berhasil diuji

All Contacts

Log File Navigation
 Latest Archive ← Fri Nov 27 00:00:00 WIB 2015 to Present..

File: /usr/local/nagios/var/nagios.log

Host	Service	Type	Time	Contact	Notification Command	Information
asus-pc	N/A	HOST UP	11-27-2015 12:58:03	nagiosadmin	notify-host-by-whatsapp	PING OK - Packet loss = 0%, RTA = 15.27 ms
asus-pc	N/A	HOST DOWN	11-27-2015 12:44:44	nagiosadmin	notify-host-by-whatsapp	CRITICAL - Host Unreachable (172.17.215.113)
fadhli	N/A	HOST UP	11-27-2015 11:30:28	nagiosadmin	notify-host-by-whatsapp	PING OK - Packet loss = 0%, RTA = 18.89 ms
asus-pc	N/A	HOST UP	11-27-2015 11:29:30	nagiosadmin	notify-host-by-whatsapp	PING OK - Packet loss = 0%, RTA = 18.89 ms
fadhli	N/A	HOST DOWN	11-27-2015 11:25:25	nagiosadmin	notify-host-by-whatsapp	CRITICAL - Host Unreachable (172.17.215.103)
asus-pc	N/A	HOST DOWN	11-27-2015 11:23:50	nagiosadmin	notify-host-by-whatsapp	CRITICAL - Host Unreachable (172.17.215.113)
fadhli	PING	OK	11-27-2015 11:09:43	nagiosadmin	notify-service-by-whatsapp	PING OK - Packet loss = 0%, RTA = 5.84 ms
fadhli	PING	CRITICAL	11-27-2015 11:04:43	nagiosadmin	notify-service-by-whatsapp	PING CRITICAL - Packet loss = 0%, RTA = 850.17 ms
fadhli	N/A	HOST UP	11-27-2015 11:03:33	nagiosadmin	notify-host-by-whatsapp	PING OK - Packet loss = 0%, RTA = 13.16 ms
tubagush-pc	N/A	HOST UP	11-27-2015 11:03:17	nagiosadmin	notify-host-by-whatsapp	PING OK - Packet loss = 0%, RTA = 0.34 ms
tubagush-pc	N/A	HOST DOWN	11-27-2015 11:02:18	nagiosadmin	notify-host-by-whatsapp	CRITICAL - Bogus ICMP: Port Unreachable (172.17.9.65)
fadhli	N/A	HOST DOWN	11-27-2015 11:01:52	nagiosadmin	notify-host-by-whatsapp	CRITICAL - Bogus ICMP: Port Unreachable (172.17.9.60)

Gambar 8. Tampilan Log File Notifikasi pada Nagios

Pengujian terhadap fungsionalitas web interface

Pengujian terhadap fungsionalitas interface diuraikan menjadi beberapa tahapan yaitu pengujian autentikasi web interface, pengujian hosts, dan pengujian services.

a. Pengujian autentikasi *web interface*

Pengujian dilakukan dengan cara mengakses *web interface* Nagios. Jika input data autentikasi benar, maka akan masuk ke dalam *web interface* Nagios

Tabel 2. Tabel Hasil Pengujian Autentikasi Web Inteface

Hasil Uji Benar			
Input	Hasil yang Diharapkan	Output	Kesimpulan
Data login: Username: Nagiosadmin Password: admin	Data login username dan password valid dapat masuk ke dalam Web Interface Nagios	Input data berhasil dilakukan dan masuk ke dalam web interface	Berhasil diuji
Hasil Uji Salah			
Mengosongkan username dan password	Tidak dapat Login dan kembali pada window autentikasi	Tidak dapat login dan diarahkan kembali pada window autentikasi	Berhasil diuji

b. Pengujian hosts

Pengujian Hosts diuji dengan menonaktifkan hosts agar tidak terkoneksi dengan server Nagios. Untuk melihat hasil monitoring recovery hosts.

Tabel 3. Data Pengujian Hosts “DOWN”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	11:33:03	11:33:29	0:00:26
2	12:01:34	12:01:51	0:00:17
3	12:17:14	12:17:22	0:00:08
4	12:32:35	12:33:22	0:00:47
5	12:48:16	12:48:53	0:00:37

*Durasi = Selisih Waktu Deteksi Nagios dengan Waktu Penonaktifan

Tabel 4. Data Pengujian Hosts “UP”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi
1	11:48:23	11:48:24	0:00:01
2	12:13:25	12:13:26	0:00:01
3	12:29:34	12:29:47	0:00:13
4	12:44:10	12:44:11	0:00:01
5	13:01:32	13:03:27	0:01:30

*Durasi = Selisih Waktu Deteksi Nagios dengan Waktu Recovery

c. Pengujian services

Pengujian dilakukan dengan menonaktifkan 2 *services* yang *dimonitoring* dalam 1 *hosts*. Kondisi awal pengujian adalah *services* yang akan diuji dalam keadaan *OK* atau tidak dalam masalah/error. Pengujian dilakukan dengan menghentikan *services* HTTP (Apache) dan FTP (Filezilla) yang sedang berjalan, maka akan terlihat Nagios mendeteksi bahwa *service* FTP dan HTTP pada waktu yang sama dalam keadaan bermasalah/error.

Tabel 5. Data Pengujian Service FTP “CRITICAL”

Pengujian	Waktu Penonaktifan	Waktu Deteksi Nagios	Durasi*
1	13:16:47	13:24:22	0:07:35
2	13:49:08	13:58:22	0:09:14
3	14:21:08	14: 22:22	0:01:14
4	16:08:14	16:16:34	0:08:20
5	16:34:27	16:42:34	0:08:07

*Durasi = Selisih Waktu Deteksi Nagios dengan Waktu Penonaktifan

Tabel 6. Data Pengujian Service FTP “OK”

Pengujian	Waktu Penonaktifan	Waktu Deteksi Nagios	Durasi*
1	13:35:23	13:38:22	0:02:59
2	14:06:35	14:12:22	0:05:47
3	14:35:33	14:36:22	0:00:49
4	16:23:39	16:30:22	0:06:43
5	16:51:33	16:54:22	0:02:49

*Durasi = Selisih Waktu Deteksi Nagios dengan Waktu Recovery

Tabel 7. Data Pengujian Service HTTP “CRITICAL”

Pengujian	Waktu Penonaktifan	Waktu Deteksi Nagios	Durasi
1	13:16:47	13:25:56	0:09:09
2	13:49:08	13:49:55	0:00:47
3	14:21:08	14:23:55	0:02:47
4	16:08:14	16:17:42	0:09:28
5	16:34:27	16:41:42	0:07:15

*Durasi = Selisih Waktu Deteksi Nagios dengan Waktu Penonaktifan

Tabel 8. Data Pengujian Service HTTP “OK”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	13:35:23	13:39:55	0:04:32
2	14:06:35	14:13:55	0:07:20
3	14:35:33	14:37:55	0:02:22
4	16:23:39	16:31:42	0:08:03
5	16:51:33	16:55:42	0:04:09

*Durasi = Selisih Waktu Deteksi Nagios dengan Waktu Recovery

Hasil pengujian terhadap fungsionalitas notifikasi Telegram

Tabel 9. Data Pengujian Notifikasi terhadap Host “DOWN”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	11:33:29	11:40:56	0:07:27
2	12:01:51	12:07:53	0:06:02
3	12:17:22	12:25:54	0:08:32
4	12:33:22	12:41:53	0:08:31
5	12:48:53	12:58:17	0:09:24

*Durasi = Selisih Waktu Pengiriman Notifikasi dengan Waktu Deteksi Nagios

Tabel 10. Data Pengujian Notifikasi terhadap Hosts “UP”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	11:48:24	11:48:24	0:00:00
2	12:13:25	12:13:22	0:00:00
3	12:29:51	12:29:51	0:00:00
4	12:44:15	12:44:15	0:00:00
5	13:03:27	13:03:27	0:00:00

*Durasi = Selisih Waktu Pengiriman Notifikasi dengan Waktu Deteksi Nagios

Tabel 11. Data Pengujian Notifikasi terhadap Service FTP “CRITICAL”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	13:24:22	13:28:22	00:04:00
2	13:58:22	14:02:22	00:04:00
3	14:22:22	14:26:22	00:04:00
4	16:16:34	16:20:34	00:04:00
5	16:42:34	16:44:34	00:04:00

*Durasi = Selisih Waktu Pengiriman Notifikasi dengan Waktu Deteksi Nagios

Tabel 12 Data Pengujian Notifikasi terhadap Service FTP “OK”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	13:38:22	13:38:22	00:00:00
2	14:12:22	14:12:22	00:00:00
3	14:36:22	14:36:22	00:00:00
4	16:30:22	16:30:22	00:00:00
5	16:54:22	16:54:22	00:00:00

*Durasi = Selisih Waktu Pengiriman Notifikasi dengan Waktu Deteksi Nagios

Tabel 13 Data Pengujian Notifikasi terhadap Service HTTP “CRITICAL”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	13:25:56	13:29:55	00:03:59
2	13:49:55	13:53:55	00:04:00
3	14:23:55	14:27:55	00:04:00
4	16:17:42	16:21:42	00:04:00
5	16:41:42	16:45:42	00:04:00

*Durasi = Selisih Waktu Pengiriman Notifikasi dengan Waktu Deteksi Nagios

Tabel 14. Data Pengujian Notifikasi terhadap Service HTTP “OK”

Pengujian	Waktu Penonaktifan	Waktu Nagios	Durasi*
1	13:39:55	13:39:55	00:00:00
2	14:13:55	14:13:55	00:00:00
3	14:37:55	14:37:55	00:00:00
4	16:31:42	16:31:42	00:00:00
5	16:55:42	16:55:42	00:00:00

*Durasi = Selisih Waktu Pengiriman Notifikasi dengan Waktu Deteksi Nagios

Hasil pengujian terhadap fungsionalitas Event Handler

Tabel 15. Data Pengujian Event Handler terhadap Service Printer Spooler

Pengujian	Waktu Penonaktifan	Waktu Deteksi Nagios	Waktu Aksi Event Handler	*Durasi
1	13:10:57	13:13:45	13:15:45	00:02:00
2	13:32:13	13:35:45	13:37:45	00:02:00
3	14:23:41	13:27:45	14:29:45	00:02:00
4	16:34:40	16:38:35	-	-
5	16:43:31	16:47:34	-	-

*Durasi = Selisih Waktu Aksi Event Handler dengan Waktu Deteksi Nagios

Berdasarkan hasil semua pengujian, maka NSClient++, NRPE, dan SNMP pada *monitoring host* berfungsi dengan baik untuk memberikan informasi *service/resource* kepada Nagios untuk dilakukan *monitoring*. Durasi Nagios dalam mendeteksi perubahan status *hosts* dalam kondisi “DOWN” dan “UP” terjadi selama selang waktu kurang dari 5 menit. Durasi Nagios dalam mendeteksi perubahan status kedua *services* dalam kondisi “CRITICAL” dan “OK” terjadi selama selang waktu kurang dari 10 menit. Nagios melakukan pengiriman notifikasi terjadi selama selang waktu kurang dari 10 menit. Pengiriman notifikasi

terhadap *services* melalui Telegram *Messenger* yang dilakukan oleh Nagios memerlukan waktu 2 menit. *Event Handler* melakukan aksi terhadap *service* terjadi selama selang waktu 2-6 menit. Semua hal tersebut berlangsung baik sesuai konfigurasi yang dilakukan.

KESIMPULAN

Penggunaan *Network Monitoring System* Nagios dengan memanfaatkan WhatsApp sebagai notifikasi alert merupakan cara efektif, mudah dan praktis untuk menangani permasalahan dalam *memonitoring* perangkat yang ada, karena selain dapat memantau keadaan perangkat tanpa mengecek keadaan fisik secara langsung, admin pun akan diberi informasi jika ada perangkat yang mengalami gangguan atau dalam keadaan *down* melalui WhatsApp *messenger* secara *real time*. Selain itu juga tampilan *Web interface* Nagios yang fakultatif untuk melihat status network, urutan masalah dan notifikasi, dan log file.

DAFTAR PUSTAKA

- Ahmad, Anita. 2015. Pengertian Web Server Sebenarnya. <https://bukainfo.com/pengertian-web-server-sebenarnya/> [11 November 2015]
- Anismaj. 2015. *Send Nagios Alert Notification using WhatsApp*. <http://www.unixmen.com/send-nagios-alert-notification-using-whatsapp/> [10 November 2015]
- Balai Pelatihan dan Pengembangan Teknologi Informasi dan Komunikasi (BPPTIK). 2015. *Modul Pelatihan Junior Network Administrator*. Modul Pelatihan Berbasis Standar Kerja Nasional, KOMINFO. Cikarang.
- Kocjan, Wojciech. 2014. *Learning Nagios 4*. Birmingham: Packt Publishing Ltd.
- Kurniawan, Wiharsono. 2007. *Computer Starter Guide: Jaringan Komputer*. Yogyakarta:ANDI; Semarang: SmithDev Community.
- Petersen, Richard. 2014. *Ubuntu 14.04 LTS Desktop: Applications and Administration*. Alameda, CA: Surfing Turtle Press.
- Rasyid, B.A. Solikin. Sularsa, Anang. 2011. " *Realisasi Monitoring Server Menggunakan Nagios dengan Memanfaatkan Event Handler, Email, dan Sms Gateway*". Jurnal Program Studi Teknik Komputer, Politeknik Telkom. Bandung.
- SK. 2015. *How to Install Nagios 4.1 in Ubuntu 15.04*. Tamilnadu, India. <http://www.unixmen.com/how-to-install-nagios-4-1-in-ubuntu-15-04/> [30 Oktober 2015].
- Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer (Ed.I)*. Yogyakarta: ANDI.
- Tarek. 2015. *Yowsup Architecture*. <https://github.com/tgalal/yowsup/wiki/Architecture> [26 November 2015]