



IMPLEMENTASI *NETWORK MONITORING SYSTEM* TERINTEGRASI DENGAN *TICKETING SYSTEM* MENGGUNAKAN NAGIOS DAN OSTICKET

Debi Onas Saputra¹, Henry Saptono²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
saputraonas@gmail.com, henry@nurulfikri.ac.id

Abstract

This research proposes integrating a network monitoring system based on Nagios with a ticketing system based on ticket os. In this study, there are steps to integrate the Nagios-based monitoring system with the ticketing os-based ticketing system and provide convenience for network managers in carrying out network maintenance with a network monitoring system supported by an automatic ticketing system monitoring system that detects infrastructure failures in the network. There are four tests, namely testing the up and down status of the host, then testing the warning and critical service status. The results obtained from testing the integration of the Nagios-based network monitoring system with the osTicket-based ticketing system, namely utilizing the alerting feature on Nagios and the web API feature on osTicket showed effective results. This is evidenced by the automatic formation of tickets on the ticket when the host status is down and when the service status is warning or critical.

Keywords: *Network monitoring system, Alerting, Python, Nagios, osTicket*

Abstrak

Penelitian ini diajukan untuk mengintegrasikan *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket. Pada penelitian ini terdapat langkah-langkah dalam mengintegrasikan *monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket dan memberikan kemudahan bagi pengelola jaringan dalam melakukan pemeliharaan jaringan dengan adanya *network monitoring system* yang didukung dengan *ticketing system* otomatis ketika *network monitoring system* mendeteksi kegagalan infrastruktur dalam jaringan. Terdapat 4 pengujian yaitu, pengujian *host status up* dan *down*, kemudian pengujian *service status warning* dan *critical*. Hasil yang didapat dari pengujian pada integrasi *network monitoring system* berbasis Nagios dengan *system ticketing* berbasis osTicket ini yaitu memanfaatkan fitur alerting pada nagios dan fitur web API yang ada pada osTicket menunjukkan hasil yang efektif. Hal ini dibuktikan dari terbentuknya tiket secara otomatis pada osTicket pada saat status host down dan ketika status *service warning* atau *critical*.

Kata kunci: *Network monitoring system, Alerting, Python, Nagios, osTicket*

1. PENDAHULUAN

Network Monitoring System adalah sebuah sistem yang memiliki fungsi untuk mengamati atau memonitor sistem jaringan komputer yang sedang berjalan dan memungkinkan deteksi kesalahan pada jaringan. Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program atau memantau perubahan, yang fokus pada proses dan keluaran.

Network Monitoring System dilakukan dengan menggunakan sebuah komputer yang dijadikan sebagai

server yang dihubungkan menggunakan jaringan lokal. Pada umumnya sebuah monitoring jaringan mengharuskan administrator untuk terus berada didepan layar agar mengetahui setiap perubahan yang terjadi hal ini kurang efisien. Untuk mengatasi masalah tersebut akan dilakukan sebuah monitoring yang menggunakan perangkat lunak yaitu Nagios.

Sebelumnya *system monitoring* berbasis Nagios hanya melakukan metode alerting melalui email. Setelah itu *system monitoring* tersebut bisa dikembangkan pada metode alerting. Oleh karena itu peneliti mengembangkan

system monitoring berbasis Nagios yang terintegrasi pada *system ticketing* berbasis osTicket dengan menerapkan alerting pada ticketing berbasis osTicket. Supaya pengguna bisa memanfaatkan *system ticketing* pada osTicket untuk memonitoring *system*.

Oleh karena itu dapat memberikan kemudahan bagi pengelola jaringan dalam melakukan pemeliharaan dengan adanya *network monitoring system* yang didukung dengan *system ticketing* otomatis ketika *network monitoring system* mendeteksi kendala infrastruktur dalam jaringan. Oleh karena itu dengan adanya sistem ticketing atau IT *helpdesk* dapat memudahkan semua penggunaan perangkat IT melaporkan setiap permasalahan yang dihadapi. Sehingga, dapat menghasilkan solusi yang tepat dalam mengatur sumber daya yang ada.

Nagios dapat memberi informasi tentang kondisi komponen jaringan secara *realtime* dengan mengirimkan pesan kepada administrator jaringan melalui osTicket. Sehingga osTicket ini berfungsi untuk mengetahui notifikasi dan permasalahan yang terjadi pada sebuah *monitoring* tersebut.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, rumusan masalah dari penelitian ini adalah:

1. Proses apakah yang dapat digunakan atau diterapkan dalam mengintegrasikan *network monitoring system* berbasis Nagios dengan ticketing system osTicket?
2. Bagaimanakah cara integrasi *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket?
3. Bagaimanakah efektifitas dari integrasi *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket?

1.2 Tujuan penelitian

1. Mengimplementasikan *network monitoring system* berbasis Nagios dan *ticketing system* berbasis osTicket.
2. Merancang dan membuat program untuk mengintegrasikan *network monitoring system* berbasis Nagios dengan *system ticketing* berbasis osTicket.
3. Menyediakan mekanisme alerting dan notifikasi pada *network monitoring* Nagios yang dapat menciptakan ticket secara otomatis pada ticket berbasis OS Ticket.

1.3 Manfaat Penelitian

1. Memberikan kemudahan bagi pengelola jaringan dalam melakukan pemeliharaan jaringan dengan adanya *network monitoring system* yang didukung

dengan *ticketing system* otomatis ketika *network monitoring system* mendeteksi kegagalan infrastruktur dalam jaringan.

2. Menghasilkan dokumen tugas akhir yang dapat menjadi sebuah alternatif referensi bagi siapa pun yang ingin mempelajari dan menerapkan sebuah *monitoring system* berbasis Nagios yang terintegrasi dengan *ticketing system* berbasis osTicket.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Implementasi *network monitoring system* dan *ticketing system* dilakukan pada lingkungan percobaan.
2. Mekanisme peringatan (*alerting*) dari *network monitoring system* hanya berfokus pada pembuatan ticket otomatis di osTicket.
3. Hal hal yang dimonitoring oleh *network monitoring system* hanya yang berkaitan dengan status dan performa sistem.
4. Penelitian tidak mengembangkan atau memodifikasi dashboard monitoring system Nagios maupun ticketing system pada osTicket.

2. LANDASAN TEORI

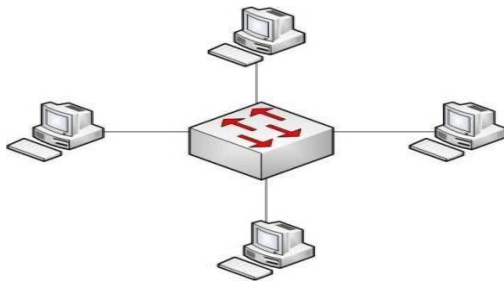
2.1 Jaringan Komputer

Jaringan komputer merupakan sekelompok komputer otonom yang saling berhubungan satu sama lain, dengan menggunakan satu protokol komunikasi sehingga seluruh komputer yang saling terhubung tersebut dapat berbagi informasi, program, sumber daya dan juga dapat saling menggunakan perangkat keras lainnya secara bersamaan, seperti printer, harddisk dan lain sebagainya. Jaringan komputer merupakan kumpulan sejumlah terminal komunikasi yang tersebar di berbagai tempat yang terdiri lebih dari satu komputer yang saling berhubungan.

Ada dua bagian dalam sebuah jaringan komputer yaitu pihak yang meminta layanan pada jaringan di sebut klien (*client*) sedangkan yang memberikan layanan pada jaringan komputer disebut peladen (*server*).

2.2 LAN (*Lokal Area Network*)

LAN adalah singkatan dari *Lokal Area Network*. Jenis jaringan LAN ini sangat sering kita temui di warnet-warnet, kampus, sekolah ataupun perkantoran yang membutuhkan hubungan atau koneksi antara dua komputer atau lebih dalam suatu ruangan. Jaringan LAN juga merupakan jaringan yang sangat di pengaruhi oleh topologi jaringannya.

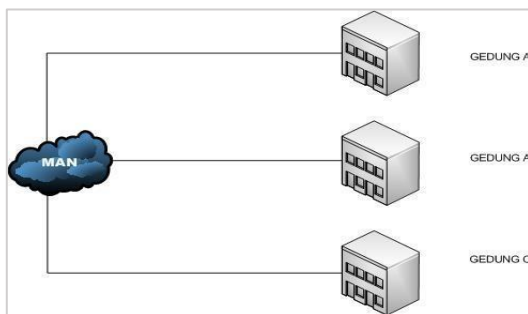


Gambar 1. LAN

2.3 MAN (Metropolitan Area Network)

MAN singkatan dari *Metropolitan Area Network*. Jenis jaringan komputer MAN ini adalah suatu jaringan komputer dalam suatu kota dengan transfer data berkecepatan tinggi yang menghubungkan suatu lokasi seperti sekolah, kampus, perkantoran dan pemerintahan.

Sebenarnya jaringan MAN ini adalah gabungan dari beberapa jaringan LAN. Jangkauan dari jaringan MAN ini bisa mencapai 10 - 50 kilometer.



Gambar 2. MAN

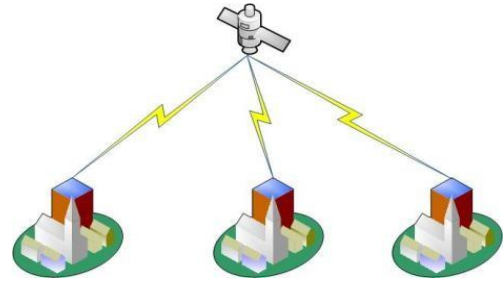
2.4 WAN (Wide Area Network)

WAN singkatan dari *Wide Area Network*. WAN adalah jenis jaringan komputer yang mencakup area yang cukup besar. contohnya adalah jaringan yang menghubungkan suatu wilayah atau suatu negara dengan negara lainnya [6]. Adapun beberapa karakteristik dari jaringan WAN, diantaranya sebagai berikut ini:

- Biasanya WAN digunakan untuk menghubungkan perangkat-perangkat yang tidak dapat dihubungkan melalui jaringan LAN dan jaringan MAN. Oleh karena itu jaringan WAN digunakan untuk menghubungkan jaringan yang sangat luas.
- WAN memiliki cakupan area yang sangat luas. Biasanya pada jaringan WAN akan melibatkan operator telekomunikasi. Tujuannya menggunakan operator telokomunikasi yaitu supaya perangkat-perangkat yang ada dalam jaringan WAN dapat saling berkomunikasi satu sama lain.
- Menggunakan koneksi serial dari berbagai macam jenis untuk dapat mengakses *bandwith* dalam lokasi yang berjauhan atau luas.
- Dapat melakukan pertukaran paket data maupun

frame antar *router* atau *switch* dan jaringan LAN yang sudah dibangun.

- Bekerja pada layer fisik dan pada layer data link dari layer OSI.



Gambar 3. WAN

2.5 TCP/IP

TCP/IP (*Transmission Control Protokol/Internet Protokol*) adalah sekumpulan protocol yang terdapat di dalam jaringan komputer yang digunakan untuk berkomunikasi atau bertukar data antar komputer. TCP/IP merupakan protokol standar pada jaringan internet yang menghubungkan banyak komputer yang berbeda jenis mesin maupun sistem operasi agar dapat berinteraksi satu sama lain. TCP/IP secara umum berfungsi untuk memilih rute terbaik transmisi data, memilih rute alternative jika suatu rute tidak dapat digunakan, mengatur, dan mengirimkan paket-paket pengiriman data dan lain-lain. TCP/IP merupakan protokol yang memungkinkan sistem di seluruh dunia berkomunikasi pada jaringan tunggal yang disebut *Internet*.

2.6 Network Monitoring System

Network Monitoring System adalah sistem yang memiliki fungsi untuk mengamati atau memonitor sistem jaringan komputer yang sedang berjalan dan memungkinkan deteksi dini terjadinya kesalahan pada jaringan. *Network Monitoring System* dilakukan dengan menggunakan sebuah komputer yang dijadikan sebagai *server* yang dihubungkan menggunakan jaringan *local*. *Network Monitoring System* adalah sistem ekstra atau kumpulan sistem yang memiliki tugas mengamati/memonitor sistem-sistem terhadap kemungkinan terjadinya masalah-masalah pada sistem tersebut untuk dapat dideteksi secara dini. Suatu *network monitoring system* dapat secara periodik menghubungi suatu server untuk menjamin adanya respon dari server, jika tidak ada respon maka monitoring sistem kemudian mengirimkan pesan atau notifikasi ke administrator.

2.7 Server

Server adalah komputer yang hanya memberikan layanan bagi komputer lain [9]. *Server* didukung oleh perangkat lunak yang memadai untuk mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya. Tugas utama *server* adalah melayani beberapa komputer

client.

2.8 Client

Client adalah komputer yang hanya bisa menggunakan resource yang disediakan *server* sesuai dengan otoritas yang diberikan oleh administrator. Aplikasi yang dijalankan pada sisi *client* bisa saja merupakan *resource* yang tersedia di server atau aplikasi yang diinstall di sisi *client* namun hanya bisa di jalankan setelah terkoneksi ke sever.

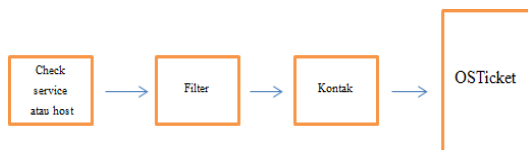
2.9 SNMP (Simple Network Management Protocol)

Simple Network Management Protocol merupakan *protocol* aplikasi yang mampu menjalankan tugas untuk memonitoring kondisi jaringan. SNMP akan mempengaruhi proses monitoring dan manajemen jaringan karena dengan menggunakan SNMP akan dapat diketahui informasi tentang kondisi perangkat jaringan yang diamati.

SNMP adalah sebuah *protocol Application layer* pada standard 7 OSI *layer* dan merupakan bagian dari *protocol TCP/IP* yang banyak digunakan untuk mengatur pertukaran *database* informasi yang menyangkut sistem manajemen dari sebuah perangkat jaringan.

2.10 Alert

Alert (peringatan) adalah notifikasi laporan yang dikirimkan Nagios kepada kontak administrator. *Alert* dapat dikirim melalui osTicket. *Alert* dapat dikirim ketika nagios mendeteksi perubahan status pada suatu *host* atau servis, maka Nagios akan memberitahukan melalui web *interface* dan *log file* serta mengirimkan notifikasi berupa *ticketing system* pada osTicket kealamat administrator. Filter pesan merupakan aturan-aturan yang dibuat dalam konfigurasi agar aktivitas *monitoring* dan notifikasi yang dikirim ke administrator yang diinginkan.



Gambar 4. Alert

2.11 Nagios

Nagios adalah aplikasi monitoring yang dapat digunakan untuk monitoring sistem komputer, monitoring jaringan dan monitoring infrastruktur jaringan. Nagios diciptakan oleh Ethan Galstad. Nagios berbasis *open source* yang dapat dijalankan pada sistem operasi Linux. Sistem monitoring Nagios memonitor seluruh infrastruktur IT untuk memastikan sistem, aplikasi, layanan, dan proses bisnis yang berfungsi dengan baik. Jika jaringan mengalami masalah, Nagios dapat langsung memberikan alert kepada teknisi jaringan sehingga memungkinkan

teknisi untuk mengatasi masalah tersebut lebih awal [4]. Adapun fungsi dari Nagios:

- Memantau layanan jaringan (SMTP, POP3, HTTP, NNTP, PING, dll).
- Memantau *resource server* (*Processor load, Disk usage, memory, dll*).
- Kemudahan untuk membuat *plugin*.
- Pengecekan layanan secara paralel.
- Dengan implementasi “*parents*” dari suatu *host*, maka bisa dengan mudah dibedakan *host* yang benar-benar *down*, dan *host* yang terkena dampaknya.

2.12 Helpdesk Ticketing System

Ticketing System adalah sebuah karcis gangguan atau disebut laporan masalah adalah mekanisme yang digunakan dalam suatu organisasi untuk melacak deteksi, pelaporan, dan resolusi dari beberapa jenis masalah. Trouble sistem *ticketing* berasal dari manufaktur sebagai dasar sistem pelaporan kertas, sekarang kebanyakan berbasis web dan terkait dengan hubungan manajemen pelanggan (CRM) lingkungan, seperti *call center* atau *e-business* [8]. Adapun *helpdesk* bertujuan untuk memudahkan semua penggunaan perangkat IT dalam ruang lingkup perusahaan maupun lembaga instansi pemerintah yang melaporkan setiap permasalahan yang dihadapi. Sehingga, dapat menghasilkan solusi yang tepat dalam mengatur sumber daya yang ada.

2.13 osTicket

osTicket adalah aplikasi *open source* yang digunakan untuk *ticketing system* gratis. osTicket bisa menjadi solusi murah untuk sebuah perusahaan menengah ke bawah untuk mengarsipkan kegiatan-kegiatan atau mungkin pendataan *problem-problem* yang terjadi pada perusahaan [9].

Aplikasi osTicket memiliki beberapa kelebihan dibandingkan aplikasi sistem pengelolaan tiket yang lain, beberapa kelebihan yang dimiliki osTicket adalah sebagai berikut :

- osTicket merupakan aplikasi yang berbasis *web-based platform* sehingga dapat diakses secara luas.
- Memiliki fitur-fitur diantaranya: *auto responder* tiket, email yang terintegrasi, dilengkapi dengan *role-based access* untuk manajemen *user, ticket assignment, alert* dan *notice* untuk tiket yang dikelola, *ticket filtering, email templates, service level agreements* dan *knowledge base management*
- Mendukung pemanfaatan komunikasi data dan informasi menggunakan layanan API.

2.14 Apache Web Server

Apache adalah sebuah *server web* yang dapat dijalankan di banyak system operasi (Unix, BSD, Linux, Microsoft Windows dan *platform* lainnya) yang berguna untuk

melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat diatur autentikasi berbasis basis data dan lainnya. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik yang memungkinkan penanganan server menjadi mudah [5].

2.15 Python

Python adalah bahasa pemrograman yang bersifat *open source*. Bahasa pemrograman ini dioptimalisasikan untuk *software quality, developer productivity, program portability, dan component integration*. Python telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti *internet scripting, systems programming, user interfaces, product customization, numeric programming* dll. Python saat ini telah menduduki posisi 4 atau 5 bahasa pemrograman paling sering digunakan di seluruh dunia. Bahasa pemrograman Python memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak.

Berikut adalah beberapa fitur yang ada pada bahasa pemrograman Python :

1. Multi Paradigm Design
2. Open Source
3. Simplicity
4. Library Support
5. Portability
6. Extendable

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Hardware

Di dalam hardware ini menggunakan 2 virtual dengan spesifikasi sebagai berikut :

- Processor : Dual Core , 1,2 Ghz
- RAM : 1 GB
- Harddisk : 20 GB

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

- Processor : Intel® Core i5-7200U
CPU @2.50GHz
- RAM : 8.00 GB
- Hardisk : 1 TB

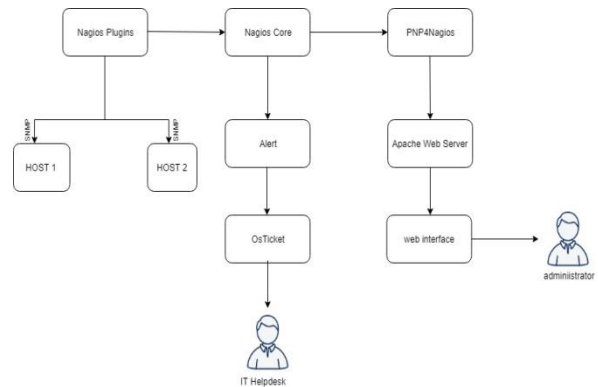
3.2 Analisis Kebutuhan Software

Dalam penelitian analisis kinerja HDFS sebagai infrastruktur pembangunan big data, peneliti menggunakan perangkat lunak (*software*) yang akan digunakan:

1. Platform Monitoring : Nagios 4.3.4
2. Apache web server 2.4.37

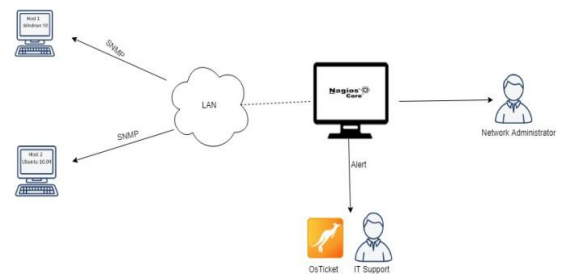
3. SNMP package
4. Platform Ticketing : osTicket
5. Apache web server 2.4.37
6. PHP 7.0
7. Mysql-server
8. Virtual Machine : Virtualbox dan VMWare
9. Web browser Chrome
10. Sistem Operasi : Linux Ubuntu 16.04
11. Bahasa Pemrograman : Python 2.7.123

3.3 Perancangan Arsitektur Sistem Logic



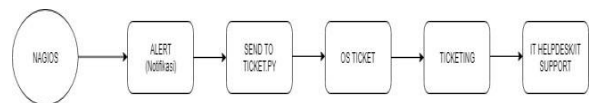
Gambar 4. Perancangan Arsitektur Sistem Logic

3.4 Perancangan Arsitektur Sistem Fisik



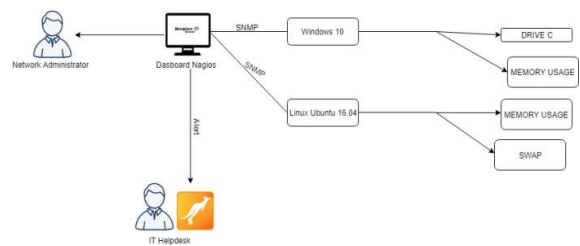
Gambar 5. Perancangan Arsitektur Sistem Fisik

3.5 Perancangan Alerting



Gambar 6. Perancangan Alerting

3.6 Rancangan Pengujian



Gambar 7. Rancangan Pengujian

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

1. Instalasi Nagios dengan *men-download* pada *website*-nya. Kemudian ekstrak dengan perintah:

```
$tar xzf nagios-4.3.4.tar.gz
```

2. Setelah mengekstrak, kemudian masuk ke direktori Nagios dan jalankan *configure* skrip untuk menentukan pengguna dan grup pada, dengan ketik perintah:

```
./configure--with-nagiosgroup=nagios--with-command-group=nagcmd
```

```
$make all
```

3. Kemudian untuk mengeluarkan perintah eksternal melalui antarmuka *web* ke Nagios, maka tambahkan pengguna server *web* yaitu *www-data* ke grup *nagcmd*:

```
$sudo usermod -G nagcmd www-data
```

4. *Men-download plugins* pada *website* Nagios kemudian ekstrak file *download*-nya

```
$tar xzf nrpe-3.2.1.tar.gz
```

5. Kemudian *configure* *nrpe* dengan perintah *configure*

```
./configure
```

6. Kemudian pasang dan install *Check_nrpe* dengan perintah *make*:

```
$ make check_nrpe
```

```
$ sudo make install-plugin
```

7. Kemudian instalasi *Apache web server*, dengan ketik perintah

```
$sudo apt-get install apache2
```

8. Selanjutnya konfigurasi *Apache* untuk melayani antarmuka pengguna Nagios. Aktifkan modul penulisan ulang dan *cgi* *Apache* dengan perintah *a2enmod*

```
$ sudo a2enmod rewrite
```

```
$ sudo a2enmod cgi
```

9. Selanjutnya membuat *user* dan *password* untuk *login* ke *web interface* Nagios. Dengan menjalankan perintah seperti di bawah ini

```
$sudo htpasswd -c
```

```
/usr/local/nagios/etc/htpasswd.users
```

10. Setelah itu membuat tautan untuk *nagios.conf* ke direktori yang didukung situs *nagios*

```
$sudo ln -s /etc/apache2/sites-available/nagios.conf
```

```
/etc/apache2/sites-enabled/
```

11. Setelah itu masuk ke direktori *nagios.conf* dan ubah seperti dibawah ini :

```
# SSLRequireSSL (Sebelum)
SSLRequireSSL (Sesudah)
```

```
Order allow,deny (sebelum) Allow
from all (sebelum)
```

```
#Order allow,deny (sesudah)
```

```
#Allow from all (sesudah)
```

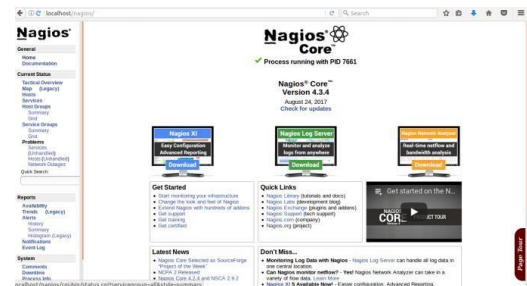
```
#Order deny,allow (sebelum) #Deny
from all (sebelum)
```

```
#Allow from 127.0.0.1 (sebelum)
```

```
Order deny,allow (sesudah) Deny
from all (sesudah)
```

```
Allow from 127.0.0.1 (sesudah)
```

12. Kemudian *restart* *nagios* dan membuka browser *localhost/nagios/* seperti gambar di bawah ini



Gambar 8. Restart Nagios

13. Kemudian Instalasi *osticket* dengan *men-download* pada *website*-nya. Kemudian ekstrak filenya seperti printer dibawah ini

```
$unzip osticket-v1.11.0-rc-1.zip
```

14. Kemudian mengubah *permission* direktori *osticket-v1.11.0-rc1* menjadi sebagai berikut

```
$ sudo chmod 755 -R osticket-
v1.11.0-rc1/
```

15. Selanjutnya, membuat *virtual host* untuk konfigurasi dengan perintah seperti dibawah ini

```
$ sudo nano/etc/apache2/sites-
available/000-default.conf
```

```
ServerAdmin webmaster@localhost
```

```

DocumentRoot /var/www/html
ServerName localhost

Alias /tiket
"/var/www/html/osticket/"

<Directory
/var/www/html/osticket/> Options
FollowSymLinks AllowOverride All

Order allow,deny allow from all

</Directory>

<Directory /var/www/html> Options
FollowSymLinks AllowOverride All

Order allow,deny

```

16. Setelah itu lakukan konfigurasi SSL-nya. Kemudian masukkan perintah seperti dibawah ini untuk masuk ke file yang ingin dikonfigurasi

```

$ sudo nano

/etc/apache2/sites
available/default-ssl.conf

<VirtualHost _default_:443>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/osticket-
v1.11.0-rc1/

ServerName localhost

ServerAlias http://localhost/

Alias /localhost
"/var/www/html/osticket/osticket-
v1.11.0-rc1/"

<Directory
/var/www/html/osticket/> Options
FollowSymLinks

AllowOverride All

Order allow,deny

allow from all

</Directory>

```

17. Selanjutnya menghidupkan konfigurasi yang sudah dibuat tadi untuk hak akses dengan perintah dibawah ini:

```

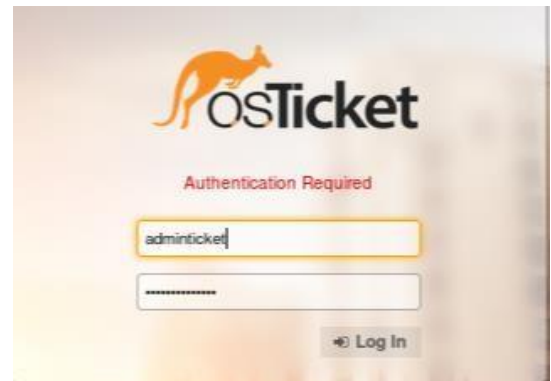
$a2ensite 000-default.conf

$a2ensite default-ssl.conf

```

18. Kemudian *restart* Apache dan membuka browser osTicket localhost/tiket/upload/scp/login.php.

berikut hasil tampilan dashboard pada osTicket.



Gambar 9. Tampilan Dashboard osTicket

19. Membuat API key osTicket. kemudian masuk ke admin panel, selanjutnya pilih *add API key*.

APIKey	IP Address	Status	Date Added	Last Updated
FD20114693E79F76069B310B3FE72C7D	192.168.32.7	Active	2/27/19	2/27/19, 2:01 PM

Gambar 10. API Key

20. Kemudian membuat kode program Python untuk mengintegrasikan *network monitoring system* berbasis Nagios dengan *system ticketing* berbasis osTicket. Pada sebelumnya peneliti mendapatkan referensi bentuk kode program pada *website* <http://sirlagz.net/2016/08/10/integrating-zabbix-with-osticket/> kemudian kode program tersebut dimodifikasi sesuai kebutuhan penelitian dalam mengintegrasikan *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket.

```

url =

'http://192.168.32.5/tiket/api/ticket
s.json'

api_key =
'FD20114693E79F76069B310B3FE72C7D'

title = " "

def parse_args():

    parser =
    argparse.ArgumentParser(description
    =title)

        parser.add_argument('-
o', '-- object_type', nargs='?',
required=True)

        parser.add_argument('--
contact', nargs='?',
required=True)

```

```

        parser.add_argument('--
notificationtype', nargs='?')
        parser.add_argument('--
hoststate', nargs='?')
        parser.add_argument('--
hostname', nargs='?')
        parser.add_argument('--
hostaddress', nargs='?')

        parser.add_argument('--
servicestate', nargs='?')
        parser.add_argument('--
servicedesc', nargs='?')
        parser.add_argument('--
output', nargs='?')
args = parser.parse_args()

    return
args argsx =
parse_args()

def send_ticket(title,
message): priority = 1

if argsx.object_type == 'host':
    title = title +
argsx.object_type + " of " +
argsx.hostname + " " +
argsx.hoststate

    if argsx.hoststate
== "UP":
        priority = 2

    if argsx.hoststate
==
        priority = 4

    title = title +

"DOWN":

    else:

argsx.object_type + " of " +
argsx.hostname + " " +
argsx.servicestate

        if argsx.servicestate
==

"OK":

```

```

        priority = 2

"WARNING":    if argsx.servicestate ==
                priority = 3

"Critical":    if argsx.servicestate ==
                priority = 4

    payload = {
        "autorespond": False,
        "source": "API",
        "name": "Debi Onas",
        "email":
            "saputraonas@gmail.com",
        "subject": title,
        "message": message
        "priority": priority
    }

    header = {
        'Content-Type':
            'application/json', 'X-API-Key': api_key
    }

    r =
requests.post(url,json=payload,header
s=header)

def host_notification(args):
    state = ''

    return '%s (%s) state is %s -
%s'%( args.hostname,
args.hostaddress,
args.hoststate,
args.output
)

def service_notification(args):
    state = ''

    return '%s (%s) state is %s -
%s '

%(args.hostname,args.servicedesc,
args.servicestate, args.output)

if_name_== '_main_':
    args = parse_args()

```



```

if args.object_type ==
    'host':
        message =
host_notification(args)

elif args.object_type ==
    'service': message
=

service_notification(args)

send_ticket(title, message)

```

21. Membuka direktori command.cfg pada nagios untuk mendefinisikan argument service dan host.

```

define command {
    command_name notify-host-
by- osticket

    command_line

/usr/lib/nagios/plugins/check_ser
ver.py --object_type host --

contact "$CONTACTPAGER$" --
notificationtype
"$NOTIFICATIONTYPE$" --
hoststate "$HOSTSTATE$" --
hostname "$HOSTNAME$" -
hostaddress

"$HOSTADDRESS$" --output
"$HOSTOUTPUT$"
}

```

```

define command {
    command_name notify-
service- by-osticket

command_line

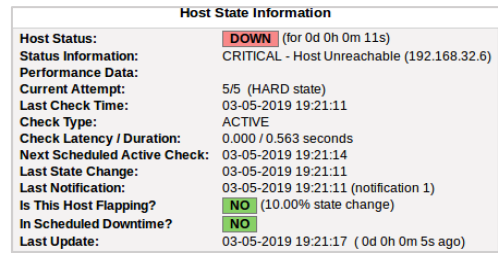
/usr/lib/nagios/plugins/check_s
er ver.py --object_type service
-- contact "$CONTACTPAGER$" --
notificationtype
"$NOTIFICATIONTYPE$" --

servicestate "$SERVICESTATE$"
-- hostname "$HOSTNAME$" --
servicedesc "$SERVICEDESC$" --
output "$SERVICEOUTPUT$"
}

```

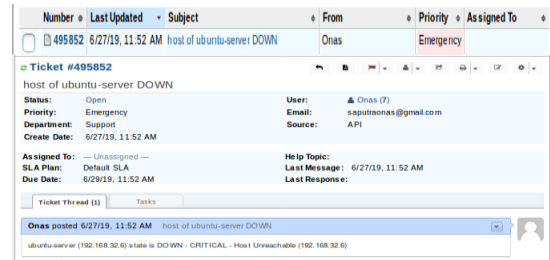
4.2 Pengujian Host

a. Mematikan host Ubuntu dengan cara mematikan panel koneksi network. Maka hasil Nagios seperti dibawah ini.



Gambar 11. Host State Information

b. Melihat pesan notifikasi pada osTicket bahwa host Ubuntu down.



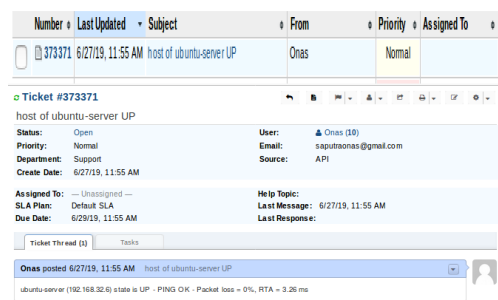
Gambar 12. Melihat Pesan Notifikasi pada osTicket

c. Kemudian menghidupkan kembali panel koneksi network pada host Ubuntu, kemudian terdapat perubahan pada nagios bahwa host Ubuntu berstatus up.



Gambar 13. Menghidupkan panel koneksi network

d. Kemudian melihat notifikasi pesan pada osTicket bahwa host Ubuntu mengalami perubahan status up



Gambar 14. Melihat Notifikasi Pesan

4.3 Hasil Pengujian Host

Setelah melakukan pengujian monitoring status host, peneliti membuat hasil pengujian dalam bentuk tabel untuk keberhasilan yang telah di ujikan sebanyak 4 kali. Hal ini menyatakan bahwa efektifitas keberhasilan sebuah notifikasi pada host Ubuntu, akan tetapi script python

tersebut bisa diujikan juga untuk jenis sistem operasi lain selain sistem operasi Ubuntu. Karena pada penelitian ini peneliti hanya melakukan salah satu *host* yaitu *host* Ubuntu. Berikut peneliti memaparkan keberhasilan terbentuknya notifikasi tiket secara otomatis pada osTicket untuk *host* Ubuntu.

Tabel 1. Hasil Pengujian Host

Host	Pengujian	Status host	Notifikasi
Ubuntu 16	Mematikan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Down	Tiket Terbentuk
Ubuntu 16	Menghidupkan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Up	Tiket Terbentuk
Ubuntu 16	Mematikan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Down	Tiket Terbentuk
Ubuntu 16	Menghidupkan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Up	Tiket Terbentuk

4.4 Pengujian Service

a. Melakukan perubahan ukuran *threshold* memory pada *command.cfg* di nagios. Pada sebelumnya -w 80-c 90 menjadi -w 1 -c 2. Untuk -w dalam artian maksimum nilai persen dalam status warning jika dia dari 1%-2% dia berstatus warning dan -c itu jika dia melebihi 2% maka berstatus critical. Setelah itu simpan perubahan.

```
define command {
    command_name    check_mem_linux
    command_line    $USER1$/nagios_plugins/check_smp_storage.pl -H $HOSTADDRESS$ -c $ARGC1$ -n "Memory" -w 1 -c 2 -f
}
```

Gambar 15. Perubahan Ukuran *Threshold* Memory

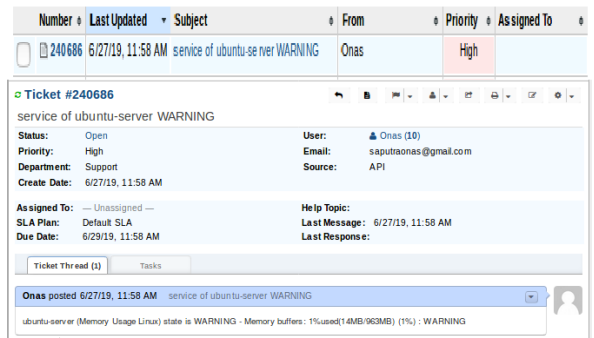
b. Kemudian melihat *dashboard* Nagios untuk melihat hasil sttus perubahan pada *service*, bahwa *service* berstatus *warning*.



Gambar 16. Melihat *Dashboard* Nagios

c. Kemudian melihat hasil notifikasi berupa tiket pada osTicket untuk melihat pesan dan membuka isi pesan

bahwa *memory* berstatus *warning*.



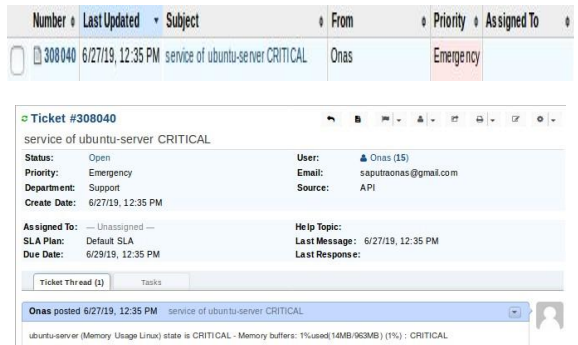
Gambar 17. Melihat Hasil Notifikasi

d. Kemudian peneliti menguji lagi, menjalankan program seperti Firefox, Libre Office di *host* Ubuntu tersebut. Maka hasilnya di Nagios *host* Ubuntu status berubah menjadi *critical*.



Gambar 18. Nagios *Host* Ubuntu Berubah menjadi *Critical*

e. Selanjutnya melihat hasil notifikasi berupa tiket pada osTicket untuk melihat pesan dan membuka pesan bahwa *memory* berstatus *critical*.



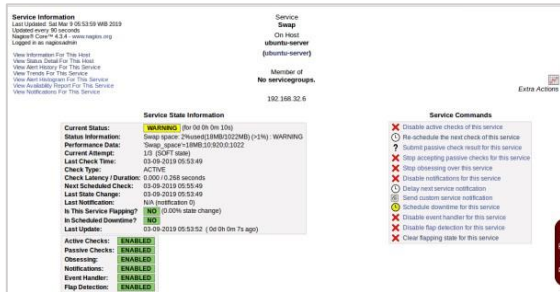
Gambar 19. Melihat Hasil Notifikasi berupa Tiket

f. Kemudian peneliti mengubah ukuran *threshold* swap pada *command.cfg* di nagios. Pada sebelumnya -w 80 - c 90 menjadi -w 1 -c 90. Untuk -w dalam artian maksimum nilai persen dalam status warning jika dia dari 1%-89% dia berstatus warning dan -c itu jika dia melebihi 90% maka berstatus critical. Setelah itu simpan perubahan.

```
define command {
    command_name check_swap_linux
    command_line SUSEB15/nagios_plugins/check_smp_storage.pl -H $HOSTADDRESS$ -C $ARG1$ -n $Swap -w 11 -c 90 -f
}
```

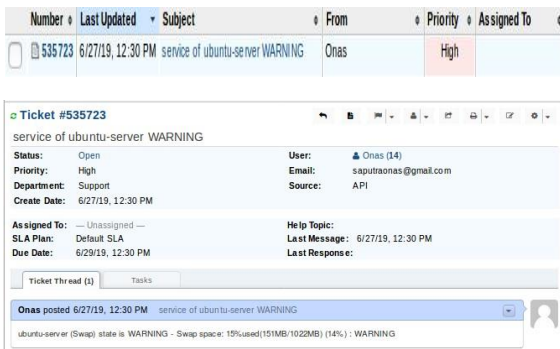
Gambar 20. Mengubah Ukuran *Threshold Swap* pada *Command*

- g. Setelah itu melihat dashboard nagios pada perubahan swap di nagios.



Gambar 21. Melihat Dashboard Nagios pada Perubahan Swap

- h. Kemudian mengecek notifikasi pada osticket. Untuk melihat pesan dan isi pesan notifikasi tersebut.



Gambar 22. Mengecek Notifikasi pada OS Ticket

- i. Kemudian peneliti mengubah ukuran *threshold* ukuran *swap* pada Nagios untuk mengubah status menjadi *critical*.

```
define command {
    command_name check_swap_linux
    command_line SUSEB15/nagios_plugins/check_smp_storage.pl -H $HOSTADDRESS$ -C $ARG1$ -n $Swap -w 0 -c 1 -f
}
```

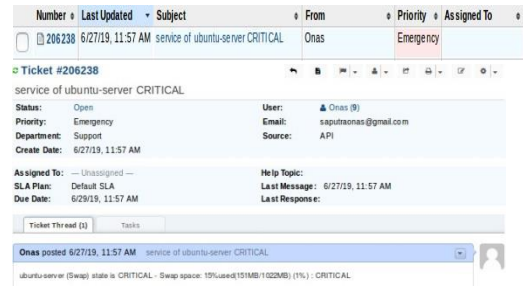
Gambar 23. Mengubah Ukuran *Threshold* Ukuran *Swap* pada Nagios

- j. Kemudian melihat hasil perubahan pada Nagios.



Gambar 24. Melihat Hasil Perubahan pada Nagios

- k. Kemudian mengecek notifikasi berupa tiket pada osTicket untuk melihat pesan dan membuka isi pesan bahwa *swap* berstatus *crit*



Gambar 25. Melihat dan Membuka Isi Pesan berstatus *Critical*

4.5 Hasil Pengujian *Service*

Setelah melakukan pengujian monitoring *service*, peneliti melakukan tahap pengujian 2 *service* yaitu *ram* dan *swap*. Kemudian peneliti membuat hasil pengujian dalam bentuk tabel sebanyak 4 kali pengujian ketika *service* mendapatkan kendala problem yaitu *warning* dan *critical*. Hal tersebut menyatakan efektifitas keberhasilan bahwa pada saat terjadinya problem di *service* maka menciptakan tiket secara otomatis pada osTicket. Berikut peneliti memaparkan keberhasilan terbentuknya notifikasi tiket secara otomatis pada osTicket untuk *service*.

Tabel 2. Hasil Pengujian *Service*

Host	Pengujian	Status	Notifikasi
Ubuntu	Mengubah Ukuran <i>Threshold</i> atau ukuran maksimum <i>memory</i>	<i>Warning</i>	Tiket Terbentuk
Ubuntu	Menjalankan program seperti Firefox, Libre Office pada host Ubuntu	<i>Critical</i>	Tiket Terbentuk
Ubuntu	Menjalankan program seperti Firefox, Libre Office pada <i>host</i> Ubuntu	<i>Warning</i>	Tiket Terbentuk
Ubuntu	Mengubah Ukuran <i>Threshold</i> atau ukuran Maksimum <i>Swap</i>	<i>Critical</i>	Tiket terbentuk

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

- a. Berdasarkan fitur *alerting* atau notifikasi yang terdapat pada *network monitoring system* berbasis Nagios, peneliti memanfaatkan fitur tersebut sebagai

metode atau proses yang digunakan dalam mengintegrasikan *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket.

- b. Dengan adanya fitur *alerting* pada Nagios, peneliti membuat *script* dengan bahasa pemrograman Python yang melakukan akses ke *web API* dari osTicket *system*. Akses ke *web API* osTicket ini akan melakukan pembuatan tiket baru yang menunjukkan sebuah notifikasi yang terjadi pada nagios saat terjadi *host down* dan *service warning* dan *critical*.
- c. Berdasarkan hasil perancangan, penerapan, dan pengujian yang sudah dilakukan oleh peneliti, proses integrasi *network monitoring system* berbasis nagios dengan *ticketing system* berbasis osTicket, yang memanfaatkan fitur *alerting* pada nagios dan fitur *web API* yang ada pada osTicket menunjukkan hasil yang efektif. Hal ini dibuktikan dari terbentuknya tiket secara otomatis pada osTicket pada saat status *host down* dan ketika status *service warning* atau *critical*.

5.2 Saran

1. Penelitian berikutnya dapat mengimplementasikan kedalam sistem yang real atau tidak dalam lingkungan virtualisasi.
2. Melakukan peningkatan atau optimasi kode program Python yang dibuat oleh peneliti untuk pembuat tiket otomatis pada osTicket pada saat terjadinya perubahan status down menjadi status *up* dan dari status *service warning* atau *critical* menjadi status ok akan memproses terjadinya *close* tiket.
3. Dalam penelitian ini, peneliti tidak menerapkan sisi keamanan sistem monitoring pada Nagios maupun sistem *ticketing* pada osTicket. oleh karena itu penelitian selanjutnya dapat menyertakan pertimbangan sisi keamanan seperti penerapan HTTPS.

DAFTAR PUSTAKA

- [1] M. D. Haryanto, I. Riadi, "Analisis dan Optimalisasi Jaringan Menggunakan Teknik *Load Balancing*," Yogyakarta, 2014.
- [2] H. Amnur, et al., "Perancangan dan Implementasi *Network Monitoring System* Menggunakan Nagios dengan Email dan SMS *Alert*," Politeknik Negeri Padang, 2014.
- [3] H. Saptono, "*Network Monitoring System* dengan Nagios," 2008.
- [4] M. E. "Implementasi Nagios untuk Merancang *System Monitoring Client* Jaringan Komputer menggunakan Sms dan Email," Jakarta: Universitas Islam Syarif Hidayatullah, 2009.
- [5] J. P. Jumri, "Perancangan Sistem Monitoring Konsultasi Bimbingan Akademik Mahasiswa dengan Notifikasi *Realtime* berbasis *SMS Gateway*," Universitas Tanjungpura, 2012.
- [6] M. R. Zaini, "Implementasi Protokol SNMP untuk Jaringan," Univesitas Andalas, 2013.
- [7] Qoyyimah, "Rancang Bangun *Helpdesk Ticketing System* Studi Kasus PT Primus Indojaya," Jakarta: Universitas Syarif Hidayatullah, 2011.
- [8] M. S. Daulay, "Pengantar Jaringan Komputer," Yogyakarta: Andi, 2005.
- [9] M. Lutz, "*Programming Python*," *Fourth Edition ed.*, Sebastopo: O'Reilly Media, Inc., 2010.