

IMPLEMENTASI SISTEM MONITORING SERVER MENGGUNAKAN NAGIOS

Juli Yanto ¹⁾ Martin Ruswanda ²⁾

¹⁾Dosen Program Studi Teknik Informasi
Sekolah Tinggi Teknologi Informasi NIIT I-Tech
Jl. Asem 2 No. 22, Cipete- Jakarta Selatan
www.i-tech.ac.id
juliyanto@i-tech.ac.id

²⁾Mahasiswa Program Studi Teknik Informasi
Sekolah Tinggi Teknologi Informasi NIIT I-Tech
Jl.Asem 2 No. 22, Cipete- Jakarta Selatan
www.i-tech.ac.id
<http://www.i-tech.ac.id>

ABSTRAK

Sistem *monitoring* merupakan komponen penting yang diperlukan suatu sistem *datacenter*. Bagaimana jika suatu sistem *datacenter* tidak memiliki sistem *monitoring*? tentunya hal ini akan menjadi kendala bagi seorang *system administrator datacenter* dalam mengelola dan memonitoring *server - server* dalam suatu *datacenter*. *Nagios* merupakan *tools monitoring* yang sifatnya *open source*, yang memiliki fleksibilitas pada komponen - komponen yang akan dimonitoring.

Pada penelitian ini akan dirancang sebuah sistem *monitoring* menggunakan *tools nagios* dengan menerapkan fungsionalitas *email notification*, *automatic startup service*, dan *monitoring HTTP status code* pada suatu *website*. Pada sistem yang dirancang ini, proses *monitoring nagios* menggunakan metode *agent base*, dimana pada setiap *server* atau komputer yang akan dimonitoring akan diinstall *agent nagios* yang bertugas untuk memberikan informasi komponen - komponen yang dimonitoring.

Dengan adanya penelitian ini diharapkan dapat membantu *system administrator* dalam memonitoring *server - server* dalam suatu *datacenter* dan dapat membantu siapa saja yang ingin membangun sistem *monitoring* melalui jaringan lokal dengan menggunakan *tools nagios* dengan fungsionalitas *email notification*, fungsionalitas *automatic startup service*, dan fungsionalitas *monitoring HTTP status code* pada suatu *website*.

Kata kunci : *nagios, open source, monitoring, dan email notification.*

1 PENDAHULUAN

1.1 Latar Belakang

Sistem *monitoring server* merupakan salah satu aspek yang penting dan diperlukan dalam suatu *datacenter*. Dengan adanya suatu sistem *monitoring server* secara terpusat, seorang *system administrator* dapat mengetahui kondisi keseluruhan *server* dengan satu tampilan tanpa perlu melakukan proses *remote* satu per satu ke setiap *server*, untuk mengetahui kondisi keseluruhan *server - server* dalam suatu *datacenter*.

Idealnya suatu sistem *monitoring*, baik yang sifatnya *free* (gratis) atau pun bersifat berbayar, memiliki fungsionalitas untuk memonitoring *free space disk*, *memory usage*, *CPU usage*, dan *service* yang berjalan pada suatu *server*. Fungsionalitas -

fungsionalitas dari sistem *monitoring server* tersebut, didukung dengan sistem *email notification* yang dapat memberikan informasi berupa pesan *email* kepada *system administrator* apabila terjadi perubahan kondisi suatu *server* yang dimonitoring.

Nagios merupakan salah satu sistem *monitoring server* yang berbasiskan *open source* yang memiliki fungsionalitas - fungsionalitas yang dapat di *custom* sesuai dengan kebutuhan *system administrator*, selain dari itu lisensi *software nagios* berupa GPL (*General Public License*). PT. Duta Anggada Realty, Tbk merupakan perusahaan yang bergerak dibidang pembangunan *real estate* yang memiliki *datacenter* dengan akses *user* kurang lebih 500 *user*. Kondisi *datacenter* pada PT. Duta Anggada Realty, Tbk tersebut belum memiliki sistem

monitoring yang dapat memonitor kondisi keseluruhan *server*.

1.2 Rumusan Masalah

Terdapat beberapa rumusan masalah yang akan dibahas dalam penelitian yang berjudul “Implementasi Sistem Monitoring Server Menggunakan Nagios” ini, yaitu sebagai berikut:

1. Bagaimana memudahkan seorang system administrator dalam memonitoring kondisi keseluruhan server dalam suatu datacenter?
2. Bagaimana menerapkan sistem notifikasi berupa pesan email kepada system administrator apabila terjadi perubahan kondisi server yang dimonitoring?
3. Bagaimana menerapkan fungsionalitas untuk menjalankan suatu service aplikasi, apabila terdapat kondisi service aplikasi server tidak berjalan ?
4. Bagaimana menerapkan fungsionalitas untuk memonitoring suatu halaman website tertentu apabila terjadi error pada sisi halaman website ?

2 LANDASAN TEORI

2.1 Open Source

Open source adalah istilah yang digunakan untuk *software* yang membuka/membebaskan *source codenya* untuk dilihat oleh orang lain dan membiarkan orang lain mengetahui cara kerja *software* tersebut dan sekaligus memperbaiki kelemahan-kelemahan yang ada pada *software* tersebut. Dan yang menarik dan salah satu keunggulannya adalah bahwa *open source software* dapat diperoleh dan digunakan secara gratis tanpa perlu membayar lisensi. Biasanya orang mendapatkan *software* ini dari internet. Salah satu *open source software* yang terkenal yaitu *Linux*. (Sto, Mmta, 2008)

2.2 Linux

Linux merupakan sistem operasi yang sifatnya bebas (*free*) dan gratis. Bebas (*free*) dan gratis dalam artian bahwa *source code* untuk *linux* tersedia secara bebas (tanpa mengeluarkan biaya *license*), dimana setiap *programmer* atau *developer* dapat memodifikasi *source code* dari operating sistem *linux* tersebut secara bebas yang tentunya mengikuti aturan dari GNU GPL (*general public license*) sendiri. (Sto, Mmta, 2008)



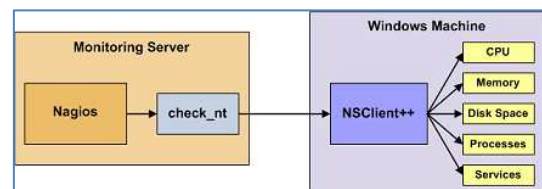
Gambar 2.1 Distribusi sistem operasi linux

2.3 Pengertian Nagios

Nagios merupakan *tool monitoring* yang berbasis *open source*, yang digunakan untuk melakukan proses *monitoring* terhadap *resource* atau sumber daya yang berada dalam suatu sistem. *Resource* atau sumber daya tersebut dapat berupa *host server* ataupun perangkat - perangkat *network* seperti *router*, *switch*, *firewall*, dan lain - lainnya. *Nagios* dapat melakukan proses *monitoring* dengan menggunakan komunikasi dengan *agent* yang diinstall pada komputer *host* atau dengan menggunakan protokol *SNMP* (*Simple Network Monitoring Protocol*), untuk perangkat yang tidak dapat diinstall *agent*. (Josephshen, David, 2010)

a. *Monitoring* dengan menggunakan distribusi *agent*

Pada proses *monitoring* yang menggunakan komunikasi *agent*, *nagios* memerlukan suatu *agent* yang diinstall pada komputer *client*. *NSClient++* merupakan komponen *agent* yang diinstall pada *host* atau *server* yang akan dimonitoring, *agent* tersebut bekerja pada *port TCP* 12489. *Server Nagios* akan melakukan *request* kepada *NSClient++*, untuk memberikan informasi mengenai komponen - komponen yang akan dimonitoring seperti *CPU usage*, *memory usage*, *disk space*, dan lain - lainnya. Dari *request nagios* tersebut *NSClient++* memberikan informasi status yang direquest oleh *nagios server*. Sebagai contoh *plugin check_nt* pada *nagios* melakukan *request* terhadap proses dengan nama *MsExchange*, *NSClient++* memberikan response mengenai status proses *MsExchange* apakah dalam kondisi *up* atau dalam kondisi *down*. (Josephshen, David, 2010)

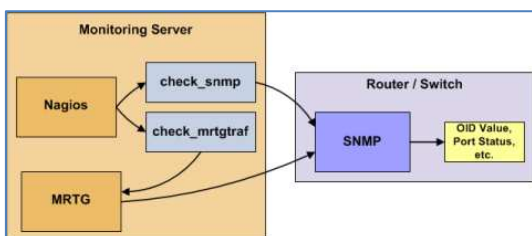


Gambar 2.2 Monitoring nagios dengan agent client

b. *Monitoring* dengan menggunakan protokol SNMP

Karena keterbatasan perangkat - perangkat network yang tidak dapat diinstall *agent nagios*, dibutuhkan protokol SNMP (*Simple Network Monitoring Protocol*) sebagai jembatan antara *nagios* dan perangkat *network* yang dimonitoring. SNMP (*Simple Network Monitoring Protocol*) merupakan standar protokol *monitoring*, yang umum digunakan untuk melakukan *monitoring* terhadap perangkat - perangkat *network* seperti *router*, *switch*, *firewall*, dan lain - lain.

Proses kerja dari monitoring dengan SNMP (*Simple Network Monitoring Protocol*) adalah *nagios* akan melakukan *request* pada *OID (Object Id)*, yang kemudian *OID (Object Id)* tersebut akan memberikan response berupa status yang sedang berjalan. (Josephshen, David, 2010)



Gambar 2.2 Monitoring nagios dengan menggunakan SNM

2.4 Mail Server

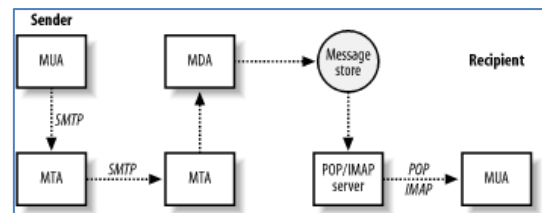
Mail server merupakan salah satu *internet protocol* yang bertugas sebagai penyedia layanan *e-mail*. *Mail server* bekerja dengan menggunakan protokol SMTP (*simple mail transfer protocol*), yang merupakan *protocol* yang dikhususkan untuk proses pengiriman email dengan menggunakan *port 25*. Selain dari *protocol* SMTP sendiri, terdapat beberapa *protocol* lain yang mendukung layanan *e-mail*, antara lain POP3 dan IMAP yang bertugas sebagai CAS (*client access server*) atau yang lebih dikenal dengan MDA (*mail delivery agent*). (Rafiudin, Rahmat, 2006)

Sistem *e-mail* biasanya melibatkan banyak sekali aplikasi ataupun program yang biasanya melaksanakan fungsi dan tugas-tugas tertentu dari sistem *mail server* itu sendiri. Fungsi-fungsi yang menjadi bagian dari *mail server*, yaitu MTA, MSA, MDA, dan MUA. MTA, MSA, dan MDA berada pada *server side*, sedangkan untuk MUA berada pada *client side*. Berikut penjelasan dari beberapa fungsionalitas server tersebut :

a. MTA atau *mail transfer agent* merupakan perangkat lunak yang mengirimkan *email* dari satu *host* ke *host* yang lainnya dengan menggunakan arsitektur *client server*. MTA menerima email yang berasal dari MTA lainnya, MSA, ataupun MUA.

b. MDA atau *mail delivery agent* merupakan perangkat lunak yang berfungsi untuk mengirimkan *email* ke *local mailbox* (kotal surat lokal).

c. MUA atau *mail user agent* merupakan perangkat lunak yang berfungsi untuk mengelola *email client*. Aplikasi ini merupakan aplikasi *client* yang dapat berhubungan dengan MTA, ataupun MSA untuk menerima ataupun mengirimkan email.



Gambar 2.3 Komponen penyusun mail server

3 ANALISIS DAN PERANCANGAN SISTEM

3.1 Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam pengerjaan proyek akhir berdasarkan referensi pada "*Ubuntu Server Guide*" perangkat keras yang dibutuhkan ini antara lain:

- Minimal *processor* 1 *gigahertz*.
- Minimal RAM 512 Mb.
- Minimal kapasitas *hard disk* 5 Gb .
- *Keyboard*

3.2 Kebutuhan Perangkat Lunak

Selain perangkat keras yang digunakan, dalam pengerjaan penelitian ini juga ada beberapa perangkat lunak yang dibutuhkan diantaranya:

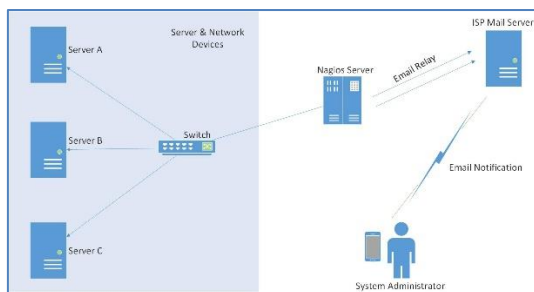
- *Sistem operasi linux*, digunakan untuk sistem operasi *nagios server*.
- *Nagios server*, digunakan sebagai aplikasi sistem monitoring *server*.
- *Web server*, digunakan untuk menjalankan sistem *nagios* agar dapat diakses dengan protokol HTTP.
- *Web browser*, untuk mengakses aplikasi *nagios*.
- *Console putty*, digunakan untuk melakukan *remote access* ke *server nagios*.

3.3 Kebutuhan Yang Harus Dipenuhi

Analisis kebutuhan merupakan suatu proses yang mengidentifikasi dan melakukan evaluasi terhadap permasalahan yang ada, sehingga dari identifikasi dan evaluasi tersebut dapat dibangun sistem yang sesuai dengan yang diharapkan. Pada sistem *monitoring server* yang akan dibangun ini ada beberapa analisis kebutuhan yang akan dibangun, antara lain :

1. Sistem *monitoring nagios* yang akan dibangun, melakukan proses *monitoring* untuk komponen - komponen *server* yang sifatnya penting seperti *CPU usage*, *memory usage*, *diskspace*, *port* dan *service* aplikasi yang berkaitan dengan aplikasi yang berjalan pada suatu *server*.
2. Sistem notifikasi berupa pesan *email* akan dikirimkan oleh *server nagios* kepada *system administrator* apabila terjadi kondisi perubahan pada suatu *server* yang dimonitoring. Untuk dapat mengirimkan pesan *email*, *server nagios* akan melakukan proses *relay email* kepada *mail server ISP* terlebih dahulu.
3. Pada sistem *monitoring nagios* akan diterapkan sistem *monitoring* dengan fungsionalitas *automatic startup*, yang merupakan suatu proses dimana *server nagios* akan melakukan suatu aksi apabila terjadi perubahan kondisi suatu *server*. Sebagai contoh apabila *service* pada suatu *server* tidak berjalan, sistem *monitoring nagios* akan mencoba menjalankan kembali *service* tersebut.
4. Pada sistem *monitoring nagios*, akan diterapkan fungsionalitas untuk melakukan *monitoring* pada suatu *website*, yang merupakan proses *monitoring* terhadap *HTTP status code* yang diberikan oleh *web server* kepada *web client*. Sehingga *system administrator* dapat mengetahui kondisi yang terjadi pada *website* yang dimonitoring.

3.4 Sistem Yang Akan Dibangun



Gambar 3.1 Desain sistem *nagios*

Sistem *monitoring server* yang berbasis *Nagios* merupakan sistem yang berbasiskan *open source*, yang digunakan untuk melakukan proses *monitoring resource* dari suatu *server* yang meliputi, *free space disk*, *memory usage*, *CPU usage*, dan *service* yang berjalan pada suatu *server*. *Nagios server* yang berperan sebagai *server monitoring* akan mengirimkan notifikasi berupa pesan *email* kepada *system administrator* apabila terjadi perubahan kondisi pada suatu *server* yang dimonitoring, seperti terjadi kondisi dimana *service* aplikasi yang berjalan pada suatu *server* tersebut

mati atau kondisi *disk space server* tersebut akan habis.

Untuk dapat mengirimkan suatu pesan *email* kepada *system administrator*, *server nagios* perlu melakukan proses *relay email* kepada *mail server ISP*, dalam studi kasus ini *mail server* yang digunakan adalah *mail server* dari *ISP*. Dengan diterapkannya sistem *monitoring server* ini, maka *system administrator* dapat mengetahui kondisi dari suatu *server* sebelum terjadi suatu *issue* pada suatu *server* yang digunakan oleh *user*.

a. Perancangan Nagios Sebagai Server Monitoring

Nagios server yang akan dibangun pada penelitian ini, digunakan untuk melakukan proses *monitoring* kondisi pada suatu *server* yang sifat nya penting, kondisi *server* tersebut antara lain:

- a. *CPU usage server*.
- b. *Memory usage server*.
- c. *Diskspace server*.
- d. *Application port server*.
- e. *Application service server*

Untuk dapat mendapatkan informasi kondisi dari suatu *server* tersebut, pada tiap-tiap *server* yang akan dimonitoring terlebih dahulu akan dilakukan proses *installasi agent nagios*, yang nantinya bertugas untuk mengumpulkan informasi dari *server*. Yang mana proses distribusi *agent* tersebut dilakukan *installasi* secara *manual* pada tiap-tiap *server* yang akan dimonitoring.

b. Perancangan Nagios Dengan Notifikasi Email

Agar *system administrator* dapat mengetahui kondisi suatu *server* kapanpun tanpa melakukan *access* ke *console nagios*, perlu nya ada suatu sistem notifikasi yang memberikan informasi kepada *system administrator* apabila terjadi perubahan kondisi dari *server* yang dimonitoring. Dalam penelitian ini, sistem notifikasi yang akan digunakan adalah pesan *email*. Untuk dapat mengirimkan pesan *email* tersebut sistem *nagios* akan melakukan proses *relay email* ke *mail server ISP*.

c. Perancangan Nagios Dengan Fungsi Automatic Startup Service

Selain menggunakan *secure sockets layer* untuk melakukan *enskripsi username* dan *password*, *FTP server* juga bisa melakukan transfer data yang aman dengan menggunakan *protokol secure shell* yang berbasiskan *remote access*. *FTP server* yang menggunakan *secure shell* sebagai jalur aman dalam melakukan transfer datanya, memerlukan *username* dan *password* yang berada dalam *operating system*. *SFTP (secure file transfer protokol)* merupakan penerapan *FTP server* dengan

menggunakan *secure shell* sebagai jalur transfer datanya.

d. Perancangan Nagios Dengan Fungsi Website Monitoring

Pada perancangan ini akan diterapkan fungsionalitas untuk melakukan *monitoring* pada suatu *website* tertentu untuk mengecek apakah terdapat *HTTP error* pada *website* yang dimonitoring, yang mana secara *default plugin* untuk melakukan *monitoring website* belum terdapat dalam *plugin nagios*. Untuk *plugin* yang akan diimplementasikan tersebut menggunakan bahasa pemrograman *php*.

4 IMPLEMENTASI DAN PENGUJIAN

4.1 Persiapan Instalasi

Dalam Pengerjaan penelitian ini ada beberapa perangkat keras, perangkat lunak, dan perangkat jaringan yang digunakan, berikut spesifikasi perangkat yang digunakan:

4.1.1 Perangkat Keras Yang Dipakai

Perangkat keras yang digunakan dalam penelitian berdasarkan referensi pada analisis kebutuhan minimal perangkat keras yang meliputi:

Tabel 4.1 Perangkat keras yang dipakai

No	Nama Perangkat	Spesifikasi Perangkat
1	Processcor	Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz
2	Memory RAM	Memory 4 GB
3	Hard Disk	Harddisk 100 GB
4	Type Operating System	System type 32 bit Operating Sistem

4.1.2 Perangkat Lunak Yang Dipakai

Untuk perangkat lunak yang digunakan sesuai analisis kebutuhan perangkat lunak adalah:

Tabel 4.2 Perangkat lunak yang dipakai

No	Nama Perangkat	Spesifikasi Perangkat
1	Operating System Server	Ubuntu Server 14.04 LTS
2	Monitoring Server	Nagios Core 3.5.1
3	Monitoring Client	Windows Server 2012 R2
4	Web Server	Apache 2.4.7
5	SMTP Server	Postfix 2.11.0

6	Web Browser	Chrome 48.0.2564
7	SSH Server	OpenSSH 6.6.1
8	PHP Programming	Php 5.5.9

4.1.3 Perangkat Jaringan Yang Dipakai

Selain perangkat keras dan perangkat lunak, perangkat jaringan yang digunakan yang sesuai dengan analisis kebutuhan perangkat jaringan adalah:

Tabel 4.3 Perangkat jaringan yang dipakai

No	Nama Perangkat	Spesifikasi Perangkat
1	Kabel UTP	Cat-5e
2	NIC	Intel(R) 82562GT

4.2 Instalasi Dan Konfigurasi

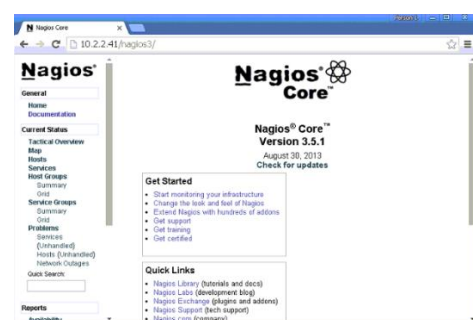
4.2.1 Instalasi Nagios

Langkah-langkah yang dilakukan dalam proses instalasi *nagios server* adalah sebagai berikut:

- Lakukan instalasi sistem operasi terlebih dahulu untuk komputer *server* dengan menggunakan sistem operasi *ubuntu server 14.04*.
- Langkah selanjutnya yaitu lakukan instalasi untuk *nagios server*, pada penelitian ini digunakan *nagios server core*.

```
root@dar-nagios:~# apt-get install nagios3
```
- Setelah proses instalasi paket selesai *restart* terlebih dahulu *service nagios* dengan langkah.

```
root@dar-nagios:~# /etc/init.d/nagios3 restart
```
- Lakukan pengecekan *nagios* pada *web browser* dengan URL http://alamat_server/nagios3



Gambar 4.1 Halaman *web page nagios*

4.2.2 Instalasi Postfix

Postfix mailserver digunakan sebagai SMTP server untuk melakukan proses pengiriman *email* dari *server nagios* kepada *email system administrator*.

- Instalasi *postfix* melalui *terminal*.

```
root@dar-nagios:~# apt-get install postfix
```
- Langkah selanjutnya adalah menjalankan *service postfix*



Gambar 4.2 Tes port SMTP postfix

4.2.3 Konfigurasi Nagios

- Agar dapat memonitoring komputer *client* yang sebelumnya telah diinstall *agent nagios*, terlebih dahulu perlu dibuat *object* komputer tersebut, lakukan dengan perintah *cp*.

```

root@dar-nagios:~# cp
/etc/nagios3/objects/win7-test.cfg
/etc/nagios3/objects/dar-mbx02.cfg

```

- Setelah dicopy *edit file object* tersebut.
- Edit* bagian *object* yang akan dimonitoring seperti gambar berikut ini :



Gambar 4.3 Edit object client nagios

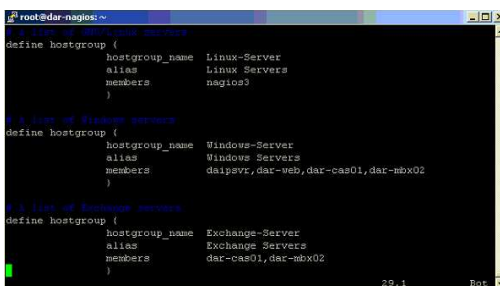
- Langkat selanjutnya *edit file hostgroup*, yang digunakan untuk melakukan *grouping* terhadap *object - object* yang dimonitoring oleh *nagios* dengan perintah :

```

root@dar-nagios:~# vi
/etc/nagios3/conf.d/hostgroups_nagios2.cfg

```

- Edit file hostgroup* sesuai dengan *grouping* yang digunakan



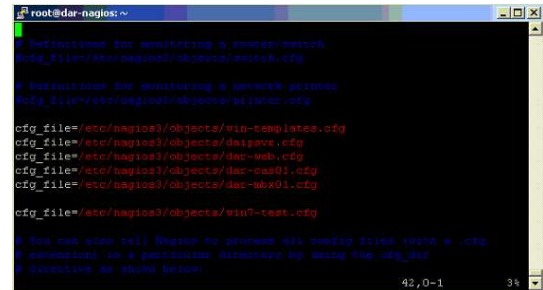
Gambar 4.4 Konfigurasi hostgroup nagios

- Apabila proses *edit hostgroup* telah selesai, *edit file nagios.cfg* untuk menambahkan *object* komputer yang sebelum nya telah di buat tersebut.

```

root@dar-nagios:~# etc/nagios3/nagios.cfg

```



Gambar 4.5 Konfigurasi file nagios.cfg

- Setelah semua *object* dari *client* yang akan dimonitoring selesai dibuat, verifikasi konfigurasi yang telah dibuat tersebut. Verifikasi dilakukan untuk melakukan pengecekan apakah terdapat *syntax* yang error atau tidak.

```

root@dar-nagios:~# nagios3 -v
/etc/nagios3/nagios.cfg

```

- Pastikan dalam proses verifikasi konfigurasi tidak terdapat error.

- Langkah terakhir, *restart service nagios*

```

root@dar-nagios:~# /etc/init.d/nagios3 restart

```

4.3 Pengujian

4.3.1 Pengujian Fungsionalitas Nagios Sebagai Server Monitoring

Tujuan pengujian:

Pengujian *nagios* sebagai *server monitoring*.

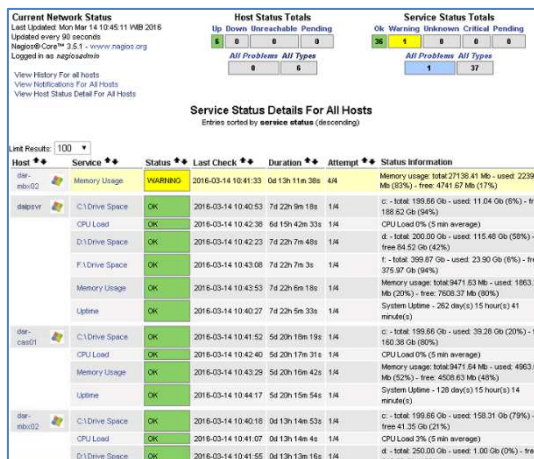
Yang akan diuji:

Interface nagios dalam menampilkan informasi *server* yang dimonitoring.

Parameter keberhasilan :

- Berhasil apabila *nagios* dapat menampilkan informasi *status service* dari *server - server* yang dimonitoring.
- Gagal apabila *nagios* tidak dapat menampilkan informasi *server-server* yang dimonitoring, atau terdapat *error* dengan jenis *unknown* dalam *console web nagios*.

Interface nagios service status detail, menampilkan informasi mengenai *service* secara detail dari *host* atau *server* yang dimonitoring, seperti nama *service* yang dimonitoring, *status service*, serta informasi dari *status service* tersebut.



Gambar 4.6 Interface nagios service status detail

Keterangan gambar :

Tabel 4.4 Keterangan nagios service status detail

Nama	Keterangan
Host	Nama <i>server</i> yang dimonitoring.
Service	<i>Service</i> yang dimonitoring dari suatu <i>server</i> .
Status	<i>Status service</i> yang dimonitoring.
Last Check	Pengecekan terakhir <i>nagios</i> terhadap suatu <i>service</i> .
Duration	<i>Uptime</i> dari status <i>service</i> selama tidak ada perubahan.
Attempt	Jumlah pengecekan <i>server nagios</i> terhadap suatu <i>service</i> .
Status Information	Informasi status dari <i>service</i> yang dimonitoring.

4.3.2 Pengujian Fungsionalitas Notifikasi Email

Tujuan pengujian:

Pengujian fungsionalitas *nagios* untuk sistem notifikasi *email*.

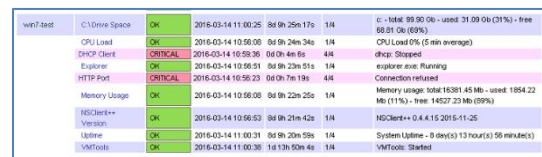
Yang akan diuji:

Proses pengiriman notifikasi *via email* apabila terjadi kondisi perubahan *service* suatu *server*.

Parameter keberhasilan :

- Berhasil apabila sistem *nagios* mengirimkan notifikasi *email*, apabila terjadi perubahan kondisi dari *service* atau *host* yang dimonitoring.
- Gagal apabila sistem *nagios* tidak mengirimkan notifikasi *email*, apabila terjadi perubahan kondisi dari *service* atau *host* yang dimonitoring.

Kondisi *service DHCP Client* dan *HTTP Port* dalam status *CRITICAL*.



Gambar 4.7 Service DHCP Client dan HTTP Port dalam status Critical

Nagios mengirimkan notifikasi *service HTTP Port* dalam status *CRITICAL*.



Gambar 4.8 Notifikasi HTTP Port dalam status CRITICAL

4.3.3 Pengujian Fungsionalitas Start-up Services

Tujuan pengujian:

Pengujian fungsionalitas *nagios* untuk proses *start-up service*.

Yang akan diuji:

Proses *start-up* service secara otomatis apabila terjadi kondisi dimana *service* yang dimonitoring terdapat kondisi *down*.

Parameter keberhasilan :

- Berhasil apabila *nagios* dapat menstart-up *service* yang dalam kondisi *down*.
- Gagal apabila *nagios* tidak dapat menstart-up *service* yang dalam kondisi *down*.

Service DHCP Client dalam kondisi *down*.



Gambar 4.9 Service DHCP Client dalam kondisi down

Nagios mentrigger *service DHCP Client* untuk *start-up*.



Gambar 4.10 Service DHCP Client dalam kondisi up

Service DHCP Client yang dimonitoring nagios dalam status OK.

init/Ans	C:\Drive Space	OK	2016-03-14 11:30:25	8d 9h 55m 57s	1/4	c: - total: 99.90 Gb - used: 31.08 Gb (31%) - free: 68.81 Gb (69%)
	CPU Load	OK	2016-03-14 11:31:08	8d 9h 55m 14s	1/4	CPU Load 0% (5 min average)
	DHCP Client	OK	2016-03-14 11:30:36	8d 9h 55m 46s	1/4	dhcpcd: Started
	Explorer	OK	2016-03-14 11:26:51	8d 9h 54m 31s	1/4	explorer.exe: Running
	HTTP Port	OK	2016-03-14 11:26:23	8d 9h 54m 58s	1/4	TCP OK - 0.000 second response time on port 80
	Memory Usage	OK	2016-03-14 11:31:08	8d 9h 53m 5s	1/4	Memory usage: total 16381.45 Mb - used: 2020.90 Mb (12%) - free: 14360.55 Mb (88%)
	NSClient++ Version	OK	2016-03-14 11:26:53	8d 9h 52m 22s	1/4	NSClient++ 0.4.4.15 2015-11-25
	Uptime	OK	2016-03-14 11:30:31	8d 9h 51m 29s	1/4	System Uptime - 9 day(s) 14 hour(s) 26 minute(s)
	VirtTools	OK	2016-03-14 11:30:38	1d 14h 20m 44s	1/4	VirtTools: Started

Gambar 4.11 DHCP Client yang dalam status OK

4.3.4 Pengujian Fungsionalitas Website Monitoring

Tujuan pengujian:

Pengujian fungsionalitas *nagios* untuk fungsionalitas *website monitoring*.

Yang akan diuji:

1.4 Tabel Hasil Pengujian

Dari hasil pengujian, dapat ditarik tabel hasil pengujian sebagai berikut :

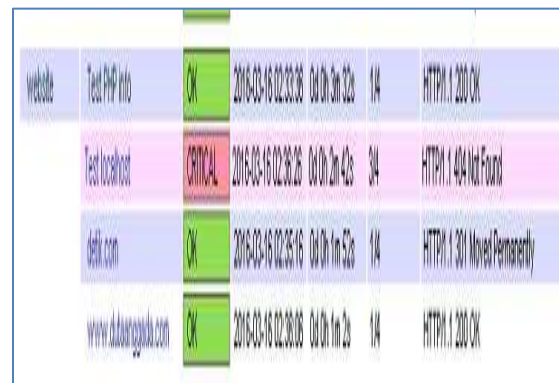
Tabel 4.5 Tabel hasil pengujian

NO	Fungsionalitas	Yang di Uji	Hasil Pengujian
1	Fungsionalitas <i>nagios</i> sebagai <i>server monitoring</i>	<i>Interface nagios</i> dalam menampilkan <i>host</i> dan <i>service</i> yang dimonitoring	<i>Nagios</i> menampilkan kondisi <i>host</i> dan <i>service</i> yang dimonitoring
2	Fungsionalitas notifikasi <i>email</i>	Proses notifikasi <i>email</i> apabila terjadi kondisi perubahan suatu <i>service</i>	<i>Nagios</i> mengirimkan notifikasi <i>email</i> apabila terjadi kondisi perubahan suatu <i>service</i> .
3	Fungsionalitas <i>start-up service</i>	Proses <i>start-up</i> otomatis apabila terdapat <i>service</i> yang <i>down</i> .	<i>Nagios</i> menstart-up <i>service</i> yang <i>down</i> , yang sebelumnya telah diterapkan fungsionalitas <i>start-up service</i> .
4	Fungsionalitas <i>website monitoring</i>	<i>Interface nagios</i> dalam menampilkan informasi <i>HTTP status code</i> , <i>website</i> yang dimonitoring	<i>Nagios</i> dapat menampilkan informasi <i>HTTP status code</i> dari <i>website</i> yang dimonitoring

Interface nagios dalam menampilkan *HTTP status code* dari *website* yang dimonitoring dan kondisi percabangan *if then else* dari *plugin* yang dibuat.

Parameter keberhasilan :

- Berhasil apabila *interface nagios* dapat menampilkan informasi *HTTP status code*
- dari *website* yang dimonitoring dan kondisi percabangan *if then else* dapat berjalan.
- Gagal apabila *interface nagios* tidak dapat menampilkan informasi *HTTP status code* dari *website* yang dimonitoring dan kondisi percabangan *if then else* tidak dapat berjalan.



Gambar 4.12 *Interface nagios* untuk fungsionalitas *website monitoring*

5 PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari implementasi penelitian ini adalah :

1. Sistem *monitoring nagios* dapat memonitoring kondisi *host* dan *service* suatu *server* secara terpusat dalam satu *console web*, sehingga memudahkan seorang *system administrator* dalam memonitoring kondisi keseluruhan *server*.
2. Fungsionalitas notifikasi *email* yang telah diterapkan, dapat membantu *system administrator* dalam mengetahui kondisi *server* yang dimonitoring, apabila terdapat kondisi dimana *system administrator* sedang tidak berada didepan computer.
3. Fungsionalitas *start-up service* secara otomatis apabila terdapat *service* yang *down* dapat berjalan dengan normal, sehingga *service* suatu *server* dapat tetap berjalan sebagai mana mestinya.
4. Fungsionalitas *website monitoring nagios* dalam menampilkan informasi *HTTP status code*, dapat berjalan normal. Dengan adanya *website monitoring* ini dapat membantu *system administrator* dalam memonitoring *website* - *website* yang dianggap penting.

5.2 Saran

Beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut :

1. Mengintegrasikan sistem monitoring nagios dengan sistem ticketing, agar setiap event yang terjadi dalam proses monitoring dapat terecord.
2. Sistem notifikasi nagios dapat dikembangkan lebih lanjut dengan sistem message chat, seperti Whatsapp, gtalk, dan sistem chat lainnya.
3. Monitoring nagios dapat diterapkan pada perangkat network seperti router, switch, dan lainnya dengan menggunakan protokol SNMP.
4. Untuk saran akademik pada penelitian ini dapat dikembangkan analisa untuk kebutuhan pembuatan aplikasi untuk manage Nagios.

6 DAFTAR PUSTAKA

- [1] Barth, Wolfgang. (2006). *Nagios: System and Network Monitoring 2nd Edition*. Germany.
- [2] Burgess, Chris. (2008). *The Nagios Book*.
- [3] Cartealy, Imam (2013). *Linux Networking*. Jakarta : Jasakom.
- [4] Eddy, Komala. (2009). *Rekayasa Perangkat Lunak*. Bandung
- [5] Dent, Kyle D. (2007). *Postfix: The Definitive Guide*.
- [6] Jogyanto. (2006). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi.
- [7] Josephshen, David. (2010) *.Building A Monitoring Infrastructure With Nagios*. Boston : Prentice Hall.
- [8] Mauro, Schmidt. (2007). *Essential SNMP, 2nd Edition*. USA : O'Reilly Media
- [9] M Syamsudin. (2014) *60 Menit Belajar Monitoring Jaringan*. Bukudigital.net.
- [10] Rafiudin, Rahmat. (2006). *Membangun Server E-mail Berbasis FreeBSD/Linux*, Jakarta : Andi Publisher.
- [11] S'to, mmta. (2008). *Fedora 8 Zero To Hero*. Jakarta : Jasakom.
- [12] Wibowo, Eri. (2011). *Ubuntu From Zero #2*. Jakarta : Jasakom.