

DESAIN, IMPLEMENTASI, DAN ANALISIS *NETWORK MANAGEMENT SYSTEM* (NMS) BERBASIS CACTI

DESIGN, IMPLEMENTATION, AND ANALYSIS OF *NETWORK MANAGEMENT SYSTEM* (NMS) BASED ON CACTI

Anggit Wulandoro¹, Fitriyani², Galih Nugraha Nurkahfi³

¹Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom

²Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom

³Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom

¹anggit.wulandoro@gmail.com, ²fitriyani.y@gmail.com, ³galihnugrahanurkahfi@gmail.com

Abstrak

Perkembangan infrastruktur jaringan saat ini sangat pesat, sehingga diperlukan adanya suatu manajemen perangkat jaringan untuk pengelolaannya. Untuk mengelola perangkat jaringan tersebut, dibutuhkan suatu sistem seperti *Network Management System* (NMS). NMS merupakan suatu sistem untuk mengelola perangkat-perangkat jaringan berbasis *Internet Protocol* (IP). Untuk menjalankan NMS, dibutuhkan suatu aturan atau protokol yang mendasari sistem tersebut, protokol tersebut adalah *Simple Network Management Protocol* (SNMP). Protokol SNMP memungkinkan administrator untuk mengelola dan mengatur kinerja perangkat jaringan secara jarak jauh. Pada tugas akhir ini akan diimplementasikan sistem pemantauan jaringan pada *cluster* yang terdapat di laboratorium *High Performance Computing* (HPC) Universitas Telkom. Implementasi NMS ini menggunakan sebuah aplikasi yaitu Cacti. Cacti merupakan aplikasi *open source* yang menyediakan sistem pemberitahuan melalui email, menampilkan hasil dalam bentuk grafik secara harian, mingguan, bulanan bahkan tahunan. Hasil penelitian ini menggambarkan keadaan sistem berdasarkan indikator seperti : *bandwith*, beban kerja prosesor, dan penggunaan memori.

Kata kunci : NMS, Cacti, SNMP.

Abstract

The growth of network infrastructure is very fast, therefore a network management system for its management. In order to manage those network system, a system such as *Network Management System* (NMS). NMS is a system that could be used to manage IP based network devices. In order to run NMS, a set of rules or protocol that work as the base of the system is essential, such protocol called *Simple Network Management Protocol* (SNMP). SNMP protocol give the administrator a capability to manage and set the entire network device performance remotely. This final project implements a network monitoring system in a cluster that located in *High Performance Computing* laboratory in Telkom University. The NMS implementation use Cacti. Cacti is an open source application that provide system notification via email, showing daily, weekly, monthly and annual monitoring graph. This research describe the system condition using various indicators like bandwidth, processor workload and memory usage.

Keywords: NMS, Cacti, SNMP.

1. Pendahuluan

Perkembangan teknologi dan infrastruktur jaringan di Indonesia saat ini sangat pesat. Dengan berkembangnya perangkat jaringan, dibutuhkan suatu manajemen perangkat jaringan yang baik, sehingga setiap perangkat dapat disesuaikan kebutuhannya. NMS mampu memantau kinerja dari perangkat-perangkat jaringan tersebut. *Network Management System* (NMS) sebagai sistem pemantau jaringan memberikan kemampuan untuk memantau dan mengelola perangkat-perangkat jaringan.

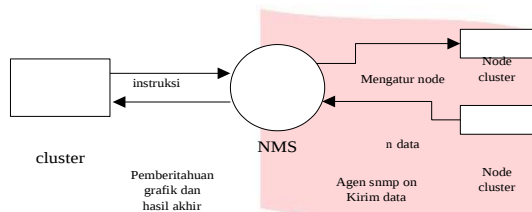
Sehingga diperlukan suatu aplikasi untuk memantau penggunaan sumber daya perangkat dan *bandwith* jaringan internet supaya besar ketersediaan

bandwith bisa memenuhi kebutuhan penggunaannya. Fungsi dari NMS adalah melakukan pemantauan terhadap kualitas *Service Level Agreement* (SLA) dari *bandwith* yang digunakan [1]. SLA adalah perjanjian yang disepakati antara penyedia layanan dengan pengguna dengan ruang lingkup untuk menentukan karakteristik dan kualitas layanan yang akan diberikan [2]. Cacti adalah salah satu aplikasi untuk memantau jaringan. Cacti cukup terkenal, karena sudah banyak penggunaannya. Dengan fitur untuk menampilkan grafik dan tambahan fitur lain, mudah digunakan dan gratis. Ada beberapa aplikasi untuk pemantauan jaringan, seperti : Zenoss, Nagios, Solarwinds, dan MRTG [3]. Hasil dari pemantauan jaringan akan dapat dianalisis oleh pengelola atau

administrator, apakah ada kejanggalan dalam jaringan atau tidak, dan juga untuk perencanaan pengembangan jaringan. Protokol yang digunakan NMS dengan Cacti adalah *Simple Network Management Protocol* (SNMP). Fungsi SNMP ini adalah memungkinkan pengelola dapat mengatur dan mengontrol elemen-elemen jaringan secara remote [4].

2. Metodologi dan Perancangan

Secara umum, sistem pemantauan jaringan adalah sebagai berikut :

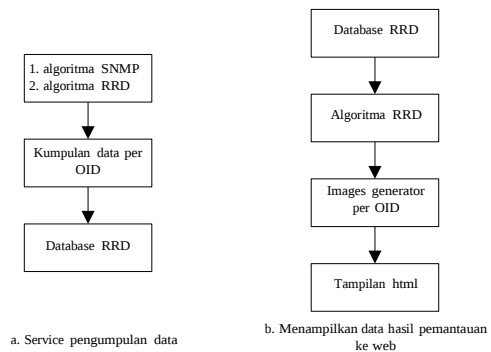


Gambar 2. Alur proses sistem

Head node (server) akan memberi instruksi ke NMS. Kemudian NMS meneruskan instruksi tersebut ke perangkat yang dipantau. Dari node cluster (sebagai agen) akan mengirimkan sumber data ke NMS, kemudian data tersebut dikumpulkan di dalam database RRD, dan akan diteruskan ke head node untuk ditampilkan datanya sebagai grafik pada tampilan web cacti.

2.1 Model komunikasi sistem

Sistem pemantauan ini menggunakan protokol SNMP. Dibutuhkan agen untuk memperoleh data yang dikumpulkan ke database dan menampilkan grafiknya menggunakan RRDTool, sebagai alat untuk menampilkan data pemantauan dalam bentuk grafik. Komunikasi antara SNMP, pengumpulan data, sampai tampilan grafik dapat dilihat pada gambar berikut :



Gambar 2.1. Komunikasi protokol SNMP dengan RRDtool

Algoritma SNMP “*pesan SNMP -v 2c + self._host + “-c” + self._password + OID*” Dengan menggunakan perintah *snmpwalk*, semua data dapat dibaca oleh SNMP servis pada head node (server). Kemudian, perintah *snmpgetnext* akan menampilkan nilai dari suatu OID pada node (klien/agen). Perintah *snmpgetnext* menampilkan nilai OID selanjutnya, dan kemudian *snmpstatus* untuk melihat status OID. -v 2c menunjukan versi SNMP yang digunakan. *Self._host* adalah alamat ip dari node (klien/agen). Dan *self._password* adalah community (otentikasi) pada klien.

Untuk algoritma RRD yang digunakan yaitu *create* dan *update*. *create rrdtool (rrd_name, item_type, interval) command create = “rrdtool create” + self.__RRD_DIRS_DATA + rrd_name + “.rrd” + “-step” + interval \ DS : item_name :item_type : interval*2 :u: RRA:AVERAGE:0.5:1* Perintah diatas adalah perintah untuk membuat database dengan nama *rrd_name.rrd*, dengan *Data Source (DS)* merupakan tipe dari suatu nilai data yang dinamakan *item_name* dengan *item_type* bergantung pada hasil data, standar interval waktunya 300 detik atau 5 menit.

RRD menyimpan data kedalam *Round Robin Archives (RRA)*, RRA berfungsi untuk menyimpan data dari semua sumber data yang telah didefinisikan. *Self.__RRD_DIRS_DATA* merupakan direktori tempat disimpannya database yang dihasilkan. Lokasinya di */data/syste/rrd*.

Untuk memperbarui data pemantauan yang masuk, dilakukan perintah *update Command update = “rrdtool update + self.__RRD_DIRS_DATA + rrd_name + .rrd N”*

Database di rrd diperbarui berdasarkan waktu sesungguhnya, sehingga akan didapat kumpulan data per OID yang dipantau. Data mentah dapat dilihat dengan perintah *fetch*. Data inilah yang kemudian dibuatkan grafiknya.

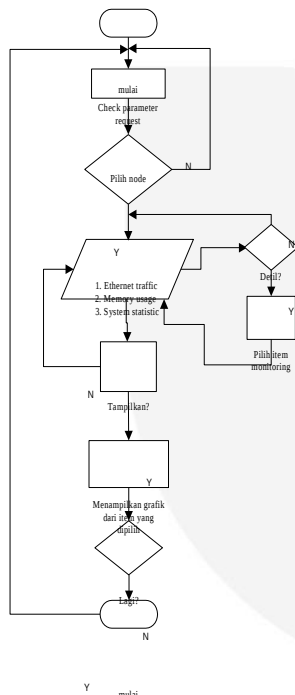
Kemudian, algoritma untuk menampilkan data ke dalam grafik pada tampilan web, algoritma untuk menampilkannya

Proses ini dibuat dalam bentuk grafik. Berikut algoritma untuk menampilkannya, Rrdtool graph (name_rrd) Command rrdtool graph : rrdtool graph + self._RRD_DIRS_IMAGE + rrd_name + .png + \

Masukan adalah data dari database rrd. Keluaran berupa grafik per OID yang diberi nama rrd_name.png. Perintah ini untuk menampilkan database rrd yang disimpan dalam bentuk grafik dan akan disimpan di self._RRD_DIRS_IMAGE berada di /images/rrd. Grafik akan ditampilkan data maksimum (MAX), minimum (MIN), dan rata-rata (AVERAGE). Keluaran grafik per OID inilah yang akan ditampilkan pada tampilan web.

2.2 Perancangan Sistem pemantauan jaringan dengan Cacti

Perancangan proses sistem pemantauan jaringan agar dapat diakses melalui web

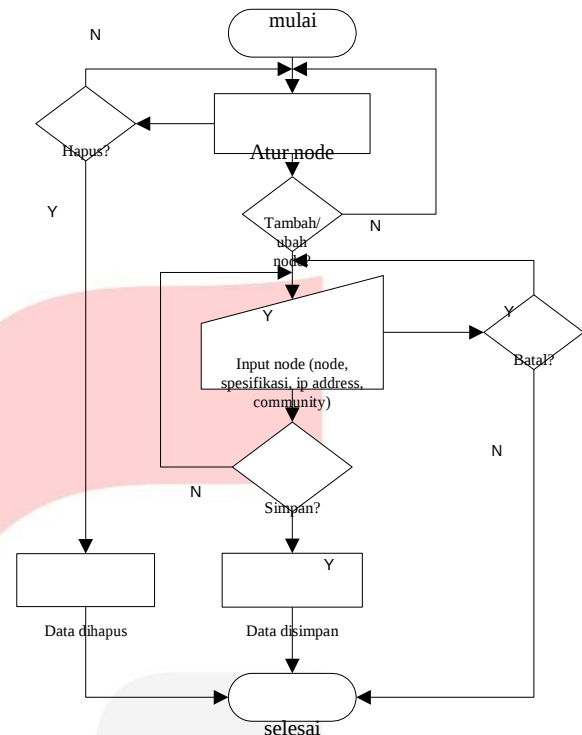


Gambar 2.2. Perancangan sistem pemantauan jaringan dengan Cacti

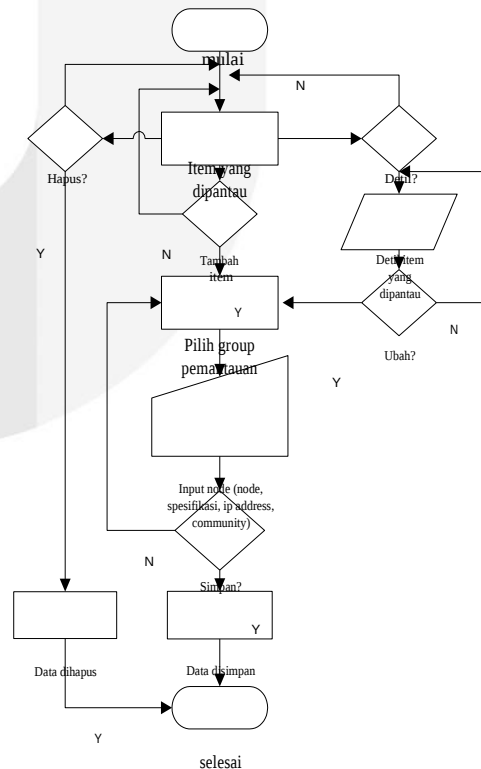
Pertama, dapatkan informasi perangkat yang akan dipantau dahulu. Informasi dapat berupa spesifikasi perangkat, alamat IP perangkat. Cek semua permintaan parameter. Kemudian pilih node yang akan dipantau. Untuk setiap node, ada tiga indikator yaitu *ethernet traffic*, penggunaan memori, statistik sistem (beban kerja CPU). Masing-masing indikator item dapat dipilih sendiri sesuai keinginan.

2.3 Proses manajemen cluster komputer di laboratorium komputasi kinerja tinggi

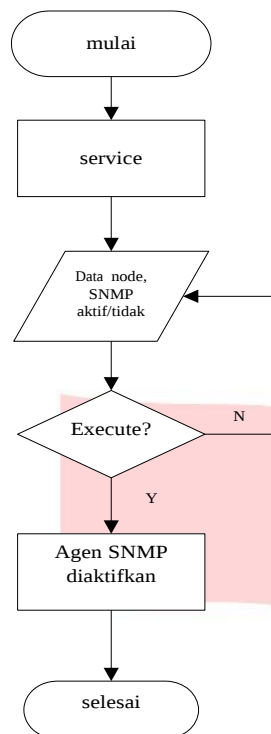
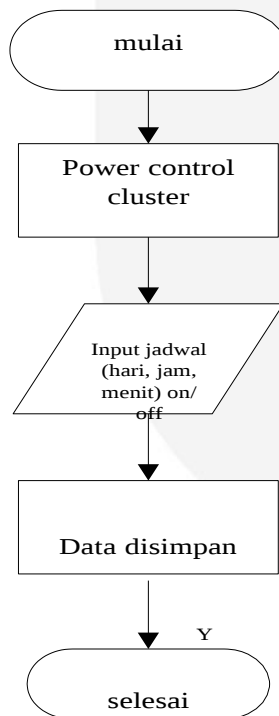
Proses ini terdiri dari pengaturan node dan item yang dipantau.



Gambar 2.3. Proses mengatur node



Gambar 2.3. Proses memantau node

Gambar 2.3. Proses *setting* servis SNMPGambar 2.3. *Power control*

Proses *atur node* dibuat untuk mempermudah ketika ada penambahan atau perubahan node pada *cluster*. Untuk menambah atau mengubah node, data yang diperlukan dari node yang ingin ditambah. Masukan yang dibutuhkan adalah nama node, spesifikasi, alamat ip, dan *community*. Jika semua

data masukan yang dimasukkan benar, maka akan tersimpan pada `self.__DIRS_DATA_SYSTEM + self.__NODES_FILE (/data/system/nodes.dat)`. Begitu juga dengan pengaturan item yang dipantau, prosesnya adalah data yang dibutuhkan untuk menambah atau mengubah item yang dipantau adalah nama group pemantauan, nama item, tipe, OID dan interval. Jika diisi dengan benar, maka akan tersimpan di `self.__DIRS_DATA_SYSTEM+self.__MONITORING_ITEM_FILE (/data/system/monitoring_item.dat)`.

2.4 Pengujian Sistem

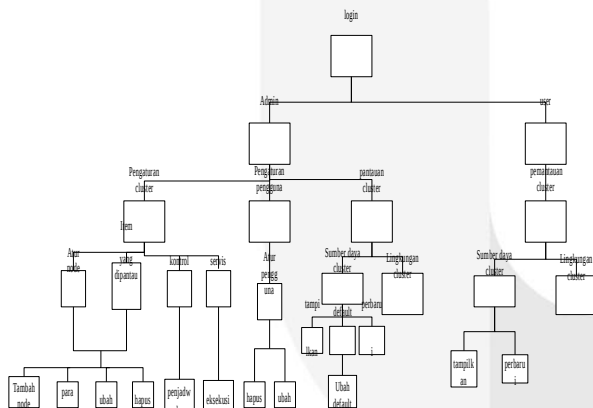
Pengujian sistem pemantauan ini dilakukan pada *cluster* komputer yang telah dibangun. Uji coba dilakukan pada komputer *cluster*. Komputer pertama sebagai head node yang juga sebagai server sistem pemantauan, dan komputer lainnya menjadi klien yang akan dipantau. Tetapi dalam kasus ini, saya membedakan kelas alamat IP yang akan dipantau. Alamat IP server 10.5.1.216, sedangkan untuk klien yang dipantau adalah 192.168.0.1. Karena ingin membedakan kelas alamat IP, maka perlu dilakukan konfigurasi dari alamat IP, dan pembatasan *broadcast* alamat IP, supaya proses untuk mengambil data menjadi lebih cepat. klien tadi terhubung ke sebuah *switch* di laboratorium komputasi kinerja tinggi. Oleh karena itu, server dipasang dengan menggunakan dua buah NIC (*Network Interface Card*) karena server akan mempunyai dua kelas alamat IP yang berbeda. Dari ethernet satu (eth1) di server, terhubung ke *switch*. Spesifikasi dari masing-masing klien yaitu :

Komputer	Spesifikasi	IP address
Head node	Intel Core i7-2600 3,4GHz, RAM 8GB, HDD 500GB, 100Mbps fast Ethernet adapter	10.5.1.216 192.168.0.1
Node 1	Intel Core i7-2600 3,4GHz, RAM 8GB, HDD 500GB, 100Mbps fast Ethernet adapter	192.168.0.2

Node 2	Intel Core i7-2600 3,4GHz, RAM 8GB, HDD 500GB, 100Mbps fast Ethernet adapter	192.168.0.3
Node 3	Intel Core i7-2600 3,4GHz, RAM 12GB, HDD 500GB, 100Mbps fast Ethernet adapter	192.168.0.4

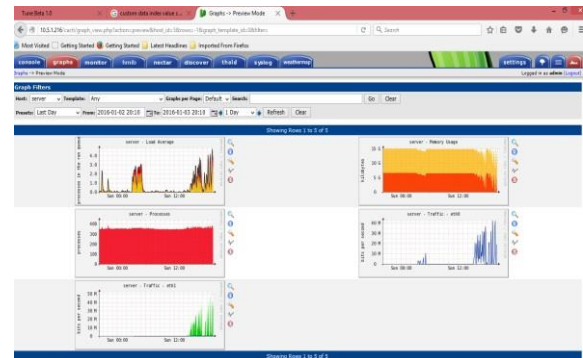
Gambar 2.4. Spesifikasi komputer cluster

2.4.1 Tampilan pengguna sistem pemantauan

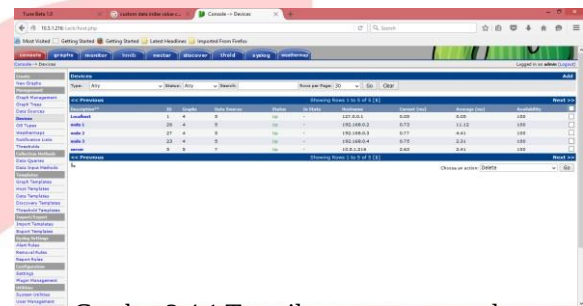


Gambar 2.4.1 Tampilan pengguna sistem pemantauan

Proses login pada diagram sistem diatas terdiri dari dua user, user pertama masuk sebagai admin, dan yang satunya masuk sebagai user biasa. Admin memiliki peranan penuh dalam sistem pemantauan. Admin memiliki semua akses terhadap sistem. Sedangkan user biasa hanya dapat melihat atau memonitor *traffic* untuk keperluan mereka. Tampilan untuk sistem pemantauan cacti seperti dibawah ini :



Gambar 2.4.1. Tampilan server cacti



Gambar 2.4.1 Tampilan pengaturan node

3. Analisa Sistem

3.1 Analisis Performansi Cacti

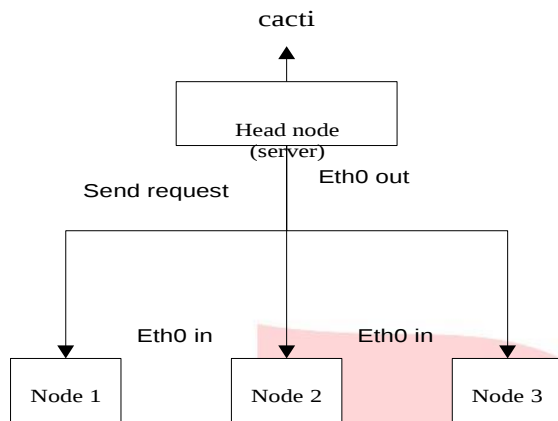
Analisis dari sistem pemantauan menggunakan cacti disini ditunjukkan dengan performansi selama proses pengumpulan data pemantauan, yaitu ketika servis pengumpulan data dari node-node yang dipantau ternyata tidak menggunakan sumber daya yang terlalu banyak. Hanya sedikit pemakaian sumber daya untuk menjalankan cacti. Tetapi ketika di server menjalankan program paralel, penggunaan sumber

daya meningkat. Dari penggunaan memori, beban kerja prosesor, dan *Ethernet traffic*. Ketika sistem pemantauan sedang dijalankan, sistem tersebut tidak mengganggu kinerja server.

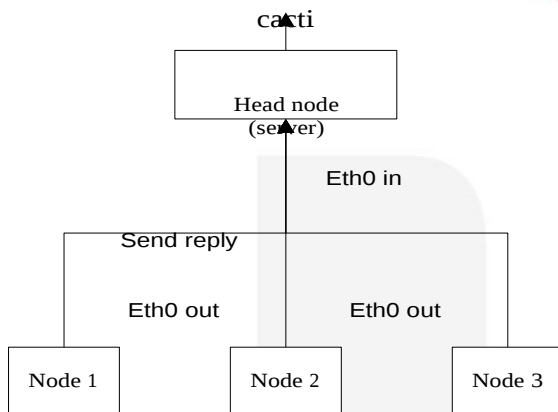
3.2 Aliran data pada cacti

Ilustrasi pengukuran performansi dilihat dari *Ethernet traffic* di head node (server). Ketika head node mengirim permintaan ke node dibawahnya, maka dari *ethernet* di head node mengirimkan paket (*ethernet out*). Kemudian pada setiap node yang dipantau, akan menerima permintaan dari head node tadi melalui paket (*ethernet in*). Setelah menerima permintaan dari head node, kemudian node akan mengirimkan kembali (*reply*) paket yang diminta oleh head node. Pada saat mengirimkan kembali, *ethernet* di masing-masing node mengirimkan paket keluar (*ethernet out*) dan head node akan menerima kembali pesan balasan tersebut untuk ditampung

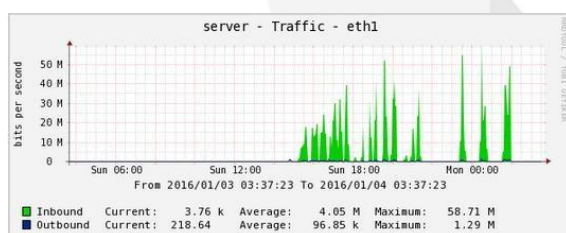
dalam database RRA, dan akan diproses untuk ditampilkan grafiknya pada tampilan web cacti.



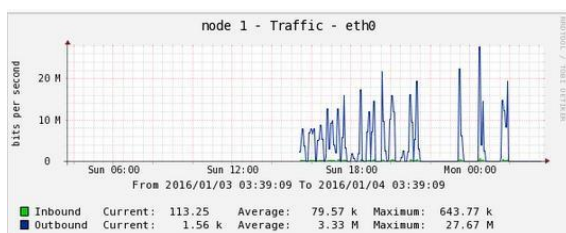
Gambar 3.2. Ilustrasi *traffic out* dari server



Gambar 3.2. Ilustrasi *traffic in* ke server

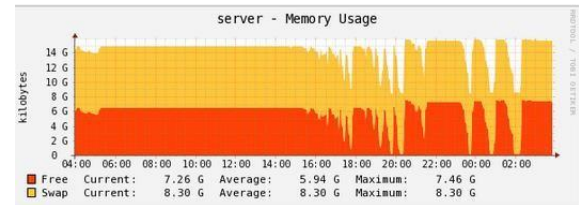


Gambar 3.2. *Traffic eth1* pada server



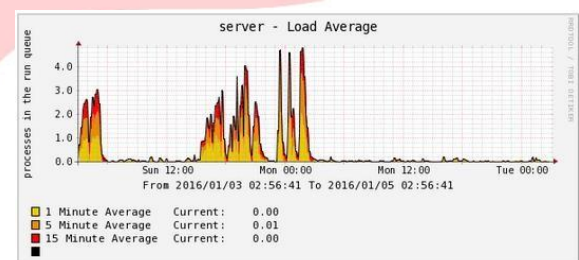
Gambar 3.2. *Traffic eth0* pada node 1

Penggunaan memori pada server pun dapat terlihat ketika sistem berjalan. Pada grafik dibawah, terdapat perbedaan penggunaan karena server digunakan untuk menjalankan program



Gambar 3.2. Grafik penggunaan memori pada cacti

Ketika pengguna memberikan instruksi untuk menjalankan suatu program yang cukup besar, maka terjadi peningkatan proses pada server dan node. Grafik dibawah ini berdasarkan pada proses yang dijalankan dan waktu tunggu untuk dijalankan.



Gambar 3.2. Grafik beban CPU pada cacti

3.3 Analisis RRDTools

Dengan aplikasi Round robin, data yang diambil berjumlah tetap. Ketika ada data baru yang masuk, data yang lama akan otomatis dibuang. Dengan cara tersebut, dataset tidak akan berkembang, jadi tidak memakan media penyimpanan. Data dapat tersimpan pada arsip Round Robin karena, sebelumnya telah menginstal RRD tools. Dalam aplikasi RRD terdapat arsip Round Robin (RRA) yang menjadi tempat terkumpulnya data. Pada arsip round robin, *counter* dibaca setiap 5 menit atau 300 detik, ini adalah pengaturan standar dari *counter*. Tetapi, hasil dari perhitungan tadi yang terbaca adalah rata-ratanya. Rrdtool bekerja dengan pengaturan waktu standar pada sistem. Data yang diperoleh dalam rentang waktu tertentu dibagi dengan satuan waktu pemantauan maka menghasilkan rata-rata kecepatan data. Setelah perhitungan kecepatan data, dengan hasil rata-ratanya, maka data akan ditampilkan dalam format .png sebagai grafik pada cacti. Warna standar untuk grafik adalah Red, Green, Blue (RGB). Warna-warna tersebut dibaca dengan standar bilangan hexadecimal.

Red = #FF0000

Green = #00FF00

Blue = #0000FF

Magenta = #FF00FF (mixed red with blue)

Gray = #555555 (one third of all components)

Black = 000000

White = FFFFFFFF (mixed of all)

4. Kesimpulan dan Saran

1. Sistem pemantauan menggunakan cacti telah terimplementasi pada *cluster* laboratorium komputasi kinerja tinggi Universitas Telkom.
2. Dengan membatasi *broadcast* alamat ip pada jaringan untuk masing-masing node, waktu yang dibutuhkan untuk komputasi pada cacti menjadi lebih cepat. Sehingga tidak membutuhkan proses lama untuk mengambil datanya. Grafik yang dimunculkan pun lebih baik.
3. Cacti tidak mempengaruhi kinerja server pada saat server sedang menjalankan program. Karena penggunaan sumber daya dari sistem pemantauan sedikit.

Daftar Pustaka:

- [1] Fachruddin, Farid. 2009. *Implementasi Sistem Monitoring SLA Bandwidth Dalam Aplikasi Cacti*, tersedia : <http://202.6.229.23/Download/artikel/cacti.pdf>
- [2] ITU-T Standardization E.860, *Framework of a Service Level Agreement*. (2002, June).
- [3] Web-based Automatic Network Discovery/Map System, (ICCAIE 2011)
- [4] Case, J. Fedor, M. Schoffstall, M. Davin, J. 1988. *A Simple Network Management Protocol*, tersedia : <http://www.faqs.org/rfcs/rfc1098.html>
- [5] A Scalable Architecture for Network Traffic Monitoring and Analysis Using Free Open Source Software, tersedia : <http://www.scirp.org/journal/PaperDownload.aspx?paperID=696>
- [6] Ipswitch. 2010. *The Value Of Network Monitoring*, tersedia : <http://www.whatsupgold.com/mailers/0809/valueofnetworkmanagement.pdf>
- [7] Management Information Base for Network Management of TCP/IP-based internets, tersedia : <https://www.ietf.org/rfc/rfc1213.txt>
- [8] A Simple Network Management Protocol (SNMP), tersedia : <https://www.ietf.org/rfc/rfc1157.txt>
- [9] Protokol SNMP, tersedia : <http://net.comlabs.itb.ac.id/blog/?p=557>
- [10] SNMPGET, tersedia : <http://www.net-snmp.org/docs/man/snmpget.html>
- [11] Object Identifier (OID), tersedia : <http://www.debianhelp.co.uk/linuxoids.htm>
- [12] rrdtool-round robin database tool, tersedia : <http://www.msg.ucsf.edu/local/rrdtool/html/rrdtool.html>
- [13] An Efficient Network Monitoring and Management System, *International Journal of Information and Electronics Engineering*, Vol. 3, No. 1, January 2013
- [14] Design and Implementation of the Campus Network Monitoring System, 2014 IEEE Workshop on Electronics, Computer and Applications
- [15] One approach to the development of custom SNMP agents and integration with management systems, MIPRO 2012, May 21-25, 2012, Opatija, Croatia