

Sistem Monitoring Server Dengan Menggunakan SNMP

Alehandrew Michael^{1,*}, Hendi Hermawan^{2,*}, Heny Ispur Pratiwi³

¹Program Studi Informatika, Universitas Pembangunan Jaya
alehandrewmichael@gmail.com

²Program Studi Informatika, Universitas Pembangunan Jaya
hendi.hermawan@upj.ac.id

³Program Studi Informatika, Universitas Pembangunan Jaya
heny.pratiwi@upj.ac.id

Received 31 August 2019, Revised 14 September 2019, Accepted 23 September 2019

Abstract — Servers are the main components in a computer networks system to provide services to the users or commonly known as the clients. The activities and operational services of a server to the client include many processes to fulfill the client requests sent to the server. A system to monitor all activities in the server is needed such that the administrator (operator) can monitor and detect if any problem occurs on the server. This system also applies server service livelihood automation for a server that is experiencing suddenly dead conditions. Programming languages used in building this system are python, php, and shell script. The reason of using Python is due to superior capabilities in connecting the Simple Network Management Protocol (SNMP). The SNMP is a network management protocol that regulates and stores all network information in it. The system is implemented by a web that provides information about the monitored server.

Keywords: Server, SNMP, Python, PHP, Shell Script

Abstrak — Server merupakan komponen utama sistem jaringan komputer yang berfungsi untuk menyediakan suatu layanan kepada pengguna yang biasa disebut sebagai *client*. Aktifitas dan operasional pelayanan sebuah *server* terhadap *client* dalam penerapannya terdiri dari begitu banyak proses untuk memenuhi segala permintaan *client* yang dikirimkan pada *server* tersebut. Diperlukan suatu sistem yang mampu melakukan pemantauan segala aktivitas di dalam *server* sehingga *administrator* (operator) bisa memonitor dan mendeteksi apabila terjadi permasalahan pada *server* yang digunakan. Sistem ini juga menerapkan otomatisasi penghidupan layanan server terhadap layanan *server* yang mengalami kondisi mati secara tiba-tiba. Bahasa pemrograman yang digunakan untuk membangun sistem ini adalah python, php, dan shell script. Alasan menggunakan bahasa pemrograman python pada sistem ini adalah memiliki kemampuan yang unggul dalam melakukan koneksi protokol SNMP (*Simple Network Management Protocol*). Hal ini dikarenakan protokol SNMP merupakan protokol manajemen jaringan yang mengatur dan menyimpan segala informasi jaringan di dalamnya. Hasil luaran dari sistem ini adalah sebuah web yang mampu mengeluarkan beberapa informasi mengenai *server* yang dipantau.

Kata Kunci: Server, SNMP, Python, PHP, Shell Script

PENDAHULUAN

Server adalah sebuah sistem komputer yang terdapat pada jaringan komputer untuk menyediakan suatu layanan kepada pengguna yang disebut sebagai *client*. Dibalik penggunaannya, *server* telah melakukan banyak proses untuk memenuhi permintaan dari *client*, oleh sebab itu sering kali *server* mengalami gangguan yang dikarenakan *server* tidak memiliki sumber daya (*resource*) yang mumpuni untuk memenuhi kebutuhan tersebut. Hal ini menyebabkan layanan *server* mati secara tiba-tiba dikarenakan kernel memutuskan untuk non-aktifkan layanan *server* yang membutuhkan *resource* yang besar. Kernel adalah komponen inti dari sistem operasi. Kernel bertanggung jawab untuk mengerjakan tugas-tugas tingkat rendah seperti

manajemen disk, manajemen tugas dan manajemen memori.

Permasalahan ini dapat teratasi dengan membangun sistem pemantauan *server* yang mampu mengumpulkan beberapa informasi *server* seperti IS (*information system*), informasi sumber daya sistem (*resource*), dan informasi *ethernet interface*. Kebutuhan yang diperlukan dalam membangun sistem ini adalah protokol SNMP (*Simple Network Management Protocol*) yang merupakan suatu protokol yang digunakan untuk mengolah dan memonitor perangkat jaringan. SNMP menggunakan sistem log yang menyimpan segala informasi dari perangkat jaringan yang terdaftar. Sehingga keuntungan dari menggunakan SNMP dapat mengambil segala kebutuhan informasi perangkat

yang digunakan dan dapat ditampilkan untuk memenuhi kebutuhan sistem *monitoring*.

Penelitian ini bertujuan untuk menghasilkan sebuah aplikasi *monitoring server* berbasis web agar mendukung pemantauan *server* secara *remote*. Selain itu web dapat diakses menggunakan *smartphone*, laptop, pc, dan device pintar lainnya. Protokol SNMP digunakan pada sistem ini untuk menerima informasi seperti *resource server*, jumlah paket yang diterima dan dikirim, informasi status, dan notifikasi yang berisi status dan waktu pada layanan *server* yang dipantau. Lingkup jaringan sistem yang dibangun mengadaptasi dari jaringan virtual yang kerap kali digunakan dalam melakukan pembangunan VPS (*Virtual private server*). Keunggulan dari sistem ini adalah mampu melakukan otomatisasi penghidupan layanan *server* menggunakan shell scripts sebagai basis perintah yang terdapat pada sistem operasi linux. Layanan *server* yang menjadi subjek dari fitur otomatisasi penghidupan antara lain adalah apache, mysql, dan ftp *server*.

KAJIAN PUSTAKA

Harrington, Presuhn, dan Wijnen (2002) memaparkan standar yang terdapat pada protokol SNMP. Standar tersebut meliputi konsep dan cara kerja SNMP yang digunakan setiap versinya. Nugroho, Afandi, dan Rahardjo (2014) melakukan perancangan aplikasi *monitoring* jaringan menggunakan protokol SNMP. Aplikasi *monitoring* yang dibangun bukan hanya melakukan *monitoring* jaringan saja, namun aplikasi dapat melakukan *mapping* terhadap jumlah *device* yang terdapat didalamnya. Selain itu, aplikasi ini juga dikembangkan untuk melakukan sistem peringatan dini terhadap perangkat jaringan yang *dimonitoring*. Taftazanie, Prasetyo, dan Widiyanto (2017) menjelaskan aplikasi pemantau jaringan berbasis web dengan menggunakan protokol SNMP dan notifikasi sms. SNMP digunakan untuk mengambil informasi dari perangkat pemantau jaringan dan notifikasi sms yang dibangun dikerjakan dengan protokol *syslog*. Liu dan Lu (2012) memaparkan aplikasi *monitoring* jaringan menggunakan protokol SNMP. SNMP digunakan untuk mengambil informasi *CPU usage rate*, *memory usage*, dan *traffic flow* yang ada pada perangkat jaringan yang dipantau.

METODOLOGI PENELITIAN

Tahapan penelitian ini dilakukan secara berurutan yang terdiri dari sebagai berikut :

- observasi literatur
- Rancangan desain aplikasi & topologi jaringan
- Instalasi sistem operasi *server*
- Implementasi arsitektur jaringan
- Instalasi & konfigurasi *software* pendukung
- Implementasi sistem
- Pengujian sistem
- Kesimpulan

Pada tahapan observasi literatur dilakukan pencarian terhadap teori dan jurnal terkait dengan penelitian yang sedang dikerjakan. Hal ini diperlukan untuk menemukan dasar dari penelitian yang dikerjakan.

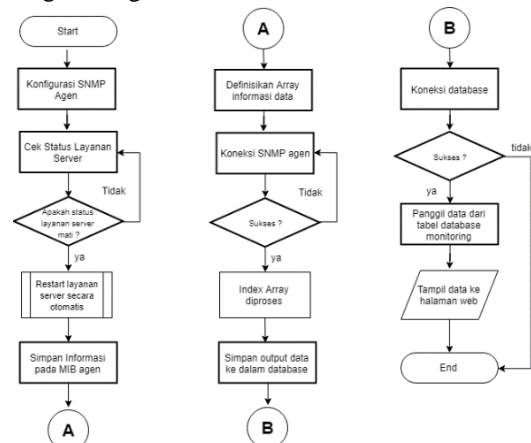
Selanjutnya memasuki tahapan perancangan desain aplikasi dan topologi jaringan untuk memberikan gambaran bentuk aplikasi yang akan dibangun serta kebutuhan jaringan sistem *monitoring server*.

Setelah menyusun desain topologi jaringan maka selanjutnya memasuki tahapan instalasi sistem operasi *server*. Pada tahapan ini perangkat *virtual* akan dipasang sistem operasi yakni Ubuntu 16.04 Desktop untuk manajer dan Ubuntu 16.04.5 *Server* untuk *agent*. *Agent* dan *manager* merupakan sebutan untuk membedakan deskripsi pekerjaan dalam menjalankan sistem monitoring *server*. Definisi dari peran *manager* adalah melakukan pengambilan informasi dari data *server* yang dipantau, sedangkan definisi dari peran *agent* adalah melakukan pengiriman data informasi dari *server* yang dipantau ke *server* yang melakukan pemantauan (manajer). Di bawah ini merupakan spesifikasi komputer yang merupakan perangkat keras yang menampung semua *resource* perangkat *virtual* didalamnya.

Tabel 1. Spesifikasi Komputer

| Komponen | Jenis Perangkat |
|----------|-----------------------------|
| Prosesor | Intel i5-7200U CPU @2.50GHz |
| RAM | 8 GB |
| HDD | 1 TB |

Setelah *server* untuk manajer dan agen siap digunakan maka selanjutnya memasuki tahapan implementasi sistem jaringan. Pada tahapan ini, lingkungan jaringan yang dibangun akan mengadaptasi jaringan *virtual* sehingga memerlukan *software* pendukung untuk pembangunan sistem jaringan tersebut. Setelah implementasi jaringan selesai dikerjakan maka akan dilakukan pengujian konektivitas antar *server* dengan melakukan *ping* ke masing-masing *server*.



Gambar 1. Flowchart Sistem Monitoring Server

Jika semua *ping* ke *server* yang akan dimonitor berhasil, maka selanjutnya memasuki tahapan instalasi *software* pendukung untuk menyiapkan sistem *monitoring server*. *Software* tersebut terdiri dari SNMP, *Text Editor*, Python, dan XAMPP yang akan digunakan dalam melakukan implementasi sistem *monitoring server*.

Setelah kebutuhan *software* pendukung terpenuhi maka selanjutnya memasuki tahapan implementasi sistem. Pada tahapan ini dibagi menjadi empat bagian yaitu sebagai berikut:

- Implementasi sistem pengambilan nilai SNMP
- Implementasi sistem penyimpanan nilai SNMP ke dalam *database*
- Implementasi sistem menampilkan data dari *database*
- Implementasi sistem notifikasi waktu dari status layanan *server*
- Implementasi sistem otomatisasi penghidupan layanan *server*

Pada bagian implementasi pertama dilakukan dengan cara membuat program otomatisasi layanan *server*, hal ini dikarenakan SNMP akan melakukan pemanggilan data setiap menit untuk memproses informasi *server* yang dikonfigurasi. Setelah proses pembuatan program otomatisasi layanan *server* dinyatakan berhasil maka selanjutnya memasuki tahapan implementasi proses definisi *array*. Pada tahapan ini dibuatlah sebuah program yang menyimpan sebuah *array* berisi *index* informasi *server* yang akan dieksekusi oleh SNMP. Setelah *array* dieksekusi dan data informasi yang dipanggil dapat disimpan pada *database* maka tahapan selanjutnya melakukan implementasi pemanggilan data yang telah disimpan pada *database* sebelumnya. Data yang telah dipanggil akan diproses kembali untuk ditampilkan pada halaman *monitoring* dan notifikasi di halaman web sistem *monitoring server*. Setelah keseluruhan proses implementasi dilakukan maka tahapan pengujian dilakukan untuk mengetahui tingkat keberhasilan dari sistem yang dibangun. Setelah hasil pengujian diketahui maka dibuatlah beberapa kesimpulan sebagai evaluasi yang dicapai dari penelitian ini.

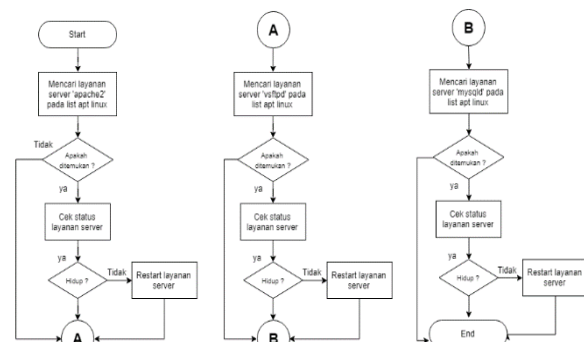
HASIL DAN PEMBAHASAN

Pengujian sistem dilakukan untuk mengetahui fungsionalitas dari setiap fitur yang dibangun. Seperti yang disebutkan sebelumnya, pengujian dilakukan dengan menggunakan metode *white box* dan *black box*. Agar pengujian yang dilakukan memiliki dasar yang dapat diikuti pada penelitian ini maka di bawah ini merupakan gambar sebuah *flowchart* yang mempresentasikan cara kerja sistem yang seharusnya sebelum dilakukan pengujian.

Flowchart yang terdapat pada Gambar 1 dimulai dari pembuatan program otomatisasi penghidupan layanan *server*. Hal ini dikarenakan luaran dari program ini adalah informasi status layanan *server* yang datanya diambil setiap menitnya oleh SNMP.

Proses di atas berawal dari konfigurasi SNMP agen yang merupakan proses pengaturan SNMP pada perangkat *server* agen.

Setelah itu program melakukan pengecekan status layanan *server* seperti apache, mysql, dan ftp. Jika terdapat layanan *server* mati maka program melakukan *restart* secara otomatis pada layanan *server* yang mati tersebut dan informasi status layanan *server* disimpan pada MIB (*Management Information Base*) *agent*. Proses yang ditunjukkan pada *restart* layanan *server* merupakan sub proses dari alur logika otomatisasi penghidupan layanan *server*. Sub proses *restart* layanan *server* ditunjukkan pada Gambar 2.



Gambar 2. Flowchart Proses Restart Layanan Server

Pada proses ini adalah akhir dari alur logika program otomatisasi penghidupan layanan *server* dan proses selanjutnya adalah alur logika program pengambilan informasi data *monitoring* dari SNMP agen. Program tersebut ditanam pada perangkat admin atau manajer sehingga diperlukan koneksi yang menghubungkan SNMP manajer dan SNMP agen.

Awal dari proses ini adalah melakukan definisi *array* yang berisi informasi data yang diperlukan untuk dipanggil dari SNMP agen. Selanjutnya program melakukan koneksi SNMP untuk menyesuaikan konfigurasi SNMP yang dilakukan pada alur proses pertama agar data yang didapatkan memang berasal dari perangkat *server* agen. Jika koneksi SNMP agen berhasil maka *index array* yang didefinisikan sebelumnya diproses dan hasil proses tersebut adalah melakukan pencocokan *index array* dengan nilai *Object Identifier* (OID) agar dapat memanggil data informasi yang berasal dari *Management Information Base* (MIB). Setelah data informasi didapatkan maka selanjutnya program melakukan penyimpanan data tersebut ke dalam *database*. Proses ini adalah akhir dari alur logika pemrograman pengambilan data *monitoring server* dan selanjutnya adalah proses alur logika dari pemrograman web sistem *monitoring server*.

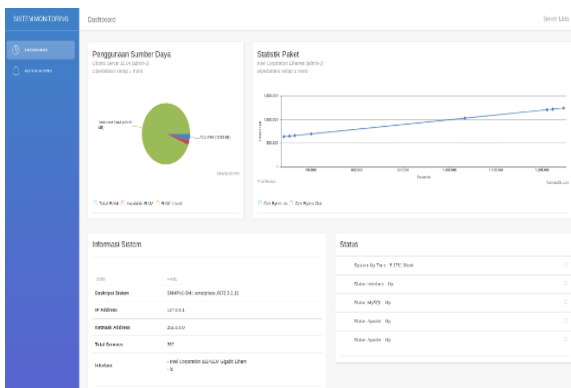
Awal dari proses ini dimulai dari melakukan koneksi *database* dengan mendefinisikan *user* dan *port* yang digunakan oleh *database*. Jika koneksi *database* berhasil maka proses pemanggilan data dari *database* dapat dilakukan. Data dipanggil dan ditampilkan pada konten halaman web yang sudah diimplementasikan sebelumnya supaya data yang

ditampilkan sudah siap disaji dalam bentuk yang lebih baik dan informatif. Hal ini merupakan proses akhir dari alur logika pemrograman web. Berdasarkan *flowchart* diatas berikut merupakan pengujian secara *blackbox*.

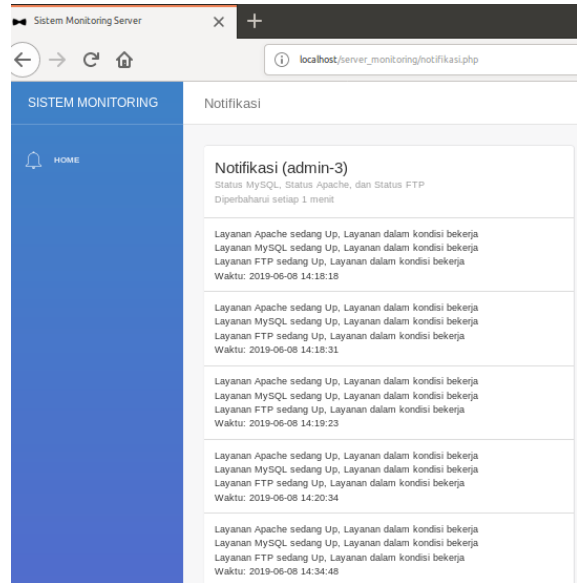
Tabel 2. Pengujian *Blackbox*

| No | Skenario Pengujian | Hasil yang diharapkan | Keterangan |
|----|--|--|------------|
| 1 | Program dapat melakukan koneksi dengan protokol SNMP | Program dapat mengeksekusi perintah pada protokol SNMPv2 | √ |
| 2 | Program dapat mengambil data <i>monitoring</i> | Program berhasil mengambil seluruh data <i>monitoring</i> | √ |
| 3 | Program dapat melakukan penyimpanan data <i>monitoring</i> ke dalam <i>database</i> | Program berhasil menyimpan seluruh data <i>monitoring</i> ke dalam <i>database</i> | √ |
| 4 | Program dapat menampilkan data <i>monitoring</i> yang terdapat di <i>database</i> ke dalam halaman web | Program berhasil memanggil seluruh data <i>monitoring</i> dari <i>database</i> dan menampilkan ke dalam halaman web | √ |
| 5 | Program dapat menampilkan notifikasi waktu dari status layanan <i>server</i> | Program berhasil memperbaharui informasi status layanan <i>server</i> dengan menggunakan waktu pada setiap menitnya | √ |
| 6 | Program dapat melakukan otomatisasi penghidupan layanan <i>server</i> <i>mysql</i> , <i>apache</i> , dan <i>ftp server</i> | Program dapat melakukan otomatisasi penghidupan layanan <i>server</i> <i>mysql</i> , <i>apache</i> , dan <i>ftp server</i> | √ |

Gambar 3 dan Gambar 4 merupakan tampilan akhir dari sistem *monitoring server*.



Gambar 3. Tampilan Halaman Utama Sistem *Monitoring Server*



Gambar 4. Tampilan Halaman Notifikasi Sistem *Monitoring Server*

KESIMPULAN

Aplikasi sistem *monitoring server* berhasil memberikan informasi mengenai aktivitas *server* yang dipantau. Informasi seperti sistem *server*, *resource*, dan statistik paket sangat bermanfaat dalam memantau konektivitas jaringan *server* dan pemeliharaan memori *server*. Sistem otomatisasi juga bekerja dengan baik apabila terdapat layanan *server* yang mati secara tiba-tiba. Hal ini untuk mengurangi penghidupan secara manual.

DAFTAR PUSTAKA

Harrington, D., Presuhn, R. & Wijnen, B., (2002). An architecture for describing simple network management protocol (SNMP) management frameworks (No. RFC 3411).

Nugroho, M., Affandi, A. & Rahardjo, D. S., (2014). Rancang Bangun Aplikasi Monitoring Jaringan Menggunakan SNMP (Simple Network Management Protokol) dengan Sistem Peringatan Dini dan Mapping Jaringan. *Jurnal Teknik Pomits*, 3(1), 35-39.

Taftazanie, S., Prasetijo, A. B. & Widiyanto, E. D., (2017). Aplikasi Pemantau Perangkat Jaringan Berbasis Web Menggunakan Protokol SNMP dan Notifikasi SMS. *Jurnal Teknologi dan Sistem Komputer*, 5(2), 62-68.

Liu, R. J. & Lu, R. (2012). Monitoring Network through SNMP-based System. *International Journal of Intelligent Engineering and Systems*, 5, 1-10.