Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)
Edisi. 1 Volume. 1 Februari 2016 ISSN : 2089-9033

PEMBANGUNAN APLIKASI SISTEM MONITORING JARINGAN MENGGUNAKAN OPENNMS BERBASIS SMARTPHONE ANDROID (STUDI KASUS PT. SKYLINE SEMESTA)

Muhamad Usman Rusmana

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : man_33us@yahoo.co.id

ABSTRAK

PT. Skyline Semesta merupakan sebuah perusahaan yang bergerak dibidang teknologi informasi dan komunikasi, salah satu layanan yang diberikan adalah *Internet Service Provider* (ISP). Aplikasi sistem monitoring The Dude merupakan aplikasi berbasis desktop, sehingga monitoring hanya dapat dilakukan pada ruang kerja. Administrator tidak dapat memantau jaringan didepan layar komputer secara terus menerus.

Sistem monitoring jaringan menggambarkan bagaimana suatu sistem dapat memantau kondisi jaringan baik itu *server*, *router*, maupun *device* jaringan lainnya yang terhubung ke jaringan dapat dimonitoring secara konstan. Aplikasi Sistem Monitoring Jaringan Berbasis Android adalah sebuah aplikasi yang memanfaatkan OpenNMS sebagai *Network Monitoring System*. Data hasil monitoring akan diakses oleh aplikasi ini. Aplikasi ini diharapkan dapat membantu Administrator dalam memonitoring jaringan yang dikelolanya dan ketika terjadi masalah pada jaringan dapat ditangani dengan segera.

Berdasarkan hasil yang didapat dari penelitian yang telah dilakukan maka dapat diperoleh kesimpulan bahwa aplikasi ini dapat membantu Administrator dalam memonitoring jaringan ketika berada di ruang kerja dan diluar ruang kerja. Adanya fasilitas *remote* memudahkan Administrator dalam melakukan konfigurasi perangkat secara jarak jauh.

Kata kunci : Sistem monitoring, jaringan, OpenNMS, *remote*, android.

1. PENDAHULUAN

Sistem monitoring jaringan menggambarkan bagaimana suatu sistem dapat memantau kondisi jaringan baik itu *server*, *router*, maupun *device* jaringan lainnya yang terhubung ke jaringan dapat dimonitoring secara konstan. Setiap perubahan-perubahan yang terjadi di *server* dapat dipantau melalui sistem monitoring. Jika pada jaringan terdapat perangkat yang tidak berfungsi ataupun terjadi masalah pada jaringan, maka sistem akan memberikan notifikasi kepada seorang Administrator berupa alarm.

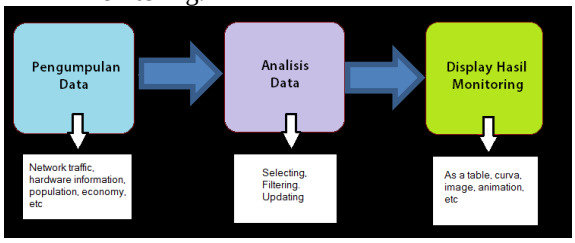
Aplikasi sistem monitoring The Dude masih berbasis *desktop* sehingga memerlukan sebuah komputer untuk melihat hasil monitoring jaringan. Masalah timbul ketika seorang Administrator tidak dapat memantau kondisi jaringan selama 24 jam penuh. Seperti kita ketahui bahwa Administrator juga manusia yang membutuhkan istirahat dan tidak selalu berada pada ruang kerja untuk melakukan aktivitas lainnya sehingga tidak dapat memantau didepan layar komputer secara terus menerus.

Memperhatikan beberapa permasalahan tersebut dan setelah melakukan wawancara dengan Administrator PT. Skyline Semesta semua Administrator jaringan menggunakan *smartphone* android, maka aplikasi sistem monitoring jaringan berbasis android dirasa dapat menjadi sebuah solusi untuk permasalahan ini. OpenNMS akan digunakan sebagai *Network Monitoring System*, data hasil monitoring dapat diakses oleh platform android. Hasil dari penelitian ini adalah Pembuatan sebuah **Aplikasi Sistem Monitoring Jaringan Menggunakan OpenNMS Berbasis *Smartphone* Android.**

1.1 Sistem Monitoring

Sistem monitoring merupakan suatu proses untuk mengumpulkan data dari berbagai sumber daya. Biasanya data yang dikumpulkan merupakan data yang *real time*. Secara garis besar tahapan dalam sebuah sistem monitoring terbagi ke dalam tiga proses besar seperti yang terlihat pada gambar 1, yaitu:

1. Proses di dalam pengumpulan data monitoring.
2. Proses di dalam analisis data monitoring.
3. Proses di dalam menampilkan data hasil monitoring.



Gambar 1. Proses Sistem Monitoring

Aksi yang terjadi di antara proses-proses dalam sebuah sistem monitoring adalah berbentuk *service*, yaitu suatu proses yang terus-menerus berjalan pada interval waktu tertentu. Proses-proses yang terjadi pada suatu sistem monitoring dimulai dari pengumpulan data seperti data dari *network traffic*, *hardware information*, dan lain-lain yang kemudian data tersebut dianalisis pada proses analisis data dan pada akhirnya data tersebut akan ditampilkan.

Pada beberapa aplikasi sistem monitoring, akses benar-benar dibatasi dari *local host terminal* saja. Pertanyaannya apakah bisa dilakukan monitoring dari jarak jauh, dimana semua data yang dikumpulkan dari terminal komputer yang berada di lokasi berbeda dengan instrumennya misalnya dengan menggunakan jaringan LAN (*Local Area Network*) atau bahkan internet. Untuk menjalankan sistem monitoring yang seperti ini sangat memungkinkan sekali dapat dilakukan dengan menggunakan *interface* program yang dapat menjembatani pengguna melalui *web browser* pada *remote terminal*. *Interface* program ini disebut CGI (*Common Gateway Interface*) yang biasanya tersedia pada linux [1].

1.2 Jaringan Komputer

Menurut Ardiansyah (2010), jaringan komputer (*Computer Network*) yang disebut secara singkat dengan jaringan adalah kumpulan komputer dan alat alat yang lain yang saling dihubungkan bersama menggunakan media komunikasi tertentu. Informasi yang melintas sepanjang media komunikasi, memungkinkan pengguna jaringan untuk saling bertukar data atau menggunakan perangkat lunak maupun perangkat keras secara berbagi. Masing - masing komputer yang atau alat - alat lain yang di hubungkan pada jaringan disebut *node*. Jaringan dapat terdiri dari puluhan, ratusan atau bahkan ribuan *node*.

1.3 *Network Monitoring System* (NMS)

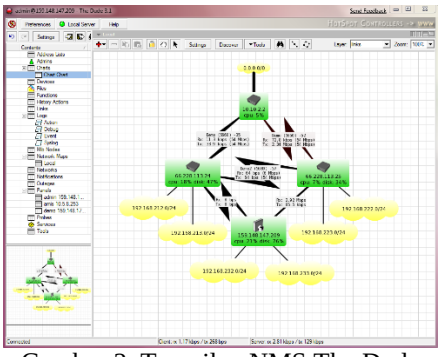
NMS merupakan *tools* untuk melakukan monitoring/pengawasan pada elemen-elemen dalam jaringan komputer. Fungsi dari NMS adalah melakukan pemantauan terhadap kualitas SLA (*Service Level Agreement*) dari *bandwidth* yang digunakan (Fachruddin, 2009). Hasil dari pantauan tersebut biasanya dijadikan bahan dalam pengambilan keputusan oleh pihak manajemen, disisi lain digunakan oleh Administrator jaringan (*technical person*) untuk menganalisa apakah terdapat kejanggalan dalam operasional jaringan.

1.3.1 Aplikasi NMS

Terdapat beberapa aplikasi *Network Management System*(NMS) yang digunakan dalam pengerjaan penelitian ini adalah sebagai berikut :

a. The Dude

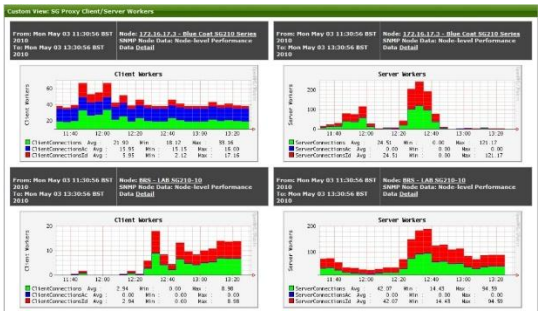
The Dude Network Monitor merupakan aplikasi MikroTik yang dapat secara signifikan meningkatkan jangkauan Administrator dalam mengelola lingkungan jaringan. The Dude akan otomatis memindai semua perangkat di dalam subnet tertentu, menggambar dan membuat tata letak peta jaringan, memberikan layanan memonitor perangkat-perangkat yang terhubung dalam jaringan dan memberikan peringatan jika suatu layanan jaringan mengalami masalah. Bentuk monitoring NMS The Dude dapat dilihat pada gambar 2.



Gambar 2. Tampilan NMS The Dude

b. OpenNMS

OpenNMS ialah sebuah alat manajemen jaringan berbasis *open source* skala enterprise. OpenNMS membantu seorang administrator jaringan untuk memantau layanan yang terganggu pada peralatan atau perangkat di sisi *remote* (jauh) dan mengumpulkan formasi dari simpul perangkat *remote* tersebut dengan menggunakan SNMP. OpenNMS menyediakan. OpenNMS ditulis dalam bahasa JAVA namun, konfigurasi data tersedia secara langsung lewat *data extensible markup language* (XML). Satu hal yang cukup penting ialah OpenNMS mampu bekerja pada sistem bertingkat (hierarkikal) dan mampu untuk memonitoring beberapa *service*/layanan seperti : ICMP, SNMP, FTP, HTTP, SMTP, DNS, Router TCP, Sybase (TCP), MySQL, Postgres, Oracle (TCP), DHCP, MS Exchange, IMAP, POP3 dan lain-lain. OpenNMS juga menawarkan jasa, dukungan, serta pengembangan dan pelatihan untuk membuat solusi custom yang dibuat untuk menyesuaikan proses bisnis [3].



Gambar 3. Tampilan NMS OpenNMS

Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)
Edisi. 1 Volume. 1 Februari 2016 ISSN : 2089-9033

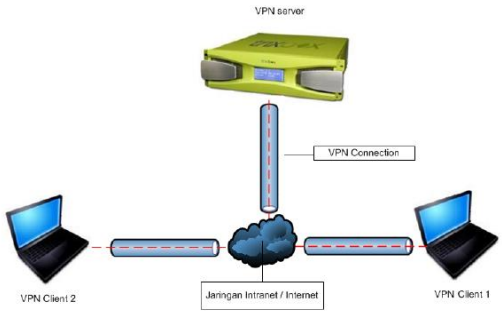
1.4 *Remote Access*

Remote access merupakan mekanisme yang biasa digunakan dalam pengendalian suatu perangkat jaringan, dimana Administrator dapat dengan mudah mengontrol dan mengawasi komputer *server*, *backup data*, atau aktifitas lainnya yang berhubungan dengan *server* [4].

Pada saat ini banyak media yang dapat digunakan untuk dapat melakukan *remote access* ini seperti komputer/laptop, *smartphone* dan perangkat terkomputerisasi lainnya. Untuk dapat melakukan *remote access* ini diharuskan agar perangkat yang akan terlibat *remote* harus terkoneksi dengan internet atau intranet.

1.5 *Virtual Private Network* (VPN)

Menurut Andri (2010), VPN (*Virtual Private Network*) adalah teknik pengamanan jaringan dengan cara membuat suatu *tunnel* misalkan pada jaringan publik atau *internet* sehingga jaringan bersifat *private* dan aman. VPN dikatakan bersifat *private* karena ketika akan terbentuknya sebuah koneksi VPN dibutuhkannya *authentikasi* untuk memastikan bahwa kedua ujung dalam koneksi adalah *user* yang sesuai yang diberi kewenangan untuk mengakses suatu *server*. Setelah terbentuknya suatu koneksi VPN maka data akan di *enkripsi* untuk menjaga kerahasiaan paket yang dikirimkan. Penggunaan *authentikasi* dan *enkripsi* yang dipadukan dalam VPN membuat suatu sistem keamanan yang lebih baik seperti pada Gambar 4.

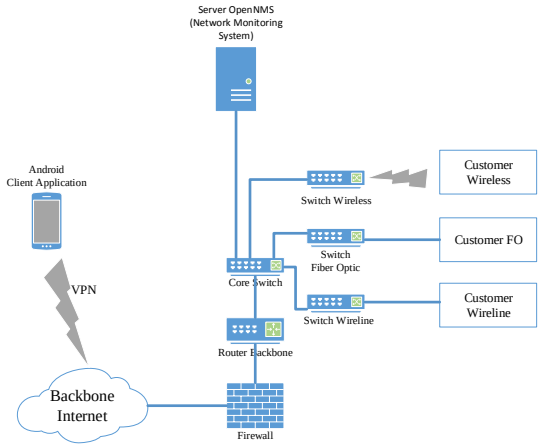


Gambar 4. Pelewatan Data dengan VPN

2. ISI PENELITIAN

2.1 Deskripsi Sistem

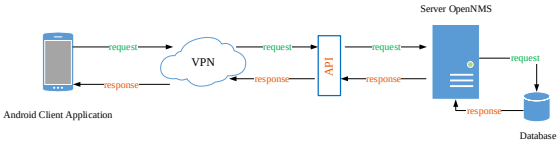
Sistem yang akan dibangun menggunakan konsep *client server* dalam arsitekturnya, dimana arsitektur pada sistem ini terdiri dari *Android Client Application* sebagai *client* dan *Server* OpenNMS sebagai *server*. *Android Client Application* mengakses perangkat jaringan melalui jaringan VPN.



Gambar 5. Arsitektur Sistem Keseluruhan

Server OpenNMS merupakan sistem yang menganalisa jaringan mulai dari *core router* sampai jaringan yang terhubung ke *Customer*. OpenNMS akan me*monitoring network element* seperti *switch*, *router* dan perangkat jaringan lainnya. OpenNMS akan mengenali, mengisolasi, mengoreksi dan mencatat kesalahan yang terjadi di dalam jaringan. Ini bertujuan untuk memprediksi *error* agar jaringan dapat selalu tersedia. *Server* OpenNMS juga sebagai penyedia data dari hasil analisis jaringan. Data yang disediakan oleh API OpenNMS yaitu data alarm, data ketersediaan jaringan(*availability*), data *event*, data *outages*, data *nodes* yang terhubung ke jaringan. Data tersebut akan diakses oleh *Android Client Application* berupa data JSON.

Android Client Application merupakan aplikasi yang digunakan oleh Administrator untuk mengetahui informasi mengetahui kondisi jaringan yang di*monitoring* oleh *Server* OpenNMS. Aplikasi akan melakukan *request* data ke *server* OpenNMS kemudian data tersebut diolah pada aplikasi. Informasi jaringan yang didapatkan adalah performasi jaringan, daftar masalah pada jaringan berupa data alarm, daftar *history* terakhir yang terjadi pada jaringan yaitu berupa data *event*, daftar *node* yang terhubung ke jaringan dan data *outages*. Aplikasi akan melakukan *request* data alarm setiap satu menit dan apabila terdapat data alarm maka aplikasi akan menampilkan notifikasi sehingga seorang Administrator dapat mengetahui bahwa terjadi masalah pada jaringan yang dikelolanya.

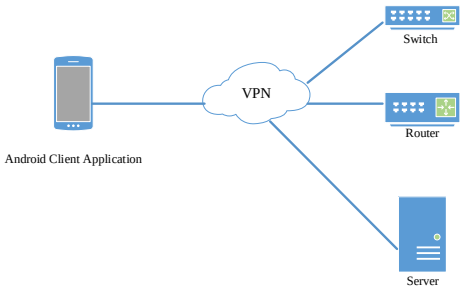


Gambar 6. Arsitektur Sistem Monitoring Dengan Android

Berikut adalah deskripsi dari Gambar 6 Arsitektur sistem monitoring dengan android :

1. *Android Client Application* terhubung ke VPN kemudian melakukan *request* data ke *server* OpenNMS melalui API.
2. *Server* menerima *request* data dari *Android Client Application* dan menentukan jenis *request* yang diminta.
3. Setelah *server* menerima data yang diminta data tersebut akan dikembalikan dalam bentuk JSON untuk diproses pada *Android Client Application.*

Terdapat fitur berupa *remote* yang dapat digunakan untuk melakukan akses secara jarak jauh dan konfigurasi langsung ke perangkat jaringan atau *server*. Konsep untuk melakukan *remote* menggunakan aplikasi pihak ketiga yaitu ConnectBot yang dapat diakses melalui aplikasi yang dibangun. Agar dapat melakukan *remote* ke semua perangkat jaringan yang berada di perusahaan baik itu IP Lokal maupun IP Publik maka perangkat android harus terhubung dengan jaringan VPN milik perusahaan seperti yang diilustrasikan pada gambar 7 berikut.



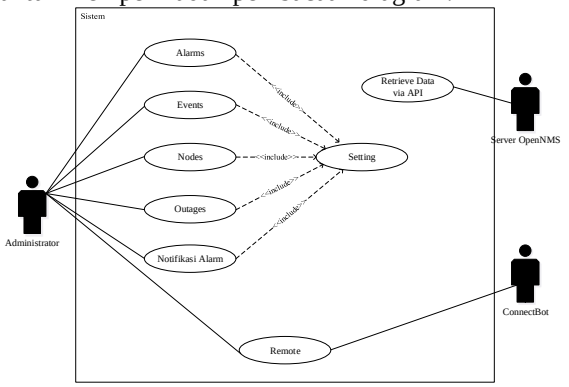
Gambar 7. Arsitektur *Remote* Dengan Android

2.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional *menggunakan use case diagram* untuk mengambarkan fungsionalitas yang ada pada perangakat lunak yang akan dibangun.

1. *Use Case Diagram*

Use case diagram digunakan untuk menjelaskan apa yang akan dilakukan sistem serta aktor-aktor yang akan berhubungan dengan proses-proses yang ada pada sistem. Garis keterhubungan yang dimiliki *use case Retrieve data via API* tidak digambarkan untuk mempermudah pembacaan diagram.



Gambar 8. *Use Case Diagram*

2.3 Instalasi *Server* OpenNMS

Server OpenNMS merupakan sebuah perangkat keras yang berupa komputer yang dijadikan *server*. Diperlukan proses instalasi dan konfigurasi pada *server* agar sistem dapat berjalan sebagaimana mestinya. Dalam proses ini akan dilakukan dalam beberapa tahap yaitu :

1. Pada proses pertama yaitu diawali dengan instalasi sistem operasi pada komputer *server* sebagai monitoring jaringan. Sistem yang digunakan adalah Linux Ubuntu 12.04.
2. Selanjutnya instalasi perangkat lunak pendukung untuk menjalankan OpenNMS yaitu Java Development Kit dan PostgreSQL.
3. Instalasi *snmp* dan *snmpd* pada Linux.
4. Kemudian instalasi dan konfigurasi OpenNMS.

2.4 Implementasi Perangkat Keras Pembangun Sistem

Perangkat keras yang diimplentasikan pada pembangunan aplikasi sistem monitoring jaringan berbasis android merupakan kebutuhan perangkat keras yang ditentukan pada tahap analisis.

1. Perangkat Keras Pembangun Aplikasi

Berikut adalah perangkat keras yang digunakan dalam pembangunan aplikasi sistem monitoring jaringan berbasis android ini.

Tabel 1. Perangkat Keras Pembangun Aplikasi

No	Perangkat Keras	Spesifikasi
1	Prosesor	Intel Core i3 2.4Ghz
2	RAM	6 GB
3	VGA	AMD Radeon 1 GB
4	Hardisk	500 GB
5	Layar	14 inch
6	*Input Device*	Mouse dan Keyboard

2. Perangkat Keras Pembangun *Server* OpenNMS

Berikut merupakan spesifikasi perangkat keras yang digunakan pada *server* OpenNMS.

Tabel 2. Perangkat Keras *Server* OpenNMS

No	Perangkat Keras	Spesifikasi
1	Prosesor	Intel Xeon E5-2667 3.3Ghz
2	RAM	4 GB
3	VGA	VGA compatible controller
4	Hardisk	SCSI 20 GB

3. Perangkat Keras *Smartphone*

Berikut merupakan spesifikasi perangkat keras *smartphone* yang digunakan yaitu berupa *smartphone* android yang berperan sebagai *client* untuk mengakses sistem.

Tabel 3. Perangkat Keras *Smartphone*

No	Perangkat Keras	Spesifikasi
1	Prosesor	Quad Core 1.2Ghz
2	Memori RAM	1 GB
3	Layar	540x960 piksel 4.5 inch
4	Memori Internal	4 GB
5	*Input Device*	Touch Screen

2.5 Implementasi Perangkat Lunak Pembangun Sistem

Perangkat lunak yang diimplentasikan pada pembangunan aplikasi sistem monitoring jaringan berbasis android merupakan kebutuhan perangkat lunak yang ditentukan pada tahap analisis.

Berikut adalah perangkat lunak yang digunakan dalam pembangunan aplikasi sistem monitoring jaringan berbasis android.

1. Sistem Operasi Windows 7 64-bit
2. Android Studio versi 1.4
3. Android SDK versi 22
4. Java Development Kit (JDK) versi 8

Sedangkan perangkat lunak yang digunakan pada *server* OpenNMS adalah sebagai berikut:

1. Sistem Operasi Linux Ubuntu 12.04
2. OpenNMS versi 16.0.3
3. PostgreSQL versi 9.1
4. Java Development Kit (JDK) 8

2.6 Pengujian Sistem

Pengujian merupakan tahapan yang dilakukan untuk menguji serta mengetahui apakah parameter-parameter konfigurasi sistem dan merepresentasikan kajian pokok dari spesifikasi, perancangan, dan pengkodean yang di implementasikan berjalan dengan baik dan sesuai dengan tujuan sebelumnya.

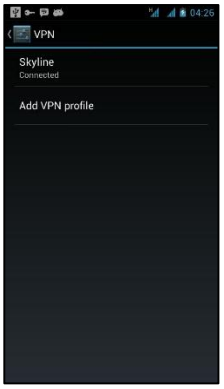
2.6.1 Hasil Pengujian

2.6.1.1 Menguji Fungsionalitas Setting Perangkat Lunak

Terdapat dua mekanisme pengujian yaitu pengujian dengan data yang sesuai dan pengujian dengan data salah. Berikut pengujian fungsionalitas *setting* dengan memasukan data yang sesuai.

1. Pengujian Fungsionalitas *Setting* dengan Data Benar

Pada pengujian ini terdapat dua komponen yang diuji yaitu *VPN Connection* dan *Server Settings*. Hasil pengujian *VPN Connection* dengan memasukan data yang benar ditandai dengan status "*Connected*" bisa terlihat pada Gambar 9 berikut.



Gambar 9 Tampilan *VPN Connection* (Data Benar)

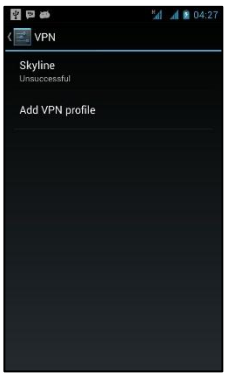
Pengujian komponen selanjutnya yaitu *Server Settings*. Hasil pengujian *Server Settings* dengan memasukan data yang benar bisa terlihat pada Gambar 10 berikut.



Gambar 10 Tampilan *Server Settings* (Data Benar)

2. Pengujian Fungsionalitas *Setting* dengan Data Salah

Pada pengujian ini terdapat dua komponen yang diuji yaitu *VPN Connection* dan *Server Settings*. Hasil pengujian *VPN Connection* dengan memasukan data yang salah ditandai dengan status "*Unsuccessful*" bisa terlihat pada Gambar 11 berikut.



Gambar 11 Tampilan *VPN Connection* (Data Salah)

Pengujian komponen selanjutnya yaitu *Server Settings*. Hasil pengujian *Server Settings* dengan memasukan data yang salah ditandai dengan

Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)
Edisi. 1 Volume. 1 Februari 2016 ISSN : 2089-9033

munculnya ikon peringatan seperti yang terlihat pada Gambar 12 berikut.



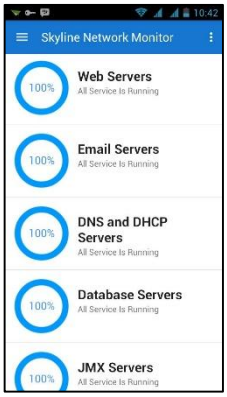
Gambar 12 Tampilan *Server Settings* (Data Salah)

2.6.1.2 Menguji Fungsionalitas Perangkat Lunak untuk Monitoring

Berikut pengujian dari setiap fungsionalitas yang diuji.

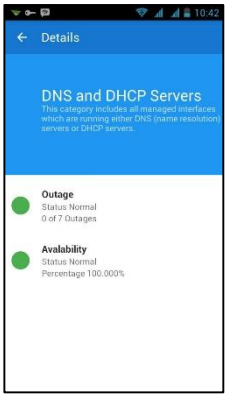
1. Menguji fungsionalitas menu *dashboard.*

Pada aplikasi sistem monitoring jaringan berbasis android yang dibangun ini terdapat menu *dashboard.* Berikut tampilan ketika Administrator mengakses menu tersebut.



Gambar 13 Tampilan *Dashboard*

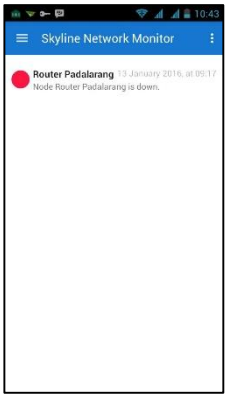
Pada Gambar 13 tersebut menampilkan statistik jaringan yang dimonitoring secara keseluruhan. Terdapat juga informasi secara detail mengenai data tersebut. Berikut tampilan informasi secara detail.



Gambar 14 Data *DNS and DHCP Servers* Secara Detail

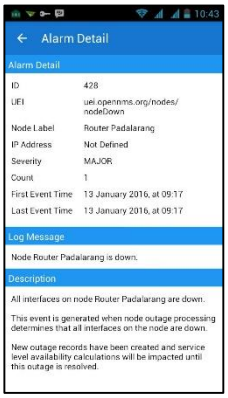
2. Menguji fungsionalitas akses data *alarms.*

Pada aplikasi sistem monitoring jaringan berbasis android yang dibangun ini terdapat menu *alarms.* Berikut tampilan ketika Administrator mengakses menu *alarms* tersebut.



Gambar 15 Daftar Data *Alarms*

Pada Gambar 15 tersebut menampilkan data alarm berupa informasi *node* yang bermasalah, waktu *node* tersebut bermasalah dan informasi singkat mengenai masalah yang terjadi. Terdapat juga informasi secara detail mengenai data alarm tersebut. Tampilan alarm secara detail dapat dilihat pada Gambar 16 berikut.



Gambar 16 Data Alarm Secara Detail

Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)
Edisi. 1 Volume. 1 Februari 2016 ISSN : 2089-9033

3. Menguji fungsionalitas akses data *Nodes.*

Pada aplikasi sistem monitoring jaringan berbasis android yang dibangun ini terdapat menu *nodes.* Berikut tampilan ketika Administrator mengakses menu *nodes* tersebut.



Gambar 17 Daftar *Node*

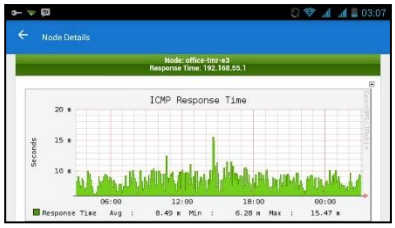
Pada Gambar 17 tersebut menampilkan data node berupa nama *node* atau *IP address node* yang dimonitoring. Terdapat juga informasi secara detail mengenai data *node* tersebut. Berikut tampilan *node* secara detail serta performansi *node*.



Gambar 18 Detail *Node*



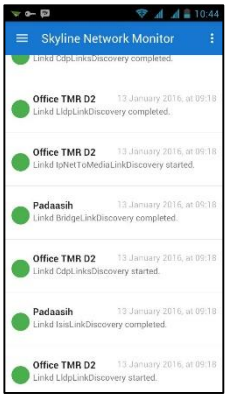
Gambar 19 *Physical Interface Node* Office-tmr-e3



Gambar 20 Grafik *Node Response Time*

4. Menguji fungsionalitas akses data *Events.*

Pada aplikasi sistem monitoring jaringan berbasis android yang dibangun ini terdapat menu *Event.* Berikut tampilan ketika Administrator mengakses menu *Events* tersebut.



Gambar 21 Daftar Data *Events*

Pada Gambar 21 tersebut menampilkan data *events* berupa informasi yang terjadi pada sistem, waktu *event* tersebut dan informasi singkat mengenai masalah yang terjadi. Terdapat juga informasi secara detail mengenai data *event* tersebut. Berikut tampilan *events* secara detail.



Gambar 22 Data *Event* Secara Detail

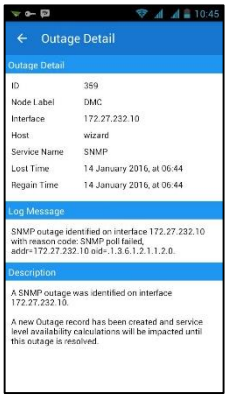
5. Menguji fungsionalitas akses data *Outages.*

Pada aplikasi sistem monitoring jaringan berbasis android yang dibangun ini terdapat menu *Outages.* Berikut tampilan ketika Administrator mengakses menu *Outages* tersebut.



Gambar 23 Daftar Data *Outages*

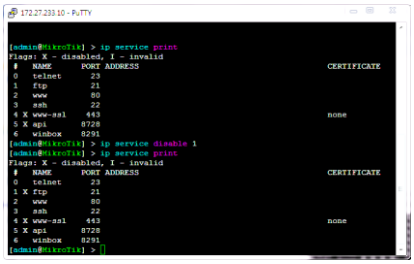
Pada Gambar 23 tersebut menampilkan data *outages* berupa nama *node*, *interface* dan *service*. Terdapat juga informasi secara detail mengenai data *outage* tersebut. Berikut tampilan *outage* secara detail.



Gambar 24 Data *Outage* Secara Detail

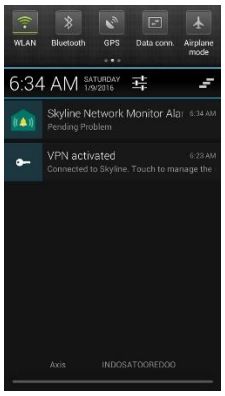
2.6.2.3 Menguji Fungsionalitas Notifikasi Alarm

Pada tahap ini akan dilakukan simulasi gangguan pada jaringan dengan mematikan *service* FTP pada salah satu *router* yang dimonitoring. Untuk melakukan simulasi tersebut digunakan aplikasi putty pada komputer untuk mengakses *router*. Setelah berhasil masuk ke *router* ketikan perintah "*ip service disable 1*" untuk mematikan *service* FTP. Berikut gambaran simulasi mematikan *service* FTP menggunakan aplikasi putty.



Gambar 25 Mematikan *Service* FTP

Setelah berhasil melakukan perubahan konfigurasi pada *router* maka *router* yang di*monitoring* memiliki *service* FTP dengan parameter *disable* yang dianggap sebagai gangguan pada jaringan. Maka Aplikasi sistem monitoring jaringan berbasis android akan memunculkan notifikasi sebagai pemberitahuan kepada Administrator bahwa sedang terjadi gangguan. Berikut tampilan notifikasi pada *smartphone* android.

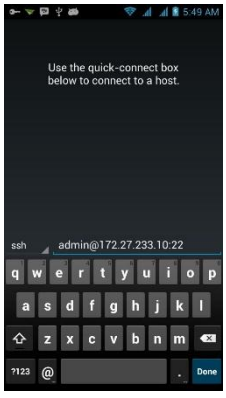


Gambar 26 Notifikasi Alarm Gangguan Jaringan

2.6.2.4 Menguji Remote Access

Pada poin ini akan dilakukan pengujian *remote access* dari perangkat lunak yang dibangun ke perangkat jaringan yang akan dikonfigurasi sebagai penanganan dalam menanggulangi masalah yang terjadi pada jaringan.

Sebelumnya telah dilakukan simulasi gangguan dengan mematikan *service* FTP pada salah satu *router*. Pada tahap ini akan dilakukan konfigurasi pada *router* tersebut dengan menghidupkan kembali *service* yang bermasalah. Untuk melakukan *remote* diperlukan ssh *login* ke *router* dengan menginputkan *username*, IP *address*, *port* dan *password*. Berikut gambaran *login* ke *router*.



Gambar 27 *Remote Login* Ke *Router*

Setelah proses autentikasi berhasil, ketikan perintah "*ip service print*" maka akan terlihat *service* FTP dengan parameter *disable* (tanda "X"). Untuk menghidupkan *service* tersebut ketikan perintah "*ip service enable 1*". Kemudian periksa kembali bahwa *service* tersebut sudah dalam parameter *enable* dengan perintah "*ip service print*". Berikut gambaran

konfigurasi mengatur *service* ftp pada *router* yang diakses secara *remote* melalui aplikasi pada *smartphone* android.



Gambar 28 Konfigurasi *Enable Service* FTP

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil pengujian dan aplikasi sistem monitoring jaringan berbasis android maka diperoleh kesimpulan sebagai berikut:

1. Aplikasi sistem monitoring jaringan berbasis android sudah membantu Administrator dalam memonitoring jaringan yang dikelolanya.
2. Administrator mendapatkan pemberitahuan (notifikasi) ketika mengalami gangguan melalui aplikasi yang dibangun.
3. Aplikasi sistem monitoring jaringan berbasis android memberikan informasi performa *server* yang dimonitoring.
4. Administrator dapat melakukan *remote* ke *server* atau perangkat jaringan melalui aplikasi yang dibangun.

3.2 Saran

Aplikasi sistem monitoring jaringan berbasis android yang dibangun masih memiliki beberapa kekurangan. Oleh karena itu ada beberapa saran yang dapat digunkan sebagai panduan pengembangan aplikasi ke arah yang lebih baik lagi. Adapun saran-saran yang terhadap pengembangan aplikasi ini adalah sebagai berikut:

1. Menggunakan *Google Cloud Messaging* sebagai *push notification* untuk notifikasi alarm.
2. *Remote* pada aplikasi ini menggunakan aplikasi pihak ketiga. Akan lebih baik lagi jika membangun modul sendiri untuk *remote* ke *server* agar tidak perlu menginstal aplikasi pihak ketiga.
3. Tampilan *dashboard* hanya menampilan statistik sistem secara keseluruhan. Akan lebih lagi jika pada *dashboard* terdapat grafik *realtime* dari setiap *node* yang monitor.

DAFTAR PUSTAKA

[1] Ohara, Gheyb Jhuana, Aplikasi Sistem Monitoring Berbasis Web Untuk Open Cluster, Bandung : 2005.
[2] Kristanto, Andri, Jaringan Komputer, Graha Ilmu, 2006.
[3] Wikipedia.org, “Wikipedia.org” 2015. [Online]. Available: https://en.wikipedia.org/wiki/OpenNMS. [Accessed 5 April 2015].
[4] I Made Yonatan Wijaya Ndon Eko Satriyo Utomo, "IMPLEMENTASI VISUAL BASIC 6 UNTUK PEMBUATAN APLIKASI REMOTE ACCESS BERBASIS TCP/IP," p. 1, 2010.