

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/279494115>

Perancangan Simple Network Management Protocol (SNMP) Agent Simulator Berbasis Open Source Pada Jaringan...

Article · October 2012

CITATIONS

0

READS

321

3 authors, including:



Muhamad Komarudin

Lampung University

11 PUBLICATIONS 15 CITATIONS

[SEE PROFILE](#)



Mona Batubara

Lampung University

3 PUBLICATIONS 1 CITATION

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Intensity average value of image segmentation for infrared image of environmental condition [View project](#)



Precision agriculture [View project](#)

Perancangan *Simple Network Management Protocol (SNMP) Agent Simulator* Berbasis *Open Source* Pada Jaringan *World Wide Interoperability For Microwave Access (WiMAX)*

M. Komarudin¹, Mona A M Batubara¹, Johan²

¹Staf Pengajar Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung
Jl.Prof.Dr. Soemantri Brojonegoro No.1 Bandar Lampung

²Astra Graphia Information Technology, Jakarta

Abstrak—WiMAX merupakan salah satu teknologi dalam perangkat jaringan yang memerlukan Network Management System (NMS) yang dapat memantau, memelihara dan mengelolanya. Dalam proses pengelolaan jaringan diperlukan protokol proses pertukaran informasi yang dapat diimplementasikan menggunakan Simple Network Management Protocol (SNMP). Penelitian dilakukan dengan merancang dan mengimplementasikan suatu SNMP agent simulator yang diterapkan sebagai Base Station (BS) atau Subscriber Station (SS) dengan menggunakan Personal Computer (PC) yang telah diinstalasi dengan suatu sistem operasi. Pertukaran informasi dalam NMS pada jaringan WiMAX menggunakan Management Information Base (MIB) standar IEEE 802.16f. Hasil yang diperoleh dari penelitian ini merupakan suatu SNMP agent simulator yang kemudian diuji dengan aplikasi OpenNMS sebagai managing system menggunakan data simulasi yang tersimpan dalam database yang dapat diatur melalui website

Kata kunci: WiMAX, MIB, Network Management System (NMS), SNMP, OpenNMS

Abstract—*Network Management System (NMS) is one of network technologies which is required by Interoperability for Microwave Access (WiMAX) for monitoring, maintaining and managing. In managing the network, information transfer process that is occurred can be implemented by Simple Network Management Protocol (SNMP). This research involves designing and implementing an SNMP agent simulator on Personal Computer (PC) that is implemented as Base Station (BS) or Subscriber Station (SS). Information exchange of NMS in WiMAX network utilizes Management Information Base (MIB) IEEE 802.16f standard*

The resulting SNMP agent simulator is tested on the OpenNMS application using a database that can be modified from a website.

Keywords: WiMAX, MIB, Network Management System (NMS), SNMP, OpenNMS

A. Pendahuluan

World Wide Interoperability for Microwave Access (WiMAX) merupakan jaringan komunikasi nirkabel yang distandarisasikan berdasarkan **IEEE 802.16**. Sebagaimana jaringan komunikasi pada umumnya, jaringan WiMAX yang telah dibangun memerlukan sistem pengelolaan agar tetap bekerja sebagaimana mestinya. Agar memudahkan pengguna atau *administrator* dalam mengatur dan memonitor jaringan WiMAX dan perangkat-perangkatnya perlu dikembangkan sebuah aplikasi *Network Management System (NMS)*. Untuk memperbolehkan ketersambungan *management* melalui *platform* jaringan yang berbeda-beda, standar *network management* dibutuhkan sehingga *vendor* dapat mengimplementasikan dan menerapkannya. *Simple Network Management Protocol (SNMP)* merupakan sekumpulan standar untuk *network management*, termasuk sebuah protokol, sebuah spesifikasi struktur *database*, sekumpulan objek data. Cakupan utama dalam penelitian ini adalah bagaimana merancang suatu *SNMP agent simulator* yang berfungsi seolah-olah sebagai *Base Station (BS)* dan *Subscriber Station (SS)* dalam NMS pada jaringan WiMAX. Pengembangan NMS di Indonesia

Naskah ini diterima pada tanggal 28 Juni 2008, direvisi pada tanggal 20 Juli 2008 dan disetujui untuk diterbitkan pada tanggal 1 Agustus 2008

dilakukan dengan melibatkan pemerintah, swasta dan akademisi dimana Unila bersama-sama ITB berperan aktif didalam pengembangannya [1].

B. Teori Dasar

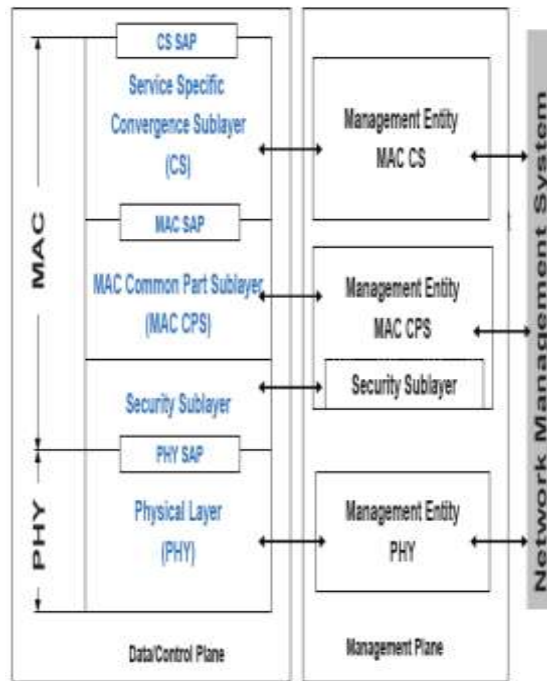
WiMAX adalah istilah yang digunakan untuk menggambarkan standar dan implementasi yang mampu beroperasi berdasarkan jaringan nirkabel IEEE 802.16 [2]. WiMAX merupakan standar *Broadband Wireless Access* (BWA) dengan kemampuan untuk menyalurkan data kecepatan tinggi (layaknya teknologi xDSL pada jaringan wireline). Banyak kemampuan lebih yang ditawarkan oleh teknologi WiMAX dibandingkan teknologi sebelumnya seperti kemampuan yang diterapkan dalam kondisi NLOS, aplikasinya baik untuk *fixed*, *nomadic*, portabel maupun *mobile*.

Lapisan protocol standar IEEE 802.16 ditunjukkan pada Gambar 1. Dengan model *OSI Layer*, *Media Access Control* (MAC) pada WiMAX menggunakan metode akses yang berbasis algoritma penjadualan (*scheduling algorithm*) sehingga bila setelah sebuah terminal mendapat garansi untuk memperoleh sejumlah sumber daya (seperti *timeslot*), maka jaringan nirkabel akan terus memberikan sumber daya ini selama terminal membutuhkannya. MAC WiMAX terdiri atas tiga *sublayer*. *Service-Specific Convergence Sublayer* (CS) menyediakan semua transformasi dan *mapping* dari data eksternal yang diterima melalui *CS service access point* (SAP), kepada MAC SDUs diterima oleh MAC *Common Part Sublayer* (CPS) melalui MAC SAP. MAC CPS menyediakan inti fungsi MAC dari sistem akses, alokasi *bandwidth*, membuat koneksi, dan memelihara koneksi. MAC juga terdiri atas *sublayer* keamanan yang menyediakan autentifikasi, pertukaran kunci dengan aman, dan enkripsi. Data PHY *control*, dan statistik di transfer di

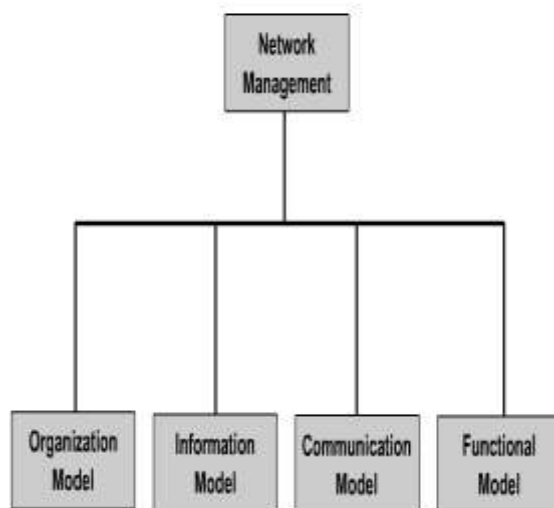
antara MAC CPS dan PHY, melalui PHY SAP.

Network Management System (NMS)

International Standard Organization (ISO) telah membuat suatu komite untuk menghasilkan suatu model untuk *network management* dibawah *OSI group* yang diperlihatkan pada Gambar 2 di bawah ini. *Organization model* menggambarkan komponen dari *network management* seperti suatu *manager*, *agent*, dan seterusnya serta hubungannya. Pengaturan komponen-komponen tersebut untuk membedakan tipe arsitektur suatu *network management*. *Information model* berfokus pada struktur dan *storage* dari *network management information*. Informasi ini disimpan dalam sebuah *database*, yang disebut *Management Information Base* (MIB). ISO mendefinisikan *Structure of Management Information* (SMI) untuk mendefinisikan *syntax* dan semantik dari *management information* yang tersimpan didalam MIB. *Communication model* berhubungan dengan bagaimana *management data* dikomunikasikan antara proses *agent* dan *manager*. Model ini berfokus pada *transport protocol*, *application protocol*, dan *command* serta respon antara *peer*. *Functional model* mengalami aplikasi *network management* yang berada pada *network management station*. Model *OSI network management* mengategorikan menjadi lima wilayah fungsi, yang sering disebut juga sebagai model FCAPS (*Fault, Configuration, Accounting, Performance, Security*).



Gambar 1. Layer PHY dan MAC pada Standar IEEE 802.16 [2]



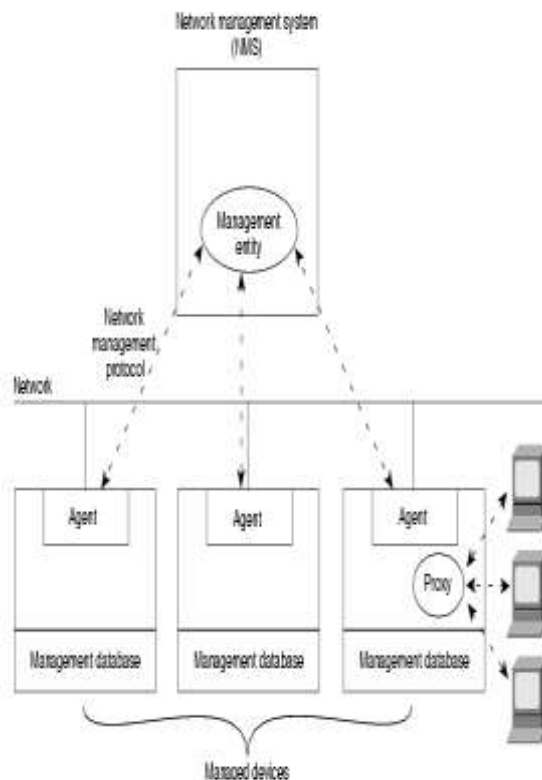
Gambar 2. Struktur Model *Network Management* pada standar OSI

Arsitektur *network management*

Sebagian besar arsitektur *network management* menggunakan struktur dasar yang sama dan sekumpulan hubungan. *Managed device* seperti sistem komputer dan *device* jaringan yang lain, menjalankan perangkat lunak yang mengizinkan mereka

mengirimkan tanda peringatan ketika mengenali masalah (contohnya ketika satu atau lebih *user* yang ditentukan melebihi ambang batas ketentuan). Ketika menerima tanda peringatan ini, *management entity* diprogram untuk bereaksi dengan mengeksekusi satu, beberapa, atau suatu kelompok aksi, termasuk pemberitahuan operator, mencatat *event*, *system shutdown*, dan usaha otomatis pada *system repair*.

Management entity juga dapat melakukan *poll* terhadap *end station* untuk mengecek nilai variabel yang pasti. *Polling* dapat otomatis atau diinisiasikan oleh *user*, tetapi *agent* pada *managed device* merespon semua *poll*. *Agent* adalah modul perangkat lunak yang pertama meng-*compile* informasi tentang *managed device* didalamnya, lalu menyimpan informasi ini didalam sebuah *management database*, dan terakhir menyediakannya (secara proaktif atau reaksi) ke *management entity* didalam *network management system* (NMS) melalui suatu protokol *network management*. Protokol *network management* yang umum adalah *Simple Network Management Protocol* (SNMP) dan *Common Management Information Protocol* (CMIP). *Management proxy* adalah *entity* yang menyediakan *management information* atas nama *entity* yang lain. Gambar 3 menggambarkan suatu tipe arsitektur *network management*. Komponen utama dalam suatu *network management* terdiri dari *device* yang harus di-*manage*.



Gambar 3. Arsitektur *network management*

Simple Network Management Protocol (SNMP)

SNMP merujuk kepada sekumpulan standar untuk *network management*, termasuk sebuah protokol, sebuah spesifikasi struktur *database*, sekumpulan objek data. SNMP diadopsi sebagai standar untuk *internet TCP/IP* pada 1989 dan menjadi sangat populer. Kemudian dilakukan sebuah *upgrade*, dikenal sebagai SNMP versi 2c (SNMPv2c) yang diadopsi pada tahun 1993. SNMPv2c menyediakan dukungan untuk strategi *network management* tersentral dan terdistribusi, dan termasuk peningkatan dalam *structure of management information* (SMI), operasi protokol, arsitektur *management*, dan *security*. Ini didesain untuk diterapkan pada jaringan berdasarkan OSI atau jaringan berdasarkan TCP/IP. Kemudian di-release SNMPv3 untuk menyelesaikan kelemahan dari *security* pada SNMPv1 dan SNMPv2c, yaitu dengan menyediakan keamanan akses ke MIB dengan

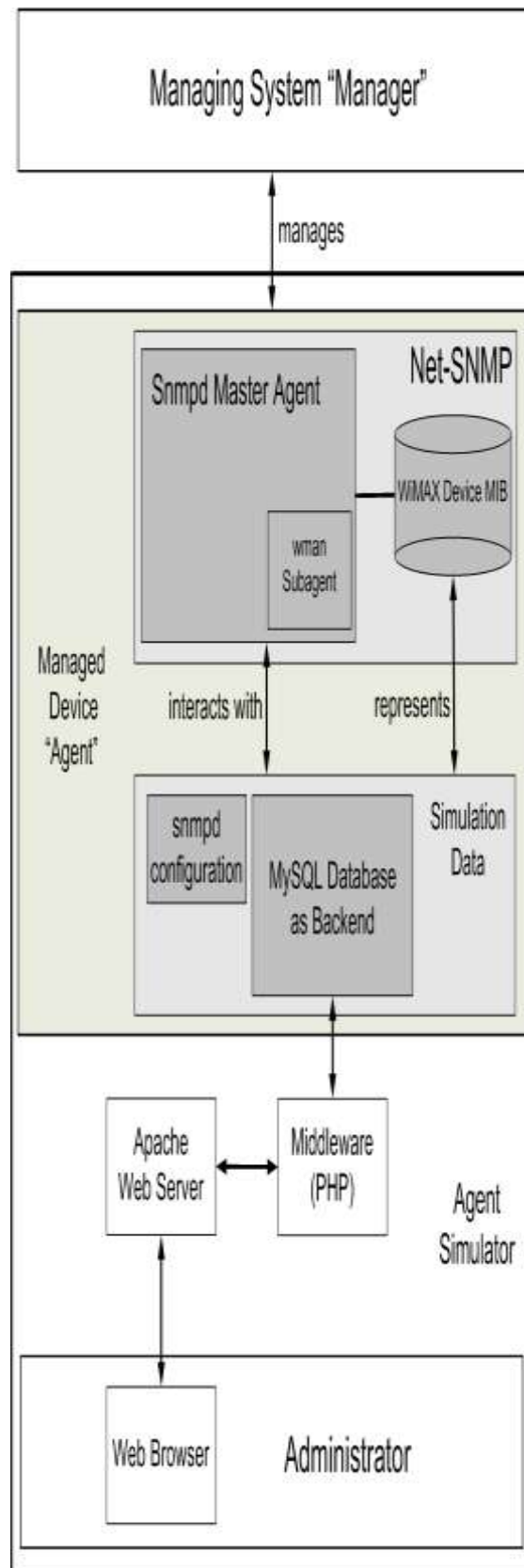
melakukan autentikasi dan mengenkripsi paket melalui jaringan.

SNMP yang berupa protokol *application layer* didesain untuk memfasilitasi pertukaran *management information* antara *device* jaringan. Dengan SNMP, kita dapat mengakses data *management information*, seperti paket per detik dikirim pada suatu *interface* atau sejumlah koneksi *open TCP*. *Network administrator* dapat lebih mudah mengatur *performance* dari *network management* untuk mencari dan menyelesaikan masalah jaringan. *Organizational model* dari *network management* berdasarkan SNMP terdapat empat elemen: *Management station*, *Management agent*, *Management information base*, Protokol *network management*.

SNMP menggunakan *User Datagram Protocol* (UDP) dan berkomunikasi melalui port 161 dan 162 berdasarkan pada suatu pertukaran pesan. Dan terdapat tiga tipe pesan secara umum:

- **Get:** Memungkinkan *management station* mendapatkan nilai objek MIB dari *agent*.
- **Set:** Memungkinkan *management station* mengeset nilai objek MIB pada *agent*.
- **Trap:** Memungkinkan *agent* untuk memberitahukan *management station* tentang *event* yang signifikan.

Model ini disebut sebagai sebuah model *two-tier* dengan mengasumsikan bahwa semua elemen jaringan bersifat *manageable* oleh SNMP. Model *two-tier* tidak selalu terjadi, sebab beberapa *device* memiliki suatu *management interface* yang bersifat *proprietary*. Dalam hal ini, model *three-tiered* dibutuhkan.



Gambar 4. Rancangan SNMP *agent simulator*

C. Metode Penelitian

Pada tahap ini, dilakukan perancangan sistem SNMP *agent simulator* yang diterapkan pada Ubuntu Gutsy 7.10 sebagaimana ditunjukkan pada Gambar 3. Perancangan sistem SNMP *agent simulator* baik untuk BS maupun SS adalah sama. Yang membedakan keduanya adalah MIB pada device WiMAX tersebut. Pada device WiMAX, terdapat beberapa bagian diantaranya:

- *snmpd Master Agent*, merupakan SNMP daemon dari Net-SNMP yang berjalan sebagai *agent* utama.
- *wman Subagent*, merupakan daemon yang berjalan dibawah *agent* utama yang akan melakukan permintaan data pada MySQL Database.
- **WiMAX Device MIB**, merupakan MIB dari Net-SNMP yang berisi mengenai device WiMAX.
- *snmpd configuration*, merupakan konfigurasi utama yang tersimpan yang akan digunakan oleh *snmpd Master Agent*.
- *MySQL Database as Backend*, merupakan informasi yang menyimpan struktur dan isi data yang dirancang dan dibuat pada database MySQL sebagai *backend* berdasarkan MIB device WiMAX baik BS maupun SS yang digunakan untuk sumber informasi utama data simulasi.
- PHP sebagai *middleware* untuk menginterpretasikan permintaan dari *web browser* agar dapat mengakses data pada MySQL Database.
- Apache sebagai perangkat lunak *web server*.

Setelah gambaran hasil perancangan diperoleh, maka dilakukan tahapan pembuatan sistem SNMP *agent simulator*. Berikut ini adalah tahapan proses pembuatan sistem SNMP *agent simulator*:

- Melakukan konfigurasi sistem pada Linux Ubuntu Gutsy 7.10 sebagai SNMP *agent simulator*.
- Melakukan proses pembuatan MIB Device WiMAX.
- Melakukan perancangan MySQL *Database*.
- Melakukan perancangan sistem pengaturan data simulasi.
- Implementasi SNMP *agent simulator*.

Setelah sistem SNMP *agent simulator* dilakukan maka dilanjutkan dengan implementasi sistem pengaturan data simulasi dan pengujian sistem dengan menggunakan aplikasi *command line* snmpwalk dan aplikasi OpenNMS.

D. Hasil dan Pembahasan

Pada setiap *managed system* yang terpasang SNMP *agent* seperti *router*, *switch*, *web host*, komputer *desktop*, *network printer* dan sebagainya, masing-masing menjalankan suatu proses layanan yang disebut *daemon*. Setelah dilakukan instalasi Net-SNMP pada sistem operasi Ubuntu Gutsy 7.10, maka proses SNMP *agent* yang berjalan pada sistem yang digunakan sebagai *agent simulator* adalah *snmpd daemon*. Pengembangan SNMP *agent* dengan melakukan kompilasi secara menyeluruh merupakan proses yang cukup menyita banyak waktu. Sebab ketika melakukan pembuatan kode program suatu modul MIB baru, kemudian dilakukan proses kompilasi secara menyeluruh pada setiap modul MIB yang ada pada *source code* Net-SNMP akan memerlukan waktu yang lebih lama. Dan apabila terjadi kesalahan (*error*) pada kode program yang dibuat, proses kompilasi akan berhenti sehingga harus dilakukan perbaikan pada kode program tersebut baru kemudian proses kompilasi dimulai kembali dari awal. Cara yang cukup cepat dan mudah dalam pengembangan dan proses pembuatan kode program tersebut adalah dengan menggunakan model *agentX*

dimana *subagent* yang melayani suatu *tree object* ID tertentu akan berjalan pada proses yang berbeda. Artinya apabila *subagent* dijalankan, maka akan ada *daemon* baru yang berjalan pada sistem.

Proses Pembuatan MIB Standar IEEE 802.16f

Pada standar IEEE 802.16f, terdapat informasi mengenai MIB untuk BS dan SS yang didefinisikan dalam dua modul MIB yaitu *wmanIfMib* dan *wmanDevMib*. Proses pembuatan SNMP *agent* dalam penelitian ini menggunakan *wmanDevMib* dan *wmanIfMib* yang diperlukan untuk mendefinisikan *management object* dari *device* WiMAX baik itu BS maupun SS yang dapat digantikan oleh suatu *simulator* dalam hal ini yaitu sistem operasi Ubuntu Gutsy 7.10. Modul MIB dari standar IEEE 802.16f dapat di-*download* dari berbagai sumber, dan dalam penelitian ini *source code* modul MIB didapat dari website IEEE 802.16. Setelah *file* modul MIB *wmanDevMib* dan *wmanIfMib* didapatkan, yaitu WMAN-DEV-MIB-11-30.txt dan WMAN-IF-MIB-11-30.txt yang didefinisikan dalam format ASN.1, kemudian kedua *file* tersebut dipindahkan ke dalam lokasi program Net-SNMP yaitu */usr/local/snmp/share/snmp/mibs*. Setelah itu menjalankan aplikasi *command line* “*snmptranslate -Tp -IR .iso.std*” pada lokasi */usr/local/snmp/bin*. Hal ini dilakukan untuk menampilkan *tree* atau struktur MIB dari modul MIB tersebut yang kemudian akan dilakukan proses kompilasi. Maka hasil keluaran yang muncul dari *snmptranslate* adalah:

```

root@wimaxunila:/usr/local/snmp/bin#./snmptranslate -Tp -IR .iso.std
+--std(0)
|
+--iso8802(8802)
|
+--wman(16)
|
+--wmanDevMib(1)
|
+--wmanDevMibObjects(1)
| |
| | +--wmanDevBsObjects(1)
| | |
| | | +--wmanDevBsSoftwareUpgradeTable(1)
| | | |
| | | | +--wmanDevBsSoftwareUpgradeEntry(1)
| | | | | Index: wmanDevBsDeviceIndex
| | | | |
| | | | +-- ---- INTEGER wmanDevBsDeviceIndel)
| | | | | Range: 1..10
| | | | +-- -R-- String wmanDevBsVendorId(2)
| | | | | Size: 2..256
| | | | +-- -R-- String wmanDevBsHwId(3)
| | | | | Size: 2..256
| | | | +-- -R-- String wmanDevBsCurrentSwVersion(4)
| | | | | Size: 2..256
| | | | +-- -RW- String wmanDevBsDownloadSwVersion(5)
| | | | | Size: 2..256
| | | | +-- -RW- String wmanDevBsUpgradeFileName(6)
| | | | | Size: 2..256
| | | | +-- -RW- EnumVal wmanDevBsSoftwareUpgradeAdminState(7)
| | | | | Values: null(0), download(1), activate(2)
| | | | +-- -R-- INTEGER wmanDevBsDownloadSwProgress(
| | | | | Range: 0..100
| | | | +-- -R-- String wmanDevBsSoftwareUpgradeTimeStamp(9)
| | | | Textual Convention: DateAndTime
| | | | Size: 8 | 11
| | |
| | | .....
| | | .....
| | | .....
| | | +--wmanDevMibCompliances(2)
| | | |
| | | +--wmanDevMibCompliance(1)

```

Dari hasil keluaran tersebut, dapat dilihat *tree* atau struktur modul MIB. Setelah memastikan bahwa struktur MIB sudah sesuai dengan standar IEEE 802.16f, kemudian modul MIB wmanDevMib dan wmanIfMib dikompilasi dengan menjalankan aplikasi *command line* mib2c. Aplikasi ini mampu membaca suatu *file* MIB dan menghasilkan *skeleton code* untuk pengembangan *agent* nantinya

Penerapan Sistem SNMP Agent Simulator

Setelah melakukan konfigurasi pada SNMP *agent* dan MySQL Server, serta

pembuatan MIB *device* WiMAX, selanjutnya adalah menerapkan sistem SNMP *agent simulator* sebagai BS ataupun SS pada SNMP *agent* yang telah dikonfigurasi. Untuk mengidentifikasi suatu *device* apakah sebagai BS atau SS, maka perlu dilakukan pembuatan suatu fungsi spesifik dan penentuan nilai-nilai identitas objek *device* tersebut. Kemudian *device* tersebut umumnya “mengiklankan” fungsi spesifik tersebut melalui suatu protokol NMS seperti SNMP. Sehingga identifikasi terhadap suatu fungsi, salah satunya yaitu fungsi *hardware*, bisa dilakukan secara tepat. Dalam konvensi

yang umum yang digunakan oleh NMS, nilai objek yang diambil untuk mengidentifikasi fungsi spesifik salah satunya adalah *Object Identifier* (OID), yaitu `sysObjectId.0` yang digunakan sebagai identifikasi jenis *hardware*.

Setelah sebelumnya melakukan analisa struktur MIB standar IEEE 802.16f dengan menampilkan *tree* modul MIB dengan aplikasi `snmptranslate`, maka hasil keluaran dapat dilihat bahwa OID

`wmanDevBsObjects` adalah

`.1.0.8802.16.1.1.1`. Nilai OID tersebut menggantikan nilai yang sebelumnya telah dikonfigurasi pada isi *file* `snmpd.conf`, yaitu:

```
syslocation SSC Universitas Lampung
sysdescr Device BS WiMAX
syscontact wimaxunila@unila.ac.id
sysObjectId .1.0.8802.16.1.1.1
rocommunity public
rwcommunity wimaxdev
```

Dalam perubahan yang dilakukan, selain nilai `sysObjectId`, nilai `sysdescr` dan `rwcommunity` juga dilakukan perubahan. Hal ini dilakukan untuk lebih memastikan bahwa *SNMP agent simulator* merupakan suatu *device* BS WiMAX. Apabila perubahan pada isi *file* `snmpd.conf` selesai dilakukan, kemudian `snmpd daemon` di-*restart* untuk menjalankan hasil perubahan konfigurasi yang terakhir diubah. Adapun pembuatan *SNMP agent simulator* yang akan diterapkan sebagai SS, prosesnya tidak jauh berbeda dengan penerapan pada BS.

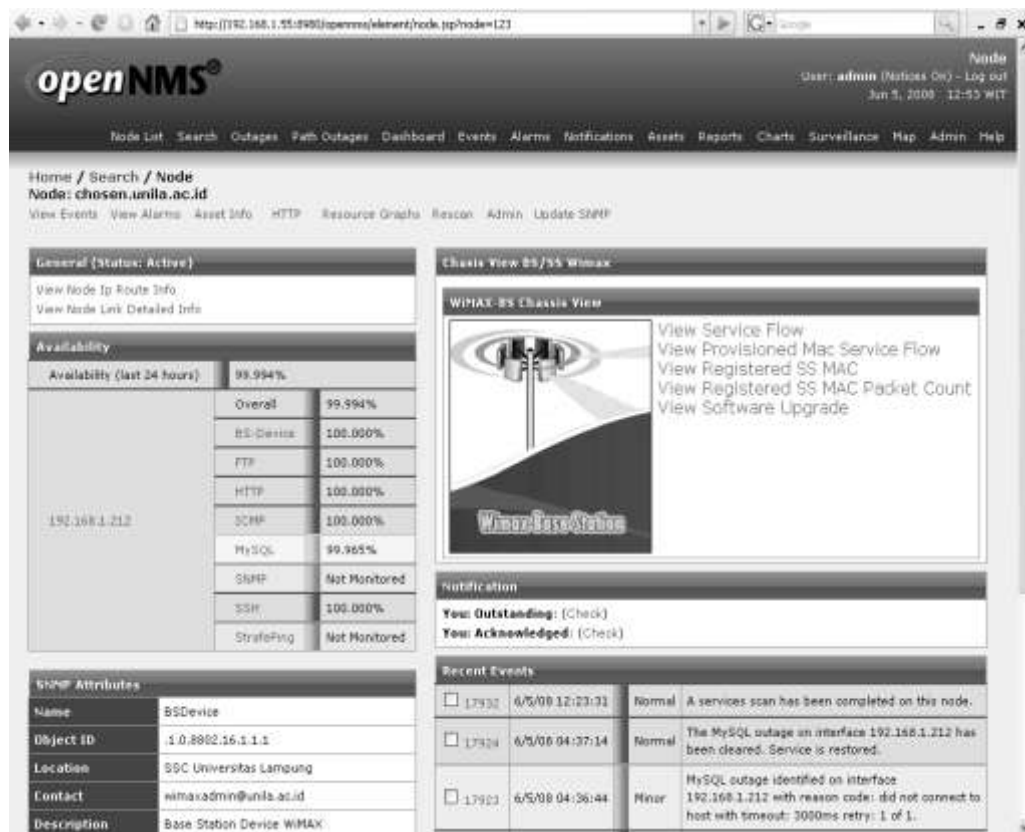
Implementasi dan Pengujian Sistem SNMP Agent Simulator

Setelah melakukan perancangan sistem pengaturan data simulasi dan MySQL *database*, kemudian mengimplementasikan aplikasi sistem pengaturan data simulasi. Jika implementasi aplikasi sistem pengaturan data simulasi telah dilakukan, kemudian dilakukan pengujian terhadap *SNMP agent simulator* menggunakan

aplikasi *command line snmpwalk* dari perangkat lunak Net-SNMP dan menggunakan aplikasi OpenNMS. Sistem pengaturan data simulasi ini dibuat dengan menggunakan bahasa pemrograman PHP. Untuk *database* sebagai sumber data digunakan MySQL untuk data simulasi pada *SNMP agent simulator*. Perancangan yang telah dilakukan, hanya terdapat seorang pengguna pada sistem ini yaitu *administrator*.

Aplikasi *command line snmpwalk* merupakan salah satu perintah untuk mendapatkan *tree* informasi suatu *Object Identifier* (OID). Semua variabel dalam *subtree* yang diberikan OID akan di-*query* dan nilainya akan disajikan kepada pengguna. Dalam pengujian ini, perintah `snmpwalk` digunakan untuk menampilkan informasi mengenai *tree* dan variabel *subtree* modul `wmandevbssoftwareupgrade`. Data modul `wmandevbssoftwareupgrade` telah disiapkan ketika proses memasukkan data simulasi dengan menggunakan aplikasi sistem pengaturan data simulasi.

Pada pengujian ini, telah dijalankan aplikasi OpenNMS yang ditujukan sebagai *managing system* pada *SNMP agent simulator* yang telah diimplementasikan. Aplikasi OpenNMS yang digunakan, telah dioptimalkan untuk *device* WiMAX baik sebagai BS *device* maupun SS *device*. OpenNMS yang telah dijalankan harus bisa menghubungkan antara OpenNMS sebagai *managing system* dengan *SNMP agent simulator* sebagai *managed device* dalam jaringan. Dalam pengujian ini, kemudian pengguna memasuki menu "Node List" untuk melihat semua *device* yang telah dipilih untuk dilakukan pemantauan (*monitoring*) yang salah satunya adalah *SNMP agent simulator* yang telah dikonfigurasi sebagai BS *device*, yaitu "chosen.unila.ac.id".

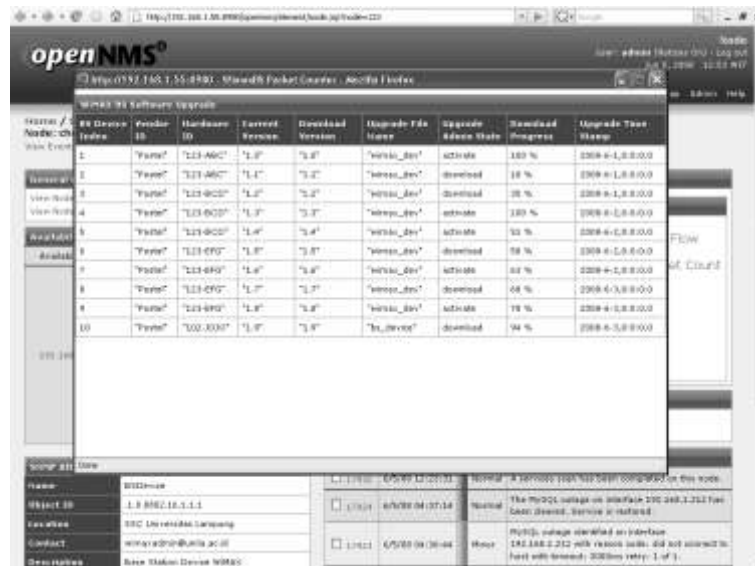


Gambar 5 Menu Node

Setelah memilih SNMP *agent simulator* chosen.unila.ac.id, maka akan ditampilkan menu mengenai informasi lengkap *device* tersebut. Apabila sysObjectID pada sistem yang didapatkan merupakan OID sebagai untuk BS device WiMAX, seperti yang telah dilakukan dalam penerapan SNMP *agent simulator* sebagai BS *device*, maka akan tampil menu "WiMAX-BS Chasis View" (lihat Gambar 5). Pada menu "SNMP Attributes", terdapat informasi mengenai "Object ID", "Location", "Contact" dan "Description". Dan nilai-nilai untuk informasi tersebut sesuai dengan nilai-nilai yang telah dimasukkan dalam isi *file snmpd.conf* untuk penerapan SNMP *agent simulator* sebagai BS *device*, yaitu "sysObjectID", "syslocation", "syscontact" dan "sysdescr".

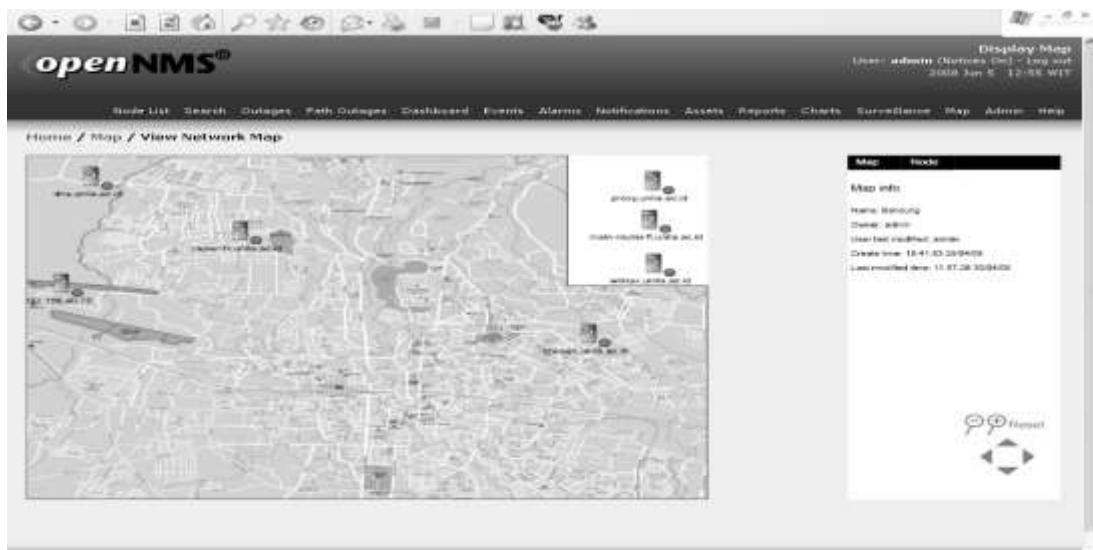
Untuk mendapatkan informasi mengenai data suatu modul MIB, dalam pengujian ini merupakan data simulasi wmandevbssoftwareupgrade, maka melakukan pemilihan menu "View Software Upgrade" yang telah disediakan pada menu "Node". Pada gambar 6 ditampilkan nilai-nilai setiap kolom untuk data simulasi wmandevbssoftwareupgrade yang telah dimasukkan dengan menggunakan sistem pengaturan data simulasi.

Pada aplikasi OpenNMS terdapat juga menu "Map" yang digunakan untuk memantau status seluruh *device* yang telah dimasukkan ke dalam suatu peta (*map*). Sehingga dapat dilakukan pemantauan terhadap seluruh *device* tersebut berdasarkan tata letak yang sesuai dengan lokasi pada peta yang telah ditampilkan sebagai *background* sebagaimana ditunjukkan oleh Gambar 7



Item	Device Type	Vendor	Hardware ID	Current Version	Download Version	Upgrade File Name	Upgrade Action State	Download Progress	Upgrade Time Range
1	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	activate	100 %	2008-6-18 0:00
2	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	download	10 %	2008-6-18 0:00
3	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	download	10 %	2008-6-18 0:00
4	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	activate	100 %	2008-6-18 0:00
5	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	activate	100 %	2008-6-18 0:00
6	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	download	10 %	2008-6-18 0:00
7	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	activate	100 %	2008-6-18 0:00
8	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	download	10 %	2008-6-18 0:00
9	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	activate	100 %	2008-6-18 0:00
10	Router	Yamaha	YR13-AGC	1.3	1.3	YR13-AGC	download	10 %	2008-6-18 0:00

Gambar 6 Menu View Software Upgrade



Gambar 7. View Network Map

E. Simpulan

Pembuatan MIB standar IEEE 802.16f dengan melalui proses kompilasi menghasilkan *skeleton code* harus dimodifikasi karena kode yang dihasilkan hanya berupa kerangka dari struktur MIB tersebut. Modul MIB yang diuji *wmandevbssoftwareupgrade* yang merupakan salah satu MIB standar IEEE 802.16f dapat dijalankan dengan optimal dengan menggunakan sistem pengaturan data simulasi dan aplikasi OpenNMS. Hasil perancangan sistem SNMP *agent*

yang diuji dengan menggunakan sistem operasi Ubuntu Gutsy 7.10 sebagai *simulator*, dapat berjalan dengan baik. Kendala yang dihadapi dalam proses pembuatan sistem SNMP *agent simulator* adalah pembuatan kode program untuk MySQL API pada bagian bagaimana menghubungkan antara modul MIB dengan data simulasi yang tersimpan dalam MySQL *database*. Dengan menggunakan AgentX sebagai *subagent* dapat mengefisiensikan proses kompilasi kode program yang telah dibuat.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih atas bantuan dana hibah dari Direktorat POSTEL dengan SPPT No. 02J/DJPT.5/KOMINFO/II/2008.

Daftar Pustaka

- [1] Surat Perintah Pelaksanaan Tugas Nomor: 02J/DJPT.5/KOMINFO/II/2008 Tentang Program Lanjutan Dukungan Penelitian dan Pengembangan Produk Telekomunikasi, Direktorat Jenderal Pos dan Telekomunikasi, Depkominfo RI
- [2] IEEE std 802.16e – 2005, IEEE standard for local and Metropolitan Area Network, Part 16 : Air Interface for Fixed and Mobile Broadband Wireless Access System <http://www.ieee.org/> diakses 14 desember 2007
- [3] Internetworking Technology Handbook. *Network Management Basics*. Cisco System. <http://www.cisco.com/en/US/docs/inter-networking/technology/handbook/NM-Basics.html> diakses 31 januari 2008
- [4] <http://www.cisco.com/en/US/docs/inter-networking/technology/handbook/SNMP.html>, diakses 31 januari 2008).
- [5] Official Documentattion, <http://www.opennms.org/> diakses 20 Februari 2008
- [6] SNMP and OpenNMS. www.haifux.org/lectures/118/2_opennms.pdf, diakses 17 maret 2008
- [7] FCAPS Overview: White paper. Flextronics Software System. <http://www.futsoft.com/pdf/fcapswp.pdf> diakses 17 mei 2008
- [8] Gunawan Wibisono dkk. 2007. *WiMAX Teknologi Broadband Wireless Access (BWA) Kini dan Masa Depan*. Informatika. Jakarta
- [9] <http://www.net-snmp.org/tutorial/tutorial-5/>. NET-SNMP v5.x Tutorial (diakses Maret 2008).