

BAB II

LANDASAN TEORI

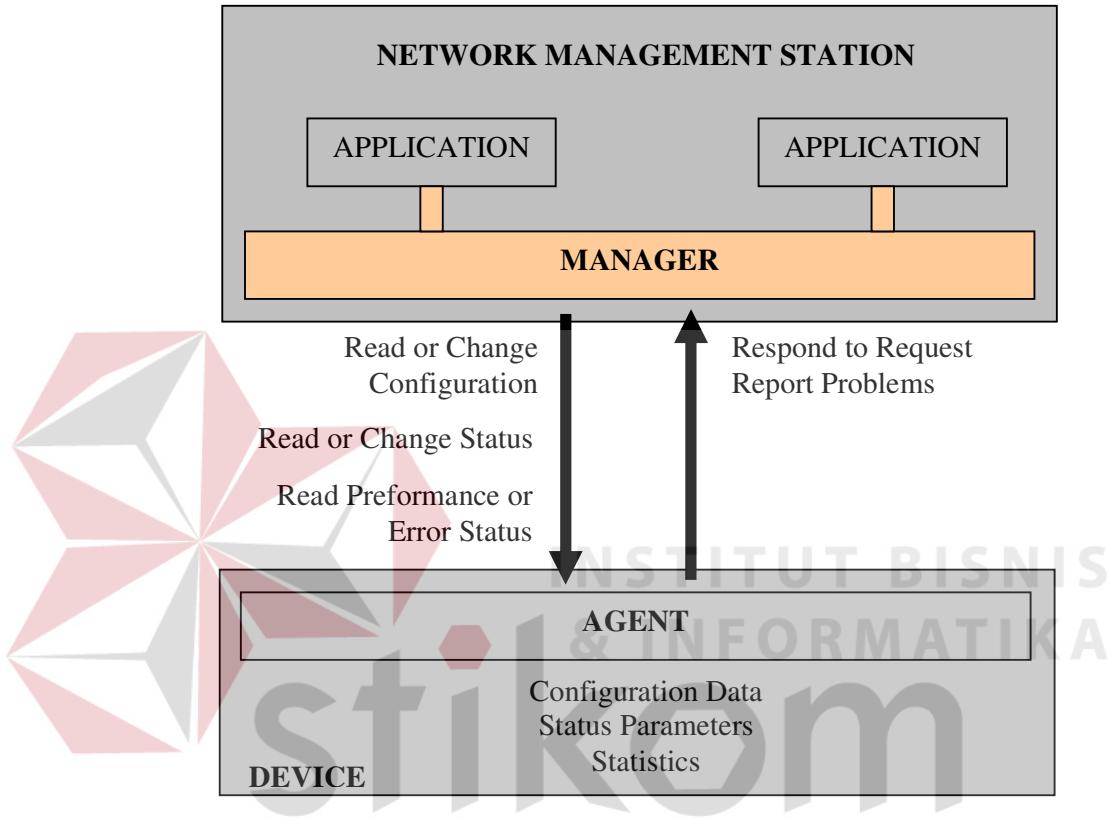
2.1 SNMP (Simple Network Management Protocol)

Pada awalnya sekitar tahun 1988 beberapa kelompok orang membutuhkan suatu peralatan yang dapat mengontrol sistem kerja jaringan melalui TCP/IP jaringan. Dan untuk itu maka beberapa orang tersebut mulai menyusun konsep untuk menyelesaikan masalah tersebut diatas dan diantara orang – orang tersebut adalah Keith McCLOGHRIE, Marshall ROSE, Jeffrey D. CASE, Mark FEDOR, Martin LEE SCHOFFSTALL, dan James R. DAVIN.

Mereka bersama – sama bekerja sebagai team yang kemudian membentuk suatu standarisasi yang dibutuhkan untuk manajemen jaringan. Standar ini diberikan oleh suatu badan khusus (*IAB - Internet Architecture Board*) yang menangani bidang ini yaitu RFC 1157 tentang *Simple Network Management Protocol* atau SNMP.

SNMP ini adalah sebuah protocol yang digunakan untuk memonitor peralatan – peralatan jaringan seperti server, router, switch dan peralatan server lain. Dan untuk melakukan monitoring tentunya SNMP juga mendukung hubungan client server dimana program client disebut dengan *Network Manager* yang membentuk *virtual connection* ke server yang disebut *SNMP Agent*. Agent ini akan terpasang pada setiap peralatan (server, router, switch atau peralatan lain) yang mana Agent ini akan menerima message (data) dari Manager. Agent akan menerima message dan mengirim tanggapan ke Manager. Agent tidak perlu menunggu permintaan untuk dilaksanakan jika ada masalah tertentu yang sangat

serius, dan Agent akan mengirimkan peringatan kepada Manager untuk memberikan tanggapan. Dibawah ini adalah interaksi antara Agent dan Manager.



Gambar 2.1. Gambar interaksi Agent dan Manager

Manager disini akan berfungsi sebagai stasiun manajemen yang akan mengirim pesan ke Agent dan menerima Trap dan Response.

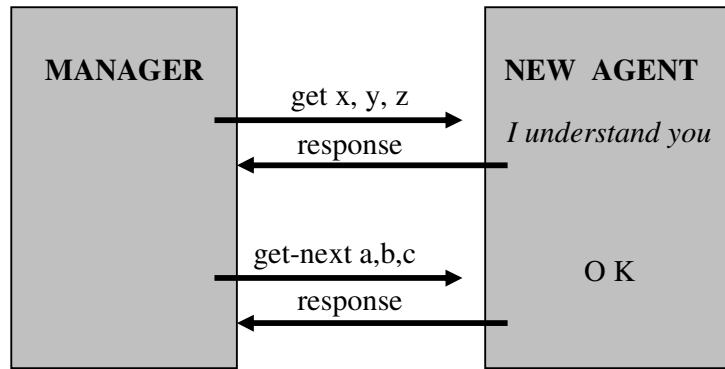
2.1.1 Perintah dasar SNMP

Untuk memenuhi dari apa yang dibutuhkan untuk mendapatkan data pada peralatan yang dimonitor, SNMP memasukkan beberapa perintah dasar ke dalam

library-nya diantaranya adalah :

1. *Get-request*, digunakan untuk meminta data pada MIB variable.
2. *Get-next-request*, digunakan untuk meminta data selanjutnya secara berurutan pada MIB variable. Perintah ini sering digunakan pada saat pengambilan data pada table. Setelah menggunakan get-request, maka untuk melanjutkan pengambilan data selanjutnya dalam table digunakan get-next-request.
3. *Set-request*, digunakan untuk memperbaharui (*update*) nilai dalam MIB.
4. *Get-response*, berfungsi untuk memberikan respon atau tanggapan setelah perintah get-request, get-next-request dan set-request.
5. *Trap*, digunakan untuk memberikan peringatan jika aplikasi atau peralatan berjalan tidak normal.

Pada dasarnya, Manager tidak mengetahui variabel apa saja yang ada, manager akan menggunakan fungsi – fungsi tersebut diatas dan kemudian akan mendapatkan tanggapan. Tanggapan ini lah yang dimengerti oleh Manager sehingga data bisa digunakan. Secara garis besar dapat digambarkan seperti dibawah ini.



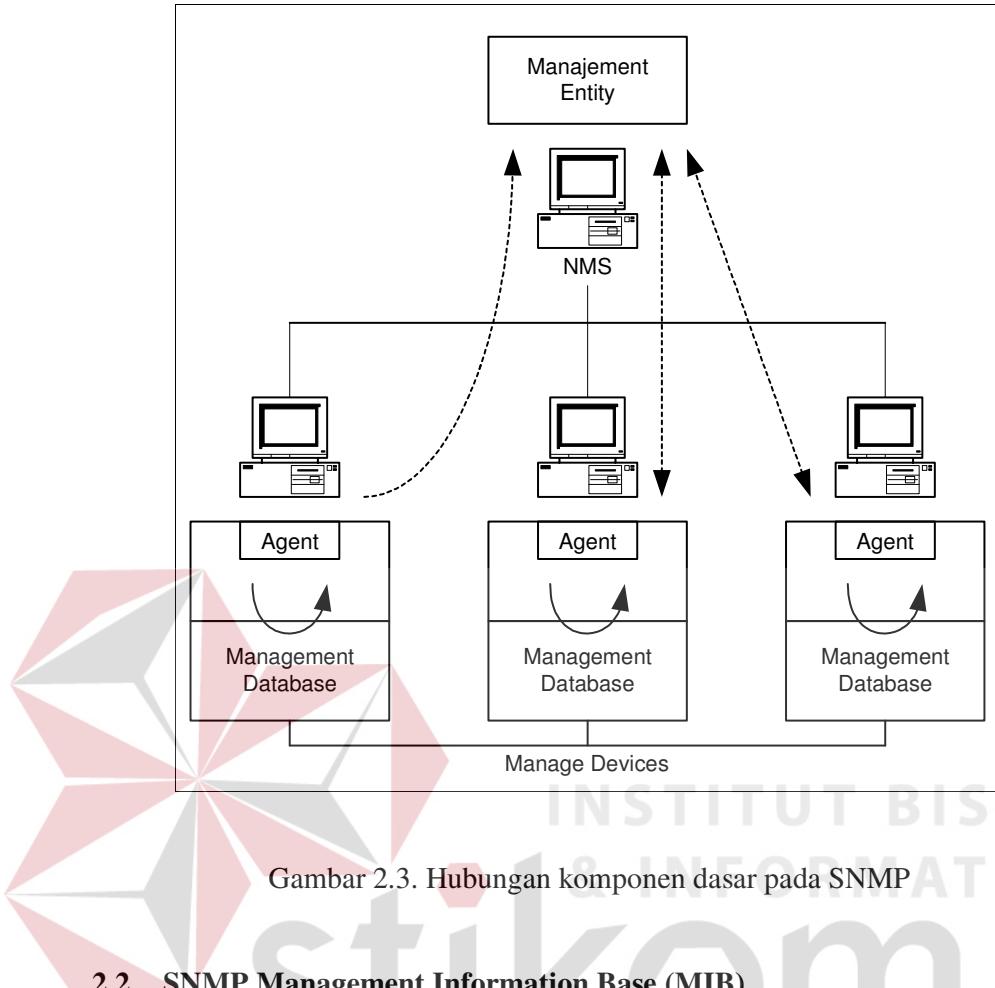
Gambar 2.2. Interaksi perintah dasar pada SNMP

2.1.2 Komponen dasar SNMP

SNMP memiliki tiga komponen dasar yaitu *Managed Device*, *Agent*, dan *Network Management Systems (NMS)*. Managed Device adalah titik jaringan yang berisi SNMP Agent dan berada pada manajemen jaringan. Managed Device menyimpan dan mengumpulkan manajemen informasi dan menggunakan informasi yang berada pada NMS untuk keperluan SNMP. Managed Device terkadang disebut element – element jaringan, bisa berupa router, server, switch dan bridge, hubs, komputer host, atau printer.

Agent adalah software yang tersimpan pada Managed Device. Sehingga Agent ini merupakan tempat berkumpulnya informasi dan menerjemahkan informasi tersebut agar sesuai dengan kebutuhan SNMP.

NMS akan menjalankan aplikasi untuk memonitor dan mengontrol Managed Device. Sehingga NMS ini merupakan pengguna terbesar (proses dan memori) yang dibutuhkan untuk manajemen jaringan. Dibawah ini adalah penggambaran hubungan antara komponen – komponen dasar.



2.2 SNMP Management Information Base (MIB)

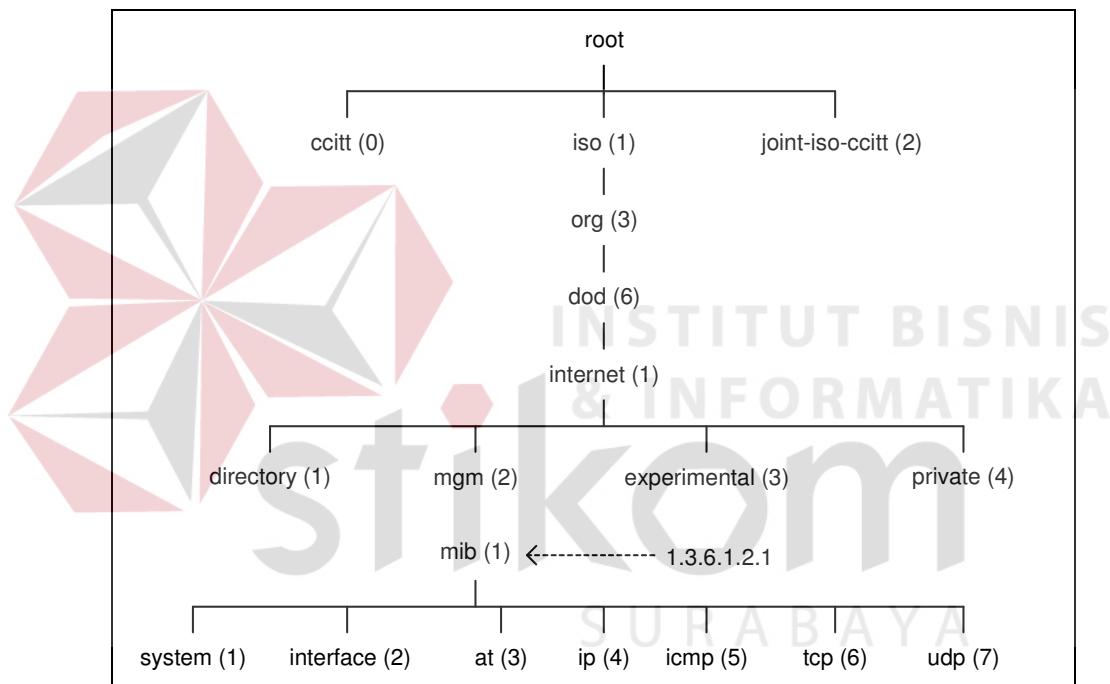
Management Information Base (MIB) menyimpan seluruh informasi secara berurutan. Untuk mengakses MIB harus menggunakan protocol SNMP. Protocol SNMP merupakan objek dan dapat mengidentifikasi suatu object.

Managed Object adalah nomer yang mengidentifikasi karakteristik dari *managed object*. Managed Object juga termasuk satu atau lebih object yang pada pokoknya merupakan variable.

Dua tipe dari Managed Object adalah *scalar* dan *tabular*. Scalar objek di definisikan sebagai objek tunggal. Sedangkan tabular objek di definisikan sebagai hubungan antara objek – objek yang membentuk suatu group dalam table MIB.

Sebagai contoh, Managed Object dari “Input” adalah sebuah scalar objek yang berisi objek instance tunggal, nilai integer di identifikasi sebagai nomer total dari *input AppleTalk* paket pada router interface.

Object Identifikasi atau *Object ID* adalah unik *identify* sebuah Managed Object dalam MIB hirarki. MIB hirarki dapat digambarkan sebagai tree dimana level – level dari hirarki ditentukan oleh organisasi – organisasi yang berbeda.



Gambar 2.4. Hirarki MIB

Pada level atas dari MIB, object ID dimiliki oleh standar organisasi yang berbeda, dan bawahnya merupakan object ID yang diberikan oleh organisasi yang bersangkutan. Vendor dapat menentukan cabang dan memasukkan Managed Object untuk product-nya sendiri.

Melanjutkan contoh diatas dimana Managed Object dari “system” dapat ditentukan sehingga membentuk urutan indentify yang unik. Nama objek: *iso.org.dod. internet.mgm.mib.system* dan unik objeknya dapat di definisikan sebagai berikut : *1.3.6.1.2.1.1*

2.3 Format Message SNMP

SNMPv1 memiliki dua bagian pada datanya. Bagian pertama berisi header yaitu versi dan nama komuniti (*community name*), dan bagian kedua berisi data utama yaitu SNMP *Protocol Data Unit (PDU)*. Untuk lebih jelasnya, gambaran format message dari SNMPv1 adalah sebagai berikut :



Gambar 2.5. Format data SNMPv1

Bagian *version* berikan versi dari SNMP yang digunakan dan *community name* mendefinisikan group dari NMS (*Network Management System*). Sedangkan SNMP PDU berisi perintah – perintah (get, set dan lainnya) dan operasi yang mengindikasikan penggunaan object dalam transaksi. Detail SNMP PDU digambarkan seperti berikut :

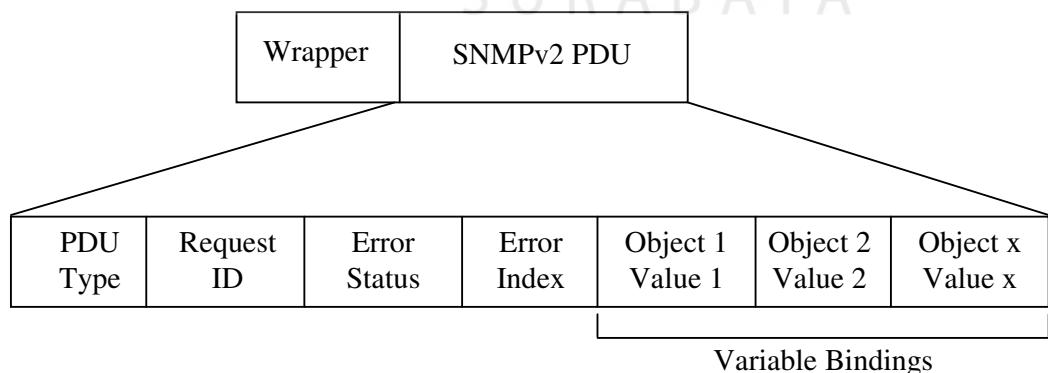
PDU Type	Request ID	Error Status	Error Index	Object 1 Value 1	Object 2 Value 2	Object x Value x
Variable Bindings						

Gambar 2.6. Detail SNMP PDU

Keterangan dari detail format SNMP PDU:

- *PDU Type* – spesifikasi tipe dari PDU.
- *Request ID* – asosiasi SNMP request ID.
- *Error Status* – indikasi nomer kesalahan (error) dan tipe kesalahan.
- *Error Index* – instan error object dari data.
- *Variable Bindings*.

Pada SNMPv2 ada sedikit perbedaan dimana pada format data yang terdapat *wrapper* di bagian kepala data. Wrapper ini memiliki fungsi sebagai autentikasi dan pelindung informasi. Secara jelas format data SNMPv2 digambarkan sebagai berikut:



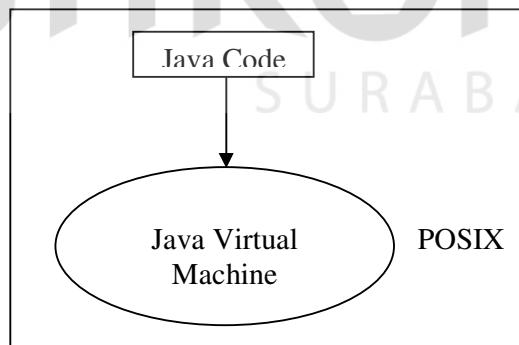
Gambar 2.7. Format data SNMPv2

2.4 Bahasa Pemrograman

Pengembangan suatu perangkat lunak dilakukan dengan mempertimbangkan bahasa pemrograman yang digunakan. Dasar penentuannya adalah terletak pada kemampuan bahasa tersebut dalam berinteraksi maupun dalam berkesesuaian dengan sarana pendukung lainnya. Kehandalan suatu bahasa pemrograman dalam mengatasi berbagai kendala yang mungkin akan dihadapi dalam penyusunan suatu perangkat lunak inipun menjadi pertimbangan yang kuat dalam pemilihan suatu bahasa pemrograman. Pada pembuatan perangkat lunak ini, bahasa pemrograman yang digunakan adalah Java khususnya Java Applet.

2.4.1 Java

Java adalah termasuk *Object-Oriented Programming language*. Dan sintak-sintak java yang telah di *compile* akan menghasilkan *byte code* yang kemudian akan dijalankan pada *virtual machine* pada *POSIX*.

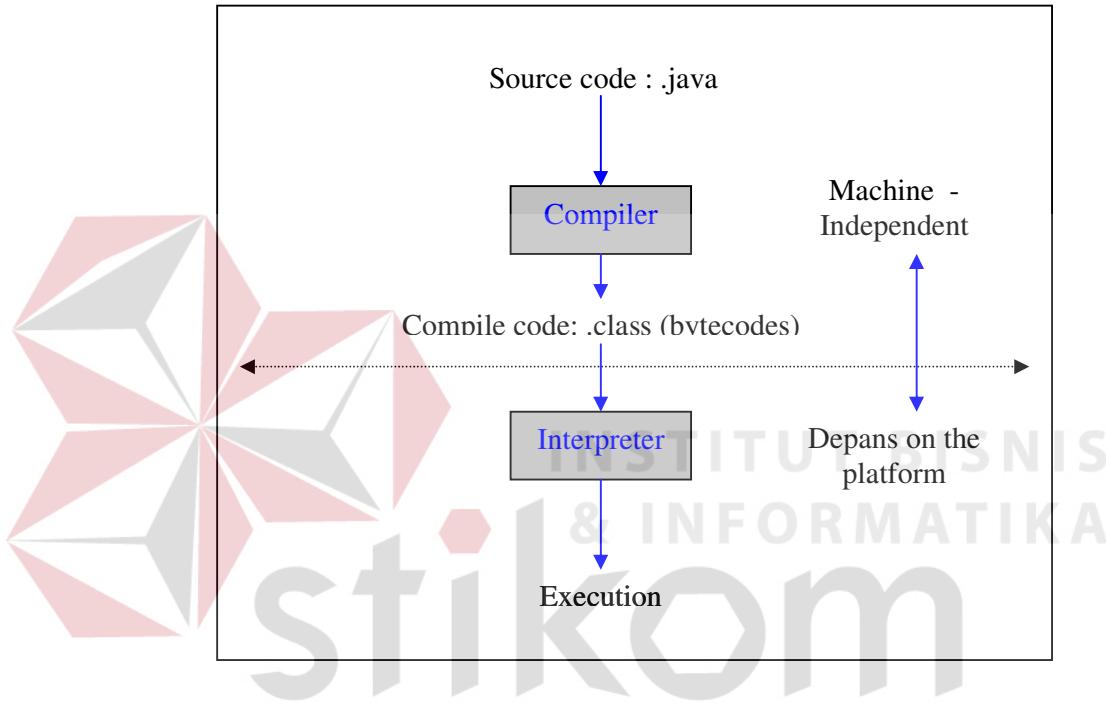


Gambar 2.8. Struktur Virtual Machine

Keuntungannya adalah dengan menggunakan virtual machine bisa dibuat sesederhana yang diinginkan dan juga penerjemahan oleh mesin - mesin yang

menyebabkan aplikasi java memdukung *multi platform*, ini adalah kelebihan yang harus dimiliki oleh bahasa pemrogramman.

Java code (*bytecodes*) ini sangat kecil yang menyebabkan java sangat ideal dalam mengirim data di jaringan.



Gambar 2.9. Struktur compiler java

2.4.2 Applet

Java adalah *Object – Oriented Programming Language* yang dibangun oleh Sun Microsystems dimana telah disediakan oleh Sun sebuah library standar yang menyediakan objek – objek. Objek ini sangat mudah digunakan. Sebuah program java aplikasi akan dijalankan *stand-alone* dari *console* atau java applet yang akan dapat dijalankan pada web browser. Java aplikasi dan applet memiliki

perbedaan dimana java aplikasi memiliki method *main()* sedangkan applet memiliki method *init()*.

Source code dari java applet diletakkan pada sebuah file dengan ekstensi *.java*. Dan kode java ini jika di compile ke bentuk bytecode yang akan membentuk sebuah file dengan ekstensi *.class*. File berekstensi *.class* ini nantinya yang akan dijalankan dengan menggabungkan ke code HTML yang telah ada.

Sebagai contoh dari applet tag adalah sebagai berikut:

```
<applet code="FileApplet.class" with="170" height="150">
```

Java Applet ini akan dijalankan oleh browser ketika browser memanggil kode-kode HTML yang berisikan perintah/tag applet.

