

# APLIKASI PEMANTAU STATISTIK LAYANAN-LAYANAN TCP DAN UDP BERBASIS SNMP++ PADA SEBUAH JARINGAN AREA LOKAL

Arthop Sagita<sup>1</sup>, Agung Budi Prasetyo<sup>2</sup>, R. Rizal Isnanto<sup>2</sup>  
Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang

**ABSTRAK** - Seiring dengan perkembangan jaringan komputer dan teknologinya, berkembang pula mekanisme manajemen terhadap jaringan komputer. Khusus untuk jaringan TCP/IP, IETF (Internet Engineering Task Force) selaku badan pekerja Internet yang bertugas menetapkan standar-standar Internet, telah mengeluarkan sebuah standar mengenai manajemen jaringan komputer TCP/IP, yaitu SNMP (Simple Network Management Protocol). Untuk menerapkan semua aturan manajemen jaringan komputer yang terdapat pada protokol SNMP, banyak pihak yang mencoba untuk membuat kerangka-kerja (framework) SNMP yang membungkus protokol SNMP ke dalam antarmuka pemrograman aplikasi (application programming interface). Salah satu antarmuka pemrograman aplikasi SNMP yang telah lahir dan sampai sekarang terus mengalami pengembangan adalah SNMP++ yang berbasis bahasa C++. Dengan berbasis bahasa C++, keuntungan pemrograman berorientasi objek dapat diperoleh dalam penggunaan SNMP++. Selain itu, SNMP++ pun diedarkan secara cuma-cuma. Karena keunggulannya tersebut, sebuah penelitian perlu dilakukan untuk mengkaji lebih dalam dan menerapkan penggunaan SNMP++ dalam membuat sebuah aplikasi manajemen jaringan TCP/IP, khususnya statistik layanan-layanan TCP dan UDP.

Sebuah komunikasi antara dua buah komputer yang tersambung melalui jaringan TCP/IP dilakukan dengan menggunakan protokol TCP atau UDP. Untuk melengkapi kebutuhan manajemen jaringan, SNMP juga mengatur penanganan statistik layanan-layanan TCP dan UDP pada sebuah hubungan TCP/IP. Salah satu informasi yang dapat diperoleh dari statistik ini yaitu informasi mengenai port-port TCP dan UDP yang aktif dan status koneksinya. Untuk membuat sebuah aplikasi yang menampilkan informasi layanan-layanan TCP dan UDP, proses penelitian dilakukan melalui tahap perancangan aplikasi yang kemudian dilanjutkan dengan tahap implementasi perancangan ke dalam kode-kode pemrograman menggunakan Microsoft<sup>TM</sup> Visual C++ 6.0. Untuk memastikan aplikasi dapat berjalan sesuai dengan yang dikehendaki, tahap berikutnya yang dilakukan pada penelitian ini adalah menguji aplikasi melalui metode pengujian kotak-hitam (black-box testing).

Dari hasil penelitian, aplikasi yang dibuat berbasis SNMP++ dapat menjalankan fungsi utamanya dalam menampilkan statistik layanan-layanan TCP dan UDP. Bahkan, dengan konsep penggunaan kembali (reusability), proses pembuatan aplikasi menjadi relatif lebih mudah dilakukan melalui penggunaan SNMP++ daripada proses pembuatan aplikasi yang tidak memanfaatkan kelebihan pemrograman berorientasi objek. Namun demikian, dalam penelitian ditemukan adanya perbedaan keluaran antara aplikasi yang dibuat dengan aplikasi pembanding (*netstat.exe*). Hal ini kemungkinan disebabkan oleh waktu pemantauan yang tidak dapat dilakukan secara bersamaan oleh kedua aplikasi.

**Kata kunci** : Manajemen jaringan komputer, TCP/IP, TCP, UDP, SNMP, SNMP++

## I. PENDAHULUAN

### A. LATAR BELAKANG

Jaringan komputer dipergunakan oleh berbagai pihak karena beberapa alasan. Secara umum, banyak pihak menggunakan jaringan komputer untuk mempermudah pekerjaannya. Sejalan dengan perkembangan jaringan komputer, berkembang pula mekanisme yang melakukan proses manajemen terhadap jaringan komputer yang bertujuan untuk merencanakan, memantau, dan mengatur komponen sistem dan jaringan komputer<sup>[3]</sup>. Mulai dari aktivitas yang memantau hal-hal yang terjadi di balik sebuah jaringan komputer, atau pencatatan secara berkala unjuk kerja sebuah jaringan komputer, sampai dengan aktivitas yang dapat mengatur komponen-komponen dari sebuah jaringan komputer, merupakan contoh kegiatan manajemen jaringan komputer.

Proses manajemen tersebut dapat diimplementasikan dalam sebuah sistem perangkat lunak. Sistem ini diharapkan dapat dijadikan sebagai alat bantu untuk me-

ngumpulkan informasi dan mengambil keputusan berkaitan dengan kondisi sebuah jaringan komputer.

### B. TUJUAN

Penelitian ini bertujuan untuk mempelajari dan menerapkan penggunaan pustaka SNMP++ yang berorientasi objek dalam membuat sebuah aplikasi perangkat lunak yang berfungsi menginformasikan statistik layanan-layanan TCP dan UDP dalam sebuah jaringan lokal TCP/IP.

### C. BATASAN MASALAH

Ruang lingkup penelitian Tugas Akhir ini dibatasi sebagai berikut.

1. Aplikasi yang dibuat pada penelitian ini bersifat pasif, yaitu hanya-baca (*read-only*).
2. Keamanan pengiriman data antara aplikasi dan objek yang dipantau tidak termasuk dalam bahasan Tugas Akhir ini.
3. Aplikasi tidak dapat melakukan pemantauan secara otomatis.
4. Aplikasi yang dibuat pada Tugas Akhir ini hanya khusus diujicobakan pada mesin berbasis Microsoft<sup>TM</sup> Windows 98 dan Microsoft<sup>TM</sup> Windows 2000.

<sup>1</sup> Mahasiswa Teknik Elektro Undip

<sup>2</sup> Staf Pengajar Teknik Elektro Undip

5. Pengujian aplikasi hanya dilakukan pada sebuah sistem jaringan area lokal yang terdiri atas dua buah komputer yang berbasis sistem operasi Microsoft™ Windows 32-bit yang tersambung pada jaringan TCP/IP.

## II. PROTOKOL TCP DAN UDP, SNMP, DAN SNMP++

### A. MODEL REFERENSI TCP/IP

Model referensi TCP/IP dibagi menjadi empat buah lapisan, yaitu lapisan Aplikasi (*Application layer*), lapisan Transportasi (*Transportation layer*), lapisan Internet (*Internet layer*) dan lapisan Antarmuka Jaringan (*Network Interface layer*)<sup>[5]</sup>. Model referensi TCP/IP ditunjukkan oleh Gambar 2.1.

Tiap lapisan memiliki protokol-protokolnya masing-masing. Protokol-protokol inilah yang membentuk tugas dan fungsi dari lapisan-lapisan pada TCP/IP. Masing-masing protokol memiliki tugas dan fungsi yang bersifat unik bagi protokol itu saja<sup>[5]</sup>.



Gambar 2.1 Model referensi TCP/IP

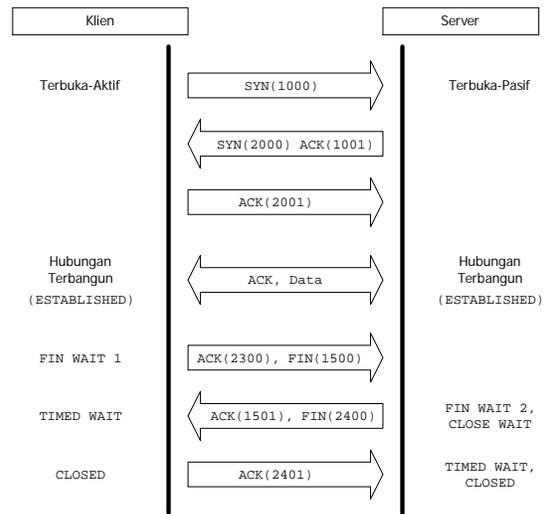
### B. PROTOKOL TCP

TCP bersifat andal (*reliable*) dan berorientasi koneksi (*connection oriented*)<sup>[5]</sup>. TCP memberikan layanan yang andal, karena ia melakukan proses pengiriman *acknowledgment*, yaitu bahwa TCP penerima segmen akan memberitahukan kepada pengirim bahwa segmen telah diterimanya. Bila sampai batas waktu yang telah ditetapkan *acknowledgment* penerima belum sampai kepada pengirim, maka pengirim melakukan pengiriman ulang segmen ke penerima. Sifat layanan TCP disebut berorientasi koneksi karena TCP pengirim akan memberitahukan terlebih dahulu bahwa ia akan membangun komunikasi kepada penerima. Bila penerima setuju, komunikasi baru dapat berjalan. Selain itu, layanan TCP juga membuat pihak penerima data untuk melakukan pengurutan segmen yang diterimanya, jika segmen datang tidak berurutan. Proses ini disebut proses pengurutan data (*data sequencing*)<sup>[5]</sup>.

Layanan TCP dibangun dengan membuat hubungan antara pihak pengirim dan penerima data dan dianggap telah terjalin jika keduanya telah membuat suatu *end-point*, yang disebut *socket*<sup>[7]</sup>. *Socket* terdiri atas pasangan nomor alamat IP 32-bit dan *port* 16-bit<sup>[7]</sup>.

Sebuah contoh manajemen koneksi dalam hubungan dalam TCP ditunjukkan oleh Gambar 2.2. Yang bertindak untuk meminta pembentukan hubungan disebut sebagai klien, sedangkan yang memutuskan menerima atau menolak hubungan disebut sebagai

server<sup>[5]</sup>. Proses meminta pembentukan hubungan oleh klien disebut sebagai terbuka-aktif (*active-open*), sedangkan proses yang dilakukan oleh server disebut terbuka-pasif (*passive-open*)<sup>[5]</sup>.



Gambar 2.2 Contoh kasus manajemen hubungan TCP

Pertama-tama, klien mengirimkan paket SYN (*synchronize*) yang disertai nomor urut. Bila server menerima ajakan klien, maka server akan mengirimkan paket ACK (*acknowledgment*) yang diberi nomor urut setelah nomor urut SYN milik klien, yang disambung juga dengan mengirimkan SYN miliknya<sup>[5]</sup>. Bila klien mengkonfirmasi SYN dari server dengan mengirimkan paket ACK, dan server membalasnya dengan ACK miliknya, maka dikatakan hubungan telah terbentuk (*established connection*)<sup>[5]</sup>, selanjutnya klien dan server dapat bertukar data. Pemutusan hubungan baru dimulai bila klien maupun server telah melepaskan paket FIN<sup>[5]</sup>. Bila paket FIN belum dikonfirmasi oleh lawannya, maka keadaan ini disebut CLOSE WAIT<sup>[5]</sup>. Bila kedua paket FIN sudah dikonfirmasi, maka hubungan baru dikatakan terputus (keadaan CLOSED)<sup>[5]</sup>.

### C. PROTOKOL UDP

UDP tidak berorientasi koneksi (*connectionless oriented*) dan juga tidak andal. Hal ini disebabkan ia tidak mengenal mekanisme *acknowledgment* dan pengiriman ulang<sup>[7]</sup>.

Dalam melakukan pengiriman paket data, layanan UDP tidak mengenal nomor urut paket<sup>[7]</sup>. Hal ini membuat layanan UDP tidak melakukan proses pengurutan paket.

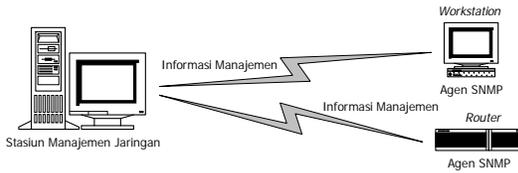
### D. SNMP

Pada dasarnya, SNMP merupakan sebuah protokol yang memungkinkan seseorang yang bekerja di balik sebuah komponen manajemen jaringan untuk mengelola jaringan komputernya dari jarak jauh (*remote*)<sup>[3]</sup>. Pengelolaan dapat dilakukan secara pasif ataupun aktif<sup>[3]</sup>.

Pengelolaan pasif adalah pengelolaan yang bersifat hanya-baca (*read-only*), sehingga tidak dapat mengubah kondisi komponen-komponen yang dikelolanya, misalnya pemantauan kondisi jaringan

komputer secara berkala. Stasiun Manajemen Jaringan dikatakan melakukan pengelolaan aktif, jika ia memiliki kekuasaan untuk dapat memerintahkan salah satu atau lebih komponen yang dikelolanya untuk mengubah kondisinya.

Secara garis besar, SNMP dibangun oleh dua buah komponen utama yang terlibat dalam mekanisme kerja SNMP, yaitu Stasiun Manajemen Jaringan (*Network Management Station*) dan Agen SNMP (*SNMP Agent*)<sup>[3]</sup>, seperti yang ditunjukkan oleh Gambar 2.3. Manajer melakukan pengelolaan tidak secara langsung, tetapi melalui Agen SNMP yang terdapat pada peralatan jaringan tersebut. Sesuai standar SNMP, Stasiun Manajemen Jaringan menghubungi Agen SNMP melalui layanan UDP pada *port* 161, kecuali untuk permintaan *Trap* yang memiliki *port* 162<sup>[5,7]</sup>.



Gambar 2.3 Komponen-komponen utama protokol SNMP

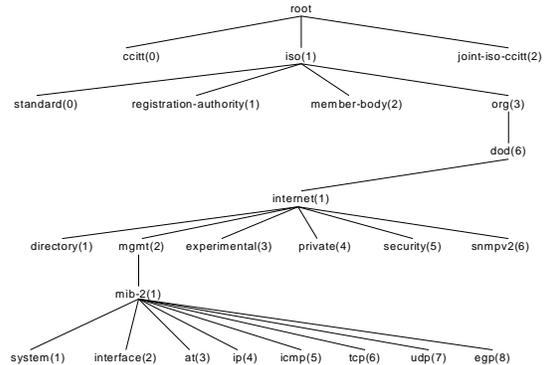
Permintaan dari manajer dan tanggapan dari agen dikirimkan melalui sebuah struktur pesan tertentu. Pada dasarnya, bagian pesan SNMP yang paling penting adalah Unit Data Protokol (*Protocol Data Unit*) yang berisi perintah atau tanggapan SNMP dan *Variable Binding*<sup>[3]</sup>. Perintah atau tanggapan SNMP merupakan representasi permintaan manajer atau tanggapan agen, misalnya *Get*, *Set* atau *Response*. Sedangkan *Variable Binding* merupakan representasi dari objek SNMP.

Agen SNMP menanggapi permintaan manajer dengan melakukan pemeriksaan terhadap suatu struktur basisdata yang berisi berbagai informasi manajemen sehubungan dengan peralatan yang dikelola tersebut. Struktur basisdata inilah yang dinamakan Basis Informasi Manajemen (*Management Information Base*)<sup>[3,5]</sup>. Struktur basisdata ini didefinisikan melalui standar ASN.1 (*Abstract Syntax Notation One*) yang juga menyumbangkan berbagai tipe data dasar bagi SNMP<sup>[3,5,7]</sup>. Perkembangan SNMP membuat bertambahnya tipe-tipe data SNMP yang sebenarnya merupakan pengembangan tipe-tipe data dasar (bawaan dari ASN.1). Berbagai tipe data hasil pengembangan tersebut dinamakan Struktur Informasi Manajemen (*SMI – Structure of Management Information*)<sup>[3,5,7]</sup>.

Informasi SNMP disebut sebagai objek, yang disajikan dalam bentuk pengidentifikasi objek (*object identifier*) dan nilai objek (*object value*)<sup>[3,5,7]</sup>. Pengidentifikasi objek SNMP memiliki tipe data khusus, yaitu *OBJECT IDENTIFIER* yang merupakan tipe data dasar SNMP<sup>[3,5,7]</sup>. Nilai objek memiliki tipe data seperti yang sudah ditetapkan dalam ASN.1 atau SMI<sup>[3,5,7]</sup>. Pengidentifikasi objek ditulis dalam urutan nomor berbentuk bilangan bulat positif bertitik atau urutan nama label bertitik, sesuai dengan diagram pohon pengidentifikasi objek yang telah ditetapkan menjadi standar oleh ISO<sup>[5,7]</sup>, seperti yang diperlihatkan oleh

Gambar 2.4. Objek-objek informasi SNMP memperoleh kedudukan di bawah titik *mib(1)*.

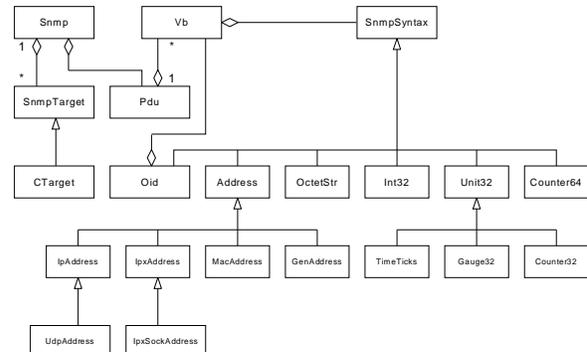
Kelompok (*group*) TCP diberi nomor 6 di bawah titik *mib(1)*, serta diberi label *tcp(6)*. Pengidentifikasi objek untuk mengakses TCP adalah *.iso.org.dod.internet.mgmt.mib-2.tcp* atau *1.3.6.1.2.1.6* (dalam bentuk angka bertitik)<sup>[5,7]</sup>. Sedangkan objek-objek UDP ditempatkan di bawah titik *udp(7)* dan pengidentifikasi objeknya adalah *1.3.6.1.2.1.7*.



Gambar 2.4 Bagian diagram pohon pengidentifikasi objek SNMP

## E. SNMP++

SNMP++ dibuat dalam bentuk kelas-kelas C++ yang terpisah-pisah, atau sudah dalam bentuk pustaka dinamis (*DLL – Dynamic Linking Library*)<sup>[3]</sup>. Hubungan kelas-kelas SNMP++ dijelaskan dalam bentuk diagram kelas seperti pada Gambar 2.5.



Gambar 2.5 Diagram kelas dari kelas-kelas dalam pustaka SNMP++

Kelas *SnmpSyntax* merupakan kelas dasar yang bertugas untuk menangani tipe-tipe data SNMP<sup>[3]</sup>. Kelas-kelas turunan dari kelas *SnmpSyntax* merupakan spesifikasi terhadap berbagai tipe data SNMP.

Kelas *Vb* merupakan kelas yang bertanggung jawab menangani *Variable Binding* dalam struktur pesan SNMP<sup>[3]</sup> dan merupakan bagian dari kelas *Pdu*. Kelas *Pdu* sendiri merupakan kelas yang merepresentasikan bagian Unit Data Protokol dalam struktur pesan SNMP<sup>[3]</sup>. Kelas *SnmpTarget* merupakan kelas dasar yang bertugas menangani Target SNMP<sup>[3]</sup>. Kelas ini memiliki kelas turunan yang diberi nama kelas *CTarget* yang menangani alamat Target SNMP.

Kelas *Snmp* membungkus (*encapsulate*) semua komponen utama dalam SNMP, termasuk perintah-perintah SNMP, penanganan versi SNMP dan pesan-

pesan kesalahan dasar dari SNMP<sup>[3]</sup>. Dalam melakukan operasinya, objek dari kelas ini bekerjasama dengan objek-objek dari kelas Pdu dan kelas CTarget. Secara konsep, objek-objek dari kelas Pdu dan kelas CTarget merupakan bagian dari objek kelas Snmp.

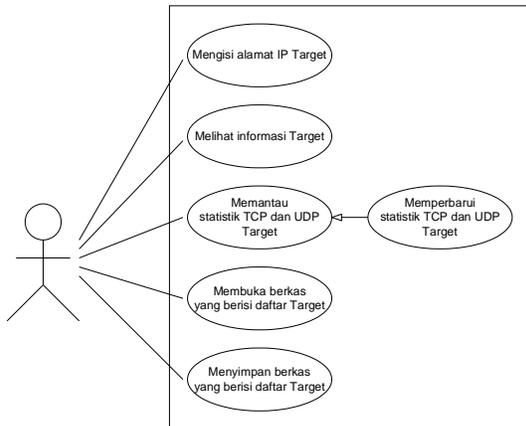
### III. PERANCANGAN APLIKASI

Perancangan aplikasi dilakukan dalam dua tahap. Tahap pertama memusatkan perhatian pada perancangan dasar dan bersifat umum, yang berarti tidak spesifik terhadap sistem operasi dan bahasa pemrograman yang dipakai dalam membuat aplikasi. Tujuan lain adalah untuk mendapatkan pandangan aplikasi dari sisi pengguna aplikasi, bukan pembuat aplikasi. Dari sini diharapkan berbagai kebutuhan fungsional aplikasi dapat diidentifikasi.

Tahap kedua merupakan tahap implementasi berbagai sarana dasar yang sudah diidentifikasi dalam tahap pertama. Kegiatan yang dilakukan adalah menyesuaikan hasil-hasil perancangan dasar dengan lingkungan sistem operasi dan bahasa pemrograman yang dipakai, dan yang lebih penting adalah menjawab “apa saja” yang terdapat pada tahap pertama dengan “bagaimana caranya”.

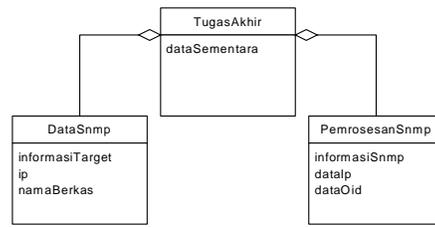
#### A. PERANCANGAN DASAR

Tahap ini memfokuskan pada dua kegiatan. Kegiatan pertama adalah mengidentifikasi kebutuhan fungsional aplikasi menggunakan diagram UML *use case* seperti ditunjukkan oleh Gambar 3.1.



Gambar 3.1 Proses mengidentifikasi kebutuhan fungsional menggunakan diagram *use case*

Kegiatan selanjutnya adalah mengidentifikasi kelas-kelas yang menangani berbagai kebutuhan fungsional yang sudah teridentifikasi. Kelas-kelas dasar yang diidentifikasi ditunjukkan oleh Gambar 3.2 yang memaparkan kelas-kelas dan hubungannya yang telah diidentifikasi pada tahap ini. Kelas-kelas yang telah teridentifikasi hanya dilengkapi oleh data (atribut-atribut), sedangkan rincian operasi-operasi yang terkait diidentifikasi pada tahap implementasi perancangan dasar.

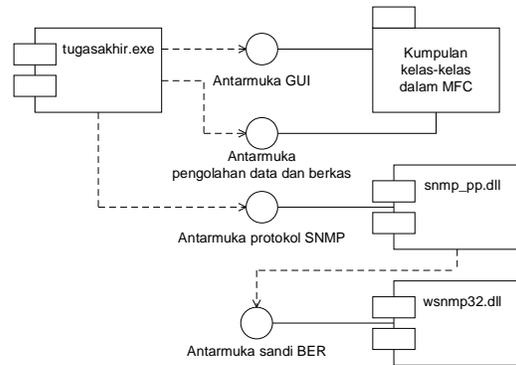


Gambar 3.2 Kelas-kelas dasar dan hubungannya

#### B. IMPLEMENTASI PERANCANGAN DASAR

##### 1. IDENTIFIKASI KEBUTUHAN PEMBUATAN APLIKASI

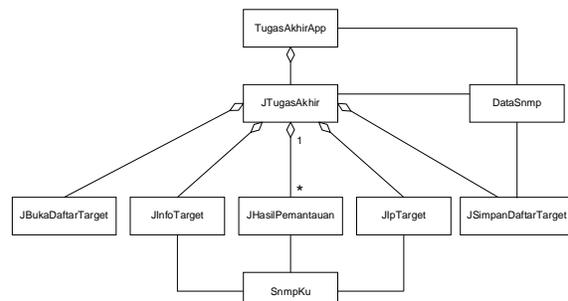
Kegiatan ini bertujuan untuk membuat spesifikasi komponen-komponen yang digunakan dalam proses pembuatan aplikasi yang telah disesuaikan dengan lingkungan sistem operasi dan bahasa pemrograman yang dipakai. Gambar 3.3 menunjukkan berbagai kebutuhan dan hubungannya yang diidentifikasi pada tahap ini.



Gambar 3.3 Hubungan aplikasi dengan komponen-komponen pendukungnya

##### 2. PENGEMBANGAN KELAS-KELAS DASAR

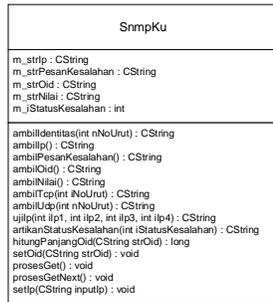
Tahap ini berisi kegiatan mengidentifikasi kembali kelas-kelas dasar yang sudah diidentifikasi pada perancangan dasar yang disesuaikan dengan lingkungan sistem operasi dan bahasa pemrograman yang dipakai. Kelas-kelas dasar kembali diobservasi dan kemudian dipecah-pecah dan ditambah dengan kelas-kelas baru. Gambar 3.4 menunjukkan berbagai kelas-kelas hasil pengembangan dan hubungannya.



Gambar 3.5 Kelas-kelas baru dan hubungan antarkelas yang lengkap

Kegiatan berikutnya adalah mendefinisikan tugas dari tiap-tiap kelas. Selanjutnya tiap-tiap kelas diobservasi untuk menemukan berbagai atribut (data) dan berbagai operasi yang mungkin diperlukan oleh kelas yang bersangkutan dalam menjalankan tugasnya.

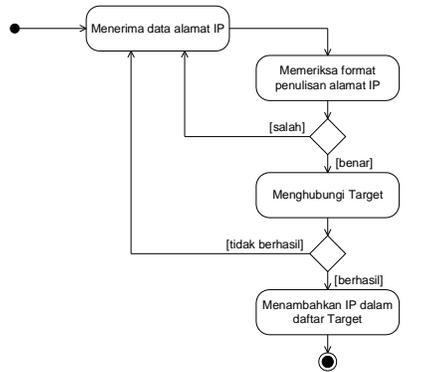
Gambar 3.6 menunjukkan sebuah contoh observasi berbagai atribut dan operasi dari kelas `SnmppKu` yang bertugas sebagai antarmuka kelas-kelas lain dengan pustaka `SNMP++`.



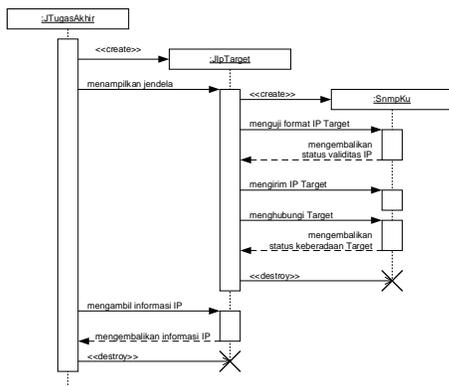
Gambar 3.6 Kelas `SnmppKu`

### 3. PERANCANGAN SISI DINAMIS APLIKASI

Kegiatan yang dilakukan di sini adalah merancang tingkah laku aplikasi berkaitan dengan sebuah kasus yang terjadi. Perancangan sisi dinamis ini dilakukan dengan bantuan dua buah diagram UML, yaitu diagram urutan (*sequence diagram*) dan diagram aktivitas (*activity diagram*). Gambar 3.7 merupakan contoh diagram aktivitas dan diagram urutan untuk kasus mengisi alamat IP Target.



(a)



(b)

Gambar 3.7 Kasus mengisi alamat IP Target

- (a) Diagram aktivitas
- (b) Diagram urutan

## IV. PENGUJIAN APLIKASI

Aplikasi diuji dengan menggunakan metode pengujian kotak-hitam (*black-box testing*) melalui dua buah pengujian. Pengujian dilakukan menggunakan dua buah komputer dengan spesifikasi seperti yang ditunjukkan oleh Tabel 3.1.

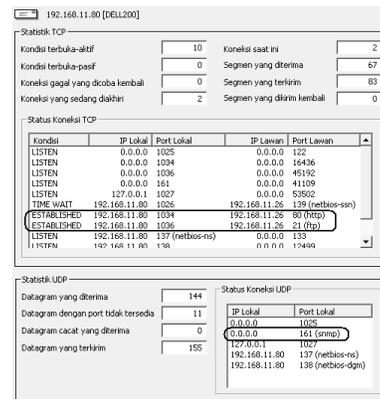
Tabel 3.1 Spesifikasi komputer pengujian

Spesifikasi	Komputer I	Komputer II
Prosesor	AMD Duron 1 GHz	Intel Pentium 200 MHz
Memori	DDR-RAM 128 MB	SDRAM 32 MB
Sistem Operasi	MS Windows 2000	MS Windows 98
Alamat IP	192.168.11.26	192.168.11.80

Pengujian pertama bertujuan untuk memastikan bahwa aplikasi dapat menjalankan fungsi utamanya. Pengujian kedua adalah pengujian aplikasi dengan membandingkan hasil pengukuran keluaran aplikasi, yaitu statistik TCP dan UDP, dengan keluaran perangkat lunak yang juga menginformasikan statistik TCP dan UDP. Perangkat lunak tersebut adalah `netstat.exe`.

### A. PENGUJIAN FUNGSIONAL APLIKASI

Pada pengujian ini, komputer II bertindak sebagai klien yang sedang mengakses layanan FTP dan HTTP yang disediakan oleh komputer I (server). Aplikasi yang diuji dijalankan pada komputer I dan mengakses Agen SNMP pada komputer II melalui *port* 161. Gambar 4.1 menunjukkan hasil pengujian yang dimaksud.



Gambar 4.1 Hasil pengujian fungsional

### B. PEMBANDINGAN KELUARAN APLIKASI DENGAN APLIKASI PEMBANDING

Proses perbandingan pada pengujian ini dilakukan dengan menjalankan aplikasi secara lokal dan jarak-jauh (*remote*).

Gambar 4.2 menunjukkan potongan tampilan aplikasi pada salah satu pengujian dengan data yang tidak berbentuk tabel.

192.168.11.80 [MORDEON3755]

Statistik TCP

Kondisi terbuka-aktif	27	Koneksi saat ini	0
Kondisi terbuka-pasif	0	Segmen yang diterima	281
Koneksi gagal yang dicoba kembali	0	Segmen yang terkirim	288
Koneksi yang terpasang kembali	9	Segmen yang dikirim kembali	0

Statistik UDP

Datagram yang diterima	1304
Datagram dengan port tidak tersedia	123
Datagram cacat yang diterima	0
Datagram yang terkirim	1411

(a)

```

TCP Statistics
Active Opens           = 27
Passive Opens         = 0
Failed Connection Attempts = 0
Reset Connections     = 9
Current Connections   = 0
Segments Received     = 281
Segments Sent         = 288
Segments Retransmitted = 0

UDP Statistics
Datagrams Received    = 1355
No Ports              = 123
Receive Errors        = 0
Datagrams Sent        = 1460
  
```

(b)

**Gambar 4.2** Hasil pengukuran statistik yang bukan berupa tabel  
 (a) Aplikasi yang diuji  
 (b) Aplikasi pembandingan

Gambar 4.3 menunjukkan potongan tampilan aplikasi pada salah satu pengujian dengan data yang berbentuk tabel. Pada kesempatan ini, Target tidak melakukan permintaan layanan apapun kepada server.

192.168.11.80 [MORDEON3755]

Status Koneksi TCP

Kondisi	IP Lokal	Port Lokal	IP Lawan	Port Lawan
LISTEN	0.0.0.0	1025	0.0.0.0	32907
LISTEN	0.0.0.0	161	0.0.0.0	53426
LISTEN	127.0.0.1	1030	0.0.0.0	24821
LISTEN	192.168.11.80	137 (netbios-ns)	0.0.0.0	45268
LISTEN	192.168.11.80	138	0.0.0.0	49158
LISTEN	192.168.11.80	139 (netbios-ssn)	0.0.0.0	32775

Status Koneksi UDP

IP Lokal	Port Lokal
0.0.0.0	1025
0.0.0.0	161 (snmp)
127.0.0.1	1030
192.168.11.80	137 (netbios-ns)
192.168.11.80	138 (netbios-dgm)

(a)

```

C:\>netstat -a
Active Connections
Proto Local Address Foreign Address State
TCP DeLL200:1025 DeLL200:0 LISTENING
TCP DeLL200:161 DeLL200:0 LISTENING
TCP DeLL200:1030 DeLL200:0 LISTENING
TCP DeLL200:137 DeLL200:0 LISTENING
TCP DeLL200:138 DeLL200:0 LISTENING
TCP DeLL200:nbssession DeLL200:0 LISTENING
UDP DeLL200:1025 *:*
UDP DeLL200:snmp *:*
UDP DeLL200:1030 *:*
UDP DeLL200:nbname *:*
UDP DeLL200:nbdatagram *:*
  
```

(b)

**Gambar 4.2** Pembandingan data berupa tabel  
 (a) Aplikasi yang diuji  
 (b) Aplikasi pembandingan

Dari hasil-hasil pengujian, terlihat adanya perbedaan terutama pada saat pengujian hasil-hasil keluaran aplikasi yang bukan berupa tabel. Adanya perbedaan dari hasil pembandingan dengan aplikasi

**netstat.exe** dapat disebabkan oleh hal-hal sebagai berikut.

1. Secara fisik, proses pemantauan tidak dapat dilakukan secara bersamaan.
2. Aplikasi yang diuji memantau Target melalui jaringan TCP/IP secara jarak-jauh (*remote*), sedangkan aplikasi **netstat.exe** hanya dapat berjalan pada komputer lokal. Hal ini juga mengurangi ketepatan waktu pemantauan yang seharusnya tiap aplikasi melakukan pemantauan pada saat yang sama.
3. Kesulitan memantau pada saat yang sama juga terjadi karena aplikasi yang diuji menggunakan protokol SNMP yang melakukan proses penyandian (*encode*) pada saat data akan dikirimkan, dan penerjemahan-sandi (*decode*) terhadap data yang baru diterima. Adanya penyandian dan penerjemahan-sandi menyebabkan waktu pemrosesan data menjadi bertambah lama.

## V. PENUTUP

### A. KESIMPULAN

Dari hasil penelitian ini, diperoleh beberapa kesimpulan sebagai berikut.

1. Dengan pemakaian pustaka SNMP++ yang berorientasi objek, pembuatan aplikasi menjadi lebih mudah dilakukan. Hal ini disebabkan keuntungan pemakaian kembali (*reusability*) dari konsep pemrograman berorientasi objek.
2. Untuk memudahkan mengakses komponen-komponen pustaka SNMP++, perlu dibuat sebuah kelas yang menjadi antarmuka dengan pustaka SNMP++.
3. Aplikasi telah berjalan dengan cukup baik dan mampu menjalankan pengujian fungsionalnya.
4. Pada pengujian melalui metode pembandingan dengan perangkat lunak **netstat.exe**, terdapat perbedaan. Adanya perbedaan kemungkinan disebabkan kerangka-kerja kedua aplikasi yang berlainan sehingga menyebabkan kedua aplikasi tidak dapat melakukan pemantauan pada waktu yang sama.

### B. SARAN

1. Aplikasi dapat dikembangkan untuk memantau statistik jaringan yang lain, seperti statistik IP dan statistik ICMP.
2. Aplikasi dapat dikembangkan agar dapat berjalan otomatis melalui pemasangan pewaktu, sehingga kegiatan pemantauan dapat berjalan otomatis, tidak manual.
3. Aplikasi dapat dikembangkan agar tidak hanya dapat melakukan aksi pasif saja, tetapi juga dapat melakukan aksi aktif dengan memanfaatkan perintah Set pada SNMP.

#### DAFTAR PUSTAKA

- [1] Booch, G., J. Rumbaugh, dan I. Jacobson, “*The Unified Modeling Language User Guide*”, Addison Wesley Longman, Inc., Massachusetts, 1999
- [2] Heywood, D., “Konsep dan Penerapan Microsoft TCP/IP”, Penerbit Andi Yogyakarta, Yogyakarta, 2001
- [3] Mellquist, P.E., “SNMP++, Pendekatan Berorientasi Objek”, Penerbit Andi Yogyakarta, Yogyakarta, 2002
- [4] Pressman, R. L., *Software Engineering: A Practitioner’s Approach (3<sup>rd</sup> Edition)*, McGraw-Hill, Inc., 1992
- [5] Purbo, O.W., A. Basalamah, I. Fahmi, dan A.H. Thamrin, “TCP/IP: Standar, Desain dan Implementasi”, Elex Media Komputindo, Jakarta, 2001
- [6] Tanenbaum, A.S., “Jaringan Komputer Jilid I”, Prenhallindo, Jakarta, 1997
- [7] Tanenbaum, A.S., “Jaringan Komputer Jilid II”, Prenhallindo, Jakarta, 1997
- [8] Taufan, R., “Manajemen Jaringan TCP/IP”, Elex Media Komputindo, Jakarta, 2001
- [9] Whitten, J.L., L.D. Bentley, K.C. Dittman, “*System Analysis and Design Methods*”, McGraw-Hill Irwin, 1997



**Arthop Sagita** (NIM. L2F097614)  
Mahasiswa Jurusan Teknik Elektro  
Konsentrasi Informatika  
Fakultas Teknik UNDIP Semarang

Mengetahui,  
Pembimbing II

R. Rizal Isnanto, S.T., M.M., M.T  
NIP. 132 288 515