

# Rancang Bangun Aplikasi *Monitoring* Jaringan dengan Menggunakan *Simple Network Management Protocol*

Reza Pradikta, Achmad Affandi, Eko Setijadi

Jurusan Teknik Elektro-FTI, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

*E-mail:* Pradikta09@mhs.ee.its.ac.id

**Abstrak**—Semakin meningkatnya ukuran dan jumlah perangkat jaringan akan semakin kompleks masalah pada jaringan sehingga diperlukan adanya pengawasan secara terus-menerus untuk menjamin ketersediaan atau *availability* layanan. *Simple Network Management Protocol* (SNMP) merupakan protokol aplikasi yang mampu menjalankan tugas untuk memonitoring kondisi jaringan. Pada tugas akhir ini akan dilakukan perancangan dan pembuatan Aplikasi *monitoring* jaringan dengan menggunakan protokol SNMP yang dilengkapi dengan sistem *database* untuk menyimpan dan mengolah nilai SNMP. Kemudian dilakukan pengujian untuk mengetahui tampilan dan fungsi dari Aplikasi yang telah dibuat. Pengujian juga dilakukan terhadap hasil aplikasi untuk mengetahui keakuratan. Hasil pengujian *availability device* dan *availability* sistem menunjukan bahwa aplikasi yang dibuat memiliki tingkat kesalahan 0 % jika dibandingkan dengan hasil perhitungan. Hasil pengujian trafik TCP menunjukan bahwa aplikasi yang dibuat cukup akurat jika dibandingkan dengan *software* Wireshark dan Netstat dengan nilai selisih terbesar untuk hasil monitoring adalah 0,2784%.

**Kata Kunci**—*Monitoring jaringan, Simple Network Management Protocol, Sistem database, Availability.*

## I. PENDAHULUAN

Perkembangan teknologi telekomunikasi yang sangat cepat akan menghadirkan beragam bentuk layanan bagi konsumen. Hal tersebut berbanding lurus dengan bisnis untuk membangun jaringan yang lebih besar dan lebih baik dengan harga yang lebih terjangkau. Dengan peningkatan ukuran dan jumlah perangkat jaringan maka akan semakin tinggi resiko terjadi gangguan jaringan. Manajemen jaringan terutama sistem *monitoring* menjadi sesuatu yang penting dilakukan.

*Simple Network Management Protocol* (SNMP) adalah sebuah protokol aplikasi pada jaringan TCP/IP yang dapat digunakan untuk pengelolaan dan pemantauan sistem jaringan komputer. Hampir semua peralatan jaringan telah mendukung penggunaan SNMP untuk pemantauannya. SNMP akan mempermudah proses *monitoring* dan manajemen jaringan karena dengan menggunakan SNMP akan dapat diketahui tentang kondisi perangkat jaringan yang diamati [1]. Tetapi layanan dan informasi SNMP hanya dapat diakses melalui tampilan pada *command prompt* atau *terminal* sehingga dalam penggunaannya tidak efektif dan sulit dilakukan karena masih membutuhkan pengolahan dan tampilannya sulit dimengerti.

Solusi yang pernah dilakukan adalah membuat *Graphical User Interface* (GUI) sebagai perantara untuk

mengambil dan menampilkan nilai SNMP [2], [3]. Tetapi solusi yang ditawarkan masih mempunyai kekurangan, karena hasil yang ditampilkan hanya sebatas informasi kondisi jaringan pada saat itu dan masih belum ada sistem untuk menyimpan dan mengolah nilai SNMP lebih lanjut. Padahal jika data tersebut diolah akan dihasilkan laporan tentang kondisi jaringan sehingga mempermudah *administrator* jaringan dalam pemantauan kondisi jaringan dan menganalisis kebutuhan serta pengembangan jaringan yang akan datang [4]. Solusi yang dapat dilakukan adalah menyediakan sistem *database* untuk menyimpan nilai-nilai kondisi jaringan yang didapat dari pesan SNMP dan mengolah lebih lanjut.

Pada tugas akhir ini akan dilakukan perancangan dan pembuatan aplikasi *monitoring* jaringan yang dapat digunakan sebagai perantara untuk mengambil dan mengolah nilai SNMP sekaligus terdapat sistem penyimpanan atau *database* sehingga dapat ditampilkan laporan informasi tentang kondisi jaringan yang meliputi *availability* perangkat dan trafik pada *transport* TCP.

## II. TINJAUAN PUSTAKA

### A. *Monitoring Jaringan*

Manajemen jaringan adalah kemampuan untuk memonitor, mengontrol, dan merencanakan suatu jaringan komputer dan komponen sistem. *Monitoring* jaringan merupakan bagian dari manajemen jaringan. Hal yang paling mendasar dalam konsep manajemen jaringan adalah tentang adanya manajer atau perangkat yang memanajemen dan agen atau perangkat yang dimanajemen.

Dalam implementasinya, ada berbagai macam arsitektur manajemen jaringan yang didasarkan pada tipe dan ukuran masing-masing. Ada dua arsitektur yang dapat digunakan yaitu manajemen terpusat (*centralized management*) dan manajemen menyebar (*distributed management*) [5].

### B. *Simple Network Management Protocol (SNMP)*

SNMP adalah sebuah protokol aplikasi pada jaringan TCP/IP yang menangani manajemen jaringan. Protokol ini didesain sehingga pengguna dapat dengan mudah memantau kondisi jaringan komputer [1]. Pemantauan kondisi jaringan dapat dilakukan dengan cara pengumpulan nilai-nilai informasi dari kondisi jaringan secara jarak jauh atau menggunakan satu pusat pengamatan. SNMP menjadi protokol yang terus dikembangkan karena banyak perangkat jaringan yang mendukung dan tersedia

layanan SNMP seperti *router*, *switch*, *server*, *workstation*, dan *printer*. Protokol SNMP pada jaringan TCP/IP menggunakan *transport* UDP oleh karena itu dalam penggunaannya tidak akan membebani trafik jaringan.

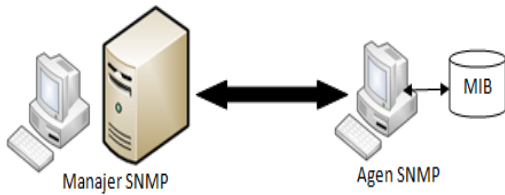
Pada sistem pemantauan jaringan dengan menggunakan layanan SNMP, terdapat tiga komponen dasar antara lain :

1. Manajer SNMP

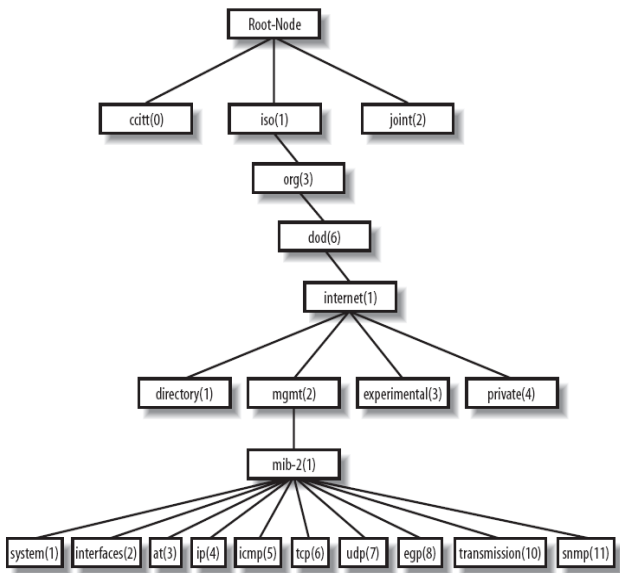
Manajer adalah perangkat yang menjalankan dan dapat menangani tugas-tugas manajemen jaringan.

2. Agen SNMP

Agen SNMP adalah perangkat pada jaringan yang akan diamati dan dikelola. Setiap agen akan merespon dan menjawab permintaan manajer SNMP.



Gambar 1. Manajer, Agen, dan MIB SNMP



Gambar 2. Diagram pohon MIB [1]

3. Management Information Base (MIB)

MIB pada SNMP dapat dikatakan sebagai tempat penyimpanan informasi yang dimiliki agen. MIB yang terdapat pada SNMP didefinisikan secara hirarki dan setiap bagian mempunyai identifikasi objek (OID).

C. Availability

*Availability system* atau ketersediaan sistem adalah keadaan dimana suatu sistem, subsistem, atau peralatan dalam keadaan beroperasi atau berfungsi. Ketersediaan sistem biasanya diukur sebagai faktor kehandalan atau *reability*. Ketersediaan dapat mengacu pada kemampuan dari sistem atau perangkat untuk memberikan layanan pada *user*. Periode ketika sistem atau perangkat dalam kondisi hidup disebut dengan *uptime* dan untuk kondisi sebaliknya disebut *downtime*.

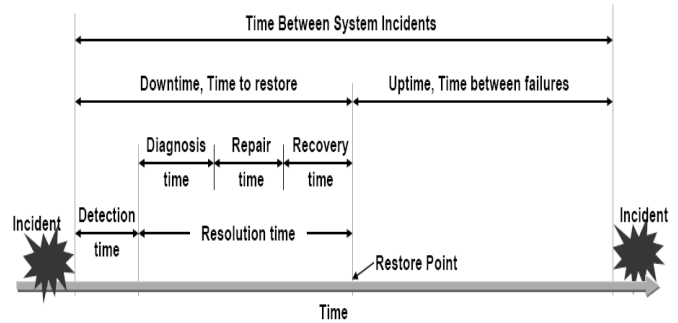
Nilai ketersediaan dapat dihitung dan secara spesifik dinyatakan dalam desimal, misalnya 0,9998 atau dalam persen, 99,998 %. Representasi sederhana dari

perhitungan dalam rasio matematika dapat dinyatakan sebagai berikut [4].

$$Availability = \frac{Uptime}{Uptime + Downtime} \tag{1}$$

Atau dapat juga dituliskan sebagai rasio,

$$Availability = \frac{MTBF}{MTBF + MTTR} \tag{2}$$



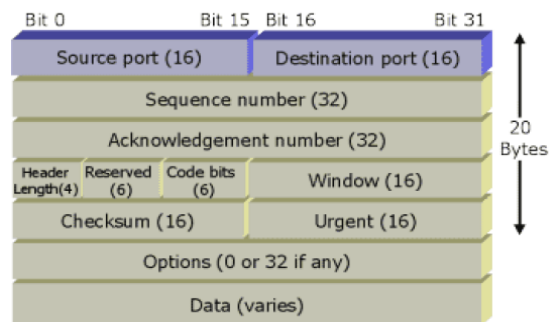
Gambar 3. Uptime dan Downtime [4]

*Downtime* menurut Gambar 3 dapat diartikan sebagai *mean time to restore* (MTTR) dan *uptime* disebut juga *mean time between failures* (MTBF), periode saat kondisi normal sampai terjadi gangguan berikutnya.

*Availability* dapat digunakan sebagai salah satu parameter dalam *Service Level Agreement* (SLA) [6]. SLA sering merujuk pada *downtime* atau *availability* untuk menentukan ketersediaan layanan.

D. Transport TCP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan dalam proses tukar data dari satu komputer ke komputer lain di dalam jaringan. *Transport Layer* melakukan segmentasi dan menyatukan kembali data yang tersegmentasi. Dua protokol lapisan yang paling umum *Transport TCP / IP* adalah *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP).



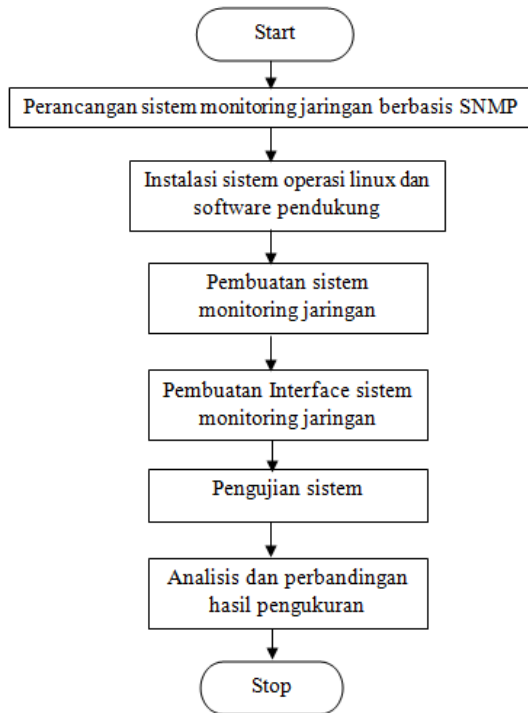
Gambar 4. Segmen TCP [7]

Perbedaan antara keduanya adalah fungsi khusus pada setiap protokol dalam mengimplementasikan. UDP adalah protokol sederhana dan bersifat *connectionless*. TCP merupakan protokol berorientasi koneksi (*connection oriented*), *reliable*, pencegahan duplikasi data, *congestion control*, dan *flow control* [7]. Beberapa contoh aplikasi yang menggunakan *transport* TCP adalah *Browser Web* (HTTP), *E-mail* (SMTP), dan *Transfer File* (FTP).

III. PERANCANGAN DAN IMPLEMENTASI SISTEM

A. Perancangan Aplikasi

Pada Tugas Akhir ini dibuat Aplikasi *monitoring* jaringan dengan memanfaatkan protokol SNMP. Berikut ini adalah *flowchart* perancangan dan pembuatan Aplikasi.



Gambar 5. Flowchart perancangan dan pembuatan

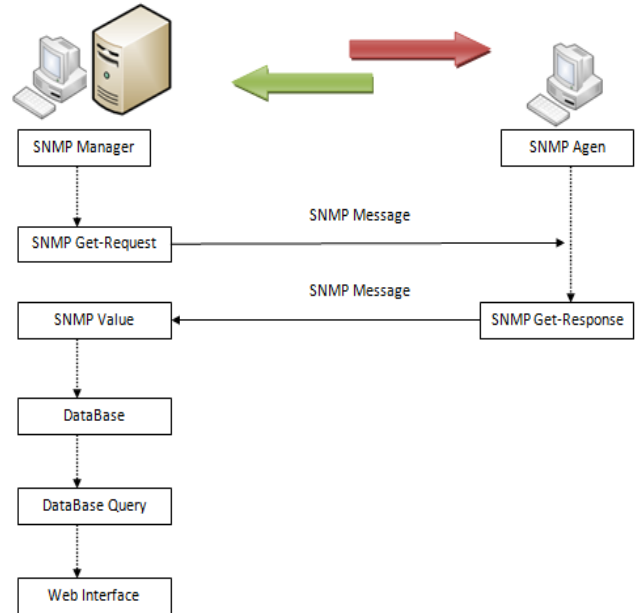
Pada tahap perancangan ditentukan MIB SNMP yang terdapat pada agen yang akan diambil dan diolah yaitu bagian TCP. Kemudian digunakan sistem operasi Linux distro Ubuntu 10.04 Long Term Support (LTS) dan instalasi *software* pendukung seperti PHP5 untuk bahasa pemrograman, MySQL sebagai sistem *database*, dan Apache2 untuk *web* server monitoring jaringan.

Pembuatan sistem monitoring meliputi pembuatan program pengambilan dan pengolahan nilai SNMP, pembuatan sistem *database*, pengambilan informasi pada *database*, dan pembuatan *web* sebagai *interface*.

B. Gambaran Umum Sistem Monitoring

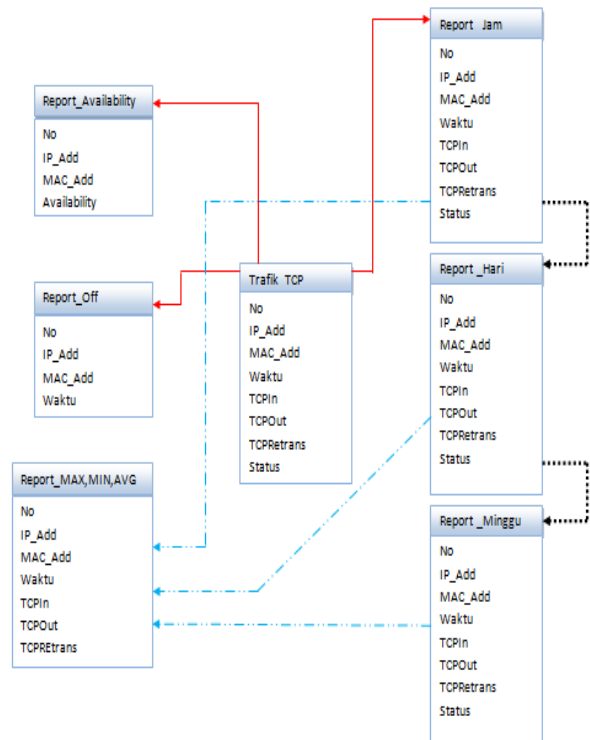
Sebuah manajer atau *monitoring* server akan memantau sebuah *device* atau agen. Manajer tersebut akan memanfaatkan protokol SNMP yaitu dengan mengirimkan pesan SNMP kepada agen secara priodik setiap menit. Kemudian agen tersebut membalas pesan SNMP dan mengirimkannya kepada manajer. Pesan yang diterima manajer berisi informasi yang dibutuhkan dan diinginkan oleh manajer. Kemudian pesan tersebut diolah dan disimpan kedalam *database* server.

Pengolahan pesan SNMP dilakukan untuk mengetahui jumlah data yang keluar, masuk, dan dikirim ulang pada *transport* TCP sekaligus mengindikasikan status perangkat yang damati dalam kondisi *on* atau *off* yang digunakan untuk mengetahui *availability*.



Gambar 6. Ilustrasi proses aplikasi *monitoring* jaringan

Dalam *database* akan dibuat *report* nilai *availability* dan trafik pada *transport* TCP dalam bentuk menit, jam, dan hari sesuai dengan struktur pada *database*. Selanjutnya akan dibuat *database query* untuk mengambil nilai pada *database* untuk ditampilkan pada *interface* berbasis *web*.



Gambar 7. Struktur dan relasi tabel pada *database*

IV. PENGUJIAN DAN ANALISIS

A. Pengujian Tampilan dan Fungsi Aplikasi

Pengujian Aplikasi *monitoring* jaringan dilakukan untuk menguji tingkat keberhasilan tampilan dan fungsi dari setiap halaman Aplikasi *monitoring* jaringan sesuai desain yang telah dirancang.

Tampilan halaman utama Aplikasi akan terdapat 4 tab menu yang dapat diakses yaitu :

1. *Home* : Menu ini digunakan untuk menuju bagian halaman utama Aplikasi.
2. *Monitoring* : Menu ini digunakan untuk melakukan proses *monitoring*. Setelah memilih menu ini, *user* akan menuju kehalaman *monitoring device* yang terdapat *form* untuk memasukkan IP *Address* dari perangkat yang ingin dimonitor.
3. *Availability* : Merupakan menu yang digunakan untuk melihat hasil *monitoring availability*. Pada halaman hasil *availaibility* juga terdapat fitur untuk melihat *availability* sistem dan bagian untuk menghapus hasil *monitoring availability*.
4. *TCP* : Tab menu ini digunakan untuk menuju halaman hasil *monitoring* trafik TCP. Hasil dari pengamatan trafik TCP akan ditampilkan dalam bentuk grafik per menit, per jam, dan per hari.



Gambar 8. Tampilan halaman utama Aplikasi

Hasil pengujian dari tampilan dan fungsi halaman aplikasi *monitoring* jaringan dapat dilihat dalam Tabel 1 berikut.

Tabel 1. Hasil pengujian fungsi Apikasi

Bagian Halaman	Pengujian	Hasil Pengujian	
		Berhasil	Gagal
Awal	Proses <i>login</i>	√	
<i>Home</i>	Akses halaman <i>home</i>	√	
	Akses proses <i>monitoring</i>	√	
	Akses hasil <i>availability</i>	√	
	Akses hasil trafik TCP	√	
Proses <i>Monitoring</i>	Melakukan proses <i>monitoring</i>	√	
Hasil <i>Monitoring Availability</i>	Melihat hasil <i>availability device</i>	√	
	Melihat informasi <i>availability</i>	√	
	Melihat hasil <i>availability sistem</i>	√	
	Hapus data <i>device</i>	√	
Hasil <i>Monitoring Trafik TCP</i>	Melihat hasil trafik TCP <i>device</i>	√	
	Melihat grafik trafik TCP	√	
	Hapus data <i>device</i>	√	

B. Pengujian Hasil Aplikasi Monitoring Jaringan

Pengujian hasil aplikasi monitoring jaringan dilakukan untuk mengetahui keakuratan hasil. Proses pengujian digunakan *software* Cacti untuk perbandingan

hasil *availability* dan *software* Wireshark dan *tools* Netstat untuk perbandingan hasil trafik TCP.

C. Pengujian Hasil Availability

Pada pengujian *availability* perangkat dilakukan berdasarkan variasi durasi *off* dari *device* yang diamati yaitu 0 menit, 2 menit, 5 menit, 10 menit, dan 15 menit dengan durasi pengamatan total selama 15 menit.

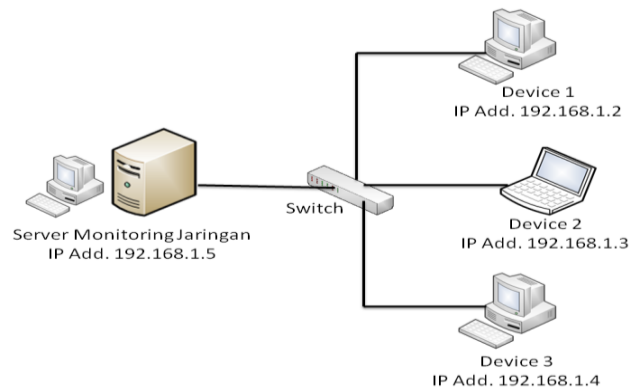
Tabel 2. Data hasil pengukuran *availability device*

Durasi Pengamatan (menit)	Durasi Off Rata-rata (menit)	Durasi On Rata-rata (menit)	Availability % Rata-rata (perhitungan)	Availability % Rata-rata (aplikasi)	Availability % Rata-rata (cacti)
15	0	15	100	100	100
15	2	13	86.66667	86.66667	100
15	5	10	66.66667	66.66667	66.667
15	10	5	33.33333	33.33333	33.333
15	15	0	0	0	0

Dari Tabel 2 ditunjukkan bahwa semakin lama durasi *on* suatu *device* maka nilai *availability* akan semakin besar. Nilai *availability* yang paling besar terjadi pada saat *device* selalu *on* atau tidak pernah mati pada saat durasi pengujian selama 15 menit. Nilai *availability* paling kecil didapatkan pada saat *device off* dengan durasi 15 menit.

Selain pengujian untuk satu *device*, juga dilakukan pengujian untuk *availability* sistem yaitu pengujian *availability* yang terdiri dari 3 *device*.

Pengujian dilakukan dilihat berdasarkan variasi durasi waktu *off* pada 3 *device* yang berbeda. Variasi pertama masing-masing *device* akan dimatikan selama 2, 4, dan 8 menit. Variasi kedua, masing-masing akan dimatikan selama 3, 6, dan 9 menit dan untuk variasi yang ketiga, masing-masing akan dimatikan selama 5, 10, dan 15 menit. Durasi pengamatan total selama 15 menit.



Gambar 9. Topologi jaringan *monitoring availability* sistem

Tabel 3. Data hasil pengukuran *avaialability* sistem

Variasi	Device	Durasi Pengamatan (menit)	Durasi Off (Rata-rata) (menit)	Durasi On (Rata-rata) (menit)	Rata-Rata Durasi On (Rata-rata) (menit)	Availability sistem % (Rata-rata) (perhitungan)	Availability sistem % (Rata-rata) (aplikasi)	Availability sistem % (Rata-rata) (cacti)
1	1	15	2	13	10.33	68.89	68.89	88.89
	2	15	4	11				
	3	15	8	7				
2	1	15	3	12	9	60	60	77.78
	2	15	6	9				
	3	15	9	6				
3	1	15	0	15	10	66.67	66.67	66.67
	2	15	5	10				
	3	15	10	5				

Hasil pada Tabel 3 dapat diketahui semakin lama rata-rata durasi *on* maka nilai *availability* akan semakin besar. begitu juga sebaliknya. Nilai *availability* paling besar terjadi saat variasi pengujian pertama, didapatkan rata-rata durasi *on* selama 10,33333 menit.

**D. Perbandingan Hasil Availability**

Hasil pengujian pada Tabel 2 dan Tabel 3 dapat dilihat perbandingan hasil antara perhitungan, *software* Cacti, dan Aplikasi yang telah dibuat. Dari perbandingan tersebut, dapat disimpulkan bahwa Aplikasi bekerja dengan baik untuk mengetahui *availability* dan hasilnya akurat 100 % jika dibandingkan dengan hasil perhitungan.

**Tabel 4.** Detail hasil pengukuran *availability* sistem

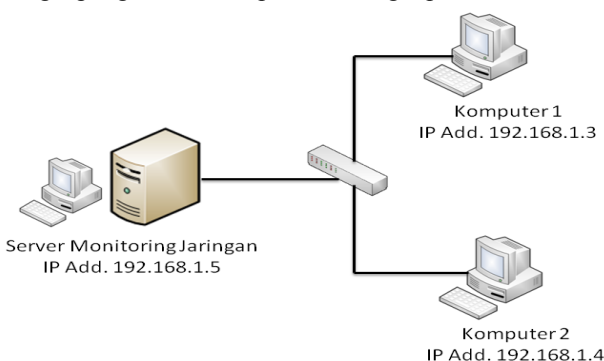
Variasi	Device	Durasi Pengamatan (menit)	Off Rata-rata (menit)	On Rata-rata (menit)	Availability % Rata-rata (Perhitungan)	Availability % Rata-rata (Aplikasi)	Availability % Rata-rata (Cacti)
1	1	15	2	13	86.67	86.67	100
	2	15	4	11	73.33	73.33	100
	3	15	8	7	46.67	46.67	66.67
2	1	15	3	12	80	80	100
	2	15	6	9	60	60	66.67
	3	15	9	6	40	40	66.67
3	1	15	0	15	100	100	100
	2	15	5	10	66.67	66.67	66.67
	3	15	10	5	33.33	33.33	33.33

Pada Tabel 2 dan Tabel 4 dapat dilihat perbedaan antara hasil *monitoring availability* dengan menggunakan Cacti. Perbedaan disebabkan karena *software* Cacti melakukan proses pengamatan atau *polling* secara periodik setiap 5 menit. Sehingga jika terjadi *off* kurang dari 5 menit tidak akan dapat diketahui dan dianggap *device* tersebut masih dalam kondisi *on*. Oleh karena itu, hasil *availability* dari Aplikasi yang dibuat lebih akurat jika dibandingkan dengan *software* Cacti.

Aplikasi yang dibuat juga terdapat fitur untuk mengetahui *availability* sistem atau hasil *availability* rata-rata dari beberapa perangkat yang diamati. Sedangkan jika menggunakan perhitungan dan *software* Cacti perlu dilakukan perhitungan secara manual untuk mencari nilai *availability* tiap perangkat kemudian dilakukan perhitungan untuk mencari rata-rata seperti pada Tabel 4.

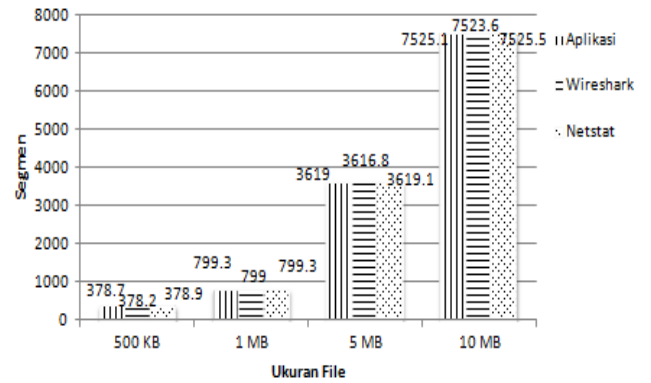
**E. Pengujian Hasil Trafik TCP**

Pada pengujian hasil trafik TCP, hasil yang didapatkan dari aplikasi akan dibandingkan dengan *software* Wireshark dan Netstat. Pengujian dilakukan berdasarkan variasi ukuran *file*, yaitu 500 KB, 1 MB, 5 MB, dan 10 MB yang akan ditransfer dari komputer 1 sebagai pengirim ke komputer 2 sebagai penerima.

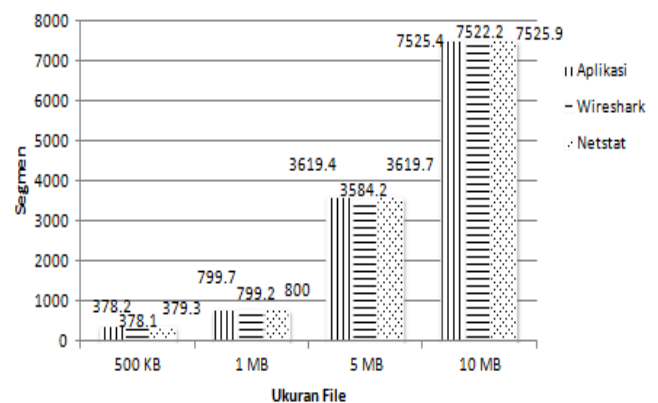


**Gambar 10.** Topologi jaringan *monitoring* Trafik TCP

Dari Gambar 11 ditunjukkan bahwa semakin besar ukuran *file* yang dikirim, maka jumlah segmen dikirim atau yang keluar semakin besar.



**Gambar 11.** Pengukuran trafik TCP yang keluar pada komputer 1



**Gambar 12.** Pengukuran trafik TCP yang masuk pada komputer 2

Pada komputer 2 atau penerima juga dilakukan pengukuran untuk melihat jumlah segmen yang masuk. Gambar 12 menunjukkan semakin besar *file* yang dikirim maka jumlah segmen diterima atau segmen yang masuk semakin besar. Jumlah segmen masuk paling besar terjadi pada saat pengiriman *file* dengan ukuran 10 MB yaitu sebesar 7520 segmen.

Dari hasil data pengujian trafik TCP dapat disimpulkan bahwa jumlah segmen yang keluar dan masuk pada *transport* TCP dipengaruhi oleh besarnya *file*. Jumlah maksimal data yang dapat dikirim dalam satu segmen TCP adalah sebesar 1460 Bytes [8], [9]. Oleh karena itu, saat melakukan transfer *file*, data tersebut akan dipisah kedalam segmen-segmen.

**F. Perbandingan Hasil Pengukuran Trafik TCP**

Perbedaan dan selisih hasil pengukuran trafik TCP yang keluar dan masuk dapat dilihat pada tabel-tabel berikut.

**Tabel 5.** Selisih hasil Aplikasi dan Wireshark untuk segmen TCP yang keluar pada komputer 1

Ukuran File	Selisih Aplikasi & Wireshark (segmen)	Selisih Aplikasi & Wireshark (%)	Selisih rata-rata Aplikasi & Wireshark (%)
500 KB	0.5	0.132205182	0.062629158
1 MB	0.3	0.037546934	
5 MB	2.2	0.060827251	
10 MB	1.5	0.019937264	

**Tabel 6.** Selisih hasil Aplikasi dan Netstat untuk segmen TCP yang keluar pada komputer 1

Ukuran File	Selisih Aplikasi & Netstat (segmen)	Selisih Aplikasi & Netstat (%)	Selisih rata-rata Aplikasi & Netstat (%)
500 KB	0.2	0.052784376	0.015215689
1 MB	0	0	
5 MB	0.1	0.002763118	
10 MB	0.4	0.005315261	

**Tabel 7.** Selisih hasil Aplikasi dan Wireshark untuk segmen TCP yang masuk pada komputer 2

Ukuran File	Selisih Aplikasi & Wireshark (segmen)	Selisih Aplikasi & Wireshark (%)	Selisih rata-rata Aplikasi & Wireshark (%)
500 KB	0.1	0.02644803	0.278409848
1 MB	0.5	0.062562563	
5 MB	35.2	0.982088053	
10 MB	3.2	0.042540746	

**Tabel 8.** Selisih hasil Aplikasi dan Netstat untuk segmen TCP yang masuk pada komputer 2

Ukuran File	Selisih Aplikasi & Netstat (segmen)	Selisih Aplikasi & Netstat (%)	Selisih rata-rata Aplikasi & Netstat (%)
500 KB	1.1	0.290007909	0.085609903
1 MB	0.3	0.0375	
5 MB	0.3	0.00828798	
10 MB	0.5	0.006643724	

Perbedaan atau selisih antara Aplikasi dengan Netstat hasilnya selalu lebih kecil jika dibandingkan selisih antara hasil Aplikasi dengan Wireshark. Hal ini terjadi karena prinsip kerja dari Aplikasi yang dibuat hampir sama dengan Netstat yaitu mengambil nilai pada saat tertentu secara periodik. Sedangkan Wireshark secara terus menerus mengambil nilai untuk diamati pada saat proses berlangsung.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan hasil rancang bangun Aplikasi *monitoring* jaringan menggunakan protokol SNMP, juga dari data-data yang didapatkan dari hasil pengujian Aplikasi, dapat diambil kesimpulan sebagai berikut :

1. Penerapan sistem *database* pada aplikasi yang dibuat berhasil dilakukan. Hal ini dapat dilihat pada saat pengujian fungsi-fungsi aplikasi *monitoring* jaringan dan tampilan hasil *monitoring* terdapat *report* atau laporan mengenai *availability* dan trafik tcp.
2. Nilai *availability* dipengaruhi oleh *Uptime* dan *Downtime* suatu perangkat. Semakin besar nilai *Uptime* maka nilai *availability* akan semakin besar. Hasil pengujian *availability* menunjukkan bahwa aplikasi yang dibuat memiliki tingkat kesalahan 0 % jika dibandingkan dengan hasil perhitungan.
3. Pada saat pengujian untuk mengetahui jumlah segmen yang keluar, selisih jika dibandingkan dengan *software* Wireshark dan Netstat adalah sebesar 0,0626 % dan 0.0152 %. Sedangkan pada saat pengujian jumlah segmen yang masuk didapatkan selisih 0,2784 % dan 0,0856 % jika dibandingkan dengan *software* Wireshark dan Netstat.

### B. Saran

Beberapa saran dan rekomendasi yang berguna untuk pengembangan aplikasi *monitoring* jaringan selanjutnya sebagai berikut :

1. Perlu dilakukan eksplorasi mengenai nilai-nilai dan parameter jaringan yang dapat diambil pada layanan SNMP sehingga dapat menambah fungsi dan fitur dari Aplikasi *monitoring* jaringan.
2. Aplikasi *monitoring* jaringan yang telah dibuat dapat diintegrasikan dengan sistem peringatan dini untuk mempermudah kerja *administrator* jika terjadi gangguan jaringan.

## DAFTAR PUSTAKA

- [1] G Mauro, Douglas., "Essential SNMP Second Edition", O'Reilly Media, Sebastopol, 2005.
- [2] Limpraptono, Yudi., "Pengembangan Aplikasi Protocol SNMP Untuk Manajemen Peralatan Jaringan Intranet", Jurnal Elektro ELTEK Vol. 1, No. 1, 2010.
- [3] Utami, Sri Puji., "Perancangan Online Network Monitoring Berbasis PHP dan SNMP", Seminar Nasional Teknologi Informasi (SNATI), Yogyakarta.
- [4] ITIL, "How to Develop, Implement and Enforce ITIL v3 Best Practice", The Art of Service, Brisbane, 2008.
- [5] Hui Lee, Kwang., "A Distributed Network Management System", Proceeding IEEE, Korea Research Foundation, Agustus, 1994.
- [6] ITU- T Standardization E.860, "Framework of a Service Level Agreement", Juni, 2002.
- [7] Purnomo, Agus., "Jaringan Komunikasi Dasar", UNS, Surakarta, 2010.
- [8] IBM, "TCP workload tuning", AIX documentation, <http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp>, di akses tanggal 12 Desember 2012.
- [9] Borman, D., "TCP Options and Maximum Segment Size (MSS)", RFC 6691, IETF, Juli 2012.



### REZA\_PRADIKTA

Dilahirkan di Mojokerto pada tanggal 23 Oktober, merupakan putra kedua dari dua bersaudara pasangan Bapak Slamet Budi dan Ibu Yuni Akhiriyah. Penulis memulai pendidikan formalnya di SDN Miji IV Mojokerto, setelah itu meneruskan pendidikan di SMPN 1 kota Mojokerto, kemudian melanjutkan pendidikan di SMAN 1 Sooko Mojokerto. Setelah lulus dari SMA, penulis melanjutkan studi di Jurusan teknik Elektro Institut Teknologi Sepuluh Nopember

(ITS) Surabaya dengan mengambil Bidang Studi Telekomunikasi Multimedia.

Selama masa studinya di Teknik Elektro ITS, penulis pernah aktif di Himpunan Mahasiswa Teknik Elektro di Departemen Pengabdian Masyarakat, UKM Tenis Lapangan ITS, dan sebagai asisten di Laboratorium Jaringan Telekomunikasi dan Multimedia.