

PERANCANGAN APLIKASI *CAPTURE TRAFFIC NETWORK TCP/IP* MENGGUNAKAN **NODE.JS** BERBASIS WEB

Mody Unedo Ritonga¹, Naikson F Saragih², Asaziduhu Gea³

Teknik Informatika¹, Fakultas Ilmu Komputer, Universitas Methodist Indonesia
Sistem Informasi³, Fakultas Ilmu Komputer, Universitas Methodist Indonesia

modyunedo98@gmail.com¹, saragihnaikson@gmail.com², gea.asaziduhu@gmail.com³

ABSTRACT

With more and more people accessing the internet to get information and the absence of settings on data traffic it will cause the network to be slow. The network traffic settings in Linux Ubuntu so far still use scripts which are relatively more difficult to use and understand for some people, so tools are needed to configure them. Capture network traffic application is Software designed to capture communications connected to servers. Capture network traffic application can only log data and read TCP/IP and HTTP protocols. The Capture network traffic application has a variety of features, including checking whether the server is on or off, and capturing traffic packets on a LAN network and displaying a map of the URL location. So the results of this study explain the creation of a Traffic Management Configurator to facilitate traffic monitoring settings in gateways that connect different networks. Furthermore, this traffic monitoring application is used to view the actual packets of traffic on the network and network statistics. Both applications use the web as an interface to the user.

Keywords: *Traffic Monitoring, Linux Ubuntu, Packet Logger, TCP/IP, Configuration, Web.*

I. PENDAHULUAN

Perkembangan teknologi komunikasi antar komputer memegang peranan yang sangat penting dalam mempercepat proses aliran informasi. Berbagai bentuk jaringan komputer dikembangkan untuk memenuhi kebutuhan akan informasi. Pembangunan infrastruktur jaringan yang terhubung ke *intranet* maupun internet sangat diperlukan. Salah satu usaha yang dapat dilakukan adalah dengan pembangunan *server intranet* di instansi yang memerlukan, sehingga kebutuhan akan informasi yang cepat dapat dipenuhi, karena hal tersebut dirasakan sangat penting dan mendesak. Saat ini telah banyak protokol komunikasi komputer yang telah dikembangkan untuk membentuk jaringan komputer.

Seorang administrator jaringan melakukan pemeriksaan terhadap paket data yang melintas pada jaringan tersebut, namun proses pengamatan tersebut akan menghasilkan suatu log data dalam jumlah yang sangat besar dan akan memakan waktu dalam melakukan pengumpulan paket data.

Masalah yang terjadi jika seorang administrator jaringan sedang melakukan proses pengumpulan paket data dengan cara manual adalah memperlama waktu yang ada serta seorang administrator jaringan akan mengalami keterlambatan saat menangani masalah yang ada. Contohnya pada saat ini adalah ketika ada 4 perangkat yang terhubung ke 1 jaringan LAN atau router maka yang tercatat pada saat itu hanya ada 4 perangkat saja, dan ketika ada 2 perangkat baru yang terhubung ke 1 jaringan LAN atau router maka hanya 4 perangkat saja yang tercatat, hal yang biasa dilakukan seorang administrator adalah dengan cara refresh ulang agar perangkat baru bisa

muncul, serta paket data yang tertangkap tidak lengkap, biasanya tampilan paket data yang tertangkap hanya *MAC Address, IP Address, MCS, RSSI, TX Rate, Rx Rate dan STA Mode*. Paket data yang tertangkap cuma beberapa tampilan saja, maka akan kurang maksimal dalam melakukan proses analisa. Dalam hal ini sering sekali *client* dari pengguna mengalami keterlambatan dalam mengakses situs web dan banyaknya keluhan atau laporan yang masuk sehingga jika hal ini berlangsung lama akan menghilangkan kepercayaan *client*.

Untuk itu dibutuhkan sebuah aplikasi yang dapat menangkap data dalam jaringan seperti aplikasi *capture traffic network*. Aplikasi *Capture traffic network* dapat mencatat data dan membaca protokol TCP/IP dan HTTP. Aplikasi *Capture traffic network* memiliki beragam fitur, antara lain memeriksa apakah server menyala atau mati, *meng-capture traffic packet* pada sebuah jaringan LAN, menampilkan bandwidth yang terhubung ke server dan menampilkan peta lokasi URL tersebut. Berdasarkan latar belakang masalah di atas, penulis tertarik untuk membangun aplikasi yaitu "**Perancangan Aplikasi *Capture Traffic Network TCP/IP* Menggunakan Node.Js Berbasis Web**".

II. LANDASAN TEORI

A. *Capture Traffic Network*

Aplikasi *Capture traffic network* adalah Perangkat lunak yang dirancang untuk menangkap komunikasi yang terhubung ke server. Aplikasi *Capture traffic network* hanya dapat mencatat data dan membaca protokol TCP/IP dan HTTP. Aplikasi *Capture traffic network* memiliki beragam fitur, antara lain memeriksa apakah server menyala atau mati, *meng-capture traffic packet*

pada sebuah jaringan LAN, menampilkan bandwidth yang terhubung ke server dan menampilkan peta lokasi URL tersebut.

B. Tujuan dan Manfaat Penelitian

Adapun tujuan dan manfaat dengan dibuatnya penelitian ini adalah untuk merancang aplikasi *Capture traffic network* pada protocol *TCP/IP*. Tujuan dan manfaat dari penelitian ini sebagai berikut yakni :

1. Membahas penggunaan fungsi-fungsi java script untuk mengakses Node.Js dan JSON
2. *Capture traffic packet* pada sebuah jaringan LAN.
3. Dapat melihat lalu lintas paket data yang berjalan di dalam jaringan.
4. Membantu administrator jaringan untuk menangkap data *traffic* dalam satu jaringan.
5. Dapat mengetahui IP client yang terhubung ke *server*.

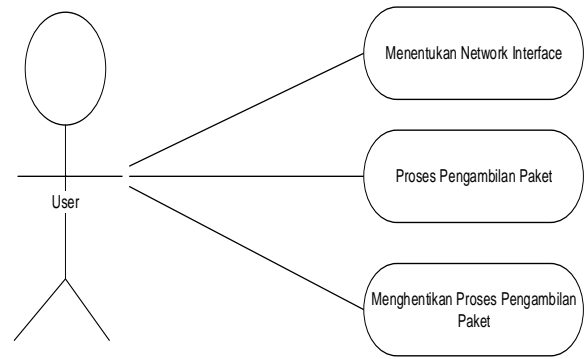
III. ANALISIS DAN PERANCANGAN SISTEM

A. Analisis Masalah

Seorang administrator jaringan melakukan pemeriksaan terhadap paket data yang melintas pada jaringan tersebut, namun proses pengamatan tersebut akan menghasilkan suatu log data dalam jumlah yang sangat besar dan akan memakan waktu dalam melakukan pengumpulan paket data. Masalah yang terjadi jika seorang administrator jaringan sedang melakukan proses pengumpulan paket data dengan cara manual adalah memperlama waktu yang ada serta seorang administrator jaringan akan mengalami keterlambatan saat menangani masalah yang ada.

B. Use case Diagram

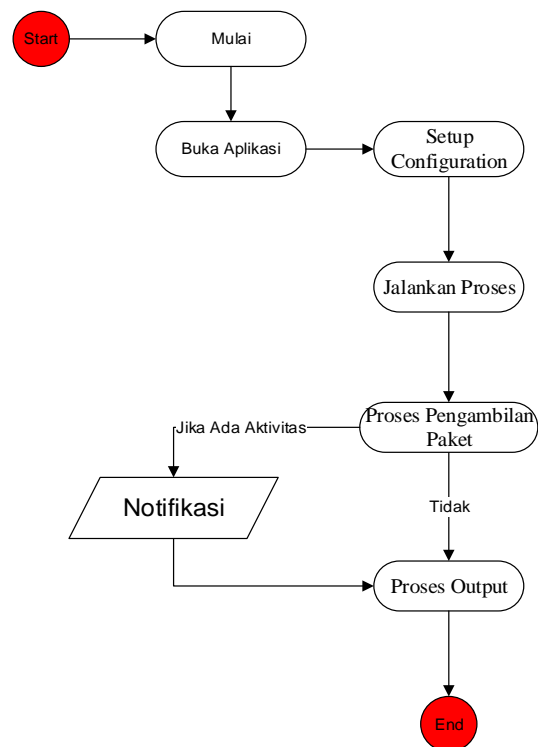
Aplikasi *Capture traffic network* pada Protokol *TCP/IP* ini ditujukan kepada *user* yang ingin melihat aktivitas client yang ditangkap oleh network interface pada suatu komputer host yang berinteraksi melalui gateway. Tidak ada tingkatan aktor/user di dalam menggunakan aplikasi ini. *User* terlebih dahulu memerintahkan aplikasi untuk berhenti menangkap paket. Ketika aplikasi mulai menangkap aktivitas client, *user* dapat melihat segala aktivitas komputer client tersebut. Aplikasi ini bertujuan untuk menangkap aktivitas client yang ditangkap oleh network interface pada suatu komputer host yang berinteraksi melalui gateway. Aplikasi ini memisahkan jenis protocol pada application layer dan jenis protocol pada transport layer. Pada network layer, ditampilkan alamat IP pengirim dan alamat IP tujuan. *Use case* dari keseluruhan sistem aplikasi ini dapat dilihat pada gambar 1 :



Gambar 1. Use Case Aplikasi *Capture traffic network*

C. Activity Diagram.

Activity diagrams adalah sesuatu yang menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya atau internal processing. *Activity diagram* memperlihatkan keadaan dari awal (start) hingga akhir (end). *Activity diagram* dari aplikasi ini dapat dilihat pada gambar 2 :



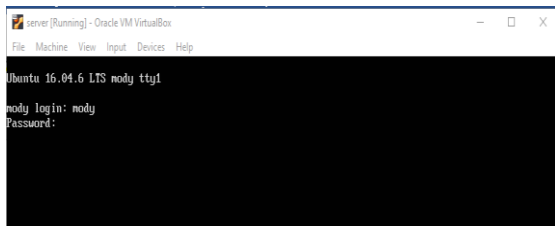
Gambar 2. Activity Diagram Aplikasi *Capture traffic network*

IV. IMPLEMENTASI DAN PENGUJIAN SISTEM

A. Tampilan Program

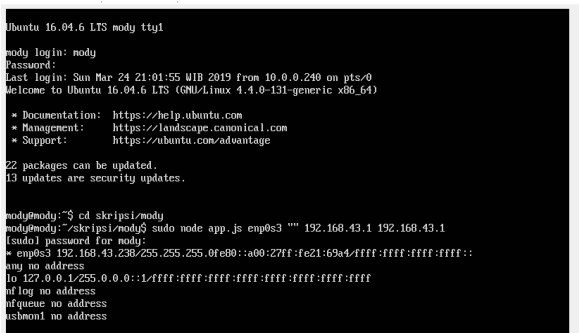
1. Perancangan Login Admin

Pada tahap ini penulis menampilkan perintah untuk membuka file terminal, yang mana merupakan suatu tempat untuk mengaktifkan terminal sebagai Command *Prompt* di dalam ubuntu. Ada 2 tahap yang penting di dalam implementasi konfigurasi program. Tahap menjalankan aplikasi menggunakan server node.js dan tahap perintah konfigurasi ip address pada sistem operasi ubuntu. selanjutnya perintah untuk membuka dan mengkonfigurasi aplikasi detector dengan perintah sudo node app.js. Berikut tampilan untuk menjalankan aplikasi dengan server node js dapat dilihat pada gambar 3 :



Gambar 3. Login Admin

Berikut tampilan *server* Node.js telah aktif dapat dilihat pada gambar 4 :

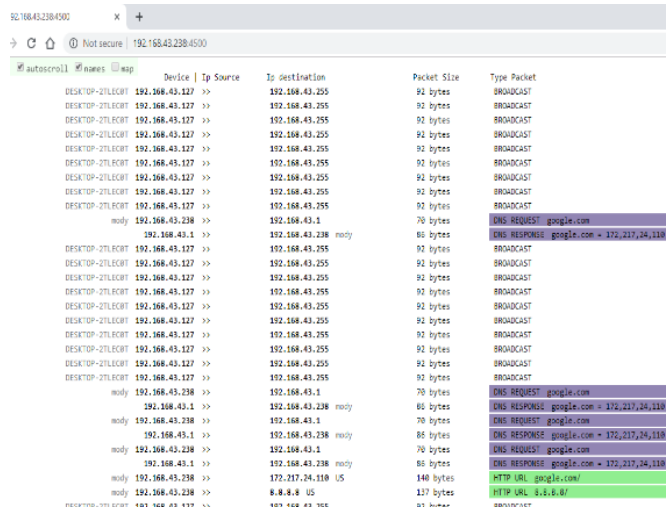


Gambar 4. Server Node.js telah aktif

2. Tampilan Menu Utama

Berikut ini adalah tampilan implemetasi dari perancangan keamanan jaringan pada Aplikasi *Capture traffic network*. Berikut ini adalah implementasi setelah melakukan konfigurasi pada sever ubuntu.

Untuk mengakses aplikasi dalam bentuk web aplikasi, dapat akses dengan web browser client dengan menginput ip address server dengan port untuk aplikasi tersebut. <http://192.168.43.238:4500/>. Tampilan pertama setelah aplikasi terbuka adalah kosong, dikarenakan belum menangkap komunikasi pada jaringan. Untuk menangkap salah satu komunikasi yang ada pada network tersebut dapat dilakukan dengan perintah sederhana, contoh: curl google.com, ssh remote, ftp dan lain-lain. Tampilan untuk menangkap salah satu komunikasi dapat dilihat pada gambar 5:

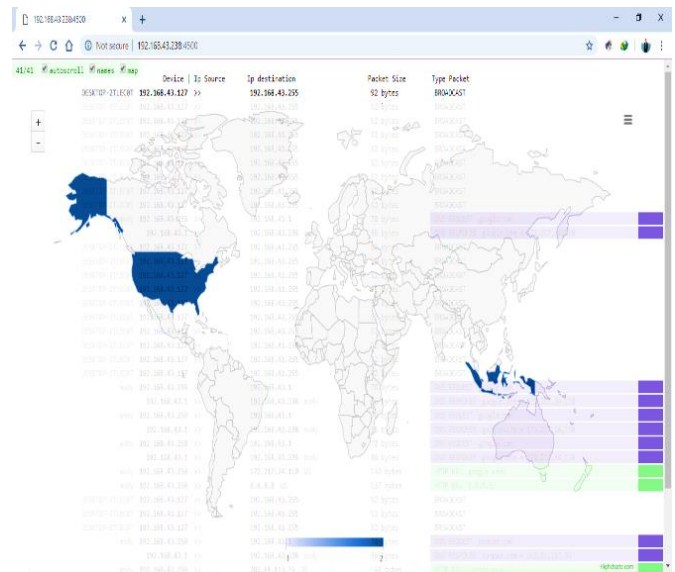


Gambar 5. Tampilan Menangkap Salah Satu Komunikasi

Proses untuk menampilkan halaman awal aplikasi capture traffic network dapat ditemukan pada *function newRow(table, dat) {}* di client/index.html.

3. Tampilan Map Ip Address

Pada halaman ini akan menampilkan status map atau letak posisi pada Ip Address yang memakai komunikasi. Berikut hasil menampilkan geo location ip address yang diakses melalui HTTP dan HTTPS dapat dilihat pada gambar 6 :



Gambar 6. Tampilan Map

Proses untuk menampilkan map pada web browser dapat ditemukan pada

```
var load_map, toggle_map;
$(function () {} Di client/index.html.
```

V. HASIL DAN PEMBAHASAN

Sebuah paket data mengandung segmen data yang 089 menyimpan informasi yang digunakan seperti *protocol*, *device name*, dan sebagainya. Dengan menggunakan sebuah Aplikasi *Capture Traffic Network*, dapat *capture* segala aktivitas lalu lintas yang terjadi pada sebuah jaringan computer ketika saat memulai terhubung keserver dan interaksi *client* ke *server*. Berikut hasil paket data yang telah di *capture* oleh Aplikasi *Capture traffic network* dapat dilihat pada table 1:

Tabel 1. Hasil paket data yang telah ditangkap

No	Device Name IP Source	IP Source	Device Name IP Destination	IP Destination	Paket Size	Type Packet
1	DESKTOP-2TLEC0T	192.168.43.127	-	192.168.43.255	70 bytes	BROADCAST
2	MODY	192.168.43.238	-	192.168.43.1	86 bytes	DNS REQUEST google.com
3	-	192.168.43.1	MODY	192.168.43.238	70 bytes	DNS RESPONSE google.com = 74,125,68,139
4	MODY	192.168.43.238	US	74.125.68.100	86 bytes	HTTP URL google.com/

1. Host DESKTOP-2TLEC0T dengan IP Source (192.168.43.127) mengirimkan HTTP Request kepada IP Destination (192.168.43.255) dengan type packet broadcast.
2. Host MODY dengan IP Source (192.168.43.238) mengirimkan HTTP request kepada web server google.com melalui remote.
3. IP Source (192.168.43.1) mendapatkan respon DNS dari google untuk IP (192.168.43.238).
4. Host MODY dengan IP Source (192.168.43.238) mengirim HTTP Request kepada google.com (74.125.68.100).

VI. PENUTUP
A. Kesimpulan

Aplikasi *Capture traffic network* yang dibangun menggunakan bahasa pemrograman java script dengan Server yang digunakan adalah Node.JS Linux Ubuntu 18.04, serta implementasi hanya terdapat pada layer 4 dan Ipv4 dalam satu LAN, Output data berupa JSON yang dikonversi ke dalam HTML, aplikasi yang di buat hanya dapat meng-*capture* lalu lintas pada sebuah

jaringan LAN. Dengan menggunakan Node.js kita dapat membuat programming model yang mudah dengan skala server yang cukup besar dengan mudah. Maka dari itu, penulis menarik kesimpulan sebagai berikut :

1. Node.js menggunakan bahasa pemrograman JavaScript yang diklaim sebagai bahasa pemrograman yang paling populer dan banyak dikenal oleh masyarakat luas
2. Node.js mampu menangani ribuan koneksi bersamaan dengan penggunaan *resource* minimum untuk setiap prosesnya
3. Node.js sangat diandalkan terutama untuk membuat aplikasi *real-time*
4. Node.js adalah project open source, sehingga siapapun dapat melihat struktur kode dan juga dapat berkontribusi untuk pengembangannya
5. Penggunaan JavaScript di sisi server dan juga client meminimalisir ketidakcocokan antar dua sisi lingkungan pemrograman, seperti terkait komunikasi data yang mana menggunakan struktur JSON yang sama di kedua sisi, validasi form yang sama yang dapat dijalankan di sisi server dan client, dan sebagainya.
6. Database NoSQL seperti MongoDB dan CouchDB mendukung langsung Javascript sehingga interfacing dengan database ini akan jauh lebih mudah.
7. Node.js memakai V8 yang selalu mengikuti perkembangan standar ECMAScript (nama standar resmi dari JavaScript, Namun JavaScript yang lebih dikenal dalam implementasinya), sehingga tidak perlu ada kekhawatiran bahwa browser tidak mendukung fitur-fitur di Node.js

Berdasarkan hasil rancang Aplikasi *Capture traffic network* yang dibangun penulis menarik kesimpulan sebagai berikut:

1. Dengan adanya Aplikasi *Capture traffic network* dapat memberikan informasi tentang lalu lintas jaringan LAN dan sebagainya
2. Dengan adanya Aplikasi *Capture traffic network* dapat mengcapture *traffic packet* pada sebuah jaringan LAN.
3. Dengan adanya Aplikasi *Capture traffic network* dapat membantu administrator jaringan untuk memonitoring *traffic* data dalam satu jaringan
4. Aplikasi *Capture traffic network* memudahkan pengguna atau seorang administrator jaringan dalam memantau Client dan kinerja Server melalui computer melihat lalu lintas jaringan.

DAFTAR PUSTAKA

- [1] Abdul Wahhab Solichin, 2015, Analisis Kebijakan Dari Formulasi Ke Penyusunan Model-model Implementasi Kebijakan Publik, (Jakarta : PT Bumi Aksara)
- [2] A. Herlina and P. M. Rasyid, "Sistem Informasi Monitoring Pengembangan Software Pada Tahap Development Berbasis Web," Jurnal Informatika, 2016.
- [3] Andrian, Allegra Shawabi, Sistem Informasi Penjualan Dan Monitoring Di Toko Royal Motor Bandung. Bandung: Jurnal Ilmiah Komputer dan Informatika (KOMPUTA). Edisi, 1. Vol, 1, 2014.
- [4] B. K. Williams and Sawyer, "Using information technology: a practical introduction to computers &

- communications : Complete version,” New York, NY: McGraw-Hill, 2011.
- [5] Deitel, Paul. Deitel, Harvey,”Java How To Program”,9th ed, USA : Prentice Hall, 2012.
- [6] Heryanto, A., Hermansyah, A. and Nizar, M. 2017, Sistem Monitoring Server Dan Perangkat Jaringan Pada Enterprise Resource Planning Fasilkom Unsri Menggunakan Protokol Icmp Dan Snmp. Palembang : Jurnal SISTEMASI. Volume 6 Nomor 3, 2017.
- [7] <http://cisco.netacad.net> semester 1, modul9.1
- [8] <https://medium.com/@tuplesedu/tcp-ip-6a2455ab9fa6>
- [9] Melwin Syafrizal, “Pengantar Jaringan Komputer”, 2015. Andi, Yogyakarta.
- [10] Mercy Corps, 2005, Design, Monitoring and Evaluation Guidebook. 28 September 2015, URL:<http://www.mercycorps.org/sites/default/files/1157150018.pdf>
- [11] M. Iqbal, M. Husni and H. Studiawan, "Implementasi Klien SIP Berbasis Web menggunakan HTML 5 dan Node.js," Jurnal Teknik ITS, vol. 1, pp. 242-243, 2012.
- [12] Morse, David R, Kissinger and the Yom Kippur War, New York: McFarland, 2015.
- [13] Rosa AS dan M.Shalahuddin. 2015. Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek. Bandung : INFORMATIKA.
- [14] Septian, Randy. 16 Maret 2016. Pengertian HTTP, Sejarah HTTP dan Cara Kerja HTTP.<http://www.randyseptian.web.id/pengertian-dan-cara-kerja-http.htm>.
- [15] Suryana, Taryana dan Koesheryatin. 2014. Aplikasi Internet Menggunakan HTML, CSS, & JavaScript. Jakarta: PT Elex Media Komputindo.