
Jurnal ***Rekayasa Elektrika***

VOLUME 13 NOMOR 2

AGUSTUS 2017

Analisis TCP Cubic dan Simulasi untuk Menentukan Parameter Congestion Window dan Throughput Optimal pada Jaringan Nirkabel Ad Hoc 65-69

Ary Firnanda, Teuku Yuliar Arif, dan Syahrial

JRE	Vol. 13	No. 2	Hal 65–118	Banda Aceh, Agustus 2017	ISSN. 1412-4785 e-ISSN. 2252-620X
-----	---------	-------	------------	-----------------------------	--------------------------------------

Analisis TCP Cubic dan Simulasi untuk Menentukan Parameter Congestion Window dan Throughput Optimal pada Jaringan Nirkabel Ad Hoc

Ary Firmanda, Teuku Yuliar Arif, dan Syahrial
Program Studi Magister Teknik Elektro Universitas Syiah Kuala
Jl. Tgk. Syech Abdurrauf No. 7, Banda Aceh 23111
e-mail: aryfirmanda2014@gmail.com

Abstrak—Performansi jaringan nirkabel ad hoc seringkali menurun yang diakibatkan terjadinya *congestion* pada proses pengiriman data. TCP Congestion Control digunakan untuk menyelesaikan permasalahan tersebut. TCP Cubic adalah salah satu variant TCP Congestion Control. Penelitian ini dilakukan dengan menguji dan membandingkan antara nilai variabel $\beta=0,2$ dengan nilai variabel β yang lain untuk digunakan pada jaringan nirkabel ad hoc. Penelitian ini dilakukan dengan metode eksperimental menggunakan perangkat lunak simulasi jaringan NS-3. Hasil penelitian menunjukkan bahwa nilai variabel β yang tepat pada jaringan nirkabel ad hoc dengan packet loss 5% untuk menghasilkan *congestion window max* yang optimal adalah 0,5 dan pada *throughput* rata-rata optimal adalah 0,1.

Kata kunci: Jaringan Nirkabel Ad Hoc, TCP congestion control, TCP cubic, congestion window, throughput

abstract—Ad hoc wireless network performance often declines caused the onset of congestion on the process of sending data. TCP Congestion Control used to solve these problems. TCP Cubic is one variant TCP Congestion Control. This research was conducted with the test and compare between variable value $b=0.2$ with the value of the variable b to be used in ad hoc wireless networks. This research was conducted with the experimental method using network simulation software NS-3. The results showed that the value of variable b is the right on ad hoc wireless network with packet loss by 5% to generate optimal congestion window max is $b=0.5$ and the average throughput optimal is $b=0.1$.

Keywords: Ad hoc Wireless Network, TCP congestion control, TCP cubic, congestion window, throughput

copyright © 2017 Jurnal Rekayasa Elektroika. All right reserved

I. PENDAHULUAN

Jaringan nirkabel *ad hoc* merupakan jaringan nirkabel yang akan banyak digunakan pada masa mendatang. Hal tersebut disebabkan untuk membangun jaringan nirkabel *ad hoc* tidak memerlukan infrastruktur yang memadai. Jaringan nirkabel *ad hoc* merupakan suatu jaringan yang dibangun tanpa infrastruktur dimana masing-masing *node* dilengkapi dengan *transceiver wireless* [1].

Berbeda dengan jaringan jenis lain yang memerlukan infrastruktur yang memadai. Hal tersebut menunjukkan bahwa jaringan nirkabel *ad hoc* lebih efisien dalam hal infrastruktur. Performansi jaringan nirkabel *ad hoc* seperti halnya jaringan nirkabel jenis lain seringkali menurun karena terjadi *congestion* pada proses pengiriman data. *Congestion* merupakan kemacetan dalam proses pengiriman data. Oleh karena itu, dibutuhkan protokol untuk mengatasi permasalahan *congestion*.

Transmission Control Protocol Congestion Control

(*TCP Congestion Control*) merupakan protokol yang telah digunakan secara luas untuk mengatasi permasalahan *congestion*. Ada beberapa variant *TCP Congestion Control* diantaranya yaitu *TCP Reno*, *TCP NewReno*, dan *TCP Cubic*. *TCP Reno* merupakan salah satu variant *TCP Congestion Control* yang hanya dapat bekerja dengan baik selama *packet loss* yang terjadi kecil. *TCP NewReno* yang merupakan pengembangan dari *TCP Reno* juga demikian. Pada jaringan nirkabel *ad hoc* sangat dimungkinkan terjadi *packet loss* yang tinggi. *TCP Cubic* merupakan salah satu variant *TCP Congestion Control* yang menggabungkan ide dasar dari *High-Speed TCP* dan *H-TCP* [2]. *TCP Cubic* [3] akan diuji untuk digunakan pada jaringan nirkabel *ad hoc*. Berdasarkan hal tersebut maka penelitian ini diperlukan sehingga mendapatkan data performansi jaringan nirkabel *ad hoc* dengan penggunaan *TCP Cubic*. Kemudian, meningkatkan performansi jaringan nirkabel *ad hoc* dengan penggunaan *TCP Cubic* agar performansi jaringan nirkabel *ad hoc* lebih optimal.

Pada *TCP Cubic* terdapat persamaan matematis yang berkaitan dengan perubahan ukuran *congestion window*. Pada persamaan tersebut terdapat parameter yang berfungsi untuk penurunan dan peningkatan ukuran *congestion window* [4].

Penelitian ini menggunakan metode penelitian eksperimental dengan penggunaan perangkat lunak simulasi jaringan *NS-3* [5].

II. STUDI PUSTAKA

TCP CUBIC [2] merupakan salah satu algoritma *congestion control*. *TCP CUBIC* adalah algoritma *TCP congestion control* yang saat ini digunakan secara *default* dalam *kernel linux* sejak versi kernel 2.6.19. Algoritma ini melakukan modifikasi pertumbuhan *window* secara linear yang terdapat dalam *TCP* standar menjadi fungsi *cubic* [4]. Algoritma pertumbuhan *window TCP CUBIC* adalah sebagai berikut, setelah terjadi sebuah *loss event* dan pengurangan *window*, maka W_{max} akan menjadi ukuran *cwnd* dan berkurang secara multiplikatif berdasarkan faktor β dimana β adalah sebuah konstanta penurunan dan *fast recovery*. Pertumbuhan *window* pada *TCP Cubic* mengikuti Persamaan (1) dan diilustrasikan pada Gambar 1 [4]:

$$W_{(t)} = C(t - K)^3 W_{max}, \quad (1)$$

Keterangan :

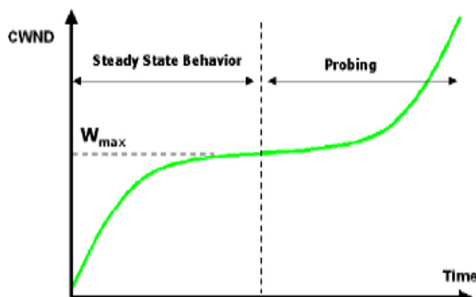
$W_{(t)}$: Ukuran *congestion window*
 C : Parameter *Cubic*
 t : Waktu dari pengurangan *window* terakhir
 K : Waktu dari pengubahan nilai dari W ke W_{max} jika tidak terdapat *loss event*
 W_{max} : *Window maksimum*

K dihitung dengan persamaan berikut [4]:

$$K = \sqrt[3]{\frac{W_{max} \beta}{C}}, \quad (2)$$

Keterangan :

K : Waktu dari pengubahan nilai dari W ke W_{max} jika tidak terdapat *loss event*
 W_{max} : *Window maksimum*
 β : Konstanta penurunan dan *fast recovery*
 C : Parameter *Cubic*



Gambar 1. Ilustrasi algoritma *TCP CUBIC* [4]

Saat *ACK* diterima selama fase *congestion avoidance*, *Cubic* menghitung laju pertumbuhan *window* selama periode *RTT* selanjutnya menggunakan persamaan (1) dengan memasukan nilai $W(t+RTT)$ sebagai kandidat baru dari *cwnd*. Bergantung pada besarnya nilai *cwnd*, *CUBIC* dapat berada dalam 3 kondisi/daerah. Pertama, jika *cwnd* bernilai lebih kecil dari besarnya *window* yang dapat dicapai oleh *TCP* standar (*Reno*) pada waktu t setelah *loss event*, maka *CUBIC* berada pada daerah *TCP* standar. Kedua, jika *cwnd* kurang dari W_{max} maka *CUBIC* berada dalam daerah *concave*, dan terakhir jika *CUBIC* lebih besar dari W_{max} , maka *CUBIC* berada dalam daerah *convex* [4].

A. Daerah *TCP*

Daerah *TCP* yang dimaksud pada *TCP Cubic* adalah kondisi dimana penambahan *window* menyerupai penambahan *window* pada *TCP Reno*. Untuk mengetahui *TCP Cubic* berada pada daerah *TCP* dapat dilihat dari besarnya *cwnd* dibandingkan dengan W_{tcp} yang didapatkan dengan rumus [4]:

$$W_{tcp(t)} = W_{max} (1 - \beta) + 3 \frac{\beta}{2 - \beta} \frac{t}{RTT}, \quad (3)$$

Keterangan :

$W_{tcp(t)}$ = *Window* yang dicapai oleh *TCP* standar (*Reno*) pada waktu t
 W_{max} = *Window maksimum*
 β = Konstanta penurunan dan *fast recovery*
 t = Waktu dari pengurangan *window* terakhir
 RTT = *Round Trip Time*

dimana jika *cwnd* kurang dari $W_{tcp}(t)$ maka *TCP Cubic* berada dalam daerah *TCP* dan besarnya nilai *cwnd* dihitung berdasarkan persamaan (3) untuk tiap *ACK* yang diterima.

B. Daerah *Concave*

Kondisi *concave* dicapai saat *TCP Cubic* menerima *ACK* pada mode *congestion avoidance*. Jika *TCP Cubic* tidak dalam kondisi *TCP* dan *cwnd* kurang dari W_{max} maka *TCP Cubic* berada dalam daerah *Concave*. Pertambahan *cwnd* diatur dengan persamaan berikut [4]:

$$cwnd_{new} = cwnd + \frac{W_{cubic}(t + RTT) - cwnd}{cwnd}, \quad (4)$$

Keterangan :

W = *Window*
 t = Waktu dari pengurangan *window* terakhir
 RTT = *Round Trip Time*
 $cwnd$ = *Congestion Window*.

C. Daerah *Convex*

Daerah *Convex* diperoleh ketika *cwnd* bernilai lebih besar dari W_{max} yaitu dimana *TCP CUBIC* telah melewati "dataran" nilai W_{max} . Setelah *cwnd* bernilai lebih besar

dari W_{max} maka dapat dianggap bahwa kondisi jaringan telah lebih baik daripada kondisi ketika terjadi *loss event* terakhir [4].

$$cwnd_{new} = cwnd + \frac{W_{cubic}(t + RTT) - cwnd}{cwnd}, \quad (5)$$

Keterangan :

W = Window

t = Waktu dari pengurangan *window* terakhir

RTT = Round Trip Time

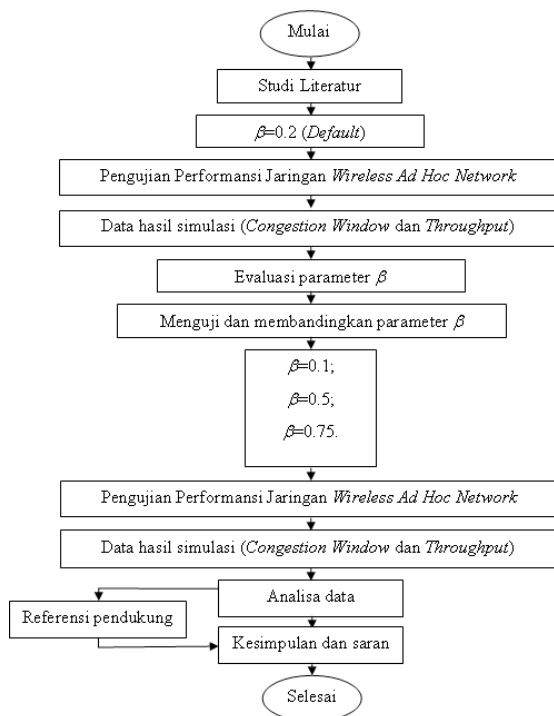
$cwnd$ = Congestion Window.

III. METODE

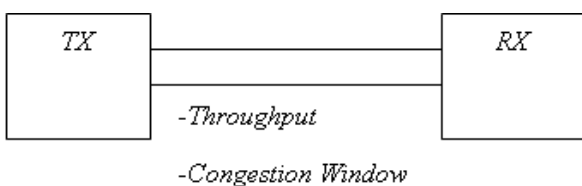
A. Objek dan Alur Penelitian

Metodologi penelitian yang digunakan adalah eksperimental dengan penggunaan perangkat lunak simulasi jaringan NS-3. Objek yang dikaji adalah parameter *TCP Cubic* yaitu parameter β . Alur penelitian disusun dalam beberapa tahapan agar penelitian berjalan secara sistematis seperti diperlihatkan pada Gambar 2.

Pengujian parameter β tersebut dilakukan pada jaringan dengan *single flow*. Kemudian, pengujian



Gambar 2. Blok diagram alur penelitian



Gambar 3. Kerangka pikir riset

dilakukan pada *packet loss* 5%. *Packet loss* 5% didapatkan dengan melakukan *ping* pada jarak berapa terdapat *packet loss* 5%. Hasil *ping* menunjukkan bahwa *packet loss* 5% terdapat pada jarak 6,913 m. Adapun mengenai kerangka pikir riset diperlihatkan pada Gambar 3.

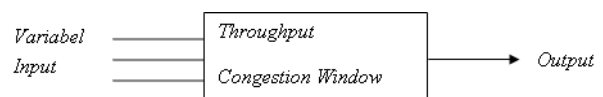
Berdasarkan Gambar 3 mengenai kerangka pikir riset dapat dijelaskan bahwa penelitian dilakukan untuk mendapatkan data mengenai kinerja *congestion window* dan *throughput* pada jaringan nirkabel *ad hoc* dengan *TCP Cubic*. Transmitter (TX) mengirimkan *packet* ke Receiver (RX). Kemudian, dari pengiriman *packet* tersebut akan didapatkan *throughput* yang diterima oleh Receiver (RX) terhadap *packet* yang dikirimkan oleh Transmitter (TX). Selain *throughput*, juga akan didapatkan data kinerja *congestion window* terhadap pengiriman *packet* dari Transmitter (TX) ke Receiver (RX).

B. Teknik yang Digunakan

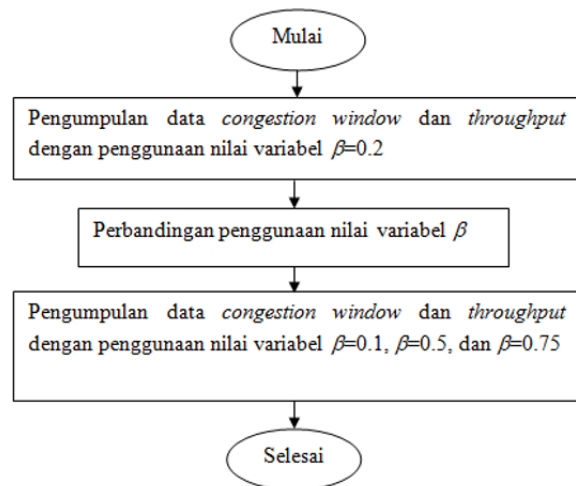
Teknik yang digunakan pada penelitian ini adalah pengumpulan data menggunakan perangkat lunak simulasi jaringan NS-3. Teknik yang digunakan diperlihatkan pada Gambar 4.

Berdasarkan Gambar 4 teknik yang digunakan adalah melakukan *input* terhadap nilai variabel pada jaringan nirkabel *ad hoc* dan *TCP Cubic*. *Input* terhadap nilai variabel pada jaringan nirkabel *ad hoc* yaitu variabel *Tx Power*, *Gain Antenna*, dan *Data Rate*. Sedangkan *input* terhadap nilai variabel pada *TCP Cubic* yaitu variabel β . Kemudian, akan didapatkan *output* berupa data kinerja *congestion window* dan *throughput*.

Data yang dikumpulkan yaitu *congestion window* dan *throughput*. Alur pengumpulan data yang akan dilakukan diperlihatkan pada Gambar 5.



Gambar 4. Teknik yang digunakan



Gambar 5. Alur pengumpulan data

C. Topologi Jaringan Yang Digunakan

Topologi jaringan yang digunakan pada penelitian ini adalah Topologi *Multi-hop*. Topologi *Multi-hop* diperlihatkan pada Gambar 6.

Berdasarkan Gambar 6 topologi *multi-hop* yang diuji pada penelitian ini menggunakan 5 *node*. *Node* tersebut terdiri dari *node source*, *router*, dan *destination*. *Node source* mengirimkan *packet* ke *node destination* melewati *node router* dimana *node router* terdiri dari 3 *node* yaitu *node router A*, *router B*, dan *router C*.

D. Konfigurasi Jaringan

Konfigurasi jaringan nirkabel *ad hoc* yang digunakan pada penelitian diperlihatkan pada Tabel 1.

Tabel 1. Konfigurasi jaringan

No.	Konfigurasi	Nilai	Keterangan
1.	Tx Power	16,0206 dB	Tx Power adalah daya yang ditransmisikan oleh pemancar.
2.	Gain Antenna	1 dB	Gain Antenna adalah karakter antena yang terkait dengan kemampuan antena mengarahkan radiasi sinyalnya, atau penerimaan sinyal dari arah tertentu.
3.	Jarak	6,913 m (Packet Loss 5%)	Jarak adalah panjang lintasan antar node.
4.	Data Rate	54 Mbps	Data Rate adalah Ukuran kecepatan bit data dalam proses transmisi, dihitung dalam bit per detik.
5.	IP Address	10.1.3.0/ 255.255. 255.0	IP Address adalah sebuah alamat pada komputer agar komputer bisa saling terhubung dengan komputer lain.
6.	Port	8080	Port adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan.
7.	Time	120 detik	Time adalah waktu yang digunakan untuk mengirimkan packet.
8.	Packet	1500 byte	Packet adalah kumpulan informasi yang dikirimkan kepada perangkat yang terletak pada jaringan intranet maupun internet.
9.	Jumlah Node	5 Node	Node adalah sebuah titik persambungan. Node dalam jaringan komputer adalah sebuah perangkat elektronik aktif yang terhubung ke satu jaringan, mampu mengirim dan menerima maupun meneruskan informasi melalui saluran komunikasi.



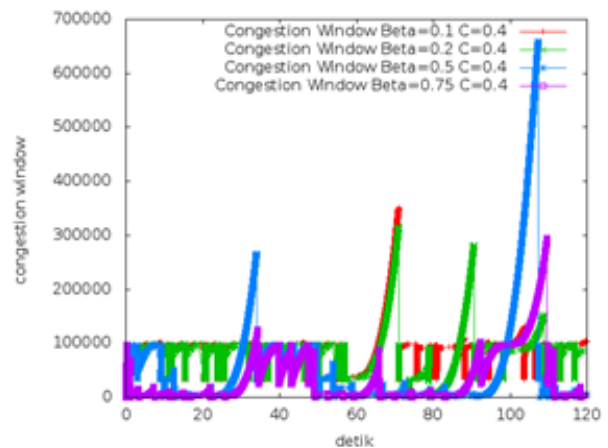
Gambar 6. Topologi multi-hop

IV. HASIL DAN PEMBAHASAN

Hasil pengujian *congestion window* dengan *single flow* pada *packet loss* 5% ditunjukkan pada Gambar 7.

Berdasarkan hasil pengujian yang ditunjukkan pada Gambar 7, pada *packet loss* 5% dari semua nilai β yang diuji menunjukkan *congestion window* berada pada fase *slow start*, *timeout*, *fast recovery*, dan *congestion avoidance*. *TCP Cubic* secara *default* menetapkan nilai β dengan 0.2. Variabel β merupakan konstanta penurunan untuk melakukan pengaturan *congestion window* setelah *packet loss*. Fase *congestion window TCP Cubic* diawali dengan *slow start*. Pada fase *slow start congestion window* meningkat secara eksponensial sehingga dapat terjadi *packet loss* sehingga *congestion window* berada pada fase *timeout*. Setelah fase *timeout congestion window* berada pada fase *fast recovery*. *TCP Cubic* menggunakan β untuk melakukan pengaturan *congestion window* dari fase *timeout* menuju fase *fast recovery*. Setelah fase *fast recovery*, *congestion window* menuju fase *congestion avoidance* dimana pada *TCP Cubic* fase *congestion avoidance* dikelompokkan dalam 3 daerah yaitu Daerah *TCP*, *Concave*, dan *Convex*. Penelitian ini melakukan pengujian untuk membandingkan performansi *congestion window max* antara penggunaan $\beta=0.2$ dengan penggunaan nilai β yang lain yaitu $\beta=0.1$, $\beta=0.5$, dan $\beta=0.75$.

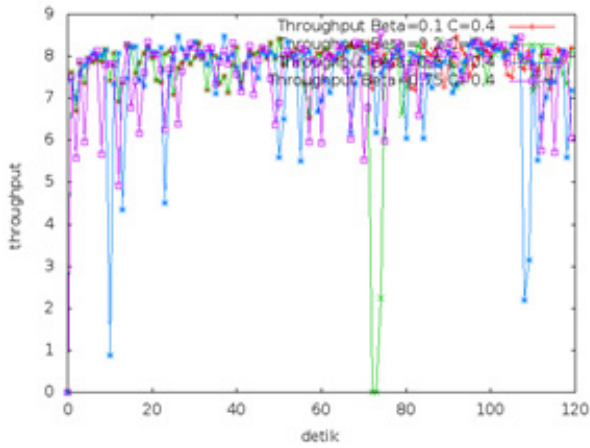
Hasil pengujian menunjukkan bahwa penggunaan $\beta=0.5$ dapat meningkatkan *congestion window max* sebesar 345479 MSS dibandingkan dengan penggunaan $\beta=0.2$. Hasil pengujian *congestion window max* pada *packet loss* 5% dengan penggunaan $\beta=0.1$, $\beta=0.2$, $\beta=0.5$, dan $\beta=0.75$ ditunjukkan pada Tabel 2.



Gambar 7. *Congestion window* dengan *single flow* pada *packet loss* 5%

Tabel 2. Perubahan fasa

β	Congestion Window Max
0.1	351401 MSS
0.2	314739 MSS
0.5	660218 MSS
0.75	294398 MSS



Gambar 8. Throughput dengan single flow pada packet loss 5%

Tabel 3. Throughput rata-rata pada packet loss 5%

β	Throughput Rata-Rata
0,1	7,807297417 Mbps
0,2	7,621077333 Mbps
0,5	7,448434983 Mbps
0,75	7,591904 Mbps

Hasil pengujian throughput dengan single flow pada packet loss 5% ditunjukkan pada Gambar 8.

Hasil pengujian throughput menunjukkan bahwa performansi throughput dengan penggunaan $\beta=0,1$, $\beta=0,2$, $\beta=0,5$, dan $\beta=0,75$ sangat fluktuatif. Penggunaan $\beta=0,1$ dapat meningkatkan throughput rata-rata lebih besar 0,18622 Mbps dibandingkan dengan penggunaan $\beta=0,2$. Akan tetapi, penggunaan $\beta=0,5$ dan $\beta=0,75$ menghasilkan throughput rata-rata lebih kecil yaitu 0,17264235 Mbps dan 0,029173333 Mbps dibandingkan dengan penggunaan $\beta=0,2$. Hasil pengujian throughput rata-rata pada packet loss 5% dengan penggunaan $\beta=0,1$, $\beta=0,2$, $\beta=0,5$, dan $\beta=0,75$ ditunjukkan pada Tabel 3.

V. KESIMPULAN

Berdasarkan data-data yang telah didapatkan dari hasil penelitian maka dapat disimpulkan bahwa nilai variabel β yang tepat pada jaringan nirkabel ad hoc dengan packet loss 5% untuk menghasilkan congestion window max yang optimal adalah 0.5. Sedangkan nilai variabel β yang tepat pada packet loss 5% untuk menghasilkan throughput rata-rata yang optimal adalah 0.1. Penelitian dilakukan pada jaringan nirkabel ad hoc dengan single flow. Penelitian dilakukan dengan menggunakan nilai variabel β yang bervariasi yaitu $\beta=0,1$, $\beta=0,2$, $\beta=0,5$, dan $\beta=0,75$. Hasil penelitian juga menunjukkan bahwa performansi congestion window dan throughput yang optimal tidak ditentukan oleh besar atau kecil nilai variabel β . Akan tetapi, performansi congestion window dan throughput yang optimal ditentukan oleh kesesuaian nilai variabel β yang digunakan. Kesesuaian yang dimaksudkan adalah penggunaan nilai variabel β yang tepat. Jadi, memperbesar nilai variabel β belum tentu akan dapat menghasilkan congestion window dan throughput yang optimal. Sebaliknya, memperkecil nilai variabel β belum tentu akan dapat menghasilkan congestion window dan throughput yang kurang optimal.

REFERENSI

- [1] K. Syarif, "Analisa Kinerja Ad-Hoc On Demand Distance Vector (AODV) Pada Komunikasi VMeS," Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
- [2] R. Fahrizal, W. Dewanto, S. Sumaryono, "Perbandingan Kinerja Algoritma BIC, CUBIC dan HTCP pada Topologi Parkinglot dan Multithome Menggunakan NS2" Seminar Nasional Informatika 2012 (semnasIF 2012) UPN "Veteran" Yogyakarta, 30 Juni 2012.
- [3] S. Ha, I. Rhee, L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," SIGOPS Oper. Syst. Rev., 42(5):64-74, Jul. 2008.
- [4] P.P. Rakimanputra, "Analisa Perbandingan Performansi TCP Cubic dan TCP Westwood+ pada Jaringan IEEE 802.11S," Laporan Tugas Akhir S1, ITT, Bandung, 2011.
- [5] B. Levasseur, M. Claypool, R. Kinicki, "A TCP CUBIC Implementation in ns-3," In Proceedings of the 2014 Workshop on ns-3 (WNS3 '14), pp. 1-8, May. 7, 2014.

Penerbit:

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala

Jl. Tgk. Syech Abdurrauf No. 7, Banda Aceh 23111

website: <http://jurnal.unsyiah.ac.id/JRE>

email: rekayasa.elektrika@unsyiah.net

Telp/Fax: (0651) 7554336

