

IMPLEMENTASI DAN ANALISIS PERFORMANSI METODE PFIFO FAST DAN RED PADA JARINGAN WIRELESS AD HOC MENGGUNAKAN PROTOKOL B.A.T.M.A.N

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF PFIFO FAST AND RED METHODS ON WIRELESS AD HOC NETWORK USING B.A.T.M.A.N PROTOCOL

Raden Bagus Nurhadi Wibowo¹, Istikmal, ST., MT.², Tody Ariefianto Wibowo, ST. MT.³

^{1,2}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

³Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom

¹bagoesnoerhadi@gmail.com ²istikmal@telkomuniveristy.co.id, ³tody@telkomuniversity.ac.id

Abstrak

Jaringan *wireless ad hoc* merupakan kumpulan *node wireless mobile* yang secara dinamis keberadaannya tanpa menggunakan infrastruktur yang ada atau administrasi terpusat. Pada jaringan *wireless ad hoc* ada beberapa permasalahan timbul yang diantaranya adalah pengaruh jumlah *node*, pengaruh pergerakan *node*, dan pengaruh besar ukuran paket yang dikirimkan sehingga menyebabkan performansi pengiriman datanya mengalami degradasi atau penurunan. Permasalahan tersebut dapat diminimalisasi dengan cara menggunakan metode antrian (*queue*) data. Dalam hal ini peneliti menggunakan 2 metode antrian data yaitu PFIFO Fast dan RED. Kedua metode antrian tersebut diimplementasikan dengan menggunakan ruting protokol B.A.T.M.A.N pada perangkat lunak berbasis Linux, yang kemudian dianalisis performansinya menggunakan *Simple Adittive Weighted Method* (SAW). Proses pengiriman dan penerimaan trafik pada sistem digunakan *Distributed Internet Traffic Generator* (DITG). Implementasi ini melibatkan 4 *nodes* dan dilakukan dalam 3 kondisi, yaitu seluruh *node* dalam kondisi diam (*fixed*), *node receiver* bergerak statis, dan *node sender* bergerak statis. Dari hasil implementasi jaringan ad-hoc dengan menggunakan kedua jenis metode antrian (*queue*) data tersebut secara menyeluruh dalam tiga kondisi (kondisi 1, kondisi 2, kondisi 3) menunjukkan bahwa metode RED dapat dinyatakan lebih baik daripada metode PFIFO Fast, dengan nilai rata-rata V untuk RED = 29.145 sedangkan V untuk PFIFO Fast = 25.073.

Kata kunci : *Wireless ad-hoc*, PFIFO Fast, RED, BATMAN, SAW, DITG

Abstract

Wireless ad hoc network is a collection of nodes wireless mobile that dynamically existence without the use of the existing infrastructure or centralized administration. In the ad hoc wireless network there are some problem arises that such are the effect of the number of nodes, the effect of the movement of the nodes, and the effect of the size of packets are sent thus causing the performance of data transmission are degraded. Those problems can be minimized by using a method of queuing data. In this case the researchers used two methods of data queuing that are PFIFO Fast and RED. Both methods queues are implemented using routing protocols BATMAN on software based Linux, which is then analyzed its performance using Simple Adittive Weighted Method (SAW). The process of sending and receiving traffic on the system using Distributed Internet Traffic Generator (DITG). This implementation involves four nodes and performed in three conditions, in example all nodes in the idle state (fixed), moving static of the receiver node, and moving static of the sender node. From the results of the implementation of ad-hoc networks using both types of methods queue data thoroughly under three conditions (condition 1, condition 2, condition 3) show that the RED method can be expressed better than PFIFO Fast method, the average value The average V for RED = 29.145 while V for PFIFO Fast = 25.073.

Keywords: *Wireless ad-hoc*, PFIFO Fast, RED, BATMAN, SAW, DITG

1. Pendahuluan

Jaringan *wireless ad hoc* [4][11] merupakan kumpulan *node (router) wireless mobile* yang secara dinamis keberadaannya tanpa menggunakan infrastruktur yang ada atau administrasi terpusat. Pada jaringan ini setiap *node* dapat bergerak bebas keluar masuk jaringan setiap saat tanpa dibatasi oleh cakupan daerah transmisi kartu jaringan. Sehingga diperlukan beberapa *node (multihop)* untuk dapat saling menghubungkan *node* dalam jaringan.

Pada jaringan *ad hoc*, mobilitas dari *node-node* nya dan kecenderungan *error* pada media nirkabelnya menghadapi banyak tantangan, termasuk perubahan frekuensi ruting dan paket loss. Permasalahan-permasalahan tersebut dapat meningkatkan jumlah *delay* paket serta mengurangi total paket yang diterima. Beban trafik di jaringan yang meningkat menyebabkan performansinya juga mengalami degradasi atau penurunan. *Active Queue Management (AQM)* diharapkan dapat memperkecil kemungkinan yang ditimbulkan beban trafik yang meningkat, sehingga bisa dihasilkan nilai *throughput* yang maksimal.

2. Dasar Teori

2.1. Jaringan Wireless

Jaringan *wireless (ad hoc)* [4] merupakan jaringan *wireless multihop* yang terdiri dari kumpulan *mobile node (mobile station)* yang bersifat dinamik dan spontan, dapat diaplikasikan dimana pun tanpa menggunakan jaringan infrastruktur (seluler ataupun PSTN) yang telah ada. Contohnya *mobile node* adalah *notebook* dan ponsel.

Jaringan *wireless* dapat diklasifikasikan menjadi dua kategori utama berdasarkan struktur jaringan yaitu terdiri dari jaringan dengan mode infrastruktur dan jaringan dengan mode tanpa infrastruktur (mode *ad hoc*). Mode infrastruktur memiliki konfigurasi yang sederhana dimana terdapat *Base Stasiun (BS)* yang memiliki cakupan area tertentu dan memberikan *service* pada *node* atau *user* yang berada pada cakupan areanya. Struktur jaringan yang kedua adalah mode *ad hoc* dimana dalam jaringan ini tidak terdapat infrastruktur seperti BS.

2.2. PFIFO Fast

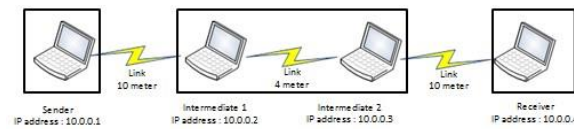
PFIFO Fast [10] merupakan mekanisme yang berdasarkan First In First Out. Antrian ini terdiri dari 3 unik antrian (qdisc) yang disebut "*Band*". Dalam masing-masing *band* aturan FIFO berlaku, namun selama ada paket yang menunggu dalam band 0 maka band 1 tidak diproses. Berlaku sama untuk band 1 dan band 2. Antrian ini tidak dikonfigurasi kecuali untuk *priority*, yang menentukan bagaimana paket dipetakan pada *band* yang berbeda berdasarkan Type of Service (TOS) pada paket IP. PFIFO Fast tidak mengubah urutan paket data, hanya menahan dan menyalurkan bila sudah memungkinkan. Jika *buffer* penuh, maka paket data akan *drop*. PFIFO Fast baik digunakan bila jalur tidak *congested*.

2.3. Random Early Detection (RED)

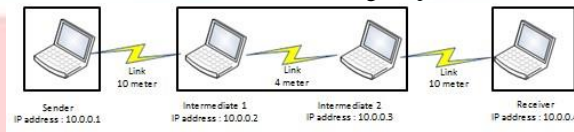
RED (*Random Early Detection*)[3] merupakan mekanisme antrian yang mencoba untuk menghindari kemacetan jaringan dengan mengontrol nilai rata-rata antrian. RED tidak melimit berdasarkan ukuran paket, tetapi bila *buffer* sudah penuh maka secara tidak langsung akan menyeimbangkan data *rate* setiap *user*. Saat ukuran *queue* rata-rata mencapai *min threshold* RED secara random akan memilih paket data untuk di *drop*. Saat ukuran rata-rata mencapai *max threshold* paket data akan didrop. Jika ukuran *queue* sebenarnya (bukan rata-ratanya) jauh lebih besar dari RED *max threshold* maka semua paket yang melebihi RED limit akan di *drop*. RED digunakan jika memiliki trafik yang *congested*. [8]

2.4. Percancangan Sistem

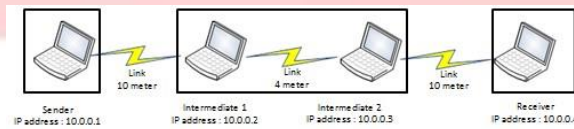
Skenario yang diuji terdiri dari tiga kondisi skenario, yaitu skenario *Fixed ad-hoc network* dimana empat *node static* (diam), *Static ad-hoc network* dengan pergerakan pada *Receiver*, *Static ad-hoc network* dengan pergerakan pada *sender*. Pergerakan *node* diimplementasikan dengan cara membawa *notebook* dengan berjalan kaki dengan kecepatan ± 3 km/jam. Model pergerakan dibatasi maksimal satu *node* yang bergerak, karena terbatasnya sumber daya manusia untuk melakukan pergerakan *node*.



Gambar 1. Model sistem tiga hop static



Gambar 2. Model sistem pergerakan receiver



Gambar 3. Model sistem pergerakan sender

Spesifikasi perangkat keras yang digunakan pada setiap nodes, sebagai berikut,

1. 2 Unit Advan G2T 80
Advan G2T 80 memiliki parameter seperti *processor* Intel Core 2 Duo T5800 @2 GHz, *wireless adapter* Broadcom, serta RAM 1 Gbytes
2. 2 Unit Acer Aspire One 532h
Acer Aspire One memiliki parameter seperti *processor* Intel Atom N450, *wireless adapter* Atheros, serta RAM 1 Gbytes

Selain dari konfigurasi model *hop* dan mobilitas yang digunakan, penelitian ini juga memiliki konfigurasi parameter sebagai berikut.

Tabel 1. Spesifikasi pengaturan sistem jaringan *ad-hoc*

No	Parameter	Keterangan
1.	<i>Transport protocol</i>	UDP
2.	Luas area implementasi	12 x 25 m
3.	<i>Environment</i>	<i>Indoor</i>
4.	Ukuran per paket	100, 500, 1000Bytes
5.	Durasi pengiriman paket per <i>sample</i>	10 detik
6.	Jumlah paket per <i>sample</i>	100 paket
7.	Jumlah sampel pengujian per skenario	10 sampel
8.	Jumlah <i>node</i> pengujian	4 <i>node</i>
9.	Frekuensi komunikasi	2.412 GHz
10.	<i>Traffic generator</i>	Distributed Internet Traffic Generator

Selain perangkat keras, perangkat lunak juga dibutuhkan untuk mengimplementasikan sistem. Berikut ini merupakan spesifikasi perangkat lunak yang dibutuhkan.

- | | |
|---|------------------|
| a. Ubuntu 12.10 | b. Batmand-0.3.2 |
| c. Distributed Internet Traffic Generator 2.8.0-rc1 | d. Iptables |
| e. Ping | |

3. Pembahasan

Untuk melakukan analisis data tersebut diatas, dalam pengambilan kesimpulan, peneliti menggunakan metode *Simple Additive Weighting* (Metode SAW) [5], metode SAW ini dipilih berdasarkan tingkat kecocokan yang tinggi dengan masalah dan data yang didapat dari hasil pengukuran.

Simple Additive Weighting Method (Metode SAW) mempunyai aturan sebagai berikut:

1. Peneliti memilih atribut yang bisa dijadikan kriteria yang mempengaruhi kinerja (dalam kasus ini, kinerja jaringan *Ad-hoc*) tidak semua atribut bisa menjadi kriteria kinerja jaringan,
2. Kemudian memilih atribut /kriteria yang menguntungkan dan yang merugikan, dengan indikator menguntungkan adalah apabila nilai atribut /kriteria semakin besar maka kinerja jaringan semakin baik, dan atribut /kriteria merugikan adalah apabila nilai atribut /kriteria semakin besar maka kinerja jaringan semakin tidak baik, (contoh ; jumlah paket yang diterima adalah kriteria yang menguntungkan dan paket loss merupakan kriteria yang merugikan).
3. Langkah berikutnya membuat matrik normalisasi dengan rumus sebagai berikut : (Max Min)

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\max X_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{i}{\min x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

4. Peneliti harus memberikan bobot (*Weighting / W*) pada setiap kriteria terhadap alternatif, dalam hal ini terhadap tingkat kepentingan keberhasilan kinerja jaringan *Ad-hoc*. (contoh: Paket diterima sangat penting, Paket Loss sangat tidak penting), dengan skala (1 sangat tidak penting, 2 tidak penting, 3 sedang, 4 penting, 5 sangat penting).
5. Menghitung nilai V, dengan cara mengalikan nilai W dengan nilai matrik ternormalisasi kriteria, dengan rumus sebagai berikut : (Sigma)

$$V_i = \sum_{j=1}^n W_j \cdot r_{ij}$$

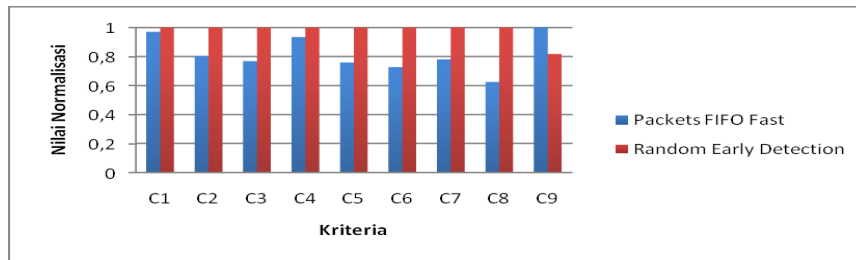
Nilai V yang lebih besar, itulah alternatif terbaik (dalam kasus penelitian ini adalah PFIFO Fast atau RED)

3.1. Fixed ad-hoc network dengan implementasi empat node static

Tabel 2. Nilai Kriteria Tiap Alternatif (Nilai V)

Kriteria	Packets FIFO Fast	Random Early Detection	Bobot (W)
C1	0.970	1.000	4
C2	0.804	1.000	5
C3	0.768	1.000	2
C4	0.935	1.000	3
C5	0.760	1.000	5
C6	0.727	1.000	4
C7	0.780	1.000	4
C8	0.625	1.000	1
C9	1.000	0.818	2
Nilai V	24.693	29.635	

Berdasarkan hasil perhitungan maka Nilai V yang terbesar adalah merupakan alternatif terbaik. Pada kondisi *Fixed ad-hoc network* dengan implementasi empat *node static* maka RED lebih baik kinerjanya dari pada PFIFO Fast.



Gambar 4. Perbandingan Hasil Pengukuran Antara PFIFO Fast dengan RED pada kondisi *Fixed ad-hoc network* empat node statis

Berdasarkan pada gambar grafik tersebut diatas menunjukkan bahwa pengiriman data pada kondisi *Fixed ad-hoc network* empat node statis maka pengiriman data dengan implementasi RED (*Random Early Detection*) lebih baik kinerjanya dibandingkan dengan implementasi PFIFO Fast.

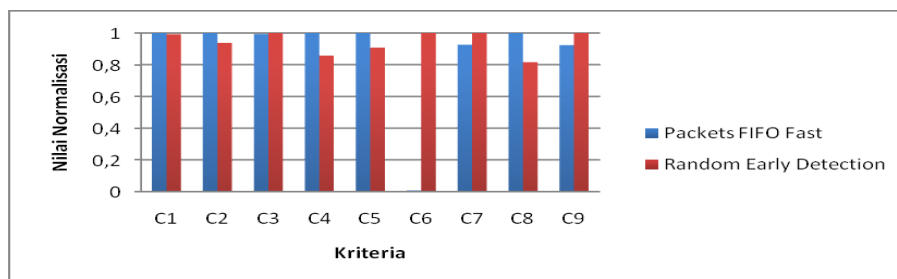
Hal ini terjadi karena PFIFO Fast tidak mengubah urutan paket data, hanya menahan dan menyalurkan bila sudah memungkinkan. Jika *buffer* sudah penuh, maka paket data akan langsung di *drop*. Lain halnya dengan RED (*Random Early Detection*) merupakan mekanisme antrian yang mencoba untuk menghindari kemacetan jaringan dengan mengontrol nilai rata-rata antrian. RED tidak melimit berdasarkan ukuran paket, tetapi bila *buffer* sudah penuh maka secara tidak langsung akan menyeimbangkan data *rate* setiap *user*. Saat ukuran antrian rata-rata mencapai *min threshold* maka RED secara random akan memilih paket data untuk didrop. Saat ukuran rata-rata mencapai *max threshold* paket data akan langsung didrop. Jika ukuran antrian sebenarnya (bukan rata-ratanya) jauh lebih besar dari RED *max threshold* maka semua paket yang melebihi RED limit akan langsung didrop.

3.2. *Static ad-hoc network* dengan pergerakan pada *Receiver*

Tabel 3. Nilai Kriteria Tiap Alternatif (Nilai V) pada *Receiver* bergerak

Kriteria	Packets FIFO Fast	Random Early Detection	Bobot (W)
C1	1.000	0.993	4
C2	1.000	0.937	5
C3	0.994	1.000	2
C4	1.000	0.857	3
C5	1.000	0.908	5
C6	0.007	1.000	4
C7	0.926	1.000	4
C8	1.000	0.816	1
C9	0.924	1.000	2
Nilai V	25.569	28.584	

Berdasarkan hasil perhitungan maka Nilai V yang terbesar adalah merupakan alternatif terbaik. Pada kondisi *Static ad-hoc network* dengan implementasi *receiver* bergerak maka RED lebih baik kinerjanya dari pada PFIFO Fast.



Gambar 5. Perbandingan Hasil Pengukuran Antara PFIFO Fast dengan RED pada kondisi *Static ad-hoc network* dengan *receiver* bergerak.

Berdasarkan pada gambar grafik tersebut diatas menunjukkan bahwa pengiriman data pada kondisi *Static ad-hoc network* dengan *receiver* bergerak maka pengiriman data dengan implementasi RED (*Random*

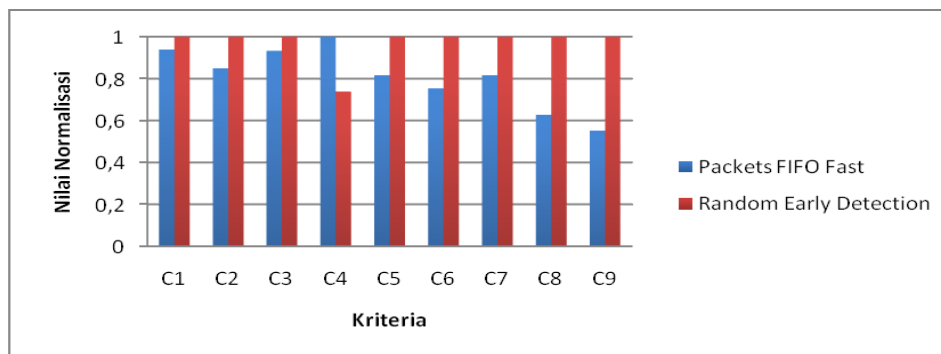
Early Detection) lebih baik kinerjanya dibandingkan dengan implementasi PFIFO Fast. Secara umum ide dasar di balik manajemen antrian RED adalah untuk mendeteksi kemacetan baru di awal dan menyampaikan pemberitahuan kemacetan ke-host end, yang memungkinkan mereka untuk mengurangi tingkat penulanan mereka sebelum antrian di *overflow* jaringan dan paket yang dijatuhkan (*drop*). Sedangkan PFIFO Fast merupakan algoritma yang didasarkan pada algoritma FIFO (*First in Firs out*) dengan cara kerja utamanya adalah tidak mengubah urutan paket data, hanya menahan dan menyalurkan bila sudah memungkinkan. Jika *bufer* penuh, maka paket data akan di *drop*. PFIFO Fast baik digunakan bila jalur tidak *congested*.

3.3. Static ad-hoc network dengan pergerakan pada sender

Tabel 4. Nilai Kriteria Tiap Alternatif (Nilai V) Pada *Sender* Bergerak

Kriteria	Packets FIFO Fast	Random Early Detection	Bobot (W)
C1	0.938	1.000	4
C2	0.849	1.000	5
C3	0.933	1.000	2
C4	1.000	0.738	3
C5	0.816	1.000	5
C6	0.753	1.000	4
C7	0.816	1.000	4
C8	0.627	1.000	1
C9	0.552	1.000	2
Nilai V	24.956	29.215	

Berdasarkan hasil perhitungan maka Nilai V yang terbesar adalah merupakan alternatif terbaik. Pada kondisi *Static ad-hoc network* dengan implementasi *sender* bergerak maka RED lebih baik kinerjanya dari pada PFIFO Fast.



Gambar 6. Perbandingan Hasil Pengukuran Antara PFIFO Fast dengan RED pada kondisi *Static ad-hoc network* dengan *sender* bergerak.

Berdasarkan pada gambar grafik tersebut diatas menunjukkan bahwa pengiriman data pada kondisi *Static ad-hoc network* dengan *sender* bergerak maka pengiriman data dengan implementasi RED (*Random Early Detection*) lebih baik kinerjanya dibandingkan dengan implementasi PFIFO Fast. Hal ini dikarenakan RED mengendalikan trafik jaringan sehingga terhindar dari kemacetan pada saat trafik tinggi berdasarkan pemantauan perubahan nilai antrian minimum dan maksimum. Jika isi antrian dibawah nilai minimum maka mode 'drop' tidak berlaku, saat antrian mulai terisi hingga melebihi nilai maksimum maka RED akan membuang (*drop*) paket data secara acak sehingga kemacetan pada jaringan dapat dihindari. Sedangkan PFIFO Fast tidak mengubah urutan paket data, hanya menahan dan menyalurkan bila sudah memungkinkan. Jika *bufer* penuh, maka paket data akan langsung di *drop*. PFIFO Fast baik digunakan bila jalur tidak *congested*.

4. Kesimpulan

1. Pada kondisi dimana empat *node* dalam kondisi diam /statis, *node* 1 sebagai *sender* dan *node* 4 sebagai *receiver* dengan menggunakan *simple additive weighthged method* (SAW) yang lebih baik adalah metode RED (*Random Early Detection*) dengan nilai $V = 29.635$ sedangkan PFIFO Fast (Packets FIFO Fast) dengan nilai $V = 24.693$.
2. Pada kondisi dimana tiga *node* dalam kondisi diam /statis, dan *node* 4 sebagai *receiver* dalam keadaan bergerak (*mobile*), dengan menggunakan *simple additive weighthged method* (SAW) yang lebih baik adalah metode RED (*Random Early Detection*) dengan nilai $V = 28.584$ sedangkan PFIFO Fast (Packets FIFO Fast) dengan nilai $V = 25.569$.
3. Pada kondisi tiga dimana tiga *node* dalam kondisi diam /statis, *node* 1 sebagai *sender* dalam keadaan bergerak (*mobile*), dengan menggunakan *simple additive weighthged method* (SAW) yang lebih baik adalah metode RED (*Random Early Detection*) dengan nilai $V = 29.215$ sedangkan PFIFO Fast (Packets FIFO Fast) dengan nilai $V = 24.956$.
4. Secara menyeluruh dalam tiga kondisi (kondisi 1, kondisi 2, kondisi 3), metode RED dapat dinyatakan lebih baik dengan nilai rata-rata V lebih besar, untuk RED = 29.145 sedangkan untuk PFIFO Fast = 25.073.

Daftar Pustaka:

- [1] Demichelis, C., Philip Chimento. 2002. *IP Packet Delay Variation Metric for IP Performance Metrics*. Internet Engineering Task Force.
- [2] COMICS (*COMputer for Interaction and CommunicationS*) Group. October 2013. "*D-ITG 2.8 Manual*" Department of Electrical Engineering and Information Technologies. University of Napoli "Federico II": Italy
- [3] Floyd, S. & Jacobson, V. *Random Early Detection Gateways for Congestion Avoidance*. Lawrence Berkeley Laboratory: University of California. IEEE/ACM Transactions on Networking Vol. 1, No. 4, August 1993, pp. 397-413. <http://ieeexplore.ieee.org>
- [4] Frodigh, M., Per Johansson, Peter Larsson. 2000. *Wireless ad hoc networking- The art of networking without a network*. Ericsson Review.
- [5] Kusumadewi, dkk. 2206. *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. Cahaya Ilmu . Yogyakarta.
- [6] Neumann, A., Corinna, A., Marek, L., Simon, W. 2008. *Better Approach to Mobile Ad-hoc Networking (B.A.T.M.A.N.)*. Internet Engineering Task Force.
- [7] Oehlmann, Fabian. 2011. *Simulation of the "Better Approach to Mobile Adhoc Networking" Protocol*. Technische Universitat Munchen.
- [8] Patil Ganesh, McClean Sally, and Raina Gaurav. *Drop Tail And RED Queue Management With Small Buffers: Stability And HOPF Bifurcation*. ICTACT Journal On Communication Technology: Special Issue On Next Generation Wireless Networks And Applications, June 2011, Volume – 2, Issue – 2
- [9] Pfifo_fast, the default Linux qdisc, [online], (<http://linux-ip.net/articles/Traffic-Control-HOWTO/classless-qdiscs.html#qs-pfifofast>, diakses tanggal 22 Juni 2015)
- [10] Simple, classless Queueing Disciplines, [online], (<http://lartc.org/howto/lartc.qdisc.classless.html#AEN658>, diakses tanggal 22 juni 2015)
- [11] Subir Kumar Sangkar, T.G. Basavaraju, C. Puttamadappa. 2008. *Ad hoc mobile Wireless Networks*. Auerbach Publications.