

Optimasi Algoritma *Rate Adaptation Control CARA* untuk Meningkatkan *Throughput* pada *Wireless Mesh Network*

Zahrul Maizi^{1*}, Teuku Yuliar Arif², Nasaruddin³

¹Jurusan Teknik Komputer, Fakultas Teknik, Universitas Serambi Mekkah, Banda Aceh

^{2,3}Jurusan Teknik Elektro dan Komputer, Fakultas Teknik, Universitas Syiah Kuala, Jln. Teuku Nyak Arief, Darussalam, Banda Aceh, 23111, Indonesia

*Koresponden email: zahrulmaizi@serambimekkah.ac.id

Diterima: 13 Mei 2020

Disetujui: 3 Desember 2020

Abstract

This paper examined throughput optimization issue in wireless mesh network (WMN), the weakest point of this network in regard to this matter. A number of previous studies on this issue have been conducted, but most focus has been on general wireless network, only few studies so far have attempted on this network. This research aimed to optimize the CARA rate adaptation control algorithm in WMN network. The optimization was performed by adjusting the successtreshold and timeout parameters in the CARA algorithm to obtain an optimal throughput. The optimal result is showed that the optimal points of the success threshold and timeout are at range of 30-35. It is obviously seen in grid 4×5 and 5×5 where the throughput value of the optimization result continues to increase. Moreover, by adding the data transmission time for 100 seconds on grid 5×5 resulting the throughput value of 0.52206412 Mbps and after the optimization the throughput value increases up to 117% to 1.1350768 Mbps, when the success threshold and timeout value are 30. For an additional 150 seconds, the throughput value is 0.5074419333 Mbps and after the optimization the throughput increases up to 120% to 1.1211402 Mbps, when the success threshold and timeout value are 35.

Keywords: *Wireless mesh network, throughput, rate adaptation control algorithm, CARA, NS-3.*

Abstrak

Penelitian ini membahas tentang masalah optimasi *throughput* pada jaringan *Wireless Mesh Network* (WMN), dimana hal ini sering menjadi titik lemah pada jaringan WMN. Penelitian ini bertujuan untuk melakukan optimasi terhadap algoritma *rate adaptation control CARA* pada jaringan WMN. Optimasi dilakukan melalui penyesuaian parameter *success threshold* dan *timeout* pada algoritma CARA untuk mendapatkan *throughput* yang optimal. Hasil optimasi menunjukkan bahwa nilai parameter *success threshold* dan *timeout* paling optimal berada pada range antara 30-35. Hal ini terlihat jelas pada grid 4×5 dan grid 5×5 dimana nilai *throughput* hasil optimasi yang didapatkan terus meningkat. Penambahan nilai waktu pengiriman data sebesar 100 detik pada grid 5×5 menghasilkan nilai *throughput* sebesar 0.52206412 Mbps, setelah dilakukan optimasi nilai *throughput*-nya meningkat hingga 117% menjadi sebesar 1.1350768 Mbps pada saat nilai *success threshold* dan *timeout* masing-masing adalah 30. Sedangkan pada penambahan nilai waktu pengiriman data sebesar 150 detik menghasilkan nilai *throughput* sebesar 0.5074419333 Mbps, setelah dilakukan optimasi nilai *throughput* meningkat hingga 120% menjadi sebesar 1.1211402 Mbps pada saat nilai *success threshold* dan *timeout* masing-masing 35.

Kata kunci: *Jaringan WMN, throughput, rate adaptation control algorithm, CARA, NS-3.*

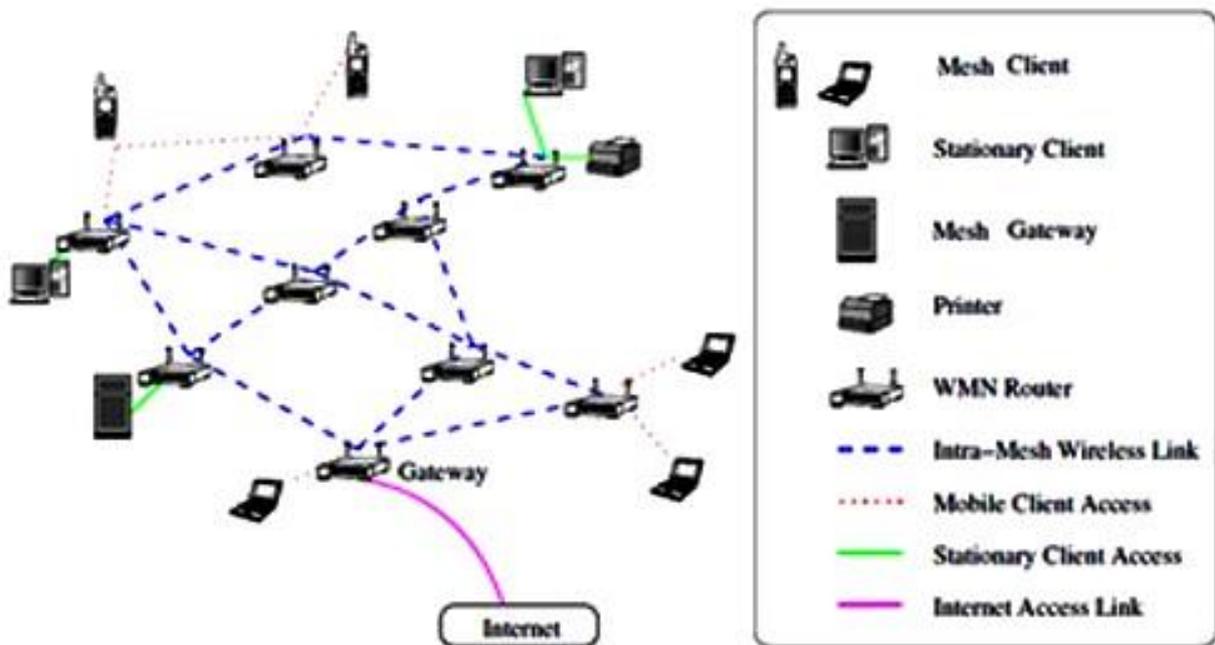
1. Pendahuluan

Jaringan *wireless mesh network* (WMN) merupakan teknologi jaringan alternatif baru yang sangat pesat perkembangannya dan dapat secara luas diterapkan untuk berbagai aplikasi jaringan karena berbagai keunggulan dan karakteristik yang dimilikinya, diantaranya adalah fitur biaya yang rendah dan konfigurasi yang lebih cepat [1]. Namun demikian, karena merupakan sebuah jaringan multihop yang kompleks, jaringan WMN ini masih memiliki beberapa keterbatasan misalnya dalam hal *throughput*, *loss rate*, dan *delay* yang merupakan penentu kinerja dan kualitas layanan akses internet di dalam jaringan ini. Ada banyak algoritma *rate adaptation control* yang telah dikembangkan dalam beberapa tahun terakhir, diantaranya dan yang paling umum digunakan adalah *auto rate fallback* (ARF), *adaptive auto rate fallback* (AARF), dan *collision aware rate adaptation* (CARA) [2]. Namun, algoritma-algoritma tersebut

masih memiliki keterbatasan dalam hal *throughput* yang dihasilkan dalam kondisi jarak yang jauh dan lalu lintas yang padat.

Pemilihan algoritma *rate adaptation control* CARA dalam penelitian ini berdasarkan penelitian sebelumnya [3] terkait evaluasi ketiga algoritma di atas, CARA menghasilkan *throughput* paling yang optimal dan yang paling sesuai dengan kondisi jaringan WMN yang kompleks untuk grid 4x5 dan 5x5 dibandingkan dengan ARF dan AARF. Pada penelitian sebelumnya, hasil evaluasi hanya untuk melihat perbandingan nilai *throughput* yang dihasilkan untuk setiap algoritma *rate adaptation control* tanpa dilakukan optimasi pada setiap parameter pendukung atau hanya menggunakan parameter bawaan untuk menghasilkan *throughput default*-nya dengan menggunakan simulator NS-3. Dalam penelitian ini algoritma CARA dilakukan optimasi lagi untuk meningkatkan *throughput default*-nya, dikarenakan pada jaringan WMN yang kompleks, *throughput* tersebut masih mengalami penurunan dalam keadaan tertentu.

Pada dasarnya WMN adalah sebuah jaringan wireless yang secara khusus terdiri dari sejumlah *mesh clients*, *gateway*, dan *mesh routers*. Biasanya *mesh clients* terdiri dari komputer, *handphone*, *bluetooth*, dan peralatan *wireless* lainnya. *Mesh router* dilengkapi dengan kemampuan *traffic aggregation* untuk menyediakan konektivitas jaringan ke *mobile clients* di dalam jangkauan area. *Mesh routers* tersebut membentuk sebuah *multihop wireless backbone* untuk mengirimkan trafik dari *clients* dan internet (*upstream traffic*) atau dari internet ke *clients* (*downstream traffic*). *Gateway* adalah router yang secara langsung terhubung dengan internet. Oleh karena itu, dalam jaringan tersebut, trafik biasanya dikirimkan oleh *wireless backbone* antara *mobile clients* dan internet melalui *gateway*.



Gambar 1. Bentuk umum jaringan WMN
Sumber: [4]

Salah satu karakteristik penting pada jaringan ini yaitu *throughput* dapat menurun karena tingginya volume lalu lintas yang diharapkan dapat secara efisien dikirimkan di dalam jaringan ini. *Rate adaptation control* merupakan penentu penting di dalam transmisi data yang optimal untuk kondisi jaringan seperti yang ada saat ini serta cukup menantang karena kondisi fluktuasi channel yang tidak menentu [5]. Tujuan *rate adaptation control* di dalam jaringan *wireless* adalah untuk meningkatkan sebesar mungkin transmisi untuk setiap node dan efektivitasnya sangat bergantung pada seberapa cepat responnya terhadap variasi channel jaringan. Namun untuk lingkungan yang *multi-user*, seperti pada jaringan WMN, efektivitasnya akan juga sangat bergantung pada pendeteksian keberadaan tabrakan dan penanganannya yang tepat.

Ada dua pendekatan yang umum digunakan untuk mengukur kondisi channel, yaitu dengan pendekatan *statistic-based* dan *signal-strength-based* [6]. Namun beberapa mekanisme yang diajukan ini masih memiliki beberapa keterbatasan. Mekanisme *signal-strength-based* mempunyai kekurangan dalam

hal korelasi yang kuat antara indikator SNR dan probabilitas pengiriman pada *data rate* yang diberikan. Selain itu, konfigurasi *data rate* diterapkan pada *sender node*, sementara informasi sinyal tersedia pada receiver, yang menyebabkan komunikasinya overhead [7]. Fakta ini membatasi keefektifan dari pendekatan ini di dalam praktek [8]. Mekanisme *statistic-based* dipengaruhi oleh kesulitan dalam penemuan *threshold* yang sesuai untuk seleksi *data rate* yang optimal.

ARF merupakan algoritma *rate adaptation control* pertama yang dipublikasikan dan desain untuk mengoptimasi *throughput* aplikasi pada WaveLan II yang menerapkan standar 802.11 DSSS. Di dalam ARF ini, setiap sender menggunakan status dari paket yang ditransmisikan sebelumnya untuk memilih *transmission rate* selanjutnya [9]. Secara spesifik, sebuah paket dikirimkan menggunakan skema ARF setelah sepuluh transmisi berturut-turut, baik gagal maupun berhasil, dimana setiap paket *probe* tersebut ditransmisikan pada suatu *bit-rate* yang lebih tinggi daripada yang sedang digunakan. ARF akan meningkatkan *transmission rate* jika paket *probe* tersebut berhasil, atau sebaliknya mengurangi *transmission rate* jika dua transmisi berturut-turut gagal [6].

Algoritma ini merupakan salah satu algoritma *rate adaption* generasi awal yang dikembangkan dari algoritma ARF. Algoritma ini menggunakan *frame error threshold* dinamis untuk meningkatkan atau mengurangi *bit-rate*. Seperti halnya ARF, AARF menggunakan paket *single probe* dalam menilai kondisi channel pada *rate* yang lebih tinggi, jadi setiap paket *single probe* tersebut berhasil ditransmisikan pada *rate* yang lebih tinggi, maka algoritma tersebut akan memutuskan untuk meningkatkan *rate* dengan sesuai [10]. AARF mampu meningkatkan stabilitas dengan menggandakan *probing threshold* hingga 50 ketika paket *probe* gagal ditransmisikan dan *probing threshold* tersebut akan kembali ke nilai awalnya 10 kapanpun *rate* tersebut berkurang.

Umumnya, algoritma *rate adaptation control* yang didasarkan pada jumlah paket yang ingin dikirimkan kembali dipengaruhi oleh adanya tabrakan paket yang menyebabkan pengurangan *rate* yang tidak perlu. Hal inilah yang mendorong dikembangkannya algoritma *Collision Aware Rate adaptation* (CARA) yang memanfaatkan teknik *loss differentiation* untuk mengatasi keterbatasan tersebut [11].

Pada dasarnya, prinsip kerja CARA adalah sama dengan ARF dimana keduanya sama-sama menggunakan mode penyesuaian *datarate*, dengan menghitung jumlah transmisi sukses berturut-turut dan juga jumlah transmisi gagal berturut-turut [12]. Namun bedanya, pada CARA, semua kegagalan transmisi diasumsikan sebagai akibat dari tabrakan antar data. Oleh karena itu, secara keseluruhan CARA lebih unggul daripada ARF karena kemampuannya untuk membedakan *frame collision* dari kegagalan *frame transmission* yang disebabkan oleh *channel error* dengan menggunakan metode *RTS probing* dan *clear channel assessment* (CCA) [11]. Dengan metode *RTS probing*, pada awalnya data ditransmisikan tanpa RTS/CTS. Penggunaan RTS/CTS baru dilakukan pada transmisi kedua setelah transmisi pertama gagal. Jika transmisi ulang ini kembali gagal, maka nilai *rate* akan diturunkan, sebaliknya jika transmisi ulang ini berhasil, maka nilai *rate* akan bertahan dan mentransmisi data selanjutnya tanpa RTS/CTS. Pertukaran RTS/CTS ini sendiri dilakukan secara adaptif pada *transmitter station*. Tahap selanjutnya, jika ditemukan kembali tabrakan, maka modul CCA akan digunakan.

Evaluasi terhadap kinerja algoritma CARA dilakukan melalui beberapa skenario topologi jaringan, ukuran *data frame* dan model *channel* wireless yang menunjukkan bahwa CARA melampaui kinerja algoritma ARF pada semua lingkungan yang melibatkan banyak pengguna [13]. Kemampuan CARA dalam mendeteksi tabrakan data menjadikan algoritma ini unggul pada lingkungan yang banyak pengguna. Oleh karena itu, pada jaringan mesh yang kompleks dengan aktivitas user dan jalur yang begitu rumit masih bisa menjadikan algoritma CARA menjadi algoritma yang unggul.

Salah satu hal penting dalam proses penyebaran paket data di dalam sebuah jaringan WMN adalah protokol *routing* (*routing protocol*). Pada dasarnya, *protocol routing ad hoc* dibagi menjadi dua, yaitu *on-demand* dan *table-driven*. Pada protokol *on-demand* (yang digunakan dalam penelitian ini), sebuah rute terbentuk hanya ketika sebuah node memiliki paket data untuk dikirimkan, sebaliknya pada protokol *table-driven* rute terbentuk secara berkelanjutan terlepas dari aktivitas trafik di dalam jaringan tersebut dimana setiap node membangkitkan pesan pengatur secara periodik. Salah satu protokol *on-demand* yang paling terkenal adalah *ad-hoc on-demand distance vector* (AODV).

AODV merupakan salah satu protokol *routing* yang paling banyak digunakan untuk jaringan ad-hoc dan WMN dan merupakan protokol yang reaktif atau dengan kata lain lebih peka terhadap jalur koneksi karena menggunakan teknik *on-demand* seperti telah dijelaskan di atas. Sebagai sebuah protokol yang reaktif, AODV mampu beradaptasi dengan cepat terhadap perubahan topologi di dalam sebuah jaringan dan memperbarui hanya host-host yang terkena dampak dari perubahan tersebut [14]. Pada awalnya, AODV ini dikembangkan untuk node-node pada radio-tunggal (*single-radio*) pada sebuah jaringan wireless ad-hoc, namun kemudian dikembangkan lagi untuk node-node pada *multiple-radio*.

IEEE 802.11 merupakan protocol jaringan yang sudah sangat populer dan paling banyak digunakan saat ini yang bekerja pada frekuensi 2.4 sampai dengan 2.5 GHz. *Physical layer* pada protocol ini mendukung kemampuan transmisi yang *multi-rate* dengan cara memilih teknik modulasi yang paling sesuai untuk *signal strength* yang diterima secara dinamis. Standar untuk protocol yang populer saat ini juga menyertakan beberapa parameter, salah satunya adalah *transmission rate* [8].

Software Network Simulator-3 (NS3) telah berkembang cukup pesat dalam menyediakan *tools* yang fleksibel dan mudah digunakan untuk simulasi berbagai kondisi jaringan, khususnya jaringan wireless dan telah digunakan secara luas khususnya para peneliti dan akademisi. NS-3 merupakan software *open source* yang dikembangkan di bawah lisensi GNU GPLv2 serta dari awal didesain agar mudah digunakan dan dimengerti [15]. NS-3 menggunakan bahasa C++ dan python. Karena berbasis *open source*, maka *source code* untuk software ini tersedia untuk hampir semua sistem operasi linux, seluruh varian unix, OS-X, dan windows dengan cygwin.

2. Metode Penelitian

Penelitian ini menggunakan metode kualitatif dengan menganalisis pengoptimasian algoritma *rate adaptation control* untuk mendapatkan *throughput* yang optimal dari suatu jaringan WMN melalui simulasi terhadap jaringan tersebut menggunakan sebuah software, yaitu NS3. Software NS3 merupakan sebuah simulator jaringan yang target utama penggunaannya adalah untuk kepentingan penelitian dan pendidikan. NS3 merupakan software gratis dibawah lisensi GNU GPLv2 yang tersedia secara publik. Pengoptimasian dilakukan terhadap algoritma *rate adaptation control* CARA dengan menyesuaikan beberapa parameter yang dapat mempengaruhi nilai *throughput*-nya. CARA dipilih karena menunjukkan kinerja yang paling sesuai untuk jaringan WMN.



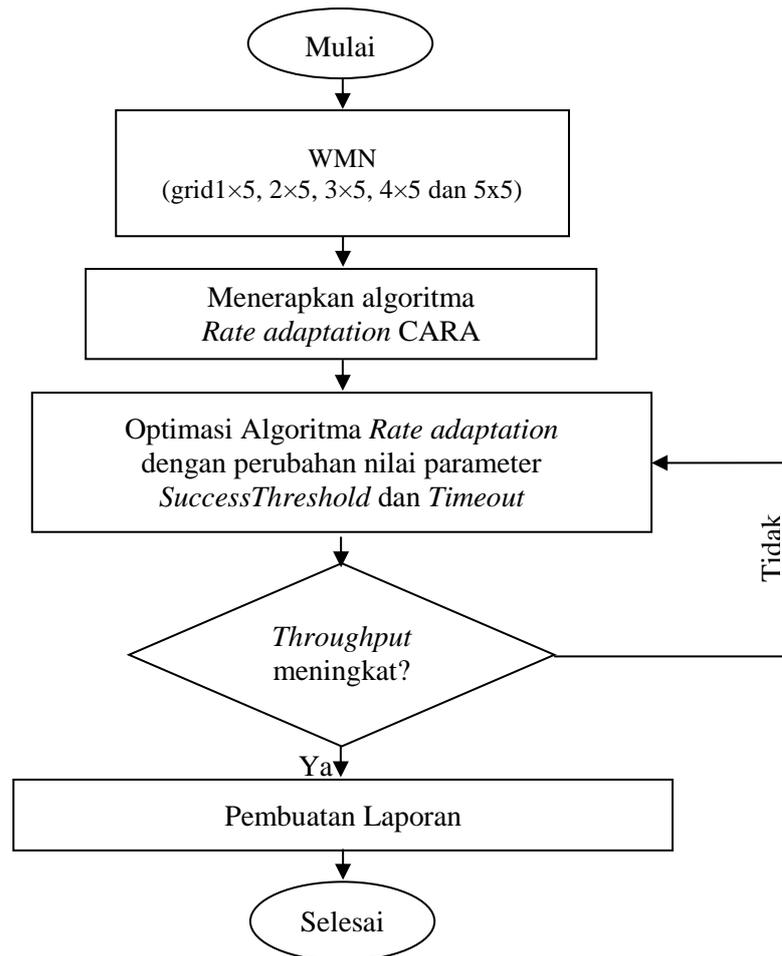
Gambar 2. Logo NS3

Software NS3 di *install* di sistem operasi Ubuntu dijalankan dengan menggunakan terminal linux dengan mode *text* tanpa ada grafis atau visual, **Gambar 2** hanya merupakan logo NS3. Persamaan yang digunakan untuk menghitung besarnya *throughput* adalah:

$$\text{Throughput} = \frac{\text{Rx bite paket diterima} * 8}{\text{total time Rx paket} - \text{timefirst Tx paket}} \dots\dots\dots(1)$$

Dimana:

- Rx bite* = jumlah paket yang diterima.
- Total time Rx paket* = waktu total paket diterima.
- Time first Tx paket* = waktu awal pengiriman paket.



Gambar 3. Tahapan pengujian

Gambar 3 merupakan alur dari pengujian akhir dengan tahapan-tahapan, yaitu (1) menerapkan algoritma *rate adaptation control* CARA yang akan digunakan untuk dioptimasi di dalam penelitian ini untuk kondisi grid 1x5, 2x5, 3x5, 4x5 dan 5x5; (2) optimasi dilakukan dengan mengubah nilai parameter *Success Threshold* dan *Timeout* pada algoritma CARA untuk mendapatkan nilai *throughput* yang paling optimal untuk kondisi jaringan WMN; (3) penilaian kondisi *throughput* dilakukan untuk melihat apakah nilai *throughput* setelah optimasi lebih besar dari sebelumnya atau tidak. Jika nilai hasil optimasi seperti yang diharapkan yaitu meningkat dibandingkan sebelumnya, maka proses akan dilanjutkan ke tahap selanjutnya, sebaliknya jika tidak, maka proses akan diulangi ke tahap sebelumnya yaitu pada tahap optimasi.

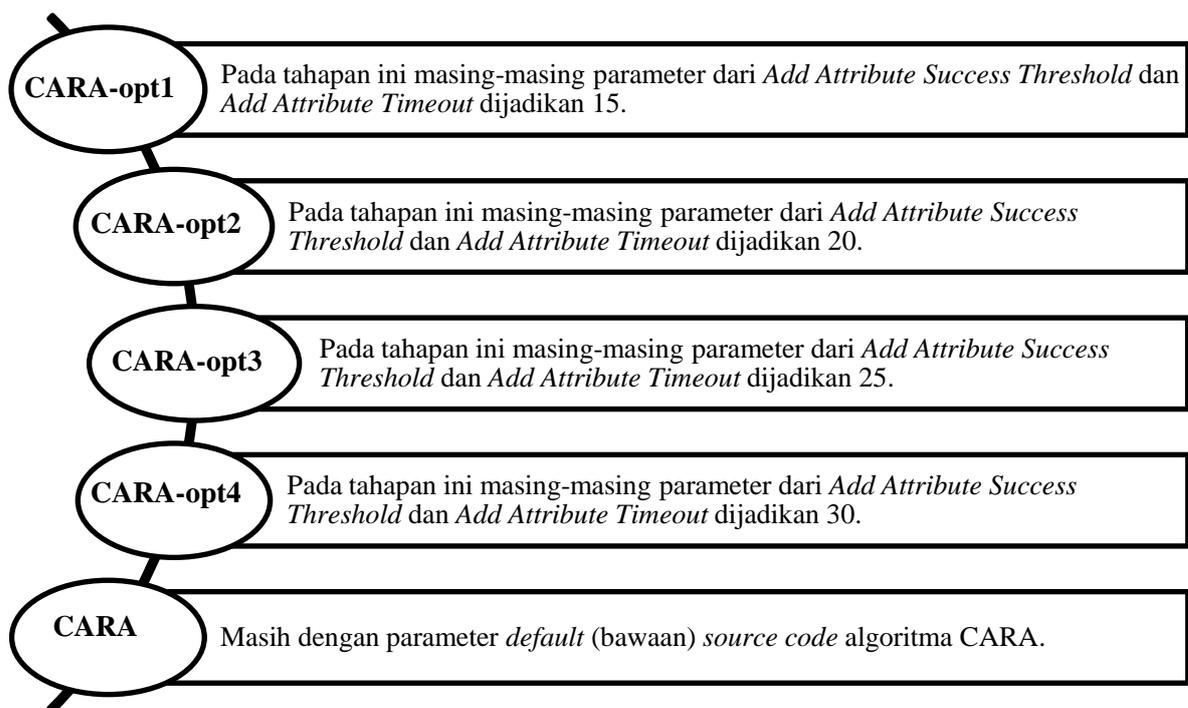
3. Hasil dan Pembahasan

Dalam penelitian ini, jaringan WMN terdiri dari beberapa node yang dibentuk dalam topologi grid, yaitu grid 1x5, 2x5, 3x5, 4x5, dan 5x5. Jarak antar setiap node-node tersebut dibuat konstan tanpa ada pengaruh interferensi dan *packet noise* di dalamnya, dimana jarak antar setiap node adalah 50 meter, memiliki ukuran paket data sebesar 1024 kbps menggunakan protokol *TCP* sebagai protokol komunikasi data dengan port 8080 dan *Internet Protocol* (IP) 10.1.3.0/24. Pada pengujian ini data rate yang dibangkitkan adalah 54 Mbps dan jenis protokol WiFi yang digunakan adalah 802.11a dengan waktu pengujian selama 50 detik untuk setiap node. Model *routing* yang digunakan dalam pengujian ini adalah *ad-hoc on-demand distance vector routing* (AODV). Pemilihan AODV sebagai *routing* adalah karena pertimbangan AODV merupakan protokol *routing* yang umum digunakan untuk kondisi jaringan dengan jumlah node yang banyak seperti pada jaringan WMN.

Transmisi data pada jaringan WMN memiliki banyak kemungkinan untuk terjadinya tabrakan data yang berakibat pada kegagalan transmisi. Kegagalan transmisi ini tentunya akan sangat berpengaruh terhadap *throughput* yang dihasilkan oleh jaringan tersebut. Oleh karena itu, untuk menjaga agar nilai

throughput tetap tinggi dan stabil, maka nilai jumlah *Attribute Success Threshold* dan *Attribute Timeout* harus dibuat lebih besar dari nilai pada kondisi *default*-nya. Dengan kata lain, memperbesar nilai *Attribute Success Threshold* dan *Attribute Timeout* berarti sama dengan menjaga kondisi *throughput* untuk mampu bertahan lebih lama pada nilai yang tinggi.

Attribute SuccessThreshold adalah jumlah minimum transmisi yang sukses untuk mencoba nilai data rate baru, sedangkan *Attribute Timeout* merupakan *timer* pada algoritma CARA. Nilai *Attribute Success Threshold* dan *Attribute Timeout default* dari algoritma CARA adalah 10 dan 15. Namun, dalam penelitian ini, nilai keduanya dibuat sama dan dinaikkan untuk keseluruhan kondisi optimasi. Nilai *Success Threshold* dibuat lebih besar dari kondisi *default*-nya, sedangkan nilai *Timeout* sama dengan nilai *Success Threshold*, hal ini tidak mengikuti kondisi *default* dimana nilai *Timeout* lebih besar daripada nilai *Success Threshold*. Penyamaan nilai ini dilakukan dengan pertimbangan bahwa ketika nilai *Success Threshold* telah dinaikan, maka nilai *Timeout*, yang menjadi waktu tunggu, tidak perlu diperbesar lagi karena kondisi *throughput* bisa tetap dipertahankan. Jika nilai *Timeout* dibuat lebih besar dari nilai *Success Threshold*, maka waktu tunggu yang diperlukan untuk menaikkan *data rate* tentunya akan lebih lama lagi dan hal ini tidak baik untuk kondisi jaringan WMN yang keadaan transmisinya sangat cepat karena jumlah *nodenya* yang begitu banyak.

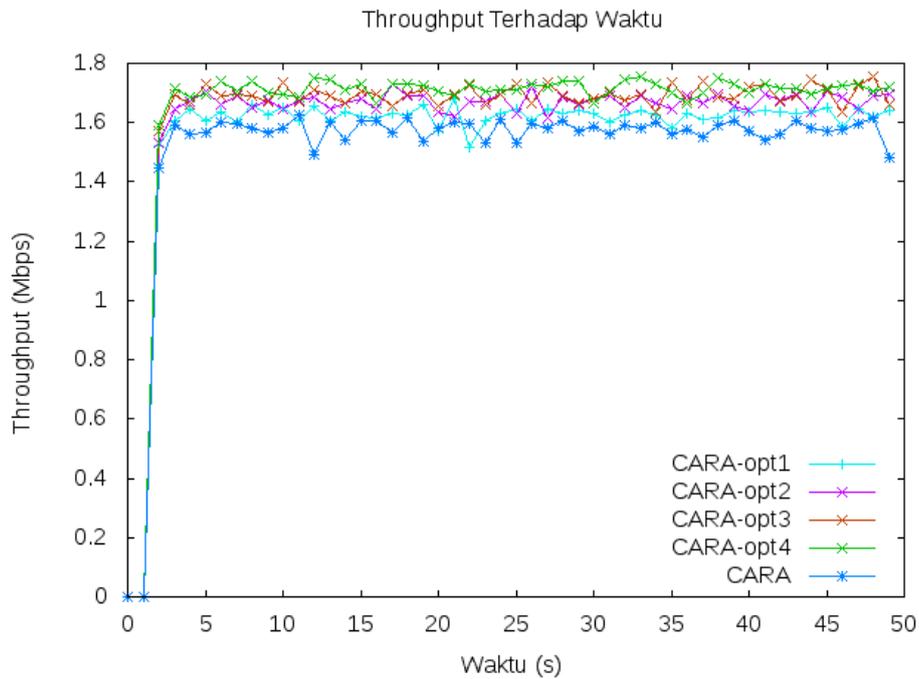


Gambar 4. Tahapan optimasi

Hasil pengujian throughput terhadap waktu pada algoritma CARA

a. Grid 1×5

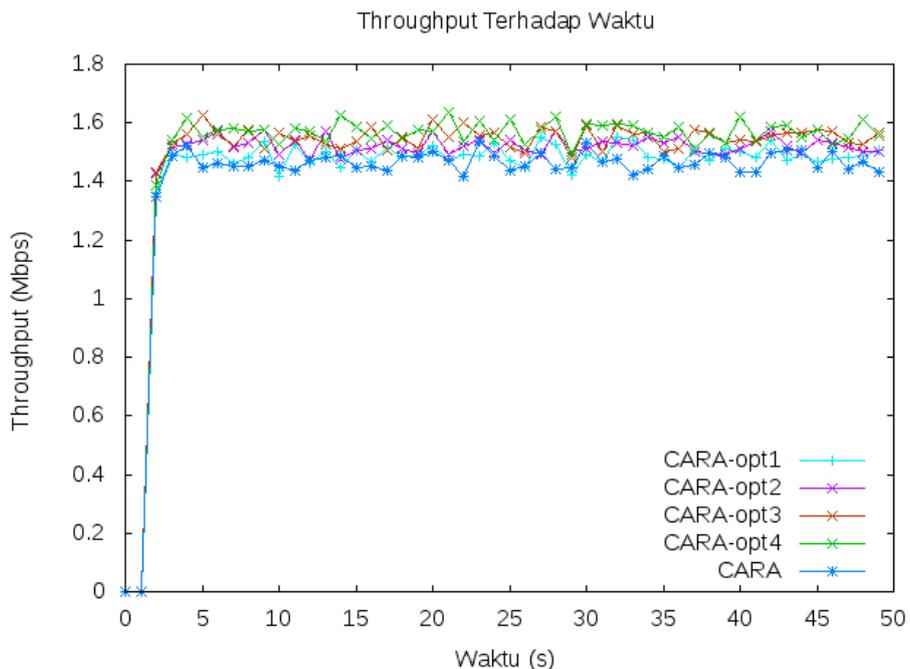
Nilai rata-rata *throughput* (**Gambar 5**) sebelum optimasi atau nilai *throughput* pada kondisi *default* adalah 1.5120344 Mbps. Setelah dioptimasi nilai rata-rata *throughput* CARA untuk topologi grid 1×5 secara berturut-turut adalah sebagai berikut: optimasi pertama CARA-opt1 adalah 1.5580872 Mbps; optimasi kedua CARA-opt 21.6019964 Mbps; optimasi ketiga CARA-opt3 1.6241232 Mbps; dan optimasi keempat CARA-opt4 1.6448762 Mbps.



Gambar 5. Optimasi Algoritma CARA grid 1×5
Sumber: Hasil penelitian, 2018.

b. Grid 2×5

Nilai rata-rata *throughput* (**Gambar 6**) sebelum optimasi adalah 1.4067214 Mbps. Setelah dioptimasi nilai rata-rata *throughput* CARA untuk topologi grid 2×5 secara berturut-turut adalah sebagai berikut: optimasi pertama CARA-opt1 adalah 1.4283326 Mbps; optimasi kedua CARA-opt2 1.4602354 Mbps; optimasi ketiga CARA-opt3 1.48562 Mbps; dan optimasi keempat CARA-opt 4 1.5043162 Mbps.

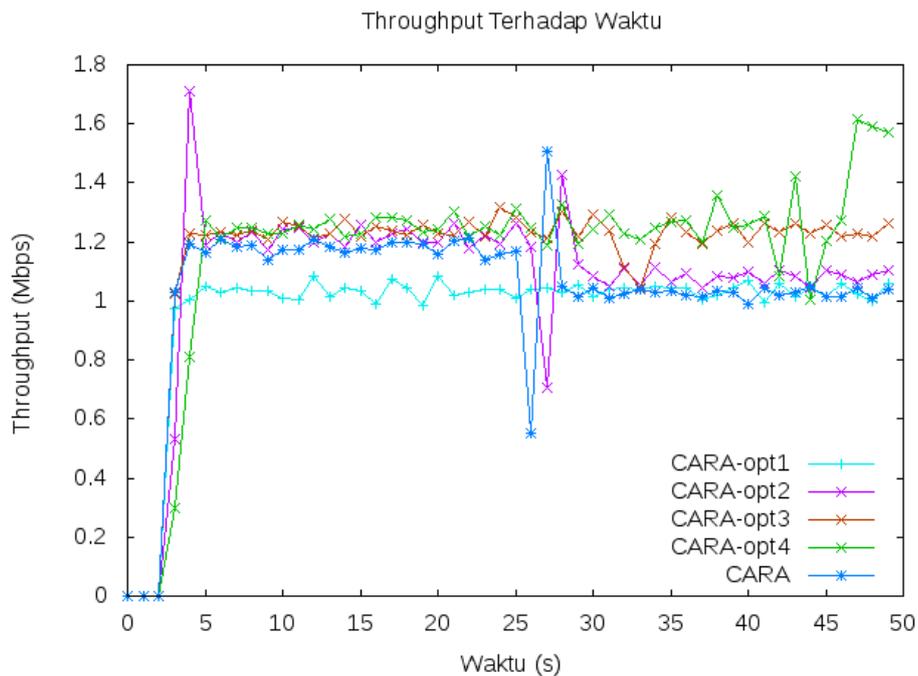


Gambar 6. Optimasi Algoritma CARA grid 2×5
Sumber: Hasil penelitian, 2018.

c. Grid 3×5

Nilai rata-rata *throughput* (**Gambar 7**) sebelum optimasi adalah 1.0327222 Mbps. Setelah dioptimasi nilai rata-rata *throughput* CARA untuk topologi grid 3×5 secara berturut-turut adalah sebagai berikut: optimasi pertama CARA-opt1 adalah 0.97011708 Mbps; optimasi kedua CARA-opt2

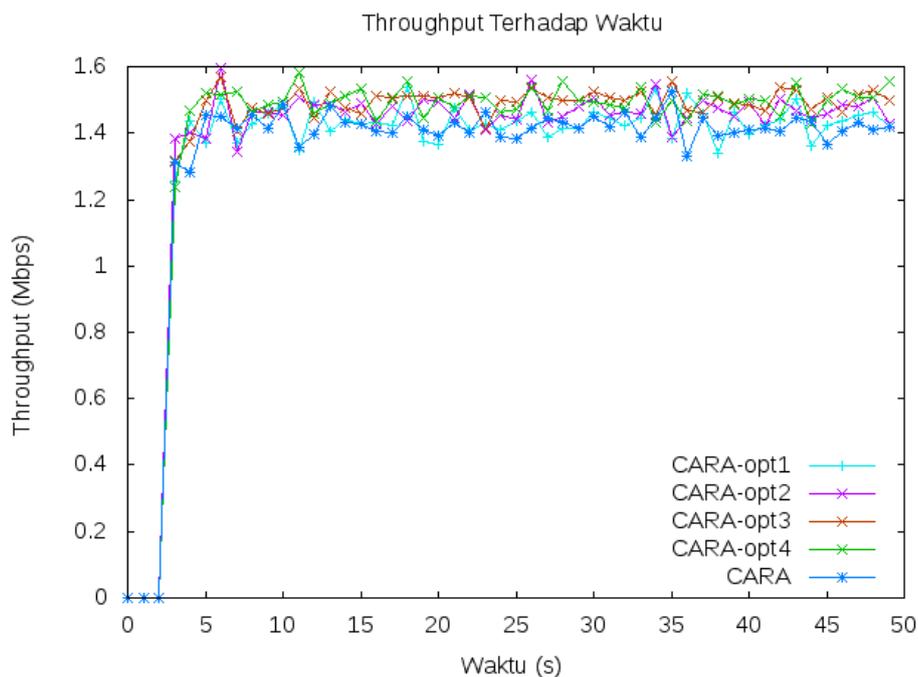
1.07748804 Mbps; optimasi ketiga CARA-opt3 1.1554436 Mbps; dan optimasi keempat CARA-opt4 1.16475468 Mbps.



Gambar 7. Optimasi Algoritma CARA grid 3×5
Sumber: Hasil penelitian, 2018.

d. Grid 4×5

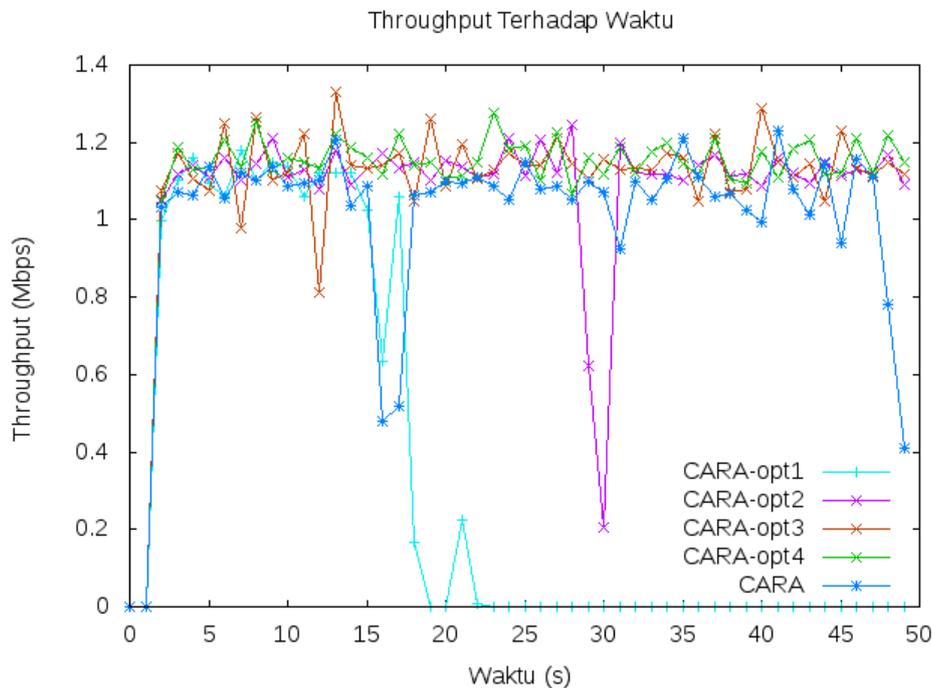
Nilai rata-rata *throughput* (**Gambar 8**) sebelum optimasi adalah 1.3321956 Mbps. Setelah dioptimasi nilai rata-rata *throughput* CARA untuk topologi grid 4×5 secara berturut-turut adalah sebagai berikut: optimasi pertama CARA-opt1 adalah 1.3438592 Mbps; optimasi kedua CARA-opt2 1.3762766 Mbps; optimasi ketiga CARA-opt3 1.400804 Mbps; dan optimasi keempat CARA-opt4 1.4023468 Mbps.



Gambar 8. Optimasi Algoritma CARA grid 4×5
Sumber: Hasil penelitian, 2018.

e. Grid 5×5

Nilai rata-rata *throughput* (**Gambar 9**) sebelum optimasi atau nilai *throughput* pada kondisi *default* adalah 0.99901848 Mbps. Setelah dioptimasi nilai rata-rata *throughput* CARA untuk topologi grid 5×5 secara berturut-turut adalah sebagai berikut: optimasi pertama CARA-opt1 adalah 0.35058744 Mbps; optimasi kedua CARA-opt2 1.05990848 Mbps; optimasi ketiga CARA-opt3 1.09215312 Mbps; dan optimasi keempat CARA-opt4 1.1108496 Mbps. Pada optimasi grid 5×5 memperlihatkan hasil optimasi yang mampu memperbaiki keadaan nilai *throughput* yang sebelumnya (tanpa optimasi) menjadi meningkat dan mampu mempertahankan kondisi *throughput* yang lebih stabil sampai pada detik ke-50 akhir pengiriman.



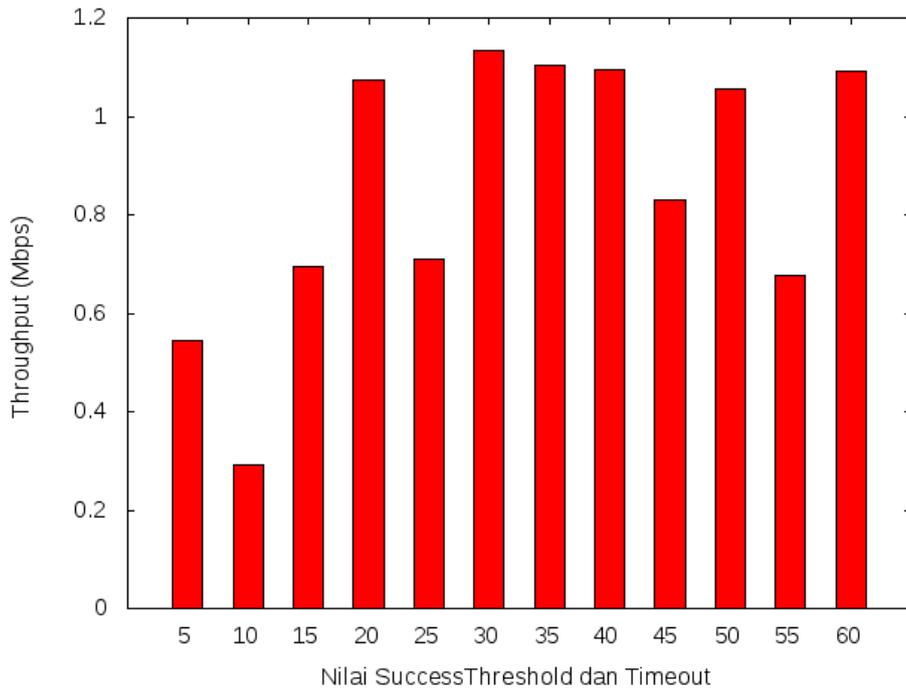
Gambar 9. Optimasi Algoritma CARA grid 5×5
Sumber: Hasil penelitian, 2018.

Dari keseluruhan pengujian di atas memperlihatkan bahwa *throughput* akan semakin meningkat pada saat nilai *Add Attribute Success Threshold* dan *Add Attribute Timeout* dinaikkan dengan waktu pengiriman data sebesar 50 detik. Dalam penelitian ini, pengujian yang dilakukan adalah untuk mempertahankan kondisi jumlah minimum transmisi yang berhasil dikirim dengan nilai dan waktu yang lebih lama untuk menaikkan dan mempertahankan *data rate*-nya sehingga nilai *throughput* menjadi lebih tinggi dan lebih stabil, pada jaringan WMN yang sangat kompleks penerapan kondisi seperti ini sangat bagus dikarenakan tabrakan (*collision*) pada proses pengiriman data sangat besar, sedangkan pada jaringan biasa tabrakannya tidak terlalu besar. Dengan mempertahankan nilai *Add Attribute Success Threshold* dalam waktu yang lebih lama maka kenaikan *throughput* akibat adanya kenaikan *data rate* akan lebih terjaga.

Hasil pengujian *throughput* terhadap nilai *success treshhold*, *timeout* dan waktu

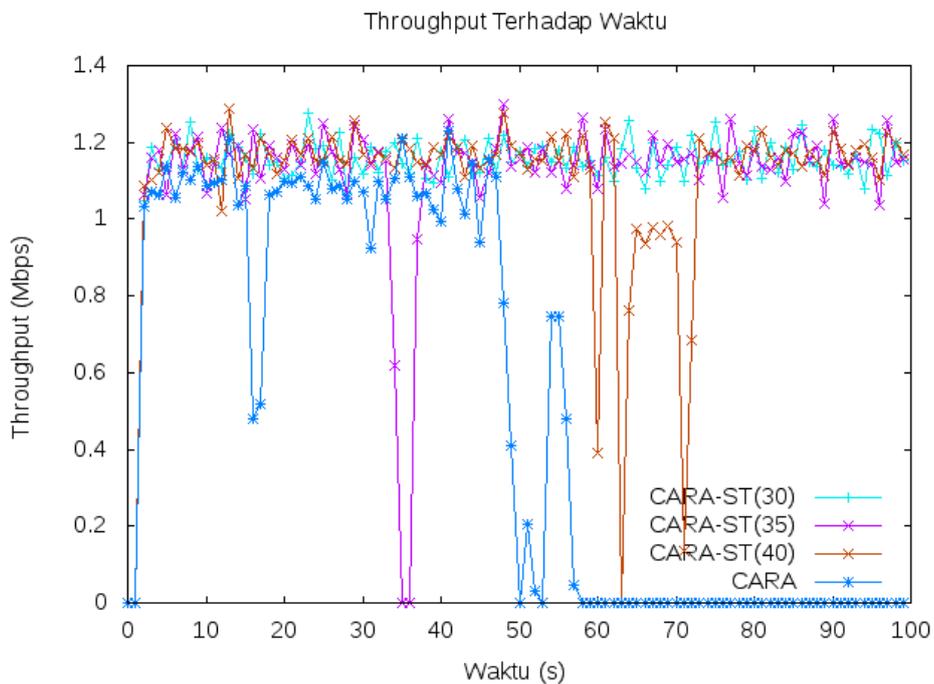
a. Waktu pengiriman data 100 detik

Gambar 10 merupakan hasil pengujian *throughput* terhadap nilai *Success Treshold* dan *Timeout*. Dengan memperbesar waktu pengiriman data sebesar 100 detik, nilai optimasi parameter *Success Treshold* dan nilai *Timeout* yang optimal berada pada range 30 hingga 40. Penambahan nilai waktu pengiriman data dilakukan agar hasil optimasi lebih terlihat dibandingkan sebelum dilakukannya optimasi, memudahkan untuk melihat nilai *Success Treshold* dan nilai *Timeout* yang optimal untuk kondisi jaringan WMN yang kompleks dengan grid 5x5.



Gambar 10. Hasil pengujian *throughput* untuk waktu pengiriman data sebesar 100 detik
 Sumber: Hasil penelitian, 2018.

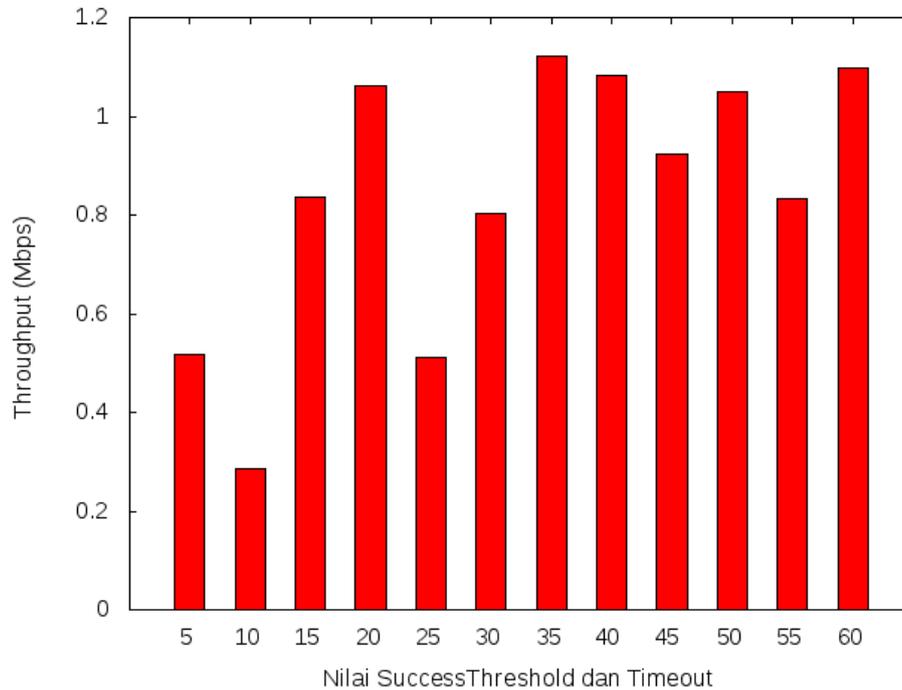
Gambar 11 menunjukkan hasil pengujian *throughput* terhadap waktu, dengan range optimasi *Success Threshold* dan *Timeout* 30 sampai 40. Penambahan waktu pengiriman data sebesar 100 detik menyebabkan jatuhnya nilai *throughput* yang signifikan selama 40 detik, yaitu mulai detik ke-60 sampai dengan detik ke-100 pada kondisi dimana belum dilakukan optimasi. Akan tetapi setelah dilakukan optimasi, nilai *throughput* dapat meningkat kembali. Nilai parameter *Success Threshold* dan *Timeout* yang optimal berada pada nilai 30 (CARA-ST(30)) tanpa ada penurunan nilai *throughput* a sampai pada detik ke-100.



Gambar 11. Hasil pengujian *throughput* terhadap waktu untuk waktu pengiriman data sebesar 100 detik
 Sumber: Hasil penelitian, 2018.

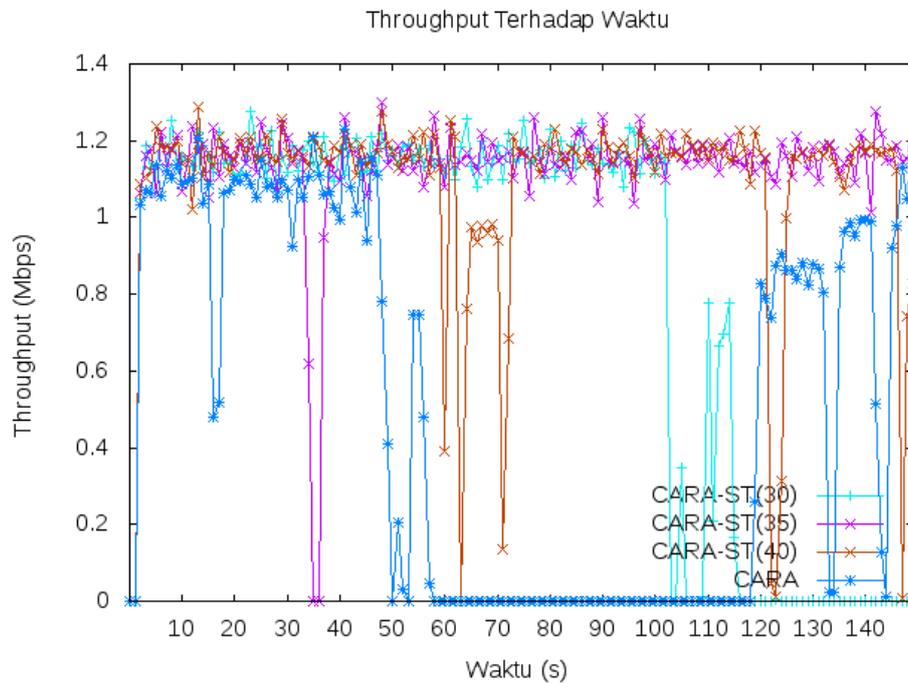
b. Waktu pengiriman data 150 detik

Gambar 12 merupakan hasil pengujian *throughput* dengan memperbesar waktu pengiriman data hingga 150 detik, kondisi nilai optimasi parameter *Success Threshold* dan nilai *Timeout* yang optimal berada pada range 35 hingga 40. Range tersebut merupakan kondisi yang paling ideal untuk sebuah jaringan WMN, karena jika nilai *Success Threshold* dan *Timeout* semakin tinggi, maka waktu tunggu untuk menaikkan *throughput* juga semakin besar.



Gambar 12. Hasil pengujian *throughput* untuk waktu pengiriman data sebesar 150 detik
Sumber: Hasil penelitian, 2018.

Untuk waktu pengiriman data sebesar 150 detik, waktu jatuhnya nilai *throughput* yang signifikan sebelum dilakukan optimasi menjadi lebih lama yaitu selama 60 detik dari detik ke-60 hingga detik ke-120. Walaupun kemudian meningkat kembali, namun kondisi nilai *throughput* masih belum stabil hingga detik ke-150. Setelah dilakukan optimasi nilai *throughput* mampu dinaikkan kembali pada range waktu tersebut dengan kondisi yang lebih stabil. Nilai parameter *Success Threshold* dan *Timeout* yang optimal berada pada nilai 35 (CARA-ST(35)) yang hanya mengalami sedikit penurunan *throughput*, seperti yang terlihat pada **Gambar 13**.



Gambar 13. Hasil pengujian *throughput* terhadap waktu untuk waktu pengiriman data sebesar 150 detik
 Sumber: Hasil penelitian, 2018.

4. Kesimpulan

Penelitian ini bertujuan untuk melakukan optimasi nilai *throughput* untuk algoritma CARA pada jaringan WMN kompleks (grid 4×5 dan grid 5×5). Algoritma ini dipilih karena menunjukkan performa yang paling baik dibandingkan dua algoritma lainnya, yaitu ARF dan AARF, untuk kondisi jaringan yang kompleks dikarenakan algoritma ini mampu melihat tabrakan data yang sangat umum terjadi pada jaringan WMN yang sering menjadi penyebab gagalnya transmisi data yang pada akhirnya menjatuhkan nilai *throughput*.

Optimasi dilakukan terhadap dua parameter yang mampu mempengaruhi nilai *throughput* pada algoritma CARA, yaitu *success threshold* dan *timeout*. Hasil optimasi menunjukkan bahwa nilai parameter *success threshold* dan *timeout* paling tinggi berada pada range antara 30-35, yang berarti bahwa nilai *throughput* paling optimal didapatkan pada range tersebut. Hal ini terlihat jelas pada grid 4×5 dan grid 5×5 dimana nilai *throughput* hasil optimasi yang didapatkan terus meningkat.

Lebih lanjut, penambahan nilai waktu pengiriman data sebesar 100 detik pada grid 5×5 menghasilkan nilai *throughput* sebesar 0.52206412 Mbps, kemudian setelah dilakukan optimasi nilai *throughput*-nyameningkat hingga 117% menjadi sebesar 1.1350768 Mbps pada saat nilai *success threshold* dan *timeout* masing-masing adalah 30. Sedangkan pada penambahan nilai waktu pengiriman data sebesar 150 detik menghasilkan nilai *throughput* sebesar 0.5074419333 Mbps, kemudian setelah dilakukan optimasi nilai *throughput* meningkat hingga 120% menjadi sebesar 1.1211402 Mbps pada saat nilai *success threshold* dan *timeout* masing-masing 35.

5. Referensi

- [1] A. Capone & G. Carello, Scheduling optimization in *wireless mesh networks* with power control and rate adaptation. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, (Vol. 1, pp. 138-147. IEEE), 2006.
- [2] D. Senthilkumar & A. Krishnan, "Throughput analysis of IEEE 802.11 multirate WLANs with collision aware rate adaptation algorithm," *International Journal of Automation and Computing*, 7(4), 571-577, 2010.
- [3] Z. Maizi, T. Arif dan Y. Nasaruddin, "Aceh Physics Society: Evaluasi Algoritma Rate Adaptation Untuk Wireless Mesh Network," Vol. 7, No. 2 pp.85-91, 2008.
- [4] M. L. Sichitiu, "Wireless mesh networks: opportunities and challenges," in *Proc. Wireless World Congress*, pp. 1-6, 2005.
- [5] W. Biaz & S. Wu, "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison," In *Computers and Communications*, pp. 130-136, IEEE, 2008.

- [6] S. H. Wong, H. Yang, S. Lu & V. Bharghavan, "Robust *rate adaptation* for 802.11 wireless networks," In *Proceedings of the 12th Annual International Conference on Mobile computing and networking* (pp. 146-157), ACM, 2011.
- [7] Q. Xia & M. Hamdi, "Smart sender: a practical *rate adaptation* algorithm for multi-rate IEEE 802.11 WLANs," *Wireless Communications, IEEE*, Vol. 7, pp. 1764-1775.
- [8] S. Pal, S.R. Kundu, K. Basu, & S.K. Das, "IEEE 802.11 rate control algorithms: Experimentation and performance evaluation in infrastructure mode," In *Passive and Active Measurement Conference*, 2006.
- [9] Y. Song, X. Zhu, Y. Fang, & H. Zhang, "Threshold optimization for *rate adaptation* algorithms in IEEE 802.11 WLANs," *Wireless Communications, IEEE Transactions on*, 9 (1), 318-327, 2010.
- [10] M. Lacage, M.H. Manshaei, & T. Turletti, IEEE 802.11 *rate adaptation*: a practical approach. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, (pp. 126-134). ACM, 2004.
- [11] J. Kim, S. Kim, S. Choi, & D. Qiao, "CARA: Collision-Aware *Rate adaptation* for IEEE 802.11 WLANs," In *INFOCOM*, Vol. 6, pp. 1-11, 2006.
- [12] M.A. Mohamed, W. M. Bahget, W. M., & Mohamed, S. S. (2014). A performance evaluation for *rate adaptation* algorithms in IEEE 802.11 wireless networks. *Int J Comput Appl*, 99(4), 54-59.
- [13] J. Chen, Y.Z. Lee, D. Maniezzo, & M. Gerla, "Performance comparison of AODV and OFLSR in wireless mesh network," In *IFIP Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)* (pp. 1-8), 2006.
- [14] A. Pirzada & M. Portman, "High performance AODV routing protocol for hybrid wireless mesh networks," In *Mobile and Ubiquitous System: Networking and Services, Mobiquitous 2007. 4th Annual International Conference on* (pp. 1-5), IEEE, 2007.
- [15] G. Brar, D. M. Blough & P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," In *Proceedings of the 12th annual international conference on Mobile computing and networking* (pp. 2-13), ACM, 2006.