ANALISIS KINERJA PROTOKOL AODV (*AD HOC ON-DEMAND DISTANCE VECTOR*) DAN AOMDV (*AD HOC ON-DEMAND MULTIPATH DISTANCE VECTOR*) TERHADAP SERANGAN AKTIF PADA JARINGAN MANET (*MOBILE AD HOC NETWORK*)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muhammad Alif Bahari
NIM: 145150200111132

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

Analisis Kinerja Protokol
AODV (*Ad Hoc On-Demand Distance Vector*) dan
AOMDV (*Ad Hoc On-Demand Multipath Distance Vector*) Terhadap Serangan
Aktif Pada Jaringan MANET (*Mobile Ad Hoc Network*)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Alif Bahari
NIM: 145150200111132

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Ir.Primantara Hari Trisnawan, M. Sc.	Reza Andria Siregar, S.T., M.Kom.
NIP. 19680912 199403 1 002	NIP. 19790621 200604 1 003

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Desember 2018

Muhammad Alif Bahari

NIM: 145150200111132

PRAKATA

Puji syukur kehadirat Allah SWT yang telah memberikan Rahmad, Taufik dan Hidayah-Nya sehingga laporan skripsi yang berjudul "Analisis Kinerja Protokol AODV (*Ad Hoc On Demand Distance Vector*) Dan AOMDV (*Ad Hoc On Demand Multipath Distance Vector*) Terhadap Serangan Aktif Pada Jaringan MANET (*Mobile Ad Hoc Network*)" ini dapat terselesaikan dengan baik.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ir.Bapak Primantara Hari Trisnawan, M. Sc. dan Bapak Reza Andria Siregar, S.T., M.Kom. selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika.
4. Kedua orang tua dan seluruh keluarga besar atas segala nasehat, motivasi, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa, semangat dan arahan demi terselesaikannya skripsi ini.
5. Teman-teman satu angkatan Program Studi Informatika 2014 tercinta yang selalu memberikan informasi, semangat, dorongan dan bantuan pikiran.
6. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penyelesaian skripsi ini.
7. Semua pihak yang telah membantu dan berbagi ilmu dalam penyelesaian skripsi ini yang tidak dapat penulis sebutkan satu per satu.
8. Semua sahabat yang telah memberikan doa, Reyhan, Kemas, Restu, Arig, Reza, dan Eno. Serta sahabat bermain dota Rifqi, Yogi, Aji, Angky, dan Adit. Teman-teman seperjuangan skripsi Ficry, Imam, Eko, Asroful, Ilin, KY, dan Bli.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 26 Desember 2018

Muhammad Alif Bahari

ABSTRAK

Muhammad Alif Bahari, Analisis Kinerja Protokol AODV (*Ad Hoc On-Demand Distance Vector*) dan AOMDV (*Ad Hoc On-Demand Multipath Distance Vector*) Terhadap Serangan Aktif Pada Jaringan MANET (*Mobile Ad Hoc Network*)

Pembimbing: Ir.Primantara Hari Trisnawan, M. Sc. dan Reza Andria Siregar, S.T., M.Kom.

MANET (*Mobile Ad hoc Network*) adalah jaringan yang tidak memiliki infrastruktur yang tetap dan *node-node* yang berada di dalamnya berfungsi sebagai router untuk meneruskan informasi yang dikirimkan. MANET memiliki 3 jenis protokol *routing* yaitu protokol *routing proactive*, *reactive*, dan *hybrid.* AODV (*Ad Hoc On Demand Distance Vector*) dan AOMDV (*Ad Hoc On Demand Multipath Distance Vector*) merupakan protokol berjenis *reactive* yang terdapat pada MANET. Keamanan pada MANET adalah sebuah tantangan yang besar, karena *node* akan saling terhubung pada sebuah jaringan nirkabel yang tidak terproteksi. Berdasarkan permasalahan di atas, maka penulis membuat penelitian yang berjudul Analisis Kinerja Protokol AODV dan AOMDV Terhadap Serangan Aktif Pada Jaringan MANET (Mobile Ad hoc Network). Hasil yang didapatkan dari penelitian ini adalah terbukti bahwa serangan aktif berupa serangan *Blackhole* dan DDoS bertipe *Flooding RREQ Attack* memengaruhi kinerja kedua protokol. Pengujian pada penelitian ini dilakukan pada *Network Simulator 2* dan diukur berdasarkan 3 parameter yaitu *Packet Delivery Ratio, End-To-End Delay,* dan *Normalized Routing Load.* Hasil pengujian menunjukkan bahwa rata-rata nilai *Packet Delivery Ratio* paling rendah terdapat pada skenario 100 *node* menggunakan protokol AOMDV dengan jenis serangan DDoS yang bernilai 52,00%. Sedangkan nilai rata-rata *End-to-End Delay* paling tinggi terdapat pada skenario 100 *node* menggunakan protokol AODV dengan jenis serangan DDoS yang bernilai 134,25 ms. Nilai rata-rata *Normalized Routing Load* paling tinggi terdapat pada skenario 100 *node* menggunakan protokol AOMDV dengan jenis serangan DDoS yang bernilai 158,8.

Kata Kunci : MANET, AODV, AOMDV, Blackhole, DDoS

ABSTRACT

Muhammad Alif Bahari, ***Performance Analysis of AODV (Ad Hoc On-Demand Distance Vector) and AOMDV (Ad Hoc On-Demand Multipath Distance Vector) Protocols Against Active Attacks on MANET Networks (Mobile Ad Hoc Network)***

Supervisors: Ir.Primantara Hari Trisnawan, M. Sc. and Reza Andria Siregar, S.T., M.Kom.

MANET (Mobile Ad hoc Network) is a network that does not have a fixed infrastructure and the nodes in it function as routers to forward the information that is sent. MANET has 3 types of routing protocols, namely proactive, reactive and hybrid routing protocols. AODV (Ad Hoc On Demand Distance Vector) and AOMDV (Ad Hoc On Demand Multipath Distance Vector) are reactive type protocols found on MANET. Security on MANET is a big challenge, because nodes will be connected to one unprotected wireless network. Based on the above problems, the authors made a study entitled Performance Analysis of AODV Protocol and AOMDV Against Active Attacks on MANET (Mobile Ad hoc Network) networks. The results obtained from this study are proven that active attacks in the form of Blackhole and DDoS attacks of type Flooding RREQ Attack affect the performance of the two protocols. Tests in this study were conducted on Network Simulator 2 and measured based on 3 parameters, namely Packet Delivery Ratio, End-To-End Delay, and Normalized Routing Load. The test results show that the lowest value of Packet Delivery Ratio is found in the 100 node scenario using the AOMDV protocol with the type of DDoS attack that is worth 52.00%. While the highest end-to-end delay value is found in the 100 node scenario using the AODV protocol with DDoS attack types that are worth 134.25 ms. The highest average value of Normalized Routing Load is found in the 100 node scenario using the AOMDV protocol with DDoS attack types valued at 158.8.

Keywords: MANET, AODV, AOMDV, Blackhole, DDoS

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar belakang

Kebutuhan manusia untuk berkomunikasi merupakan salah satu kebutuhan utama. Tetapi bencana alam dapat terjadi dimana saja dan kapan saja seperti contohnya gempa bumi, tsunami, dan gunung meletus. Bencana alam seperti ini tidak dapat terhindari dan seringkali dapat merusak sarana dan prasarana infrastuktur jaringan komunikasi suatu daerah tertentu seperti *Base Transceiver Station*, *access point*, *dan router*. Kerusakan insfratuktur komunikasi ini dapat menyebabkan suatu daerah terisolir dalam hal berkomunikasi sehingga sulit untuk melakukan evakuasi. Berdasarkan situasi tersebut maka diperlukan suatu penanggulangan yang memungkinkan adanya jaringan komunikasi tanpa memerlukan infrastuktur tetap. Kehadiran MANET (*Mobile Ad hoc Network*) menjadi jawaban untuk komunikasi yang belum mempunyai infrastruktur yang tetap. MANET merupakan jaringan yang tidak memiliki infrastruktur yang tetap dan *node-node* yang berada di dalamnya berfungsi sebagai router untuk meneruskan informasi yang dikirimkan. (Harahap, 2014).

MANET memiliki beberapa *routing protocol, routing protocol* pada MANET berfungsi untuk menentukan *path/*jalur terbaik diantara dua *node* yang bergerak yaitu *node* sumber dan *node* tujuan. Sehingga data yanga dikirimkan dapat sampai tepat waktu. Protokol *routing* memiliki informasi tentang *node* yang terkoneksi dan *node* tetangganya. Saat sebuah *node* sumber ingin melakukan hubungan denga *node* tujuan maka *node* sumber akan melakukan *broadcast status* pada *node* tetangganya (Jayanti, 2014). Beberapa *routing protocol* yang terdapat pada MANET adalah *routing protocol reactive*, *proactive,* dan *hybrid.* AODV (*Ad Hoc On-Demand Distance Vector*) dan AOMDV (*Ad Hoc On-Demand Multipath Distance Vector*) adalah protokol *reactive,* protokol *reactive* adalah protokol yang melakukan pencarian path routing hanya ketika dibutuhkan yaitu saat node-node akan saling berkomunikasi. Reaktif *routing* lebih efisien daripada *proactive routing* dalam hal mencari dan mempertahankan rute antara node yang akan berkomunikasi satu sama lain (Paulose & Paulose, 2016).

Keamanan pada MANET adalah sebuah tantangan yang besar, karena node yang mengarahkan data pada *node* lain akan saling terhubung pada sebuah jaringan nirkabel yang tidak terproteksi. Sebuah *node* pada jaringan dapat masuk dan keluar tanpa adanya izin. Oleh karena itu MANET sangat rentan akan gangguan dari segi keamanan yang disebabkan oleh *node* yang dengan atau tidak sengaja memasuki jaringan (Kumar, et al., 2017).

Serangan aktif yang dapat ditemui pada jaringan MANET adalah DDoS *(Distributed Denial of Serice)* dan *Blackhole.* Serangan DDoS ini akan menbanjiri jaringan yang ada dengan banyak paket *routing* yang berlangsung secara terus menerus dengan jumlah *node* penyerang yang banyak dan periode waktu yang lama, sehingga dapat membuat kualitas jaringan menurun bahkan *down* (Ramadhan, 2017). Blackhole dilakukan dengan *node* penyerang memberikan

informasi *routing* palsu sehingga seolah-olah *node* penyerang tersebut adalah rute terbaik menuju tujuan. Saat data diterima oleh *node* penyerang, maka data akan di *drop* (Kumar, et al., 2017).

Penelitian sebelumnya mengenai serangan *Blackhole* dilakukan oleh H. A. Esmaili (2011) dengan judul penelitian *Performance Analysis of* AODV *under Black Hole Attack through Use of OPNET Simulator.* Penelitian ini membahas bagaimana kinerja protokol AODV saat mengalami serangan *Blackhole* yang diimplementasikan dengan menggunakan simulator OPNET serta menggunakan 48 *nodes.* Penelitian lainnya juga dilakukan oleh Ritika Sharma (2015) dengan judul *Comparison of* AOMDV *With and Without Black Hole Attack,* yang membahas perbandingan kinerja protokol AOMDV saat tidak ada serangan *Blackhole* dan saat ada serangan *Blackhole*, penelitian ini menggunakan *software* simulasi MATLAB. Sedangkan penelitian mengenai serangan DDoS dilakukan oleh Ananda Ari Ramadhan (2017) dengan judul Analisis Perbandingan Pengujian *Distributed Denial of Service (DDoS)* dan *Rushing Attack* pada Jaringan UDP dengan Routing AODV. Penelitian ini meneliti bagaimana kinerja protokol AODV terhadap serangan *Denial of Service (DDoS)* dan *Rushing Attack* menggunakan *nodes* berjumlah 30. Parameter yang diuji pada penelitian ini diantaranya adalah *Delay, Packet Delivery Ratio,* dan *Routing Overhead*.

Seperti yang telah diketahui bahwa protokol AODV dan AOMDV adalah dua dari beberapa protokol yang sering digunakan dalam jaringan MANET. Penelitian ini bertujuan untuk menganalisis kinerja kedua protokol agar dapat mengetahui protokol manakah yang memiliki kinerja paling optimal dan protokol manakah yang paling rentan saat kedua protokol diberikan serangan berupa *Blackhole* pada skenario serangan pertama dan *DDoS* pada skenario serangan kedua, sehingga protokol yang memiliki nilai paling optimal dapat dipilih untuk menjadi protokol pada jaringan MANET yang dibuat.

Penelitian ini akan membandingkan bagaimana kinerja protokol AODV dan AOMDV sebelum dan sesudah adanya penyerangan *DDoS* dan *Blackhole* dengan menguji setiap skenario serangan secara terpisah. Penelitian dilakukan dengan mengukur QoS (*Quality of Service*) berupa PDR (*Packet Delay Ratio*), rata-rata *delay,* dan *normalized routing load* dengan menggunakan *Network Simulator-2*. Protokol AODV dan AOMDV dipilih pada penelitian ini dikarenakan pada jaringan MANET implementasi kedua protokol ini dengan serangan DDoS dan *Blackhole* masih sedikit sehingga perlu dilakukan penelitian lebih lanjut. Parameter QoS berupa *Packet Delivery Ratio, End-to-End Delay,* dan *Normalized Routing Load* dipilih untuk melihat kinerja dan kekuatan jaringan dengan mengimplementasikan kedua protokol AODV dan AOMDV.

1.2 Rumusan masalah

Bedasarkan latar belakang telah yang dijabarkan, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana nilai kinerja protokol AODV dan AOMDV berdasarkan hasil QoS (*Quality of Service)* berupa *packet delivery ratio*, rata-rata *delay*, *normalized routing load* pada jaringan MANET sebelum dan sesudah dilakukan serangan aktif?

1.3 Tujuan

Dari rumusan masalah yang sudah ditetapkan, maka dapat ditentukan tujuan skripsi ini adalah:

1. Menerapkan aturan protokol AODV dan AOMDV pada jaringan MANET
2. Menganalisis nilai kinerja jaringan MANET dengan protokol AODV dan AOMDV sebelum adanya serangan aktif dan sesudah serangan
3. Menganalisis hasil QoS (*Quality of Service)* berupa *packet delivery ratio*, rata-rata *delay*, *normalized routing load* pada jaringan MANET dengan mengimplementasikan protokol AODV dan AOMDV pada serangan DDoS dan *Blackhole*

1.4 Manfaat

Manfaat dari penelitian ini adalah melakukan analisis protokol *routing* AODV dan AOMDV dengan skema penyerangan menggunakan *DDoS* dan *BlackHole*, penelitian ini bermanfaat untuk mengetahui kinerja protokol AODV dan AOMDV sebelum dan sesudah serangan.

1.5 Batasan masalah

1. Simulasi dilakukan dengan menggunakan *Network Simulator* 2.
2. Tidak membahas mengenai konsumsi energi pada MANET
3. Parameter pengujian yang digunakan *delivery ratio*, rata-rata *delay*, dan *normalized routing load*

1.6 Sistematika pembahasan

Sistematika pembahasan dari penyusunan penelitian yang direncanakan adalah sebagai berikut

BAB 1 PENDAHULUAN

Bab ini mencakup latar belakang, rumusan masalah, tujuan yang ingin dicapai, manfaat yang dapat diperoleh, batasan masalah, dan sistematika penulisan dari penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian tentang penelitan sebelumnya yang relevan dan memaparkan teori-teori dan konsep-konsep yang didapat dari sumber-sumber yang relevan untuk digunakan sebagai panduan dalam penelitian serta penyusunan laporan.

BAB 3 METODOLOGI

Bab ini membahas metodologi yang digunakan pada penelitian agar penelitian dapat berjalan secara terstuktur. Di sini berisi pemaparan langkah kerja yang terdiri atas studi literatur, analisis kebutuhan, perancangan sistem, pengujian, teknik pengambilan kesimpulan dan saran, dan analisis.

BAB 4 PERANCANGAN DAN IMPLEMENTASI PENGUJIAN

Bab ini berisi tentang tahap-tahap perancangan, simulasi, dan pengujian implementasi serangan DDoS dan *Blackhole* dalam protokol AODV dan AOMDV.

BAB 5 HASIL DAN ANALISIS

Bab ini berisi hasil dan analisis perbandingan terhadap protokol AODV dan AOMDV terhadap serangan aktif berupa DDoS dan Blackhole.

BAB 6 KESIMPULAN DAN SARAN

Bab ini mengemukakan kesimpulan yang diambil dari hasil penelitian dan perancangan sistem, serta saran-saran untuk pengembangan selanjutnya, agar dapat dilakukan perbaikan-perbaikan di masa yang akan datang

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian ini dikaji dengan penelitian-penelitian yang telah dilakukan sebelumnya yang akan dijadikan pedoman dalam penelitian ini. Berikut adalah kajian pustaka yang dijabarkan pada tabel 2.1

Tabel 2.1 Kajian Pustaka

No	Judul dan Tahun	Penulis	Hasil Penelitian	Penelitian Penulis
1.	Analisis Perbandingan Pengujian *Distributed Denial of Service* (DDoS) dan *Rushing Attack* pada Jaringan UDP dengan Routing AODV [2017]	Ananda Ari Ramadhan, Aji Gautama Putrada, Maman Abdurrohman	Mengimplementasikan serangan Rushing Attack dan DDoS pada protokol AODV dengan pengujian node berjumlah 30 node	Mengimplementasikan serangan DDoS dan BlackHole pada protokol AODV dan AOMDV dengan pengujian node berjumlah 20,50, 100 node
2.	Evaluasi Kinerja Protokol AOMDV Terhadap Serangan *Malicious Node* dan DDoS pada MANET dengan Menggunakan *Network Simulator* 2 (NS-2) [2015]	Mellia Aisyah Aristyorini, Sukiswo, dan Ajub Ajulian Zahra	Mengimplementasikan serangan malicious node dan DDoS pada protokol AOMDV	Mengimplementasikan serangan DDoS dan BlackHole pada protokol AODV dan AOMDV
3.	*Performance Analysis of AODV under Black Hole Attack through Use of OPNET Simulator*[2011]	H. A. Esmaili, M. R. Khalili Shoja, dan Hossein gharaee	Implementasi menggunakan software simulasi OPNET dan 46 nodes	Menggunakan software Network Simluator 2 dan protokol yang diuji adalah AODV dan AOMDV

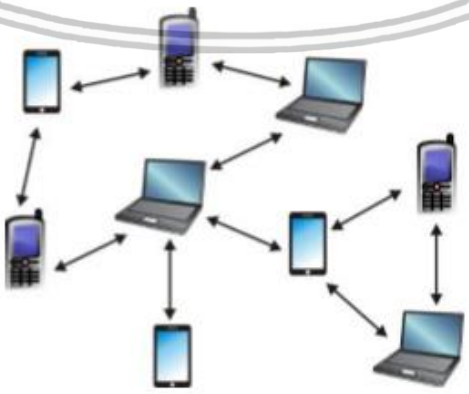
4.	*Comparison of AOMDV With and Without Black Hole Attack* [2015]	Ritika Sharma, Bhawna Singla	Implementasi menggunakan software MATLAB	Implmentasi menggunakan software Network Simluator 2

2.2 Dasar Teori

2.2.1 Mobile Ad-Hoc Network (MANET)

Mobile Ad-Hoc Network (MANET) adalah sebuah jaringan yang terdiri dari kumpulan *mobile node* yang dapat bergerak secara bebas dan dinamis sehingga dapat membentuk suatu jaringan dan tidak membutuhkan infrastruktur tetap. MANET dapat mengorganisasi dan mengonfigurasi *node-node* nya sendiri pada jaringan nirkabel yang bersifat *multihop*, dimana struktur topologi dapat berubah secara dinamis. *Node* pada jaringan MANET berfungsi sebagai *host* dan *router* sehingga *node* itu sendiri dapat mengirimkan data dari dan ke *node* lain (Abolhasan, et al., 2003).

Tanpa adanya infrastruktur tetap pada MANET perpindahan data dari *node* sumber ke *node* tujuan seringkali tidak dapat dijangkau. Sebuah *routing protocol* diperlukan untuk mengetahui *path*/jalur yang optimal antara *node* sumber dan *node* tujuan. Sehingga koneksi komunikasi didalam jaringan tetap dapat dipertahankan meskipun topologi didalam jaringan yang bersifat dinamis dan dapat berubah sewaktu-waktu karena terdapat *node* yang bertambah ataupun berkurang. Jaringan MANET bersifat desentralisasi, yaitu organinsasi jaringan dan pengiriman pesan yang terjadi didalamnya harus dijalankan oleh *node-node* itu sendiri. MANET memiliki tiga jenis *routing protocol* yaitu *reactive*, *proactive*, dan *hybrid* (Gorantala, 2006). Gambar 2.1 adalah gambaran MANET yang merupakan sekumpulan autonomus yang terdiri dari elemen *mobile* seperti *smartphone, tablet,* dan laptop.



Gambar 2.1 Jaringan MANET

Sumber : Studi Kinerja Multipath AODV dengan Menggunakan Network simulator 2 (NS-2)

MANET memiliki beberapa karakteristik yaitu :

1. Topologi yang dinamis : Seluruh *node* yang terdapat didalam jaringan ini memiliki sifat *mobile* yang dapat berpindah-pindah sewaktu-waktu, dan topologi yang bersifat *multihop* dapat berubah sewaktu-waktu saat terdapat perubahan pada pergerakan *node*.
2. *Bandwidth* yang terbatas: Link/jalur pada jaringan nirkabel (*wireless*) memiliki kapasitas yang lebih rendah/kecil dibandingkan jika jaringan menggunakan kabel sebagai media pengirimannya. *Bandwidth* yang terbatas ini adalah salahsatu masalah didalam MANET yang dapat menyebabkan *congestion* (kemacetan) jika jaringan terlalu padat.
3. Energi yang terbatas : Karena MANET merupakan jaringan yang terdiri dari *node mobile*, maka *node* ini dipastikan membutuhkan energi seperti baterai untuk dapat bekerja. Sehingga optimasi energi pada MANET sangatlah penting.
4. Keamanan yang terbatas : Jaringan nirkabel (*wireless*) dapat dipastikan memiliki keaman yang rentan oleh serangan yang dapat terjadi. Beberapa faktor ancaman yang seringkali terjadi pada jaringan MANET adalah DDoS dan *Blackhole* (Putra & Anggoro, 2016).

2.2.2 Protokol *Routing* Reaktif

Protokol *routing* reaktif memiliki mekanisme proses pencarian dan penemuan rute *routing* hanya akan berjalan saat terdapat permintaan (*on-demand*). Proses pencarian rute pada protokol ini adalah dengan menggunakan Teknik *flooding* yang digunakan untuk melakukan proses *route discovery*. Keuntungan dari jenis protokol ini adalah *bandwidth* yang dibutuhkan cukup kecil yang cocok untuk jaringan *Ad-hoc.* (Purba, 2018)

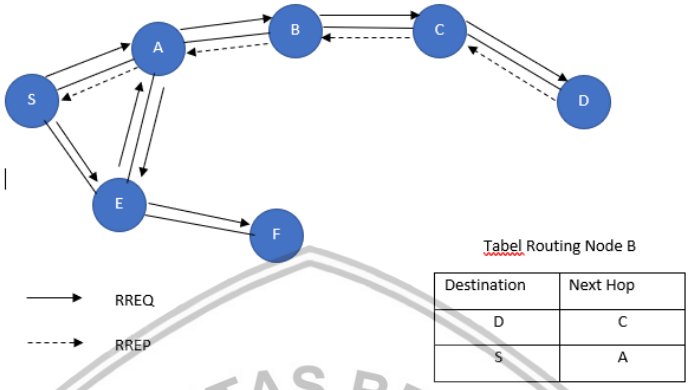
2.2.2.1 *AODV (Ad Hoc On-Demand Distance Vector)*

AODV adalah sebuah *routing protocol reactive* yang berfungsi untuk mencari *path/*jalur yang terbaik dari *node* sumber ke *node* tujuan dan protokol ini menggunakan 3 tipe pesan yaitu, *Route Request (RREQ), Route Reply (RREP),* dan *Route Error (RERR)* (Esmaili, et al., 2011). Terdapat dua proses utama pada AODV yaitu: Route Discovery dan Route Maintenance.

A. Route Discovery

Saat *node* sumber membutuhkan sebuah rute untuk mengirimkan data kepada *node* tujuan, maka hal pertama yang dilakukan adalah menginisiasi proses *route discovery*. *Node* sumber menyebarkan dengan cara melakukan *broadcast* berupa *Route Request (RREQ)* yang berisi *node* mana yang akan menjadi tujuan dan kemudian menunggu *Route Reply (RREP).* Saat sebuah *node* mendapatkan RREQ maka *node* tersebut akan membuat *reverse path* kepada sumber *node* melalui *hop* yang sebelumnya begitu juga *reverse path* pada *hop* selanjutnya. Jika rute ke *node* tujuan *valid* maka *node* akan menigrimkan sebuah RREP, jika tidak maka RREQ

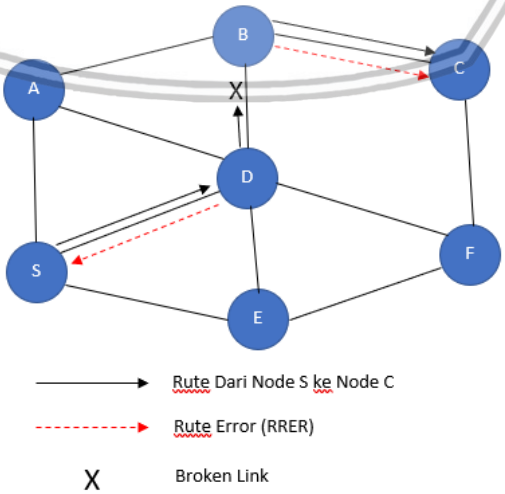
akan di *broadcast* kembali. RREQ yang terduplikasi pada sebuah *node* akan dibuang. Saat *node* tujuan mendapatkan sebuah RREQ, maka *node* sumber juga akan membuat dan mengirimkan RREP melalui *reverse path* yang telah dibuat (Purba, 2018). Gambar 2.2 merupakan gambaran proses *route discovery* pada AODV.



Gambar 2.2 Proses *Route Discovery* Pada Protokol AODV

B. *Route Maintenance*

Proses *Route Maintenance* dilakukan oleh paket *Route Error (RERR).* Saat terjadi kegagalan *Link* dideteksi, RERR akan dikirimkan kembali melalui *maintained predecessor link* kepada semua *node* yang terdeteksi memiliki *link* yang gagal. *Path/*jalur akan dihapus oleh RERR bersamaan saat RERR melewati jalur tersebut. Saat *node* sumber mendapatkan RERR, maka *node* kan melalukan kembali proses *route discovery.* Jalur yang tidak terpakai pada *routing table* akan hilang sesuai dengan waktu yang ditentukan (Marina & Das, 2001). Gambar 2.3 merupakan mekanisme *route maintenance* pada AODV.



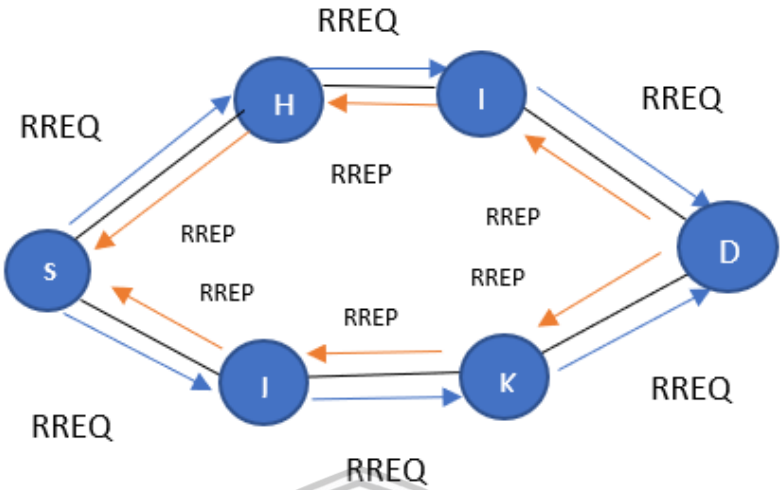
Gambar 2.3 *Route Maintenance* Pada Protokol AODV

2.2.2.2 *AOMDV (Ad Hoc On-Demand Multipath Distance Vector)*

Protokol AOMDV adalah pengembangan dari protokol *routing* AODV. Protokol AOMDV juga menggunakan konsep berbasis *vector* dengan pendekatan *hop-by-hop*. Fitur utama pada protokol AOMDV adalah *route discovery* dan *route maintenance* yang hamper mirip dengan protokol *routing* AODV, tetapi perbedaannya adalah pada protokol ini jumlah jalur/rute yang ditemukan saat proses *route discovery* lebih banyak karena protokol AOMDV yang bersifat *multipath.* Pada protokol AOMDV berupaya mengakumulasi beberapa rute (*multiple paths*) selama proses *routing discovery* berjalan. Sehingga protokol AOMDV dapat meminimalkan nilai *end-to-end delay* karena terdapat rute cadangan yang tersimpan pada table *routing* saat rute pengiriman data rusak. Proses untuk menentukan beberapa rute/jalur yang ditemukan harus bersifat *multiple loop free,* yaitu harus tidak terdapat *loop* pada setiap jalur dan harus saling lepas (*disjoint*). (Putra & Anggoro, 2016)

A. Route Discovery

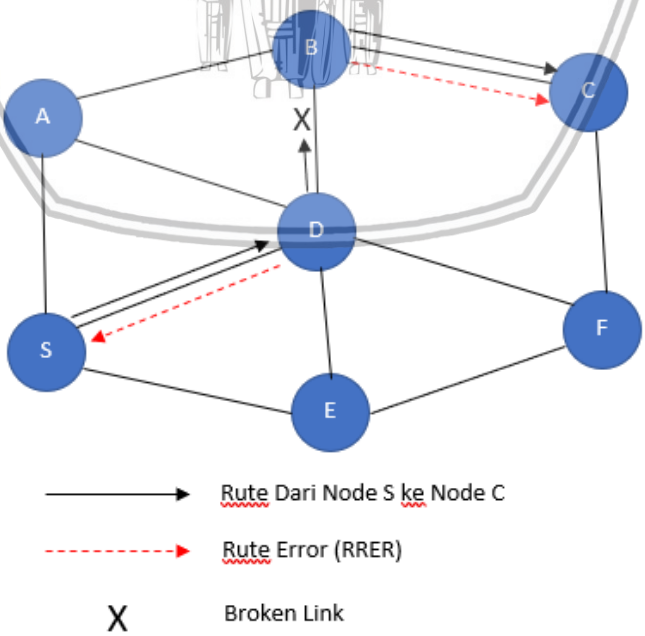
Proses *Route Discovery* protokol AOMDV hampir sama dengan proses *route discovery* pada protokol AODV, tetapi perbedaannya adalah pada jumlah rute yang ditemukan pada protokol AOMDV lebih banyak dari pada protokol AODV. *Node* sumber menyebarkan dengan cara melakukan *broadcast* berupa *Route Request (RREQ)* yang berisi *node* mana yang akan menjadi tujuan dan kemudian menunggu *Route Reply (RREP).* Saat sebuah *node intermediate* mendapatkan RREQ maka *node* tersebut akan membuat *reverse path* kepada sumber *node* melalui *hop* yang sebelumnya begitu juga *reverse path* pada *hop* selanjutnya. Kemudian *node intermediate* akan melakukan cek apakah ada beberapa *forward paths* menuju *node* tujuan ataukah tidak. Jika hasil cek tersebut valid, maka *node* ini akan membuat paket *routing route reply* dan mengirimkan paket routing tersebut melalui *reverse path* menuju ke *node* sumber. Tetapi jika tidak ditemukan beberapa *forward paths,* maka *node intermediate* akan langsung meneruskan paket *routing* RREQ menuju *node* tujuan yang kemudian *node* tujuan akan membalas dengan mengirimkan paket *routing* RREP menuju *node* sumber. Pada protokol AODV, saat *node* tujuan mendapatkan RREQ yang terduplikasi, maka paket *routing* RREQ yang terakhir diterima *node* sumber akan di buang/*drop.* Tetapi pada AOMDV, *node* tujuan akan menerima semua paket *routing* RREQ baik itu yang terduplikasi ataupun tidak yang ditandai dengan *sequence number* yang berbeda, dan akan proses ini digunakan oleh protokol AOMDV untuk membentuk beberapa rute cadangan dari *node* sumber ke *node* tujuan. Sehingga protokol AOMDV akan memiliki lebih dari satu jalur/*path (multipath)* dari *node* sumber menuju *node* tujuan*,* hal ini dimaksudkan jika terjadi kerusakan pada jalur/rute pada saat *node* didalam jaringan berpindah-pindah maka terdapat *node-node* tersebur telah memiliki informasi mengenai jalur alternatif yang tersedia, sehingga proses *delay* dapat diminimalisir. (Putra & Anggoro, 2016). Gambar 2.4 merupakan proses *route discovery* pada protokol AOMDV.



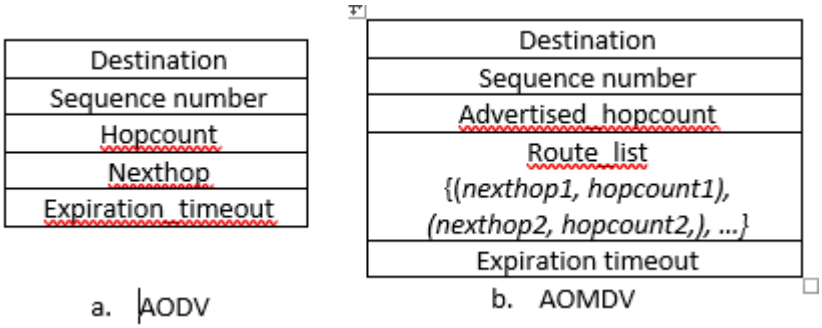
Gambar 2.4 *Route Discovery* Protokol AOMDV

B. *Route Maintenance*

Proses *Route Maintenance* pada protokol AOMDV adalah pengembangan sederhana dari protokol AODV. Protokol AOMDV juga menggunaka paket *Route Error* (RRER) untuk mengirimkan pesan *error* apabila terdapat rute yang rusak, tetapi karena protokol AOMDV memiliki banyak rute cadangan yang ditemukan pada proses *route discovery* maka *node* akan langsung memilih rute alternative jika terdapat rute menuju *node* tujuan yang rusak. (Putra & Anggoro, 2016). Gambar 2.5 adalah proses *route maintenance* protokol AOMDV.



Gambar 2.5 *Route Maintenance* Protokol AOMDV

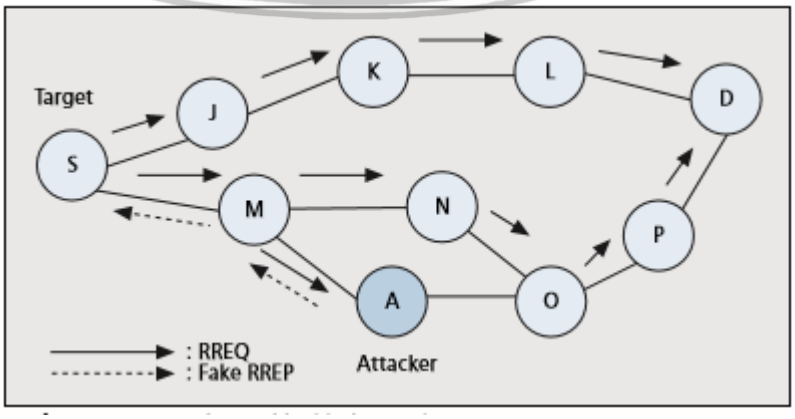


Gambar 2.6 Struktur *routing table* pada AODV dan AOMDV

Gambar 2.6 menunjukkan struktur *routing table* pada AODV dan AOMDV. Pada AOMDV, *advertised_hopcount* menggantikan *hopcount* pada AODV. Selain itu *route_list* menggantikan *nexthop*, dan mendefinisikan *multiple next hop* dengan *nexthop* yang diketahui. *Advertised_hopcount* di inisialisasi setiap kali *sequence number* diperbarui. *Advertised hopcount* berisi informasi maksimal *hop* yang terdapat pada setiap jalur menuju destinasi. (Marina & Das, 2001).

2.2.3 *Blackhole*

Serangan *blackhole* memiliki dampak yang besar dalam pengiriman paket yang diantar oleh setiap *node*, serangan ini dilakukan oleh *node* yang jahat dengan mengirimkan informasi *routing* palsu kepada jaringan yang ada, sehingga paket yang dikirimkan oleh *node* sumber akan diterima oleh *node* jahat tersebut yang kemudian paket akan di buang/*drop.* Serangan *Blackhole* memiliki beberapa jenis *Blackhole* adalah *internal blackhole* dan *external blackhole. Internal Blackhole* adalah jenis serangan *blackhole* yang telah berada pada jaringan MANET diantara *node* sumber dan *node* tujuan, serta akan mencari kesempatan dimana *node blackhole* dapat menyerang. Sedangkan *External Blackhole* adalah jenis serangan *blackhole* dimana *node* penyerang berada diluar jaringan dan akan mencari cara untuk masuk kedalam jaringan sehingga akan menimbulkan kemacetan dan mengganggu seluruh jaringan. (Dorri, et al., 2015).



Gambar 2.7 *Blackhole attack* pada MANET

Sumber: A Survey of Routing Attacks In Mobie Ad Hoc Networks

Pada Gambar 2.7 dijelaskan bagaimana *blackhole* bekerja. Penyerang mengirimkan informasi RREP palsu kepada *node* sumber S, mengaku bahwa *node* penyerang memiliki jalur terbaik dibandingkan dengan *node* lain. Karena penyerang memberikan informasi bahwa *sequence number* lebih besar dibandingkan dengan *node* lain maka paket akan melewati *node* A penyerang (Kannhavong, et al., 2007).

2.2.4 DDoS (*Distributed Denial of Service)*

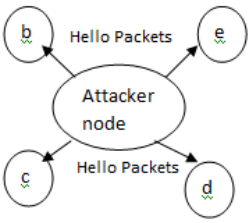
DDoS adalah kata lain dari DoS (*Denial of Service*), tetapi DDoS dilakukan secara *massive* atau banyak *node* penyerang. Cara kerja DoS adalah penyerang memberikan banyak *request* didalam jaringan sehingga *node* yang akan bertukar data tidak dapat mengakses jaringan data atau servis. Serangan ini dilakukan untuk membuang-buang *resources,* sehingga *packet delay* dan *congestion* meningkat (Dorri, et al., 2015).

Serangan DDoS secara sederhana juga melakukan apa yang dilakukan oleh DoS, tetapi pada DDoS penyerang menggunakan banyak *node* yang disebut *zombie*. Para *zombie* akan menyerang secara bersamaan sehingga *resources* yang terbuang akan semakin besar dan dapat mematikan seluruh jaringan (Moudni, et al., 2016). Contoh dari serangan DDoS adalah *Flooding Attack.*

2.2.4.1 *Flooding Attack*

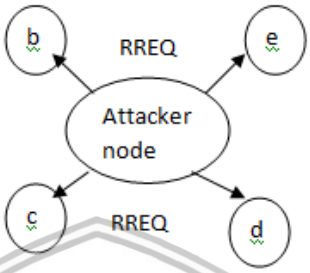
Flooding Attack dimulai dengan membanjiri jaringan dengan paket RREQ atau paket data palsu yang berfungsi untuk memblokir jaringan dan meminimalkan keberhasilan transmisi paket data yang sebenarnya pada *node* tujuan. Pada *Flooding Attack* terdapat tiga cara untuk melakukan *flooding* yaitu dengan *HELLO Flooding, RREQ Flooding, Data Flooding*.

HELLO Flooding : Beberapa protokol *routing* pada jaringan *wireless* memerlukan *nodes* untuk melakukan *broadcast hello messages* untuk mengidentifikasikan *node* itu pada *node* tetangganya. *Node* yang menerima paket *hello* akan berasumsi bahwa *node* itu berada pada jarak jangkauannya. Beberapa *node* yang bertugas untuk melakukan penyerangan akan membanjiri jaringan dengan paket *hello* secara terus menerus tanpa memerdulikan interval paket *hello* yang semestinya sehingga hal ini dapat memengaruhi jaringan. Gambar 2.8 merupakan gambar *HELLO Flooding Attack.*



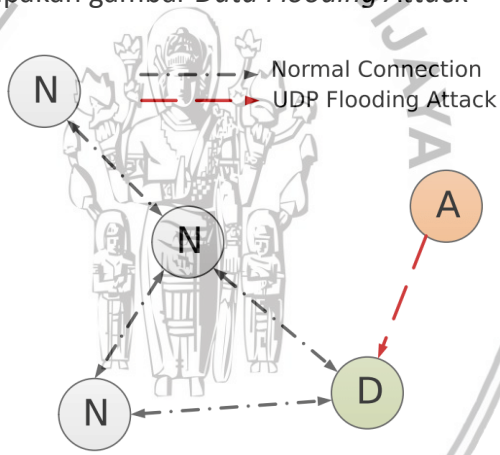
Gambar 2.8 *HELLO Flooding Attack*

RREQ Flooding : Pada serangan jenis ini penyerang membanjiri jaringan dengan melakukan *broadcast* RREQ untuk *node* yang ada ataupun tidak ada pada jaringan tersebut. Untuk melakukan serangan ini, penyerang akan melakukan serangan terus-menerus dengan interval yang sangat kecil sehingga konsumsi *bandwidth* akan meningkat dan transimisi paket data akan terganggu. Gambar 2.9 merupakan gambar *RREQ Flooding Attack.*



Gambar 2.9 *RREQ Flooding Attack*

Data Flooding : Pada serangan ini *node* penyerang membanjiri jaringan dengan paket *data* palsu sehingga *resource* jaringan akan menurun (Meher & Ladhe, 2014). Gambar 2.10 merupakan gambar *Data Flooding Attack*

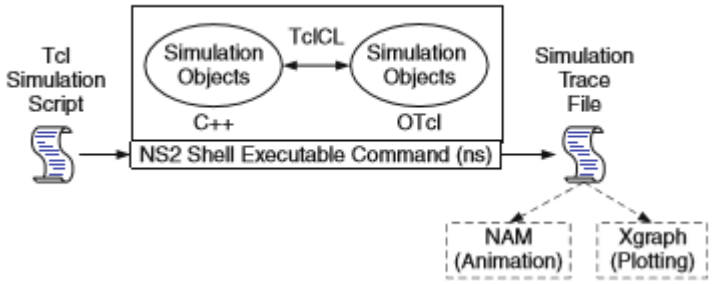


Gambar 2.10 *Data Flooding Attack*

2.2.5 *Network Simulator* 2

Network Simulator 2 (NS2) adalah sebuah simulator *event driven* yang berguna untuk pembelajaran jaringan MANET, yaitu jaringan komunikasi yang dinamis. NS2 dapat digunakan untuk jaringan kabel taupun nirkabel beserta protokolnya (algoritma *routing,* TCP, dan UDP). (Issariyakul, 2012).

NS2 terdiri dari 2 bahasa utama, yaitu C++ dan *Object-oriented Tool Command Language* (Otcl). Bahasa C++ digunakan untuk mendefinisikan mekanisme *internal* dari simulasi (*Backend*), sementara Octl berfungsi untuk mengatur proses simulasi dengan cara membangun dan mengkonfigurasi setiap objek pada simulasi dengan baik (*Frontend*).



Gambar 2.11 Arsitektur NS2

Sumber: Introduction to Network Simulator 2 (NS2)

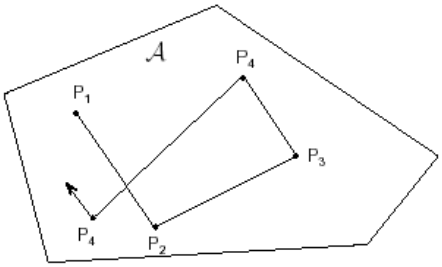
Gambar 2.11 menjelaskan mengenai arsitektur dasar dari NS2, yaitu berawal dari *script TCL,* yang akan diproses dalam simulasi sehingga menghasilkan *file trace. File trace* berisi tentang NAM yang berupa animasi yang merupakan hasil dari proses simulasi yang telah dilakukan. Pada tabel 2.2 adalah *format* dalam setiap baris yang ada pada *file trace.*

Tabel 2.2 Format pada File Trace

Event	Time	Node sumber	Node Tujuan	Nama Paket	Ukuran paket	Flags	Flow ID	Alamat Sumber	Alamat tujuan	Sequence number	ID Packet

2.2.6 *Random Way Point*

Random Way Point adalah sebuah model pergerakan acak untuk setiap *node* yang berada pada jaringan dan bagaimana lokasi, kecepatan *node* dapat berubah sewaktu-waktu dalam area simulasi. Setiap pergerakan *node* tidak terikat satu sama lain/bebas. Proses pergerakan dari mode *Random Way Point* ini adalah *node* akan diberikan tujuan, kecepatan, dan arah secara acak. Setelah *node* tersebut mencapai tujuan pertama, maka *node* tersebut akan berhenti sebentar sebelum akhirnya bergerak kembali menuju ke lokasi tujuan selanjutnya. Waktu untuk *node* berhenti sebentar ini dinamakan *pause time.* Pendefinisian berapa kecepatan dan lama *pause time* adalah penting dalam menggunakan jenis model ini, karena semakin kecil kecepatan dan semakin lama *pause time* akan menghasilkan topologi yang lebih stabil (Pandey, 2014). Gambar 2.12 merupakan gambaran *random way point.*



Gambar 2.12 *Random Way Point*

2.2.7 *Packet Delivery Ratio*

Rumus dari *packet delivery ratio* (PDR) adalah jumlah semua paket data yang diterima oleh *node* tujuan kemudian dibagi jumlah semua paket data yang dikirimkan oleh *node* sumber (Harahap, 2014). PDR berfungsi untuk mengetahui performa jaringan. Rumus untuk mencari *packet delivery ratio* adalah:

$$\textit{Pakcet Delivery Ratio} = \frac{Jumlah\ seluruh\ paket\ yang\ diterima}{jumlah\ seluruh\ paket\ yang\ dikirim}\ X\ 100\% \quad (2.1)$$

2.2.8 End-to-End Delay

End-to-end delay adalah rata-rata waktu pengiriman paket data yang berhasil terkirim dibagi dengan jumlah seluruh paket data yang terkirim (Harahap, 2014). Perhitungan *end-to-end delay* hanya menghitung paket data yang benar-benar sampai pada *node* tujuan. Rumus untuk mencari *end-to-end delay* adalah:

$$\textit{End to End Delay} = \frac{Jumlah\ total\ waktu\ pengiriman\ paket\ yang\ berhasil\ terkirim}{jumlah\ seluruh\ paket\ yang\ terkirim} \quad (2.2)$$

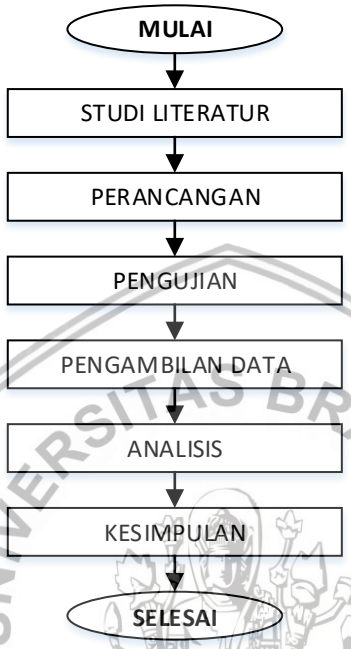
2.2.9 *Normalized Routing Load*

Normalized routing load (NRL) adalah nilai perbandingan antara seluruh paket *routing* (RREQ, RREP, dan RRER) dengan seluruh aket data yang ditermia oleh *node* tujuan. *Normalized routing load* diukur untuk mengetahui tingkat efisiensi dari protokol yang diuji, semakin tinggi nilai NRL maka semakin kurang efisien juga protokol tersebut (Harahap, 2014). Rumus untuk mencari *normalized routing load* adalah:

$$\textit{Normalized Routing Load} = \frac{Jumlah\ total\ paket\ routing\ (RREQ,RREP,RRER)}{jumlah\ seluruh\ paket\ data\ yang\ diterima} \quad (2.3)$$

BAB 3 METODOLOGI

Pada bagian ini menjelaskan metode yang digunakan dalam pengembangan penelitian ini. Gambar 3.1 merupakan tahapan-tahapan metodologi penelitian untuk penelitian ini.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literatur digunakan sebagai dasar dan landasan terhadap seluruh perancangan dan implementasi dair penelitian ini. Literatur yang digunakan adalah:

1. *Mobile Ad-Hoc Network* (MANET)
2. AODV
3. AOMDV
4. *Black Hole*
5. DDoS (*Distributed Denial of Service*)
6. *Network Simulator*-2

3.2 Kebutuhan Fungsional

Kebutuhan fungsional dibutuhkan untuk mengetahui apa saja yang akan dilakukan oleh sistem yang akan dibuat dan berisi informasi mengenai apa yang ada dan dapat dihasilkan sistem. Kebutuhan fungsional pada pengujian ini adalah:

1. Sistem dapat mengimplementasikan protokol routing AODV dan AOMDV.

2. Sistem dapat mengatur pergerakan dan letak *node* selama simulasi.
3. Sistem dapat menerapkan mekanisme serangan berupa *Blackhole* dan DDoS
4. Sistem dapat meletakkan *node* agen sebagai penyerang.
5. Sistem dapat memberikan rincian hasil simulasi berupa data selama proses dilakukan.

3.3 Kebutuhan Non-Fungsional

Kebutuhan Non-fungsional adalah kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini. Berikut kebutuhan non-fungsional sistem ini, yaitu:

1. Perangkat keras:
   a. 1 buah komputer/laptop
2. Perangkat lunak:
   a. Ubuntu linux 14.04
   b. Network Simulator-2 (Software Simulasi)

3.4 Perancangan dan Implementasi Skenario Pengujian

Perncangan dan implementasi skenario pengujian merupakan tahapan penting yang dilakukan selama pengujian yang akan dilakaukan pada *Network Simulator 2.* Pada tahap ini akan dijelaskan mengenai detail perancangan sistem, parameter dari skenario yang akan dibuat dan kemudian akan dijelaskan mengenai langkah-langkah implementasi skenario yang telah dirancang pada *Setwork Simulator 2*

3.4.1 Perancangan Skenario Pengujian

Dalam perncangan skenario pengujian akan dijabarkan bagaimana penentuan parameter-parameter yang akan digunakan seperti protokol apa yang digunakan, luas area simulasi, jumlah kepadatan *node* pada area simulasi, kecepatan *node*, model pergerakan *node*, tipe koneksi, data *rate*, besar paket data, jumlah *node* serangan *blackhole* dan DDoS.

3.4.2 Implementasi Skenario Pengujian

Setelah dilakukan proses perancangan skenario pengujian, kemudian selanjutnya adalah implementasi skenario pengujian yang dirancangan menggunakan *Network Simulator 2*. Beberapa konfigurasi dibutuhkan dalam merancang skenario pengujian ini dinataranya seperti parameter topologi jaringan, pergerakan *node*, lama waktu simulasi, waktu serangan *node* blackhole dan DDoS berlangsung, dan proses data *output.*

3.5 Pengujian

Pengujian dirancang pada jaringan MANET yang akan disimulasikan menggunakan simulator NS2 untuk mendapatkan hasil dan data yang akan di

analisis untuk pengambilan kesimpulan di akhir penelitian. Pengujian dilakukan dengan membandingkan protokol AODV dan AOMDV dengan skenario 20,50, dan 100 *node* dengan serangan *Blackhole*. Pengujian dilakukan dengan menguji 20 *node* dengan 0, 2, 4, 6, 8, dan 10 *node* blackhole. 50 *node* dengan 0, 2, 4, 6, 8, dan 10 *node* blackhole. 100 *node* dengan 0, 2, 4, 6, 8, dan 10 *node* blackhole. Pengujian selanjutnya adalah dengan menggunakan serangan DDoS dengan skenario 20, 50, dan 100 *node.* Skenario pengujian dengan serangan DDoS untuk penggunaan *node* DDoS berjumlah sama dengan skenario *Blackhole* pada tiap skenarionya.

3.6 Pengumpulan dan Pengambilan Data

Proses pengumpulan dan pengambilan data dilakukan pada saat pengujian telah selesai dilakukan, kemudian data hasil simulasi akan dimasukkan kedalam sebuah *trace file* berekstensi .tr yang memuat seluruh hasil data simulasi. Kemudian *trace file* tersebut akan dianalisis dengan menggunakan Bahasa pemrograman AWK untuk mendapatkan *Packet Delivery Ratio, End-to-End Delay,* dan *Normalized Routing Load*.

3.7 Hasil dan Analisis

Hasil dan analisis digunakan untuk menarik kesimpulan dari pengujian yang telah berlangsung. Disini juga berisi tentang penjelasan perbandingan kedua protokol yang telah diuji sehingga dapat diketahui protokol mana yang memiliki hasil lebih optimal saat dilakukan pengujian dari skenario yang telah dibuat.

3.8 Pengambilan kesimpulan dan Saran

Pengambilan kesimpulan dilakukan berdasarkan pengujian dan analisis terhadap kinerja dari implementasi protokol *routing* AODV dan AOMDV yang telah dilakukan. Berdasarkan hasil pengujian, dapat ditentukan perbandingan kedua protokol manakah yang memiliki QoS (*Packet Delivery Ratio, End-to-End Delay,* dan *Normalized Routing Load*) paling optimal.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan

Pada bab ini dijabarkan tahapan apa saja yang dibutuhkan dalam melakukan implementasi pengujian protokol AODV dan AOMDV serta serangan *DDoS* dan *blackhole* pada MANET. Implementasi penelitan ini diterapkan pada *Network Simulator 2*.

4.1.1 Perancangan Parameter Pengujian

Perancangan parameter pengujian yang akan digunakan pada simulasi dapat dilihat pada Tabel 4.1

Tabel 4.1 Perancangan Parameter Pengujian

No	Parameter	Spesifikasi
1.	Network Simulator	*Network Simulator 2*
2.	*Routing* Protocol	AODV dan AOMDV
3.	Waktu Simulasi	1000 detik
4.	Area Simulasi	1000 x 1000
5.	Jumlah *Node*	20,50,100
6.	Jumlah *node* Penyerang	2, 4, 6, 8, 10
7.	Model Pergerakan *Node*	Random Way Point
8.	Pause Time	5 detik
9.	Tipe Koneksi	UDP (CBR)
10	Data Rate	0,1 mbps
11.	Besar Paket Data	1024 bytes
12.	Keceptan node	Acak, maks 2 m/s

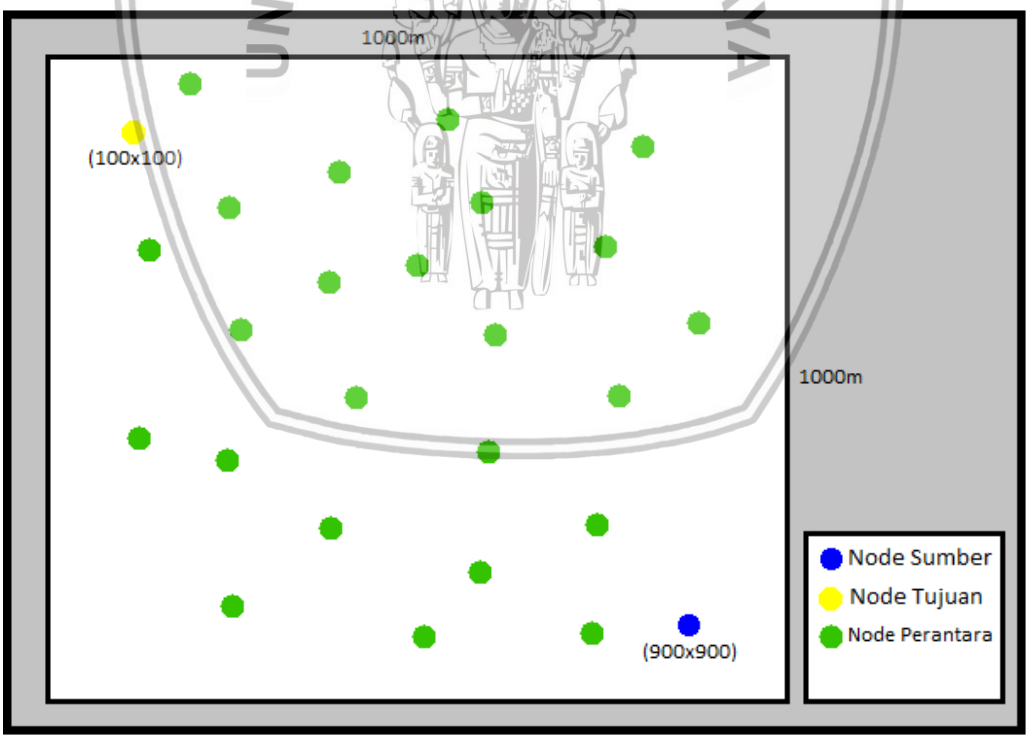
Parameter - parameter simulasi yang digunakan pada penelitian ini adalah sebagai berikut:

1. Simulator yang digunakan adalah *Netwok Simulator 2*
2. Protokol routing yang digunakan pada simulasi adalah AODV (*Ad-hoc On-Demand Distance Vector*) dan AOMDV (*Ad-hoc On-Demand Multipath Distance Vector*)
3. Waktu simulasi berlangsung selama 1000 detik
4. Ukuran wilayah simulasi yang digunakan sebesar 1000mx1000m
5. Jumlah *node* yang digunakan pada simulasi adalah 20, 50, dan 100 *node*

6. Jumlah *node* penyerang yang digunakan pada simulasi adalah 0, 2, 4, 6, 8, 10 *node blackhole* pada skenario serangan *blackhole* dan 0, 2, 4, 6, 8, 10 *node DDoS* pada skenario serangan *DDoS* jumlah *node* penyerang ini diambil berdasarkan penelitian sebelumnya.
7. Model pergerakan yang digunakan adalah Random Way Point dengan kecepatan maksimal 2 m/s dan *pause time* selama 5 detik.
8. Protokol yang digunakan untuk pengiriman paket data adalah UDP (CBR). Besar paket data yang digunakan dalam simulasi sebesar 1024 bytes. Paket data tersebut akan dikirimkan dari *node*[0] sebagai *node* sumber paket menuju *node*[1] sebagai *node* tujuan paket.
9. *Data Rate* pada simulasi ini sebesar 0,1 mbps.

4.1.2 Perancangan Topologi Jaringan

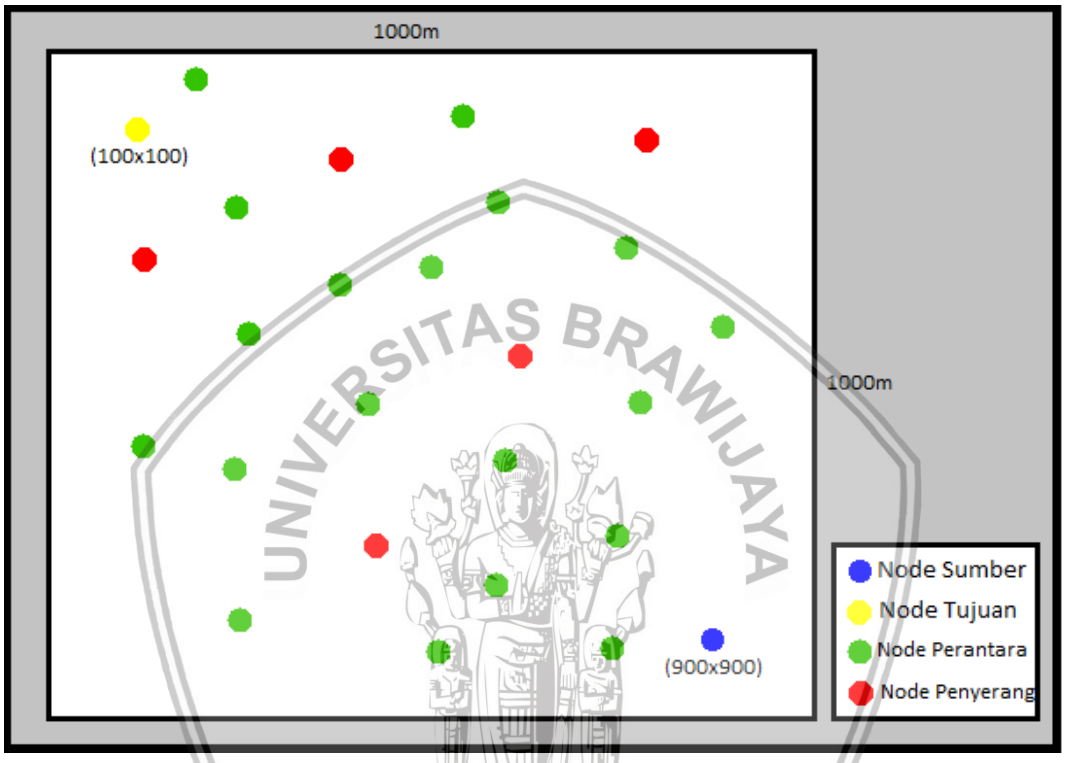
Perancangan topologi jaringan yang akan dibuat pada pengujian ini adalah perancangan topologi jaringan tanpa menggunakan serangan dan topologi dengan menggunakan serangan. Masing-masing topologi dibuat berdasarkan kepadatan *node* sebanyak 20, 50 dan 100 *node* yang diatur secara acak sesuai skenario pengujian. Perancangan ini dibuat untuk menggambarkan topologi yang akan digunakan dalam simulasi.



Gambar 4.1 Topologi Tanpa Serangan

Gambar 4.1 merupakan gambar topologi tanpa menggunakan serangan yang dibuat berdasarkan luas area simulasi sebesar 1000mx1000m. *Node* berwarna biru merupakan *node* sumber paket yang akan mengirimkan paket data pada saat simulasi berlangsung. *Node* berwarna hijau merupakan *node* perantara yang

bertugas untuk menyalurkan paket kepada *node* tujuan paket. *Node* berwarna kuning merupakan *node* tujuan paket yang terakhir dituju. Penempatan *node* sumber dan *node* tujuan diletakkan saling berjauhan sehingga protokol dapat bekerja secara optimal, berbeda jika *node* sumber dan *node* tujuan diletakkan berdekatan maka paket data akan langsung dapat diterima oleh *node* tujuan sehingga kinerja protokol tidak terlalu dapat terlihat.



Gambar 4.2 Topologi Dengan Serangan

Gambar 4.2 merupakan gambaran topologi yang digunakan pada simulasi baik itu dengan serangan *blackhole* dan *DDoS.* Topologi dibuat berdasarkan luas area simulasi sebesar 1000mx1000m. *Node* berwarna biru merupakan *node* sumber paket yang akan mengirimkan paket data pada saat simulasi berlangsung. *Node* berwarna hijau merupakan *node* perantara yang bertugas untuk menyalurkan paket kepada *node* tujuan paket. *Node* berwarna kuning merupakan *node* tujuan paket yang terakhir dituju. *Node* berwarna merah merupakan *node* penyerang yang diletakkan secara acak di dalam topologi untuk menghambat kinerja protokol yang digunakan. Penempatan *node* sumber dan *node* tujuan diletakkan saling berjauhan sehingga protokol dapat bekerja secara optimal, berbeda jika *node* sumber dan *node* tujuan diletakkan berdekatan maka paket data akan langsung dapat diterima oleh *node* tujuan sehingga kinerja protokol tidak terlalu dapat terlihat.

4.1.3 Perancangan Pergerakan *Node*

Random Way Point adalah sebuah model pergerakan acak untuk setiap *node* yang berada pada jaringan dan bagaimana lokasi, kecepatan *node* dapat berubah sewaktu-waktu dalam area simulasi. Setiap pergerakan *node* tidak terikat satu sama lain/bebas. Proses pergerakan dari mode *Random Way Point* ini adalah *node* akan diberikan tujuan, kecepatan, dan arah secara acak. Setelah *node* tersebut mencapai tujuan pertama, maka *node* tersebut akan berhenti sebentar sebelum akhirnya bergerak kembali menuju ke lokasi tujuan selanjutnya. Waktu untuk *node* berhenti sebentar ini dinamakan *pause time*. Kecepatan yang digunakan pada simulasi ini adalah maksimal 2m/s dan *pause time* sebesar 5 detik.

4.1.4 Perancangan Serangan

4.1.4.1 Perancangan Skenario Tanpa Menggunakan Serangan

Pada tahap ini akan dilakukan perancangan skenario tanpa adanya serangan pada kedua protokol yaitu AODV dan AOMDV. Pengujian dilakukan dengan satu koneksi transport jaringan UDP dengan model *traffic constant bit-rate* (CBR). Skenario pengujian dibagi menjadi 3 yaitu simulasi dengan jumlah node 20, 50, dan 100 dengan durasi waktu selama 1000 detik. Besar paket yang akan dikirimkan sebesar 1024 byte. Pada skenario pengujian paket mulai dikirimkan sejak detik 50 sampai detik 1000.

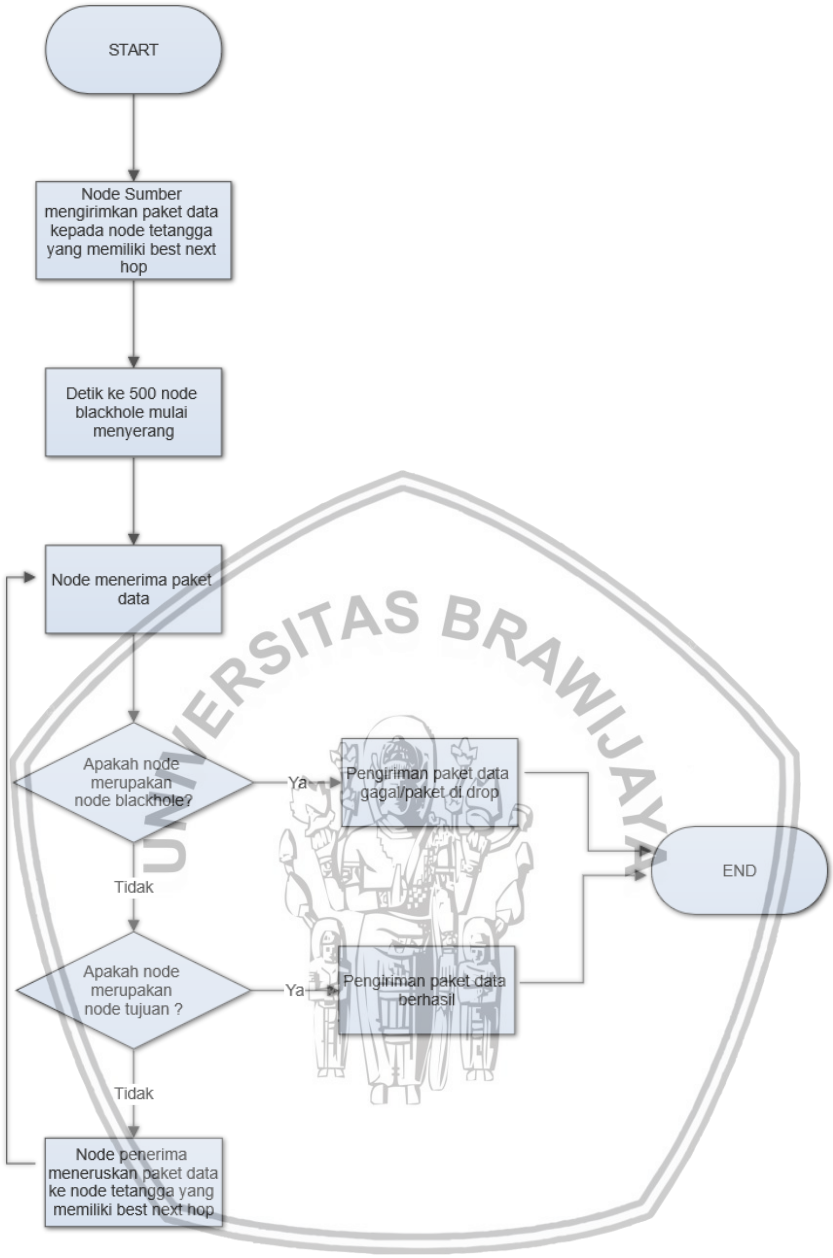
4.1.4.2 Perancangan Skenario dengan Serangan *Blackhole*

Pada simulasi ini serangan *blackhole* dilakukan dengan menambahkan *patch blackhole* pada *library* AODV dan AOMDV *Network Simulator 2*. Serangan *blackhole* yang dilakukan pada simulasi ini adalah serangan *blackhole* dengan jenis internal *blackhole attack.* Pada jenis serangan *internal blackhole attack, node* penyerang adalah *node* internal yang tidak berusaha masuk kedalam rute pengiriman paket pada jaringan. Namun jika *node* penyerang mendapatkan kesempatan dimana paket data dikirimkan melalui rute *node* penyerang, maka paket data akan di *drop* sehingga paket data tidak akan sampai pada *node* tujuan.

Pada pengujian menggunakan serangan *blackhole* skenario pengujian yang akan digunakan antara lain sebagai berikut :

1. Skenario 20 node dengan *node* penyerang sebanyak 0, 2, 4, 6, 8, 10 *node*
2. Skenario 50 node dengan *node* penyerang sebanyak 0, 2, 4, 6, 8, 10 *node*
3. Skenario 100 node dengan *node* penyerang sebanyak 0, 2, 4, 6, 8, 10 *node*

Pengujian dilakukan selama 1000 detik dengan *node* penyerang melakukan serangan pada detik ke 500 sampai 1000. Gambar 4.3 adalah *flowchart* serangan *blackhole* dilakukan:



Gambar 4.3 Flowchart Skenario Serangan Blackhole

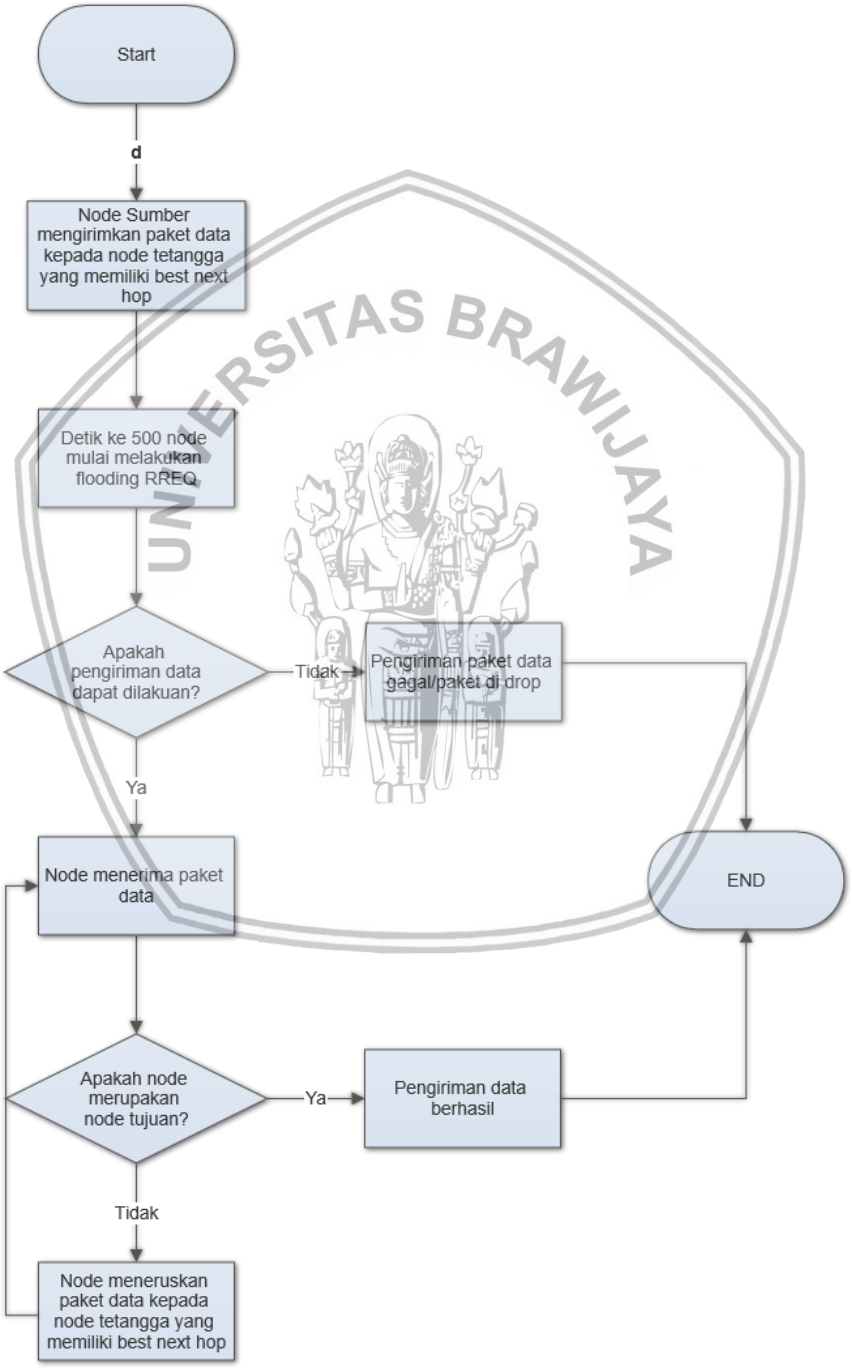
4.1.4.3 Perancangan Skenario dengan Serangan DDoS

Pada simulasi ini serangan *DDoS* dilakukan adalah serangan dengan tipe *flooding attack,* yaitu dengan membuat *node* penyerang melakukan request (RREQ) pada *node interval* waktu 0.2 detik, tujuan dari serangan ini adalah untuk mengonsumsi *bandwidth* jaringan sehingga diharapkan jaringan akan *overload* bahkan sampai *down*.

Pada pengujian menggunakan serangan DDoS skenario pengujian yang akan digunakan antara lain sebagai berikut :

1. Skenario 20 node dengan *node* penyerang sebanyak 0, 2, 4, 6, 8, 10 *node*
2. Skenario 50 node dengan *node* penyerang sebanyak 0, 2, 4, 6, 8, 10 *node*
3. Skenario 100 node dengan *node* penyerang sebanyak 0, 2, 4, 6, 8, 10 *node*

Pengujian dilakukan selama 1000 detik dengan *node* penyerang melakukan serangan pada detik ke 500 sampai 1000. Gambar 4.4 adalah *flowchart* serangan *DDoS* dilakukan:



Gambar 4.4 Flowchart Skenario Serangan DDoS

4.2 Implementasi

Pada bab ini akan dijelaskan mengenai bagaimana implementasi skenario yang telah dirancang pada subbab sebelumnya diterapkan pada *Network Simulator 2*.

4.2.1 Konfigurasi Skenario Pada *Network Simulator 2*

Dalam penelitian ini implementasi dilakukan pada perangkat lunak *Network Simulator 2* (NS2). Dalam menerapkan skenario yang telah dirancang, dibutuhkan konfigurasi untuk penerapan skenario-skenario tersebut pada NS2. Pengaturan konfigurasi untuk serangan berupa *blackhole* dan DDoS dibutuhkan perubahan *file* konfigurasi pada NS2 untuk masing-masing protokol. Pada AODV dibutuhkan perubahan pada *file* aodv.h dan aodv.cc pada direktori AODV dalam NS2, sedangkan pada AOMDV dibutuhkan perubahan pada *file* aomdv.h dan aomdv.cc pada direktori AOMDV pada NS2. Selain itu konfigurasi untuk membuat parameter skenario diatur di dalam *file script* Tcl. Pada penelitian ini implementasi dilakukan dengan menerapkan dua protokol *routing* pada MANET, yaitu AODV dan AOMDV dalam lingkungan yang berbeda berdasarkan jumlah *node* dan jumlah *node* penyerang. Berdasarkan skenario-skenario tersebut maka diperlukan empat *file script* Tcl yang mewakili masing-masing protokol dan masing-masing serangan. Sehingga dalam penelitian ini dibuat enam *file script* Tcl dengan nama aodv.tcl, aodvblackhole.tcl, aodvddos.tcl, aomdv.tcl, aomdvblackhole.tcl, dan aomdvddos.tcl ynag berisi konfigurasi parameter yang akan dibahas pada subbab selanjutnya.

Konfigurasi serangan *blackhole* pada kedua protokol diperlukan agar *node* penyerang dapat berjalan pada simulasi skenario. Konfigurasi dilakukan dengan mengubah/menambahkan kode pada *file* aodv.h, aodv.cc, aomdv.h, dan aomdv.cc yang terletak pada masing-masing direktori pada NS2.

4.2.1.1 Konfigurasi Serangan *Node Blackhole*

Konfigurasi pada protokol AODV dilakukan dengan menambahkan kode pada aodv.h dan aodv.cc. Konfigurasi dilakukan dengan menambahkan *script* pada *file* aodv.h dan aomdv.h.

Source Code 1 : Konfigurasi Serangan *Blackhole*

```
1   bool          malicious;
```

Kode pada *Source Code* 1 : Konfigurasi Serangan *Blackhole* harus ditambahkan pada *file* aodv.h dan aomdv.h dengan mendefinisikan *malicious* dengan parameter *boolean* untuk *node* penyerang. Selain itu penambahan kode pada *file* aodv.cc dan aomdv.cc juga diperlukan dengan kode pada *Source Code* 2 : Konfigurasi Serangan *Blackhole.*

Source Code 2 : Konfigurasi Serangan *Blackhole*

```
1.  if(strncasecmp(argv[1], "hacker", 6) == 0) {
2.       malicious = true;
3.        return TCL_OK;
```

```
4.      }
5.  malicious = false;
6.
7.  if (malicious == true ) {
8.              drop(p, DROP_RTR_ROUTE_LOOP);
9.              return;
10.         }
11. else if (malicious == true) {
12.     seqno = max(seqno, rq->rq_dst_seqno)+1;
13.     if (seqno%2) seqno++;
14.
15.     sendReply(rq->rq_src,
16.               1,
17.               rq->rq_dst,
18.               seqno,
19.               MY_ROUTE_TIMEOUT,
20.               rq->rq_timestamp);
21.    Packet::free(p);
22.  }
```

Penjelasan :

1. Tabel 1-4 : merupakan proses inisiasi saat terdapat kode yang menunjuk pada kata "hacker" maka nilai *malicious* akan berubah menjadi *true.*
2. Tabel 5 adalah nilai default *malicious* bernilai *false.*
3. Tabel 7-8 :Jika *malicious* bernilai *true* maka *drop* jalur *routing* yang ada pada *routing table.*
4. Tabel 11-13 : Jika *malicious* bernilai *true* maka *sequence number* akan diberikan nilai maksimal atau nilai yang lebih besar dari nilai *sequence number* yang sebenarnya untuk *node blackhole* seolah *node* tersebut adalah *node* tujuan.
5. Tabel 15-22 : Mengirimkan *Reply* dengan ke *source node* dengan informasi yang telah dimanipulasi seperti *next hop* bernilai 1, semakin kecil *next hop* menandakan *node* tersebut memiliki jalur yang lebih pendek menuju *node* tujuan. Mengirimkan *sequence number* yang telah dimanipulasi, alamat *source node*, *time_out* jalur, dan *timestamp.*

4.2.1.2 Konfigurasi Serangan *Node* DDoS

Konfigurasi serangan DDoS pada kedua protokol diperlukan agar *node* penyerang dapat berjalan pada simulasi skenario. Konfigurasi dilakukan dengan mengubah/menambahkan kode pada *file* aodv.h, aodv.cc, aomdv.h, dan aomdv.cc yang terletak pada masing-masing direktori pada NS2. Konfigurasi dilakukan dengan menambahkan kode agar *node* melakukan proses *flooding* secara berkala

sesuai waktu yang telah ditentukan dengan melakukan proses *broadcast* RREQ kepada *node* yang tidak ada dalam jaringan secara terus menerus yang megakibatkan konsumsi *bandwidth* yang berlebihan dan menyebabkan *resource* pada jaringan menurun.

Konfigurasi pada protokol AODV dan AOMDV dilakukan dengan menambahkan kode pada aodv.h dan aomdv.h. Konfigurasi dilakukan dengan menambahkan *script Source Code* 3 dan *Source Code 4* pada *file* aodv.h, aomdv.h, aodv.cc, dan aomdv.cc.

Source Code 3 : Konfigurasi Serangan DDoS

```
1.  #define FLOOD_INTERVAL  0.2
2.  class FloodTimer : public Handler
3.          {
4.           public:
5.               FloodTimer(AODV* a):    agent(a){}
6.               void     handle(Event*);
7.           private:
8.               AODV     *agent;
9.               Event     intr;
10.         };
11.
12. FloodTimer ftimer;
13. friend class FloodTimer;
14. void  FloodRREQ(nsaddr_t dst);
15. bool  flooder;
```

Penjelasan :

1. Tabel 1 : Mendefinisikan berapa interval waktu *flooding* berlangsung, pada simulasi ini didefinisikan dengan tiap 0,2 detik *node* penyerang akan melakukan *flooding.*
2. Tabel 2-10 : Merupakan penambahan *class Timer* pada AODV.h dengan nama *FloodTimer*. Kemudian mendefinisikan bahwa *class FloodTimer* akan menggunakan protokol AODV untuk setiap *agent* yang ada pada jaringan tersebut.
3. Tabel 12 : Mendefinisikan objek *class* sebagai property AODV *class.*
4. Tabel 13 : Membuat *FloodTimer class* menjadi *friend class.*
5. Tabel 14 : Mendefinisikan fungsi void untuk *FloodRREQ.*
6. Tabel 15 : Mendefinisikan variable flooder bertipe *Boolean.*

Source Code 4 : Konfigurasi Serangan DDoS

```
1.  if(strcmp(argv[1], "flooder") == 0)
2.    {
3.            flooder = true;
4.            return TCL_OK;
5.    }
6.  ftimer.handle((Event*) 0);
7.  ftimer(this)
8.  flooder=false;
9.  void FloodTimer::handle(Event*)
10.   {
11.             if (agent->flooder==true)
12.              {
13.                      agent->FloodRREQ(150);
14.               }
15.              Scheduler::instance().schedule(this,
16. &intr, FLOOD_INTERVAL);
17.   }
18. void AODV::FloodRREQ(nsaddr_t dst)
19. {
20.   Packet *p = Packet::alloc();
21.   struct hdr_cmn *ch = HDR_CMN(p);
22.   struct hdr_ip *ih = HDR_IP(p);
23.   struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
24.   aodv_rt_entry *rt = rtable.rt_lookup(dst);
```

Penjelasan :

1. Tabel 1-4 : merupakan proses inisiasi saat terdapat kode yang menunjuk pada kata “flooder” maka nilai malicious akan berubah menjadi true
2. Tabel 6 : Menambahkan ftimer.handle((Event*) 0);
3. Tabel 7 : Menambahkan ftimer(this).
4. Tabel 8 : Merupakan nilai *default flooder* bernilai *false.*
5. Tabel 9-16 : Merupakan class void FloodTimer yang akan dipanggil saat terdapat *node* yang memiliki nilai *flooder true. Node flooder* bernilai *true* tersebut akan mengirimkan *RREQ* kepada *agent* dengan id 150/*node* yang memiliki nomor 150. Kemudian interval pengirim RREQ sesuai dengan yang telah ditetapkan sebelumnya yaitu 0,2 detik.
6. Tabel 17-24 : Merupakan modifikasi dari *class SendRequest* dengan mengganti nama *class* menjadi *FloodRREQ*. *Class FloodRREQ* memiliki nilai seperti tipe protokol yang digunakan, *port node* sumber dan tujuan, IP *node* sumber dan tujuan, *sequence number, hop count, time.*

4.2.1.3 Konfigurasi Sistem

Konfigurasi parameter pengujian dilakukan sebagai implementasi dari rancangan parameter yang telah dibuat. Konfigurasi dibuat pada *file script* Tcl yang selanjutnya dijalankan pada perangkat lunak *Network Simulator 2* (NS2). Berikut ini adalah pengaturan parameter pada salah satu protokol yang dibuat.

Source Code 5 : Konfigurasi Parameter Pengujian

```
1.  set val(chan)    Channel/WirelessChannel    ;
2.  set val(prop)    Propagation/TwoRayGround   ;
3.  set val(netif)   Phy/WirelessPhy            ;
4.  set val(mac)     Mac/802_11                 ;
5.  set val(ifq)     Queue/DropTail/PriQueue    ;
6.  set val(ll)      LL                         ;
7.  set val(ant)     Antenna/OmniAntenna        ;
8.  set val(ifqlen)  50                         ;
9.  set val(nn)      50                         ;
10. set val(rp)      AODV                       ;
11. set val(sc)/home/bahari/ns-allinone-2.35/ns-2.35/indep-utils/cmu-
12. scen-gen/setdest/50node.tcl
13. set val(cp)/home/bahari/ns-allinone-2.35/ns-2.35/indep-utils/cmu-
14. scen-gen/50cbr.tcl
15. set val(x)       1000                       ;
16. set val(y)       1000                       ;
17. set val(stop)    1000.0                     ;
```

Pada *Source Code* 5 : Konfigurasi Parameter Pengujian, nomor 1-8 adalah konfigurasi system sesuai apa yang di inginkan. Pada tabel nomor 1 tipe *channel* menggunakan *wireless,* tabel nomor 2 merupakan tipe propagasi yaitu *Two Way Ground,* tipe *mac,* tipe antenna juga dapat diubah sesuai dengan simulasi yang ingin dijankan. Pada penelitian ini simulasi dilakukan dengan dua protokol *routing* yang berbeda yaitu AODV dan AOMDV. Oleh karena itu, maka didalam implementasinya dapat dilakukan dengan mengubah parameter pada tabel nomor 10 dengan mengubah menjadi AODV atau AOMDV sesuai protokol yang ingin digunakan. Simulasi akan menentukan protokol apa yang berjalan sesuai tabel nomor 10. Pada penelitian ini dilakukan simulasi dengan beberapa skenario yang berbeda yaitu berdasarkan jumlah *node* yaitu 20, 50, dan 100 hal tersebut dapat diubah pada tabel nomor 9 yang merupakan berapa banyak *node* yang ingin dibuat dalam simulasi. Pada tabel nomor 11-14 merupakan parameter untuk memanggil konfigurasi topologi dan pergerakan *node* yang terletak pada direktori yang telah ditentukan. Selanjutnya pada tabel nomor 15 dan 16 merupakan luas area simulasi yang digunakan yakni x=1000m dan y=1000m. Tabel nomor 17 merupakan lama simulasi yaitu 1000 detik.

4.2.1.4 Konfigurasi *Node* Penyerang

Pada penelitian ini konfigurasi *node* penyerang diletakkan didalam masing-masing *file script* Tcl.

A. Blackhole

Konfigurasi *node* penyerang dilakukan sesuai implementasi dari rancangan penyerangan yang telah dibuat. Setiap skenario dibuat dengan jumlah *node* penyerang yang berbeda-beda yaitu 0, 2, 4, 6, 8, 10 node. Konfigurasi diatur pada *file script* Tcl yang telah dibuat yaitu aodvblackhole.tcl dan aomdvblackhole.tcl dengan pengaturan *Source Code* 6 : Konfigurasi *Node* Penyerang *Blackhole*

Source Code 6 : Konfigurasi Node Penyerang Blackhole

```
1.  $ns_ at 500.0 "[$node_(43) set ragent_] hacker"
2.  $ns_ at 500.0 "[$node_(69) set ragent_] hacker"
3.  $ns_ at 500.0 "[$node_(7) set ragent_] hacker"
4.  $ns_ at 500.0 "[$node_(82) set ragent_] hacker"
5.  $ns_ at 500.0 "[$node_(1) set ragent_] hacker"
6.  $ns_ at 500.0 "[$node_(24) set ragent_] hacker"
7.  $ns_ at 500.0 "[$node_(91) set ragent_] hacker"
8.  $ns_ at 500.0 "[$node_(98) set ragent_] hacker"
9.  $ns_ at 500.0 "[$node_(90) set ragent_] hacker"
10  $ns_ at 500.0 "[$node_(94) set ragent_] hacker"
```

Konfigurasi dilakukan dengan membuat *node* sebagai penyerang dengan melakukan set ragent_ sebagai *hacker. Node* penyerang dijalankan saat detik ke 500. Alasan dipilihnya *node blackhole* pada tabel diatas adalah berdasarkan peletakkan *node* yang diacak dan tersebar pada topologi jaringan.

B. DDoS

Konfigurasi *node* penyerang dilakukan sesuai implementasi dari rancangan penyerangan yang telah dibuat. Setiap skenario dibuat dengan jumalh *node* penyerang yang berbeda-beda yaitu 0, 2, 4, 6, 8, 10 node. Konfigurasi diatur pada *file script* Tcl yang telah dibuat yaitu aodvddos.tcl dan aomdvddos.tcl dengan pengaturan *Source Code* 7 : Konfigurasi *Node* Penyerang DDoS

Source Code 7 : Konfigurasi *Node* Penyerang DDoS

```
1.  $ns_ at 500.0 "[$node_(43) set ragent_] flooder"
2.  $ns_ at 500.0 "[$node_(69) set ragent_] flooder"
3.  $ns_ at 500.0 "[$node_(7) set ragent_] flooder"
4.  $ns_ at 500.0 "[$node_(82) set ragent_] flooder"
5.  $ns_ at 500.0 "[$node_(1) set ragent_] flooder"
```

```
6.  $ns_ at 500.0 "[$node_(24) set ragent_] flooder"
7.  $ns_ at 500.0 "[$node_(91) set ragent_] flooder"
8.  $ns_ at 500.0 "[$node_(98) set ragent_] flooder"
9.  $ns_ at 500.0 "[$node_(90) set ragent_] flooder"
10. $ns_ at 500.0 "[$node_(94) set ragent_] flooder"
```

Tabel nomor 1-9 konfigurasi dilakukan dengan membuat *node* sebagai penyerang dengan melakaukan set ragent_ sebagai *flooder. node* penyerang dijalankan saat detik ke 500. Alasan dipilihnya *node blackhole* pada tabel diatas adalah berdasarkan peletakkan *node* yang diacak dan tersebar pada topologi jaringan.

4.2.1.5 Konfigurasi Topologi dan Pergerakan Jaringan

Pada konfigurasi topologi dan pergerakan jaringan dilakukan dengan memanfaatkan *automatic TCL code generation* yang tersedia pada *Network Simulator 2*.

Pada konfigurasi topologi jaringan digunakan *setdest* yang terdapat pada NS2. Konfigurasi topologi jaringan dilakukan sebagai implementasi dari rancangan simulasi yang dibuat. Tiap skenario simulasi memiliki jumlah *node* yang berbeda-beda yakni 20, 50, dan 100 *node.* Konfigurasi topologi jaringan diatur dalam *file script* 20node.tcl, 50node.tcl, dan 100node.tcl.

```
Konfigurasi Pergerakan Node
$  ./setdest -v 1 -n 50  -M 2  -P 1 -x 1000 -y 1000 > 50node.tcl
```

Konfigurasi Pergerakan *Node* contoh dari salah satu konfigurasi yang dibuat menggunakan *setdest*. *Setdest* berguna untuk membuat *hop* antara satu *node* dengan *node* yang lain dengan menggunakan obyek GOD (*General Operations Director*). Membuat pergerakan *node* dalam meter/detik. Membuat *node-node* bergerak secara acak menggunakan *setdest* dengan minimal dan maksimal kecepatan yang dapat diatur.

-v = versi *setdest*

-n = jumlah *node*

-M = maksimal kecepatan m/s

-P = *pause time*

-x = luas area x

-y = luas area y

File script disimpan dengan nama 50node.tcl

4.2.1.6 Konfigurasi Pengiriman paket

Konfigurasi pengiriman paket dilakukan sebagai implementasi dari perancangan pengirian paket dari *source node* ke *destination node.* Konfigurasi pengiriman paket dilakukan dengan memanfaatkan *cbrgen* yang ada pada *Network Simulator 2. Crbgen* berguna untuk membuat koneksi antar *node* sehingga saling terhubung, membuat tipe agen (cbr atau tcp). Menentukan berapa r*ate* paket yang dikirimkan. Konfigurasi pengiriman paket diatur pada *file script* Tcl yang dibuat yaitu 20cbr.tcl, 50cbr.tcl, dan 100cbr.tcl.

Source Code 8 : Konfigurasi Pengiriman Paket

```
1. set udp_(0) [new Agent/UDP]
2. $ns_ attach-agent $node_(0) $udp_(0)
3. set null_(0) [new Agent/Null]
4. $ns_ attach-agent $node_(42) $null_(0)
5. $cbr_(0) set packetSize_ 1024
6. $ns_ connect $udp_(0) $null_(0)
```

Source Code 8 : Konfigurasi Pengiriman Paket merupakan konfigurasi pengiriman paket pada simulasi. *Setup agent* pada pengiriman paket dibutuhkan untuk membangun konektifitas jaringan dalam *Network Simulator 2.* Dalam penelitian ini protokol jaringan yang dipakai adalah UDP. Pengaturan dilakukan pada tabel nomor 1, yakni set udp [new Agent/UDP] dimana Agent/UDP berperan sebagai sumber pengiriman paket data yang dilakukan dengan transport jaringan UDP. Selanjutnya memilih Agent/UDP dilakukan seperti tabel nomor 2, yakni $ns_ attach-agent $node_(0) $udp_(0) sebagai *source node*. Selanjutnya mengatur *destination node* agar dapat menerima paket data secara bebas dengan membuat Agent/Null pada tabel nomor 3. Pada tabel nomor 4 atur agar *destination node* tersambung dengan Agent/Null yang telah dibuat. Tabel nomor 5 mengatur berapa besar paket yang akan dikirmkan. Kemudian yang terakhir adalah menghubungkan *source node* dengan *destination node* seperti pada tabel nomor 6.

Source Code 9 : Konfigurasi Model Pengiriman Paket

```
1. set cbr_(0) [new Application/Traffic/CBR]
2. $cbr_(0) set packetSize_ 1024
3. $cbr_(0) set rate_ 0.1mb
4. $cbr_(0) set random_ null
5. $cbr_(0) attach-agent $udp_(0)
6. $ns_ connect $udp_(0) $null_(0)
7. $ns_ at 50.0 "$cbr_(0) start"
8. $ns_ at 1000.0 "$cbr_(0) stop"
```

Source Code 9 : Konfigurasi Model Pengiriman Paket merupakan konfigurasi model pengiriman paket pada simulasi. Model transmisi yang digunakan pada penelitian ini adalah CBR yang diatur pada tabel nomor 1. Kemudian tabel nomor 2 digunakan untuk menentukan berapa besar paket data yang akan dikirimkan. Tabel nomor 3 merupakan *transmission rate* yang dibutuhkan untuk *node* mengirimkan paket data sebesr 0,1 mbps. Pada tabel nomor 7 dan 8 adalah pada detik berapa pertukaran paket data terjadi dan detik berapa paket data berhenti dikirimkan.

4.2.1.7 Konfigurasi Pemrosesan Data Output

Konfigurasi data *output* dilakukan sebagai pengaturan simulasi agar *network Simulator 2* dapat memroses *output* sesuai skenario yang telah dirancang. Konfigurasi data *output* dilakukan dengan mengubah *file script* Tcl yang telah dibuat yaitu aodv.tcl, aodvblackhole.tcl, aodvddos.tcl, aomdv.tcl, aomdvblackhole.tcl, aomdvddos.tcl. berikut konfigurasi pemrosesan data *output* dalam *file script* Tcl.

Source Code 10 : Konfigurasi Pemrosesam Data *Output*

```
1. set tracefile [open aodv.tr w]
2. $ns_ trace-all $tracefile
3. set namfile [open aodv.nam w]
4. $ns_ namtrace-all $namfile
```

Source Code 10 : Konfigurasi Pemrosesan Data *Output* digunakan agar data dapat dianalisis. pada tabel nomor 1 *file* (.tr) di *set* dengan nama aodv.tr yang apabila simulasi dijalankan, *output* akan tersimpan dalam *file* aodv.tr tersebut. Tabel nomor 2 adalah merekam seluruh jejak paket dan *routing* dengan *script* $ns_trace-all $traecfile. Pada tabel nomor 3 *file* (.nam) disimpan dengan nama aodv.nam. dan terakhir pada tabel nomor 4 berfungsi merekam seluruh detail simulasi dalam bentuk animasi.

4.3 Pengujian

Dalam subbab ini akan membahas tentang skenario pengujian dan Teknik pengambilan data berdasarkan simulasi yang telah dijalankan pada *Network Simulator 2.*

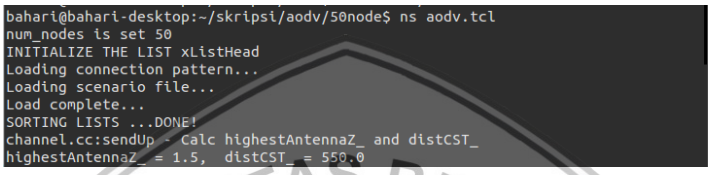
4.3.1 Skenario Pengujian

Pada penelitian ini pengukuran data hasil pengujian dilakukan dengan parameter *Packet Delivery Ratio, End-to-End Delay,* dan *Normalized Routing Load.* Protokol yang digunakan yaitu AODV dan AOMDV. Kedua protokol diuji dengan beberapa skenario yang berbeda, yaitu pengujian dengan 20, 50 dan 100 *node.* Pengujian juga dilakukan dengan beberapa variasi, yakni dengan tidak ada serangan dan dengan serangan. Serangan yang dipakai pada pengujian kali ini adalah serangan *Blackhole* dan DDoS. Jumlah *node* penyerang pada tiap skenario

adalah 0, 2, 4, 6, 8, 10 pada *Blackhole* dan DDoS. Pengujian dilakukan dengan mengirimkan paket sebesar 1024 bytes dengan memulai pengiriman pertama pada detik ke 50 dan berakhir pada detik ke 1000. *Transmission rate data* pada simulasi ini sebesar 0,1 mbps. Pada Skenario yang melibatkan serangan, *node* penyerang menyerang pada detik ke 500 hingga 1000.

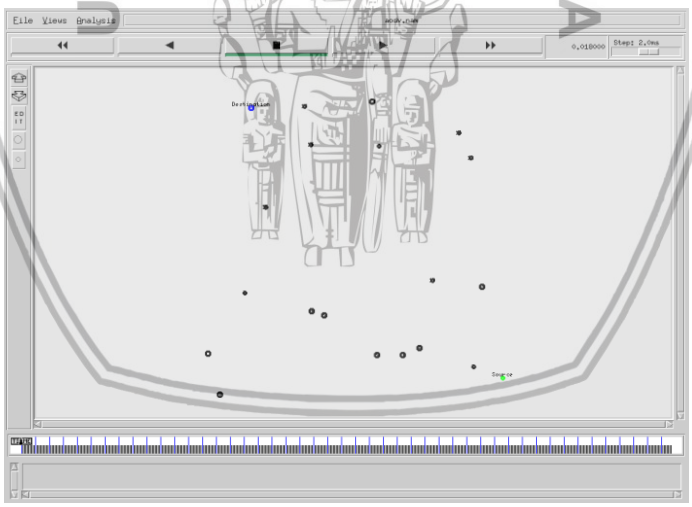
4.3.2 Pengujian Simulasi Protokol *Routing* MANET

Simulasi protokol *routing* dilakukan sebagai implementasi beberapa skenario pengujian pada protokol AODV dan AOMDV yang dirancang pada *Network Simulator 2.* Gambar 4.5 merupakan hasil eksekusi dari *file script* Tcl yang telah dibuat.



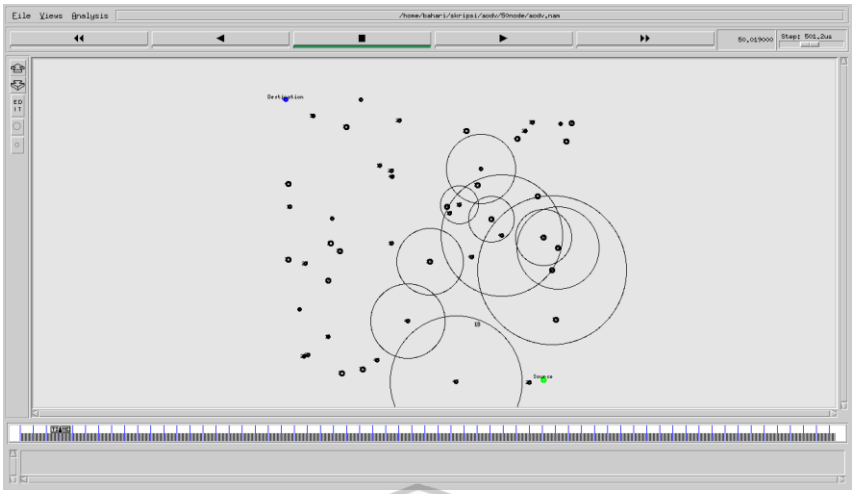
Gambar 4.5 Hasil Eksekusi *file* AODV.tcl

Dalam menjalankan *file script* Tcl yang telah dibuat dengan menggunakan perintah "ns" pada terminal. Jika *file script* Tcl berhasil dijalankan dan tidak ada pesan *error* maka secara otomatis *output* berupa *network animation file* (.nam) akan tampil seperti Gambar 4.6, Gambar 4.7, dan Gambar 4.8



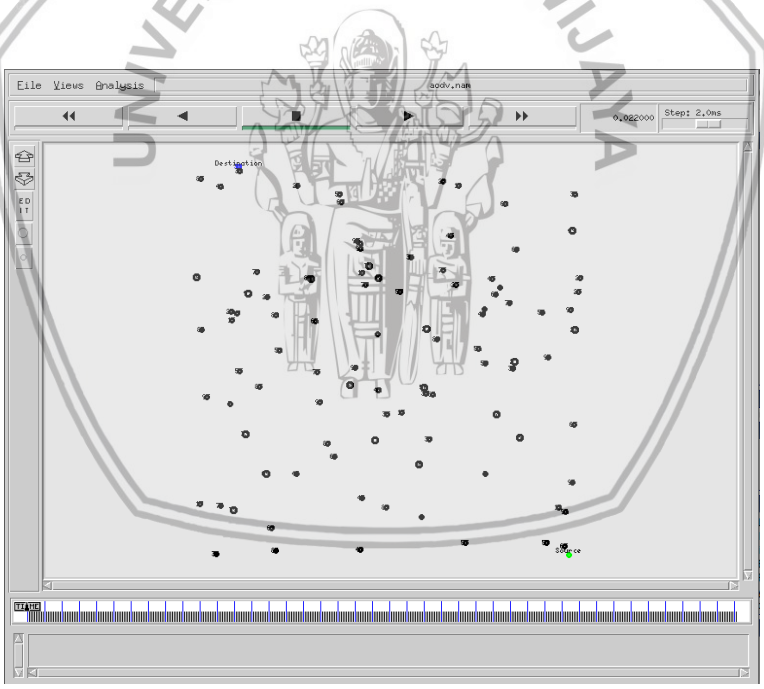
Gambar 4.6 Hasil Output Network Animation File Skenario 20 Node

Gambar 4.6 merupakan simulasi skenario dari topologi jaringan dengan menggunakan 20 *node*, jarak antara *node* sumber dan *node* tujuan sengaja ditempatkan pada posisi saling jauh agar hasil pengujian yang didapatkan lebih optimal.



Gambar 4.7 Hasil Output Network Animation File Skenario 50 Node

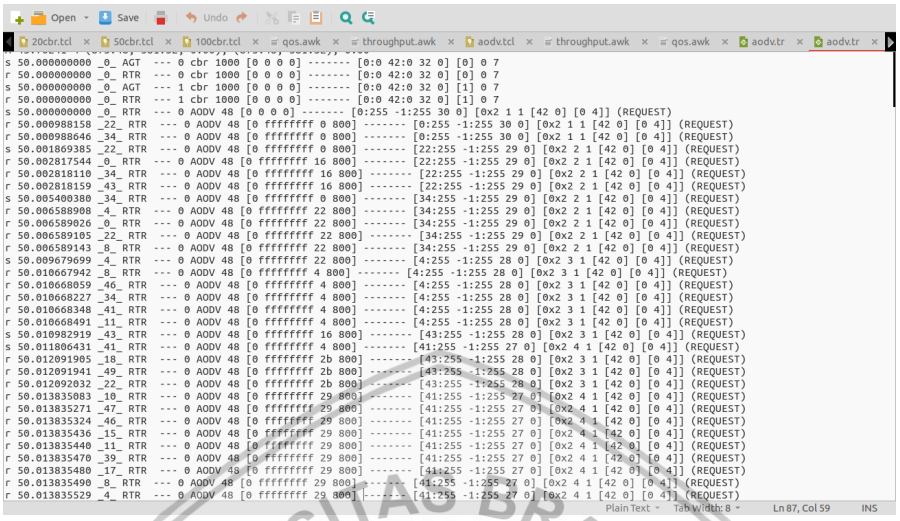
Gambar 4.7 merupakan simulasi skenario dari topologi jaringan dengan menggunakan 50 *node*, jarak antara *node* sumber dan *node* tujuan sengaja ditempatkan pada posisi saling jauh agar hasil pengujian yang didapatkan lebih optimal.



Gambar 4.8 Hasil Output Network Animation File Skenario 100 Node

Gambar 4.8 merupakan simulasi skenario dari topologi jaringan dengan menggunakan 100 *node*, jarak antara *node* sumber dan *node* tujuan sengaja ditempatkan pada posisi saling jauh agar hasil pengujian yang didapatkan lebih optimal.

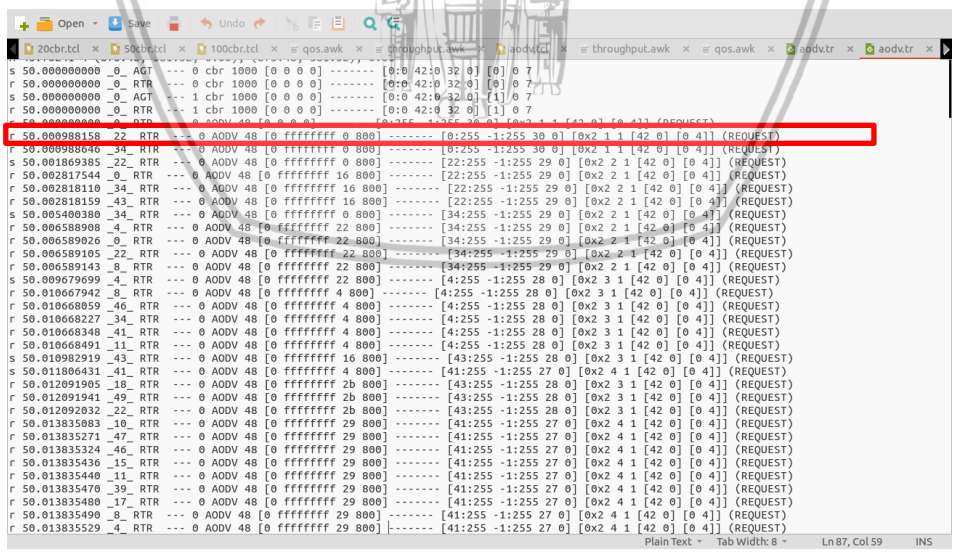
Output lainnya yang akan muncul selain *network animation file* (.nam) adalah *network trace file* (.tr) yang berisi rekaman seluruh data lalu lintas jaringan selama simulasi berlangsung secara detail seperti Gambar 4.9.



Gambar 4.9 Hasil Output Network Trace File

4.4 Pengumpulan dan Pengambilan Data

Pengumpulan dan pengambilan data dilakukan dengan mengolah salah satu *file output* simulasi, yaitu *network trace file* (.tr). Pengolahan data akan dijalankan dengan pemrograman AWK yang dapat digunakan untuk memroses data berdasarkan kolom pada *file network trace file* (.tr). berikut penjelasan Gambar 4.10 untuk struktur *output Network trace file*.



Gambar 4.10 Struktur Output Network Trace File

1. Pada kolom pertama berisi detail dari status paket yang dikirimkan. Status paket berupa s (*send*) yaitu paket dikirim, r (*received*) paket data diterima, d (*drop*) paket data di *drop,* f (*forward*) paket data di teruskan, dan c (*collision*) paket data bertabrakan pada lapisan MAC.

2. Pada kolom kedua adalah detail dari waktu aktifitas dalam satuan detik.
3. Pada kolom ketiga adalah id *node* tempat terjadi aktifitas.
4. Pada kolom keempat merupakan letak tingkatan lapisan jaringan tempat aktifitas terjadi. Contoh seperti AGT (*agent*) terdapat pada lapisan aplikasi, RTR (*routing*), MAC, IFQ(*interface queue*) merupakan antrian paket keluar yang terdapat diantara *link* dan MAC, LL (*link layer*), dan PHY (*physical*).
5. Pada kolom keenam adalah id paket dalam *sequence number.*
6. Pada kolom ketujuh adalah penjelasan dari tipe paket seperti data CBR (*constant bitrate*), ARP, RTS, CTS, dan paket *routing* (AODV, AOMDV).
7. Pada kolom kedelapan merupakan besar paket yang diproses.
8. Pada kolom kesembilan merupakan alamat *source node* dan tujuan MAC yang masing-masing adalah "0" dan "ffffffff". Nilai 800 merupakan IP paket data.
9. Pada kolom kesepuluh merupakan alamat IP *source node* dan *destination node* yaitu 0 dan 1 sedangkan 255 adalah portnya. 30 dan 0 adalah alamat *next hop* dan *time to live.*
10. Pada kolom kesebelas hingga ketiga belas merupakan rekaman informasi *routing* yang memiliki paket RREQ yang ditandai dengan ID "0x2" selanjutnya *hop* adalah "0" dan *broadcast id* adalah "1". Sedangkan alamat IP *destionasion node* dan *sequence* number adalah 1 dan 0, serta alamat IP *source node* dan *sequence number* adalah 0 dan 4.
11. Pada kolom keempat belas tanda (REQUEST) menandakan bahwa paket berjenis RREQ.

4.4.1 Pengumpulan dan Pengambilan Data *Packet Delivery Ratio*

Proses pengumpulan dan Pengambilan data *packet delivery ratio* dilakukan dengan pemrograman AWK yang dirancang untuk membaca dan menganalisis *file output network trace* dari *Network Simulator 2.* Kode program AWK *packet delivery ratio*

Source Code 11 : Program AWK *Packet Delivery Ratio*

```
1.  BEGIN {
2.  sent=0;
3.  received=0;
4.  }{
5.    if($1=="s" && $4=="AGT")
6.     { sent++;}
7.    else if($1=="r" && $4=="AGT")
8.     {received++;}
9.  }END{
10. print "Packet Sent= ",sent;
11. print "Packet Received= ",received;
```

```
12.  print "Packet Delivery Ratio=  " (received/sent) * 100 "%";
13.  }
```

Penjelasan :

1. Baris 1-4 merupakan inisialisasi awal program dengan mendefinsikan sent dan received yang berguna untuk menampung nilai paket data yang dikirim dan diterima.
2. Baris 5 dan 6 berfungsi untuk mencocokkan data yang dikirim apakah bernilai S dan memiliki tipe AGT yang kemudian disimpan pada variable sent
3. Baris 7-8 berfungsi untuk mengecek seluruh baris pada *trace file,* yaitu apabila kolom ke 4 dan 1 adalah "AGT dan "r" (*received*) maka variable received akan bertambah.
4. Baris 9 merupakan akhir dari perhitungan program.
5. Baris 10 - 13 adalah melakukan print untuk menampilkan hasil akhir perhitungan *packet delivery ratio* dengan melakukan print nilai *sent, received,* dan rata-rata *packet delivery ratio.*

Kemudian kode program dengan format *file* AWK dan dijalankan seperti pada Gambar 4.11.

```
bahari@bahari-desktop:/media/windows/skripsi2/aodv/50$ awk -f pdr.awk aodv.tr
Packet Sent=  23194
Packet Received=  22907
Packet Delivery Ratio=  98.7626%
bahari@bahari-desktop:/media/windows/skripsi2/aodv/50$
```

Gambar 4.11 Hasil Eksekusi Program AWK Packet Delivery Ratio

4.4.2 Pengumpulan dan Pengambilan Data *End-to-End Delay*

Proses pengumpulan dan Pengambilan data *End-to-end delay* dilakukan dengan pemrograman AWK yang dirancang untuk membaca dan menganalisis *file output network trace* dari *Network Simulator 2.* Kode program AWK *end-to-end delay*

Source Code 12 : Program AWK *End-to-End Delay*

```
1.  BEGIN {
2.                  seqno= -1;
3.                  count = 0;
4.  }{
5.      if ($4 == "AGT" && $1 == "s" && seqno < $6) {
6.      seqno = $6; }
7.      if($4 == "AGT" && $1 == "s") {
8.      start_time[$6] = $2; }
```

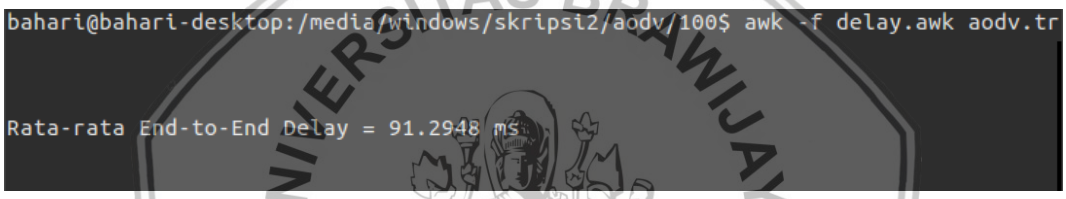
```
9.      else if(($7 == "cbr") && ($1 == "r")) {
10.     end_time[$6] = $2; }
11.     else if($1 == "D" && $7 == "cbr") {
12.     end_time[$6] = -1; }
13. }END {
14.      for(i=0; i<=seqno; i++) {
15.           if(end_time[i] >=0) {
16.           delay[i] = end_time[i] - start_time[i];
17.           count++;
18.           }
19.           else
20.           {
21.           delay[i] = -1; } }
22.           for(i=0; i<count; i++) {
23.               if(delay[i] >= 0) {
24.               n_to_n_delay = n_to_n_delay + delay[i]; } }
25.               n_to_n_delay = n_to_n_delay/count;
26. print "Average End-to-End Delay = " n_to_n_delay * 1000 " ms";
```

Penjelasan :

1. Baris 1-4 merupakan inisialisasi awal program dengan variable seqno dan count untuk menampung nilai *sequence number* dari awal paket paket dikirim hingga akhir. Variabel count berfungsi untuk membagi nilai rata-rata *delay* pada akhir program.
2. Baris 5-6 berfungsi untuk mendapatkan variable seqno dengan kondisi jika kolom 1 bernilai s dan kolom 4 bernilai AGT dan variable seqno tidak lebih dari nilai pada kolom 6. Hal ini digunakan untuk mendapatkan *sequence number* paket yang terakhir dikirim.
3. Baris 7-8 berfungsi untuk mengecek baris pada kolom *network trace file,* yaitu kolom ke 1 dan ke 4 apakah bernilai s dan AGT, jika benar maka variable array start_time dengan *index* sesuai nomor urut pake kolom 6 akan diisi dengan nilai waktu berjalannya paket tersebut ketika dikirim oleh *source node.*
4. Baris 9-10 berfungsi untuk mengecek baris pada kolom *network trace file,* yaitu kolom ke 1 dan ke 7 apakah bernilai r dan CBR, jika benar maka variable array end_time dengan *index* sesuai nomor urut pake kolom 6 akan diisi dengan nilai waktu berjalannya paket tersebut ketika dikirim oleh *destination node.*
5. Baris 11-12 berfungsi untuk mengecek baris pada kolom *network trace file,* yaitu kolom ke 1 dan ke 7 apakah bernilai D dan CBR, jika benar maka variable array end_time dengan *index* sesuai nomor urut pake kolom 6 akan diisi dengan nilai -1 yang merupakan paket gagal dikirim .

6. Baris 13-17 adalah akhir dari perngumpulan data yang ditandai denga END. Proses perhitungan estimasi waktu paket yang terkirim dilakukan dengan mengurangi waktu *end_time* dengan *start_time*. Kemudian nilai disimpan pada variable *array delay* dengan index sesuai *sequence number*. Kemudian dilakukan perhitungan jumlah paket yang berhasil terkirim pada varabel count.
7. Baris 19-21 berfungsi untuk mengisi paket berstatus D dengan nilai -1.
8. Baris 22-24 berfungsi untuk menjumlahkan seluruh waktu paket terikirim yang disimpan pada variable *array delay.*
9. Baris 25-26 berfungsi untuk menampilkan hasil rata-rata waktu *end-to-end delay* dengan membagi jumlah hasil total delay yang disimpan dalam variable n_to_n_delay dengan count.

Kemudian kode program dengan format *file* AWK dan dijalankan seperti pada Gambar 4.12.



Gambar 4.12 Hasil Eksekusi Program AWK End-to-End Delay

4.4.3 Pengumpulan dan Pengambilan Data *Normalized Routing Load*

Proses pengumpulan dan Pengambilan data *normalizes routing load* dilakukan dengan pemrograman AWK yang dirancang untuk membaca dan menganalisis *file output network trace* dari *Network Simulator 2.* Kode program AWK *normalized*

Source Code 13 : Kode Program AWK *Normalized Routing Load*

```
1.  BEGIN{
2.  recvd = 0;
3.  rt_pkts = 0;
4.  }
5.  {
6.  if (( $1 == "r") && ( $7 == "cbr") && ( $4=="AGT" )) recvd++;
7.  if (($1 == "s" || $1 == "f") && $4 == "RTR" && ($7 =="AODV" || $7
8.  =="message" || $7 =="AOMDV")) rt_pkts++;
9.  }
10. END{
11. printf("################################################\n");
12. printf("\n");
13. printf("Normalized Routing Load = %.3f\n", rt_pkts/recvd);
14. printf("\n");
```

```
15.  printf("##############################################\n");
16.  }
```

Penjelasan :

1. Baris 1-4 adalah inisiliasisasi awal program dengan membuat variable recvd dan rt_pkts yang digunakan untuk menampung nilai jumlah seluruh paket data baik yang diterima maupun yang diteruskan oleh *destination node.*
2. Baris 5 berfungsi untuk mengecek seluruh baris pada kolom *network trace file,* yakni kolom 1,7, dan 4 apakah r, CBR, dam AGT, jika benar maka nilai recvd akan bertambah.
3. Baris 7-9 berfungsi untuk status pada kolom ke 1 apakah bernilai s (*send*) ataukah f (*forwarded*), dan kolom ke 4 RTR (*routing request*). Kemudian mengecek kolom ke 7 apakah statusnya merupakan *agent routing* AODV atau AOMDV, jika benar nilai rt_pkts akan bertambah.
4. Baris ke 10-15 merupakan akhir pengumpulan dan pemrosesan data dan menampilkan hasil perhitungan *normalized routing load* dengan membagi variable rt_pkts dengan recvd.

Kemudian kode program dengan format *file* AWK dan dijalankan seperti pada Gambar 4.13.

```
bahari@bahari-desktop:/media/windows/skripsi2/aodv/100$ awk -f normalizedrouting
load.awk aodv.tr
##############################################################################
##

                        Normalized Routing Load = 0.410

##############################################################################
##
```

Gambar 4.13 Hasil Eksekusi Program AWK Normalized Routing Load

BAB 5 HASIL DAN ANALISIS

Dalam bab ini akan dilakukan pembahasan mengenai data hasil pengujian beserta analisis perbandingan antara protokol AODV dan AOMDV dari hasil yang telah didapatkan pada skenario pengujian. Analisis akan dilakukan terhadap hasil pengujian terhadap variasi jumlah *node,* jumlah *node* penyerang, dan tipe serangan yang dilakukan. Dari hasil analisis pengujian akan didapatkan bagaimana pengaruh setiap skenario pada kinerja protokol AODV dan AOMDV, serta akan diketahui bagaimana perbandingan kinerja kedua protokol tersebut pada MANET berdasarkan parameter *packet delivery ratio, end-to-end delay,* dan *normalized routing load.*

5.1 Hasil Pengujian

Pada sub bab ini terdapat penjelasan mengenai perbandingan hasil kinerja dari protokol AODV dan AOMDV dengan serangan *Blackhole* dan *DDoS* dengan setiap skenario pengujian.

5.1.1 Hasil Pengujian Pengaruh Serangan *Blackhole* Pada Protokol AODV dan AOMDV

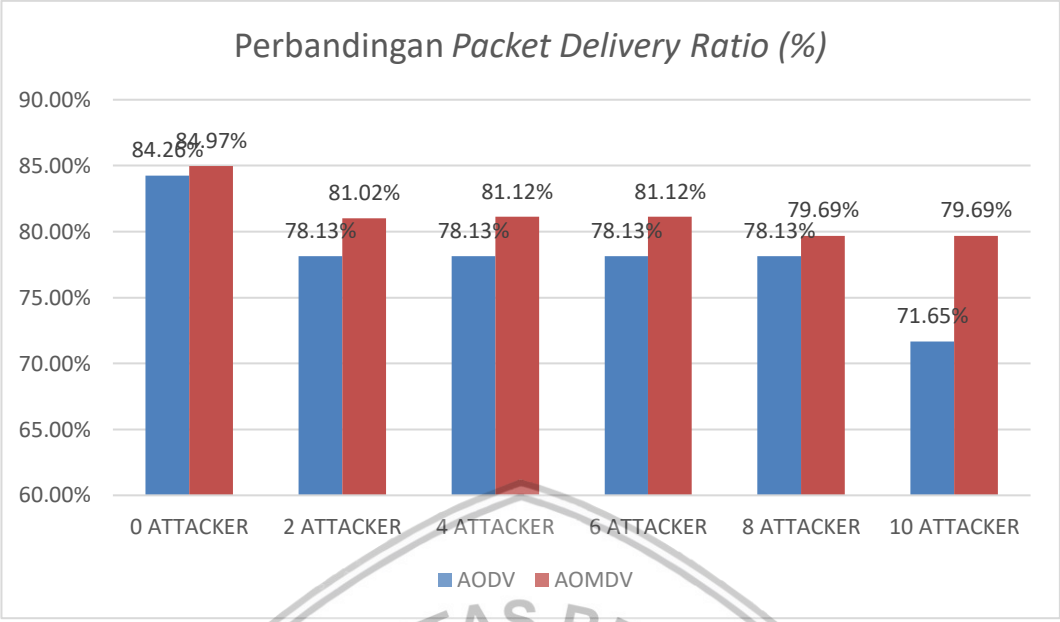
5.1.1.1 Packet Delivery Ratio

A. Skenario 20 *Node* Dengan Variasi 0, 2, 4, 6, 8, 10 *Node* Penyerang

Berdasarkan pengujian yang telah dilakukan dalam skenario 20 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.1

Tabel 5.1 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 20 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	20	84,26%	78,13%	78,13%	78,13%	78,13%	71,65%
AOMDV		84.97%	81.02%	81,11%	81,11%	79,69%	79,68%



Gambar 5.1 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 20 *Node*

Pada Gambar 5.1 merupakan grafik perbandingan hasil *packet delivery ratio* pada protokol AODV dan AOMDV dengan skenario pengujian 20 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 84,26% untuk 0 *node* penyerang, 78,13% untuk 2 *node* penyerang, 78,13% untuk 4 *node* penyerang78,13% untuk 6 *node* penyerang, 78,13% untuk 8 *node* penyerang, 31,65% untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian menunjukkan 84,97% untuk 0 *node* penyerang, 81,02% untuk 2 *node* penyerang, 81,11% untuk 4 *node* penyerang, 81,11% untuk 6 *node* penyerang, 79,69% untuk 8 *node* penyerang, 79,68% untuk 10 *node* penyerang.

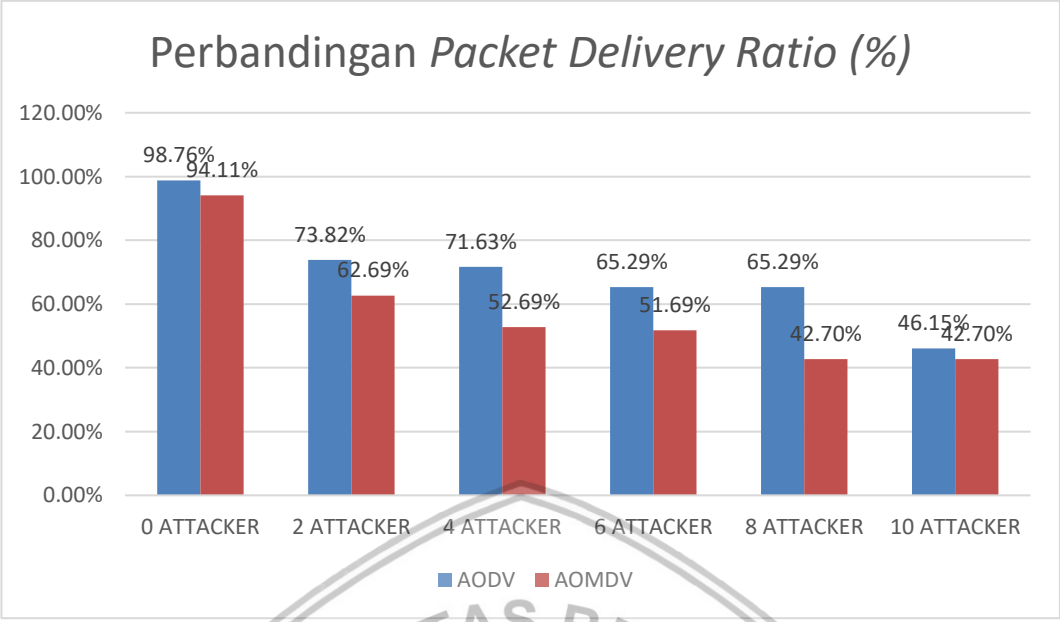
Hasil pengujian pada skenario 20 *node* menunjukkan bahwa kedua protokol memiliki rata-rata *packet delivery ratio* yang hampir sama tetapi pada protokol AODMV penurunan *packet delivery ratio* lebih baik dan pada skenario dengan 10 serangan penurunan *packet delivery ratio* pada protokol AODV sangat signifikan.

B. Skenario 50 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 50 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.2

Tabel 5.2 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 50 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	50	98,76%	73,82%	71,63%	65,29%	65,29%	46,15%
AOMDV		94,11%	62,68%	52,68%	51,68%	42,69%	42,69%



Gambar 5.2 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 50 *Node*

Pada Gambar 5.2 merupakan grafik perbandingan hasil *packet delivery ratio* pada protokol AODV dan AOMDV dengan skenario pengujian 50 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 98,76% untuk 0 *node* penyerang, 73,82% untuk 2 *node* penyerang, 71,63% untuk 4 *node* penyerang, 65,29% untuk 6 *node* penyerang, 65,29% untuk 8 *node* penyerang, 46,15% untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian menunjukkan 94,11% untuk 0 *node* penyerang, 42,68% untuk 2 *node* penyerang, 42,68% untuk 4 *node* penyerang, 42,68% untuk 6 *node* penyerang, 42,69% untuk 8 *node* penyerang, 42,69% untuk 10 *node* penyerang.

Hasil pengujian kedua protokol pada skenario 50 *node* menunjukkan bahwa semakin banyak *node* yang terdapat pada jaringan serta semakin banyak *node* penyerang maka *packet delivery ratio* akan semakin menurun hal ini disebabkan karena banyak paket yang tidak sampai pada *node* tujuan dikarenakan oleh *node blackhole*.

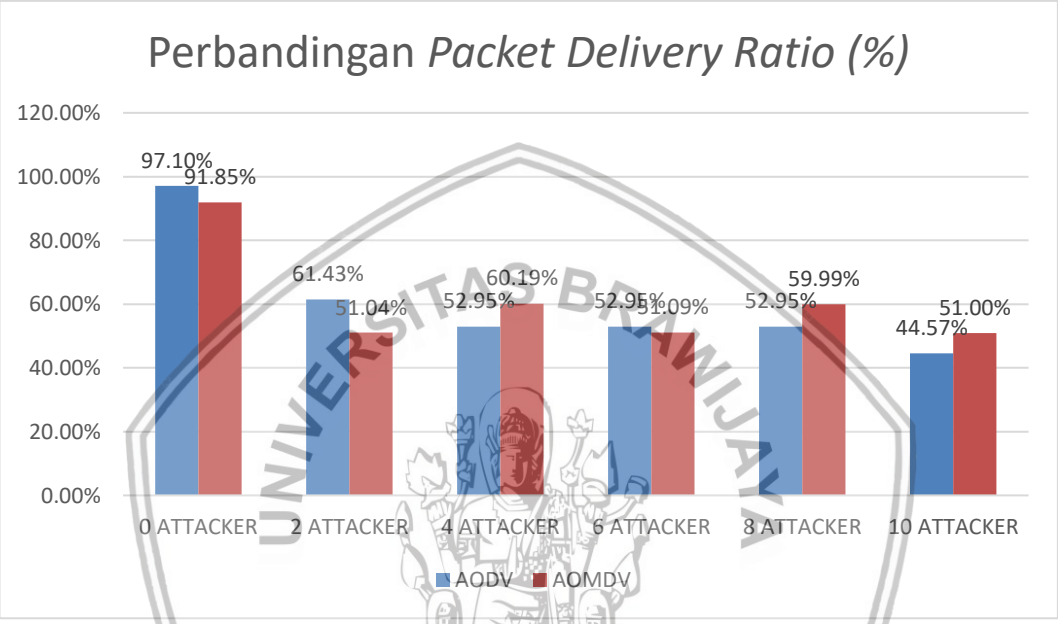
Pada protokol AODV sangat tingkat *packet delivery ratio* dipengaruhi oleh jarak *node* penyerang terhadap *node* sumber pada saat pembentukan jalur *routing,* pada saat jalur mengalami *expiration timeout* dan diperlukan pembaruan jalur lagi maka *node blackhole* akan mengirimkan RREP kepada *node* sumber bahwa seolah-olah *node blackhole* memiliki jalur terpendek dengan *hop count* paling sedikit sehingga paket data akan dikirimkan/diambil alih oleh *node blackhole.*

C. Skenario 100 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.3

Tabel 5.3 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 100 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	100	97,10%	61,43%	52,95%	52,94%	52,94%	44,57%
AOMDV		91,85%	51,04%	60,19%	51,09%	59,99%	51,00%



Gambar 5.3 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 100 *Node*

Pada Gambar 5.3 merupakan grafik perbandingan hasil *packet delivery ratio* pada protokol AODV dan AOMDV dengan skenario pengujian 20 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 97,10% untuk 0 *node* penyerang, 61,43% untuk 2 *node* penyerang, 52,95% untuk 4 *node* penyerang, 52,94% untuk 6 *node* penyerang, 52,94% untuk 8 *node* penyerang, 44,57% untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian menunjukkan 91,85% untuk 0 *node* penyerang, 51,04% untuk 2 *node* penyerang, 60,19% untuk 4 *node* penyerang, 51,09% untuk 6 *node* penyerang, 59,99% untuk 8 *node* penyerang, 51.00% untuk 10 *node* penyerang.

Hasil pengujian kedua protokol pada skenario 50 *node* menunjukkan bahwa semakin banyak *node* yang terdapat pada jaringan serta semakin banyak *node* penyerang maka *packet delivery ratio* akan semakin menurun hal ini disebabkan karena banyak paket yang tidak sampai pada *node* tujuan dikarenakan oleh *node blackhole*.

Pada protokol AODV sangat tingkat *packet delivery ratio* dipengaruhi oleh jarak *node* penyerang terhadap *node* sumber pada saat pembentukan jalur *routing,* pada saat jalur mengalami *expiration timeout* maka *node blackhole* akan

mengirimkan RREP kepada *node* sumber bahwa seolah-olah *node blackhole* memiliki jalur terpendek dengan *hop count* paling sedikit sehingga paket data akan dikirimkan/diambil alih oleh *node blackhole.*

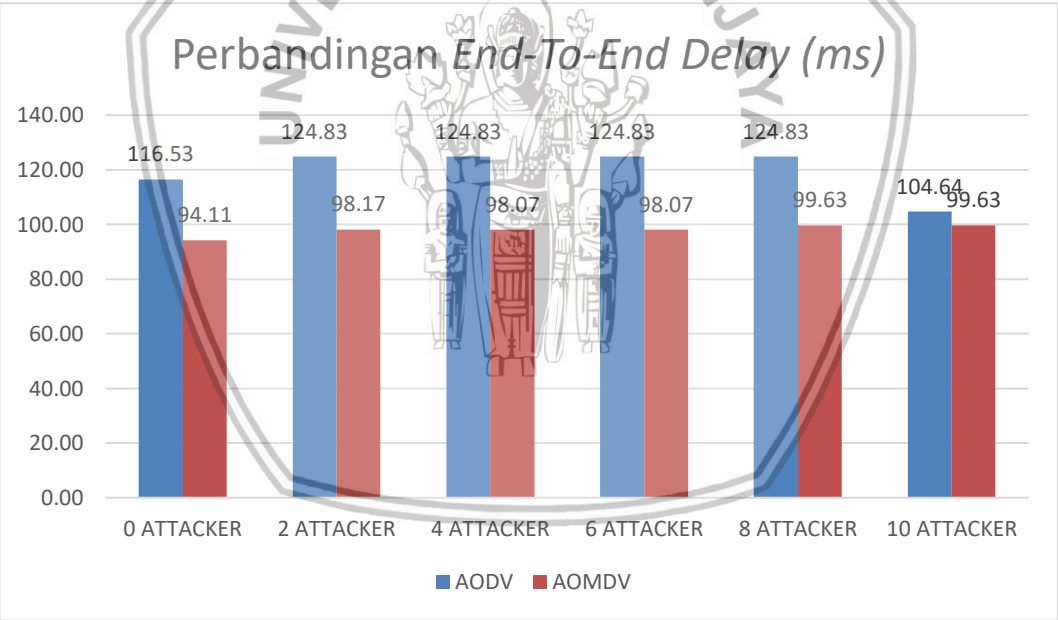
5.1.1.2 End-To-End Delay

A. Skenario 20 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 20 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *End-to-End Delay* yang dijabarkan pada tabel 5.4

Tabel 5.4 Hasil Pengujian *End-To-End Delay* Pada Skenario 20 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	20	116,53	124,83	124,83	124,83	124,83	104,65
AOMDV		94,11	98,17	98,07	98,07	99,63	99,62



Gambar 5.4 Hasil Pengujian *End-To-End Delay* Pada Skenario 20 *Node*

Pada Gambar 5.4 merupakan grafik perbandingan hasil rata-rata *End-to-End Delay* pada protokol AODV dan AOMDV dengan skenario pengujian 20 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 116,53 ms untuk 0 *node* penyerang, 124,83 ms untuk 2 *node* penyerang, 124,83 ms untuk 4 *node* penyerang, 124,83 ms untuk 6 *node* penyerang, 124,83 ms untuk 8 *node* penyerang, 104,65 ms untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 94,11 ms untuk 0 *node* penyerang, 98,17 ms untuk 2 *node* penyerang, 98,07 ms untuk 4 *node*

penyerang, 98,07 ms untuk 6 *node* penyerang, 99,63 ms untuk 8 *node* penyerang, 99,62 ms untuk 10 *node* penyerang.

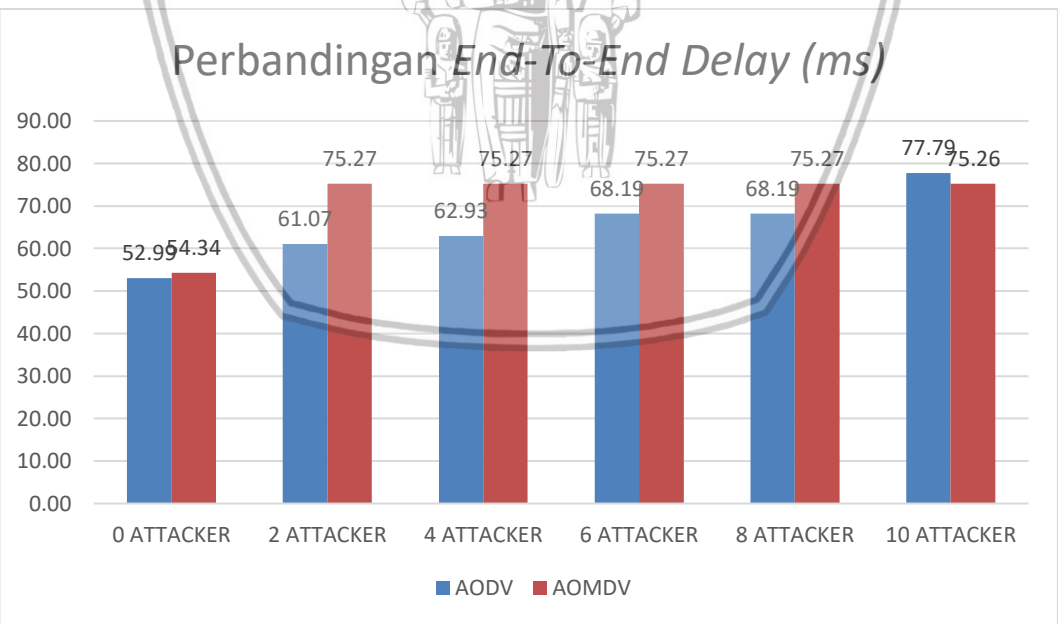
Dari hasil pengujian berdasarkan skenario 20 *node* menunjukkan bahwa protokol AOMDV memiliki nilai rata-rata *End-to-End Delay* yang lebih baik dibandingkan dengan protokol AODV. Hal ini disebabkan karena protokol AOMDV memelihara sejumlah jalur alternatif menuju *node* tujuan sehingga dapat mengurangi waktu *end-to-end delay* dalam proses pencarian jalur yang sewaktu-waktu dapat berubah. Sedangkan pada protokol AODV tidak memiliki algoritme untuk mengetahui jalur alternatif sehingga *end-to-end delay* menjadi lebih tinggi.

B. Skenario 50 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 50 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *End-to-End Delay* yang dijabarkan pada tabel 5.5

Tabel 5.5 Hasil Pengujian *End-To-End Delay* Pada Skenario 50 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	50	52,99	61,07	62,93	68,19	68,19	77,79
AOMDV		54,34	75,27	75,27	75,27	75,27	75,26



Gambar 5.5 Hasil Pengujian *End-To-End Delay* Pada Skenario 50 *Node*

Pada Gambar 5.5 merupakan grafik perbandingan hasil rata-rata *End-to-End Delay* pada protokol AODV dan AOMDV dengan skenario pengujian 50 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 52,99 ms untuk 0 *node* penyerang, 61,07 ms

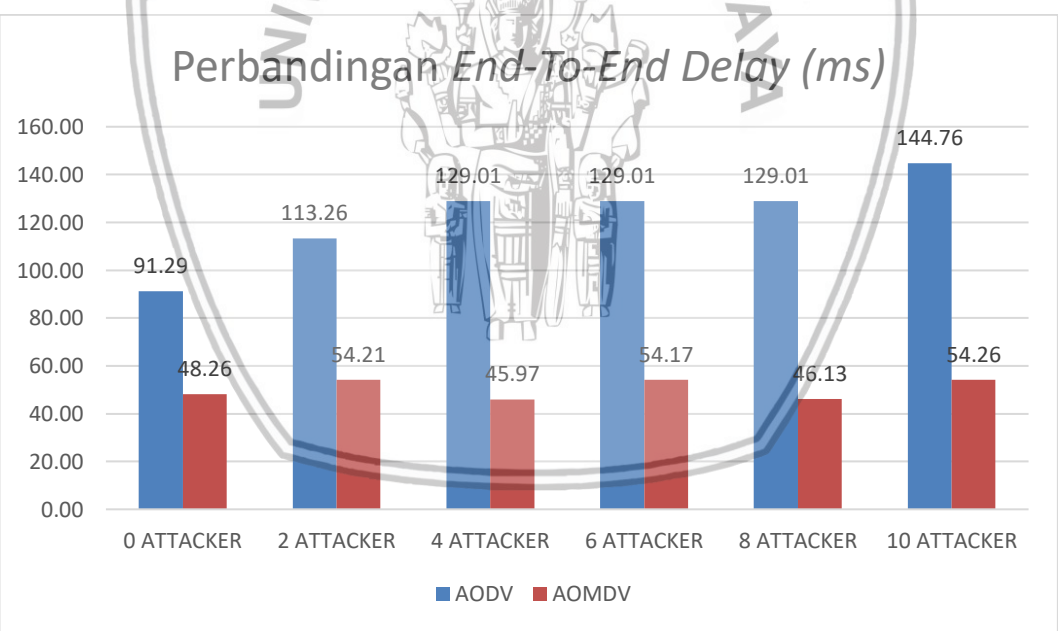
untuk 2 *node* penyerang, 62,93 ms untuk 4 *node* penyerang, 68,19 ms untuk 6 *node* penyerang, 68,19 ms untuk 8 *node* penyerang, 77,79 ms untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 54,34 ms untuk 0 *node* penyerang, 75,27 ms untuk 2 *node* penyerang, 75,27 ms untuk 4 *node* penyerang, 75,27 ms untuk 6 *node* penyerang, 75,27 ms untuk 8 *node* penyerang, 75,26 ms untuk 10 *node* penyerang.

C. Skenario 100 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *End-to-End Delay* yang dijabarkan pada tabel 5.6

Tabel 5.6 Hasil Pengujian *End-To-End Delay* Pada Skenario 100 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	100	91,29	113,26	129,00	129,00	129,00	144,76
AOMDV		48,26	54,21	45,97	54,17	46,13	54,26



Gambar 5.6 Hasil Pengujian *End-To-End Delay* Pada Skenario 100 *Node*

Pada Gambar 5.6 merupakan grafik perbandingan hasil rata-rata *End-to-End Delay* pada protokol AODV dan AOMDV dengan skenario pengujian 100 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 91,29 ms untuk 0 *node* penyerang, 113,26 ms untuk 2 *node* penyerang, 129,00 ms untuk 4 *node* penyerang, 129,00 ms untuk 6 *node* penyerang, 129,00 ms untuk 8 *node* penyerang, 144,76 ms untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 48,26 ms untuk 0 *node* penyerang, 54,21 ms untuk 2 *node* penyerang, 45,97 ms untuk 4 *node*

penyerang, 54,17 ms untuk 6 *node* penyerang 46,13 ms untuk 8 *node* penyerang, 54,26 ms untuk 10 *node* penyerang.

Dari hasil pengujian berdasarkan skenario 100 *node* menunjukkan bahwa protokol AOMDV memiliki nilai rata-rata *End-to-End Delay* yang lebih baik dibandingkan dengan protokol AODV. Hal ini disebabkan karena protokol AOMDV memelihara sejumlah jalur alternatif menuju *node* tujuan sehingga dapat mengurangi waktu *end-to-end delay* dalam proses pencarian jalur yang sewaktu-waktu dapat berubah. Sedangkan pada protokol AODV tidak memiliki algoritme untuk mengetahui jalur alternatif sehingga *end-to-end delay* menjadi lebih tinggi.

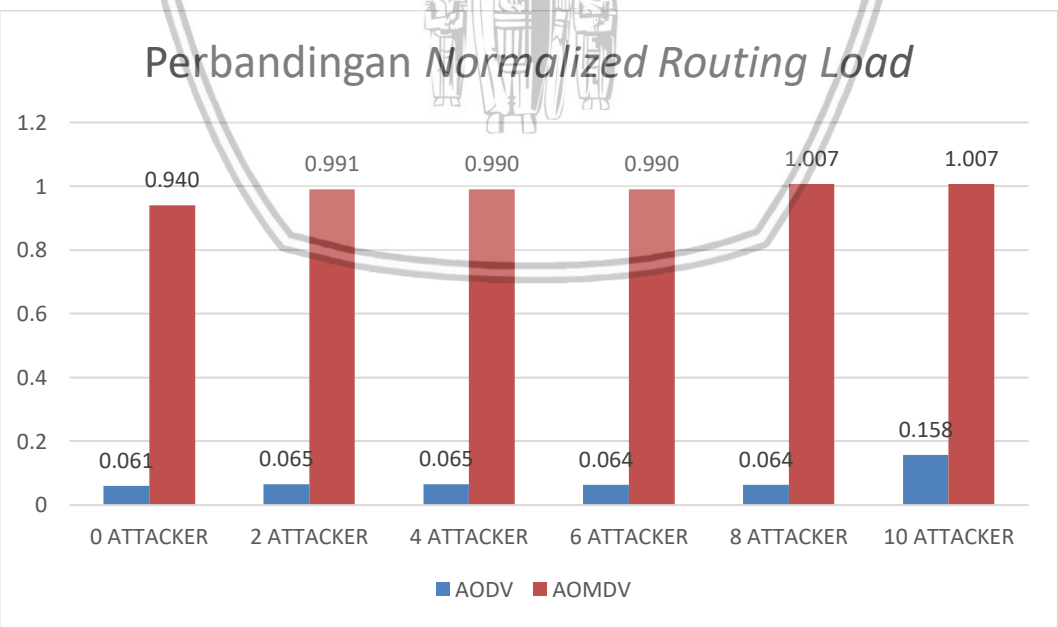
5.1.1.3 Normalized Routing Load

A. Skenario 20 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 20 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata *Normalized Routing Load* yang dijabarkan pada tabel 5.7

Tabel 5.7 Hasil Pengujian *Normalized Routing Load* Pada Skenario 20 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	20	0,061	0,065	0,065	0,064	0,064	0,158
AOMDV		0,946	0,991	0,990	0,990	1,007	1,007



Gambar 5.7 Hasil Pengujian *Normalized Routing Load* Pada Skenario 20 *Node*

Pada Gambar 5.7 merupakan grafik perbandingan hasil rata-rata *Normalized Routing Load* pada protokol AODV dan AOMDV dengan skenario

pengujian 20 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 0,061 untuk 0 *node* penyerang, 0,065 untuk 2 *node* penyerang, 0,065 untuk 4 *node* penyerang, 0,064 untuk 6 *node* penyerang, 0,064 untuk 8 *node* penyerang, 0,158 untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 0,946 untuk 0 *node* penyerang, 0,991 untuk 2 *node* penyerang, 0,990 untuk 4 *node* penyerang, 0,990 untuk 6 *node* penyerang 1,007 untuk 8 *node* penyerang, 1,007 untuk 10 *node* penyerang.

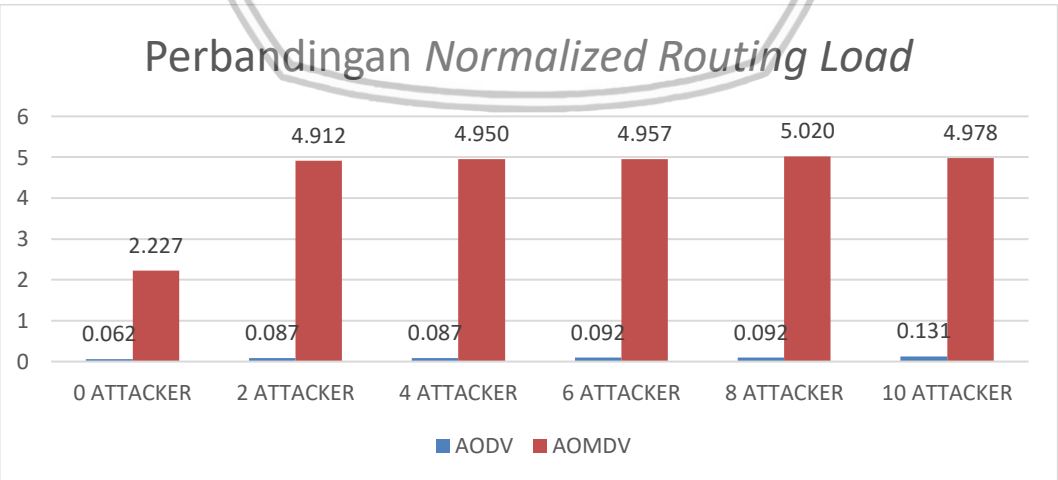
Dari hasil pengujian berdasarkan skenario 20 *node* menunjukkan bahwa protokol AODV memiliki nilai rata-rata *Normalized Routing Load* yang lebih baik dibandingkan dengan protokol AOMDV. Hal ini disebabkan karena protokol AODV membuang semua paket *RREQ* yang terduplikasi saat melakukan pencarian jalur dari *node* sumber ke *node* tujuan, berbeda dengan AOMDV yang menyimpan semua *RREQ* yang digunakan sebagai informasi cadangan saat jalur pengiriman data rusak.

B. Skenario 50 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 50 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata *Normalized Routing Load* yang dijabarkan pada tabel 5.8

Tabel 5.8 Hasil Pengujian *Normalized Routing Load* Pada Skenario 50 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	50	0,062	0,087	0,087	0,092	0,092	0,131
AOMDV		2,227	4,912	4,950	4,957	5,020	4,978



Gambar 5.8 Hasil Pengujian *Normalized Routing Load* Pada Skenario 50 *Node*

Pada Gambar 5.8 merupakan grafik perbandingan hasil rata-rata *Normalized Routing Load* pada protokol AODV dan AOMDV dengan skenario

pengujian 50 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 0,062 untuk 0 *node* penyerang, 0,087 untuk 2 *node* penyerang, 0,087 untuk 4 *node* penyerang, 0,092 untuk 6 *node* penyerang, 0,092 untuk 8 *node* penyerang, 0,131 untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 2,227 untuk 0 *node* penyerang, 4,912 untuk 2 *node* penyerang, 4,950 untuk 4 *node* penyerang, 4,957 untuk 6 *node* penyerang 5,020 untuk 8 *node* penyerang, 4,978 untuk 10 *node* penyerang.

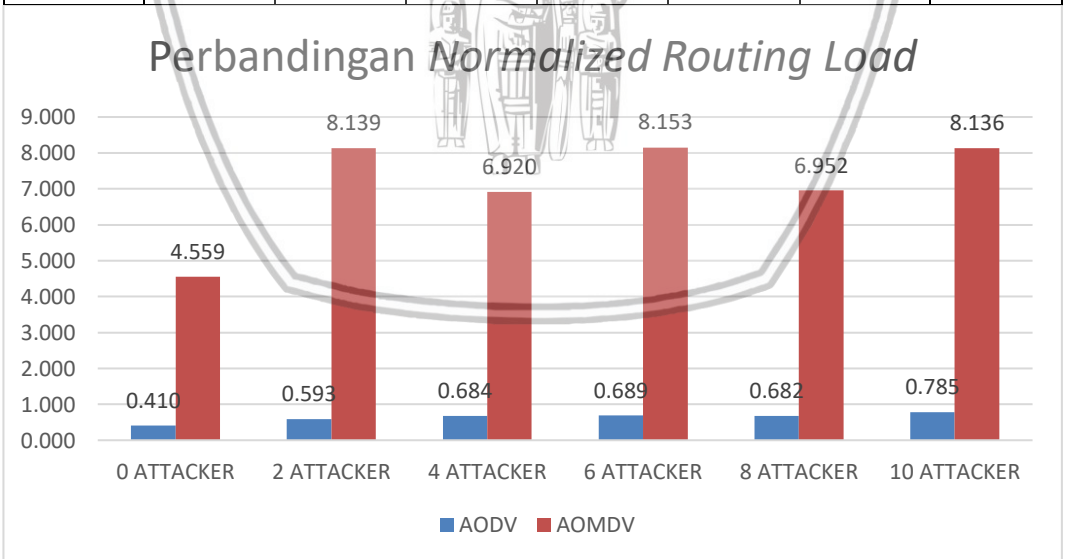
Dari hasil pengujian berdasarkan skenario 50 *node* menunjukkan bahwa protokol AODV memiliki nilai rata-rata *Normalized Routing Load* yang lebih baik dibandingkan dengan protokol AOMDV.

C. Skenario 100 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata *Normalized Routing Load* yang dijabarkan pada tabel 5.9

Tabel 5.9 Hasil Pengujian *Normalized Routing Load* Pada Skenario 100 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	100	0,410	0,593	0,684	0,689	0,682	0,785
AOMDV		4,559	8,139	6,920	8,153	6,952	8,136



Gambar 5.9 Hasil Pengujian *Normalized Routing Load* Pada Skenario 100 *Node*

Pada Gambar 5.9 merupakan grafik perbandingan hasil rata-rata *Normalized Routing Load* pada protokol AODV dan AOMDV dengan skenario pengujian 100 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node blackhole.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 0,410 untuk 0 *node* penyerang, 0,593 untuk 2 *node* penyerang, 0,684 untuk 4 *node* penyerang, 0,689

untuk 6 *node* penyerang, 0,682 untuk 8 *node* penyerang, 0,785 untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 4,559 untuk 0 *node* penyerang, 8,139 untuk 2 *node* penyerang, 6,920 untuk 4 *node* penyerang, 8,153 untuk 6 *node* penyerang 6,952 untuk 8 *node* penyerang, 8,136 untuk 10 *node* penyerang.

Dari hasil pengujian berdasarkan skenario 100 *node* menunjukkan bahwa protokol AODV memiliki nilai rata-rata *Normalized Routing Load* yang lebih baik dibandingkan dengan protokol AOMDV.

5.1.2 Hasil Pengujian Pengaruh Serangan *DDoS* Pada Protokol AODV dan AOMDV

Pada sub bab ini terdapat pen penjelasan mengenai perbandingan hasil kinerja dari protokol AODV dan AOMDV dengan serangan *DDoS.*

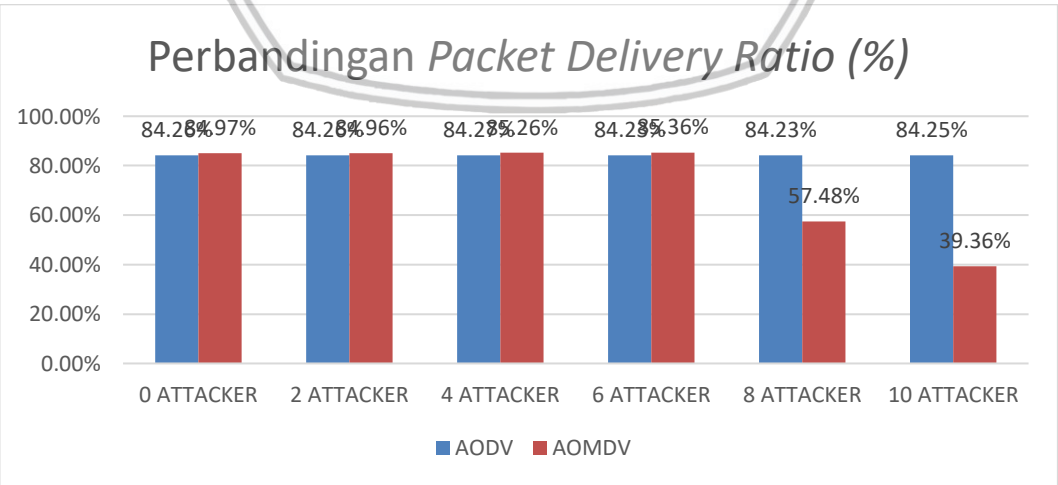
5.1.2.1 Packet Delivery Ratio

A. Skenario 20 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 20 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.10

Tabel 5.10 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 20 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	20	84,26%	84,26%	84,27%	84,23%	84,22%	84,25%
AOMDV		84,97%	84,96%	85,26%	85,36%	57,48%	39,36%



Gambar 5.10 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 20 *Node*

Pada Gambar 5.10 merupakan grafik perbandingan hasil *packet delivery ratio* pada protokol AODV dan AOMDV dengan skenario pengujian 20 *node* dengan 0,

2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 84,26% untuk 0 *node* penyerang, 84,26% untuk 2 *node* penyerang, 84,27% untuk 4 *node* penyerang, 84,23% untuk 6 *node* penyerang, 84,22% untuk 8 *node* penyerang, 84,25% untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian menunjukkan 84,97% untuk 0 *node* penyerang, 84,96% untuk 2 *node* penyerang, 85,26% untuk 4 *node* penyerang, 85,36% untuk 6 *node* penyerang, 57,48% untuk 8 *node* penyerang, 39,36% untuk 10 *node* penyerang.

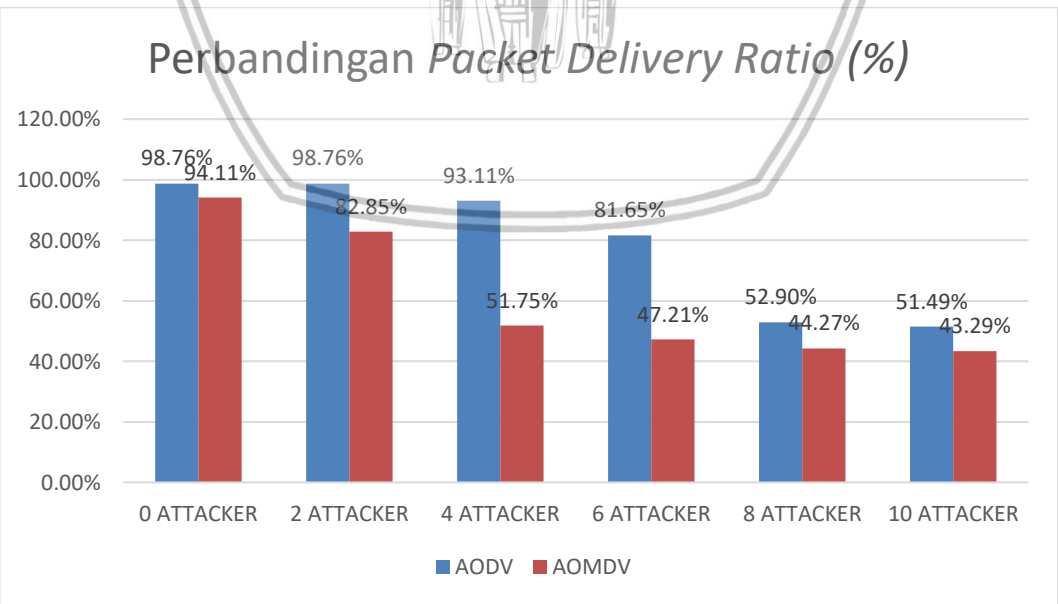
Dari hasil pengujian berdasarkan skenario 20 *node* menunjukkan bahwa kedua protokol hampir memiliki nilai rata-rata *Packet Delivery Ratio* yang sama. Semakin banyak *node* penyerang yang meyerang ke dalam sistem jaringan maka semakin turun juga nilai *Packet Delivery*.

B. Skenario 50 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 50 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.11

Tabel 5.11 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 50 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	50	98,76%	98,76%	93,11%	81,65%	52,90%	51,49%
AOMDV		94,11%	82,85%	51,75%	47,21%	44,27%	43,29%



Gambar 5.11 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 50 *Node*

Pada Gambar 5.11 merupakan grafik perbandingan hasil *packet delivery ratio* pada protokol AODV dan AOMDV dengan skenario pengujian 50 *node* dengan 0,

2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 98,76% untuk 0 *node* penyerang, 98,76% untuk 2 *node* penyerang, 93,11% untuk 4 *node* penyerang, 81,65% untuk 6 *node* penyerang, 52,90% untuk 8 *node* penyerang, 51,49% untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian menunjukkan 94,11% untuk 0 *node* penyerang, 82,85% untuk 2 *node* penyerang, 51,75% untuk 4 *node* penyerang, 47,21% untuk 6 *node* penyerang, 44,27% untuk 8 *node* penyerang, 43,29% untuk 10 *node* penyerang.

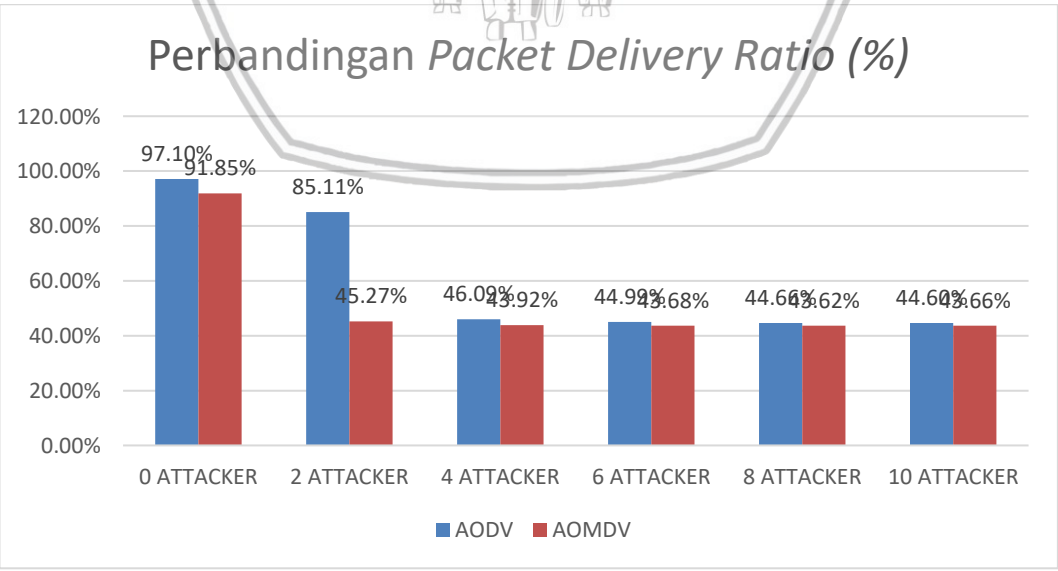
Dari hasil pengujian berdasarkan skenario 50 *node* menunjukkan bahwa protokol AODV memiliki rata-rata *packet delivery ratio* yang lebih baik dibandingkan protokol AOMDV, hal ini disebabkan karena penyerangan menggunakan skema *flooding attack* sangat merugikan protokol AOMDV karena *routing overhead* akan menjadi lebih tinggi dari protokol AODV.

C. Skenario 100 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.12

Tabel 5.12 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 100 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	100	97,10%	85,11%	46,09%	44,99%	44,66%	44,60%
AOMDV		91,85%	45,27%	43,92%	43,68%	43,62%	43,66%



Gambar 5.12 Hasil Pengujian *Packet Delivery Ratio* Pada Skenario 100 *Node*

Pada Gambar 5.12 merupakan grafik perbandingan hasil *packet delivery ratio* pada protokol AODV dan AOMDV dengan skenario pengujian 50 *node* dengan 0,

2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 97,10% untuk 0 *node* penyerang, 85,11% untuk 2 *node* penyerang, 46,09% untuk 4 *node* penyerang, 44,99% untuk 6 *node* penyerang, 44,66% untuk 8 *node* penyerang, 44,60% untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian menunjukkan 91,85% untuk 0 *node* penyerang, 45,27% untuk 2 *node* penyerang, 43,92% untuk 4 *node* penyerang, 43,68% untuk 6 *node* penyerang, 43,62%untuk 8 *node* penyerang, 43,66% untuk 10 *node* penyerang.

Dari hasil pengujian berdasarkan skenario 100 *node* menunjukkan bahwa protokol AODV memiliki rata-rata *packet delivery ratio* yang lebih baik dibandingkan protokol AOMDV, hal ini disebabkan karena penyerangan menggunakan skema *flooding attack* sangat merugikan protokol AOMDV karena *routing overhead* akan menjadi lebih tinggi dari protokol AODV, sehingga beban pada jaringan yang menggunakan protokol AOMDV menjadi lebih tinggi yang memungkinkan banyak *packet data* yang tidak sampai pada *node* tujuan.

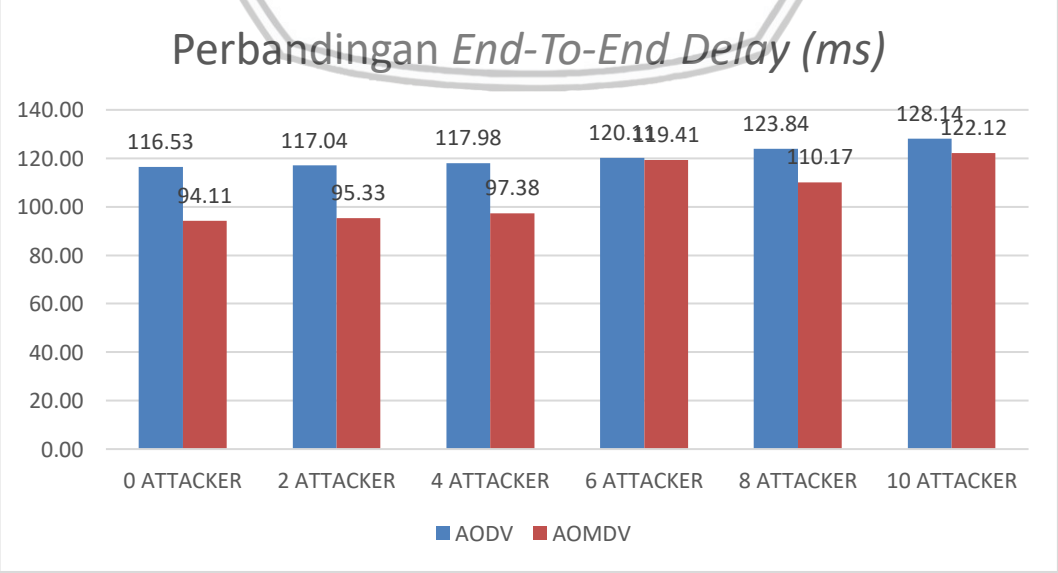
5.1.2.2 End-To-End Delay

A. Skenario 20 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 20 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *end-to-end delay* yang dijabarkan pada tabel 5.13

Tabel 5.13 Hasil Pengujian *End-To-End Delay* Pada Skenario 20 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	20	116,53	117,04	117,98	120,11	123,83	128,14
AOMDV		94,11	95,33	97,38	119,41	110,17	122,12



Gambar 5.13 Hasil Pengujian *End-To-End Delay* Pada Skenario 20 *Node*

Pada Gambar 5.13 merupakan grafik perbandingan hasil rata-rata *End-to-End Delay* pada protokol AODV dan AOMDV dengan skenario pengujian 20 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 116,53 ms untuk 0 *node* penyerang, 117,04 ms untuk 2 *node* penyerang, 117,98 ms untuk 4 *node* penyerang, 120,11 ms untuk 6 *node* penyerang, 123,83 ms untuk 8 *node* penyerang, 128,14 ms untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 94,11 ms untuk 0 *node* penyerang, 95,33 ms untuk 2 *node* penyerang, 97,38 ms untuk 4 *node* penyerang, 119,41 ms untuk 6 *node* penyerang, 110,17 ms untuk 8 *node* penyerang, 122,12 ms untuk 10 *node* penyerang.

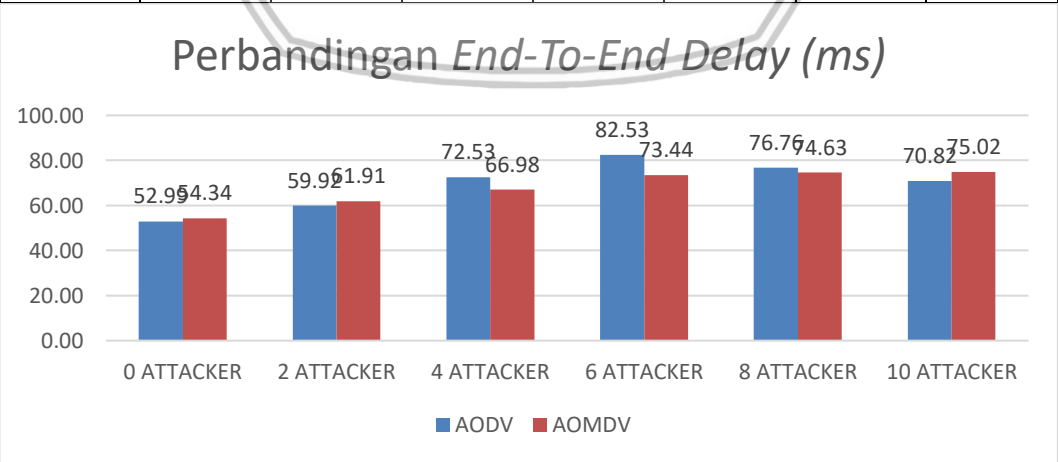
Dari hasil pengujian berdasarkan skenario 20 *node* menunjukkan bahwa protokol AOMDV memiliki nilai rata-rata *End-to-End Delay* yang lebih baik dibandingkan dengan protokol AODV. Hal ini disebabkan karena protokol AOMDV memelihara sejumlah jalur alternatif menuju *node* tujuan sehingga dapat mengurangi waktu *end-to-end delay* dalam proses pencarian jalur yang sewaktu-waktu dapat berubah. Sedangkan pada protokol AODV tidak memiliki algoritme untuk mengetahui jalur alternatif sehingga *end-to-end delay* menjadi lebih tinggi.

B. Skenario 50 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 50 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *end-to-end delay* yang dijabarkan pada tabel 5.14

Tabel 5.14 Hasil Pengujian *End-To-End Delay* Pada Skenario 50 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	50	52,99	59,92	72,53	82,53	76,76	70,82
AOMDV		54,34	61,91	66,98	73,44	74,63	75,02



Gambar 5.14 Hasil Pengujian *End-To-End Delay* Pada Skenario 50 *Node*

Pada Gambar 5.14 merupakan grafik perbandingan hasil rata-rata *End-to-End Delay* pada protokol AODV dan AOMDV dengan skenario pengujian 50 *node*

dengan 0, 2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 52,99 ms untuk 0 *node* penyerang, 59,92 ms untuk 2 *node* penyerang, 72,53 ms untuk 4 *node* penyerang, 82,53 ms untuk 6 *node* penyerang, 76,76 ms untuk 8 *node* penyerang, 70,82 ms untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 54,34 ms untuk 0 *node* penyerang, 61,91 ms untuk 2 *node* penyerang, 66,98 ms untuk 4 *node* penyerang, 73,44 ms untuk 6 *node* penyerang, 74,63 ms untuk 8 *node* penyerang, 75,02 ms untuk 10 *node* penyerang.

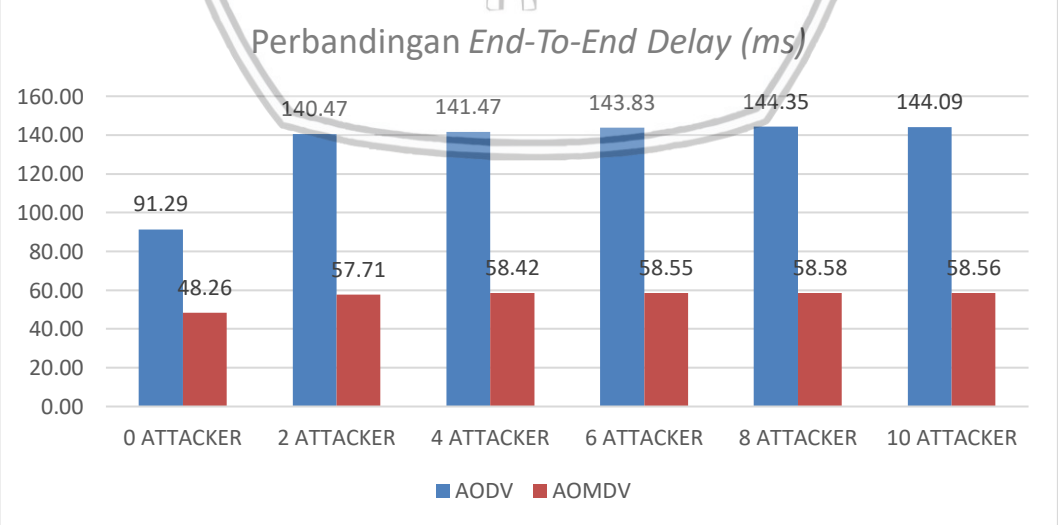
Dari hasil pengujian berdasarkan skenario 50 *node* menunjukkan bahwa protokol AOMDV memiliki nilai rata-rata *End-to-End Delay* yang lebih baik dibandingkan dengan protokol AODV. Hal ini disebabkan karena protokol AOMDV memelihara sejumlah jalur alternatif menuju *node* tujuan sehingga dapat mengurangi waktu *end-to-end delay* dalam proses pencarian jalur yang sewaktu-waktu dapat berubah. Sedangkan pada protokol AODV tidak memiliki algoritme untuk mengetahui jalur alternatif sehingga *end-to-end delay* menjadi lebih tinggi.

C. Skenario 100 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *end-to-end delay* yang dijabarkan pada tabel 5.15

Tabel 5.15 Hasil Pengujian *End-To-End Delay* Pada Skenario 100 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	100	91,29	140,47	141,47	143,82	144,35	144,09
AOMDV		48,26	57,71	58,42	58,55	58,58	58,56



Gambar 5.15 Hasil Pengujian *End-To-End Delay* Pada Skenario 100 *Node*

Pada Gambar 5.15 merupakan grafik perbandingan hasil rata-rata *End-to-End Delay* pada protokol AODV dan AOMDV dengan skenario pengujian 100 *node*

dengan 0, 2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 91,29 ms untuk 0 *node* penyerang, 140,47 ms untuk 2 *node* penyerang, 141,47 ms untuk 4 *node* penyerang, 143,82 ms untuk 6 *node* penyerang, 144,35 ms untuk 8 *node* penyerang, 144,09 ms untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 48,26 ms untuk 0 *node* penyerang, 57,71 ms untuk 2 *node* penyerang, 58,42 ms untuk 4 *node* penyerang, 58,55 ms untuk 6 *node* penyerang, 58,58 ms untuk 8 *node* penyerang, 58,56 ms untuk 10 *node* penyerang.

Dari hasil pengujian berdasarkan skenario 100 *node* menunjukkan bahwa protokol AOMDV memiliki nilai rata-rata *End-to-End Delay* yang lebih baik dibandingkan dengan protokol AODV. Hal ini disebabkan karena protokol AOMDV memelihara sejumlah jalur alternatif menuju *node* tujuan sehingga dapat mengurangi waktu *end-to-end delay* dalam proses pencarian jalur yang sewaktu-waktu dapat berubah. Sedangkan pada protokol AODV tidak memiliki algoritme untuk mengetahui jalur alternatif sehingga *end-to-end delay* menjadi lebih tinggi.

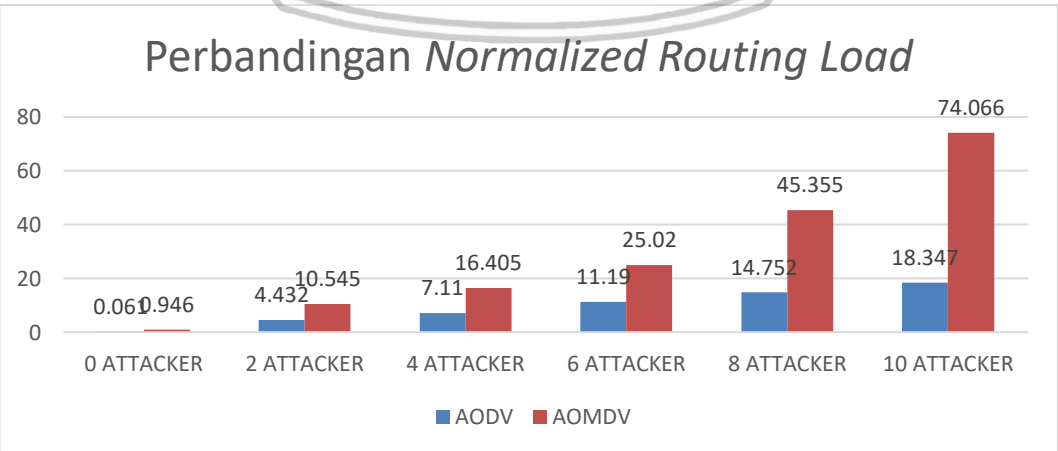
5.1.2.3 Normalized Routing Load

A. Skenario 20 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 20 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *normalized routing load* yang dijabarkan pada tabel 5.16

Tabel 5.16 Hasil Pengujian *Normalized Routing Load* Pada Skenario 20 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	20	0,061	4,432	7,110	11,190	14,752	18,347
AOMDV		0,946	10,545	16,405	25,020	45,355	74,066



Gambar 5.16 Hasil Pengujian *Normalized Routing Load* Pada Skenario 20 *Node*

Pada Gambar 5.16 merupakan grafik perbandingan hasil rata-rata *Normalized Routing Load* pada protokol AODV dan AOMDV dengan skenario pengujian 20 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 0,061 untuk 0 *node* penyerang, 4,432 untuk 2 *node* penyerang, 7,110 untuk 4 *node* penyerang, 11,190 untuk 6 *node* penyerang, 14,752 untuk 8 *node* penyerang, 18,347 untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 0,946 untuk 0 *node* penyerang, 10,545 untuk 2 *node* penyerang, 16,405 untuk 4 *node* penyerang, 25,020 untuk 6 *node* penyerang 45,355 untuk 8 *node* penyerang, 74,066 untuk 10 *node* penyerang.

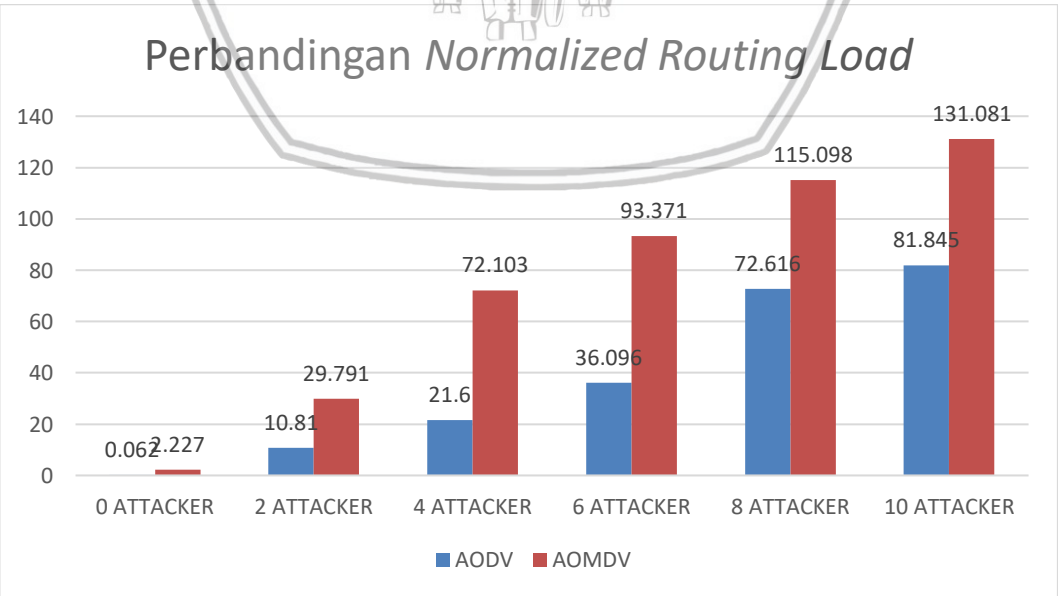
Dari hasil pengujian berdasarkan skenario 20 *node* menunjukkan bahwa protokol AODV memiliki nilai rata-rata *Normalized Routing Load* yang lebih baik dibandingkan dengan protokol AOMDV.

B. Skenario 50 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 50 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *normalized routing load* yang dijabarkan pada tabel 5.17

Tabel 5.17 Hasil Pengujian *Normalized Routing Load* Pada Skenario 50 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	50	0,062	10,810	21,600	36,096	72,616	81,845
AOMDV		2,227	29,791	72,103	93,371	115,098	131,081



Gambar 5.17 Hasil Pengujian *Normalized Routing Load* Pada Skenario 50 *Node*

Pada Gambar 5.17 merupakan grafik perbandingan hasil rata-rata *Normalized Routing Load* pada protokol AODV dan AOMDV dengan skenario pengujian 50 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 0,062 untuk 0 *node* penyerang, 10,810 untuk 2 *node* penyerang, 21,600 untuk 4 *node* penyerang, 36,096 untuk 6 *node* penyerang, 72,616 untuk 8 *node* penyerang, 81,845 untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 2,227 untuk 0 *node* penyerang, 29,791 untuk 2 *node* penyerang, 72,103 untuk 4 *node* penyerang, 93,371 untuk 6 *node* penyerang 115,098 untuk 8 *node* penyerang, 131,081 untuk 10 *node* penyerang.

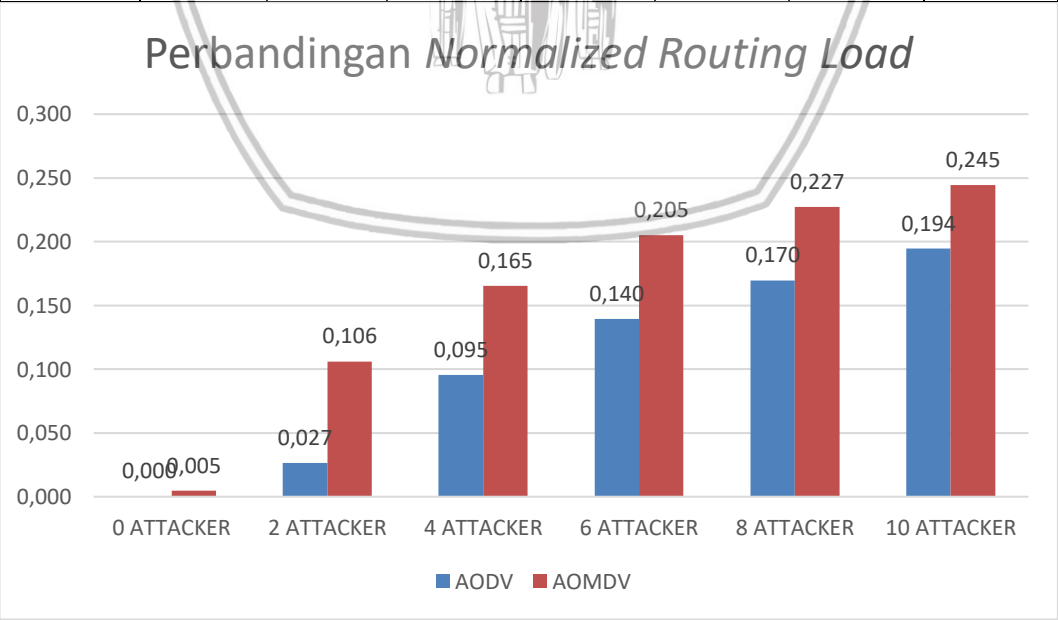
Dari hasil pengujian berdasarkan skenario 50 *node* menunjukkan bahwa protokol AODV memiliki nilai rata-rata *Normalized Routing Load* yang lebih baik dibandingkan dengan protokol AOMDV.

C. Skenario 100 *Node Dengan Variasi 0, 2, 4, 6, 8, 10 Node Penyerang*

Berdasarkan pengujian yang telah dilakukan dalam skenario 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh nilai rata-rata hasil *normalized routing load* yang dijabarkan pada tabel 5.18

Tabel 5.18 Hasil Pengujian *Normalized Routing Load* Pada Skenario 100 *Node*

Protokol Routing	Jumlah *Node*	Jumlah *Node* Penyerang					
		0	2	4	6	8	10
AODV	100	0,410	26,534	95,375	139,573	169,674	194,473
AOMDV		4,559	106,150	165,201	205,281	227,321	244,534



Gambar 5.18 Hasil Pengujian *Normalized Routing Load* Pada Skenario 100 *Node*

Pada Gambar 5.18 merupakan grafik perbandingan hasil rata-rata *Normalized Routing Load* pada protokol AODV dan AOMDV dengan skenario

pengujian 100 *node* dengan 0, 2, 4, 6, 8, 10 penyerang *node DDoS.* Dari hasil pengujian didapatkan hasil pada protokol AODV sebesar 0,410 untuk 0 *node* penyerang, 26,534 untuk 2 *node* penyerang, 95,375 untuk 4 *node* penyerang, 139,573 untuk 6 *node* penyerang, 169,674 untuk 8 *node* penyerang, 194,473 untuk 10 *node* penyerang. Sedangkan pada protokol AOMDV hasil pengujian 4,559 untuk 0 *node* penyerang, 106,150 untuk 2 *node* penyerang, 165,201 untuk 4 *node* penyerang, 205,281 untuk 6 *node* penyerang 227,321 untuk 8 *node* penyerang, 244.534 untuk 10 *node* penyerang.

Dari hasil pengujian berdasarkan skenario 100 *node* menunjukkan bahwa protokol AODV memiliki nilai rata-rata *Normalized Routing Load* yang lebih baik dibandingkan dengan protokol AOMDV. Hal ini disebabkan karena protokol AODV membuang semua paket *RREQ* yang terduplikasi saat melakukan pencarian jalur dari *node* sumber ke *node* tujuan, berbeda dengan AOMDV yang menyimpan semua *RREQ* yang digunakan sebagai informasi cadangan saat jalur pengiriman data rusak.

5.2 Analisis Hasil Pengujian

5.2.1 Analisis Pengaruh Serangan *Blackhole* Pada Protokol AODV dan AOMDV

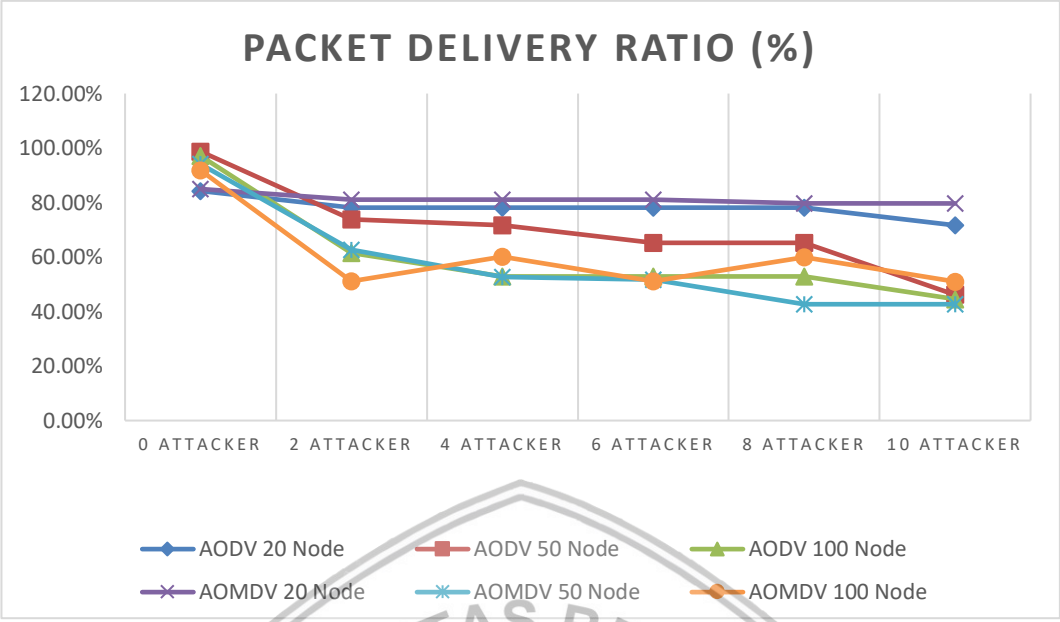
Pada subbab ini terdapat penjelesan analisis mengenai hasil pengujian yang telah dilakukan.

5.2.1.1 Packet Delivery Ratio

Berdasarkan pengujian yang telah dilakukan dalam skenario 20, 50, dan 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh perbandingan nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.19

Tabel 5.19 Perbandingan *Packet Delivery Ratio* Dengan Serangan *Blackhole*

Jumlah Node	Protokol Routing	Jumlah *Node* Penyerang						Rata-Rata
		0	2	4	6	8	10	
20	AODV	84,26%	78,13%	78,13%	78,13%	78,13%	71,65%	78,07%
50		98,76%	73,82%	71,63%	65,29%	65,29%	46,15%	70,16%
100		97,10%	61,43%	52,95%	52,94%	52,94%	44,57%	60,32%
20	AOMDV	84.97%	81.02%	81,11%	81,11%	79,69%	79,68%	80,40%
50		94,11%	62,68%	52,68%	51,68%	42,69%	42,69%	57,76%
100		91,85%	51,04%	60,19%	51,09%	59,99%	51,00%	60,86%



Gambar 5.19 Perbandingan *Packet Delivery Ratio* Dengan Serangan *Blackhole*

Gambar 5.19 menunjukkan perbandingan hasil nilai *packet delivery ratio* yang telah dilakukan pada pengujian dengan skenario 20, 50, dan 100 *node,* juga variasi serangan sebanyak 0, 2, 4, 6, 8, 10 *node blackhole* pada masing-masing protokol *routing.* Secara umum kinerja kedua protokol terlihat menurun ketika terdapat serangan *blackhole,* dan semakin bertambahnya *node* pada jaringan maka semakin menurun juga nilai *packet delivery ratio* kedua protokol.

Pengujian dengan kepadatan 20 *node* pada kedua protokol dengan serangan *blackhole* terlihat tidak memiliki dampak yang cukup besar pada penurunan nilai *packet delivery ratio*, hal ini disebabkan karena presentase paket data diambil alih oleh *node blackhole* menjadi lebih kecil. Sedangkan pada kepadatan 50 *node,* kinerja kedua protokol juga menurun seiring dengan bertambahnya kepdatan jumlah *node* dan *node blackhole* pada jaringan. Pada kepadatan 100 *node* hasil nilai *packet delivery ratio* kedua protokol juga menurun dengan signifikan.

Berdasarkan Tabel 5.19 dan Gambar 5.19 dapat ditarik kesimpulan bahwa serangan *blackhole* memengaruhi kinerja kedua protokol yang ditandai dengan *packet delivery ratio* yang semakin menurun seiring dengan bertambahnya jumlah kepadatan *node* dan jumlah *node blackhole.* Hal ini disebabkan oleh sifat kedua protokol yang tidak memiliki mekanisme untuk mendeteksi adanya serangan *blackhole*. Serangan *blackhole* bekerja dengan mengirimkan paket *routing* palsu yang menyatakan bahwa *node blackhole* adalah *node* tujuan dan memiliki jalur paling pendek sehingga pada akhirnya paket akan di *drop* saat sampai pada *node blackhole*.

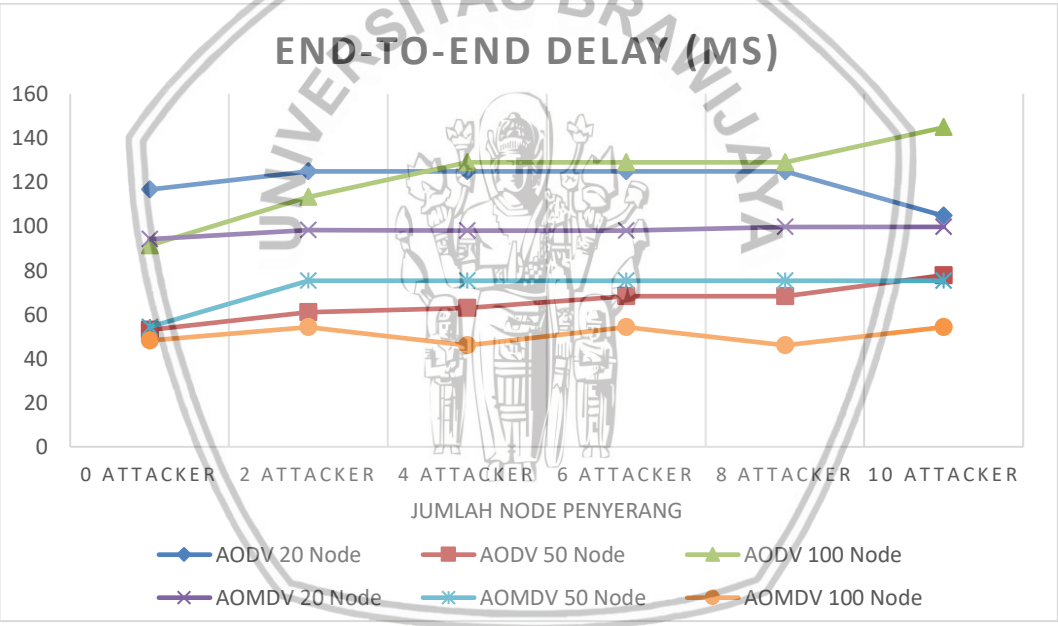
5.2.1.2 End-To-End Delay

Berdasarkan pengujian yang telah dilakukan dalam skenario 20, 50, dan 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh

perbandingan nilai rata-rata hasil *end-to-end delay* yang dijabarkan pada tabel 5.20

Tabel 5.20 Perbandingan *End-To-End Delay* Dengan Serangan *Blackhole*

Jumlah Node	Protokol Routing	Jumlah *Node* Penyerang						Rata-Rata
		0	2	4	6	8	10	
20	AODV	116,53	124,83	124,83	124,83	124,83	104,65	120,08
50		52,99	61,07	62,93	68,19	68,19	77,79	65,19
100		91,29	113,26	129,00	129,00	129,00	144,76	122,72
20	AOMDV	94,11	98,17	98,07	98,07	99,63	99,62	97,95
50		54,34	75,27	75,27	75,27	75,27	75,26	71,78
100		48,26	54,21	45,97	54,17	46,13	54,26	50,50



Gambar 5.20 Perbandingan *End-To-End Delay* Dengan Serangan *Blackhole*

Gambar 5.20 menunjukkan perbandingan hasil nilai *end-to-end delay* yang telah dilakukan pada pengujian dengan skenario 20, 50, dan 100 *node,* juga variasi serangan sebanyak 0, 2, 4, 6, 8, 10 *node blackhole* pada masing-masing protokol *routing.* Secara umum nilai rata-rata *end-to-end delay* pada protokol AOMDV lebih baik dibandingkan protokol AODV karena sifat *multipath* pada protokol AOMDV memengaruhi hasil nilai *end-to-end delay.*

Pengujian pada kepadatan 20, 50, dan 100 *node* menunjukkan bahwa *node blackhole* memengaruhi nilai *end-to-end delay* pada kedua protokol. Banyak atau tidaknya *node blackhole* tidak begitu memengaruhi nilai *end-to-end delay.* Dari Tabel 5.20 terlihat bahwa kinerja protokol AOMDV lebih baik dibandingkan

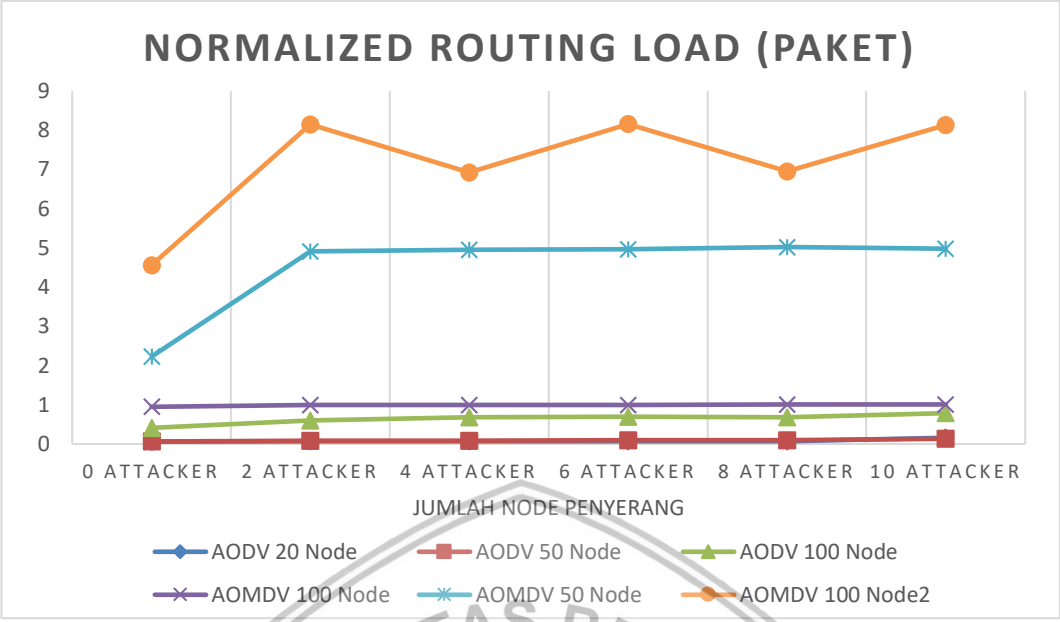
dengan kinerja protokol AODV. pada protokol AODV nilai *end-to-end delay* paling optimal terlihat pada skenario dengan 50 *node*, jika *node* terlalu sedikit atau banyak maka kinerja protokol AODV cenderung berkurang yang mengakibatkan *end-to-end delay* menjadi naik. Pada protokol AOMDV nilai *end-to-end delay* cenderung baik pada semua skenario yang dilakukan, hal ini disebabkan karena sifat *multipath* protokol AOMDV yang akan menyimpan seluruh jalur alternatif dari *node* sumber menuju *node* tujuan jika sewaktu-waktu terdapat jalur yang rusak, maka dari itu nilai *end-to-end delay* pada protokol AOMDV cenderung lebih kecil.

5.2.1.3 Normalized Routing Load

Berdasarkan pengujian yang telah dilakukan dalam skenario 20, 50, dan 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh perbandingan nilai rata-rata hasil *normalized routing load* yang dijabarkan pada tabel 5.21

Tabel 5.21 Perbandingan *Normalized Routing Load* Dengan Serangan *Blackhole*

Jumlah Node	Protokol Routing	Jumlah *Node* Penyerang						Rata-Rata
		0	2	4	6	8	10	
20	AODV	0,061	0,065	0,065	0,064	0,064	0,158	0,08
50		0,062	0,087	0,087	0,092	0,092	0,131	0,09
100		0,410	0,593	0,684	0,689	0,682	0,785	0,64
20	AOMDV	0,946	0,991	0,990	0,990	1,007	1,007	0,99
50		2,227	4,912	4,950	4,957	5,020	4,978	4,51
100		4,559	8,139	6,920	8,153	6,952	8,136	7,14



Gambar 5.21 Perbandingan *Normalized Routing Load* Dengan Serangan *Blackhole*

Gambar 5.21 menunjukkan perbandingan hasil nilai *normalized routing laod* yang telah dilakukan pada pengujian dengan skenario 20, 50, dan 100 *node,* juga variasi serangan sebanyak 0, 2, 4, 6, 8, 10 *node blackhole* pada masing-masing protokol *routing.* Secara umum nilai *normalized routing* protokol AODV cenderung lebih kecil dibandingkan dengan protokol AOMDV.

Berdasarkan hasil pengujian dengan kepadatan 20, 50, dan 100 *node* dan *node blackhole* sebanyak 0, 2, 4, 6, 8, 10 pada masing-masing protokol menunjukkan bahwa nilai *normalized routing load* pada protokol AODV lebih baik dibandingkan dengan protokol AOMDV. Hal ini disebabkan karena pada proses *routing discovery* jumlah paket *routing* yang dihasilkan oleh protokol AOMDV lebih banyak dibandingkan dengan protokol AODV. Tetapi kenaikkan nilai *normalized routing load* dengan serangan *blackhole* tidak lebih parah dari serangan DDoS yang dilakukan pada pengujian kali ini.

5.2.2 Analisis Pengaruh Serangan DDoS Pada Protokol AODV dan AOMDV

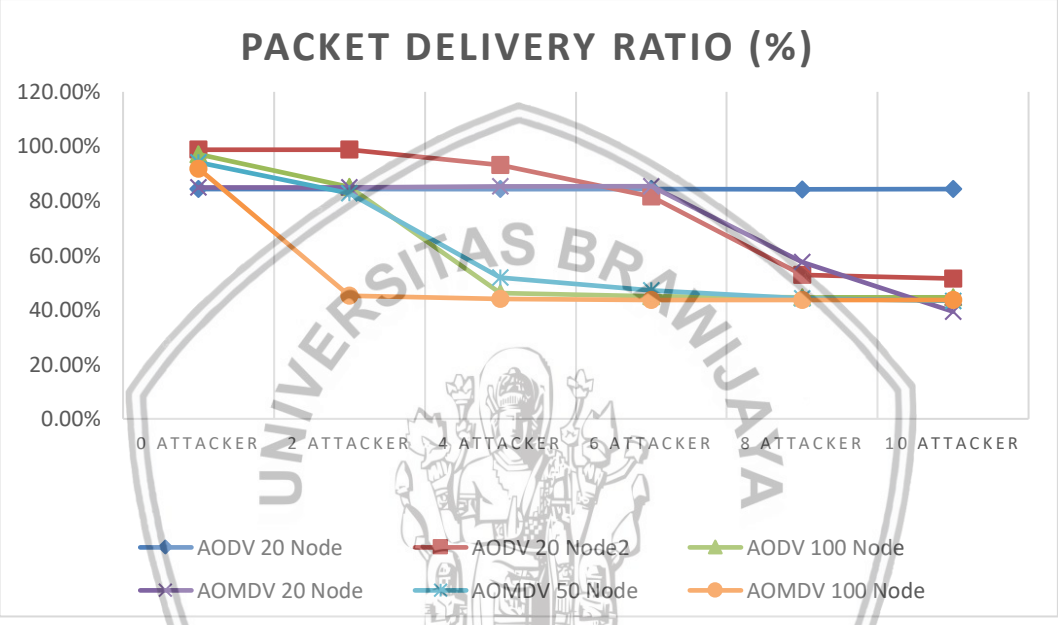
5.2.2.1 Packet Delivery Ratio

Berdasarkan pengujian yang telah dilakukan dalam skenario 20, 50, dan 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh perbandingan nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.22

Tabel 5.22 Perbandingan *Packet Delivery Ratio* Dengan Serangan *DDoS*

Jumlah Node	Protokol Routing	Jumlah *Node* Penyerang						Rata-Rata
		0	2	4	6	8	10	

20	AODV	84,26%	84,26%	84,27%	84,23%	84,22%	84,25%	84,25%
50		98,76%	98,76%	93,11%	81,65%	52,90%	51,49%	79,45%
100		97,10%	85,11%	46,09%	44,99%	44,66%	44,60%	60,43%
20	AOMDV	84,97%	84,96%	85,26%	85,36%	57,48%	39,36%	72,90%
50		94,11%	82,85%	51,75%	47,21%	44,27%	43,29%	60,58%
100		91,85%	45,27%	43,92%	43,68%	43,62%	43,66%	52,00%



Gambar 5.22 Perbandingan *Packet Delivery Ratio* Dengan Serangan *DDoS*

Gambar 5.22 menunjukkan perbandingan hasil nilai *end-to-ed delay* yang telah dilakukan pada pengujian dengan skenario 20, 50, dan 100 *node,* juga variasi serangan sebanyak 0, 2, 4, 6, 8, 10 *node blackhole* pada masing-masing protokol *routing.* Secara umum nilai *packet delivery ratio* protokol AODV lebih tinggi dibandingkan dengan protokol AOMDV. Terlihat dari rata-rata nilai *packet delivery ratio* pada Tabel 5.22 yang menunjukkan pada kepadatan berjumlah 20 *node* protokol AODV memiliki nilai 84,25% sedangkan protokol AOMDV memiliki nilai 72,90%. Pada kepadatan berjumlah 50 *node* protokol AODV memiliki nilai 79,45% sedangkan protokol AOMDV memiliki nilai 60,58%. Pada kepadatan berjumlah 100 *node* protokol AODV memiliki nilai 60,43% sedangkan protokol AOMDV memiliki nilai 52,00%.

Penurunan nilai *packet delivery ratio* pada kedua protokol sangat dipengaruhi oleh banyaknya *node* DDoS yang menyerang pada jaringan. Dampak dari serangan DDoS ini sangat memengaruhi kedua protokol, tetapi pada protokol AOMDV penurunan nilai *packet delivery ratio* terlihat lebih signifkan daripada protoko AODV. Hal ini disebabkan karena protokol AOMDV yang menyimpan dan memroses seluruh RREQ yang terdapat pada jaringan yang digunakan untuk mencari jalur alternatif, sehingga jaringan menjadi penuh dan banyak paket yang

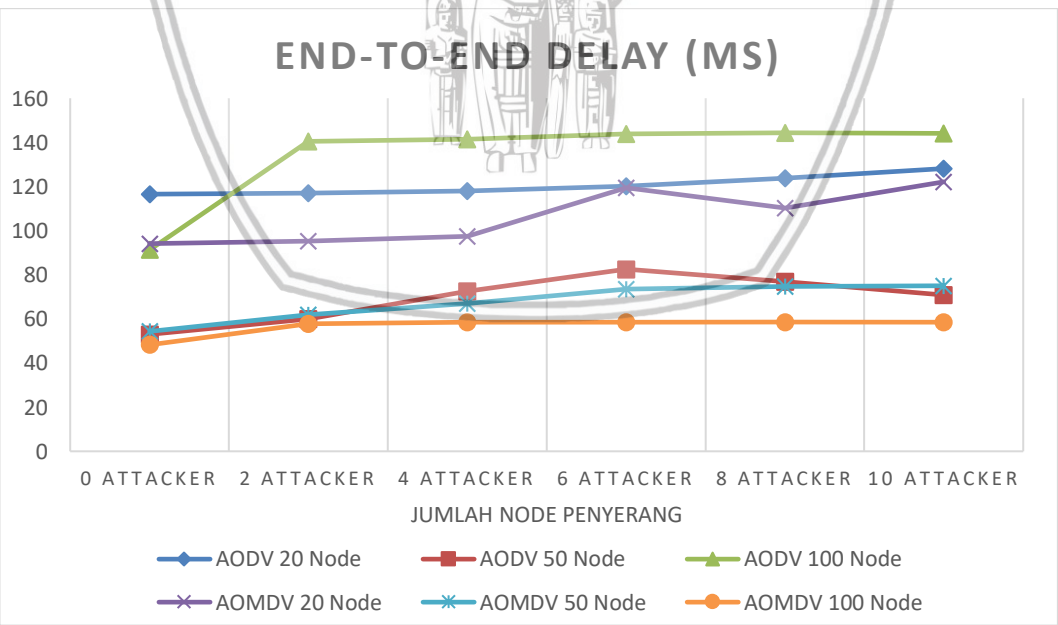
tidak sampai pada *node* tujuan. Berbeda dengan protokol AODV yang membuang paket RREQ yang tidak penting karena protokol AODV tidak memiliki mekanisme *multipath.* Sehingga jaringan tidak akan sepenuh protokol AOMDV. Berdasarkan hal tersebut maka dapat dilihat bahwa nilai *packet delivery routing* dengan derangan DDoS pada protokol AODV lebih baik daripada protokol AOMDV.

5.2.2.2 End-To-End Delay

Berdasarkan pengujian yang telah dilakukan dalam skenario 20, 50, dan 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh perbandingan nilai rata-rata hasil *packet delivery ratio* yang dijabarkan pada tabel 5.23

Tabel 5.23 Perbandingan *End-To-End Delay* Dengan Serangan *DDoS*

Jumlah Node	Protokol Routing	Jumlah *Node* Penyerang						Rata-Rata
		0	2	4	6	8	10	
20	AODV	116,53	117,04	117,98	120,11	123,83	128,14	120,605
50		52,99	59,92	72,53	82,53	76,76	70,82	69,258
100		91,29	140,47	141,47	143,82	144,35	144,09	134,25
20	AOMDV	94,11	95,33	97,38	119,41	110,17	122,12	106,42
50		54,34	61,91	66,98	73,44	74,63	75,02	67,72
100		48,26	57,71	58,42	58,55	58,58	58,56	56,68



Gambar 5.23 Perbandingan *End-To-End Delay* Dengan Serangan *DDoS*

Gambar 5.23 menunjukkan perbandingan hasil nilai *end-to-end delay* yang telah dilakukan pada pengujian dengan skenario 20, 50, dan 100 *node,* juga variasi serangan sebanyak 0, 2, 4, 6, 8, 10 *node* DDoS pada masing-masing protokol

routing. Secara umum nilai rata-rata *end-to-end delay* pada protokol AOMDV lebih baik dibandingkan protokol AODV karena sifat *multipath* pada protokol AOMDV memengaruhi hasil nilai *end-to-end delay.*

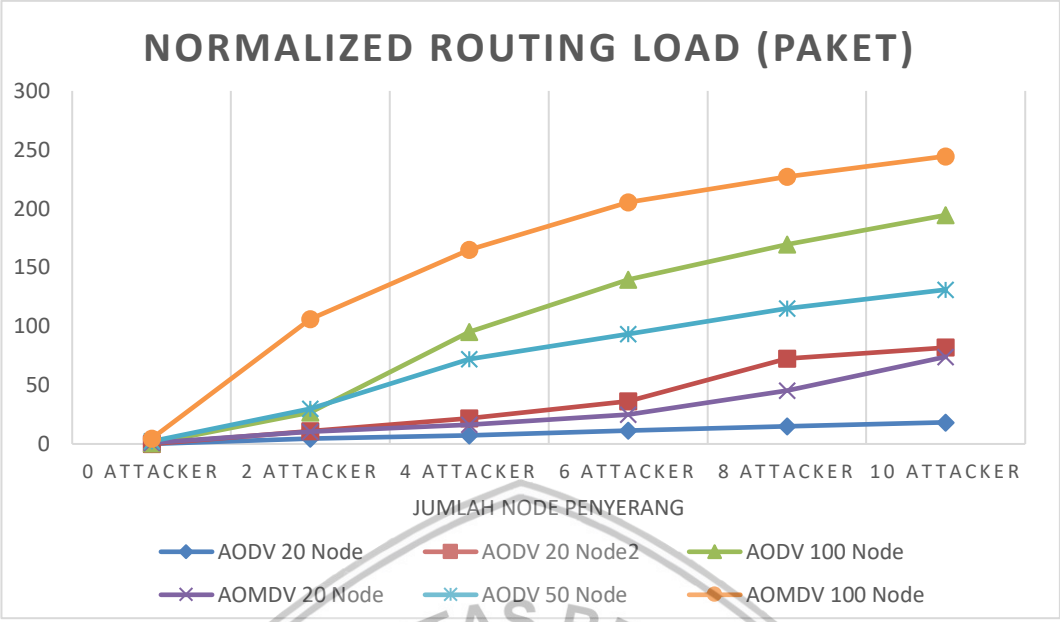
Pengujian pada kepadatan 20, 50, dan 100 *node* menunjukkan bahwa *node blackhole* memengaruhi nilai *end-to-end delay* pada kedua protokol. Semakin banyak *node* DDoS maka akan semakin bertambah nilai *end-to-end delay* kedua protokol. Dari Tabel 5.23 terlihat bahwa kinerja protokol AOMDV lebih baik dibandingkan dengan kinerja protokol AODV. pada protokol AODV nilai *end-to-end delay* paling optimal terlihat pada skenario dengan 50 *node*, jika *node* terlalu sedikit atau banyak maka kinerja protokol AODV cenderung berkurang yang mengakibatkan *end-to-end delay* menjadi naik. Pada protokol AOMDV nilai *end-to-end delay* cenderung baik pada semua skenario yang dilakukan, hal ini disebabkan karena sifat *multipath* protokol AOMDV yang akan menyimpan seluruh jalur alternatif dari *node* sumber menuju *node* tujuan jika sewaktu-waktu terdapat jalur yang rusak maka *delay* yang terjadi dapat lebih diminimalisir, maka dari itu nilai *end-to-end delay* pada protokol AOMDV cenderung lebih kecil.

5.2.2.3 Normalized Routing Load

Berdasarkan pengujian yang telah dilakukan dalam skenario 20, 50, dan 100 *node* dengan variasi 0, 2, 4, 6, 8, 10 *node* penyerang, maka diperoleh perbandingan nilai rata-rata hasil *normalized routing load* yang dijabarkan pada tabel 5.24

Tabel 5.24 Perbandingan *Normalized Routing Load* Dengan Serangan *DDoS*

Jumlah Node	Protokol Routing	Jumlah *Node* Penyerang						Rata-Rata
		0	2	4	6	8	10	
20	AODV	0,061	4,432	7,110	11,190	14,752	18,347	9,31
50		0,062	10,810	21,600	36,096	72,616	81,845	37,17
100		0,410	26,534	95,375	139,573	169,674	194,473	104,3
20	AOMDV	0,946	10,545	16,405	25,020	45,355	74,066	28,72
50		2,227	29,791	72,103	93,371	115,098	131,081	73,95
100		4,559	106,150	165,201	205,281	227,321	244,534	158,8



Gambar 5.24 Perbandingan *Normalized Routing Load* Dengan Serangan *DDoS*

Gambar 5.24 menunjukkan perbandingan hasil nilai *end-to-ed delay* yang telah dilakukan pada pengujian dengan skenario 20, 50, dan 100 *node*, juga variasi serangan sebanyak 0, 2, 4, 6, 8, 10 *node blackhole* pada masing-masing protokol *routing.* Secara umum nilai *normalized routing* protokol AODV cenderung lebih kecil dibandingkan dengan protokol AOMDV.

Berdasarkan hasil pengujian dengan kepadatan 20, 50, dan 100 *node* dan *node* DDoS sebanyak 0, 2, 4, 6, 8, 10 pada masing-masing protokol menunjukkan bahwa nilai *normalized routing load* pada protokol AODV lebih baik dibandingkan dengan protokol AOMDV. Hal ini disebabkan karena pada proses *routing discovery* jumlah paket *routing* yang dihasilkan oleh protokol AOMDV lebih banyak dibandingkan dengan protokol AODV karena sifat protokol AOMDV yang memroses seluruh paket *routing* untuk mencari jalur alternatif. Serangan DDoS lebih parah daripada serangan *blackhole*, hal ini disebabkan karena serangan DDoS akan membanjiri jaringan dengan paket RREQ yang membuat nilai *normalized routing load* akan semakin tinggi.

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil analisis perbandingan kinerja protocol AODV dan AOMDV dengan menggunakan serangan *Blackhole* dan *DDoS* maka didapatkan kesimpulan sebagai berikut.

1. Kinerja protokol AODV dan AOMDV terhadap serangan *blackhole* dan *DDoS* masing-masing protokol memiliki kelebihan dan kekurangan. Untuk kinerja protokol AODV memiliki nilai *End-to-End Delay* yang lebih tinggi dibandingkan dengan protokol AOMDV baik dengan serangan *blackhole* maupun serangan *DDoS.* Sedangkan nilai *Normalized Roting Load* pada protokol AOMDV lebih tinggi daripada protokol AODV baik dengan serangan *blackhole* atau *DDoS.* Untuk *packet delivery ratio* pada skenario dengan serangan *blackhole* kedua protokol menunjukkan kinerja yang kurang lebih sama, tetapi pada saat dilakuan skenario dengan serangan DDoS, protokol AOMDV menunjukkan hasil yang sedikit lebih buruk dibandingkan protokol AODV.

   Berdasarkan sleuruh skenario pengujian yang telah dilakukan terhadap kinerja protokol AODV dan AOMDV dengan parameter pengukuran *Packet Delivery Ratio, End-to-End Delay,* dan *Normalized Routing Load* maka dapat diambil analisis sebagai berikut:

   A. Pada pengujian parameter *Packet Delivery Ratio* dengan serangan *Blackhole,* dapat diambil kesimpulan bahwa *node blackhole* memengaruhi kinerja kedua protokol sehingga nilai *Packet Delivery Ratio* menurun, semakin banyak *node blackhole* pada jaringan maka semakin kecil nilai *packet delivery ratio* yang dihasilkan. Sedangkan pada serangan dengan *DDoS* semakin banyak *node* pada jaringan dan semakin banyak *node DDoS* maka nilai PDR kedua protokol akan menurun dengan drastis dikarenakan beban yang berlebihan pada jaringan akan membuat paket data terbuang/tidak sampai pada *node* tujuan, tetapi penurunan nilai PDR pada protokol AOMDV sedikit lebih signifikan daripada protokol AODV, hal ini dikarenakan *overload* pada jaringan membuat banyak paket data yang tidak sampai tujuan. Serangan DDoS dirasa memberikan dampak yang lebih besar dibandingkan dengan serangan *Blackhole*

   B. Pengujian dengan parameter *End-to-End* Delay dengan serangan *Blackhole* menunjukkan bahwa protokol AOMDV memiliki nilai *End-to-End Delay* yang lebih kecil dan lebih stabil daripada protokol AODV pada setiap skenario yang dijalankan. Banyak *node* pada jaringan sangat memengaruhi nilai dari *End-To-End Delay.* Sedangkan pada skenario dengan serangan *DDoS* juga menunjukkan bahwa nilai *End-To-End Delay* protokol AOMDV lebih kecil dan stabil dibandingkan dengan protokol AODV. Hal ini disebabkan karena protokol AOMDV memelihara sejumlah jalur alternatif menuju node tujuan sehingga dapat mengurangi waktu *end-to-end delay* dalam proses pencarian jalur yang sewaktu-waktu dapat berubah.

Sedangkan pada protokol AODV tidak memiliki algoritme untuk mengetahui jalur alternatif sehingga end-to-end delay menjadi lebih tinggi.

C. Pengujian dengan parameter *Normalized Routing Load* menunjukkan bahwa protokol AODV lebih optimal dibandingkan dengan protokol AOMDV baik dengan serangan *Blackhole* ataupun DDoS. Dikarenakan karena protokol AODV tidak menggunakan algoritme *multipath* dan membuang semua paket RREQ yang terduplikasi saat melakukan pencarian jalur dari *node* sumber ke *node* tujuan, berbeda dengan AOMDV yang menyimpan semua RREQ yang digunakan sebagai informasi cadangan sehingga saat jalur pengiriman data rusak. Hal ini menyebabkan protokol AOMDV memiliki lebih banyak paket *routing* sehingga nilai *Normalized Routing Load* menjadi cenderung tinggi dan kurang efisien. Serangan DDoS berdampak lebih besar dari pada serangan *Blackhole,* dikarenakan serangan DDoS akan membanjiri jaringan dengan paket RREQ yang membuat jaringan *overload* dan nilai *normalized routing load* naik

6.2 Saran

Saran yang dapat disampaikan oleh penulis untuk penelitian selanjutnya adalah.

1. Perlu dilakukan penelitian dengan menggunakan tipe protokol lain pada MANET seperti protokol *Proactive* dan protokol *Hybrid.*
2. Perlu dilakukan pengujian menggunakan serangan dengan tipe lain seperti *wormhole, sinkhole, greyhole* dll. Serta cara menanggulangi dan menghentikan serangan tersebut.
3. Menambah parameter pengujian dengan meninjau efisiensi energi tiap protokol *routing.*
4. Menambah variasi jumlah *node,* kecepatan, serta pergerakan *node* yang berbeda.

DAFTAR REFERENSI

Abolhasan, M., Wysocki, T. & Dutkiewicz, E., 2003. *A review of routing protocols for mobile ad hoc networks,* Australia: Telecommunication and Information Research Institute, University of Wollongong.

Dorri, A., Kamel, S. R. & Kheyrkhah, E., 2015. SECURITY CHALLENGES IN MOBILE AD HOC NETWORKS: A SURVEY. *International Journal of Computer Science & Engineering Survey (IJCSES),* 6(1).

Esmaili, H. A., Shoja, M. R. K. & Gharaee, H., 2011. Performance Analysis of AODV under Black Hole Attack through Use of OPNET Simulator. *World of Computer Science and Information Technology Journal (WCSIT),* 1(2), pp. 49-52.

Gorantala, K., 2006. *Routing Protocols in Mobile Ad-hoc Networks,* Sweden: Umea University Department of Computing Science .

Harahap, E. H., 2014. Analisis Performansi Protokol AODV (Ad Hoc On Demand Distance Vector) Dan DSR (Dynamic Source Routing) Terhadap Active Attack Pada MANET (Mobile Ad Hoc Network) Ditinjau Dari QoS (Quality of Service) Jaringan. *e-Proceeding of Engineering,* Volume 1, pp. 118-125.

Issariyakul, T., 2012. *Introduction to Network Simulator 2 (NS 2).* s.l.:Springer, Boston, MA.

Jayanti, V. N., 2014. Routing Protocols in MANET: Comparative Study. *International Journal of Computer Science and Mobile Computing,* 3(7), pp. 119-125.

Kannhavong, B., Nakayama, H., Nemoto, Y. & Kato, N., 2007. A Survey of Routing Attacks In Mobile Ad Hoc Networks. *IEEE Wireless Communications.*

Kumar, S., Goyal, M., Goyal, D. & Poonia, R. C., 2017. Routing Protocols and Security Issues in MANET. *2017 International Conference on Infocom Technologies and Unmanned Systems (ICTUS'2017),* pp. 818-824.

Marina, M. K. & Das, S. R., 2001. On-demand Multipath Distance Vector Routing in Ad Hoc Networks. *IEEE.*

Meher, R. & Ladhe, S., 2014. Review Paper on Flooding Attack in MANET. *International Journal of Engineering Research and Applications,* 4(1), pp. 39-46.

Moudni, H., Er-rouidi, M., Mouncif, H. & Hadadi, B. E., 2016. Performance analysis of AODV routing protocol in MANET under the influence of routing attacks. *2nd International Conference on Electrical and Information Technologies ICEIT'2016.*

Paulose, N. & Paulose, N., 2016. Comparison of On Demand Routing Protocols AODV with AOMDV. *International Journal of Science, Engineering and Technology Research (IJSETR),* 5(1), pp. 181-184.

Purba, D. U., 2018. Analisis Kinerja Protokol Ad Hoc On-Demand Distance Vector (AODV) dan Fisheye State Routing (FSR) pada Mobile Ad Hoc Network. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 2(7), pp. 2626-2636.

Putra, B. B. & Anggoro, R., 2016. Studi Kinerja Multipath AODV dengan Menggunakan Network simulator 2 (NS-2). *JURNAL TEKNIK ITS,* 5(2), pp. A652-A656.

Ramadhan, A. A., Abdurohman, M. & Putrada, A. G., 2017. *Analisis Perbandingan Pengujian Distributed Denial of Service (DDoS) dan Rushing Attack pada Jaringan UDP dengan Routing AODV,* Bandung: Telkom University.

Sharma, R. & Singla, B., 2015. Comparison of AOMDV With and Without Black Hole Attack. *International Journal Of Engineering And Computer Science,* 4(8), pp. 13892-13899.