TUGAS AKHIR – KI141502

Implementasi Pendeteksian dan Pencegahan Serangan Black Hole dengan Memanfaatkan Hop Count dan Neighbor Information pada Jaringan MANET

FANY AGRIANSYAH ROSYADA
NRP 5113 100 076

Dosen Pembimbing 1
Dr. Eng. RADITYO ANGGORO, S.Kom., M.Sc.

Dosen Pembimbing 2
Ir. F.X. ARUNANTO, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

TUGAS AKHIR – KI141502

Implementasi Pendeteksian dan Pencegahan Serangan Black Hole dengan Memanfaatkan Hop Count dan Neighbor Information pada Jaringan MANET

FANY AGRIANSYAH ROSYADA
NRP 5113 100 076

Dosen Pembimbing 1
Dr. Eng. RADITYO ANGGORO, S.Kom., M.Sc.

Dosen Pembimbing 2
Ir. F.X. ARUNANTO, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

FINAL PROJECT – KI141502

Implementation Detection and Prevention of Black Hole Utilizing Hop Count and Neighbor Information Method on MANET Network

FANY AGRIANSYAH ROSYADA
NRP 5113 100 076

Advisor 1
Dr. Eng. RADITYO ANGGORO, S.Kom., M.Sc.

Advisor 2
Ir. F.X. ARUNANTO, M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Implementasi Pendeteksian dan Pencegahan Serangan Black Hole dengan Memanfaatkan Hop Count dan Neighbor Information pada Jaringan MANET

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur Jaringan Komputer
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:
FANY AGRIANSYAH ROSYADA
NRP. 5113 100 076

Disetujui oleh Pembimbing Tugas Akhir:

Dr. Eng. Radityo Anggoro, S.Kom., M.Sc. .....
NIP: 198410162008121002 (pembimbing)

Ir. F.X. Arunanto, M.Sc. .....
NIP: 195701011983031004 (pembimbing)

SURABAYA
JANUARI, 2017

[Halaman ini sengaja dikosongkan]

Implementasi Pendeteksian dan Pencegahan Serangan Black Hole dengan Memanfaatkan Hop Count dan Neighbor Information pada Jaringan MANET

Nama Mahasiswa	**: Fany Agriansyah Rosyada**
NRP	**: 5113 100 076**
Jurusan	**: Teknik Informatika FTIf - ITS**
Dosen Pembimbing	**: 1. Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.**
	2. Ir. F.X. ARUNANTO, M.Sc.

ABSTRAK

Mobile Ad-Hoc Network (MANET) merupakan sebuah lingkungan jaringan yang terdiri dari beberapa node yang dinamis sehingga posisi dari masing-masing node dan informasi rute komunikasi antar node dapat berubah sewaktu-waktu. Hal ini memungkinkan adanya serangan yang terjadi pada suatu jaringan komunikasi. MANET tidak memiliki topologi dan posisi node yang pasti, sehingga sulit untuk mendeteksi node mana yang tidak berbahaya dan yang terpercaya untuk diakui sebagai rute yang dapat dilewati tanpa adanya paket drop.

Serangan yang dilakukan oleh node berbahaya ini (malicious node) sering disebut sebagai Black Hole Attack. Black Hole Attack memungkinkan tujuan pengiriman paket yang dikirimkan oleh node sumber diakuisisi oleh malicious node. Black hole ini akan mengatasnamakan dirinya sebagai node tujuan. Kemudian melakukan drop pada paket atau meneruskan ke black hole lain yang paket tidak akan pernah sampai ke node tujuan. Black hole akan melakukan pengiriman pesan Route Reply (RREP) kepada node sumber yang telah diakuisisi oleh Black Hole. Pada kondisi ini, node sumber tidak dapat membedakan antara node asli dan Malicious node. Rute yang telah dibangun ini merupakan rute yang tidak benar dan sulit untuk dideteksi. Sehingga node tujuan tidak akan pernah mendapat

paket/informasi yang dikirimkan oleh node sumber. Pada tugas akhir ini, akan dilakukan pendeteksian dan pencegahan black hole pada Jaringan MANET.

Metode yang digunakan adalah dengan memanfaatkan Hop Count dan Neighbour Information yang ada pada Ad-Hoc On-Demand Distance Vector (AODV). Informasi Hop Count terdapat pada pesan RREP. Sementara Neighbour Information dimiliki oleh setiap node, yang berfungsi untuk menjelaskan informasi daftar tetangga dari masing-masing node.

Tujuan pembuatan tugas akhir ini adalah untuk mencegah serangan black hole sehingga paket yang dikirimkan dari node sumber bisa tersampaikan ke node tujuan dan terhindar dari serangan Black Hole.

Kata Kunci: MANET, Network Simulator, NS2, AODV, Neighbor Information, Hop Count.

Implementation Detection and Prevention of Black Hole Utilizing Hop Count and Neighbor Information Method on MANET Network

Name : **Fany Agriansyah Rosyada**
NRP : **5113 100 076**
Major : **Informatics Engineering, IT Dept – ITS**
Advisor : **1. Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.**
2. Ir. F.X. ARUNANTO, M.Sc.

ABSTRACT

Mobile Ad-Hoc Network (MANET) is a network environment that consists of multiple dynamic nodes so the position of each node and route information can be changed at any time. This allows attacks to happen inside a communication network. MANET does not have exact topology and nodes position, making it difficult to detect which nodes are malicious and reliable to be recognized as a route that can be passed without any packet drop.

Attack by Malicious node is often referred as Black Hole Attack. Black Hole Attack allows the delivery destination of packets that sent by the source node got acquired by malicious node. These Black holes would recognize itself as the destination node. Then do packet drop or forward it to another black hole which makes the packet never make it to the destination node. The Black hole will deliver the Route Reply (RREP) Message to source node that has been acquired by the black hole. In this condition, the source node cannot distinguish between original node, and malicious node. The route that has been built is a fake route and difficult to detect. Which means the destination route will never receive any packet or information that sent by the source node. In this Final Project, I will do detection and prevention of black holes in MANET networks.

Method that used is the Hop Count and Neighbor Information Method. Hop Count Information included in RREP Message. While Neighbor Information owned by each node, that have function to explain neighbor list information from each node.

The purpose of making this Final Project is to prevent black hole attack so the packet sent from the source node can be delivered to the destination node and avoid the black hole attack.

Keywords: MANET, Network Simulator, NS2, AODV, Neighbor Information, Hop Count.

KATA PENGANTAR

Bismillahirrohmanirohim.

Alhamdulilahirabil'alamin, segala puji bagi Allah SWT, atas segala rahmat dan karunia-Nya yang tak terhingga sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

"Implementasi Pendeteksian dan Pencegahan Serangan Black Hole dengan Memanfaatkan Hop Count dan Neighbor Information pada Jaringan MANET".

Terselesaikannya Tugas Akhir ini tidak terlepas dari bantuan banyak pihak. Oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan sebesar-besarnya kepada pihak-pihak sebagai berikut.

1. Allah SWT, karena limpahan rahmat dan karunia-Nya lah penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Teknik Informatika ITS.
2. Orang tua penulis, yang tiada henti memberikan semangat, doa serta dukungan penuh kepada penulis selama ini sehingga dapat menyelesaikan Tugas Akhir ini dan perkuliahan di Teknik Informatika ITS Surabaya.
3. Adik penulis, Vony Agrilika Yuwanda yang telah memberikan dukungan dan doa kepada penulis untuk terus semangat dalam penyelesaian Tugas Akhir ini.
4. Bapak Dr. Eng. Radityo Anggoro S.Kom., M.Sc. selaku dosen pembimbing dari penulis yang telah memberikan bimbingan, dukungan, masukan, nasihat dan banyak arahan terhadap analisis dan pengujian kepada penulis dalam menyelesaikan Tugas Akhir ini.
5. Bapak Ir. F.X. ARUNANTO, M.Sc. selaku dosen pembimbing dari penulis yang telah memberikan

bimbingan dan arahan terhadap perancangan laporan dari program yang saya usulkan dari Tugas Akhir ini.

6. Bapak Hudan Studiawan, S.Kom.,M.Kom. selaku dosen wali dari penulis yang selalu memberi nasihat kepada penulis selama menjalani perkuliahan di Teknik Informatika ITS.
7. Bapak Dr. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Jurusan Teknik Informatika ITS.
8. Keluarga lab Mobile Innovation Studio, yang memberikan semangat, canda tawa, menemani menginap di lab selama penulis menjadi admin di lab tersebut.
9. Teman-teman bermain dota, yang senantiasa menemani bermain ketika penulis sedang stres.
10. Teman-teman T. Informatika ITS angkatan 2013 yang telah mendukung selama penulis menjadi Komting.
11. Pihak-pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis ke depannya. Penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum. Semoga Allah SWT. memberkati dan membalas semua kebaikan yang telah dilakukan

Surabaya, Januari 2017

Fany Agriansyah Rosyada

DAFTAR ISI

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

DAFTAR KODE SUMBER

BAB I
PENDAHULUAN

Bab ini memaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Komunikasi [1] merupakan salah satu hal yang penting dalam kehidupan sehari-hari. Dengan komunikasi, kita dapat bertukar pesan dengan orang lain, dapat mengungkapkan keinginan dengan orang lain, dapat berinteraksi dengan orang lain, dan yang paling penting bahwa komunikasi adalah kunci bahwa manusia adalah makhluk sosial yang selalu membutuhkan orang lain.

Perkembangan teknologi [2] yang begitu pesat membuat batasan-batasan komunikasi ini semakin tidak terbatas. Komunikasi tidak hanya sebatas pada komunikasi lisan atau verbal dan non-verbal serta bertatap muka dengan lawan bicara kita secara langsung. Perkembangan zaman di Indonesia saat ini sudah mulai memasuki era globalisasi dan modernisasi.

Dengan pentingnya komunikasi tersebut, manusia membutuhkan suatu teknologi yang dapat mengirim dan menerima berbagai macam informasi dengan cepat dan tepat. Ad-Hoc network merupakan sarana yang sesuai untuk dapat diterapkan untuk kasus mengirim dan menerima informasi. Yaitu yang merupakan sebuah jaringan radio bergerak tanpa bantuan infrastruktur tetap atau administrasi terpusat yang memungkinkan mereka untuk digunakan dengan mudah sebagai skalabilitas.

MANET memiliki karakteristik seperti media nirkabel yang tidak dapat diandalkan (link) yang digunakan untuk komunikasi antara host, topologi dan keanggotaan jaringan yang dapat berubah sewaktu-waktu, bandwidth terbatas, daya tahan baterai, masa hidup, dan kekuatan komputasi node.

Namun dalam setiap jaringan pasti terdapat serangan. Serangan yang terjadi sering menjadi keluhan di berbagai kalangan, baik individu maupun instansi atau kelompok. Ada banyak masalah keamanan yang telah dipelajari dalam beberapa tahun terakhir. Contohnya, snooping attacks, wormhole attacks, black hole attacks [10], routing table overflow and poisoning attacks, packet replication, denial of service (DoS) attacks, distributed DoS (DDoS) attacks, et cetera [11]. Terutama masalah routing adalah salah satu ancaman keamanan yang populer [12] seperti serangan black hole. Beberapa peneliti telah mengusulkan beberapa ide [13, 14, 15, 16] untuk memecahkan masalah keamanan ini namun tetap belum dapat mencegahnya dengan sempurna.

Ada dua jenis serangan black hole yaitu grouped black hole attack [17] dan single black hole attack. Serangan black hole yang sering dijumpai adalah serangan single black hole [17]. Serangan ini dapat menyamarkan diri sebagai node tujuan (destination node) sehingga paket yang dikirimkan ke node tujuan tidak dikirimkan dan tidak akan pernah sampai. Sehingga perlu dilakukan riset lebih untuk serangan single black hole ini yang dikarenakan riset yang sudah pernah ada belum dapat mencegahnya dengan sempurna dan masih sering terjadinya permasalahan ini. Karena serangan tersebut dapat membatalkan maupun memanipulasi suatu paket yang dikirim maupun yang diterima.

Metode pengiriman paket ada berbagai macam jenis. Salah satunya adalah AODV. AODV dapat menentukan rute penyampaian pesan atau pengiriman paket dengan melakukan broadcast message. Jenis routing ini dapat menghasilkan rute jaringan stabil paling baru dan jeda pada koneksi tidak terlalu tinggi. Di dalam pesan balasan yang dimiliki AODV terdapat hop count yang berisi informasi jarak antar node. Dan neighbor information yang berisi informasi daftar tetangga pada sebuah node.

Pada tugas akhir kali ini masalah yang akan diangkat adalah pendeteksian dan pencegahan serangan single black hole pada jaringan MANET dan protokol routing AODV. Metode yang digunakan adalah dengan memanfaatkan Hop Count dan Neighbor Information. Sehingga hasil yang diharapkan dari pengerjaan tugas akhir ini adalah pengiriman paket yang dikirimkan oleh node sumber akan tetap sampai ke node tujuan meskipun adanya black hole pada jaringan tersebut (MANET) dan dapat mengurangi paket drop yang disebabkan oleh serangan black hole.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut.

1. Bagaimana mendeteksi suatu black hole pada jaringan MANET dengan studi kasus yang menggunakan protokol AODV?
2. Bagaimana menerapkan metode pemanfaatan Hop Count dan Neighbor Information untuk mencegah adanya serangan black hole pada jaringan MANET agar paket tetap sampai pada node tujuan yang sebenarnya?

1.3. Batasan Permasalahan

Permasalahan pada Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut.

1. Protokol *routing* hanya dijalankan dan diuji coba pada aplikasi *Network Simulator 2* (NS-2).
2. Protokol yang digunakan dalam studi kasus ini adalah protokol AODV.
3. Jenis black hole yang diuji adalah single black hole attack.

1.4. Tujuan dan Manfaat

Tujuan dari pengerjaan Tugas Akhir ini adalah meningkatkan nilai Packet Delivery Ratio yang dikirimkan ke node tujuan dan mengurangi terjadinya paket drop yang diakibatkan oleh black hole.

Hasil dari pengerjaan Tugas Akhir ini memiliki manfaat untuk mengetahui adanya black hole pada suatu jaringan MANET yang menggunakan studi kasus protokol AODV dan melakukan tindakan pencegahan agar paket yang dikirim ke node tujuan tetap dapat diterima tanpa melewati black hole.

1.5. Metodologi

Tugas Akhir ini menggunakan beberapa tahapan dalam proses pengerjaannya. Metodologi yang dilakukan dalam pengerjaan Tugas Akhir ini terdiri atas beberapa tahapan yang dipaparkan sebagai berikut.

1. **Penyusunan Proposal Tugas Akhir**
   Proposal Tugas Akhir ini berisi tentang Implementasi Pendeteksian dan Pencegahan Serangan Black Hole dengan Memanfaatkan Hop Count dan Neighbor Information pada Jaringan MANET sebagai solusi dari permasalahan adanya black hole pada Jaringan MANET dan Protokol AODV.
   Proposal Tugas Akhir ini terdiri dari deskripsi pendahuluan yang menjabarkan latar belakang dan rumusan masalah yang mendasari dibangunnya aplikasi ini, batasan masalah dalam pembangunan aplikasi ini, serta tujuan dan manfaat yang diharapkan dapat dicapai dengan implementasi metode ini. Selain itu, pada proposal Tugas Akhir ini juga terdapat tinjauan pustaka yang menjelaskan teori-teori yang menjadi dasar pembuatan tugas akhir ini, yaitu Ad-Hoc On-Demand Distance Vector (AODV) Protokol, Mobile Ad Hoc Network (MANET), sekuritas yang diharapkan untuk jaringan, serangan sekuritas pada jaringan, dan grouped black hole attack.
2. **Studi Literatur**
   Studi literator yang dilakukan dalam pengerjaan Tugas Akhir ini adalah mengenai Implementasi Pemanfaatan Hop Cop Count untuk mengetahui node pengirim balasan pertama. Juga studi literator tentang implementasi pemanfaatan Neighbor

Information guna mendeteksi adanya serangan black hole pada suatu jaringan dan melakukan pencegahan serangannya. Sehingga, studi literator ini dapat diterapkan pada jaringan MANET yang menggunakan protokol AODV.

3. **Implementasi Metode**
   Implementasi metode yang dipakai dalam tahapan pendeteksian black hole ini adalah dengan menggunakan pemanfaatan hop count dan untuk pencegahan diperkuat dengan pemanfaatan Neighbor Information guna mengetahui perbedaan antara black hole dan bukan (node terpercaya). Implementasi metode ini akan dibangun pada protokol AODV sehingga protokol AODV ini akan dilakukan modifikasi

4. **Pengujian dan Evaluasi**
   Pengujian dan evaluasi hasil implementasi dari Tugas Akhir ini akan dilakukan uji coba pada kasus serangan *black hole* yang ada pada jaringan MANET menggunakan protokol AODV secara virtual. Aplikasi yang digunakan untuk virtualisasi yaitu Network Simulator 2 (NS-2) pada sistem operasi Linux Mint.

5. **Penyusunan Buku Tugas Akhir**
   Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi yang telah dibuat. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain sebagai berikut.

   1. Pendahuluan
      a. Latar Belakang
      b. Rumusan Permasalahan
      c. Batasan Permasalahan
      d. Tujuan dan Manfaat
      e. Metodologi
      f. Sistematika Penulisan
   2. Tujuan Pustaka

3. Perancangan Sistem
4. Implementasi
5. Pengujian dan Evaluasi
6. Penutup

1.6. Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang dijelaskan sebagai berikut.

- Bab I. Pendahuluan
  Bab ini berisi latar belakang masalah, permasalahan, batasan masalah, tujuan Tugas Akhir, manfaat Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan buku Tugas Akhir.
- Bab II. Tinjauan Pustaka
  Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.
- Bab III. Perancangan
  Bab ini berisi perancangan metode yang nantinya akan diimplementasikan dan dilakukan pengujian dari aplikasi yang akan dibangun.
- Bab IV. Implementasi
  Bab ini membahas implementasi dari rancangan sistem atau desain yang dilakukan pada tahap perancangan. Penjelasan berupa implementasi skenario, black hole pada AODV, hop count, neighbor information, dan simulasi pada NS-2.
- Bab V. Pengujian dan Evaluasi
  Bab ini menjelaskan tahap pengujian sistem dan performa dalam skenario adanya black hole pada lingkungan MANET.
- Bab VI. Penutup
  Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan terhadap rumusan masalah yang ada serta saran untuk pengembangan selanjutnya.

BAB II
TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir ini. Tujuan dari dilakukannya pembahasan pada bagian ini adalah untuk memberikan gambaran secara umum terhadap masalah dan metode-metode terkait yang akan digunakan untuk menyelesaikan rumusan masalah dalam pengembangan tugas akhir.

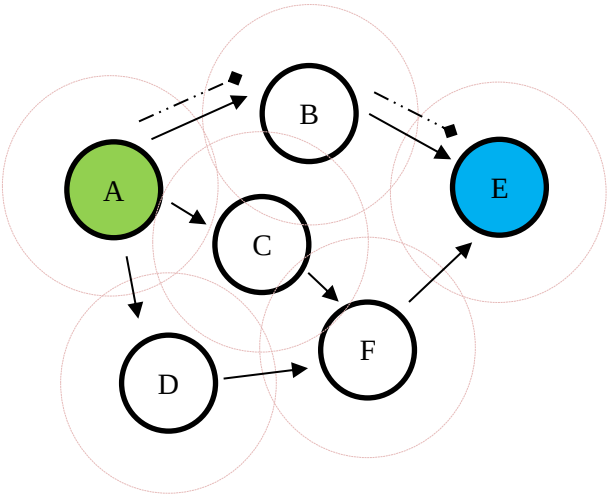
2.1. *Ad-Hoc On-Demand Distance Vector (AODV) Protocol*

Ad-Hoc On-Demand Distance Vector [3] merupakan salah satu metode penentuan rute (*routing*) jaringan. Pengaturan rute ini dilakukan di awal sebelum terbentuknya rute dan setelah rute yang sudah pernah ada rusak atau ada salah satu *node* yang berlaku sebagai *intermediate node* tersebut putus. AODV melakukan pengaturan rute *on-demand* di awal sesi pada komunikasi dan rute ini akan dipakai hingga rusak/putus seperti yang telah dijelaskan sebelumnya. Sehingga AODV melakukan inisiasi pengaturan rute dari awal. Pemeliharaan rute pada AODV menggunakan *Route Request* (RREQ), *Route Reply* (RREP) untuk mengatur pesan pada tahap *Route Discovery*, dan *Route Error* (RERR) untuk mengatur pesan pada tahap *Route Maintenance*.

AODV ini terdapat *node-node* yang memiliki perannya sendiri. Pertama adalah *source node* (*node* sumber) merupakan sebuah *node* yang ingin melakukan komunikasi terhadap *destination node* (*node* tujuan). Kedua *destination node* (*node* tujuan) yaitu sebuah *node* yang ingin dituju oleh sebuah *node* sumber. Kemudian yang ketiga adalah *intermediate node* yaitu *node* yang berfungsi menyalurkan paket dari *node* sumber ke *node* tujuan. *Node* sumber ini akan mengirimkan sebuah paket melewati *intermediate node* yang kemudian diteruskan ke *node* tujuan.

Sebelum melakukan pengiriman paket, maka dilakukan pengenalan rute jaringan terlebih dahulu. Metode yang digunakan

oleh AODV untuk penentuan rute adalah dengan melakukan *broadcast* ke *node* tetangga di sekitarnya. *Node* sumber menginisiasi sebuah *Route Discovery* dengan cara melakukan *broadcast* sebuah RREQ ke semua *node* tetangganya. Pesan RREQ ini akan tetap diteruskan oleh *intermediate node* (tetangganya) hingga pesan yang disampaikan oleh *node* sumber telah sampai ke *node* tujuan dengan syarat rute yang dilalui tetap baru. Hingga pada tahap ini RREP dibuat dan dikirimkan ke *node* sumber. Sehingga setelah *node* sumber mengirimkan RREQ maka *node* ini perlu menunggu untuk menerima sebuah RREP.



Gambar 2.1 Ilustrasi Pesan Kontrol pada AODV

Pada Gambar 2.1, terdapat ilustrasi dari pengiriman pesan kontrol pada protokol AODV. *Node* A berlaku sebagai *node* sumber. *Node* E berlaku sebagai *node* tujuan. Dan *node* lainnya sebagai *intermediate node*. Lingkaran merah muda merupakan lingkungan broadcast yang dimiliki oleh masing-masing *node.* Sehingga dihasilkan jarak terdekat untuk *node* A berkomunikasi dengan *node* E adalah dengan rute A – B – E.

2.2. *Mobile Ad Hoc Network (MANET)*

Mobile Ad Hoc Network (MANET) [5] merupakan kumpulan dari *node* yang bersifat nirkabel (tanpa kabel) dan pengaturannya dinamis, dapat di mana saja dan kapan saja tanpa menggunakan infrastruktur yang ada.

MANET memiliki tingkat kerentanan lebih tinggi dibandingkan dengan jaringan kabel. Sebuah sistem tertentu mungkin rentan terhadap manipulasi data yang tidak sah karena sistem tidak memverifikasi identitas penggunanya sebelum mengizinkan akses data. Beberapa kerentanan pada MANET adalah sebagai berikut:

- Ketersediaan sumber daya
- Skalabilitas
- Membutuhkan *intermediate node* untuk transfer informasi
- Topologi dinamis
- Sumber tenaga terbatas

2.3. Sekuritas yang Diharapkan

Sekuritas merupakan salah satu penjaminan mutu suatu sistem. Sehingga sekuritas dapat dianggap sebagai suatu hal yang penting. Dalam MANET, metode penyaluran informasi menggunakan *broadcast* ke *node* tetangga *(intermediate node)*. Hal ini membutuhkan kualitas yang baik agar paket dapat sampai ke tujuan. Tujuan untuk mengevaluasi apakah jaringan seluler ad-hoc adalah aman atau tidak adalah sebagai berikut:

- Ketersediaan
- Kerahasiaan
- Integritas
- Otentikasi
- Non-penolakan
- Anonimitas

2.4. Serangan Sekuritas

Serangan yang ada pada jaringan komputer terdapat berbagai macam tipe. William Stallings mengklasifikasikan secara umum serangan (attack) dalam jaringan menjadi dua yaitu passive attack dan active attack

2.4.1. Passive Attack (Serangan Pasif)

Serangan pasif adalah jenis serangan yang tidak membahayakan terhadap sebuah sistem jaringan. Jenis serangan ini tidak menyebabkan hilangnya sumber daya dalam sebuah jaringan maupun menyebabkan kerusakan terhadap sebuah sistem jaringan yang di serang menggunakan jenis serangan ini. Sumber daya yang terdapat dalam sistem jaringan yang berupa data, bandwidth jaringan, mesin cetak, memori dalam sebuah komputer, unit pengolah (prosesor) dan masih banyak lagi. Intinya jenis serangan ini hanya melakukan pengamatan terhadap semua sumber daya yang terdapat dalam sebuah sistem jaringan komputer.

2.4.2. Active Attack (Serangan Aktif)

Serangan aktif adalah serangan yang dilakukan oleh node berbahaya yang menanggung sebagian biaya energi untuk melakukan serangan. Serangan aktif melibatkan beberapa modifikasi aliran data atau penciptaan aliran palsu. Serangan aktif dapat internal atau eksternal.

Ketika penyerang telah memasuki lingkungan internal maka serangan internal ini lebih berbahaya dan sulit untuk diketahui. Berikut jenis-jenis serangan aktif:

- Black hole (Lubang hitam)
- Gray hole (lubang Gray)
- Worm hole (lubang Cacing)
- Jellyfish attack (Serangan Ubur-ubur)
- Spoofing (Spoofing)
- Sybil attack (serangan Sybil)

2.5. *Network Simulator 2* (NS-2)

Network Simulator 2 (NS-2) merupakan alat simulasi jaringan yang bersifat *open source* yang banyak digunakan dalam mempelajari struktur dinamika dari jaringan komunikasi. Simulator ini ditargetkan pada penelitian jaringan dan memberikan dukungan yang baik untuk simulasi *routing*, protokol *multicast* dan protokol IP, seperti UDP, TCP, RTP, jaringan nirkabel dan jaringan satelit. Beberapa keuntungan menggunakan *network simulator* sebagai perangkat lunak simulasi yaitu *network simulator* dilengkapi dengan *tools* validasi, pembuatan simulasi dengan menggunakan *network simulator* jauh lebih mudah daripada menggunakan *software developer* seperti Delphi atau C++, *network simulator* bersifat *open source* di bawah GPL (Gnu Public License) dan dapat digunakan pada sistem operasi Windows dan sistem operasi Linux.

Pada Tugas Akhir ini digunakan NS-2 versi 2.35 sebagai aplikasi simulasi jaringan skenario MANET yang dihasilkan oleh program *default* dari NS-2 yaitu *generator file node-movement*. NS-2 dijalankan pada sistem operasi Linux.

Tools yang disebut 'setdest' dikembangkan oleh CMU (Carnegie Mellon University) untuk menghasilkan *movement* dari *node* dalam jaringan nirkabel. *Node movement* dihasilkan dengan kecepatan gerak yang spesifik menuju lokasi acak atau lokasi spesifik yang berada dalam kawasan yang telah ditentukan. Ketika *node* tiba ke lokasi pergerakan, *node* tersebut bisa diatur untuk berhenti.

```
./setdest [-v version ] [-n num_of_nodes] [-p
pausetime] [-M maxspeed] [-t simtime] [-x maxx] [-y
maxy] > [outdir/movement-file]
```

Kode Sumber 2.1 Format Command Line 'setdest'

Pengguna harus menambahkan kode program ‘setdest’ pada skenario tcl. Format kode program ‘setdest’ ditunjukkan pada Kode Sumber 2.1 dan keterangannnya ditunjukkan pada Tabel 2.1.

Tabel 2.1 Keterangan pada *Command Line* 'setdest'

Parameter	Keterangan
-v version	Versi ‘setdest’ simulator yang digunakan
-n num	Jumlah *node* dalam skenario
-p pausetime	Durasi ketika sebuah *node* tetap diam setelah tiba di lokasi pergerakan. Jika nilai ini diatur ke 0, maka *node* tidak akan berhenti ketika tiba di lokasi pergerakan dan akan terus bergerak
-M maxspeed	Kecepatan maksimum sebuah *node*. *Node* akan bergerak pada kecepatan acak dalam rentang [0, amxspeed]
-t simtime	Waktu simulasi
-x max x	Panjang maksimum area simulasi
-y max y	Lebar maksimum area simulasi

Command Line ‘setdest’ menghasilkan *file output* yang berisi jumlah *node* dan mobilitas yang akan digunakan dalam *file* Tcl selama simulasi. *File output*, selain mengandung skrip pergerakan, juga mengandung beberapa statistik lain tentang perubahan *link* dan rute.

```
./setdest -v 1 -n 50 -p 2.0 -M 20.0 -t 200 -x 500 -
y 500 > scen-20-test
```

Kode Sumber 2.2 Hasil Output pada file ‘scen-20-test’

Untuk membuat skenario *node-movement* yang terdiri dari 50 *node*, bergerak dengan kecepatan maksimum 20.0 m/s dengan jeda rata-rata antar gerakan sebesar 2 detik, simulasi akan berhenti

setelah 200 detik dengan batas topologi yang diartikan sebagai 500 x 500 meter^2, *command line*-nya terlihat seperti pada Kode Sumber 2.2

File output ditulis ke "stdout" secara *default*. Di sini *output* disimpan ke dalam *file* "scen-20-test". *File* dimulai dengan posisi awal *node* dan berlanjut menetapkan *node-movement* seperti terlihat pada Kode Sumber 2.3.

```
$ns_ at 2.000000000000 "$node_(0) setdest
90.441179033457 44.896095544010
1.373556960010
```

Kode Sumber 2.3 Contoh Command Line 'setdest'

Command line pada Kode Sumber 2.3 dari 'scen-20-test' mendefinisikan bahwa *node* (0) pada detik ke 2.0 mulai bergerak ke arah tujuan (90.44, 44. 89) dengan kecepatan 1.37 m/s. *Command line* ini dapat digunakan untuk mengubah arah dan kecepatan gerak dari *mobile node*. Arahan untuk *General Operations Director* l (GOD) yang ada juga di *file node-movement*. Objek "GOD" digunakan untuk menyimpan informasi global tentang keadaan dari lingkungan jaringan dan *node* di sekitarnya. Namun isi dari *file* "GOD" tidak boleh diketahui oleh setiap bagian dalam simulasi.

Dalam simulasi di sini objek "GOD" hanya digunakan untuk menyimpan sebuah *array* dari jumlah *hop* terpendek yang diperlukan untuk mencapai satu *node* ke *node* yang lain. Objek "GOD" tidak menghitung jumlah *hop* yang diperlukan selama simulasi berjalan, karena akan cukup memakan waktu. Namun "GOD" menghitung *hop* di akhir simulasi. Informasi yang dimuat ke dalam objek "GOD" dari pola pergerakan *file* terdapat pada baris perintah di Kode Sumber 2.4.

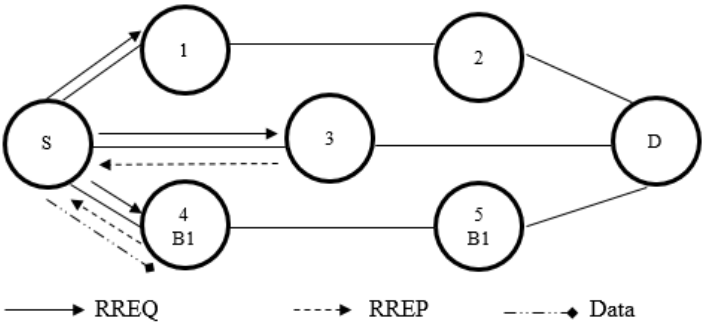
```
$ns  at 899.642 "$god  set-dist 23 46 2"
```

Kode Sumber 2.4 Command Line "GOD" pada 'scen-20-test'

Ini berarti bahwa jarak terpendek antara *node* 23 dan *node* 46 berubah menjadi 2 *hop* di waktu 899,642 detik. Program 'setdest' menghasilkan *file node-movement* menggunakan algoritma *random way point*. Perintah-perintah yang termasuk dalam program utama untuk memuat *file-file* ini dalam objek "GOD".

2.6. Single Black Hole Attack

Sebuah serangan Black Hole [7][8] adalah jenis penolakan layanan di mana node berbahaya dapat menarik semua paket dengan mengklaim sebagai rute baru untuk tujuan dan kemudian menyerapnya tanpa meneruskannya ke tujuan. *Single Black Hole Attack* merupakan serangan black hole individu sehingga penyerangannya tidak ada kerjasama dengan black hole yang lain. Serangan single black hole memiliki dua tahap. Pertama adalah node berbahaya memanfaatkan ad-hoc routing protokol seperti AODV untuk mengenalkan dirinya sebagai node yang memiliki rute yang valid ke node tujuan, dengan tujuan mencegat paket, meskipun rute ini palsu. Pada tahap kedua, node penyerang menjatuhkan paket tanpa meneruskan ke node tujuan. Black hole ini biasanya memodifikasi paket ketika meninggalkan data paket dari node lain yang tidak berpengaruh. Hal ini membuat sulit node lain untuk mendeteksi node berbahaya tersebut.



Gambar 2.2 Ilustrasi Black Hole pada AODV

Pada Gambar 2.2, terdapat ilustrasi skenario adanya beberapa black hole pada satu jaringan. Node S merupakan node sumber dan D adalah node tujuan. Sedangkan node 1 hingga node 5 adalah intermediate node.

Pada ilustrasi kali ini yang bertindak sebagai black hole adalah node 4 (B1) dan 5 (B2). Ketika node sumber ingin mengirimkan sebuah paket data ke node tujuan, pertama kali yang dia lakukan adalah mengirimkan paket RREQ ke node tetangganya. Node yang berbahaya (black hole) juga akan menerima RREQ. Karena black hole memiliki karakteristik menanggapi RREQ pertama kali dari RREQ yang lain, maka black hole ini akan segera mengirimkan RREP. RREP dari B1 telah mencapai node sumber. Sekarang ketika penerimaan RREP dari B1, node sumber mulai mengirimkan paket. Pada tahap ini, B1 akan melakukan drop pada paket yang dikirimkan oleh sumber tanpa meneruskannya ke tujuan atau dari B1 akan meneruskan semua data ke B2 dan kemudian B2 akan melakukan drop paket data yang akan diteruskan ke tujuan. Sehingga paket data akan hilang dan tidak akan pernah mencapai ke node tujuan.

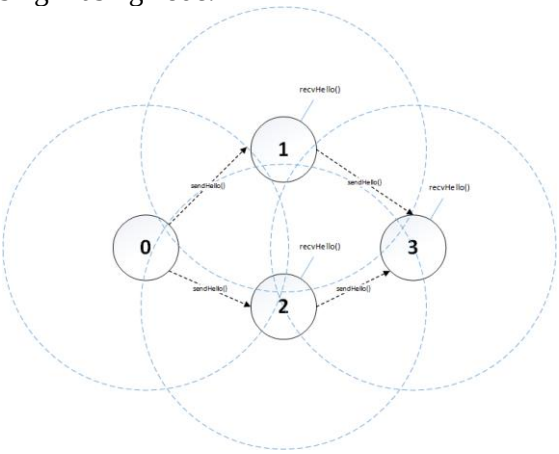
2.7. Hop Count

Hop count merupakan jumlah node yang harus dilewati oleh paket dari node tujuan untuk sampai ke node sumber. Sistem kerja hop count ini diawali ketika node tujuan menerima request dari node sumber bahwa akan ada pengiriman paket ke node tujuan. Ketika node tujuan mengetahui bahwa dia merupakan tujuan dari pengiriman paket, maka node tujuan tersebut melakukan sendReply dari request yang dikirim oleh node sumber. Informasi yang dikirimkan oleh node tujuan ini berupa alamat IP dari node sumber agar pesan balasan ini dapat sampai ke node sumber, alamat IP dari node tujuan ini, sequence number, lifetime, waktu (timestamp), dan hop count. Hop count diatur dengan diawali dari 1. Sehingga ketika paket balasan ini melewati intermediate node, maka hop count akan tambah menjadi 1. Begitu seterusnya hingga paket diterima oleh node sumber.

2.8. Neighbor Information

Pada lingkungan MANET yang menggunakan protocol AODV, telah diberikan fitur Neighbor Information. Neighbor information berisi informasi-informasi mengenai tetangga dari sebuah node. Informasi yang diberikan bisa berupa informasi node ID, hop count, node aktif, dan sebagainya.

Sistem kerja informasi neighbor ini didapatkan dengan cara mengirimkan "hello message" dari node yang ingin mengetahui tetangganya. Hello message dikirimkan secara otomatis dari protokol AODV dengan interval waktu tertentu. Ketika hello message dibroadcast, node yang ada pada jangkauannya akan menerima hello message tersebut yang kemudian akan menambahkan node pengirim dan node penerima sebagai tetangga untuk masing-masing node.



Gambar 2.3 Ilustrasi Broadcast Hello Message

Pada Gambar 2.3 dapat dilihat ilustrasi bagaimana node melakukan broadcast hello message ke node-node tetangga. Dimulai dengan node 0 melakukan sendHello dan diterima oleh node 1 dan node 2. Pada fungsi recvHello yang diterima oleh node 1 dan node 2, terdapat skenario penambahan informasi tentang tetangga dari node yang bersangkutan.

2.9. Packet Delivery Ratio (PDR)

Teknik penghitungan uji coba untuk skenario simulasi jaringan ada berbagai macam cara. Salah satunya adalah menggunakan Packet Delivery Ratio (PDR). PDR merupakan teknik penghitungan perbandingan jumlah paket yang diterima oleh node tujuan dengan jumlah paket yang dikirimkan oleh node sumber. Tujuan dari teknik penghitungan ini adalah mengetahui rasio paket yang diterima oleh node tujuan dibanding dengan paket yang dikirimkan oleh node sumber.

Penghitungan rasio dilakukan menggunakan file .tr yang dihasilkan dari NS-2 setelah menjalankan skenario pada file tcl. Berikut merupakan contoh isi dari file .tr.

```
s 1.083746959 _0_ MAC  --- 0 cbr 1078 [13a 4 0 800]
------- [0:0 1:0 30 4] [0] 0 0
```

Kode Sumber 2.5 Contoh File .tr

Pada Kode Sumber 2.5, didapatkan informasi bahwa.

s	: send, node 0 mengirim paket.
1.083746959	: waktu (timestampe) untuk operasi ini.
_0_	: id node (alamat IP) yang melakukan operasi.
MAC	: trace level (MAC layer)
0	: id paket
cbr	: tipe paket
1078	: ukuran paket yang dikirim (bytes)

2.10. Error Rate Ratio

Teknik pengujian ini melakukan penghitungan rasio dari paket yang dikirim oleh node sumber yang diterima oleh node black hole. Pada kasus ini, paket yang dikirimkan oleh node sumber memungkinkan untuk tidak diterima oleh node tujuan, karena adanya serangan black hole di lingkungan jaringan tersebut. Sehingga perlu adanya penghitungan rasio dari paket yang diterima

oleh black hole. Untuk memastikan berapa persen error yang didapat pada satu skenario simulasi tersebut.

Metode penghitungan error rate ratio ini hampir sama dengan metode penghitungan PDR, perbedaannya hanya pada node yang menerima paket dari sumber. Untuk error rate ratio, node yang menerima paket bukan node tujuan, melainkan node black hole.

BAB III
PERANCANGAN SISTEM

Pada bab ini dijelaskan mengenai dasar perancangan dari perangkat lunak yang akan dibangun dalam Tugas Akhir ini. Secara khusus akan dibahas mengenai deskripsi umum sistem, perancangan skenario, alur, serta gambaran implementasi sistem yang diterapkan pada *Network Simulator* 2 (NS-2).

3.1. Deskripsi Umum

Pada Tugas Akhir ini akan dilakukan implementasi metode deteksi dan pencegahan adanya black hole pada lingkungan jaringan MANET. Protokol yang digunakan adalah AODV. Dalam pembuatan skenario MANET menggunakan simulator yaitu Network Simulator 2 (NS-2).

Perancangan skenario uji coba mobilitas MANET diawali dengan perancangan skenario yang akan dilakukan pada simulasi black hole. Kemudian melakukan modifikasi pada AODV untuk menambahkan black hole.

Setelah itu melakukan perancangan untuk deteksi dan pencegahan serangan black hole yaitu dengan merancang metode neighbor information dan metode hop count. Dan terakhir melakukan perancangan simulasi pada NS-2 yang dilakukan pada skrip TCL.

3.2. Perancangan Skenario

Perancangan skenario uji coba pengiriman paket dari node sumber ke node tujuan dengan adanya black hole pada lingkungan MANET.

Pada simulasi ini, terdapat beberapa node dengan pembagian peran sebagai berikut:

- Node sumber
  Merupakan node awal pengiriman paket.
- Node intermediate

Merupakan node penyalur paket dari node sumber ke node tujuan.

- Node tujuan
  Merupakan node akhir pengiriman paket dari node sumber atau node yang menjadi tujuan pengiriman dari node sumber.
- Node black hole
  Merupakan node yang mengatasnamakan dirinya sebagai node tujuan, sehingga paket yang dikirim dari node sumber tidak pernah sampai ke node tujuan.

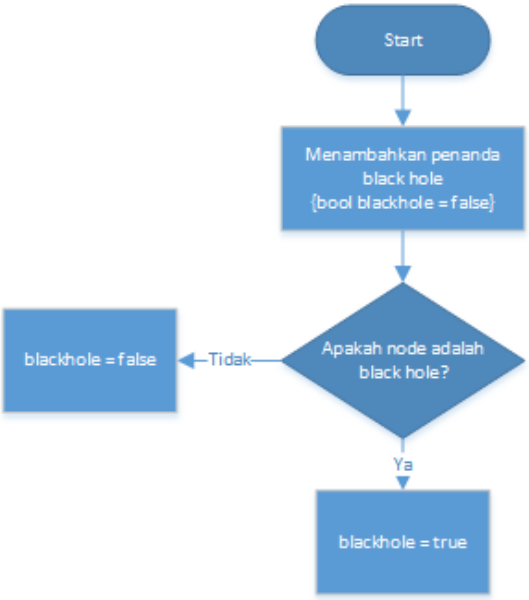
Masing-masing node akan bergerak ke suatu tujuan karena skenario ini akan diimplementasikan pada lingkungan MANET. Pemberian warna untuk masing-masing peran node agar membedakan node sumber, tujuan, intermediate, dan black hole.

Dan yang terakhir pemberian identitas untuk node black hole agar dapat dikenali oleh protokol AODV.

3.3. Perancangan Black Hole pada AODV

Pada perancangan implementasi black hole, dilakukan pada file aodv.cc dan aodv.h agar pada skrip TCL dapat dibaca ketika black hole telah ditambahkan. Metode penambahan black hole ini yaitu yang pertama adalah membaca dari file tcl (file simulasi). Beberapa node akan ditandai sebagai black hole. Kemudian pada protokol AODV dilakukan pengecekan untuk masing-masing node. Ketika node black hole telah teridentifikasi, maka penanda bahwa node tersebut merupakan blackhole diaktifkan.

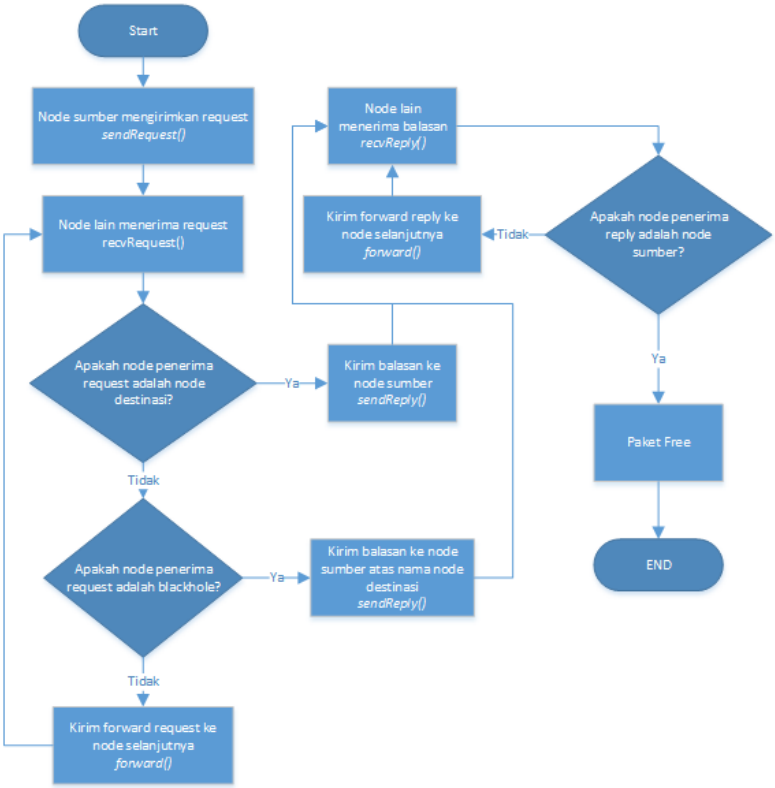
Penggunaan pertama kali (Gambar 3.1) penanda berikut adalah untuk melakukan drop paket yang diterima oleh black hole pada saat node sumber mengirim request. Kemudian penggunaan yang kedua (Gambar 3.2) adalah ketika black hole menerima request dari node sumber, black hole ini akan mengirimkan balasan dengan mengatas namakan dirinya sebagai node tujuan.



Gambar 3.1 Instansiasi Penanda Blackhole

3.4. Perancangan Hop Count

Data informasi hop count terdapat pada pesan balasan (reply) yang dikirimkan dari node tujuan maupun black hole. Metode pengecekan menggunakan hop count dilakukan ketika sebuah node menerima pesan reply (*recvRep()*) dari node yang lain. Kemudian sebelum pesan diproses, dilakukan dulu pengecekan apakah nilai hop count adalah 1. Yang berarti pengirim dari pesan reply ini adalah node tujuan atau juga dari node black hole. Selanjutnya akan diteruskan pengecekan dengan menggunakan neighbor information. Jika hop count tidak sama dengan 1, maka pesan reply dapat diproses.



Gambar 3.2 Alur Perancangan Blackhole

3.5. Perancangan Neighbor Information

Pada perancangan ini, akan diimplementasikan pada file Node.h, Node.cc, dan AODV.cc. Metode yang digunakan adalah menggunakan fitur “hello message” yang ada pada AODV. Hello message akan dikirimkan dengan interval waktu tertentu.

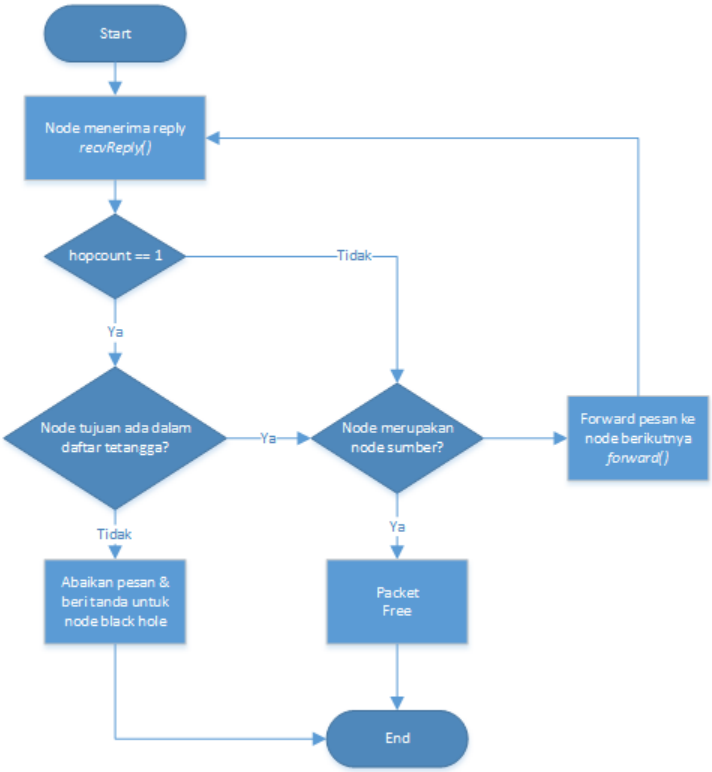
Hal pertama untuk mengimplementasi metode Neighbor Information untuk deteksi black hole adalah menambahkan penanda untuk masing-masing node apakah node tersebut

terpercaya atau tidak. Kemudian ketika masing-masing node mengirimkan hello message, maka akan dilakukan penyimpanan data/informasi untuk node yang menerima hello message dan node yang mengirimkan hello message. Tidak lupa pula untuk menghapus data neighbor ketika node tersebut sudah tidak dalam jangkauannya.

Penggunaan neighbor information ini ketika sebuah node menerima balasan (*recvReply()*). Sebelum balasan diproses, node akan melakukan pengecekan apakah node yang mengirimkan balasan merupakan black hole atau bukan. Cara pendeteksiannya adalah sebagai berikut:

1. Node menerima reply dari node yang lain.
2. Lakukan pengecekan apakah hop count dari pesan reply ini adalah 1.
    a. Jika hop count 1, maka lakukan pengecekan lagi ke poin 3.
    b. Jika hop count bukan 1, maka lanjut ke poin 4.
3. Pengecekan apakah node tujuan ada dalam daftar tetangga dari node ini.
    a. Jika node tujuan tidak ada dalam daftar tetangga, abaikan pesan balasan dan catat node tersebut sebagai node berbahaya (black hole).
    b. Jika node tujuan ada dalam daftar tetangga, lanjutkan ke poin 4.
4. Pengecekan apakah node yang menerima ini merupakan sumber.
    a. Jika node merupakan node sumber, pesan balasan berhasil sampai ke node sumber. End.
    b. Jika node bukan node sumber, lakukan forward pesan balasan ke node berikutnya. Kembali ke poin 1.

Perancangan metode pendeteksian dan pencegahan black hole dapat dilihat dalam bentuk diagram alur sebagai berikut.



Gambar 3.3 Pendeteksian dan Pencegahan Black Hole

3.6. Perancangan Simulasi pada NS-2

Pada perancangan kode NS-2 dengan konfigurasi MANET, dengan skrip TCL yang diberikan parameter-parameter untuk membangun percobaan simulasi MANET pada NS-2. Berikut parameter simulasi perancangan sistem MANET yang dapat digunakan dapat dilihat pada **Tabel 3.1**.

Tabel 3.1 Parameter Simulasi pada NS-2

No.	Parameter	Spesifikasi
1	Network simulator	NS- 2.35
2	*Routing* Protocol	AODV
3	Waktu simulasi	200 detik
4	Waktu Pengiriman Paket Data	1 – 200 detik
5	Area simulasi	500 m x 500 m
6	Banyak *node*	50
7	Banyak black hole	3 (node 0, 1, 2)
8	Radius transmisi	250 m
9	Tipe koneksi	UDP
10	Tipe data	Constant Bit Rate (CBR)
11	Source / Destination	Statik (*Node* 48 / *Node* 49)
12	Kecepatan generasi paket	1 paket per detik
13	Ukuran paket data	1024 bytes
14	Protokol MAC	IEEE 802.11
15	Tipe Antena	OmniAntenna
16	Tipe Interface Queue	Droptail/PriQueue
17	Tipe kanal	Wireless channel
18	Tipe *trace*	Old Format *Trace*

[Halaman ini sengaja dikosongkan]

BAB IV
IMPLEMENTASI

Bab ini membahas tentang implementasi dari perancangan sistem. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, implementasi skenario, implementasi black hole pada AODV, implementasi metode hop count, implementasi neighbor information, dan implementasi simulasi pada NS-2.

4.1. Lingkungan Pembangunan Sistem

Pembangunan perangkat lunak dilakukan pada lingkungan pengembangan sebagai berikut:

4.1.1. Lingkungan Perangkat Lunak

Adapun perangkat lunak yang digunakan untuk pengembangan sistem adalah sebagai berikut:

- Sistem Operasi Linux Mint 17 64 bit untuk lingkungan NS-2
- *Network Simulator* 2 (NS-2) versi 2.35.

4.1.2. Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk implementasi perangkat lunak Tugas Akhir adalah sebagai berikut:

- *Processor* AMD FX Quad Core CPU @3.30GHz
- AMD Radeon R7 Graphics
- Media penyimpanan sebesar 500GB
- RAM sebesar 4 GB DDR3.

4.2. Implementasi Skenario

Implementasi skenario mobilitas MANET dengan node-node sesuai peran masing-masing.

- Node n-2 : sebagai node sumber

(contoh: jumlah node = 50, node sumber = 48)

- Node n-1 : sebagai node tujuan
- (contoh: jumlah node = 50, node sumber = 49)
- Node 0 dan seterusnya sejumlah banyak black hole (contoh: node 0, 1, 2)
- Node sisa dan seterusnya merupakan intermediate node. (contoh: node 3, ..., 17)

Posisi node sumber dan tujuan statis dan tidak bergerak. Sedangkan node black hole dan node intermediate posisi dan pergerakannya random.

```
set n(48) [$ns node]
$n(48) set X_ 100
$n(48) set Y_ 250
$n(48) set Z_ 0.0
$ns initial_node_pos $n(48) 20

set n(49) [$ns node]
$n(49) set X_ 450
$n(49) set Y_ 250
$n(49) set Z_ 0.0
$ns initial_node_pos $n(49) 20
```

Kode Sumber 4.1 Posisi statis node sumber dan node tujuan

Pada Kode Sumber 4.1, node 48 (node sumber) posisi diatur pada posisi 100, 250 (sumbu x, y) dan kedalaman 0 (sumbu z). Sedangkan node 1 (node tujuan) posisi diatur pada posisi 450, 250 (sumbu x, y) dan kedalaman 0 (sumbu z).

Pada Kode Sumber 4.2**Error! Reference source not found.**, semua posisi node kecuali node sumber dan node tujuan diatur secara random dengan aturan untuk sumbu x random mulai dari 0 hingga 500, sumbu y random mulai dari 0 hingga 500, dan sumbu z 0.

```
set rng [new RNG]
$rng seed next-substream
for {set i 0} {$i < 48} {incr i} {
    set n($i) [$ns node]
    $n($i) set X_ [$rng uniform 0.0 500.0]
    $n($i) set Y_ [$rng uniform 0.0 500.0]
    $n($i) set Z_ 0.0 #flat ground
    $ns initial_node_pos $n($i) 20
}
```

Kode Sumber 4.2 Posisi random node black hole dan node intermediate

Pemberian warna seperti diimplementasikan pada Kode Sumber 4.3 Pemberian warna node, untuk masing-masing peran node. Node sumber diberi warna hijau, node tujuan berwarna biru, node black hole berwarna merah, dan untuk node lainnya (intermediate) berwarna standar (hitam).

```
# Source Node
$n(48) color green
$ns at 0.0 "$n(48) color green"
$ns at 0.0 "$n(48) label Source"

# Destination Node
$n(49) color blue
$ns at 0.0 "$n(49) color blue"
$ns at 0.0 "$n(49) label Destination"

# Blackhole Attacker
$n(0) color red
$ns at 0.0 "$n(0) color red"
$ns at 0.0 "$n(0) label Attacker"
```

Kode Sumber 4.3 Pemberian warna node

Pemberian identitas untuk node black hole dilakukan dengan menambahkan penanda yaitu “hacker” pada file skenario tcl sebagai berikut.

```
$ns at 0.0 "[$n(0) set ragent_] hacker"
```

Kode Sumber 4.4 Pemberian identitas node black hole

Setelah itu menjadikan semua node kecuali node sumber dan node tujuan bergerak ke suatu lokasi random. Generate pergerakan node random menggunakan fitur setdest pada NS-2.

Pada Kode Sumber 4.5, -v 1 merupakan versi dari fitur setdest ini, -n 48 merupakan jumlah node yang ingin diimplementasi memiliki pergerakan (mobile), -p 1 merupakan pause time 1 detik, -M 5 merupakan maksimal kecepatan pergerakan (m/s), -t 200 merupakan waktu simulasi 200 detik, -x 500 merupakan lebar lingkungan simulasi (koordinat x), dan -y 500 merupakan ketinggian lingkungan simulasi (koordinat y). Dari hasil running setdest ini, dimasukkan ke dalam file yang nantinya dapat digunakan untuk skenario di file tcl.

```
./setdest -v 1 -n 48 -p 1 -M 5 -t 200 -x 500 -y 500
> mal3_50.move
```

Kode Sumber 4.5 Pergerakan random node black hole dan node intermediate

4.3. Implementasi Black Hole pada AODV

Untuk implementasi adanya black hole pada AODV, perlu adanya modifikasi pada protokol AODV yang ada pada Network Simulation 2 (NS-2).

Modifikasi yang pertama dilakukan adalah dengan menambahkan variable malicious yang bertipe protected boolean sebagai penanda node black hole. Penambahan variable malicious dilakukan di dalam kelas AODV:public Agent pada file AODV.h, dapat dilihat pada Kode Sumber 4.6

```
class AODV: public Agent {
       ...
protected:
       bool malicious;
       ...
}
```

Kode Sumber 4.6 Variable malicious pada AODV.h

Kemudian penambahan juga dilakukan pada file AODV.cc dengan menambahkan beberapa kode. Pada fungsi konstruktor AODV, dilakukan instansiasi untuk variable malicious agar penanda blac hole dapat digunakan dengan benar hanya ketika node black hole berhasil diidentifikasi oleh AODV, penambahan ini dapat dilihat pada Kode Sumber 4.7.

```
AODV::AODV(nsaddr_t id) : Agent(PT_AODV),
btimer(this), htimer(this), ntimer(this),
rtimer(this), lrtimer(this), rqueue() {
       ...
       malicious = false;
       ...
}
```

Kode Sumber 4.7 Instansiasi variable pada AODV.cc

Ketika AODV mengenali node yang bersangkutan sebagai black hole (dapat diidentifikasi dari skenario tcl dengan pemberian identitas pada node-node tertentu), maka penanda malicious diatur menjadi true, penambahan ini dapat dilihat pada Kode Sumber 4.9 pada kasus ini pemberian identitas dengan nama “hacker”.

```
int AODV::command(int argc, const char*const* argv)
{
   if(argc == 2) {
       Tcl& tcl = Tcl::instance();
       ...
       if(strncasecmp(argv[1], "hacker", 6) == 0) {
              malicious = true;
              return TCL_OK;
       }
       ...
   }
   ...
}
```

Kode Sumber 4.9 Pemberian tanda malicious pada node

Alur kerja AODV mulai dari status node sumber mengirimkan request (*sendRequest()*). Kemudian node lain yang menerima request (*recvRequest()*) akan melakukan pengecekan apakah node tersebut merupakan destinasi, black hole, atau intermediate node. Sehingga modifikasi dilakukan pada fungsi recvRequest() yang ada di dalam file AODV.cc. Modifikasi kode ini dapat dilihat pada Kode Sumber 4.8 untuk melakukan drop paket dengan alasan looping dan Kode Sumber 4.10. Black hole melakukan pengiriman pesan reply ke node sumber yang mengatasnamakan dirinya sebagai node tujuan.

```
Void AODV::rt_resolve(Packet *p) {
       ...
       if (malicious == true ) {
              drop(p, DROP_RTR_ROUTE_LOOP);
              return;
       }
       ...
}
```

Kode Sumber 4.8 Drop paket oleh black hole

```
Void AODV::recvRequest(Packet *p) {
   ...
   // First check if I am the destination ..
   if(rq->rq_dst == index) {
       ...
   }
   // check if black hole ..
   else if (malicious == true) {
       seqno = max(seqno, rq->rq_dst_seqno)+1;
       if (seqno%2) seqno++;

       // IP Destination
       sendReply(rq->rq_src,

       // Hop Count is set to 1 to confuse the
       // source node!
       1,

       // Dest IP Address
       rq->rq_dst,

       // Dest Sequence Num
       seqno,

       // Lifetime
       MY_ROUTE_TIMEOUT,

       // timestamp
       rq->rq_timestamp);
       Packet::free(p);
   }
   ...
}
```

Kode Sumber 4.10 Pengiriman reply oleh Black Hole

Dengan modifikasi dari kode-kode program di atas, maka implementasi adanya black hole pada AODV NS-2 sudah dapat digunakan.

4.4. Implementasi Hop Count

Informasi hop count didapatkan dari pesan reply yang dikirimkan ketika node tujuan atau node black hole menerima request dari node sumber. Hop count pada pesan yang dikirimkan oleh node tujuan dan node black hole sama, yaitu satu (1). Sehingga dilakukan pengecekan apakah node yang menerima reply memiliki informasi hop count 1 atau tidak. Jika memiliki hop count 1 maka dapat dicurigai. Apakah node pengirim reply ini adalah node tujuan atau node black hole.

Hal yang perlu dilakukan untuk implementasi hop count ini yaitu dengan melakukan penambahan kode pada fungsi recvReply yang ada pada file AODV.cc (Kode Sumber 4.11).

```
Void AODV::recvReply(Packet *p) {
       ...
       if (rp->rp_hop_count == 1) {
              // lakukan pengecekan lanjutan
              // pengecekan neighbor information
       }
       ...
}
```

Kode Sumber 4.11 Pengecekan hop count

4.5. Implementasi Neighbor Information

Dengan pengecekan nilai hop count ini belum cukup, maka dilakukan pengecekan menggunakan metode Neighbor Information. Sebelum informasi neighbor dapat digunakan, perlu adanya modifikasi pada beberapa file yang ada pada NS-2. Yang pertama adalah menambahkan variable penanda untuk node

terpercaya (reliable node) pada struct neighbor_list_node yang ada pada file Node.h (Kode Sumber 4.12).

```
struct neighbor_list_node {
       ...
       int reliable;
       ...
}
```

Kode Sumber 4.12 Penambahan penanda reliable node

Metode mendapatkan neighbor information ini yaitu dari hello message. Hello message dikirimkan dengan interval waktu tertentu. Ketika node mulai melakukan broadcast hello message, fungsi yang dijalankan pertama kali adalah sendHello(). Sehingga pada fungsi ini dilakukan modifikasi untuk menghapus daftar tetangga ketika node yang melakukan pengiriman hello message ini adalah node sumber (dalam kasus ini node 0) agar daftar tetangga tetap segar dan mendapatkan informasi terbaru (Kode Sumber 4.14). Untuk implementasi penghapusan daftar tetangga, perlu adanya pembuatan fungsi baru pada kelas Node. Yaitu menambahkan fungsi pada Node.h (Kode Sumber 4.13) dan mengimplementasi fungsi clearNeighbor() pada file Node.cc (Kode Sumber 4.15)

Kemudian node lain akan menerima pesan hello. Pada penerimaan pesan hello ini perlu adanya modifikasi untuk menyimpan informasi tetangga. Modifikasi ini dilakukan pada fungsi recvHello() yang ada pada file AODV.cc (Kode Sumber 4.16) untuk menambahkan pengirim dan penerima pesan hello ke dalam daftar tetangga masing-masing dan modifikasi pada fungsi addNeighbor(Node*) yang ada pada file Node.cc untuk tidak menambahkan node ke dalam daftar tetangga ketika node tersebut sudah ada dalam daftar (Kode Sumber 4.17).

```
if (index == 0) {

	Node* node =
		Node::get_node_by_address(index);
	node->clearNeighbor();

}
```

Kode Sumber 4.14 Pemanggilan clear neighbor pada revHello()

```
class Node : public ParentNode {

public:

	...
	void clearNeighbor();
	...

}
```

Kode Sumber 4.13 Penambahan fungsi clearNeighbor pada Node.h

```
...
void Node::clearNeighbor() {
	neighbor_list_ = NULL;
}
...
```

Kode Sumber 4.15 Implementasi clearNeighbor pada Node.cc

```
void AODV::recvHello(Packet *p) {
	...
	Node* sender_node =
		Node::get_node_by_address(rp-
	>rp_dst);
	Node* receiver_node =
		Node::get_node_by_address(index);
	sender_node->addNeighbor(receiver_node);
	receiver_node->addNeighbor(sender_node);
	...
}
```

Kode Sumber 4.16 Menambahkan tetangga pada recvHello()

Ketika fungsi penambahan sudah diimplementasikan semua seperti di atas, maka data neighbor information dapat digunakan. Neighbor information untuk deteksi dan pencegahan black hole digunakan pada saat sebuah node menerima pesan reply seperti metode pendeteksian menggunakan hop count. Neighbor information ini merupakan metode yang digunakan untuk memperkuat deteksi dan pencegahan black hole menggunakan hop count seperti yang dijelaskan pada bagian Implementasi Hop Count. Pengecekan dilakukan sebelum pesan reply diolah. Sehingga implementasi hop count dan neighbor information jika digabungkan akan menjadi seperti Kode Sumber 4.17.

Dengan nilai hop count dari pesan reply dan informasi neighbor dari node, maka sudah dapat diketahui apakah node yang mengirimkan pesan reply ini merupakan black hole atau node tujuan sebenarnya.

```
void Node::addNeighbor(Node * neighbor) {
   int flag = 0;
   neighbor_list_node* my_neighbor_list =
       neighbor_list_;

   while(my_neighbor_list) {
       if(my_neighbor_list->nodeid ==
           neighbor->nodeid()) {
           flag = 1;
           break;
       } else {
           my_neighbor_list =
              my_neighbor_list->next;
       }
   }

   if(flag == 1) {
       //neighbour already exist do nothing
   } else {
       neighbor_list_node* nlistItem =
       (neighbor_list_node*)malloc(sizeof(neighbor
_list_node));
       nlistItem->nodeid = neighbor->nodeid();
       nlistItem->reliable = 1;
       nlistItem->next = neighbor_list_;
       neighbor_list_=nlistItem;
   }
}
```

Kode Sumber 4.17 Modifikasi existing neighbor pada Node.cc

```
Void AODV::recvReply(Packet *p) {
   ...
   Node* m_node =
      Node::get_node_by_address(this->addr());
   neighbor_list_node* my_mobile_neighbor_list;
   my_mobile_neighbor_list =
      m_node->neighbor_list_;
   int isNeighbor = 0;
   while(my_mobile_neighbor_list) {
      if (my_mobile_neighbor_list->nodeid ==
      rp->rp_dst) {
         isNeighbor = 1;
      }
      my_mobile_neighbor_list =
          my_mobile_neighbor_list->next;
   }

   if (rp->rp_hop_count == 1 && isNeighbor == 0) {
       m_node->setReliable(m_node, ih->saddr());
   } else {
       ...
       // proses pesan reply
   }
}
```

Kode Sumber 4.18 Implementasi Neighbor Information untuk deteksi dan pencegahan black hole

4.6. Implementasi Simulasi pada NS-2

Dilanjutkan dengan implementasi simulasi pada simulator yaitu NS-2. Simulasi ini akan dijalankan menggunakan kode program tcl. Hal pertama yang perlu dilakukan adalah menambahkan pengaturan untuk simulasi. Penambahan pengaturan dapat dilihat pada Kode Sumber 4.19.

Setelah melakukan pengaturan parameter simulasi, lakukan inisialisasi program untuk skenario MANET.

```
# Tipe channel
set val(chan)       Channel/WirelessChannel;
# Mode radio-propagation
set val(prop)       Propagation/TwoRayGround;
# Tipe antarmuka jaringan
set val(netif)      Phy/WirelessPhy;
# MAC type
set val(mac)        Mac/802_11;
# Tipe interface queue
set val(ifq)        Queue/DropTail/PriQueue;
# Tipe link layer
set val(ll)         LL;
# Model antenna
set val(ant)        Antenna/OmniAntenna;
# Max packet dalam ifq
set val(ifqlen)     50;
# Jumlah node
set val(nn)         20;
# Routing protocol
set val(rp)         AODV;
# X dimensi topografi
set val(x)          500;
# Y dimensi topografi
set val(y)          500;
# Waktu simulasi berakhir
set val(stop)       200.0;
```

Kode Sumber 4.19 Pengaturan parameter simulasi

```
#Create a ns simulator
set ns_ [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open $file_name.tr w]
$ns_ trace-all $tracefile

#Open the NAM trace file
set namfile [open $file_name.nam w]
$ns_ namtrace-all $namfile
$ns_ namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel
```

Kode Sumber 4.20 Pengaturan Inisialisasi NS-2

Pada Kode Sumber 4.20, ns_ merupakan variable baru untuk simulator. Pada NS-2 juga perlu adanya inisialisasi untuk pengaturan topografi. Kemudian jika ingin mencatat hasil dari simulasi, maka digunakan command trace-all yang hasilnya akan dimasukkan ke dalam file yang berekstensi .tr. Pada NS-2 juga disediakan Network Animator yang dapat menampilkan visualisasi dari simulasi NS. Network Animator atau yang biasa disebut dengan NAM, memiliki sistem kerja yang hampir sama dengan file tr. Keduanya sama-sama melakukan trace terhadap skenario pada NS. Perbedaannya terdapat pada bentuk visual untuk NAM, dan bentuk text untuk file .tr.

Setelah itu memasukkan variable yang sudah diatur pada Kode Sumber 4.19 ke dalam pengaturan sebagai berikut.

```
$ns_ node-config -adhocRouting  $val(rp) \
                 -llType        $val(ll) \
                 -macType       $val(mac) \
                 -ifqType       $val(ifq) \
                 -ifqLen        $val(ifqlen) \
                 -antType       $val(ant) \
                 -propType      $val(prop) \
                 -phyType       $val(netif) \
                 -channel       $chan \
                 -topoInstance  $topo \
                 -agentTrace    ON \
                 -routerTrace   ON \
                 -macTrace      ON \
                 -movementTrace ON
```

Kode Sumber 4.21 Pengaturan parameter mobile node

Hal selanjutnya adalah dengan melakukan definisi terhadap node-node yang akan digunakan, penentuan posisi awal, pemberian warna untuk simulasi pada NAM, pemberian label untuk masing-masing node, dan pembacaan file hasil generate pergerakan node seperti yang telah dijelaskan pada bab 4.2.

Kemudian melakukan definisi untuk koneksi. Koneksi yang digunakan pada kasus ini adalah UDP. Kode program untuk konfigurasi koneksi UDP adalah sebagai berikut.

```
set udp0 [new Agent/UDP]
$ns_ attach-agent $node_(48) $udp0
set null1 [new Agent/Null]
$ns_ attach-agent $node_(49) $null1
$ns_ connect $udp0 $null1
$udp0 set packetSize_ 1500
```

Kode Sumber 4.22 Pengaturan koneksi UDP

Dan aplikasi yang digunakan pada koneksi UDP adalah CBR. Didalam pengaturan CBR, terdapat definisi ukuran paket yang dikirimkan oleh sumber, iterval waktu pengiriman paket, dan kapan simulasi ini mulai hingga simulasi ini berakhir. Ukuran paket untuk simulasi ini diatur 1024 bytes, interval pengiriman 1 detik, waktu mulai pada detik ke-1, dan waktu berakhir pada detik ke-300.

```
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 1024
$cbr0 set interval 1
$cbr0 set random_ null
$ns_ at 1.0 "$cbr0 start"
$ns_ at 300.0 "$cbr0 stop"
```

Kode Sumber 4.23 Pengaturan CBR application

Kemudian untuk mengakhiri simulasi ini, perlu adanya penambahan kode program seperti Kode Sumber 4.24.

Pada Kode Sumber 4.24, ns memanggil fungsi finish yaitu yang berisi penutupan penggunaan file tempat menyimpan data trace .tr dan nam.

Pada akhirnya, implementasi pada NS-2 telah selesai, dan untuk menjalankan skenario tersebut perlu kode program sebagai berikut yang dijalankan pada terminal.

Ketika skenario selesai dijalankan, maka aplikasi NAM akan otomatis muncul ke layar komputer yang menampilkan hasil menjalankan skenario pada NS-2 dengan kode program exec nam.

```
proc finish {} {
    global ns_ tracefile namfile
    $ns_ flush-trace
    close $tracefile
    close $namfile
    exec nam $file_name.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns_ at $val(stop) "\$node_($i) reset"
}
$ns_ at $val(stop) "$ns_ nam-end-wireless
$val(stop)"
$ns_ at $val(stop) "finish"
$ns_ at $val(stop) "puts \"done\" ; $ns_ halt"
$ns_ run
```

Kode Sumber 4.24 Termination skenario NS-2

```
$ ns namafile.tcl
```

Kode Sumber 4.25 Menjalankan skenario NS-2

BAB V
PENGUJIAN DAN EVALUASI

Bab ini membahas mengenai pengujian dari skenario NS-2 yang telah dibuat. Pengujian fungsionalitas akan dibagi ke dalam beberapa skenario pengujian.

5.1. Lingkungan Pengujian

Uji coba dilakukan pada laptop yang telah terpasang satu buah sistem operasi yaitu Linux Mint. Spesifikasi komputer yang digunakan ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Komputer yang Digunakan

Komponen	**Spesifikasi**
CPU	Processor AMD FX Quad Core CPU @3.30GHz;
Sistem Operasi	Sistem Operasi Linux Mint 17 64 bit (NS-2, Data Routing Information, Cross Checking)
Memori	4 GB DDR3
Graphics Card	AMD Radeon R7 Graphics
Media Penyimpanan	500 GB

5.2. Kriteria Pengujian

Pengujian pada skenario yang dihasilkan oleh NS-2 menggunakan beberapa kriteria. Pada Tabel 5.2 menunjukkan kriteria-kriteria yang ditentukan di dalam skenario.

Tabel 5.2 Kriteria Pengujian

Kriteria	**Spesifikasi**
Skenario	MANET
Jumlah Node	50, 60, 70, 80, 90, 100
Jumlah Node Black Hole	3, 6, 10

Waktu simulasi	200 detik
Protokol *Routing*	AODV

Untuk mendapatkan hasil yang akurat, maka dilakukan pengujian sebanyak sepuluh kali pada kombinasi jumlah node dan jumlah black hole. Kemudian dipilih lima data terbaik (data yang stabil) dan dihitung rata-rata PDR dan Error Rate. Pengujian ini menggunakan penghitungan PDR dan Error Rate.

5.3. Pengujian

5.3.1. Nilai PDR dengan 3 Black Hole

Nilai PDR dari pengujian pada lingkungan MANET untuk jumlah black hole 3 dengan jumlah node 50, 60, 70, 80, 90, dan 100 dapat dilihat sebagai berikut.

Tabel 5.3 PDR dengan 3 Black Hole dan 50 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	80,12%	0,00%	71,00%
2.	80,33%	11,65%	30,65%
3.	64,54%	31,34%	31,34%
4.	79,40%	72,88%	72,88%
5.	70,72%	48,39%	26,15%
Rata-rata	***75,02%***	***32,85%***	***46,40%***

Tabel 5.4 PDR dengan 3 Black Hole dan 60 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	76,67%	25,96%	42,15%
2.	77,75%	0,00%	11,22%
3.	59,74%	0,01%	75,36%
4.	72,49%	64,06%	74,54%
5.	73,16%	0,47%	27,65%
Rata-rata	***71,96%***	***18,10%***	***46,18%***

Tabel 5.5 PDR dengan 3 Black Hole dan 70 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	67,11%	0,00%	45,24%
2.	66,82%	7,65%	7,67%
3.	69,50%	0,00%	69,25%
4.	82,89%	57,29%	57,29%
5.	70,84%	0,00%	43,08%
Rata-rata	***71,43%***	***12,99%***	***44,51%***

Tabel 5.6 PDR dengan 3 Black Hole dan 80 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	80,63%	0,00%	30,54%
2.	53,86%	0,00%	40,67%
3.	83,03%	0,00%	42,06%
4.	70,90%	0,00%	49,56%
5.	77,54%	0,00%	50,56%
Rata-rata	***73,19%***	***0,00%***	***42,68%***

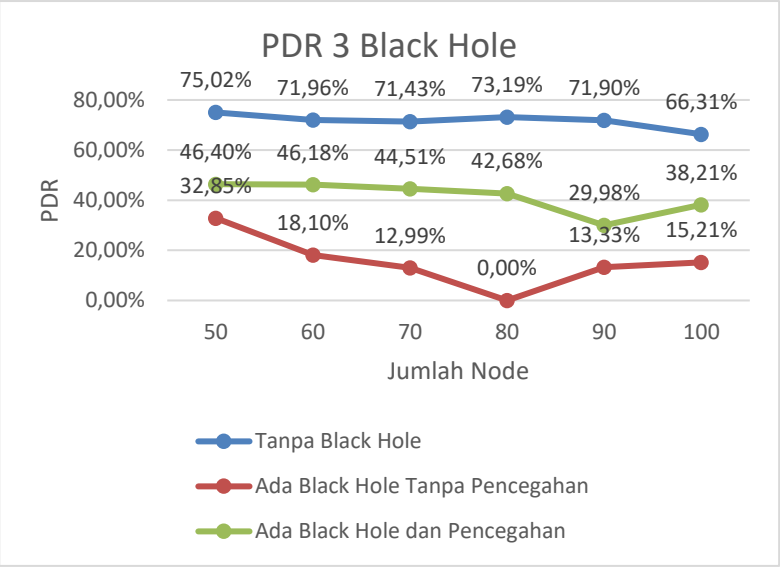
Tabel 5.7 PDR dengan 3 Black Hole dan 90 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	76,78%	0,00%	0,00%
2.	60,12%	18,73%	48,86%
3.	79,70%	0,00%	41,50%
4.	79,49%	32,94%	32,96%
5.	63,42%	14,98%	26,56%
Rata-rata	***71,90%***	***13,33%***	***29,98%***

Tabel 5.8 PDR dengan 3 Black Hole dan 100 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	63,02%	0,00%	39,94%
2.	62,03%	0,00%	41,68%
3.	52,42%	0,00%	17,47%
4.	75,71%	0,00%	15,38%
5.	78,38%	76,06%	76,56%
Rata-rata	***66,31%***	***15,21%***	***38,21%***

Dari enam tabel di atas didapatkan rata-rata untuk setiap skenario dan jika digambarkan dalam grafik garis akan seperti berikut.



5.3.2. Nilai PDR dengan 6 Black Hole

Nilai PDR dari pengujian pada lingkungan MANET untuk jumlah black hole 6 dengan jumlah node 50, 60, 70, 80, 90, dan 100 dapat dilihat sebagai berikut.

Tabel 5.9 PDR dengan 6 Black Hole dan 50 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	80,12%	0,00%	15,49%
2.	80,33%	17,95%	25,10%
3.	64,54%	73,82%	74,28%
4.	79,40%	72,79%	72,79%
5.	70,72%	15,39%	78,71%
Rata-rata	***75,02%***	***35,99%***	***53,28%***

Tabel 5.10 PDR dengan 6 Black Hole dan 60 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	76,67%	20,34%	20,49%
2.	77,75%	0,07%	73,20%
3.	59,74%	8,16%	80,93%
4.	72,49%	61,78%	67,70%
5.	73,16%	0,00%	73,08%
Rata-rata	***71,96%***	***18,07%***	***63,08%***

Tabel 5.11 PDR dengan 6 Black Hole dan 70 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	67,11%	9,80%	54,12%
2.	66,82%	17,17%	17,19%
3.	69,50%	18,99%	62,83%
4.	82,89%	57,26%	57,29%
5.	70,84%	0,00%	6,46%
Rata-rata	***71,43%***	***20,65%***	***39,58%***

Tabel 5.12 PDR dengan 6 Black Hole dan 80 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	82,82%	0,00%	82,08%
2.	80,67%	0,00%	10,44%
3.	45,26%	82,92%	42,90%
4.	51,24%	41,73%	42,13%
5.	61,19%	0,00%	54,92%
Rata-rata	***64,23%***	***24,93%***	***46,49%***

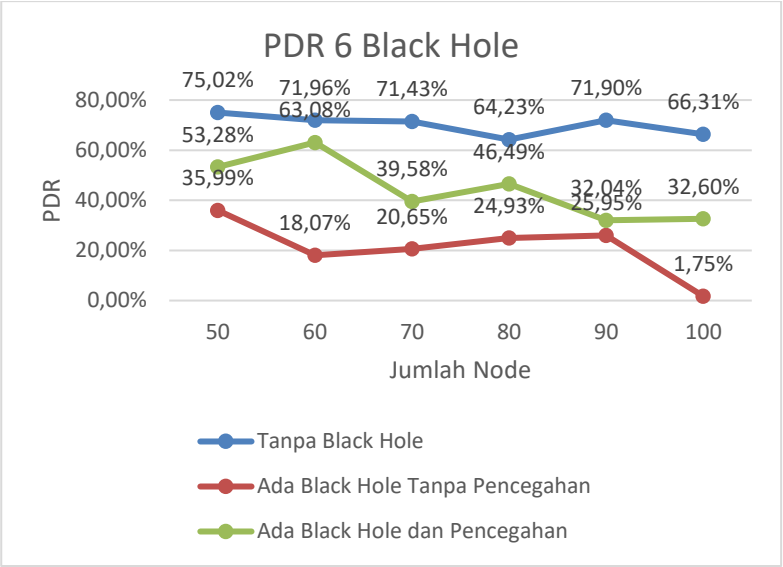
Tabel 5.13 PDR dengan 6 Black Hole dan 90 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	76,78%	0,00%	30,00%
2.	60,12%	18,72%	19,15%
3.	79,70%	47,62%	47,64%
4.	79,49%	32,95%	32,94%
5.	63,42%	30,45%	30,49%
Rata-rata	***71,90%***	***25,95%***	***32,04%***

Tabel 5.14 PDR dengan 6 Black Hole dan 100 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	63,02%	0,00%	33,47%
2.	62,03%	8,74%	48,95%
3.	52,42%	0,00%	0,00%
4.	75,71%	0,00%	0,00%
5.	78,38%	0,00%	80,60%
Rata-rata	***66,31%***	***1,75%***	***32,60%***

Dari enam tabel di atas didapatkan rata-rata untuk setiap skenario dan jika digambarkan dalam grafik garis akan seperti berikut.



5.3.3. Nilai PDR dengan 10 Black Hole

Nilai PDR dari pengujian pada lingkungan MANET untuk jumlah black hole 10 dengan jumlah node 50, 60, 70, 80, 90, dan 100 dapat dilihat sebagai berikut.

Tabel 5.15 PDR dengan 10 Black Hole dan 50 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	80,12%	0,00%	24,99%
2.	80,33%	14,93%	14,94%
3.	64,54%	31,30%	31,32%
4.	79,40%	0,00%	38,90%
5.	70,72%	11,25%	11,24%
Rata-rata	***75,02%***	***11,49%***	***24,28%***

Tabel 5.16 PDR dengan 10 Black Hole dan 60 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	76,67%	0,00%	20,96%
2.	77,75%	0,00%	33,72%
3.	59,74%	8,14%	21,69%
4.	72,49%	6,40%	7,12%
5.	73,16%	0,00%	14,53%
Rata-rata	***71,96%***	***2,91%***	***19,60%***

Tabel 5.17 PDR dengan 10 Black Hole dan 70 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	67,11%	17,11%	17,14%
2.	66,82%	0,24%	17,18%
3.	69,50%	0,00%	69,14%
4.	82,89%	0,00%	56,04%
5.	70,84%	0,00%	0,47%
Rata-rata	***71,43%***	***3,47%***	***31,99%***

Tabel 5.18 PDR dengan 10 Black Hole dan 80 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	42,32%	1,96%	82,89%
2.	61,08%	0,00%	10,25%
3.	71,28%	0,00%	28,81%
4.	59,21%	0,00%	0,00%
5.	82,16%	0,00%	24,46%
Rata-rata	***63,21%***	***0,39%***	***29,28%***

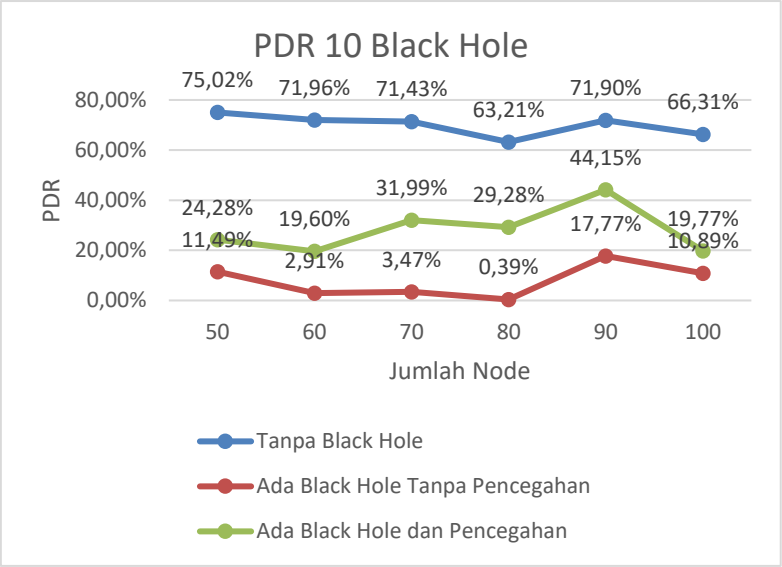
Tabel 5.19 PDR dengan 10 Black Hole dan 90 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	76,78%	0,00%	81,48%
2.	60,12%	0,00%	2,65%
3.	79,70%	0,00%	47,66%
4.	79,49%	81,82%	81,89%
5.	63,42%	7,02%	7,05%
Rata-rata	***71,90%***	***17,77%***	***44,15%***

Tabel 5.20 PDR dengan 10 Black Hole dan 100 Node

No.	Tanpa Black Hole	Ada Black Hole Tanpa Pencegahan	Ada Black Hole dan Pencegahan
1.	63,02%	0,00%	15,10%
2.	62,03%	0,00%	28,17%
3.	52,42%	0,00%	0,00%
4.	75,71%	54,43%	54,57%
5.	78,38%	0,00%	1,00%
Rata-rata	***66,31%***	***10,89%***	***19,77%***

Dari enam tabel di atas didapatkan rata-rata untuk setiap skenario dan jika digambarkan dalam grafik garis akan seperti berikut.



5.3.4. Error Rate dengan 3 Black Hole

Persentase paket yang masuk ke dalam black hole dari pengujian pada lingkungan MANET untuk jumlah black hole 3 dengan jumlah node 50, 60, 70, 80, 90, dan 100 dapat dilihat sebagai berikut.

Tabel 5.21 Error Rate dengan 3 Black Hole dan 50 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	83,07%	3,46%
2.	65,87%	49,01%
3.	41,02%	43,98%
4.	6,37%	6,37%
5.	20,06%	36,32%
Rata-rata	***43,28%***	***27,83%***

Tabel 5.22 Error Rate dengan 3 Black Hole dan 60 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	53,12%	36,59%
2.	87,79%	63,05%
3.	1,77%	1,33%
4.	8,78%	2,62%
5.	81,56%	47,83%
Rata-rata	***46,60%***	***30,28%***

Tabel 5.23 Error Rate dengan 3 Black Hole dan 70 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	100,00%	35,03%
2.	88,51%	73,45%
3.	85,05%	8,97%
4.	24,40%	24,40%
5.	100,00%	35,73%
Rata-rata	***79,59%***	***35,52%***

Tabel 5.24 Error Rate dengan 3 Black Hole dan 80 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	83,24%	80,54%
2.	100,00%	40,67%
3.	77,10%	82,06%
4.	82,82%	59,56%
5.	100,00%	50,56%
Rata-rata	***88,63%***	***62,68%***

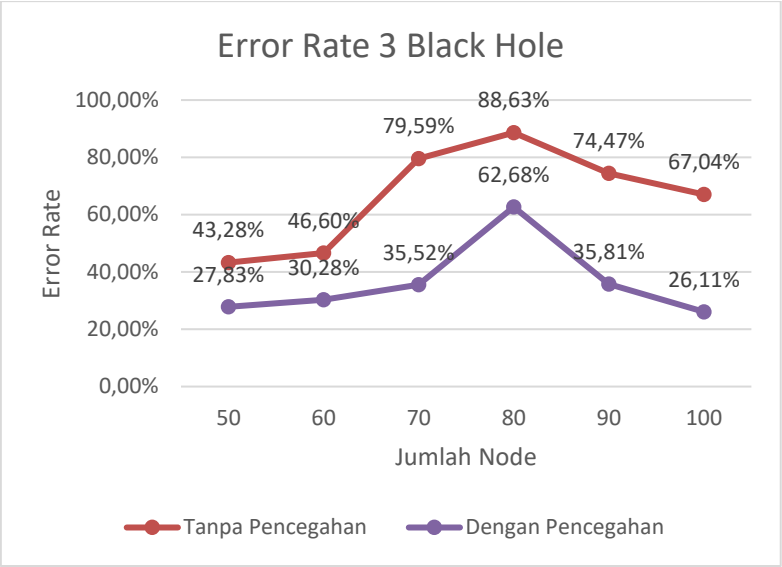
Tabel 5.25 Error Rate dengan 3 Black Hole dan 90 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	82,65%	69,26%
2.	60,72%	24,60%
3.	92,73%	22,82%
4.	56,81%	23,52%
5.	79,43%	38,83%
Rata-rata	***74,47%***	***35,81%***

Tabel 5.26 Error Rate dengan 3 Black Hole dan 100 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	81,21%	19,85%
2.	100,00%	16,80%
3.	74,57%	56,41%
4.	78,41%	37,51%
5.	1,00%	0,00%
Rata-rata	***67,04%***	***26,11%***

Dari enam tabel di atas didapatkan rata-rata untuk setiap skenario dan jika digambarkan dalam grafik garis akan seperti berikut.



5.3.5. Error Rate dengan 6 Black Hole

Persentase paket yang masuk ke dalam black hole dari pengujian pada lingkungan MANET untuk jumlah black hole 6 dengan jumlah node 50, 60, 70, 80, 90, dan 100 dapat dilihat sebagai berikut.

Tabel 5.27 Error Rate dengan 6 Black Hole dan 50 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	84,15%	61,87%
2.	70,02%	48,62%
3.	8,89%	3,99%
4.	10,17%	10,16%
5.	67,18%	1,30%
Rata-rata	***48,08%***	***25,19%***

Tabel 5.28 Error Rate dengan 6 Black Hole dan 60 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	66,47%	52,23%
2.	82,42%	0,00%
3.	87,56%	0,50%
4.	21,17%	6,45%
5.	80,21%	8,78%
Rata-rata	***67,57%***	***13,59%***

Tabel 5.29 Error Rate dengan 6 Black Hole dan 70 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	83,29%	15,37%
2.	66,85%	48,05%
3.	68,82%	8,49%
4.	21,60%	13,67%
5.	96,73%	67,36%
Rata-rata	***67,46%***	***30,59%***

Tabel 5.30 Error Rate dengan 6 Black Hole dan 80 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	81,22%	0,00%
2.	94,93%	59,35%
3.	0,10%	0,00%
4.	0,09%	0,01%
5.	100,00%	0,00%
Rata-rata	***55,27%***	***11,87%***

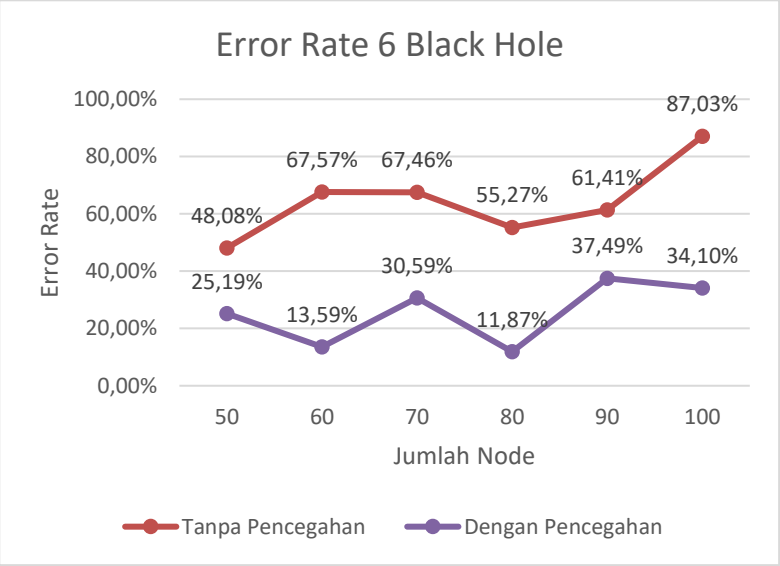
Tabel 5.31 Error Rate dengan 6 Black Hole dan 90 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	100,00%	55,44%
2.	74,47%	52,86%
3.	39,81%	21,22%
4.	57,29%	29,08%
5.	35,50%	28,86%
Rata-rata	***61,41%***	***37,49%***

Tabel 5.32 Error Rate dengan 6 Black Hole dan 100 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	97,19%	24,74%
2.	78,38%	18,04%
3.	83,27%	55,81%
4.	83,93%	71,94%
5.	92,36%	0,00%
Rata-rata	***87,03%***	***34,10%***

Dari enam tabel di atas didapatkan rata-rata untuk setiap skenario dan jika digambarkan dalam grafik garis akan seperti berikut.



5.3.6. Error Rate dengan 10 Black Hole

Persentase paket yang masuk ke dalam black hole dari pengujian pada lingkungan MANET untuk jumlah black hole 10 dengan jumlah node 50, 60, 70, 80, 90, dan 100 dapat dilihat sebagai berikut.

Tabel 5.33 Error Rate dengan 10 Black Hole dan 50 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	88,04%	45,69%
2.	76,95%	64,79%
3.	60,43%	42,99%
4.	87,21%	25,62%
5.	77,19%	45,56%
Rata-rata	***77,97%***	***44,93%***

Tabel 5.34 Error Rate dengan 10 Black Hole dan 60 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	95,04%	53,51%
2.	71,68%	42,35%
3.	82,28%	49,68%
4.	73,60%	52,26%
5.	89,97%	58,47%
Rata-rata	***82,51%***	***51,25%***

Tabel 5.35 Error Rate dengan 10 Black Hole dan 70 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	51,13%	64,20%
2.	81,28%	52,71%
3.	94,17%	11,70%
4.	97,00%	24,21%
5.	97,34%	66,74%
Rata-rata	***84,19%***	***43,91%***

Tabel 5.36 Error Rate dengan 10 Black Hole dan 80 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	94,98%	0,00%
2.	100,00%	65,41%
3.	82,53%	50,28%
4.	83,16%	55,56%
5.	100,00%	39,21%
Rata-rata	***92,13%***	***42,09%***

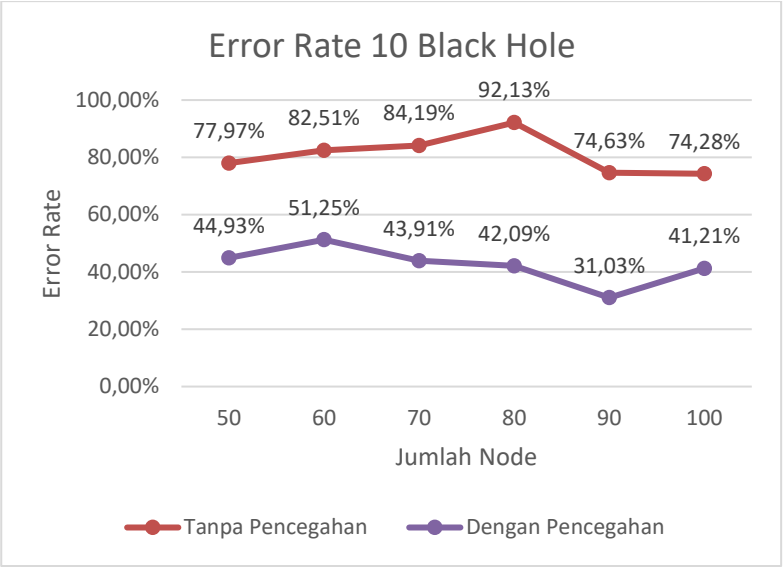
Tabel 5.37 Error Rate dengan 10 Black Hole dan 90 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	89,90%	0,00%
2.	100,00%	68,25%
3.	97,41%	22,17%
4.	0,15%	0,00%
5.	85,67%	64,73%
Rata-rata	***74,63%***	***31,03%***

Tabel 5.38 Error Rate dengan 10 Black Hole dan 100 Node

No.	Tanpa Pencegahan	Dengan Pencegahan
1.	97,62%	60,02%
2.	90,09%	28,09%
3.	89,43%	61,25%
4.	0,09%	0,00%
5.	94,17%	56,70%
Rata-rata	***74,28%***	***41,21%***

Dari enam tabel di atas didapatkan rata-rata untuk setiap skenario dan jika digambarkan dalam grafik garis akan seperti berikut.



5.3.7. Perubahan Kenaikan PDR

Dari data tabel-tabel PDR di atas, dapat disimpulkan perbandingan perubahan kenaikan PDR tanpa pencegahan dan dengan pencegahan.

Tabel 5.39 Perubahan PDR 3 Black Hole

No.	Jumlah Node	Kenaikan PDR
1.	50	13,55%
2.	60	28,08%
3.	70	31,52%
4.	80	42,68%
5.	90	16,65%
6.	100	22,99%

Tabel 5.40 Perubahan PDR 6 Black Hole

No.	Jumlah Node	Kenaikan PDR
1.	50	17,28%
2.	60	45,01%
3.	70	18,93%
4.	80	21,57%
5.	90	6,10%
6.	100	30,86%

Tabel 5.41 Perubahan PDR 10 Black Hole

No.	Jumlah Node	Kenaikan PDR
1.	50	12,79%
2.	60	16,70%
3.	70	28,52%
4.	80	28,89%
5.	90	26,38%
6.	100	8,88%

Sehingga dengan menggunakan metode pendeteksian dan pencegahan dengan memanfaatkan Hop Count dan Neighbor Information mendapatkan hasil kenaikan PDR dengan nilai kenaikan paling rendah 6,10% dan kenaikan paling tinggi 45,01% dengan skenario jumlah black hole 3, 6, 10 dan jumlah node 50, 60, 70, 80, 90, dan 100.

5.3.8. Perubahan Penurunan Error Rate

Dari data tabel-tabel Error Rate di atas, dapat disimpulkan perbandingan perubahan penurunan Error Rate tanpa pencegahan dan dengan pencegahan.

Tabel 5.42 Penurunan Error Rate 3 Black Hole

No.	Jumlah Node	Penurunan Error Rate
1.	50	15,45%
2.	60	16,32%
3.	70	44,08%
4.	80	25,95%
5.	90	38,66%
6.	100	40,92%

Tabel 5.43 Penurunan Error Rate 6 Black Hole

No.	Jumlah Node	Penurunan Error Rate
1.	50	22,89%
2.	60	53,98%
3.	70	36,87%
4.	80	43,40%
5.	90	23,92%
6.	100	52,92%

Tabel 5.44 Penurunan Error Rate 10 Black Hole

No.	Jumlah Node	Penurunan Error Rate
1.	50	33,04%
2.	60	31,26%
3.	70	40,28%
4.	80	50,04%
5.	90	43,59%
6.	100	33,07%

Sehingga dengan menggunakan metode pendeteksian dan pencegahan dengan memanfaatkan Hop Count dan Neighbor Information mendapatkan hasil penurunan Error Rate dengan nilai penurunan paling rendah 15,45% dan penurunan paling tinggi 53,98% dengan skenario jumlah black hole 3, 6, 10 dan jumlah node 50, 60, 70, 80, 90, dan 100.

BAB VI
PENUTUP

Pada bab ini diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Kesimpulan yang dapat diambil dalam Tugas Akhir ini adalah sebagai berikut:

- Dari hasil uji coba dapat disimpulkan bahwa setelah adanya metode pencegahan, nilai PDR dapat lebih tinggi dari nilai PDR sebelum adanya metode pencegahan dengan perbedaan ketinggian 45,01% (diambil dari nilai maksimal).
- Nilai persentase error/jumlah paket yang masuk ke black hole (paket drop) juga dapat lebih rendah dari sebelum adanya metode pencegahan dengan perbedaan ketinggian 53,98% (diambil dari nilai maksimal).

6.2. Saran

Dalam pengerjaan Tugas Akhir ini terdapat beberapa saran untuk perbaikan serta pengembangan sistem yang telah dikerjakan dikarenakan adanya batasan pada metode pendeteksian dan pencegahan ini, bahwa black hole tidak dapat terdeteksi ketika posisi black hole berada satu tetangga dengan node sumber atau node intermediate, maka perlu adanya optimasi untuk mengatasi hal ini. Optimasi yang dimaksudkan adalah:

- Melakukan pengecekan tambahan yaitu dilakukannya pengecekan alamat IP dari pengirim pesan balasan (sendReply) karena dalam pesan balasan juga terdapat informasi tentang alamat IP dari node tujuan.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

[1] Suprapto, Tommy. *Pengantar Teori dan Manajemen Komunikas*, Yogyakarta: Medpress. 2009.

[2] Zamroni, Mohammad. Perkembangan Teknologi Komunikasi dan Dampaknya Terhadap Kehidupan, Yogyakarta: Jurnal Dakwah, Vol. X No. 2, Juli-Desember 2009.

[3] Addison Wesley. *Chapter 6, Sections 6.1-6.3, 6.5 Ad Hoc Networking*, Perkins: 2001

[4] Mistry, Nital, dan Jinwala, C Devesh. *Improving AODV Protocol against Black Hole Attacks*, Journal: 2010.

[5] Paul, Suman. *Introduction to MANET and Clustering in MANET*, Hamburg: Diplomica Publishing GmbH. 2015.

[6] Stallings, William. *Cryptography and Network Security Principles and Practice 6th Edition*. Pearson Education, Inc (2014).

[7] Modi, Nirali. *Prevention Of Black hole Attack using AODV Routing Protocol in MANET*, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 3254 – 3258. 2014.

[8] Yi-Chun Hu, Adrian Perrig. *A Survey of Secure Wireless Ad Hoc Routing***,** IEEE Security and Privacy, 1540-7993/04/$20.00 © 2004 *IEEE*.

[9] Wang, Zehua. *Implementation of the AODV Routing Protocol in ns2 for Multi-hop Wireless Networks*.

[10] Umang S, Reddy BVR, Hoda MN. *Enhanced Intrusion Detection System for Malicious Node Detection in Ad Hoc Routing Protocols using Minimal Energy Consumption*. IET Communications. 2010.

[11] Wu B, Chen J, Wu J, Cardei M. *A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks*. In Wireless Network Security. on Signals and Communication

Technology. Edited by: Xiao Y, Shen X, Du D-Z. Springer, New York; 2007.

[12] Marti S, Giuli TJ, Lai K, Baker M. *Mitigating Routing Misbehavior in Mobile Ad Hoc Networks*. Paper presented at the 6th annual International Conference on Mobile Computing and Networking. Boston, Massachusetts, 6–11 August 2000.

[13] Tseng Y-C, Jiang J-R, Lee J-H. *Secure Bootstrapping and Routing in an IPv6-based Ad Hoc Network*. Journal of Internet Technology 2004.

[14] Hu Y-C, Perrig A. *Survey of Secure Wireless Ad Hoc Routing*. IEEE Security & Privacy 2004.

[15] Raja Mahmood RA, Khan AI. *A Survey on Detecting Black Hole Attack in AODV-based Mobile Ad Hoc Networks*. Paper presented at the International Symposium on High Capacity Optical Networks and Enabling Technologies, Dubai, United Arab Emirates, 18–20 November 2007.

[16] Saini A, Kumar H. *Comparison between Various Black Hole Detection Techniques in MANET*. Paper presented at the National Conference on Computational Instrumentation, Chandigarh, India, 19–20 March 2010.

[17] T. Nandhini, V. Sandhya: *Detection of Grouped Malicious Nodes to Avoid Black Hole Attack in Mobile Ad-Hoc Network*. IJIRST, December 2014.

BIODATA PENULIS

Fany Agriansyah Rosyada, biasa dipanggil Fany, lahir di Madiun pada 28 Oktober 1994. Penulis adalah anak kedua dari 3 bersaudara dan dibesarkan di Madiun, Jawa Timur. Penulis menempuh pendidikan formal di SDN 1 Sidomoro Gresik (2001-2004), SDN Banjarejo Madiun (2004-2007), SMPN 1 Madiun (2007-2010), SMAN 2 Madiun (2010-2013). Pada tahun 2013, penulis memulai pendidikan S1 Jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya angkatan 2013 yang terdaftar dengan NRP 5113100076.

Di jurusan Teknik Informatika, penulis mengambil bidang minat Arsitektur Jaringan Komputer (AJK) dan memiliki ketertarikan di bidang *Operating System Linux*, Jaringan Komputer. Selain bidang minat AJK, penulis juga memiliki ketertarikan di bidang pengembangan aplikasi baik web maupun mobile android. Selama menempuh kuliah, penulis juga aktif dalam organisasi kemahasiswaan seperti Himpunan Mahasiswa Teknik Computer (HMTC) dan Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi. Penulis dapat dihubungi melalui alamat email fanyagriansyah@gmail.com.

[Halaman ini sengaja dikosongkan]