PENDETEKSIAN SERANGAN *BLACK HOLE* TERHADAP PROTOKOL *ROUTING AD HOC ON-DEMAND DISTANCE* *VECTOR* (AODV) PADA *MOBILE* *AD HOC NETWORK* (MANET)

SKRIPSI

Untuk memenuhi sebagian persyaratan   
memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Taufiq Aryadi

NIM: 155150201111173



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2019

PENGESAHAN

PENDETEKSIAN SERANGAN *BLACK HOLE* TERHADAP PROTOKOL *ROUTING AD HOC ON-DEMAND DISTANCE VECTOR* (AODV) PADA *MOBILE AD HOC NETWORK* (MANET)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan

memperoleh gelar Sarjana Komputer

Disusun Oleh :

Muhammad Taufiq Aryadi

NIM: 155150201111173

Skripsi ini telah diuji dan dinyatakan lulus pada

17 Juli 2019

Telah diperiksa dan disetujui oleh:

|  |  |
| --- | --- |
| Dosen Pembimbing I  Ir. Primantara Hari Trisnawan, M.Sc.  NIP. 19680912 199403 1 002 | Dosen Pembimbing II  Reza Andria Siregar, S.T., M.Kom.  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 p*end*apat yang pernah ditulis atau diter*Bit*kan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam 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, 17 Juli 2019

­

Muhammad Taufiq Aryadi

NIM: 155150201111173

PRAKATA

Alhamdulillahirabbilalamin, puji syukur kehadirat Allah SWT atas segala limpahan rahmat dan karunianya yang telah memberikan kelancaran dan kemudahan sehingga penelitian skripsi dengan judul Pendeteksian Serangan Black Hole terhadap Protokol Routing Ad Hoc On Demand (AODV) pada Mobile Ad Hoc Network (MANET) ini dapat terselesaikan dengan baik. Penelitian ini dilakukan dalam rangka memenuhi mata kuliah skripsi untuk memperoleh gelar sarjana komputer di Fakultas Ilmu Komputer Universitas Brawijaya.

Dengan terselesaikannya tugas akhir penelitian ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang terlibat dalam penyusunan penelitian ini. Pihak-pihak yang telah banyak berkorban, membantu, mendoakan, membimbing, dan mendukung penulis selama penelitian ini berlangsung hingga penelitian ini dapat terselesaikan khususnya kepada:

1. Allah SWT.
2. Kedua orangtua penulis yang secara ikhlas telah memberikan dukungan penuh baik dalam bentuk moril maupun materil.
3. Bapak Wayan Firdaus Mahmudi, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer, Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika.
4. Bapak Ir.Primantara Hari Trisnawan, M. Sc. dan Bapak Reza Andria Siregar, S.T., M.Kom. selaku dosen pembimbing skripsi penulis yang telah memberikan arahan, bimbingan serta dukungan hingga terselesaikannya penelitian ini.
5. Ubaidillah, Nugroho, Adit, Riyad, Usman, Bagus, Ana, Adi, Fian, Vina, Vany, Rachma, Laili, Fahmi, Vio, Agus dan K Class selaku sahabat penulis yang selalu mendampingi memberikan dukungan dan bantuan serta motivasi dalam menyelesaikan penelitian ini.
6. Seluruh pihak yang telah memberikan dukungan dan bantuan selama penelitian ini berlangsung hingga dapat terselesaikan dengan baik.

Penulis sadar dalam penyusunan tugas akhir penelitian ini masih banyak kekurangan baik dalam penyajian materi maupun pembahasan. Demi pengembangan penelitian selanjutnya dan kesempurnaan laporan ini, penulis sangat mengharapkan kritik dan saran. Semoga laporan ini dapat memberikan manfaat bagi semua pihak yang membutuhkan.

Malang, Juli 2019

Penulis

mtaufiqaryadi8@gmail.com

ABSTRAK

Muhammad Taufiq Aryadi, Pendeteksian Serangan Black Hole terhadap Protokol Routing Ad Hoc On Demand (AODV) pada Mobile Ad Hoc Network (MANET)

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

MANET adalah jaringan *ad hoc* yang tersusun dari kumpulan *node ad hoc*. *Node* pada MANET bersifat dinamis yang berarti *node* dapat bebas bergabung dan meninggalkan jaringan sehingga pembentukan *topologi* jaringan bersifat dinamis. Sifat dinamis pada MANET membuat setiap *node* dapat memperoleh informasi dengan mudah. Setiap *node* juga dapat berperilaku tidak normal seperti melakukan serangan *black hole* yang dapat mengganggu proses *routing*. Serangan *black hole* bekerja dengan menyatakan dirinya memiliki rute terpendek menuju *node* tujuan kepada *node* sumber sehingga *node* sumber akan mengirimkan paketnya melalui *node black hole*. Kemudian *node black hole* akan membuang paket yang diterimanya. Hal ini akan sangat berbahaya jika paket yang dikirimkan adalah sebuah paket yang berisi informasi-informasi penting sehingga MANET membutuhkan mekanisme keamanan untuk dapat mendeteksi dan menghindari serangan *black hole*. Sistem pendeteksian serangan *black hole* terdiri dari proses pencarian *black hole* dan melakukan pencarian rute menghindari *black hole*. Proses pencarian *black hole* dilakukan dengan mengirimkan pesan RREQ dengan alamat tujuan palsu. Apabila terdapat balasan dari pesan RREQ palsu maka *node* pengirim pesan RREP ditandai sebagai black hole. Pengiriman data dilakukan dengan memilih jalur yang terdapat pada tabel *routing* dan menghindari *node black hole*. Pada hasil pengujian dapat dilihat ketika simulasi jaringan MANET dengan kondisi terdapat deteksi *black hole* dapat mempengaruhi nilai *packet delivery ratio* menjadi lebih besar dibandingkan dengan simulasi dengan simulasi dalam kondisi terdapat *black hole*. Mekanisme pendeteksian *black hole* dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. Pada hasil pengujian dapat dilihat ketika simulasi dengan kondisi terdapat mekanisme deteksi *black hole* membuat nilai *packet loss* menjadi lebih kecil dibandingkan ketika tidak terdapat mekanisme deteksi. Nilai *packet loss* pada pergerakan *random walk* turun dari 99,9959 % dalam kondisi terdapat *black hole* menjadi 0,209308 % dalam kondisi terdapat mekanisme deteksi *black hole*.

Kata kunci: MANET, AODV, *Black Hole*, *Recive Request*, Pendeteksian

ABSTRACT

Muhammad Taufiq Aryadi, Detection of Black Hole Attacks on Ad Hoc On Demand (AODV) Routing Protocol on Mobile Ad Hoc Network (MANET)

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

*MANET is an ad hoc network which composed of ad hoc node collections. MANET nodes are dynamic, the nodes are free to join and leave the network so that the network topologies formation changes dynamically. A node could act abnormally such as doing black hole attacks and might disturb the routing process. The black hole attack works by showing that it has the shortest route to the destination node to the source node so that the source node will send the source node packet through the black hole node. Moreover, the black hole node throw the packet it has received away. This is very risky if the sent package is a package which contains important information. Therefore, MANET needs a security mechanism to be able to detect and avoid from the black hole attack. The detection system for the black hole attack consists of the black hole searching and avoiding processes. The black hole searching process is performed by sending RREQ message with fake destination address. If there is a reply from fake RREQ message, The node which sent the RREP message will be marked as a black hole. Furthermore, the route for the data transmission is done by selecting the route in the routing table and avoiding the black hole node. In the test results, it can be seen as the MANET network simulation in the condition that there was a detection of the black hole might affect the packet delivery ratio value became bigger than the simulation with the simulation in a condition of black hole in it. The mechanism of detecting the black hole was able to know the black hole address and avoid black hole route in transferring data so that the data transmission could be transferred to the destination. Also, in the test results, it can be seen when the simulation with the condition that there was a black hole detection mechanism made the packet loss value smaller than when there was no detection mechanism. It in the random walk movement packet loss from 99.9959% in the condition that there was a black hole to 0.209308% under the condition where black hole detection mechanism existed.*

*Keywords : MANET, AODV, Blackhole, Receive Request, detection*

DAFTAR ISI

[PENDETEKSIAN SERANGAN *BLACK HOLE* TERHADAP PROTOKOL *ROUTING AD HOC ON-DEMAND DISTANCE VECTOR* (AODV) PADA *MOBILE AD HOC NETWORK* (MANET) i](#_Toc15243235)

[PENGESAHAN ii](#_Toc15243236)

[PERNYATAAN ORISINALITAS iii](#_Toc15243237)

[PRAKATA iv](#_Toc15243238)

[ABSTRAK v](#_Toc15243239)

[ABSTRACT vi](#_Toc15243240)

[DAFTAR ISI vii](#_Toc15243241)

[DAFTAR TABEL xi](#_Toc15243242)

[DAFTAR GAMBAR xii](#_Toc15243243)

[BAB 1 PENDAHULUAN 1](#_Toc15243244)

[1.1 Latar Belakang 1](#_Toc15243245)

[1.2 Rumusan Masalah 3](#_Toc15243246)

[1.3 Tujuan 3](#_Toc15243247)

[1.4 Manfaat 3](#_Toc15243248)

[1.5 Batasan Masalah 4](#_Toc15243249)

[1.6 Sistematika Pembahasan 4](#_Toc15243250)

[BAB 2 LANDASAN KEPUSTAKAAN 6](#_Toc15243251)

[2.1 Penelitian Terdahulu 6](#_Toc15243252)

[2.2 Mobile Ad Hoc *Network* (MANET) 8](#_Toc15243253)

[2.2.1 Pengertian MANET 8](#_Toc15243254)

[2.2.2 Karakteristik MANET 9](#_Toc15243255)

[2.2.3 Penggunaan MANET 10](#_Toc15243256)

[2.3 Protokol *Routing* pada MANET 10](#_Toc15243257)

[2.3.1 Pengertian Protokol *Routing* 10](#_Toc15243258)

[2.3.2 Ad Hoc On-Demand *Distance* *Vector* (AODV) 11](#_Toc15243259)

[2.3.3 Destination Sequenced *Distance* *Vector* (DSDV) 14](#_Toc15243260)

[2.3.4 Zone *Routing* *Protocol* (ZRP) 15](#_Toc15243261)

[2.4 Model Pergerakan pada MANET 15](#_Toc15243262)

[2.4.1 *Random* *Waypoint* 15](#_Toc15243263)

[2.4.2 *Random* *Walk* 15](#_Toc15243264)

[2.4.3 *Random* *Direction* 16](#_Toc15243265)

[2.5 Serangan pada MANET 16](#_Toc15243266)

[2.5.1 Serangan *Black hole* 16](#_Toc15243267)

[2.5.2 Serangan Grey Hole 17](#_Toc15243268)

[2.6 Jaringan Nirkabel 17](#_Toc15243269)

[2.6.1 Pengertian Jaringan Nirkabel 17](#_Toc15243270)

[2.6.2 Standar Jaringan Nirkabel 18](#_Toc15243271)

[2.7 *Network* Simulator 3 (NS3) 18](#_Toc15243272)

[2.7.1 Pengertian NS3 18](#_Toc15243273)

[2.7.2 Konsep dan Abstraksi pada NS3 18](#_Toc15243274)

[2.8 *Network* Animator (NetAnim) 19](#_Toc15243275)

[2.9 Parameter Pengujian *Quality* *of* *Service* (QoS) 19](#_Toc15243276)

[2.9.1 Pengertian Parameter Pengujian *Quality* *of* *Service* (QoS) 19](#_Toc15243277)

[2.9.2 Packet Loss 20](#_Toc15243278)

[2.9.3 *Packet delivery ratio* (PDR) 20](#_Toc15243279)

[2.9.4 *Delay* 20](#_Toc15243280)

[2.9.5 Akurasi Deteksi 20](#_Toc15243281)

[BAB 3 METODOLOGI PENELITIAN 22](#_Toc15243282)

[3.1 Studi Literatur 22](#_Toc15243283)

[3.2 Analisis Kebutuhan 22](#_Toc15243284)

[3.3 Perancangan 22](#_Toc15243285)

[3.3.1 Perancangan Serangan *Black hole* 22](#_Toc15243286)

[3.3.2 Perancangan Deteksi Serangan *Black hole* 22](#_Toc15243287)

[3.3.3 Perancangan Parameter Pengujian 23](#_Toc15243288)

[3.3.4 Perancangan Sistem 23](#_Toc15243289)

[3.3.5 Perancangan Skenario Pengujian 23](#_Toc15243290)

[3.4 Implementasi 23](#_Toc15243291)

[3.4.1 Implementasi Serangan *Black hole* 23](#_Toc15243292)

[3.4.2 Implementasi Deteksi *Black hole* 23](#_Toc15243293)

[3.4.3 Implementasi Sistem 24](#_Toc15243294)

[3.4.4 Implementasi Skenario Pengujian 24](#_Toc15243295)

[3.5 Pembahasan Hasil Pengujian 24](#_Toc15243296)

[3.6 Penutup 24](#_Toc15243297)

[BAB 4 ANALISA KEBUTUHAN DAN PERANCANGAN 25](#_Toc15243298)

[4.1 Deskripsi Umum Sistem 25](#_Toc15243299)

[4.2 Kebutuhan Sistem 25](#_Toc15243300)

[4.2.1 Kebutuhan Fungsional 25](#_Toc15243301)

[4.2.2 Kebutuhan Non-Fungsional 25](#_Toc15243302)

[4.3 Perancangan Serangan *Black hole* 26](#_Toc15243303)

[4.4 Perancangan Deteksi Serangan *Black hole* 27](#_Toc15243304)

[4.5 Perancangan Parameter Pengujian 29](#_Toc15243305)

[4.5.1 Packet Loss 29](#_Toc15243306)

[4.5.2 Packet Delivery Ratio 29](#_Toc15243307)

[4.5.3 Delay 30](#_Toc15243308)

[4.5.4 Akurasi Deteksi 30](#_Toc15243309)

[4.6 Perancangan Sistem 31](#_Toc15243310)

[4.7 Perancangan Skenario Pengujian 32](#_Toc15243311)

[4.7.1 Posisi Node Black Hole 32](#_Toc15243312)

[4.7.2 Tipe Pergerakan 33](#_Toc15243313)

[4.7.3 Jumlah *Node* 33](#_Toc15243314)

[BAB 5 IMPLEMENTASI 35](#_Toc15243315)

[5.1 Instalasi NS 3 35](#_Toc15243316)

[5.2 Implementasi Serangan *Black hole* 35](#_Toc15243317)

[5.3 Implementasi Deteksi Serangan *Black hole* 37](#_Toc15243318)

[5.4 Implementasi Sistem 39](#_Toc15243319)

[5.5 Implementasi Skenario Pengujian 40](#_Toc15243320)

[5.5.1 Posisi Node Black Hole 40](#_Toc15243321)

[5.5.2 Tipe Pergerakan 52](#_Toc15243322)

[*5.5.3* Jumlah *Node* 65](#_Toc15243323)

[BAB 6 HASIL PENGUJIAN DAN PEMBAHASAN 78](#_Toc15243324)

[6.1 Kinerja Protokol Terhadap Posisi Black Hole 78](#_Toc15243325)

[6.1.1 Hasil Pengujian 78](#_Toc15243326)

[6.1.2 Pembahasan 78](#_Toc15243327)

[6.2 Kinerja Protokol Terhadap Variasi Tipe Pergerakan 82](#_Toc15243328)

[6.2.1 Hasil Pengujian 82](#_Toc15243329)

[6.2.2 Pembahasan 83](#_Toc15243330)

[6.3 Kinerja Protokol Terhadap Variasi Jumlah *Node* 86](#_Toc15243331)

[6.3.1 Hasil Pengujian 86](#_Toc15243332)

[6.3.2 Pembahasan 86](#_Toc15243333)

[BAB 7 PENUTUP 90](#_Toc15243334)

[7.1 Kesimpulan 90](#_Toc15243335)

[7.2 Saran 91](#_Toc15243336)

[DAFTAR REFERENSI 92](#_Toc15243337)

DAFTAR TABEL

[Tabel 2.1 Kajian pustaka 8](#_Toc13078722)

[Tabel 2.2 Pesan RREQ 12](#_Toc13078723)

[Tabel 2.3 Pesan RREP 12](#_Toc13078724)

[Tabel 2.4 Pesan RRER 12](#_Toc13078725)

[Tabel 2.5 Tabel *routing* pada setiap *node* 13](#_Toc13078726)

[Tabel 4.1 Kebutuhan fungsional 25](#_Toc13078727)

[Tabel 4.2 Kebutuhan perangkat lunak 26](#_Toc13078728)

[Tabel 4.3 Kebutuhan p*end*ukung 26](#_Toc13078729)

[Tabel 4.4 Konfigurasi implementasi sistem 31](#_Toc13078730)

[Tabel 4.5 Konfigurasi implementasi variasi posisi black hole 32](#_Toc13078731)

[Tabel 4.6 Konfigurasi implementasi variasi tipe pergerakan 33](#_Toc13078732)

[Tabel 4.7 Konfigurasi implementasi variasi jumlah *node* 34](#_Toc13078733)

[Tabel 6.1 Hasil pengujian variasi posisi black hole 78](#_Toc13078734)

[Tabel 6.2 Hasil pengujian variasi tipe pergerakan 82](#_Toc13078735)

[Tabel 6.3 Hasil pengujian variasi jumlah *node* 86](#_Toc13078736)

DAFTAR GAMBAR

[Gambar 2.1 *Flow*chart ketika *node* menerima pesan RREP 7](#_Toc13078737)

[Gambar 2.2 Contoh jaringan MANET 9](#_Toc13078738)

[Gambar 2.3 Jenis protokol *routing* MANET 11](#_Toc13078739)

[Gambar 2.4 Proses *route* *discovery* 13](#_Toc13078740)

[Gambar 2.5 Proses *route* *maintenance* 14](#_Toc13078741)

[Gambar 2.6 Mobilitas *node* *random* *waypoint* 15](#_Toc13078742)

[Gambar 2.7 Mobilitas *node* *random* *walk* 16](#_Toc13078743)

[Gambar 2.8 Mobilitas *node* *random* *direction* 16](#_Toc13078744)

[Gambar 2.9 Proses serangan *black hole* 17](#_Toc13078745)

[Gambar 2.10 Tampilan NetAnim 19](#_Toc13078746)

[Gambar 4.1 Penandaan *black hole* 27](#_Toc13078747)

[Gambar 4.2 Pendeteksian *black hole* 28](#_Toc13078748)

[Gambar 5.1 Tampilan NetAnim 20 *node* kondisi normal 41](#_Toc13078749)

[Gambar 5.2 Hasil NS3 20 *node* kondisi normal 41](#_Toc13078750)

[Gambar 5.3 Tampilan NetAnim 20 *node* kondisi terdapat black hole 42](#_Toc13078751)

[Gambar 5.4 Hasil NS3 20 *node* kondisi terdapat black hole 43](#_Toc13078752)

[Gambar 5.5 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole 43](#_Toc13078753)

[Gambar 5.6 Hasil NS3 20 *node* kondisi terdapat deteksi black hole 44](#_Toc13078754)

[Gambar 5.7 Tampilan NetAnim 20 *node* kondisi terdapat black hole 45](#_Toc13078755)

[Gambar 5.8 Hasil NS3 20 *node* kondisi terdapat black hole 45](#_Toc13078756)

[Gambar 5.9 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole 46](#_Toc13078757)

[Gambar 5.10 Hasil NS3 20 *node* kondisi terdapat deteksi black hole 47](#_Toc13078758)

[Gambar 5.11 Tampilan NetAnim 20 *node* kondisi terdapat black hole 47](#_Toc13078759)

[Gambar 5.12 Hasil NS3 20 *node* kondisi terdapat black hole 48](#_Toc13078760)

[Gambar 5.13 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole 49](#_Toc13078761)

[Gambar 5.14 Hasil NS3 20 *node* kondisi terdapat deteksi black hole 49](#_Toc13078762)

[Gambar 5.15 Tampilan NetAnim 20 *node* kondisi terdapat black hole 50](#_Toc13078763)

[Gambar 5.16 Hasil NS3 20 *node* kondisi terdapat black hole 51](#_Toc13078764)

[Gambar 5.17 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole 51](#_Toc13078765)

[Gambar 5.18 Hasil NS3 20 *node* kondisi terdapat deteksi black hole 52](#_Toc13078766)

[Gambar 5.19 Tampilan NetAnim *random* *waypoint* kondisi normal 53](#_Toc13078767)

[Gambar 5.20 Hasil NS3 *random* *waypoint* kondisi normal 53](#_Toc13078768)

[Gambar 5.21 Tampilan NetAnim *random* *waypoint* kondisi terdapat black hole 54](#_Toc13078769)

[Gambar 5.22 Hasil NS3 *random* *waypoint* kondisi terdapat black hole 55](#_Toc13078770)

[Gambar 5.23 Tampilan NetAnim *random* *waypoint* kondisi terdapat deteksi black hole 55](#_Toc13078771)

[Gambar 5.24 Hasil NS3 *random* *waypoint* kondisi terdapat deteksi black hole 56](#_Toc13078772)

[Gambar 5.25 Tampilan NetAnim *random* *walk* kondisi normal 57](#_Toc13078773)

[Gambar 5.26 Hasil NS *random* *walk* kondisi normal 57](#_Toc13078774)

[Gambar 5.27 Tampilan NetAnim *random* *walk* kondisi terdapat black hole 58](#_Toc13078775)

[Gambar 5.28 Hasil NS3 *random* *walk* kondisi terdapat black hole 59](#_Toc13078776)

[Gambar 5.29 NetAnim *random* *walk* kondisi terdapat deteksi black hole 59](#_Toc13078777)

[Gambar 5.30 Hasil NS3 *random* *walk* kondisi terdapat deteksi black hole 60](#_Toc13078778)

[Gambar 5.31 Tampilan NetAnim *random* *direction* kondisi normal 61](#_Toc13078779)

[Gambar 5.32 Hasil NS3 *random* *direction* kondisi normal 61](#_Toc13078780)

[Gambar 5.33 Tampilan NetAnim *random* *direction* kondisi terdapat black hole 62](#_Toc13078781)

[Gambar 5.34 Hasil NS3 *random* *direction* kondisi terdapat black hole 63](#_Toc13078782)

[Gambar 5.35 Tampilan NetAnim *random* *direction* kondisi terdapat deteksi black hole 63](#_Toc13078783)

[Gambar 5.36 Hasil NS3 *random* *direction* dengan detksi serangan *black hole* 64](#_Toc13078784)

[Gambar 5.37 Tampilan NetAnim 30 *node* kondisi normal 65](#_Toc13078785)

[Gambar 5.38 Hasil NS3 30 *node* kondisi normal 66](#_Toc13078786)

[Gambar 5.39 Tampilan NetAnim 30 *node* kondisi terdapat black hole 66](#_Toc13078787)

[Gambar 5.40 Hasil NS3 30 *node* kondisi terdapat black hole 67](#_Toc13078788)

[Gambar 5.41 Tampilan NetAnim 30 *node* kondisi terdapat deteksi black hole 68](#_Toc13078789)

[Gambar 5.42 Hasil NS3 30 *node* kondisi terdapat deteksi black hole 68](#_Toc13078790)

[Gambar 5.43 Tampilan NetAnim 40 *node* kondisi normal 69](#_Toc13078791)

[Gambar 5.44 Hasil NS3 40 *node* kondisi normal 70](#_Toc13078792)

[Gambar 5.45 Tampilan NetAnim 40 *node* kondisi terdapat black hole 70](#_Toc13078793)

[Gambar 5.46 Hasil NS3 40 *node* kondisi terdapat black hole 71](#_Toc13078794)

[Gambar 5.47 Tampilan NetAnim 40 *node* kondisi terdapat deteksi black hole 72](#_Toc13078795)

[Gambar 5.48 Hasil NS3 40 *node* kondisi terdapat deteksi black hole 72](#_Toc13078796)

[Gambar 5.49 Tampilan NetAnim 50 *node* kondisi normal 73](#_Toc13078797)

[Gambar 5.50 Hasil NS3 50 *node* kondisi normal 74](#_Toc13078798)

[Gambar 5.51 Tampilan NetAnim 50 *node* kondisi terdapat black hole 74](#_Toc13078799)

[Gambar 5.52 Hasil NS3 50 *node* kondisi terdapat black hole 75](#_Toc13078800)

[Gambar 5.53 Tampilan NetAnim 50 *node* kondisi terdapat deteksi black hole 76](#_Toc13078801)

[Gambar 5.54 Hasil NS3 50 *node* kondisi terdapat deteksi black hole 76](#_Toc13078802)

[Gambar 6.1 Grafik packet loss variasi posisi black hole 79](#_Toc13078803)

[Gambar 6.2 Grafik *packet delivery ratio* variasi posisi black hole 80](#_Toc13078804)

[Gambar 6.3 Grafik *delay* variasi posisi black hole 81](#_Toc13078805)

[Gambar 6.4 Grafik packet loss variasi tipe pergerakan 83](#_Toc13078806)

[Gambar 6.5 Grafik *packet delivery ratio* variasi tipe pergerakan 84](#_Toc13078807)

[Gambar 6.6 Grafik *delay* variasi tipe pergerakan 85](#_Toc13078808)

[Gambar 6.7 Grafik packet loss variasi jumlah *node* 87](#_Toc13078809)

[Gambar 6.8 Grafik *packet delivery ratio* variasi jumlah *node* 88](#_Toc13078810)

[Gambar 6.9 Grafik *delay* variasi jumlah *node* 89](#_Toc13078811)

# PENDAHULUAN

## Latar Belakang

Dewasa ini, jaringan nirkabel menjalani pertumbuhan dengan pesat. Jaringan nirkabel dapat digolongkan ke dalam jaringan nirkabel berbasis infrastruktur dan berbasis tidak menggunakan infrastruktur. *Mobile* *ad hoc network* (MANET) adalah contoh dari jaringan nirkabel berbasis tidak menggunakan infrasturktur. MANETdapat diartikan sebagai jaringan *ad hoc* yang terbentuk dari beberapa node *ad hoc*. *Node* pada MANETmemiliki sifat dinamis yaitu *node* dapat bebas bergabung atau meninggalkan jaringan sehingga pembentukan *topologi*  jaringan bersifat dinamis. Sifat dinamis ini membuat MANETbanyak digunakan dalam situasi darurat seperti kegiatan militer dan ketika terjadi bencana alam. Setiap node pada jaringan MANET dapat memiliki beberapa tugas (Simamora, 2014).

Setiap *node* pada MANETakan berperan aktif ketika melakukan komunikasi dan pembentukan jalur. *Node* pada MANET memiliki beberapa peran seperti, *node* biasa, melakukan *forwarding* paket, dan dapat berperan sebagai *router* (Jain and Tokekar, 2015). Pergerakan node yang bebas membuat topologi pada jaringan MANETS sering berubah. Untuk mengatasinya, jaringan MANET membutuhkan protokol routing untuk mengatur cara berkomunikasi dan melakukan pertukaran informasi antar node (Simamora, 2014). Protokol routing memiliki peran yang penting pada jaringan MANET. Protokol routing mengatur urusan koneksi dan cara berkomunikasi antar node pada jaringan MANET yang bersifat dinamis. Penggunaan protokol routing memiliki tujuan untuk menemukan jalur dan menjaga jalur antar node pada jaringan MANET.

Protokol *routing* memiliki peran penting dalam penyebaran data pada MANET. Protokol *routing* *ad hoc on demand distance vector* (AODV) merupakan salah satu protokol *routing* yang sering digunakan pada MANET. Protokol *routing* AODV merupakan protokol yang distandardisasi oleh IETF MANET *working* *group*. AODV merupakan salah satu protokol *routing* bertipe reaktif yang berarti pembentukan rute akan dilakukan ketika terdapat permintaan(Dorri and Kamel, 2015). Pada AODV terdapat dua proses utama yaitu proses *route* *discovery* dan proses *route maintenance*. Proses *route discovery* dilakukan ketika *node* tujuan akan mengirimkan data ke *node* lain dan tidak memiliki rute pada tabel *routing* dengan mengirim pesan *route request* (RREQ) kepada *node* tetangga dan akan mendapat balasan pesan *route replay* (RREP) ketika rute telah ditemukan. Sedangkan proses *route maintenance* dilakukan jika terjadi perubahan dan kerusakan rute dengan menggunakan pesan *route error* (RERR) (Dorri and Kamel, 2015).

MANETbersifat dinamis sehingga setiap *node* dapat memperoleh informasi dengan mudah. Setiap *node* juga dapat berperilaku tidak normal seperti membuang paket yang dapat mengganggu proses *routing*. Pada MANETterdapat beberapa serangan yang mungkin dapat terjadi. Salah satu serangan pada MANETadalah *denial of Service* (DOS). DOS merupakan salah satu serangan yang *bertujan* agar suatu jaringan tidak dapat diakses oleh pengguna (Jain and Tokekar, 2015). Contoh serangan DOS pada MANETadalah *black hole*. *black hole* bekerja dengan menyatakan dirinya memiliki rute terpendek menuju *node* tujuan kepada *node* sumber *node* sumber akan menggunakan rute melewati *black hole*. *Black hole* akan membuang semua data yang melewati dirinya. Sehingga sebelum sampai tujuan, paket dihilangkan dan tidak samapi ke node tujuan. Serangan *black hole* memiliki dua jenis yaitu serangan dengan jumlah black hole satu dan serangan yang dilakukan oleh lebih dari satu *black hole* (van Glabbeek et al., 2016).

Pada umumnya MANETtidak memiliki mekanisme keamanan. MANETdirancang dengan asumsi semua *node* dapat dipercaya. Jaringan MANET rentan terhadap berbagai macam serangan yang akan menggangu komunikasi. Hal tersebet dapat berbahaya apabila paket yang dikomunikasikan merupakan paket dengan informasi yang penting. Oleh karena itu jaringan MANETmembutuhkan mekanisme keamanan untuk dapat pendeteksian dan melindungi jaringan dari beberapa serangan (Jain and Tokekar, 2015).

Penelitian terdahulu dengan judul “*Mitigating the Effects of Black hole Attacks on AODV Routing Protocol in Mobile Ad Hoc Networks*” menjelaskan *black hole* akan mengirim pesan RREP segera setelah menerima pesan RREQ dari *node* sumber tanpa meneruskan pesan RREQ ke *node* selanjutnya. Pesan RREP dari *black hole* akan sampai ke *node* sumber paling cepat dibandingkan pesan RREP dari *node* lain. Untuk mengatasinya, pada penelitian ini memberikan solusi dengan mengabaikan pesan RREP yang pertama mencapai *node* sumber (Jain and Tokekar, 2015). Penelitian lainnya dengan judul “*Implementing and improving the performance of AODV by receive reply method and securing it from Black hole attack*” menjelaskan pada AODV secara *default* akan memilih RREP yang pertama datang untuk memilih jalur. Pada penelitian ini semua pesan RREP akan disimpan terlebih dahulu dengan jangka waktu tertentu setelah menerima RREP pertama. Identifikasi *black hole* dibedakan berdasarkan nilai *delay* dan *sequence number* terbesar dari informasi *routing* yang dikumpulkan (Choudhury, Ragha and Marathe, 2015). Penelitian lainnya dengan judul “Trust Value based Algorithm to Identify and Defense GrayHole and Black-Hole attack present in MANET using Clustering Method” memberikan solusi alternatif untuk mendeteksi black hole dengan bantuan nilai kepercayaan. Solusi ini dapat menghilangkan kebutuhan IDS inbuilt di jaringan nirkabel dan menghasilkan peningkatan kinerja jaringan (Janak and Patel, 2018). Pada ketiga penelitian dilakukan pengukuran terhadap parameter pengujian *Quality of Service* (QoS*)* yaitu *packet loss, packet delivery ratio,* dan *delay*.

Berdasarkan permasalahan di atas, maka penulis membuat penelitian yang berjudul “Pendeteksian Serangan *Black hole* Terhadap Protokol *Routing Ad Hoc On Demand Distance Vector* (AODV) Pada *Mobile Ad Hoc Network* (MANET)”. Dalam penelitian ini, pendeteksian *black hole* pada MANETdilakukan dengan mengirimkan pesan RREQ palsu. Alamat tujuan pada pesan RREQ palsu diubah menjadi alamat yang tidak ada pada jaringan. Ketika *black hole* menerima pesan RREQ palsu maka akan segera membalas dengan pesan RREP dengan nilai *sequence number* tinggi dan *hop count* terendah. *Node* pendeteksi dapat mengetahui apabila terdapat balasan dari pesan RREQ palsu maka *node* tersebut merupkan *node* *black hole*.

## Rumusan Masalah

Rumusan masalah yang didapatkan berdasarkan penjelasan latar belakang diatas adalah sebagai berikut :

1. Bagaimanakah cara implementasi serangan *black hole* pada simulasi jaringan MANET menggunakan protokol AODV ?
2. Bagaimanakah cara implementasi mekanisme untuk pendeteksian *black hole* pada simulasi jaringan MANET menggunakan protokol AODV ?
3. Bagaimanakah dampak implementasi mekanisme untuk pendeteksian *black hole* terhadap kinerja protokol yang dilihat dari *packet loss, packet delivery ratio*, dan *delay* pada simulasi jaringan MANET menggunakan protokol AODV ?

## Tujuan

Berdasarkan uraian latar belakang, maka tujuan penelitian ini yaitu sebagai berikut :

1. Mengetahui cara implementasi serangan *black hole* pada simulasi jaringan MANET menggunakan protokol AODV
2. Mengetahui cara implementasi mekanisme untuk pendeteksian *black hole* pada simulasi jaringan MANET menggunakan protokol AODV
3. Mengetahui dampak implementasi mekanisme untuk pendeteksian *black hole* terhadap kinerja protokol yang dilihat dari *packet loss, packet delivery ratio*, dan *delay* pada simulasi jaringan MANET menggunakan protokol AODV

## Manfaat

Hasil penelitian diharapkan mampu memberikan beberapa manfaat untuk pembaca dan peneliti. Manfaat yang diharapkan yaitu sebagai berikut :

1. Hasil penelitian dapat digunakan sebagai sarana pembelajran untuk mengimplementasikan teori yang telah dipelajari
2. Hasil Penelitian diharapkan dapat menjadi referensi untuk melakukan penelitian lain dalam upaya meningkatkan keamanan pada protokol AODV di jaringan MANET
3. Penelitian ini dapat dijadikan pertimbangan dalam memilih protokol AODV di jaringan MANETberbasis hasil pengujian nilai *packet loss, packet delivery ratio,* dan *delay*

## Batasan Masalah

Berdasarkan uraian rumusan masalah diatas, agar penelitian dapat terfokus dan tidak mengakibatkan melebarnya masalah, maka berikut Batasan masalah pada penelitan ini :

1. Protokol *routing* yang digunakan adalah AODV
2. Serangan yang digunakan adalah *black hole*
3. Pendeteksian menggunakan pesan RREQ palsu dengan alamat tujuan “10.1.1.99”
4. Tipe pergerakan yang akan digunakan adalah *pergerakan random waypoint, pergerakan random walk, dan pergerakan random direction*
5. Jumlah *node* yang akan digunakan adalah 30, 40, dam 50
6. Ukuran paket data yang dikirim berukuran 512 *bytes*
7. Ukuran luas area dalam penelitian ini adalah 1000mx1000m
8. Waktu setiap simulasi dalam penelitian ini adalah 1000 detik
9. Jumlah *black hole* yang digunakakan adalah satu
10. Pengukuran nilai parameter dalam penelitian ini adalah *packet loss*, *delay, dan packet delivery ratio*

## Sistematika Pembahasan

Sistematika atau Gambaran umum mengenai penelitian ini disusun sebagai berikut:

1. BAB I : Pendahuluan

Pada bab ini berisi latar belakang penelitian, rumusan masalah penelitian, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

1. BAB II : Landasan Kepustakaan

Pada bab ini penulis memaparkan ladasan kepustakaan dari penelitian terdahulu dan landasan teori. Pembahasan landasan teori seperti *mobile ad-hoc network* (MANET), protokol *routing* pada MANET, model pergerakan pada MANET, serangan pada MANET, jaringan nirkabel, *network* simulator, animator, dan parameter pengujian *quality of service*.

1. BAB III : Metodologi Penelitian

Pada bab ini penulis menjelaskan urutan tahapan yang dilakukan untuk dilakukan pada penelitian ini. Bab ini terdiri dari studi literatur, analisis kebutuhan, perancangan , implementasi, pembahasan hasil pengujian, dan kesimpulan dan saran.

1. BAB IV : Analisa Kebutuhan Dan Perancangan

Pada bab ini penulis menjelaskan kebutuhan dari sistem berdasarkan penjelasan deskripsi umum sistem dan tujuan penelitian. Bab ini terdiri dari deskripsi umum sistem, kebutuhan sistem, perancangan serangan *black hole*, perancangan deteksi serangan *black hole*, perancangan parameter pengujian, perancangan sistem, dan perancangan skenario pengujian.

1. BAB V : Implementasi

Pada bab ini penulis melakukan implementsasi berdasarkan perancangan yang telah dibuat. Bab ini terdiri dari instalasi NS 3, implementasi serangan *black hole*, implementasi deteksi serangan *black hole*, implementasi sistem, implementasi skenario pengujian.

1. BAB VI : Hasil Pengujian Dan Pembahasan

Pada bab ini penulis memaparkan mengenai pembahasan dari pengujian yang telah diakukan pada penelitian ini. Bab ini terdiri dari pembahasan pengujian kinerja protokol terhadap variasi posisi *black hole*, tipe pergerakan, dan jumlah *node*.

1. BAB VII : Penutup

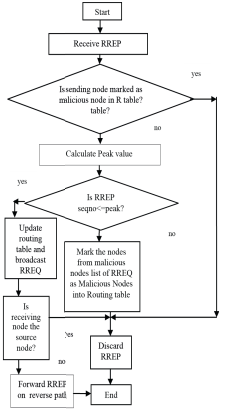
Pada bab ini penulis memaparkan kesimpulan yang didapatkan dari pengamatan hasil pengujian dan analisis. Penulis memberikan saran berdasarkan hasil pengujian dan analisis untuk menyempurnakan penelitian selanjutnya.

# LANDASAN KEPUSTAKAAN

## Penelitian Terdahulu

Sebelumnya jain dan tokekar telah melakukan penelitian dan berhasil mengatsi permasalahan serangan *black hole* pada protokol *routing* AODV di MANETyakni dengan mengabaikan pesan RREP pertama yang mencapai *node* sumber. Penelitan ini menjelaskan *black hole* akan mengirim pesan RREP segera setelah menerima RREQ dari *node* sumber tanpa meneruskan pesan RREQ ke *node* selanjutnya. Pesan RREP dari *black hole* akan sampai ke *node* sumber paling cepat dibandingkan pesan RREP dari *node* lain. *Node* sumber akan memilih rute melewati *black hole* untuk mengirimkan paket ke *node* tujuan. Ketika paket melewati *node* *malicious*, maka paket akan di buang oleh *black hole*. Jain dan tokekar melakukan modifikasi protokol *routing* AODV dengan mengabaikan pesan RREP yang pertama mencapai *node* sumber. Hal itu dilakukan karena ada kemungkinan keterlibatan *black hole* di pesan RREP pertama yang diterima *node* sumber. Dengan mengabaikan pesan RREP yang pertama maka rute yang melalui *black hole* tidak terbentuk. Jain dan tokekar melakukan pengukuran performa dengan parameter *packet delivery ratio*, *end to end delay*, *throughput*, dan *routing* overhead. Pengukuran performa menunjukan bahwa metode yang digunakan bekerja dengan baik ketika terdapat serangan *black hole* dan tidak adanya serangan (Jain and Tokekar, 2015).

Penelitian lainnya yang dilakukan choundury telah melakukan penelitian dan berhasil mengatasi permasalahan serangan *black hole* pada protokol *routing* AODV di MANETyakni dengan menampung pesan RREP dalam jangka waktu tertentu dan melakukan identifikasi *black hole* berdasarkan nilai *sequence number* dan *delay*. Penelitian ini menjelaskan beberapa masalah pada jaringan ad hoc seperti, *node* yang bersifat dinamis, sistem desentralisasi, dan *black hole* yang dapat menyebabkan penurunan kinerja yang signifikan. Pada AODV pemilihan pesan RREP yang digunakan untuk pengiriman data berdasarkan pesan yang sampai terlebih dahulu, nilai *sequence number* tertinggi, dan hop terp*end*ek. Metode yang digunakan pada penelitian ini akan menampung pesan RREP dengan jangka waktu tertentu setelah menerima RREP pertama. Selanjutnya dilakukan Identifikasi *black hole* menggunakan informasi yang terdapat pada pesan RREP. Proses penerimaan pesan RREP dan identifikasi pada *node* sumber dapat dilihat pada Gambar 2.1. Informasi yang digunakan adalah nilai *sequence number* dan *delay* . Pada penelitian ini dilakukan pengukuran performa dengan parameter *packet delivery ratio* dan *throughput*. Metode ini dapat menangkal serangan *black hole* pada protokol AODV (Choudhury, Ragha and Marathe, 2015). Pada penelitian lain yang dilakukan oleh janak dan patel memberikan algoritma yang diusulkan didasarkan pada nilai kepercayaan masing-masing *node*. Membagi semua *node* menjadi tiga *cluster*, di setiap *cluster* dipilih *cluster* utama yang memiliki tingkat energi maksimum. Pada metode ini cara membedakan *node* *black* *hole* dengan menggunakan tingkat energi dari *node* dan nilai kepercayaan dari setiap *node* di MANET. Setelah menemukan nilai kepercayaan kami membandingkannya dengan nilai ambang batas yang telah ditentukan dan mendeteksi *black* *hole*, setelah mendeteksi dilakukan pencegahan dengan paket yang konsisten diteruskan melalui beberapa jalur alternatif dalam jaringan. Terdapat 3 tahap dalam pendeteksian yaitu proses inisialisasi yang digunakan untuk menentukan nilai awal dari nilai energi dan kepercayaan pada setiap *node*. Selanjutnya proses meningkatkan nilai kepercayaan pada setiap node ketika paket-paket ditransmisikan dengan benar dari satu *node* ke *node* lain maka akan nilai kepercayaan akan ditingkatkan. Terakhir adalah proses Menghilangkan *black hole* dari jaringan dengan cara menandai *node* yang memiliki nilai kepercayaan lebih kecil dari nilai ambang batas sebagai *black hole* (Janak and Patel, 2018).



Gambar 2.1 *Flow*chart ketika *node* menerima pesan RREP

Pada penelitian ini menggunakan referensi penelitian sebelumnya yang memiliki kesamaan dengan topih pembahasan penelitian ini. Berikut merupakan penelitian sebelumnya yang dijadikan sebagai referensi penulis, untuk memudahkan dalam memahami dan perbandingan kajian pustaka disajikan pada Tabel 2.1.

Tabel 2.1 Kajian pustaka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Penulis, Tahun, Judul** | **Persamaan** | **Perbedaan** | |
| **Penelitian Terdahulu** | **Rencana Penelitian** |
| 1 | (Jain and Tokekar, 2015), *Mitigating the Effects of Black hole Attacks on AODV Routing Protocol in Mobile Ad Hoc Networks* | Mengabaikan pesan RREP | Pada penelitian ini memberikan solusi dengan mengabaikan pesan RREP yang pertama mencapai *node* sumber | Mengabaikan pesan RREP yang membalas pesan RREQ dengan alamat tujuan palsu |
| 2 | (Choudhury, Ragha and Marathe, 2015) , *Implementing and improving the performance of AODV by receive reply method and securing it from Black hole attack* | Pendeteksian dilakukan dengan menunggu pesan RREP dari node black hole | Pada penelitian memberikan solusi dengan menyimpan pesan RREP terlebih dahulu dengan jangka waktu tertentu setelah menerima pesan RREP pertama. Identifikasi *black hole* berdasarkan *delay* dan *sequence number*. | Pendeteksian serangan *black hole* dilakukan dengan menunggu node yang mengirimkan pesan RREP balasan dari pesan RREQ dengan alamat tujuan palsu |
| 3 | (Janak and Patel, 2018) , *Trust Value based Algorithm to Identify and Defense GrayHole and Black-Hole attack present in MANET using Clustering Method* | Melakukan *blacklist* terhadap *black hole* yang telah terdeteksi | Proses *blacklist* dengan cara menandai *node* yang memiliki nilai kepercayaan lebih kecil dari nilai ambang batas sebagai *black hole*. | Proses *blacklist* dilakukan oleh *node* terdekat dari *black* *hole* ketika terdapat balasan dari pesan RREQ palsu. |

## Mobile Ad Hoc *Network* (MANET)

### Pengertian MANET

*Mobile ad-hoc network* (MANET) adalah jarignan nirkabel yang bersifat independen. MANETterbentuk dari sekumpulan *node* *ad-hoc* yang dapat berperan sebagai *router* dan *host*. Konfigurasi jarignan pada MANETdilakukan oleh *node* secara mandiri tanpa menggunakan infrastruktur (Panda and Pattanayak, 2018). *Node* pada MANETbersifat dinamis sehingga pembentukan *topologi*  jaringan bersifat dinamis. *Node* pada MANET memiliki beberapa peran seperti, *node* biasa, melakukan forwarding paket, dan *node* dapat berperan sebagai *router* (Jain and Tokekar, 2015).

*Mobile ad hoc network* (MANET) bekerja independen karena dalam pengoprasian tidak memerlukan infrastruktur tetap seperti *router*, *swtich*, dan *base* *station*. Jaringan MANETterbentuk dengan cara dinamis dari beberapa *node* melalui jaringan nirkabel yang tidak menggunakan infrastrurktur tetap dan administrasi terpusat. Setiap *node* pada jaringan MANETdapat bergerak dengan bebas yang membuat *topologi* pada jaringan MANETdapat berubah dengan cepat setiap saat. Secara umum, rute antar *node* dalam jaringan MANETdapat mencakup *multihop* (Basagni et al., 2004). Contoh *multihop* ditunjukan pada Gambar berikut :



Gambar 2.2 Contoh jaringan MANET

Seperti yang ditunjukan pada Gambar 2.2, pada jaringan MANETdapat terbentuk dari sekumpulan perangkat yang berbeda seperti, laptop, komputer, dan sebagainya. Semua *node* pada jaringan MANETdapat melakukan komunikasi secara langsung dengan *node* lain yang ada pada jangkauan transmisinya. Untuk melakukan komunikasi dengan *node* yang berada diluar jangkauan transmisinya, *node* sumber akan menggunakan *node* tetangga untuk menyampaikan pesan (Basagni et al., 2004).

### Karakteristik MANET

Jarinan MANET terbentuk dari node yang dapat bergerak bebas dan tanpa membutuhkan infrasturktur. Dalam melakukan komunikasi, digunakan media transmisi nirkabel untuk mengirimkan informasi antar node. Berikut merupakan karakteristik pada jaringan MANET(Basagni et al., 2004):

1. Nirkabel. Setiap node berkomunikasi menggunakan media transmisi nirkabel dan menggunakan media yang sama (radia, inframerah, dll).
2. Berbasis ad-hoc. Terbentuk dari sekumpulan ad hoc yang merupakan jaringan sementara. Ad hoc dibentuk secara dinamis dan menyesuaikan sesuai kebutuhan.
3. Mandiri dan tidak memiliki infrastruktur. MANET tidak tergantung pada infrastruktur dan administrasi terpusat. Setiap node dapat melakukan komunikasi dengan *peer to peer*, dan dapat bertindak sebagai router maupun end device.
4. Multihop routing. MANET tidak membutuhkan router karena setiap node bertindak sebagai router dan dapat meneruskan paket untuk berbagi informasi antar node.
5. Bergerak bebas. Setiap node pada MANET dapat bergerak dengan bebas sambal berkomunikasi dengan node lain. Topologi pada MANET bersifat dinamis karena pergerakan setiap node menyebabkan pola komunikasi antar node terus berubah.

### Penggunaan MANET

Jaringan MANETmerupakan jaringan fleksibel yang dapat diatur dimana saja, kapan saja, tanpa infrastruktur, dan tanpa administrasi terpusat. Jaringan MANETdapat digunakan pada bidang militer, keadaan bencana alam, dan sebagainya. Pada awalnya jaringan MANETlebih sering digunakan pada kegiatan militer, saat ini jaringan MANETtelah berkembang dan telah menyebar diluar militer. Sebagai hasilnya, telah terdapat beberapa aplikasi jaringan MANETseperti penggunaan pribadi, jaringan rumah, operasi penyelamatan, P*end*idikan, komersial, dan sebagainya. Berikut merupakan penggunaan MANET:

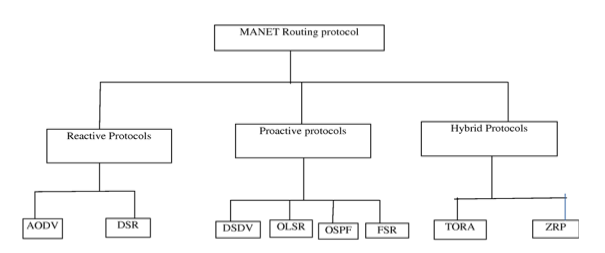
1. Militer, komunikasi militer.
2. Jaringan rumah, rumah atau perkantoran WLAN.
3. Operasi penyelamatan, operasi pencarian dan penyelamatan dalam pemulihan bencana.
4. Pendidikan, komunikasi *ad hoc* selama konfersensi, rapat, dan kuliah.
5. Komersial, pembayaran elektronik, penggunaan jaringan *ad* *hoc* *local* antara kendaraan.

Penggunaan jaringan MANETmemiliki banyak manfaat, kemudahan dalam konfigurasi, dan sebagainya. Namun dikarenakan sifatnya yang fleksibel membuat jaringan MANETmemiliki masalah seperti optimasi *bandwith*, pengkontrolan daya, kualitas transmisi, Tanpa infrastruktur, *Topologi*  yang berubah, Keandalan jaringan, Keamanan jaringan, *Quality of Service* , dan sebagainya (Basagni et al., 2004).

## Protokol *Routing* pada MANET

### Pengertian Protokol *Routing*

Protokol *routing* pada MANETmemiliki tiga cara dalam melakukan *route discovery* dan *route maintenance*, yaitu dengan selalu memperbarui informasi ( *proactive)* , memperbarui informasi ketika terdapat permintaan (*reactive)*, dan gabungan *proactive* dan *reactive* (*hybrid)* dapat dilihat pada Gambar 2.3 (Panda and Pattanayak, 2018).



Gambar 2.3 Jenis protokol *routing* MANET

Pada Gambar 2.3 terdapat 3 jenis protokol routing yaitu protokol proactive, protokol reactive, dan protokol hybrid. Protokol reactive adalah protokol yang bekerja ketika terdapat permintaan pencarian rute untuk pengiriman data atau berkomunikasi dengan node lain. Ketika node melakukan komunikasi dengan node lain, proses pencarian rute mulai dilakukan. Protokol proactive adalah protokkol yang selalu memperbarui informasi tabel routing pada setiap node. Setiap node akan selaui memperbarui informasi tabel routing miliknya meskipun tidak ada komunikasi yang terjadi antar node. Protokkol ini disebut juga table-driven dalam memperbarui informasi tabel routing. Protokol hybrid merupakan gabungan antara protokol reactive dan protokol proactive. Protokol reactive digunakan ketika topologi jaringan relatif tidak berubah. Protokol proactive digunakan ketika topologi jaringan bersifat dinamis dan sering berubah (Panda and Pattanayak, 2018).

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

#### Pengertian AODV

Protokol *routing* memiliki peran penting dalam mengatur koneksi dan komunikasi pada jaringan. Tujuan utama dari protokol *routing* yaitu untuk mencari dan menjaga jalur antar *node*. Contoh dari protokol *routing* yang bersifat *reactive* adalah *Ad hoc on demand distance vector* (AODV) (Kaneria and Rajavat, 2016). AODV memiliki kelebihan dalam penggunaan memori yang lebih hemat, proses yang lebih singkat dan merupakan *routing* protokol yang bersifat dinamis.

Protokol AODV merupakan contoh protokol *reactive* disebut juga dengan *on-demand routing* yang menggunakan p*end*ekatan berbeda dari protokol *proactive*. Kelemahan protokol *proactive* pada *overhead* yang berasal dari proses pemeliharaan rute pada setiap *node* yang dilakukan setiap saat. Protokol *reactive* dilakukan dengan tidak melakukan pemeliharaan rute setiap saat, rute dibentuk ketika dibutuhkan saja. Ketika *node* sumber perlu rute untuk melakukan pengiriman data ke *node* tujuan, *node* sumber akan memeriksa Tabel *routing* miliknya. Jika rute tidak ditemukan maka akan dilakukan proses *route discovery* untuk menemukan rute ke *node* tujuan. Protokol *reactive* memiliki kelebihan berkurangnya *overhed* dibandingkan dengan protokol *proactive* khususnya dalam jaringan dengan lalu lintas r*end*ah hingga sedang.

Protokol AODV merupakan protokol yang menyediakan *route discovery* ketika terdapat permintaan pada jaringan MANET. Pencarian rute dilakukan dengan proses *route discovery* yang akan mengirimkan pesan secara *broadcast* pada jaringan dan m*end*apatkan balasan secara *unicast* yang berisi informasi rute. AODV memiliki kesamaan dengan protokol DSDV yaitu setiap *node* menggunakan *sequence number* untuk menghindari *loop* dan pemilihan informasi rute terbaru.

#### Route Message

Protokkol routing AODV memiliki 3 jenis pesan yaitu *route* *request* (RREQ) untuk mengirmkan permintaan rute menuju *node* tujuan, *route* *replay* (RREP) untuk mengirimkan balasan dari pesan RREQ yang berisi informasi rute menuju *node* tujuan, dan *route* *error*(RERR) yang ditampilkan pada Tabel 2.2, Tabel 2.3, dan Tabel 2.4 (Kaneria and Rajavat, 2016).

Tabel 2.2 Pesan RREQ

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Flags** | **Reserved** | **Hop *count*** |
| RREQ (*Broadcast*) id | | | |
| Destination IP address | | | |
| Destination *sequence number* | | | |
| *Source* IP address | | | |

Tabel 2.3 Pesan RREP

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **A** | **Reserved** | **Hop *count*** |
| Destination IP address | | | |
| Destination *sequence number* | | | |
| *Source* IP address | | | |
| *Source* *sequence number* | | | |

Tabel 2.4 Pesan RRER

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **N** | **Reserved** | **Destination *count*** |
| Unreachable destination IP address | | | |
| Unreachable Destination *sequence number* | | | |
| Additional unreachable destination IP address | | | |
| Additional Unreachable Destination *sequence number* | | | |

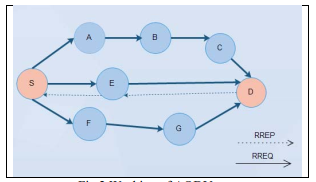
Setiap *node* akan membuat tabel *routing* dan akan dilakukan *update* konten pada saat menerima pesan RREQ, RREP, dan RERR. Tabel *Routing* memiliki struktur seperti pada Tabel 2.5.

Tabel 2.5 Tabel *routing* pada setiap *node*

|  |
| --- |
| Destination IP address |
| Destination *sequence number* |
| Hop *count* |
| Next hop |
| First hope |
| Valid *Bit* |
| *Count* |

Pada AODV memiliki dua proses utama yaitu proses *route* *discovery* dan proses *route* *maintenance*. Proses *route* *discovery* dilakukan saat suatu *node* akan mengirimkan paket data ke *node* lain dan tidak mempunyai rute di dalam tabel *routing* dengan mengirim pesan *route* *request* (RREQ) kepada *node* tetangga dan akan m*end*apat balasan pesan *route* *replay* (RREP) ketika rute telah ditemukan. Sedangkan proses *route* *maintenance* dilakukan jika terjadi perubahan dan kerusakan rute dengan menggunakan pesan *route* *error* (RERR) (Dorri and Kamel, 2015).

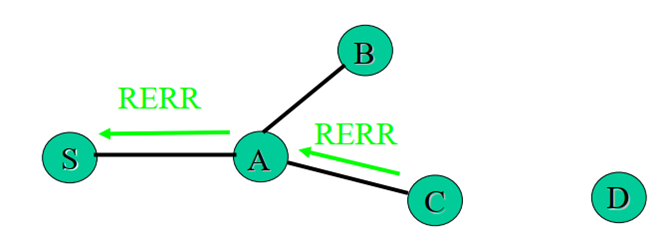
#### Route Discovery



Gambar 2.4 Proses *route* *discovery*

Ketika *source* *node* akan mengirimkan data ke tujuan, langkah pertama yang dilakukan node sumber adalah melihat tabel *routing* sendiri apakah memiliki rute menuju node tujuan . Apabila rute tidak ditemukan di tabel *routing* sendiri maka proses *broadcast* RREQ ke semua tetangga akan dilakukan oleh *node* sumber untuk mendapatkan rute menuju *node* tujuan. Balasan pesan RREP dapat dilakukan langsung oleh *node* tujuan dan *node* tetangga yang memiliki rute menuju tujuan. *Node* sumber akan mendapat balasan pesan RREP ketika rute telah ditemukan dan memilih rute dengan nilai *hop count* terkecil (Kaneria and Rajavat, 2016). Proses *route* *discovery* dapat dilihat pada Gambar 2.4.

#### Route Maintenance



Gambar 2.5 Proses *route* *maintenance*

Route maintenance adalah proses yang dilakukan jika terjadi perubahan dan kerusakan rute. Pada Gambar 2.5 Ketika koneksi antara *node* C dan D mengalami kerusakan atau perubahan rute maka *node* c yang merupakan tetangga *node* D akan menghapus rute *node* D pada tabel *routing miliknya*. *Node* C akan membuat pesan RERR dan mengirimkan pesan kepada *node* tetangga yaitu *node* A untuk diteruskan sampai *node* sumber. *Node* tetangga dan *node* sumber akan menghapus rute yang menggunakan *node* D (Panda and Pattanayak, 2018). Porses *route* *maintenance* dapat dilihat pada Gambar 2.5.

### Destination Sequenced *Distance* *Vector* (DSDV)

Protokol DSDV merupakan contoh protokol *proactive* yang memiliki karakteristik setiap *node* pada jaringan akan melakukan pemeliharaan rute ke setiap *node* lain setiap saat. Setiap *node* akan melakukan pembuatan dan pemeliharaan rute secara berkala dan ketika terdapat permintaan rute dalam komunikasi antar *node*. Protokol *proactive* memiliki kelebihan yaitu rute akan tersedia ketika dibutuhkan. Setiap *node* akan mempertahankan rute terbaru dalam jaringan secara konsisten. *Node* sumber dapat dengan mudah memeriksa Tabel *routing* untuk mengirimkan data ke beberapa tujuan.

Protokol DSDV merupakan protokol *distance* *vector* yang mengimplelmentasikan penyesuaian untuk membuat operasinya lebih cocok untuk jaringan MANET. Protokol ini menggunakan *sequence number* pada setiap *node* untuk menghindari masalah yang sering terjadi pada protokol *distance* *vector* yaitu perhitungan hingga tak terhingga. Setiap *node* akan melakukan increment pada *sequence number* ketika terjadi perubahan *topologi*  dikarnakan penambahan dan penghapusan *node*. Protokol DSDV menggunakan nilai *sequence number* terbesar untuk memilih rute untuk memastikan penggunaan informasi terbaru. Setiap *node* pada akan memelihara rute ke setiap *node* pada jaringan (Basagni et al., 2004).

### Zone *Routing* *Protocol* (ZRP)

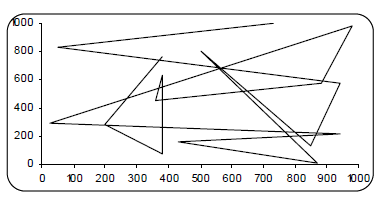
Protokol ZRP merupakan contoh Protokol *hybrid* yang mengintegrasikan protokol *proactive* dan protokol *reactive*. Protokol *hybrid* dapat berperilaku seperti protokol *proactive* dalam keadaan tertentu dan dapat berperilaku seperti protokol *reactive* dalam keadaan lain. Protokol ini dapat berperilaku dengan fleksibel menyesuaikan karakteristik jaringan.

Protokol ZRP merupakan protokol yang mengintegrasikan protokol *proactive* dan *reactive* menjadi satu protokol tunggal. Pada protokol ini akan dibagi menjadi beberapa zona yang diukur berdasarkan radius antar *node*. Setiap *node* dalam zona akan menggunakan protokol *proactive* untuk menentukan rute. Sedangkan *node* diluar zona akan menggunakan protokol *reactive* untuk menentukan rute. Ketika *node* akan mengirimkan data ke *node* tujuan maka ia akan memeriksa Tabel *routing* untuk mencari rute. Jika tujuan terletak di dalam zona maka rute berada pada Tabel. Jika tidak ditemukan rute maka tujuan berada di luar zonanya. Oleh karena itu dilakukan pencarian untuk menemukan rute ke tujuan (Basagni et al., 2004).

## Model Pergerakan pada MANET

### *Random* *Waypoint*

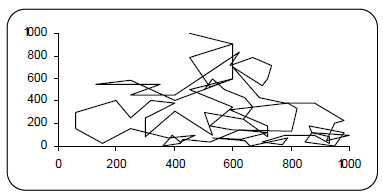
Pergerkan random waypoint adalah pergerakan yang menentukan tujuan secara acak dan memiliki pergerakan lurus dengan kecepatan minimal sampai maksimal sesuai yang ditentukan. Ketika node telah mencapai tujuan atau waypoint maka node akan berhenti (pause time) lalu akan mengulai langkah sebelumnya yaitu memilih tujuan secara acak (Technology and Science, 2013). Gambar 2.6 menunjukan arah pergerakan node menggunakan random waypoint.



Gambar 2.6 Mobilitas *node* *random* *waypoint*

### *Random* *Walk*

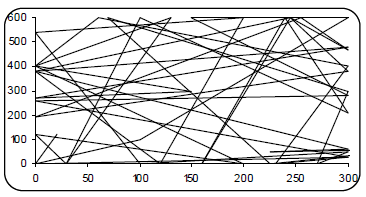
Random walk merupakan pergerakan kesamaan dengan random waypoint yang sifatnya memilih tujuan secara acak. Perbedaan dengan random waypoint adalah tidak terdapat pause time pada model random walk. Model ini tidak memiliki memori untuk menyimpan status pergeran terdahulu. Kecepatan pada kondisi sebelumnya dan kecepatn saat ini dapat berbeda. Node bergerak secara acak. Model ini juga menggunakan kecepatan maksimal dan kecepatan minimal yang dapat ditentukan (Technology and Science, 2013). Gambar 2.7 merupakan arah pergerakan node dengan model random walk.



Gambar 2.7 Mobilitas *node* *random* *walk*

### *Random* *Direction*

Pergerakan random direction menentukan arah tujuan dengan cara acak dan memiliki pergerakan ke pinggir area yang telah ditentukan. Model ini memiliki kesaam dengan model random waypoint yaitu memilih tujuan secara acak. Ketika node telah mencapai tujuan atau batas area maka node akan berhenti (pause time) dan memilih arah baru (Technology and Science, 2013). Gambar 2.8 berikut contoh pergerakan node yang menggunakan model random direction.

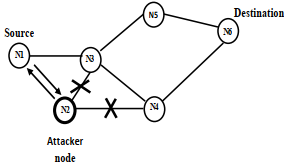


Gambar 2.8 Mobilitas *node* *random* *direction*

## Serangan pada MANET

### Serangan *Black hole*

Serangan black hole adalah serangan yang membuang paket (drop paket) yang diterimanya. Seranan ini bekerja dengan memberikan balasan setiap permintaan jalur (pesan RREQ) yang diterimanya. Node black hole akan menyatakan dirinya memiliki rute dan segara membalas dengan pesan RREP. Informasi pada pesan RREP black hole menyatakan memiliki rute terpendek menuju node tujuan dengan cara mengubah nilai hop count menjadi minimum dan nilai sequence number menjadi lebih besar. Serangan black hole dapat bekerja dengan jumlah tunggal dan dengan banyak node atau disebut dengan serangan collaborative black hole (van Glabbeek et al., 2016). Serangan *black hole* dapat dilihat pada Gambar 2.9. Sebagai contoh pada Gambar 2.9 N1 akan mengirimkan pesan paket ke N6 dan *node* sumber memulai proses *route* *discovery*. Diasumsikan *node* N2 sebagai *node* penyerang yang tidak memiliki rute ke N6 pada Tabel *routing* miliknya. *Node* 2 menyatakan dirinya memiliki rute langsung ke *node* tujuan setalah menerima pesan RREQ dari *node* sumber dan membalas dengan mengirimkan pesan RREP ke *node* sumber. Ketika *node* sumber mengirim pesan melalui N2 maka *node* penyerang dengan mudah dapat melakukan drop data (Abdelhaq et al., 2011).



Gambar 2.9 Proses serangan *black hole*

*Node* penyerang melakukan beberapa tahapan dalam serangan *black hole* seperti berikut :

1. Mengirimkan pesan RREP dengan nilai *sequence number* yang tinggi
2. Mengirimkan pesan RREP dengan nilai hop yang r*end*ah

### Serangan Grey Hole

Serangan Grey hole memiliki kemiripan dengan *black hole*. Pada serangan *black hole*, *black hole* akan membuang semua paket. Sementara pada serangan *grey* hole, *black hole* akan membuang paket dengan probabilitas yang berbeda. *Black hole* akan membuang paket secara selektif (Jain and Tokekar, 2015).

## Jaringan Nirkabel

### Pengertian Jaringan Nirkabel

Jaringan nirkabel (*wireless*) adalah teknologi komunikasi untuk menghubungkan dua atau lebih alat komunikasi tanpa menggunakan kabel sebagai media transmisi. Jaringan nirkabel menggunakan gelombang frekuensi tertentu untuk dapat saling berkomunikasi. Jaringan nirkabel memiliki beberapa macam teknologi yang penggunaannya disesuaikan dengan kebutuhan. (Arief, 2007)

### Standar Jaringan Nirkabel

Terdapat beberapa standar yang dibuat oleh *Institute of Electrical and Electronic Engineers* (IEEE), diantaranya adalah:

1. IEEE 802.11: Standar yang ditetapkan oleh IEEE pada tahun 1997 yang bekerja pada frekuensi 2,4 Ghz hanya mampu mendukung *bandwith* jaringan sebesar 2 Mbps saja.
2. IEEE 802.11b: Pada tahun 1999 IEEE membuat standar baru yang mampu mendukung *bandwith* sebesar 11 Mbps, dan masih bekerja pada frekuensi 2,4 Ghz.
3. IEEE 802.11a: Saat IEEE sedang mengembangkan standar 802.11b, secara bersamaan mereka juga mengambangkan standar 802.11a. standar ini sudah mampu mendukung *bandwith* sebesar 54 Mbps tetapi bekerja pada frekuensi 5 Ghz yang artinya standar ini tidak kompatibel dengan 802.11b karena bekerja pada frekuensi yang berbeda(Asrullah, 2015).
4. IEEE 802.11g: Standar ini ditetapkan pada tahun 2002. Standar ini menyatukan kemampuan pada standar 802.11a dan 802.11b. Standar ini mampu mendukung *bandwith* sebesar 54 Mbps dan bekerja pada frekuensi 2,4 Ghz (Arief, 2007).

## *Network* Simulator 3 (NS3)

### Pengertian NS3

*Network* simulator 3(NS3) adalah sebuah aplikasi *network* simulator, ditargetkan terutama untuk keperluan penelitian dan edukasi. NS3 adalah perangkat lunak gratis di bawah lisensi GNU GPLv2. Dibagun sebagai sisetm dari *software* *libraries* yang bekerja bersama. Ditulis dalam Bahasa c++ dan *python*. Sebelum menjalankan ns-3 perlu dilakukan proses *building* *libraries* terlebih dahulu. Ns 3 memberikan pemodelan tentang bagaimana paket data jaringan dapat bekerja dan berjalan dan menyediakan simulasi bagi pengguna untuk melakukan eksperimen simulasi (ns-3 Tutorial, 2019).

### Konsep dan Abstraksi pada NS3

Berikut merupakan konsep dan abstraksi pada ns-3 :

1. *Node*

Setiap perangkat yang terhubung pada jaringan disebut *host* atau *end* *system*. Pada ns-3 tidak menggunakan istilah *host* yang terikat dengan inter*net* dan protokol. Abstraksi perangkat komputer pada ns-3 menggunakan istilah umum *node*. *Node* menyediakan metode untuk mengelola representasi perangkat komputer dalam simulasi. *Node* pada ns-3 perlu ditambahkan hal-hal seperti aplikasi, protokol yang digunakan, perangkat jaringan untuk dapat digunakan

1. *Channel*

*Channel* merupakan media tempat data mengalir dalam jaringan. *Channel* digunakan untuk menghubungkan *node* dengan *node* lain untuk berkomunikasi. Pada ns-3, *channel* di abstraksi dengan c++ oleh *classChannel*. *classChannel* menyediakan metode untuk mengelola dan menghubungkan *node*

1. *Net device*

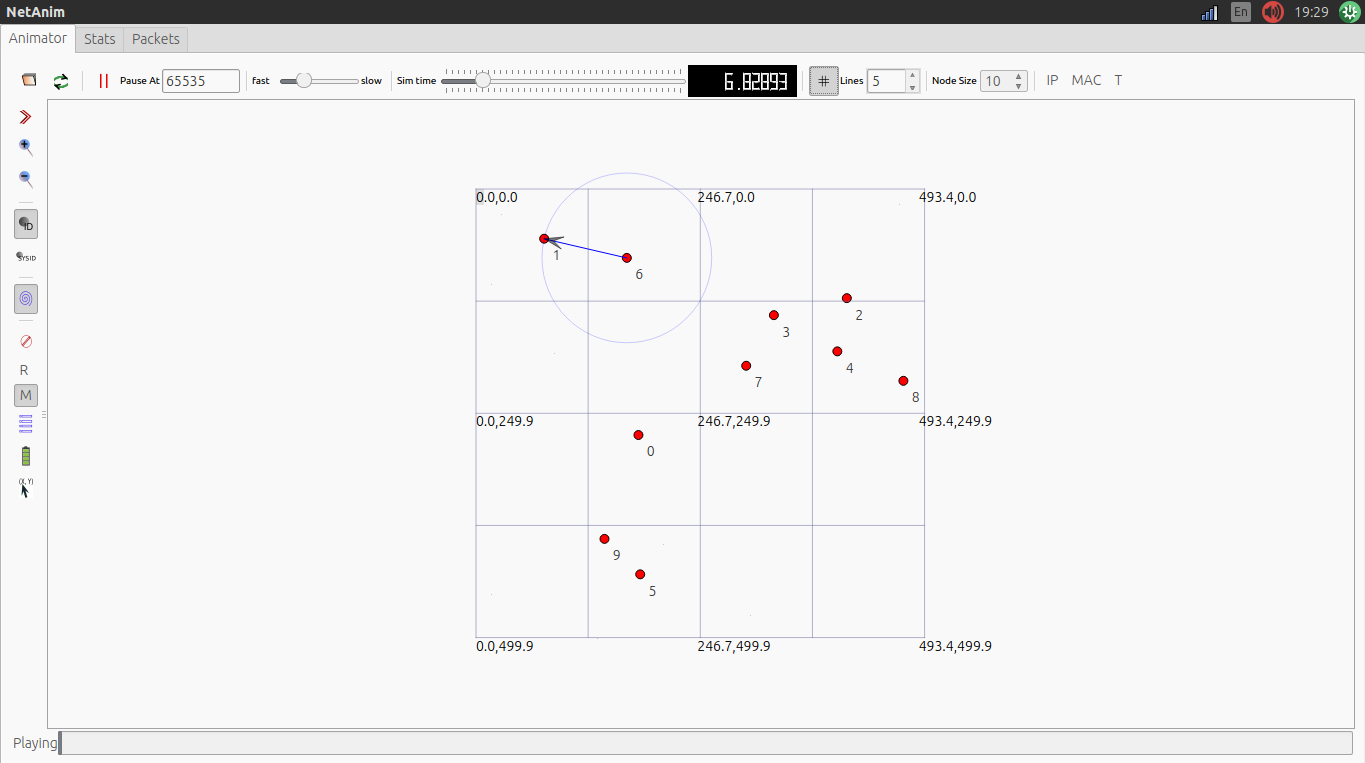
Ketika komputer akan dihubungkan ke jaringan, maka komputer memerlukan jenis kabel tertentu dan perangkat kartu NIC yang perlu dipasang. Perangkat NIC tidak berfungsi tanpa driver untuk mengkontrol perangakat keras. Pada ns-3, *net device* berfungsi sebagai perangkat keras dan perangkat lunak atau driver. *Net device* diabstraksikan dalam c++ oleh classNet*Device*

1. *Topology helpers*

Topology helpers digunakan untuk memudahkan dalam menggabungkan *node*, *Net Device*, dan *channel*. Pada *topology helpers* dapat dilakukan pengaturan IP, instalasi *Net Device* pada *node*, konfigurasi protokol yang akan digunakan dan menggabungkan *net device* dengan *channel*

## *Network* Animator (NetAnim)

Netanim adalah program yang menampilkan simulasi jaringan dengan grafik animasi berbasi Qt toolkit. NetAnim menjalankan file .XML yang dapat dihasilkan dari simulasi aplikasi NS3 (ns-3 Tutorial, 2019). Gambar 2.10 merupakan tampilan dari NetAnim yang sedang menjalankan simulasi jaringan menggunakan file XML.



Gambar 2.10 Tampilan NetAnim

## Parameter Pengujian *Quality* *of* *Service* (QoS)

### Pengertian Parameter Pengujian *Quality* *of* *Service* (QoS)

Quality of Service adalah cara mengukur kempuan dari suatu jaringan untuk diketahui seberaba baik performa jaringan. Tujuan utama QoS untuk menentukan parameter pengujian untuk mengukur kinerja jaringan dan layanan. Analisis QoS digunakan untuk melihat kemampuan jaringan dan layanan dalam pengujian tertentu (Nurusshobah, 2012).

### Packet Loss

*Packet loss* adalah kegagalan *end device* dalam mentransmisikan paket untuk mencapai tujuannya (Istiqomah, Siregar and Kartikasari, 2018). Penghitungan *packet loss* ini nantinya akan digunakan untuk membuktikan bahwa pencegahan dapat membuat rasio kegagalan pengiriman data lebih kecil daripada tanpa pencegahan. Rumus untuk *packet loss* ditunjukan pada Persamaan 2.1.

*Packet Loss = x 100%* (2.1)

### *Packet delivery ratio* (PDR)

PDR merupakan teknik penghitungan persentase jumlah paket yang berhasil diterima pada *node* tujuan dari total paket yang dikirim oleh *node* sumber (Istiqomah, Siregar and Kartikasari, 2018). Penghitungan PDR ini nantinya akan digunakan untuk membuktikan bahwa pencegahan dapat membuat rasio pengiriman data lebih tinggi daripada tanpa pencegahan. Secara rumus, perhitungan PDR ditunjukan pada Persamaan 2.2.

*PDR = x 100%* (2.2)

### *Delay*

*Delay* merupakan teknik penghitungan waktu yang dibutuhkan oleh node sumber untuk mengirimkan data sampai diterima oleh node tujuan (M, Primananda and Basuki, 2019). Pada perhitungan *Delay*, yang perlu diperhatikan adalah menghitung waktu paket diterima, dikurangi dengan waktu paket dikirim, maka akan ditemukan *delay* paket tersebut berhasil terkirim. Penghitungan *delay* dilakukan untuk mengukur seberapa lama keterlambatan paket untuk sampai ke tujuan antara dengan pencegahan dan tanpa pencegahan. Dalam penelitan ini menggunakan nilai total *delay* yang diperoleh dari total *delay* yang dibutuhkan untuk mengirimkan data dengan jumlah data dan waktu tertentu. Secara rumus, perhitungan *delay* ditunjukan pada Persamaan 2.3.

*Delay = Time Packet Recived – Time Packet Sent*

*Average Delay =* (2.3)

### Akurasi Deteksi

Akurasi didefinisikan sebagai persentase dari aliran trafik serangan yang secara benar diklasifikasikan sebagai serangan *black hole* dibandingkan dengan jumlah total serangan. Nilai akurasi akan diperoleh dengan melakukan pengujian serangan *black hole* pada sistem, kemudian membandingkan total serangan yang terdeksi dengan total jumlah serangan yang uji cobakan (Muliantara, 2012). Secara rumus, perhitungan akurasi deteksi ditunjukan pada Persamaan 2.4.

*Akurasi = x 100%* (2.4)

# METODOLOGI PENELITIAN

## Studi Literatur

Studi literatur digunakan menjadi referensi pengerjaan skripsi. Pada bagian studi literatur terdapat kajian pustaka dan dasar teori. Kajian pustaka merupakan referensi penulisan skripsi dari penelitian terdahulu yang terkait dengan serangan black hole dan mekanisme pendeteksiannya terhadap protokol routing AODV pada MANET. Dasar teori merupakan pengertian dari setiap teori yang menjadi referensi dalam penulisan skripsi pendeteksian serangan black hole pada MANET.

## Analisis Kebutuhan

Analisis kebutuhan memberikan informasi tentang kebutuhan yang harus dipenuhi pada penelitian. Pada langkah ini disebutkan beberapa kebutuhan fungsioanl yang harus dapat dipenuhi. Diperlukan kebutuhan perangkat lunak dan kebutuhan pendukung untuk memenuhi kebutuhan fungsional. Analisis kebutuhan digunakan untuk mengetahui kebutuhan apa saja yang menjadi pendukung dalam pengerjaan penelitian. Untuk menganalisis kebutuhan dapat dilihat dari kepustakaan dan dasar teori pada penelitian sebelumnya yang relevan dan dapat digunakan pada penelitian ini.

## Perancangan

Perancangan membahas tentang arsitektur sistem yang akan dibuat berdasarkan analisis kebutuhan fungsional. Pada tahap ini hal yang dirancang adlah perancangan serangan black hole, perancangan deteksi serangan black hole, perancangan parameter pengujian, perancangan sistem, dan perancangan skenario pengujian. Perancangan didasarkan pada dasar teori dan kajian pustaka penelitian terdahulu. Dengan perancangan diharapkan dapat memudahkan proses implementasi sistem.

### Perancangan Serangan *Black hole*

Perancangan serangan black hole dilakukan dengan modifikasi routing AODV dan merubah balasan pesan RREP menjadi memiliki nilai hop count rendah dan nilai sequence number lebih besar. Hal tersebut mengakibatkan penggunaan rute menuju black hole untuk pengiriman data dari node sumber menuju node tujuan. Paket data yang sampai di black hole akan dibuang sehingga tidak sampai ke node tujuan.

### Perancangan Deteksi Serangan *Black hole*

Perancangan modifikasi *routing* AODV akan dilakukan dengan menambahkan metode pengiriman pesan RREQ dengan alamat tujuan palsu. Alamat tujuan palsu menggunakan alamat 10.1.1.99. Pengiriman RREQ palsu digunakan untuk mencari apakah terdapat *black hole* pada jaringan MANET.

### Perancangan Parameter Pengujian

Perancangan parameter pengujian dibuat untuk mengukur beberapa parameter QoS dalam kondisi jaringan normal, dengan serangan black hole, dan menggunakan mekanisme deteksi serangan black hole. Perancangan dilakukan Degnan menambahkan beberapa script pada implementasi sistem. Parameter pengujian yang akan dilakukan pengukuran adalah nilai packet loss, nilai packet delivery ratio, dan nilai average delay yang dilakukan dengan mengamati aliran data dari node sumber dan node tujuan. Dalam pengujian akan dilakukan pengukuran akurasi deteksi terhadap serangan pada skenario posisi *black hole* yang berbeda.

### Perancangan Sistem

Perancangan sistem dibuat untuk menentukan parameter simulasi yang digunakan pada implementasi jaringan MANET. Pengaturan yang akan dilakukan adalah protokol routing, tipe pergerakan, jumlah node, luas area, jenis koneksi, dan waktu simulasi yang akan dilakukan. Pada perancangan sistem tidak terdapat serangan black hole.

### Perancangan Skenario Pengujian

Perancangan skenario pengujian dibuat untuk menentukan parameter uji yang digunakan untuk mengetahui kinerja dari protokol. Pengaturan skenario menjelaskan beberapa perancangan variasi posisi black hole, tipe pergerakan, dan jumlah node. Dari perancangan skenario pengujian akan digunakan untuk proses analisis kinerja protokol terhadap parameter QoS ketika jaringan normal, tanpa serangan black hole, dan dengan deteksi serangan black hole.

## Implementasi

Implementasi dilaksanakan berdasarkan perancancangan yang telah dibuat sebelumnya. Implementasi yang akan dilaksanakan adalah implementasi serangan black hole, implementasi deteksi black hole, implementasi sistem, dan implementasi skenario pengujian.

### Implementasi Serangan *Black hole*

Implementasi serangan black hole pada MANET dilakukan dengan mengububah skema balasan pesan RREP pada nilai hop count menjadi rendah dan nilai sequence number menjadi lebih besar. Selain itu juga terdapat beberapa fungsi untuk melakukan perilaku black hole pada protokol AODV.

### Implementasi Deteksi *Black hole*

Dalam pendeteksian *black hole* pada MANETdilakukan dengan mengirimkan pesan RREQ palsu dengan alamat tujuan yang sudah ditentukan. Alamat tujuan pada pesan RREQ palsu menggunakan alamat 10.1.1.99. Ketika *black hole* menerima pesan RREQ palsu maka akan segera membalas dengan pesan RREP dengan nilai *sequence number* tinggi dan hop *count* terendah. *Node* sumber dapat mengetahui apabila terdapat balasan dari pesan RREQ palsu maka *node* tersebut merupakan *black hole*.

### Implementasi Sistem

Implementasi sistem dilakukan untuk menjalankan protokol AODV pada jaringan MANET. Implementasi dilakukan sesuai dengan parameter pada perancangan sistem yang telah dilakukan. Implemenasi sistem bertujuan digunakan untuk memberikan gambaran tentang tahapan dan parameter yang digunakan dalam membangun simulasi AODV pada jaringan MANET.

### Implementasi Skenario Pengujian

Implementasi skenario pengujian dilaksanakan berdasarkan perancangan yang telah didefinisikan. Implementasi skenario pengujian meliputi variasi posisi *black hole*, tipe pergerakan dan jumlah node. Setiap skenario yang diimplementasikan menghasilkan nilai parameter pengujian QoS meliputi *packet loss, packet delivery ratio* , dan *delay*.

## Pembahasan Hasil Pengujian

Pembahasn hasil pengujian akan membahas hasil pengujian pada implementasi skenario pengujian dan dilakukan analisis terhadap parameter pengujian. Pengujian dilakukan untuk menguji fungsionalitas sistem dan kinerja sistem ketika dilakukan implementasi skenario pengujian. Tahap ini digunakan untuk menampilkan data dan mengevaluasi kinerja protokol dan mencari masalah yang timbul pada protokol.

## Penutup

Penutup akan membahas kesimpulan dan saran penulis yang diperoleh dari hasil pembahasan hasil pengujian protokol. Kesimpulan digunakan untuk menjawab rumusan masalah yang telah didefinisikan sebelumnya. Pada kesimpulan juga dipaparkan pembahasan dari hasil pengujian. Pada setiap penelitian akan memberikan saran yang membahas kekurangan yang terdapat pada sistem dan dapat dikembangkan untuk dibenahi oleh peneliti selanjutnya.

# ANALISA KEBUTUHAN DAN PERANCANGAN

## Deskripsi Umum Sistem

Sistem penanganan terhadap serangan *black hole* terdiri dari proses mencari *black hole* dan melakukan *Blacklist* terhadap *black hole*. proses mencari *black hole* dilakukan dengan cara mengirimkan pesan RREQ dengan tujuan palsu. *Blacklist* terhadap *black hole* dilakukan setelah terdapat laporan adanya *black hole*. Selanjutnya proses pencarian jalur menuju tujuan. Pengiriman data dilakukan dengan memilih jalur yang terdapat pada Tabel *routing* dan menghindari *Blacklist*.

## Kebutuhan Sistem

Kebutuhan sistem adalah kebutuhan yang diperlukan untuk perancangan sistem dan implementasi sistem menggunakan protokol AODV pada jaringan MANET. Kebutuhan pada penelitian ini terdiri dari kebutuhan fungsioanl dan kebutuhan non-fungsional.

### Kebutuhan Fungsional

Kebutuhan fungsioanl adalah kebutuhan yang harus ada di sistem dan proses yang dapat dilakukan sistem. Tabel 4.1 merupakan kebutuhan fungsional pada penelitian ini.

Tabel 4.1 Kebutuhan fungsional

|  |  |
| --- | --- |
| **No.** | **Kebutuhan Fungsional** |
| 1 | Sistem dapat mengimpIementasikan protokoI *routing* AODV pada jaringan MANET |
| 2 | Sistem dapat mengimpIementasikan serangan *black hole* terhadap AODV pada jaringan MANET. |
| 3 | Sistem dapat meIakukan pendeteksian serangan *black hole* terhadap AODV pada jaringan MANET. |

Tabel 4.1 menyajikan 3 kebutuhan fungsional yang harus dipenuhi pada penelitian ini. Kebutuhan fungsional yang harus dipenuhi adalah sistem dapat mengimpelentasikan protokol routing AODV, serangan black hole, dan melakukan pendeteksian serangan black hole terhadap AODV pada jaringan MANET.

### Kebutuhan Non-Fungsional

Kebutuhan non-fungsioanl adalah kebutuhan pendukung yang digunakan untuk perancangan dan implementasi sistem. Kebutuhan non-fungsional terdiri dari kebutuhan perangkat lunak yang menjelaskan perangkat lunak yang diperlukan dan kebutuhan pendukung yang menjelaskan perangkat laptop yang digunakan pada penelitian ini.

#### Kebutuhan perangkat lunak

Kebutuhan perangkat lunak menyajikan perangkat lunak apa saja yang digunakan untuk mengimplementasikan pendeteksian serangan black hole terhadap protokol AODV pada jaringan MANET. Kebutuhan perangkat lunak dapat dilihat pada Tabel 4.2.

Tabel 4.2 Kebutuhan perangkat lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat** | **Keterangan** |
| 1 | *Network Simulator 3 (NS3)* | Network Simulator berfungsi untuk membangun simulasi sistem dan menjalankan simulasi sistem. |
| 2 | *Linux Ubuntu 19* | Linux ubuntu berfungsi sebagai lingkunan untuk menjalankan perangkat lunak network simulator dan network animator. |
| 3 | *NetAnim* | NetAnim berfungsi untuk menampilkan implementasi sistem secara visual dalam bentuk animasi. |
| 4 | *Visual Studio Code* | Visual studio code berfungsi untuk menulis code untuk implementasi sistem, serangan black hole, dan deteksi serangan black hole. |

Penelitian pendeteksian serangan *black hole* pada MANETmembutuhkan 4 perangkat lunak seperti yang disajikan pada Tabel 4.2. perangkat lunak *network* simulator, NetAnim, dan Visual studio *code* digunakan untuk melakukan simulasi dan menulis *source* *code* untuk dapat menjalankan kebutuhan fungsional.

#### Kebutuhan pendukung

Kebuthan pendukung yang diperlukan untuk menjalankan kebutuhan non-fungsional dalam implementasi pendeteksian serangan black hole terhadap protokol AODV pada jaringan MANET dapat dilihat pada Tabel 4.3.

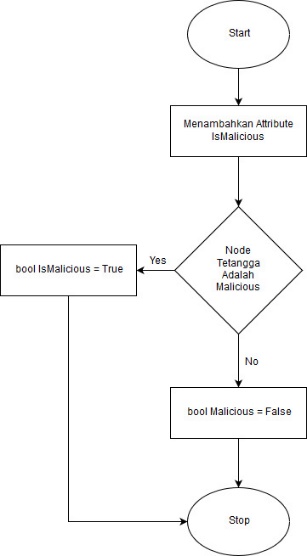
Tabel 4.3 Kebutuhan p*end*ukung

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat** | **Keterangan** |
| 1 | Laptop Asus X550ZE | Perangkat ini berfungsi untuk melakukan simulasi serangan dan pendeteksian *black hole* pada protokol AODV dan melakukan penyusunan dokumen penelitian. Perangkat memiliki spesifikasi, *processor* AMD A10, *Memory* 6 GB RAM, *Harddisk* 1000 GB, VGA AMD R6, |

Kebutuhan perangkat p*end*ukung pada penelitian pendeteksian serangan *black hole* pada MANETyang disajikan pada Tabel 4.3 adalah Laptop. Laptop digunakan untuk menjalankan berbagai perangkat lunak yang telah disebutkan pada Tabel 4.2.

## Perancangan Serangan *Black hole*

Pada perancangan implementasi *black hole*, implementasi dilakukan pada *file* aodv-*routing*-*Protocol*.cc. agar pada skrip dapat dibaca dan berlaku layaknya *black hole* asli ketika *black hole* telah ditambahkan. Metode penambahan *black hole* ini yaitu yang pertama menandiai *node* 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 *black hole* diaktifkan. Untuk memudahkan pembaca akan disajikan alur penandaan *Black hole* yang ditampilkan pada Gambar 4.1.

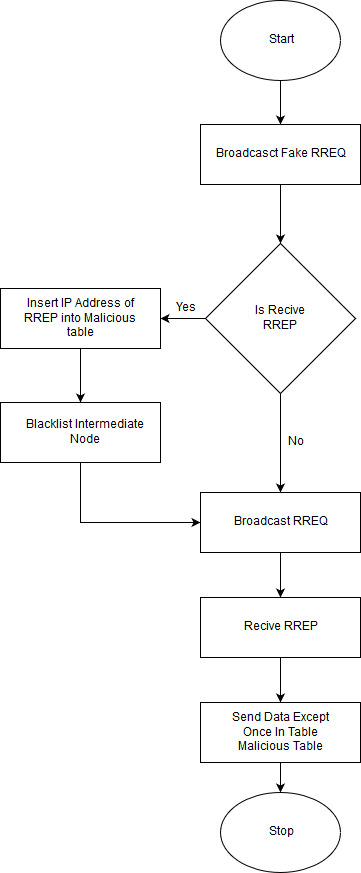


Gambar 4.1 Penandaan *black hole*

Pada Gambar 4.1 dijelaskan proses penentuan *node* *black hole* dilakukan dengan memberikan nilai ture pada attribute is*Malicious*. Setelah ditandai maka *node* akan memiliki sifat *black hole*.

## Perancangan Deteksi Serangan *Black hole*

Dalam pendeteksian *black hole* pada MANETdapat dengan mengirimkan pesan RREQ palsu. Alamat tujuan pada pesan RREQ palsu diubah menjadi alamat yang tidak ada pada jaringan. Ketika *black hole* menerima pesan RREQ palsu maka akan segera membalas dengan pesan RREP dengan nilai *sequence number* tinggi dan hop *count* ter*end*ah. *Node* sumber dapat mengetahui apabila terdapat balasan dari pesan RREQ palsu maka *node* tersebut merupkan *black hole*. Alur pendeteksian *Black hole* disajikan pada Gambar 4.2.



Gambar 4.2 Pendeteksian *black hole*

Berikut merupakan alur pendeteksian *black hole* yang digunakan pada penelitian ini :

1. Penambahan atribut *IsDetectBlack hole* dituiskan untuk melakukan simulasi maka *node* yang mengkatifkan atribut ini akan bertindak sebagai p*end*eteksi *node* *black hole*
2. Melakukan definisi alamat 10.1.1.99 sebagai tujuan pesan RREQ palsu
3. Pengecekan apakah terdapat *node* yang bersifat *malicious*. Pengecekan dilakukan oleh *node* sumber dengan mengirimkan pesan RREQ dengan alamat tujuan palsu. Apabila atribut *IsDetactBlackhole* aktif maka pada *method* s*endRequest* akan dimodifikasi dengan menambahkan *script* pengiriman pesan RREQ palsu
4. Penambahan fungsi pada *RecvReply()* untuk memberitahu pengguna bahwa terdapat serangan *black hole* pada jaringan. Ketika *node* sumber menerima pesan balasan dari RREQ palsu, maka sistem akan memberitahu bahwa pengirim RREP adalah *black hole*
5. Menandai *node* *black hole* yang membalas pesan RREQ palsu.
6. Proses pencarian rute dengan mengirim pesan RREQ asli. *Node* sumber melakukan *broadcast* RREQ untuk mencari rute ke *node* tujuan
7. *Node* sumber menerima semua pesan RREP balasan dari *node* tujuan dan *node* tetangga yang memiliki informasi rute
8. *Node* sumber dapat mengirim pesan melalui jalur yang diketahui

## Perancangan Parameter Pengujian

### Packet Loss

Parameter pengujian packet loss merupakan pengukuran jumlah paket data yang gagal transmisikan menuju tujuannya. Persamaan 4.1 merupakan rumus untuk mencari packet loss.

*Packet Loss = x 100%* (4.1)

Persamaan 4.1 dapat diimplementasikan pada network simulator yang digunakan untuk simulasi sistem dengan script berikut.

|  |  |
| --- | --- |
| 1  2 | std::cout << " Packet Loss : " << ((float)i->second.txBytes - (float)i->second.rxBytes) / (float)i->second.txBytes \* 100 << "%\n"; |

Pada script diatas merupakan persamaan rumus untuk menghitung nilai packet loss. Pada baris pertama menunjukan pengamatan transmisi data dari node sumber menuju node tujuan. Persamaan rumus packet loss dapat dihitung dengan (second.txBytes – second.rxBytes)/ second.txBytes \* 100%. Variabel txPacket merupakan jumlah paket yang dikirim node sumber dan second.rxPacket merupakan jumlah paket yang diterima node tujuan.

### Packet Delivery Ratio

Parameter pengujian packet delivery ratio (PDR) merupakan pengukuran jumlah paket yang diterima oleh node tujuan yang dibandingkan dengan jumlah paket yang dikirimkan oleh node sumber dalam satuan persen (%). Persamaan 4.2 merupakan rumus untuk mencari nilai packet delivery ratio.

*PDR = x 100%* (4.2)

Persamaan 4.2 dapat diimplementasikan pada network simulator yang digunakan untuk simulasi sistem dengan script berikut.

|  |  |
| --- | --- |
| 1  2 | std::cout << " Delivery : " << (float)i->second.rxBytes / (float)i->second.txBytes \* 100 << "%\n"; |

Pada script diatas merupakan persamaan rumus untuk menghitung nilai packet delivery ratio. Pada baris pertama menunjukan pengamatan transmisi data dari node sumber menuju node tujuan. Persamaan rumus packet delivery ratio dapat dihitung dengan second.rxBytes/ second.txBytes \* 100%. Variabel txPacket merupakan jumlah paket yang dikirim node sumber dan second.rxPacket merupakan jumlah paket yang diterima node tujuan.

### Delay

Parameter pengujian delay merupakan pengukuran waktu tunda yang dibutuhkan untuk mentransmisikan data dari node sumber sampai diterima node tujuan. Persamaan 4.3 merupakan rumus untuk menghitung nilai delay.

*Delay = Time Packet Recived – Time Packet Sent*

*Average Delay =* (4.3)

Persamaan 4.3 dapat diimplementasikan pada network simulator yang digunakan untuk simulasi sistem dengan script berikut.

|  |  |
| --- | --- |
| 1  2 | std::cout << " Delay : " << i->second.delaySum / (second.rxBytes /540) << "\n"; |

Pada script diatas merupakan perhitungan nilai *delay*. Pengukuran dilakukan dengan mengukur waktu yang dibutuhkan dalam pengiriman data dari node sumber menuju node tujuan.

### Akurasi Deteksi

Parameter akurasi deteksi merupakan persentase dari aliran *trafik* serangan yang secara benar diklasifikasikan sebagai serangan *black hole* dibandingkan dengan jumlah total serangan. Nilai akurasi akan diperoleh dengan melakukan pengujian serangan *black hole* pada sistem, kemudian membandingkan total serangan yang terdeteksi dengan total jumlah serangan yang uji cobakan. Persamaan 4.4 merupakan rumus untuk menghitung akurasi deteksi.

*Akurasi = x 100%* (4.4)

Nilai yang disimulasikan dimasukkan pada tabel input sebagaimana dalam Tabel 4.2 berikut.

**Tabel 4.2 Perancangan Input Parameter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Aktivitas | Posisi Black Hole | Hasil Pendeteksian | Status |
|  | A | B | C | D |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

Parameter yang divariasikan dapat dicatatkan dalam kolom-kolom terkait. Dalam tabel input, susunannya adalah sebagai berikut:

1. Aktivitas ketika terdapat deteksi serangan *black hole* diletakkan kolom A
2. Posisi node *black hole* yang berbeda diletakkan dalam kolom B
3. Hasil pendeteksian black hole dalam C
4. Status benar dan salah terhadap hasil pendeteksian dalam kolom D

## Perancangan Sistem

Perancangan sistem dilakukan untuk memberikan gambaran implementasi protokol AODV pada jaringan MANET dan melihat kinerja dari protokol dalam kondisi normal. Konfigurasi yang ditetapkan pada perancangan ini dengan menggunkan protokol routing AODV. Jumlah node yang dibuat berjumlah 50. Penempatan node dilakukan secara acak menggunakan tipe mobilitas random waypoint. Semua node akan disebar secara acak pada area seluas 1000 meter x 1000 meter. Node bergerak secara bebas dengan kecepatan antara 1 sampai 10m/s. dalam pengiriman data antar node menggunakan koneksi UDP dengan jenis paket CBR. Ukuran paket yang dikirim adalah 512 bytes dari node sumber (1) menuju node (16) dengan anggapan tidak ada serangan black hole. Konfigurasi implementasi sistem menggunakan protokol AODV dapat dilihat pada Tabel 4.4.

Tabel 4.4 Konfigurasi implementasi sistem

|  |  |
| --- | --- |
| **Parameter** | **Keterangan** |
| Protokol *Routing* | AODV |
| Jumlah *Node* | 50 |
| Jenis Koneksi | UDP |
| Jenis Paket | CBR |
| Ukuran Paket | 512 *Bytes* |
| Ukuran *Rate* CBR | 1000 Kb |
| Luas Area | 1000 meter x 1000 meter |
| Waktu Simulasi | 100 detik |
| Tipe Mobilitas | *Random waypoint* |
| Kecepatan *Node* | 1 sampai 10 m/s |

## Perancangan Skenario Pengujian

Perancangan skenario pengujian dilakukan untuk melihat dan mengukur kinerja protokol terhadap skenario posisi node black hole, tipe pergerakan, dan jumlah node. Dari skenario pengujian akan didapatkan data untuk menganalisis protokkol.

### Posisi Node Black Hole

Perancangan skenario variasi posisi *black hole* dilakukan untuk melihat kinerja dari protokol AODV terhadap variasi posisi *black hole*. Perancangan variasi posisi black hole dimaksudkan untuk melihat apakah mekanisme Konfigurasi yang ditetapkan pada perancangan ini dengan menggunkan protokol routing AODV. Jumlah node yang dibuat berjumlah 20. Penempatan node dilakukan secara tetap menggunakan tipe mobilitas constant. Semua node akan ditempatkan secara tetap pada area seluas 600 meter x 600 meter. Posisi node black hole berada antara node (1, 3, 8, dan 13). dalam pengiriman data antar node menggunakan koneksi UDP dengan jenis paket CBR. Ukuran paket yang dikirim adalah 512 bytes dari node sumber (1) menuju node (19) dengan anggapan tidak ada serangan black hole. Konfigurasi implementasi sistem menggunakan protokol AODV dapat dilihat pada Tabel 4.5.

Tabel 4.5 Konfigurasi implementasi variasi posisi black hole

|  |  |
| --- | --- |
| **Parameter** | **Keterangan** |
| Protokol *Routing* | AODV |
| Jumlah *Node* | 50 |
| Jumlah *Node* *black hole* | 0 dan 1 |
| Posisi *Node black hole* | Node 0, 3, 8, dan 13 |
| Jenis Koneksi | UDP |
| Jenis Paket | CBR |
| Ukuran Paket | 512 *Bytes* |
| Ukuran *Rate* CBR | 100 Kb |
| Luas Area | 600 x 600 meter |
| Waktu Simulasi | 50 detik |
| Tipe Mobilitas | *Constant* |

### Tipe Pergerakan

Perancangan skenario pengujian variasi tipe pergerakan dilakukan untuk melihat kinerja protokol terhadap serangan black hole dan mekanisme pendeteksian serangan black hole menggunakan tipe pergerakan yang berbeda. Konfigurasi yang ditetapkan pada perancangan ini dengan menggunkan protokol routing AODV. Jumlah node yang dibuat berjumlah 50. Penempatan node dilakukan secara acak menggunakan tipe mobilitas pergerakan random waypoint, pergerakan random walk, dan pergerakan random direction. Semua node akan disebar secara acak pada area seluas 1000 meter x 1000 meter. Node bergerak secara bebas dengan kecepatan antara 1 sampai 10m/s. dalam pengiriman data antar node menggunakan koneksi UDP dengan jenis paket CBR. Ukuran paket yang dikirim adalah 512 bytes dari node sumber (1) menuju node (16) dan posisi node black hole berada pada node (1). Konfigurasi implementasi sistem menggunakan protokol AODV dapat dilihat pada Tabel 4.6.

Tabel 4.6 Konfigurasi implementasi variasi tipe pergerakan

|  |  |
| --- | --- |
| **Parameter** | **Keterangan** |
| Protokol *Routing* | AODV |
| Jumlah *Node* | 50 |
| Jumlah *Node* *black hole* | 0 dan 1 |
| Jenis Koneksi | UDP |
| Jenis Paket | CBR |
| Ukuran Paket | 512 *Bytes* |
| Ukuran *Rate* CBR | 100 Kb |
| Luas Area | 1000 meter x 1000 meter |
| Waktu Simulasi | 1000 detik |
| Tipe Mobilitas | *Pergerakan random waypoint, pergerakan random walk, dan pergerakan random direction* |
| Kecepatan *Node* | 1 sampai 10 m/s |

### Jumlah *Node*

Perancangan skenario pengujian variasi jumlah node dilakukan untuk melihat kinerja protokol terhadap serangan black hole dan mekanisme pendeteksian serangan black hole menggunakan jumlah node yang berbeda. Konfigurasi yang ditetapkan pada perancangan ini dengan menggunkan protokol routing AODV. Jumlah node yang dibuat berjumlah 30, 40, dan 50. Penempatan node dilakukan secara acak menggunakan tipe mobilitas pergerakan random waypoint, pergerakan random walk, dan pergerakan random direction. Semua node akan disebar secara acak pada area seluas 1000 meter x 1000 meter. Node bergerak secara bebas dengan kecepatan antara 1 sampai 10m/s. dalam pengiriman data antar node menggunakan koneksi UDP dengan jenis paket CBR. Ukuran paket yang dikirim adalah 512 bytes dari node sumber (1) menuju node (16) dan posisi node black hole berada pada node (1). Konfigurasi implementasi sistem menggunakan protokol AODV dapat dilihat pada Tabel 4.7.

Tabel 4.7 Konfigurasi implementasi variasi jumlah *node*

|  |  |
| --- | --- |
| **Parameter** | **Keterangan** |
| Protokol *Routing* | AODV |
| Jumlah *Node* | 30, 40, dan 50 |
| Jumlah *node* *black hole* | 0 dan 1 |
| Jenis Koneksi | UDP |
| Jenis Paket | CBR |
| Ukuran Paket | 512 *Bytes* |
| Ukuran *Rate* CBR | 100 Kb |
| Luas Area | 1000 meter x 1000 meter |
| Waktu Simulasi | 1000 detik |
| Tipe Mobilitas | *Random waypoint* |
| Kecepatan *Node* | 1 sampai 10 m/s |

# IMPLEMENTASI

## Instalasi NS 3

Network simulator 3 (NS3) digunakan untuk membangun simulasi jaringan MANET. Sebelum menggunakan NS3 diperlukan instalasi dan beberapa aplikasi pendukung. Berikut merupakan cara melakukan instalasi NS3 menggunakan terminal ubuntu.

1. Instalasi aplikasi p*end*ukung yang dibutuhkan *Network* Simulator 3 seperti C++ compiler, Python, Git, tar, bunzip2. Aplikasi yang dibutuhkan dapat diinstal secara *default*.
2. Download *file* *Network* Simulator 3 untuk pengguna baru yang ingin mengunduh dan melakukan simulasi jaringan menggunakan *network* simulator 3 versi terbaru. Perintah pada ubuntu dapat dilihat pada *script* dibawah ini.

|  |
| --- |
| $ cd  $ mkdir (tempat menyimpan data NS3 contoh workspace)  $ cd workspace  $ wget (download file NS3 pada alamat [https:/www.nsnam.org/release/ns-allinone-3.29.tar.bz2](https://www.nsnam.org/release/ns-allinone-3.29.tar.bz2))  $ tar xjf ns-allinone-3.29.tar.bz2 |

1. *Building* *Network* Simulator 3. Perintah pada ubuntu dapat dilihat pada *script* dibawah ini.

|  |
| --- |
| $ ./build.py --enable-examples --enable-tests |

1. *Testing* *Network* Simulator 3. Perintah pada ubuntu dapat dilihat pada *script* dibawah ini.

|  |
| --- |
| $ ./test.py |

1. *Running* a *script*. Perintah pada ubuntu dapat dilihat pada *script* dibawah ini.

|  |
| --- |
| $ ./waf --run hello-simulator |

## Implementasi Serangan *Black hole*

Impelemtasi serangan black hole akan dijalankan dengan menambahkan beberapa fungsi pada protokol AODV. Node black hole akan membalas semua pesan RREQ yang diterima dengan pesan RREP yang memiliki nilai hop count kecil dan sequence number lebih besar. Berikut merupakan beberapa fungsi yang ditambahkan.

Pada baris pertama terdapat penambahan atribut IsMalicious untuk menandai node yang akan menjadi black hole. Pada konfigurasi awal nilai IsMalicios adalah false. IsMalicios bernilai true ketika terdapat node black hole pada jaringan MANET. Script dibawah merupakan implementasi penambahan atribut IsMalicious.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | .AddAttribute ("Is*Malicious*",  "Is the *node* *malicious*",  BooleanValue (false)  MakeBooleanAccessor (  &*RoutingProtocol*::Set*Malicious*Enable,  &*RoutingProtocol*::Get*Malicious*Enable),  MakeBooleanChecker ()) |

Penambahan script pada serangan balck hole untuk mebuang paket paket. Ketika atribut IsMalicious node bernilai true apabila terdapat data yang akan dikirimkan menuju node sumber melewati node black hole maka data akan dibuang. Simulasi pada network simulator akan menampilkan pesan packet telah dibuang(drop). Berikut script untuk membuang paket data.

|  |  |
| --- | --- |
| 1  2  3  4 | if(Is*Malicious*){  std :: cout <<"Launching *Black hole* Attack!  Packet dropped . . . \n";  return false;} |

Penambahan script untuk membalas setiap pesan RREQ yang diterima dengan mengakui memiliki rute menuju node tujuan. Terdapat beberapa informasi yang diubah pada balasan pesan RREP agar node sumber menggunakan rute yang diberikan black hole. Nilai hop count pada pesan RREP dibuat menjadi paling dekat menuju node tujuan dengan nilai 1. Nilai sequence number diubah dengan menjumlahkan sequence number dengan 100 sehingga memiliki nilai yang lebih besar. Script dibawah merupakan pembuatan pesan RREP oleh black hole.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  10  11  12  13  14  15  16  17  18  19  20  21 | if (Is*Malicious* || (!rreqHeader.GetDestinationOnly ()  && toDst.GetFlag () == VALID))  {  m\_*routing*Table.Lookup*Route* (origin, toOrigin);  if(Is*Malicious*)  {  Ptr<Net*Device*> dev = m\_ipv4->GetNet*Device*  (m\_ipv4->GetInterfaceForAddress (receiver));  *Routing*Table*Entry*  falseToDst(dev,dst,true,rreqHeader.GetDstSeqno()+100,  m\_ipv4->GetAddress (m\_ipv4->GetInterfaceForAddress  (receiver),0),1,dst,Active*Route*Timeout);  S*end*ReplyByIntermediate*Node* (falseToDst, toOrigin,  rreqHeader.GetGratiousRrep ());  return;  }  S*end*ReplyByIntermediate*Node* (toDst, toOrigin,  rreqHeader.GetGratiousRrep ());  return;  } |

Penambahan script pada node dengan nilai IsMalicious adalah true untuk membuat nilai hop count pada pesan RREP menjadi 1. Script dibawah merupakan cara menset nilai hop count menjadi 1.

|  |  |
| --- | --- |
| 1  2 | If (Is*Malicious*) {  rrepHeader.SetHop*Count* (1);} |

## Implementasi Deteksi Serangan *Black hole*

Implementasi Deteksi serangan black hole akan menjelaskan skema pendeteksian seranngan balck hole terhadap protol AODV pada jaringan MANET. pendeteksian dilakukan dengan menambahkan beberapa fungsi pada protokol AODV. Berikut merupakan beberapa fungsi yang ditambhakan pada file protokol AODV.

Pada baris pertama terdapat penambahan atribut IsDetectMalicious untuk menandai node dapat mendeteksi serangan black hole. Pada konfigurasi awal nilai IsDetectMalicious adalah false. Ketika node IsDetectMalious bernilai true maka node bertindak sebagai *node* pendeteksi. Script dibawah menunjukan implementasi penambahan atribut IsDetectMalicious.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | //tambahan atribute deteksi  .AddAttribute ("IsDetect*Black hole*", "Can the *node* detect *black hole*s", BooleanValue (false),MakeBooleanAccessor (&*RoutingProtocol*::SetDetect*Black hole*Enable, &*RoutingProtocol*::GetDetect*Black hole*Enable),  MakeBooleanChecker ()) |

Mengirimkan pesan RREQ palsu dengan alamat tujuan 10.1.1.99. *node* sumber akan mengirimkan pesan RREQ palsu terlebih dahulu sebelum mengirimkan pesan RREQ asli. Proses pengiriman pesan RREQ dapat dilihat pada *script* dibawah.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | //kirim rreq palsu  if (IsDetect*Black hole* && fakeAddress.IsEqual(Ipv4Address("0.0.0.0")))  {  std::cout<<"S*end*ing fake RREQ to detect *Black hole*  from"<< header.Get*Source*()<< " \n";  fakeAddress = Ipv4Address("10.1.1.99");  SetFakeAddress(fakeAddress);  S*endRequest*(fakeAddress);  } |

Pengecekan pada Tabel *routing* untuk melakukan *Blacklist* rute yang melewati *node* *black hole*. Pengecekan dilakukan dengan melihat Tabel *routing* memiliki alamat palsu. Apabila terdapat rute menuju alamat salah/palsu maka akan dilakukan penghapusan rute. *Node* akan mengirimkan pesan RERR untuk mencari rute lain menuju *node* sumber. Proses pengecekan rute *node* *Blacklist* dapat dilihat pada *script* dibawah.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | //cek *Blacklist*  if(*route*-> GetGateway()== Get*MaliciousNode*())  {  std::cout<<Get*MaliciousNode*()<<" *Blacklist* *node*, drop *route* !\n";  m\_*routing*Table.Delete*Route* (dst);  S*end*RerrWhenNo*Route*T*of*orward (dst, 0, origin);  return false;  } |

Pengecekan alamat pengirim pesan RREP. Pesan RREP akan diterima jika alamat tidak sama dengan alamat *node* *black hole* yang terdeteksi. Apabila alamat pengirim sama dengan alamat *node* *black hole* yang terdeteksi maka pesan akan diabaikan. Proses pengecekan asal dari pesean RREP dapat dilihat pada *script* dibawah.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | //cek *Blacklist* *black hole*  if(s*end*er != Get*MaliciousNode*())  {  RecvReply (packet, receiver, s*end*er);  break;  } |

Melakukan deteksi *node* *black hole* dengan mengecek pesan RREP yang masuk. Pengecekan dilakukan dengan melihat alamat tujuan dari pesan RREP. Apabila alamat pesan RREP adalah alamat palsu maka pengirim pesan RREP akan dicurigai. Sistem akan memberitahukan *node* *black hole* telah terdeteksi dan memasukan alamat *node* *black hole* kedalam *Blacklist*. Proses pendeteksian *node* *black hole* dapat dilihat pada *script* dibawah.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | //deteksi  //pelaku dengan tujuan rrep alamat palsu  if (IsDetect*Black hole* && dst == Ipv4Address("10.1.1.99")&& Get*MaliciousNode*() == Ipv4Address("0.0.0.0"))  {  std::cout<<"*Black hole* detected !\n";  Ipv4Address pelaku = s*end*er;  Set*MaliciousNode*(pelaku);  std :: cout << Get*MaliciousNode*() << " *Malicious*  *Node* ! receiver "<< receiver << " \n";  return;  } |

## Implementasi Sistem

Implementasi sistem terbagi menjadi beberapa tahapan. Pada langkah awal dilakukan pembuatan node berjumlah 50 dengan 49 node biasa dan 1 node black hole. Penempatan node dilakukan secara acak menggunakan pergerakan random waypoint. Node bergerak secara bebas pada area 1000 meter x 1000 meter. Node bergerak secara bebas dengan kecepatan antara 1 sampai 10 m/2. Pembuatan node dapat dilihat pada script dibawah.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | *Node*Container c;  c.Create (50);  ObjectFactory pos;  pos.SetTypeId ("ns3::*Random*RectanglePositionAllocator");  pos.Set ("X", StringValue  ("ns3::Uniform*Random*Variable[Min=0.0 | Max=1000.0]"));  pos.Set ("Y", StringValue  ("ns3::Uniform*Random*Variable[Min=0.0 |Max=1000.0]"));  *Mobility*Helper *mobility*;  Ptr<PositionAllocator> posAlloc = pos.Create() -> GetObject  <PositionAllocator>();  *mobility*.SetPositionAllocator (posAlloc);  *mobility*.Set*Mobility*Model ("ns3::*RandomWaypointMobility*Model",  "Speed", StringValue  ("ns3::Uniform*Random*Variable[Min=1|Max=10]"),  "Pause", StringValue  ("ns3::*ConstantRandom*Variable[*Constant*=1.0]"),  "PositionAllocator",  PointerValue(posAlloc));  *mobility*.Install (c); |

Pengaturan wifi yang digunakan dalam komunikasi antar node dan pengiriman data. Pada penelitian ini menggunakan wifi dengan standar 802.11b. jangkaun wifi diatur menjadi 250 meter. terdapat implementasi pembuatan cahnnel dan device untuk menghubungkan antar node. Script dibawah merupakan pengaturan media transmisi yang digunakan.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | WifiHelper wifi;  YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::*Default* ();  wifiPhy.SetPcapDataLinkType (Yans WifiPhyHelper::DLT\_IEEE802\_11);  Yans Wifi*Channel*Helper wifi*Channel* ;  wifi*Channel*.SetPropagation*Delay*  ("ns3::*Constant*SpeedPropagation*Delay*Model");  wifi*Channel*.AddPropagationLoss  ("ns3::TwoRayGroundPropagationLossModel","*System*Loss",  DoubleValue(1),"HeightAboveZ", DoubleValue(1.5));  wifiPhy.Set ("TxPower Start", DoubleValue(33));  wifiPhy.Set ("TxPower *End*", DoubleValue(33));  wifiPhy.Set ("TxPower Levels", UintegerValue(1));  wifiPhy.Set ("Tx Gain", DoubleValue(0));  wifiPhy.Set ("Rx Gain", DoubleValue(0));  wifiPhy.Set ("Energy DetectionThreshold", DoubleValue(-61.8));  wifiPhy.Set ("CcaMode1Threshold", DoubleValue(-64.8));  wifiPhy.Set*Channel* (wifi*Channel*.Create ());  WifiMacHelper wifiMac;  wifiMac.SetType ("ns3::AdhocWifiMac");  wifi.SetStandard (WIFI\_PHY\_STANDARD\_80211b);  wifi.SetRemoteStationManager  ("ns3::*ConstantRate*WifiManager","DataMode",StringValue(phyMode),  "ControlMode",StringValue(phyMode));  Net*Device* Container *device*s;  *device*s = wifi. Install (wifiPhy, wifiMac, c); |

Menentukan protokol routing dan range IP Address yang digunakan pada simulasi. Simulasi menggunakan protokol routing AODV untuk mengatur komunikasi antar node pada jaringan MANET. Range IP addres pada jaringan dengan base 10.1.1.0. script dibawah merupakan implementasi pengaturan protokol routing dan penentuan range IP Address.

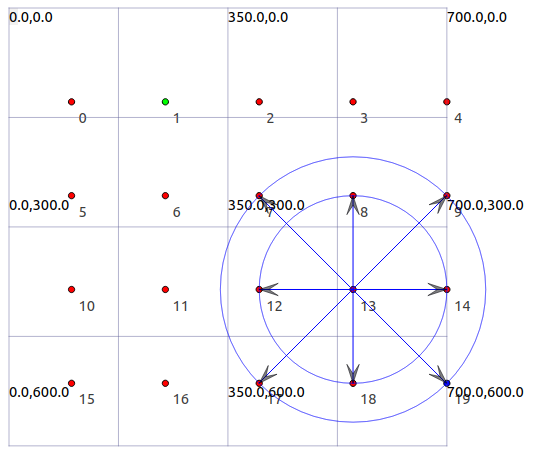
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | AodvHelper aodv;  Internet StackHelper internet;  internet.Set *Routing*Helper (aodv);  internet.Install (c);  Ipv4Address Helper address;  address.SetBase ("10.1.1.0", "255.255.255.0");  Ipv4Interface Container *node*Interfaces;  *node*Interfaces = address.Assign (*device*s); |

Membuat aplikasi untuk melakukan pengiriman data dari node sumber menuju node tujuan. Penentuan jenis koneksi dan waktu simulasi pada yang akan digunakan. Pada peneltian ini node (1) berperan sebagai pengirim dan node (15) sebagai penerima paket data. Simulasi dilakukan dalam waktu 1000 detik dengan jenis koneksi UDP. Script dibawah merupakan pembuatan aplikasi dalam penelitian ini.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | Address sinkAddress (InetSocketAddress (*node* Interfaces.GetAddress  (15), 6));  PacketSinkHelper packetSinkHelper ("ns3::UdpSocketFactory",  InetSocketAddress (Ipv4Address::GetAny (), 6));  Application Container sinkApps = packetSinkHelper.Install (c.Get  (15));  sinkApps.Start (Seconds (0.0));  sinkApps.Stop (Seconds (1000.0));  Ptr<Socket> ns3UdpSocket = Socket::CreateSocket (c.Get (1),  UdpSocket Factory::GetTypeId ());  Ptr<MyApp> app = CreateObject<MyApp> ();  app -> Setup (ns3Udp Socket, sinkAddress, 512, 100, Data*Rate*  ("100Kbps"));  c.Get (1) -> AddApplication (app);  app -> SetStartTime (Seconds (2.0));  app -> SetStopTime (Seconds (1000.0)); |

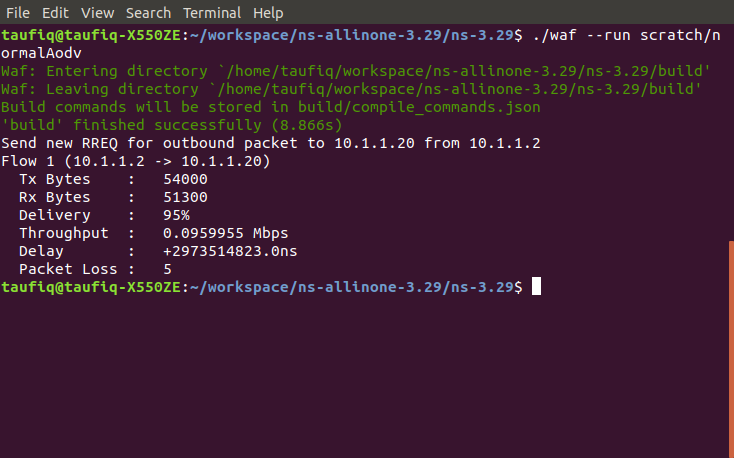
## Implementasi Skenario Pengujian

### Posisi Node Black Hole



Gambar 5.1 Tampilan NetAnim 20 *node* kondisi normal

Gambar 5.1 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.1 ada 3 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (19) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.2 Hasil NS3 20 *node* kondisi normal

Gambar 5.2 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi normal menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi normal pada Gambar 5.2 menunjukan hasil dengan nilai *packet* *loss* 5%, *packet* *delivery* *ratio* 95%, dan *delay* 31300156,03 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

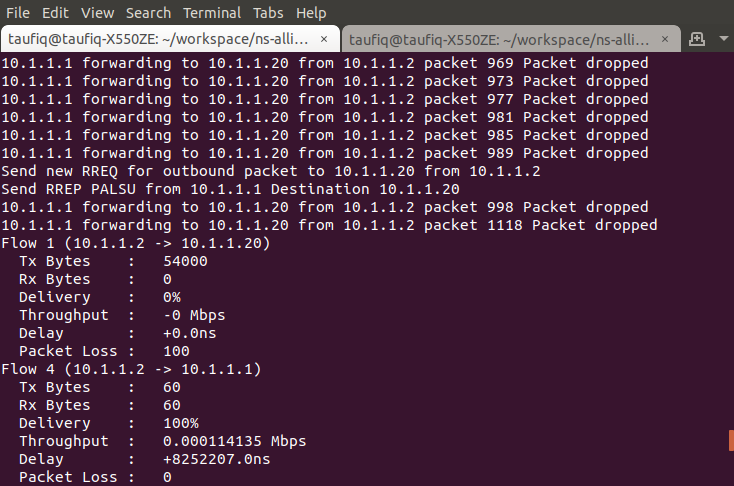
#### Node 0

##### Simulasi kondisi terdapat black hole



Gambar 5.3 Tampilan NetAnim 20 *node* kondisi terdapat black hole

Gambar 5.3 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.3 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (0) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.4 Hasil NS3 20 *node* kondisi terdapat black hole

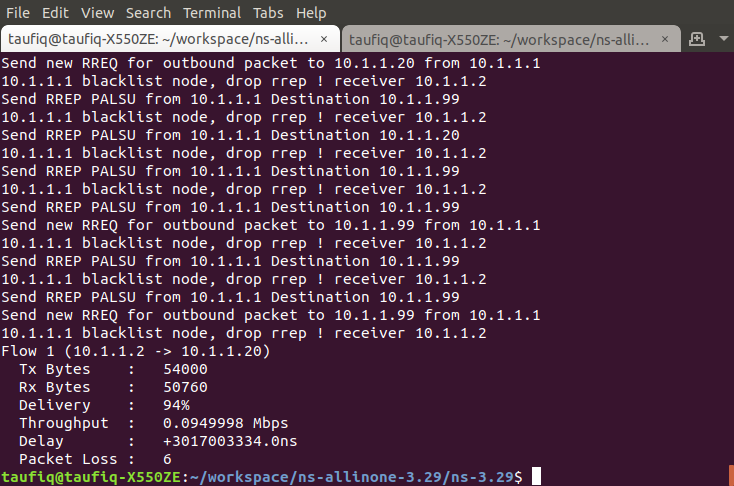
Gambar 5.4 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.4 menunjukan hasil dengan nilai *packet* *loss* 100%, *packet* *delivery* *ratio* 0%, dan *delay* 0 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

##### Simulasi kondisi terdapat deteksi black hole



Gambar 5.5 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole

Gambar 5.5 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.5 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (0) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.

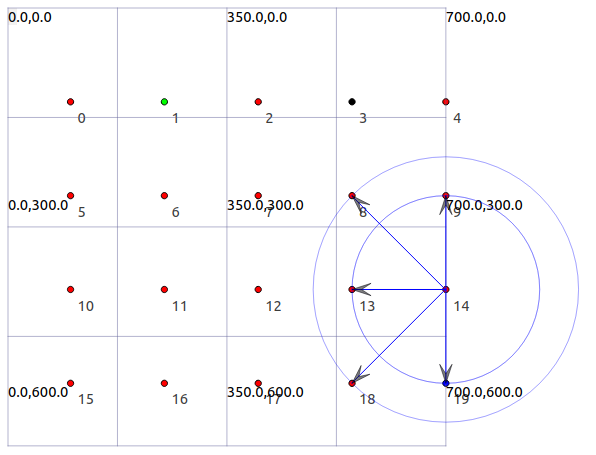


Gambar 5.6 Hasil NS3 20 *node* kondisi terdapat deteksi black hole

Gambar 5.6 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.6 menunjukan hasil dengan nilai *packet* *loss* 6%, *packet* *delivery* *ratio* 94%, dan *delay* 32095780,15 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

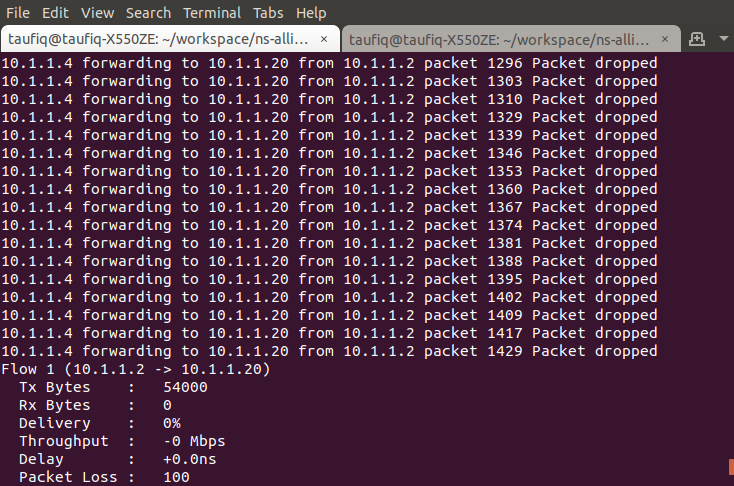
#### Node 3

##### Simulasi kondisi terdapat black hole



Gambar 5.7 Tampilan NetAnim 20 *node* kondisi terdapat black hole

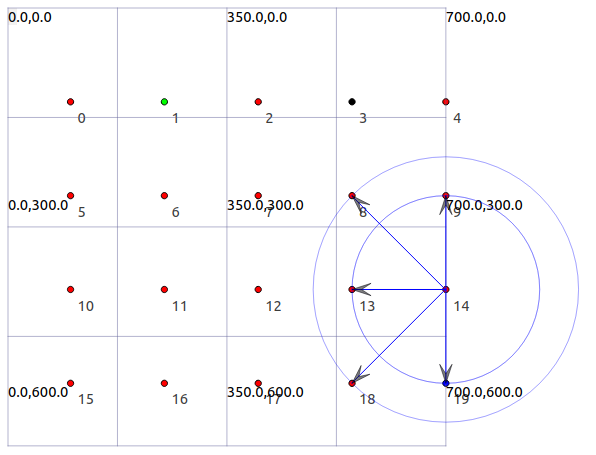
Gambar 5.7 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.7 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (3) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.8 Hasil NS3 20 *node* kondisi terdapat black hole

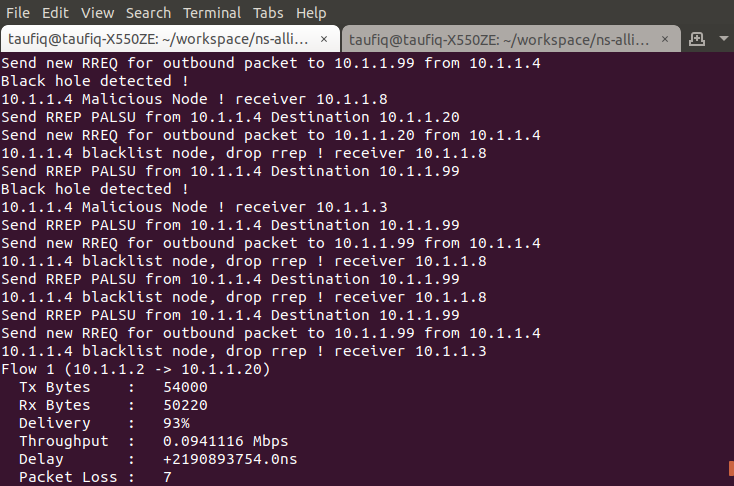
Gambar 5.8 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.8 menunjukan hasil dengan nilai *packet* *loss* 100%, *packet* *delivery* *ratio* 0%, dan *delay* 0 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

##### Simulasi kondisi terdapat deteksi black hole



Gambar 5.9 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole

Gambar 5.9 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.9 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (3) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.

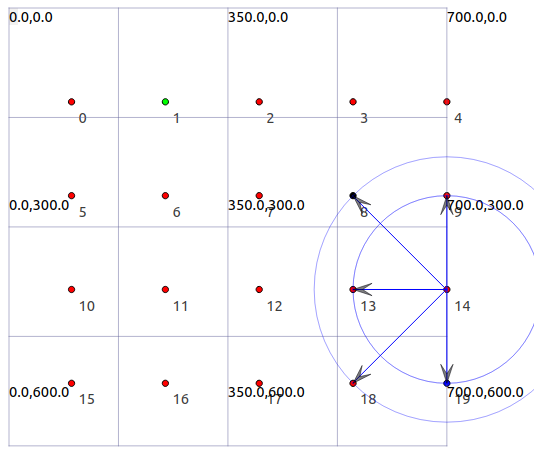


Gambar 5.10 Hasil NS3 20 *node* kondisi terdapat deteksi black hole

Gambar 5.10 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.10 menunjukan hasil dengan nilai *packet* *loss* 7%, *packet* *delivery* *ratio* 93%, dan *delay* 23557997,35 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

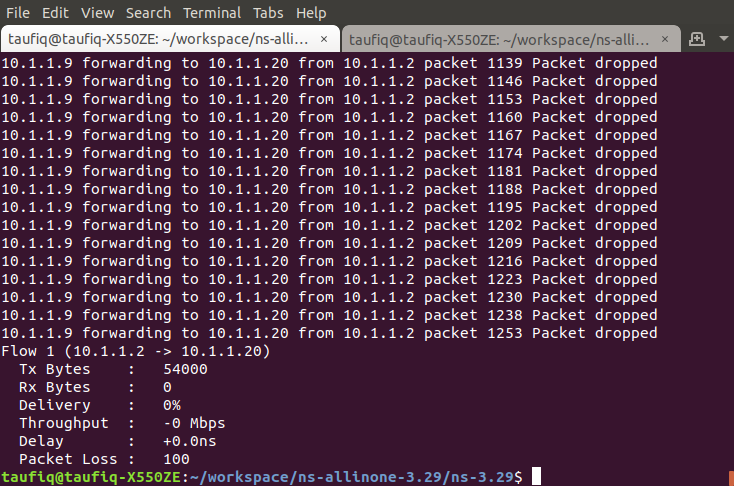
#### Node 8

##### Simulasi kondisi terdapat black hole



Gambar 5.11 Tampilan NetAnim 20 *node* kondisi terdapat black hole

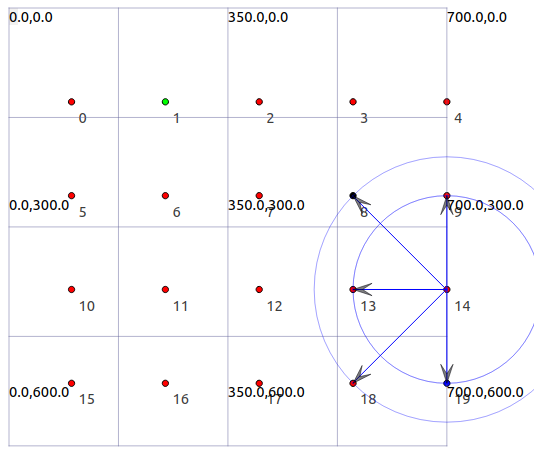
Gambar 5.11 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.11 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (0) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.12 Hasil NS3 20 *node* kondisi terdapat black hole

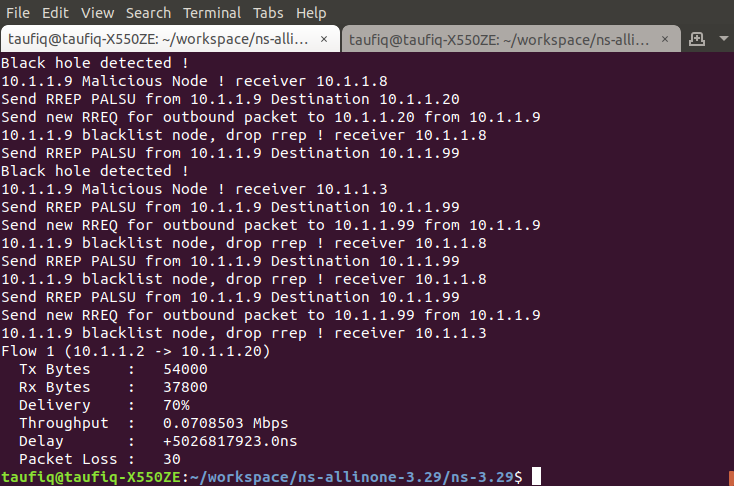
Gambar 5.12 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.12 menunjukan hasil dengan nilai *packet* *loss* 100%, *packet* *delivery* *ratio* 0%, dan *delay* 0 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

##### Simulasi kondisi terdapat deteksi black hole



Gambar 5.13 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole

Gambar 5.13 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.13 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (8) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.

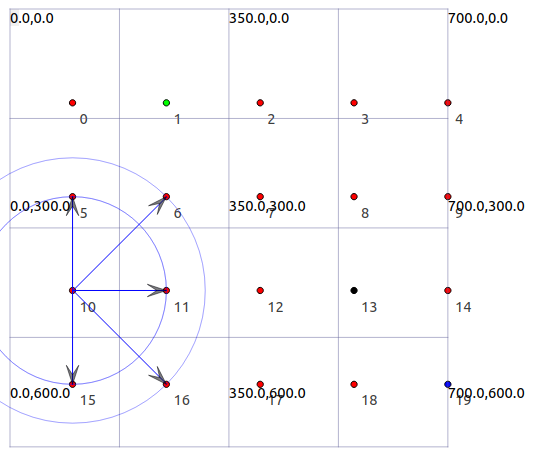


Gambar 5.14 Hasil NS3 20 *node* kondisi terdapat deteksi black hole

Gambar 5.14 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.14 menunjukan hasil dengan nilai *packet* *loss* 30%, *packet* *delivery* *ratio* 70%, dan *delay* 71811684,61 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

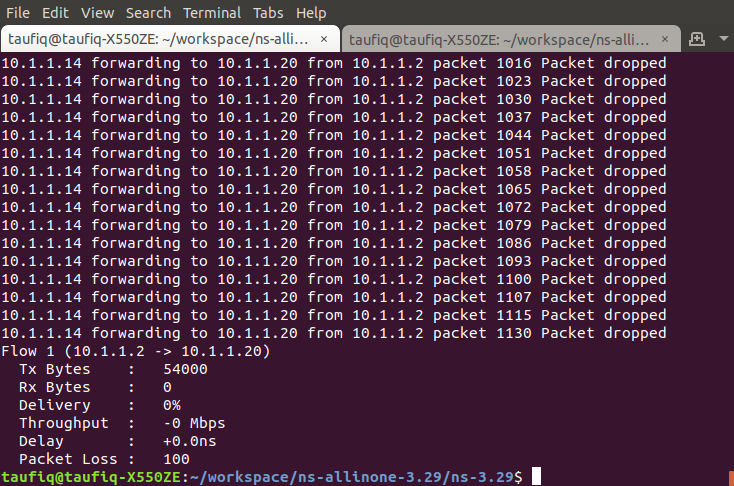
#### Node 13

##### Simulasi kondisi terdapat black hole



Gambar 5.15 Tampilan NetAnim 20 *node* kondisi terdapat black hole

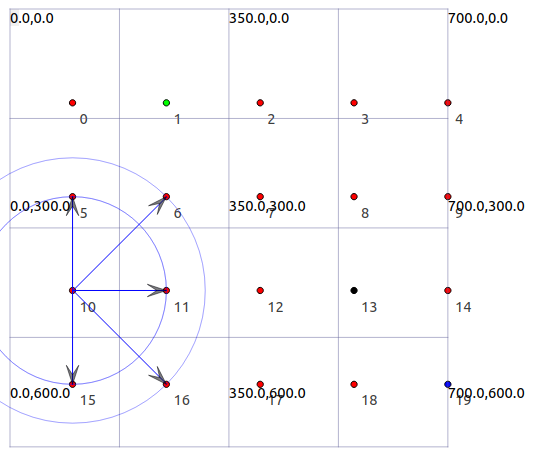
Gambar 5.15 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.15 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (13) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.16 Hasil NS3 20 *node* kondisi terdapat black hole

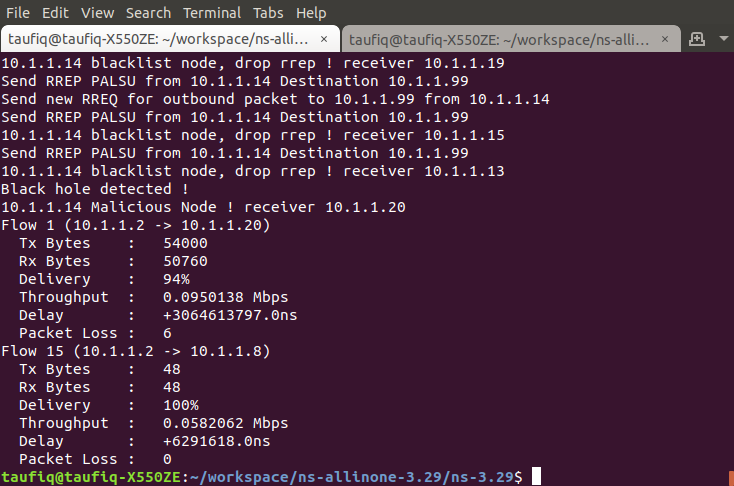
Gambar 5.16 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.16 menunjukan hasil dengan nilai *packet* *loss* 100%, *packet* *delivery* *ratio* 0%, dan *delay* 0 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

##### Simulasi kondisi terdapat deteksi black hole



Gambar 5.17 Tampilan NetAnim 20 *node* kondisi terdapat deteksi black hole

Gambar 5.17 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 600 meter x 600 meter dan menggunakan tipe mobilitas *constant*. *Node* menempati posisi tetap dan tidak dapat bergerak secara bebas. Pada Gambar 5.17 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber, *node* (19) berperan sebagai tujuan, dan *node* (13) berperan sebagai *black hole*. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.18 Hasil NS3 20 *node* kondisi terdapat deteksi black hole

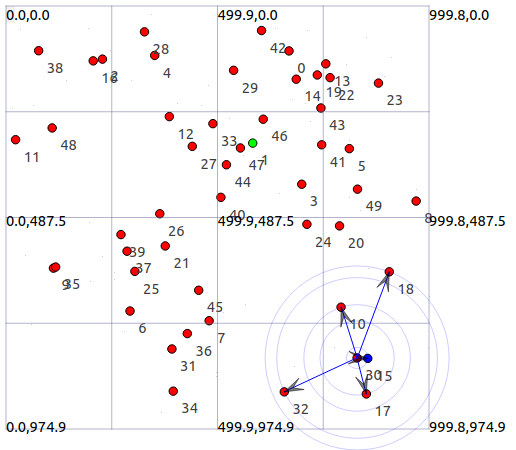
Gambar 5.18 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario posisi *black* *hole* yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.20). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.18 menunjukan hasil dengan nilai *packet* *loss* 6%, *packet* *delivery* *ratio* 94%, dan *delay* 32602274,44 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.20) selama proses simulasi dijalankan.

### Tipe Pergerakan

Implementasi variasi tipe pergerakan berdasarkan pada perancangan yang telah di definisikan. Tipe pergerakan yang digunakan adalah *pergerakan random waypoint, pergerakan random walk, dan pergerakan random direction*. Setiap skenario pengujian tipe pergerakan akan terdapat 3 jenis kondisi yaitu ketika jaringan normal, terdapat serangan *black hole*, dan terdapat deteksi serangan *black hole*.

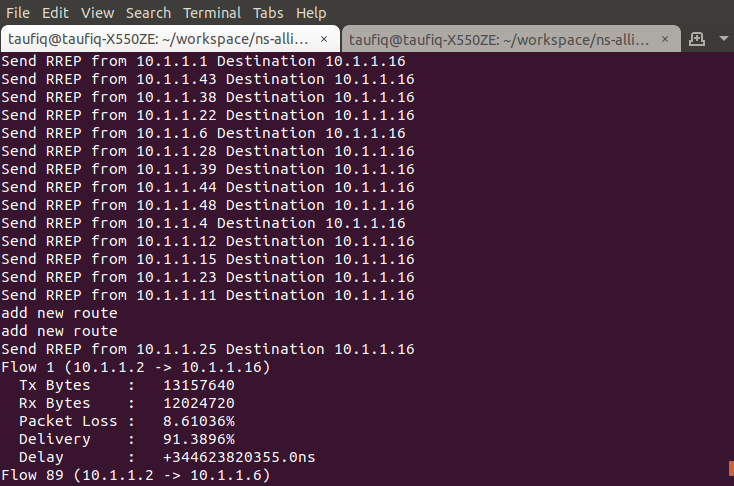
#### Implementasi dengan Tipe Pergerakan Random Waypoint

##### Simulasi Kondisi normal



Gambar 5.19 Tampilan NetAnim *random* *waypoint* kondisi normal

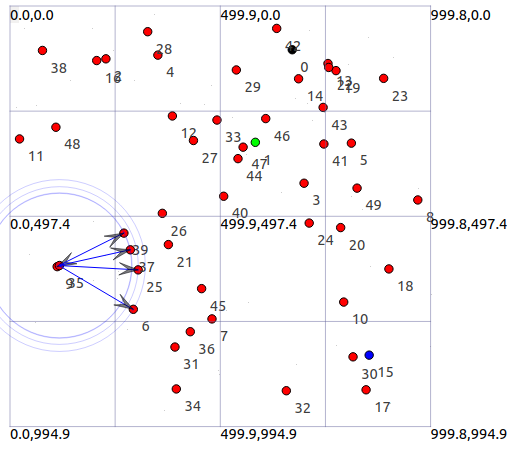
Gambar 5.19 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random waypoint*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.19 ada 3 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.20 Hasil NS3 *random* *waypoint* kondisi normal

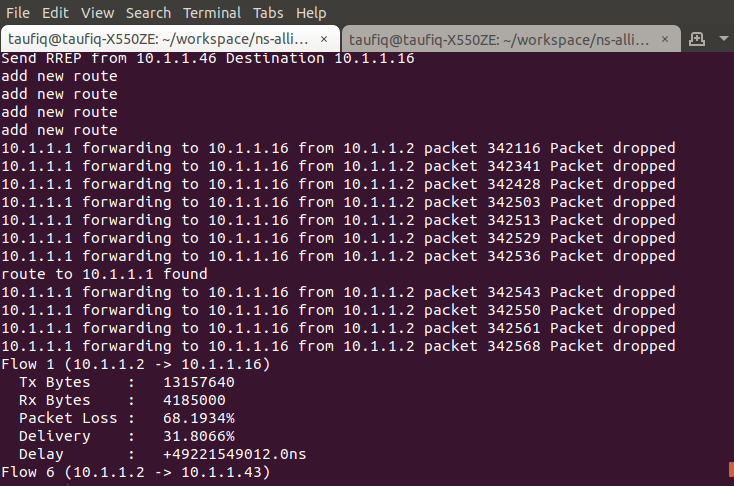
Gambar 5.20 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi normal menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi normal pada Gambar 5.20 menunjukan hasil dengan nilai *packet* *loss* 8,61036 %, *packet* *delivery* *ratio* 91,3896 %, dan *delay* 15476190,96 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat black hole



Gambar 5.21 Tampilan NetAnim *random* *waypoint* kondisi terdapat black hole

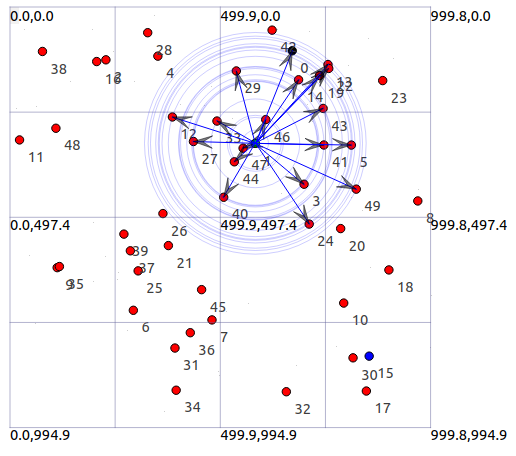
Gambar 5.21 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random waypoint*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.21 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.22 Hasil NS3 *random* *waypoint* kondisi terdapat black hole

Gambar 5.22 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.22 menunjukan hasil dengan nilai *packet* *loss* 68,1934 %, *packet* *delivery* *ratio* 31,8066 %, dan *delay* 6351167,614 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat deteksi black hole



Gambar 5.23 Tampilan NetAnim *random* *waypoint* kondisi terdapat deteksi black hole

Gambar 5.23 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random waypoint*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.23 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.

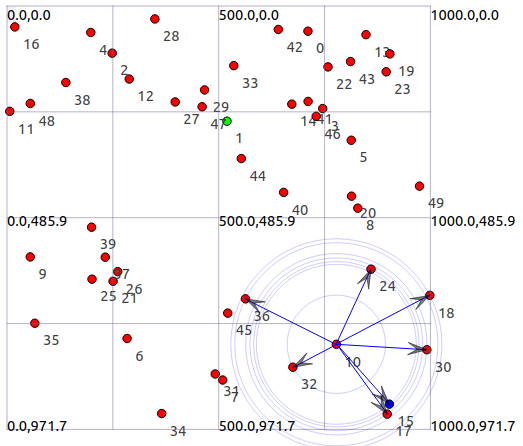


Gambar 5.24 Hasil NS3 *random* *waypoint* kondisi terdapat deteksi black hole

Gambar 5.24 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.24 menunjukan hasil dengan nilai *packet* *loss* 23,2004 %, *packet* *delivery* *ratio* 76,7996 %, dan *delay* 13056686,94 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

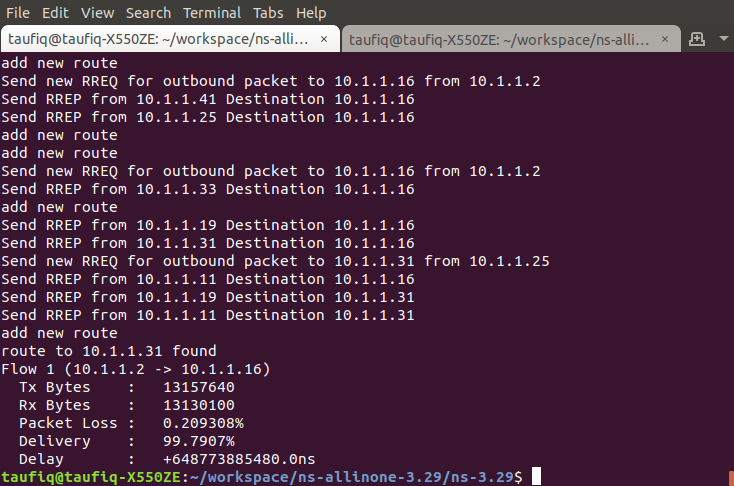
#### Implementasi dengan Tipe Pergerakan Random Walk

##### Simulasi Kondisi normal



Gambar 5.25 Tampilan NetAnim *random* *walk* kondisi normal

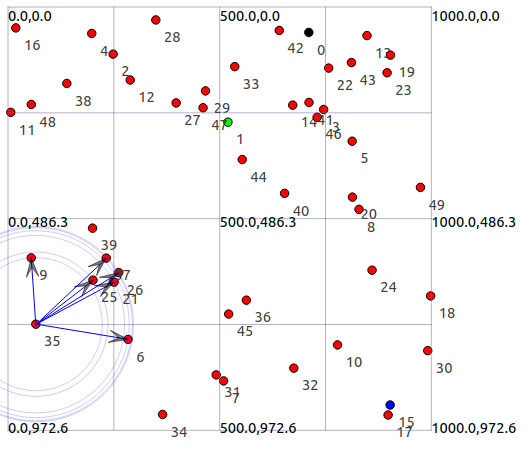
Gambar 5.25 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random walk*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.25 ada 3 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.26 Hasil NS *random* *walk* kondisi normal

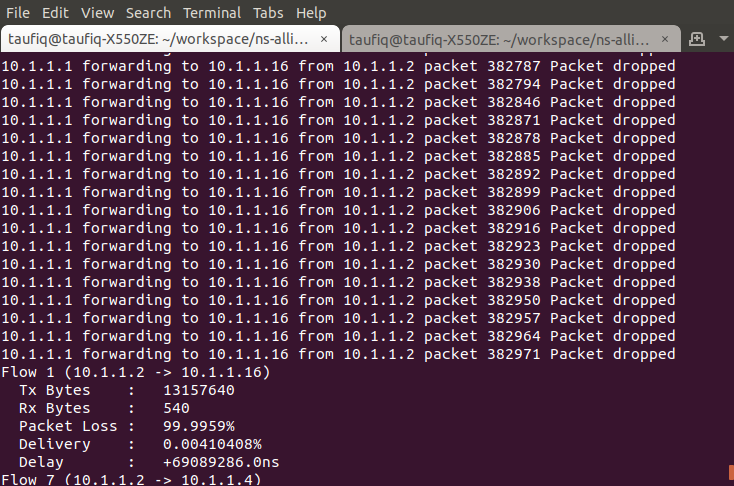
Gambar 5.26 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi normal menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi normal pada Gambar 5.26 menunjukan hasil dengan nilai *packet* *loss* 0,209308 %, *packet* *delivery* *ratio* 99,7907 %, dan *delay* 26682043,41 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat black hole



Gambar 5.27 Tampilan NetAnim *random* *walk* kondisi terdapat black hole

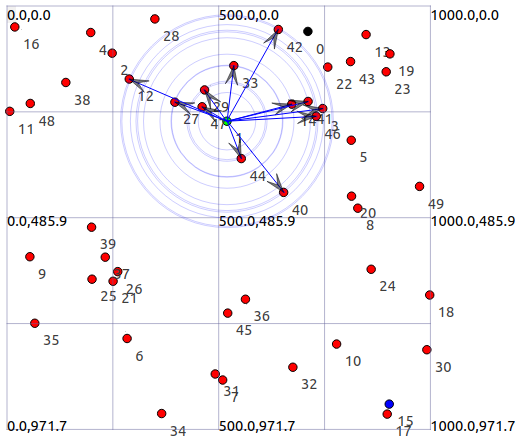
Gambar 5.27 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random walk*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.27 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.28 Hasil NS3 *random* *walk* kondisi terdapat black hole

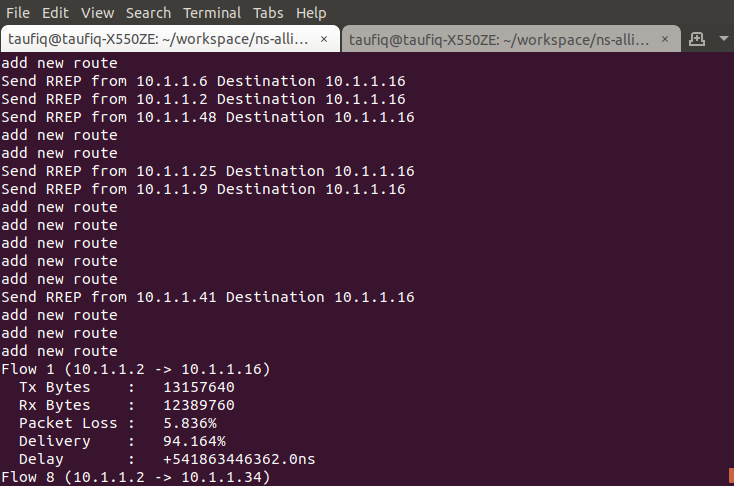
Gambar 5.28 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.28 menunjukan hasil dengan nilai *packet* *loss* 99,9959 %, *packet* *delivery* *ratio* 0,00410408 %, dan *delay* 69089286 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat deteksi black hole



Gambar 5.29 NetAnim *random* *walk* kondisi terdapat deteksi black hole

Gambar 5.29 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random walk*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.29 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.

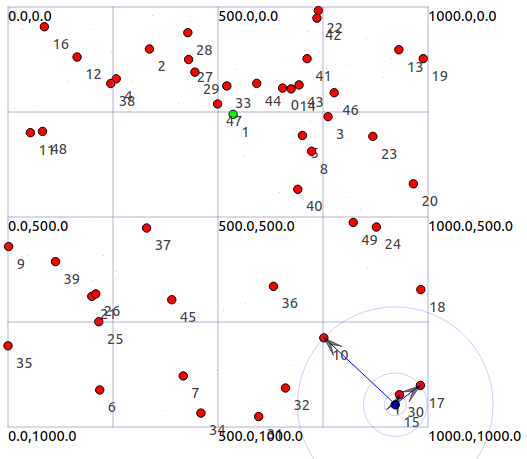


Gambar 5.30 Hasil NS3 *random* *walk* kondisi terdapat deteksi black hole

Gambar 5.30 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.30 menunjukan hasil dengan nilai *packet* *loss* 5,836 %, *packet* *delivery* *ratio* 94,164 %, dan *delay* 23616782,01 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

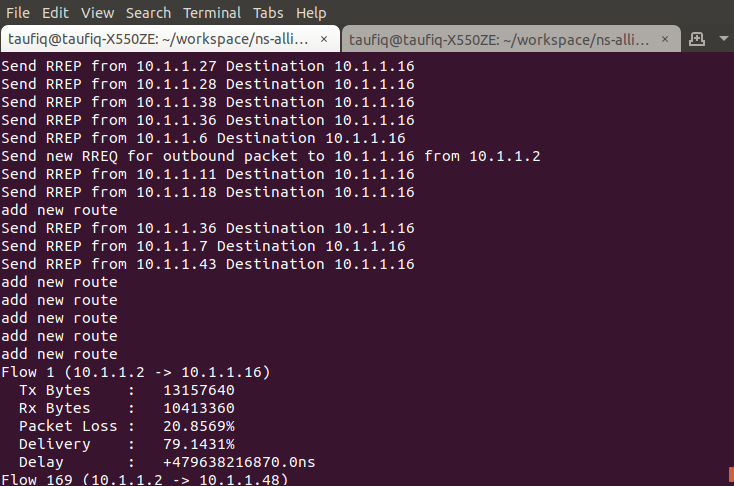
#### Implementasi dengan Tipe Pergerakan Random Direction

##### Simulasi Kondisi normal



Gambar 5.31 Tampilan NetAnim *random* *direction* kondisi normal

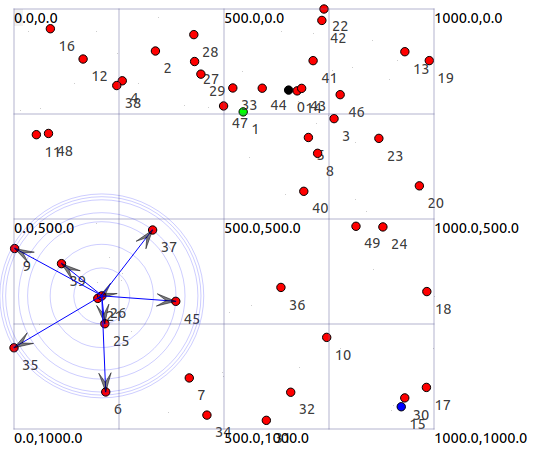
Gambar 5.31 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random direction*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.31 ada 3 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.32 Hasil NS3 *random* *direction* kondisi normal

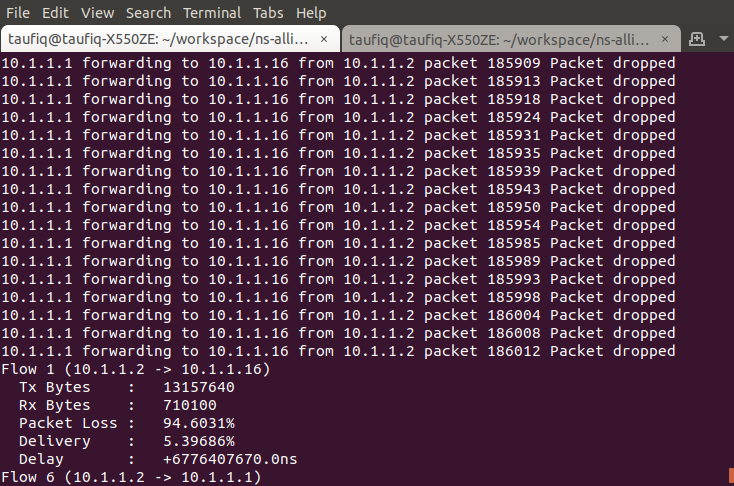
Gambar 5.32 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi normal menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi normal pada Gambar 5.32 menunjukan hasil dengan nilai *packet* *loss* 20,8869 %, *packet* *delivery* *ratio* 79,1431 %, dan *delay* 24872340,64 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat black hole



Gambar 5.33 Tampilan NetAnim *random* *direction* kondisi terdapat black hole

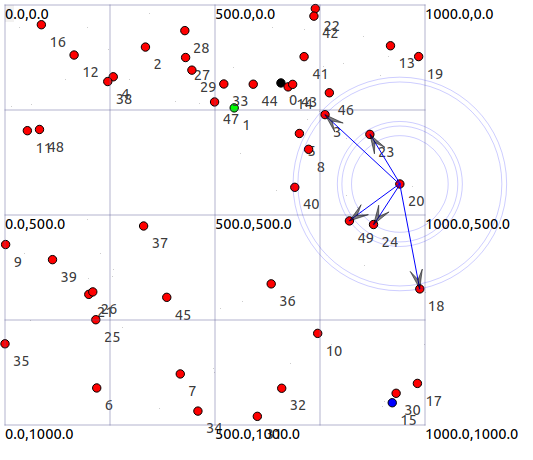
Gambar 5.33 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random direction*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.33 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.34 Hasil NS3 *random* *direction* kondisi terdapat black hole

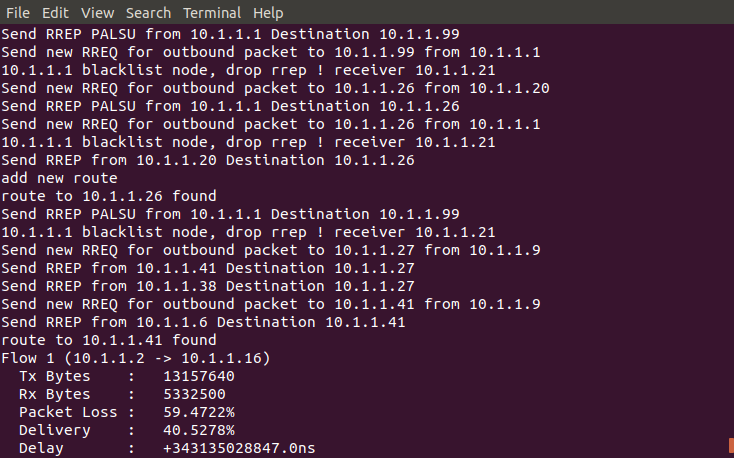
Gambar 5.34 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.34 menunjukan hasil dengan nilai *packet* *loss* 94,6031 %, *packet* *delivery* *ratio* 5,39686 %, dan *delay* 5153161,726 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat deteksi black hole



Gambar 5.35 Tampilan NetAnim *random* *direction* kondisi terdapat deteksi black hole

Gambar 5.35 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas *random direction*. *Node* menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.35 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna merah sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru sebagai node tujuan yang berperan untuk menerima data. *Node* berwarna hitam sebagai *black hole* yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui *node black hole*. Pada simulasi diatas *node* (1) berperan sebagai sumber dan *node* (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap *node* mencapai 250 meter. Apabila *node* tujuan berada diluar jangkauan transmisi *node* sumber maka data akan dikirimkan menggunakan perantara *node* tetangga.



Gambar 5.36 Hasil NS3 *random* *direction* dengan detksi serangan *black hole*

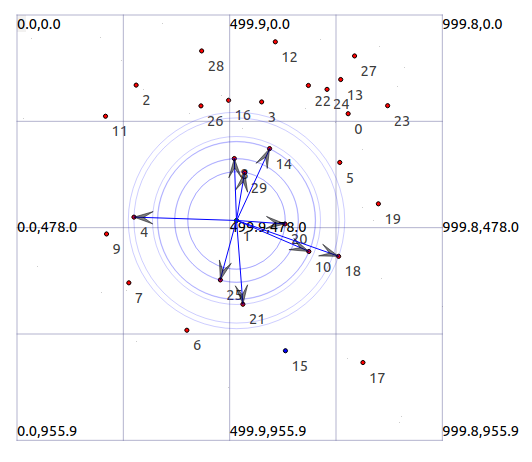
Gambar 5.36 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario tipe mobilitas yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.36 menunjukan hasil dengan nilai *packet* *loss* 59,47722 %, *packet* *delivery* *ratio* 40,5278 %, dan *delay* 34747851,02 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

### Jumlah *Node*

Implementasi variasi jumlah node berdasarkan pada perancangan yang telah di definisikan. Jumlah node yang digunakan adalah 30, 40, dan 50. Setiap skenario pengujian tipe pergerakan akan terdapat 3 jenis kondisi yaitu ketika jaringan normal, terdapat serangan *black hole*, dan terdapat deteksi serangan *black hole*.

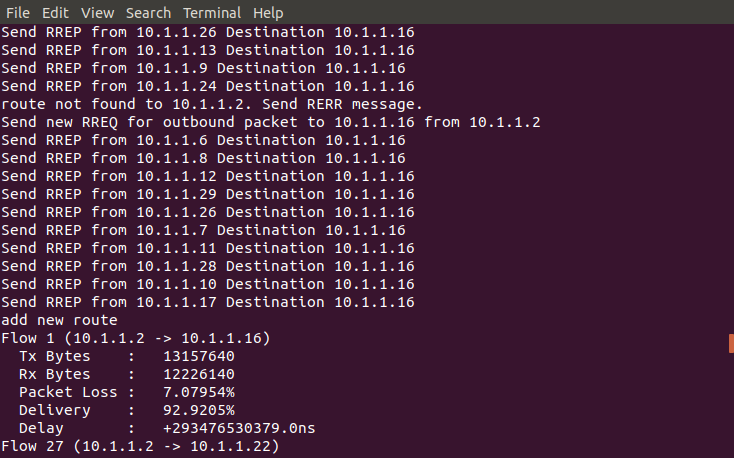
#### Implementasi dengan 30 Node

##### Simulasi Kondisi normal



Gambar 5.37 Tampilan NetAnim 30 *node* kondisi normal

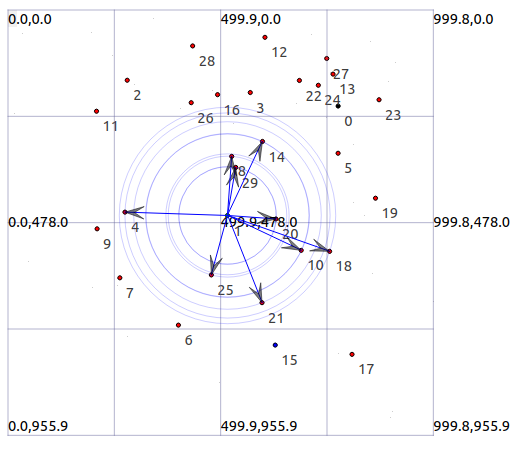
Gambar 5.37 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 30. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.37 ada 3 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.



Gambar 5.38 Hasil NS3 30 *node* kondisi normal

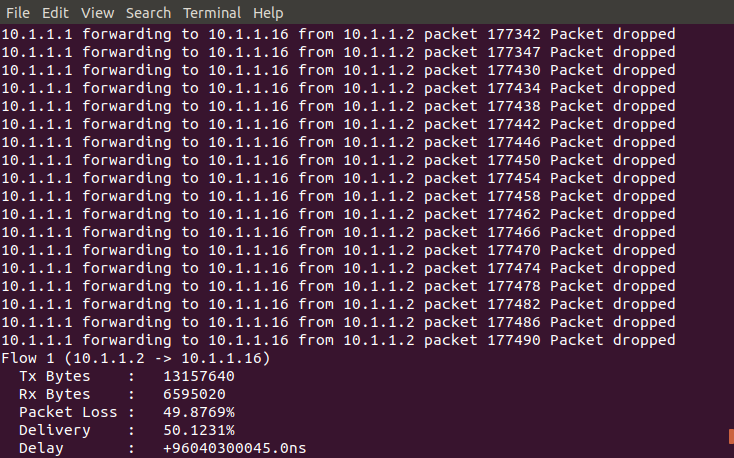
Gambar 5.38 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi normal menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh node sumber (10.1.1.2) kepada node tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi normal pada Gambar 5.38 menunjukan hasil dengan nilai packet loss 7,0954 %, packet delivery ratio 92,9205 %, dan delay 12962171,74 ns yang dihitung dari aliran data antara node sumber (10.1.1.2) dengan node tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat black hole



Gambar 5.39 Tampilan NetAnim 30 *node* kondisi terdapat black hole

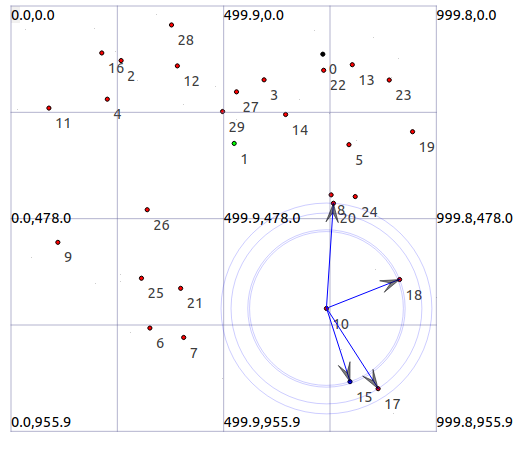
Gambar 5.39 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 30. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.39 ada 4 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Node berwarna hitam sebagai black hole yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui node black hole. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.



Gambar 5.40 Hasil NS3 30 *node* kondisi terdapat black hole

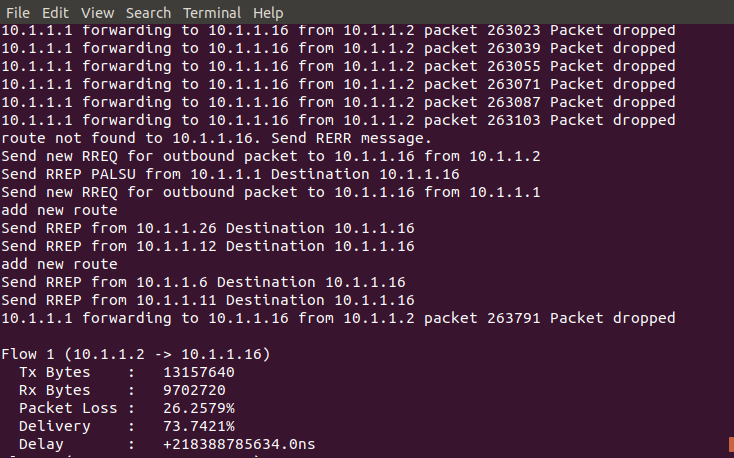
Gambar 5.40 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.40 menunjukan hasil dengan nilai *packet* *loss* 49,8769 %, *packet* *delivery* *ratio* 50,1231 %, dan *delay* 7863776,308 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat deteksi black hole



Gambar 5.41 Tampilan NetAnim 30 *node* kondisi terdapat deteksi black hole

Gambar 5.41 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 30. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.41 ada 4 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Node berwarna hitam sebagai black hole yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui node black hole. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.

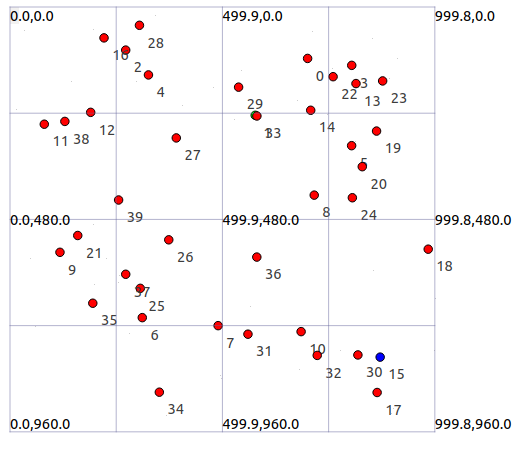


Gambar 5.42 Hasil NS3 30 *node* kondisi terdapat deteksi black hole

Gambar 5.42 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat deteksi *black hole* pada Gambar 5.42 menunjukan hasil dengan nilai *packet* *loss* 26,2579 %, *packet* *delivery* *ratio* 73,7421 %, dan *delay* 12154317,99 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

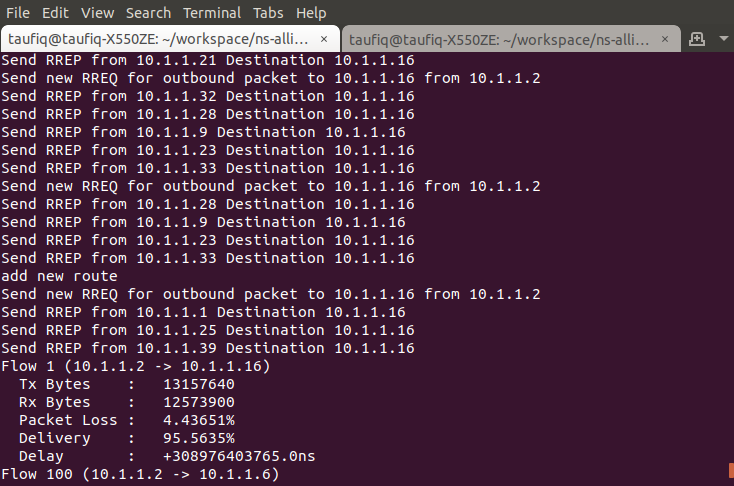
#### Implementasi dengan 40 Node

##### Simulasi Kondisi normal



Gambar 5.43 Tampilan NetAnim 40 *node* kondisi normal

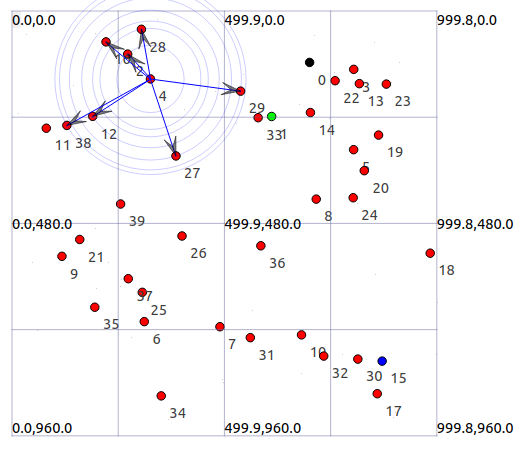
Gambar 5.43 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 40. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.43 ada 3 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.



Gambar 5.44 Hasil NS3 40 *node* kondisi normal

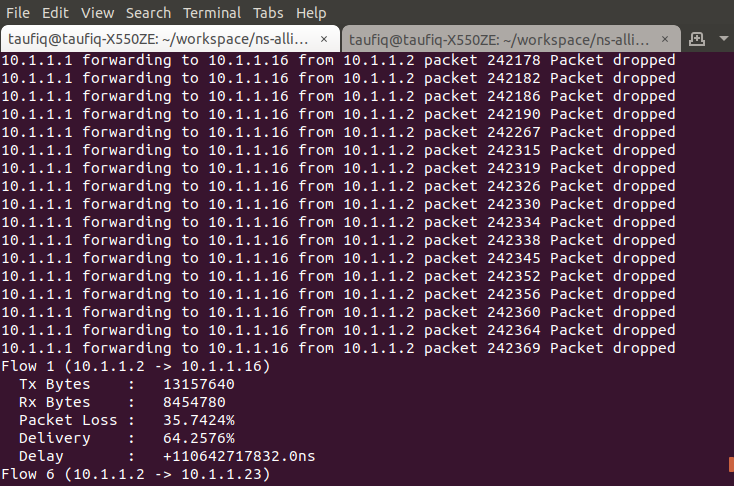
Gambar 5.44 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi normal menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh node sumber (10.1.1.2) kepada node tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi normal pada Gambar 5.44 menunjukan hasil dengan nilai packet loss 4,43651 %, packet delivery ratio 95,5635 %, dan delay 13269332,35 ns yang dihitung dari aliran data antara node sumber (10.1.1.2) dengan node tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat black hole



Gambar 5.45 Tampilan NetAnim 40 *node* kondisi terdapat black hole

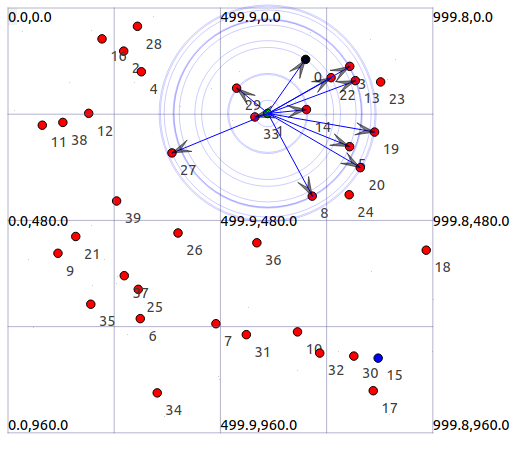
Gambar 5.45 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 40. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.45 ada 4 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Node berwarna hitam sebagai black hole yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui node black hole. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.



Gambar 5.46 Hasil NS3 40 *node* kondisi terdapat black hole

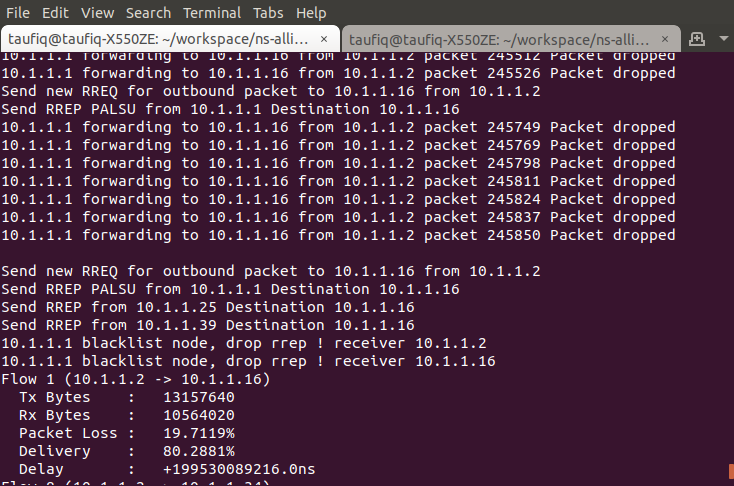
Gambar 5.46 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.46 menunjukan hasil dengan nilai *packet* *loss* 35,7424 %, *packet* *delivery* *ratio* 64,2576 %, dan *delay* 7066661,419 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat deteksi black hole



Gambar 5.47 Tampilan NetAnim 40 *node* kondisi terdapat deteksi black hole

Gambar 5.47 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 40. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.47 ada 4 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Node berwarna hitam sebagai black hole yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui node black hole. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.

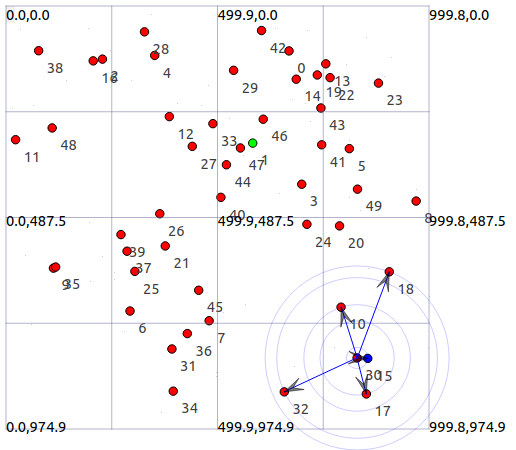


Gambar 5.48 Hasil NS3 40 *node* kondisi terdapat deteksi black hole

Gambar 5.48 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi black hole pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh node sumber (10.1.1.2) kepada node tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat deteksi black hole pada Gambar 5.48 menunjukan hasil dengan nilai packet loss 19,7119 %, packet delivery ratio 80,2881 %, dan delay 10199360,49 ns yang dihitung dari aliran data antara node sumber (10.1.1.2) dengan node tujuan (10.1.1.16) selama proses simulasi dijalankan.

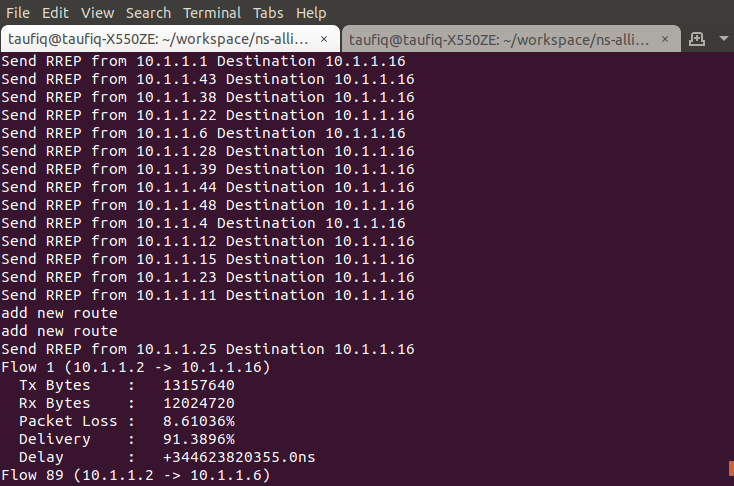
#### Implementasi dengan 50 Node

##### Simulasi Kondisi normal



Gambar 5.49 Tampilan NetAnim 50 *node* kondisi normal

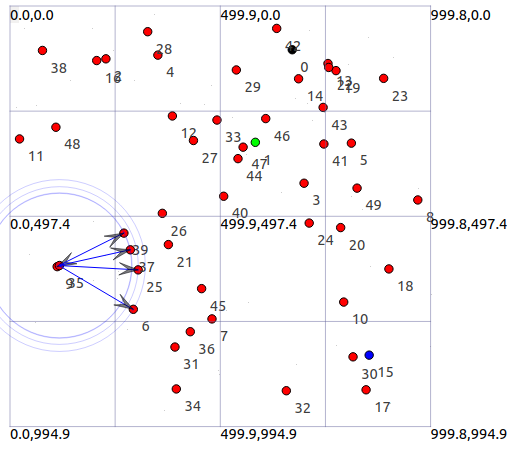
Gambar 5.49 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 50. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.49 ada 3 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.



Gambar 5.50 Hasil NS3 50 *node* kondisi normal

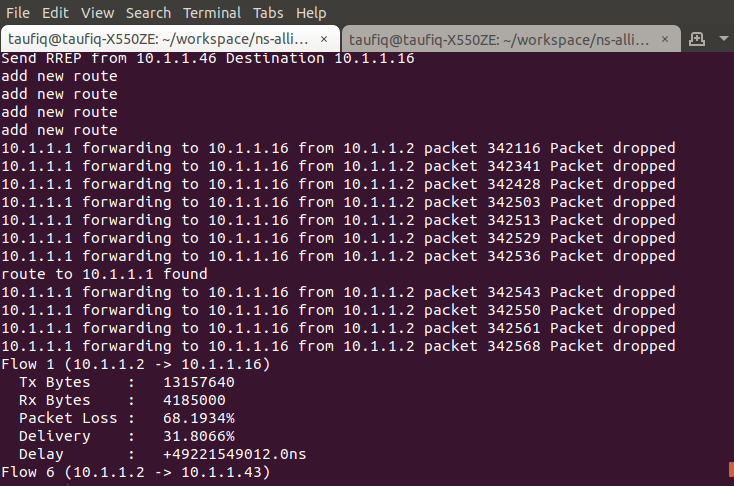
Gambar 5.50 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi normal menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh node sumber (10.1.1.2) kepada node tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi normal pada Gambar 5.50 menunjukan hasil dengan nilai packet loss 8,61036 %, packet delivery ratio 91,3896 %, dan delay 15476190,96 ns yang dihitung dari aliran data antara node sumber (10.1.1.2) dengan node tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat black hole



Gambar 5.51 Tampilan NetAnim 50 *node* kondisi terdapat black hole

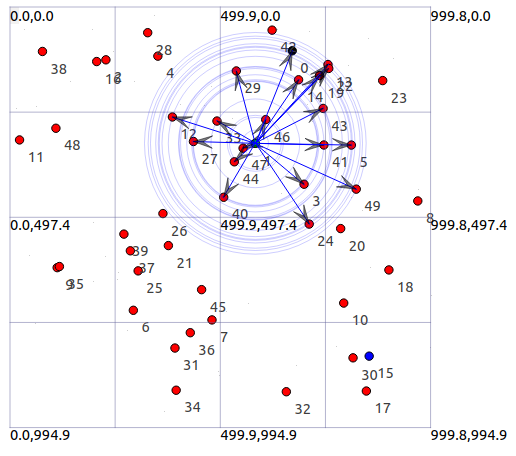
Gambar 5.51 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 50. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.51 ada 4 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Node berwarna hitam sebagai black hole yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui node black hole. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.



Gambar 5.52 Hasil NS3 50 *node* kondisi terdapat black hole

Gambar 5.52 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat *black hole* pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh *node* sumber (10.1.1.2) kepada *node* tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat *black hole* pada Gambar 5.52 menunjukan hasil dengan nilai *packet* *loss* 68,1934 %, *packet* *delivery* *ratio* 31,8066 %, dan *delay* 6351167,614 ns yang dihitung dari aliran data antara *node* sumber (10.1.1.2) dengan *node* tujuan (10.1.1.16) selama proses simulasi dijalankan.

##### Simulasi Kondisi terdapat deteksi black hole



Gambar 5.53 Tampilan NetAnim 50 *node* kondisi terdapat deteksi black hole

Gambar 5.53 merupakan tampilan dari simulasi jaringan MANET dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan node yang berjumlah 50. Posisi setiap node ditempatkan dengan acak pada area seluas 1000 meter x 1000 meter dan menggunakan tipe mobilitas random waypoint. Node menempati posisi acak dan dapat bergerak secara bebas di dalam area. Pada Gambar 5.53 ada 4 macam warna node dan memiliki peran yang berbeda. Node dengan warna merah sebagai node tetangga yang digunakan sebagai perantara pengiriman data. Node berwarna hijau sebagai node sumber yang berperan untuk melakukan pengiriman data. Node berwarna biru sebagai node tujuan yang berperan untuk menerima data. Node berwarna hitam sebagai black hole yang akan melakukan pembuangan data setiap terdapat data yang akan dikirimkan melalui node black hole. Pada simulasi diatas node (1) berperan sebagai sumber dan node (15) berperan sebagai tujuan. Jangkauan transmisi pada setiap node mencapai 250 meter. Apabila node tujuan berada diluar jangkauan transmisi node sumber maka data akan dikirimkan menggunakan perantara node tetangga.



Gambar 5.54 Hasil NS3 50 *node* kondisi terdapat deteksi black hole

Gambar 5.54 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan MANET dalam kondisi terdapat deteksi black hole pada jaringan menggunakan aplikasi NS3. Simulasi dilakukan dengan skenario jumlah node yang berbeda. Pada tampilan terminal terdapat informasi aliran paket data yang dikirimkan oleh node sumber (10.1.1.2) kepada node tujuan (10.1.1.16). Skenario jaringan MANET dalam kondisi terdapat deteksi black hole pada Gambar 5.54 menunjukan hasil dengan nilai packet loss 23,2004 %, packet delivery ratio 76,7996 %, dan delay 13056686,94 ns yang dihitung dari aliran data antara node sumber (10.1.1.2) dengan node tujuan (10.1.1.16) selama proses simulasi dijalankan.

# HASIL PENGUJIAN DAN PEMBAHASAN

## Kinerja Protokol Terhadap Posisi Black Hole

### Hasil Pengujian

Tabel 6.1 menampilkan data hasil pengujian simulasi jaringan MANET. Data didapatkan dari 9 simulasi jaringan MANET menggunakan aplikasi NS 3. Simulasi dilakukan dalam 3 kondisi yang berbeda. Kondisi normal ketika tidak terdapat *black hole* pada simulasi jaringan MANET. Kondisi black hole ketika terdapat *node black hole* pada simulasi jaringan MANET. Kondisi deteksi ketika terdapat *black hole* dan mekanisme pendeteksian *black hole* pada jaringan MANET. Skenario pengujian yang dilakukan dengan mengubah posisi *node black hole* pada setiap simulasi. Posisi black hole berada pada salah satu *node* (0), *node* (3), *node* (8), dan *node* (13). Pada Tabel 6.1 menunjukan hasil pengujian dengan 3 parameter *Quality of Service* (QoS) yaitu nilai *packet loss* dengan satuan persen (%), *packet delivery ratio* (PDR) dengan satuan (%), dan *delay* dengan satuan *nano seconds* (ns).

Tabel 6.1 Hasil pengujian variasi posisi black hole

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Posisi Black Hole** | **Simulasi** | **Paket Loss (%)** | **PDR (%)** | **Average Delay (ns)** |
| 1 | Normal | Normal | 1 | 99 | 31300156,03 |
| 2 | Node 0 | Black Hole | 100 | 0 | 0 |
| 3 | Node 0 | Deteksi | 6 | 94 | 32095780,15 |
| 4 | Node 3 | Black Hole | 100 | 0 | 0 |
| 5 | Node 3 | Deteksi | 7 | 93 | 23557997,35 |
| 6 | Node 8 | Black Hole | 100 | 0 | 0 |
| 7 | Node 8 | Deteksi | 30 | 70 | 71811684,61 |
| 8 | Node 13 | Black Hole | 100 | 0 | 0 |
| 9 | Node 13 | Deteksi | 6 | 94 | 32602274,44 |

### Pembahasan

#### Packet Loss

Dari Tabel 6.1, data diolah menjadi grafik yang menunjukkan nilai *packet loss* dari pengujian protokol terhadap variasi posisi black hole.

Gambar 6.1 Grafik packet loss variasi posisi black hole

Gambar 6.1 merupakan grafik nilai *packet loss* hasil pengujian simulasi jaringan MANET dengan skenario penempatan *node black hole* yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai *packet loss* sebesar 5 %. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai *packet loss* mengalami kenaikan pada keempat skenario posisi *black hole* yang berbeda menjadi 100 %. Kenaikan *packet loss* disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menurunkan nilai *packet loss* pada keempat skenario posisi *black hole* yang berbeda. Penurunan *packet loss* disebabkan karena mekanisme pendeteksian dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. Pada simulasi dengan posisi *black hole* berada pada *node* (0) dan *node* (13) menunjukan nilai *packet loss* menjadi 6 %. Simulasi lain dengan posisi *black hole* berada pada *node* (8) menunjukan nilai *packet loss* menjadi 30 %.

#### Packet delivery ratio

Dari Tabel 6.1, data diolah menjadi grafik yang menunjukkan nilai *packet delivery ratio* dari pengujian protokol terhadap variasi posisi.

Gambar 6.2 Grafik *packet delivery ratio* variasi posisi black hole

Gambar 6.2 merupakan grafik nilai *packet delivery ratio* (PDR) hasil pengujian simulasi jaringan MANET dengan skenario penempatan *node black hole* yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai PDR sebesar 95 %. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai PDR mengalami penurunan pada keempat skenario posisi *black hole* yang berbeda menjadi 0 %. Penurunan PDR disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menaikan nilai PDR pada keempat skenario posisi *black hole* yang berbeda. Kenaikan PDR disebabkan karena mekanisme pendeteksian dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. Pada simulasi dengan posisi *black hole* berada pada *node* (0) dan *node* (13) menunjukan nilai PDR menjadi 94 %. Simulasi lain dengan posisi *black hole* berada pada *node* (8) menunjukan nilai PDR menjadi 70 %.

#### Delay

Dari Tabel 6.1, data diolah menjadi grafik yang menunjukkan nilai *delay* dari pengujian protokol terhadap variasi posisi black hole.

Gambar 6.3 Grafik *delay* variasi posisi black hole

Gambar 6.3 merupakan grafik nilai *delay* hasil pengujian simulasi jaringan MANET dengan skenario penempatan *node black hole* yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai *delay* sebesar 31300156,03 ns. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai *delay* mengalami penurunan pada keempat skenario posisi *black hole* yang berbeda menjadi 0 ns. Penurunan *delay* disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menaikan nilai *delay* pada beberapa skenario posisi *black hole* yang berbeda. Kenaikan *delay* disebabkan karena mekanisme pendeteksian dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga diperlukan tambahan waktu untuk mencari posisi *black hole* dan mencari rute lain untuk menghindari *black hole*. Pada simulasi dengan posisi *black hole* berada pada *node* (8) menunjukan nilai *delay* menjadi 71811684,61 ns . Simulasi lain dengan posisi *black hole* berada pada *node* (13) menunjukan nilai *delay* menjadi 32602274,44 ns.

#### Akurasi Deteksi

Tabel 6.2 Hasil Akurasi deteksi variasi posisi black hole

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Aktivitas** | **Posisi Black Hole** | **Hasil Pendeteksian** | **Hasil** | **Paket Loss** |
| 1 | Serangan | Node 0 | Terdeteksi | Benar | 6 |
| 2 | Serangan | Node 3 | Terdeteksi | Benar | 7 |
| 3 | Serangan | Node 8 | Terdeteksi | Benar | 30 |
| 4 | Serangan | Node 13 | Terdeteksi | Benar | 6 |

Tabel 6.2 merupakan grafik akurasi deteksi hasil pengujian simulasi jaringan MANET dengan skenario penempatan *node black hole* yang berbeda pada setiap simulasi. Pada grafik terdapat 4 posisi black hole berbeda yang menunjukkan posisi *black hole* pada jaringan MANET ketika simulasi dijalankan. Skenario dengan posisi *black hole* pada *node* 0, 3, dan 13 dapat dilakukan pendeteksian dan dapat dilihat nilai paket loss bernilai 6 dan 7 menunjukkan *black hole* dapat segera terdeteksi. Pada skenario posisi *black hole* pada *node* 8 dapat dilakukan pendeteksian dan memiliki nilai *paket loss* lebih tinggi dari ketiga skenario lain. Hasil akurasi deteksi pada keempat skenario posisi *black hole* berbeda dapat mengetahui posisi *black hole*. Pada hasil pengujian deteksi serangan, metode pendeteksian dengan pesan RREQ palsu dapat mendeteksi total 4 dari 4 skenario upaya serangan dengan posisi *black hole* yang berbeda. Berdasarkan perhitungan tingkat akurasi pendeteksian serangan *black hole* memperoleh presentase akurasi 100 %.

## Kinerja Protokol Terhadap Variasi Tipe Pergerakan

### Hasil Pengujian

Tabel 6.1 menampilkan data hasil pengujian simulasi jaringan MANET. Data didapatkan dari 9 simulasi jaringan MANET menggunakan aplikasi NS 3. Simulasi dilakukan dalam 3 kondisi yang berbeda. Kondisi normal ketika tidak terdapat *black hole* pada simulasi jaringan MANET. Kondisi black hole ketika terdapat *node black hole* pada simulasi jaringan MANET. Kondisi deteksi ketika terdapat *black hole* dan mekanisme pendeteksian *black hole* pada jaringan MANET. Skenario pengujian yang dilakukan dengan mengubah tipe mobilitas pada setiap simulasi. Tipe mobilitas yang digunakan dalam simulasi adalah mobilitas random waypoint, mobilitas random walk, dan mobilitas random direction. Pada Tabel 6.1 menunjukan hasil pengujian dengan 3 parameter *Quality of Service* (QoS) yaitu nilai *packet loss* dengan satuan persen (%), *packet delivery ratio* (PDR) dengan satuan (%), dan *delay* dengan satuan *nano seconds* (ns).

Tabel 6.2 Hasil pengujian variasi tipe pergerakan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Tipe Pergerakan** | **Simulasi** | **Paket Loss (%)** | **PDR (%)** | **Average Delay (ns)** |
| 1 | Random Waypoint | Normal | 8,61036 | 91,3896 | 15476190,96 |
| 2 | Random Walk | Normal | 0,209308 | 99,7907 | 26682043,41 |
| 3 | Random Direction | Normal | 20,8869 | 79,1431 | 24872340,64 |
| 4 | Random Waypoint | Black Hole | 68,1934 | 31,8066 | 6351167,614 |
| 5 | Random Walk | Black Hole | 99,9959 | 0,004104 | 6908928 |
| 6 | Random Direction | Black Hole | 94,6031 | 5,39686 | 5153161,726 |
| 7 | Random Waypoint | Deteksi | 23,2004 | 76,7996 | 13056686,94 |
| 8 | Random Walk | Deteksi | 5,836 | 94,164 | 23616782,01 |
| 9 | Random Direction | Deteksi | 59,47722 | 40,5278 | 34747851,02 |

### Pembahasan

#### Packet Loss

Dari Tabel 6.2, data diolah menjadi grafik yang menunjukkan nilai *packet loss* dari pengujian protokol terhadap variasi tipe pergerakan.

Gambar 6.4 Grafik packet loss variasi tipe pergerakan

Gambar 6.4 merupakan grafik nilai *packet loss* hasil pengujian simulasi jaringan MANET dengan skenario penempatan *node black hole* yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai *packet loss* sebesar 8,61036 % pada mobilitas *random waypoint*, 0,209308 % pada mobilitas *random walk*, dan 20,8869 % pada mobilitas *random walk*. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai *packet loss* mengalami kenaikan pada ketiga skenario tipe mobilitas yang berbeda menjadi 68,1934 % pada mobilitas random waypoint, 99,9959 % pada mobilitas random walk, dan 94,6031 % pada mobilitas random walk. Kenaikan *packet loss* disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menurunkan nilai *packet loss* pada keempat skenario posisi *black hole* yang berbeda. Penurunan *packet loss* disebabkan karena mekanisme pendeteksian dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. Pada simulasi dengan tipe mobilitas *random walk* menunjukan nilai *packet loss* menjadi 5,836 %. Simulasi lain tipe mobilitas *random waypoint* menunjukan nilai *packet loss* menjadi 23,2004 %.

#### Packet delivery ratio

Dari Tabel 6.2, data diolah menjadi grafik yang menunjukkan nilai *packet delivery ratio* dari pengujian protokol terhadap variasi tipe pergerakan.

Gambar 6.5 Grafik *packet delivery ratio* variasi tipe pergerakan

Gambar 6.5 merupakan grafik nilai *packet delivery ratio* (PDR) hasil pengujian simulasi jaringan MANET dengan skenario tipe mobilitas yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai PDR 91,3896 % pada mobilitas *random waypoint*, 99,7907 % pada mobilitas *random walk*, dan 79,1431 % pada mobilitas *random walk*. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai PDR mengalami penurunan pada ketiga skenario tipe mobilitas yang berbeda menjadi 31,8066 % pada mobilitas *random waypoint*, 0,00410408 % pada mobilitas *random walk*, dan 5,39686 % pada mobilitas *random walk*.. Penurunan PDR disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menaikan nilai PDR pada ketiga skenario tipe mobilitas yang berbeda. Kenaikan PDR disebabkan karena mekanisme pendeteksian dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. Pada simulasi dengan tipe mobilitas random walk menunjukan nilai PDR menjadi 94,164 %. Simulasi lain dengan tipe mobilitas random waypoint menunjukan nilai PDR menjadi 76,7996 %.

#### Delay

Dari Tabel 6.2, data diolah menjadi grafik yang menunjukkan nilai *delay* dari pengujian protokol terhadap variasi tipe pergerakan.

Gambar 6.6 Grafik *delay* variasi tipe pergerakan

Gambar 6.6 merupakan grafik nilai *delay* hasil pengujian simulasi jaringan MANET dengan skenario tipe mobilitas yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai *delay* sebesar 15476190,96 ns pada mobilitas *random waypoint*, 26682043,41 ns pada mobilitas *random walk*, dan 24872340,64 ns pada mobilitas *random walk*. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai *delay* mengalami penurunan pada ketiga skenario tipe mobilitas yang berbeda menjadi 6351167,614 ns pada mobilitas *random waypoint*, 6908928 ns pada mobilitas *random walk*, dan 5153161,726 ns pada mobilitas *random walk*. Penurunan *delay* disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menurunkan nilai *delay* pada ketiga skenario tipe mobilitas yang berbeda. penurunan *delay* disebabkan karena mekanisme pendeteksian menghindari rute *black hole* dalam pengiriman data, ketika proses mencari posisi *black hole* dan mencari rute lain untuk menghindari *black hole* membutuhkan waktu tambahan membuat beberapa data menjadi gagal terkirim atau melewati *black hole*. Pada simulasi dengan pendeteksian *black hole* menunjukan nilai *delay* menjadi 13056686,94 ns pada mobilitas *random waypoint*, 23616782,01 ns pada mobilitas *random walk*, dan 34747851,02 ns pada mobilitas *random walk*.

## Kinerja Protokol Terhadap Variasi Jumlah *Node*

### Hasil Pengujian

Tabel 6.1 menampilkan data hasil pengujian simulasi jaringan MANET. Data didapatkan dari 9 simulasi jaringan MANET menggunakan aplikasi NS 3. Simulasi dilakukan dalam 3 kondisi yang berbeda. Kondisi normal ketika tidak terdapat *black hole* pada simulasi jaringan MANET. Kondisi black hole ketika terdapat *node black hole* pada simulasi jaringan MANET. Kondisi deteksi ketika terdapat *black hole* dan mekanisme pendeteksian *black hole* pada jaringan MANET. Skenario pengujian yang dilakukan setiap simulasi dengan mengubah jumlah *node* pada jaringan MANET. Jumlah node yang digunakan pada simulasi berjumlah 30, 40, dan 50. Pada Tabel 6.1 menunjukan hasil pengujian dengan 3 parameter *Quality of Service* (QoS) yaitu nilai *packet loss* dengan satuan persen (%), *packet delivery ratio* (PDR) dengan satuan (%), dan *delay* dengan satuan *nano seconds* (ns).

Tabel 6.3 Hasil pengujian variasi jumlah *node*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Jumlah Node** | **Simulasi** | **Paket Loss (%)** | **PDR (%)** | **Average Delay (ns)** |
| 1 | 30 Node | Normal | 7,0954 | 92,9205 | 12962171,74 |
| 2 | 40 Node | Normal | 4,43651 | 95,5635 | 13269332,35 |
| 3 | 50 Node | Normal | 8,61036 | 91,3896 | 15476190,96 |
| 4 | 30 Node | Black Hole | 49,8769 | 50,1231 | 7863776,308 |
| 5 | 40 Node | Black Hole | 35,7424 | 64,2576 | 7066661,419 |
| 6 | 50 Node | Black Hole | 68,1934 | 31,8066 | 6351167,614 |
| 7 | 30 Node | Deteksi | 26,2579 | 73,7421 | 12154317,99 |
| 8 | 40 Node | Deteksi | 19,7119 | 80,2881 | 10199360,49 |
| 9 | 50 Node | Deteksi | 23,2004 | 76,7996 | 13056686,94 |

### Pembahasan

#### Packet Loss

Dari Tabel 6.3, data diolah menjadi grafik yang menunjukkan nilai packet loss dari pengujian protokol terhadap variasi jumlah *node*.

Gambar 6.7 Grafik packet loss variasi jumlah *node*

Gambar 6.7 merupakan grafik nilai *packet loss* hasil pengujian simulasi jaringan MANET dengan skenario jumlah *node* yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai *packet loss* sebesar 7,0954 % pada simulasi dengan jumlah 30 *node*, 4,43651 % pada simulasi dengan jumlah 40 *node*, dan 8,61036 % pada simulasi dengan jumlah 50 *node*. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai *packet loss* mengalami kenaikan pada ketiga skenario jumlah *node* yang berbeda menjadi 49,8769 % pada simulasi dengan jumlah *30* *node*, 35,7424 % pada simulasi dengan jumlah 40 *node*, dan 68,1934 % pada simulasi dengan jumlah 50 *node*. Kenaikan *packet loss* disebabkan karena adanya *black hole* pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menurunkan nilai *packet loss* pada ketiga skenario jumlah *node* yang berbeda. Penurunan *packet loss* disebabkan karena mekanisme pendeteksian dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. *Packet loss* mengalami penurunan daripada kondisi *black hole* menjadi 26,2579 % pada simulasi dengan jumlah 30 *node*, 19,7119 % pada simulasi dengan jumlah 40 *node*, dan 23,2004 % pada simulasi dengan jumlah 50 *node*.

#### Packet delivery ratio

Dari Tabel 6.3, data diolah menjadi grafik yang menunjukkan nilai *packet delivery ratio* dari pengujian protokol terhadap variasi jumlah *node*.

Gambar 6.8 Grafik *packet delivery ratio* variasi jumlah *node*

Gambar 6.8 merupakan grafik nilai *Packet delivery ratio* (PDR) hasil pengujian simulasi jaringan MANET dengan skenario jumlah *node* yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai PDR sebesar 92,9205 % pada simulasi dengan jumlah 30 *node*, 95,5635 % pada simulasi dengan jumlah 40 *node*, dan 91,3896 % pada simulasi dengan jumlah 50 *node*. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai PDR mengalami penurunan pada ketiga skenario jumlah *node* yang berbeda menjadi 50,1231 % pada simulasi dengan jumlah *30* *node*, 64,2576 % pada simulasi dengan jumlah 40 *node*, dan 31,8066 % pada simulasi dengan jumlah 50 *node*. Penurunan PDR disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menaikan nilai PDR pada ketiga skenario jumlah node yang berbeda. Kenaikan PDR disebabkan karena mekanisme pendeteksian dapat mengetahui posisi *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. PDR mengalami kenaikan daripada kondisi *black hole* menjadi 73,7421 % pada simulasi dengan jumlah 30 *node*, 80,2881 % pada simulasi dengan jumlah 40 *node*, dan 76,7996 % pada simulasi dengan jumlah 50 *node*.

#### Delay

Dari Tabel 6.3, data diolah menjadi grafik yang menunjukkan nilai *delay* dari pengujian protokol terhadap variasi jumlah *node*.

Gambar 6.9 Grafik *delay* variasi jumlah *node*

Gambar 6.9 merupakan grafik nilai *delay* hasil pengujian simulasi jaringan MANET dengan skenario jumlah node yang berbeda pada setiap simulasi. Pada grafik terdapat 3 warna berbeda yang menunjukan kondisi jaringan MANET ketika simulasi dijalankan. Warna biru menunjukan kondisi normal, warna merah menunjukan kondisi *black hole*, dan warna hijau menunjukan kondisi deteksi. Simulasi jaringan MANET dalam kondisi normal menunjukan nilai *delay* sebesar 12962171,74 ns pada simulasi dengan jumlah 30 *node*, 13269332,35 ns pada simulasi dengan jumlah 40 *node*, dan 15476190,96 ns pada simulasi dengan jumlah 50 *node*. Ketika pada simulasi jaringan MANET terdapat *black hole* membuat nilai *delay* mengalami penurunan pada ketiga skenario jumlah *node* yang berbeda menjadi sebesar 7863776,308 ns pada simulasi dengan jumlah 30 *node*, 7066661,419 ns pada simulasi dengan jumlah 40 *node*, dan 6351167,614 ns pada simulasi dengan jumlah 50 *node*. Penurunan *delay* disebabkan karena adanya black hole pada jaringan MANET sehingga semua data yang melewati *black hole* akan dibuang (*drop data*). Selanjutnya dilakukan penambahan mekanisme pendeteksian *black hole* pada jaringan MANET yang terdapat *black hole*. Mekanisme dapat menurunkan nilai *delay* pada ketiga skenario jumlah *node* yang berbeda. penurunan *delay* disebabkan karena mekanisme pendeteksian menghindari rute *black hole* dalam pengiriman data, ketika proses mencari posisi *black hole* dan mencari rute lain untuk menghindari *black hole* membutuhkan waktu tambahan membuat beberapa data menjadi gagal terkirim atau melewati *black hole*. Pada simulasi dengan pendeteksian *black hole* menunjukan nilai *delay* menjadi sebesar 12154317,99 ns pada simulasi dengan jumlah 30 *node*, 10199360,49 ns pada simulasi dengan jumlah 40 *node*, dan 13056686,94 ns pada simulasi dengan jumlah 50 *node*.

# PENUTUP

## Kesimpulan

Dari hasil pengujian yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Implementasi jaringan MANET dengan kondisi terdapat *black hole* dapat dijalankan. *Black hole* berhasil membuang semua paket yang dilewatkan pada dirinya, sehingga data yang dikirimkan tidak dapat terkirim ke tujuan. Pada hasil pengujian ketiga skenario dapat dilihat ketika simulasi jaringan MANET dengan kondisi terdapat *black hole* dapat mempengaruhi nilai *packet loss* menjadi lebih besar dibandingkan dengan simulasi dalam kondisi normal.
2. Implementasi jaringan MANET dengan kondisi terdapat mekanisme deteksi *black hole* dapat dijalankan. Mekanisme pendeteksian *black hole* dapat mengetahui identitas *black hole* dan menghindari rute *black hole* dalam pengiriman data, sehingga data yang dikirimkan dapat terkirim ke tujuan. Pada hasil pengujian ketiga skenario dapat dilihat ketika simulasi jaringan MANET dengan kondisi terdapat deteksi *black hole* dapat mempengaruhi nilai *packet delivery ratio* menjadi lebih besar dibandingkan dengan simulasi dengan simulasi dalam kondisi terdapat *black hole*.
3. Pada hasil pengujian dapat dilihat ketika simulasi dengan kondisi terdapat mekanisme deteksi *black hole* membuat nilai *packet loss* menjadi lebih kecil dibandingkan ketika tidak terdapat mekanisme deteksi. Nilai *packet loss* pada pergerakan *random walk* turun dari 99,9959 % dalam kondisi terdapat *black hole* menjadi 0,209308 % dalam kondisi terdapat mekanisme deteksi *black hole*. Pada hasil pengujian dapat dilihat ketika simulasi dengan kondisi terdapat mekanisme deteksi *black hole* membuat nilai *packet delivery ratio* menjadi lebih besar dibandingkan ketika tidak terdapat mekanisme deteksi. Nilai *packet delivery ratio* pada pergerakan *random walk* naik dari 0,00410408 % dalam kondisi terdapat *black hole* menjadi 94,164 % dalam kondisi terdapat mekanisme deteksi *black hole*. Pada hasil pengujian dapat dilihat ketika simulasi dengan kondisi terdapat mekanisme deteksi *black hole*, Nilai delay dalam skenario posisi *black hole* menjadi lebih besar dibandingkan dengan simulasi tanpa mekanisme deteksi. Nilai *delay* mengalami kenaikan karena diperlukan tambahan waktu untuk mencari posisi *black hole* dan mencari rute lain untuk menghindari *black hole.* Jumlah data yang dikirimkan pada skenario posisi *black hole* memiliki jumlah yang sama pada setiap simulasi. nilai *delay* dalam skenario tipe pergerakan dan jumlah *node* menjadi lebih kecil dibandingkan dengan simulasi tanpa mekanisme deteksi. Nilai *delay* mengalami penurunan karena jumlah data yang terkirim pada setiap simulasi memiliki jumlah yang berbeda.

## Saran

Dari hasil pengujian yang telah dilakukan dapat diambil beberapa saran sebagai berikut:

1. Penelitian berikutnya dapat dikembangkan dengan menggunakan mekanisme lain untuk menghindari atau mengisolasi *node* yang berperan sebagai *black hole* dari jaringan MANET
2. Penelitian berikutnya dapat dikembangkan dengan protokol *routing* dan serangan lain pada jaringan MANET

DAFTAR REFERENSI

Abdelhaq, M., Serhan, S., Alsaqour, R. and Hassan, R., 2011. A local intrusion detection routing security over MANET network. *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011*, (July), pp.3–8.

Anon 2019. *ns-3 Tutorial*. [online] Available at: <https://www.nsnam.org/docs/release/3.29/tutorial/ns-3-tutorial.pdf> [Accessed 13 Feb. 2019].

Arief, M.R., 2007. TEKNOLOGI JARINGAN TANPA KABEL ( WIRELESS ). 2007(November), pp.1–8.

Basagni, S., Conti, M., Giordano, S. and Stojmenovic, I., 2004. *Mobile Ad Hoc Networking*.

Choudhury, D.R., Ragha, L. and Marathe, N., 2015. Implementing and improving the performance of AODV by receive reply method and securing it from Black hole attack. *Procedia Computer Science*, [online] 45(C), pp.564–570. Available at: <http://dx.doi.org/10.1016/j.procs.2015.03.109>.

Dorri, A. and Kamel, S.R., 2015. Security Challenges in Mobile Ad Hoc Networks: A Survey. *International Journal of Computer Science & Engineering Survey*, [online] 6(1), pp.15–29. Available at: <http://www.airccse.org/journal/ijcses/papers/6115ijcses02.pdf>.

van Glabbeek, R., Höfner, P., Portmann, M. and Tan, W.L., 2016. Modelling and verifying the AODV routing protocol. *Distributed Computing*, 29(4), pp.279–315.

Istiqomah, T., Siregar, R.A. and Kartikasari, D.P., 2018. Implementasi Serangan Black Hole pada Mobile Ad-Hoc Network dengan Pergerakan Dinamis Terstruktur menggunakan Protokol Dynamic Source Routing. 2(12), pp.7262–7270.

Jain, A.K. and Tokekar, V., 2015. Mitigating the effects of Black hole attacks on AODV routing protocol in mobile ad hoc networks. *2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015*, 00(c).

Janak, N. and Patel, K., 2018. Trust Value based Algorithm to Identify and Defense Gray- Hole and Black-Hole attack present in MANET using Clustering Method. 4(4), pp.281–287.

Kaneria, P. and Rajavat, A., 2016. Detecting and avoiding of worm hole attack on MANET using trusted AODV routing algorithm. *2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016*.

M, L.R.P., Primananda, R. and Basuki, A., 2019. Pengaruh Model Mobilitas Node Pada Protokol Routing AODV dalam. 3(1), pp.563–572.

Muliantara, A., 2012. Program Studi Teknik Informatika , Jurusan Ilmu Komputer , Fakultas Matematika dan Ilmu Pengetahuan Alam , Universitas Udayana Kata Kunci : Anomali , Threshold , Denial of Services , Sistem Deteksi Intrusi nilai threshold merupakan suatu keharusan untuk membantu sistem deteksi intrusi ( IDS ) dalam membuat suatu keputusan yang baik dalam mengidentifikasi atau mendeteksi adanya suatu serangan [ 10 ]. Nilai threshold bisa ditentukan dengan. 1(2), pp.8–13.

Panda, N. and Pattanayak, B.K., 2018. Analysis of blackhole attack in AODV and DSR. *International Journal of Electrical and Computer Engineering*, 8(5), pp.3093–3102.

Simamora, P., 2014. *Model Pembelajaran Teknologi Informasi Dengan Teknik*.

Technology, I.J.I. and Science, C., 2013. Performance Analysis of MANET Routing Protocols in Different Mobility Models. (May), pp.73–82.