BAB II
TINJAUAN PUSTAKA

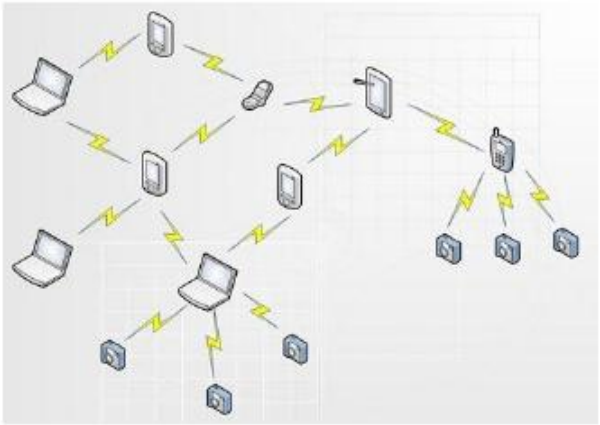
Bab ini membahas mengenai teori-teori penting yang dapat menjadi acuan dalam skripsi ini. Bagian tersebut meliputi penjelasan tentang pengenalan MANET (*Mobile Ad Hoc Network*), protokol *routing*, AODV (*Ad Hoc On Demand Vector)*, Serangan *Blackhole*, mekanisme *watchdog*, NS 2, dan penelitian terkait.

2.1 Umum

2.1.1 MANET (*Mobile Ad Hoc Network*)

Merupakan suatu jaringan yang terdiri dari berbagai perangkat bergerak (*mobile)* yang tergabung menjadi satu tanpa adanya infrastruktur yang memadai, sehingga membentuk suatu jaringan yang bersifat sementara (Irawan, Dedy; Roestam, Rusdianto, 2011). Dimana tiap perangkat yang ada memiliki tampilan nirkabel yang saling terhubung melalui gelombang radio.

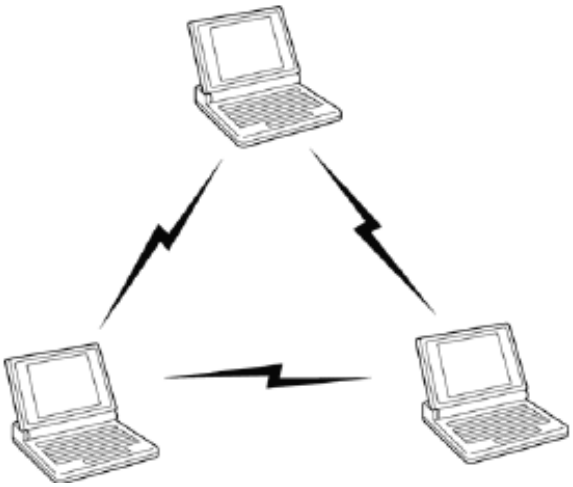
Perangkat - perangkat tersebut disebut *node*. Contoh nyata dari suatu node pada jaringan ad hoc adalah sebuah laptop dan *smartphone* yang saling terhubung secara langsung. Berikut adalah contoh gambar dari perangkat - perangkat yang ada pada jaringan ad hoc :



Gambar 2.1 Perangkat Heterogen

(Sumber :KNS&I11-052, 2011)

Perangkat heterogen (Gambar 2.1) terdiri dari perangkat yang berbeda - beda sedangkan perangkat yang sama disebut sebagai homogen seperti Gambar 2.2.



Gambar 2.2 Perangkat Homogen
(Sumber : KNS&I11-052, 2011)

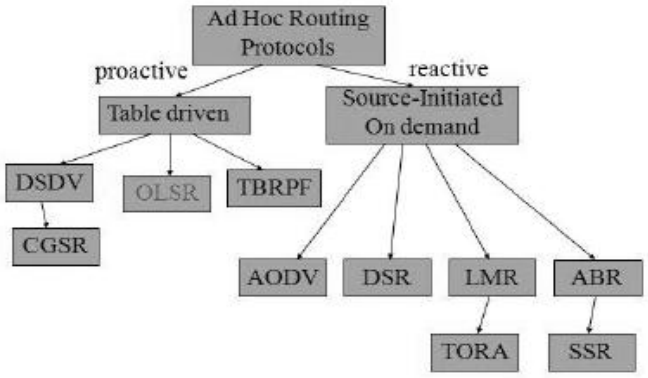
2.1.2 Protokol Routing

Terdapat banyak jenis protokol yang digunakan saat ini. Masing – masing memiliki kelebihan dan kekurangan. Tujuan dari penggunaan protokol seperti yang tercatat pada (Outline, 2004) yaitu :

a. Menyederhanakan proses manajemen jaringan karena alamat – alamat yang dicapai dapat segera diketahui secara otomatis.
b. Menemukan jalur – jalur "bebas – loop" di dalam jaringan.
c. Menetapkan jalur "terbaik" di antara beberapa pilihan yang tersedia.
d. Memastikan bahwa semua router yang ada di dalam jaringan "menyetujui" jalur – jalur terbaik yang telah ditetapkan.

Ada dua pernyataan umum mengenai tipe protokol untuk jaringan *Ad Hoc*, yaitu protokol proaktif dan protokol reaktif (Irawan, Dedy; Roestam, Rusdianto, 2011). Protokol routing reaktif bersifat *on demand*, yang artinya membentuk suatu rute dari sebuah node sumber ke node tujuan hanya berdasarkan pada permintaan node sumber tersebut. Sementara protokol routing proaktif bersifat *table driven*, dimana tiap *node* menyimpan tabel yang berisi informasi rute ke setiap node yang diketahui. Informasi rute diperbaharui secara berkala jika terjadi perubahan *link*. Penggunaan protokol routing proaktif secara mendasar memberikan solusi terpendek *End to End delay*, karena informasi routing selalu tersedia dan diperbaharui secara berkala dibandingkan protokol

routing reaktif. Berikut adalah gambar yang menjelaskan beberapa tipe protokol routing *Ad Hoc*.



Gambar 2.3 Karakteristik Protokol Routing

(Sumber :KNS&I11-052, 2011)

Gambar 2.3 menjelaskan mengenai beberapa protokol yang ada pada MANET seperti AODV, OLSR, dan DSDV. Masing - masing mempunyai karakteristik yang berbeda. Pada ad hoc ada dua tipe protokol routing, yaitu :

1. Proaktif:
   - Destination Sequenced Distance Vector (DSDV), Prinsip kerja protokol routing ini mengacu kepada algoritma penentuan route *Bellman-Ford* berdasarkan nilai pembobotan setiap link. Setiap node menjaga tabel routingnya yang berisi arah tujuan, jumlah hop setiap tujuan dan *sequence number*. Proses update routing dilakukan secara periodik. Protokol routing ini bebas dari kejadian *looping* route. Tetapi salah satu kelemahan DSDV adalah tidak mendukung *multipath routing* (*routing* ke banyak tujuan).
   - Cluster Switch Gateway Routing (CSGR), dalam protokol routing ini node dikelompokkanke dalam beberapa *cluster.*
   - Optimized Link State Routing Protocol (OLSR), adalah protokol routing proaktif yang menggunakan pesan Halo dan kontrol topologi (TC) untuk menemukan dan kemudian menyebarkan informasi link state seluruh jaringan *ad hoc mobile*. Implementasi: dioptimalkan untuk jaringan MANET tetapi juga dapat digunakan di jaringan

wireless lain. Implementasi: NU/Linux, FreeBSD, NetBSD, OSX dan Windows systems.

– Topology broadcast based on reverse-path forwarding (TBRPF), berfungsihanya mentransmisikan perbedaan antara keadaan jaringan sebelumnya dan keadaan jaringan saat ini. Oleh karena itu, pesan routing lebih kecildan dapat dikirim lebih sering. Ini berarti bahwa tabel routing nodelebih *up-to-date*. Implementasi: MANET dengan IPv4

2. Reaktif:

– Dynamic Source Routing (DSR), Protokol routing ini bekerja berdasarkan routing dari node sebelumnya. Node akan memperbarui rute berdasarkan rute baru yang didapatnya. Proses routing terdiri atas dua bagian, pencarian rute dan pemeliharaan rute. Pencarian rute digunakan untuk meminta dan meneruskan informasi rute. Pemeliharaan rute digunakan untuk informasikejadian kesalahan rute dan *acknowledgements*. Sama halnya dengan AODV, protokol ini akan membebani link. Semakin besar jaringan, *control packets* dan *message packets* akan semakin banyak, yang akan berakibat meminta alokasi bandwith.. Implementasi: MANET dengan IPv4.

– Ad hoc On-demand Distance Vector (AODV), AODV berfungsi bekerja selama koneksi rute dari pengirimke penerima yang valid, maka AODV tidak akan melakukan pencarian lagi. AODV memelihara rute selama dibutuhkan. Implementasi: MANET dengan IPv4 dan IPv6.

– Temporally Ordered Routing Algorithm (TORA),Protokol routing ini bersifat adaptif dan bebas dari kemungkinan looping sehingga sangat cocok untuk kondisi jaringan yang berubah-ubah. Node pengirim menyediakan beberapa route untuk node tujuan, sehingga jika satu route gagal dapat digunakan route lain. Dengan adanya banyak route dari node pengirim, maka pengiriman paket data dapat tidak terganggu saat pertama kali terjadinya perubahan jaringan.

Terjadi 3 proses didalam protokol ini, yaitu *route creation, route maintenance* dan *route erasure*.

– Associativy Based Routing (ABR), ABR berfungsi hanya mempertahankan rute untuk sumber yang benar-benar menginginkan rute. Namun, ABR tidak menjalankan rute berdasarkan informasi rute alternatif yang tersimpan.
– Signal Stability Routing (SSR),SSR memilih route berdasarkan kuat sinyal antar node dan terbagi atas dua protokol, *Dynamic Routing Protocol* (DRP) dan *Static Routing Protocol* (SRP). DRP bertanggung jawab untuk menjaga tabel stabilitas sinyal dan tabel routing. SRP memproses paket dengan melewatkan paket ke link dengan intensitas sinyal yang lebih besar.

2.1.3 AODV (Ad Hoc On Demand Vector)

AODV merupakan sebuah routing yang menyampaikan pesan antar komputer portabel. Dimana tiap komputer portabel atau biasa disebut node, menyampaikan pesan yang tidak bisa dikirim langsung, melalui node – node yang berada di sekitarnya untuk meneruskan pesan ke node tujuan dari node sumber. Proses ini dilakukan oleh AODV dengan cara menemukan rute untuk pengiriman pesan. AODV memastikan rute ini tidak terdapat *loops* dan mencoba untuk mencari rute terpendek agar sampai pada node tujuan. AODV juga dapat menangani perubahan rute dan menciptakan rute baru jika terjadi kegagalan. AODV merupakan '*on demand routing protocol'* dengan *delay* sedikit. Ini berarti sebuah rute akan terbentuk saat dibutuhkan untuk mengurangi tingkat atas. AODV mendukung Unicast, Broadcast dan Multicast.

Ada dua jenis rute pada AODV, yaitu penemuan rute dan pemeliharaan rute. Penemuan rute berupa RREQ (*Route Request*) dan RREP (*Route Reply*) sedangkan pemeliharaan rute berupa RERR (*Route Erorr*).

a. RREQ (Route Request)

RREQ dan RREP diciptakan oleh AODV saat sumber membutuhkan suatu rute untuk pengiriman paket data menuju tujuan namun belum terbentuk sebuah rute yang sesuai dengan permintaan sumber. Sehingga node sumber akan menyebarkan proses penemuan rute untuk menemukan rute yang sesuai menuju node tujuan. RREQ dikirimkan pada node yang berada di luar wilayah node

sumber dan belum terbentuk sebuah rute antara dua node tersebut. Node yang menerima RREQ akan memperbarui informasi untuk dikirim kembali sebagai balasan dan akan memelihara informasi yang telah valid tersebut menciptakan sebuah *path* cadangan menuju node sumber dalam bentuk tabel routing.

Node penerima RREQ memiliki dua pilihan, yaitu :

a. Saat node tersebut mengetahui node tujuan, maka node akan mengirim RREP sebagai balasan kepada node sumber.
b. Saat node tidak mengetahui node tujuan, maka node tersebut akan mengirim ulang RREQ dari node sumber ke node lainnya sehingga nilai pada *hop counter* akan bertambah.

b. RREP (Route Reply)

Saat node sumber menerima RREP, sebuah jalur akan terbentuk diantara node sumber dengan node penerima. Sehingga pengiriman paket menuju node tujuan akan mulai diproses. Ketika node sumber menerima RREP dengan nomer yang berurut yang lebih besar atau sama dan *hop count* yang lebih kecil, maka node sumber akan memperbarui informasi *routing* menuju node tujuan.

c. RERR (Route Erorr)

Saat proses pengiriman berlangsung dan terjadi perubahan topologi jaringan yang mengakibatkan berubahnya rute yang telah terbentuk sehingga node tidak dapat mengirim paket, maka sebuah node akan mengirimkan pesan *erorr* pada *node* lainnya. Setiap nodeyang menerima pesan *erorr* tersebut akan melakukan perubahan informasi dalam tabel routing. Setelah node sumber menerima pesan *erorr* tersebut, maka proses pengiriman akan diulangi kembali dari awal saat node sumber menyebarkan proses penemuan rute pada node yang berada di sekitarnya.

AODV tidak akan membangun sebuah rute jika itu tidak diperlukan. Informasi rute tersimpan hanya pada node sumber, node tujuan, dan node – node perantara. Skenario ini mengurangi pemakaian memori yang berlebihan, mengurangi pemakaian sumber jaringan dan dapat berjalan dengan baik pada situasi dengan tingkat pergerakan node yang tinggi. Ciri utama dari AODV adalah menjaga *timer-based state* pada setiap

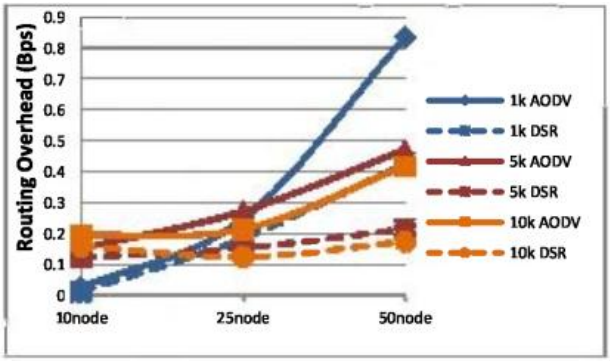
node sesuai dengan penggunaan tabel *routing*. Tabel r*outing* akan kadaluarsa jika jarang digunakan (Sari, Riri Fitri; Syarif, Abdusy dan Budiardjo, Bagio , 2008).

Di sisi lain, disebutkan oleh Yonas Sidharta dan Damar Widjaja tahun 2013, tingginya nilai *overhead* pada AODV sangat dipengaruhi oleh kapasitas jaringan. *Routing overhead* AODV semakin tinggi tingkatannya pada saat kondisi jaringan dengan jumlah *node* 25 dan 50. Pada tabel 2.1 menampilkan hasil penelitian *overhead* yang dilakukan oleh Yonas dan Damar.

Tabel 2.1 Hasil Perhitungan Rata – Rata *Overhead*

	10 *node*		
	1k	5k	10k
AODV	0.0255	0.1511	0.1937
DSR	0.0112	0.1223	0.1591
	25 *node*		
AODV	0.2474	0.2723	0.2139
DSR	0.1842	0.1529	0.1236
	50 *node*		
AODV	0.8353	0.47112	0.4181
DSR	0.4262	0.2139	0.1737

(Sumber : Jurnal Teknologi Vol 6 No.1, 2013)



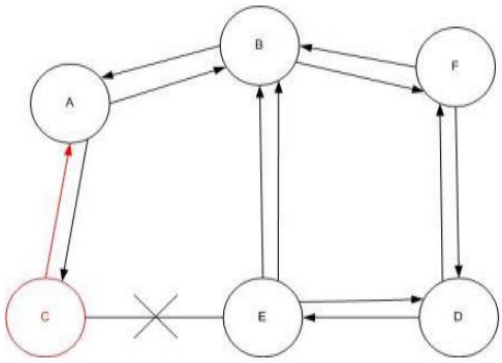
Gambar 2.4 Pengaruh Jumlah *Node* Terhadap Rata – Rata *Overhead*

(Sumber : Jurnal Teknologi Vol 6 No.1, 2013)

Yonas dan Damar (2013) menjelaskan dalam laporan mereka mengenai gambar 2.4 di atas bahwa *routing* AODV memiliki nilai *routing overhead* lebih besar. Tinggi nilai *routing overhead* pada AODV sangat dipengaruhi oleh kapasitas jaringan. *Routing overhead* AODV semakin meningkat saat kondisi jaringan berjumlah 25 *node* dan 50 *node.*

2.1.4 *Black Hole*

Serangan *Black Hole* merupakan serangan yang ada pada MANET. Serangan ini merupakan sebuah node yang mengirimkan RREP palsu pada node sumber yang telah menyebarkan RREQ untuk memulai pengiriman sebuah paket. *Node* ini akan memberi informasi baru mengenai rute baru yang lolos dari pengecekan tabel *routing*. Dengan cara ini, node tersebut dapat mengecoh node sumber agar menerima RREP yang dikirimkan. Sehingga paket yang terkirim akan dicegah dan proses pengiriman ke node tujuan mengalami gangguan.



Gambar 2.5 Serangan Black Hole

(Sumber : IJSIT Vol.5, 2014)

Gambar 2.5 memperlihatkan bagaimana proses serangan *Black Hole* berlangsung. Proses serangan *Black hole* diuraikan sebagai berikut:

- nodeA akan melakukan pengiriman paket ke nodeD dan menyebarkan proses penemuan rute pada semua node yang berada didekatnya.
- Jika *node* C adalah node berbahaya atau gangguan, maka node tersebut akan membentuk sebuat rute aktif menuju node tujuan dengan cepat setelah node tersebut menerima paket RREQ.
- Balasan RREP palsu akan dikirimkan oleh *node* C kepada *node* A sebelum node yang lain membalas. Sehingga *node*A akan mengira jika rute aktif tersebut adalah sebuah rute pengiriman yang telah selesai terbentuk.
- *Node* A tidak akan menghiraukan seluruh balasan dari *node* lain yang melakukan RREP padanya dan pengiriman paket akan mulai berlangsung pada *node* C. Karena hal inilah, paket akan mengalami

kehilangan ataupun pengurangan informasi data(*Ullah, Irshad and Rehman, Shoaib Ur, 2010*).

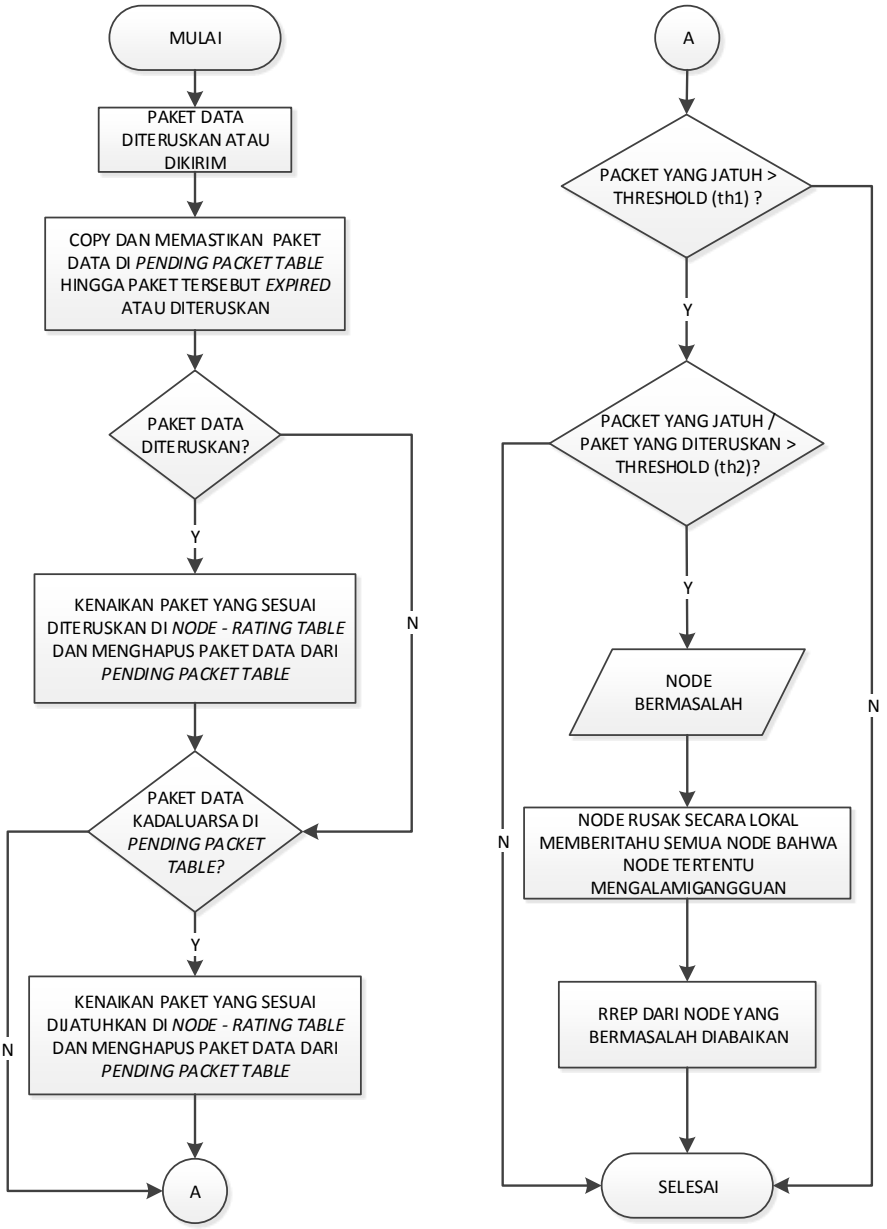
Terdapat dua tipe serangan *Black Hole* pada AODV, yaitu *Internal Black Hole Attack* dan *External Black Hole Attack*. *Internal Black Hole Attack* adalah jenis serangan yang terjadi pada rute asal paket dan rute tujuan paket dikirim. Secepat setelah mendapatkan kesempatan, *node* penyerang akan membuat dirinya menjadi sebuah bagian dari aktif *router* data. Pada tahap ini *node* penyerang mampu untuk melakukan serangan dengan memulainya pada transmisi data. Ini termasuk serangan internal karena *node* tersebut berada dalam rute pengiriman data. Serangan internal lebih rentan dan sulit dicegah karena sulitnya mendeteksi *node* internal yang berbahaya. Sementara *External Black Hole Attack* secara fisik berada di luar jaringan dan menolak akses menuju lalu lintas jaringan atau menciptakan kemacetan dalam jaringan atau dengan mengganggu seluruh jaringan. Serangan eksternal dapat menjadi jenis serangan internal jika mengambil alih kontrol dari *node* penyerang yang berada di bagian internal dan mengontrol untuk melakukan suatu serangan terhadap *node* lain yang ada pada MANET. Serangan *Black Hole* eksternal dapat dilihat dari poin – pon berikut ini :

a. Node penyerang mendeteksi rute yang aktif dan mencatat alamat tujuan.
b. Node penyerang mengirim rute paket balasan (RREP) termasuk alamat tujuan *field spoofed* menuju alamat tujuan yang tidak diketahui. Nilai hop menjadi lebih rendah dan pengurutan nomer menjadi lebih tinggi.
c. Node penyerang mengirim RREP ke node terdekat yang mana termasuk dalam rute aktif. Hal ini juga termasuk dalam pengiriman langsung menuju node sumber data yang ada.
d. RREP diterima oleh node terdekat yang ada kepada node penyerang yang menyesuaikan diri dengan rute berbeda yang dibentuk dari nodesumber.
e. Informasi baru diterima dalam rute yang memberi balasan akan mengijinkan node sumber untuk memperbarui tabel *routing*.
f. Rute baru dipilih oleh node sumber untuk seleksi data.
g. Node penyerang akan menjatuhkan seluruh data yang ada pada rute yang terbentuk.

2.1.5 Mekanisme *Watchdog*

Inti dari mekanisme *watchdog* adalah pengawasan tepat (*promiscuous monitoring)*. Jika terdeteksi node berbahaya, node sumber akan memilih rute baru yang bebas dari node berbahaya dengan bantuan "*Path rater*". Mekanisme ini tidak akan menampilkan hasil dengan baik pada saat kondisi jaringan tidak menguntungkan dimana terdapat gangguan dari node berbahaya yang mengakibatkan data rusak saat probabilitas tinggi. Terdapat dua jenis pendekatan pada mekanisme *watchdog*, yaitu pendekatan pertama adalah deteksi serangan *End-to-End* menggunakan deteksi kesalahan. Kekurangannya adalah *Throughput* jaringan dapat berkurang bahkan saat tidak ada node berbahaya.

Dalam skema ini, node sumber dan node tujuan tidak mengetahui tentang lokasi node berbahaya. Yang kedua adalah pendekatan yang diambil secara umum untuk mengeksploitasi sifat penyiaran dari media nirkabel dimana sebuah node akan mengawasi arus pergerakan data pada suatu jaringan, node tersebut merupakan *watchdog*. Untuk pendekatan yang kedua ini memiliki tantangan utama, yaitu jaringan yang perlahan menghilang, terjadinya tabrakan pengirimandan gangguan lain yang dapat muncul tanpa diduga. Sehingga *watchdog* tidak dapat menangkap pengiriman yang berlangsung. Di sini, perilaku *next - hop* diukur dengan evaluasi catatan lokal terbagi menjadi 2, yaitu *ratio byte and ratio packet* dan *forwarded by the next - hop neighbor*. Tingkat kepercayaan dari*node* adalah gabungan dari pengamatan lokal dan informasi yang disiarkan. Tingkat kepercayaan dimasukan pada RREQ tersebut kemudian rute dipilih dengan cara yang sama dengan AODV. Algoritma prinsip kerja mekanisme *watchdog* dapat dijelaskan pada Gambar 2.6 sebagai berikut:



Gambar 2.6 Diagram alur mekanisme *watchdog*

Mekanisme watchdog pada Gambar 2.6 dijelaskan sebagai berikut:

1. Data paket diteruskan atau dikirim kepada node tujuan.
2. Meng- *Copy* dan menjaga paket data di table *pending packet* hingga paket tersebut kadaluarsa atau diteruskan
3. Jika paket data diteruskan maka paket yang peningkatannya sesuai dengan tabel *node rating* diteruskan dan menghapus paket data dari tabel *pending packet*.

4. Jika paket data berakhir di tabel *pending packet*, maka paket yang peningkatannya sesuai dengan tabel *node rating* dihentikan dan menghapus paket data dari tabel *pending packet.*
5. Jika paket yang dihentikan lebih besar dari threshold (TH1) kemudian jika paket yang dihentikan ataupun paket yang diteruskan lebih besar dari threshold (Th2) *node* tersebut dianggap bermasalah *(misbehave)* kemudian memberi tahu semua *node* yang ada pada jaringan bahwa *node* tersebut bermasalah.
6. RREP yang datang dari *node* yang bermasalah tersebut diabaikan.

2.1.6 *Bayesian Watchdog*

Untuk mendeteksi node yang bermasalah, monitoring jaringan diperlukan. Setiap node harus mengawasi node terdekatnya, dan pengawas merupakan komponen populer dalam system deteksi gangguan didedikasikan untuk tugas ini. Masalah utama adalah bahwa pengawas ditandai dengan ketidakakuratan dan kecepatan deteksi rendah, pada dasarnya karena mobilitas dan sinyal gangguan. Hasil-hasil sebelumnya dari kelompok telah dievaluasi pengawas Bayesian pada Ad-hoc On-demand Distance Vector (AODV) routing dalam MANET. Pengawas bayesian ini hasil dari agregasi filter Bayesian dengan implementasi pengawas standar. Pengawas standar hanya mendengar paket yang dikirimkan dan diterima oleh tetangganya, menghitung paket yang harus ditransmisikan secara berulang, dan komputasi kepercayaan untuk setiap tetangga sebagai rasio "paket yang dikirimkan" untuk "paket yang seharusnya dikirimkan kembali". Jika node mentransmisikan semua paket yang seharusnya telah ditransmisikan ulang, memiliki tingkat kepercayaan dari 1 jika node memiliki tingkat kepercayaan yang lebih rendah dari ambang toleransi dikonfigurasi, node yang ditandai sebagai node berbahaya.

Peran filter Bayesian di pengawas adalah untuk tingkat kemungkinan memperkirakan kondisi sistem. Dasar matematika dari bayesian filter berikut: pada waktu t, state diperkirakan oleh variabel θ acak, yang tidak diketahui, dan ketidakpastian ini dimodelkan dengan asumsi bahwa θ sendiri diambil sesuai dengan distribusi yang diperbarui seperti terjadi pengamatan yang baru. Hal ini biasa disebut kepercayaan atau Belt (θ). Untuk menggambarkan hal ini, mari kita asumsikan pada persamaan berikut:

$$Bel_t(\vartheta) = p(\vartheta|z_1, z_2, .., z_n, .., z_t) = Beta(\alpha_t, \beta_t, \vartheta)$$

Keterangan:

$Bel_t(\vartheta)$= tingkat kepercayaan

ϑ = variable acak

α dan β= keadaan sistem

Z_t = informasi

Dalam pendekatan ini, variabel θ acak milik interval [0,1]. Bayesian filtering bergantung pada distribusi Beta, yang cocok untuk memperkirakan keyakinan dalam interval ini, seperti yang ditunjukkan dalam ekspresi 1; α dan β mewakili keadaan sistem, dan mereka diperbarui sesuai dengan persamaan berikut:

$$\alpha_{t+1} = \alpha_t + Z_t; \beta_{t+1} = \beta_t + Z_t$$

Keterangan:

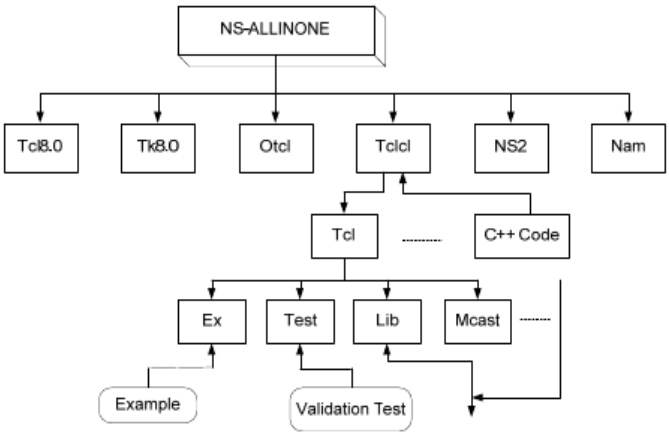
α dan β= keadaan sistem

Z_t = informasi pengawas lokal

Fungsi Beta hanya membutuhkan dua parameter yang terus diperbarui sebagai pengamatan yang dibuat atau dilaporkan. Dalam pendekatan ini, pengamatan ZT merupakan informasi dari pengawas lokal yang diperoleh dalam interval waktu [t, t + AT] tentang persentase paket yang tidak diteruskan. Pengawas bayesian menggunakan tiga parameter: dua parameter pertama adalah α dan β, yang ditangani ke fungsi Beta untuk mendapatkan estimasi jalur node. Dengan demikian, kita dapat mengatakan bahwa α dan β adalah representasi numerik dari reputasi node. Parameter ketiga adalah γ, yang mewakili devaluasi bahwa pengamatan harus berusaha keras untuk beradaptasi dengan perilaku pengawas untuk skenario yang terus berubah tanpa mempengaruhi node tertentu. Jadi itu adalah mekanisme untuk mengintegrasikan kembali node ke MANET jika mereka mengubah perilaku mereka ke yang lebih kooperatif.

2.1.7 NS2 *(Network Simulator 2)*

NS2 (*Network Simulator 2*) dikembangkan pertama kali di UCB (University of California Berkeley) yang didukung oleh DARPA. NS2 merupakan suatu *system* yang bekerja pada *system* Unix/Linux , NS2 juga dapat dijalankan dalam *system* Windows namun harus menggunakan Cygwin sebagai Linux *Enviroment*nya. NS2 dibangun dari 2 bahasa pemrograman yaitu C++, sebagai *library* yang berisi *event scheduler*, protokol , dan *network component* yang diimplementasikan pada simulasi oleh *user*. Otcl digunakan pada *script* simulasi yang ditulis oeh NS *user*. Otcl juga berperan sebagai *interpreter*. Bahasa C++ digunakan pada library karena C++ mampu mendukung *runtime* simulasi yang cepat, meskipun simulasi melibatkan simulasi jumlah paket dan sumber data dalam jumlah besar. Sedangkan bahasa Tcl memberikan respon *runtime* yang lebih lambat daripada C++, namun jika terdapat kesalahan , respon Tcl terhadap kesalahan *syntax* dan perubahan *script* berlangsung dengan cepat dan interaktif. Pengetahuan tentang komponen pembangun NS2 dan letaknya akan sangat berguna dalam membangun simulasi. Komponen pembangun NS2 dijelaskan pada blok diagram gambar 2.7 berikut.



Gambar 2.7 Komponen Pembangun NS 2(Sumber : IT Training Center, 2011)

Keterangan:

Tcl :*Tool command language*
Otcl :*Object* Tcl

TK :*Tool Kit*

Tclcl : Tcl/C++ *Interface*

NS2 : NS versi 2

Nam : *Network animator*

Penjelasan secara singkat hubungan antara *header file* (.h) adalah untuk menemukan arti dari fungsi tanpa *body,* sementara *file* .cc untuk menemukan fungsi *body* dan file .tcl untuk melakukan konfigurasi jaringan. (seperti jumlah *node* dan parameter lainnya).

Transport agent pada NS 2 di jaringan internet, terdapat 4 layer komunikasi TCP/IP. Yaitu layer aplikasi, transport, IP dan network. Lapisan transport merupakan layer komunikasi yang mengatur komunikasi data yang akan digunakan oleh lapisan aplikasi di atasnya. NS berada pada lapisan transport dengan objek simulasi yang bernama *transport agent*. Pada simulasi pengiriman data, *transport agent* tidak dapat bergerak sendiri. *Transport agent* membutuhkan lapisan aplikasi di atasnya yang berfungsi sebagai *traffic generator*. Protokol lapisan transport data yang didukung network simulator 2 antara lain :

a. TCP (*Transport Control Protocol*)
Network simulator 2 mendukung 2 jenis TCP *agent*, yaitu *one way TCP* agent dan *two way TCP agent*.

b. UDP (*User Datagram Protocol*)
Koneksi dengan menggunakan UDP pada NS2 dilakukan dengan menggunakan *agent* UDP sebagai pengirim dan *agent* Null sebagai penerima.

c. RTP (*Real Time Transport Protocol*)
RTP menyelenggarakan *end to end delivery services* untuk data yang memiliki karakteristik *real time*, seperti VoIP (Voice Over IP) dan video interaktif. Layanan tersebut termasuk identifikasi tipe *payload*, pengurutan, *timestamping*, dan monitor pengiriman data. Sama seperti UDP, pemakai RTP sebagai *agent* pengirim dipasangkan dengan *agent* Null sebagai penerima.

Kedudukan lapisan aplikasi NS 2 pada sistem dunia nyata, aplikasi terhubung dengan lapisan transport yang ada di bawahnya melalui sebuah *Aplication Program Interface* (API). Jenis API yang umum digunakan yaitu *socket*. Ada 2 tipe dasar aplikasi yang disimulasikan pada NS2, yaitu:

a. Simulated Application

Pada saat ini baru terdapat dua jenis aplikasi yang disimulasikan oleh NS yaitu :

a. FTP dibangun untuk mensimulasikan *bulk* data transfer.

b. Telnet masing-masing aplikasi diatur oleh *transport agent*. Jumlah paket yang ditransmisikan diatur oleh mekanisme *flow control* dan *congestion control* TCP.

b. Generator Traffic

Object generator traffic dibagi atas 4 type, yaitu:

a. Eksponensial, *generator traffic* ini membangkitkan *traffic* dengan *inter arrival time* antar paket sesuai dengan fungsi eksponensial.

b. Pareto, *generator traffic* ini membangkitkan *traffic* dengan *inter arrival time* antar paket sesuai dengan fungsi pareto.

c. CBR, fungsi ini membangkitkan data secara berkelanjutan dengan *bit rate* yang konstan.

d. Traffic race, generator ini membangkitkan traffic dari sebuah file trace.

Secara umum simulasi *routing* pada NS2 dilakukan melalui 3 blok fungsi, yaitu *routing agent* yang mempertukarkan paket *routing* antar node, *routing logic* yang menggunakan informasi yang dikumpulkan oleh *routing agent* untuk melakukan perhitungan rute dalam bentuk tabel *routing* dan peng-klasifikasi yang ada pada node.

2.2 Penelitian Terkait

Bagian ini menjelaskan tentang penelitian yang telah dilakukan oleh Kanika Lakhani, Himani Bathla dan Rajesh Yadav dengan judul "*A Simulation Model to Secure the Routing Protocol AODV againts Black Hole Attack in MANET*" serta penelitian yang dilakukan oleh Surana K.A, Rathi S.B, Thosar T.P dan Snehal Mehatre dengan judul "*Securing Black Hole Attack in Routing Protocol AODV in MANET with WATCHDOG MECHANISM*" yang dijadikan sebagai acuan dalam tugas akhir kali ini.

Kedua penelitian tersebut bertujuan untuk mengetahui dampak dari serangan *Black Hole* pada routing protocolAODVdengan melakukan simulasi. Untuk mencapai tujuannya, mereka memperkenalkan sebuah protokolbaru yang disebut "*Modified AODV"*. Protokolbaru ini merupakan turunan dari AODV yang sudah ada. Dijelaskan penggunaan mekanisme *watchdog* memerlukan dua tabel tambahan di tiap node. Dari dua tabel ini nantinya dapat dilacak apakah terdapat serangan yang berupa nilai 0 dan 1,

dimana 0 sebagai keterangan prilaku baik dan 1 untuk node berbahaya. Dalam mekanisme *watchdog* juga dijelaskan jika tiap node yang ada akan tetap melakukan pelacakan saat melakukan pengiriman data. Penilaian dan penentuan apakah terdapat serangan dilakukan dengan perbandingan jumlah paket yang berhasil dikirim dengan jumlah paket yang gagal terkirim. Hal ini juga tergantung pada nilai ambang, yang berarti selama meneruskan paket sebanyak dua kali sama halnya dengan saat menjatuhkan paket tersebut maka tidak akan terdeteksi. Jika diambil nilai terendah maka akan menambah presentase positif palsu. Setelah mendeteksi node berbahaya, sebuah nodeakan melakukan perbaikan lokal untuk seluruh rute yang dilewati oleh node berbahaya tersebut.

Mekanisme serangan *Blackhole* saat simulasi berlangsung, node sumberakan menyebarkan proses penemuan rute untuk memulai pengiriman paket. Sebuah *node* yang diidentifikasikan sebagai Blackhole akan melakukan RREP pada node sumber dan melakukan serangan dengan menjatuhkan paketatau menjatuhkan paket.

Simulasi yang telah dilakukan dalam penelitian ini menggunakan pola pergerakan sebanyak 50 node dalam area berukuran 1000 x 1000 meter dan kecepatan maksimum tiap node sebesar 5 m/sec. Lalu lintas pergerakan tiap 50 *node* akan dimaksimalkan oleh 5 koneksi menggunakan CBR (*Constant bit Rate*) dan perbedaan nilai yang telah digunakan dalam simulasi. Nilai yang digunakan untuk memetakan lalu lintas pola secara acak. Dengan perubahan yang hanya memetakan nilai dengan koneksi CBR ini, dapat merubah lalu lintas pola secara keseluruhan. Parameter Simulasi yang telah digunakan dapat dilihat pada Tabel 2.2 di bawah ini :

Tabel 2.2 Simulasi Parameters yang telah digunakan dalam peneltian

Communication Type	CBR
Number of Nodes	50
Maximum mobility speed of nodes	5 m/sec
Simulation Area	1000m x 1000m
Simulation Time	200 sec
Packet Rate	4 packet/sec
Packet Size	512 bytes
Number of Connections	5
Transmission Range	250 m

Pause Times	0, 40, 120, 160 sec
Number of malicious node	0, 3, 5
Transmission speed	10 Mbps

(Sumber : IJCSNS, Vol 10 No 5, 2010)

Dengan mekanisme *watchdog*, tiap node membentuk dua buah tabel tambahan, yaitu *pending packet table* dan *node rating table.* Pada *pending packet table* terdapat empat kolom yang terdiri dari *Packet ID*, *Next Hop*, *Expiry Time*, dan *Packet Destination.*Tabel 2.3 ini berisi daftar paket yang proses pengirimannya tertunda karena masih menunggu update informasi dari node rating table untuk mencari daftar node yang aman untuk meneruskan paket. Setelah node yang aman ditemukan, paket akan dikirimkan ke tujuan melewati node tersebut dengan catatan masa expired belum habis.

Tabel 2.3 Pending Packet Table

Packet ID	Next Hop	Expiry Time	Packet Destination

(Sumber : IJCSNS, Vol 10 No 5, 2010)

Keterangan :

Packet ID : ID dari paket yang dikirim

Next Hop : alamat *node next - hop*

Expiry Time : batas waktu paket yang dikirim

Packet Destination : alamat tujuan pengiriman paket

Pada *node rating table* juga terdiri dari empat kolom, yaitu *Node Address*, *Packet Drops*, *Packet Forwards* dan *Serangan*(*Node Malware/*penyerang). Tabel ini berfungsi untuk melakukan *update* informasi saat proses berlangsung pada *Pending Packet Table*. Tabel 2.4 ini berisi node-node yang ada di jaringan.

Tabel 2.4 Node Rating Table

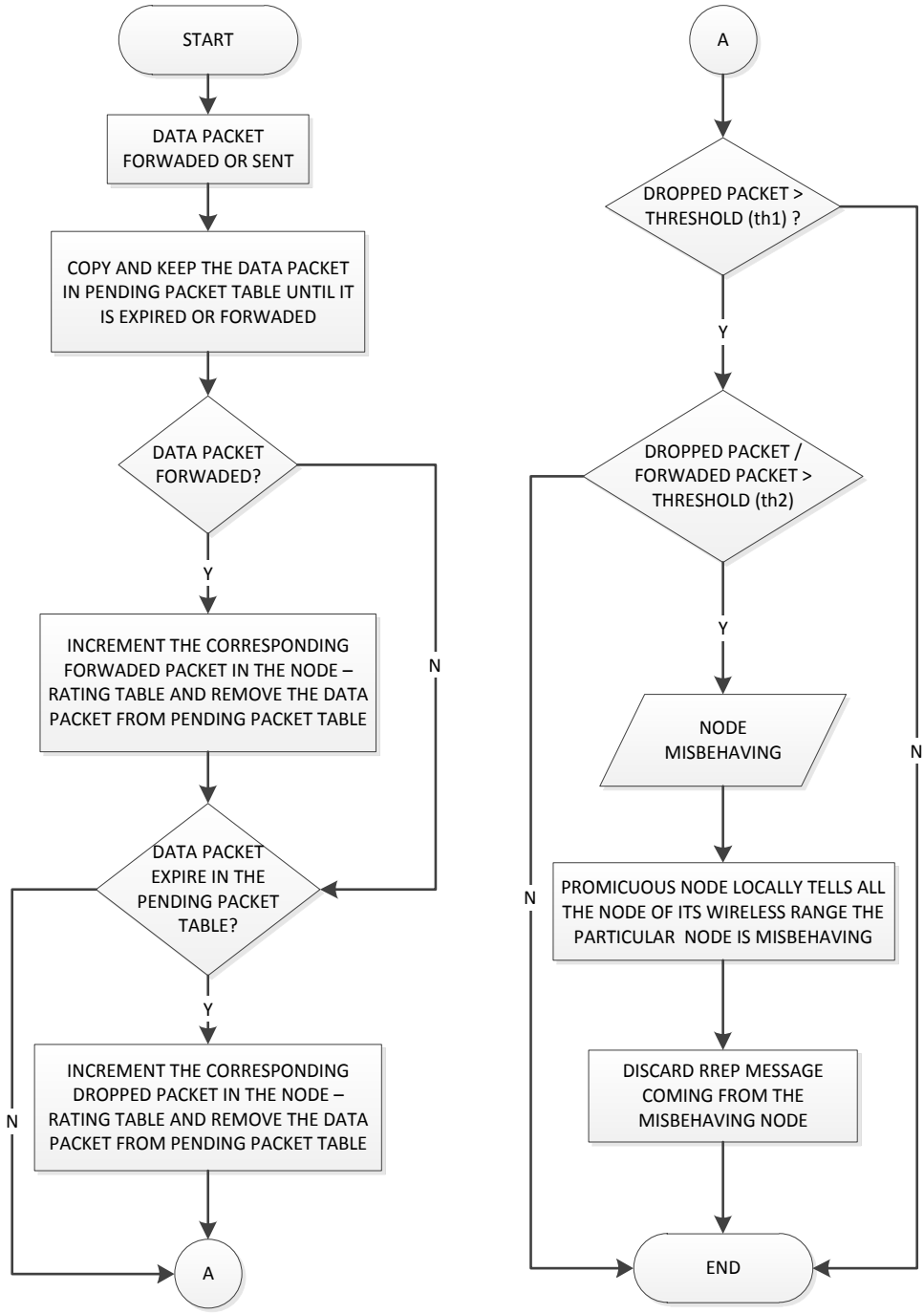
node Address	Packet Drops	Packet Forwards	Serangan

(Sumber : IJCSNS, Vol 10 No 5, 2010)

Keterangan :

node Address	: alamat *node next - hop*
Packet Drops	: perhitungan jumlah paket yang dijatuhkan
Packet Forwards	: perhitungan jumlah paket yang diteruskan
Serangan (node berbahaya)	: memiliki dua nilai 0 dan 1, 0 untuk node normal dan 1 untuk node berbahaya

Algoritma yang digunakan pada penelitian tim Kanika (2010) adalah sebagai berikut :



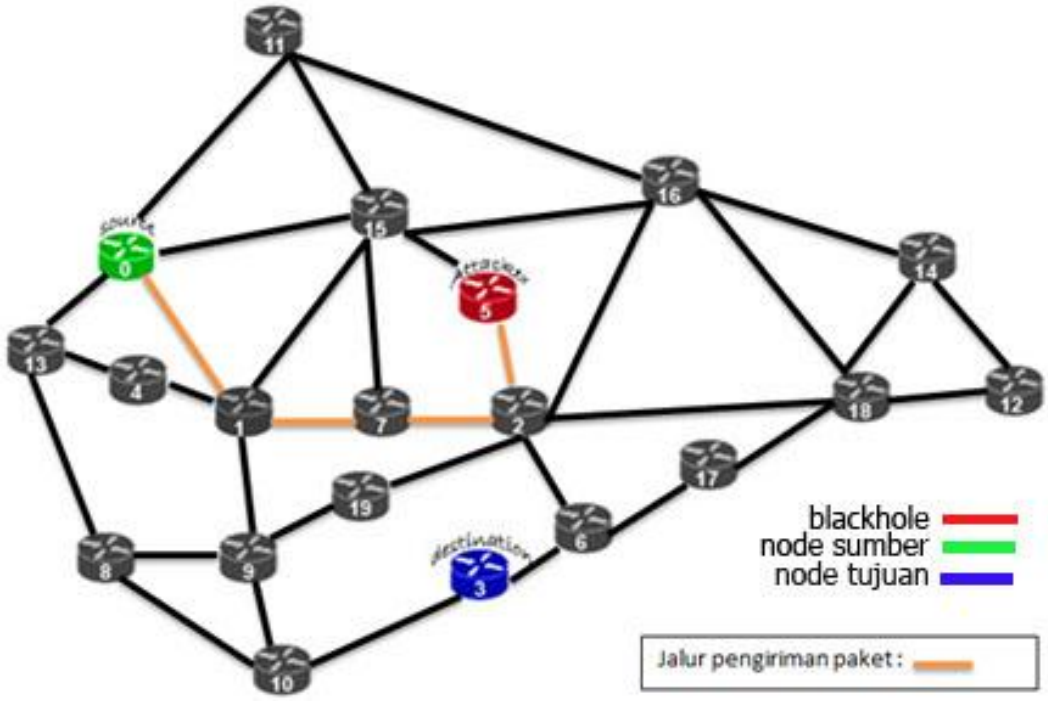
Gambar 2.8 Algoritma Mekanisme *Watchdog* pada penelitian tim Kanika

Hasil analisa dari penelitian ini ditampilkan pada ns-2.31 yang dijalankan pada sistem operasi Red Hat Linux Enterprise Server. Sebuah jaringan

dengan 50 node dipilih untuk simulasi dengan perbedaan waktu jeda 0, 40, 120 dan 160 detik. Perbandingan *Throughput* dan *Packet Delivery* dikalkulasi pada AODV yang telah ada menggunakan perbedaan skenario dengan *Black Hole* node pada node0, 3, dan 5. Dengan menggunakan parameter yang sama pada AODV modifikasi yang telah dilakukan tes seperti yang telah disebutkan sebelumnya dengan *Black Hole* pada *node* 0, 3 dan 5, dimana pada keduanya mekanisme *watchdog* dalam posisi aktif dan tidak aktif. Hasil yang ditampilkan saat serangan *Black Hole* node meningkat hingga 6% dari jumlah total *node* yang ada pada jaringan maka keberadaan *watchdog* secara aktif akan meningkat hingga 3% sampai 8% untuk skenario yang lain. Ketika serangan *Black Hole* meningkat hingga 10% dari total *node* yang ada pada jaringan maka keberadaan *watchdog* secara aktif meningkat hingga 10% sampai 18% untuk skenario yang lain.

2.3 Topologi

Topologi yang digunakan merupakan jenis topologi dinamis dimana pergerakan *routing protocol* AODV merupakan pergerakan secara acak dan berubah – ubah dalam beberapa waktu.

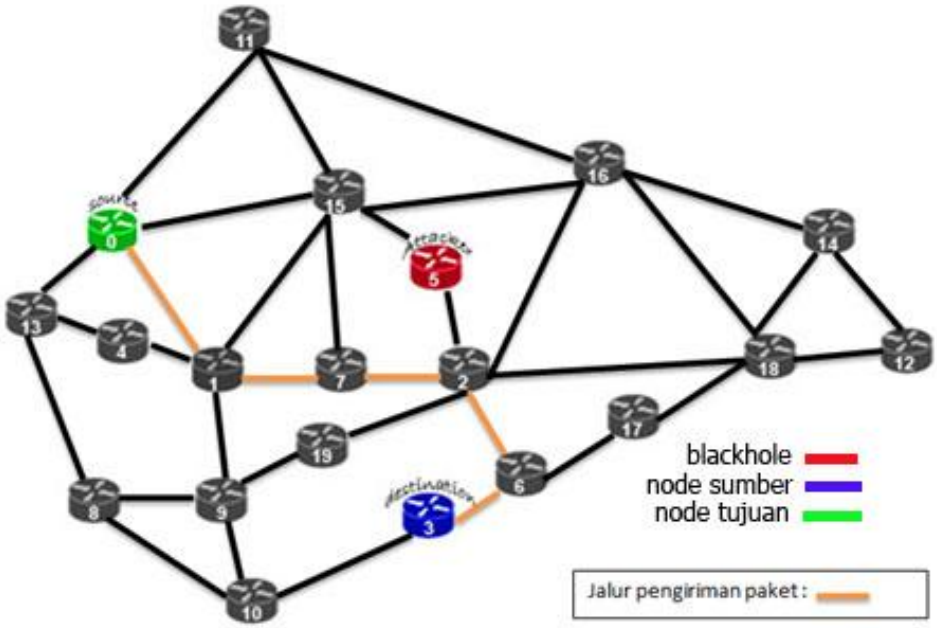


Gambar 2.9 Topologi tanpa Mekanisme *Watchdog*

Topologi pada Gambar 2.9, alur pengiriman data yang dilakukan oleh *node* sumber menuju *node* tujuan terganggu akibat *node* yang bermasalah

mengirimkan RREP palsu terhadap *node* – *node* yang berada dalam cakupan jalur pengiriman paket yang telah dibentuk oleh *node* sumber selama melakukan proses pencarian rute.

Sementara pada saat penggunaan mekanisme *watchdog* pada Gambar 2.10, RREP palsu yang dikirimkan oleh *node* bermasalah tidak akan dihiraukan sehingga paket dapat dikirimkan pada *node* tujuan.



Gambar 2.10 Topologi dengan mekanisme *Watchdog*