

## BAB 2

### DASAR TEORI

#### 2.1 KAJIAN PUSTAKA

Penelitian yang membandingkan *routing* protokol OLSR dan DSDV pada MANET [5]. Penelitian tersebut berfokus pada analisis kinerja keduanya berdasarkan pergerakan *node*. Penelitian ini menggunakan simulasi jaringan MANET dengan dua *model mobilitas node*, yaitu *random waypoint* dan *random direction*. Parameter *Quality of Service* (QoS) yang diukur meliputi *average end-to-end delay*, PDR, dan *routing overhead*. Skenario pengujian melibatkan variasi jumlah *node* (20, 30, 40, dan 50 *node*), luas area simulasi (200m<sup>2</sup>, 500m<sup>2</sup>, 800m<sup>2</sup>, dan 1000m<sup>2</sup>), dan mode *mobilitas* (*random waypoint* dan *random direction*). Hasil simulasi menunjukkan bahwa OLSR secara keseluruhan memiliki kinerja yang lebih baik dibandingkan dengan DSDV dalam hal PDR dan *routing overhead*. Namun, pada nilai *delay*, DSDV menunjukkan kinerja yang lebih efektif daripada OLSR.

Studi tentang analisis protokol *routing* OLSR terhadap pergerakan *gauss-markov* dan *random walk* pada manet dilakukan dalam penelitian tersebut [6]. Fokus utama penelitian adalah menganalisis kinerja *routing* protokol OLSR terhadap dua jenis pergerakan tersebut. Simulasi dilakukan dengan menggunakan OLSR pada pergerakan *gauss-markov* dan *random walk*. Variasi dalam simulasi mencakup jumlah *node* (20, 40, 60 dan 80), serta rentang kecepatan minimal dan maksimal (0-5 m/s, 5-10 m/s, 10-15 m/s, dan 15-20 m/s). Parameter QoS yang dievaluasi meliputi *packet delivery ratio* (PDR) , *end-to-end delay*, dan *routing overhead*. Hasil penelitian menunjukkan bahwa pergerakan *Gauss-Markov* memberikan kinerja yang lebih efektif dalam hal PDR dan *delay*, sementara pergerakan *random walk* menunjukkan nilai yang lebih efektif dalam hal *routing overhead*.

Adapun penelitian yang membandingkan *routing* protokol AODV, DSR dan OLSR pada MANET berdasarkan parameter QoS [7]. Penelitian mencakup analisis

kinerja *routing* protokol AODV, DSR, dan OLSR pada MANET, dengan fokus pada parameter QoS. Pergerakan yang digunakan adalah *random motion model*, sedangkan parameter QoS yang dievaluasi meliputi PDR, *packet loss*, *throughput*, dan *delay*. skenario pengujian melibatkan variasi jumlah *node*, yaitu 25, 50, 75, 100, 150, dan 200 *node*. Hasil simulasi menunjukkan bahwa OLSR memiliki nilai yang lebih efektif dibandingkan dengan AODV dan DSR pada semua parameter yang di evaluasi, termasuk PDR, *packet loss*, *throughput*, dan *delay*.

Terdapat penelitian yang membahas perbandingan *routing protocol* AOMDV, DSDV, Dan ZRP sebagai protokol *routing* Pada MANET [8]. penelitian tersebut mengevaluasi kinerja protokol AOMDV, DSDV, dan ZRP pada jaringan MANET dengan menggunakan pergerakan *random waypoint*. Parameter QoS yang digunakan meliputi *throughput*, *end-to-end delay*, PDR, dan *normalized routing load*. Simulasi dilakukan menggunakan NS2.35. skenario pengujian melibatkan *node* yang bervariasi yaitu (20, 40, 80, dan 100) dengan variasi luas area (500x500, 800x800, 1000x1000 meter). Hasil penelitian menunjukkan bahwa protokol AOMDV memiliki nilai PDR dan *throughput* tertinggi. Namun, pada pengukuran *end-to-end delay* dan *normalized routing load*, protokol DSDV menunjukkan nilai terbaik dibandingkan dengan protokol *routing* lainnya.

Terdapat penelitian yang membahas pengaruh *model mobilitas* terhadap konsumsi energi protokol OLSR pada MANET [9]. Penelitian ini mengadakan simulasi untuk mengevaluasi kinerja *routing* protokol OLSR dengan menggunakan berbagai jenis pergerakan *node*, termasuk *random waypoint*, *random walk*, dan *random direction*. Dalam skenario pengujian, variasi jumlah *node* (20, 40, 60, 80, dan 100) diimplementasikan, dengan fokus pada parameter konsumsi energi. Hasil penelitian menunjukkan bahwa pada pergerakan *random waypoint*, persentase total energi sisa mencapai 17,48%, sedangkan pada *random direction* mencapai 21,36%, dan pada *random walk* mencapai 21,05%.

Penelitian ini bertujuan untuk mengevaluasi kinerja *routing* protokol OLSR dan DSDV dalam konteks pergerakan *node random walk*. Berbeda dengan penelitian sebelumnya, penelitian ini berfokus pada kedua protokol tersebut. Pergerakan yang dipertimbangkan adalah *random walk*. Simulasi dilakukan menggunakan NS-3, dengan parameter QoS yang mencakup PDR, *throughput*,

*delay*, dan *packet loss*. Skenario penelitian mencakup variasi jumlah *node*, yaitu 20, 30, 40, dan 50.

**Tabel 2.1 Kajian peneliti sebelumnya**

Referensi	Tujuan	<i>Routing</i> Protokol	Pergerakan <i>node</i>	Parameter	Hasil
[5]	Melakukan Studi Analisis Perbandingan Kinerja Protokol <i>Routing</i> OLSR dan DSDV Pada MANET Berdasarkan Pergerakan <i>Node</i>	OLSR dan DSDV	<i>Random Waypoint</i> dan <i>Random Direction</i>	<i>Average end to end delay, Packet Delivery Ratio,</i> dan <i>Routing Overhead</i>	OLSR lebih baik dibandingkan <i>routing protocol</i> DSDV namun pada <i>delay</i> DSDV lebih baik dari OLSR.
[6]	Melakukan Studi Analisis kinerja OLSR menggunakan pergerakan <i>node gauss-markov mobility</i> dan <i>random walk in Mobile AdHoc Network</i> (MANET)	OLSR	<i>Random walk</i> dan <i>Gauss-Markov</i>	<i>Packet Delivery Ratio, end-to-end delay</i> dan <i>Routing Overhead</i>	PDR dan <i>delay</i> lebih baik saat menggunakan <i>gauss-markov</i> , sedangkan pada <i>routing overhead</i> pergerakan <i>random walk</i> memiliki nilai lebih baik
[7]	Melakukan Studi kinerja <i>routing protocol</i> AODV, DSR, dan OLSR pada MANET Berdasarkan <i>Quality Of Service</i>	AODV, DSR, dan OLSR	<i>Random Motion Model</i>	<i>Packet Delivery Ratio, Packet loss, Throughput,</i> dan <i>Delay</i>	OLSR Lebih baik dari AODV dan DSR

**Tabel 2.1 Kajian peneliti sebelumnya (lanjutan)**

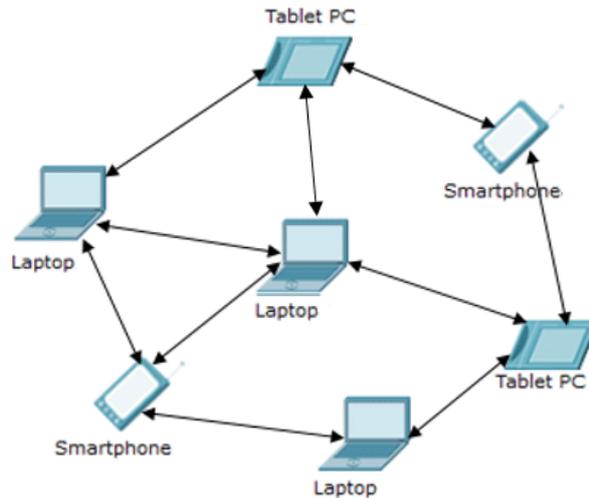
Referensi	Tujuan	<i>Routing</i> Protokol	Pergerakan <i>node</i>	Parameter	Hasil
[8]	Melakukan studi perbandingan <i>routing protocol</i> AOMDV, DSDV, dan ZRP sebagai protokol <i>routing</i> pada <i>mobile ad-hoc network</i> (MANET)	AOMDV, DSDV, Dan ZRP	<i>Random Waypoint</i>	<i>Packet Delivery Ratio, Throughput, End-to-end Delay, dan Normalized Routing Load</i>	AOMDV Lebih baik pada PDR dan <i>throughput</i> , Sedangkan pada <i>end-to-end delay</i> dan <i>normalized routing load</i> DSDV lebih baik.
[9]	Melakukan studi pengaruh <i>model mobilitas</i> terhadap konsumsi energi protokol <i>routing Optimized Link State Routing</i> (OLSR) pada <i>Mobile Ad Hoc Network</i> (MANET)	OLSR	<i>Random Waypoint, Random walk, dan Random Direction</i>	Konsumsi Energi	Total energi sisa pada <i>Random Waypoint</i> sebesar 17,48% sedangkan <i>Random Direction</i> 21,36% dan <i>Random walk</i> 21,05%.

## 2.2 DASAR TEORI

### 2.2.1 MANET

*Mobile Ad Hoc Network* (MANET) adalah sebuah jaringan tanpa kabel yang terdiri dari sejumlah *node mobile* yang dinamis, beroperasi tanpa memerlukan infrastruktur tetap. Dalam jaringan MANET *node* memiliki peran sebagai penerima dan pengirim data dan juga sebagai penunjang *node* lainnya untuk dapat meneruskan paket data kepada *node* lainnya [10]. Dalam MANET, *mobile host* yang terhubung melalui jaringan nirkabel memiliki kemampuan untuk dapat

bergerak secara bebas dan juga bertindak sebagai *router*, *mobile host* bertanggung jawab untuk menemukan dan menentukan rute komunikasi yang digunakan diantara *node-node* dalam jaringan [11]. Pada gambar 2.1 merupakan arsitektur jaringan MANET.



**Gambar 2.1** *Mobile ad-hoc network*

Jaringan MANET sering digunakan pada beberapa sektor seperti sektor komersial, militer dan juga sektor yang bersifat privat. Jaringan MANET memungkinkan pengguna untuk berkomunikasi atau bertukar informasi tanpa perlu memikirkan masalah geografis maupun infrastruktur jaringan. Pada sektor militer jaringan MANET dapat digunakan untuk berkomunikasi antara tentara, kendaraan militer, dan markas militer. Pada sektor komersial jaringan manet dapat digunakan ketika terjadi operasi darurat atau penyelamatan korban jiwa pada saat terjadi bencana. Penyelamat harus dapat terus berkomunikasi ketika melakukan penyelamatan karena kerusakan insfratruktur penggunaan jaringan MANET sering digunakan untuk operasi ini [12].

Adapun karakteristik yang dimiliki manet antara lain adalah:

1. MANET memiliki sifat yang mandiri di mana setiap *node* berfungsi sebagai *router* dan *host*, dan tidak bergantung pada otentikasi terpusat.
2. Sifat terdistribusi pada MANET disesuaikan untuk mengatur konfigurasi *host*, *routing*, dan keamanan, dengan topologi yang dinamis mengatur prosesnya.

3. Kemampuan MANET untuk menambahkan atau menghapus *node* dari jaringan kapan saja, bersama dengan ciri khas *multi-hop*, menjadikannya sistem yang sangat fleksibel.
4. *Routing multi-hop* di MANET memberikan opsi ketika *node* sumber dan *node* tujuan berada di luar jangkauan radio satu sama lain.
5. Tingkat perilaku yang selaras dan impulsif di MANET mengurangi kebutuhan campur tangan manusia, menjadikannya lebih otonom.
6. *Node* dalam MANET menciptakan lokasi asimetris dengan fitur, tanggung jawab, dan kemampuan yang serupa.
7. Dibandingkan dengan jaringan kabel, jaringan nirkabel memiliki stabilitas, keandalan, efisiensi, dan kapasitas yang lebih rendah [13].

### 2.2.2 Protokol *Routing* MANET

Protokol *routing* dalam MANET bertanggung jawab untuk menetapkan cara komunikasi antar *router*, serta mengirimkan informasi yang relevan agar setiap *node* dapat memilih rute komunikasi yang optimal dalam jaringan. Ada beberapa jenis protokol *routing* yang telah mapan dan sering digunakan dalam penelitian, seperti AODV, DSDV, OLSR dan DSR [14]. Protokol *routing* MANET dapat diklasifikasikan menjadi tiga kategori utama: proaktif, reaktif, dan hibrida. Pada *routing protocol* proaktif *table routing* disimpan pada setiap *node* dimana digunakan untuk melacak perubahan topologi dan untuk menyimpan informasi tentang rute yang bisa digunakan untuk mengirim paket. Sedangkan dalam *routing* protokol reaktif tidak ada *table routing*, dan rute dibuat hanya jika diperlukan komunikasi. Dan *routing protocol hybrid* merupakan kombinasi dari fitur terbaik yang ada pada *routing protocol* proaktif dan reaktif.

#### 1. *Routing Protocol* Reaktif

Protokol *Routing* Reaktif, memiliki ciri khusus di mana tidak ada rute tetap yang disimpan. Sebaliknya, rute dibangun hanya pada saat dibutuhkan. Jika jalur ke tujuan tidak tersedia, *cache* diperiksa untuk setiap jalur yang ada; jika tidak, langkah berikutnya adalah mencari rute baru. Beberapa contoh protokol *routing* reaktif adalah *Temporally Ordered Routing Algorithm* (TORA), *Dynamic Source*

*Routing (DSR), Ad Hoc On-demand Distance Vector Routing (AODV), Link Quality Source Routing (LQSR) and Land Mobile Radio (LMR).*

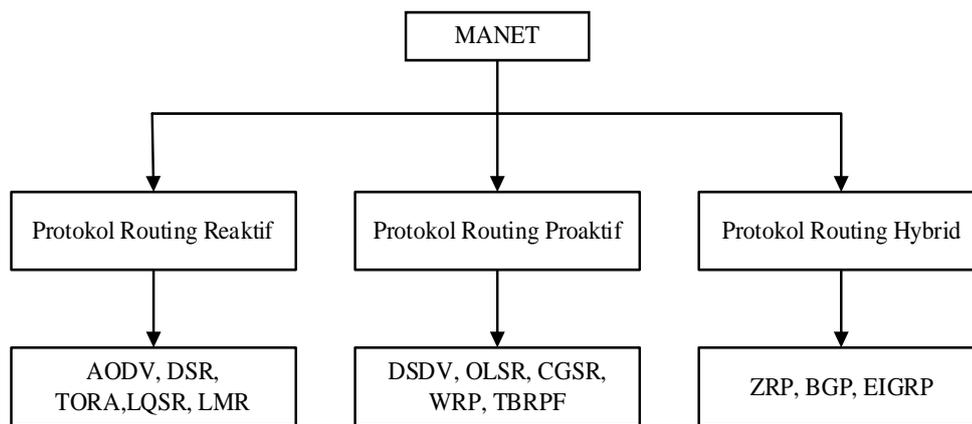
## 2. *Routing Protocol Proaktif*

Protokol proaktif sering disebut sebagai protokol berbasis tabel, melibatkan setiap *node* yang memiliki tabel perutean yang berisi rincian lengkap tentang topologi jaringan. Meskipun fungsi ini berguna untuk mengatur lalu lintas datagram, namun memiliki kelemahan terkait konsumsi daya dan lalu lintas sinyal yang cukup besar. Tabel perutean diperlukan pembaruan setiap kali terjadi perubahan dalam topologi jaringan. Protokol proaktif kurang ideal untuk jaringan besar karena setiap *node* harus menyimpan catatan lengkap dalam tabel perutean, yang dapat memakan waktu lama. Sebaliknya, untuk mengatasi tantangan ini, tabel *routing* terpisah dibuat untuk setiap pasangan protokol. Beberapa protokol *routing* pada *protocol routing* proaktif adalah *Destination-Sequenced Distance Vector Routing (DSDV), Optimized Link State Routing Protocol (OLSR), Cluster-head Gateway Switch Routing Protocol (CGSR), Wireless Routing Protocol (WRP), dan Topology Dissemination Based on Reverse-Forwarding Jalur (TBRPF).*

## 3. *Routing Protocol hybrid*

Protokol *routing hybrid* dapat dianggap sebagai langkah evolusi dalam dunia protokol MANET. Protokol ini menggabungkan fitur-fitur algoritma *routing* reaktif dan proaktif, memberikan fleksibilitas dan kinerja yang dapat diukur. Dibandingkan dengan protokol murni reaktif atau proaktif, protokol *routing hybrid* memiliki keunggulan tertentu. Mereka dapat melakukan pengukuran yang lebih terkontrol, sementara juga memperkenalkan fitur-fitur baru, seperti kemampuan untuk mendeteksi titik kegagalan tunggal dan menciptakan jenis *node* khusus untuk mengatasi masalah kemacetan. Protokol ini memungkinkan transmisi data dari sejumlah *node* tertentu hanya jika rute favorit tidak tersedia, memungkinkan perutean hierarki secara umum. Meskipun memiliki keunggulan, penggunaan protokol *routing hybrid* tidak tanpa tantangan. Salah satu kesulitan utamanya adalah perlunya mereset jaringan berdasarkan kondisi yang ada. Namun, tujuan utama dari protokol ini untuk memanfaatkan keunggulan dari kedua sistem (reaktif dan proaktif) sambil meminimalkan kelemahan mereka. Sebagai contoh, penundaan

dalam identifikasi rute baru dapat diminimalkan dengan mempertahankan beberapa jenis tabel perutean daripada membuatnya berdasarkan permintaan. Beberapa contoh protokol *routing hybrid* meliputi *Border Gateway Protocol (BGP)*, *Zone Routing Protocol (ZRP)*, dan *Enhanced Interior Gateway Routing Protocol (EIGRP)*. Mereka mencerminkan upaya untuk mengintegrasikan keunggulan keduanya, menciptakan solusi yang lebih efektif dan efisien dalam konteks MANET [15]. Pada gambar 2.2 merupakan *routing* protokol pada manet.



**Gambar 2.2** Klasifikasi *routing* protokol pada MANET

Dalam pemilihan *routing* protokol perlu dilihat beberapa faktor yang perlu dipertimbangkan antara lain:

1. *Multicasting* : Kemampuan untuk mengirim paket ke beberapa *node* sekaligus.
2. *Loop - free*: Rute yang ditempuh oleh suatu paket tidak pernah melibatkan *node* perantara yang sama dua kali sebelum mencapai destinasi. Untuk meningkatkan kinerja keseluruhan dalam protokol *routing*, langkah- langkah diambil untuk memastikan bahwa rute yang ditentukan tidak membentuk *loop*. Hal ini bertujuan untuk menghindari pemborosan *bandwidth* dan penggunaan *cpu* yang efisien.
3. *Multiple routes*: Ketika salah satu jalur rusak karena suatu masalah, data harus dapat dikirim melalui jalur lain. Oleh karena itu, protokol harus memungkinkan pembuatan banyak rute.
4. *Distributed operation* : protokol harus dibagikan, tidak boleh bergantung pada *node* terpusat.

5. Keamanan fisik : jaringan rentan terhadap ancaman fisik seperti penyadapan dan serangan pengacauan.
6. Operasi *on-demand*: Protokol harus mampu beradaptasi dengan pola lalu lintas secara *on-demand* atau berdasarkan kebutuhan, sehingga sumber daya dan *bandwidth* dapat digunakan secara lebih efisien.
7. Dukungan *link* unidireksional: Lingkungan radio dapat menyebabkan pembentukan tautan unidireksional. Pemanfaatan tautan ini dan bukan hanya tautan dua arah meningkatkan kinerja protokol *routing* .
8. *Node* masuk/keluar: Protokol *routing* harus dapat dengan cepat beradaptasi dengan masuk atau keluarnya *node* dalam jaringan, tanpa harus merestrukturisasi seluruh jaringan [13].

### 2.2.3 OLSR

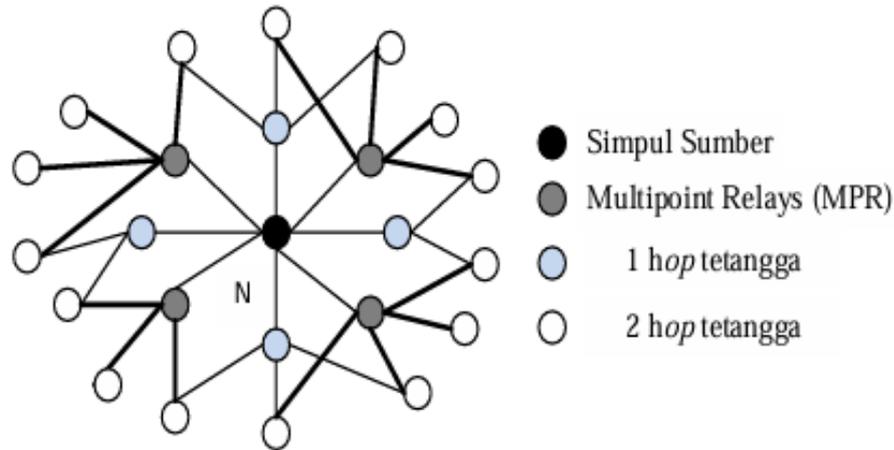
Protokol *routing* proaktif *Optimized Link State Routing* (OLSR) merupakan *routing* yang dikembangkan berdasarkan Algoritma *link state routing* dan menggunakan teknik dalam mengekstrak informasi yang berkaitan dengan topologi jaringan. OLSR memiliki kelebihan dalam hal *delay* karena bersifat *routing table*. Namun, sifatnya yang konstan dalam *update routing table* menyebabkan *routing overhead* [7]. OLSR sendiri merupakan pengembangan dari *routing* sebelumnya yaitu *Link State Routing* (LSR), dengan penambahan *Multipoint Relay* (MPR) untuk proses *packet forwarding*.

Adanya mekanisme MPR pada OLSR membuat paket-paket OLSR yang diterima tidak akan langsung diteruskan, akan tetapi hanya *node-node* MPR yang akan meneruskan paket kontrol yang diterima. Hal ini bertujuan untuk mengurangi adanya penerimaan duplikasi pesan kontrol yang tidak diperlukan yang dapat membanjiri jaringan.

Berikut merupakan urutan cara kerja OLSR :

1. *Link Sensing*
2. *Neighbour Detection*
3. *MPR Selection*
4. Pengiriman pesan *TC*
5. *Route Calculation* [16]

Setiap *node* MPR yang dipilih dihitung sebagai *node* tetangga simetris dan semua *hop* tetangga dapat dijangkau melalui MPR. Mekanisme pemilihan standar MPR umumnya menggunakan algoritma Greedy atau Dijkstra dalam menghitung jumlah MPR yang terbentuk atau memberikan hasil rute terpendek. Pemilihan *node* MPR dengan melihat *node* satu *hop* yang memiliki lebih banyak tetangga dua *hop* yang belum dicakup oleh MPR lain. [17].



**Gambar 2.3 Pemilihan *node* MPR**

Gambar 2.3 menunjukkan bagaimana pemilihan *node* MPR pada OLSR dimulai dengan *node* N sebagai *node* Sumber. *Node* MPR dipilih empat dari delapan *node* yang terhubung dengan satu *hop* tetangga. *Node* sumber dapat berkomunikasi dengan semua tetangga *hop* melalui *node* MPR dan hanya *node* MPR yang akan meneruskan pesan [18].

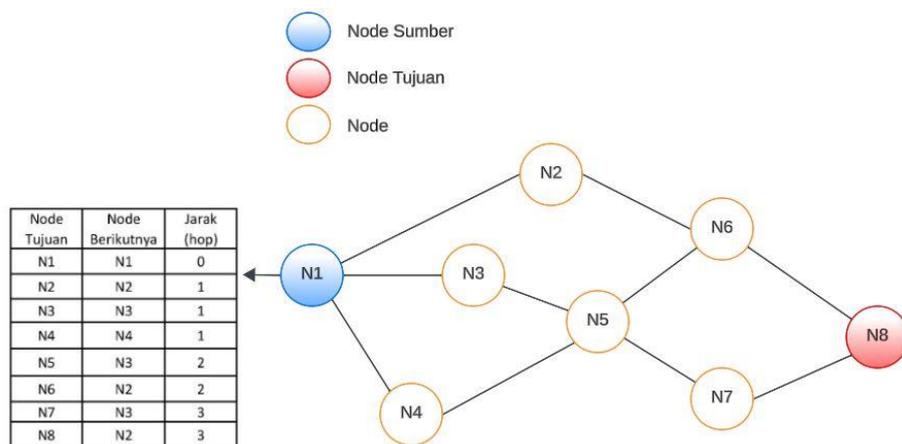
Dalam menentukan rute, OLSR menggunakan dua pesan kontrol utama *Hello message* dan *Topology control messages (TC)*. *Hello messages* memungkinkan setiap *node* untuk mengetahui status *Link State* dan tetangga hingga dua *hop*, informasi ini digunakan oleh setiap *node* untuk menentukan MPR yang akan digunakan komunikasi. sementara pesan TC digunakan untuk memelihara database perutean dengan menyiarkan informasi topologi secara teratur untuk membuat set pemilih MPR mereka. OLSR secara periodik mengoptimalkan kinerjanya dengan mengurangi interval waktu antara pengiriman pesan TC, sehingga cocok untuk digunakan dalam jaringan yang padat dengan menggunakan teknik MPR. Keuntungan dalam penggunaan *routing* protokol OLSR yaitu jalur

*routing* sudah tersedia ketika dibutuhkan karena pembaruan *routing* tabel yang dilakukan secara berkala [19].

### 2.2.4 DSDV

*Destination Sequence Distance vector* (DSDV) adalah protokol *routing* yang merupakan hasil pengembangan dari algoritma *routing bellman-ford*. DSDV termasuk dalam kategori *table driven routing protocol* dalam jaringan manet. DSDV menggunakan metode *routing distance vector*, memungkinkan setiap *node* dalam jaringan untuk dapat bertukar *table routing* melalui *node* tetangganya. DSDV memiliki kelebihan dalam menghindari *loops*, mengurangi *routing overhead* yang tinggi dan mengurangi latensi untuk penemuan rute yang rendah [20].

DSDV merupakan protokol proaktif yang memiliki sifat *table-driven*. Sifat *table-driven* dalam *routing* jaringan berarti setiap *node* memiliki peta lengkap jaringan, memungkinkan mereka untuk menentukan jalur terbaik untuk mentransmisikan data., tiap *node* memiliki informasi untuk mengetahui semua rute ke *node* tetangga atau *node* lain yang berada didalam jaringan tersebut dimana setiap *node* akan selalu mengupdate tabel *routing*nya secara periodik dengan menggunakan *node* yang masuk dan keluar pada suatu jaringan [21].



**Gambar 2.4 Routing tabel DSDV**

Pada gambar 2.4 merupakan tabel *routing* pada *node* N1 *routing protocol* DSDV. Dalam DSDV, informasi perutean disiarkan oleh *node* yang bergerak ke *node* tetangga. Setiap *node* akan menyimpan tabel *routing* yang berisi informasi *node* sekitar, *node* yang dapat dijangkau, dan jumlah *hop*. Dengan kata lain setiap

*node* akan wajib menemukan *node* mereka sendiri pada lingkungan sekitar karena adanya perubahan dalam topologi jaringan. Struktur yang terdapat pada *table routing* DSDV sangat sederhana, karena setiap *entri* tabel mempunyai nomor urut yang dapat bertambah saat pesan pembaruan dikirimkan oleh *node* tersebut. Ketika terjadi perubahan pada topologi jaringan, maka *table routing* akan diperbarui secara periodik [3]. Sebelum mengirimkan pesan untuk pembaruan rute, *node* akan menunggu waktu untuk penyelesain untuk memastikan bahwa *node* tidak mendapatkan pesan pembaruan dari *node* tetangga. Namun, jika salah satu *table routing* telah diperbarui, *node* akan segera memilih rute untuk mencapai *node* tujuan [22]. Pesan yang dikirimkan pada setiap *node* Ketika terjadi perubahan dalam topologi jaringan adalah, alamat tujuan, jumlah *hop* yang diperlukan untuk mencapai tujuan, dan nomor urut informasi yang diterima mengenai tujuan tersebut, sesuai dengan stempel asli dari tujuan tersebut. Keuntungan dalam penggunaan *routing protocol* ini adalah efisiensi dalam menentukan rute, latensi dalam penentuan rute sangat rendah, dan dapat menghindari *loop* [23].

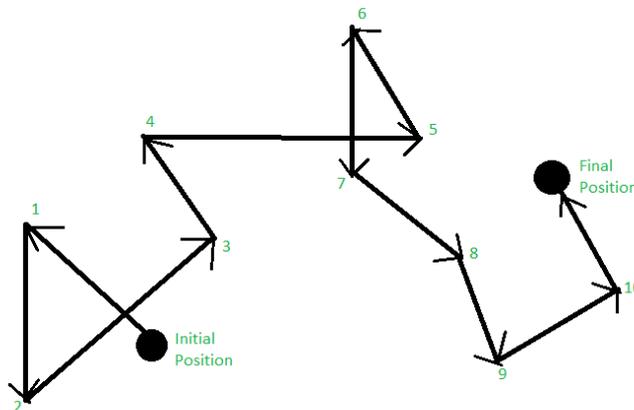
### 2.2.5 Pergerakan *Node*

Pergerakan *node* merupakan salah satu parameter penting yang perlu diperhatikan dalam membangun sebuah simulasi dalam lingkungan jaringan MANET. Pergerakan *node* merupakan teknik dalam simulasi untuk menentukan pola pergerakan *node*, bagaimana letak posisi *node*, percepatan dan perubahan kecepatan *node* setiap waktu. Tujuan dari penggunaan pergerakan *node* atau *mobilitas node* adalah untuk menunjukkan arah, percepatan dan kecepatan *node* yang digunakan pada suatu simulasi. Dalam pemilihan pergerakan *node* sangat penting untuk menyesuaikan dengan penggunaan *routing* protokol yang akan digunakan untuk menganalisis masing - masing *routing* protokol. Pergerakan *node* memiliki beberapa *model* menurut karakteristik pergerakannya seperti *temporal*, *spasial* dan *geografis*. Pada pergerakan *node temporal* riwayat pergerakan memepengaruhi pergerakan *node* yang bergerak. Sedangkan pada pergerakan *node spasial*, *node* bergerak searah. Dan pada *model geografis* kendala dalam pergerakan *node* adalah hambatan seperti jalan raya. Dalam simulasi penggunaan pergerakan *node* acak sering digunakan untuk menentukan kinerja *routing* protokol yang baik. Meskipun

pergerakan tersebut sering mempengaruhi topologi jaringan. Penting untuk mengidentifikasi pergerakan *node* yang sesuai untuk memastikan kesesuaian dengan di dunia nyata [24].

### 2.2.6 *Random walk*

*Random walk* atau biasa dikenal dengan nama pergerakan *Brownian*, dijelaskan secara matematis oleh Einstein pada 1926, untuk menyaingi pergerakan partikel yang tidak dapat diprediksi dalam fisika. Pada MANET beberapa *node* dianggap bergerak secara tiba-tiba. Model pergerakan *random* dibuat untuk meniru pergerakan *node* yang tak dapat diprediksi *random walk model* adalah proses *mobilitas* tanpa memori dimana informasi tentang status masa lalu tidak digunakan untuk pilihan di masa depan. dengan kata lain kecepatan saat ini bersifat otonom dengan kecepatan masa lalunya dan kecepatan di masa mendatang juga tidak bergantung pada kecepatannya saat ini [4].



**Gambar 2.5** Pergerakan *random walk*

Pada gambar 2.5 merupakan pergerakan *random walk*. Pergerakan *random walk* juga merupakan pergerakan *node* dimana *node* akan bergerak secara acak dimana *node* akan bergerak dari posisi awal ke posisi baru dengan laju dan tujuan dengan acak. Selain laju dan arah yang sudah ditentukan. Tujuan baru akan ditentukan setelah *node* bergerak dalam waktu dan jarak yang telah ditentukan. Karena beberapa *node* bergerak dianggap bergerak secara tiba-tiba, *random walk* dibuat agar menyesuaikan pergerakan *node* yang tidak dapat diprediksi [9].

### 2.2.7 Packet Delivery Ratio (PDR)

PDR merupakan salah satu parameter QoS yang penting untuk dipertimbangkan dalam pemilihan protokol *routing*. Pada tabel 3.1 merupakan nilai Batasan PDR.

**Tabel 3.1 Nilai batasan PDR [25]**

<b>PDR (%)</b>	<b>Kategori</b>
95% - 100%	Sangat Bagus
80% - 95%	Bagus
50% - 80%	Sedang
< 50%	Jelek

PDR metrik penting dalam jaringan komputer yang mengukur persentase paket data yang berhasil diterima oleh *node* tujuan. Nilai PDR yang tinggi menunjukkan bahwa protokol *routing* yang digunakan efektif dalam mengantarkan paket data ke tujuannya. PDR dihitung dengan membagi jumlah paket yang diterima dengan jumlah paket yang dikirim. Rumus PDR dapat dilihat pada persamaan 2.1.

$$PDR = \frac{\text{paket yang diterima}}{\text{paket yang dikirim}} \times 100\% \quad (2.1)$$

### 2.2.8 Throughput

*Throughput* merupakan metrik penting dalam jaringan komputer yang mengukur kecepatan transfer data dalam satuan *bit per second* (bps). Nilai *throughput* yang tinggi menunjukkan bahwa jaringan mampu mengirimkan data dengan cepat dan efisien. Besarnya selang waktu dalam pengukuran dapat mempengaruhi hasil gambaran perilaku jaringan. *Throughput* dapat dihitung dengan membagi jumlah data yang diterima dengan waktu yang dibutuhkan untuk mentransfer data tersebut. Rumus *throughput* dapat dilihat pada persamaan 2.2 [25].

$$\text{Throughput} = \frac{\text{jumlah data yang diterima (bit)}}{\text{waktu pengiriman paket (sec)}} \text{ kbps} \quad (2.2)$$

### 2.2.9 Delay

*Delay* merupakan metrik penting dalam jaringan komputer yang mengukur waktu yang diperlukan untuk mengirimkan data dari pengirim ke penerima, diukur dalam satuan detik (s).

**Tabel 3.2 Nilai batasan *delay* standar TIPHON [26]**

<b><i>Delay</i> (ms)</b>	<b>Kategori</b>
< 150 ms	Sangat Bagus
151 ms – 300 ms	Bagus
301 ms – 450 ms	Sedang
> 450 ms	Jelek

Nilai *delay* yang rendah menunjukkan bahwa jaringan mampu mengirimkan data dengan cepat dan responsif. Pada tabel 3.2 merupakan nilai batasan *delay* menurut standar TIPHON. Waktu yang diperlukan *routing* dalam memilih ataupun menemukan jalur akan digunakan saat pengiriman data merupakan faktor yang dapat mempengaruhi *delay*. Rumus *delay* dapat dilihat pada persamaan 2.3.

$$Delay = \frac{\text{waktu pengiriman}}{\text{jumlah paket yang diterima}} \text{ ms} \quad (2.3)$$

### 2.2.10 Packet loss

*Packet loss* merupakan metrik penting dalam jaringan komputer yang mengukur banyaknya paket data yang hilang selama proses pengiriman. Pada tabel 3.2 merupakan nilai batasan yang digunakan menurut standar TIPHON.

**Tabel 3.2 Nilai batasan *packet loss* standar TIPHON [26]**

<b><i>Packet loss</i> (%)</b>	<b>Kategori</b>
0% - 2%	Sangat Bagus
3% - 14%	Bagus
15% - 24%	Sedang
> 25%	Jelek

Nilai *packet loss* yang tinggi menunjukkan bahwa jaringan mengalami masalah dan tidak dapat mengirimkan data dengan handal. *Packet loss* dapat dihitung dengan membagi jumlah paket yang hilang dengan jumlah paket yang dikirim. Rumus *packet loss* dapat dilihat pada persamaan 2.4.

$$Packet Loss = \frac{Paket\ yang\ hilang}{paket\ yang\ dikirim} \times 100\% \quad (2.4) [26]$$