

BAB II

LANDASAN TEORI

2.1. Studi Literatur

Pada penelitian sebelumnya yang juga membahas perbandingan performa protokol OLSR dan DSDV, yang berjudul Analisis Perbandingan Kinerja Protokol Routing DSDV dan OLSR Untuk Perubahan Kecepatan Mobilitas pada Standar IEEE 802.11ah (Irfan et al., 2016). Tujuan dari penelitian tersebut adalah untuk mengevaluasi routing protokol OLSR dan DSDV pada standar IEEE 802.11ah dengan menggunakan NS3. Skenario yang digunakan pada penelitian tersebut adalah perubahan kecepatan mobilitas pada setiap *station*, pada skenario ini rentan kecepatan mobilitas yang digunakan adalah 0 - 0,6 m/s, 0,6 - 1,2 m/s, 1,2 - 1,8 m/s, 1,8 - 2,4 m/s, 2,4 - 3 m/s. Penelitian tersebut menganalisis dengan parameter *QoS throughput*, *packet delivery ratio*, *delay*, dan konsumsi energi.

Berdasarkan hasil simulasi pada penelitian tersebut, didapat kesimpulan bahwa routing protokol OLSR memiliki kinerja yang lebih baik dibandingkan routing protokol DSDV pada skenario perubahan kecepatan. Nilai rata-rata *throughput* untuk OLSR adalah 28400 *Bps* sedangkan untuk DSDV adalah 2934 *Bps*. Nilai rata-rata PDR untuk OLSR adalah 14,582% sedangkan untuk DSDV adalah 2,7%. Nilai rata-rata *delay* untuk OLSR adalah 0,04453994 *s* sedangkan untuk DSDV adalah 0,6261986 *s*. Sedangkan rasio perbandingan untuk konsumsi energi antara routing protokol OLSR dan DSDV adalah 1,48% untuk skenario perubahan kecepatan. Diketahui juga pada penelitian tersebut lebih berfokus pada mengevaluasi routing protokol DSDV dan OLSR pada standar IEEE 802.11ah.

2.1.1. Perbandingan Penelitian

Pada penelitian ini akan lebih berfokus untuk membandingkan performa dari routing protokol OLSR dan DSDV secara umum, tidak terfokus pada implementasi terhadap jenis *wireless* tertentu dan juga pada penelitian ini akan menggunakan jenis *wireless* yang berbeda, yaitu standar IEEE 802.11b.

Berdasarkan parameter *throughput*, *packet delivery ratio*, *packet loss* dan *delay* yang akan dijalankan pada dua skenario, yaitu skenario penambahan node dan penambahan ukuran paket data. Beberapa perbandingan parameter antara penelitian sebelumnya dengan penelitian ini, dapat dilihat pada tabel 2.1.

Tabel 2.1. Perbandingan parameter

Parameters	Penelitian Sebelumnya	Penelitian Ini
Routing Protokol	OLSR/DSDV	OLSR/DSDV
<i>Wireless</i>	IEEE 802.11ah	IEEE 802.11b
Parameter <i>QoS</i>	<i>throughput</i> , <i>packet delivery ratio</i> , <i>delay</i> , dan konsumsi energi	<i>throughput</i> , <i>packet delivery ratio</i> , <i>packet loss</i> dan <i>delay</i>
Skenario	<ul style="list-style-type: none"> •Perubahan kecepatan mobilitas pada setiap STA 	<ul style="list-style-type: none"> •Penambahan <i>node</i> •Penambahan ukuran paket data
Transport Layer	UDP	UDP
Data Rate	1,2 Mbps	11 Mbps
Ukuran Paket Data	100 <i>bytes</i>	64 <i>bytes</i> dan 128 <i>bytes</i>

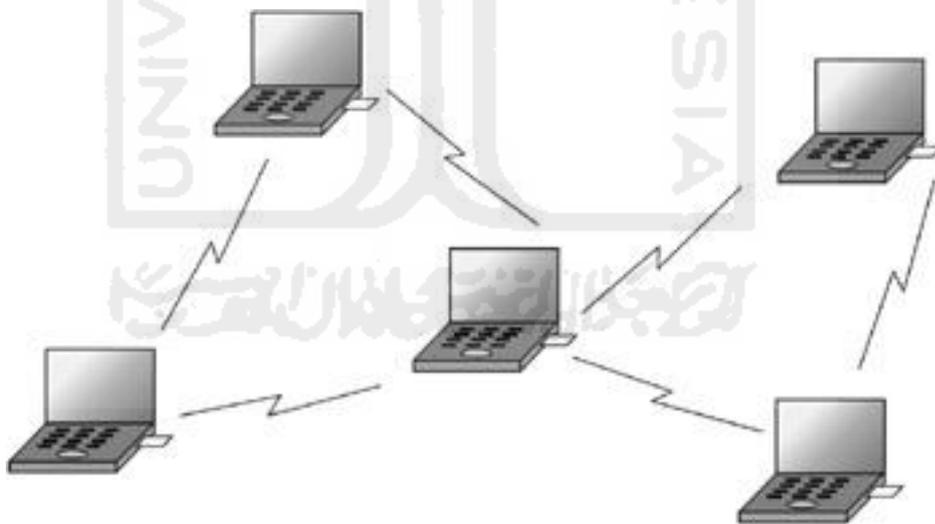
2.2. Jaringan Nirkabel (*Wireless*)

Jaringan nirkabel adalah bidang disiplin yang berkaitan dengan komunikasi antar sistem komputer tanpa menggunakan kabel. Jaringan nirkabel ini sering dipakai untuk jaringan komputer baik pada jarak yang dekat maupun pada jarak jauh (lewat satelit). Bidang ini erat hubungannya dengan bidang telekomunikasi, teknologi informasi, dan teknik komputer. Jenis jaringan yang populer dalam kategori jaringan nirkabel ini meliputi jaringan kawasan lokal nirkabel (*wireless LAN/WLAN*), dan *Wi-Fi*. Jaringan nirkabel biasanya menghubungkan satu sistem komputer dengan sistem yang lain dengan menggunakan beberapa macam media transmisi tanpa kabel, seperti gelombang radio, gelombang mikro, maupun cahaya infra merah (id.wikipedia.org, 2016). Beberapa kategori yang dimiliki jaringan

nirkabel adalah *Wireless Wide Area Networks*, *Wireless Metropolitan Area Networks*, *Wireless Local Area Networks*, *Wireless Personal Area Networks*, *Peer to Peer Network/Ad-Hoc Wireless LAN* (Jiatmiko & Prayudi, 2016).

2.3. *Mobile Ad Hoc Network (MANET)*

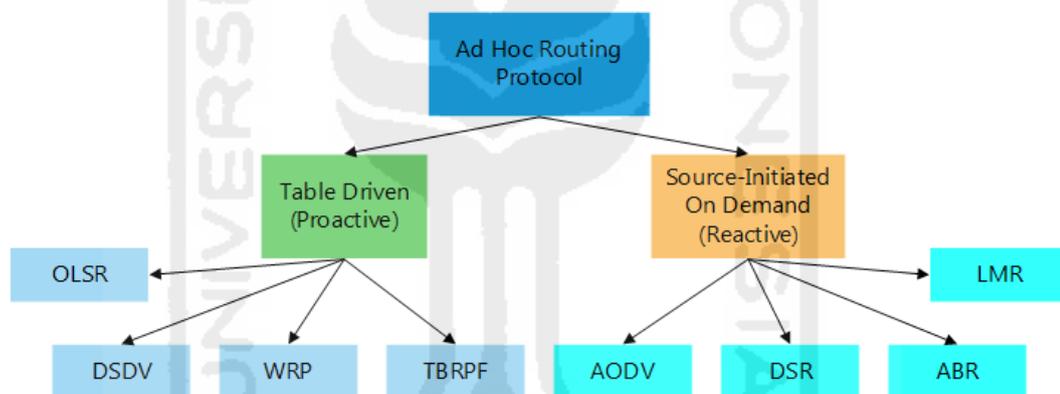
Mobile Ad Hoc Network (MANET) adalah salah satu jenis teknologi *wireless* yang terdiri dari kumpulan *node-node (device)* bergerak dan bersifat dinamis yang bekerja tanpa menggunakan infrastruktur atau administrasi terpusat. *Node* pada MANET dapat bertindak sebagai *user* dan router, *node* bertindak sebagai *user* saat mengirim dan menerima paket dan *node* bertindak sebagai router saat membangun rute dan meneruskan paket ke *node* lainnya. MANET tidak memerlukan dukungan *back-bone* infrastruktur sehingga mudah diimplementasikan dan sangat berguna ketika infrastruktur tidak ada ataupun tidak berfungsi lagi (Mulyanta, 2005). Contoh gambar dari MANET dapat dilihat pada gambar 2.1.



Gambar 2.1. MANET (globalspec.com, 2008)

2.4. Routing Protokol

Routing tak lain adalah untuk menentukan arah paket data dari satu jaringan ke jaringan lain. Penentuan arah ini disebut juga sebagai route, routing dapat diberikan secara dinamis (*dynamic routing*) atau secara statis (*static routing*) (ICT, 2005). Protokol adalah aturan-aturan dan konversi yang menentukan bagaimana sebuah *device* di dalam jaringan komputer untuk saling berkomunikasi. Sehingga routing protokol diperlukan untuk mengatur bagaimana router berkomunikasi antara satu dengan yang lain dalam menyebarkan informasi, yang memungkinkan router untuk memilih rute pada jaringan komputer (Kopp, 1999). Pada jaringan *ad-hoc*, routing protokol dibagi menjadi dua jenis yaitu *Table Driven Routing Protocol* (Proaktif) dan *One-Demand Routing Protocol* (Reaktif).



Gambar 2.2. Routing protokol

2.4.1. Table Driven Routing Protocol (Proaktif)

Pada *Table Driven Routing Protocol* (Proaktif), masing-masing *node* memiliki tabel routing yang lengkap. Artinya sebuah *node* akan mengetahui semua rute ke *node* lain yang berada dalam jaringan tersebut. Setiap *node* akan melakukan pembaruan tabel routing yang dimilikinya secara periodik sehingga perubahan topologi jaringan dapat diketahui setiap interval waktu tersebut. Contoh *table driven routing* adalah OLSR (*Optimized Link State Routing*), DSDV (*Destination Sequenced Distance Vector Routing*), WRP (*Wireless Routing Protocol*) (Sidharta & Widjaja, 2013).

2.4.1.1. OLSR (*Optimized Link State Routing*)

OLSR merupakan routing protokol yang mewarisi algoritma *link state* dan khusus digunakan untuk MANET (Jacquet et al., 2001). Pada OLSR tabel route selalu diperbarui dengan cara mengirim pesan control secara berkala, hal tersebut membuat beban jaringan bertambah. Untuk mengurangi beban jaring OLSR menggunakan mekanisme MPR (*multipoint relay*). Konsep MPR adalah mengurangi pesan broadcast yang memiliki isi informasi yang sama dan mengurangi routing overhead. MPR adalah *node* tetangga yang dipilih oleh suatu *node* dengan spesifikasi tertentu. *Node* ini digunakan untuk memecah paket kemudian setiap *node* MPR memilih sekumpulan node dari tetangga satu *hop*-nya yang digunakan untuk mentransmisikan paket broadcast. Setiap *node* dari MPR memelihara sebuah tabel untuk mengidentifikasi *node* yang akan mengirimkan paket (Mapa, Djanali, & Shiddiqi, 2014). Keuntungan dari penggunaan OLSR adalah dapat mengoptimalkan penggunaan bandwidth yang tersedia (Jacquet et al., 2001).

2.4.1.2. DSDV (*Destination Sequenced Distance Vector Routing*)

DSDV adalah routing protokol konvensional yang diadopsi dari *routing information protocol* (RIP) untuk digunakan pada jaringan *ad hoc*. DSDV menambahkan atribut baru, yaitu *sequence number* untuk setiap entri tabel rute yang masuk. Dengan menggunakan *sequence number*, node dapat membedakan informasi yang kedaluwarsa dan yang baru dengan demikian mencegah pembentukan *routing loops*. Pada DSDV, setiap node di dalam jaringan *ad hoc* harus menyimpan dan menjaga tabel routing yang berisi seluruh alamat tujuan, *metric*, *hop* selanjutnya, dan *sequence number* (He, 2002). Tabel rute pada DSDV akan ditransmisikan dengan cara *broadcast* secara berkala untuk menjaga konsistensi.

2.5. Bahasa Pemrograman C++

Setiap masalah yang akan diselesaikan menggunakan komputer harus dituliskan kedalam sebuah kode program dengan bahasa pemrograman tertentu. Bahasa pemrograman tersebut akan diimplementasikan pada komputer yang akan

menjalankan setiap program yang ditulis dalam satu bahasa. Dari sekian banyak bahasa pemrograman yang ada, pada penelitian ini akan menggunakan bahasa pemrograman C++.

Bjarne Stroustrup pada Bell Labs pertama kali mengembangkan C++ pada awal tahun 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (*low level coding*). Pada C++ ditambahkan konsep-konsep baru seperti *class* dengan sifat-sifatnya seperti *inheritance* dan *overloading*. Selain itu, C++ merupakan bahasa pemrograman yang memiliki sifat *Object Oriented Programming* (OPP) (id.wikipedia.org, n.d.). Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan mendefinisikan *class-class* yang merupakan *a.-class* yang dibuat sebelumnya sebagai abstraksi dari objek-objek fisik. *Class* tersebut berisi keadaan objek, anggota-anggotanya, dan kemampuan dari objeknya. Setelah beberapa *Class* dibuat, masalah dipecahkan menggunakan *class* (Al Fatta, 2007).

2.6. Network Simulator 3 (NS3)

NS3 adalah sebuah aplikasi simulasi jaringan yang terutama fokus untuk kegiatan penelitian dan pendidikan. NS3 merupakan aplikasi open source yang di bawah lisensi *GNU GPLv2* dan tersedia untuk penelitian dan pengembangan. Simulasi jaringan yang dibuat di NS3 bersifat *real-time* dan hampir sama dengan kejadian nyata (nsgn.org, 2011). Pembuatan simulasi menggunakan NS3 memudahkan untuk menganalisa dan mengevaluasi hasil terhadap suatu model jaringan karena representasi hasil data simulasi dapat ditampilkan dalam bentuk grafik (Jiatmiko & Prayudi, 2016). Aplikasi NS3 ditulis sepenuhnya dalam bahasa C++ dan *Python* sebagai *optional binding*.

2.7. Parameter Kinerja Jaringan

Quality of Service (QoS) adalah parameter yang dibutuhkan untuk mengukur kinerja jaringan. Dengan melihat kinerja jaringan dapat menganalisis konsistensi jaringan, tingkat keberhasilan pengiriman data, waktu *delay* data dan lain-lain. Adapun beberapa parameter yang digunakan sebagai pengukur kinerja

jaringan, sebagai berikut:

2.7.1. *Throughput*

Throughput adalah laju data aktual per satuan waktu. *Throughput* dapat dikatakan sebagai *bandwidth* dalam kondisi sebenarnya (Sidharta & Widjaja, 2013). *Bandwidth* dan *throughput* memiliki perbedaan pada sifat laju datanya. *Bandwidth* bersifat tetap, sementara *throughput* bersifat dinamis atau tidak tetap karena tergantung pada *traffic*. Satuan dari *throughput* adalah Kbps (*Kilo Bit per second*). Rumus untuk menghitung *throughput* adalah:

$$\textit{Throughput} = \frac{\textit{ukuran data yang diterima}}{\textit{waktu pengiriman data}} \quad (2.1)$$

2.7.2. *Packet Delivery Ratio*

Packet Delivery Ratio (PDR) adalah rasio antara banyaknya paket yang diterima oleh tujuan dengan banyaknya paket yang dikirim oleh sumber (Sidharta & Widjaja, 2013). Rumus untuk menghitung PDR adalah:

$$\textit{PDR} = \frac{\textit{paket yang diterima}}{\textit{paket yang dikirim}} \times 100\% \quad (2.2)$$

2.7.3. *Packet Loss*

Packet Loss adalah banyaknya paket yang hilang selama proses pengiriman paket dari *node* asal ke *node* tujuan (Sidharta & Widjaja, 2013). Rumus untuk menghitung *packet loss* adalah:

$$\textit{PL} = \frac{\textit{paket yang dikirim} - \textit{paket yang diterima}}{\textit{paket yang dikirim}} \times 100\% \quad (2.3)$$

2.7.4. *Delay*

Delay adalah jeda waktu antara paket dikirim dengan diterimanya paket tersebut ke tujuan. Rumus untuk menghitung *delay* adalah:

$$\textit{Delay} = \textit{waktu data diterima} - \textit{waktu data dikirim} \quad (2.4)$$