

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Sebelumnya

Penelitian yang membahas tentang protokol *routing* sudah banyak dilakukan, mulai dari pengimplementasiannya menggunakan simulator hingga menggunakan perangkat keras.

Penelitian berjudul “Analisis Performansi Routing Protokol DSR, DSDV dan ZRP pada Manet Menggunakan Network Simulator 2” membandingkan protokol *routing* ZRP dengan DSR dan DSDV. Pada penelitian ini mengamati performa dari rata – rata *end to end delay*, *packet delivery ratio*, dan *routing overhead*. Pengamatan dilakukan dengan menggunakan *Network Simulator-2* (NS-2) [5].

Penelitian yang berjudul “Analisis Performa Routing Protokol DSR dan DSDV Menggunakan NS-3 pada Mobile Ad-Hoc Network Manet” membandingkan performa antara protokol DSR dan DSDV berdasarkan jumlah dari jumlah *node*. menggunakan metode simulasi dengan tahapan *problem formulation, conceptual model input/output data, modelling, simulation verification and validation, experimentation, and output analysis*. Penelitian ini menggunakan NS-3 sebagai simulator jaringannya [6].

Tabel 2.1 menampilkan persamaan dan perbedaan antara penelitian saat ini dan penelitian sebelumnya..

Table 2.1 Penelitian Sebelumnya

No	Judul	Penulis	Permasalahan	Metode yang diusulkan	Solusi/Hasil
1	Analisis Performansi Routing Protokol DSR, DSDV dan ZRP pada Manet Menggunakan	ISMAIL, A (2019)	Kurangnya pengetahuan tentang bagaimana setiap jenis protokol routing bekerja dalam keadaan	Metode analisis performansi dengan menggunakan aplikasi <i>Network Simulator-2</i>	Skenario protokol routing DSDV dengan 10 node menghasilkan nilai parameter penggunaan energi terbaik dengan 0,9701 joule. Dapatkan hasil 0 m/s untuk parameter

No	Judul	Penulis	Permasalahan	Metode yang diusulkan	Solusi/Hasil
	Network Simulator 2		lingkungan jaringan tertentu		delay. Sebaliknya, kasus dengan 100 node menghasilkan nilai 4,6707% untuk parameter perutean overhead. ZRP memiliki nilai terbesar dalam hasil parameter Packet Delivery Ratio (PDR), dengan hasil 100% untuk skenario dengan 20, 50, dan 100 node. [5]
2	Analisis Performa Routing Protokol DSR dan DSDV Menggunakan NS-3 pada Mobile Ad-Hoc Network Manet	Fiade, A Suseno, H B (2018)	Menganalisis cara yang paling efektif untuk mengirim paket data dari sumber ke tujuan.	Metode analisis performansi dengan menggunakan aplikasi NS-3	Hasil dari penelitian ini menunjukkan bahwa Protokol <i>routing</i> DSR lebih unggul dari protocol <i>routing</i> DSDV, protocol <i>routing</i> DSR menghasilkan throughput yang lebih besar dibandingkan protokol <i>routing</i> DSDV [6].
3	Simulasi Dan Analisis Perbandingan Kinerja Routing Protocol Ad Hoc On Demand Distance Vector (AODV) dan Dynamic Source Routing (DSR) Saat Melakukan Data Broadcast Storm pada Jaringan Manet	Laksono, F F (2018)	Jaringan membutuhkan sistem routing dengan kinerja yang lebih baik karena adanya gangguan broadcast storm.	Metode analisis performansi dengan menggunakan <i>software</i> OPNET modeler 14.5.	Penelitian ini menunjukkan ketika masalah broadcast storm muncul, protokol AODV lebih baik daripada DSR [7].
4	Analisis Optimasi Kinerja Protokol Routing AODV dan AOMDV dengan Menggunakan Metode RFAP untuk Mencegah RREQ Flooding	Jatmika A, Yudistiana I, Zubaidi A (2019)	Terjadinya serangan banjir RREQ pada jaringan MANET	Metode analisis performansi dengan menggunakan metode RFAP untuk mencegah RREQ <i>Flooding Attacks</i> menggunakan	Hasil dari penerapan metode RFAP pada protokol <i>routing</i> AODV dapat menghasilkan kualitas jaringan yang lebih baik dari pada protokol AOMDV, baik dari segi <i>throughput</i> , <i>average end-to-end delay</i> dan

No	Judul	Penulis	Permasalahan	Metode yang diusulkan	Solusi/Hasil
	Attacks pada Jaringan MANET			software OPNET modeler 14.5.	normalized routing load. [8]
5	Analisis Kinerja Protokol Routing OLSR dan TORA pada Jaringan Manet Menggunakan NS2	Erlan, A C (2018)	Kurangnya pengetahuan tentang kinerja masing - masing protokol	Metode analisis performansi dengan menggunakan aplikasi NS-2	Hasil pengujian menunjukkan bahwa, jika dibandingkan dengan TORA, nilai kinerja OLSR lebih tinggi dalam hal <i>throughput</i> , latensi, dan PDR. OLSR sedikit lebih unggul dari TORA dalam hal <i>throughput</i> dan PDR pada area yang luas (1500m x 1500m) dan sejumlah kecil node (25). [9]

2.2. Dasar Teori

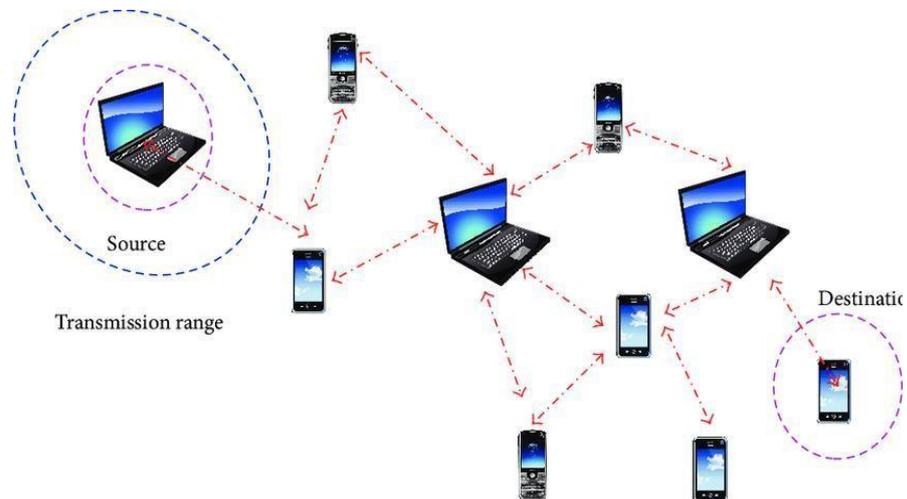
Sub bab ini akan menjelaskan referensi dasar teori dari teknologi dan komponen yang digunakan dalam Tugas Akhir.

2.2.1. MANET

2.2.1.1. Definisi MANET

Jaringan yang tidak menggunakan infrastruktur yang sedang berlangsung disebut *Mobile Ad-Hoc Network* (MANET). Karena tingginya tingkat mobilitas node, MANET tidak memerlukan infrastruktur permanen seperti topologi jaringan lainnya. MANET adalah jaringan nirkabel berbasis IEEE 802.11. [3]. Setiap node seluler di MANET memiliki antarmuka jaringan nirkabel, dan semuanya menggunakan media transmisi untuk berbicara satu sama lain. Mobile node terhubung secara nirkabel dapat dengan mudah berpindah antar node karena daya pancar terbatas media transmisi [10], yang memungkinkan node untuk berfungsi ganda sebagai router, yang bertugas menentukan dan mengelola rute ke setiap node dalam jaringan [11].

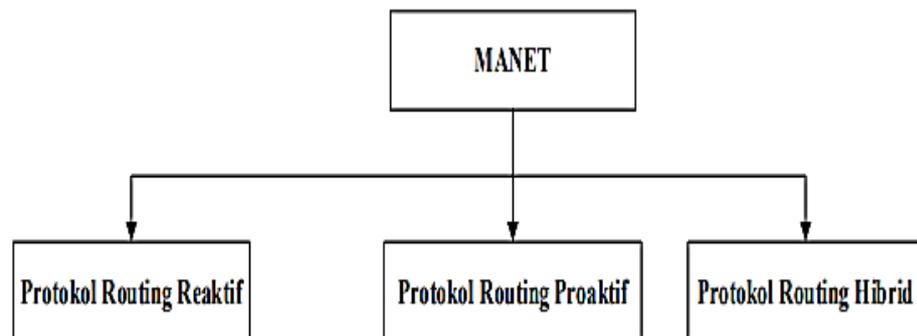
Gambar 2.1 menunjukkan bentuk dari jaringan MANET ketika melakukan komunikasi antar *node*.



Gambar 2.1 Jaringan Manet [2]

2.2.1.2. Klasifikasi MANET

Teknik routing reaktif, proaktif, dan hybrid membentuk tiga kategori utama protokol routing yang digunakan pada jaringan MANET. Kategorisasi jaringan MANET ditampilkan pada Gambar 2.2.



Gambar 2.2 Klasifikasi Manet [12]

a. Protokol Reaktif

Protokol ini juga disebut sebagai protokol *routing on-demand* karena tidak melacak aktivitas *routing* atau data pada node ketika tidak ada komunikasi antar-*node*. Dengan mengirimkan permintaan rute ke semua tetangga jaringan MANET, jalur optimal ditemukan [13].

b. Protokol Proaktif

Tabel routing dibuat oleh protokol routing proaktif, yang kemudian memperbarui routing setiap saat dalam jangka waktu yang telah ditentukan. Tabel perutean mencakup informasi tentang perutean ke node tujuan dan node lainnya [14].

c. Protokol *Hybrid*

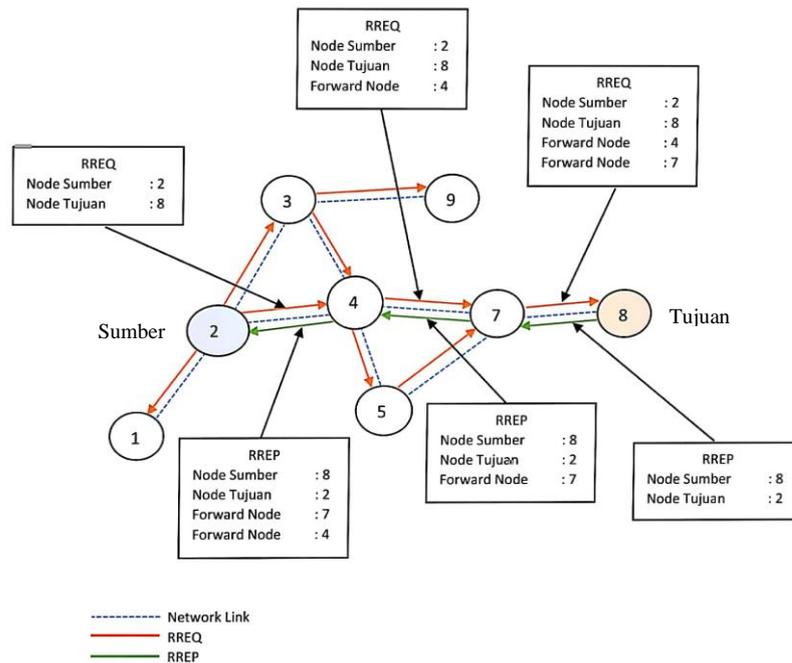
Protokol routing reaktif dan protokol routing proaktif digabungkan untuk membentuk protokol routing hybrid [2].

2.2.2. *Dynamic Manet On-Demand (DYMO)*

Protokol reaktif seperti *Dynamic Manet On-Demand (DYMO)* memenuhi jaringan dengan paket permintaan saat mencoba menemukan rute. Fitur operasi *Route Discovery* dan *Route Maintenance* digunakan oleh protokol DYMO. Saat mengirim paket yang rute targetnya tidak termasuk dalam database routing, node sumber akan menyiarkan *Route Request (RREQ)* ke sisa jaringan untuk menemukan node tujuan [15].

2.2.2.1. *Rute Discovery*

Gambar 2.3 menggambarkan prosedur penemuan rute. Ketika DYMO ingin mentransfer paket data dari node 2 tetapi tidak memiliki rute penerusan dalam tabel *routing* ke alamat *node 8*, *node 2* mengirim RREQ untuk mengidentifikasi rute ke *node 8*. *Node 8* bereaksi terhadap RREQ dengan mengirimkan *Route Reply (RREP)* ke *node 2* setelah menerima RREQ. Setiap *node* yang dilalui RREP akan merekam rute ke *node 8* di log rutanya. Akibatnya, rute antara *node 2* dan *node 8* telah terbentuk ketika *node 2* mendapat RREP. *Node 2* dapat melepaskan rute yang ditetapkan sebelumnya ketika akan mengirim paket data ke *node 8*.



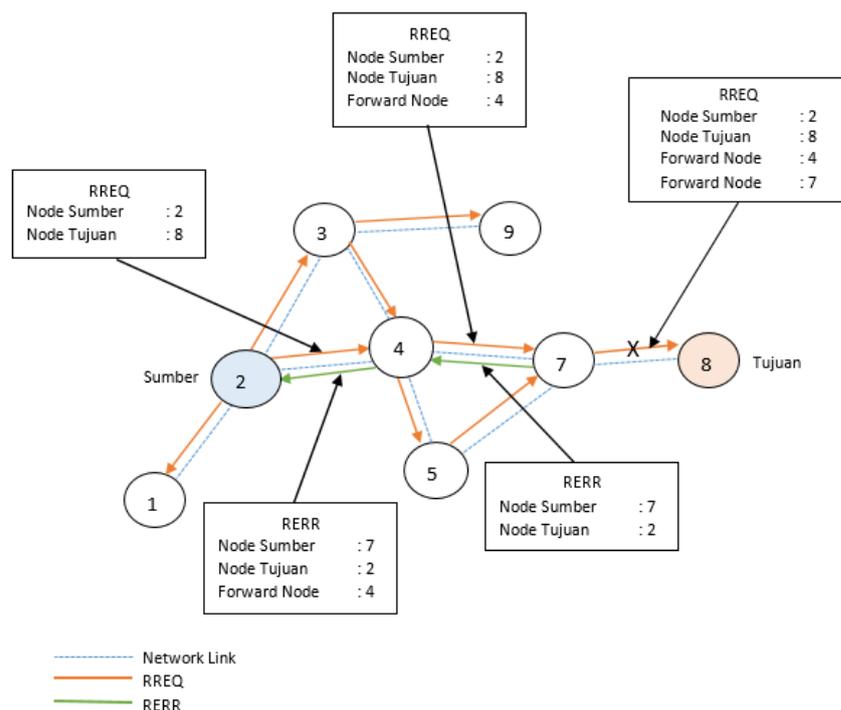
Gambar 2.3 Route Discovery [3]

Protokol DYMO menggunakan *sequence number* untuk menentukan jumlah iterasi dari *Route Message* (RM) yang telah dikirim. *Node* yang meneruskan RREQ akan menambahkan informasi baru ke tabel *routing*, seperti alamat, informasi gateway, dan *sequence number*. *Sequence number* sangat membantu untuk mencegah jalur yang rusak karena akan terus meningkat hingga jalur diambil [15].

2.2.2.2. Rute Maintenance

Rute *Maintenance* adalah rute pertama dibangun dengan prosedur bereaksi terhadap perubahan topologi. Bidang *entri Valid Timeout database routing* diperbarui oleh node saat mengirim dan menerima paket data, dan terus memantau tautan aktif untuk mempertahankan rute. Jika sebuah node mendapatkan paket data dengan rute yang salah sebagai tujuannya, ia harus merespons dengan mengirimkan pesan *Route Error* (RERR). Pesan RRR mencantumkan alamat dan urutan jumlah node yang tidak dapat diakses, node per node. Selain itu, setiap entri tabel perutean yang menentukan tujuan sebagai item hop berikutnya dimasukkan oleh node. Tujuannya adalah untuk mengingatkan pengguna tentang rute baru yang telah dihentikan. Daftar rute ini dikirim oleh node dalam paket ERR.

Penyebaran RERR dilihat pada gambar 2.4. ketika tautan antar *node 7* dan *node 8* terputus, maka *node 7* akan menerima paket yang seharusnya tertuju ke *node 8*. Penyebab terjadinya tautan tersebut rusak dikarenakan *entry* yang tidak valid pada tabel *routing*. Karena paket tidak dapat diteruskan ke tujuan, *Node 7* sebagai *node* terakhir yang menerima paket akan mengirim pesan RERR menuju *node 2*.



Gambar 2.4 *Route Maintenance* [3]

Node yang mendapat RERR memeriksa daftar node pesan terhadap entri terkait dalam tabel routing. Jika node RERR yang diterima dan node hop berikutnya cocok dan ada lebih banyak entri urutan daripada nomor urut di RERR, maka setiap entri tabel rute pada node tersebut akan diabaikan. Untuk menjaga jalur tetap hidup setiap saat, node harus terus mencoba mengidentifikasi kegagalan koneksi dan mengambil tindakan setelah menerima paket ke tujuan tanpa tabel entri rute yang valid [16][17].

2.2.2.3. *Path Accumulation*

Path accumulation merupakan fitur penting pada protokol routing DYMO yang memungkinkan setiap *node* di jaringan MANET untuk menyimpan informasi

tentang rute yang ditemukan oleh *node* lain. Dengan adanya *path accumulation*, setiap *node* di jaringan dapat menggunakan informasi tersebut untuk memperbarui tabel rute dan melakukan *routing* dengan lebih efisien

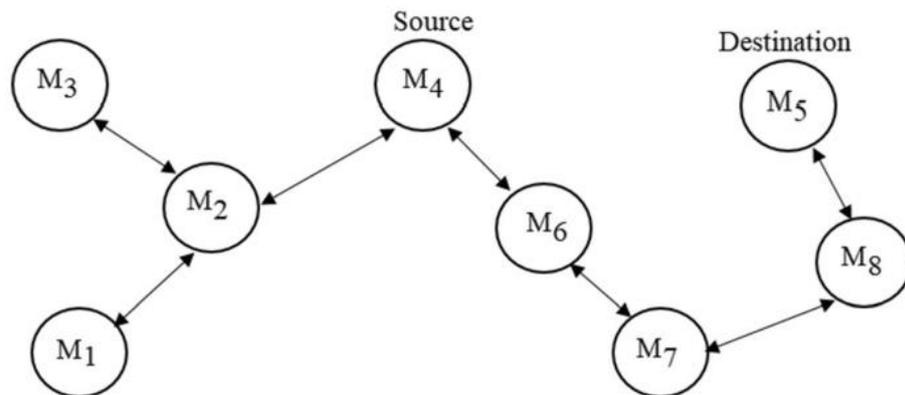
Setiap *node* pada protokol DYMO secara periodik mengirimkan pesan RREQ untuk mencari rute ke tujuan. Ketika *node* menerima RREQ, ia akan mengevaluasi informasi tentang rute yang disimpan di dalam pesan tersebut. Jika *node* tersebut memiliki informasi yang lebih baik tentang rute yang dituju, maka *node* tersebut akan memperbarui cache rute dan mengirimkan RREP dengan informasi rute yang baru.

Fungsi *path accumulation* pada DYMO membantu setiap *node* dalam jaringan MANET untuk mengumpulkan informasi tentang jalur terbaik ke setiap tujuan yang diketahui, sehingga memungkinkan setiap *node* untuk memilih jalur terbaik untuk mengirimkan paket data [18].

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

Protokol DSDV adalah protokol *routing ad-hoc* proaktif atau *table-driven* untuk mengatasi *routing loop*. Untuk berkomunikasi dalam jaringan, protokol DSDV menggunakan *sequence number* pada pesan yang dikirimkan [19]. *Sequence number* terjadi karena tabel routing proaktif yang digunakan DSDV, yang terus memperbarui informasi, nomor urut juga digunakan ketika perubahan jaringan terjadi.

Protokol DSDV menggunakan pembaruan rute berbasis waktu dan peristiwa. Time-driven berarti bahwa *node* akan berkomunikasi satu sama lain untuk berbagi informasi sehingga mereka dapat secara berkala memperoleh informasi terbaru. *Node* lain akan dipindahkan untuk mengirimkan data dari tabel routing baru dengan pemicu atau kejadian tertentu sementara event driven terjadi ketika ada pembaruan penting. Karena persyaratan sistem routing proaktif bahwa DSDV secara rutin memperbarui tabel routing, protokol DSDV membutuhkan sejumlah besar listrik [12]. Proses pengiriman paket pada protokol DSDV dapat dilihat dari gambar 2.5. M4 akan melakukan pengiriman sebuah paket data menuju M5.



Gambar 2.5 Protokol *Routing* DSDV [20]

M4 akan memeriksa tabel *routing* dan menemukan hop berikutnya untuk merutekan paket tersebut. Terlihat hop selanjutnya adalah M6 untuk menuju ke M5. Kemudian M4 akan mengarahkan paket menuju ke M6 dan rute yang dilalui paket dari M4 ke M6 akan terdaftar di tabel *routing*. M6 yang menerima paket akan memeriksa hop berikutnya untuk sampai ke M5. M7 yang merupakan hop selanjutnya dari M6 akan menerima paket dan memeriksa hop selanjutnya agar paket sampai ke M5. Begitu seterusnya hingga paket sampai tujuan.

Ketika sebuah node menerima paket, ia memulai proses inisialisasi sumber dan merutekan semua tujuan, lalu paket akan diperiksa pada tabel *routing* dan diterima untuk semua informasi rute. Jika nomor urut saat ini pada tabel *routing* = nomor urut yang diterima dan matriks saat ini $>$ matriks yang diterima, maka matriks saat ini = matriks yang diterima. Tetapi jika kondisinya berbeda, maka akan dicek apakah urutan saat ini $<$ urutan yang diterima dan matriks saat ini $>$ matriks yang diterima, maka urutan saat ini dan matriks saat ini = urutan dan matriks yang diterima. Namun jika kondisinya masih tidak sama dengan kondisi yang sudah disebutkan sebelumnya, maka pembaruan dari matriks yang diterima akan diabaikan [20].

2.2.4. Simulator OMNeT++

OMNET++ adalah simulator *discrete event* bersifat *open source* yang memiliki *library* dan *framework* simulasi C++ yang dapat diperluas, modular, berbasis komponen yang digunakan dalam membangun simulasi jaringan. *Discrete event* artinya simulasi bertindak pada peristiwa langsung di dalam *event* (kejadian). Keunggulan OMNET++ antara lain *open source*, mudah dioperasikan, dan sifatnya *extensible*. [21]

OMNeT++ merupakan *development environment* terintegrasi berdasarkan platform Eclipse. Selain fitur standar yang disediakan oleh Eclipse, OMNeT++ menambahkan fungsionalitas baru yang memungkinkan konfigurasi dan pembuatan model simulasi menggunakan bahasa NED, editor grafik jaringan, dan file yang digunakan untuk konfigurasi model. Memungkinkan untuk menganalisis hasil simulasi dari data yang dikumpulkan.

Source model yang disediakan oleh OMNeT++ berisi file berikut:

- a. Implementasi modul sederhana dan kode lainnya disertakan dalam file C++ (.cc dan .h).
- b. Definisi pesan yang akan diterjemahkan ke kelas C++ terdapat dalam file pesan (.msg).
- c. Deskripsi topologi dan komponen yang disertakan dalam file NED (.ned).
- d. Pengaturan parameter model yang dipertimbangkan dan pengaturan lainnya disertakan dalam file konfigurasi (.ini) [22].

Omnnet++ memiliki dukungan yang luas terhadap berbagai standar komunikasi wireless, menjadi alat yang efektif untuk memodelkan dan menguji jaringan nirkabel dalam berbagai skenario dan lingkungan. Berikut standar komunikasi *wireless* yang dapat digunakan pada Omnet++.

- a. IEEE 802.11a
IEEE 802.11a adalah standar jaringan nirkabel yang menggunakan gelombang mikro dalam frekuensi 5 GHz. Standar ini dikembangkan

oleh *Institute of Electrical and Electronics Engineers* (IEEE) dan merupakan salah satu varian dari teknologi Wi-Fi. 802.11a menggunakan modulasi OFDM (*Orthogonal Frequency Division Multiplexing*) untuk mentransmisikan data dengan kecepatan hingga 54 Mbps [23].

b. IEEE 802.11b

IEEE 802.11b adalah standar yang diperluas oleh IEEE pada tahun 1999 untuk produk *wireless* yang beroperasi pada kecepatan 11 Mbps, dan disebut sebagai IEEE 802.11b. Frekuensi yang digunakan dalam IEEE 802.11b adalah 2.4 GHz. Standar ini menjamin kompatibilitas produk dari berbagai produsen melalui *Wireless Ethernet Compatibility Alliance* (WECA). Standar ini digunakan untuk akses nirkabel ke jaringan lokal dan mendukung kecepatan transfer data 5,5 Mbps dan 11 Mbps. Metode yang digunakan dalam standar ini adalah Direct Sequence Spread Spectrum (DSSS) [24].

c. IEEE 802.11g

IEEE 802.11g adalah standar jaringan nirkabel yang beroperasi di frekuensi 2,4 GHz. 802.11g adalah pengembangan dari standar IEEE 802.11b dan memberikan kecepatan transfer data yang lebih tinggi, hingga 54 Mbps. Standar 802.11g menggunakan modulasi *Orthogonal Frequency Division Multiplexing* (OFDM) untuk meningkatkan kualitas sinyal dan mengurangi gangguan. Ini umumnya digunakan dalam aplikasi perumahan dan komersial karena biaya rendah dan kompatibel digunakan dengan perangkat 802.11b, yang berarti bahwa perangkat yang mendukung standar 802.11g juga dapat bekerja dengan 802.11b [25].

d. IEEE 802.11p

IEEE 802.11p adalah standar yang diperkenalkan oleh IEEE pada tahun 2010 dan merupakan evolusi dari standar IEEE 802.11a yang dioptimalkan untuk jaringan kendaraan. Standar ini menggunakan

Orthogonal Frequency Division Multiplexing (OFDM) dengan lebar saluran 10 MHz. IEEE 802.11p memiliki *Control Channels (CCH)* dan *Service Channels (SCH)* serta menggunakan skema pengkodean yang sama dengan standar IEEE 802.11a. Teknik akses yang digunakan adalah *Carrier Sense Multiple-Access with Collision-Avoidance (CSMA/CA)*. Standar ini mendukung kecepatan transfer data antara 3 hingga 54 Mbps dengan menggunakan berbagai skema pengkodean seperti *binary phase shift keying, convolutional coding, quadrature phase shift keying, 16 quadrature amplitude modulation, atau 64-quadrature amplitude modulation* [26].