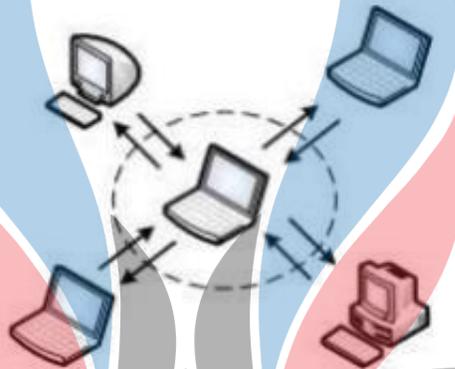


TINJAUAN PUSTAKA

2.1 MANET (*Mobile Ad-hoc Network*)



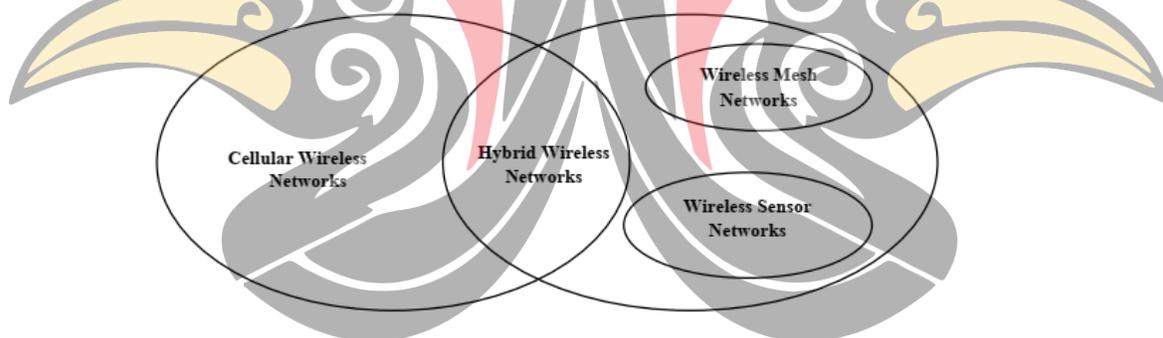
Gambar 2.1 Perancangan sederhana jaringan MANET (Manapa, Sampetoding and Lewakabessy, 2019)

Jaringan *Ad-hoc* merupakan salah satu jenis jaringan *wireless* atau nirkabel yang dirancang untuk tujuan tertentu. Pada jaringan *ad-hoc* pengguna dapat terhubung maupun mengirimkan data dengan pengguna lainnya tanpa memerlukan perantara atau infrastruktur jaringan. Salah satu jenis jaringan *ad-hoc* yang sering digunakan adalah jaringan MANET. *Mobile Ad-hoc Network* merupakan salah satu jenis jaringan *ad-hoc* yang dimana merupakan pelopor jaringan nirkabel. Sebuah jaringan MANET dapat berjalan tanpa memerlukan infrastruktur jaringan yang tetap, dimana pengguna dapat berkomunikasi dengan pengguna lain hanya dengan menggunakan gelombang radio dan pada jaringan MANET *user* dapat bergerak secara bebas. Jaringan MANET sendiri merupakan jaringan yang dibuat dengan tujuan tertentu.

Jaringan ini juga sering disebut jaringan yang dimana *node* nya dapat bergerak bebas atau otonom. Dalam perancangannya jaringan MANET sendiri diperlukan beberapa perhitungan dan pemilihan konfigurasi *routing* secara tepat dikarenakan jaringan ini memiliki sifat *multihop* dan *user* pada jaringan dapat bergerak secara

www.itk.ac.id

bebas yang membuat *routing* dan topologi pada jaringan dapat berubah sewaktu-waktu MANET sebagai jaringan *ad-hoc* tidak memerlukan administrasi terpusat seperti titik akses atau pusat stasiun, dan dalam perancangannya dapat diatur sesuai kebutuhan dan termasuk jaringan yang murah biaya dalam perancangannya. Jaringan MANET memiliki *host* seluler yang dapat bergerak bebas dan independen ke segala arah dan menghubungkan ke *host* lain dapat dilakukan dengan mudah. Selain dapat menghubungkan dengan mudah jaringan MANET juga dapat dengan mudah mengaktifkan maupun menonaktifkan *host* tanpa memberitahu *host* lain . Setiap *host* seluler harus meneruskan lalu lintas, yang tidak terkait dengan penggunaannya sendiri dan oleh karena itu bertindak sebagai *router*. Setiap *host* berperan sebagai *router* dimana harus menentukan maupun meneruskan lalu lintas data pada jaringan ini dan tantangan pada jaringan ini adalah tingkat perubahan *node* tergantung pada jumlah *node* dan kecepatan *node* yang membuat daya transmisi yang dapat digunakan terbatas (Lawrence, Latha and Lawrence, 2016).



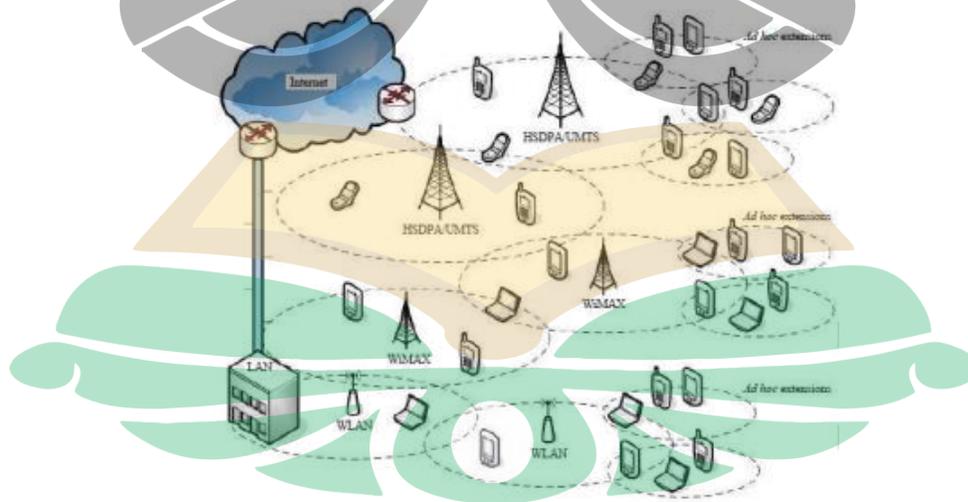
Gambar 2.2 Jaringan *celular* dan *Ad-hoc* Network (Lawrence, Latha and Lawrence, 2016)

Jaringan *ad-hoc* sendiri saat ini sudah mencapai generasi ketiga. dimana pada generasi pertama jaringan *ad-hoc* diteliti pada tahun 1970-an yang dimana tercipta sebuah jaringan bernama *Packet Radio Network* (PRNET). Jaringan tersebut diteliti oleh badan *Defense Advanced Research Project Agency* (DARPA) yang diawali dengan penelitian tentang penggunaan komunikasi radio untuk menyediakan komunikasi yang andal dan PRNET dengan *packet-switched*. *Packet switched* merupakan sebuah metode pengiriman data dengan cara memecah-mecah data menjadi

www.itk.ac.id

beberapa paket sebelum dikirimkan ke pengguna yang akan dituju akan tetap PRNET memiliki beberapa kekurangan yaitu sering terjadi komunikasi *multihop* antar *user* yang bergerak. PRNET kemudian berkembang diawal tahun 1980-an menjadi *Survivable Adaptive Radio Network* (SURAN).

Jaringan SURAN tercipta untuk meningkatkan kinerja radio dengan membuat jaringan yang lebih kecil, lebih murah dan hemat daya, dan tahan terhadap serangan elektronik. Ditahun yang sama, *United State Department of Defense* (DOD) mendanai sebuah proyek jaringan bernama *Near Term Digital Radio* (NTDR) yang memiliki sebuah kelebihan dimana jaringan ini menggunakan teknik *clustering*, *link state routing* dan mengatur jaringan *ad-hoc* yang lebih baik. NTDR ini menggunakan *clustering* dan *link state routing* dan mengatur jaringan *ad-hoc*. NTDR sendiri digunakan oleh tentara amerika serikat untuk menambah kekuatannya. Setiap tahunnya jaringan *ad-hoc* mengalami perkembangan dan mulai berkembang besar disekitar tahun 1990-an. Badan *Internet Engineering Task Force* (IETF) mulai mengembangkan jaringan MANET dengan menggunakan protokol standar dan mulai berkembang menggunakan *protokol* IEEE 802.11, Bluetooth dan HIPERLAN yang membuat jaringan *ad-hoc* memiliki performa yang lebih baik (Lawrence, Latha and Lawrence, 2016).



Gambar 2.3 Integrasi MANET dengan berbagai jaringan (Manapa, Sampetoding and Lewakabessy, 2019)

www.itk.ac.id

Pada jaringan MANET setiap *user* dapat terhubung walaupun dalam jarak yang jauh asalkan dapat mengirimkan *broadcast communication* yang dimana merupakan proses penyebaran informasi ke setiap *node* yang ada pada jaringan menggunakan *transmitter* dan *receiver wireless*. Hal ini membuat Jaringan MANET merupakan jaringan yang fleksibel dalam luas wilayahnya dan jaringan MANET dapat menggunakan transmisi *wireless* seperti HSDP, WiMax, dan WLAN dalam mendukung perancangan jaringan MANET. Jaringan MANET juga cocok digunakan sebagai jaringan komunikasi darurat apabila terjadi bencana alam disuatu daerah yang mengakibatkan kerusakan jaringan komunikasi di daerah tersebut. Adapun beberapa karakteristik jaringan MANET antara lain yaitu : (Manapa, Sampetoding and Lewakabessy, 2019).

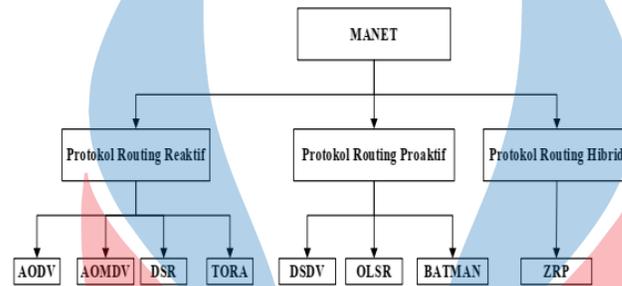
1. *Autonomous and infrastructure-less*, yang memiliki arti sistem tidak memiliki server pusat atau infrastruktur khusus dalam perancangan. Semua komunikasi antar *user* hanya disediakan oleh konektivitas nirkabel yang ada pada jaringan tersebut . Manajemen jaringan harus didistribusikan ke berbagai *node* sehingga pemilihan protokol sangat penting dalam perancangannya
2. *Self-Configuring*, yaitu jaringan ini konfigurasi dan aktivitas jaringan dilakukan oleh *node* itu sendiri termasuk penemuan topologi dan pengiriman pesan ke *node* lain atau bisa dikatakan *user* berperan sebagai *host* dan *router* dalam jaringan MANET.
3. *Dynamic Topology*, yang dimana topologi dapat berubah secara *dynamic* tergantung pergerakan *node*. Pada jaringan MANET *user* dapat bergerak secara bebas sehingga topologi akan berubah-berubah sewaktu-waktu. Setiap *node* dapat keluar dari jangkauan *user* lain sehingga *routing* akan berubah-ubah terus menerus
4. *Network scalability*, dimana kebanyakan perancangan jaringan MANET melibatkan jaringan dan *node* ber skala besar dikarenakan pada wilayah yang berskala besar membutuhkan jumlah *node* yang banyak untuk keberhasilan penyebaran jaringan.

- www.itk.ac.id
5. *Power Limitation*, dimana pada jaringan ini *user* menggunakan perangkat seluler yang menggunakan memori berukuran kecil dan ringan dengan kemampuan CPU yang lebih sedikit, dan penyimpanan daya yang rendah karena dipasok oleh sumber daya yang terbatas. *Multi-hop routing*, merupakan sebuah kondisi dimana *node* mencoba mengirim informasi ke *node* lain yang berada di luar jangkauan komunikasinya, informasi tersebut harus diteruskan melalui *node* perantara. Setiap *node* berperan sebagai *router* sehingga hal tersebut dapat dilakukan.
 6. *Wireless Links*, merupakan kekurangan dari jaringan *mobile ad-hoc* karena t rentan terhadap berbagai jenis serangan dan memiliki kapasitas penyimpanan yang rendah hal tersebut disebabkan karena pergerakan *node* dan *power supply* yang terbatas sehingga kegagalan komunikasi akan terjadi apabila ada *user* yang keluar dari cakupan *user* tersebut.
 7. *Variation in link and node capabilities*, merupakan variasi dimana setiap *node* yang ada pada jaringan dapat memiliki lebih dari satu *radio interface* yang memiliki kemampuan transmisi yang berbeda-beda setiap *node* nya. Hal ini dapat menyebabkan terjadinya distribusi lalu lintas yang padat sehingga dibutuhkan protokol dan algoritma jaringan dalam perancangannya.

Dari karakteristik jaringan MANET diatas dapat disimpulkan bahwa jaringan MANET merupakan jaringan yang digunakan untuk keperluan tertentu dan dapat digunakan sebagai jaringan darurat apabila terjadi kerusakan pada infrastruktur jaringan. Dalam perancangannya memerlukan pemilihan faktor-faktor pendukung seperti *protokol* jaringan, model mobilitas dan lain sebagainya. Adapun jaringan MANET memiliki beberapa kelebihan lain antara lain bebas *router* dikarenakan *user* dapat berperan sebagai *router* itu sendiri, kecepatan perancangan dan instalasi karena pembangunan infrastruktur tidak dibutuhkan, toleransi kesalahan, konektivitas dan harga yang murah dalam perancangannya. Adapun beberapa kekurangan dari perancangan jaringan MANET yaitu link yang tidak simetris, kemungkinan terjadi *routing overhead*, rentan terhadap serangan dari luar, *double connection*. dan topologi

yang berubah sewaktu-waktu tergantung dari pergerakan *user* atau pengguna. Sehingga jaringan MANET biasanya dipakai untuk kepentingan tertentu atau sebagai jaringan darurat. (Lawrence, Latha and Lawrence, 2016).

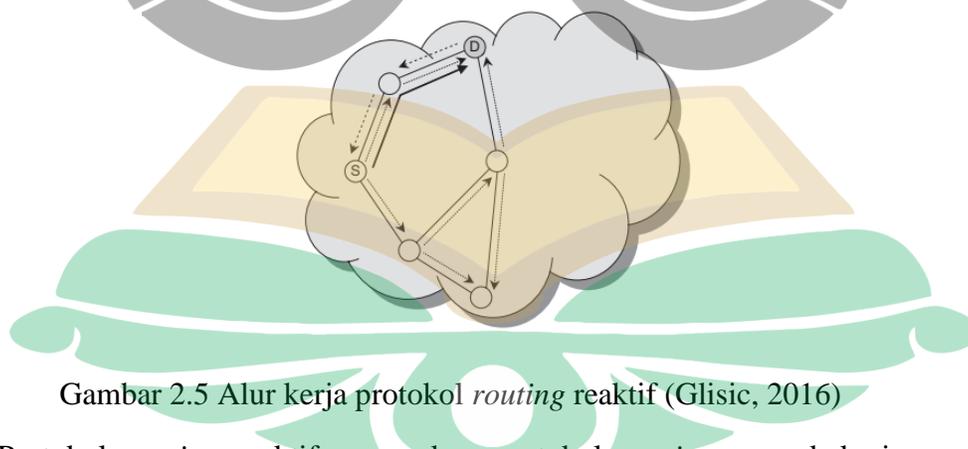
2.2 Protokol Routing



Gambar 2.4 Protokol *routing* MANET (Glisic, 2016)

Perancangan Jaringan MANET membutuhkan protokol *routing* yang berguna untuk *node* dapat menentukan jalur atau rute yang ingin mereka tuju untuk mengirimkan informasi atau data. Protokol *routing* terbagi menjadi tiga kelas berdasarkan performansi dan fungsi pada jaringan MANET yaitu *protokol routing* reaktif, proaktif, dan *hybrid* (Moraes and Sadjadpour, 2011).

2.2.1 Protokol Routing Reaktif



Gambar 2.5 Alur kerja protokol *routing* reaktif (Glisic, 2016)

Protokol *routing* reaktif merupakan protokol *routing* yang bekerja sesuai dengan permintaan yang ada dan mempertahankan rute tersebut jika diperlukan. Setiap kali *node* ingin mengirimkan informasi ke *node* yang lain pada protokol ini maka akan

dilakukan pembuatan rute baru berdasarkan situasi jaringan saat itu, dan rute akan dihilangkan ketika data sudah mencapai tujuan (Bai, Mai and Wang, 2017).

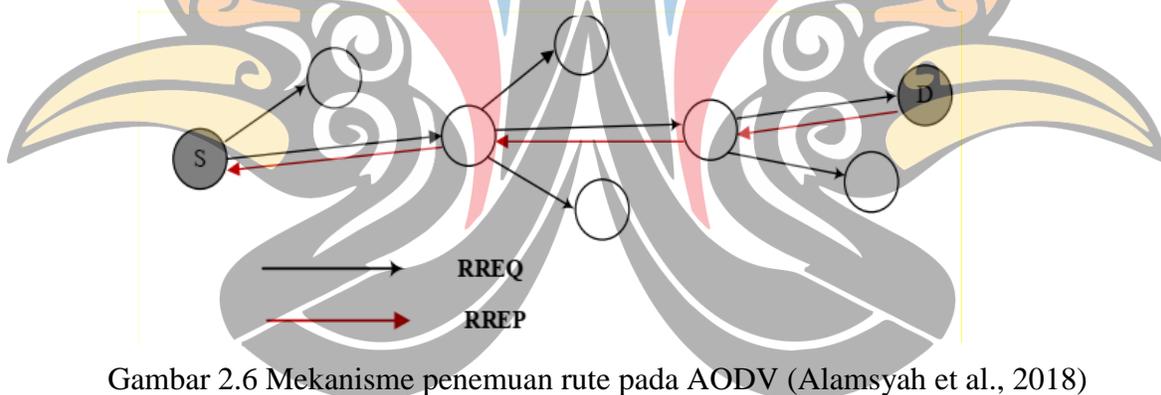
Adapun kekurangan pada protokol *routing* reaktif adalah sering terjadi penundaan pengiriman informasi ketika lalu lintas jaringan mengalami kesibukan. Oleh sebab itu protokol ini cocok digunakan pada jaringan komunikasi yang memiliki lalu lintas jaringan yang rendah. Adapun beberapa contoh *protokol routing* reaktif antara lain (Bai, Mai and Wang, 2017)

A. AODV (*Ad-hoc On-demand Distance Vector*)

AODV merupakan *protokol routing* yang dimana semua *node* akan bekerja secara terpisah dan informasi dari *node* terdekat tidak akan tersimpan. Akan tetapi semua *node* akan memiliki informasi tentang rute yang telah ditentukan sebelumnya di mana informasi dapat dikirim ke *node* yang ingin dituju. Sebuah rute akan dibentuk hanya ketika informasi tersebut akan dikirimkan ke *node* tujuan. Protokol AODV menggunakan algoritma yang memungkinkan *node* untuk mendapatkan rute untuk *node* tujuan dengan cepat, hal ini di tandai dengan tidak diperlukannya *node* untuk mempertahankan rute ke tujuan yang tidak dalam satu komunikasi yang telah dibuat. Protokol AODV memiliki beberapa kelebihan yaitu protokol ini dapat memungkinkan *node* merespon perubahan rute maupun topologi secara dinamis (Bai, Mai and Wang, 2017).

Protokol AODV menggunakan konfigurasi pesan *route discovery* dan *route maintenance*. Proses *route discovery* merupakan proses dimulai setiap kali *node* sumber perlu berkomunikasi dengan *node* lain yang tidak memiliki informasi *routing* didalam *routing* tabel. Setiap *node* memiliki dua penghitung terpisah yaitu nomor urutan id siaran dan *node*. Pada konfigurasi *node* sumber akan memulai mencari jalur dengan menyiarkan paket permintaan rute (RREQ) ke tetangganya. Ada penomoran yang termasuk dalam RREQ yaitu nomor urut *node* sumber dan nomor urut *node* tujuan terakhir yang dimiliki *node* sumber (Perkins and Royer, 1999).

Nomor urut yang ada pada sumber berguna untuk menjaga *update* informasi tentang rute yang mengarah ke *node* sumber, dan nomor urutan tujuan menentukan seberapa *update* rute ke tujuan *node* yang diinginkan. Pada saat melakukan pengiriman data *node* penerima pertama kali memeriksa bahwa RREQ diterima melalui tautan dua arah. Jika *node* perantara memiliki entri rute untuk tujuan yang diinginkan kemudian akan dibandingkan nomor urut tujuan dalam entri rutenya sendiri dengan nomor urut tujuan yang ada pada RREQ. Jika nomor urut yang ada pada RREQ untuk *node* tujuan lebih besar dari yang dicatat oleh *node* perantara, *node* perantara tidak boleh digunakan menanggapi RREQ. Maka *node* perantara akan menyiarkan ulang RREQ dan *node* hanya dapat membalas jika ia memiliki rute dengan nomor urut yang lebih besar. Jika *node* memiliki rute tujuan dan jika RREQ belum aktif, maka *node* kemudian meng-unicast paket balasan rute (RREP) (Perkins and Royer, 1999).



Gambar 2.6 Mekanisme penemuan rute pada AODV (Alamsyah et al., 2018)

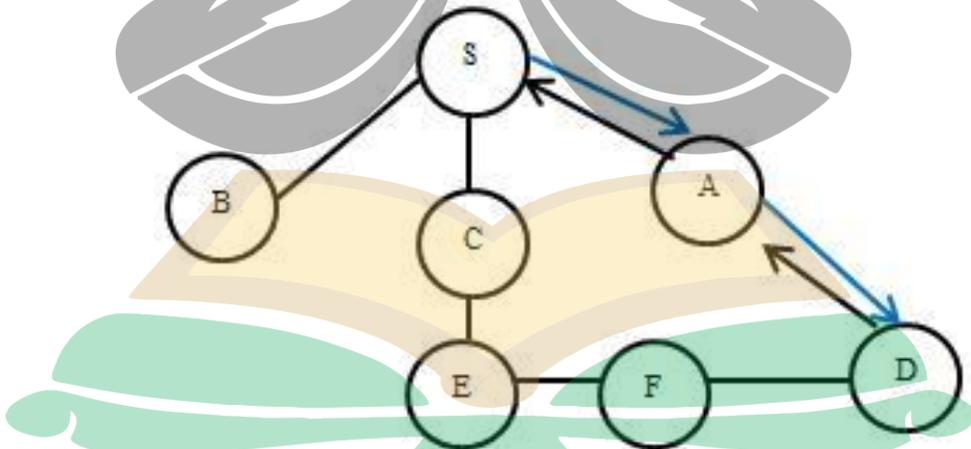
B. AOMDV (*Ad-hoc* On-demand Multipath Distance Vector)

AOMDV Merupakan pengembangan dari protokol AODV di protokol ini memiliki karakteristik yang sama dengan protokol AODV dimana pada protokol ini ditambahkan kinerja berbasis *multipath*. AOMDV memiliki karakteristik menghitung beberapa jalur selama penemuan *routing* rute. Protokol *routing* AOMDV biasanya digunakan pada jaringan *ad-hoc* yang sangat dinamis di mana kegagalan *routing* dan pemutusan rute sering terjadi pada jaringan. Ketika protokol ini mendapatkan permintaan jalur tunggal maka dilakukan penemuan rute baru yang diperlukan sebagai respons untuk setiap pemutusan rute yang ada pada jaringan.

Setiap penemuan rute akan dikaitkan dengan *overhead* tinggi dan latensi pada pengiriman data. Hal ini dapat dihindari dengan menyediakan beberapa jalur tambahan. Akan tetapi pada protokol ini penemuan rute baru diperlukan hanya jika semua jalur ke tujuan telah di matikan. Fitur penting dari protokol AOMDV adalah penggunaan informasi *routing* yang telah tersedia sebanyak mungkin sama seperti protokol AODV. Dengan demikian diperlukan sedikit biaya untuk penambahan jalur baru (Marina and Das, 2006).

C. DSR (*Dynamic Source Routing*)

DSR merupakan protokol *routing* di mana *node* sumber menentukan urutan lengkap *node* yang akan digunakan untuk meneruskan paket ke *node* tujuan. Pengirim akan mencantumkan rute ini di header paket secara eksplisit, dan mengidentifikasi setiap jangkauan *node* dengan alamat *node* berikutnya yang akan mengirimkan paket dalam perjalanan ke *host* tujuan. DSR telah banyak digunakan dalam sejumlah konteks untuk *routing* dalam jaringan kabel, baik menggunakan rute sumber yang ditentukan maupun dibangun secara statis dan dinamis. DSR telah dikonfigurasi di *Tucson Amateur Packet Radio* (TAPR) secara statis dan bekerja untuk *routing* dalam jaringan nirkabel (Johnson and Maltz, 1996).



Gambar 2.7 Mekanisme penemuan rute pada DSR ((Anggraini, Nugroho and Cahyadi, 2017)

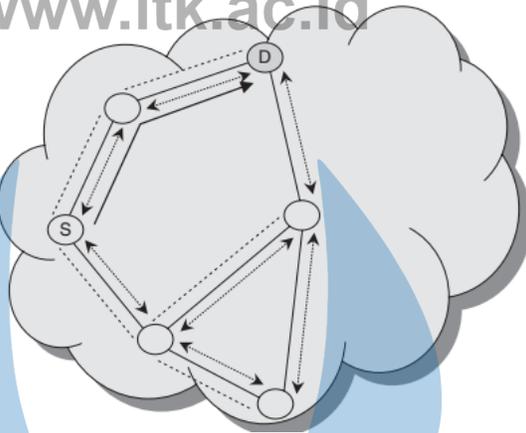
Protokol ini disajikan dan secara eksplisit dirancang untuk digunakan dalam lingkungan nirkabel jaringan *ad-hoc* seperti pada jaringan MANET. Pada protokol DSR tidak ada iklan *router* berkala dan ketika sebuah *node* membutuhkan rute *node* lain (Johnson and Maltz, 1996).

D. TORA (*Temporally Ordered Routing Algorithm*)

Protokol TORA merupakan salah satu jenis protokol *routing* yang bekerja sangat efisien, adaptif, dan dapat diskalakan untuk MANET yang memiliki topologi dan kinerja sangat dinamis. Protokol TORA memiliki tiga fungsi dasar dalam konfigurasinya yaitu pembuatan rute, pemeliharaan rute, dan penghapusan rute. Di protokol ini semua *node* memiliki informasi *routing* keseluruhan *node* yang ada pada jaringan. Kemudian setelah itu protokol akan memilih rute yang paling efisien dari sumber *node* ke *node* tujuan. Protokol TORA menyediakan rute bebas *loop* dan beberapa rute cadangan untuk *node* yang ingin mengirimkan paket ke *node* tujuan. Protokol ini meminimalkan reaksi terhadap perubahan topologi ketika *node* mengalami perubahan tempat dengan melokalisasi reaksi ke sekelompok yang lebih kecil dan juga menyediakan pemulihan yang cepat. Protokol ini juga mendeteksi partisi jaringan dan menghapus rute yang tidak valid (Bai, Mai and Wang, 2017).

2.2.2 Protokol Routing Proaktif

Pada *Protokol Routing Proaktif*, setiap *node* akan memperbarui dan memelihara *routing* tabel yang ada dan juga mempertahankan rute ke semua *node* di jaringan MANET. *Routing* tabel berisi informasi *routing* ke *node* lain maupun *node* tujuan. Informasi yang berada pada *routing* tabel akan diperbarui secara konsisten untuk menjaga informasi *routing* tetap konsisten dan diperbarui saat status jaringan berubah sesuai dengan pergerakan *node* (Bai, Mai and Wang, 2017).

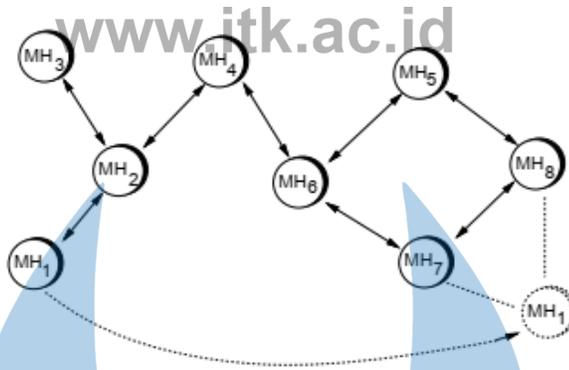


Gambar 2.8 Alur kerja protokol *routing* proaktif (Glisic, 2016)

Protokol *routing* proaktif biasanya menggunakan algoritma Bellman-Ford dan protokol ini memiliki beberapa kekurangan yaitu dalam pembaruan *routing* tabel membutuhkan biaya tambahan yang besar dan informasi *routing* yang tersedia pada semua *node* mungkin tidak digunakan oleh semua *node*. Namun protokol *routing* proaktif cocok untuk jenis jaringan yang memiliki jaringan lalu lintas yang tinggi. Adapun beberapa contoh dari protokol *routing* proaktif antara lain sebagai berikut (Glisic, 2016; Bai, Mai and Wang, 2017).

A. DSDV (*Destination-Sequenced Distance Vector*)

DSDV merupakan protokol *routing* yang dibuat berdasarkan algoritma *Bellman-Ford* klasik. Protokol DSDV memiliki kelemahan yaitu *routing* berulang dapat terjadi di protokol ini. Akan tetapi pembentukan *loop* dapat dihilangkan atau diminimalisir dengan cara *node* harus sering berkomunikasi dan berkoordinasi satu sama lainnya. Permasalahan ini muncul karena perubahan topologi yang sering terjadi pada perancangan jaringan MANET. Pada protokol DSDV memiliki fitur protokol informasi *routing* (RIP). Penerapan RIP ini pada jaringan *ad-hoc* yang bersifat mobilitas yang berfungsi untuk mengurangi *routing* berulang (Glisic, 2016).



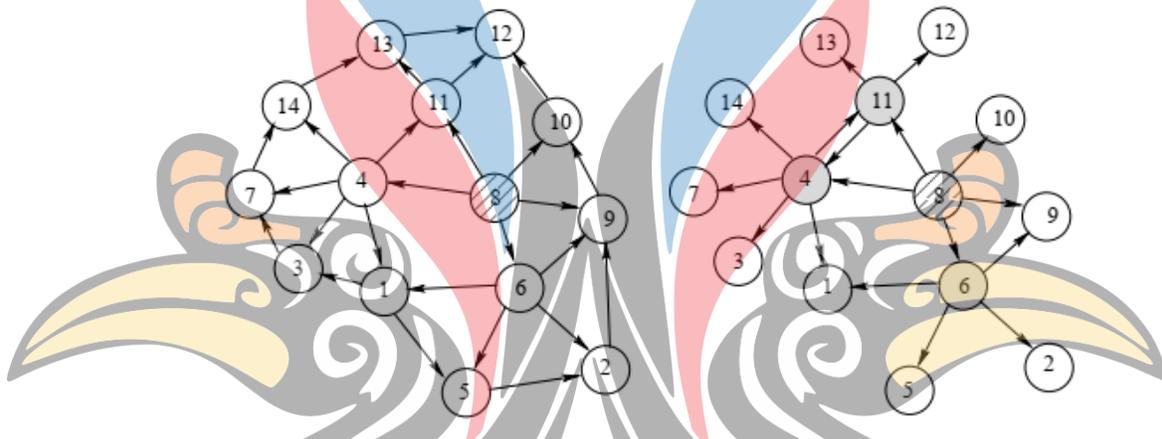
Gambar 2.9 Mekanisme penemuan rute pada DSDV (Perkins and Bhagwat, 1994)

Pada Protokol DSDV, setiap *node* harus memberikan informasi mengenai *routing* tabel ke *node* tetangga yang ada pada jaringan. Selain itu, setiap *node* setuju untuk menyampaikan paket ke *node* lain berdasarkan permintaan yang berada pada rute pengiriman paket. Persetujuan berfungsi agar *node* dapat menentukan jumlah lompatan terpendek untuk rute ke tujuan *node*. Hal itu memungkinkan untuk tidak terjadi gangguan yang tidak perlu pada saat mengirim paket. Informasi *routing* akan disebarkan dengan cara menyiarkan atau *multicasting* paket yang ditransmisikan secara berkala dan bertahap saat perubahan topologi terjadi pada jaringan. Data yang disimpan berupa lamanya waktu antara kedatangan pertama dan kedatangan rute terbaik untuk setiap tujuan tertentu. Berdasarkan data ini, keputusan terbaik dapat dibuat untuk menunda rute penyebaran yang ada, sehingga meredam fluktuasi pada tabel *routing* (Perkins and Bhagwat, 1994).

B. OLSR (*Optimized Link State Routing*)

Protokol OLSR merupakan protokol dikembangkan berdasarkan algoritma link state *routing* dan protokol ini memiliki teknik optimal dalam mengkaji informasi yang berkaitan dengan topologi jaringan MANET. Protokol ini cocok untuk aplikasi jaringan nirkabel yang berisifat *mobile*. Protokol OLSR merupakan protokol proaktif yang menggunakan mekanisme relai *multipoint* yang merupakan skema penerusan paket status tautan yang efisien dalam pengerjaannya. Optimasi dapat dicapai dengan cara mengurangi informasi kontrol overhead dengan memperpendek

panjang paket kontrol dan dengan mengurangi jumlah link yang digunakan untuk meneruskan paket yang akan dikirimkan ke *node* tujuan. Karenanya, Protokol OLSR memiliki fitur yaitu multipoint relay (MPR) yang memungkinkan *node* yang bertanggung jawab untuk meneruskan pesan kontrol siaran karena proses flooding yang diperlukan untuk menyebarkan informasi status tautan. Berbeda dari algoritma klasik lainnya di mana informasi status dibanjiri melalui semua *node* jaringan, dalam skema OLSR hanya informasi status tautan yang bersifat parsial yang akan didistribusikan di jaringan. Berikut contoh gambaran *routing* protokol OLSR dengan fitur MPR (Moraes and Sadjadpour, 2011).



Gambar 2.10 *Routing* protokol OLSR dengan MPR (Moraes and Sadjadpour, 2011)

Protokol OLSR memiliki beberapa kelebihan dalam perancangan jaringan nirkabel yang bersifat *mobile* adalah pengurangan *overhead-routing* dibandingkan dengan protokol proaktif lainnya. Akan tetapi protokol ini masih menggunakan protokol berbasis tabel yang memungkinkan protokol ini untuk mengirim pesan *routing* secara berkala melalui jaringan untuk memperbarui pembaruan *routing* table (Moraes and Sadjadpour, 2011).

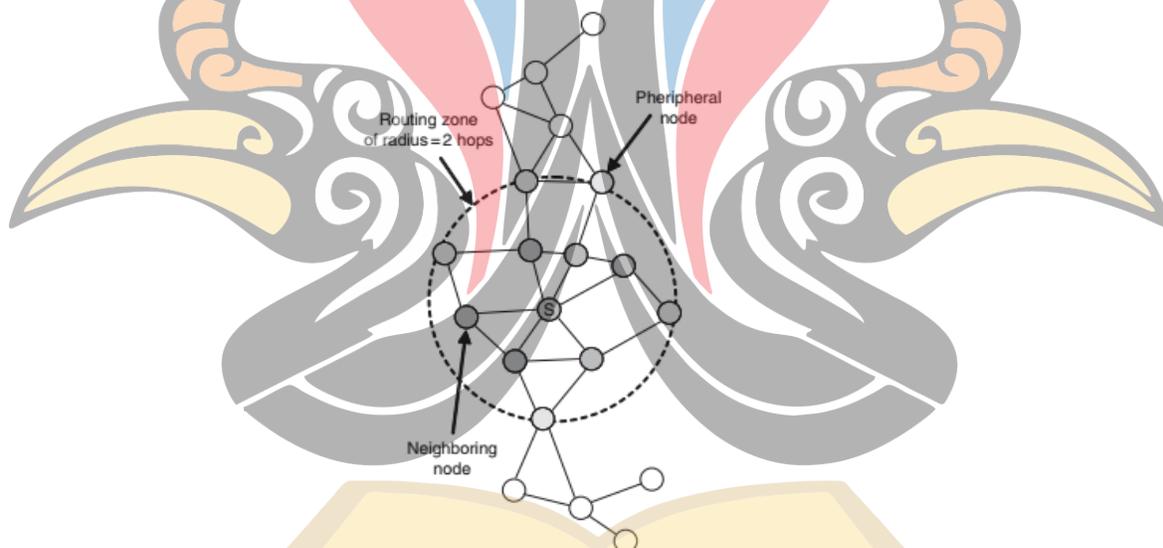
C. BATMAN (*Better Approach to Mobile Ad-hoc Network*)

Protokol BATMAN adalah protokol *routing* proaktif untuk jaringan mesh *Ad-hoc* nirkabel, dan jaringan *Ad-hoc* Seluler (MANET). Protokol ini secara proaktif memelihara informasi tentang keberadaan semua *node* dalam cakupan jaringan yang

www.itk.ac.id

dapat diakses melalui link komunikasi *single-hop* maupun komunikasi *multi-hop*. Protokol BATMAN dapat menentukan setiap tujuan yang ada dalam cakupan jaringan satu *single-hop* yang berada didekatnya, yang kemudian dapat digunakan sebagai *gateway* terbaik untuk berkomunikasi dengan *node* yang ingin dituju. Pada protokol ini dilakukan *routing* berbasis IP *multi-hop*, tabel *routing* suatu *node* berisikan gateway tautan-lokal untuk setiap *host* atau rute jaringan ke *node* yang ingin dituju. Pada protokol BATMAN tidak diperlukan mencari tahu atau menghitung rute lengkap yang ada, hal ini memungkinkan implementasi dapat dilakukan secara cepat dan efisien untuk mengirim paket ke *node* tujuan yang ada pada cakupan jaringan yang bersifat *mobile* atau bergerak (Neumann *et al.*, 2008).

2.2.3 Protokol *Routing Hybrid*



Gambar 2.11 *Routing* protokol ZRP (Glisic, 2016)

Protokol *routing hybrid* merupakan protokol yang menggunakan fitur protokol proaktif dan reaktif dalam menjalankan proses *routing*. *Zone routing protocol* (ZRP) adalah contoh dari protokol *routing hybrid* dimana kinerja ditingkatkan dengan memilih protokol proaktif atau reaktif berdasarkan algoritma mana yang paling memiliki kinerja terbaik yang dapat digunakan dalam pengiriman data. Protokol *routing zona* (ZRP) adalah protokol *routing hybrid* yang secara proaktif memelihara

www.itk.ac.id

rute dalam cakupan wilayah jaringan MANET. Protokol ZRP menggunakan topologi zona untuk meningkatkan efisiensi mekanisme *routing* yang ada. ZRP dapat dikonfigurasi untuk jaringan tertentu melalui penyesuaian parameter yang ada pada jaringan (Glisic, 2016). Pada protokol *routing* ini sebuah zona *routing* radius disimbolkan dengan r dan ditentukan untuk setiap *node* dan termasuk *node* yang berada pada jarak minimum dalam *hop*. *Node* yang berada pada lingkaran seperti pada gambar, dikatakan berada di dalam zona perutean dari *node* pusat yang disimbolkan dengan S . *Node* yang berada di luar lingkaran dikatakan berada di luar zona *routing*. *S node perifer* merupakan *node* yang jarak minimumnya ke S sama persis dengan radius zona (Glisic, 2016).

2.3 Model Propagasi

Jaringan nirkabel memiliki karakteristik yang menyebabkan keterbatasan dalam kinerja jaringan *ad-hoc*. Kualitas dari jaringan nirkabel adalah kombinasi efek yang kompleks karena hilangnya rute maupun pemudaran multi rute. Propagasi radio dapat menyebabkan variasi yang signifikan tergantung dari lingkungan, frekuensi, kecepatan *node* dan sumber gangguan, dan faktor dinamisnya lainnya yang ada propagasi radio. *Path loss* merupakan variabel yang mengukur kerugian dalam kekuatan sinyal yang disebabkan karena jarak antar *node* yang ada pada jaringan. *Shadow fading* mengartikan fluktuasi di sekitar dimana hilangnya jalur rata-rata (Das, Priyadarshini and Mohapatra, 2013).

2.3.1 Model Propagasi Non-Fading

Model propagasi biasanya dibagi menjadi dua jenis yaitu *nonfading* dan *fading*. Model propagasi *non-fading* merepresentasikan tentang pengaruh bahwa gelombang radio harus mencakup area yang diinginkan dan berkembang ketika jarak ke *node* tujuan meningkat. Contoh dari model propagasi *non-fading* adalah *free-space* dan *two ray ground*. Adapun penjelasan lebih lengkap mengenai model propagasi tersebut sebagai berikut (Das, Priyadarshini and Mohapatra, 2013).

A. Free-Space Model

Model *free-space* merupakan model propagasi yang biasanya digunakan untuk memprediksi kekuatan sinyal ketika pemancar dan penerima memiliki jalur pandang yang jelas dan tidak terhalang atau terjadi *blocking* di antara keduanya. Hal ini memungkinkan memprediksi bahwa daya yang diterima akan berkurang sebagai fungsi jarak pemancar ke penerima yang dinaikkan ke beberapa daya dan biasanya ke tingkatan kedua. Pada model propagasi ini menggunakan persamaan Friis yang terkenal digunakan untuk menghitung daya yang diterima (Das, Priyadarshini and Mohapatra, 2013).

$$P_{r,FS} = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (2.1)$$

Sebuah jalur langsung antara pemancar t dan penerima r diasumsikan pada model propagasi ini. Daya yang diterima pada P_r tergantung pada daya yang ditransmisikan P_t , penguatan penerima G_t dan antena pemancar G_r yang ada pada persamaan dan panjang gelombang k , jarak d antara kedua *node* dan koefisien kehilangan sistem L . Semua parameter yang dijelaskan sebelumnya, kecuali jarak d , adalah parameter konstan lebar sistem pada model propagasi *free-space*. Oleh sebab itu, daya sinyal yang akan diterima (P_r) hanya bisa berubah dengan jarak antara *node* pengirim dan *node* penerima. Hal tersebut dikarenakan kedua parameter penerima $RXThresh$ dan $CSThresh$ juga konstan, maka diperlukan berada di dalam disk yang sempurna untuk *node* penerima (Eltahir, 2007; Das, Priyadarshini and Mohapatra, 2013).

B. Two Ray Ground Model

Pada Model propagasi *Two ray ground* disumsikan bahwa ada dua rute antara *node* sumber dan *node* tujuan. Jalur pertama merupakan jalur garis pandang dan yang kedua adalah jalur pantulan dari tanah. Ini adalah versi perbaikan dari model *free-space*. Ketinggian kedua antena di atas permukaan tanah digambarkan dengan h_t dan h_r yang konstan. Hingga jarak saling silang akan menghasilkan $d < d_{Thers}$ (Das, Priyadarshini and Mohapatra, 2013).

$$d_{Threshold} = \frac{4\pi h_t h_r}{\lambda}$$

(2.2)

Persamaan diatas merupakan persamaan model *free-space*. Di luar jarak ini ($d \geq d_{Threshold}$), pantulan tanah akan mengakibatkan destruktif dan mengganggu sinar langsung dan selanjutnya mengurangi kekuatan sinyal yang terkirim. Kekuatan sinyal *node* penerima kemudian berbanding terbalik dengan pangkat keempat dari jarak d . Hal yang sama terjadi pada model *free-space*, yaitu model propagasi *two-ray ground* hanya berisi jarak antara *node* pengirim dan *node* penerima sebagai parameter variabel yang ada (Das, Priyadarshini and Mohapatra, 2013).

$$P_{r,FS} = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L}$$

(2.3)

2.3.2 Model Propagasi *Fading*

Model propagasi *fading* merupakan model propagasi yang menghitung kekuatan sinyal tergantung pada pergerakan *node* atau kerangka waktu yang kecil pada jaringan tersebut. Beberapa model statistik digunakan untuk menggambarkan *fading* dalam cakupan jaringan nirkabel dan distribusi yang paling sering digunakan untuk *fading* skala besar adalah model propagasi shadowing dan model propagasi nakagami. Adapun penjelasan lebih lengkap mengenai model propagasi *fading* tersebut sebagai berikut (Eltahir, 2007; Das, Priyadarshini and Mohapatra, 2013).

A. Shadowing Model

Pada model propoagasi *shadowing*, model propagasi tersebut menggunakan variabel acak X . Diperlukan jarak referensi d_0 untuk menghitung kekuatan sinyal ruang bebas rata-rata yang diterima $P_r, FS (d_0)$. Eksponen *path loss* β akan dijelaskan pada persamaan dibawah ini dan untuk hasil tergantung pada lingkungan dan konstan. X terdistribusi normal dengan rata-rata nol dan deviasi standar σ (disebut deviasi bayangan) dan untuk nilai β dan σ ditentukan secara empiris (Das, Priyadarshini and Mohapatra, 2013).

www.itk.ac.id

$$P_{r,SH} = \frac{P_t G_t G_r h_t^2 h_r^2}{d_0^4 L} \frac{d}{d_0} 10^x \quad (2.4)$$

Didapatkan model propagasi *shadowing* memperkenalkan beberapa jenis ketidakpastian dalam mentransmisi paket menuju *node* tujuan. Penerimaan yang benar dijamin untuk jarak dekat dan tidak mungkin untuk jarak jauh, sedangkan penerimaan yang benar untuk jarak menengah tidak dapat diprediksi. Meskipun memiliki kekurangan pada penerimaan, namun pada area penerimaan yang benar tetap membentuk lingkaran ketika mempertimbangkan banyak transmisi. Variasi kekuatan sinyal tidak bergantung pada arah dan kemungkinan kesalahan tetap dapat terjadi selama setiap transmisi dilakukan. Hal ini bervariasi secara signifikan pada model propagasi ini karena antara (Nassef, 2010; Das, Priyadarshini and Mohapatra, 2013).

B. Nakagami Model

Model nakagami ini dikembangkan oleh Nakagami dan banyak digunakan karena cocok dengan data empiris yang *fading*. Model ini lebih realistis dalam perancangannya. Model nakagami mencakup beberapa distribusi lain untuk beberapa kasus tertentu. Model ini sangat umum dan lebih realistis yang memungkinkan perkiraan sinyal yang serupa. Distribusi gamma digunakan untuk memodelkan daya penerimaan radio dapat dilihat pada persamaan dibawah ini (Rhattoy and Zatni, 2013)

$$P_r(r) = \frac{2m^m r^{2m-1}}{\gamma(m)\Omega^m} \exp\left(-\frac{mr^2}{\Omega}\right), r > 0 \quad (2.5)$$

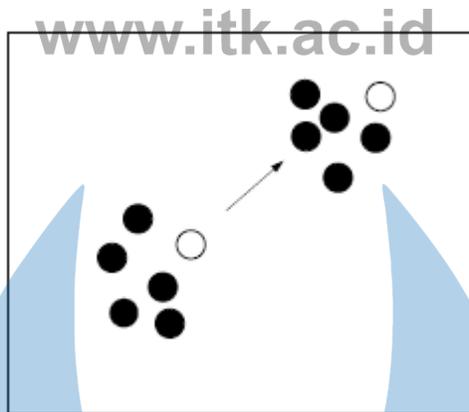
dimana m pada persamaan tersebut merupakan intensitas efek *fading*. Model nakagami adalah model propagasi yang baik untuk jaringan *ad-hoc* di daerah perkotaan, karena memperhitungkan pengaruh berbagai jenis dinding bangunan pada sinyal yang diterima. Model propagasi Nakagami cocok digunakan untuk tipikal jaringan yang bergerak (Rhattoy and Zatni, 2013).

2.4 Mobility models

Model Mobilitas merupakan gambaran pola pergerakan perangkat atau *user* dalam jaringan yang mengalami perubahan posisi dan kecepatan. Perubahan kecepatan maupun posisi dari waktu ke waktu mengakibatkan perpindahan perangkat seluler di wilayah jaringan MANET. Adapun perubahan kecepatan *user* dapat dibagi menjadi 4 jenis yang pertama adalah *fast car model* yang memiliki kecepatan 30 m/s atau 108 km/jam ini merepresentasikan mobil berkecepatan tinggi. Kemudian ada *slow car model* yang memiliki kecepatan berkisar 10 m/s atau 36 km/jam yang merepresentasikan pergerakan Mobil dalam keadaan pelan. Setelah itu ada *race walking model* yang memiliki kecepatan 4 m/s atau 14.4 km/jam yang menggambarkan kecepatan pengguna dalam posisi berlari. Adapun yang terakhir adalah *human walking model* yang memiliki kecepatan 2 m/s atau 7.2 km/jam yang dimana hal ini merepresentasikan manusia ketika sedang berjalan. Model mobilitas dikategorikan sebagai model mobilitas homogen dan heterogen. Adapun penjelasan lebih lanjut dari dua model tersebut sebagai berikut (Abdullah, Ozen and Bayramoglu, 2019).

2.4.1 Mobilitas Homogen

Pada model mobilitas homogen, *node* atau *user* cenderung bekerjasama dengan *node* tetangga ataupun *node* yang mewakili pola gerakan tunggal atau sama. *Node* seluler melintasi menurut model mobilitas tertentu. Pada model mobilitas homogen terdapat dua jenis model yaitu model acak dan model terkontrol. Model acak selanjutnya dibagi lagi menjadi acak sebagian dan acak total. Adapun contoh jenis mobilitas homogen adalah *pursue mobility model*. Model ini menunjukkan bahwa *node* dalam jaringan dibatasi ke satu *node* di depan yaitu *node* bergerak menuju *node* yang ingin dituju bersifat tunggal dengan memiliki kecepatan yang seragam. *Node* bergerak dalam kelompok dengan kecepatan dan arah yang seragam untuk mencapai *node* yang dituju pada jaringan MANET, mirip dengan model mobilitas *random way point* (Taleb *et al.*, 2013).



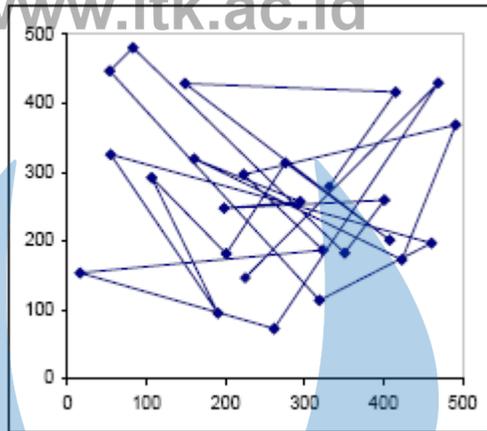
Gambar 2.12 Pursue mobility models (Gupta, Sadawarti and Verma, 2013)

2.4.2 Mobilitas Heterogen

Pada model mobilitas heterogen, *node* merepresentasikan pola pergerakan berbeda yang bervariasi dari *node* sebelumnya. Model mobilitas ini tidak tergantung pada model mobilitas lain yang digunakan dalam jaringan MANET. Salah satu contoh dari model mobilitas heterogen adalah *random way point* berdasarkan model *random* dan ada juga model *manhattan grid* berdasarkan model geografis pada jaringan. Adapun contoh lainnya dari mobilitas heterogen sebagai berikut (Taleb *et al.*, 2013).

A. Random Way Point

Pada model *random way point*, *node* seluler diterapkan secara acak dan memilih tujuannya secara independen atau dapat memilih tujuannya sendiri. *Node* akan bergerak menuju tujuan dengan kecepatan acak setiap saat, saat mencapai *node* tujuan, *node* akan berhenti untuk periode waktu tertentu yang disebut periode jeda yang dilambangkan T_{max} , yang digunakan untuk menghindari penghentian dan permulaan *node* secara tiba-tiba. Setelah waktu jeda, secara acak akan memilih tujuan lain di jaringan dan bergerak ke arahnya. Proses ini akan terus dilakukan hingga mengunjungi semua *node* di jaringan MANET. Dimisalkan bahwa V_{min} dan V_{max} menunjukkan kecepatan minimum dan maksimum yang dikonfigurasi pada *node*. Jika kecepatan V_{maks} maksimum dan T_{maks} kecil, maka jaringan tersebut dikatakan stabil. Jika kecepatan V_{min} minimum dan T_{max} besar, maka jaringan tersebut akan dikatakan dinamis (Gupta, Sadawarti and Verma, 2013).

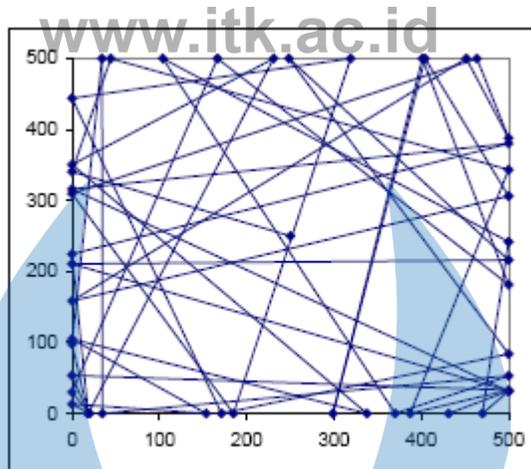


Gambar 2.13 *Random way point* (Gupta, Sadawarti and Verma, 2013)

Pada Model *random way point* memiliki beberapa keunggulan dalam mensimulasikan semua jenis protokol *routing* karena kemudahannya pada *node*, model mobilitas ini sering digunakan dalam perancangan jaringan MANET, dan model mobilitas ini memiliki pergerakan *node* yang realistis dalam perancangannya. Akan tetapi model ini memiliki kelemahan dalam kaitannya dengan waktu nyata karena hambatan membuat *node* tidak bergerak secara dinamis (Gupta, Sadawarti and Verma, 2013).

B. *Random Direction Mobility Model (RDMM)*

Model mobilitas arah acak merupakan pengembangan dalam model mobilitas sebelumnya. Dalam model ini, *user* memilih arah acak daripada tujuan acak. *User* pada mobilitas ini cenderung melakukan perjalanan ke tengah area simulasi hingga ke batas ujung simulasi. Hal tersebut dilakukan untuk mengekang gelombang kepadatan yang dihasilkan dalam model mobilitas *way point* acak.



Gambar 2.14 *Random direction mobility* (Gupta, Sadawarti and Verma, 2013)

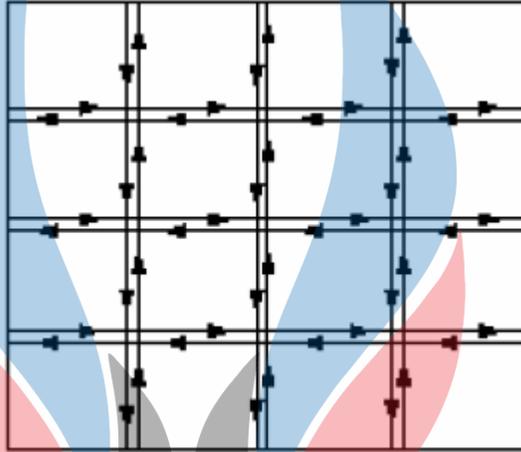
Semua pengguna ditempatkan secara acak di area jaringan dan diberi arah sudut dalam kisaran dari 0 hingga 2π pada kecepatan acak yang seragam antara t_0 dan t_{max} . Pada model mobilitas ini *user* akan menuju batas dari wilayah simulasi, Pengguna berhenti untuk waktu tertentu dan sudut gerakan baru dan kecepatan baru ditetapkan ke *node* yang ada pada model mobilitas ini. Proses ini dilanjutkan sampai simulasi berakhir. Karena pengguna digunakan untuk melakukan perjalanan secara acak dan berhenti di batas untuk beberapa waktu, jumlah *hop* rata-rata untuk paket jauh lebih tinggi dan rute yang dipilih seragam dibandingkan dengan model mobilitas lainnya pada jaringan MANET. Akan tetapi pada model mobilitas ini memiliki kekurangan yaitu pergerakan *node* yang tidak realistis dan jarak rata-rata antar *user* berjauhan sehingga evaluasi *protokol* akan sulit dilakukan. (Gupta, Sadawarti and Verma, 2013).

C. *Manhattan Grid*

Pada model mobilitas Manhattan, *node* akan dapat bergerak baik dalam arah vertikal maupun horizontal. Model Manhattan menggunakan topologi jalan grid. Sebuah *node* dapat mengubah arah pergerakannya pada titik perpotongan. Kecepatan *node* akan bersifat dinamis, yang bergantung pada kecepatan *node* sebelumnya pada model ini. Model mobilitas ini dirancang sedemikian rupa sehingga satu *node* dapat bergerak pada satu waktu berjalan dalam jalur dan arah

www.itk.ac.id

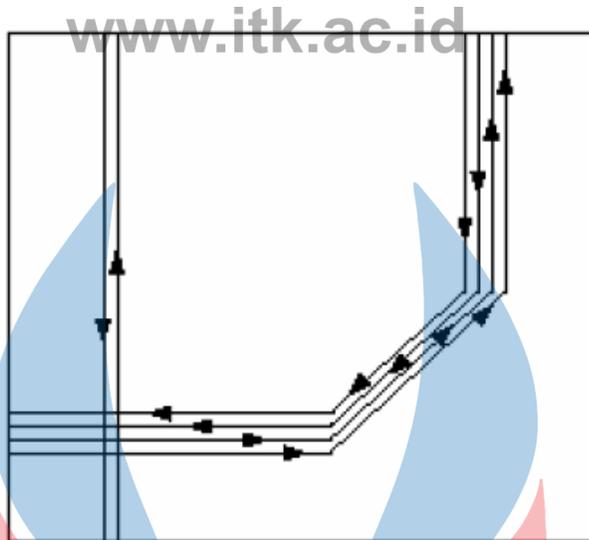
yang sama. Titik-titik yang berjalan dalam vektor yang sama menjaga jarak tertentu di antara mereka. Probabilitas *node* bergerak lurus adalah 0,5. Probabilitas *node* berubah arah baik secara vertikal maupun horizontal adalah nilai setengahnya yaitu 0,25 (Gupta, Sadawarti and Verma, 2013).



Gambar 2.15 Manhattan *grid* (Gupta, Sadawarti and Verma, 2013).

D. *Freeway Mobility*

Model mobilitas ini biasanya digunakan untuk bertukar status lalu lintas atau melacak kendaraan di jalan bebas hambatan. Pada model mobilitas ini digambarkan dengan beberapa jalan bebas hambatan di peta dan setiap jalan bebas hambatan memiliki lajur di kedua arah. Pola mobilitas ini menggambarkan jalan bebas hambatan diharapkan memiliki ketergantungan spasial dan ketergantungan temporal yang tinggi. Adapun perbedaan model mobilitas ini dengan model mobilitas *random way point* adalah setiap *node* bergerak dibatasi dengan jalur yang dibuat dengan bebas hambatan, kecepatan *user* bersifat temporan atau bergantung dengan kecepatan *user* sebelumnya *user* tersebut dan kecepatan *user* yang ada tidak boleh melebihi kecepatan *user* yang telah lewat sebelumnya (Gupta, Sadawarti and Verma, 2013).



Gambar 2.16 *Freeway mobility* (Gupta, Sadawarti and Verma, 2013)

2.5 *Traffic Agent*

MANET (*Mobile Ad-hoc Network*) merupakan salah satu jenis jaringan *ad-hoc* yang dimana merupakan pelopor jaringan nirkabel. Setiap *node* pada Jaringan MANET berperan sebagai server atau menentukan data dan menentukan tujuan dimana *node* akan di kirimkan. *Traffic agent* merupakan komponen yang berperan sebagai pengatur lalu lintas data dan memungkinkan terjadinya komunikasi antar *user* maupun penghubung antara *node* satu dengan yang lainnya dan setiap *traffic agent* memiliki karakterisitiknya masing-masing. Terdapat dua tipe *traffic agent* yang digunakan pada jaringan MANET antara lain TCP/FTP dan UDP /CBR (Sharma and Gupta, 2012).

Pada skenario lalu lintas jaringan MANET, TCP mewakili tipe data yang akan dikirimkan ke *node* tujuan dan FTP mewakili agen lalu lintas atau berperan sebagai pengirim data TCP ke *node* yang akan dituju. TCP berada protokol lapisan transport dan FTP adalah protokol lapisan aplikasi atau menghubungkan *user* antar *user*. Pada TCP/FTP menggunakan transmisi berorientasi koneksi, dimana terdapat beberapa tahapan yang terjadi pada pengirimannya antara lain pembentukan koneksi, transmisi data dan pemutusan koneksi. Adapun perbandingan dari karakteristik dari masing-masing *traffic agent* antara lain sebagai berikut (Jaiswal and Bhadauria, 2015).

Tabel 2.1 Karakteristik *Traffic Agent* (Jaiswal and Bhadauria, 2015)

Karakteristik	TCP/FTP	UDP/CBR
Deskripsi	TCP merupakan protokol lapisan <i>transport</i> yang mewakili tipe data dan FTP adalah protokol lapisan aplikasi yang mewakili agen lalu lintas aplikasi yang mengangkut data TCP.	UDP adalah protokol lapisan <i>transport</i> yang mewakili tipe data dan CBR adalah protokol lapisan aplikasi yang mewakili agen lalu lintas aplikasi yang mengangkut data UDP.
Koneksi	Koneksi harus diatur sebelum dilakukannya transmisi data	Data akan ditransmisikan sebelum membuat koneksi pada <i>traffic agent</i> ini.
Interface Data	Data dikirim dalam bentuk aliran <i>byte</i> , dan tidak mempertahankan batas data	Paket dikirim secara independen dan untuk integritas, mereka diverifikasi hanya jika tercapai, dan mempertahankan batas data.
Keandalan	TCP menyediakan pengiriman data yang terjamin.	Tidak memberikan jaminan pengiriman data, oleh karena itu dikatakan <i>unreliable</i> pada <i>traffic</i> UDP.
Pengurutan Data	Pengiriman pesan akan berada dalam urutan yang sama seperti saat dikirim.	Paket datagram dapat tiba dalam urutan apa pun
Pengiriman Ulang Data	Data dapat dikirimkan ulang apabila ada data yang hilang	Data tidak dikirimkan ulang.
Kecepatan Transmisi	TCP lebih lambat dibandingkan dengan UDP	UDP lebih cepat dikarenakan tidak ada pengecekan kembali data

2.6 Quality of Service (QoS)

Quality of Service (QoS) merepresentasikan kemampuan suatu sistem jaringan untuk mengukur seberapa baik kinerja jaringan dalam melayani *user*-nya. QoS dapat juga dimaknai dengan metode pengukuran untuk menjelaskan karakteristik atau sifat suatu layanan jaringan pada segi lalu lintas jaringan data. Tujuan utama QoS yaitu memberikan prioritas pada sekumpulan parameter uji untuk mengukur kinerja pada suatu konfigurasi jaringan. Parameter QoS yang digunakan dalam penelitian ini adalah *Throughput*, *packet delivery ratio* dan *residual energy*. berdasarkan skenario simulasi yang telah dibuat (Nurusshobah, Trisnawan and Amron, 2019).

A. Throughput

Throughput adalah jumlah total paket data atau *bandwidth* aktual dari paket data yang diterima ke *node* tujuan selama waktu simulasi pada simulasi jaringan MANET. *Throughput* bisa dikatakan juga sebagai jumlah total paket yang berhasil dikirim ke tujuan selama waktu simulasi. Satuan yang digunakan adalah Kbps dan paket yang dikirimkan memiliki satuan *byte*. Protokol *routing* dengan *throughput* tinggi lebih efisien. (Jaiswal and Bhadauria, 2015).

$$TO = \frac{\text{Paket yang diterima}}{\text{waktu simulasi}} \times \left(\frac{8}{1000}\right) \quad (2.8)$$

B. Packet Delivery Ratio

Packet delivery ratio adalah perbandingan banyaknya paket yang diterima dengan paket yang dikirimkan pada suatu jaringan MANET. Semakin tinggi *packet delivery ratio* maka akan semakin bagus jaringan tersebut. Satuan yang digunakan pada *packet delivery ratio* adalah persen. (Jaiswal and Bhadauria, 2015).

$$PDR = \frac{\text{Paket yang diterima}}{\text{Paket yang dikirim}} \times 100\% \quad (2.9)$$

Tabel 2. 2 Parameter *Packet delivery ratio* (ETSI, 1999)

Kategori <i>Packet delivery ratio</i>	<i>Packet Delivey Ratio (%)</i>
Sangat Bagus	100% - 98%
Bagus	97% - 86%
Sedang	85% - 76%
Kurang Bagus	>75 %

C. Energy Model

Pada pencarangannya energy model berperan dalam menginisiasikan daya dan energi yang dibutuhkan pada setiap *user* agar dapat berjalan dan aktif dalam mengirimkan maupun meneruskan data ke tujuan yang diinginkan. Adapun beberapa parameter energy model yang akan digunakan dalam perancangan jaringan MANET. Menurut standar IEEE 802.11 MAC pada saat berada dalam posisi idle tenaga yang dibutuhkan berkisar 800 mW. Dalam penerimaan dan pengiriman paket data dibutuhkan tenaga yang berkisar antara 1000 mW – 1300 mW (Can *et al.*, 2000; Bello, 2013).

1. *Transmit energy (Tx)* : *Energy* yang dibutuhkan pada posisi saat *node* mentransmisikan atau mengirimkan data ke *node* yang akan dituju pada perancangan jaringan MANET (*Mobile ad-hoc Network*) pada simulasi yang akan digunakan nantinya pada jaringan MANET (Sultan and Zaki, 2017).

$$Tx \text{ Energy} = \frac{Tx \text{ Power} \times Packet \text{ Size}}{2 \times 10^6} \quad (2.10)$$

2. *Received energy (Rx)* : *Energy* yang dibutuhkan pada posisi saat *node* dalam keadaan menerima data dari *node* atau *user* pada perancangan jaringan MANET (Sultan and Zaki, 2017).

$$Rx \text{ Energy} = \frac{Rx \text{ Power} \times Packet \text{ Size}}{2 \times 10^6} \quad (2.11)$$

- www.itk.ac.id
3. *Idle Energy* : *Energy* yang dibutuhkan pada posisi saat *node* dalam keadaan stand by atau tidak dalam posisi mengirimkan data maupun meneruskan data (Sultan and Zaki, 2017).
 4. *Sleep* : *Energy* yang dibutuhkan pada posisi saat *node* dalam keadaan mati sehingga *node* tidak dapat menerima sinyal (Sultan and Zaki, 2017).

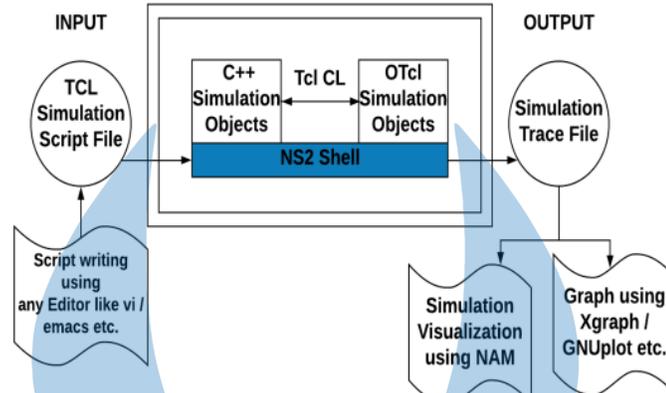
Quality of Service yang digunakan dalam *energy* model adalah *residual energy* yang merupakan *energy* yang tersisa dari simulasi yang telah dilakukan pada jaringan MANET. Adapun persamaan yang digunakan dalam *residual energy* sebagai berikut (Bello, 2013).

$$\text{Residual Energy} = \text{Energi Awal} - \text{Konsumsi Energi} \quad (2.12)$$

2.7 Network Simulator

Network simulator merupakan salah satu simulator jaringan yang paling banyak digunakan yang digunakan saat ini dalam perancangan maupun pemodelan jaringan. *Network Simulator* merupakan simulator jaringan kejadian diskrit berorientasi objek yang awalnya dikembangkan di *Lawrence Berkeley Laboratory* di University of California, Berkeley, sebagai bagian dari proyek *Virtual InterNetwork Testbed* (VINT).

Aplikasi ini dirancang untuk komunitas riset jaringan untuk mensimulasikan algoritma *routing*, *multicast*, dan protokol TCP / IP. Berkeley merilis kode awal yang memungkinkan simulasi jaringan nirkabel dapat dilakukan di *Network Simulator*. Adapun Proyek *Monarch* di *Carnegie Mellon University* telah memperbaiki dan melakukan riset pada *Network Simulator*. (Sarkar and Halim, 2008).



Gambar 2.17 Arsitektur network simulator (L., J. and J., 2018)

2.8 Penelitian Terdahulu

Penelitian Terdahulu merupakan rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang telah dilakukan.

Peneliti	Judul	Metode dan Hasil
Sharma and Gupta, 2012	<i>Comparison based Performance Analysis of UDP/CBR and TCP/FTP Traffic under AODV Routing Protokol in MANET</i>	Metode : Merancang jaringan MANET menggunakan software NS 2.34 dengan menggunakan protokol AODV dan DSDV dengan model Traffic Agent TCP/FTP dan UDP/CBR . Jumlah <i>node</i> yang digunakan 30 <i>nodes</i> . Hasil : Diketahui bahwa nilai <i>Throughput</i> TCP/FTP lebih baik . <i>Packet delivery ration</i> UDP/CBR memiliki nilai yang lebih baik namun TCP/FTP lebih dapat diandalkan pada setiap percobaan simulasi pada jaringan MANET.
Dash, 2013	<i>Impact of Propagation models on Distance Vector Routing Protokols in MANET</i>	Metode : Merancang jaringan MANET menggunakan software NS 3 dengan

menggunakan protokol AODV dan DSDV dengan model propagasi *free space, two ray ground dan shadowing*.

Hasil : Terlihat bahwa model *two ray ground* mengungguli dalam semua kasus dalam hal rasio pengiriman paket, energi yang dikonsumsi dan analisis penundaan. Namun kinerja protokol DSDV ditemukan lebih baik daripada AODV mengingat model *two ray ground*. Masih dalam beberapa kasus dengan jumlah nodus yang lebih tinggi, model bayangan berkinerja lebih baik.

Jiatmoko dkk, 2015 Simulasi Jaringan MANET Dengan NS3 Untuk Membandingkan Performa Routing Protokol AODV dan DSDV

Metode : Merancang jaringan MANET menggunakan *software NS 3* dengan menggunakan protokol AODV dan DSDV dengan skenario variasi *node* (20,30 dan 50) dengan koneksi antar *node* statis dan koneksi diujikan sebanyak 1,3 dan 5.

Hasil : Pada skenario variasi *node* didapatkan nilai *Throughput* AODV sebesar 1.75563168 dan DSDV sebesar 1.48138038 dan nilai delay AODV sebesar 15.653937 dan DSDV sebesar 4.923451. Untuk PDR AODV sebesar 87.17% dan DSDV sebesar 81.83%. Pada skenario penambahan koneksi nilai *Throughput* AODV sebesar 1.307329 dan DSDV sebesar 1.342511, untuk delay AODV sebesar 8.364059 dan DSDV sebesar 2.472702, untuk PDR nilai AODV sebesar 83.86% dan DSDV sebesar 82.48%.

Sharma dkk, 2016 *Performance Comparison of AODV, ZRP and AODVDR Routing Protocols in MANET*

Metode : Merancang jaringan MANET menggunakan software NS 3 dengan menggunakan protokol AODV, AODVDR dan ZRP dengan skenario variasi *connection* (5,10, 15, 20 dan 25).

Hasil : ZRP berkinerja lebih baik dari AODV, AODVDR dan AODV memberikan AODVDR kinerja yang lebih baik ketika kemungkinan jumlah koneksi jaringan dari setiap *node* kurang dari 25% dari total *node* dalam jaringan. Tetapi ketika koneksi jaringan lebih besar dari 40% dari total *node* di jaringan, AODVDR memberikan kinerja yang lebih baik daripada dua protokol perutean lainnya. Ketika koneksi jaringan antara 25% hingga 40%, sangat sulit untuk mengatakan protokol mana yang berkinerja terbaik

Ali dkk, 2017 *Comparing and Analyzing Reactive Routing Protocols (AODV, DSR and TORA) in QoS of MANET*

Metode : Merancang jaringan MANET menggunakan software NS 3 dengan menggunakan protokol AODV, DSR dan TORA dengan skenario variasi *node* (25,50, 75 dan 100).

Hasil : Didapatkan bahwa *Network Load, Retransmission Attempts, End to End Delay, Media Access Delay, Throughput, and etc.* Hasilnya, kami menemukan bahwa AODV mengungguli protokol lain dalam hal penundaan akses media, pemuatan, penundaan, percobaan ulang data drop, dan transmisi ulang upaya.

**Appiah,
2017**

*Performance Comparison Of
Mobility models In Mobile Ad-
hoc Network (Manet)*

Metode : Merancang jaringan MANET menggunakan software NS 3 dengan menggunakan protokol OLSR dengan variasi model propagasi *random way point* dan MANET *down left* dengan skenario variasi *node* (5,10, dan 30) dan variasi *size packet* (64,256 dan 512)

Hasil : RWP akan bekerja lebih baik di MANET bila digunakan dengan OLSR di lingkungan di mana waktu jeda adalah 5 atau 15 detik dan kecepatan mobilitas 5-10m / s atau 25-30m / s. RWP OLSR 2 memiliki kinerja yang lebih baik daripada MDL OLSR 2 dengan memberikan 89% *Throughput* dan penundaan yang lebih rendah sebesar 42%

**Bai dkk,
2017**

*Performance Comparison and
Evaluation of the Proactive and
Reactive Routing Protocols for
MANETs*

Metode : Merancang jaringan MANET menggunakan software NS 3 dengan menggunakan protokol AODV, DSDV dan FSR dengan skenario variasi *node* (5,10, dan 30) dan variasi *size packet* (64,256 dan 512).

Hasil : Hasil pengujian menunjukkan bahwa algoritma AODV dari tipe reaktif berkinerja lebih baik dalam hal *Throughput* dan penundaan end-to-end rata-rata, sedangkan DSR dari tipe reaktif sedikit lebih baik di antara algoritma perutean dalam hal rasio pengiriman paket. Ketika ukuran jaringan meningkat, protokol reaktif (terutama algoritma perutean AODV) menjadi dominan di semua

kategori kinerja, sedangkan pengaruh ukuran paket tidak signifikan.

Rangaraj dan Anitha, 2017 *Performance Comparison Of Mobility models In Mobile Ad-hoc Network (Manet)*

Metode : Merancang jaringan MANET menggunakan software NS dengan menggunakan protokol AODV dengan variasi model propagasi *random way point, manhattan,* dan *pursue mobility* dengan skenario 100 node.

Hasil : Kinerja protokol dievaluasi untuk model *Random way point (RWP), Manhattan Mobility (MM)* dan *Pursue Mobility (PM)* dalam hal *Throughput*, rasio pengiriman paket, penundaan, energi dan overhead. Perbandingan hasil simulasi menggambarkan bahwa kinerja protokol *routing* sangat bervariasi di berbagai model mobilitas.

Alamsyah dkk, 2018 Analisis Kinerja Protokol *Routing* Reaktif dan Proaktif pada MANET Menggunakan NS2

Metode : Merancang jaringan MANET menggunakan software NS 2 dengan menggunakan protokol AODV, DSR, DSDV dan OLSR dengan skenario variasi *node* (25,50,75,100,150 dan 200)

Hasil : Didapatkan Rata-rata PDR pada OLSR adalah sebesar 39,997%, DSR sebesar 39,197%, AODV sebesar 33,06%, dan DSDV sebesar 31,24%. Rata-rata kinerja *Throughput* pada OLSR adalah sebesar 417,383 Kbps, DSR sebesar 245,2 Kbps, AODV sebesar 280,917 Kbps, dan DSDV sebesar 253,25 Kbps. Rata-rata *packet loss* pada OLSR adalah sebesar 60,003%, DSR sebesar 60,803%, AODV

		sebesar 62,635%, dan DSDV sebesar 68,76%. Rata-rata nilai delay pada OLSR adalah sebesar 15,52 milidetik, DSR sebesar 60,31 milidetik, AODV sebesar 39,07 milidetik, dan DSDV sebesar 24,63 milidetik.
Alamsyah dkk, 2018	<i>Performance Comparative of AODV, AOMDV and DSDV Routing Protokols in MANET Using NS2</i>	Metode : Merancang jaringan MANET menggunakan software NS 2 dengan menggunakan protokol AODV,AOMDV, dan DSDV dengan skenario variasi <i>node</i> (25,50,75,100,150 dan 200) Hasil : Didapatkan Rata-rata PDR AOMDV adalah 60.09%, AODV adalah 57.42%, dan DSDV adalah 38.86%. Didapatkan rata-rata <i>Packet Loss</i> AOMDV adalah 39.9%, AODV adalah 42.58%, dan DSDV adalah 61.14%. Didapatkan rata-rata <i>Delay</i> DSDV adalah 27.8 milidetik, AOMDV adalah 61.19 milidetik, dan AODV adalah 102.83 milidetik. Didapatkan rata-rata <i>Routing Overhead</i> AODV adalah 1809 paket, DSDV adalah 139586 paket, dan AOMDV adalah 171464 paket.
Chbib dkk, 2018	<i>Improvement of OLSR Protokol by Using Back Up MPR and Routing Table Mechanisms</i>	Metode : Merancang jaringan MANET menggunakan software NS 2 dengan menggunakan protokol AODV,AOMDV, dan DSDV dengan skenario variasi <i>node</i> (20, 30 dan 40) Hasil : Didapatkan Rata-rata PDR OLSR adalah 58%, sedangkan pada OLSR <i>Backup</i> mencapai 63%. Didapatkan rata-rata <i>Routing Overhead</i>

dengan 20 *node*, rasio OLSR adalah 7%, sedangkan OLSR Cadangan menurun menjadi 6,1%. Ketika jumlah *node* meningkat menjadi 30, rasio overhead meningkat di OLSR menjadi 9,3% sedangkan di Cadangan mencapai 8,8%. Didapatkan rata-rata *Throughput* Dengan 20 *node*, kecepatan OLSR adalah 197 kb / s, sedangkan protokol OLSR cadangan telah mencapai 216 kb / s. Ketika jumlah *node* meningkat menjadi 30 *Throughput* meningkat di kedua protokol (OLSR 221 kb / s dan *Backup* OLSR 228 kb / s).

Nisac dkk, Perbandingan Performansi
2018 Protokol BATMAN dan AODV
Pada Jaringan MANET (*Mobile Ad-hoc Network*)

Metode : Merancang jaringan MANET menggunakan *software* NS 2 dengan menggunakan protokol BATMAN dan AODV dengan skenario variasi *node* (20, 30 dan 40)

Hasil : Didapatkan Rata-rata nilai dari protokol AODV didapatkan nilai rata-rata *delay* sebesar 59,71 ms nilai *delay* dan untuk rata-rata *packet loss* sebesar 4,57% masuk dalam klasifikasi dengan kualitas sedang dengan *packet loss* kurang dari 15%. Sedangkan nilai rata-rata waktu *handover* didapatkan sebesar 927,974 ms. Didapatkan Rata-rata nilai dari protokol BATMAN didapatkan nilai rata-rata *delay* sebesar 93,96 ms nilai, untuk rata-rata *packet loss* sebesar 11,11% masuk dalam klasifikasi dengan kualitas sedang dengan *packet loss* kurang dari 15%.

Sedangkan nilai rata-rata waktu *handover* didapatkan sebesar 1685,55 ms.

Taufan, 2018	Analisis Perbandingan Kinerja Protokol <i>Routing</i> OLSR dan DSDV Pada MANET Berdasarkan Pergerakan <i>Node</i>	Metode : Merancang jaringan MANET dengan menggunakan dua protokol proaktif, yakni OLSR dan DSDV dengan skenario pengujian berupa perbedaan jumlah <i>node</i> sebanyak 20,30,40 dan 50 <i>node</i> , luas area simulasi sebesar 200m ² , 500m ² , 800m ² dan 1000m ² , serta mode mobilitas <i>Random Waypoint</i> dan <i>Random Direction</i> . Hasil : Untuk masing-masing kinerja dari <i>average end to end delay</i> protokol DSDV lebih baik <i>disbanding</i> dengan protokol OLSR. Ditunjukan dengan nilai rata-rata <i>delay</i> dari protokol DSDV yaitu 0,00107591ms. Namun pada hasil pengukuran <i>packet delivery ratio</i> dan <i>routing overhead</i> OLSR lebih baik daripada DSDV disebabkan OLSR mempunyai <i>system</i> yang disebut <i>Multi poin relays (MPRs)</i> MPR dapat menurunkan banyaknya paket atau pesan informasi <i>broadcast</i> yang mempunyai informasi yang sama dan bisa menurunkan nilai dari paket <i>routing</i> .
Nurusshobah dkk, 2019	Analisis Kinerja Protokol <i>Routing Dynamic</i> MANET <i>On-Demand (DYMO)</i> dan <i>Cluster Based Routing Protokol (CBRP)</i> pada <i>Mobile Ad-hoc Network (MANET)</i>	Metode : Merancang jaringan MANET menggunakan <i>software NS</i> dengan menggunakan protokol AODV,AOMDV, dan DSDV dengan skenario variasi luas area, jumlah <i>node</i> , ukuran paket data, konvergensi. Hasil : 1. Skenario Jumlah <i>Node</i> , Didapatkan nilai

Throughput DYMO dan CBRP berturut-turut yaitu 102,09 Kbps dan 96,55 Kbps. Didapatkan nilai *end to end delay* DYMO dan CBRP adalah 73,55 ms dan 1517,56 ms. Didapatkan nilai *jitter* DYMO dan CBRP adalah 11,81 ms dan 259,21 ms. Didapatkan nilai *packet loss* DYMO dan protokol CBRP berturut-turut adalah 1,78 % dan 15,45 %.

2. Skenario Variasi Luas Wilayah, Didapatkan nilai *Throughput* DYMO dan CBRP berturut-turut yaitu 102,09 Kbps dan 96,55 Kbps. Didapatkan nilai *end to end delay* DYMO sebesar 18,98 ms, sedangkan protokol CBRP sebesar 19,98 ms. Didapatkan nilai *jitter* DYMO adalah 3,06 ms, sedangkan protokol CBRP adalah 3,71 ms. Didapatkan nilai *packet loss* DYMO adalah 0,35 %, sedangkan protokol CBRP adalah 0,36 %.

3. Skenario Variasi Paket data, Didapatkan nilai *Throughput* DYMO dan CBRP berturut-turut yaitu 102,78 Kbps dan 104,86 Kbps. Didapatkan nilai *end to end delay* DYMO sebesar 32,86 ms, sedangkan protokol CBRP sebesar 37,89 ms. Didapatkan nilai *jitter* DYMO sebesar 8,44 ms, sedangkan protokol CBRP sebesar 9,66 ms. Didapatkan nilai *packet loss* DYMO sebesar 0,53 %, sedangkan protokol CBRP sebesar 1,04 %.

4. Skenario Konvergensi
didapatkan nilai *packet loss*
yang lebih besar dari
DYMO, yaitu: 14,90 %
sedangkan protokol DYMO
14,07 %

