

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisis Sistem

Jenis jaringan *wireless* memiliki keterbatasan jangkauan transmisi, sehingga menyebabkan penggunaan *routing* dibutuhkan untuk mengirim data melalui jaringan. penggunaan *mobile node* dalam *wireless* sendiri menimbulkan masalah dalam *routing*. Protokol *routing* konvensional tidak didesain untuk topologi dinamis. Oleh karena itu muncullah berbagai jenis protokol *routing* yang mampu untuk mengatasi hal tersebut (Khristian, E. 2013).

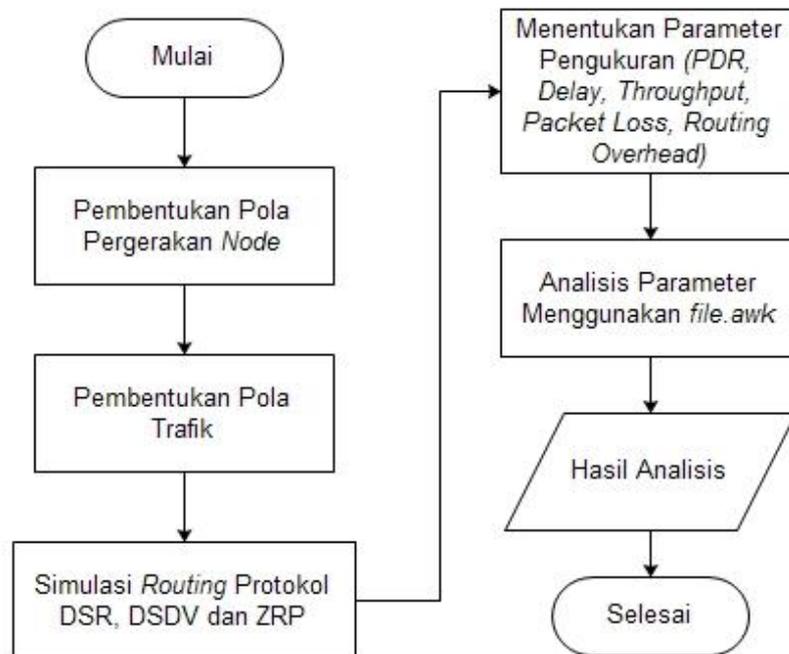
*Dynamic Source Routing* (DSR) dalam melakukan pengiriman paket data dari sumber ke tujuan memerlukan proses *route discovery* dan *route maintenance*. Pada *route discovery* terdapat dua kegiatan RREQ dan RREP. RREQ dilakukan ketika *node* sumber tidak mengetahui rute menuju *node* tujuan. RREP merupakan respon *node* tujuan ketika adanya permintaan rute dari *node* sumber. DSR termasuk dalam jenis *routing* protokol *reaktif*, dimana pada proses pencarian rute hanya dilakukan ketika *node* sumber membutuhkan komunikasi dengan *node* tujuan. *Destination Sequenced Distance Vector* (DSDV) dalam melakukan pengiriman paket data dari sumber ke tujuan dengan melihat rute berdasarkan tabel *routing* yang ada. Dalam tabel *routing* tersebut terdapat informasi rute menuju *node* tujuan. DSDV termasuk dalam jenis *routing* protokol *proaktif*, dimana pada protokol ini setiap *node* melakukan *broadcast* tabel *routing* ke semua *node* dalam jaringan walaupun tidak ada permintaan rute. *Zone Routing Protokol* (ZRP) bekerja berdasarkan zona dengan menggunakan konsep zona terbatas menggunakan fitur dari *routing* protokol *proaktif*, sedangkan zona luar menggunakan fitur dari *routing* protokol *reaktif*. ZRP terdiri dari dua *sub routing* protokol utama yaitu *Intra Zone Routing Protokol* (IARP) dan *Inter Zone Routing Protokol* (IERP). IARP mengacu pada jaringan padat yang menjadi batas dari zona *routing* protokol

*proaktif* sedangkan IERP mengacu pada jaringan zona luar dari *routing* protokol *reaktif*. IARP mempertahankan informasi topologi jaringan dengan selalu *update* jalur ketika *node* berada didalam zona. ZRP termasuk dalam jenis *routing* protokol *hybrid*, protokol ini merupakan kombinasi dari kedua tipe *routing* protokol yaitu *routing* protokol *proaktif* dan *routing* protokol *reaktif*. (Yanuar, G.C. 2016).

Ketiga jenis protokol *routing* tersebut memiliki karakteristik yang berbeda dalam melakukan proses *routing* karena adanya perbedaan jenis protokol *routing* sehingga memiliki kinerja yang berbeda. Maka diperlukan analisis parameter pengukuran untuk mengetahui kinerja dari protokol *routing* tersebut.

### 3.2 Hasil Analisis

Bahwa untuk mengetahui hasil kinerja dari *routing* protokol DSR, DSDV, dan ZRP melalui proses simulasi dengan *network simulator 2* yaitu berupa perancangan pembentukan pola pergerakan *node* dan pembentukan pola trafik. Selanjutnya dilakukan penghitungan terhadap parameter pengukuran menggunakan *file.awk* untuk mengetahui kinerjanya dari masing-masing *routing* protokol yang meliputi *packet delivery ratio*, *delay*, konsumsi energi, *packet loss*, dan *routing overhead*. Untuk lebih jelasnya dapat dilihat pada gambar 3.1 berikut ini :



**Gambar 3.1** Diagram Alir Analisis *Routing*

Penjelasan Gambar 3.1 :

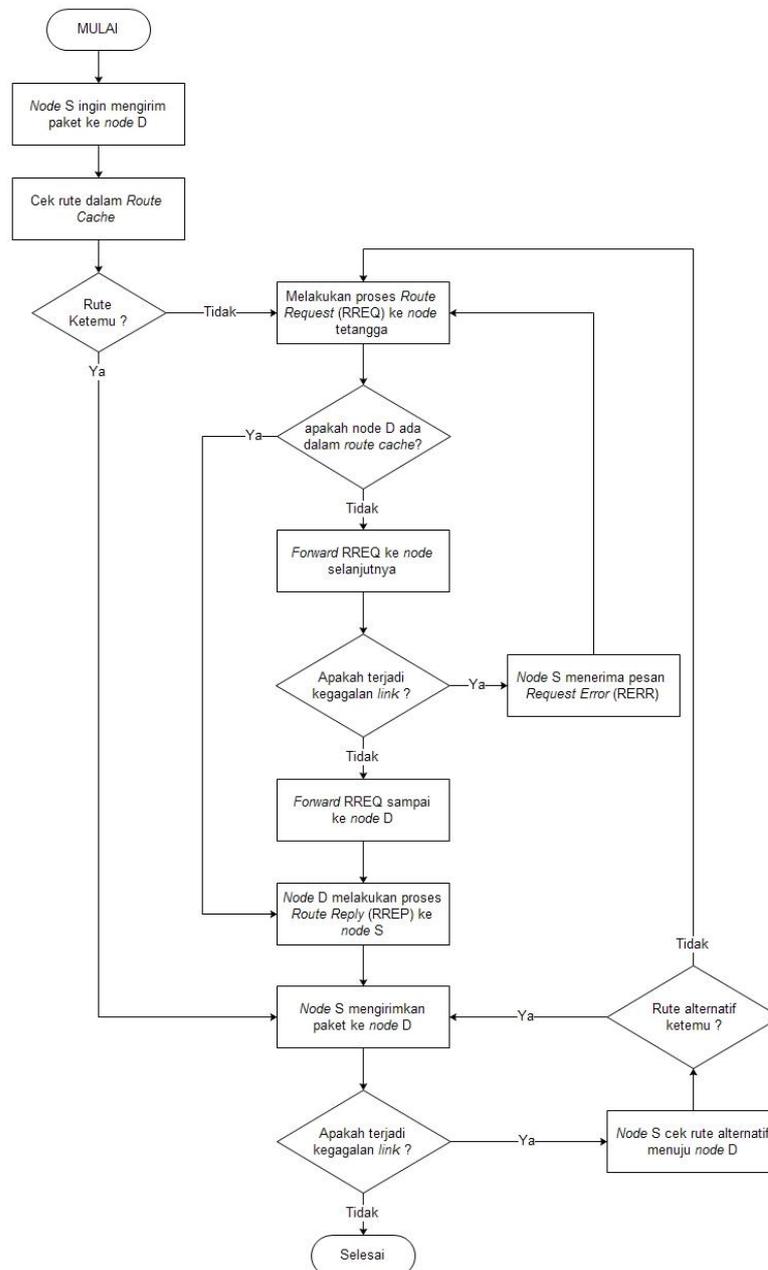
1. Dimulai dengan pembentukan pola pergerakan *node*, pada model ini *node-node* bergerak secara acak dan dinamis.
2. Pembentukan pola trafik ini untuk menentukan jumlah *node*, jumlah keseluruhan paket data yang ditransmisikan, juga menentukan *node* sumber yang akan mengirimkan paket data dan *node* tujuan yang akan menerima paket data.
3. Selanjutnya dilakukan proses simulasi *routing* protokol DSR, DSDV, dan ZRP yang nantinya akan menghasilkan *output* berupa *file.tr* dan *file.nam*. Dimana *file.tr* merupakan *file trace* yang digunakan analisa numerik dan *file.nam* merupakan tampilan grafis simulasi.
4. Dilakukan penentuan perhitungan parameter *Packet Delivery Ratio*, *Delay*, Konsumsi Energi, *Packet Loss*, dan *Routing Overhead* pada *file.tr*.

5. Selanjutnya dilakukan analisis perhitungan nilai parameter menggunakan *file.awk* terhadap *file.tr* yang nantinya akan menghasilkan angka untuk mengetahui performansi dari *routing* protokol DSR, DSDV, dan ZRP.

### **3.3 Perancangan Algoritma Routing**

Pada algoritma ini ada 3 jenis *routing* protokol yang akan dianalisis yaitu *routing* protokol DSR, DSDV, dan ZRP. Masing-masing dari *routing* protokol tersebut memiliki diagram alir yang berbeda. Berikut ini merupakan penjelasan algoritma tersebut :

### 3.3.1 Algoritma Routing DSR



**Gambar 3.2** Flowchart DSR

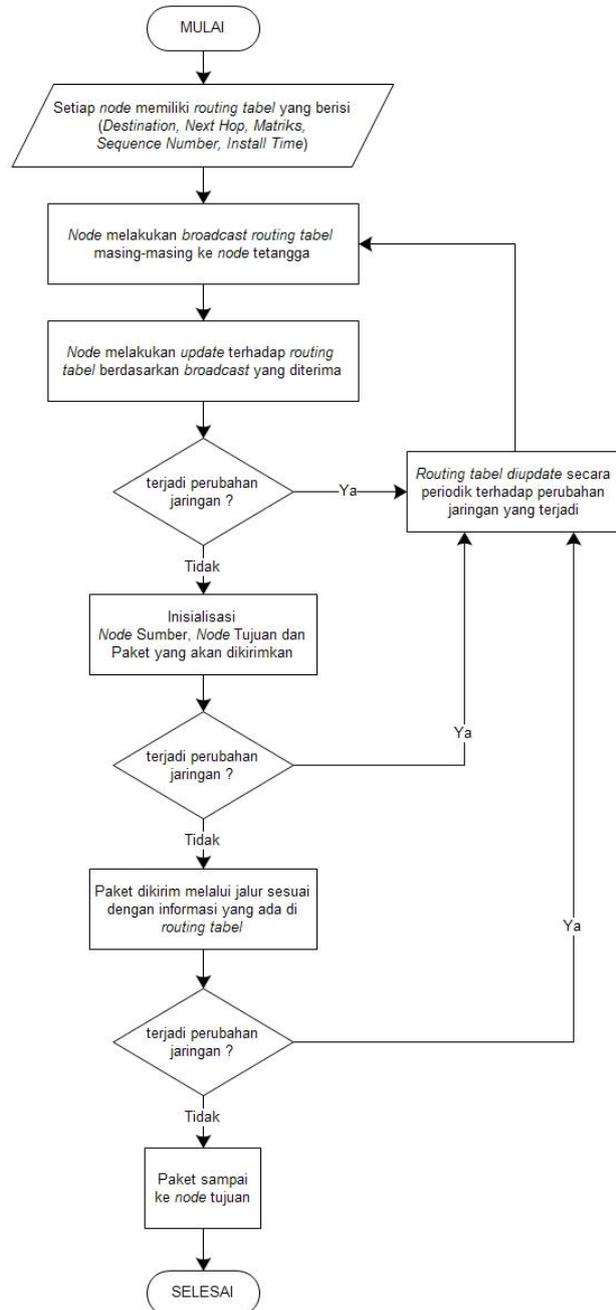
Penjelasan gambar 3.2:

1. Dimulai dari *node S* (*source*) yang ingin mengirimkan paket data ke *node D* (*destination*).
2. *Node S* mengecek *route cachenya* apakah terdapat rute menuju ke *node D*. Jika *node S* memiliki rute menuju *node D*, maka *node S* akan

mengirim paket data ke *node* D tanpa melakukan RREQ. Jika *node* S tidak memiliki rute ke *node* D, maka *node* S akan melakukan proses *Route Request* (RREQ) ke *node* tetangga.

3. *Node* yang menerima RREQ dari *Node* S akan mengetahui bahwa *Node* S ingin mengirimkan paket ke *node* D dengan mencantumkan *node source* dan *node destination* pada *packet header*nya. Masing-masing *node* tersebut mengecek apakah mengetahui keberadaan *node* D dengan mengecek *route cachenya*.
4. Jika ada yang memiliki *route cache* menuju *node* D, maka akan meneruskan RREQ ke *node* D dan *node* D akan melakukan proses *Route Reply* (RREP) menuju *node* S. Jika tidak ada, maka *node* yang menerima RREQ akan meneruskan RREQ ke *node* berikutnya sampai menuju *node* D.
5. Jika pada proses RREQ ditengah perjalanan terjadi kegagalan *link* atau terputusnya koneksi antar *node*, maka akan otomatis proses RREQ tersebut berubah menjadi *Route Error* (RERR) dan kembali menuju *node* S sehingga *node* S akan mengetahui jika terjadi kegagalan *link*.
6. Jika tidak terjadi kegagalan *link*, maka RREQ akan diteruskan sampai menuju *node* D dan melakukan proses RREP menuju *node* S.
7. *Node* S menerima RREP sehingga mengetahui jalur terpendek yang akan dilalui dalam pengiriman paket menuju *node* D.
8. *Node* S mengirimkan paket menuju *node* D, jika ditengah perjalanan pengiriman paket terjadi kegagalan *link* atau *link* terputus, maka *node* S akan mengirimkan paket melalui rute alternatif yang sudah tersimpan dalam *route cache*. Jika *node* tidak ada yang memiliki rute alternatif menuju *node* D, maka *node* S akan melakukan proses RREQ ulang.
9. Jika tidak terjadi kegagalan *link*, maka paket akan sampai ke tujuan yaitu *node* D.

### 3.3.2 Algoritma Routing DSDV



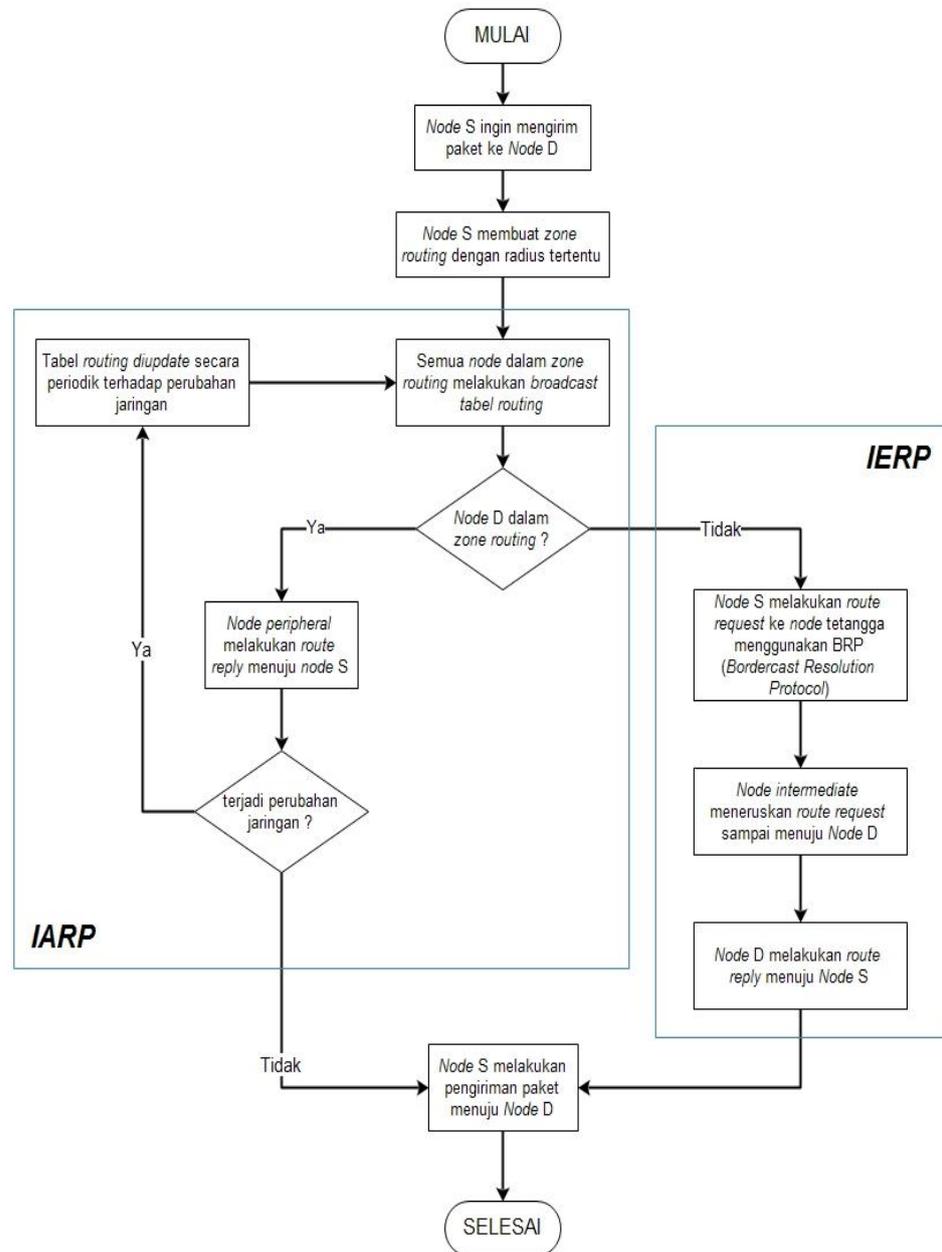
**Gambar 3.3** Flowchart DSDV

Penjelasan Gambar 3.3 :

1. Setiap *node* dalam jaringan memiliki *routing tabel* masing-masing yang berisi (*Node Destination, Next Hop, Matriks, Sequence Number* dan *Install Time*).

2. *Node* melakukan *broadcast routing tabel* ke *node* tetangga dengan tujuan agar semua *node* dalam jaringan tersebut mengetahui jarak antar *node*.
3. *Node* akan melakukan *update* terhadap *routing tabel* berdasarkan *broadcast* yang diterima dari *node* tetangga.
4. Jika terjadi perubahan jaringan, maka *node* tersebut akan melakukan *update* terhadap *routing tabelnya* dan melakukan *broadcast* ke *node tetangga* dengan tujuan agar *node* lain mengetahui *routing tabel* yang baru.
5. Inisialisasi jaringan dengan memberikan informasi tentang *node* sumber yang akan melakukan pengiriman paket data menuju *node* tujuan tertentu dengan menyertakan paket data yang akan dikirim.
6. Jika terjadi perubahan jaringan, maka *node* tersebut akan melakukan *update* terhadap *routing tabelnya* dan melakukan *broadcast* ke *node* tetangga dengan tujuan agar *node* lain mengetahui *routing tabel* yang baru.
7. Paket akan dikirim melalui jalur sesuai dengan informasi yang ada dalam *routing tabel*.
8. Jika terjadi perubahan jaringan, maka *node* tersebut akan melakukan *update* terhadap *routing tabelnya* dan melakukan *broadcast* ke *node* tetangga dengan tujuan agar *node* lain mengetahui *routing tabel* yang baru.
9. Paket akan diteruskan sampai menuju ke *node* tujuan.

### 3.3.3 Algoritma Routing ZRP



**Gambar 3.4** Flowchart ZRP

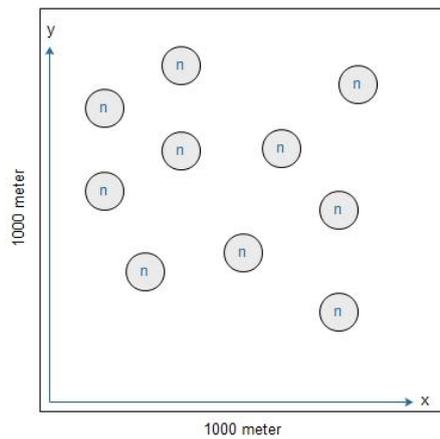
Penjelasan Gambar 3.4 :

1. Dimulai dari *node S* (Source) ingin mengirimkan paket menuju *node D* (Destination).
2. *Node S* membuat zona *routing* dengan radius *hop* (loncatan) tertentu.

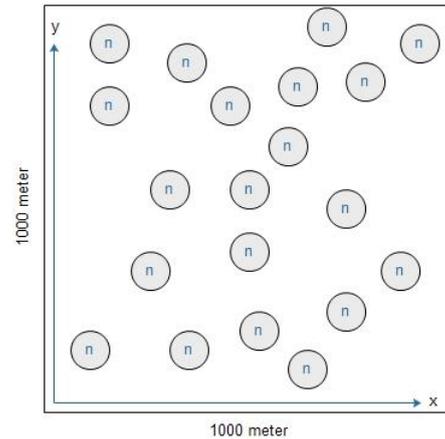
3. Pada wilayah dalam zona *routing* ini disebut dengan *Intra Zone Routing Protocol* (IARP).
4. Dalam IARP, semua *node* yang tercakup di zona *routing* melakukan proses *broadcast* tabel *routing* ke semua *node* agar memiliki informasi rute dalam zona tersebut.
5. Jika *node* D berada didalam zona *routing* tersebut, maka *node peripheral* akan memberi informasi kepada *node* S dengan cara melakukan *route reply*.
6. Jika terjadi perubahan jaringan maka tabel *routing* akan diupdate secara *periodic* dengan cara setiap *node* melakukan *broadcast* tabel *routing* untuk memperbarui informasi.
7. Jika tidak terjadi perubahan jaringan maka *node* S akan melakukan pengiriman paket menuju *node* D dan selesai.
8. Jika *node* D berada diluar zona *routing*, maka perlu dilakukan pencarian rute dimana proses ini termasuk dalam wilayah tanggung jawab *Inter Zone Routing Protocol* (IERP).
9. Dalam IERP, *node* S melakukan proses pencarian rute dengan cara melakukan proses *route request* ke *node* tetangga menggunakan layanan BRP (*Bordercast Resolution Protocol*).
10. *Node intermediate* atau *node* yang menerima *route request* akan meneruskannya sampai menuju ke *node* D.
11. Jika *route request* sudah sampai pada *node* D, selanjutnya *node* D akan membalas dengan melakukan *route reply* menuju *node* S dengan menyertakan informasi berupa rute mana saja yang dapat dilewati.
12. *Node* S akan memutuskan untuk memilih melewati rute yang terpendek dan mengirimkan paket menuju ke *node* D.
13. Paket akan sampai ke *Node* D dan selesai.

### 3.4 Desain Topologi Simulasi

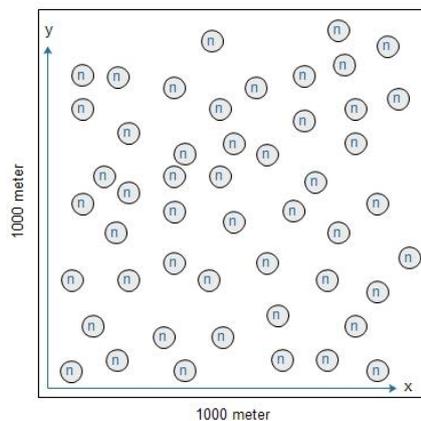
Desain topologi simulasi yang digunakan dibagi menjadi 4 skenario yaitu topologi 10 *node*, 20 *node*, 50 *node*, dan 100 *node*. Desain topologi diilustrasikan pada gambar 3.5, gambar 3.6, gambar 3.7, dan gambar 3.8 :



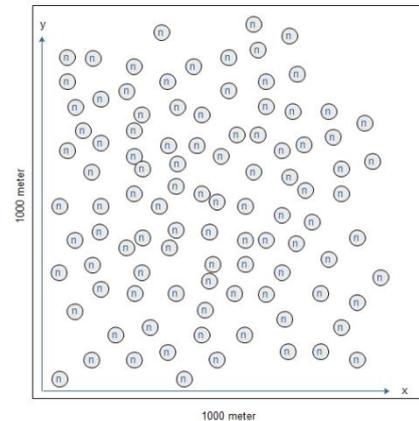
**Gambar 3.5** Topologi 10 *Node*



**Gambar 3.6** Topologi 20 *Node*



**Gambar 3.7** Topologi 50 *Node*



**Gambar 3.8** Topologi 100 *Node*

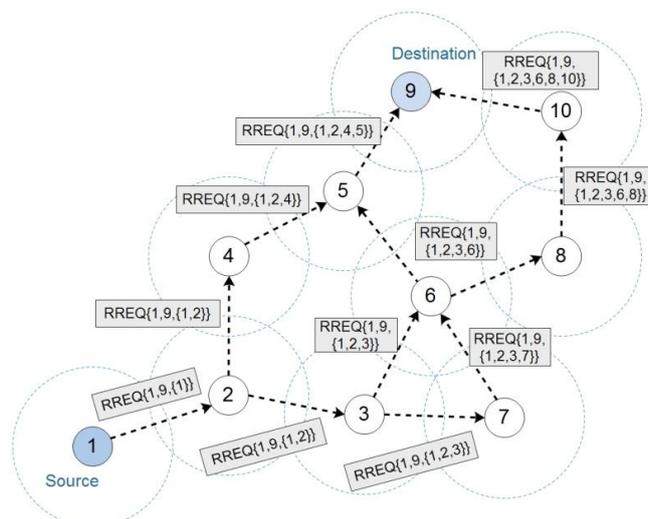
Desain topologi sistem seperti yang diilustrasikan pada gambar 3.5, gambar 3.6, gambar 3.7 dan gambar 3.8, masing-masing posisi *nodenya* ditentukan secara random. Jadi, untuk masing-masing percobaan akan menghasilkan posisi-posisi *node* yang berbeda. Akan tetapi, *node* sumber dan *node* tujuannya sudah ditentukan.

### 3.5 Arsitektur Jaringan

Pada jaringan *Mobile Ad hoc Network*, *node* dapat berubah-ubah secara dinamis sesuai dengan perubahan topologi jaringan yang terjadi. *Node-node* pada MANET dapat berperan sebagai *transmitter* dan *router*. Dimana setiap *node* memiliki tanggung jawab untuk meneruskan paket sampai ke *node* tujuan. Jadi semua *node* disini dapat berfungsi sebagai *router*.

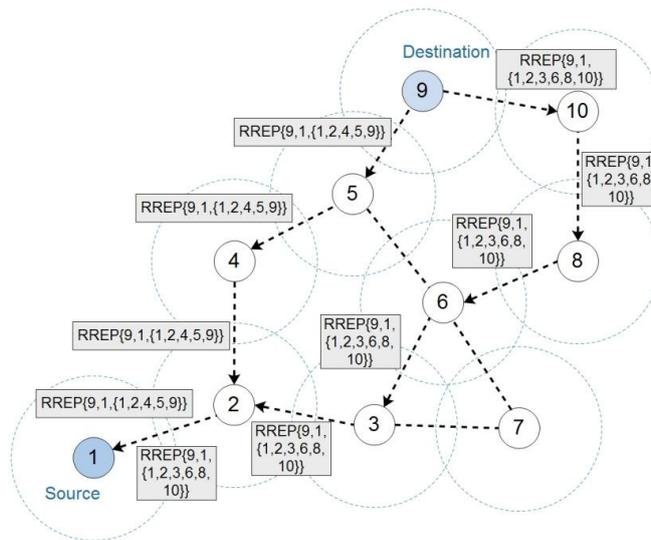
#### 3.5.1 Proses *Routing* Protokol DSR

Pada proses *routing* protokol DSR terdapat dua proses, yaitu proses *route request* dan proses *route reply* yang akan dijelaskan pada gambar 3.9 dan gambar 3.10 berikut ini :



**Gambar 3.9** Proses *Route Request*

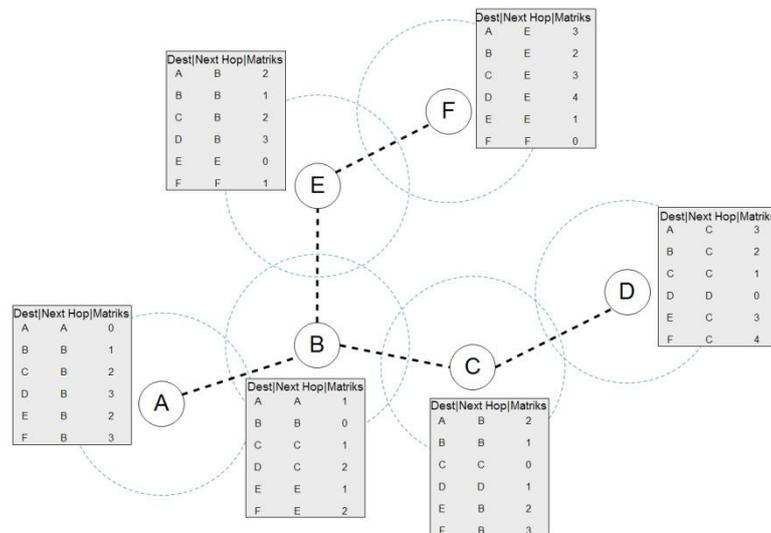
Gambar 3.9 menunjukkan bahwa *node* nomer 1 melakukan proses *route request* dengan membroadcast menuju *node* 2, *node* 2 melanjutkan *route request* sampai menuju *node destination* menuju *node* 9 dengan melewati *node* 1,2,4,5 dan 9. Rute lain menuju *node destination* yaitu dengan melewati *node* 1,2,3,6,8,10 dan 9.



**Gambar 3.10** Proses *Route Reply*

Gambar 3.10 menunjukkan bahwa *node destination* 9 melakukan proses *route reply* menuju *node source* yaitu node 1. RREP dari *node* 9 melewati rute 9,5,4,2 dan 1. Terdapat rute lain menuju *node* 1 yaitu dengan melewati *node* 10,8,6,3,2 dan 1. *Node source* akan mengirim paket dengan melewati *node* terpendek.

### 3.5.2 Proses Routing Protokol DSDV

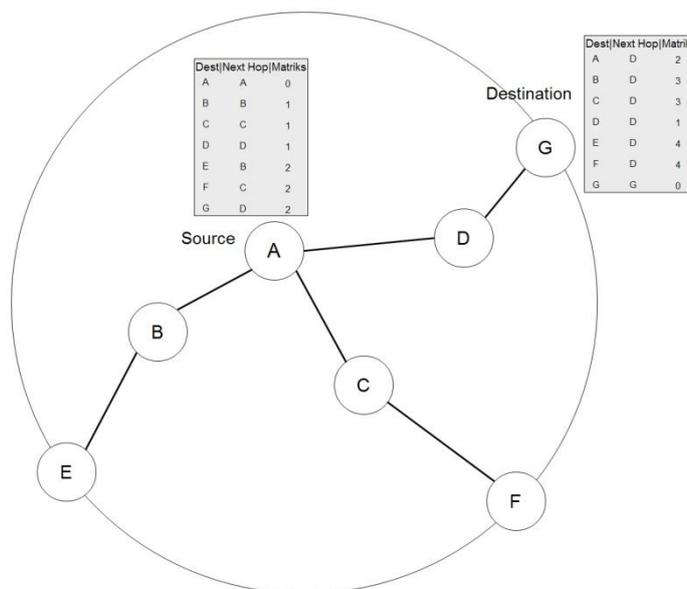


**Gambar 3.11** Proses *Routing DSDV*

Gambar 3.11 menunjukkan proses *routing* protokol DSDV dengan cara melakukan *broadcast routing* tabel ke seluruh *node* dalam jaringan. Jadi semua *node* dalam jaringan memiliki informasi rute menuju *node destination*. DSDV selalu memelihara *routing* tabel saat terjadi perubahan jaringan walaupun tidak ada permintaan rute.

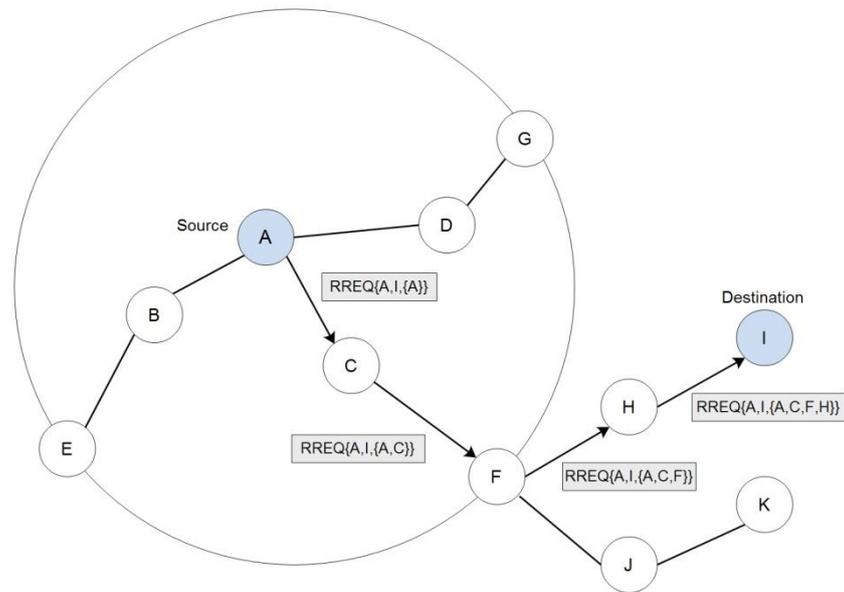
### 3.5.3 Proses Routing Protokol ZRP

Pada proses ini ZRP mempunyai dua *sub protokol*, yaitu *Intra Zone Routing Protokol* (IARP) dan *Inter Zone Routing Protokol* (IERP). Penjelasan terdapat pada gambar 3.12, gambar 3.13 dan gambar 3.14 berikut ini :



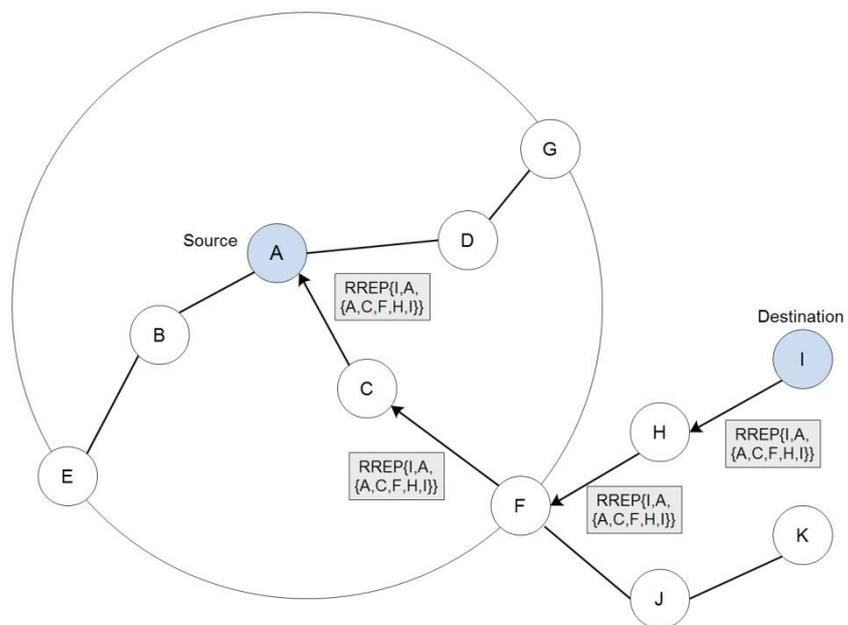
**Gambar 3.12** Proses IARP

Gambar 3.12 menjelaskan proses *Intra Zone Routing Protokol* (IARP), proses ini mengacu pada karakteristik dari *routing* protokol *proaktif* dengan cara melakukan *broadcast* ke seluruh *node* dalam zona *routing* sehingga *node* dalam cakupan zona *routing* mengetahui informasi rute dalam zona *routing* tersebut.



**Gambar 3.13** Proses IERP *Route Request*

Gambar 3.13 menunjukkan proses *route request* pada *Inter Zone Routing Protokol* (IERP). Proses ini dilakukan karena *node destination* berada diluar zona *routing*. Proses ini mengacu pada karakteristik *routing* protokol *reaktif* dengan cara melakukan RREQ menuju *node destination* hanya ketika terjadi permintaan rute.



**Gambar 3.14** Proses IERP *Route Reply*

Gambar 3.14 menunjukkan proses route reply pada *Inter Zone Routing Protokol* (IERP). Proses ini dilakukan ketika *node destination* melakukan balasan terhadap RREQ yang diterima dari *node source* dengan memberikan RREP menuju *node source*.

### 3.6 Skenario Simulasi

Simulasi dilakukan menggunakan skenario jaringan *wireless* standart IEEE 802.11n dengan jangkauan 250 m. Jika sebuah *node* memiliki jarak lebih dari 250 meter atau diluar jangkauan transmisi, maka *node* tersebut tidak dapat berkomunikasi dengan *node* yang lain. Variasi pola trafik yang akan digunakan dalam mengukur performa *routing* protokol dalam penelitian ini, antara lain :

1. Kapasitas jaringan : Ukuran area simulasi yang digunakan yaitu dengan luas area  $1000 \text{ m}^2 \times 1000 \text{ m}^2$  .
2. Pola pergerakan *node* : dalam simulasi jaringan *ad hoc* yaitu menggunakan random *waypoint*. Pada model ini *node-node* bergerak secara acak menuju arah tujuannya dengan kecepatan maksimal sampai 10m/s. Paket terdiri dari file *setdest.h* dan *setdest.cc*. Dengan penjelasan parameter :
  - -n : menyatakan jumlah *node* dalam jaringan
  - -p : menunjukkan jeda waktu sebuah *node* dalam keadaan tidak bergerak
  - -M : maksimum kecepatan *node* dalam bergerak
  - -t : durasi waktu simulasi
  - -x : ukuran (dalam sumbu x) topologi jaringan
  - -y : ukuran (dalam sumbu y) topologi jaringan
3. Pola trafik : menunjukkan jumlah keseluruhan paket data yang ditransmisikan setiap detik. *Packet rate* bernilai 1 paket/detik dengan ukuran paket 512 *bytes*, yang artinya setiap detik akan ditransmisikan 1 paket yang berukuran 512 *bytes*. Jenis trafik data yang digunakan adalah *Constant Bit Rate* (CBR). Dengan pemodelan CBR, berarti setiap sumber trafik yang berada pada suatu jaringan akan mengirimkan datanya secara

terus menerus. Dalam menggunakan *script* ini, kita harus menentukan beberapa parameter yang digunakan untuk simulasi. Parameter tersebut antara lain :

- Tipe koneksi yang digunakan
- Jumlah *node* dalam simulasi
- Jumlah *seed*
- Maksimum koneksi yang dikehendaki
- Paket *rate*

### 3.7 Parameter Pengukuran

Berikut ini merupakan beberapa parameter yang digunakan untuk menguji performansi *routing* protokol DSR, DSDV, dan ZRP yaitu (Sasongko, S.A., dkk. 2012) :

#### 1. *Packet Delivery Ratio* (PDR)

*Packet Delivery Ratio* merupakan perbandingan banyaknya jumlah paket yang diterima oleh *node* penerima dengan total paket yang dikirimkan dalam suatu periode waktu tertentu atau bisa juga dihitung dengan cara mengurangi jumlah paket keseluruhan yang dikirim dengan paket yang hilang. Secara sistematis *Packet Delivery Ratio* dapat dihitung dengan persamaan berikut :

$$\text{PDR} = \frac{\sum \text{data yang diterima}}{\sum \text{data yang dikirim}} \times 100\%$$

#### 2. *Delay*

*Delay* adalah jumlah total waktu pengiriman paket dalam satu kali pengamatan. Dalam hal ini satu kali simulasi dibagi dengan jumlah usaha pengiriman yang berhasil dalam satu kali pengamatan tersebut. Secara umum *delay* rata-rata dapat dinyatakan dengan persamaan berikut :

$$\text{Average Delay} = \frac{\text{total delay}}{\text{total paket yang diterima}}$$

Dimana

$$\text{Total Delay} = \text{Waktu Paket Diterima} - \text{Waktu Paket Dikirim}$$

### 3. *Konsumsi Energi*

Merupakan jumlah energi yang dibutuhkan oleh *node* untuk melakukan proses transmisi data pada sebuah jaringan. Secara umum konsumsi energi dinyatakan dalam persamaan berikut :

$$\text{Energi} = \frac{\text{total jumlah energi}}{\text{jumlah node}} \times 100$$

### 4. *Packet Loss*

*Packet Loss* merupakan banyaknya jumlah paket yang hilang selama proses pengiriman paket dari node asal menuju node tujuan. (Sidharta, Y., dkk. 2013). *Packet loss* dapat dihitung dengan persamaan berikut :

$$\text{Packet Loss} = \frac{\Sigma \text{ paket yang dikirim} - \Sigma \text{ paket yang diterima}}{\Sigma \text{ paket yang dikirim}} \times 100\%$$

### 5. *Routing Overhead*

*Routing overhead* merupakan rasio antara jumlah paket *routing* dengan paket data yang berhasil diterima. *Routing overhead* dapat dihitung dengan persamaan berikut (Dhamayanti, Y., dkk. 2013) :

$$\text{Routing Overhead} = \frac{\Sigma \text{ paket routing}}{\Sigma \text{ paket data yang diterima}}$$

## 3.8 Spesifikasi Kebutuhan Pembuatan Sistem

Dalam melakukan simulasi dan analisis performansi *routing* protokol (DSR, DSDV dan ZRP) dibutuhkan perangkat keras dan perangkat lunak.

### **A. Kebutuhan Perangkat Keras**

Perangkat keras adalah komponen fisik peralatan yang membentuk sistem komputer, serta peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Adapun rekomendasi perangkat keras yang dibutuhkan untuk menjalankan aplikasi ini adalah :

1. Prosesor Intel Core i3-3217U 1.8 Ghz
2. Random Access Memory 2 GB
3. Monitor VGA atau SVGA 14 inch
4. Harddisk 500 GB
5. Keyboard
6. Mouse

### **B. Kebutuhan Perangkat Lunak**

Sedangkan untuk *spesifikasi software* (kebutuhan perangkat lunak) untuk merancang aplikasi ini adalah:

1. Sistem Operasi Windows 7 Service Pack 1
2. Microsoft Office
3. Edraw Max 7.9
4. Cygwin version 2.738
5. Network Simulator 2 versi 2.30
6. Network Animator versi 1.15
7. Notepad++