

# KINERJA PROTOKOL DSR PADA JARINGAN MANET DENGAN METODE NODE DISJOINT AND ALTERNATIVE MULTIPATH ROUTING

Henni Endah Wahanani  
Jurusan Teknik Informatika, Fakultas Teknologi Industri, UPN “Veteran” Jatim  
Jl. Rungkut Madya, Surabaya  
E-mail : henni\_endah@yahoo.com

**Abstrak:** *Mobile Ad Hoc Network* (MANET) menyediakan pengguna dengan jaringan di saat mereka terus menerus bergerak. Setiap pergerakan *mobile node* mempengaruhi topologi jaringan dan rute transmisi sehingga menyebabkan kegagalan jalur rute. Jika jarak antara *mobile node* sangat besar, maka terjadi transmisi yang rendah atau terjadi kegagalan jalur rute. Kegagalan jalur rute yang tak terelakkan menyebabkan tidak hanya hubungan komunikasi antar *node* yang terputus tetapi juga kualitas dalam pengiriman data. Hal ini menjadi tantangan utama pengaturan rute dalam MANET. Penelitian ini mengajukan strategi *routing* untuk mendapatkan pengiriman data secara optimal. Untuk mendapatkan kualitas pengiriman data yang optimal, digunakan metode *Node Disjoint and Alternative Multipath Routing* (NDAMR) yang diterapkan pada kerangka protokol *Dynamic Source Routing* (DSR). Implementasi protokol *routing* yang diusulkan dilakukan dengan menggunakan tool *Network Simulator 2 (NS-2)*. Hasil penelitian menunjukkan bahwa kinerja penyelamatan data pada protokol DSR menggunakan metode NDAMR dengan berbagai variasi ukuran jaringan sampai 150 node kurang optimal dalam pengiriman paket yaitu dibawah 50%, sedangkan untuk *delay* memberikan hasil yang optimal yaitu diatas 50% dan *normalized routing load* juga memberikan hasil yang optimal yaitu diatas 50%.

**Kata kunci :** MANET, *Node Disjoint Path*, NDAMR, protokol DSR

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan terbaru dari *wireless mobile device* adalah peningkatan kebutuhan akan lingkungan *ad hoc networking* yang menghubungkan banyak *node-node mobile* seperti *notebook*, dan lain-lain. Biasanya *wireless mobile device*, seperti *notebook*, dapat diakses mengakses jaringan melalui jaringan infrastruktur *access point*. Namun, kadang muncul kebutuhan komunikasi antar *wireless mobile device* dalam situasi dimana tidak tersedia infrastruktur jaringan. Sebagai contoh komunikasi antar personal yang menggunakan *wireless mobile device* di medan peperangan. Di medan perang, tidak tersedia infrastruktur jaringan. *Wireless mobile device* seorang personel dapat berkomunikasi dengan *wireless mobile device* personel lain jika keduanya saling berada dalam jangkauan sinyal. Jika keduanya tidak saling berada dalam jangkauan sinyal, maka pengiriman paket-paket data dilakukan melalui *wireless mobile device* lain sebagai perantara. Pada kondisi demikian, sekumpulan *wireless mobile device* tersebut akan membentuk sebuah *temporary network* tanpa adanya infrastruktur jaringan yang terpasang.

Salah satu cara untuk mengurangi ketergantungan tersebut adalah dengan memanfaatkan MANET yang merupakan kumpulan *node* bergerak secara dinamis membentuk suatu jaringan sementara tanpa menggunakan struktur jaringan yang telah ada [6]. Setiap pergerakan *mobile node* mempengaruhi topologi jaringan dan rute transmisi sehingga menyebabkan kegagalan jalur rute. *Mobile node* berkomunikasi menggunakan gelombang radio. Dalam jaringan *wireless* tanpa infrastruktur akan memiliki area komunikasi gelombang radio yang terbatas. Jika jarak antara *mobile node* sangat besar, maka terjadi transmisi yang rendah atau terjadi kegagalan jalur rute [13]. Kegagalan jalur rute yang tak terelakkan menyebabkan tidak hanya hubungan komunikasi antar *node* yang terputus tetapi juga kualitas dalam pengiriman data.

Protokol *routing* adalah standarisasi yang melakukan kontrol terhadap arah pergerakan *node* dalam meneruskan paket diantara perangkat komputasi dalam MANET, dimana protokol *routing* berfungsi untuk mencarikan *route link* (jalur rute) yang terbaik dari jalur yang akan di lalui melalui mekanisme pembentukan tabel *routing*. Pemilihan router terbaik tersebut didasarkan atas beberapa pertimbangan seperti *bandwith link* dan

jaraknya. Dalam MANET terdapat dua model protokol *routing* dimana salah satunya disebut protokol *routing* yang bersifat reaktif. DSR merupakan salah satu protokol *routing* yang bersifat reaktif, efisien dan sederhana dirancang khusus untuk digunakan dalam multi-hop jaringan nirkabel *ad hoc mobile*. Jaringan sudah mengatur konfigurasi diri sendiri, tidak membutuhkan jaringan infrastruktur atau administrasi. Antara *node* tidak secara langsung dalam transmisi nirkabel berbagai satu sama lain. Semua *routing* ditentukan secara otomatis dan dipelihara oleh protokol DSR. Karena jumlah atau urutan antara *hop* yang diperlukan untuk mencapai tujuan dapat berubah setiap saat, topologi jaringan yang dihasilkan cukup banyak dan cepat berubah. Protokol DSR memungkinkan *node* secara dinamis menemukan sumber rute di beberapa jaringan hop dalam jaringan *ad hoc* [2].

Protokol *routing* reaktif mendeteksi jalur rute yang gagal difasilitasi oleh paket data secara per hop atau menggunakan *backup route* (jalur cadangan). Algoritma *backup routing* mencoba untuk mengirim data melalui jalur penemuan kembali *backup route* lokal tanpa *re-discovery* (pencarian rute kembali) secara global bila terjadi kegagalan pada jalur *routing*. Jalur yang gagal sering terjadi ketika *node* bergerak cepat. Sedangkan algoritma ini hanya mengatur rute cadangan di *route discovery* (pencarian rute). Ketika topologi jaringan berubah secara dinamis, awal pencarian rute pada *backup routing* tidak tersedia. Karena tidak adanya rute cadangan, algoritma yang ada menghasilkan sejumlah *routing* paket menggunakan *backup route* [3]. Selain itu, memiliki *overhead* yang tinggi untuk mengkonfirmasi apakah *backup route* secara berkala gagal atau tidak. *Backup routing* diklasifikasikan sebagai protokol *routing* yang memiliki jalur tunggal. Sebuah protokol *routing multipath* biasanya menetapkan beberapa jalur yang berlaku untuk sepasang *node* selama protokol dijalankan. Tetapi pada *backup routing* terdapat masalah dalam tabrakan transmisi yang disebabkan oleh tambahan lalu lintas yang dikeluarkan oleh strategi cadangan [9]. Pada penelitian S.-J. Lee, M. Gerla [12]: AODV-BR tidak efektif dan efisien dalam jaringan karena adanya tabrakan paket, sehingga karakteristik kinerja protokol *routing* cadangan terus dianalisa dan di kaji berdasarkan konsep sehingga diusulkan *node disjoint path* untuk mengatasi kekurangan protokol yang ada. *Node disjoint path* merupakan protokol *routing* yang mempertahankan hanya jalur cadangan dua terpendek di *node* sumber dan *node* tujuan seperti yang dijelaskan pada penelitian [9], dimana pada

penelitian ini menerapkan *node disjoint path* dengan metode NDAMR pada protokol AODV menghasilkan peningkatan pengiriman data secara optimal.

Oleh karena adanya penelitian penerapan metode NDAMR pada protokol AODV berhasil dalam pengiriman data maka penelitian ini mengajukan metode *Node Disjoint and Alternative Multipath Routing* (NDAMR) yang akan diterapkan pada kerangka protokol *Dynamic Source Routing* (DSR) yang di implementasikan menggunakan tool *Network Simulator 2* (NS-2).

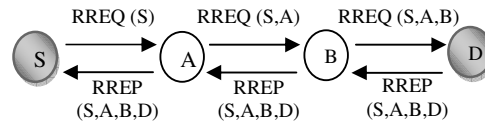
Hasil penelitian diharapkan mampu meningkatkan kinerja pengiriman data secara optimal pada protokol DSR di jaringan *ad hoc*.

## 2. KAJIAN PUSTAKA

### 2.1 Protocol Dynamic Source Routing (DSR)

Protokol DSR adalah protokol *routing* sederhana dan efisien yang dirancang khusus digunakan di *node multihop* jaringan nirkabel *mobile ad hoc* [11]. Hal ini berdasarkan pada konsep sumber *routing* yang diutarakan oleh Johnson, David B.[5], dimana sumber *routing* adalah teknik *routing* di mana pengirim paket menentukan urutan lengkap dari *node* yang akan digunakan untuk meneruskan paket, pengirim secara eksplisit mendaftarkan jalur ini dalam *header* paket, mengidentifikasi setiap forwarding "hop" dengan alamat dari *node* berikutnya untuk mengirimkan paket dalam perjalanannya ke *node* tujuan.

Protokol DSR ini terdiri dari dua mekanisme utama, yaitu *Route Discovery* (pencarian rute) dan *Route Maintenance* (pemeliharaan rute).



Gambar 2.2 Mekanisme Protokol DSR

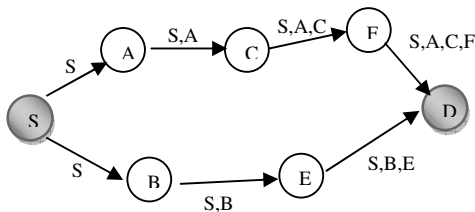
#### 2.2.1 Mekanisme Route Discovery

*Route discovery* adalah suatu mekanisme pada protokol yang berfungsi untuk melakukan pencarian *path* (jalur) secara dinamis dalam jaringan *ad hoc*, baik secara langsung di dalam *range* transmisi ataupun dengan melewati beberapa *node intermediate*. Penentuan *path* ini terbagi menjadi dua bagian yaitu *Route Request* (RREQ) dan *Route Reply* (RREP).

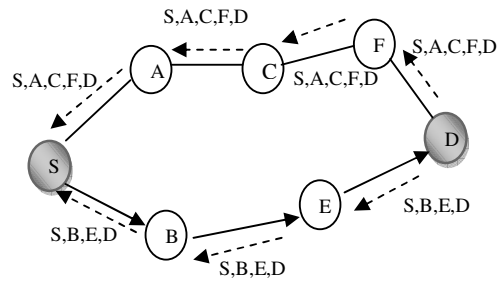
Ketika sebuah *node* sumber (S) ingin mengirim beberapa data ke *node* tujuan (D) dimulai

dari pencarian rute. *Route discovery* melakukan *broadcast*, yaitu paket RREQ ke jaringan. *Node intermediate* ketika menerima Paket RREQ melihat ke dalam *cache*-nya untuk mengetahui apakah telah ada beberapa rute ke tujuan, jika ada maka balasan ke pengirim dengan mengirim paket RREP (balasan rute), dimana mengandung rute. Jika *node intermediate* tidak memiliki rute, maka melakukan *broadcast* permintaan ulang setelah menambahkan alamat ke dalam rute sumber (lihat Gambar 2.3). Ketika *query* mencapai *node* tujuan, yang menemukan *node* yang dipesan pada urutan hop dalam paket RREQ dan menggunakan itu untuk memberikan paket RREP ke pengirim (Ashish K Shukla, 2006).

Pada Gambar 2.3 paket RREQ yang dipancarkan oleh *node* S diterima oleh *node* B. Pada *node* B, paket RREQ yang diterima diperiksa apakah sebelumnya pernah singgah di B. Karena ternyata paket RREQ tersebut belum pernah singgah di B, maka paket RREQ *diforward* dengan cara *flooding* oleh *node* B. Sebelum *diforward*, pada paket RREQ disisipi *identifier* B sehingga pada paket RREQ yang dipancarkan oleh *node* B tersebut terdapat catatan [S,B] yang menunjukkan paket RREQ telah menempuh rute S→B. Hal yang sama dilakukan ketika paket RREQ tersebut singgah ke *node* E. Dengan demikian, ketika paket RREQ tersebut dipancarkan oleh *node* E, pada paket RREQ terdapat catatan yang berisi [S,B,E] yang menunjukkan paket RREQ tersebut telah menempuh rute S→B→E. begitu pula jika dari [S,A,C,F] menunjukkan paket RREQ tersebut telah menempuh rute S→A→C→F. Ketika paket RREQ sampai pada *node* yang dituju yaitu *node destination* D, maka paket tersebut tidak di *forward*. *Node* D membalas paket RREQ tersebut dengan paket RREP yang ditujukan kepada *node* S. Rute yang ditempuh oleh paket RREP merupakan kebalikan rute yang ditempuh paket RREQ. Ingat bahwa rute yang ditempuh paket RREQ dapat diketahui dengan memeriksa catatan *node* yang tersimpan pada paket RREQ tersebut. Pada RREP yang dikirim oleh D juga disisipkan rute dari *node* S menuju *node* D.



Gambar 2.3 Route Request



Gambar 2.4 Route Reply

Pada Gambar 2.4 ditunjukkan bahwa *node* D menerima paket RREQ[S,A,C,F,D] dan RREQ[S,B,E,D]. Kemudian *node* D membalas paket RREQ tersebut dengan RREP[S,A,C,F,D] dan RREP[S,B,E,D] yang dikirim kepada *node* S. Rute yang ditempuh oleh RREP[S,A,C,F,D] adalah D→F→C→A→S dan rute yang ditempuh oleh RREP[S,B,E,D] adalah D→E→B→S. *Node* S yang menerima RREP[S,A,C,F,D] dan RREP[S,B,E,D] akan mengetahui rute yang harus ditempuh untuk mengirim paket data ke *node* D. *Node* S dapat memilih salah satu rute S→A→C→F→D atau S→B→E→D untuk mengirim paket data. Ketika *node* S mengirim paket-paket data ke *node* D, *node* S menyisipkan informasi rute yang harus ditempuh oleh paket-paket data tersebut untuk menuju *node* tujuan D.

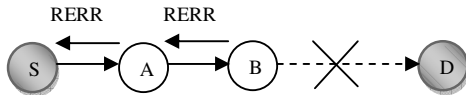
## 2.2.2 Mekanisme Route Maintenance

Protokol DSR memiliki mekanisme *route maintenance* dimana sumber mendeteksi adanya perubahan topologi jaringan sehingga pengiriman paket mengalami *kongesti* atau mekanisme yang dipanggil ketika sebuah jalur komunikasi rusak atau gagal terdeteksi selama transmisi data. Hal ini disebabkan karena salah satu *node* yang terdaftar dalam rute sebelumnya bergerak menjauh dari range *node* yang lain. Versi saat ini dari protokol DSR biasa di kenal dengan nama data salvation (penyelamatan data). Saat *node* mendeteksi masalah pada rute yang ada, paket RERR akan dikirim pada *node* pengirim. Saat paket RERR diterima, *hop* ke *node* yang menjauh akan dihilangkan dari *route cache*. Kemudian rute lain yang masih tersimpan di *cache* akan digunakan. Jika tidak ada rute lagi maka protokol DSR akan melakukan proses *route discovery* lagi untuk menemukan rute baru.

Pada Gambar 2.5, jika *node* B tidak menerima pengakuan dari *node* tujuan D setelah beberapa permintaan karena terjadi kegagalan jalur, ia mengembalikan paket RERR ke *node* sumber S. Begitu *node* sumber S menerima pesan RERR, maka akan menghapus jalur rute yang rusak dari

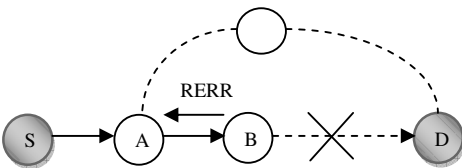
cache. Node sumber S kemudian mengulangi proses *route discovery*.

Dalam protokol DSR, setiap kali jalur *node* terdeteksi mengalami kegagalan, rute yang ada terputus, maka *node* sumber harus memulai proses *route discovery*. Dalam topologi yang sangat dinamis kegagalan *node* sering terjadi, yang menghasilkan proses *route discovery* yang menghabiskan waktu jika *node* B jauh dari *node* sumber S yang tidak memiliki rute alternatif dimana mengakibatkan *delay*.



Gambar 2.5 Ilustrasi *Route Discovery*

Melihat bahwa ketika *node* bergerak lebih dekat ke *node* tujuan D maka probabilitas mendapatkan rute alternatif yang lebih dan mengurangi waktu untuk memulai *route discovery* dan mendapatkan balasan jika *route discovery* dimulai dari *node* B itu sendiri. Antara B dan S mungkin ada *node intermediate* A yang memiliki rute ke *node* tujuan D. Jadi, bukannya kesalahan pengiriman rute kembali ke S, B dapat mengirimkan paket kembali ke A dengan paket RERR dalam paket data (diasumsikan bahwa B tidak memiliki jalur alternatif untuk *node* tujuan D dalam *cache*). Dalam cara ini *node* A kemudian akan meneruskan paket saat ini dan selanjutnya ke *node* tujuan D dengan rute alternatif yang telah di *cache*. Skema ini mengurangi *delay* yang akan terjadi.



Gambar 2.6 Penyelamatan Data

Keterbatasan protokol DSR adalah membangun dan bergantung pada rute yang *unipath* untuk setiap data transmisi. Setiap kali ada jalur yang gagal pada rute, protokol *routing* DSR perlu melakukan proses pencarian rute baru. Hal ini menyebabkan beban *routing* yang tinggi. DSR merupakan protokol *routing node* yang didasarkan pada konsep *routing*.

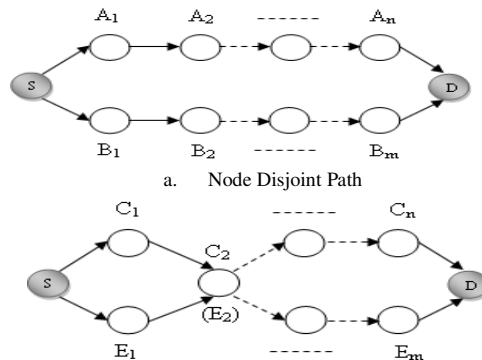
Keuntungan penggunaan protokol DSR ini adalah pada *node intermediate* tidak perlu memelihara secara *up to date* informasi *routing* pada saat melewati paket, karena setiap paket selalu berisi informasi *routing* di dalam headernya. *Routing* jenis ini juga menghilangkan juga proses

*periodic route advertisement* dan *neighbor detection* yang dijalankan oleh *routing* pada jaringan *ad hoc* lainnya. Dibandingkan dengan *routing on demand* lainnya protokol DSR memiliki kinerja yang paling baik dalam hal *throughput*, *routing overhead* (pada paket) dan rata-rata panjang *path*, akan tetapi DSR memiliki *delay* yang buruk bagi proses untuk pencarian *route* baru. Penggunaan *routing* ini akan sangat optimal pada jumlah *node* yang kecil atau kurang dari 200 *node*. Untuk jumlah yang lebih besar akan mengakibatkan *collision* antar paket dan menyebabkan bertambahnya *delay* pada saat akan membangun koneksi baru.

### 2.3 Disjoint Path Multipath Routing

Istilah *disjoint path* digunakan untuk menentukan berapa banyak *node* berpartisipasi lebih dari dua kali dalam pencarian beberapa *path* dari sumber ke tujuan. Beberapa studi diusulkan jenis *disjoint path*, seperti pada penelitian yang dilakukan oleh [7], [10] dan [8].

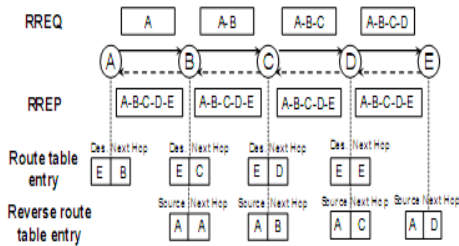
*Node disjoint* dalam jaringan yang *multipath* menghasilkan peningkatan dalam *bandwidth* yang efektif, dan pengurangan kemacetan dan probabilitas menjatuhkan paket dalam jaringan. Ada dua jenis *disjoint path*: *link disjoint* dan *node disjoint*. *Link disjoint path* tidak memiliki *link* bersama, tetapi mungkin memiliki *node* bersama. Sebaliknya, *node disjoint path* tidak memiliki *node* bersama, kecuali sumber dan tujuan. Hal ini serupa dengan konsep dari penelitian [4] bahwa di *node disjoint*, setiap *node* berpartisipasi dalam hanya satu jalur sedangkan di *link disjoint*, *node* dapat berpartisipasi dalam beberapa jalur, tapi bukan dua *node* atau titik lagi berurutan. *Node disjoint path* dan *link disjoint path* digambarkan sebagai berikut [7] :



b. Link Disjoint Path

Gambar 2.7 *Disjoint Path*

Tujuan utama dari *node disjoint multipath routing* adalah untuk membangun jalur *node disjoint* pada jaringan *multipath* dengan menghasilkan *routing overhead* yang rendah selama *route discovery* (pencarian rute). Pada waktu modifikasi protokol DSR dengan menyertakan jalur secara akumulasi dalam paket RREQ. Ketika paket RREQ dihasilkan atau diteruskan oleh *node* dalam jaringan, setiap *node* menambahkan alamat sendiri untuk permintaan paket routing. Ketika sebuah paket RREQ tiba di tempat *node* tujuan, *node* tujuan bertanggung jawab untuk menilai apakah ada atau tidak jalur routing pada *node disjoint path*. Sebagai contoh mempertimbangkan lima *node* A, B, C, D dan E seperti yang ditunjukkan pada Gambar 2.8 [14]. *Node* A ingin mengirim data ke *node* E. Karena A tidak memiliki rute untuk E dalam tabel routing, maka melakukan broadcast permintaan rute. *Node* B menerima permintaan rute, menambahkan nya alamat sendiri untuk meminta, dan meneruskan permintaan padahal juga tidak memiliki rute ke E. Demikian pula, ketika *node* C dan *node* D menerima RREQ, mereka tambahkan alamat mereka ke permintaan dan meneruskannya.



Gambar 2.8 Akumulasi Path didalam Node Disjoint Multipath Routing

*Node disjoint multipath* lebih kuat daripada yang lain, hal ini terbukti dalam penelitian [7] menggunakan probabilitas *path break*. Oleh karena itu, penemuan *node disjoint multipath* merupakan bagian penting dari *multipath routing*. Akibatnya, pencarian rute semua jalur tersedia dari sumber ke tujuan bisa lebih efisien untuk *node* dengan jumlah yang banyak.

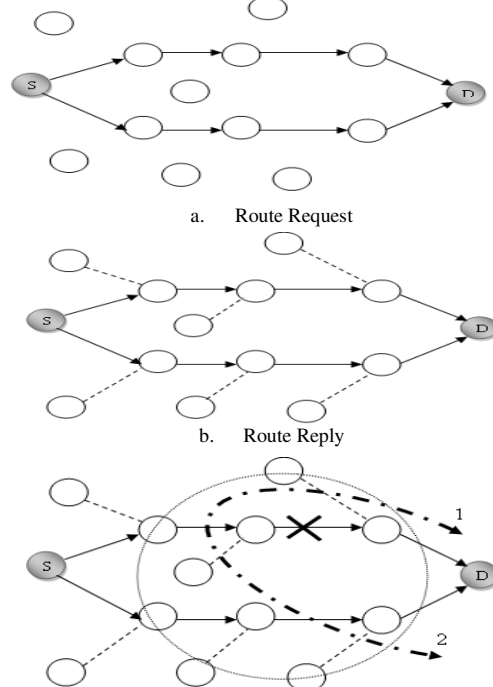
## 2.4 Metode Node Disjoint and Alternative Multipath Routing

Pencarian rute dilakukan melalui *route request* (rute permintaan) dan *route reply* (jawaban rute). *Data Salvation* dimulai ketika pertemuan mobile *node* mengalami kegagalan jalur. Mekanisme permintaan rute diadopsi untuk membuat beberapa *node*, yaitu, beberapa jalur yang

tidak memiliki *node intermediate* (*node* tengah). Namun, hanya satu tambahan *node* jalan utama dibuat di samping jalur primer pertama untuk sesi pasangan. Strategi ini dipilih sebagai kebebasan rute milik *node disjoint path* memungkinkan transmisi data pada jalur yang berbeda dari sesi pasangan tanpa mempengaruhi satu sama lain. Selain itu, mekanisme pencarian rute telah terbukti memiliki biaya routing kecil dan keterlambatan transmisi. Selanjutnya, multiple jalur pada *node-disjoint* dari sepasang sesi ditetapkan oleh *node* tujuan pada saat yang sama menjalankan routing protokol dan karena itu keduanya dijamin cocok untuk transmisi data.

Hal ini dibangun suatu pendekatan yaitu sebuah metode *Node Disjoint and Alternative Multipath Routing* (NDAMR). NDAMR menetapkan dua *node disjoint* jalur utama untuk *node* yang berpasangan. *Node disjoint path* mempunyai rute independen yang berarti bahwa tidak memiliki jalur bersama antara *node* dan tidak mempengaruhi satu sama lain selama transmisi data.

Metode NDAMR memiliki dua aspek dibandingkan dengan jaringan *multipath* pada protokol *routing on-demand* lainnya yaitu mengurangi *routing overhead* secara dramatis dan mencapai berbagai jalur *routing node disjoint*.



Gambar 2.9 Route Discovery dan Route Maintenance pada Metode NDAMR

### 3. METODOLOGI PENELITIAN

#### 3.1 Pseudocode NDAMR

Pertama kali yang dilakukan pada algoritma NDAMR yaitu permintaan rute. Tahap permintaan rute dilakukan pengumpulan informasi rute. Sebuah *node* sumber yang berasal dari tanpa jalur yang sah kepada *node* tujuan menambahkan paket RREQ ke alamat sendiri dalam catatan rute sebelum melakukan *broadcast*. Jika *path*-nya terlalu jauh maka paket RREQ tersebut di jatuhkan atau di abaikan. kemudian ternyata ada paket RREQ memiliki *path* terpendek ke sumber, maka di hitung *hop* paket RREQ tersebut lalu di simpan di masukkan dalam daftar akumulasi paket RREQ ke alamat di *node* tujuan. *Node* tujuan menerima beberapa identik paket RREQ dengan akumulasi jalur yang berbeda. Paket RREQ yang satu dengan yang lain dibandingkan disimpan dalam tabel rute. Hal ini dilakukan secara berulang-ulang dengan melakukan *broadcast* kembali paket RREQ. Hanya satu *node* yang terpendek dan beriringan dalam satu cakupan sinyal disimpan. Selanjutnya, paket RREP dikembalikan ke *node* sumber.

```

If (paket RREQ diduplikasikan) and
  (path jauh) then
  drop paket RREQ;
else if (path terpendek ke sumber) then
  hop terpendek := hitung hop paket RREQ;
  hop selanjutnya dari reverse path :=
  hop sebelumnya yang mengirim paket RREQ;
  Simpan daftar akumulasi paket RREQ ke (Alamat);
  Hitung hop paket RREQ += 1;
  Broadcast kembali paket RREQ;
else drop paket RREQ;
endif
  
```

Gambar 3.1 Permintaan Rute

Pada gambar 3.2 dilakukan proses penyelamatan data dimana terjadi pada saat kegagalan jalur. Ketika jalur utama mengalami kegagalan *link* atau hubungan maka dilakukan *broadcast* data yang ditransmisikan kepada para *node* tetangga. *Node* tetangga memiliki setidaknya dua pilihan, yaitu menggunakan jalur primer kedua atau jalur alternatif.

```

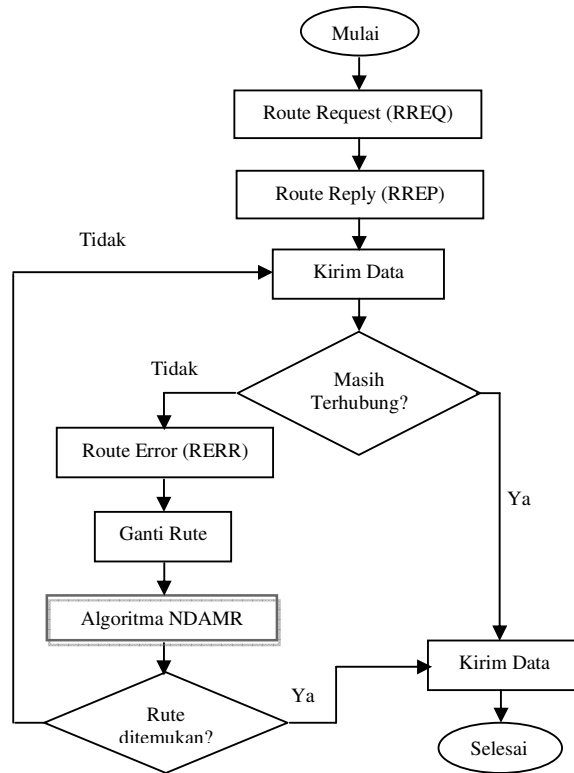
if (path alternatif aktif ke node tujuan) then
  if (kondisi 1) and (kondisi 2) then
    paket data di broadcast pada hop berikutnya = hop berikutnya di sepanjang path
    alternatif;
    forward paket data;
    break;
  endif
else if (path primer aktif ke node tujuan) then
  if (kondisi 1) then
    paket data di broadcast pada hop berikutnya = hop berikutnya sepanjang path primer;
    forward paket data;
    break;
  endif
else drop paket data;
endif
  
```

kondisi :  
 1. *Path* aktif pada hop berikutnya adalah bukan hop yang membutuhkan bantuan  
 2. *Path* alternatif pada hop berikutnya adalah hop pada saat terputus atau *disconnected*

Gambar 3.2 Penyelamatan Data

#### 3.2 Modifikasi Protokol DSR

Tahap desain model algoritma ini digunakan untuk merancang model algoritma yang memadukan antara protokol DSR dengan algoritma NDAMR. Modifikasi yang dilakukan adalah melakukan perubahan pada proses *route discovery* dan *route maintenance* pada protokol *routing* DSR.



Gambar 3.3 Modifikasi Protokol DSR dengan Algoritma NDAMR

## 4. Uji Coba dan Analisis Hasil

### 4.1 Skenario Uji Coba

Pada bagian ini akan dijelaskan skenario uji coba dan hasil uji coba dari protokol routing yang telah dirancang dan akan diimplementasikan menggunakan NS-2 versi 2.30. Hasil uji coba tersebut nantinya akan dianalisa apakah sesuai dengan tujuan yang diharapkan atau tidak. Karena keterbatasan sumber daya yang dipergunakan dalam proses uji coba, maka pada uji coba ini jumlah *node* hanya dibatasi sampai 150 *node*. Parameter-parameter yang digunakan pada uji coba ini mengacu pada penelitian yang dilakukan oleh [9], adalah sebagai berikut :

- a. Waktu simulasi : 180 detik
- b. Jumlah node : 50 node, 100 node, 150 node
- c. Packet rate : 1 packet/s
- d. Packet size : 512 byte
- e. Maksimum speed : 5 m/s, 10 m/s, 15 m/s, 20 m/s, 25 m/s, 30 m/s
- f. Maksimum koneksi : 5 node, 10 node, 15 node
- g. Network area : 500x500m<sup>2</sup>, 1000x1000m<sup>2</sup>, 1000x1500 m<sup>2</sup>

### 4.2 Hasil Uji Coba

Pada bagian ini akan dijelaskan hasil uji coba dari algoritma yang telah dirancang pada bab sebelumnya. Hasil uji coba ini akan memberikan gambaran terhadap performa penyelamatan data dari protokol *routing* yang diusulkan pada jaringan MANET. Hasil dari uji coba yang dilakukan dengan melihat nilai parameter *packet delivery ratio*, *average delay* dan *normalized routing load*. Hasil simulasi jaringan yang menunjukkan perbandingan kinerja penyelamatan data dari protokol DSR dan protokol DSR dengan menggunakan NDAMR seperti PDR, *average delay*, dan NRL ditunjukkan pada Tabel 4.1 sampai Tabel 4.3. Fokus analisis hasil uji coba pada penelitian ini adalah analisis terhadap hasil pengukuran *Packet Delivery Ratio*, *Average delay* dan *Normalized Routing Load* ketika melakukan penyelamatan data pada saat pengiriman data di *node* sebesar 50 *node*, 100 *node* dan 150 *node* dengan masing-masing maksimum koneksi 5 *node*, 10 *node* dan 15 *node* di *network area* 500 x 500 m<sup>2</sup>, 1000 x 1000 m<sup>2</sup>, 1000 x 1500 m<sup>2</sup>.

### 3.2.1 Analisis Packet Delivery Ratio

*Packet Delivery Ratio* akan menurun pada saat kecepatan gerak *node* bertambah besar. Hal ini disebabkan bertambahnya kecepatan gerak *node* maka akan sering terjadi perubahan posisi *node* sehingga rute rusak, menyebabkan hilangnya paket data yang lewat (*drop*).

$$PDR = \frac{\text{paket data yang diterima}}{\text{paket data yang dikirim}}$$

**Tabel 4.1** Kinerja Penyelamatan Data untuk *Packet Delivery Ratio*

Maksimum koneksi ( <i>node</i> )	<i>Network area</i> (m <sup>2</sup> )	<i>Packet Delivery Ratio</i>		
		50 <i>node</i>	100 <i>node</i>	150 <i>node</i>
5	500x500	1.42%	1.27%	11.38%
	1000x1000	3.43%	3.00%	0.82%
	1000x1500	0.60%	1.03%	1.58%
10	500x500	1.90%	3.97%	0.34%
	1000x1000	3.46%	8.48%	1.89%
	1000x1500	7.57%	1.82%	1.17%
15	500x500	0.44%	0.65%	0.71%
	1000x1000	1.15%	2.63%	1.57%
	1000x1500	2.04%	10.10%	3.65%
Nilai rata-rata		2.45%	3.66%	2.57%

Pada tabel 4.1 diatas, untuk 50 *node* dengan maksimum koneksi 10 *node* di *network area* 1000x1500 m<sup>2</sup> menghasilkan penyelamatan data yang lebih optimal yaitu 7,57%. Sedangkan pada 100 *node*, penyelamatan data yang optimal ada di maksimum koneksi 15 *node* di *network area* 1000x1500 m<sup>2</sup>. 150 *node* dengan maksimum koneksi 5 *node* di *network area* 500x500 m<sup>2</sup>. Jadi dapat disimpulkan, bahwa jumlah paket yang terkirim pada protokol DSR dengan NDAMR terhadap protokol DSR yaitu dibawah 50%.

### 3.2.2 Analisis Average end to end Delay

*Delay* merupakan waktu pengiriman paket data dari sumber ke penerima *Delay* akan meningkat ketika kecepatan gerak *node* bertambah besar. Hal ini disebabkan pada saat kecepatan tinggi, maka kondisi jaringan akan semakin tidak stabil. Hal ini akan mengakibatkan semakin banyak paket RREQ yang dikirimkan, sehingga akan mengakibatkan peluang tabrakan antar paket semakin besar. Pada saat *node* bergerak dengan kecepatan tinggi, *route* semakin tidak tersedia

sehingga paket data akan tertahan di *buffer* sementara sampai *route* baru di temukan. Hal ini akan mengakibatkan rata-rata *delay* pengiriman paket data akan semakin besar.

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

**Tabel 4.2** Kinerja Penyelamatan Data untuk *Average Delay*

Maksimum koneksi (node)	Network area (m <sup>2</sup> )	Average Delay		
		50 node	100 node	150 node
5	500x500	53.25%	22.67%	13.26%
	1000x1000	47.83%	29.88%	65.58%
	1000x1500	41.71%	51.67%	51.57%
10	500x500	63.93%	55.95%	42.78%
	1000x1000	59.41%	22.35%	30.62%
	1000x1500	48.23%	48.19%	32.95%
15	500x500	25.97%	33.39%	34.15%
	1000x1000	47.45%	30.92%	26.23%
	1000x1500	19.07%	28.95%	70.67%
Nilai rata-rata		45.21%	36.00%	40.87%

Kinerja penyelamatan data untuk *delay* di 50 *node* yang terlihat pada tabel 4.2 menunjukkan bahwa *delay* yang optimal di maksimum koneksi 10 *node* dengan *network area* 500x500 m<sup>2</sup> yaitu 63,93%. Sedangkan pada 100 *node* di maksimum koneksi 10 *node* dengan *network area* 500x500 m<sup>2</sup> yaitu 55,95% , dan 150 *node* di maksimum koneksi 15 *node* dengan *network area* 1000x1500 m<sup>2</sup> yaitu 70,67%. Hal ini menunjukkan bahwa penyelamatan data untuk *delay* protokol DSR dengan NDAMR terhadap protokol DSR lebih optimal diatas 50%.

### 3.2.3 Analisis Normalized Routing Load

*Normalized routing load* merupakan rasio antara jumlah paket *routing* dengan paket data yang berhasil diterima. *Normalized routingload* meningkat ketika kecepatan gerak *node* bertambah besar. Hal ini disebabkan karena ketika kecepatan gerak *node* bertambah, posisi *node* juga semakin mudah berubah dan konfigurasi jaringan sering berubah-ubah, *node* pengirim akan lebih banyak melakukan *broadcast* paket *routing* pada proses *Route Discovery* (RREQ dan RREP) untuk mendapatkan *route* yang baru. Kedua, adanya mekanisme *flooding* pada protokol DSR. Dengan mengirimkan paket RREQ dan RREP pada saat *route discovery* dan ketika jumlah *node* bertambah

maka jumlah paket RREQ dan RREP yang dikirim juga semakin bertambah besar.

$$\text{NRL} = \frac{\text{jumlah paket routing}}{\text{paket data yang diterima}}$$

**Tabel 4.3** Kinerja Penyelamatan Data untuk *Normalized Routing Load*

Maksimum koneksi (node)	Network area (m <sup>2</sup> )	Normalized Routing Load		
		50 node	100 node	150 node
5	500x500	23.01%	29.75%	53.48%
	1000x1000	57.62%	13.29%	26.35%
	1000x1500	20.85%	17.89%	24.85%
10	500x500	62.38%	57.76%	27.71%
	1000x1000	42.61%	35.78%	51.45%
	1000x1500	30.75%	34.00%	42.69%
15	500x500	41.81%	40.86%	17.99%
	1000x1000	31.59%	39.74%	29.01%
	1000x1500	10.73%	26.75%	23.90%
Nilai rata-rata		35.71%	32.87%	33.05%

Kinerja penyelamatan data untuk NRL di 50 *node* yang terlihat pada tabel 4.3 menunjukkan bahwa NRL yang optimal di maksimum koneksi 10 *node* dengan *network area* 500x500 m<sup>2</sup> yaitu 62,38%. Sedangkan pada 100 *node* di maksimum koneksi 10 *node* dengan *network area* 500x500 m<sup>2</sup> yaitu 57,76% , dan 150 *node* di maksimum koneksi 5 *node* dengan *network area* 500x500 m<sup>2</sup> yaitu 53,48%. Hal ini menunjukkan bahwa penyelamatan data untuk *delay* protokol DSR dengan NDAMR terhadap protokol DSR lebih optimal diatas 50%.

## 4. Kesimpulan

Berdasarkan hasil analisis pada simulasi pengukuran jumlah paket yang terkirim, *average delay* dan *normalized routing load*, dapat diambil beberapa kesimpulan sebagai berikut :

- Penyelamatan data pada protokol DSR dengan metode NDAMR terhadap protokol DSR memberikan kualitas pengiriman paket yang tidak optimal yaitu dibawah 50%..
- Penyelamatan data untuk *delay* pada protokol DSR dengan metode NDAMR terhadap protokol DSR memberikan hasil yang optimal yaitu diatas 50% untuk 50 *node* dengan maksimum koneksi 10 *node* dengan *network area* 500x500m<sup>2</sup> yaitu 63,93%. Sedangkan pada 100 *node* di maksimum koneksi 10 *node* dengan



*network area* 500x500 m<sup>2</sup> yaitu 55,95% , dan 150 *node* di maksimum koneksi 15 *node* dengan *network area* 1000x1500 m<sup>2</sup> yaitu 70,67%.

- c. Penyelamatan data untuk *normalized routing load* pada protokol DSR dengan metode NDAMR terhadap protokol DSR memberikan hasil yang optimal yaitu diatas 50% untuk 50 *node* dengan maksimum koneksi 10 *node* dengan *network area* 500x500 m<sup>2</sup> yaitu 62,38%. Sedangkan pada 100 *node* di maksimum koneksi 10 *node* dengan *network area* 500x500 m<sup>2</sup> yaitu 57,76%, dan 150 *node* di maksimum koneksi 5 *node* dengan *network area* 500x500m<sup>2</sup> yaitu 53,48%.

## 5. Daftar Pustaka

- [1] Bobade, Niles P. and Mhala, Nitiket N., 2012. Performance Evaluation of AODV and DSR on Demand Routing Protocols with Varying Manet Size. *International Journal of Wireless & Mobile Networks (IJWMN)* Vol. 4, No. 1, February.
- [2] David B. Johnson, David A. Maltz, dan Josh Broch., 2001. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. in *Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley.
- [3] Jung, Se-Won., Choi, Jin-Chul., dan Lee, Chae-Woo., 2008. *A Study on Backup Route Setup Scheme for Ad Hoc Networks*. School of Electrical and Computer Engineering, Ajou University.
- [4] Jung, S.H., Talipov, E., Ahn, M.W., dan Kim, C., 2007. *A Discovery Method of Node-Disjoint Multipaths by Valid Source-Destination Edges*. In the Proceedings of 4th IASTED Asian conference Communication systems and network.
- [5] Johnson, David B. dan Maltz, David A., 1996. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Computer Science Department, Camegie Mellon University.
- [6] Li, D., Liu, Q., Hu, X., dan Jia, X., 2007. *Energy Efficient Multicast Routing in Ad Hoc Wireless Network*. *Computer Communications*, vol.30, hal. 3746–3756.
- [7] Li, Xuefei dan Cuthbert, Laurie., 2004. *Stable Node-Disjoint Multipath Routing with Low Overhead in Mobile Ad hoc Networks*. Proceedings of the IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems.
- [8] Li, Zhi dan Kwok, Yu-Kwong., 2005. *A New Multipath Routing Approach to Enhancing TCP Security in Ad Hoc Wireless Networks*. Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'05).
- [9] Lin, Chu-Hsing., Jiang, Fuu-Cheng dan Chang, Jen-chieh., 2008. *Node-disjoint Alternative Dual-path Routing for Data Salvation in Mobile Ad hoc Networks*. Proceedings of the IEEE international Conference on Parallel and Distributed Computing, Applications and Technologies.
- [10] Nasipuri and S. R. Das., 1999. *On-Demand Multipath Routing for Mobile Ad Hoc Networks*. Proc. ICCN 1999, pp. 64–70, Oct.
- [11] Ramesh, V dkk., 2010. *Modified DSR (Preemptive) to Reduce Link Breakage and Routing Overhead for MANET using Proactive Route Maintenance (PRM)*. *Global Journal of Computer Science and Technology*.
- [12] S-J. Lee., M.Gerla., 2000. *AODV-BR: Backup Routing in Ad Hoc Networks*. Proceedings of IEEE Wireless Communications and Networking Conference, Vol. 3, Chicago.
- [13] Wang, Ying-Hong., dan Chao, Chih-Feng. 2006. *Dynamic Backup Routes Routing Protocol for Mobile Ad Hoc Networks*. Department of Computer Science and Information Engineering, Tamkang University, Taiwan.
- [14] X. Li, L. Cuthbert., 2004. *On-demand Node-disjoint Multipath Routing in Wireless Ad Hoc Networks*, Proceedings of the 29<sup>th</sup> Annual IEEE International Conference on Local Computer Network (LCN 2004), Tampa, FL, 16-18 Nov. pp. 419-420.