

Analisis Perbandingan Metode *Routing Spray and Wait* dengan *Prophet* untuk Daerah Terpencil

Imron Sazali¹, Achmad Basuki²

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹sazaliimron@gmail.com, ²abazh@ub.ac.id

Abstrak

Kondisi geografis Indonesia beberapa diantaranya adalah pedalaman yang didalamnya tidak ada koneksi internet secara *reliable*. *Routing* merupakan sebuah proses untuk menentukan rute dari sumber ke sebuah tujuan dalam sebuah skema komunikasi. Dibutuhkan sebuah skema *Routing* yang mempunyai toleransi terhadap *delay*, mobilitas, dan membutuhkan sumberdaya yang minimal. Pemilihan model *Routing* sangat bergantung pada kondisi geografis daerah tersebut. Teknologi *Delay Tolerant Network* (DTN) merupakan sebuah mekanisme pengiriman data yang tahan dengan *latency*, dan bisa membawa paket data dalam ukuran besar dengan *resource system* yang minimal. Dengan adanya teknologi DTN (*Delay Tolerant Network*) masalah tersebut mungkin untuk diselesaikan. Dua protokol DTN dalam tugas akhir ini adalah *Prophet* yang menggunakan *knowledge* dan *Spray and Wait* dengan strategi replika. Kedua protokol tersebut disimulasikan dalam lingkungan yang telah ditentukan menggunakan *One Simulator* lalu dianalisis berdasarkan parameter pengukur kinerja yaitu *delivery probability*, *average delay* dan *overhead ratio*. Dari hasil penelitian yang dilakukan, didapatkan data bahwa *Routing Spray and Wait* memiliki kecenderungan kinerja yang lebih baik dalam hal *Delivery Probability* dan *Average Delay*. Sedangkan dalam hal *Overhead Ratio*, *Routing Prophet* menunjukkan nilai yang lebih kecil. Untuk lingkungan yang memiliki karakteristik seperti dalam penelitian ini *Spray and Wait* lebih dipilih dibandingkan dengan *Prophet*.

Kata kunci: Pedalaman, DTN, *Routing Prophet*, *Routing Spray and Wait*, *One Simulator*

Abstract

Indonesia's geographical conditions, some of which are inland, in which there is no reliable internet connection. Routing model selection is highly dependent on the geographical conditions of the area. Routing is a process for determining the route from source to destination in a communication scheme. It takes a Routing scheme that has tolerance for delay, mobility, and requires minimal resources. Delay Tolerant Network (DTN) technology is a data delivery mechanism that resists latency, and can carry large data packets with minimal system resources. With the DTN (Delay Tolerant Network) technology the problem is likely to be resolved. Two DTN protocols in this final project are Prophets that use knowledge and Spray and Wait with replica strategies. Both protocols are simulated in a predetermined environment using One Simulator and then analyzed based on performance measurement parameters of delivery probability, average delay and overhead ratio. From the results of research conducted, obtained data that Routing Spray and Wait has a tendency better performance in terms of Delivery Probability and Average Delay. Whereas in the case of Overhead Ratio, Routing Prophet shows a smaller value. For the environment that has characteristics as in this study Spray and Wait is better than Prophet.

Keywords: Rural, DTN, *Routing Prophet*, *Routing Spray and Wait*, *One Simulator*

1. PENDAHULUAN

Kondisi geografis Indonesia yang beragam terkadang mengakibatkan beberapa daerah belum bisa terjangkau infrastruktur jaringan

yang fixed. Hal ini menyebabkan daerah terpencil tidak dapat dijangkau pertukaran informasi digital, karena sulitnya membangun sebuah jaringan yang reliable. Dalam siaran pers Kementerian Komunikasi dan Informatika menyebutkan masih ada sekitar 43.000 desa

tertinggal yang belum terjangkau jaringan telekomunikasi (Kominfo, 2006). Penyebaran informasi digital ke daerah terpencil dirasa sangatlah penting karena kemajuan ekonomi, pendidikan, dan hal lain harus dirasakan pula oleh masyarakat terpencil. Diharapkan dengan adanya kesetaraan informasi antara masyarakat kota dan masyarakat terpencil, setiap masyarakat bisa menikmati hasil dari pembangunan Indonesia. Supaya terjadi kesetaraan informasi antara perkotaan dan daerah terpencil, diperlukan skema pengiriman data yang sistematis dan efisien. Proses pengiriman informasi data digital antar daerah bisa disebut dengan proses *Routing*.

Routing merupakan sebuah proses untuk menentukan rute dari sumber ke sebuah tujuan dalam sebuah skema komunikasi. Penentuan model *Routing* yang tepat menjadi inti dari keberhasilan sebuah pengembangan teknologi jaringan, karena pemilihan *Routing* yang tepat bisa meningkatkan kinerja dari jaringan itu sendiri. Untuk mendapatkan metode yang sesuai untuk diterapkan pada suatu jaringan, maka harus dilakukan perbandingan antara 2 atau lebih skema *Routing*. Pemilihan model *Routing* juga akan sangat berpengaruh pada model lingkungan tempat jaringan tersebut diimplementasikan. Dibutuhkan sebuah skema *Routing* yang mempunyai toleransi terhadap delay, mobilitas, dan membutuhkan resource yang minimal (Warthman, 2003).

Delay Tolerant Network (DTN) merupakan sebuah mekanisme pengiriman data yang memiliki toleransi tinggi terhadap latency, dan bisa membawa paket data dalam ukuran besar dengan resource system yang minimal. Didalam komunikasi jaringan DTN menawarkan sebuah solusi *Routing* dan pengiriman data dengan kondisi lingkungan yang tidak dimungkinkan adanya jaringan yang reliabel (Thrasyvoulos, 2005). Di beberapa penelitian sebelumnya DTN berhasil diterapkan, tetapi karena karakteristik geografis Indonesia yang beraneka ragam, maka perlu dilakukan perbandingan dari beberapa protokol *Routing* DTN pada suatu lingkungan.

Protokol yang digunakan dalam DTN bisa dikategorikan dalam dua kelompok berdasarkan knowledge dan strategi replikasi untuk mengirimkan pesan dari pengirim ke penerima. Salah satu protokol yang menggunakan knowledge adalah *Prophet*, sedangkan untuk strategi replika salah satunya adalah *Spray and Wait* (Fall, 2003).

Penelitian yang pernah dilakukan sebelumnya adalah “Analisis Penggunaan Protokol *Routing Prophet* pada DTN untuk Sistem Berbagi Informasi Digital di Daerah Pedalaman” oleh Magdalena (2014). Dalam pembahasan ini didapatkan informasi bahwa dalam protokol *Prophet*, masih ditemukan lamanya delay dalam forwarding paket. Dalam penelitian lain “*Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks*” oleh Thrasyvoulos S., Konstantinos P., dan Cauligi S. R., tahun 2005 diinformasikan bahwa protokol *Spray and Wait* memiliki sifat replikasi dan flooding based, sehingga dengan sifat ini *Spray and Wait* bisa diharapkan mengurangi delay yang terjadi.

2. LANDASAN KEPUSTAKAAN

2.1. Delay Tolerant Network (DTN)

Delay Tolerant Network (DTN) merupakan arsitektur jaringan end-to-end yang didesain untuk menyediakan komunikasi pada lingkungan dengan konektivitas intermitten, penundaan yang besar, dan error rate yang tinggi (Ilham A., 2012). DTN menyediakan suatu mekanisme yang dapat menghubungkan suatu jaringan regional yang memiliki karakteristik *delay* yang berbeda-beda (Fall, 2003). DTN merupakan suatu arsitektur jaringan yang menawarkan toleransi yang cukup tinggi terhadap *delay*. DTN dirancang untuk mengatasi komunikasi pada suatu jaringan dimana tidak terdapat konektivitas yang secara terus menerus.

2.2. Prophet

Prophet merupakan salah satu protokol *Routing* pada DTN dengan karakteristik penggunaan pengetahuan dari pertemuan *node-node* untuk lebih mengoptimalkan penerusan paket. Sifat lain yang unik dari *Routing* ini adalah mempunyai *delivery predictability* (DP), nilai DP ini digunakan untuk memutuskan apakah sebuah paket akan diteruskan atau tidak. Tiap *node* memiliki DP tabel dan setiap pertemuan suatu *node* dengan *node* lain, maka akan dilakukan pertukaran informasi DP tabel. Setiap terjadi pertemuan antar *node* maka DP akan diperharui. *Delivery Predictability* dapat dihitung dengan beberapa cara berdasarkan pertemuan antar *node*. *Node* yang sering bertemu memiliki DP yang tinggi. Perhitungan tersebut ditunjukkan pada persamaan 1.

$$P_{(a,b)_{new}} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) \times P_{init} \quad (1)$$

Dimana :

$P_{(a,b)new}$ = DP untuk *node* B yang disimpan dalam *node* A setelah pertemuan

$P_{(a,b)old}$ = DP sebelum pertemuan dengan *node*

P_{init} = konstanta inisialisasi

2.3. Spray and Wait

Sistem *Routing Spray and Wait* mempunyai 2 tahapan dalam dalam proses *Routing*nya, yakni tahapan *spray* dan *wait*. Pada tahapan *Spray*, *node* sumber meneruskan paket ke *node* yang lain. Jika *path*/jalur yang ditentukan sesuai, maka paket diteruskan. Jika tidak, maka proses *Wait* akan diteruskan sampai proses atau jalur ditemukan. Dan pada tahap *Spray and Wait* tidak diketahui bagaimana awal salinan pesan disebar. *Spray and Wait* mengambil keuntungan dari *Routing Epidemic* dengan transmisi message yang lebih cepat dan *Delivery Probability* yang tinggi dan proses *forwarding* ke tujuan secara langsung. Tujuan dari *Spray and Wait* adalah mengontrol *flooding* dengan membatasi atau mengurangi jumlah salinan (L) yang dibuat dengan mengurangi *overhead* di *Epidemic Routing* yang berhubungan dengan jumlah *Node* (N).

Proses *Routing* pada *Spray and Wait* mempunyai 2 tahap *Routing* yaitu tahap *Spray* dan tahap *Wait*, akan dijelaskan pada uraian berikut ini :

a) Tahap *Spray*

Adalah fase saat *node* Source membuat salinan pesan untuk disebar ke *relay node*. Tahap ini membatasi pesan yang di salin untuk meminimalkan penggunaan sumberdaya jaringan. Pada tahap ini, proses *multicast* dilakukan untuk mengirim beberapa salinan pesan dari *source* ke *relay node*. Jika *destination* tidak ditemukan maka *node* akan memasuki tahap *wait* dimana setiap *relay node* yang memiliki salinan pesan menunggu sampai *destination* ditemukan untuk meneruskan pesan.

b) Tahap *Wait*

Jika *destination* tidak ditemukan dalam tahap *spray*, maka setiap *relay node* yang membawa salinan melakukan transmisi langsung ke *destination* yaitu meneruskan pesan hanya untuk *destination*. Pada tahap *wait*, *node* diperbolehkan untuk menyampaikan pesan ke *destination* menggunakan penerusan pesan secara langsung ketika *time to live (TTL)*

berakhir. Pada tahap *wait* sebuah *node* akan meneruskan pesan ke *relay node* yang lain sampai tersisa satu pesan saja, dan *relay node* yang hanya memegang satu pesan salinan akan masuk ke tahap *wait*. Pada tahap ini *relay node* akan menunggu sampai bertemu *destination* dan melakukan penerusan pesan.

2.4. Parameter Pengukur Kinerja

Kinerja *Routing* pada DTN bisa diukur dari nilai parameter seperti *Delivery Probability*, *Average Delay*, dan *Overhead Ratio*. *Delivery Probability* adalah suatu nilai yang menentukan probabilitas pesan penerusan pesan dalam *Routing*. Nilai dari DP ini adalah berkisar $0 \leq DP \leq 1$. Semakin besar nilai DP maka kinerja *Routing* bisa disebut lebih baik. Dalam penerusan pesan di sistem DTN terdapat jeda waktu antara pesan dikirim dan pesan diterima oleh *node* tujuan. Waktu yang dibutuhkan untuk sebuah pesan sampai di *node* tujuan ini biasa disebut *delay*. Sedangkan *Average Delay* adalah nilai untuk mengukur waktu rata rata yang dibutuhkan pesan bisa sampai di *node* tujuan. *Overhead Ratio* adalah sebuah negasi dari beberapa pesan yang diinformasikan ke sejumlah pesan yang terkirim. Semakin rendah nilai overhead maka semakin optimal kinerja sebuah *Routing*. *Overhead Ratio* ini dipengaruhi oleh lama paket disimpan sebelum diteruskan ke *node* tujuan.

2.5. Aplikasi Pendukung Simulasi

Untuk memudahkan dalam pembuatan simulasi DTN ini digunakan beberapa aplikasi pendukung seperti *NetBeans* (NetBeans, 2017), *OpenJump* (OpenJump, 2017), *OpenStreetMap* (OpenStreetMap, 2017), dan *One Simulator* (OneSimulator, 2017). *NetBeans* akan digunakan untuk melakukan pengaturan lanjutan supaya simulasi bisa dimodelkan sesuai dengan lingkungan. *OpenJump* digunakan untuk melakukan digitasi peta supaya bisa digunakan dalam simulasi. *OpenStreetMap* digunakan untuk mengambil data peta dari daerah yang diinginkan untuk simulasi. *One Simulator* digunakan untuk mensimulasikan *Node* DTN supaya bisa berjalan pada peta yang sudah ditentukan.

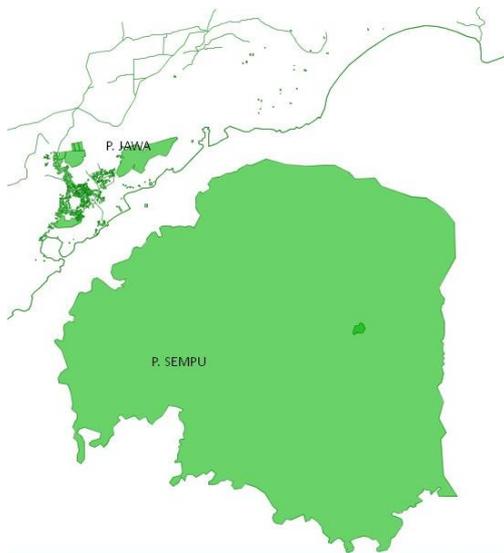
3. METODOLOGI

3.1. Implementasi

Pada tahap implementasi dilakukan

pemodelan terhadap daerah pedalaman. Agar daerah pedalaman ini memenuhi syarat, dilakukan pemodelan secara *real* terhadap daerah pedalaman tersebut dengan menggunakan *software* *OpenStreetMap* dan *OpenJump* untuk menghasilkan file *.wkt* lalu diuji menggunakan *One Simulator*.

Pembuatan skenario DTN pada *One Simulator* membutuhkan *map* yang memiliki skala yang tepat sesuai dengan kenyataan sebenarnya, dimana saat ini *map* dengan mudah didapatkan melalui website GIS seperti *OpenStreetMap.org* seperti ditunjukkan pada Gambar 1



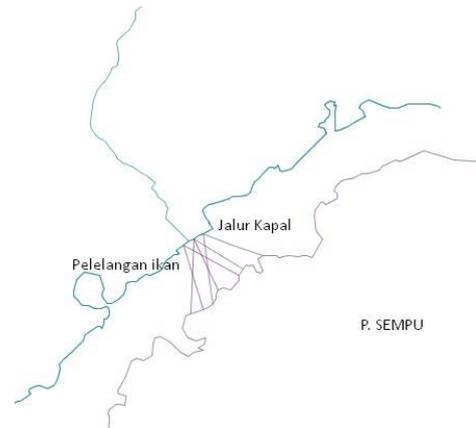
Gambar 1. Hasil peta dari *OpenStreetMap.org*

Setelah mendapatkan peta yang diinginkan, langkah selanjutnya adalah menyeleksi *path*/jalan utama yang digunakan. Setelah ini peta diolah kembali sehingga mendapatkan jalan utama yang nantinya digunakan sebagai simulasi seperti ditunjukkan Gambar 2. Untuk menyeleksi *path*/jalan utama digunakan aplikasi *OpenJump*.



Gambar 2. Jalan utama yang digunakan untuk simulasi

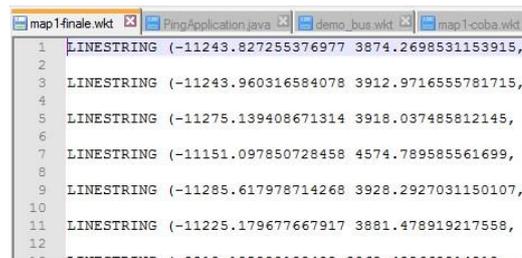
Jalan utama sudah ditentukan, selanjutnya adalah memodelkan rute kapal. Dikarenakan daerah pantai ini adalah tempat pelelangan ikan, maka kapal bergerak dari satu daerah secara bersamaan. Rute kapal dimodelkan pada Gambar 3



Gambar 3. Peta jalan utama dan jalur kapal

Dari data *map* tersebut kemudian didigitasi menggunakan aplikasi *OpenJump*, data hasil digitasi ini berupa file *.wkt* (*well known format*). File dengan format *.wkt* berisi angka-angka koordinat yang digunakan dalam *One Simulator* sebagai *path* atau peta agar *node* (kendaraan) bisa berjalan seperti kondisi *real* lingkungan.

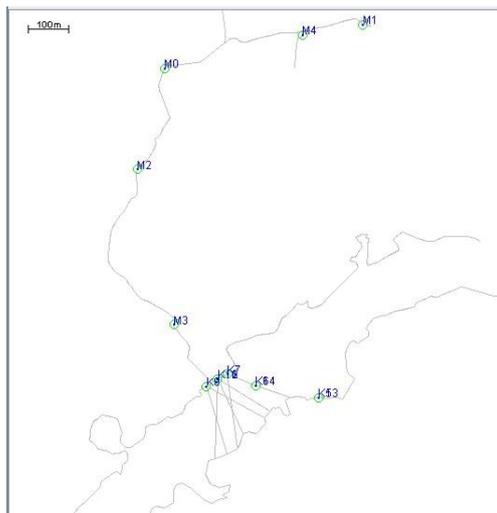
Angka koordinat dalam file *.wkt* seperti yang ditunjukkan oleh Gambar 4



Gambar 4. File format *.wkt*

Dalam skenario ini pergerakan *node* pada peta diatur sedemikian rupa supaya menyerupai kondisi sebenarnya. *Node* berjalan sesuai dengan *path* yang sesuai dengan peta jalan menuju daerah pedalaman.

Pada Penelitian ini menggunakan dua *node* yaitu *node* M mewakili kendaraan yang berjalan di jalan utama pulau Jawa yaitu Mobil yang menuju Tempat Pelelangan Ikan. Dan *node* K mewakili Kapal yang melintasi laut menuju Pulau Sempu seperti ditunjukkan pada Gambar 5.



Gambar 5. Pergerakan *Node* M dan K mengikuti *path*

Setelah itu dibutuhkan pengaturan dengan melakukan penyesuaian kebutuhan skenario simulasi dengan cara mengedit file `default_setting.txt`

3.2. Konfigurasi Simulasi

Tabel 1 Konfigurasi Simulasi

<i>Simulation Time</i>	7200 second
<i>Buffer Size</i>	5M
<i>Number of Nodes</i>	10 - 50 Node
<i>Message Size</i>	20k - 50k
<i>Message Generation (Event Interval)</i>	25-35
<i>Mobility Speed</i>	30 – 50 (km/j)
<i>Mobility Model</i>	Map Route Movement
<i>Signal Range</i>	10 m
<i>Transmit Speed</i>	2Mbps

Penjelasan konfigurasi simulasi pada Tabel 1 bisa dijelaskan dalam uraian berikut ini :

- *Simulation Time* : Merupakan lama waktu simulasi, penentuan lama waktu ini adalah menurut pengalaman penulis yang pernah ke kawasan Pantai sendang biru, waktu yang dibutuhkan dari Jalan Raya Lintas Selatan Jawa menuju Pantai Sendang Biru lalu menyeberang ke Pulau Sempu memiliki estimasi 2 jam (7200s)
- *Buffer Size* : *Buffer Size* dalam simulasi memang dari awal ditentukan sebesar 5Mb. *Buffer Size* sebesar ini dirasa sudah cukup untuk memuat banyak pesan.
- *Number of Nodes* : Jumlah *Node* yang dipilih adalah berkisar 10 – 50 *Node*. Jumlah ini adalah asumsi jumlah kendaraan dan kapal

yang lalu lalang dalam jumlah minimal dan maksimalnya.

- *Message Size* : ukuran pesan yang berada dalam konfigurasi simulasi ini secara default adalah 20k-50k, yang diharapkan dari besarnya ukuran pesan ini adalah memaksimalkan potensi pesan terkirim/bertukar informasi.
- *Mobility Speed* : Kecepatan kendaraan yang akan digunakan, kecepatan 30-50 Km/j adalah asumsi kecepatan rata rata.
- *Mobility Model* : adalah model pergerakan node, *Map Route Model* adalah model pergerakan yang mengikuti Peta yang sudah ditentukan.
- *Signal Range* : jangkauan sinyal yang digunakan dalam simulasi ini adalah menggunakan sinyal WiFi. Sinyal WiFi memiliki sifat yang mudah di implementasikan pada banyak perangkat dan memiliki kecepatan pengiriman data yang cepat
- *Transmit Speed* : Adalah kecepatan pengiriman data. Nilai sebesar 2Mbps ini dipilih karena dengan kecepatan ini akan lebih banyak pesan yang akan terkirim.

3.3. Pengujian

Pengujian dilakukan dengan mengubah parameter seperti *Routing*, jumlah *node* serta ukuran file yang dikirim didalam command file `default_setting.txt`. Pada pengujian dibandingkan 2 model *Routing* yakni *Prophet* dan *Spray and Wait*. Pengaturan dalam *One Simulator* untuk menggunakan kedua model *Routing* ini dengan cara mengisikan command pada syntaks pada masing masing Group. Kemudian ditentukan juga ukuran file yang dikirim yaitu 20kb-50kb serta jumlah *node* diubah pada masing-masing kelompok.

4. HASIL PENGUJIAN DAN ANALISIS

4.1. Hasil Pengujian

Setelah melakukan pengujian menggunakan *Routing Prophet* dengan mengganti jumlah *node* antara 10 – 50 *node* didapatkan hasil seperti ditunjukkan Tabel 2

Tabel 2 Pengujian *Routing Prophet*

No	Param	10	20	30	40	50
1	<i>time (s)</i>	7200	7200	7200	7200	7200
2	<i>Created</i>	244	244	244	244	244
3	<i>Started</i>	84	100	227	258	413
4	<i>Relayed</i>	84	100	227	258	413
5	<i>Aborted</i>	0	0	0	0	0

6	<i>Dropped</i>	0	0	0	0	0
7	<i>Removed</i>	0	0	0	0	0
8	<i>Delivered</i>	40	36	41	30	40
9	<i>Delivery prob</i>	0.163	0.147	0.168	0.123	0.16
10	<i>Overhead ratio</i>	1.100	1.777	4.536	7.600	9.32
11	<i>Latency avg</i>	1486.12	1806.97	1887.35	1570.89	1871.3
		5	2	8	3	37

Sedangkan untuk pengujian *Routing Spray and Wait*. Setelah melakukan pengujian dengan mengganti jumlah *node* didapatkan hasil seperti ditunjukkan Tabel 3.

Tabel 3 Pengujian *Routing Spray and Wait*

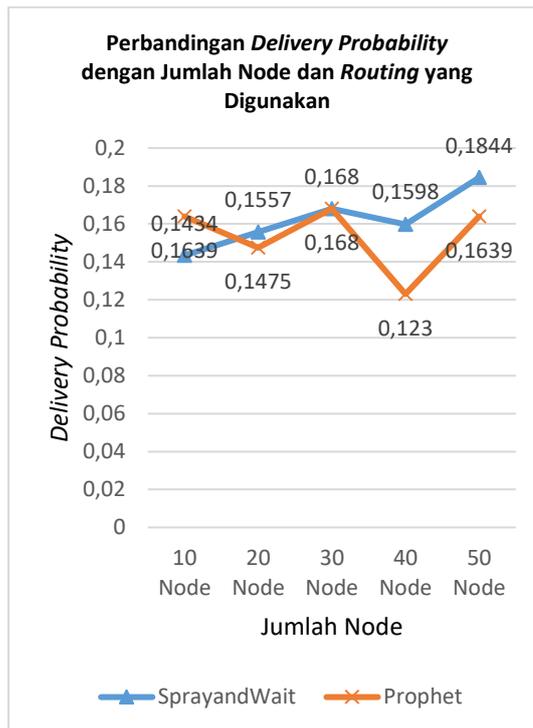
No	Param	10	20	30	40	50
1	<i>time (s)</i>	7200	7200	7200	7200	7200
2	<i>Created</i>	244	244	244	244	244
3	<i>Started</i>	246	404	634	759	884
4	<i>Relayed</i>	246	404	634	759	884
5	<i>Aborted</i>	0	0	0	0	0
6	<i>Dropped</i>	0	0	0	0	0
7	<i>Remove d</i>	0	0	0	0	0
8	<i>Delivered</i>	35	38	41	39	45
9	<i>Delivery prob</i>	0.143	0.155	0.168	0.159	0.184
10	<i>Overhead ratio</i>	6.028	9.631	14.46	18.46	18.64
		6	6	34	15	44
11	<i>Latency avg</i>	1566.888	1642.668	1825.768	1408.423	1498.977

4.2. Analisis

4.2.1. Delivery Probability

Setelah pengujian didapatkan hasil laporan kinerja termasuk perhitungan *Delivery Probability*, *Overhead Ratio* dan *latency average*. Hasil dari perhitungan *Delivery Probability* menggunakan *Routing Prophet* dengan menggunakan 10, 20, 30, 40 dan 50 *node* terdapat pada Tabel 2. Hasil dari perhitungan *Delivery Probability* menggunakan *Routing Spray and Wait* dengan menggunakan 10, 20, 30, 40 dan 50 *node* terdapat pada Tabel 3.

Perbandingan hasil *Delivery Probability* dengan menggunakan kedua *Routing* tersebut terlihat pada Gambar 6.



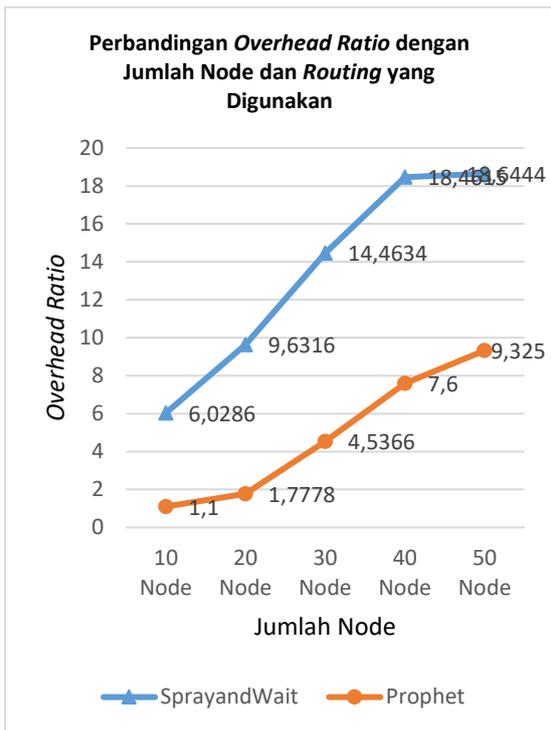
Gambar 6 Hasil *Delivery Probability*

Untuk *Routing Prophet* kinerja optimal saat jumlah *node* 30 yaitu 0,1680 dan kinerja terendah saat jumlah *node* 40 yaitu 0,1230. Sedangkan *Delivery Probability* pada *Routing Spray and Wait* paling optimal ketika jumlah *node* 50 yaitu 0,1844 dan memiliki kinerja terendah ketika jumlah *node* 10 yaitu 0,1434. Sehingga dapat diambil kesimpulan bahwa *Routing Spray and Wait* memiliki probabilitas pengiriman yang lebih baik dibanding *Routing Prophet* disaat *traffic* ramai (lebih dari 10 *node*). Dan disaat *traffic* sepi (10 *Node*) *Routing Prophet* lebih baik dibandingkan dengan *Routing Spray and Wait*.

4.2.2. Overhead Ratio

Hasil dari perhitungan *Overhead Ratio* menggunakan *Routing Prophet* dengan menggunakan 10, 20, 30, 40 dan 50 *node* terdapat pada Tabel 2. Hasil dari perhitungan *Overhead Ratio* menggunakan *Routing Spray and Wait* dengan menggunakan 10, 20, 30, 40 dan 50 *node* terdapat pada Tabel 3

Perbandingan hasil *Overhead Ratio* dengan menggunakan kedua *Routing* tersebut terlihat pada Gambar 7.



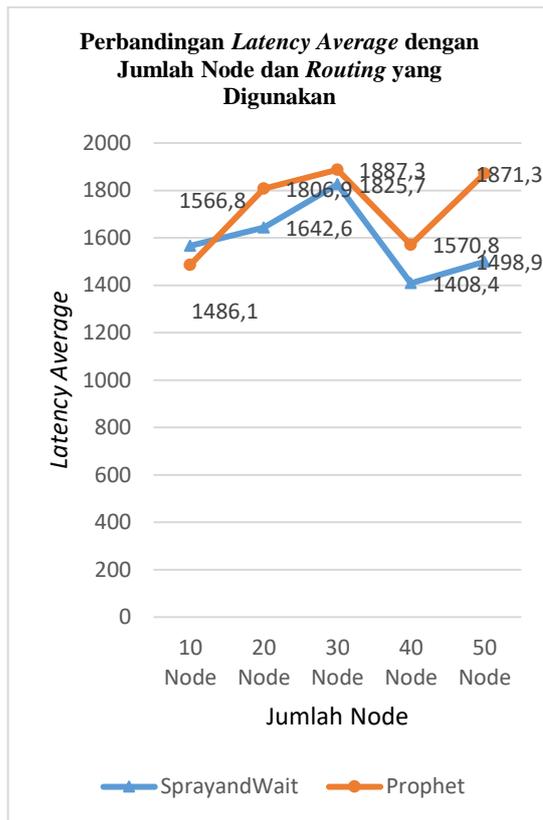
Gambar 7 Hasil Overhead Ratio.

Nilai dari *Overhead Ratio* adalah dipengaruhi oleh lamanya paket disimpan sebelum diteruskan ke *node* tujuan. Dari Gambar 7 bisa didapatkan informasi bahwa antara *Prophet* maupun *Spray and Wait* memiliki nilai *Overhead Ratio* yang paling kecil saat *node* berjumlah 10. Semakin rendah nilai *Overhead Ratio* menunjukkan keoptimalan *Routing* tersebut. Sehingga dapat diambil kesimpulan semakin banyak *node* maka paket disimpan semakin lama dan membutuhkan *resource* yang lebih tinggi. Dari data tersebut terlihat bahwa *Routing Prophet* memiliki *Overhead Ratio* yang lebih kecil dibandingkan dengan *Routing Spray and Wait*.

4.2.3. Latency Average

Hasil dari perhitungan *latency average* menggunakan *Routing Prophet* dengan menggunakan 10, 20, 30, 40 dan 50 *node* terdapat pada Tabel 2. Hasil dari perhitungan *latency average* menggunakan *Routing Spray and Wait* dengan menggunakan 10, 20, 30, 40 dan 50 *node* terdapat pada Tabel 3.

Perbandingan hasil *latency average* dengan menggunakan kedua *Routing* tersebut terlihat pada Gambar 8.



Gambar 8 Hasil Latency Average

Nilai *latency average* ini menunjukkan seberapa lama pesan akan tertunda sebelum berhasil diterima *node* tujuan, semakin rendah nilai *latency average* maka menunjukkan keoptimalan sebuah *Routing*. Dari Gambar 8 bisa didapatkan informasi bahwa *Routing Spray and Wait* memiliki waktu tunda yang paling rendah saat *node* berjumlah 40 yaitu 1408.4231 ms . Sedangkan pada *Routing Prophet* nilai *latency average* terendah terdapat pada jumlah *node* 10 yaitu 1486.1250 ms.

Sehingga dapat diambil kesimpulan bahwa *Routing Spray and Wait* memiliki *latency average* yang lebih baik dibanding *Routing Prophet* disaat *traffic* ramai (lebih dari 10 *node*). Dan disaat *traffic* sepi (10 *Node*) *Routing Prophet* lebih optimal dibandingkan dengan *Routing Spray and Wait*.

Routing Spray and Wait lebih optimal dalam hal *latency average* karena tidak membutuhkan syarat untuk melakukan *forwarding* pesan. Sementara pada *Routing Prophet* ketika *node-node* bertemu, waktu untuk *forwarding* pesan cukup lama karena saat bertemu dibutuhkan waktu untuk saling bertukar informasi *Routing* dan perhitungan nilai *delivery predictability*, sampai akhirnya melakukan *forwarding* pesan.

5. KESIMPULAN

Berdasarkan pengujian terhadap lingkungan terpercil dapat diambil kesimpulan bahwa:

1. Nilai Delivery Probability cenderung semakin tinggi jika jumlah *Node* juga bertambah. Perbedaan nilai Delivery Probability pada *Prophet* dan *Spray and Wait* ini didasarkan pada sifat masing masing *Routing*. Pada *Prophet* proses penerusan pesan dilakukan setelah *node* mengolah informasi/*knowledge* dari *node* sebelumnya. Sedangkan *Spray and Wait* langsung mereplika pesan saat bertemu *node* dan langsung meneruskannya. Hal ini yang mengakibatkan nilai delivery probability dari *Spray and Wait* lebih tinggi dibanding dengan *Prophet*.
2. Pada kinerja Overhead Ratio, *Spray and Wait* memiliki nilai yang lebih besar dibandingkan dengan *Prophet*. Pertambahan jumlah *Node* juga akan menambah nilai dai Overhead ratio ini. Hal ini dikarenakan pada mekanisme replica di *Spray and Wait* pesan banyak yang ganda/*Redundant*. Sedangkan pada *Prophet* pesan baru akan diteruskan ketika mempelajari informasi pada pertemuan *node* sebelumnya, maka dari itu Pesan ganda pada *Prophet* akan lebih sedikit dibandingkan dengan *Spray and Wait*. Semakin tinggi nilai Overhead Ratio ini maka semakin banyak sumberdaya yang dibutuhkan.
3. Untuk kinerja Latency Average terlihat *Prophet* memiliki latency yang cenderung lebih besar dibandingkan dengan *Spray and Wait* meskipun hanya sedikit. Dari Gambar 5.3 bisa dilihat bahwa dengan bertambahnya jumlah *Node* maka nilai *Latency Average* nya pun cenderung bertambah. Penyebab nilai ini lebih tinggi pada *Prophet* adalah *Prophet* membutuhkan lebih banyak waktu untuk mengirimkan pesan karena dalam system pertukaran pesan, *Prophet* perlu mengolah informasi dari pertemuan *node* sebelumnya. Berbeda dengan *Spray and Wait* yang langsung mengirimkan pesan ketika bertemu *node* lainnya.

Dari ketiga kinerja *Routing DTN* diatas dapat dinyatakan bahwa pada *Spray and Wait*, memiliki nilai Delivery Probability yang lebih besar, dengan nilai *Average delay* yang lebih kecil, tetapi lebih boros sumberdaya. Sedangkan untuk *Prophet* memiliki nilai *Delivery*

Probability yang lebih kecil, dengan *Average Delay* yang lebih besar, tetapi keunggulan nya adalah pada lebih hemat sumberdaya. Dengan demikian *Spray and Wait* lebih dipilih untuk digunakan di daerah terpercil.

DAFTAR PUSTAKA

- Fall, K., "A delay-tolerant network architecture for challenged internets." Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM, 2003.
- Ilham A. A. ., Niswar M., Agussalim. 2012. "Evaluasi dan Optimisasi Model *Routing* pada *Delay Tolerant Network (DTN)* untuk Pengiriman Data ke Desa Terpercil ", Universitas Hasanuddin, Makassar.
- Kominfo, 2006. "Siaran Pers No. 104/DJPT.1/KOMINFO/VIII/2006", http://www.postel.go.id/info_view_c_2_6_p_1463.htm, diakses 7 Juli 2017
- Magdalena, T. P. 2014, "Analisis Penggunaan Protokol *Routing Prophet* Pada *DTN* untuk Sistem Berbagi Informasi Digital di Daerah Pedalaman", Universitas Brawijaya, Malang.
- NetBeans IDE . "NetBeans IDE Features". <https://NetBeans.org/features/> , diakses 8 Januari 2017
- OpenJump. "Open Jump". <http://OpenJump.org/index.html>, diakses 20 Pebruari 2017
- OpenStreet. "Open Street Map", https://wiki.OpenStreetMap.org/wiki/Main_Page, diakses 20 Pebruari 2017
- OneSimulator."The ONE", <https://www.netlab.tkk.fi/tutkimus/DTN/theone/>, diakses 20 Pebruari 2017
- Thrasylvoulos S., Konstantinos P., Cauligi S. R., "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks" SIGCOMM'05 Workshops, August 22–26, 2005, Philadelphia, PA, USA. Copyright 2005 ACM 1-59593-026-4/05/0008
- Warthman, Forrest. "Delay-Tolerant Networks (DTNs) A Tutorial" Warthman

Associates. 2003.