

**ANALISIS *TRANSFER DATA PADA BUNDLE
PROTOCOL WIRELESS ROUTER* BERGERAK DI
JARINGAN *DELAY TOLERANT NETWORK (DTN)***

SKRIPSI

Diajukan Sebagai Syarat Untuk Menyelesaikan
Pendidikan Program Strata-1 Pada Jurusan Sistem Komputer



Oleh :

**FITRI NOVIANI
09111001005**

**JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2016**

**ANALISIS *TRANSFER DATA PADA BUNDLE
PROTOCOL WIRELESS ROUTER* BERGERAK DI
JARINGAN *DELAY TOLERANT NETWORK (DTN)***

TUGAS AKHIR

Diajukan Sebagai Syarat Untuk Menyelesaikan

Pendidikan Program Strata-1 Pada Jurusan Sistem Komputer



Oleh :

Fitri Noviani

09111001005

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2016

LEMBAR PENGESAHAN

ANALISIS *TRANSFER DATA PADA BUNDLE PROTOCOL* *WIRELESS ROUTER BERGERAK DI JARINGAN* *DELAY TOLERANT NETWORK (DTN)*

TUGAS AKHIR

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh :

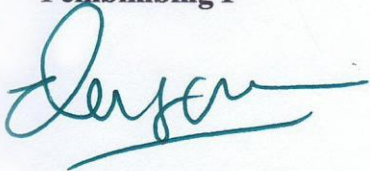
Fitri Noviani

09111001005

Mengetahui

Inderalaya, Oktober 2016

Pembimbing I



Deris Stiawan, Ph.D

NIP. 197806172006041002

Pembimbing II



Sri Desy Siswanti, M. T.

NIPUS. 197802232015109201

Ketua Jurusan Sistem Komputer



Rossi Passarella, M. Eng

NIP. 197806112010121004

HALAMAN PERSETUJUAN

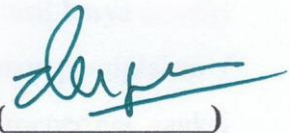
Telah diuji pada :

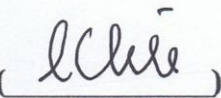
Hari : Jum'at

Tanggal : 16 September 2016

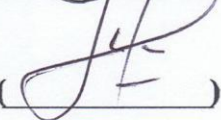
Tim Penguji :

- | | |
|---------------|----------------------------|
| 1) Ketua | : Deris Stiawan, Ph.D. |
| 2) Sekretaris | : Sri Desy Siswanti, M.T. |
| 3) Anggota I | : Ahmad Fali Oklilas, M.T. |
| 4) Anggota II | : Huda Ubaya, M.T. |

()

()

()

()

Mengetahui,
Ketua Jurusan Sistem Komputer,



Rossi Passarella, M. Eng
NIP. 197806112010121004

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Fitri Noviani

NIM : 09111001005

Judul : Analisis *Transfer Data Pada Bundle Protocol Wireless Router Bergerak*
di Jaringan *Delay Tolerant Network (DTN)*


Menyatakan bahwa laporan tugas akhir saya merupakan hasil karya sendiri dan bukan hasil penjiplakan / *plagiat*. Apabila ditemukan unsur penjiplakan / *plagiat* dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.



Inderalaya, Oktober 2016




Fitri Noviani

HALAMAN PERSEMBAHAN

“Dalam perkataan, tidak mengapa Anda merendahkan diri, tetapi dalam aktivitas tunjukkan kemampuan Anda”

– Nabi Muhammad SAW-

“Yakinlah ada sesuatu yang menantimu selepas banyak kesabaran (yang kau jalani) yang akan membuatmu terpana hingga kau lupa pedihnya rasa sakit ”

-Ali bin Abi Thalib-

“Kau hanya harus lebih Kuat dari rasa Lelah dan Malasmu”

-Fitri Noviani-

Tugas Akhir ini ku persembahkan untuk :

- *Allah SWT*
- *Ibu dan bapakku, yang selalu mendo'akan di setiap langkahku*
- *Adik-adikku yang mensupport selalu*
- *Seluruh keluargaku tercinta*
- *Sahabat-sahabatku yang luar biasa*
- *Bidikmisi, karnanya-lah saya bisa mendapatkan kesempatan melanjutkan studi*
- *Serta almamater-ku Universitas Sriwijaya*

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah swt karena berkat, rahmat dan karunia-Nya lah, penulis dapat menyelesaikan laporan Tugas Akhir ini yang berjudul “Analisis *Transfer Data* pada *Bundle Protocol Wireless Router* Bergerak di Jaringan *Delay Tolerant Network* (DTN)”.

Sholawat dan salam untuk baginda Nabi Muhammad SAW yang telah menerangi bumi indah ini dengan agama *rahmatan lil alamin*, memberikan percikan kedamaian lewat kesabaran. Yang dengan tauladannya mampu membuat siapa saja terus ingin menjadi baik seperti beliau.

Penulis berharap tulisan ini dapat bermanfaat bagi semua, baik menjadi tambahan bahan bacaan ataupun sebagai referensi bagi yang tertarik mengembangkan lebih lanjut sistem DTN ini.

Dalam penyusunan laporan ini, penulis banyak mendapatkan bantuan dari berbagai pihak yang berupa bimbingan, saran, petunjuk, keterangan dan data, baik yang diberikan secara lisan maupun tulisan. Oleh karena itu, dalam kesempatan ini, dengan segala ketulusan dan keikhlasan penulis menghanturkan terima kasih yang tak terhingga kepada pihak-pihak yang membantu penulis dalam menyelesaikan laporan ini. Semoga apa yang telah diberikan oleh mereka kepada penulis mendapatkan balasan dari Tuhan Yang Maha Esa. Disamping itu penulis juga mengucapkan terima kasih kepada :

1. Kedua Orangtuaku Tercinta (Bapak Sukardi dan Ibu Fatimah yang telah mengajar, mendidik, membesarkan, merawat, memenuhi segala kebutuhanku dan tak pernah lelah mendoakanku).
2. Saudara-saudariku Tersayang (Dwi Novika Amelia dan Octa Wahyu Tri Prasetya) yang selalu mendukung dan menyemangatiku.
3. Rossi Passarella, M.Eng. selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Deris Stiawan, Ph.D. selaku Dosen Pembimbing I Tugas Akhir penulis yang telah memberikan semangat, pengetahuan dan bimbingan secara penuh.

5. Sri Desy Siswanti, M.T. selaku Dosen Pembimbing II Tugas Akhir yang telah memberikan penulis saran untuk mengangkat DTN sebagai bahasan Tugas Akhir penulis dan tidak henti-hentinya memberikan semangat serta bimbingan secara penuh.
6. Erwin, M.Si. selaku Dosen Pembimbing Akademik penulis.
7. Ahmad Fali Oklilas, M.T. dan Huda Ubaya, M.T. selaku Dosen Penguji sidang Tugas Akhir yang telah memberi banyak masukan berupa kritik dan saran serta ilmu yang bermanfaat sehingga tulisan ini menjadi lebih baik.
8. Iis Oktaria, A.Md. selaku admin sistem komputer yang telah memberikan pelayanan yang sangat baik kepada mahasiswa sistem komputer umumnya dan kepada penulis khususnya mulai dari pertama kali masuk hingga kini telah menjadi alumni jurusan sistem komputer fasilkom unsri.
9. Seluruh Dosen Jurusan Sistem Komputer Fasilkom Unsri.
10. Sahabat baik dan seperjuanganku *DTN Team* (Putri dan Lina) serta sahabat baikku (Septiyani) yang selalu bersedia mendengarkan keluh dan kesah penulis. Teman-teman Sistem Komputer 2011 (Ade, Chandra, Zalbin, Akhram, Mala, Wulan, Hikmah, Robby, Cacing, Hafizh, Yogi, Amel dan teman” lainnya yang tidak dapat saya sebutkan satu persatu), kakak dan adik tingkat dari Sistem Komputer (Kak Atika, Kak Karlina, Kak Ipeh, Cora), Keluarga Besar Himasisko, Keluarga Besar Ilmu Komputer Universitas Sriwijaya, dan semua pihak yang turut membantu dengan ikhlas sehingga laporan ini bisa terselesaikan.
11. Teman-teman baikku Fikri, Diah, dan Topan yang selalu memberi semangat dengan tulus menemaniku 10 tahun lebih ini.
12. Civitas akademika Jurusan Sistem Komputer, *Office Boy*, dan Staff Administrasi Fakultas Ilmu Komputer Universitas Sriwijaya.
13. Bidikmisi, yang telah memberikanku kesempatan untuk melanjutkan studiku hingga selesai.
14. Semua pihak yang telah terlibat dalam penyelesaian tugas akhir yang tidak dapat disebutkan namanya satu-persatu.

Sejatinya kesempurnaan hanyalah milik Allah, pembaca akan mendapati kekurangan dan ke-*khilaf*-an penulis dalam menyelesaikan tugas akhir ini. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan, sebagai bentuk perbaikan dikemudian hari. Semoga peneliti selanjutnya dapat mengambil pelajaran yang baik dan memperbaiki yang kurang, sehingga mampu memberikan karya yang lebih baik lagi.

Palembang, Oktober 2016

Penulis,

Fitri Noviani

Analisis Transfer Data Pada Bundle Protocol Wireless Router Bergerak Di Jaringan Delay Tolerant Network (DTN)

Fitri Noviani (09111001005)

Departement of Computer Engineering, Faculty of Computer Science

Sriwijaya University

Email : fitrinoviani13@yahoo.com

Abstract

Delay Tolerant Network (DTN) is a tolerant computer network or it does not concern about delay (time delay). It provides store-carry-forward communication model which is based on the mobility node to transfer data between separated nodes geographically so it can be used as an alternative solution for network communications in remote areas which possibility have bad connection problem during data transfer is still ongoing. Custody transfer is been a mechanism to improve the reliability of the delivery by using reability hop-by-hop. It provides responsibility for each data transfer to a destination between its node. In DTN, there is a special layer which is called bundle layer. Bundle layer stores and forwards the entire bundle between nodes communicates with DTN. Examination was carried out in LoS (Line of Sight) and N-LoS (non Line of Sight) condition with range test was up to 70 meters and maximum data test were 2000 MB, bundle can only last as long as 3600 seconds, while it was passing from 3600 seconds (time expiration), the bundle lost. The examination result showed that at the time of data transfer is doing, it had a tendency to send size of data / files larger, then the time required to send the files will also be greater.

keywords : *DTN, custody transfer, bundle protocol, DTN wireless router*

Analisis Transfer Data Pada Bundle Protocol Wireless Router Bergerak Di Jaringan Delay Tolerant Network (DTN)

Fitri Noviani (09111001005)

Departement of Computer Engineering, Faculty of Computer Science

Sriwijaya University

Email : fitrinoviani13@yahoo.com

Abstrak

Delay Tolerant Network (DTN) merupakan jaringan komputer yang bersifat toleran atau tidak mempermasalahkan *delay* (waktu tunda). *Delay Tolerant Network (DTN)* menyediakan model komunikasi *store-carry-forward* yang bertumpu pada mobilitas node untuk *mentransfer* data antara node yang terpisah secara geografis sehingga dapat digunakan sebagai solusi alternatif untuk jaringan komunikasi pada daerah terpencil yang memiliki permasalahan kemungkinan putusnya koneksi saat *transfer* data masih berlangsung. *Custody transfer* merupakan sebuah mekanisme untuk meningkatkan kehandalan dari pengiriman dengan menggunakan *reability hop-by-hop*. Pada *custody transfer* memberikan tanggung jawab dalam setiap transfer data ke tujuan antar nodenya. Pada DTN terdapat layer khusus yang disebut *bundle layer*. *Bundle layer* menyimpan dan meneruskan keseluruhan bundle antara node-node yang berkomunikasi dengan DTN. Pengujian dilakukan pada kondisi LoS (*Line of Sight*) maupun N-LoS (*Non Line of Sight*) dengan jarak uji mencapai 70 meter dan data uji maksimum yaitu 2000 MB, bundle hanya dapat bertahan selama 3600 detik, apabila lewat dari 3600 detik (*time expiration*), maka bundle akan hilang. Hasil pengujian menunjukkan bahwa pada saat transfer data dilakukan memiliki kecenderungan semakin besar ukuran data/file yang dikirimkan, maka waktu yang dibutuhkan untuk mengirim file tersebut juga akan semakin besar.

kata kunci : *DTN, custody transfer, bundle protocol, DTN wireless router*

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR.....	vi
ABSTRACT	ix
ABSTRAK	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xvii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Tujuan dan Manfaat	2
1.2.1 Tujuan.....	2
1.2.2 Manfaat.....	2
1.3 Perumusan Masalah dan Batasan Masalah	3
1.3.1 Perumusan Masalah.....	3
1.3.2 Batasan Masalah.....	3
1.4 Metodologi Penelitian	4
1.5 Sistematika Penulisan	6

BAB II TINJAUAN PUSTAKA

2.1 Pendahuluan.....	7
2.2 Delay Tolerant Network (DTN).....	9
2.3 Bundle Protocol	11
2.4 Custody Transfer.....	15
2.5 DTN2	17

	Halaman
2.5 Propagasi Gelombang	18

BAB III METODOLOGI PENELITIAN

3.1 Pendahuluan	20
3.2 Kerangka Kerja Penelitian (<i>Framework</i>)	20
3.3 Perancangan Sistem	22
3.4 Spesifikasi Data	24
3.5 Spesifikasi Perangkat	24
3.5.1 Perangkat Hardware	25
3.5.2 Perangkat Software	26
3.6 Konfigurasi Sistem DTN	27
3.6.1 Konfigurasi Routing DTN	27
3.6.2 Konfigurasi Interface	29
3.6.3 Konfigurasi Routing DHCP	30
3.7 Skenario Penelitian	30
3.8 Mekanisme Pengiriman Data	31
3.9 Mekanisme Pengambilan Data	34

BAB IV PENGUJIAN DAN ANALISA

4.1 Pendahuluan	37
4.2 Pengujian dan Analisa	37
4.2.1 Pengujian dan Analisa Waktu Transfer Data	39
4.2.1.1 Pengujian dalam Kondisi LoS (<i>Line of Sight</i>)	39
4.2.1.2 Pengujian dalam Kondisi N-LoS (<i>Non Line of Sight</i>)	47
4.2.1.3 Perbandingan Lama Pengiriman Data antara LoS dan N-LoS	54
4.2.2 Data Uji dan Lama Pengiriman Maksimum pada Setiap Jarak	60
4.2.3 Parameter Uji dalam <i>Log Bundle</i>	64
4.2.4 <i>Bundle Expiration</i> pada DTN	67
4.2.5 Perhitungan Kondisi Pengujian apakah LoS, n-LoS, atau N-LoS	71

	Halaman
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	76
5.2 Saran	77
DAFTAR PUSTAKA	xviii
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar 1.1 Blok Diagram Metodologi Penelitian.....	5
Gambar 2.1 Diagram Hiraerkial Penelitian Tugas Akhir.....	8
Gambar 2.2 Lapisan Internet dan Lapisan DTN.	12
Gambar 2.3 Letak Bundle Layer pada Lapisan OSI	12
Gambar 2.4 Metode Simpan dan Teruskan (Store and Forward)	13
Gambar 2.5 DTN Routing.....	15
Gambar 2.6 Custody Transfer pada DTN.	16
Gambar 2.7. Perbedaan LoS, n-LoS, dan N-LoS.....	19
Gambar 3.1 Konsep Penelitian.....	20
Gambar 3.2 Kerangka Kerja Penelitian.....	22
Gambar 3.3 Skenario Penelitian Sistem DTN.....	23
Gambar 3.4 Skenario Penelitian DTN dalam Kondisi <i>LoS</i>	31
Gambar 3.5 Skenario Penelitian DTN dalam Kondisi <i>N-LoS</i>	31
Gambar 3.6 Diagram Alir Pengiriman Data.....	33
Gambar 4.1 t Pengiriman Pada Log Server.....	39
Gambar 4.2 t Pengiriman Pada Log Klien	39
Gambar 4.3 Grafik Lama Pengiriman dalam Jarak 15 Meter.	41
Gambar 4.4 Grafik Lama Pengiriman dalam Jarak 30 Meter.	43
Gambar 4.5 Grafik Lama Pengiriman dalam Jarak 50 Meter.	45
Gambar 4.6 Grafik Lama Pengiriman dalam Jarak 70 Meter.	47
Gambar 4.7 Grafik Lama Pengiriman dalam Jarak 15 Meter.	49
Gambar 4.8 Grafik Lama Pengiriman dalam Jarak 30 Meter.	51
Gambar 4.9 Grafik Lama Pengiriman dalam Jarak 50 Meter.	51
Gambar 4.10 Grafik Lama Pengiriman dalam Jarak 70 Meter.	54
Gambar 4.11 Grafik Perbandingan LoS dan N-LoS pada Jarak 15 Meter.....	56
Gambar 4.12 Grafik Perbandingan LoS dan N-LoS pada Jarak 30 Meter.....	57
Gambar 4.13 Grafik Perbandingan LoS dan N-LoS pada Jarak 50 Meter.....	59
Gambar 4.14 Grafik Perbandingan LoS dan N-LoS pada Jarak 70 Meter.....	60
Gambar 4.15 Sample Log Bundle Server.....	65
Gambar 4.16 Sample Log Bundle Klien	65

	Halaman
Gambar 4.17 Expiration pada Saat Data Uji Dikirim ke klien	67
Gambar 4.18 Data Tidak Terkirim Ke Klien	68
Gambar 4.19 Bundle Antrian	71

DAFTAR TABEL

	Halaman
Tabel 1 Tabel Spesifikasi Laptop.....	24
Tabel 2 Penamaan <i>Hostname</i> , <i>user</i> dan Pengaturan IP address.....	25
Tabel 3 IP Address Perangkat DTN.	33
Tabel 4 Tabel Ukuran Data Uji dalam <i>Bytes</i>	38
Tabel 5 Lama Pengiriman Data LoS dalam Jarak 15 Meter	40
Tabel 6 Lama Pengiriman Data LoS dalam Jarak 30 Meter.	42
Tabel 7 Lama Pengiriman Data LoS dalam Jarak 50 Meter.	44
Tabel 8 Lama Pengiriman Data LoS dalam Jarak 70 Meter.	46
Tabel 9 Lama Pengiriman Data N-LoS dalam Jarak 15 Meter	48
Tabel 10 Lama Pengiriman Data N-LoS dalam Jarak 30 Meter.	50
Tabel 11 Lama Pengiriman Data N-LoS dalam Jarak 50 Meter.	52
Tabel 12 Lama Pengiriman Data N-LoS dalam Jarak 70 Meter.	53
Tabel 13 Perbandingan Lama Pengiriman Data (s) dalam Kondisi LoS.	55
Tabel 14 Perbandingan Lama Pengiriman Data (s) dalam Kondisi N-LoS.	55
Tabel 15 Perbandingan LoS dan N-LoS pada Jarak 15 Meter	56
Tabel 16 Perbandingan LoS dan N-LoS pada Jarak 30 Meter.	57
Tabel 17 Perbandingan LoS dan N-LoS pada Jarak 50 Meter.	58
Tabel 18 Perbandingan LoS dan N-LoS pada Jarak 70 Meter.	59
Tabel 19 Data Uji yang Terkirim ke Tujuan pada Setiap Jarak dalam Kondisi LoS dan N-LoS.	62
Tabel 20 Lama Pengiriman Maksimum yang Terkirim ke Tujuan pada Setiap Jarak dalam Kondisi LoS dan N-LoS	62
Tabel 21 Data Uji Maksimum yang Terkirim ke Tujuan pada Setiap Jarak dalam Kondisi LoS dan N-LoS.	63

DAFTAR LAMPIRAN

Lampiran A. Lokasi Pengujian

Lampiran B. Program Autoconnech.sh

Lampiran C. Tampilan DTN

Lampiran D. Sample log Bundle DTN

Lampiran E. Berkas Sidang

BAB I

PENDAHULUAN

1.1. Latar Belakang

Delay Tolerant Network (DTN) merupakan arsitektur jaringan untuk menyediakan solusi bagi jaringan yang memiliki konektivitas yang terputus-putus, *long delay*, kecepatan data yang berbeda dan kesalahan rate yang tinggi [1]. DTN mengimplementasikan metode store and forward message switching dengan menggunakan sebuah layer baru, yaitu *bundle layer*. Bundle layer menyimpan dan meneruskan keseluruhan *bundle* (atau sebuah fragmen saja) antara node-node yang berkomunikasi dengan DTN. Model komunikasi ini bertumpu pada mobilitas node untuk mengirimkan data antara node yang terpisah secara geografis.

Dalam DTN, terdapat istilah yang disebut *custody transfer* [2]. *Custody transfer* adalah mekanisme yang untuk meningkatkan kehandalan dari pengiriman dengan menggunakan *reability hop-by-hop*, satu atau lebih hop, dengan tidak adanya konektivitas *end-to-end* dengan mentransfer tanggung jawab pengiriman yang handal. *Custody Transfer* memberikan tanggung jawab dalam setiap transfer data ke tujuan antar nodenya (node satu ke node lain).

Konsep DTN dapat diterapkan pada jaringan dengan memanfaatkan kendaraan yang berperan sebagai kurir digital untuk menyebarluaskan data dari satu tempat ke tempat lain [3], Kurir digital ini sendiri berupa *Router Wireless* yang bergerak ke daerah yang membutuhkan akses jaringan dan mengumpulkan data-data, selanjutnya kurir digital bergerak ke tempat yang memiliki akses internet dan memproses permintaan-permintaan tersebut sehingga layanan internet dapat diterapkan dan disajikan untuk di suatu area yang memiliki karakteristik *delay* yang panjang, tingkat loss yang tinggi, dan tingkat konektivitas yang rendah.

Berdasarkan hal tersebut sistem jaringan ini dapat sebagai solusi alternatif untuk jaringan komunikasi pada daerah terpencil yang memiliki permasalahan putusnya koneksi saat *transfer* data masih berlangsung. Jika koneksi terputus, DTN akan menyimpan sebagian data yang sudah berhasil dikirimkan, sehingga saat koneksi tersambung kembali, proses transfer data dapat dilanjutkan tanpa mengulang proses dari awal, sedangkan pada jaringan yang menggunakan protokol TCP/IP, pengiriman data akan gagal apabila koneksi terputus saat proses pengiriman.

Merujuk kepada penelitian-penelitian [4][5][6][7] yang telah dilakukan sebelumnya mengenai topik *Delay Tolerant Network* (DTN), dalam tugas akhir ini, akan dilakukan penelitian dengan merancang sebuah skenario pengiriman data yang mengimplementasikan *wireless router* bergerak pada jaringan DTN dengan memperhatikan *custody transfer bundle* protokol serta memperhatikan waktu transfer dalam setiap data dari hasil log DTN.

1.2. Tujuan dan Manfaat

1.2.1 Tujuan

Adapun tujuan yang hendak dicapai dalam penelitian ini adalah:

1. Merancang dan mengimplementasikan konsep *wireless router* bergerak pada jaringan DTN.
2. Menganalisis *bundle* protokol *wireless router* bergerak DTN dengan pendekatan *custody transfer*.
3. Merancang skenario jaringan DTN dalam jaringan multihop yang mengandalkan *reability hop-by-hop*.

1.2.2 Manfaat

Adapun manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut

1. Dapat memberikan solusi alternatif pada jaringan di daerah yang koneksi internetnya sering terputus (*intermittent*), tingkat error yang tinggi dan aliran data yang tidak simetris.

2. Dapat digunakan pada daerah yang tidak memiliki infrastruktur internet dan telekomunikasi kabel maupun nirkabel yang menjangkau suatu daerah (*digital divide*) dengan memanfaatkan alternatif DTN.
3. Dapat memberikan gambaran mengenai *bundle* protokol yang berperan penting dalam arsitektur dari DTN.

1.3. Perumusan Masalah dan Batasan Masalah

1.3.1 Perumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang akan dibahas pada penelitian ini adalah :

1. Merancang arsitektur dari skenario jaringan DTN yang terdiri dari satu laptop sebagai *client*, satu laptop digunakan sebagai *wireless router* bergerak, dan satu server data, serta *Wifi Access Point (AP)* dan *Wireless USB Adapter*.
2. Melakukan instalasi dan mengkonfigurasi perangkat sehingga mendukung jaringan DTN.
3. Melakukan pengujian sistem DTN dalam 2 kondisi yaitu LoS (*Line of Sight*) dan N-LoS (*non Line of Sight*).
4. Menganalisis hasil log dari skenario DTN.

1.3.2 Batasan Masalah

Adapun batasan masalah pada penelitian ini, yakni sebagai berikut :

1. Menekankan pada sisi analisis dari *bundle* protokol DTN *wireless router* dengan memperhatikan waktu transfer setiap data serta analisis dari log DTN.
2. Dalam penelitian ini menggunakan pendekatan *custody transfer*.
3. Tidak melihat kecepatan dari kendaraan yang dipakai untuk membawa *router* bergerak, hanya melihat pada jarak berapa server-client masih dapat terkoneksi dengan baik.
4. Penelitian hanya dilakukan pada kondisi LoS (*Line of Sight*) dan N-LoS (*non Line of Sight*).

1.4. Metodologi Penelitian

Agar tujuan penelitian ini dapat tercapai berikut merupakan tahapan penelitian yang dilakukan:

1. Tahap Pertama (Perumusan Masalah)

Tahap ini dilakukan dengan cara menentukan permasalahan apa saja yang muncul pada penelitian sebelumnya tentang *Delay Tolerant Network (DTN)* menggunakan *wireless router* bergerak yang kemudian akan dibatasi lagi apa saja yang akan menjadi rumusan masalah yang nantinya akan dibahas dalam tugas akhir ini.

2. Tahap Kedua (Studi Pustaka / Literatur)

Tahap ini dilakukan dengan cara mencari dan membaca *literature* / referensi mengenai DTN, sehingga dapat menunjang penulisan laporan Tugas Akhir.

3. Tahap Ketiga (Perancangan Sistem Router Bergerak)

Tahap ini merupakan proses dari perancangan sistem dari skenario DTN yang akan dibangun. Tahap perancangan ini juga untuk menentukan perangkat komputer, sistem operasi dan simulator yang nantinya akan membantu dalam jalannya skenario DTN tersebut dalam proses pengiriman data.

4. Tahap Keempat (Pengujian)

Pada tahap ini merupakan tahap pengujian dan validasi dari perancangan *wireless router* bergerak dengan menggunakan prosedur pengujian yang telah ditentukan.

5. Tahap Kelima (Analisis Hasil)

Pada tahap ini merupakan tahap analisis data dari hasil *log* DTN yang dihasilkan dari perancangan *wireless router* bergerak menggunakan parameter yang telah ditentukan.

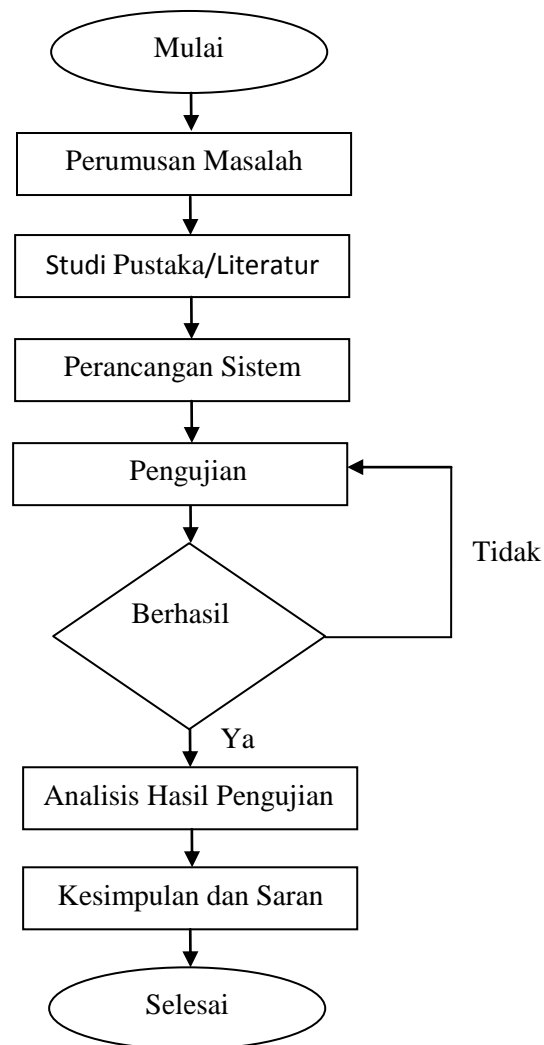
6. Tahap Keenam (Analisis Hasil)

Pada tahap ini akan ditarik kesimpulan dari setiap pengujian dari tahap sebelumnya kemudian dianalisis dengan tujuan mengetahui karakteristik hasil perancangan dan pengujian yang telah dilakukan.

7. Tahap Ketujuh (Penarikan Kesimpulan)

Pada tahap ini akan ditarik kesimpulan dari hasil analisis dan studi literatur serta untuk penulis selanjutnya apabila akan digunakan untuk bahan referensi penelitian selanjutnya.

Berikut ditampilkan metodologi penelitian secara visual dalam bentuk blok diagram yang merepresentasikan proses pelaksanaan penelitian :



Gambar 1.1. Blok Diagram Metodologi Penelitian

1.5. Sistematika Penulisan

Untuk lebih memudahkan dalam menyusun tugas akhir ini dan memperjelas isi dari setiap bab yang ada pada laporan ini, maka dibuatlah sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi menjelaskan tentang latar belakang penelitian, perumusan dan batasan masalah, tujuan dan manfaat dari penelitian, metodologi penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang dasar teori dari *DTN*, *bundle protocol*, *custody transfer*, *wireless router* bergerak, dan Routing DTN.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan secara bertahap dan terperinci tentang langkah-langkah (metodologi) yang digunakan untuk membuat kerangka kerja (*framework*) yang berisi perancangan kebutuhan *hardware* dan *software* yang menunjang tugas akhir.

BAB IV PENGUJIAN DAN ANALISA

Bab ini berisi pengujian hasil sementara dari penyelesaian tugas akhir serta analisa dari data yang didapat.

BAB V KESIMPULAN

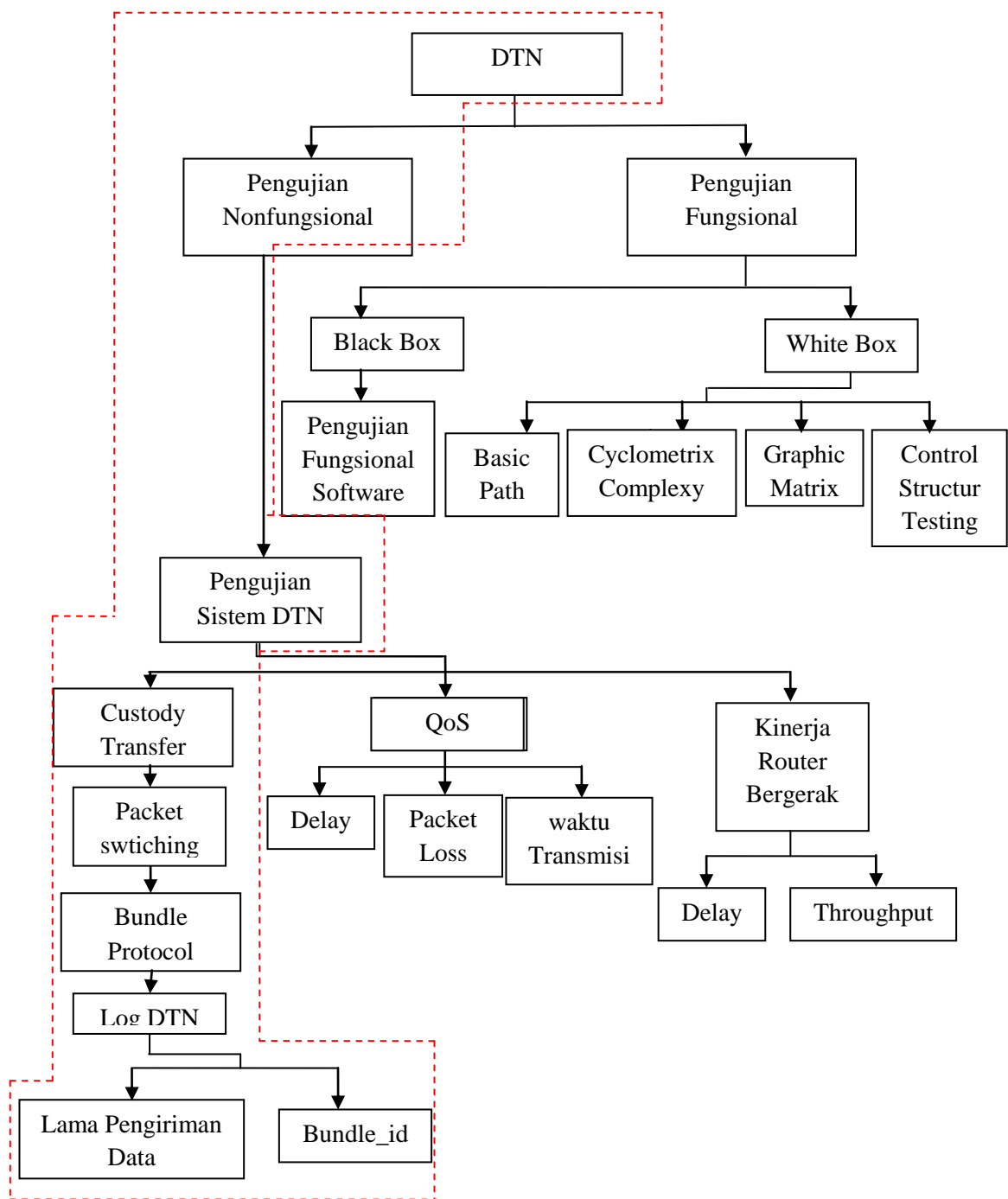
Bab ini berisi kesimpulan tentang apa yang diperoleh oleh penulis serta merupakan jawaban dari tujuan yang ingin dicapai pada bab 1 (pendahuluan)

BAB II

TINJAUAN PUSTAKA

2.1 Pendahuluan

Dalam penelitian tugas akhir ini dilakukan pengujian *non-fungsional*, yaitu pengujian yang dilakukan untuk menguji sistem dari DTN yang telah dibangun [8]. Pengujian dan pengambilan data yang dilakukan dapat memperlihatkan batasan dari sistem untuk mengirim *file* dengan ukuran data uji dari ukuran data terkecil hingga terbesar. Diagram hirarkial penelitian pada tugas akhir ini dapat dilihat pada gambar 2.1.



Gambar 2.1. Diagram *Hiraerkial* Penelitian Tugas Akhir

2.2 Delay Tolerant Network (DTN)

Konsep DTN pertama kali dikenalkan oleh Kevin Fall [9] untuk diterapkan pada jaringan dengan waktu tunda yang lama dan koneksi yang tidak dapat dipastikan. DTN mengatasi masalah-masalah yang terkait dengan *intermittent connectivity*, waktu tunda yang panjang, laju data asimetris, dan tingkat kesalahan yang tinggi. DTN menerapkan konsep simpan dan teruskan seperti yang digunakan sistem pos. Keseluruhan pesan atau pecahannya (*fragment*) dipindahkan dari tempat penyimpanan di sebuah titik (*node*) ke titik lain pada jalur antara sumber dan tujuan. Jaringan DTN ini sangat cocok di implementasikan pada daerah yang memiliki karakteristik jaringan yang tidak mumpuni untuk internet [10].

Arsitektur DTN [11] menyediakan metode interkoneksi antara *gateway* yang heterogen atau *proxy* dengan menyediakan rute untuk pesan secara *store and forward*. Arsitektur DTN menyediakan layanan yang hampir sama dengan email, namun lebih menekankan pada penamaan (*naming*), rute (*routing*), dan kemampuan keamanan. Node tidak selalu mendukung kemampuan yang diperlukan oleh arsitektur tersebut, kemungkinan karena tidak didukung oleh lapisan aplikasi proxy yang berfungsi sebagai aplikasi DTN.

Pada [3] menyatakan protokol Internet yang ada, tidak bekerja dengan baik pada beberapa lingkungan menyangkut beberapa asumsi fundamental yang dibangun dalam arsitektur Internet yaitu :

1. jalur *end-to-end* antara sumber dan tujuan tersedia satu sesi durasi komunikasi,
2. keandalan komunikasi dimana transmisi ulang berlangsung tepat waktu dan umpan balik yang stabil dari penerima data merupakan sarana yang efektif dalam memperbaiki kesalahan,
3. *end-to-end loss* yang relatif kecil,
4. *router* dan stasiun akhir mendukung protokol TCP/IP,
5. aplikasi tidak perlu mengetahui tentang kinerja komunikasi,
6. mekanisme keamanan *endpoint-based* cukup untuk mengatasi kekhawatiran yang terkait dengan keamanan,

7. pertukaran paket (*packet switching*) merupakan abstraksi yang paling tepat untuk interoperabilitas dan kinerja,
8. pemilihan satu rute antara pengirim dan penerima dalam menunjang kinerja komunikasi.

Fall [9] menyatakan bahwa arsitektur DTN di dasarkan pada sejumlah prinsip sebagai berikut.

1. Penggunaan variabel tunda (*delay*) yang panjang dalam komunikasi bertujuan membantu dan meningkatkan kemampuan jaringan dalam menyediakan penjadwalan atau keputusan pemilihan jalur yang mungkin dilalui.
2. Penggunaan sintaks penamaan yang mendukung berbagai penamaan dan konvensi pengalamatan untuk meningkatkan interoperabilitas.
3. Penggunaan penyimpanan (*storage*) dalam jaringan untuk mendukung operasi *store and forward* pada beberapa jalur dimana tidak terdapat jalur *end-to-end*, dalam rentang waktu yang panjang; dalam hal ini tidak mengharuskan kehandalan jalur end-to-end.
4. Menyediakan mekanisme keamanan yang melindungi infrastruktur dari penggunaan yang tidak sah dengan pemutusan lalu lintas (*traffic*) secepat mungkin.
5. Menyediakan layanan, pilihan pengiriman, dan mempertahankan umur data dan memberikan kesempatan pada jaringan agar mengirim data dengan lebih baik sehingga memenuhi kebutuhan pada aplikasi.

DTN bergantung pada pertukaran pesan (*message switching*) secara *store and forward*. Secara khusus setiap *node* DTN dalam jaringan lokal akan mengirimkan pesan ke *node-node* lainnya melewati rute, dimana pesan tersebut akan diteruskan dari sumber hingga ke tujuan [12].

Aplikasi berbasis DTN [13] mengirimkan pesan dengan panjang yang berubah-ubah sehingga disebut *Application Data Units* (ADUs). ADU secara khusus dikirimkan dan terkirim ke aplikasi dalam unit yang lengkap. ADU diubah oleh *bundles* lapisan ke dalam satu atau lebih unit data protokol yang disebut *bundles*, yang diteruskan oleh *nodes* DTN. Sumber dan tujuan diidentifikasi oleh *endpoint identifiers* (EIDs) [14]. Suatu EID dapat mengacu pada satu atau lebih DTN *nodes*. Sedangkan IP jaringan berdasarkan pada operasi *store and forward*,

dengan asumsi bahwa tidak ada penyimpanan data dalam satuan waktu, karena antrian dan *delay* transmisi. Sebaliknya arsitektur DTN tidak selalu menganggap tersedianya jaringan dan mengatur bahwa node akan menyimpan *bundles* dalam selang waktu tertentu.

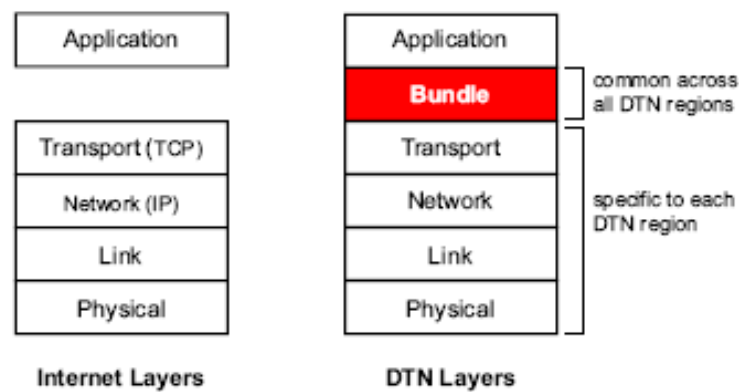
Bundles-based forwarding [5] memiliki elemen esensial dimana *bundles* akan menunggu dalam antrian hingga terdapat kesempatan komunikasi atau kontak pada jaringan. Hal ini didukung oleh beberapa asumsi yaitu :

1. media penyimpanan tersedia dan terdistribusi secara merata pada keseluruhan jaringan,
2. media penyimpanan mampu menyimpan *bundles* hingga terjadi pengiriman data pada node selanjutnya, secara implisit model *store and forward* merupakan pilihan yang lebih baik dibandingkan dengan usaha agar terdapat koneksi yang terus menerus.

Asumsi diatas perlu dipertimbangkan dan diperhitungkan agar dapat diterapkan secara efektif pada jaringan yang menunjang arsitektur DTN.

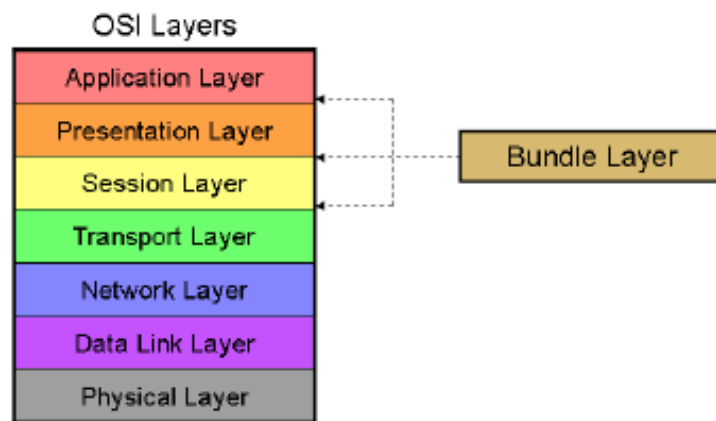
2.3 Bundles Protokol

Lapisan *bundles* (*bundle layer*) atau *bundle* protokol merupakan protokol utama yang digunakan dalam DTN. *Bundle layer* [15] terkait dengan spesifikasi region lapisan dibawahnya, sehingga memungkinkan program aplikasi dapat berkomunikasi dengan beberapa region lainnya. *Bundle layer* bertugas menyimpan dan meneruskan keseluruhan atau sebagian *bundles* diantara *nodes*. Suatu *bundle layer* protokol digunakan pada saat melintasi semua jaringan (region) dalam DTN. Sebaliknya, lapisan dibawah *bundle layer* (transport, network, *link*, physical lapisan), dipilih sesuai dengan lingkungan komunikasi pada setiap region. Gambar 2.2 menunjukkan letak *bundle layer* pada DTN dalam perbandingan dengan lapisan internet.



Gambar 2.2. Lapisan Internet dan Lapisan DTN [15]

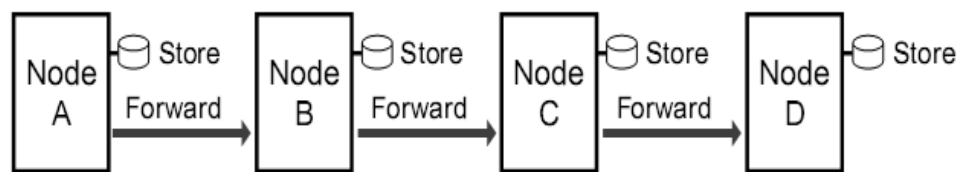
Bundles layer berada diantara lapisan transport dan application dalam lapisan OSI. Sebuah protokol *bundles* layer digunakan di seluruh jaringan yang tercakup dalam DTN [16]. Gambar 2.3 menunjukkan letak *bundles* lapisan pada lapisan OSI.



Gambar 2.3 Letak bundles layer pada lapisan OSI [16]

DTN menerapkan konsep *store and forward switching* [17] seperti yang digunakan sistem pos. Keseluruhan pesan atau pecahannya (*fragment*) dipindahkan dari tempat penyimpanan di sebuah titik (*node*) ke titik lain pada jalur antara sumber dan tujuan. Durasi penyimpanan pesan atau data pada setiap

node DTN, secara signifikan lebih lama dari durasi penyimpanan pada *router* TCP/IP. Jadi semua DTN *node* yang berada pada suatu jaringan, perlu mempunyai kapasitas penyimpanan yang cukup besar. Media yang dapat digunakan yaitu *hard disk* dan *flash memory* sebagai tempat penyimpanan sementara. Sehubungan dengan jaringan sementara (*link-intermittency*), DTN mendukung retransmisi *node* ke *node* dengan metode *store and forward* [18], seperti ditunjukkan pada Gambar 2.4.



Gambar 2.4. Metode *Store and Forward* Pada DTN [18]

Pesan yang disimpan (*stored*) dan diteruskan (*forwarded*) oleh setiap DTN *node*. Pesan atau data disebut *bundles*. *Bundles* merupakan unit dasar data berupa ukuran variable dan signal yang diperlukan untuk melintasi jaringan DTN. *Bundles* memuat tiga jenis informasi yaitu sumber-aplikasi data pengguna, kontrol informasi yang disediakan oleh sumber-aplikasi untuk aplikasi tujuan (informasi kontrol menjelaskan bagaimana menangani user data), dan sebuah *bundles header* yang terdapat pada *bundles layer*. *Bundles* [19] dikirimkan di antara *nodes* DTN yang ditandai dengan *round trip times* minimal, untuk menghindari terjadinya *link delay* [4].

Bundle router membutuhkan informasi yang detail tentang keadaan sistem yang akan mendasari keputusan tentang bagaimana *routing* akan dilakukan. Keputusan *routing* yang dibuat *bundle router* diwujudkan dalam seperangkat instruksi kepada *bundle forwarder* yang bertugas mengeksekusi instruksi tersebut. *Bundle forwarder* mengeksekusi instruksi dengan berinteraksi dengan *convergence layer*, *registrations store*, dan *persistent store*.

Persistent store digunakan untuk menyimpan *bundle* dalam waktu yang cukup lama selama proses simpan dan teruskan. Mode penyimpanan di *persistent store* dapat bervariasi mulai bentuk sebuah *file* sederhana hingga dalam bentuk sebuah basis data relasional.

Sebuah node DTN membutuhkan tempat penyimpanan yang permanen seperti harddisk untuk menyimpan data. Hal ini tentu berbeda dengan *router* biasa yang hanya memiliki memori jangka pendek untuk menyimpan data dalam jangka waktu beberapa milidetik. Tempat penyimpanan diperlukan karena :

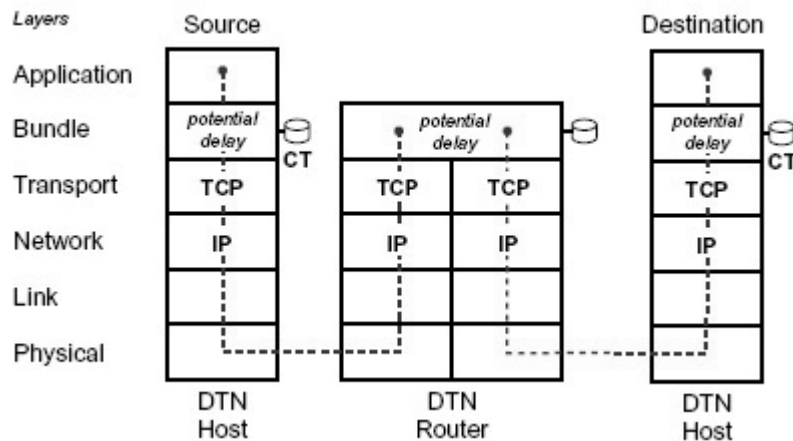
1. jalur komunikasi ke node berikutnya belum tentu tersedia,
2. kemampuan mengirim dan menerima data pada sebuah *node* terkadang berbeda dengan *node* lainnya,
3. data yang telah dikirim perlu dikirim ulang jika terdapat kesalahan dalam pengiriman.

Bundle terdiri dari tiga bagian, yaitu data dari lapisan aplikasi, informasi kontrol (yang diatur oleh aplikasi di sumber dan tujuan), dan *bundle header*. Saat pengiriman data, lapisan *bundle* dapat memecah keseluruhan *bundle* menjadi beberapa fragmen seperti halnya IP dapat memecah datagram menjadi fragmen. Jika *bundle* dipecah menjadi beberapa fragmen, lapisan *bundle* di tujuan akan menyusun ulang *bundle* tersebut.

Dalam DTN, sebuah node adalah sebuah perangkat yang memiliki lapisan *bundle*. Sebuah node dapat berupa sebuah *host*, *router*, atau *gateway* yang berperan sebagai pengirim atau penerima data. Sebuah *host* adalah perangkat yang menerima atau mengirim *bundle*, tetapi tidak meneruskannya. Sebuah *router* adalah perangkat yang bertugas untuk meneruskan *bundle*. *Router* dapat beroperasi dalam jaringan dengan waktu tunda yang lama. Jika tidak tersedia koneksi ke node berikutnya, *router* menyimpan sebuah *bundle* hingga koneksi tersedia. Sebuah *gateway* adalah perangkat yang bertugas untuk meneruskan *bundle* seperti halnya *router*.

Cara kerja DTN *routing* tidak jauh berbeda dengan Internet *routing*. Yang membedakan adalah lapisan-lapisan yang terlibat di dalamnya. Pada Internet *routing*, TCP/IP digunakan pada keseluruhan jaringan. IP beroperasi pada semua *node* dalam jaringan sedangkan TCP hanya beroperasi pada sumber dan tujuan.

Sebuah *router* Internet tidak memerlukan lapisan *transport*. Pada DTN *routing*, semua *node* menggunakan lapisan *transport* dan *bundle*. Mekanisme kerja *router* DTN ditunjukkan dalam Gambar 2.6



Gambar 2.5. DTN Routing [4].

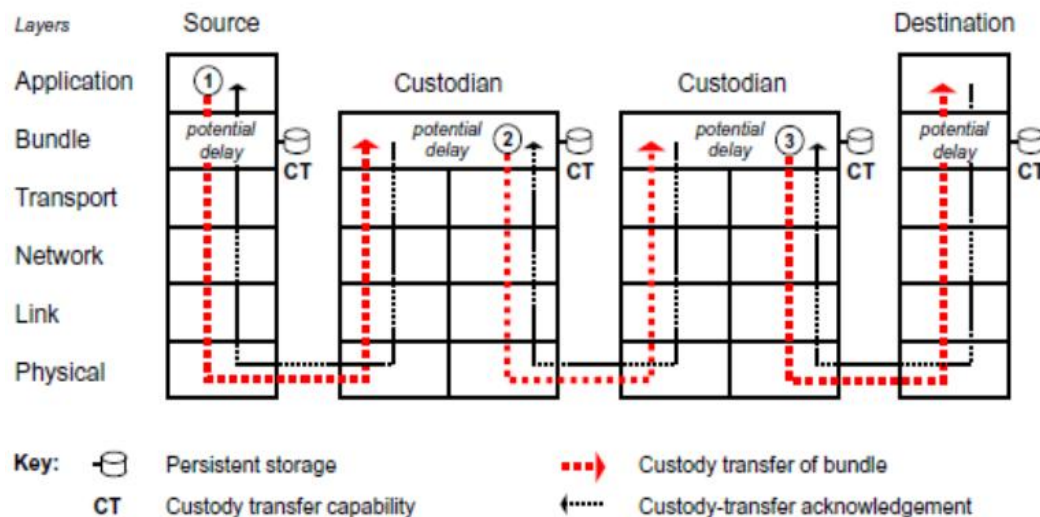
Sebuah *router* DTN dapat berupa *router* tetap maupun *router* bergerak (*mobile*). *Router* bergerak transit dari satu tempat ke tempat yang lain untuk meneruskan data. *Router* bergerak sering disebut juga kurir data. *Router* bergerak dapat memanfaatkan sepeda motor, mobil, pesawat terbang, kendaraan umum, hingga manusia.

2.4 Custody Transfer

DTN mendukung pengiriman kembali dari satu *node* ke *node* lain untuk data yang rusak dan hilang pada kedua lapisan *transport* dan *bundle* [20]. Akan tetapi karena tidak ada lapisan *transport* tunggal yang bekerja *end-to-end* pada DTN, maka realibilitas *end-to-end* hanya dapat diimplementasikan di lapisan *bundle*.

Pengiriman kembali pada DTN ini menggunakan *custody transfer*. Custody transfer atau “pengiriman data tahanan” ini dilakukan pada *node-node* yang telah berhasil menerima sebuah *bundle*. Sebuah *node* yang akan mengirim suatu *bundle*

akan meminta *node* berikutnya mengirim *bundle acknowledgment*. Apabila *node* selanjutnya telah siap menerima sebuah *bundle* maka *acknowledgment* akan dikirim ke *node* sebelumnya. Apabila waktu yang diberikan *node* sumber untuk menunggu *acknowledgment* dari *node* berikutnya telah melebihi batas *time-to-live* (waktu hidup) *bundle*, maka *bundle* akan dihapuskan dari *storage node* tersebut.



Gambar 2.6. Custody Transfer pada DTN [8]

Custody transfer dan *time-to-acknowledge retransmission timer* diperlukan ketika *current node custodian* mengirim *bundle* menuju *next node*. *Next node* akan mengirimkan *acknowledgment* kepada pengirim ketika *custody* diterima. Jika tidak ada *acknowledgment* yang dikirimkan ke pengirim, maka *bundle* akan dikirimkan ulang sebelum *time-to-acknowledge retransmission timer expired*. *Bundle custodian* akan menyimpan *bundle* sampai *node* lain menerima *custody* atau *bundle time-to-live expired*.

Custody transfer tidak menyediakan realibilitas *end-to-end*. Hal ini dapat dilakukan apabila sebuah sumber meminta custody transfer sekaligus dengan *return receipt*. *Return receipt* dikirim kepada *node* sebelumnya untuk memberitahu bahwa *bundle* telah diterima. Sehingga, *node* sumber harus memiliki salinan *bundle* tersebut hingga menerima *return receipt* dari *node* selanjutnya, dan

bundle akan dikirim kembali ketika *return receipt* tidak diterima dalam waktu yang ditentukan.

2.5 DTN2

DTN2 adalah nama perangkat lunak yang mengimplementasikan protokol DTN. DTN2 bertindak sebagai standar aplikasi DTN[4]. DTN2 juga berfungsi sebagai bahan eksperimen bagi peneliti di seluruh dunia untuk menguji apakah protokol DTN berjalan sesuai dengan standar yang telah ditetapkan. Meskipun DTN2 merupakan aplikasi standar dan dibuat untuk kepentingan penelitian, DTN2 juga merupakan perangkat lunak yang berkualitas tinggi, dan siap digunakan bagi aplikasi-aplikasi yang ingin memanfaatkan DTN. DTN2 telah diterapkan di India dan Kamboja sebagai bagian dari TIER *project*.

DTN2 mengimplementasikan protokol DTN sesuai spesifikasi yang tercantum pada RFC 5050. Sesuai RFC 5050, DTN mengirimkan data dalam bentuk *bundle* yang bisa saja berukuran lebih besar dari besar paket yang umum pada jaringan IP. DTN2 juga mendukung protokol *bundle security* untuk menyediakan fasilitas autentikasi pada *bundle* jika diperlukan oleh aplikasi. *Bundle* dapat dikirimkan melalui TCP/IP ataupun protokol lain. DTN2 juga menyediakan mekanisme *routing* sehingga *bundle* dapat dikirim baik secara langsung ke tujuan maupun melalui *router*.

DTN2 terdiri atas sebuah aplikasi utama dan beberapa aplikasi pendukung. Aplikasi utama DTN2 adalah *dtnd* yang berfungsi sebagai *daemon* DTN2. *dtnd* harus dijalankan selama penggunaan protokol DTN. Semua *bundle* yang dikirimkan maupun diterima aplikasi DTN harus melewati *dtnd*. *Bundle* yang dapat dikirimkan kepada *dtnd* melalui perantara sebuah file maupun hanya berupa parameter-parameter yang dikirimkan oleh aplikasi pendukung. Beberapa aplikasi pendukung DTN2 antara lain:

1. *dtnping*

dtnping merupakan aplikasi untuk menguji koneksi antara dua *node* DTN. *dtnping* dapat mengirim *bundle* ke *daemon* lain. Alamat *daemon* tersebut menjadi parameter saat memanggil *dtnping*.

2. Dtnsend dan dtnrecv

dtnsend dan *dtnrecv* merupakan aplikasi dasar pengirim dan penerima bundle dalam DTN2. *dtnsend* dan *dtnrecv* dapat digunakan untuk sekedar mengirim tanggal, pesan, hingga mengirim file.

3. Dtncp dan dtncpd

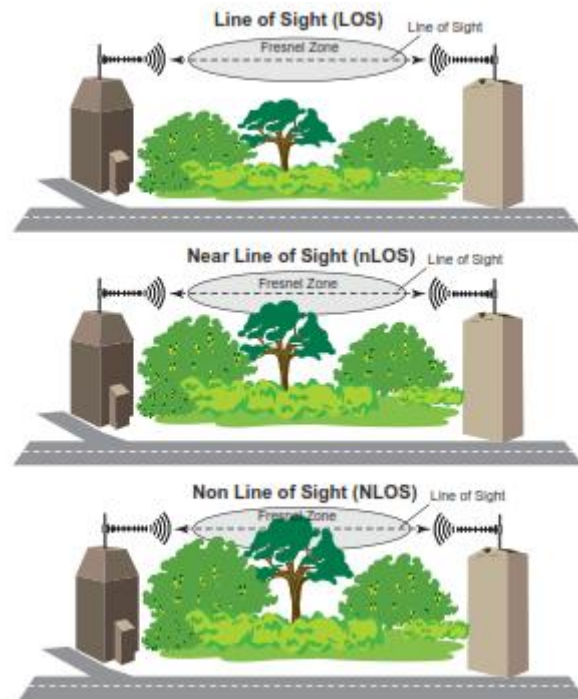
dtncp dan *dtncpd* merupakan aplikasi yang berfungsi untuk melakukan transfer file. *dtncp* mengirim file ke daemon lain. *dtncpd* menunggu file yang dikirim *dtncp* di tujuan. *dtncpd* akan meletakkan file yang diterima pada lokasi yang dimasukkan sebagai parameter saat *dtncpd* dipanggil.

4. Dtnperf

dtnperf merupakan aplikasi yang berfungsi untuk menguji performa dari jaringan DTN. *dtnperf* melaporkan status dari *bundle* yang dikirimkan dari *client* ke *server*. *dtnperf* juga dapat digunakan untuk mengukur *round trip times* (RTT).

2.6. Propagasi Gelombang

Propagasi gelombang adalah transmisi atau penyebaran sinyal dari suatu tempat ke tempat lain. Media perambatan atau saluran transmisi gelombang dapat berupa fisik maupun non fisik (nirkabel). Saluran transmisi gelombang fisik yaitu berupa kabel koaksial dan sepasang kawat konduktor, sedangkan untuk non fisik berupa gelombang radio dari sistem komunikasi yang biasanya digambarkan dengan kondisi LoS (*Line of sight*), N-LoS (*Non Line of sight*), dan n-LoS (*Near Line of sight*).



Gambar 2.7. Perbedaan LoS, n-LoS, dan N-LoS [Global Connectivity]

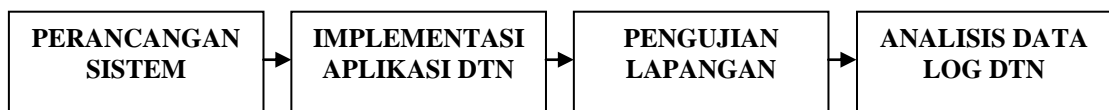
Kondisi LoS (*Line of Sight*) artinya suatu kondisi di mana pemancar dapat dilihat secara jelas tanpa halangan sebuah penerima. Tipe koneksi ini biasanya lebih stabil dan dapat mengirimkan data yang cukup banyak dengan sedikit kesalahan. Selanjutnya n-LoS (*near Line of Sight*) adalah kondisi dimana pandangan sedikit terhalang atau mencakup penghalang parsial seperti puncak pohon, perumahan, dan lain-lain. Sedangkan kondisi N-LoS (*non Line of Sight*) terdapat penghalang pada lintasan sinyalnya. Penghalang dapat bersifat buatan seperti gedung bertingkat maupun pegunungan ataupun perbukitan.

BAB III

METODOLOGI PENELITIAN

3.1. Pendahuluan

Bab ini berisi penjelasan mengenai metodologi penelitian tugas akhir. Metodologi penelitian ini melalui beberapa tahapan yang direpresentasikan dalam suatu kerangka kerja (*framework*). Tahapan awal yang akan dilakukan adalah menginstal paket-paket yang menunjang aplikasi DTN. DTN berjalan di sistem operasi linux. Ketiga device yang berperan sebagai *server*, *router*, dan klien diinstal aplikasi *DTN2* dan *Oasys* beserta paket-paket lain yang dibutuhkan DTN. Kemudian melakukan konfigurasi `dtm.conf` untuk *routing* DTN nya. Setelah aplikasi DTN telah berjalan, pengambilan data dilakukan dalam keadaan LoS (*Line of Sight*) dan N-LoS (*non Line of Sight*) dengan 4 kali percobaan dengan jarak masing-masing adalah 15 meter, 30 meter, 50 meter, dan 70 meter dan variasi ukuran data yang diuji yaitu 5 MB, 50 MB, 200 MB, 500 MB, 750 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Setelah percobaan selesai, maka data akan dianalisa berdasarkan log yang tersimpan di DTN. Berikut adalah gambar konsep penelitian :



Gambar 3.1. Konsep Penelitian

3.2. Kerangka Kerja Penelitian (*Framework*)

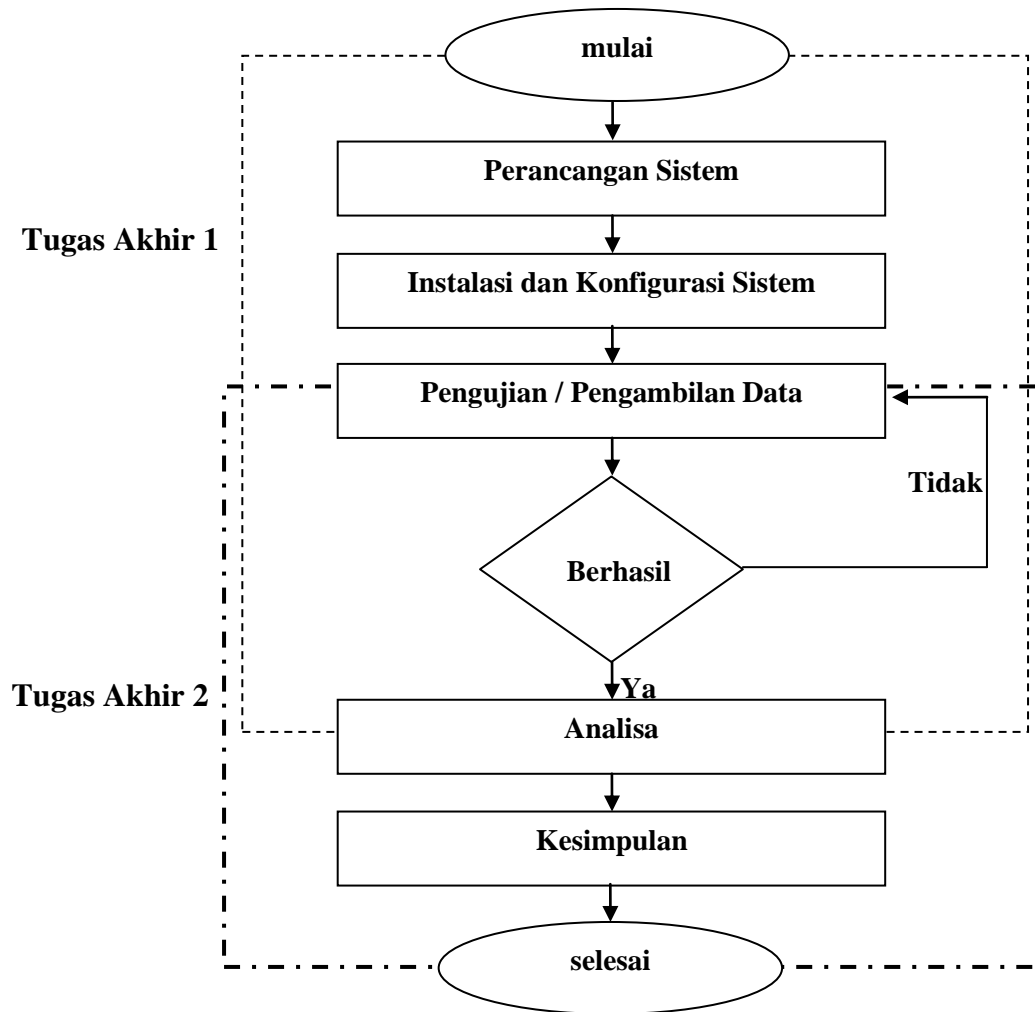
Proses percobaan dalam penelitian ini melalui beberapa tahapan. Tahapan yang dilakukan mengikuti kerangka kerja (*Framework*) yang telah dirancang

sehingga proses pengerjaan tugas akhir ini terstruktur dan mengikuti alur penelitian yang telah dirancang.

Tahap pertama adalah perancangan topologi yang akan digunakan dalam penelitian DTN ini sehingga nantinya akan diketahui apa saja perangkat yang dibutuhkan untuk perancangan dalam sistem DTN. Kemudian persiapan kebutuhan *hardware* (perangkat keras), dalam hal ini digunakan 3 laptop yang berperan sebagai server, router, dan klien, serta *AP (access point)* dan *usb wireless*. Setelah kebutuhan perangkat keras terpenuhi, selanjutnya adalah mempersiapkan kebutuhan perangkat lunak yang menunjang DTN. Yang pertama dilakukan adalah instalasi sistem operasi setiap device yang digunakan ke OS Linux. Kemudian dilakukan instalasi aplikasi *DTN2* dan *oasys* serta paket-paket yang mendukung untuk jalannya aplikasi DTN tersebut. Setelah semua alat yang digunakan telah siap, langkah selanjutnya adalah perancangan sistem. Dalam perancangan sistem yang meliputi konfigurasi *dtn.conf*, *DTN routing*, serta *routing DHCP*. Setelah itu dilakukan integrasi antara perangkat keras dan perangkat lunak sehingga aplikasi DTN dapat berjalan semestinya dan dapat dilakukan percobaan lanjut.

Dalam penelitian ini digunakan parameter jarak 15-70 meter dengan kondisi LoS (*line of Sight*) dan N-LoS (*non Line of Sight*), maka yang dilakukan selanjutnya adalah survei lapangan yang nantinya memungkinkan untuk digunakan sebagai lokasi penelitian atau pengambilan data. Setelah lokasi telah didapatkan, maka penelitian dan pengambilan data bisa dilakukan.

Penelitian dilakukan dengan pengujian atau pengambilan data sesuai skenario DTN dengan memperhatikan parameter-parameter yang nantinya akan dianalisis pada BAB IV. Setelah semua data dianalisa, maka akan ditarik kesimpulan yang dibahas pada bab V. Adapun bentuk *framework* yang telah dibuat dapat dilihat pada gambar 3.2.



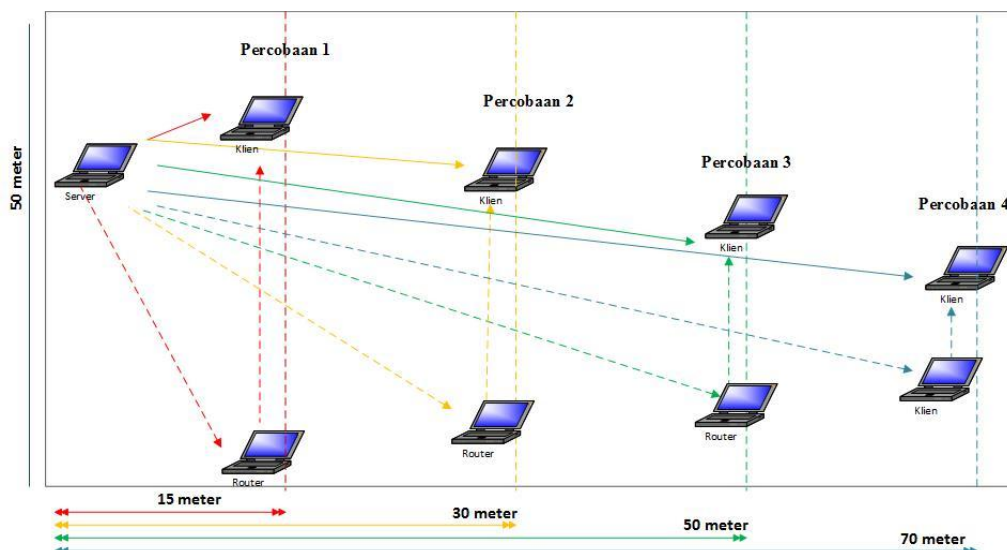
Gambar 3.2. Kerangka Kerja Penelitian

3.3. Perancangan Sistem

Penelitian dilakukan dalam 2 kondisi, yaitu Los (*Line of Sight*) dan N-LoS (*non Line of Sight*). Perancangan sistem dibangun dengan menggunakan 3 device laptop, 1 *Access Point* (AP), dan *usb wireless portable*. Pertimbangan digunakan laptop bukan PC dikarenakan dalam penelitian ini dilakukan dalam keadaan *line of sight* (LoS) atau *outdoor* dengan memanfaatkan lapangan yang sisinya mencapai 70 meter. Untuk mencapai konsep *router bergerak*, maka dalam pergerakan *router* ketika menjemput data dari server serta memindahkannya ke

klien memanfaatkan manusia yang membawa router tersebut. Ketika pengambilan data, semua device telah terkoneksi dengan jaringan *wifi*. Jaringan *Wifi* memiliki karakteristik penerapan yang fleksibel dan tidak memerlukan koneksi perkabelan yang rumit sehingga memudahkan proses instalasi perangkat.

Router adalah penghubung antara dua jaringan yang terpisah. Dalam jaringan DTN juga dapat memfungsikan salah satu node sebagai router untuk menghubungkan dua jaringan yang terpisah (tidak ada koneksi *end-to-end*). Konsep router DTN ini seperti kurir data yang menyampaikan data dari satu tempat ke tempat lain yang terpisah. Analoginya, *router* DTN dapat berperan sebagai petugas pos yang mengantarkan surat dari satu alamat (rumah) ke alamat yang lain. Pada skenario jaringan DTN ini, terdapat dua buah jaringan yang terpisah yaitu yang berperan sebagai server dan klien. Laptop server akan mengirim data yang ditujukan ke klien (*destination*). Antara laptop *server* dan *klien* tidak terdapat koneksi apapun, tetapi terdapat sebuah laptop *router* DTN yang akan dibawa (oleh manusia) dan bergerak mendekati server dan klien. Jaringan DTN memungkinkan sebuah *router* DTN dapat menjadi perantara server dan klien agar kedua jaringan tersebut berkomunikasi meskipun tidak memiliki koneksi *end-to-end*. Skenario penelitian yang digunakan pada penelitian ini dapat dilihat pada gambar 3.3.



Gambar 3.3. Skenario Penelitian Sistem DTN

3.4. Spesifikasi Data

Data/*file* yang dikirim pada saat pengujian sistem DTN ini meliputi data yang berukuran mulai dari 5 MB sampai dengan 2000 MB. Ukuran data uji dapat berbentuk format *.mp3* (audio) maupun *.mkv*, *mp4*, dan *.avi* (video). Adapun format setiap data uji dari pengujian sistem DTN ini dapat dilihat pada tabel 1 sebagai berikut

Tabel 1
Spesifikasi Laptop

Ukuran Data (Size)	Format	Tipe Data	Bit Rate	Frame Rate
5 MB	.mp3	audio	192 kbps	-
50 MB	.mkv	video	N/A	30 frames per second
200 MB	.mp4	video	399 kbps	30 frames per second
500 MB	.mkv	video	N/A	25 frames per second
750 MB	.avi	video	N/A	24 frames per second
1000 MB	.mkv	video	N/A	24 frames per second
1200 MB	.avi	video	N/A	30 frames per second
1500 MB	.avi	video	N/A	24 frames per second
1700 MB	.mkv	video	N/A	24 frames per second
2000 MB	.avi	video	N/A	30 frames per second

3.5. Spesifikasi Perangkat

Dalam penelitian ini digunakan perangkat *hardware* (perangkat keras) dan juga perangkat *software* (perangkat lunak) yang mendukung skenario penelitian jaringan DTN ini.

3.5.1. Perangkat Hardware

Perangkat *Hardware* (perangkat keras) yang digunakan dalam penelitian ini sebagai berikut:

1. 3 unit Laptop yang berperan sebagai *server*, *router*, dan klien dengan spesifikasi yang dapat dilihat pada Tabel 2

Tabel 2
Spesifikasi Laptop

Peran	Spesifikasi	Sistem Operasi
Server	Intel [®] Core i5, 2.4 GHz, 2 GB RAM	Linux, Ubuntu 15.04 LTS
Router	Intel [®] Core i3, 2.3 GHz, 2 GB RAM	Linux, Ubuntu 15.04 LTS
Klien	Intel [®] Atom [™] , 1.8 GHz, 2 GB RAM	Linux, Ubuntu 15.04 LTS

2. 1 unit *wifi Access Point* WRT54G, 802.11 B/G dengan kecepatan 54 Mbps.
3. 1 unit *wireless USB adapter* TL-WN722N, dengan 4dBi antenna eksternal dan kecepatan 150 Mbps.

Device yang berperan sebagai *server*, *router* maupun klien di instal dengan sistem operasi *Linux*. Dalam penelitian ini digunakan *Linux Ubuntu* versi 15.04 LTS dengan spesifikasi laptop seperti pada Tabel 1. Setelah instalasi sistem operasi, perlu dilakukan konfigurasi interfaces pada setiap PC masing-masing untuk menentukan alamat IP nya. Untuk mengatur dan mengubah IP *interfaces*nya, dapat membuka direktori */etc/network/interfaces*. Adapun penamaan hostname, user, serta pengaturan IP *interfaces* pada setiap masing-masing PC dapat dilihat pada tabel 3.

Tabel 3

Penamaan *hostname*, *user*, dan *Pengaturan IP Address*

Peran	Hostname	User	<i>IP address</i>
Server	fitri	Fitri	192.168.10.2
Router	putri	Putri	192.168.10.1
Klien	klien	Klien	192.168.20.12

3.5.2. Perangkat *Software*

Sistem DTN ini berjalan di sistem operasi *Linux Ubuntu* 15.04 LTS. Adapun Perangkat *Software* (perangkat lunak) yang digunakan dalam sistem DTN ini adalah aplikasi DTN2 dan Oasys. Aplikasi DTN2 merupakan aplikasi inti dari sistem layanan berbasis DTN yang berfungsi mengatur mekanisme pengiriman, penerimaan dan penyimpanan data yang disimpan dalam bentuk *log* DTN menggunakan protokol DTN. Di bagian pengirim Aplikasi DTN2 akan mengubah data-data yang akan dikirim menjadi *bundle* dan mengubahnya kembali menjadi bentuk data semula di bagian penerima. Perangkat lunak yang digunakan sebagai aplikasi DTN2 dalam penelitian ini adalah *dtm-2.8.9* dan komponen pendukung *oasys-1.5.0* sebagai platform untuk program DTN2. Untuk dapat mendukung penggunaan perangkat lunak DTN2, diperlukan instalasi paket-paket aplikasi pendukung yang harus tersedia di sistem linux ubuntu. Adapun paket-paket aplikasi tersebut adalah sebagai berikut:

- *build-essential*
- *libavahi-compat-libdnssd-dev*
- *libdb5.3*
- *libdb5.3-dev*
- *libxerces-c28*
- *libxerces-c3-dev*
- *libexpat1-dev*
- *mono-**

- *tcl8.4*
- *tcl8.4-dev*
- *tcllib*
- *tclreadline*
- *tclx8.4-dev*
- *tclx8.4*
- *zlib1g-dev*

3.6. Konfigurasi Sistem DTN

3.6.1. Konfigurasi Routing DTN

Sistem DTN ini berjalan sesuai dengan pengaturan konfigurasi routing yang telah dilakukan di `dtm.conf` pada masing-masing node secara statis. Konfigurasi `dtm.conf` diatur pada setiap laptop yang berperan sebagai server, router, maupun klien. Untuk merubah pengaturan konfigurasi `dtm.conf` dapat dilakukan dengan membuka direktori tempat `dtm.conf` tersebut disimpan. Pada sistem DTN penelitian ini, direktori `dtm` dapat ditemukan dengan perintah `/etc/dtm.conf` dan untuk merubah pengaturan `dtm.conf` sesuai skenario penelitian ini digunakan perintah `nano /etc/dtm.conf`. Adapun beberapa parameter yang harus diperhatikan dalam merubah konfigurasi dalam `dtm.conf` yaitu sebagai berikut

1. *route local_eid "dtm://[info hostname]"*

Penamaan parameter *[info hostname]* disesuaikan dengan *hostname* diikuti dengan nama *domain* dari masing-masing komputer server, misalnya *local_eid* untuk komputer server adalah *route local_eid "dtm://server.dtm"*

2. *link add <name> <nexthop> <type> <clayer>*

Parameter *<name>* diisi dengan nama link, misalnya *link1*. Parameter *<nexthop>* diisi dengan alamat IP atau nama *domain* dari *host* yang akan terhubung secara langsung dengan *host* pengirim. Parameter *<type>* dalam implementasi ini menggunakan tipe koneksi *ONDEMAND*. Parameter

<clayer> dalam implementasi menggunakan *tcp*. Contoh penulisan atau pengaturan parameter *link add* yaitu :

```
link add link1 192.168.10.1 ONDEMAND tcp
```

3. *route add <dest> <link/peer>*

Parameter *<dest>* diisikan dengan *alamat host* yang menjadi tujuan pengiriman data *bundle* dari suatu komputer. Parameter *<link/peer>* diisikan dengan nama link tujuan. Contoh penulisan atau pengaturan parameter *route add* secara lengkap adalah sebagai berikut:

```
route add dtn://router.dtn/* link1.
```

Berdasarkan beberapa parameter tersebut, adapun konfigurasi untuk routing *dtm.conf* pada setiap laptop yang berperan sebagai server, router, maupun klien adalah sebagai berikut:

- Konfigurasi *dtm.conf* pada Server

```
storage set dbname DTN
Route set type static
route local_eid "dtm://fitri.dtm"
link add link1 192.168.10.1 ONDEMAND tcp
route add dtm://putri.dtm/* link1
route add dtm://klien.dtm/* link1
```

- Konfigurasi *dtm.conf* pada Router

```
storage set dbname DTN
Route set type static
route local_eid "dtm://putri.dtm"
link add link1 192.168.10.2 ONDEMAND tcp
link add link1 192.168.20.12 ONDEMAND tcp
route add dtm://server.dtm/* link1
route add dtm://klien.dtm/* link2
```

- Konfigurasi *dtm.conf* pada Klien

```
storage set dbname DTN
Route set type static
```

```

route local_eid "dtn://klien.dtn"
link add link1 192.168.10.1 ONDEMAND tcp
route add dtn://router.dtn/* link1
route add dtn://server.dtn/* link1

```

3.6.2. Konfigurasi *Interface*

Konfigurasi *interface* dilakukan untuk mengatur IP address pada setiap perangkat untuk konfigurasi jaringan dalam sistem DTN tersebut. Konfigurasi interface ini diatur pada server dan router sesuai dengan alamat IP dari kartu jaringan secara statis. Berikut pengaturan konfigurasi interface pada setiap perangkat server dan router.

- pada server

```

auto wlan0
iface wlan0 inet static
    address 192.168.10.2
    subnetmask 255.255.255.0
    wireless-channel 1
    wireless-essid MYNETWORK
    wireless-mode ad-hoc

```

- pada router

```

auto wlan0
iface wlan0 inet static
    address 192.168.10.1
    subnetmask 255.255.255.0
    wireless-channel 1
    wireless-essid MYNETWORK
    wireless-mode ad-hoc

```

```

auto wlan1
iface wlan1 inet static

```

```

address 192.168.20.1
subnetmask 255.255.255.0
wireless-channel 1

```

3.6.3. Konfigurasi Routing DHCP

DHCP (*Dinamic Host Configuration Protocol*) merupakan metode protocol berbasis client/server yang digunakan untuk memudahkan pengalokasian IP address dalam suatu jaringan. Komputer yang menjadi server DHCP disini adalah Router. DHCP disini digunakan untuk menyebarkan IP ke perangkat klien yang telah terhubung dalam sistem DTN sehingga ketika klien aktif akan otomatis mendapatkan IP address dari server DHCP. Pengaturan konfigurasi *dhcp* ini dapat diatur pada direktori `/etc/dhcp/dhcpd.conf`. Adapun konfigurasi DHCP dilakukan pada perangkat router dengan pengaturan sebagai berikut:

```

default-lase-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.20.255;
option routers 192.168.20.1;
option domain-name-servers 192.168.20.1;
option domain-name "mydomain.example";

subnet 192.168.20.0 netmask 255.255.255.0 {
range 192.168.20.10 192.168.20.100;
}

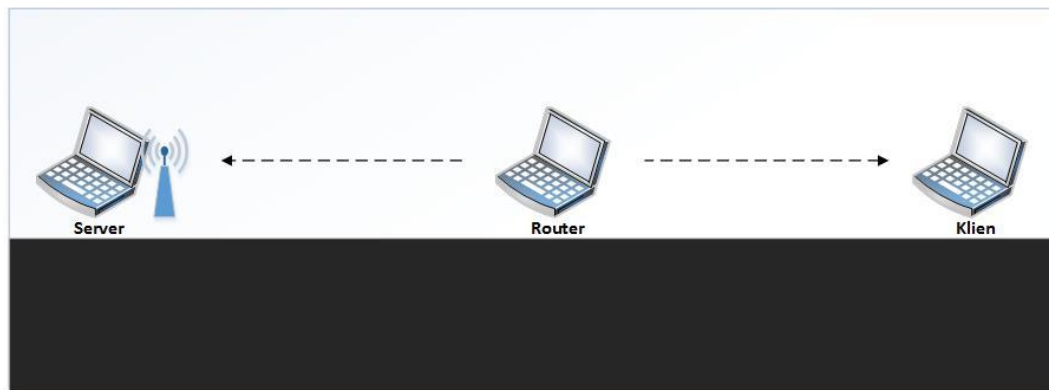
host klien {
    hardware ethernet 44:6D:57:A9:DE:5B;
    fixed-address 192.168.20.12;
}

```

3.7. Skenario Penelitian

Pada penelitian jaringan DTN ini dilakukan dengan implementasi arsitektur jaringan DTN dengan komponen perangkat keras dan perangkat lunak yang telah dijelaskan diatas. Pengujian dengan membuat pemodelan sistem, orang dimodelkan sebagai pembawa router yang menjadi perantara pengiriman data antara server dan klien menggunakan DTN seperti yang terlihat pada gambar 3.4. Lokasi pengujian/pengambilan data dilakukan di lapangan *LoS* (*Line of Sight*) dengan pengujian jarak 15 dan 30 meter bertempat di lapangan basket Perumahan Kedamaian Permai sedangkan untuk pengujian jarak 50 dan 70 meter bertempat di lapangan Sekolah SD dan SMP Pupuk Sriwijaya Palembang (PT.PUSRI). Sedangkan untuk kondisi N-LoS (*non Line of Sight*) berlokasi di Perumahan Kedamaian Permai.

Pengujian dilakukan dengan 4 kali percobaan pada jarak masing-masing adalah 15 meter, 30 meter, 50 meter dan 70 meter dan variasi ukuran data yang diuji yaitu 5 MB, 50 MB, 200 MB, 500 MB, 750 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB.



Gambar 3.4. Skenario Penelitian DTN dalam Kondisi *LoS*

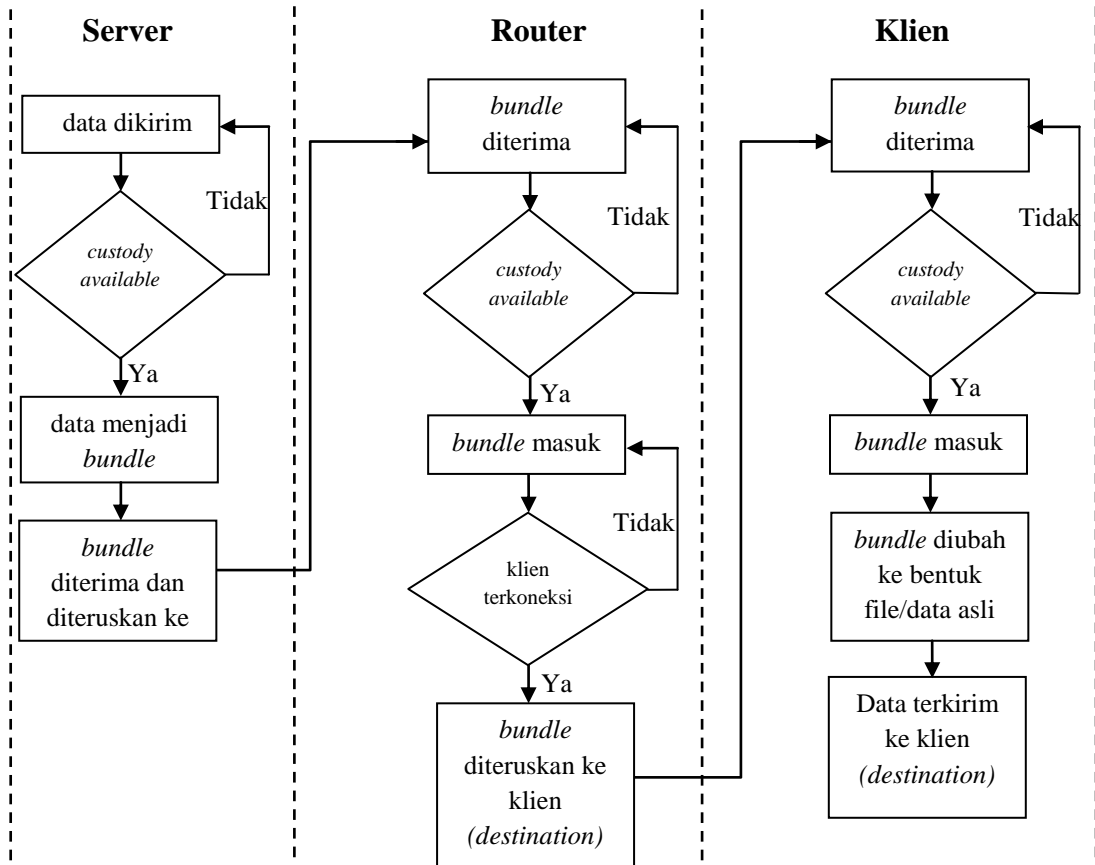


Gambar 3.5. Skenario Penelitian DTN dalam Kondisi N-LoS

3.8. Mekanisme Pengiriman Data

Konsep DTN ini memungkinkan dalam pengiriman data dari satu node ke node lainnya yang memiliki keadaan koneksi yang terputus-putus (*intermittent*). Dalam penelitian ini mengkondisikan ketika server mengirimkan file/data ke klien (*destination*), *bundle* tersebut akan *transit* dulu ke *router* dengan cara laptop *router* yang dibawa oleh manusia akan mendekati server untuk menjemput file yang dikirim *server*. Setelah file tersebut sampai di *router*, *wireless router* bergerak akan mendekati lokasi klien, apabila koneksi ke klien sudah terdeteksi dengan *wifi* otomatis di perangkat *router*, maka *bundle* yang tersimpan di *router* tersebut akan otomatis diteruskan ke klien (*destination*). Jika klien tidak terkoneksi/terjangkau dengan perangkat *wifi* otomatis, maka *bundle* tersebut akan tetap tersimpan di *router* dengan waktu tertentu. Ketika *bundle* tersebut terkirim ke klien, maka *bundle* tersebut akan diubah menjadi *file* data sesuai dengan data yang dikirim dan masuk ke *direktori* komputer klien yang telah ditentukan sebelumnya. Dalam pengiriman data ini *custody transfer* sangat berperan penting dalam pengiriman *bundle* protocol pada setiap node yang dilalui dari pengirim sampai data yang dikirim tiba di tujuan. *Custody* disini bertanggung jawab untuk *custody transfer* bertanggung jawab untuk memberikan konfirmasi pengiriman suatu pesan dari satu hop ke hop yang lain. *Custody transfer* digunakan untuk

mengurangi tingkat kemungkinan data yang dikirim hilang/tidak terkirim dalam setiap hop nya. Dengan kata lain *custody transfer* ini bertanggung jawab untuk pemeliharaan koneksi *end-to-end*. Diagram alir dari pengiriman data dari server ke klien (*destination*) dapat dilihat pada gambar 3.5.



Gambar 3.6. Diagram Alir Pengiriman Data DTN

Dalam pengiriman *file/data*, harus memperhatikan *IP address* yang telah ditentukan sebelumnya. Untuk Pengalamatan IP pada setiap perangkat DTN dapat dilihat pada tabel 4.

Tabel 4
IP address Perangkat DTN

Perangkat	Kartu Jaringan	IP address	Subnet Mask
Server	wlan0	192.168.10.2	255.255.255.0
Router	wlan0	192.168.10.1	255.255.255.0
Klien	wlan1	192.168.20.12	255.255.255.0
Access Point (AP)	wlan1	192.168.20.2	255.255.255.0
USB wireless Portable	wlan1	192.168.20.11	255.255.255.0

3.8. Mekanisme Pengambilan Data

Dalam pengambilan data, hal yang pertama kali diperhatikan adalah sistem dari skenario DTN ini sudah berjalan dengan semestinya. Semua *device* dalam perancangan DTN ini sudah dapat berkomunikasi/terkonfigurasi menjadi sistem DTN. Pengambilan data ini memperhatikan setiap *device* yang terhubung. Untuk penyimpanan *log bundle* ini dilakukan di ketiga *device* yang berperan sebagai server, router, maupun klien.

Hal yang dilakukan pertama kali dilakukan dalam pengujian untuk pengambilan data ini adalah mengaktifkan *access point (AP)* dan *USB wireless adapter* yang telah di *setting* sebelumnya. Untuk melihat apakah semua *device* telah terhubung/terkonfigurasi antar satu dengan yang lain dapat digunakan perintah *dtnping*, dengan contoh

```
dtnping -c 6 dtn://klien.dtn
```

Adapun hal-hal yang harus dilakukan pada setiap device DTN ini adalah sebagai berikut

- **Pada Server**

Server disini bertugas untuk mengirimkan data ke tujuan yaitu klien. Langkah-langkah yang dilakukan pada *device server* adalah

1. Pada terminal pertama (T1) yaitu mengaktifkan *daemon* dan menyimpan *log bundle* untuk setiap data yang dikirim dengan perintah:

```
dtnd -t -c etc dtn.conf -l info -o
/home/fitri/dtn/log/521koma6MB15m.log
```

Pada pengujian ini data yang dikirim berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 750 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB dengan 3 kali percobaan dengan jarak 15 meter, 30 meter, 50 meter, dan 70 meter.

2. Di terminal lainnya (T2), server mengirimkan data ke tujuan (*klien*) dengan perintah:

```
dtcp -expiration 3600
/home/fitri/Documents/KIRIMDATA/521koma6MB.mkv
dtn://klien.dtn
```

3. Pada terminal 1, akan dilakukan pengamatan mengenai status pengiriman data yang telah dilakukan dengan perintah:

```
bundle list
```

maka, apabila status *bundle list* menunjukkan

```
Currently Pending Bundles (521643373):
```

berarti bundle yang dikirim masih dalam proses pengiriman ke node selanjutnya. Sedangkan apabila status *bundle list* nya ketika di cek kembali dengan mengetik *bundle list* namun yang terlihat pesan

```
Currently Pending Bundles (0):
```

maka bundle yang dikirim sudah sampai ke node selanjutnya (router).

▪ Pada Router

Router disini berperan penting dalam skenario penelitian DTN ini. Di router inilah *bundle* dari data yang dikirimkan dapat disimpan sampai koneksi ke node tujuan telah tersedia. Hal-hal yang harus dilakukan pada device router adalah sebagai berikut:

1. Hal yang harus dilakukan pertama kali pada *router* yaitu dengan mengaktifkan *service DHCP* di terminal pertama. Cara mengaktifkan *service DHCP* yang telah diatur sebelumnya adalah:

```
service isc-dhcp-server start
service network-manager restart
/etc/init.d/networking restart
```

2. Selain *service DHCP*, dirouter juga menggunakan skrip *wifi otomatis* yaitu *autoconnect.sh*. Program ini digunakan untuk memastikan klien sudah terkoneksi dengan *access point* (AP). Untuk mengaktifkan *wifi otomatis* ini digunakan perintah `sh autoconnect.sh`.
3. Mengaktifkan *service daemon* dan penyimpanan *log bundle* otomatis seperti yang dilakukan pada server.
4. Memonitoring *bundle list* seperti yang dilakukan server dengan tujuan melihat berapa jumlah antian *bundle* yang sedang menuju ke router. Data yang dikirim ke klien oleh *server* akan otomatis diteruskan ke *router* ini sebelum akhirnya akan diteruskan lagi ke node selanjutnya (klien). Dalam *bundle list* akan terlihat juga ukuran data (*bytes*) dalam setiap data pengirimannya. *Bundle* tersebut akan terus berkurang hingga pada akhirnya *bundle* pada router akan tersisa 0. Jika *bundle* pada router sudah selesai dikirim, maka *bundle* tersebut akan diteruskan ke *node* selanjutnya (klien).

▪ Pada Klien

Pada prinsipnya, *klien* disini hanya menerima data yang dikirimkan oleh server. Hal yang harus dilakukan pada klien adalah

1. Mengaktifkan *service daemon* DTN dan penyimpanan *log bundle* otomatis seperti yang dilakukan pada *server* maupun *router*.
2. Dari *monitoring bundle list* terlihat `Currently Pending Bundles (521643373)` :
berarti *bundle* yang diteruskan oleh *router* masih dalam proses pengiriman. Sedangkan apabila status *bundle list* nya ketika di cek kembali dengan mengetik `bundle list` namun yang terlihat pesan, `Currently Pending Bundles (0)` :
berarti *bundle* sudah sampai di klien.
3. Untuk melihat apakah data sudah benar-benar sampai di klien:
`dtncpd /home/klien`

BAB IV

PENGUJIAN DAN ANALISA

4.1. Pendahuluan

Pada bab ini akan dilakukan pengujian perangkat lunak dan analisis terhadap hasil pengujian. Analisa dilakukan untuk melihat apakah hasil yang didapat sesuai dengan kinerja yang diharapkan.

4.2. Pengujian dan Analisa

Pada pengujian ini menggunakan analisis metode *custody transfer* yang merupakan metode pengiriman data yang mengandalkan hop by hop pada setiap nodenya. Dalam tugas akhir ini, objek analisis dilakukan berdasarkan *file log* yang dihasilkan oleh perangkat lunak DTN2 dan disimpan menggunakan skrip penyalinan *log* otomatis. Adapun yang akan dibahas dalam pengujian jaringan DTN ini yaitu, sebagai berikut.

1. Waktu yang dibutuhkan untuk mentransfer data dari server ke klien melalui router sehingga akan didapatkan waktu rata-rata pengiriman dengan menggunakan jaringan DTN.
2. Mengenal istilah-istilah dalam *log bundle* yang didapat dari pengiriman data pada saat pengujian.
3. Ukuran data maksimum yang memungkinkan untuk data dapat terkirim dari server ke klien pada setiap jarak pengujian.

Pengambilan data dilakukan dalam 2 kondisi yaitu kondisi LoS (*Line of Sight*) dan N-LoS (*non Line of Sight*) dengan 4 kali percobaan dengan jarak masing-masing adalah 15 meter, 30 meter, 50 meter dan 70 meter dan variasi ukuran data yang diuji yaitu 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Setelah percobaan selesai, maka data akan dianalisa berdasarkan *log* yang tersimpan di DTN.

Ukuran data pengujian yang terdapat pada tabel bukanlah ukuran sebenarnya. Hal tersebut dilakukan untuk memudahkan dalam mengingat skala yang digunakan dalam pengiriman data tersebut. Hal ini juga disebabkan karna sulit mencari data yang benar-benar dalam skala data yang telah ditentukan, sehingga data yang digunakan berdasarkan ukuran data yang mendekati dari skala data pengujian. Adapun data sebenarnya yang dikirim dalam satuan *bytes* dapat dilihat pada tabel 5.

Tabel 5

Ukuran Data Uji dalam *Bytes*

Ukuran Data (dalam Skala)	Ukuran Data Sebenarnya (dalam Bytes)
5 MB	5259105 bytes
50 MB	51097785 bytes
200 MB	209696318 bytes
500 MB	521643214 bytes
750 MB	734322688 bytes
1000 MB	1048097165 bytes
1200 MB	1366835824 bytes
1500 MB	1458949864 bytes
1700 MB	1721254389 bytes
2000 MB	2045884082 bytes

Dalam analisa waktu yang dibutuhkan untuk *mentransfer* data dari *server* ke *router* ini digunakan parameter *log bundle* pada *server* dan *log bundle* klien. Pada saat data dikirim oleh *server*, maka dalam *log* tersebut akan terlihat waktu (jam) pengiriman data dimulai dan pada saat data diterima atau sampai di *klien*

juga ada *log* data yang menunjukkan waktu (jam) data yang dikirim tersebut sampai atau diterima oleh klien.

4.2.1. Pengujian dan Analisa Waktu Transfer Data

Pengujian hasil *log bundle* pada setiap *server* dan klien dapat dilihat dengan parameter waktu pengiriman (*t* pengiriman) data pada server (jam saat data dikirim) dan waktu diterima (*t* diterima) pada *log* data di klien. Contoh *log* data di server (*t* pengiriman) dan di klien (*t* diterima) pada pengiriman data 500 MB dalam jarak 15 meter dapat dilihat pada gambar 4.1 dan 4.2.

```
rfc3339,time_t.usec,node,event,size,src,id,dest,expiry,flags,peer
,ob-src,ob-id,otime.ousec,conttype,duration,cmt
[1451221275.923784 /s10 info] 2015-12-27
13:01:15,1451221275.923687,dtn://fitri.dtn,STARTING,0,,,,,,,,,0.0
00000,,0,
[1451221275.924697 /dtn/bundle/daemon notice] loading bundles
from data store
```

Gambar 4.1. *t* pengiriman pada log server

```
[1451222281.057028 /s10 info] 2015-12-27
13:25:53,1451222281.056981,dtn://klien.dtn,RX,521643214,dtn://fit
ri.dtn/dtncp/send?
source=/home/fitri/Documents/KIRIMDATA/521KOMA6MB.mkv,504536528.0
,dtn://klien.dtn/dtncp/recv?
file=/521KOMA6MB.mkv,3600,,192.168.20.1:41873,,,0.000000,,0,link
[1451222281.057112 /dtn/bundle/daemon info] BUNDLE_RECEIVED
bundle id 0 [dtn://fitri.dtn/dtncp/send?
source=/home/fitri/Documents/KIRIMDATA/521KOMA6MB.mkv ->
dtn://klien.dtn/dtncp/recv?file=/521KOMA6MB.mkv 521643214 byte
payload] prevhop dtn://putri.dtn (521643373 bytes recvd)
```

Gambar 4.2. *t* pengiriman pada log klien

4.2.1.1. Pengujian dalam Kondisi LoS (*Line of Sight*)

Pada percobaan pertama, paket data dikirim dengan jarak 15 meter dengan data yang diuji berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 700 MB,

1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Adapun hasil dari tabel perhitungan waktu *transfer* dan grafiknya adalah sebagai berikut (Tabel 6 dan Gambar 4.3).

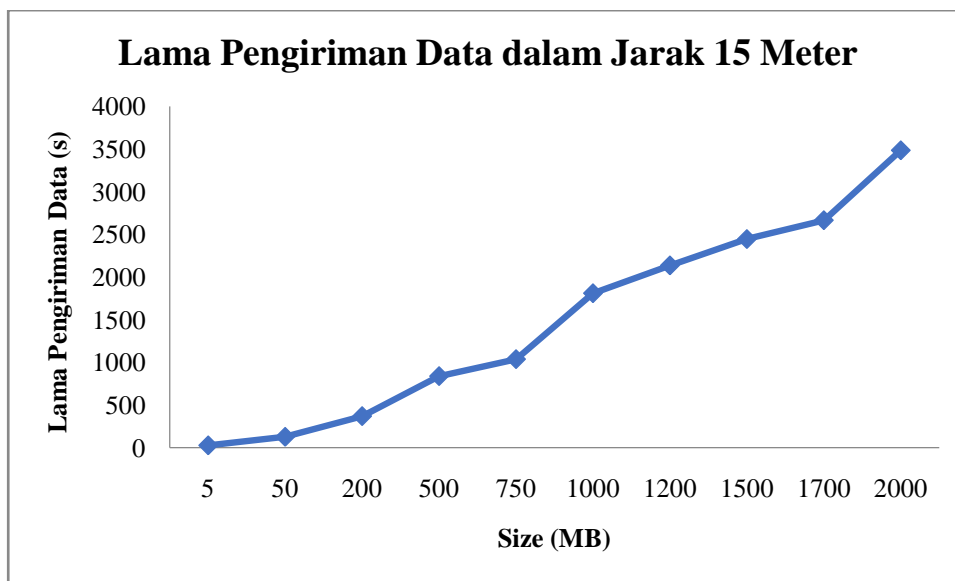
Tabel 6

Lama Pengiriman Data LoS dalam Jarak 15 Meter

Size (MB)	<i>t</i> pengiriman	<i>t</i> diterima	Lama pengiriman	<i>Lama</i> pengiriman (s)
5	12:36:42	12:37:08	0:00:26	26
50	12:39:54	12:42:00	0:02:06	126
200	12:43:11	12:49:20	0:06:09	369
500	12:51:29	13:05:27	0:13:58	838
750	13:11:18	13:28:35	0:17:17	1037
1000	12:05:09	12:35:19	0:30:10	1810
1200	12:23:01	12:58:37	0:35:36	2136
1500	13:44:03	14:24:48	0:40:45	2445
1700	13:00:18	13:44:44	0:44:26	2666
2000	14:04:34	15:02:40	0:58:06	3486

Size Data (MB)	Lama Pengiriman Data (detik)
5	26
50	126
200	369
500	838
750	1037
100	1810
1200	2136
1500	2445
1700	2666
2000	3486

Pada tabel 5 menunjukkan waktu yang dibutuhkan untuk data yang dikirim oleh server diterima oleh klien pada setiap ukuran data ujinya. Pada tabel tersebut, menunjukkan waktu pengiriman data mengalami peningkatan seiring ukuran data ujinya. Pada tabel juga menunjukkan lama pengiriman data paling cepat yaitu 5 MB sedangkan paling lama 2000 MB dan semua variasi data uji yang dikirim dari *server* sampai ke klien.



Gambar 4.3. Grafik Lama Pengiriman dalam Jarak 15 Meter

Pada gambar 4.3, grafik lama pengiriman dalam jarak 15 meter menunjukkan semakin besar ukuran data yang dikirim, semakin lama pula waktu pengiriman datanya. Dalam jarak 15 meter, ukuran data uji paling besar yaitu 2000 MB dapat diterima di klien dalam waktu pengiriman 3486 detik atau 58 menit 6 detik.

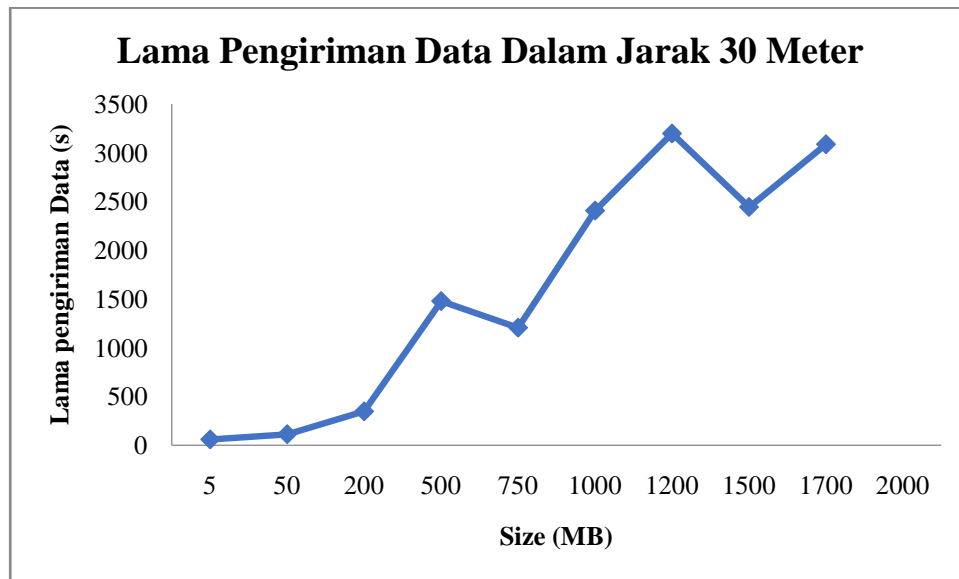
Pada percobaan kedua, paket data dikirim dengan jarak 30 meter dengan data yang diuji berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Adapun hasil tabel perhitungan waktu transfer dan grafiknya adalah sebagai berikut (Tabel 7 dan Gambar 4.4).

Tabel 7

Lama Pengiriman Data LoS dalam Jarak 30 meter

Size (MB)	t pengiriman	t diterima	Lama pengiriman	<i>Lama</i> pengiriman (s)
5	13:40:03	13:41:01	0:00:58	58
50	13:37:05	13:38:57	0:01:52	112
200	13:29:18	13:35:04	0:05:46	346
500	13:01:15	13:25:53	0:24:38	1478
750	12:39:26	12:59:31	0:20:05	1205
1000	12:04:50	12:44:56	0:40:06	2406
1200	5:03:42	5:56:59	0:53:17	3197
1500	12:24:08	13:04:51	0:40:43	2443
1700	13:07:29	13:58:56	0:51:27	3087
2000	-	-	-	-

Size Data (Mb)	Lama Pengiriman Data (detik)
5	58
50	112
200	346
500	1478
750	1205
1000	2406
1200	3197
1500	2443
1700	3087
2000	-



Gambar 4.4. Grafik Lama Pengiriman dalam Jarak 30 Meter

Pada percobaan kedua yang ditunjukkan oleh tabel 6 dan gambar 4.4, terlihat ketika pengiriman data 1900 MB, data sudah tidak dapat terkirim atau gagal diterima oleh klien. Grafik menunjukkan lama pengiriman data naik turun, hal ini mungkin dikarenakan pada saat pengambilan data pada jarak 30 meter ini, koneksi wifi sedang tidak baik (koneksi hilang / tidak tersedia), sehingga pada saat pengiriman data cenderung waktu pengiriman data tidak stabil. Pada pengiriman dalam jarak 30 meter, ukuran data terbesar yang sampai di klien adalah data dengan ukuran 1200 MB dengan waktu lama pengiriman 3197 detik atau 53 menit 17 detik.

Pada percobaan ketiga, paket data dikirim dengan jarak 50 meter dengan data yang diuji berturut-turut 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Namun pada pengiriman data dengan ukuran data 1200 MB, data tidak terkirim ke klien. Hal ini dikarenakan jarak yang tidak memungkinkan untuk router terkoneksi dengan *wifi* sehingga waktu *expired* yang ditentukan habis sehingga koneksi pun terputus dan data tidak dapat terkirim.

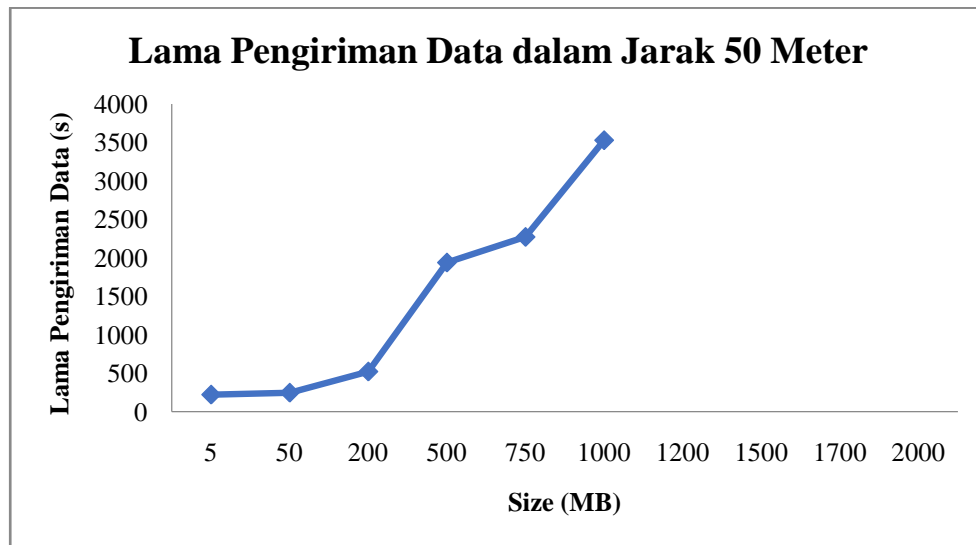
Pada Tabel 8 dan gambar 4.5 menunjukkan data terbesar yang dapat terkirim ke klien dalam jarak 50 meter adalah 1000 MB dengan waktu 3629 detik atau 1 jam 29 detik.

Tabel 8

Lama Pengiriman Data LoS dalam Jarak 50 meter

Size (MB)	t pengiriman	t diterima	Lama pengiriman	<i>Lama pengiriman (s)</i>
5	2:34:16	2:37:59	0:03:43	223
50	2:42:20	2:46:28	0:04:08	248
200	3:37:40	3:46:23	0:08:43	523
500	3:48:37	4:20:56	0:32:19	1939
750	4:21:36	4:59:28	0:37:52	2272
1000	2:27:37	3:28:06	0:59:29	3569
1200	-	-	-	-
1500	-	-	-	-
1700	-	-	-	-
2000	-	-	-	-

Size Data (MB)	Lama Pengiriman Data (detik)
5	223
50	248
200	523
500	1939
750	2272
1000	3529
1200	-
1500	-
1700	-
2000	-



Gambar 4.5. Grafik Lama Pengiriman dalam Jarak 50 Meter

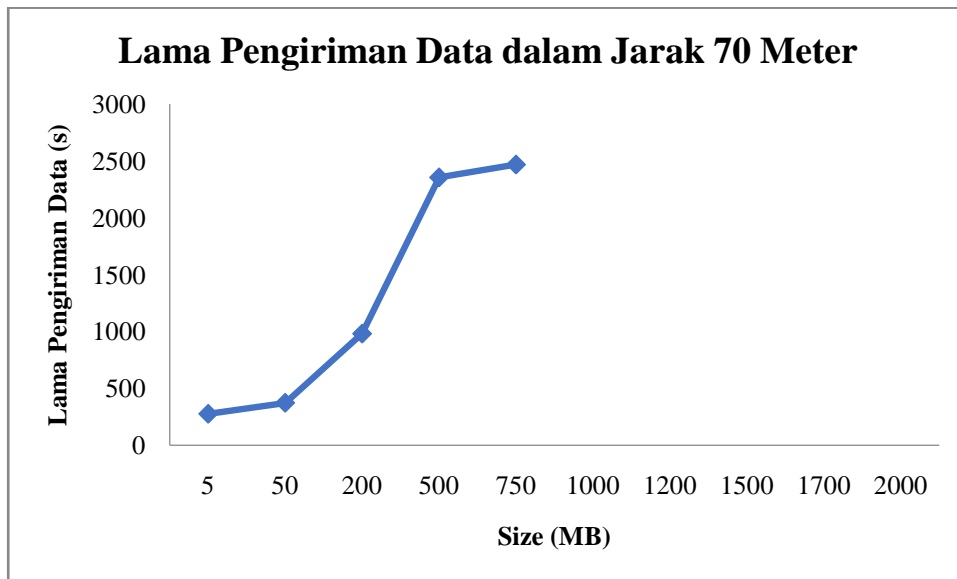
Pada percobaan keempat, data uji dikirim dengan jarak 70 meter dengan data yang diuji berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Pada jarak 70 meter, koneksi sistem DTN sudah mulai tidak stabil, data yang terkirim ke klien yaitu hanya sampai pada ukuran data 700 MB dengan lama pengiriman data 2467 detik. Hasil tabel perhitungan lama pengiriman data dalam jarak 70 meter dan grafiknya dapat dilihat pada tabel 9 dan gambar 4.6 berikut.

Tabel 9

Lama Pengiriman Data LoS dalam Jarak 70 meter

Size (MB)	t Pengiriman	t Diterima	Lama Pengiriman	Lama Pengiriman (s)
5	1:59:25	2:04:00	0:04:35	275
50	2:04:53	2:11:05	0:06:12	372
200	2:14:45	2:31:05	0:16:20	980
500	2:30:26	3:09:40	0:39:14	2354
750	3:13:16	3:54:23	0:41:07	2467
1000	-	-	-	-
1200	-	-	-	-
1500	-	-	-	-
1700	-	-	-	-
2000	-	-	-	-

Size (MB)	Lama Pengiriman (s)
5	275
50	372
200	980
500	2354
750	2467
1000	-
1200	-
1500	-
1700	-
2000	-



Gambar 4.6. Grafik Lama Pengiriman dalam Jarak 70 meter

4.2.1.2. Pengujian dalam Kondisi N-LoS (*Non Line of Sight*)

Pengujian N-LoS (*non-Line of Sight*) dilakukan sama seperti pengujian LoS (*line of Sight*), hanya saja LoS dalam keadaan antar satu point dengan point lain saling bebas dari apapun yang akan menghalangi pandangan, sedangkan non LoS, kondisi pandangan antar pointnya terhalang (pohon, rumah, gedung, dan lain-lain). Pengujian antar LoS dan N-LoS dilakukan pada lokasi yang berbeda sesuai spesifikasi dari keduanya, hanya saja jarak dan data uji yang digunakan dalam pengujian/pengambilan data sama seperti LoS.

Pada percobaan pertama, paket data dikirim dengan jarak 15 meter dengan data yang diuji berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Adapun hasil dari tabel perhitungan waktu transfer dan grafiknya adalah sebagai berikut (Tabel 10 dan Gambar 4.7).

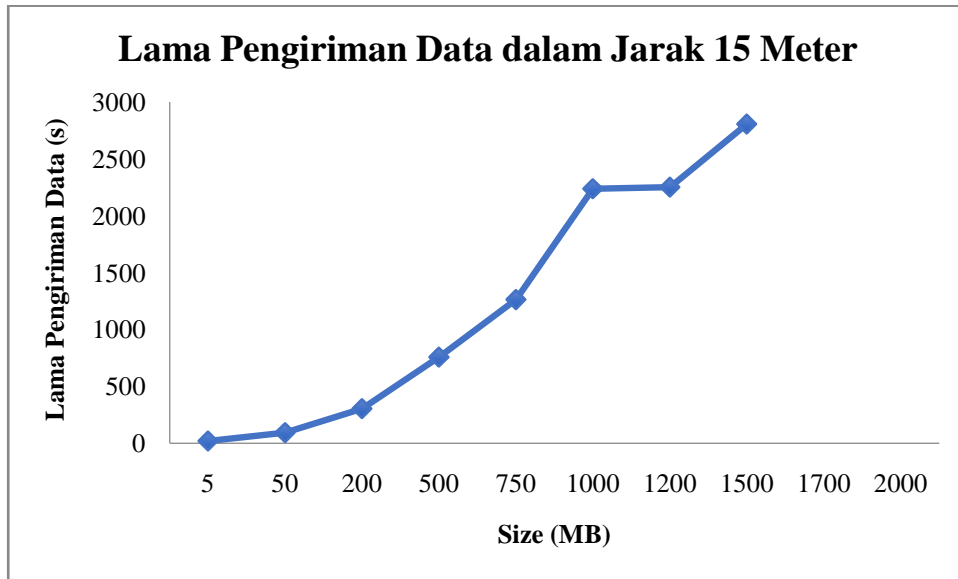
Tabel 10

Lama Pengiriman Data N-Los dalam Jarak 15 Meter

Size (MB)	t Pengiriman	t Diterima	Lama Pengiriman	Lama Pengiriman (s)
5	1:54:12	1:54:32	0:00:20	20
50	1:58:19	1:59:52	0:01:33	93
200	2:03:40	2:08:43	0:05:03	303
500	2:11:54	2:24:31	0:12:37	757
750	2:26:29	2:47:31	0:21:02	1262
1000	2:50:04	3:27:21	0:37:17	2237
1200	3:29:55	4:07:26	0:37:31	2251
1500	4:10:51	4:57:36	0:46:45	2805
1700	-	-	-	-
2000	-	-	-	-

Size (MB)	Lama Pengiriman (s)
5	20
50	93
200	303
500	757
750	1262
1000	2237
1200	2251
1500	2805
1700	-
2000	-

Pada jarak 15 meter, lama pengiriman data maksimum dari node server ke node klien yang dicapai pada kondisi N-LoS adalah 2805 detik atau 46 menit 45 detik dengan data uji maksimum yang sampai ke tujuan adalah 1500 MB.



Gambar 4.7. Grafik Lama Pengiriman dalam Jarak 15 meter

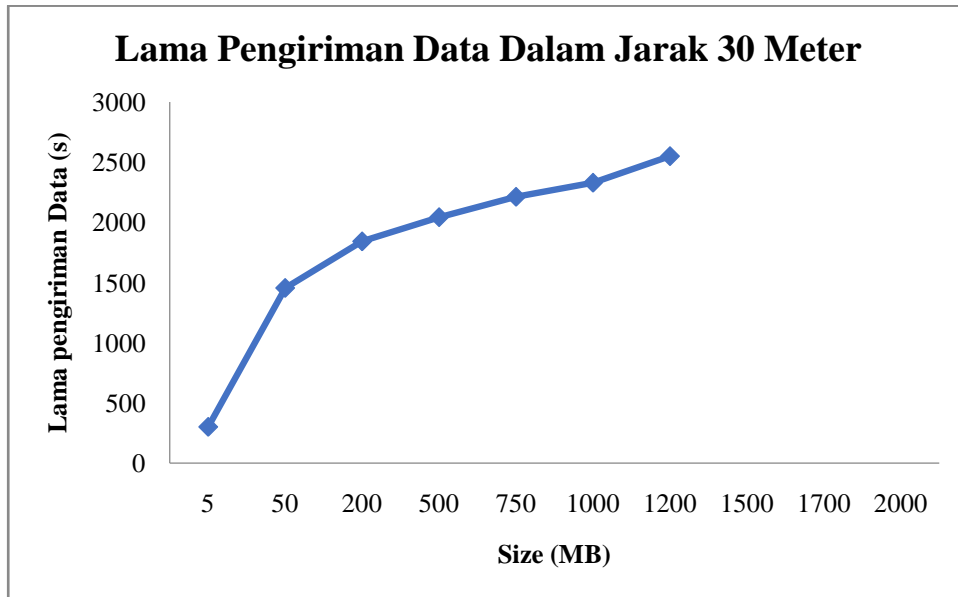
Pengujian kemudian dilanjutkan pada jarak 30 meter. Pada jarak 30 meter, data yang terkirim sampai ke klien adalah 1200 MB. Lama pengiriman data dari node server ke node klien yang dicapai pada kondisi N-LoS dengan jarak 30 meter adalah 2551 detik atau 42 menit 31 detik. Tabel dan grafik dari lama pengiriman data N-LoS pada jarak 30 meter dapat dilihat pada tabel 12 dan gambar 4.7 berikut ini.

Tabel 11

Lama Pengiriman Data N-LoS dalam Jarak 30 Meter

Size (MB)	t Pengiriman	t Diterima	Lama Pengiriman	Lama Pengiriman (s)
5	1:18:49	1:23:51	0:05:02	302
50	1:25:15	1:49:31	0:24:16	1456
200	1:49:44	2:20:27	0:30:43	1843
500	3:46:55	4:20:59	0:34:04	2044
750	0:19:13	0:56:06	0:36:53	2213
1000	1:08:35	1:47:25	0:38:50	2330
1200	1:48:49	2:31:20	0:42:31	2551
1500	-	-	-	-
1700	-	-	-	-
2000	-	-	-	-

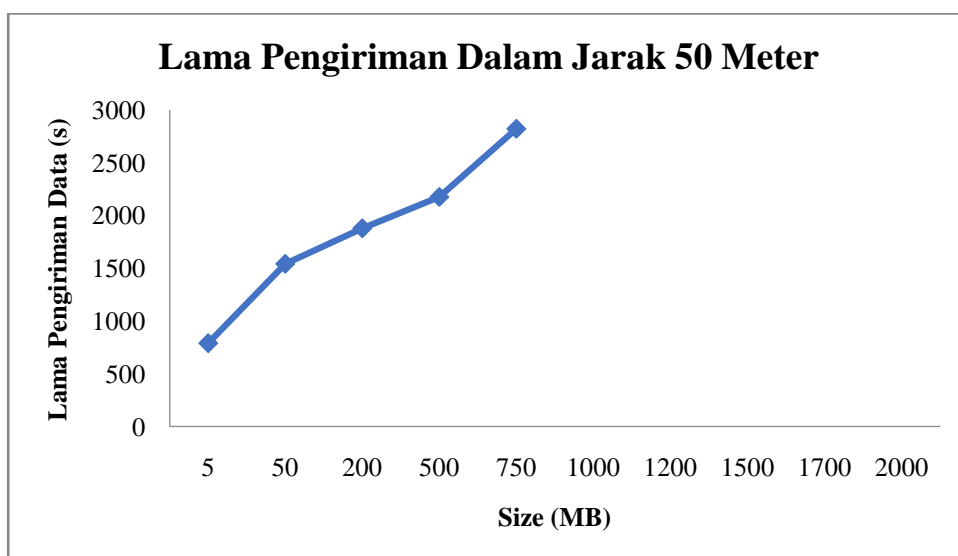
Size (MB)	Lama Pengiriman (s)
5	302
50	1456
200	1843
500	2044
750	2213
1000	2330
1200	2551
1500	-
1700	-
2000	-



Gambar 4.8. Grafik Lama Pengiriman dalam Jarak 30 meter

Pada percobaan ketiga, paket data dikirim dengan jarak 50 meter dengan data yang diuji berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB.

Dapat dilihat pada Tabel 13 dan gambar 4.9 menunjukkan dalam kondisi N-LoS pada jarak 50 meter data terbesar yang dapat terkirim/sampai ke klien dalam jarak 50 meter dengan kondisi N-LoS adalah 700 MB dengan waktu transfer data mencapai 2820 detik atau 47 menit.



Gambar 4.9. Grafik Lama Pengiriman dalam Jarak 50 meter

Tabel 12

Lama Pengiriman Data N-LoS dalam Jarak 50 Meter

Size (MB)	t Pengiriman	t Diterima	Lama Pengiriman	Lama Pengiriman (s)
5	4:25:13	4:38:21	0:13:08	788
50	5:00:59	5:26:39	0:25:40	1540
200	6:22:26	6:53:43	0:31:17	1877
500	6:53:45	7:29:59	0:36:14	2174
750	7:32:57	8:19:57	0:47:00	2820
1000	-	-	-	-
1200	-	-	-	-
1500	-	-	-	-
1700	-	-	-	-
2000	-	-	-	-

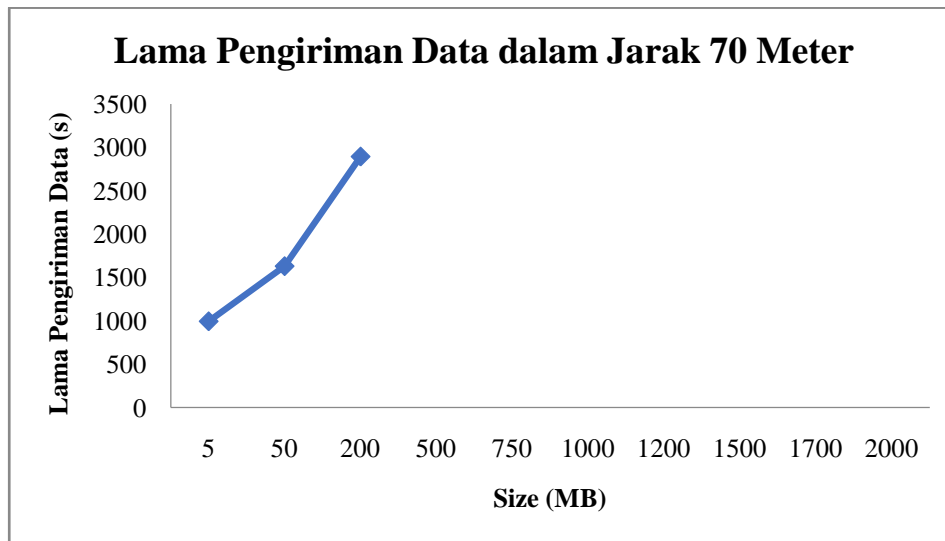
Size (MB)	Lama Pengiriman (s)
5	788
50	1540
200	1877
500	2174
750	2820
1000	-
1200	-
1500	-
1700	-
2000	-

Tabel 13

Lama Pengiriman Data N-LoS dalam Jarak 70 Meter

Size (MB)	t Pengiriman	t Diterima	Lama Pengiriman	Lama Pengiriman (s)
5	6:51:09	7:07:43	0:16:34	994
50	7:14:16	7:41:26	0:27:10	1630
200	7:48:56	8:37:10	0:48:14	2894
500	-	-	-	-
750	-	-	-	-
1000	-	-	-	-
1200	-	-	-	-
1500	-	-	-	-
1700	-	-	-	-
2000	-	-	-	-

Size (MB)	Lama Pengiriman (s)
5	994
50	1630
200	2894
500	-
750	-
1000	-
1200	-
1500	-
1700	-
2000	-



Gambar 4.10. Grafik Lama Pengiriman dalam Jarak 70 meter

Pengujian terakhir dilakukan dengan data uji yang sama dengan pengujian sebelumnya dalam jarak 70 meter dengan kondisi N-LoS. Seperti yang terlihat pada tabel 12 dan gambar 4.10, data yang terkirim pada jarak 70 meter ini hanya mumpuni untuk mencapai data uji 200 MB dengan lama pengiriman data adalah 2894 detik atau 48 menit 14 detik.

4.2.1.3. Perbandingan Lama Pengiriman Data antara LoS dan N-LoS

Dari hasil pengujian antara LoS dan N-LoS diperoleh lama pengiriman data pada setiap kondisinya dalam jarak ukur 15 meter, 30 meter, 50 meter dan 70 meter. Pada setiap jarak, terlihat perbedaan hasil lama pengiriman data antara node server ke node klien (tujuan) pada setiap ukuran data uji yang digunakan. Semakin besar data uji yang digunakan, maka semakin lama pula waktu pengiriman data untuk sampai ke tujuan. Berikut Tabel 13 dan tabel 14 menunjukkan perbandingan lama pengiriman data pada setiap jaraknya di kondisi LoS dan kondisi N-LoS dengan data uji 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB.

Tabel 14

Perbandingan Lama Pengiriman Data (s) dalam Kondisi LoS

Size (MB) Jarak (m)	Lama Pengiriman Data dalam Kondisi <i>LoS</i> (detik)			
	15 meter	30 meter	50 meter	70 meter
5	26	58	223	275
50	126	112	248	372
200	369	346	523	980
500	838	1478	1939	2354
750	1037	1205	2272	2467
100	1810	2406	3629	-
1200	2136	3197	-	-
1500	2445	2443	-	-
1700	2666	3087	-	-
2000	3486	-	-	-

Tabel 15

Perbandingan Lama Pengiriman Data (s) dalam Kondisi N-LoS

Size (MB) Jarak (m)	Lama Pengiriman Data dalam Kondisi <i>N-LoS</i> (detik)			
	15 meter	30 meter	50 meter	70 meter
5	20	302	788	994
50	93	1456	1540	1630
200	303	1843	1877	2894
500	757	2044	2174	-
750	1262	2213	2820	-
100	2237	2330	-	-
1200	2251	2551	-	-
1500	2805	-	-	-
1700	-	-	-	-
2000	-	-	-	-

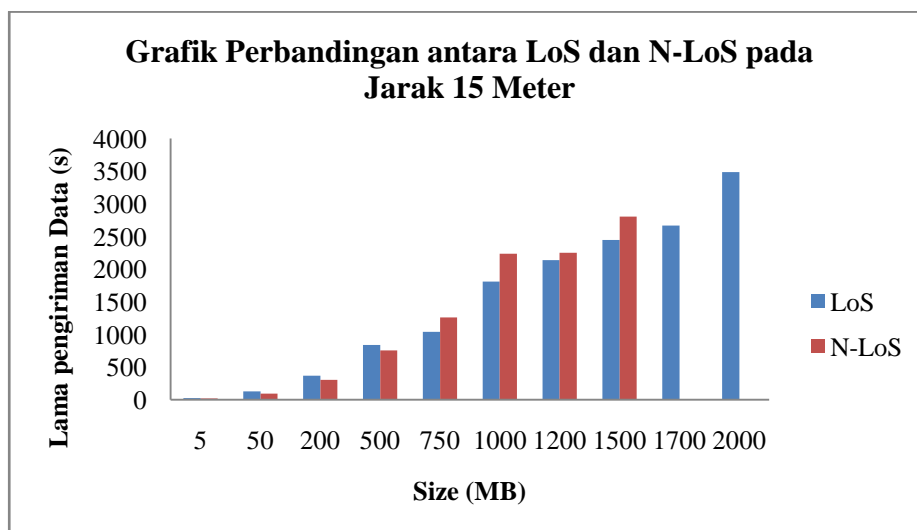
Pada pengujian pertama dilakukan dengan mengambil data dengan jarak 15 meter. Pada Tabel 15 dan Gambar 4.11 terlihat grafik perbandingan lama

pengiriman data dalam kondisi LoS dan N-LoS pada jarak 15 meter. Garis berwarna biru menunjukkan kondisi LoS sedangkan garis merah menunjukkan kondisi N-LoS. Pada kondisi LoS ukuran data yang diujikan terkirim semua ke klien yaitu dengan data uji 2000 MB dalam waktu 3486 detik sedangkan pada kondisi N-LoS data yang diterima di klien hanya sampai 1500 MB dengan lama pengiriman data yaitu 3075 detik.

Tabel 16

Perbandingan LoS dan N-LoS pada Jarak 15 Meter

Size (Mb)	Lama Pengiriman Data (s)	
	LoS	N-LoS
5	26	20
50	126	93
200	369	303
500	838	757
750	1037	1262
1000	1810	2237
1200	2136	2251
1500	2445	2805
1700	2666	-
2000	3486	-



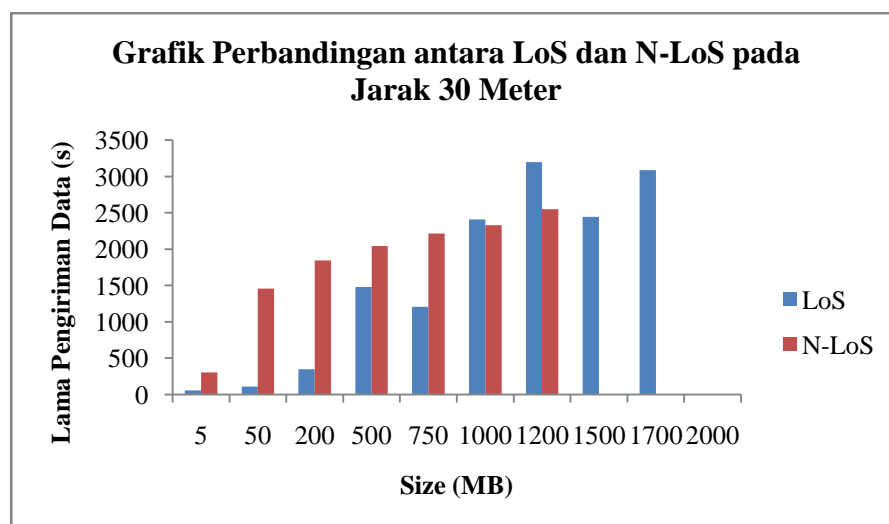
Gambar 4.11 Grafik Perbandingan LoS dan N-LoS pada Jarak 15 Meter

Pada pengujian kedua, paket data dikirim dengan jarak 30 meter dengan data yang diuji berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 700 MB, 1000 MB, 1200 MB, 1700 MB, dan 2000 MB. Adapun grafik perbandingan antara LoS dan N-LoS dapat dilihat pada gambar 4.12.

Tabel 17

Perbandingan LoS dan N-LoS pada Jarak 30 Meter

Size (MB)	Lama Pengiriman Data (s)	
	LoS	N-LoS
5	58	302
50	112	1456
200	346	1843
500	1478	2044
750	1205	2213
1000	2406	2330
1200	3197	2551
1500	2443	-
1700	3087	-
2000	-	-



Gambar 4.12 Grafik Perbandingan LoS dan n-LoS pada Jarak 30 Meter

Pada grafik terlihat ketika pengiriman data 2000 MB, data sudah tidak dapat terkirim atau gagal diterima oleh klien. Grafik menunjukkan lama pengiriman data naik turun, hal ini mungkin dikarenakan pada saat pengambilan data pada

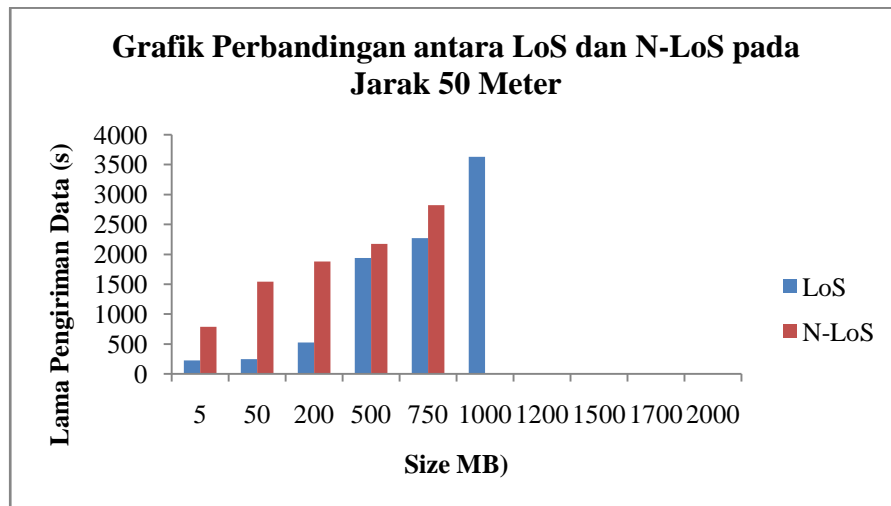
jarak 30 meter ini, koneksi wifi sedang tidak baik (koneksi hilang/tidak tersedia), sehingga pada saat pengiriman data cenderung waktu pengiriman data tidak stabil. Pada pengiriman dalam jarak 30 meter, ukuran data terbesar yang sampai di klien adalah data dengan ukuran 1700 MB dengan waktu lama pengiriman 3087 detik. Sedangkan pada kondisi N-LoS data uji maksimal yang sampai di klien adalah data uji 1200 MB dengan durasi transfer data 2551 detik.

Pada pengambilan data selanjutnya dilakukan pada jarak 50 meter. Pada gambar 4.13 terlihat pada kondisi LoS dengan ukuran data 1200 MB, data tidak terkirim ke klien. maksimal data uji yang terkirim yaitu berukuran 1000 MB dengan waktu transfer yaitu 3629 detik sedangkan pada kondisi N-LoS data uji yang sampai hanya 750 MB dalam waktu pengiriman data selama 2820 detik. Hal ini dikarenakan jarak yang tidak memungkinkan untuk *router* terkoneksi dengan wifi sehingga waktu *expired* yang ditentukan habis sehingga koneksi pun terputus dan data tidak dapat terkirim.

Tabel 18

Perbandingan LoS dan N-LoS pada Jarak 50 Meter

Size (Mb)	Lama Pengiriman Data (s)	
	LoS	N-LoS
5	223	788
50	248	1540
200	523	1877
500	1939	2174
750	2272	2820
1000	3629	-
1200	-	-
1500	-	-
1700	-	-
2000	-	-



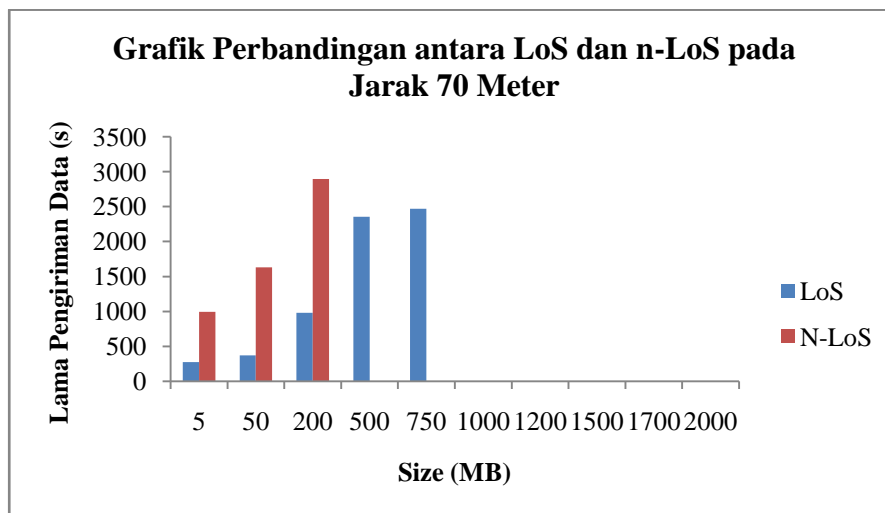
Gambar 4.13 Grafik Perbandingan LoS dan n-LoS pada Jarak 50 Meter

Pada pengujian terakhir dilakukan pada jarak 70 meter. Dalam kondisi LoS, data uji maksimal yang terkirim ke tujuan yaitu 750 MB dengan lama pengiriman data 2467 detik, sedangkan dalam kondisi N-LoS hanya dapat mengirim data berukuran 200 MB dengan waktu transfer selama 2894 detik. Grafik dari perbandingan pengiriman data antara LoS dan n-LoS dapat dilihat pada gambar 4.14.

Tabel 19

Perbandingan LoS dan N-LoS pada Jarak 70 Meter

Size (MB)	Lama Pengiriman Data (s)	
	LoS	N-LoS
5	275	994
50	372	1630
200	980	2894
500	2354	-
750	2467	-
1000	-	-
1200	-	-
1500	-	-
1700	-	-
2000	-	-



Gambar 4.14 Grafik Perbandingan LoS dan n-LoS pada Jarak 70 Meter

Pengujian sistem DTN ini dilakukan dengan menempatkan jarak semaksimal mungkin hingga koneksi wifi sistem DTN tidak tersedia / tidak terkoneksi ke klien.

Dari hasil pengujian dan pengambilan data, keseluruhan sistem berjalan secara fungsional. Sistem dari skenario DTN dalam penelitian ini mampu menghasilkan output yang sesuai dengan yang diharapkan sebelumnya. Hasil pengambilan data dalam jarak 15 meter, 30 meter, 50 meter dan 70 meter dengan ukuran besar data berturut-turut adalah 5 MB, 50 MB, 200 MB, 500 MB, 750 MB, 1000 MB, 1200 MB, 1500 MB, 1700 MB, dan 2000 MB ini memiliki kecenderungan bahwa semakin besar ukuran data/file yang dikirim maka waktu transfer / durasi yang dibutuhkan untuk mengirim file tersebut juga akan semakin besar.

4.2.2. Data Uji dan Lama Pengiriman Maksimum pada Setiap Jarak

Pada setiap pengujian tidak semua data uji yang diujikan dapat terkirim ke klien. Hal ini bergantung pada jarak ukur dari setiap pengambilan data. Tabel data uji yang sampai / terkirim ke tujuan pada setiap jarak uji 15 meter, 30 meter, 50 meter dan 70 meter dalam kondisi LoS (*Line of Sight*) maupun N-LoS (*Non Line of Sight*) dapat dilihat pada tabel 19.

Pada tabel 19 terlihat pada jarak 15 meter, dalam kondisi LoS data uji terkirim semua ke klien sedang kan pada kondisi N-LoS hanya mencapai data uji maksimum 1500 MB. Kemudian pada jarak 30 meter, dalam kondisi LoS hanya mampu mengirim data ke klien dengan data uji 1700 MB, sedangkan pada kondisi n-LoS, hanya mencapai data uji 1200 MB. Pada jarak berikutnya yaitu 50 meter, dalam kondisi LoS dan N-LoS berturut-turut data uji maksimum yang terkirim mencapai 100 MB dan 700 MB. Sedangkan jarak uji terakhir yang dilakukan yaitu pada jarak 70 meter, pada kondisi LoS data yang terkirim ke tujuan hanya mencapai data maksimum 700 MB sedangkan pada kondisi N-LoS hanya mencapai 200 MB.

Setiap data uji yang terkirim ke tujuan (klien), memiliki lama pengiriman yang bervariasi. Hasil dari pengolahan data dari pengujian sistem menunjukkan bahwa lama pengiriman maksimum pada setiap jarak dalam kondisi LoS maupun n-LoS cenderung semakin besar data uji yang dikirim, semakin lama pula waktu yang dibutuhkan agar data sampai ke klien. Lama pengiriman data maksimum pada setiap jarak pengujian dalam kondisi LoS maupun N-LoS ini dapat dilihat pada tabel 20.

Namun pada pengujian sistem ini, seperti yang terlihat pada tabel 21, pada pengambilan data LoS dalam jarak 30 meter, data uji 1200 MB lebih lama terkirim daripada data uji 1500 MB dan 1700 MB. hal ini dikarenakan pada saat pengambilan data pada jarak 30 meter ini, koneksi wifi sistem DTN sedang tidak baik (koneksi hilang / tidak tersedia), sehingga pada saat pengiriman data berlangsung cenderung waktu pengiriman data tidak stabil sehingga waktu yang diperlukan data sampai ke node selanjutnya pun semakin banyak diperlukan.

Tabel 20

Data Uji yang Terkirim ke Tujuan pada Setiap Jarak dalam Kondisi LoS dan N-LoS

Jarak	5 MB		50 MB		200 MB		500 MB		750 MB		1000 MB		1200 MB		1500 MB		1700 MB		2000 MB	
	los	N-los	los	N -los	los	N -los	los	N-los	los	N-los	los	N-los	los	N-los	los	N-los	los	N-los	los	N-los
15 m	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	√	-
30 m	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	√	-	-	-
50 m	√	√	√	√	√	√	√	√	√	√	√	-	-	-	-	-	-	-	-	-
70 m	√	√	√	√	√	√	√	-	√	-	-	-	-	-	-	-	-	-	-	-

Tabel 21

Lama Pengiriman Maksimum yang Terkirim ke Tujuan pada Setiap Jarak dalam Kondisi LoS dan N-LoS

	15 Meter		30 Meter		50 Meter		70 Meter	
	los	N-los	los	N-los	los	N-los	los	N-los
Lama Pengiriman	3486 s	2805 s	3197 s	2551 s	3569 s	2820 s	2467 s	2894 s
Data Uji Maksimum	2000 MB	1500 MB	1200 MB	1200 MB	1000 MB	700 MB	750 MB	200 MB

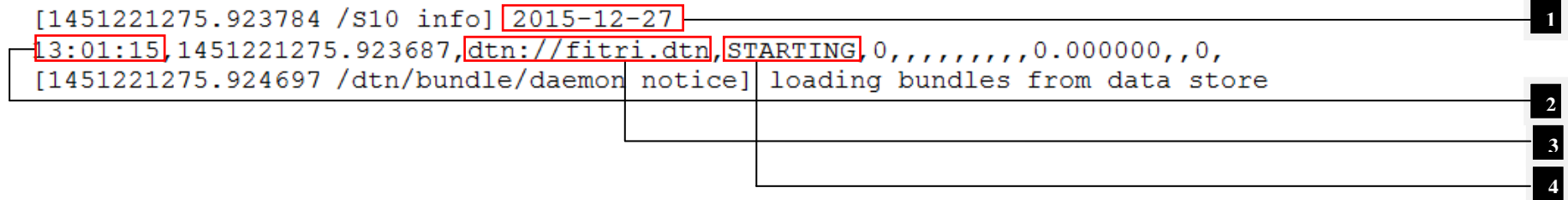
Tabel 22

Data Uji Maksimum yang Terkirim ke Tujuan pada Setiap Jarak dalam Kondisi LoS dan N-LoS

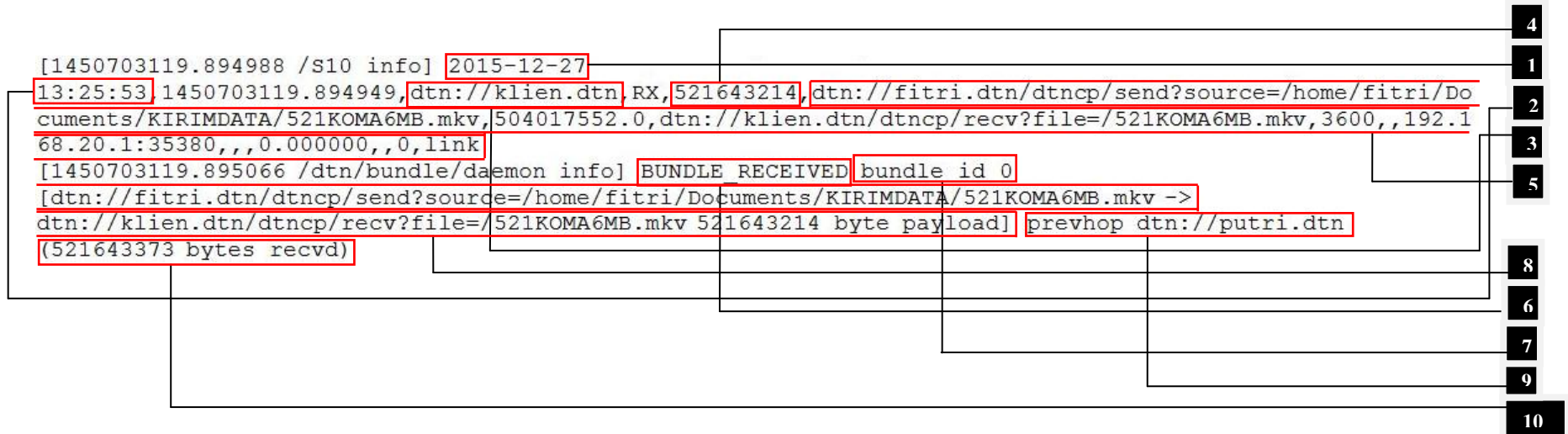
	15 Meter		30 Meter		50 Meter		70 Meter	
	los	N-los	los	N-los	los	N-los	los	N-los
Data Uji Maksimum	2000 MB	1500 MB	1700 MB	1200 MB	1000 MB	700 MB	750 MB	200 MB
Lama Pengiriman	3486 s	2805 s	3087 s	2551 s	3569 s	2820 s	2467 s	2894 s

4.2.3. Parameter Uji dalam *Log Bundle*

Dalam pengujian skenario jaringan DTN, didalam *log bundle dtn* terdapat parameter-parameter uji yang harus diperhatikan dalam proses pengiriman data. Penjelasan mengenai parameter uji yang dilihat dari contoh log data server pada pengiriman data 500 MB dapat dilihat pada gambar 4.6. Gambar 4.6 menunjukkan hasil *log bundle* pada server yang didapatkan ketika pengambilan data dilakukan.



Gambar 4.15. sample log bundle server



Gambar 4.16. sample log bundle klien

Untuk melihat waktu transfer pengiriman data dapat dilihat di *log bundle* pada device server. Adapun keterangan data *log* tersebut yaitu:

1. Pada poin pertama dan kedua menunjukkan tanggal dari pengiriman oleh server.
2. Pada *log server*, terdapat waktu (jam) sesaat data dikirim oleh server dan merupakan parameter untuk mengukur berapa lama waktu transfer data.
3. `dtm://fitri.dtm` disini menunjukkan alamat dari server (pengirim).
4. Pada poin 4 ditunjukkan bahwa server sedang memulai mengirim data.

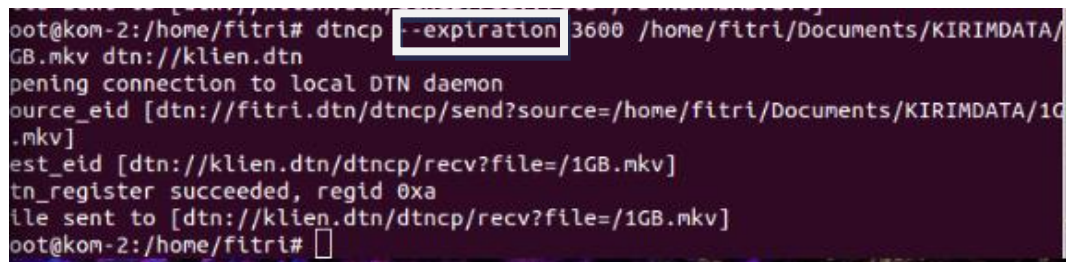
Dalam log data klien (penerima) tampilannya berbeda, dimana parameter yang dipakai adalah waktu dari `bundle_received` yang artinya bundle telah diterima di klien. Pada gambar 4.7 merupakan contoh *log* data klien pada pengiriman data 500 MB. Gambar 4.7 menunjukkan hasil *log bundle* pada klien yang didapatkan ketika pengambilan data dilakukan. Untuk melihat waktu transfer penerimaan data (data sampai ke tujuan) dapat dilihat di *log bundle* pada klien. Adapun keterangan data dari log bundle tersebut yaitu sebagai berikut:

1. Pada poin pertama dan kedua menunjukkan tanggal dari pengiriman oleh server.
2. Pada log server, terdapat waktu (jam) sesaat data dikirim oleh server dan merupakan parameter untuk mengukur berapa lama waktu transfer data.
3. `dtm://klien.dtm` disini menunjukkan tempat bundle saat ini berada.
4. Pada poin 4 ditunjukkan ukuran data yang sedang dikirim.
5. Pada poin 5, menunjukkan direktori data yang dikirim oleh komputer server (pengirim).
6. `BUNDLE_RECEIVED` disini menunjukkan bahwa bundle telah terkirim di klien.
7. Sedangkan `bundle id 0` menunjukkan penomoran untuk `bundle_id` dari data yang dikirim.
8. Pada poin 8 menunjukkan direktori tempat data yang terkirim / diterima oleh klien disimpan.

9. `prevhop dtn://putri.dtn` disini menunjukkan node router yang melanjutkan pengiriman data oleh server sehingga data sampai di tujuan (klien).
10. Pada poin 10 menunjukkan bahwa ukuran data yang diterima di klien dalam satuan bytes.

4.2.4. *Bundle Expiration* pada DTN

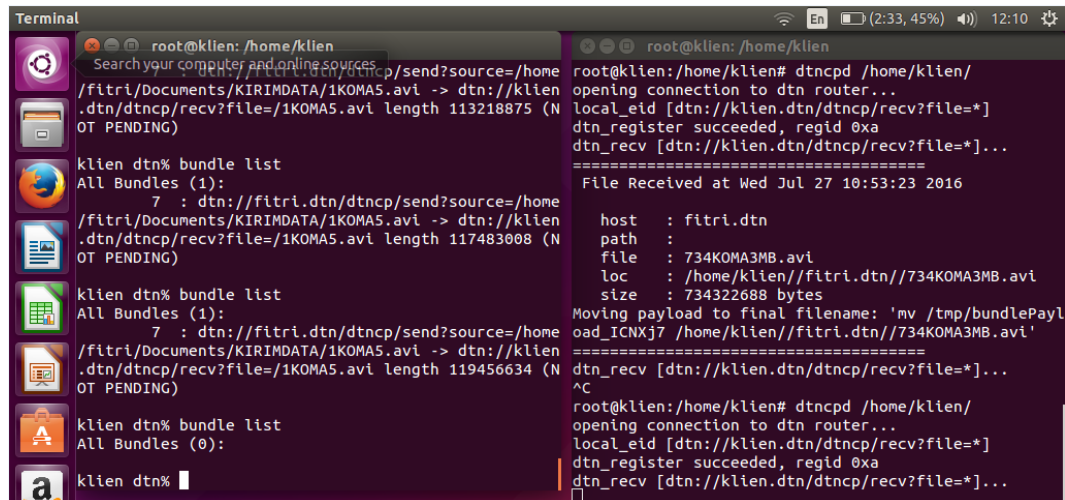
Pada saat pengambilan data, *bundle* hanya bertahan melakukan transmisi ulang selama 3600 detik seperti yang telah diatur pada saat pengiriman data ketika aplikasi `dtnd` dijalankan seperti yang terlihat pada gambar 4.16. Apabila lewat dari waktu *expiration*, maka antrian bundle tidak terkirim ke tujuan / tidak sampai ke klien.



```
oot@kom-2:/home/fitri# dtncp --expiration 3600 /home/fitri/Documents/KIRIMDATA/1GB.mkv dtn://klien.dtn
opening connection to local DTN daemon
source_eid [dtn://fitri.dtn/dtncp/send?source=/home/fitri/Documents/KIRIMDATA/1GB.mkv]
dest_eid [dtn://klien.dtn/dtncp/recv?file=/1GB.mkv]
tn_register succeeded, regid 0xa
file sent to [dtn://klien.dtn/dtncp/recv?file=/1GB.mkv]
oot@kom-2:/home/fitri#
```

Gambar 4.17. Expiration pada Saat Data Uji Dikirim ke Klien

Ketika pengambilan data dilakukan, *device* yang terhubung pada sistem DTN tidak selalu dalam kondisi stabil. Ketika jarak dari klien tidak memungkinkan untuk terkoneksi ke sistem DTN / terputus-putus sehingga sistem DTN akan melakukan transmisi ulang sampai waktu *expired* habis. Apabila sampai pada saat waktu *expiration* habis, maka koneksi pun terputus dan data tidak terkirim ke klien seperti yang terlihat pada gambar 4.17.



Gambar 4.18. Data tidak terkirim ke Klien

Pada saat data tidak terkirim, log DTN akan menunjukkan BUNDLE EXPIRED seperti salah satu contoh log pengiriman data berikut ini:

```
[1469594932.079971 /S10 info] 2016-07-27
04:48:52,1469594932.079932,dtn://klien.dtn,CONTUP,0,,,,,192.1
68.20.1:41006,,,0.000000,OPPORTUNISTIC,0,
[1469595596.063725 /dtn/bundle/daemon info] BUNDLE_EXPIRED
bundle id 0
[dtn://fitri.dtn/dtnrcp/send?source=/home/fitri/Documents/KIRIM
DATA/1KOMA5.avi -> dtn://klien.dtn/dtnrcp/recv?file=/1KOMA5.avi
49161089 byte payload, fragment @0/1458949864]
[1469595596.258317 /dtn/bundle/daemon info] BUNDLE_EXPIRED
bundle id 5
[dtn://fitri.dtn/dtnrcp/send?source=/home/fitri/Documents/KIRIM
DATA/1KOMA5.avi -> dtn://klien.dtn/dtnrcp/recv?file=/1KOMA5.avi
26587365 byte payload, fragment @149769613/1458949864]
[1469595596.514093 /dtn/bundle/daemon info] BUNDLE_EXPIRED
bundle id 6
[dtn://fitri.dtn/dtnrcp/send?source=/home/fitri/Documents/KIRIM
DATA/1KOMA5.avi -> dtn://klien.dtn/dtnrcp/recv?file=/1KOMA5.avi
14628405 byte payload, fragment @174865646/1458949864]
[1469595596.587336 /dtn/bundle/daemon info] BUNDLE_EXPIRED
bundle id 3
[dtn://fitri.dtn/dtnrcp/send?source=/home/fitri/Documents/KIRIM
DATA/1KOMA5.avi -> dtn://klien.dtn/dtnrcp/recv?file=/1KOMA5.avi
17668171 byte payload, fragment @52121291/1458949864]
[1469595596.789816 /dtn/bundle/daemon info] BUNDLE_EXPIRED
bundle id 4
[dtn://fitri.dtn/dtnrcp/send?source=/home/fitri/Documents/KIRIM
DATA/1KOMA5.avi -> dtn://klien.dtn/dtnrcp/recv?file=/1KOMA5.avi
84467553 byte payload, fragment @67317292/1458949864]
[1469595596.993259 /dtn/bundle/daemon info] BUNDLE_EXPIRED
bundle id 2
[dtn://fitri.dtn/dtnrcp/send?source=/home/fitri/Documents/KIRIM
```

```

DATA/1KOMA5.avi -> dtn://klien.dtn/dtnctp/recv?file=/1KOMA5.avi
6016915 byte payload, fragment @47521642/1458949864]
[1469595948.065416 /dtn/cl/tcp/conn/0xb3900490 info] remote
socket closed connection -- returned POLLHUP
[1469595948.066939 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPEN -> CLOSED] (connection broken)
for link link-0 [192.168.20.1:41006 dtn://putri.dtn
OPPORTUNISTIC tcp state=OPEN]
[1469595948.067273 /dtn/cl/tcp info] close_contact contact
192.168.20.1:41006 (started 1469594932.78787)
[1469595948.068685 /dtn/bundle/daemon info] LINK UNAVAILABLE
link-0 [192.168.20.1:41006 dtn://putri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE]
[1469595948.069053 /dtn/bundle/daemon info] CONTACT_DOWN link-
0 [192.168.20.1:41006 dtn://putri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE] (connection broken) (contact 0xb1b01138)
[1469595948.069215 /S10 info] 2016-07-27
05:05:48,1469595948.069175,dtn://klien.dtn,CONTDOWN,0,,,,,192
.168.20.1:41006,,,1469594932.078787,OPPORTUNISTIC,1016,
[1469595948.069539 /dtn/bundle/protocol info]
handle_bundle_received: begin
[1469595948.069691 /S10 info] 2016-07-27
05:05:48,1469595948.069647,dtn://klien.dtn,RX,119961617,dtn://
fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIRIMDATA/1K
OMA5.avi,522907196.0,dtn://klien.dtn/dtnctp/recv?file=/1KOMA5.a
vi,3600,,,192.168.20.1:41006,,,0.000000,,,0,link
[1469595948.069821 /dtn/bundle/daemon info] BUNDLE_RECEIVED
bundle id 7
[dtntn://fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIRIM
DATA/1KOMA5.avi -> dtn://klien.dtn/dtnctp/recv?file=/1KOMA5.avi
119961617 byte payload, fragment @187292751/1458949864]
prevhop dtn://putri.dtn (119961776 bytes recvd)
[1469595948.069971 /dtn/bundle/daemon info]
handle_bundle_received: validating bundle: calling
BlockProcessors?
[1469595948.070077 /dtn/bundle/protocol info]
BundleProtocol::validate begin
[1469595948.070182 /dtn/bundle/protocol info] validating each
individual blocks
[1469595948.070298 /dtn/bundle/protocol info]
BundleProtocol::validate end
[1469595948.070432 /dtn/bundle/expiration info] adding bundle
to pending list
[1469595948.113517 /dtn/bundle/expiration warning] scheduling
IMMEDIATE expiration for bundle id 7: [expiration 3600,
creation time 522907196.0, offset 946684800, now
1469595948.113495]
[1469595948.114266 /dtn/bundle/daemon info] BUNDLE_EXPIRED
bundle id 7
[dtntn://fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIRIM
DATA/1KOMA5.avi -> dtn://klien.dtn/dtnctp/recv?file=/1KOMA5.avi
119961617 byte payload, fragment @187292751/1458949864]
[1469595977.086387 /dtn/contact/link-0 info] new contact
contact 192.168.20.1:41007 (started 1469595977.86375)

```

```

[1469595977.087682 /dtn/bundle/daemon info] CONTACT_UP link-0
[192.168.20.1:41007 dtn://putri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE] (contact 0xb1b04c78)
[1469595977.087883 /S10 info] 2016-07-27
05:06:17,1469595977.087852,dtn://klien.dtn,CONTUP,0,,,,,192.1
68.20.1:41007,,,0.000000,OPPORTUNISTIC,0,
[1469596281.019628 /dtn/cl/tcp/conn/0xb3900490 info] got
SHUTDOWN (node is busy) [reconnect delay 0]
[1469596281.020866 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPEN -> CLOSED] (peer shut down)
for link link-0 [192.168.20.1:41007 dtn://putri.dtn
OPPORTUNISTIC tcp state=OPEN]
[1469596281.021326 /dtn/cl/tcp info] close_contact contact
192.168.20.1:41007 (started 1469595977.86375)
[1469596281.021922 /dtn/bundle/daemon info] LINK UNAVAILABLE
link-0 [192.168.20.1:41007 dtn://putri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE]
[1469596281.022282 /dtn/bundle/daemon info] CONTACT_DOWN link-
0 [192.168.20.1:41007 dtn://putri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE] (peer shut down) (contact 0xb1b04c78)
[1469596281.022439 /S10 info] 2016-07-27
05:11:21,1469596281.022399,dtn://klien.dtn,CONTDOWN,0,,,,,192
.168.20.1:41007,,,1469595977.086375,OPPORTUNISTIC,304,
[1469596331.443079 /dtn/tclcmd notice] exiting simple command
loop
[1469596331.443627 /dtn/tclcmd error] tcl error in
command_loop: "error writing "stdout": I/O error"
[1469596331.443756 /dtnd notice] command loop exited...
shutting down daemon
[1469596337.094476 /dtn/tclcmd notice] shutting down
interpreter
[1469596337.096274 /dtnd notice] shutting down dtn server
[1469596337.096405 /dtnd info] DTNServer shutdown called,
posting shutdown request to daemon
[1469596337.096586 /dtn/bundle/daemon notice] Received
shutdown request
[1469596337.096745 /dtnd notice] closing persistent data store
[1469596337.302095 /dtn/storage info] db closed
[1469596337.302930 /S10 info] 2016-07-27
05:12:17,1469596337.302877,dtn://klien.dtn,EXITING,0,,,,,,0
.000000,,0,
[1469596337.303173 /dtnd notice] daemon exiting..

```

Dari contoh *log* DTN tersebut dapat dilihat, antrian dari *bundle* yang dikirim oleh server, dimana apabila koneksi terputus-putus, akan otomatis mentransmisi ulang dan membentuk *bundle* antrian. Ilustrasi dari bentuk *bundle* antrian dapat dilihat pada gambar 4.18.

```

root@klien: /home/klien
cp/recv?file=/521KOMA6MB.mkv length 55955879
1 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtnv [dtn://klien.dtn/dtn/recv?file=*]...
cp/recv?file=/521KOMA6MB.mkv length 338415
2 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtn connection to dtn router...
cp/recv?file=/521KOMA6MB.mkv length 521643214 (NOT PENDING) [dtn://klien.dtn/dtn/recv?file=*]
)
3 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtn
cp/recv?file=/521KOMA6MB.mkv length 1414938
4 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtn
cp/recv?file=/521KOMA6MB.mkv length 5648684
5 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtn
cp/recv?file=/521KOMA6MB.mkv length 2495802
6 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtn payload to final filename: 'mv /tmp/bundlePayload
cp/recv?file=/521KOMA6MB.mkv length 9234203
7 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtnv [dtn://klien.dtn/dtn/recv?file=*]...
cp/recv?file=/521KOMA6MB.mkv length 25875
8 : dtn://fitri.dtn/dtn/recv?source=/home/fitri...
/Documents/KIRIMDATA/521KOMA6MB.mkv -> dtn://klien.dtn/dtn connection to dtn router...
cp/recv?file=/521KOMA6MB.mkv length 1647895 (NOT PENDING) [dtn://klien.dtn/dtn/recv?file=*]
klien dtn%

```

Gambar 4.19. Bundle Antrian

4.2.5. Perhitungan Kondisi Pengujian apakah LoS, n-LoS, atau N-LoS

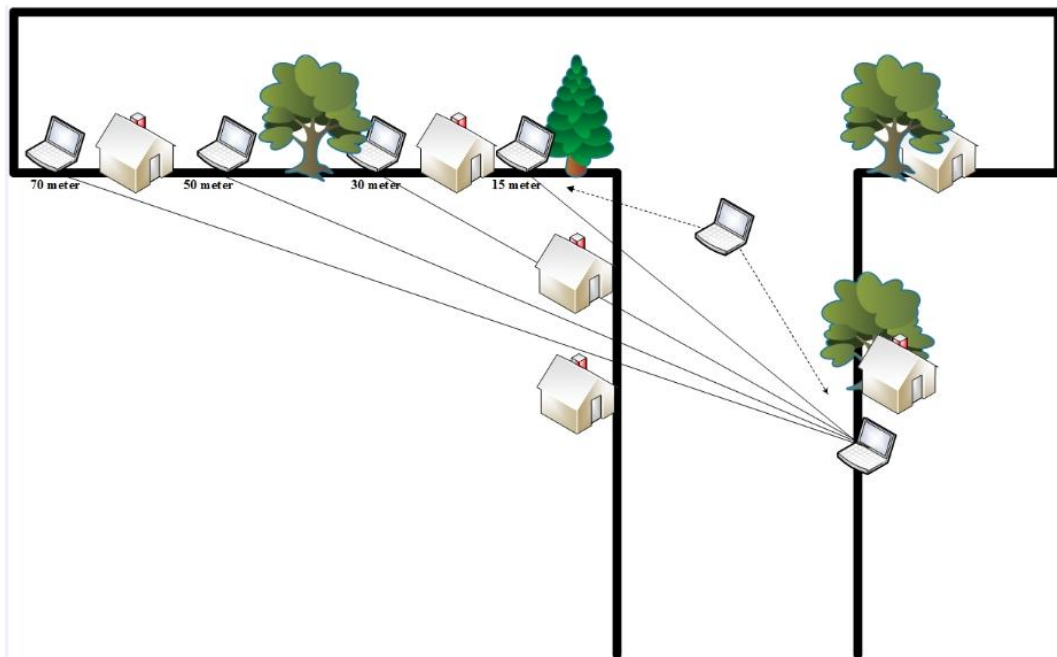
Pengujian dilakukan dalam 2 kondisi yaitu kondisi LoS (*Line of Sight*) dan kondisi N-LoS (*Non Line of Sight*). Pada kondisi LoS, dilakukan di lapangan terbuka yang mencakup luas 90 meter x 40 meter. Sedangkan kondisi N-LoS, dilakukan di daerah perumahan dengan adanya penghalang seperti rumah, pohon, dan lain-lain.

Adapun kategori yang harus terpenuhi agar kondisi dapat dikatakan LoS, n-LoS, ataupun N-LoS dalam sudut pandang adanya halangan/penghalang dari antara server ke routernya adalah sebagai berikut

1. Jika daerah yang terhalang < 40 %, daerah tersebut masuk dalam kategori LoS.
2. Jika dalam kondisi $40\% < \text{n-LoS} < 70\%$, maka daerah tersebut masuk dalam kategori *near-LoS*.

3. Jika tertutup > 70%, maka kondisi tersebut sudah masuk dalam kondisi Non-LoS.

Untuk mengetahui apakah daerah tersebut n-LoS ataupun sudah dalam keadaan N-LoS, bisa dilakukan perhitungan dengan mencari *Fresnel Zone* nya. Untuk melihat keadaan pada saat pengambilan data dalam kondisi Non-LoS dapat dilihat pada gambar 4.20.



Gambar 4.20. Skenario Pengambilan Data Pada Kondisi N-LoS

Berikut perhitungan *Fresnel zone* yang dilakukan pada setiap jarak ujinya yaitu pada jarak 15 meter, 30 meter, 50 meter dan 70 meter.

- *Fresnel Zone* pengujian pada jarak 15 meter

Diketahui : $d1 = 10 \text{ m} = 0,01 \text{ km}$

$d2 = 5 \text{ m} = 0,005 \text{ km}$

$D = 15 \text{ m} = 0,015 \text{ km}$

$f = 2,4 \text{ GHz}$

lebar daerah singgung ke penghalang = 3,8 meter = 380 cm

Maka, F pada 15 meter adalah

$$F_{15} = 17,3 \times \sqrt{\frac{d1 \times d2}{f \times D}}$$

$$\begin{aligned}
&= 17,3 \times \sqrt{\frac{0,01 \times 0,005}{2,4 \times 0,015}} \\
&= 17,3 \times \sqrt{\frac{0,00005}{0,036}} \\
&= 17,3 \times \sqrt{0,001388888888889} \\
&= 17,3 \times 0,0372677 \\
&= 0,644 \text{ m}
\end{aligned}$$

$$F_{15} = 60 \text{ cm}$$

maka, untuk daerah Fresnel Zone dalam jarak 15 meter, daerah terhalang sudah lebih dari 70% menghalangi daerah *fresnel zone* nya ($380 \text{ cm} > 60 \text{ cm}$).

- *Fresnel Zone* pengujian pada jarak 30 meter

Diketahui : $d_1 = 20 \text{ m} = 0,02 \text{ km}$

$$d_2 = 10 \text{ m} = 0,01 \text{ km}$$

$$D = 30 \text{ m} = 0,03 \text{ km}$$

$$f = 2,4 \text{ GHz}$$

$$\text{lebar daerah singgung ke penghalang} = 5,5 \text{ meter} = 550 \text{ cm}$$

Maka, F pada 30 meter adalah

$$\begin{aligned}
F_{15} &= 17,3 \times \sqrt{\frac{d_1 \times d_2}{f \times D}} \\
&= 17,3 \times \sqrt{\frac{0,02 \times 0,01}{2,4 \times 0,03}} \\
&= 17,3 \times \sqrt{\frac{0,0002}{0,072}} \\
&= 17,3 \times \sqrt{0,0027777777778} \\
&= 17,3 \times 0,05270462 \\
&= 0,911790 \text{ m}
\end{aligned}$$

$$F_{30} = 91,179 \text{ cm}$$

maka, untuk daerah Fresnel Zone dalam jarak 30 meter, daerah terhalang sudah lebih dari 70% menghalangi daerah *fresnel zone* nya ($550 \text{ cm} > 91,179 \text{ cm}$).

- *Fresnel Zone* pengujian pada jarak 50 meter

Diketahui : $d_1 = 27,5 \text{ m} = 0,0275 \text{ km}$

$$d_2 = 22,5 \text{ m} = 0,0225 \text{ km}$$

$$D = 50 \text{ m} = 0,05 \text{ km}$$

$$f = 2,4 \text{ GHz}$$

$$\text{lebar daerah singgung ke penghalang} = 7,2 \text{ meter} = 720 \text{ meter}$$

Maka, F pada 30 meter adalah

$$\begin{aligned} F_{15} &= 17,3 \times \sqrt{\frac{d_1 \times d_2}{f \times D}} \\ &= 17,3 \times \sqrt{\frac{0,0275 \times 0,0225}{2,4 \times 0,05}} \\ &= 17,3 \times \sqrt{\frac{0,00061875}{0,12}} \\ &= 17,3 \times \sqrt{0,00515625} \\ &= 17,3 \times 0,0718 \\ &= 0,51198414 \text{ m} \end{aligned}$$

$$F_{50} = 51,198414 \text{ cm}$$

maka, untuk daerah Fresnel Zone dalam jarak 50 meter, daerah terhalang sudah lebih dari 70% menghalangi daerah *fresnel zone* nya ($720 \text{ cm} > 51,198414 \text{ cm}$).

- *Fresnel Zone* pengujian pada jarak 70 meter

Diketahui : $d_1 = 39 \text{ m} = 0,039 \text{ km}$

$$d_2 = 31 \text{ m} = 0,031 \text{ km}$$

$$D = 70 \text{ m} = 0,07 \text{ km}$$

$$f = 2,4 \text{ GHz}$$

$$\text{lebar daerah singgung ke penghalang} = 8,8 \text{ meter} = 880 \text{ cm}$$

Maka, F pada 30 meter adalah

$$\begin{aligned} F_{15} &= 17,3 \times \sqrt{\frac{d_1 \times d_2}{f \times D}} \\ &= 17,3 \times \sqrt{\frac{0,039 \times 0,031}{2,4 \times 0,07}} \\ &= 17,3 \times \sqrt{\frac{0,001209}{0,168}} \\ &= 17,3 \times \sqrt{0,007196428} \\ &= 17,3 \times 0,0848317 \end{aligned}$$

$$= 1,4675 \text{ m}$$

$$F_{70} = 146,75 \text{ cm}$$

maka, untuk daerah Fresnel Zone dalam jarak 70 meter, daerah terhalang sudah lebih dari 70% menghalangi daerah *fresnel zone* nya ($880 \text{ cm} > 146,75 \text{ cm}$).

Hasil perhitungan tersebut menunjukkan bahwa pada kondisi tersebut merupakan termasuk dalam kondisi Non-LoS dimana lebar daerah yang menghalangi (lebar daerah singgung ke penghalang) lebih dari 70% daerah *fresnel zone* nya.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan pengujian dan analisis data yang diperoleh dari pengujian skenario penelitian DTN, dapat disimpulkan bahwa :

1. Sistem DTN ini dapat digunakan sebagai solusi alternatif di daerah yang *digital divide*, yaitu daerah yang tidak menjangkau infrastruktur jaringan komunikasi internet.
2. *Custody Transfer* berperan penting sebagai penanggung jawab untuk pemeliharaan koneksi *end-to-end* sehingga data bisa sampai ke *node* tujuan.
3. *Log bundle* DTN berisi log data hasil monitoring dari berjalannya sistem DTN.
4. Pada kondisi LoS dengan jarak 15 meter, data uji terkirim semua. Namun pada jarak 30 meter hanya terkirim dengan data berukuran 1700 MB dengan waktu transfer data adalah 3087 detik atau 51 menit 27 detik, sedangkan pada jarak 50 meter data uji yang sampai ke tujuan yaitu 1000 MB, dengan waktu yang dibutuhkan untuk transfer data adalah 3629 detik atau 1 jam 29 detik dan pada jarak 70 meter maksimal data uji yang sampai ke tujuan yaitu 750 MB dengan waktu yang dibutuhkan yaitu 2467 detik. Sedangkan pada kondisi N-LoS dengan jarak 15 meter, data yang terkirim dengan data uji 1500 MB dengan waktu 2805 detik. Namun pada jarak 30 meter hanya terkirim dengan data berukuran 1200 MB dengan waktu transfer data adalah 2551 detik, sedangkan pada jarak 50 meter data uji yang sampai ke tujuan yaitu 750 MB, dengan waktu yang dibutuhkan untuk transfer data adalah 2820 detik dan pada jarak 70 meter maksimal data uji yang sampai ke tujuan hanya 200 MB dengan waktu yang dibutuhkan yaitu 2894 detik atau 48 menit 14 detik.

5. Pada pengujian N-LoS memerlukan lebih banyak waktu untuk sampai ke klien dibanding dengan pengujian pada kondisi LoS dan ukuran data uji pada kondisi N-LoS yang sampai ke klien juga lebih kecil daripada pengujian LoS pada setiap jarak pengujiannya.
6. Hasil pengujian dan pengambilan data menunjukkan bahwa pada saat transfer data dilakukan memiliki kecenderungan semakin besar ukuran data/file yang dikirimkan maka waktu yang dibutuhkan untuk mengirim file tersebut juga akan semakin besar.

5.2. Saran

Dari penelitian yang telah dilakukan, saran yang dapat diberikan untuk pengembangan penelitian sistem DTN ini adalah sebagai berikut:

1. Diperlukan penelitian lebih jauh dengan skenario sistem DTN yang lebih kompleks.
2. Menerapkan aplikasi tambahan yang mendukung pengiriman data menggunakan sistem DTN.
3. Melakukan penelitian sistem DTN dalam kondisi n-LoS (*near Line of Sight*).
4. Diperlukan pengujian sistem DTN dengan penambahan klien dalam jumlah besar, agar hasil yang didapat lebih maksimal.

DAFTAR PUSTAKA

- [1] K. Fall. 2003. **A delay-tolerant network architecture for challenged internets**” *Proc. 2003 Conf. Appl. Technol. Archit. Protoc. Comput. Commun. - SIGCOMM '03*, p. 27.
- [2] K. Fall. **Custody Transfer for Reliable Delivery in Delay Tolerant Networks**.
- [3] S. Burleigh, A. Hooke, L. Torgerson, B. Durst, K. Scott, and T. M. Corporation. 2003. **Delay-Tolerant Networking: An Approach to Interplanetary Internet**. IEEE. no. June. pp. 128–136.
- [4] Siswanti, Sri Desy. 2013. **Pengembangan Sistem Aplikasi Pengiriman Data Daerah Terpencil Berbasis Delay Tolerant Network**. Jurnal Generic, vol. 8, no. 2, pp. 238–253.
- [5] M. Naisuty, A. Urip, A. Wibowo, R. Suhatman. 2012. **Analisa Kinerja Protokol TCP/IP dan DTN Pada Jaringan Multi Jalur**. vol. 1. no. September, pp. 1–6.
- [6] E. M. Husni and A. R. Sumarmo. 2011. **Delay Tolerant Network Utilizing Train for News Portal and Email Services**.
- [7] E. Husni and J. Waworundeng. 2011. **Telemedicine System Based On Delay Tolerant Network**. no. July, pp. 1–5.
- [8] L. Amalia, R. 2013. **Penerapan Delay Tolerant Network (DTN) untuk Sistem Konsultasi Kesehatan Jarak Jauh Berbasis Web**. pp. 1–8.
- [9] K. Fall. 2003. **A Delay-Tolerant Network Architecture for Challenged Internets**. Intel Research Technical Report, IRB TR-03.
- [10] J. N. Isento, J. A. Dias, J. J. P. C. Rodrigues, M. Chen, and K. Lin. 2011. **The Effect of Bundle Aggregation on the Performance of Vehicular Delay - Tolerant Networks**. pp. 813–818.
- [11] K. Fall, S. Member, and S. Farrell. 2008. **DTN: An Architectural Retrospective**. vol. 26, no. 5, pp. 828–836.
- [12] J. Banumathi. 2014. **Survey Paper on the Performance Analysis of Routing Protocols in DTN**. no. March, pp. 13–18.

- [13] J. Seguí and E. Jennings. 2006. **Delay Tolerant Networking – Bundle Protocol Simulation**. IEEE.
- [14] R. Patra and S. Nedevski. 2003. **DTNLite : A Reliable Data Transfer Architecture for Sensor Networks**,.
- [15] E. Koutsogiannis, F. Tsapeli, and V. Tsaoussidis. 2011. **Bundle Layer End-to-end Retransmission Mechanism**. pp. 109–115.
- [16] T. Jonson, J. Pezeshki, V. Chao, K. Smith, and J. Fazio. 2008. **Application of Delay Tolerant Networking (DTN) In Airbone Networks**. IEEE. pp. 1–7.
- [17] J. Segui, E. Jennings, J. S. Segui, and E. H. Jennings. 2006. **Delay Tolerant Networking - Bundle Protocol Simulation**. IEEE.
- [18] M. Chuah, P. Yang, B. D. Davison, and L. Cheng. **Store-and-Forward Performance in a DTN**.
- [19] M. Alfonzo, J. A. Fraire, and E. Kocian. 2014. **Development of a DTN Bundle Protocol Convergence Layer for SpaceWire**. pp. 0–5.
- [20] M. Seligman. **Storage Usage of Custody Transfer in Delay Tolerant Networks with Intermittent Connectivity**.
- [21] Warthman. 2003. **Delay Tolerants Network (DTNs)**. versi 1.1 hal.3.

LAMPIRAN

LOKASI PENGUJIAN

1. Lokasi Pengujian LoS dalam Jarak 15 Meter dan 30 Meter



2. Lokasi Pengujian LoS dalam Jarak 50 Meter dan 70 Meter



3 . Lokasi Pengujian n-LoS (*Non-Line of Sight*)



Program autoconnech.sh

```
while true
do
    sudo ifconfig eth1:1 192.168.10.1
    echo
    echo
    "=====
=====
    echo " set IP alias : "
    sudo ifconfig eth1:1 | grep "inet addr"
    echo " -----
----- "
    echo " mencari Access Point          "
    echo
    "=====
=====
    echo
    if ping -c 1 -W1 192.168.20.1
    then
        echo
        echo "=====
        date +%d-%b-%Y_%R:%S,%N
        echo "-----
        echo " Access Point ditemukan"
        echo "-----
        echo " Terhubung ke Access Point"
        echo "=====
        echo

        echo
        sudo ifconfig eth1 192.168.10.1
        echo
    "=====
=====
    echo " Set IP Router : "
    sudo ifconfig eth1 | grep "inet addr"
    echo " -----
----- "
    echo " Cek Koneksi ke Server klien"
    echo
    "=====
=====
    echo
    while ping -c 1 -W1 192.168.20.12
    do
        echo
        echo "=====
        date +%d-%b-%Y_%R:%S,%N
        echo "-----"
```

```

        echo " terhubung ke Server klien_1"
        echo " IP klien_1 : 192.168.20.12"
        echo " IP Router   : 192.168.10.1"
        echo "===== "
        echo
            sleep 2
        done
        echo
"===== "
        echo " Koneksi ke Server klien_1 Tidak
Tersedia/Terputus"
        echo
"===== "
        echo

        echo
        echo "===== "
        echo " Koneksi Tidak Tersedia
        echo " ----- "
        echo " Tidak Dapat Terhubung ke Server"
        echo "===== "
        echo

    else
        echo "===== "
        echo " Access Point Tidak Ditemukan"
        echo "===== "
        echo
    fi

    sleep 1

done

```

TAMPILAN DTN

1. Tampilan dtnping

```
root@kom-2: /home/fitri
PING [dtn://putri.dtn/ping] (expiration 30)...
^C
root@kom-2: /home/fitri# dtnping -c 6 dtn://putri.dtn/ping
source_eid [dtn://fitri.dtn/ping.2953]
dtn_register succeeded, regid 12
PING [dtn://putri.dtn/ping] (expiration 30)...
20 bytes from [dtn://putri.dtn/ping]: 'dtnping!' seqno=0, time=516 ms
20 bytes from [dtn://putri.dtn/ping]: 'dtnping!' seqno=1, time=516 ms
20 bytes from [dtn://putri.dtn/ping]: 'dtnping!' seqno=2, time=494 ms
20 bytes from [dtn://putri.dtn/ping]: 'dtnping!' seqno=3, time=495 ms
20 bytes from [dtn://putri.dtn/ping]: 'dtnping!' seqno=4, time=408 ms
20 bytes from [dtn://putri.dtn/ping]: 'dtnping!' seqno=5, time=485 ms
root@kom-2: /home/fitri# dtnping -c 6 dtn://klien.dtn/ping
source_eid [dtn://fitri.dtn/ping.2959]
dtn_register succeeded, regid 13
PING [dtn://klien.dtn/ping] (expiration 30)...
^C
root@kom-2: /home/fitri# dtnping -c 6 dtn://klien.dtn/ping
source_eid [dtn://fitri.dtn/ping.2985]
dtn_register succeeded, regid 10
PING [dtn://klien.dtn/ping] (expiration 30)...
20 bytes from [dtn://klien.dtn/ping]: 'dtnping!' seqno=0, time=3235 ms
20 bytes from [dtn://klien.dtn/ping]: 'dtnping!' seqno=1, time=2300 ms
20 bytes from [dtn://klien.dtn/ping]: 'dtnping!' seqno=2, time=1178 ms
20 bytes from [dtn://klien.dtn/ping]: 'dtnping!' seqno=3, time=562 ms
20 bytes from [dtn://klien.dtn/ping]: 'dtnping!' seqno=4, time=529 ms
20 bytes from [dtn://klien.dtn/ping]: 'dtnping!' seqno=5, time=516 ms
root@kom-2: /home/fitri#
```

2. Tampilan Koneksi Wifi Otomatis

```
root@kom-1: /home/putri
IP klien_1 : 192.168.20.12
IP Router  : 192.168.10.1
=====

PING 192.168.20.12 (192.168.20.12) 56(84) bytes of data.
64 bytes from 192.168.20.12: icmp_seq=1 ttl=64 time=3.02 ms

--- 192.168.20.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.024/3.024/3.024/0.000 ms

=====
27-Dec-2015_20:35:59,918778025
-----
terhubung ke Server klien_1
IP klien_1 : 192.168.20.12
IP Router  : 192.168.10.1
=====

PING 192.168.20.12 (192.168.20.12) 56(84) bytes of data.
64 bytes from 192.168.20.12: icmp_seq=1 ttl=64 time=7.54 ms

--- 192.168.20.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.549/7.549/7.549/0.000 ms

=====
27-Dec-2015_20:36:00,932885972
-----
terhubung ke Server klien_1
IP klien_1 : 192.168.20.12
IP Router  : 192.168.10.1
=====
```

3. Tampilan Koneksi Wifi Otomatis saat Koneksi ke Klien Terputus

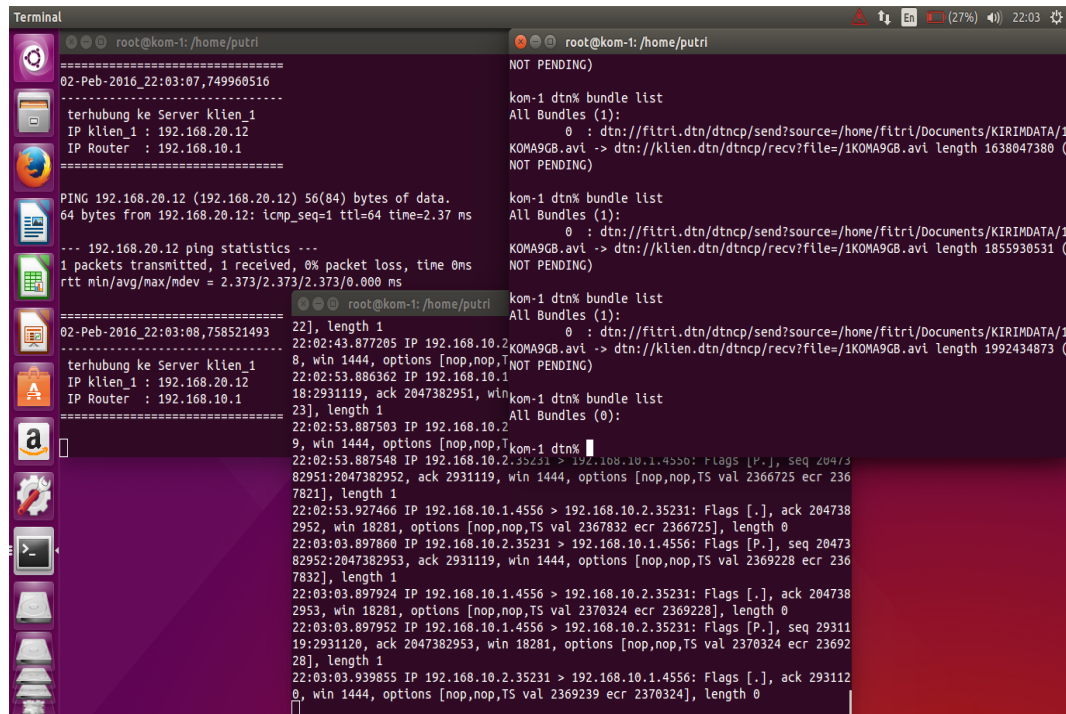
```
root@kom-1: /home/putri
=====
PING 192.168.20.12 (192.168.20.12) 56(84) bytes of data.
--- 192.168.20.12 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
=====
Koneksi ke Server klien_1 Tidak Tersedia/Terputus
=====
Koneksi Tidak Tersedia
-----
Tidak Dapat Terhubung ke Server
=====
root@kom-1: /home/putri
```

4. Tampilan Komputer Server saat Proses Pengujian dan Pengambilan Data

```
terminal
root@kom-2: /home/fitri
invoked from within
"uplevel \#0 $command($input)"
kon-2 dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/5KOMA3MB.mp3
  1 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA27GB.avi
kon-2 dtn% ^C
root@kom-2: /home/fitri# dtnd -t -c /etc/dtn.conf -l info -o /home/fitri/dtn/log/cubocubo5m.log
kon-2 dtn% ^C
root@kom-2: /home/fitri# dtnd -t -c /etc/dtn.conf -l info -o /home/fitri/dtn/log/1KOMA27GB.log
kon-2 dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/5KOMA3MB.mp3
  1 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA27GB.avi
kon-2 dtn% bundle list
All Bundles (0):
kon-2 dtn%

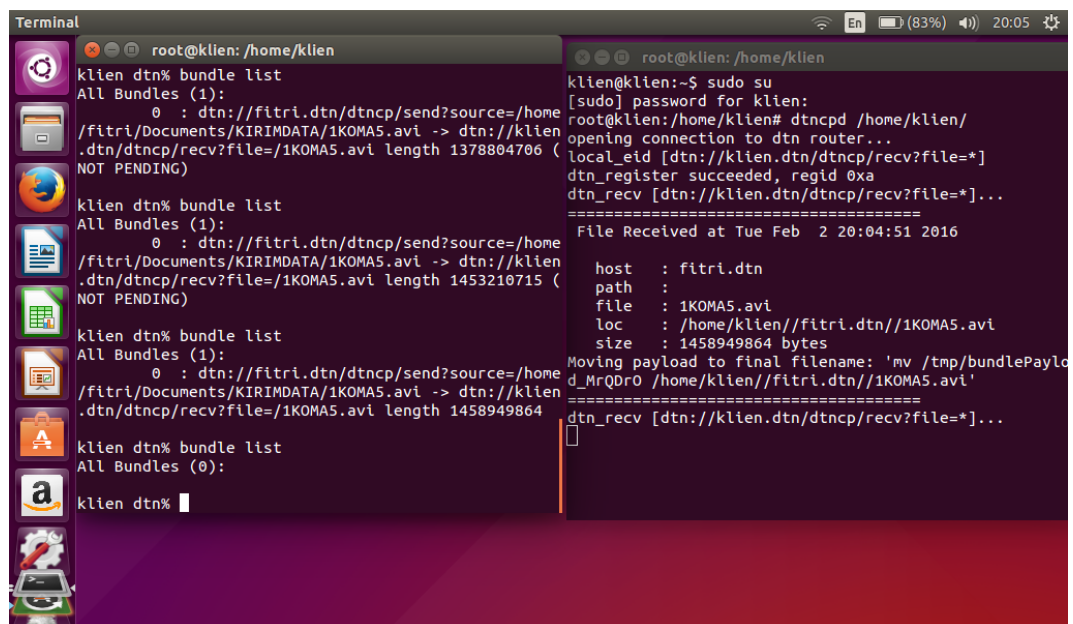
root@kom-2: /home/fitri
5KOMA3MB.mp3 dtn://klien.dtn
Opening connection to local DTN daemon
source_eid [dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/5KOMA3MB.mp3]
dest_eid [dtn://klien.dtn/dtn/recv?file=5KOMA3MB.mp3]
file sent to [dtn://klien.dtn/dtn/recv?file=5KOMA3MB.mp3]
root@kom-2: /home/fitri# dtnd --expiration 3600 /home/fitri/Documents/KIRIMDATA/5KOMA3MB.mp3 dtn://klien.dtn
Opening connection to local DTN daemon
source_eid [dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/5KOMA3MB.mp3]
dest_eid [dtn://klien.dtn/dtn/recv?file=5KOMA3MB.mp3]
file sent to [dtn://klien.dtn/dtn/recv?file=5KOMA3MB.mp3]
root@kom-2: /home/fitri# dtnd --expiration 3600 /home/fitri/Documents/KIRIMDATA/1KOMA27GB.avi dtn://klien.dtn
Opening connection to local DTN daemon
source_eid [dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA27GB.avi]
dest_eid [dtn://klien.dtn/dtn/recv?file=1KOMA27GB.avi]
file sent to [dtn://klien.dtn/dtn/recv?file=1KOMA27GB.avi]
root@kom-2: /home/fitri#
```

5. Tampilan Komputer Router saat Proses Pengujian dan Pengambilan Data



```
Terminal
root@kom-1: /home/putri
=====
02-Feb-2016 22:03:07,749960516
-----
terhubung ke Server klien_1
IP klien_1 : 192.168.20.12
IP Router : 192.168.10.1
=====
PING 192.168.20.12 (192.168.20.12) 56(84) bytes of data.
64 bytes from 192.168.20.12: icmp_seq=1 ttl=64 time=2.37 ms
--- 192.168.20.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.373/2.373/2.373/0.000 ms
=====
02-Feb-2016 22:03:08,758521493
-----
terhubung ke Server klien_1
IP klien_1 : 192.168.20.12
IP Router : 192.168.10.1
=====
kon-1 dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA9GB.avi -> dtn://klien.dtn/dtn/recv?file=/1KOMA9GB.avi length 1638047380 (NOT PENDING)
kon-1 dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA9GB.avi -> dtn://klien.dtn/dtn/recv?file=/1KOMA9GB.avi length 1855930531 (NOT PENDING)
kon-1 dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA9GB.avi -> dtn://klien.dtn/dtn/recv?file=/1KOMA9GB.avi length 1992434873 (NOT PENDING)
kon-1 dtn% bundle list
All Bundles (0):
22:02:43.877205 IP 192.168.10.2 > 192.168.10.1.4556: Flags [P.], seq 2047382951, win 1444, options [nop,nop,TS val 2366725 ecr 2367821], length 1
22:02:53.886362 IP 192.168.10.1 > 192.168.10.2.35231: Flags [P.], seq 2047382952, win 18281, options [nop,nop,TS val 2367832 ecr 2366725], length 0
22:03:03.897860 IP 192.168.10.2.35231 > 192.168.10.1.4556: Flags [P.], seq 2047382952, win 1444, options [nop,nop,TS val 2369228 ecr 2367832], length 1
22:03:03.897924 IP 192.168.10.1.4556 > 192.168.10.2.35231: Flags [P.], seq 2047382953, win 18281, options [nop,nop,TS val 2370324 ecr 2369228], length 0
22:03:03.897952 IP 192.168.10.1.4556 > 192.168.10.2.35231: Flags [P.], seq 2931119, win 1444, options [nop,nop,TS val 2370324 ecr 2369228], length 1
22:03:03.939855 IP 192.168.10.2.35231 > 192.168.10.1.4556: Flags [P.], seq 2931119, win 1444, options [nop,nop,TS val 2369239 ecr 2370324], length 0
```

6. Tampilan Komputer Klien saat Proses Pengujian dan Pengambilan Data



```
Terminal
root@klien: /home/klien
klien dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA5.avi -> dtn://klien.dtn/dtn/recv?file=/1KOMA5.avi length 1378804706 (NOT PENDING)
klien dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA5.avi -> dtn://klien.dtn/dtn/recv?file=/1KOMA5.avi length 1453210715 (NOT PENDING)
klien dtn% bundle list
All Bundles (1):
  0 : dtn://fitri.dtn/dtn/recv?source=/home/fitri/Documents/KIRIMDATA/1KOMA5.avi -> dtn://klien.dtn/dtn/recv?file=/1KOMA5.avi length 1458949864
klien dtn% bundle list
All Bundles (0):
klien dtn%

root@klien: /home/klien
klien@klien:~$ sudo su
[sudo] password for klien:
root@klien: /home/klien# dtncpd /home/klien/
opening connection to dtn router...
local_eid [dtn://klien.dtn/dtn/recv?file=]
dtn_register succeeded, regid 0xa
dtn_rcv [dtn://klien.dtn/dtn/recv?file=]...
=====
File Received at Tue Feb 2 20:04:51 2016
host : fitri.dtn
path :
file : 1KOMA5.avi
loc : /home/klien/fitri.dtn/1KOMA5.avi
size : 1458949864 bytes
Moving payload to final filename: 'mv /tmp/bundlePayload_MrQDrO /home/klien/fitri.dtn/1KOMA5.avi'
=====
dtn_rcv [dtn://klien.dtn/dtn/recv?file=]...
```

SAMPLE LOG BUNDLE DTN

1. Log Bundle Pada Server Data

```
[1454421870.459144 /dtnd notice] random seed is 459126
[1454421870.459328 /dtnd notice] DTN daemon starting up...
(pid 2325)
[1454421870.480171 /dtnd info] parsing configuration file
/etc/dtn.conf...
[1454421870.488812 /dtnd info] dtnd parsing configuration...
[1454421870.504166 /dtn/interface/table info] adding
interface tcp0 (tcp)
[1454421870.504564 /dtn/link/link1 info] new link link1
[192.168.10.1 dtn:none ONDEMAND tcp state=AVAILABLE]
[1454421870.504769 /dtnd info] dtnd configuration parsing
complete
[1454421870.504834 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 3 SECONDS
[1454421871.505014 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 2 SECONDS
[1454421872.505267 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 1 SECONDS
[1454421873.505529 /dtn/storage notice] tidy option removing
directory '/bin/rm -rf /var/dtn/db'
[1454421873.609522 /dtn/storage notice] creating new
database directory /var/dtn/db
[1454421873.609898 /dtn/storage info] initializing db
name=DTN (not shared), dir=/var/dtn/db
[1454421873.610840 /dtn/storage info] datastore /var/dtn/db
was cleanly shut down
[1454421873.610901 /dtnd notice] tidy option removing
directory '/bin/rm -rf /var/dtn/bundles'
[1454421873.770155 /dtn/storage/globals info] initializing
global table
[1454421874.178256 /S10 notice] Lines marked "S10" are log
lines added to make parsing
[1454421874.178325 /S10 notice] the logs easier after
extended tests. These were added for
[1454421874.178343 /S10 notice] the summer 2010 N4C summer
trial and are mainly new lines
[1454421874.178358 /S10 notice] that allow tracing bundles
across multiple hops.
[1454421874.178374 /S10 notice] To extract these do the
following...
[1454421874.178389 /S10 notice]   grep "S10 info" <dtnd-log-
files> | grep -v grep | awk '{print $4,$5}' | sort -n
[1454421874.178623 /S10 info]
rfc3339,time_t,usec,node,event,size,src,id,dest,expiry,flags
,peer,ob-src,ob-id,otime.ousec,conttype,duration,cmt
```

```

[1454421874.178658 /S10 info] 2016-02-02
14:04:34,1454421874.178403,dtn://fitri.dtn,STARTING,0,,,,,,,,
,,0.000000,,0,
[1454421874.179318 /dtn/bundle/daemon notice] loading
bundles from data store
[1454421874.179430 /dtn/bundle/daemon info]
REGISTRATION_ADDED 0 dtn://fitri.dtn
[1454421874.179513 /dtn/bundle/daemon info]
REGISTRATION_ADDED 2 dtn://fitri.dtn/ping
[1454421874.179629 /dtn/bundle/daemon/loop warning] event
LINK_CREATED was in queue for 3.674984 seconds
[1454421874.188719 /dtn/bundle/daemon info] LINK_CREATED
link1 [192.168.10.1 dtn:none ONDEMAND tcp state=AVAILABLE]
[1454421874.188953 /dtn/bundle/daemon/loop warning] event
ROUTE_ADD was in queue for 3.684233 seconds
[1454421874.188984 /dtn/bundle/daemon info] ROUTE_ADD
dtn://putri.dtn/* -> link1 (FORWARD)
[1454421874.189037 /dtn/bundle/daemon/loop warning] event
ROUTE_ADD was in queue for 3.684288 seconds
[1454421874.189058 /dtn/bundle/daemon info] ROUTE_ADD
dtn://klien.dtn/* -> link1 (FORWARD)
[1454421874.189132 /dtn/bundle/daemon info] LINK_AVAILABLE
link1 [192.168.10.1 dtn:none ONDEMAND tcp state=AVAILABLE]
[1454421874.679266 /dtnd info] starting console on
127.0.0.1:5050
[1454421875.656742 /dtn/tclcmd info] can't load
oasys_tclreadline: can't find package oasys_tclreadline
[1454421875.656842 /dtn/tclcmd info] fall back to simple
command loop
[1454421900.583502 /dtn/apiclient/25 info] new session
127.0.0.1:5010 -> 127.0.0.1:45647
[1454421900.627917 /dtn/bundle/daemon info]
REGISTRATION_ADDED 10
dtn://fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIRI
MADATA/1KOMA9GB.avi
[1454421900.627990 /dtn/registration/table info] adding
registration
10/dtn://fitri.dtn/dtnctp/send?source=/home/fitri/Documents/K
IRIMDATA/1KOMA9GB.avi
[1454421900.683658 /dtn/apiclient/25 info] spec: 0 0
[1454421900.684004 /dtn/bundle/daemon info]
BUNDLE_ACCEPT_REQUEST: bundle bundle id 0
[dtn://fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIR
IMDATA/1KOMA9GB.avi ->
dtn://klien.dtn/dtnctp/recv?file=/1KOMA9GB.avi 2045884082
byte payload] accepted (reason no additional information)
[1454422001.732697 /dtn/apiclient/25 info] DTN_SEND bundle
bundle id 0
[dtn://fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIR
IMDATA/1KOMA9GB.avi ->
dtn://klien.dtn/dtnctp/recv?file=/1KOMA9GB.avi 2045884082
byte payload

```



```

[1454422004.201206 /dtm/bundle/protocol info]
handle_bundle_received: begin
[1454422004.201301 /S10 info] 2016-02-02
14:06:44,1454422004.201276,dtm://fitri.dtm,FROMAPP,204588408
2,dtm://fitri.dtm/dtncp/send?source=/home/fitri/Documents/KI
RIMDATA/1KOMA9GB.avi,507737100.0,dtm://klien.dtm/dtncp/recv?
file=/1KOMA9GB.avi,3600,,dunno,,,0.000000,,0,
[1454422004.201331 /dtm/bundle/daemon info] BUNDLE_RECEIVED
(from app) bundle id 0
[dtm://fitri.dtm/dtncp/send?source=/home/fitri/Documents/KIR
IMDATA/1KOMA9GB.avi ->
dtm://klien.dtm/dtncp/recv?file=/1KOMA9GB.avi 2045884082
byte payload] prevhop dtm:none (0 bytes recvd)
[1454422004.201359 /dtm/bundle/expiration info] adding
bundle to pending list
[1454422004.234741 /dtm/bundle/protocol info]
BundleDaemon::handle_bundle_received: end
[1454422004.234961 /dtm/contact/link1 info] new contact
contact 192.168.10.1 (started 1454422004.234959)
[1454422004.235792 /dtm/apiclient/25 info] received
DTM_CLOSE message; closing API handle
[1454422004.238623 /dtm/bundle/daemon info]
REGISTRATION_EXPIRED 10
[1454422004.238676 /dtm/registration/table info] removing
registration 10
[1454422004.253701 /dtm/bundle/daemon info]
REGISTRATION_DELETE 10
[1454422004.253871 /dtm/bundle/daemon info] CONTACT_UP link1
[192.168.10.1 dtm://putri.dtm ONDEMAND tcp state=OPENING]
(contact 0x7f464c005e20)
[1454422004.253920 /S10 info] 2016-02-02
14:06:44,1454422004.253905,dtm://fitri.dtm,CONTUP,0,,,,,192
.168.10.1,,,0.000000,ONDEMAND,0,
[1454422004.254099 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks begin
[1454422004.254180 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks looking for processors
[1454422004.254201 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks age block outbound is
DISABLED
[1454422004.254217 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks age block inbound support is
ENABLED
[1454422004.254233 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks age block zero creation is
ENABLED
[1454422004.254259 /dtm/bundle/extblock/aeb info] prepare()
begin
[1454422004.254276 /dtm/bundle/extblock/aeb info]
age_outbound_enabled != true, no AEB, returning BP_FAIL
[1454422004.254291 /dtm/bundle/extblock/aeb info] prepare()
end

```



```

[1454422004.254329 /dtn/bundle/protocol info]
BundleProtocol::prepare_blocks end
[1454422004.254346 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: calling generate() on each
block processor
[1454422004.254364 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x0
type 0x0) data_offset 0 data_length 0 contents length 0
[1454422004.254386 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x0
type 0x0) data_offset 0 data_length 0 contents length 0
[1454422004.254411 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x1
type 0x1) data_offset 7 data_length 2045884082 contents
length 7
[1454422004.254430 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x1
type 0x1) data_offset 7 data_length 2045884082 contents
length 7
[1454422004.254450 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: calling generate() on
PRIMARY block
[1454422004.254535 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: calling finalize() on all
blocks
[1454422004.254558 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: end
[1454422950.975836 /dtn/bundle/daemon info]
BUNDLE_TRANSMITTED id:0 (2045884236 bytes_sent/2045884236
reliable) -> link1 (192.168.10.1)
[1454422950.975933 /S10 info] 2016-02-02
14:22:30,1454422950.975912,dtn://fitri.dtn,TX,2045884082,dtn
://fitri.dtn/dtncp/send?source=/home/fitri/Documents/KIRIMDA
TA/1KOMA9GB.avi,507737100.0,dtn://klien.dtn/dtncp/recv?file=
/1KOMA9GB.avi,3600,,192.168.10.1,,,0.000000,,0,

```

2. Log Bundle Wireless Router

```
[1453350136.412060 /dtnd notice] random seed is 412042
[1453350136.412221 /dtnd notice] DTN daemon starting up...
(pid 12425)
[1453350136.454222 /dtnd info] parsing configuration file
/etc/dtn.conf...
[1453350136.465649 /dtnd info] dtnd parsing configuration...
[1453350136.466800 /dtn/interface/table info] adding
interface tcp0 (tcp)
[1453350136.467139 /dtn/link/link1 info] new link link1
[192.168.10.2 dtn:none ONDEMAND tcp state=AVAILABLE]
[1453350136.467298 /dtn/link/link2 info] new link link2
[192.168.20.12 dtn:none ONDEMAND tcp state=AVAILABLE]
[1453350136.467465 /dtnd info] dtnd configuration parsing
complete
[1453350136.467527 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 3 SECONDS
[1453350137.467671 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 2 SECONDS
[1453350138.467935 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 1 SECONDS
[1453350139.468160 /dtn/storage notice] tidy option removing
directory '/bin/rm -rf /var/dtn/db'
[1453350139.524586 /dtn/storage notice] creating new
database directory /var/dtn/db
[1453350139.524927 /dtn/storage info] initializing db
name=DTN (not shared), dir=/var/dtn/db
[1453350139.525746 /dtn/storage info] datastore /var/dtn/db
was cleanly shut down
[1453350139.525794 /dtnd notice] tidy option removing
directory '/bin/rm -rf /var/dtn/bundles'
[1453350139.716033 /dtn/storage/globals info] initializing
global table
[1453350140.127834 /S10 notice] Lines marked "S10" are log
lines added to make parsing
[1453350140.127880 /S10 notice] the logs easier after
extended tests. These were added for
[1453350140.127896 /S10 notice] the summer 2010 N4C summer
trial and are mainly new lines
[1453350140.127910 /S10 notice] that allow tracing bundles
across multiple hops.
[1453350140.127923 /S10 notice] To extract these do the
following...
[1453350140.127936 /S10 notice]   grep "S10 info" <dtnd-log-
files> | grep -v grep | awk '{print $4,$5}' | sort -n
[1453350140.128063 /S10 info]
rfc3339,time_t,usec,node,event,size,src,id,dest,expiry,flags
,peer,ob-src,ob-id,otime.ousec,conttype,duration,cmt
[1453350140.128087 /S10 info] 2016-01-21
04:22:20,1453350140.127948,dtn://putri.dtn,STARTING,0,,,,,,
,,0.000000,,0,
```

```

[1453350140.129144 /dtn/bundle/daemon notice] loading
bundles from data store
[1453350140.129261 /dtn/bundle/daemon info]
REGISTRATION_ADDED 0 dtn://putri.dtn
[1453350140.129355 /dtn/bundle/daemon info]
REGISTRATION_ADDED 2 dtn://putri.dtn/ping
[1453350140.129489 /dtn/bundle/daemon/loop warning] event
LINK_CREATED was in queue for 3.662274 seconds
[1453350140.138458 /dtn/bundle/daemon info] LINK_CREATED
link1 [192.168.10.2 dtn:none ONDEMAND tcp state=AVAILABLE]
[1453350140.138602 /dtn/bundle/daemon/loop warning] event
LINK_CREATED was in queue for 3.671255 seconds
[1453350140.149532 /dtn/bundle/daemon info] LINK_CREATED
link2 [192.168.20.12 dtn:none ONDEMAND tcp state=AVAILABLE]
[1453350140.149641 /dtn/bundle/daemon/loop warning] event
ROUTE_ADD was in queue for 3.682235 seconds
[1453350140.149665 /dtn/bundle/daemon info] ROUTE_ADD
dtn://server.dtn/* -> link1 (FORWARD)
[1453350140.149757 /dtn/bundle/daemon/loop warning] event
ROUTE_ADD was in queue for 3.682312 seconds
[1453350140.149776 /dtn/bundle/daemon info] ROUTE_ADD
dtn://klien.dtn/* -> link2 (FORWARD)
[1453350140.149819 /dtn/bundle/daemon info] LINK_AVAILABLE
link1 [192.168.10.2 dtn:none ONDEMAND tcp state=AVAILABLE]
[1453350140.149888 /dtn/bundle/daemon info] LINK_AVAILABLE
link2 [192.168.20.12 dtn:none ONDEMAND tcp state=AVAILABLE]
[1453350140.629122 /dtnd info] starting console on
127.0.0.1:5050
[1453350142.526049 /dtn/tclcmd info] can't load
oasys_tclreadline: can't find package oasys_tclreadline
[1453350142.526143 /dtn/tclcmd info] fall back to simple
command loop
[1453350182.707398 /dtn/link/link-0 info] new link link-0
[192.168.10.2:38297 dtn:none OPPORTUNISTIC tcp
state=UNAVAILABLE]
[1453350182.707545 /dtn/contact/link-0 info] new contact
contact 192.168.10.2:38297 (started 1453350182.707543)
[1453350182.748280 /dtn/bundle/daemon info] LINK_CREATED
link-0 [192.168.10.2:38297 dtn://fitri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE]
[1453350182.748579 /dtn/bundle/daemon info] CONTACT_UP link-
0 [192.168.10.2:38297 dtn://fitri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE] (contact 0x7f0438001400)
[1453350182.748625 /S10 info] 2016-01-21
04:23:02,1453350182.748610,dtn://putri.dtn,CONTUP,0,,,,,192
.168.10.2:38297,,,0.000000,OPPORTUNISTIC,0,
[1453350545.792346 /dtn/bundle/protocol info]
handle_bundle_received: begin
[1453350545.792421 /S10 info] 2016-01-21
04:29:05,1453350545.792403,dtn://putri.dtn,RX,734322688,dtn:
//fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIRIMDAT
A/734KOMA3MB.avi,506665308.0,dtn://klien.dtn/dtnctp/recv?file
=/734KOMA3MB.avi,3600,,192.168.10.2:38297,,,0.000000,,0,link

```

```

[1453350545.792451 /dtm/bundle/daemon info] BUNDLE_RECEIVED
bundle id 0
[dtm://fitri.dtm/dtncp/send?source=/home/fitri/Documents/KIR
IMDATA/734KOMA3MB.avi ->
dtm://klien.dtm/dtncp/recv?file=/734KOMA3MB.avi 734322688
byte payload] prevhop dtm://fitri.dtm (734322847 bytes
rcvd)
[1453350545.792487 /dtm/bundle/daemon info]
handle_bundle_received: validating bundle: calling
BlockProcessors?
[1453350545.792503 /dtm/bundle/protocol info]
BundleProtocol::validate begin
[1453350545.792520 /dtm/bundle/protocol info] validating
each individual blocks
[1453350545.792541 /dtm/bundle/protocol info]
BundleProtocol::validate end
[1453350545.792565 /dtm/bundle/expiration info] adding
bundle to pending list
[1453350545.892974 /dtm/bundle/protocol info]
BundleDaemon::handle_bundle_received: end
[1453350545.893190 /dtm/contact/link2 info] new contact
contact 192.168.20.12 (started 1453350545.893188)
[1453350545.899147 /dtm/bundle/daemon info] CONTACT_UP link2
[192.168.20.12 dtm://klien.dtm ONDEMAND tcp state=OPENING]
(contact 0x7f04400066c0)
[1453350545.899217 /S10 info] 2016-01-21
04:29:05,1453350545.899203,dtm://putri.dtm,CONTUP,0,,,,,192
.168.20.12,,,0.000000,ONDEMAND,0,
[1453350545.899390 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks begin
[1453350545.899469 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks looking for processors
[1453350545.899491 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks age block outbound is
DISABLED
[1453350545.899509 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks age block inbound support is
ENABLED
[1453350545.899526 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks age block zero creation is
ENABLED
[1453350545.899554 /dtm/bundle/extblock/aeb info] prepare()
begin
[1453350545.899572 /dtm/bundle/extblock/aeb info]
age_outbound_enabled != true, no AEB, returning BP_FAIL
[1453350545.899589 /dtm/bundle/extblock/aeb info] prepare()
end
[1453350545.899629 /dtm/bundle/protocol info]
BundleProtocol::prepare_blocks end
[1453350545.899647 /dtm/bundle/protocol info]
BundleProtocol::generate_blocks: calling generate() on each
block processor

```

```

[1453350545.899666 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x0
type 0x0) data_offset 0 data_length 0 contents length 0
[1453350545.899688 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x0
type 0x0) data_offset 0 data_length 0 contents length 0
[1453350545.899712 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x1
type 0x1) data_offset 7 data_length 734322688 contents
length 7
[1453350545.899733 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: generated block (owner 0x1
type 0x1) data_offset 7 data_length 734322688 contents
length 7
[1453350545.899754 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: calling generate() on
PRIMARY block
[1453350545.899790 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: calling finalize() on all
blocks
[1453350545.899813 /dtn/bundle/protocol info]
BundleProtocol::generate_blocks: end
[1453352376.884471 /dtn/bundle/daemon info]
BUNDLE_TRANSMITTED id:0 (734322847 bytes_sent/734322847
reliable) -> link2 (192.168.20.12)
[1453352376.884535 /S10 info] 2016-01-21
04:59:36,1453352376.884517,dtn://putri.dtn,TX,734322688,dtn:
//fitri.dtn/dtnctp/send?source=/home/fitri/Documents/KIRIMDAT
A/734KOMA3MB.avi,506665308.0,dtn://klien.dtn/dtnctp/recv?file
=/734KOMA3MB.avi,3600,,192.168.20.12,,,0.000000,,0,
[1453352454.672464 /dtn/cl/tcp/conn/0x7f04400067c0 info]
remote connection unexpectedly closed
[1453352454.672733 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPEN -> CLOSED] (connection
broken) for link link2 [192.168.20.12 dtn://klien.dtn
ONDEMAND tcp state=OPEN]
[1453352454.672814 /dtn/cl/tcp info] close_contact contact
192.168.20.12 (started 1453350545.893188)
[1453352454.673091 /dtn/bundle/daemon info] LINK UNAVAILABLE
link2 [192.168.20.12 dtn://klien.dtn ONDEMAND tcp
state=UNAVAILABLE]
[1453352454.673219 /dtn/bundle/daemon info] CONTACT_DOWN
link2 [192.168.20.12 dtn://klien.dtn ONDEMAND tcp
state=UNAVAILABLE] (connection broken) (contact
0x7f04400066c0)
[1453352454.673262 /S10 info] 2016-01-21
05:00:54,1453352454.673248,dtn://putri.dtn,CONTDOWN,0,,,,,1
92.168.20.12,,,1453350545.893188,ONDEMAND,1909,
[1453352459.675735 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [UNAVAILABLE -> OPEN] (re-
establishing connection) for link link2 [192.168.20.12
dtn://klien.dtn ONDEMAND tcp state=UNAVAILABLE]

```

```

[1453352459.675838 /dtn/contact/link2 info] new contact
contact 192.168.20.12 (started 1453352459.675835)
[1453352459.679096 /dtn/cl/tcp/conn/0x7f04400066c0 info]
remote socket closed connection -- returned POLLHUP
[1453352459.679236 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPENING -> CLOSED] (connection
broken) for link link2 [192.168.20.12 dtn://klien.dtn
ONDEMAND tcp state=OPENING]
[1453352459.679290 /dtn/cl/tcp info] close_contact contact
192.168.20.12 (started 1453352459.675835)
[1453352459.679457 /dtn/bundle/daemon info] LINK UNAVAILABLE
link2 [192.168.20.12 dtn://klien.dtn ONDEMAND tcp
state=UNAVAILABLE]
[1453352469.682181 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [UNAVAILABLE -> OPEN] (re-
establishing connection) for link link2 [192.168.20.12
dtn://klien.dtn ONDEMAND tcp state=UNAVAILABLE]
[1453352469.682291 /dtn/contact/link2 info] new contact
contact 192.168.20.12 (started 1453352469.682288)
[1453352469.685304 /dtn/cl/tcp/conn/0x7f04400066c0 info]
remote socket closed connection -- returned POLLHUP
[1453352469.685364 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPENING -> CLOSED] (connection
broken) for link link2 [192.168.20.12 dtn://klien.dtn
ONDEMAND tcp state=OPENING]
[1453352469.685386 /dtn/cl/tcp info] close_contact contact
192.168.20.12 (started 1453352469.682288)
[1453352469.685482 /dtn/bundle/daemon info] LINK UNAVAILABLE
link2 [192.168.20.12 dtn://klien.dtn ONDEMAND tcp
state=UNAVAILABLE]
[1453352489.688278 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [UNAVAILABLE -> OPEN] (re-
establishing connection) for link link2 [192.168.20.12
dtn://klien.dtn ONDEMAND tcp state=UNAVAILABLE]
[1453352489.688378 /dtn/contact/link2 info] new contact
contact 192.168.20.12 (started 1453352489.688376)
[1453352489.691414 /dtn/cl/tcp/conn/0x7f04400066c0 info]
remote socket closed connection -- returned POLLHUP
[1453352489.691652 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPENING -> CLOSED] (connection
broken) for link link2 [192.168.20.12 dtn://klien.dtn
ONDEMAND tcp state=OPENING]
[1453352489.691738 /dtn/cl/tcp info] close_contact contact
192.168.20.12 (started 1453352489.688376)
[1453352489.691901 /dtn/bundle/daemon info] LINK UNAVAILABLE
link2 [192.168.20.12 dtn://klien.dtn ONDEMAND tcp
state=UNAVAILABLE]
[1453352529.692561 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [UNAVAILABLE -> OPEN] (re-
establishing connection) for link link2 [192.168.20.12
dtn://klien.dtn ONDEMAND tcp state=UNAVAILABLE]
[1453352529.692672 /dtn/contact/link2 info] new contact
contact 192.168.20.12 (started 1453352529.692670)

```

```

[1453352529.695564 /dtn/cl/tcp/conn/0x7f04400066c0 info]
remote socket closed connection -- returned POLLHUP
[1453352529.695771 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPENING -> CLOSED] (connection
broken) for link link2 [192.168.20.12 dtn://klien.dtn
ONDEMAND tcp state=OPENING]
[1453352529.695851 /dtn/cl/tcp info] close_contact contact
192.168.20.12 (started 1453352529.692670)
[1453352529.696042 /dtn/bundle/daemon info] LINK UNAVAILABLE
link2 [192.168.20.12 dtn://klien.dtn ONDEMAND tcp
state=UNAVAILABLE]
[1453352546.960544 /dtn/cl/tcp/conn/0x7f0448000940 info]
remote connection unexpectedly closed
[1453352546.960771 /dtn/bundle/daemon info]
LINK_STATE_CHANGE_REQUEST [OPEN -> CLOSED] (connection
broken) for link link-0 [192.168.10.2:38297 dtn://fitri.dtn
OPPORTUNISTIC tcp state=OPEN]
[1453352546.960851 /dtn/cl/tcp info] close_contact contact
192.168.10.2:38297 (started 1453350182.707543)
[1453352546.961064 /dtn/bundle/daemon info] LINK UNAVAILABLE
link-0 [192.168.10.2:38297 dtn://fitri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE]
[1453352546.961126 /dtn/bundle/daemon info] CONTACT_DOWN
link-0 [192.168.10.2:38297 dtn://fitri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE] (connection broken) (contact
0x7f0438001400)
[1453352546.961176 /S10 info] 2016-01-21
05:02:26,1453352546.961147,dtn://putri.dtn,CONTDOWN,0,,,,,1
92.168.10.2:38297,,,1453350182.707543,OPPORTUNISTIC,2364,

```

3. Log Bundle Client

```
[1467245861.606150 /dtnd notice] random seed is 606128
[1467245861.606481 /dtnd notice] DTN daemon starting up...
(pid 2541)
[1467245861.617120 /dtnd info] parsing configuration file
/etc/dtn.conf...
[1467245861.617709 /dtnd info] dtnd parsing configuration...
[1467245861.636617 /dtn/interface/table info] adding
interface tcp0 (tcp)
[1467245861.637581 /dtn/link/link1 info] new link link1
[192.168.10.1 dtn:none ONDEMAND tcp state=AVAILABLE]
[1467245861.637981 /dtnd info] dtnd configuration parsing
complete
[1467245861.638094 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 3 SECONDS
[1467245862.638503 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 2 SECONDS
[1467245863.639031 /dtn/storage warning] PRUNING CONTENTS OF
/var/dtn/db IN 1 SECONDS
[1467245864.639376 /dtn/storage notice] tidy option removing
directory '/bin/rm -rf /var/dtn/db'
[1467245864.678814 /dtn/storage notice] creating new
database directory /var/dtn/db
[1467245864.679356 /dtn/storage info] initializing db
name=DTN (not shared), dir=/var/dtn/db
[1467245864.680506 /dtn/storage info] datastore /var/dtn/db
was cleanly shut down
[1467245864.680600 /dtnd notice] tidy option removing
directory '/bin/rm -rf /var/dtn/bundles'
[1467245864.885559 /dtn/storage/globals info] initializing
global table
[1467245865.475527 /S10 notice] Lines marked "S10" are log
lines added to make parsing
[1467245865.475786 /S10 notice] the logs easier after
extended tests. These were added for
[1467245865.475878 /S10 notice] the summer 2010 N4C summer
trial and are mainly new lines
[1467245865.475945 /S10 notice] that allow tracing bundles
across multiple hops.
[1467245865.476041 /S10 notice] To extract these do the
following...
[1467245865.476138 /S10 notice]   grep "S10 info" <dtnd-log-
files> | grep -v grep | awk '{print $4,$5}' | sort -n
[1467245865.476572 /S10 info]
rfc3339,time_t,usec,node,event,size,src,id,dest,expiry,flags
,peer,ob-src,ob-id,otime,ousec,conttype,duration,cmt
[1467245865.476695 /S10 info] 2016-06-30
00:17:45,1467245865.476226,dtn://klien.dtn,STARTING,0,,,,,,
,,0.000000,,0,
[1467245865.478557 /dtn/bundle/daemon notice] loading
bundles from data store
```



```

[1467245865.478738 /dtn/bundle/daemon info]
REGISTRATION_ADDED 0 dtn://klien.dtn
[1467245865.478878 /dtn/bundle/daemon info]
REGISTRATION_ADDED 2 dtn://klien.dtn/ping
[1467245865.479087 /dtn/bundle/daemon/loop warning] event
LINK_CREATED was in queue for 3.841346 seconds
[1467245865.485497 /dtn/bundle/daemon info] LINK_CREATED
link1 [192.168.10.1 dtn:none ONDEMAND tcp state=AVAILABLE]
[1467245865.485798 /dtn/bundle/daemon/loop warning] event
ROUTE_ADD was in queue for 3.847913 seconds
[1467245865.485911 /dtn/bundle/daemon info] ROUTE_ADD
dtn://router.dtn/* -> link1 (FORWARD)
[1467245865.486471 /dtn/bundle/daemon/loop warning] event
ROUTE_ADD was in queue for 3.848503 seconds
[1467245865.486743 /dtn/bundle/daemon info] ROUTE_ADD
dtn://server.dtn/* -> link1 (FORWARD)
[1467245865.487084 /dtn/bundle/daemon info] LINK_AVAILABLE
link1 [192.168.10.1 dtn:none ONDEMAND tcp state=AVAILABLE]
[1467245865.978101 /dtnd info] starting console on
127.0.0.1:5050
[1467245867.017506 /dtn/tclcmd info] can't load
oasys_tclreadline: can't find package oasys_tclreadline
[1467245867.017686 /dtn/tclcmd info] fall back to simple
command loop
[1467245872.819877 /dtn/apiclient/25 info] new session
127.0.0.1:5010 -> 127.0.0.1:54834
[1467245872.841628 /dtn/bundle/daemon info]
REGISTRATION_ADDED 10 dtn://klien.dtn/dtncp/recv?file=*
[1467245872.841781 /dtn/registration/table info] adding
registration 10/dtn://klien.dtn/dtncp/recv?file=*
[1467246337.116464 /dtn/link/link-0 info] new link link-0
[192.168.20.1:58424 dtn:none OPPORTUNISTIC tcp
state=UNAVAILABLE]
[1467246337.118121 /dtn/contact/link-0 info] new contact
contact 192.168.20.1:58424 (started 1467246337.118109)
[1467246339.545419 /dtn/bundle/daemon info] LINK_CREATED
link-0 [192.168.20.1:58424 dtn://putri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE]
[1467246339.546040 /dtn/bundle/daemon/loop warning] event
LINK_CREATED took 2427 ms to process
[1467246339.546234 /dtn/bundle/daemon info] CONTACT_UP link-
0 [192.168.20.1:58424 dtn://putri.dtn OPPORTUNISTIC tcp
state=UNAVAILABLE] (contact 0xb1a00b18)
[1467246339.546425 /S10 info] 2016-06-30
00:25:39,1467246339.546379,dtn://klien.dtn,CONTUP,0,,,,,192
.168.20.1:58424,,,0.000000,OPPORTUNISTIC,0,
[1467247266.898877 /dtn/bundle/protocol info]
handle_bundle_received: begin
[1467247266.899042 /S10 info] 2016-06-30
00:41:06,1467247266.899008,dtn://klien.dtn,RX,734322688,dtn:
//fitri.dtn/dtncp/send?source=/home/fitri/Documents/KIRIMDAT
A/734KOMA3MB.avi,520561203.0,dtn://klien.dtn/dtncp/recv?file
=/734KOMA3MB.avi,3600,,192.168.20.1:58424,,,0.000000,,0,link

```

```

[1467247266.899117 /dtn/bundle/daemon info] BUNDLE_RECEIVED
bundle id 0
[dt://fitri.dtn/dtncp/send?source=/home/fitri/Documents/KIR
IMDATA/734KOMA3MB.avi ->
dt://klien.dtn/dtncp/recv?file=/734KOMA3MB.avi 734322688
byte payload] prevhop dt://putri.dtn (734322847 bytes
recvd)
[1467247266.899179 /dtn/bundle/daemon info]
handle_bundle_received: validating bundle: calling
BlockProcessors?
[1467247266.899219 /dtn/bundle/protocol info]
BundleProtocol::validate begin
[1467247266.899257 /dtn/bundle/protocol info] validating
each individual blocks
[1467247266.899300 /dtn/bundle/protocol info]
BundleProtocol::validate end
[1467247266.899349 /dtn/bundle/expiration info] adding
bundle to pending list
[1467247267.024143 /dtn/bundle/protocol info]
BundleDaemon::handle_bundle_received: end
[1467247267.024652 /dtn/registration/10 info]
deliver_bundle: queuing bundle id 0 for active delivery to
dt://klien.dtn/dtncp/recv?file=*
[1467247283.374750 /dtn/apiclient/25 info] DTN_RECV:
successfully delivered bundle 0 to registration 10
[1467247283.375253 /dtn/bundle/daemon info] BUNDLE_DELIVERED
id:0 (734322688 bytes) -> regid 10
(dt://klien.dtn/dtncp/recv?file=*)
[1467247283.375479 /S10 info] 2016-06-30
00:41:23,1467247283.375423,dt://klien.dtn,DELIVERED,7343226
88,dt://fitri.dtn/dtncp/send?source=/home/fitri/Documents/K
IRIMDATA/734KOMA3MB.avi,520561203.0,dt://klien.dtn/dtncp/re
cv?file=/734KOMA3MB.avi,3600,,dt://klien.dtn/dtncp/recv?fil
e=*,,,0.000000,,0,
[1467247341.007238 /dtn/apiclient/25 info] IPC socket closed
while blocked in read... application must have exited

```