

ANALISIS KINERJA DAN PERFORMANSI PENGIRIMAN FILE HASIL TANGKAPAN KAMERA DENGAN IBR-DTN

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Abid Hidayat
NIM: 125150202111009



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017

PENGESAHAN

ANALISIS KINERJA DAN PERFORMANSI PENGIRIMAN FILE HASIL TANGKAPAN
KAMERA DENGAN IBR-DTN
SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Abid Hidayat
NIM: 125150202111009

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Februari 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Eko Sakti P., S.Kom, M.Kom
NIK: 201102 860805 1 001

Ari Kusyanti, S.T, M.Sc
NIK: 201102 831228 2 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

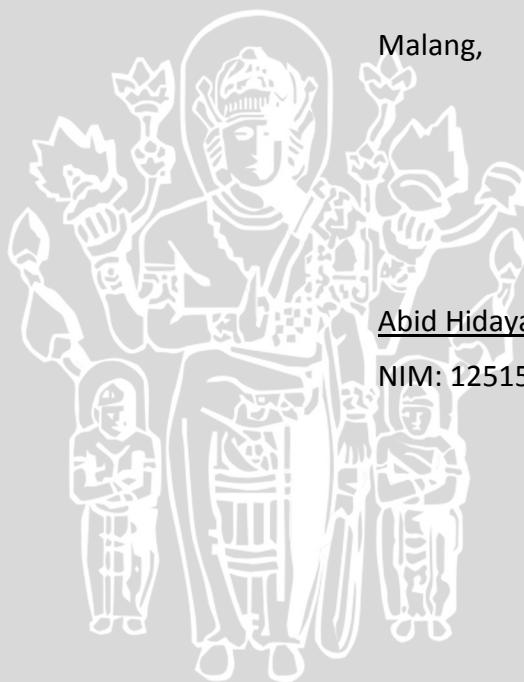
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang,

Abid Hidayat

NIM: 125150202111009



KATA PENGANTAR

Syukur Alhamdulillah penulis ucapkan atas kehadiran Allah Subhanahu wa Ta'ala, karena berkat rahmat dan anugerah-Nya penulis dapat menyelesaikan skripsi yang berjudul "Analisis Kinerja dan Performa Pengiriman File Pada IBR-DTN" shalawat serta salam semoga selalu tercurah pada baginda Rasulullah Muhammad SAW. Penulisan skripsi ini akan sulit terwujud apabila tidak ada rahmat dari sang pemilik langit dan bumi yaitu Allah Subhanahu wa Ta'ala serta bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan rasa terimakasih kepada:

1. Ayah H. Suparman, Ibu Hj. Suyatni yang telah mendukung dan memotivasi selama pengerjaan skripsi.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph,D selaku ketua jurusan teknik informatika yang telah memeberikan kesempatan untuk mengikuti ujian skripsi
3. Bapak Eko Sakti P., S.Kom, M.Kom dan Ibu Ari Kusyanti, S.T, M.Sc selaku dosen pembimbing yang telah banyak memberikan bimbingan, saran dan motivasi kepada penulis.
4. Seluruh Dosen Program Studi Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
5. Seluruh Civitas Akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
6. Bapak M. Misbachul Munir selaku Kadiv. UPLA, Bapak Gatot Selaku Manager Umum dan rekan-rekan kerja Wonokoyo Breeding Departemen Umum.
7. Dita Pramawati yang telah memberikan semangat, memberikan nasihat, memberikan canda dan tawanya dalam suka duka menjalani kuliah dan mengerjakan skripsi.
8. Teman-teman yang membantu penulis dalam pengerjaan Aji Fathul Huda, Moh, Zahrul Muttaqin, Yazid, Kukuh Bhaskara, Nining Nahdiah Satriani, Ellsa Nur Amilushofia, Fadli Hakim, Giassurahman, Hadi Abdurrahman, Dody Kurniawan, Sirojul Hadi, Ellsa Yuniar Rahmawati, Lestari Sintika Syarah, UKM Merpati Putih, Grup Intelijen Tusbolser, Grup Bajiroad dan Grup Skripsi Bahagia

Penulis menyadari bahwa skripsi ini tidaklah sempurna, kritik dan saran yang membangun sangat diharapkan dari para pembaca sehingga penulis dapat melakukan perbaikan yang diperlukan. Akhir kata, penulis berharap skripsi ini dapat bermanfaat bagi siapapun yang membacanya.

Malang,

Penulis

Abid Hidayat



ABSTRAK

Informasi adalah data yang telah diolah menjadi suatu bentuk lain yang lebih berguna seperti teks, gambar, suara dan video. Seiring perkembangan teknologi, pertukaran informasi dilakukan dengan berbagai macam cara. Salah satunya adalah dengan media internet. Internet adalah jaringan komputer yang saling terhubung menggunakan standar sistem global *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP). Pada pengiriman paket dalam jaringan TCP/IP waktu yang dibutuhkan tidak terlalu lama (*short round trips*). Apabila waktu pengiriman data terlalu lama, maka paket data akan di *drop*. Untuk memperbaiki jaringan TCP/IP, maka diperlukan jaringan yang tidak memperlambat *delay*. *Delay Tolerant Network* (DTN) adalah arsitektur jaringan untuk menyediakan solusi bagi jaringan yang memiliki konektivitas yang terputus-putus, *long delay*, kecepatan data yang berbeda dan tingkat kesalahan yang tinggi. Metode komunikasi pada DTN adalah *store and forward*, yaitu metode yang menggunakan lapisan baru untuk menyimpan dan meneruskan pesan yang disebut lapisan *bundle*. Salah satu pengembangan dari DTN adalah IBR-DTN. IBR-DTN adalah perangkat lunak yang efisien untuk *embedded system*. Pada penelitian ini dilakukan analisis terhadap kinerja dan performa pengiriman *file* pada IBR-DTN. Terdapat beberapa skenario pengujian yang telah dirancang sebelumnya antara lain adalah pengujian pengiriman *file* dengan konektivitas putus-sambung, pengujian dengan mengirimkan *file* dalam jumlah tertentu dan yang terakhir adalah pengiriman *file* selama 30 menit. Hasil dari pengujian yang didapatkan adalah simulasi yang dibangun dapat melakukan pengiriman *file* dengan keadaan konektivitas yang putus sambung. Pengiriman file membutuhkan waktu rata-rata selama 18.7 detik untuk tiap *file* yang dikirim.

Kata kunci: DTN, Bundle, Store and Forward, IBR-DTN.

ABSTRACT

Information is data that has been processed into another form that is more useful such as text, images, sound and video. Along with the development of technology, exchange of information is done in various ways. One of them is the internet media. The Internet is a network of computers connected together using the global system standard Transmission Control Protocol / Internet Protocol Suite (TCP / IP). On the delivery of packets in a TCP / IP time needed is not too long (short round trips). If the data transmission time is too long, then the data packet will be dropped. To fix the TCP / IP network, it is necessary that the network did not make delay. Delay Tolerant Network (DTN) is a network architecture to provide solutions for network connectivity is intermittent, long delay, different data speeds and high error rates. This method of communication at DTN is –store and forward, a method that uses a new layer to store and forward messages called bundle layer. One is the development of IBR-DTN DTN. IBR-DTN is efficient software for embedded systems. In this research, analysis of the performance and the performance file delivery on the IBR-DTN. There are several scenarios testing that has been previously designed, among others, is testing the transmission of files with connectivity off and on, the test by sending a certain number of files and the last one is sending the file for 30 minutes. The results of the simulation tests obtained are built to perform transmission of files with broken connection connectivity state. Sending files takes on average for 18.7 seconds for every file sent.

Keywords: DTN, Bundle, Store and Forward, IBR-DTN.



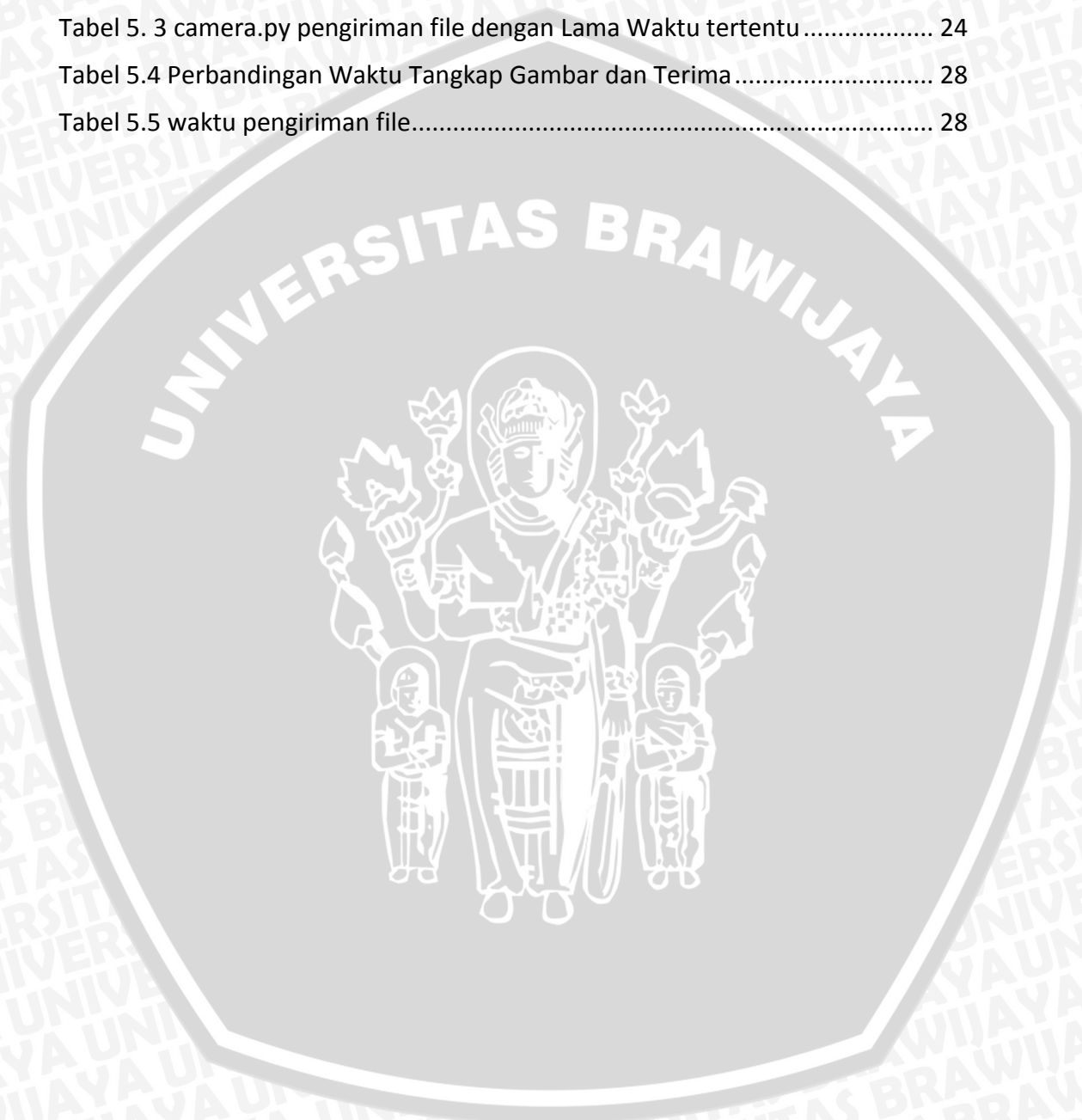
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Landasan Teori.....	5
2.2.1 Delay Tolerant Network (DTN).....	5
2.2.2 IBR-DTN	8
2.2.3 Raspberry Pi	8
BAB 3 METODOLOGI	10
3.1 Studi Literatur	11
3.2 Analisis Kebutuhan Simulasi	11
3.2.1 Perangkat Keras	11
3.2.2 Perangkat Lunak.....	12
3.3 Perancangan Sistem.....	12
3.3.1 Instalasi dan Konfigurasi IBR-DTN	12
3.3.2 Konfigurasi Kamera Modul.....	13

3.3.3 Topologi.....	13
3.4 Implementasi Simulasi	14
3.5 Hasil dan Pembahasan	14
3.6 Kesimpulan.....	14
BAB 4 Implementasi Simulasi.....	15
4.1 Implementasi	15
4.1.1 Access Point	15
4.1.2 IBR-DTN	17
4.2 Pengujian Test Pengiriman gambar	19
BAB 5 HASIL DAN Pembahasan	21
5.1 Skenario Pengujian	21
5.1.1 Pengujian Pengiriman File dengan Konektivitas Putus-Sambung	21
5.1.2 Pengiriman File dengan Jumlah Tertentu	22
5.1.3 Pengiriman File Selama 30 Menit	24
5.2 Pembahasan.....	25
5.2.1 Hasil Pengujian Pengiriman File dengan Konektivitas Putus-Sambung.....	25
5.2.2 Hasil Pengujian Pengiriman File Dengan Jumlah Tertentu	26
5.2.3 Hasil Pengujian Pengiriman File selama 30 menit	28
BAB 6 Penutup	30
6.1 Kesimpulan.....	30
6.2 Saran	30
DAFTAR PUSTAKA.....	31
LAMPIRAN	32

DAFTAR TABEL

Tabel 5.1 camera.py pengujian konektivitas putus-sambung	21
Tabel 5.2 camera.py pengiriman file dengan Jumlah tertentu.....	23
Tabel 5. 3 camera.py pengiriman file dengan Lama Waktu tertentu	24
Tabel 5.4 Perbandingan Waktu Tangkap Gambar dan Terima	28
Tabel 5.5 waktu pengiriman file.....	28



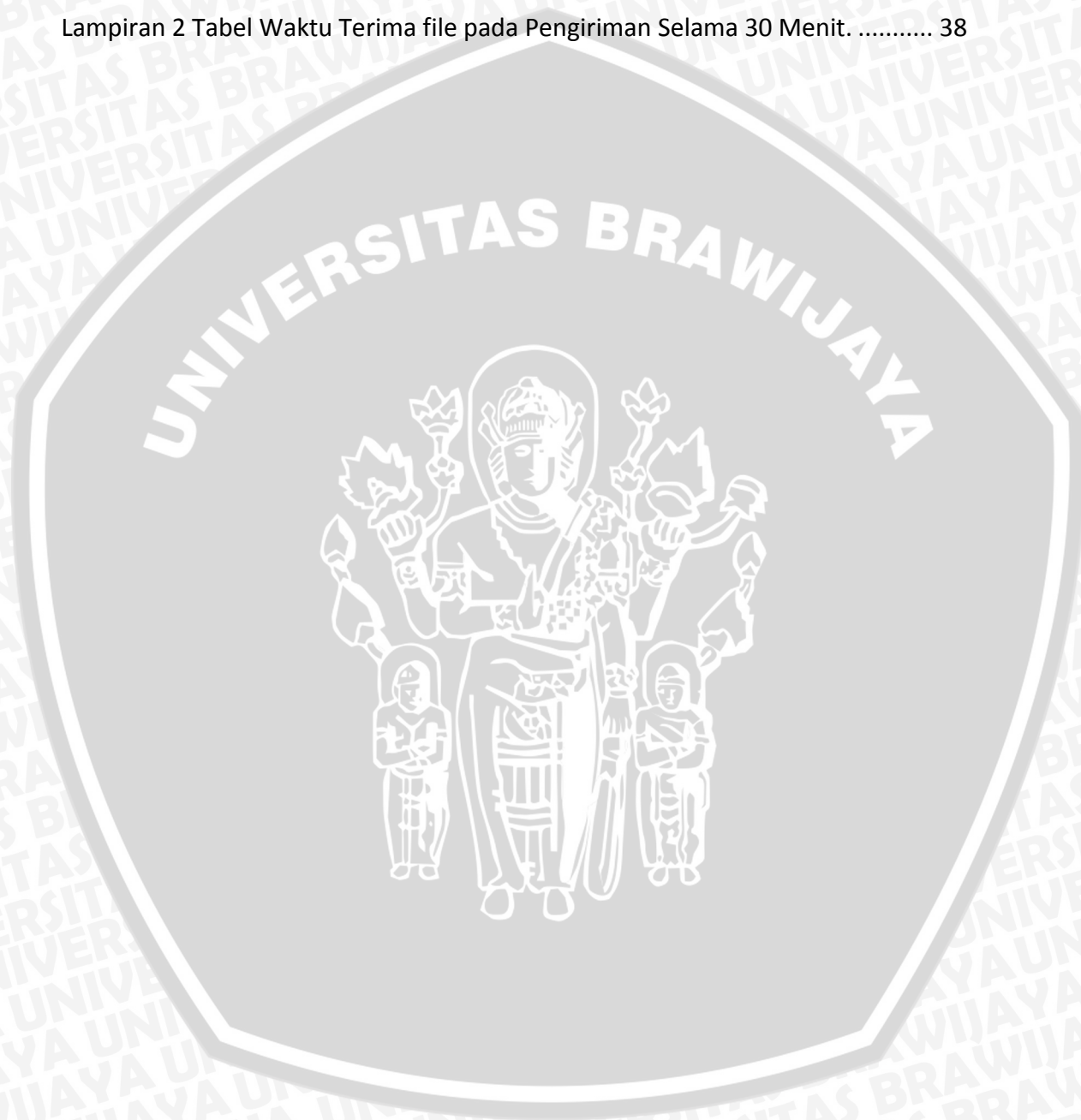
DAFTAR GAMBAR

Gambar 2.1 Metode Store and Forward.....	6
Gambar 2.2 Letak Bundle Layer	7
Gambar 2.3 ACK pada Bundle Layer	7
Gambar 2.4 Arsitektur IBR-DTN	8
Gambar 2.5 Komponen Raspberry Pi Model B+	9
 Gambar 3.1 Diagram Alir Proses Penelitian.....	10
Gambar 3.2 Langkah Instalasi dan Konfigurasi IBR-DTN.....	13
Gambar 3.3 Topologi Simulasi	14
 Gambar 4.1 Access Point yang telah tersambung	17
Gambar 4.2 konfigurasi IBR-DTN	18
Gambar 4.3 File gambar ditemukan oleh dtnoutbox	19
Gambar 4.4 File gambar diterima oleh dtninbox.....	20
 Gambar 5.1 pengujian konektivitas putus-sambung	22
Gambar 5.2 Node Penerima tidak Ditemukan.....	25
Gambar 5.3 Node Pengirim Terhubung Kembali	26
Gambar 5.4 File Masuk ke Node Penerima.....	26
Gambar 5.5 Pengiriman file sebanyak 100 kali.....	27
Gambar 5.6 Pengiriman file sebanyak 200 kali.....	27
Gambar 5.7 Hasil Pengiriman File dengan Lama Waktu Tertentu.....	28
Gambar 5.8 Pengiriman File selama 30 Menit.....	29

DAFTAR LAMPIRAN

Lampiran 1 IBRDTN.CONF. 32

Lampiran 2 Tabel Waktu Terima file pada Pengiriman Selama 30 Menit. 38



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada era ini, kebutuhan akan informasi semakin meningkat. Informasi merupakan aset yang paling bernilai dan dibutuhkan oleh semua kalangan. Informasi adalah data yang telah diolah menjadi suatu bentuk lain yang lebih berguna seperti teks, gambar, suara dan video (Siswanti, 2013). Seiring perkembangan teknologi, pertukaran informasi dilakukan dengan berbagai macam cara. Salah satunya adalah dengan media internet. Internet adalah jaringan komputer yang saling terhubung menggunakan standar sistem global *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP) sebagai protokol pertukaran paket (Wikipedia, 2016). Sebagai protokol yang paling banyak digunakan dalam jaringan komputer, TCP/IP tidak dapat bekerja apabila terjadi delay yang cukup lama. Agar pengiriman paket dengan TCP/IP dapat bekerja, terdapat syarat-syarat yang harus dipenuhi oleh jaringan. Syarat-syarat tersebut adalah adanya koneksi end-to-end, *short round trips* dan *low error rates* (Suharsono, 2012). Koneksi end-to-end adalah koneksi antara pengirim dan tujuan yang kontinyu dan bidirectional sehingga koneksi jaringan tidak boleh terputus. *Short round trips* adalah waktu pengiriman paket dari pengirim ke tujuan tidak terlalu lama. *Low error rates* adalah tingkat kesalahan pengiriman kecil.

Pada pertukaran informasi yang dilakukan di internet dengan protokol pengiriman paket adalah TCP/IP, maka harus terdapat koneksi end-to-end yang selalu tersedia. Koneksi end-to-end melewati beberapa router dari pengirim ke tujuan. Apabila koneksi terputus ditengah-tengah pengiriman, maka koneksi internet akan terputus. Biasanya, sistem akan mencari rute lain untuk sampai pada tujuan. Namun, apabila semua rute gagal dalam melakukan pengiriman, maka koneksi internet akan putus. Semua paket yang sudah dikirim, tetapi belum sampai pada tujuan akan dibuang atau di "drop". Selanjutnya, apabila koneksi sudah kembali normal, maka proses pengiriman paket akan dimulai dari awal (Suharsono, 2012). Selain itu, pada pengiriman paket dalam jaringan TCP/IP waktu yang dibutuhkan tidak terlalu lama (*short round trips*). Apabila waktu pengiriman data terlalu lama, maka paket data akan di drop. Untuk memperbaiki jaringan TCP/IP, maka diperlukan jaringan yang tidak memperlama *delay*.

Delay Tolerant Network (DTN) adalah arsitektur jaringan untuk menyediakan solusi bagi jaringan yang memiliki konketivitas yang terputus-putus, *long delay*, kecepatan data yang berbeda dan tingkat kesalahan yang tinggi (Siswanti, 2013). Metode komunikasi pada DTN adalah *store and forward*, yaitu metode yang menggunakan lapisan baru untuk menyimpan dan meneruskan pesan yang disebut lapisan *bundle*. Jika koneksi terputus, DTN akan menyimpan sebagian data yang telah dikirimkan. Ketika koneksi telah kembali normal, maka proses pengiriman paket akan dilanjutkan tanpa mengulang proses dari awal. DTN merupakan sistem jaringan yang dapat digunakan sebagai alternatif untuk komunikasi jaringan pada daerah terpencil yang memiliki permasalahan koneksi

selama proses pengiriman data berlangsung. Dalam beberapa tahun terakhir DTN menjadi sangat populer untuk diterapkan dan banyak dikembangkan. Salah satu pengembangan dari DTN adalah IBR-DTN. IBR-DTN adalah perangkat lunak yang efisien untuk *embedded system* (Yuliandoko dkk, 2015).

Penelitian terkait pengiriman data dengan jaringan DTN adalah “Perancangan Sistem Pertukaran Informasi di Pedesaan Berbasis *Delay Tolerant Network* Menggunakan Raspberry Pi” (Prakista, 2013). Penelitian ini membahas performa dan kinerja IBR-DTN dalam sistem pertukaran informasi di pedesaan. Dalam penelitian ini dilakukan tiga buah macam pengujian untuk menganalisa performa dan kinerja dari IBR-DTN antara lain adalah pengujian konektifitas, pengujian besaran *file* yang dikirim dan pengujian pengiriman *file* dengan skema putus-sambung. Kesimpulan dari penelitian ini adalah IBR-DTN dapat diterapkan untuk melakukan pertukaran informasi pada konektifitas yang terbatas. Penelitian lain terkait kinerja dan performa infrastruktur IBR-DTN adalah “Analisis Penggunaan Protokol Routing Prophet Pada IBR-DTN untuk Sistem Berbagi Informasi Digital di Daerah Pedalaman” (Magdalena, 2014). Penelitian ini membahas performa pengiriman *file* dengan konektifitas terbatas pada IBR-DTN dengan menggunakan routing prophet.

Dari penelitian-penelitian yang telah dijelaskan, data yang dikirim berupa file statis yang dibuat atau telah disiapkan sebelumnya, belum dibahas tentang data yang dikirim berupa file dinamis contohnya file dari tangkapan kamera. Oleh karena itu dalam penelitian ini dilakukan "**Analisis Kinerja dan Performansi Pengiriman File Hasil Tangkapan Kamera dengan IBR-DTN**". Hasil dari analisis ini diharapkan dapat menjadi salah satu bahan informasi dalam mengoptimalkan performa untuk pertukaran informasi berbasis gambar dalam jaringan DTN.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dikhususkan pada:

1. Bagaimana Membangun simulasi pertukaran informasi berbasis gambar hasil tangkap kamera pada IBR-DTN?
2. Bagaimana analisis pengiriman data berbasis gambar hasil tangkap kamera pada IBR-DTN?

1.3 Tujuan

Tujuan dari penelitian ini adalah :

1. Mampu merancang simulasi pertukaran informasi berbasis gambar hasil tangkap kamera pada IBR-DTN.
2. Mengetahui hasil dari analisis pertukaran informasi berbasis gambar hasil tangkap kamera pada IBR-DTN.

1.4 Manfaat

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Sebagai salah satu bahan informasi yang dapat ditambahkan ke dalam kepastakaan dalam mengoptimalkan performa untuk pertukaran informasi berbasis gambar hasil tangkap kamera dalam jaringan DTN.
2. Dapat dijadikan sebagai referensi lebih lanjut untuk para peneliti dalam menganalisa jaringan DTN.

1.5 Batasan masalah

Batasan masalah dalam penelitian diberikan agar pembahasan tidak melebar dan lebih terperinci. Adapun penelitian ini dibatasi oleh hal-hal sebagai berikut:

1. Perancangan dan analisis menggunakan perangkat lunak IBR-DTN dan perangkat keras Raspberry Pi.
2. *File* yang dikirim berupa gambar.
3. Gambar ditangkap melalui kamera modul secara *realtime*.

1.6 Sistematika pembahasan

Sistematika penulisan dalam penelitian ini dibagi menjadi beberapa bab, yang terdiri dari:

BAB I Pendahuluan

Berisi tentang penjelasan latar belakang masalah terkait alasan analisa kinerja dan performa pengiriman *file* pada IBR-DTN . Terdapat juga rumusan masalah serta tujuan adanya rumusan masalah ini. Kemudian terdapat manfaat yang ditujukan untuk peneliti selanjutnya ataupun yang membaca. Serta adanya batasan masalah yang berguna untuk membatasi pembahasan yang terdapat pada penelitian ini.

BAB II Landasan Kepustakaan

Memuat kajian teori sebagai landasan penyusunan topik skripsi yang diawali dengan penelitian terkair yang merujuk pada penelitian yang diangkat. Kemudian terdapat kajain teori tentang *Delay Tolerant Network* (DTN), IBR-DTN, Bundle dan *Store and Forward*.

BAB III Metodologi

Metodologi terdiri dari sekumpulan metode yang digunakan untuk menyelesaikan masalah dalam penelitian. Diawali dengan studi pendahuluan yang terdiri dari studi literatur, kemudian melakukan analisa kebutuhan seperti perangkat keras dan lunak yang digunakan dalam penelitian, kemudian masuk keperancangan sistem, setelah perancangan sistem masuk ke implementasi, melakukan pengujian dan Analisa dan yang terakhir adalah pengambilan keputusan.

BAB IV Implementasi Simulasi

Bab implementasi dan pengujian lebih menjelaskan mengenai langkah langkah implementasi dari sistem tersendiri dan pengujian yang dilakukan terhadap sistem yang dibangun.

BAB V Hasil dan Pembahasan

Bab hasil dan pembahasan memuat analisa hasil pengujian analisis kinerja dan performa pengiriman *file* pada IBR-DTN. Dengan parameter uji yang sudah ditentukan di setiap skenario pengujian. Hasil pengujian ini kemudian dilakukan analisa sesuai dengan rumusan masalah. Terdapat pula tabel dan grafik yang dapat menunjang informasi dari hasil pengujian.

BAB VI Penutup

Pada bab ini memuat kesimpulan yang merangkum jawaban dari rumusan masalah berdasarkan hasil penelitian yang sudah dilakukan dan saran untuk pengembangan penelitian lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan mengenai kajian pustaka dan dasar-dasar teori yang menunjang penelitian ini. Pada kajian pustaka akan dijelaskan mengenai penelitian-penelitian yang terkait sebelumnya. Dasar teori menjelaskan teori teori yang berkaitan dengan penelitian ini.

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini membahas tentang penelitian sebelumnya terkait kinerja dan performa infrastruktur IBR-DTN adalah “Perancangan Sistem Pertukaran Informasi di Pedesaan Berbasis Delay Tolerant Network Menggunakan RaspberryPI” (Evan, 2013). Penelitian ini membahas performa dan kinerja IBR-DTN dalam sistem pertukaran informasi di pedesaan. Dalam penelitian ini dilakukan tiga buah macam pengujian untuk menganalisa performa dan kinerja dari IBR-DTN antara lain adalah pengujian konektifitas, pengujian besaran *file* yang dikirim dan pengujian pengiriman *file* dengan skema putus-sambung. Kesimpulan dari penelitin ini adalah IBR-DTN dapat diterapkan untuk melakukan pertukaran informasi pada konektifitas yang terbatas.

Penelitian lain terkait kinerja dan performa infrastruktur IBR-DTN adalah “Analisis Penggunaan Protokol Routing Prophet Pada IBR-DTN untuk Sistem Berabagi Informasi Digital di Daerah Pedalaman” (Magdalena, 2014). Penelitian ini membahas performa pengiriman *file* dengan konektifitas terbatas pada IBR-DTN dengan menggunakan routing prophet. Kesimpulan dari penelitian ini adalah routing prophet berguna untuk menghubungkan node-node bergerak dengan pengetahuannya terhadap jumlah pertemuan node sebelumnya, dan kinerja dari penggunaan prophet menunjukkan peningkatan *average delay* jika node berjumlah 2 node.

Dari penelitian-penelitian yang telah dijelaskan, berkaitan dengan penelitian penulis yaitu analisis kinerja dan performa pengiriman *file* pada IBR-DTN. Dimana *file* yang akan dikirimkan berupa gambar yang ditangkap melalui sebuah kamera secara realtime.

2.2 Landasan Teori

2.2.1 Delay Tolerant Network (DTN)

Delay Tolerant Network (DTN) adalah arsitektur jaringan untuk menyediakan solusi bagi jaringan yang memiliki konketivitas yang terputus-putus, *long delay*, kecepatan data yang berbeda dan tingkat kesalahan yang tinggi (Siswanti, 2013). Metode komunikasi pada DTN adalah *store and forward*, yaitu metode yang menggunakan lapisan baru untuk menyimpan dan meneruskan pesan yang disebut lapisan *bundle*. Jika koneksi terputus, DTN akan menyimpan sebagian data yang telah dikirimkan. Ketika koneksi telah kembali normal, maka proses pengiriman paket akan dilanjutkan tanpa mengulang proses dari awal. DTN

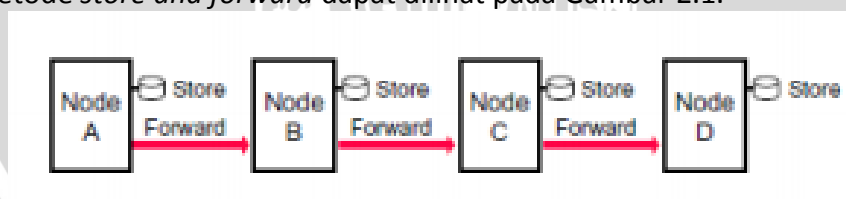
merupakan sistem jaringan yang dapat digunakan sebagai alternatif untuk komunikasi jaringan pada daerah terpencil yang memiliki permasalahan koneksi selama proses pengiriman data berlangsung.

Arsitektur DTN lebih menekankan pada rute (*routing*), penamaan (*naming*) dan kemampuan keamanan. Arsitektur DTN didasarkan pada sejumlah prinsip diantaranya adalah penggunaan *delay* yang panjang, penggunaan penyimpanan dalam jaringan dan menyediakan mekanisme keamanan. Penggunaan *delay* yang panjang bertujuan untuk meningkatkan kemampuan jaringan dalam menyediakan penjadwalan dan keputusan pemilihan jalur yang akan dilalui. Penggunaan penyimpanan dalam jaringan bertujuan untuk mendukung operasi *store and forward* pada beberapa jalur dimana tidak terdapat jalur *end-to-end* dan dalam rentang waktu yang panjang.

Pada awalnya DTN digunakan untuk komunikasi luar angkasa (*Interplanetary Network*). Pada komunikasi luar angkasa tidak dapat menggunakan TCP/IP karena koneksi *end-to-end* yang tidak selalu ada dan *memiliki delay* pengiriman yang lama (Fall, 2003). Seiring perkembangannya, DTN dapat digunakan juga *Military AdHock Network* yaitu jaringan yang digunakan oleh pasukan militer yang ditempatkan di daerah-daerah terpencil dan tidak memiliki koneksi yang memadai. DTN juga dapat digunakan sebagai jaringan dengan media penghantar gelombang radio contohnya *Handy Talkie* (HT) dan jaringan sensor/actuator seperti penerapan *Wireless Sensor Network* (WSN).

2.2.1.1 Store and Forward

Metode *store and forward* adalah metode komunikasi dalam DTN. Metode ini menggunakan lapisan baru untuk menyimpan dan meneruskan pesan yang disebut lapisan *bundle*. Metode ini bekerja dengan meneruskan seluruh pesan atau sebagian dari pesan (*fragment*) yang disimpan di sebuah node. Pesan akan diteruskan dari satu node ke node lain apabila telah terdapat koneksi (Rahmania, 2013). Metode *store and forward* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Metode *Store and Forward*
Sumber: Ramhania (2013)

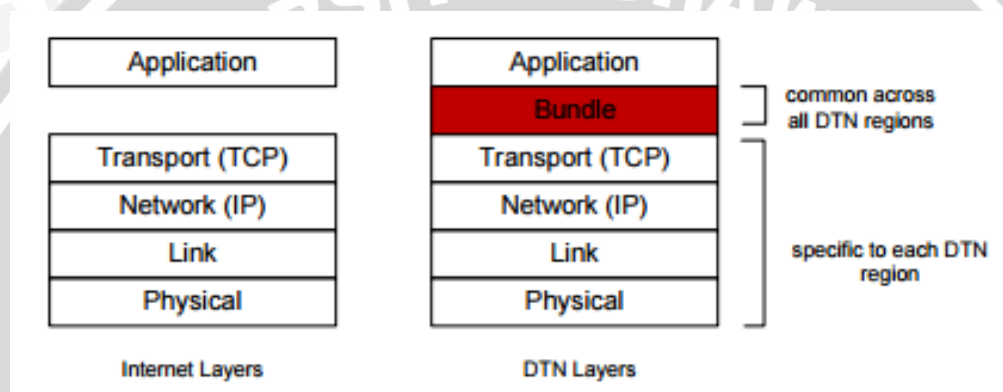
Pada Gambar 2.1, menunjukkan pada proses pengiriman data dari Node A dengan tujuan Node D. Pada TCP/IP, yang dalam pengirimannya router hanya menerima data dan langsung memforward. Jika koneksi terputus di tengah-tengah proses pengiriman, data yang dikirim akan langsung hilang. Pada DTN dengan metode *store and forward*, data akan terlebih dahulu disimpan sebelum dikirimkan. Data akan dikirimkan kembali saat koneksi telah siap. Dalam menggunakan metode *store and forward*, setiap node harus memiliki media penyimpanan yang digunakan untuk menyimpan data apabila terjadi gangguan

koneksi ditengah-tengah proses pengiriman. Media penyimpanan yang dapat digunakan adalah *hard disk* dan *flash memory*.

Metode *store and forward* saat ini telah banyak digunakan dalam sistem voicemail dan email. Metode *store and forward* menjadi pilihan yang tepat untuk dapat meringankan permasalahan yang ada di daerah terpencil yang memiliki koneksi tidak stabil. Permasalahan yang dapat diselesaikan terutama dalam lingkup pertukaran informasi atau pengiriman data.

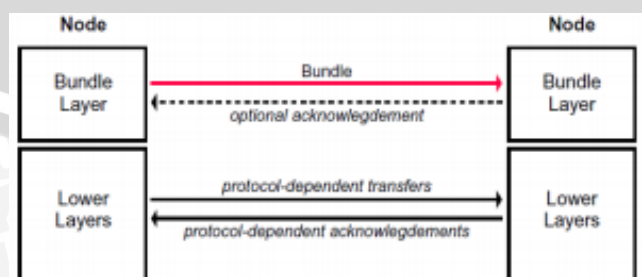
2.2.1.2 Bundle

Dalam DTN, proses metode *store and forward* dilakukan pada layer tambahan yaitu *bundle layer*. Data yang tersimpan sementara disebut sebagai *bundle*. *Bundle layer* adalah sebuah layer tambahan untuk memodifikasi paket. *Bundle layer* terletak diantara layer aplikasi dan layer transport seperti pada Gambar 2.2.



Gambar 2.2 Letak *Bundle Layer*
Sumber: Siswanti (2013)

Pada Gambar 2.2 dapat dilihat perbedaan dari layer-layer pada internet dengan protokol TCP/IP dan layer DTN. DTN tidak hanya beroperasi di jaringan TCP/IP. Layer dibawah *bundle* dapat berupa protokol-protokol lain. Oleh karena itu, DTN dapat menjadi perantara jaringan dengan protokol yang berbeda. *Bundle layer* menyimpan dan meneruskan pesan atau sebagian pesan dari satu node ke node lain. *Bundle layer* saling berkomunikasi dengan menggunakan sesi sederhana yang terdapat fasilitas ACK. Pengiriman ACK antar *bundle layer* dapat dilihat pada Gambar 2.3.

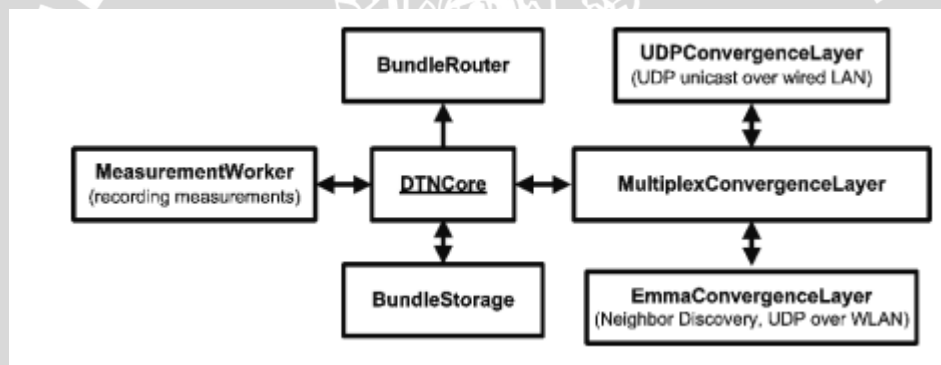


Gambar 2.3 ACK pada *Bundle Layer*
Sumber: Rahmania (2013)

2.2.2 IBR-DTN

Delay Tollerant Network (DTN) merupakan sebuah jaringan yang menawarkan komunikasi untuk jaringan yang tidak selalu terdapat koneksi *end-to-end*. Dalam beberapa tahun terakhir DTN menjadi sangat populer untuk diterapkan dan banyak dikembangkan. Salah satu pengembangan dari DTN adalah IBR-DTN. IBR-DTN adalah perangkat lunak yang efisien untuk *embedded system* (Yuliandoko dkk, 2015).

Saat ini IBR-DTN dikembangkan pada Mikrotik Routerboard 532, yang dilengkapi dengan konsol serial untuk debugging. IBR-DTN terdiri dari tiga modul utama yang menyediakan layanan dasar bagi modul DTN. Modul tersebut adalah *Bundle Router*, *Bundle Storage* dan *Convergence Layer*. *Bundle router* dirancang untuk memungkinkan *modular routing* yang *plug-in*. *Bundle storage* adalah manajemen, penyimpanan dan pengambilan *bundle*. Selain itu, *bundle storage* memberikan informasi setiap kali *bundle* yang telah lama dihapus sehingga laporan status dapat dikirim jika diperlukan. *Convergence layer* menyediakan adaptor untuk mekanisme transportasi. Arsitektur perangkat lunak IBR-DTN dapat dilihat pada Gambar 2.4.



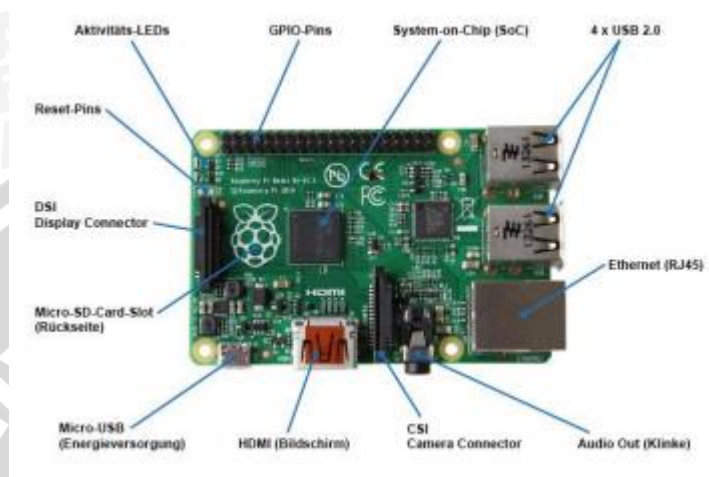
Gambar 2.4 Arsitektur IBR-DTN
Sumber: Doering,dkk (2008)

2.2.3 Raspberry Pi

Raspberry Pi adalah mini komputer yang awalnya dibuat hanya untuk keperluan edukasi. Seiring perkembangan zaman, penggunaan Raspberry Pi tidak hanya untuk edukasi. Penggunaannya sudah dikembangkan untuk membuat proyek-proyek seperti *game console*, instrument musik, dan robot. Untuk mengoperasikannya, komputer mini ini dilengkapi dengan port I/O untuk menghubungkan monitor komputer atau layar TV, *keyboard* dan *mouse*. Meskipun ukurannya jauh lebih kecil dibandingkan komputer pada umumnya, tidak mengurangi kemampuan Raspberry Pi dalam melakukan pengolahan kata, memutar video bahkan memainkan permainan seperti yang dilakukan *computer desktop* (Raspberry Pi Foundation, 2012).

Penggunaan Raspberry Pi dapat disesuaikan dengan kebutuhan penggunanya. Beberapa penggunaan *Raspberry Pi* diantaranya adalah sebagai *desktop* mini, *file server*, *download server*, *access point*, *server DNS* dan *multimedia player*

(Ristianto, 2015). Terdapat 8 model Raspberry Pi dengan spesifikasi yang berbeda, yaitu Raspberry Pi model A, Raspberry Pi model A+, Raspberry Pi model B, Raspberry Pi model B+, Raspberry Pi model compute modul, Raspberry Pi versi 2, Raspberry Pi Zero, dan Raspberry Pi versi 3 yang dilengkapi dengan Wi-Fi dan Bluetooth (Suranata, 2015). Komponen-komponen Raspberry Pi model B+ dapat dilihat pada Gambar 2.5.

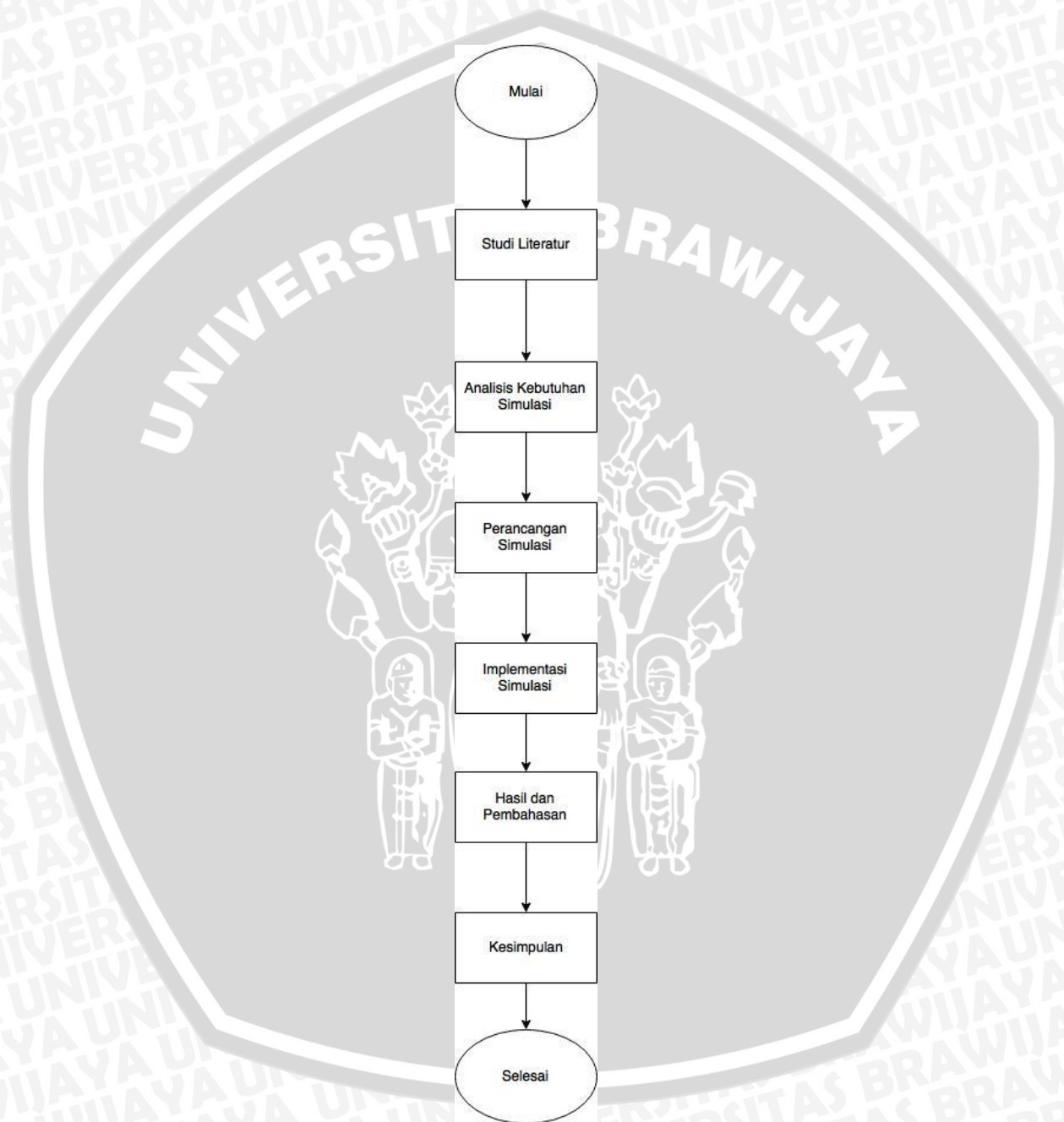


Gambar 2.5 Komponen Raspberry Pi Model B+
Sumber: (Raspberry Pi Foundation, 2012)

Pada komponen internal Raspberry Pi tidak dilengkapi dengan Network Interface Card, sehingga dibutuhkan USB wireless adapter agar Raspberry Pi memiliki koneksi *wireless*. Sistem operasi yang dijalankan pada Raspberry Pi adalah *Raspbian Jessie*.

BAB 3 METODOLOGI

Pada bab ini dijelaskan metode-metode penelitian yang dilakukan dalam penelitian, analisis kebutuhan, perancangan, dan teknik yang digunakan dalam pengujian. Gambar 3.1 menjadi gambaran diagram alir proses penelitian.



Gambar 3.1 Diagram Alir Proses Penelitian

3.1 Studi Literatur

Studi literatur adalah kegiatan yang dilakukan untuk menentukan objek penelitian yang sesuai dengan topik yang diambil. Dalam studi literatur dapat juga digunakan untuk referensi dalam melakukan penelitian. Studi literature berfungsi untuk mendukung penelitian dalam menyelesaikan masalah agar tercapainya tujuan penelitian. Teori Teori pendukung dapat diperoleh dari jurnal, buku dan penelitian sebelumnya. Adapun studi literatur dari penelitian ini antara lain *Delay Tolerant Network (DTN)*, *Metode Store and Forward, Bundle*, *IBR-DTN* dan *Raspberry Pi*.

3.2 Analisis Kebutuhan Simulasi

Analisis kebutuhan didapat dari hasil studi literatur yang sudah disusun menjadi dasar teori yang menunjang penelitian. Kebutuhan dari penelitian ini dibagi menjadi kebutuhan perangkat keras dan perangkat lunak.

3.2.1 Perangkat Keras

Dalam membangun infrastruktur berbasis IBR-DTN diperlukan 2 buah raspberry pi yang berfungsi sebagai pengirim dan penerima informasi (*file*) dan kamera modul yang berfungsi untuk menangkap gambar. Dalam penelitian ini memakai 2 buah raspyberry pi yang berbeda dimana raspberrypi 2 model B sebagai Pengirim dan raspberrypi 3 model B sebagai penerima. Spesifikasi dari masing masing raspberry pi yang akan di gunakan adalah sebagai berikut:

1. Raspberrypi 2 model B:

- Processor : ARM 900MHz quad-core ARM Cortex-A7
- RAM : 1GB
- Interface : Wireless LAN
- Wifi : Ralink RT5730
- Hardisk : SD Card 8GB

2. Raspberrypi 3 model B:

- Processor : ARM 1.2GHz 64-bit quad-core ARMv8
- RAM : 1GB
- Interface : Wireless LAN
- Wifi : 802.11
- Bluetooth : 4.1
- Hardisk : SD Card 32GB

Sedangkan untuk kamera modul yang di gunakan adalah kamera module v2 dengan spesifikasi sebagai berikut:

- Resolusi gambar : 2592 x 1944
- Resolusi video : 1080p @ 30fps, 720p @ 60 fps dan merekam 640x480p 60/90

- Interface : 15pin CSI
- Ukuran : 20x25x9mm

3.2.2 Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam penelitian ini adalah yang dapat menyediakan layanan pengiriman *file* dengan kondisi jaringan yang tidak selalu tersedia secara terus menerus. Delay Tolerant Network digunakan untuk mengatasi masalah konektivitas yang tidak tersedia secara terus menerus. Salah satu perangkat lunak yang digunakan untuk mendukung DTN adalah IBR-DTN. Perangkat lunak lain yang dibutuhkan adalah *hostapd* dan *isc-dhcp-server*, dimana kedua perangkat lunak tersebut berfungsi untuk membuat jaringan access point yang akan digunakan sebagai jaringan penghubung antar node DTN.

3.3 Perancangan Sistem

Dalam mengimplementasikan sebuah sistem, dibutuhkan proses perancangan sistem yang merupakan proses merancang atau mendesain sebuah sistem yang isinya adalah kebutuhan serta prosedur teknik yang akan di buat. Pada tahap ini akan dijelaskan mengenai perancangan sistem yang akan dibangun. Perancangan sistem terdiri dari proses instalasi dan konfigurasi IBR-DTN, Proses konfigurasi kamera modul dan pembangunan topologi.

3.3.1 Instalasi dan Konfigurasi IBR-DTN

IBR-DTN merupakan salah satu perangkat lunak yang merupakan pengembangan dari DTN. Sebelum menggunakan IBR-DTN, terdapat proses instalasi dan konfigurasi. Langkah-langkah instalasi dan konfigurasi IBR-DTN dapat dilihat pada Gambar 3.2.

Adapun penjelasan mengenai tahapan instalasi dan konfigurasi IBR-DTN sebagai berikut:

1. Install dan konfigurasi OS

Pada tahap ini dilakukan instalasi sistem operasi yang berguna sebagai wadah penyedia layanan umum untuk perangkat lunak. Perangkat lunak IBR-DTN dapat berjalan pada banyak sistem operasi salah satu contohnya adalah Raspbian. Raspbian adalah salah satu sistem operasi yang dapat berjalan pada perangkat keras Raspberry pi. Pada tahap

2. Update OS

Pada tahap ini dilakukan *update* pada sistem operasi yang berguna untuk memperbarui library yang telah ada pada sistem operasi yang terinstall pada perangkat lunak.

3. Install dan konfigurasi IBR-DTN

Pada tahap ini dilakukan instalasi dan konfigurasi IBR-DTN sebagai perangkat lunak yang mendukung DTN. Instalasi IBR-DTN dilakukan pada saat sistem operasi telah melakukan update dan menambahkan *repository*. IBR-DTN

dapat dikonfigurasi sesuai kebutuhan peneliti, dengan mengkonfigurasi pada *file ibrdtn.conf*.

4. Test Daemon

Pada tahap ini dilakukan percobaan untuk menjalankan daemon dari IBR-DTN yang berfungsi untuk mengetahui apakah konfigurasi yang telah dilakukan dapat berjalan atau tidak pada daemon IBR-DTN.



Gambar 3.2 Langkah Installasi dan Konfigurasi IBR-DTN

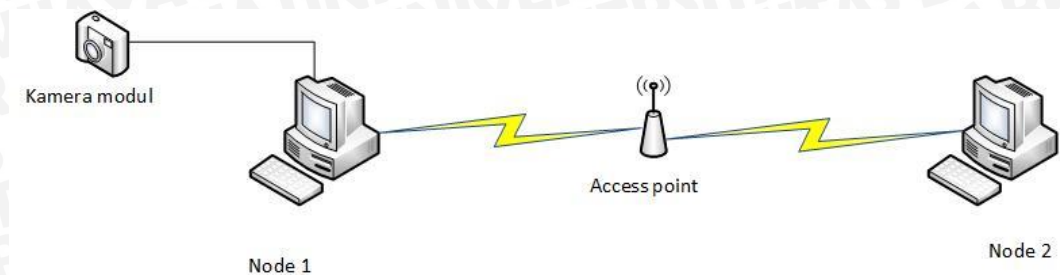
3.3.2 Konfigurasi Kamera Modul

Kamera modul adalah perangkat keras pendukung yang dapat dihubungkan pada raspberrypi yang berguna sebagai penangkap gambar dan perekam video. Kamera modul tidak bisa untuk digunakan langsung seperti kamera pada perangkat keras lainnya akan tetapi dengan melakukan konfigurasi pada raspberrypi untuk mengaktifkan fungsi kamera. Untuk mengakap gambar, kamera modul juga membutuhkan kode program untuk mengambil gambar yang di realisasikan dalam Bahasa python atau juga bisa menggunakan *command* pada terminal yang telah disediakan raspberry.

3.3.3 Topologi

Pada Gambar 3.3 dijelaskan terdapat dua buah node DTN yang dihubungkan oleh jaringan *access point*, dimana node 1 sebagai pengirim informasi dan node 2 sebagai penerima informasi. Pada node 1 disematkan sebuah kamera modul yang

berfungsi untuk menangkap gambar secara realtime dan node 1 akan mengirimkan gambar yang ditangkap oleh kamera saat itu juga ke node 2.



Gambar 3.3 Topologi Simulasi

3.4 Implementasi Simulasi

Pada tahap ini, dilakukan pembangunan infrasturktur dari sistem pengiriman gambar secara realtime yang sudah dilakukan pada tahap sebelumnya. Implementas infrastuktur dapat dimulai dengan menyiapkan perangkat keras dan perangkat lunak yang sesuai dengan kebutuhan. Tahap selanjutnya adalah dengan melakukan instalasi dan konfigurasi raspberrypi dan kamera modul. Setelah perngkat keras siap digunakan, masuk ke tahap melakukan instalasi dan kofigurasi *acces point* yang berguna untuk penguhung antar node DTN, dan terakhir melakukan instalasi dan konfigurasi *perangkat lunak* IBR-DTN.

3.5 Hasil dan Pembahasan

Pengujian yang dilakukan oleh peneliti adalah dengan menggunakan *perangkat lunak* IBR-DTN. Pengujian menggunakan IBR-DTN dilakukan untuk mengukur kinerja dan performa infrastruktur, apakah telah memenuhi spesifikasi kebutuhan yang melandasinya. Pengujian yang akan dilakukan mengacu pada penelitian yang dilakukan oleh Evan Hardyanto Prakasita.

Terdapat tiga pengujian untuk skenario yang telah ditentukan, yaitu:

1. Pengiriman *file* dengan koneksi putus-sambung.
2. Pengiriman *file* dengan jumlah banyak.
3. Pengiriman *file* selama waktu tertentu.

3.6 Kesimpulan

Pengambilan kesimpulan dilakukan saat semua tahapan perancangan dan implementasi sistem yang dibuat telah selesai dilaksanakan dan sesuai dengan teori penerapan. Kesimpulan dibuat untuk menjawab rumusan masalah yang telah ditetapkan sebelumnya dan menjadi panutan untuk penelitian kinerja dan performa pengiriman *file* pada IBR-DTN selanjutnya.

BAB 4 IMPLEMENTASI SIMULASI

Pada bab ini akan dijelaskan langkah langkah implementasi dan pengujian yang sesuai dengan skenario yang telah di tentukan. Pada tahap implementasi akan dijeaskan bagaimana perangkat keras ataupun perangkat lunak. Pada tahap pengujian dilakukan penujian terhadap sistem yang telah diimplementasikan.

4.1 Implemantasi

Hasil dari analisis kebutuhan dan perancangan sistem yang telah dilakukan sebelumnya menghasilkan landasan dasar implementasi. Implementasi dilakukan pada perangkat keras dan perangkat lunak yang telah di siapkan. Tahapan-tahapan yang dilakukan pada proses implementasi ini antara lain access point dan IBR-DTN.

4.1.1 Access Point

Untuk menghubungkan node-node agar saling terhubung, maka diperlukan sebuah jaringan yang akan *mengcover* node-node terebut agar saling terhubung. Pada penelitian ini menggunakan jaringan *access point* yang akan menghubungkan node-node agar dapat melakukan pengiriman *file*.

4.1.1.1 Instalasi

Sebelum menggunakan *access point*, maka terlebih dahulu melakukan proses instalasi. *Perangkat lunak* yang perlu di install agar *access point* ini dapat bekerja adalah *hostapd*, *isc-dhcp-server*, dan *iptables-persistent*. *Hostapd* dan *isc-dhcp-server* adalah perangkat lunak ini yang digunakan untuk membuat *access point*. Untuk melakukan instalasi *hostapd* dan *isc-dhcp-server*, maka perlu menjalankan perintah yang di bawah.

```
$ sudo apt-get install hostapd isc-dhcp-server
```

4.1.1.2 Konfigurasi

a. Konfigurasi dhcp, isc-dhcp-server dan interfaces WLAN0

Setelah melalukan instalasi *hostapd*, *isc-dhcp-server* dan *iptables-persistent*, selanjutnya masuk ketahap konfigurasi. Sebelum masuk ke tahap konfigurasi *access point*, terlebih dahulu melakukan konfigurasi DHCP server dan juga melakukan setting pada interface *wlan0*. Selanjutnya mengedit *file dhcpd.conf* dengan memasukan perintah yang di bawah.

```
$ sudo nano /etc/dhcp/dhcpd.conf
```

Temukan baris yang berisi kode di bawah dan tambahkan tanda pagar (#) di setiap awal baris.

```
option domain-name "example.org";  
option domain-name-servers ns1.example.org,  
ns2.example.org;
```

Setelah kode diatas di rubah, temukan baris yang berisi kode di bawah dan hilangkan tanda pagar di awal baris.

```
#authoritative;
```

Setelah kode diatas dirubah, tambahkan kode di bawah pada akhir layar.

```
subnet 192.168.42.0 netmask 255.255.255.0 {
    range 192.168.42.10 192.168.42.50;
    option broadcast-address 192.168.42.255;
    option routers 192.168.42.1;
    default-lease-time 600;
    max-lease-time 7200;
    option domain-name "local";
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Setelah melakukan konfigurasi pada dhcpd.conf, langkah selanjutnya masuk untuk melakukan konfigurasi isc-dhcp-server. Jalankan perintah dibawah untuk masuk konfigurasi isc-dhcp-server dan isi pada INTERFACES="" menjadi INTERFACES="wlan0".

```
sudo nano /etc/default/isc-dhcp-server
```

Untuk konfigurasi dhcp dan isc-dhcp-server sudah selesai, selanjutnya konfigurasi interfaces wlan0 untuk memberikan ip static pada interfaces wlan0.

```
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
```

b. Konfigurasi Access Point.

Konfigurasi *access point* berfungsi untuk memberikan nama *access point* dan juga memberikan *password protect*. Untuk masuk konfigurasi *access point*, masuk dengan perintah di bawah ini dan lakukan konfigurasi dengan kode yang di bawah (Lady, 2016).

```
$ sudo nano /etc/hostapd/hostapd.conf
```

Kemudian, tambahkan baris berikut:

```
interface=wlan0
driver=nl80211
ssid=raspi
channel=1

macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```

```
wpa=2
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

ieee80211n=1
hw_mode=g
```

Setelah melakukan konfigurasi terhadap *access point*, selanjutnya melakukan konfigurasi daemon agar *access point* dapat berjalan otomatis dengan masuk ke perintah.

```
$ sudo nano /etc/default/hostapd
```

Setelah masuk ke konfigurasi, ganti baris kode `#DAEMON_CONF=""` menjadi `DAEMON_CONF="/etc/hostapd/hostapd.conf"`.

Setelah melakukan konfigurasi, *access point* telah siap di gunakan. Untuk menjalankan *access point* dapat dilakukan dengan memasukan perintah dibawah, cukup memasukan perintah satu kali saja dikarenakan *daemon* akan berjalan secara otomatis saat perangkat di hidup matikan.

```
$ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

Access point yang telah berhasil tersambung oleh perangkat lain akan memberikan tampilan seperti pada Gambar 4.1.

```
root@raspy1:/home/pi# sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
sudo: unable to resolve host raspy1
Configuration file: /etc/hostapd/hostapd.conf
Using interface wlan0 with hwaddr 7c:dd:90:69:57:c6 and ssid "raspi"
random: Only 19/20 bytes of strong random data available from /dev/random
random: Not enough entropy pool available for secure operations
WPA: Not enough entropy in random pool for secure operations - update keys later when the first station connects
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA b8:27:eb:13:59:23 IEEE 802.11: authenticated
wlan0: STA b8:27:eb:13:59:23 IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED b8:27:eb:13:59:23
wlan0: STA b8:27:eb:13:59:23 RADIUS: starting accounting session 5877C503-00000000
wlan0: STA b8:27:eb:13:59:23 WPA: pairwise key handshake completed (RSN)
```

Gambar 4.1 *Access Point* yang telah tersambung

Pada Gambar 4.1 dapat terlihat bahwa status dari *access point* dengan SSID “raspi” telah berstatus “ENABLE” yang berarti *access point* telah siap untuk digunakan. Kemudian, suatu perangkat dengan *mac address* b8:27:eb:13:59:23 telah tersambung dengan *access point* “raspi”.

4.1.2 IBR-DTN

IBR-DTN adalah adalah salah satu perangkat lunak pengembangan dari DTN. IBR DTN merupakan salah satu dari sekian banyak DTN yang menggnakan bundle protocol sebagai standar protokolnya.

4.1.2.1 Instalasi

Instalasi IBR-DTN dilakukan pada setiap raspberrypi yang akan menjadi node untuk dapat saling mengirim dan menerima informasi dalam keadaan jaringan

yang tidak selalu ada koneksi secara terus menerus. Untuk melakukan instalasi IBR-DTN pada raspberry pi, diperlukan penambahann repositori. Repository yang perlu di tambahkan adalah:

```
deb http://jenkins.ibr.cs.tu-bs.de/download/debian/ wheezy
main
```

Setelah selesai penambahan reporitory pada raspberrypi, maka dilakukan proses update pada raspberrypi. Sebelum melakukan instalasi, maka perlu menjalankan perintah dibawah terlebih dahulu.

```
$ wget -O - http://jenkins.ibr.cs.tu-
bs.de/download/repository.gpg.key | apt-key add -
```

Setelah perintah tersebut dijalankan, maka dilakukan proses update kembali pada raspberrypi dan instalasi IBR-DTN dengan menjalankan perintah di bawah

```
$ sudo apt-get install ibrdtn
```

4.1.2.2 Konfigurasi

Untuk dapat menggunakan IBR-DTN sesuai kebutuhan, sebelum menjalankan perangkat lunak IBR-DTN diperlukan beberapa konfigurasi. Konfigurasi dilakukan dengan cara mengedit *file* konfigurasi ibrdtn.conf yang berada di /etc/ibrdtn seperti pada Gambar 4.2.

```
GNU nano 2.2.6 File: ibrdtn.conf
#####
# IBR-DTN daemon
#####
#
# the local eid of the dtn node
# default is the hostname
#
local_uri = dtn://raspy2.dtn
#
# specifies an additional logfile
#
logfile = /var/log/ibrdtn/ibrdtn.log
#
# Limit the block size of all bundles.
# The value accepts different multipliers.
# G = 1,000,000,000 bytes
# M = 1,000,000 bytes
# K = 1,000 bytes
#limit_blocksize = 1.3G
#
# Limit the block size of foreign bundles.
# Foreign bundles are not address from or to the
# local node.
#
# The value accepts different multipliers.
# G = 1,000,000,000 bytes
# M = 1,000,000 bytes
# K = 1,000 bytes
^G Get Help      ^O WriteOut      ^R Read File      ^Y Prev Page      ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify        ^W Where Is       ^V Next Page      ^U UnCut Text    ^T To Spell
```

Gambar 4.2 konfigurasi IBR-DTN

Pada Gambar 4.2 terdapat beberapa hal yang perlu di atur untuk penelitian ini adalah:

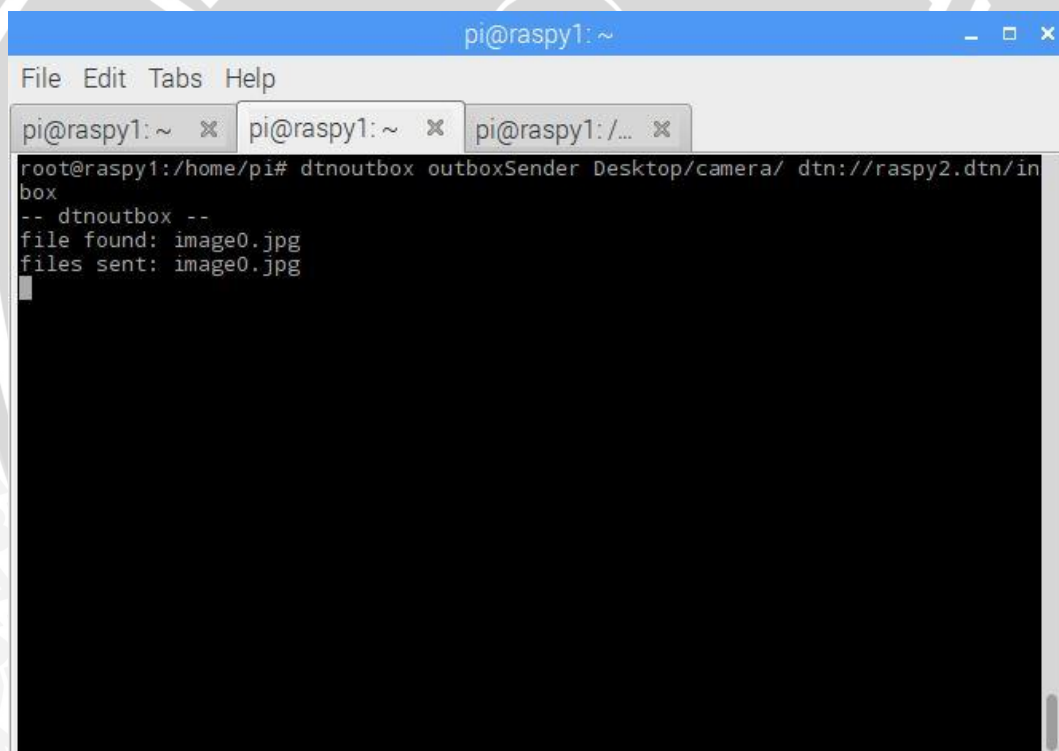
- Nama id untuk node dtn yang dapat diubah menjadi dtn://(nama node).dtn.
- Lifetime juga perlu untuk diatur dikarenakan lifetime sangat berperan untuk menentukan lama penyimpanan bundle apabila terjadinya perangkat keras yang tidak mendapatkan jaringan.

- Penyimpanan log dapat diatur di dalam `ibrdtn.conf` dengan cara mengisi alamat *file* yang akan diisi oleh log.
- Penyimpanan *file* yang telah diterima dapat diatur dengan cara mengisi `storage_path` dengan lokasi yang diinginkan.
- Kapasitas `storage_path` dapat diatur dengan mengisi `limit_storage`.

4.2 Pengujian Test Pengiriman gambar

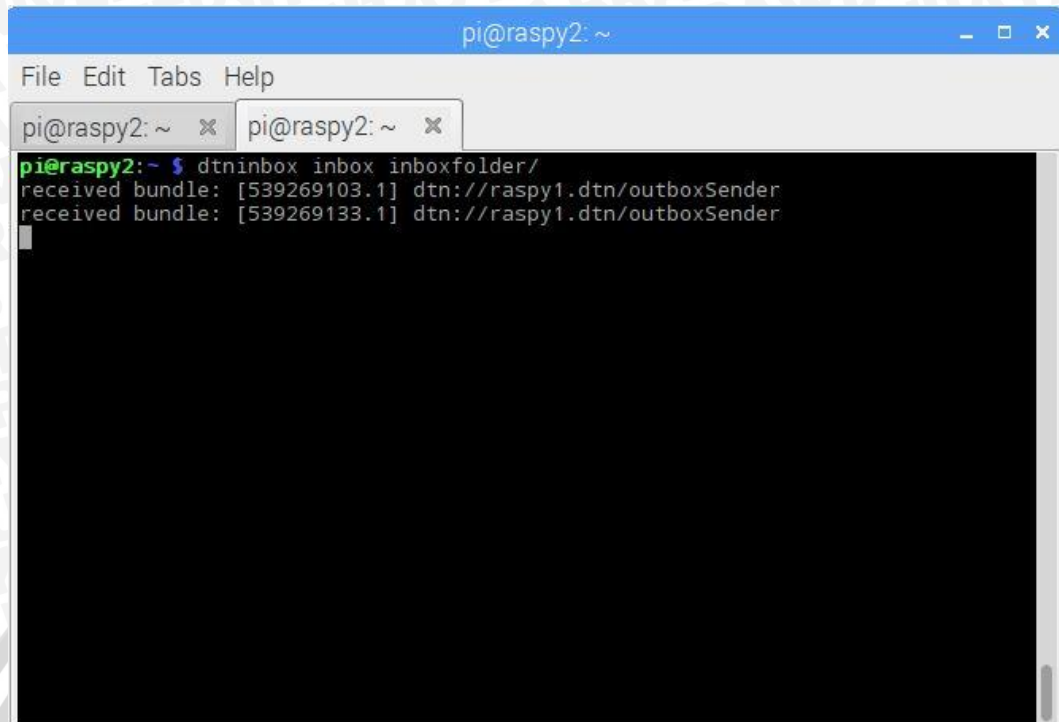
Pada tahap ini dilakukan pengujian untuk mencoba mengirimkan *file* tangkap gambar pada simulasi yang telah dibuat. Pengujian ini dilakukan untuk mengetahui apakah simulasi yang telah dibuat dapat bekerja sesuai tujuan.

Pertama kali pastikan *daemon* IBR-DTN pada kedua node berjalan dan dapat saling terhubung dengan melakukan pengujian ping kepada kedua node. Setelah kedua node saling terhubung, maka dilakukan pengujian untuk melakukan pengiriman file secara otomatis dengan menjalankan kode program yang telah dibuat dan juga menjalankan *tools* dari IBR-DTN.



Gambar 4.3 File gambar ditemukan oleh dtnoutbox

Pada Gambar 4.3, *tools* dtnoutbox telah menemukan *file* gambar hasil tangkap dari kamera yang berada pada *folder* camera dan melakukan pengiriman menuju alamat `dtn://raspy2.dtn/inbox`. Kemudian *tools* dtninbox menerima file yang telah dikirim oleh node pengirim lalu menyimpan hasil kiriman yang didapat menuju inboxfolder seperti pada Gambar 4.4. Pada *tools* dtninbox dapat memberikan keterangan seri bundle yang telah diterima dan asal file yang diterima oleh node penerima.



The image shows a terminal window titled 'pi@raspy2: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. There are two tabs, both labeled 'pi@raspy2: ~'. The terminal content shows a command 'dtinbox inbox inboxfolder/' being executed, followed by two lines of output: 'received bundle: [539269103.1] dtn://raspy1.dtn/outboxSender' and 'received bundle: [539269133.1] dtn://raspy1.dtn/outboxSender'. The cursor is at the end of the second line.

```
pi@raspy2: ~ $ dtinbox inbox inboxfolder/
received bundle: [539269103.1] dtn://raspy1.dtn/outboxSender
received bundle: [539269133.1] dtn://raspy1.dtn/outboxSender
```

Gambar 4.4 File gambar diterima oleh dtinbox



BAB 5 HASIL DAN PEMBAHASAN

Pada Bab ini akan dijelaskan mengenai skenario pengujian yang dilakukan dan juga hasil dari pengujian yang dilakukan.

5.1 Skenario Pengujian

Pengujian dilakukan untuk mengetahui performa dan kinerja dari sebuah simulasi yang telah dibuat. Terdapat beberapa skenario pengujian yang telah dirancang sebelumnya antara lain adalah pengujian pengiriman *file* dengan konektivitas putus-sambung, pengujian dengan mengirimkan *file* dalam jumlah tertentu dan yang terakhir adalah pengiriman *file* dengan lama waktu tertentu

5.1.1 Pengujian Pengiriman *File* dengan Konektivitas Putus-Sambung

Pada tahap ini dilakukan pengujian pengiriman dari *device* pengirim ke *device* penerima dengan kondisi konektivitas putus-sambung. Node IBR-DTN pertama merupakan pengirim yang telah dilengkapi dengan kamera modul untuk menangkap gambar secara *realtime*, node IBR-DTN kedua berfungsi sebagai penerima. Untuk dapat menangkap gambar dan mengirimkannya melalui IBR-DTN, maka pada node pengirim dibuat sebuah program seperti pada Tabel 5.1.

Tabel 5.1 *camera.py* pengujian konektivitas putus-sambung

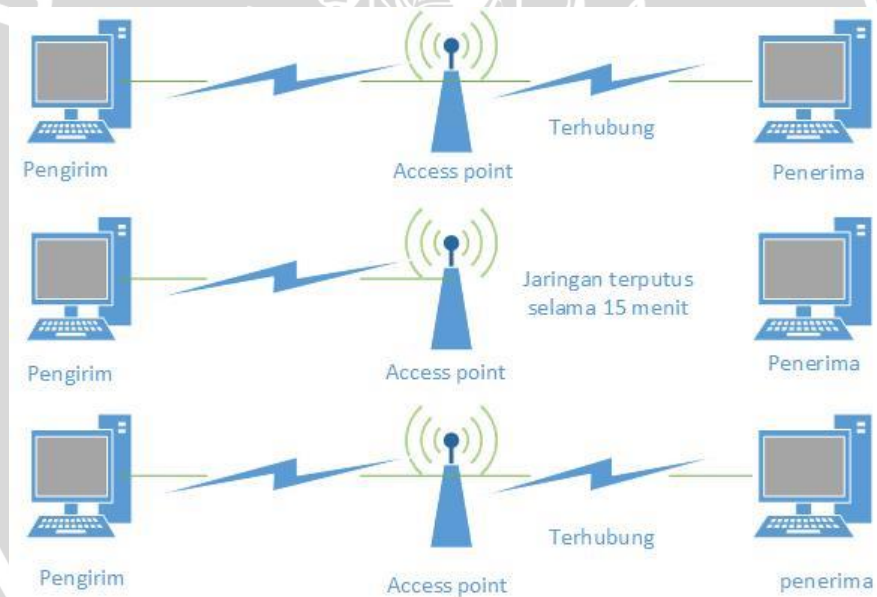
camera.py	
Baris	Source Code
1	import pikamera
2	from time import sleep
3	from subprocess import call
4	
5	kamera = pikamera.PiKamera()
6	
7	
8	kamera.start_preview()
9	for i in range(100):
10	
11	kamera.capture('/home/pi/Desktop/kamera/image%s.jpg' % i)
12	file='/home/pi/Desktop/kamera/image%s.jpg' % i
13	print file
14	sleep(30)
15	kamera.stop_preview()

Source code pada Tabel 5.1 berfungsi untuk menangkap gambar secara otomatis selama dua menit. Sedangkan untuk node penerima tidak menjalankan perintah apapun dikarenakan *file* yang terikirim akan masuk secara otomatis ke dalam *storage_path*. Penjelasan *source code* program dari Tabel adalah sebagai berikut:

- Baris 5 adalah inisialisasi modul kamera yang disimpan kedalam variabel kamera.
- Baris 8 adalah fungsi kamera mulai dinyalakan.

- Baris 9 adalah perulangan yang akan dilakukan sebanyak 100 kali. Perulangan ini adalah banyaknya data yang akan dikirim.
- Baris 11 adalah fungsi pengambilan gambar melalui kamera dan disimpan dalam folder /home/pi/Dekstop/kamera.
- Baris 12 adalah Inisialisasi variabel file yang berisi keterangan gambar yang telah di ambil
- Baris 13 adalah menampilkan variable file yang telah dinisialisasi pada baris 12
- Baris 14 adalah fungsi untuk mengatur jeda waktu pengiriman *file* selama 2 menit.
- Baris 15 adalah fungsi untuk menutup kamera.

Pada saat kamera menangkap gambar, secara otomatis *file* akan tersimpan di *folder* camera yang berada pada Dekstop. Secara bersamaan, akan mengirimkan gambar yang di tangkap menuju node yang bernama raspy2 menggunakan *tools* dari IBR-DTN yaitu dtnoutbox dan dtninbox. Pengujian dengan konektivitas putus-sambung dapat dilihat pada Gambar 5.1.



Gambar 5.1 pengujian konektivitas putus-sambung

Pada Gambar 5.1, digambarkan bahwa node pengirim masih tersambung dengan node penerima. Setelah itu, node penerima dilakukan dua perlakuan pemutusan jaringan dan mematikan secara paksa raspberry pi selama 15 menit, setelah 15 menit node pengirim dinyalakan kembali dan di sambungkan ke jaringan *access point* agar terhubung dengan node pengirim.

5.1.2 Pengiriman *File* dengan Jumlah Tertentu

. Pada node pengirim dilengkapi dengan kamera modul yang berguna untuk menangkap gambar secara realtime. Pengujian ini, memiliki 4 buah skenario yaitu:

1. Menangkap gambar dan mengirimkannya sampai 100 gambar,
2. Menangkap gambar dan mengirimkannya sampai 200 gambar,
3. Menangkap gambar dan mengirimkannya sampai 300 gambar,
4. Menangkap gambar dan mengirimkannya sampai 500 gambar.

Pengujian ini dilakukan dengan *source code* program seperti pada Tabel 5.2.

Tabel 5.2 *camera.py* pengiriman *file* dengan Jumlah tertentu

camera.py	
Baris	Source Code
1	import pikamera
2	from time import sleep
3	from subprocess import call
4	
5	kamera = pikamera.PiKamera()
6	
7	
8	kamera.start_preview()
9	for i in range(100):
10	
11	kamera.capture('/home/pi/Desktop/kamera/image%s.jpg' % i)
12	file='/home/pi/Desktop/kamera/image%s.jpg' % i
13	print file
14	kamera.stop_preview()

Source code pada Tabel 5.2 berfungsi untuk mengirimkan *file* sebanyak 100 kali. Penjelasan *source code* program dari Tabel 5.2 adalah sebagai berikut:

- Baris 5 adalah inisialisasi modul kamera yang disimpan kedalam variabel kamera.
- Baris 8 adalah fungsi kamera mulai dinyalakan.
- Baris 9 adalah perulangan yang akan dilakukan sebanyak 100 kali. Perulangan ini adalah banyaknya data yang akan dikirim. Pada baris ini parameter dapat diganti sesuai banyaknya *file* yang akan dikirim sesuai pengujian.
- Baris 11 adalah fungsi pengambilan gambar melalui kamera dan disimpan dalam folder /home/pi/Desktop/kamera.
- Baris 12 adalah inisialisasi variabel file yang berisi keterangan gambar yang telah di ambil
- Baris 13 adalah menampilkan variabel file yang telah diinisialisasi pada baris 12
- Baris 14 adalah fungsi untuk menutup kamera.

Pada pengujian ini dilakukan untuk mengetahui perbandingan total waktu yang dibutuhkan untuk pengambilan gambar dan total waktu yang dibutuhkan untuk menerima gambar yang dikirim menggunakan *tools* dtnoutbox.

5.1.3 Pengiriman *File* Selama 30 Menit

Pada pengujian ini dilakukan dengan *source code* program seperti pada Tabel 5.3.

Tabel 5. 3 *camera.py* pengiriman *file* dengan Lama Waktu tertentu

camera.py	
Baris	Source Code
1	import pikamera
2	from time import sleep
3	from subprocess import call
4	
5	kamera = pikamera.PiKamera()
6	i=1
7	
8	kamera.start_preview()
9	while True:
10	
11	kamera.capture('/home/pi/Desktop/kamera/image%s.jpg' %
12	i)
13	file='/home/pi/Desktop/kamera/image%s.jpg' % i
14	print file
15	sleep(30)
16	kamera.stop_preview()

Penjelasan *source code* program dari Tabel 5.3 adalah sebagai berikut:

- Baris 5 adalah inisialisasi modul kamera yang disimpan kedalam variabel kamera.
- Baris 6 adalah inisialisasi variabel i.
- Baris 8 adalah fungsi kamera mulai dinyalakan.
- Baris 9 adalah perulangan yang akan dilakukan sebanyak data yang akan dikirim. Pada baris ini paramaeter dapat diganti sesuai banyaknya *file* yang akan dikirim sesuai pengujian.
- Baris 11 adalah fungsi pengambilan gambar melalui kamera dan disimpan dalam folder /home/pi/Dekstop/kamera.
- Baris 12 adalah Inisialisasi variabel file yang berisi keterangan gambar yang telah di ambil
- Baris 13 adalah menampilkan variable file yang telah dinisialisasi pada baris 12
- Baris 14 adalah untuk mengaatur waktu perulangan selama 30 detik.
- Baris 15 adalah fungsi untuk menutup kamera.

Pada pengujian ini dilakukan untuk mengetahui waktu yang dibutuhkan oleh *tools* dtninbox untuk menerima *file* yang dikirimkan dari node pengirim.

5.2 Pembahasan

Pada sub-bab ini akan membahas hasil pengujian dari tiga skenario yang telah diuji.

5.2.1 Hasil Pengujian Pengiriman *File* dengan Konektivitas Putus-Sambung

Pada pengujian ini dilakukan dengan mengirimkan *file* sebanyak 100 *file* dengan jeda waktu selama 30 detik dari node pengirim ke node penerima. Pada awalnya node pengirim dan *penerima* saling terhubung dan melakukan pengiriman *file*. Dilakukan dua buah perlakuan kepada node penerima dengan mematikan raspberry pi pada node penerima dan melakukan pemutusan jaringan ke *access point* selama 15 menit, kemudian setelah 15 menit node penerima di sambungkan lagi ke jaringan *access point*.

Pada saat node pengirim dan penerima saling terhubung, node pengirim dapat mengirim *file* ke node penerima, pada saat node penerima terputus dengan jaringan *access point*, node pengirim tidak dapat menemukan node penerima, maka dari itu *file* yang akan dikirimkan menuju node penerima disimpan terlebih dahulu di dalam sebuah bundle. Bundle akan disimpan pada node pengirim sampai batas waktu *lifetime* yang telah di atur di dalam *ibrdsn.conf*. pada Gambar 5.2 menunjukkan bahwa saat node penerima tidak ditemukan, *file* yang akan dikirim disimpan di dalam bundle.

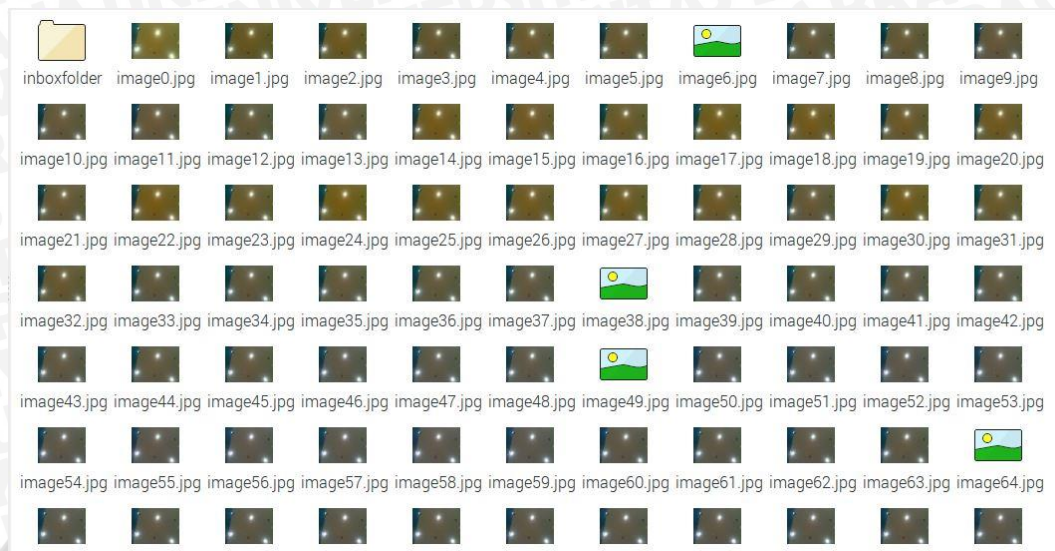
```
root@raspy1:/var/log/ibrdsn# tail ibrdtn.log
Thu Jan 12 18:15:20 2017 NOTICE NodeEvent: Node dtn://raspy2.dtn unavailable
Thu Jan 12 18:16:43 2017 NOTICE BundleCore: Bundle received [537560203.1] dtn://raspy1.dtn/twFCzysPqyEfpPdJ (local)
Thu Jan 12 18:16:43 2017 NOTICE BundleEvent: bundle [537560203.1] dtn://raspy1.dtn/twFCzysPqyEfpPdJ received
Thu Jan 12 18:16:43 2017 NOTICE QueueBundleEvent: New bundle queued [537560203.1] dtn://raspy1.dtn/twFCzysPqyEfpPdJ
Thu Jan 12 18:18:44 2017 NOTICE BundleCore: Bundle received [537560324.1] dtn://raspy1.dtn/FyqeyJwOhCvYrzju (local)
Thu Jan 12 18:18:44 2017 NOTICE BundleEvent: bundle [537560324.1] dtn://raspy1.dtn/FyqeyJwOhCvYrzju received
Thu Jan 12 18:18:44 2017 NOTICE QueueBundleEvent: New bundle queued [537560324.1] dtn://raspy1.dtn/FyqeyJwOhCvYrzju
Thu Jan 12 18:20:44 2017 NOTICE BundleCore: Bundle received [537560444.1] dtn://raspy1.dtn/NnrlWpCGSxgjYqgQ (local)
Thu Jan 12 18:20:44 2017 NOTICE BundleEvent: bundle [537560444.1] dtn://raspy1.dtn/NnrlWpCGSxgjYqgQ received
Thu Jan 12 18:20:44 2017 NOTICE QueueBundleEvent: New bundle queued [537560444.1] dtn://raspy1.dtn/NnrlWpCGSxgjYqgQ
```

Gambar 5.2 Node Penerima tidak Ditemukan

Pada saat node penerima telah tersambung ke jaringan *access point*, maka node penerima akan mengenali node penerima dan mengirimkan bundle yang telah disimpan sementara waktu pada node pengirim. Tidak dibutuhkan waktu lama untuk melakukan pengiriman *file* setelah node penerima terhubung kembali. Hasil pengujian dapat dilihat pada Gambar 5.3 dan Gambar 5.4.

```
^Cpi@raspy2:~$ sudo dtnd -c ibrdtn-repo/ibrdsn/daemon/etc/ibrdsnd.conf -i wlan0
sudo: unable to resolve host raspy2
Thu Jan 12 18:31:12 2017 INFO NativeDaemon: IBR-DTN daemon 1.0.1 (build 517ff60)
Thu Jan 12 18:31:12 2017 INFO Configuration: Configuration: ibrdtn-repo/ibrdsn/daemon/etc/ibrdsnd.conf
Thu Jan 12 18:31:12 2017 INFO NativeDaemon: use logfile for output: /var/log/ibrdsn/ibrdsn.log
Thu Jan 12 18:31:12 2017 INFO NativeDaemon: Parallel event processing enabled using 4 processes.
Thu Jan 12 18:31:12 2017 INFO BundleCore: Local node name: dtn://raspy2.dtn
Thu Jan 12 18:31:12 2017 INFO BundleCore: Forwarding of bundles enabled.
Thu Jan 12 18:31:12 2017 INFO BundleCore: Non-singleton bundles are accepted.
Thu Jan 12 18:31:12 2017 INFO NativeDaemon: using simple bundle storage in /var/spool/ibrdsn/bundles
Thu Jan 12 18:31:12 2017 INFO SimpleBundleStorage: 2 Bundles restored.
Thu Jan 12 18:31:12 2017 INFO NativeDaemon: API initialized using tcp socket: loopback:4550
Thu Jan 12 18:31:12 2017 INFO NativeDaemon: TCP ConvergenceLayer added on wlan0:4556
Thu Jan 12 18:31:12 2017 INFO DiscoveryAgent: listen to [224.0.0.142]:4551
Thu Jan 12 18:31:12 2017 INFO DiscoveryAgent: listen to [ff02::142]:4551
Thu Jan 12 18:31:12 2017 INFO DiscoveryAgent: add interface wlan0
Thu Jan 12 18:31:12 2017 INFO NativeDaemon: Using prophet routing extensions with GRTR as forwarding strategy.
Thu Jan 12 18:31:12 2017 INFO ProphetRoutingExtension: Initializing PROPHET routing module
```

Gambar 5.3 Node Pengirim Terhubung Kembali



Gambar 5.4 File Masuk ke Node Penerima

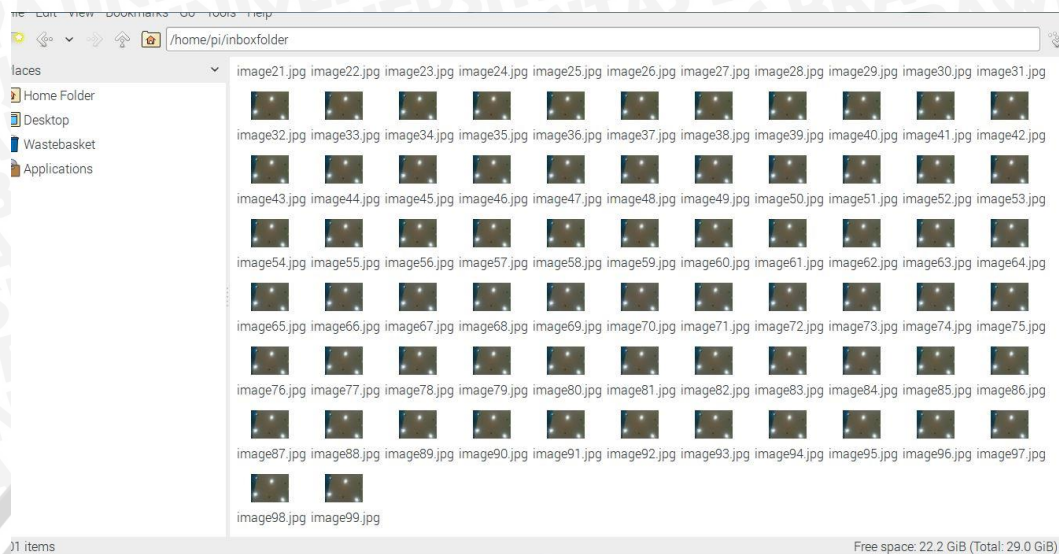
Pada Gambar 5.3 dan Gambar 5.4 membuktikan node pengirim terhubung kembali pada jam 18:31 dan pada saat itu juga *file* masuk ke node penerima. Sehingga, dapat membuktikan untuk waktu tunggu pengiriman kembali tidak memakan waktu yang lama, akan tetapi terdapat *file* yang rusak jika diberikan perlakuan mematikan secara paksa raspberry pi pada node penerima. Hal ini dikarenakan proses render *file* yang telah diterima tidak berjalan dengan sukses, berbeda dengan perlakuan memutuskan jaringannya saja dimana tidak terdapat file yang rusak pada saat menerima file.

IBR-DTN dapat mengatasi masalah ketidaktersediaannya jaringan dengan cara menyimpan *file* yang akan dikirimkan ke bundle storage dengan lifetime yang telah ditentukan terlebih dahulu dan juga kapasitas besaran file yang dapat disimpan didalam sebuah bundle akan tetapi apabila node penerima mengalami masalah pada *hardware* seperti pengujian yang dilakukan dengan mematikan paksa raspberry pi pada node penerima maka akan mendapatkan file yang rusak dikarenakan proses render yang belum berjalan dengan sempurna, berbeda apabila yang terjadi masalah hanya jaringan yang tidak tersedia, apabila tersambung kembali masih dapat menerima *file* yang belum diterima oleh node penerima dengan catatan file yang blum terkirim dan masih tersimpan di bundle pengirim masih memiliki lifetime untuk menyimpan file yang akan dikirim ke node penerima hingga node penerima terhubung kembali ke jaringan.

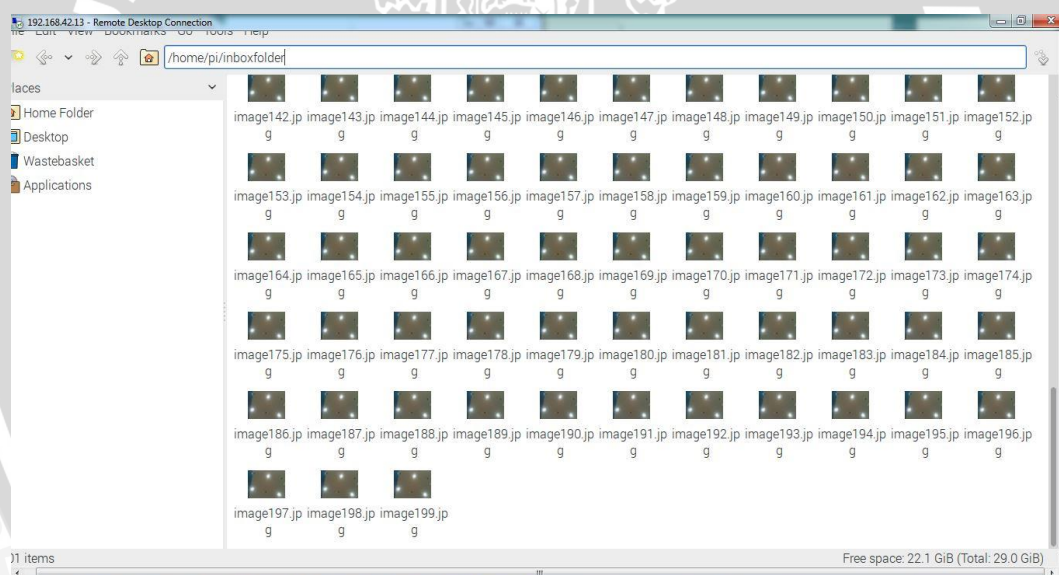
5.2.2 Hasil Pengujian Pengiriman *File* Dengan Jumlah Tertentu

Pada tahap ini dilakukan pengujian dengan beberapa skenario berbeda dengan mengirimkan *file* sebanyak 100, 200, 300 dan 500 *file* dengan ukuran tiap *file* sama. Pada saat pengiriman *file* tidak diberikan waktu tunggu untuk penangkapan gambar oleh kamera modul, sehingga pada saat *file* telah dikirim kamera pada saat

itu juga langsung mengambil gambar kembali dan melakukan pengiriman kembali. Hasil pengujian dapat dilihat pada Gambar 5.5 dan Gambar 5.6.



Gambar 5.5 Pengiriman *file* sebanyak 100 kali



Gambar 5.6 Pengiriman *file* sebanyak 200 kali

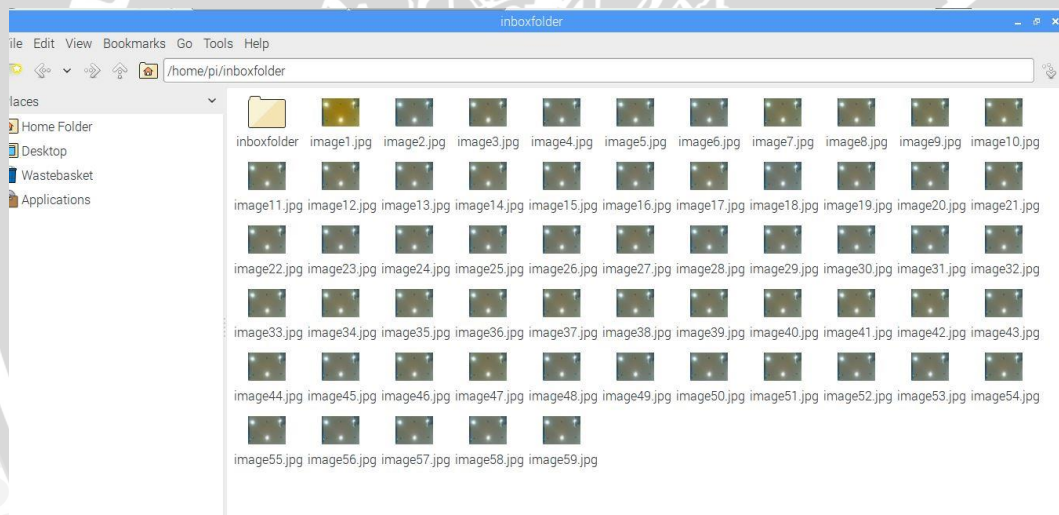
Pada pengujian ini didapatkan perbedaan waktu antara total waktu yang dibutuhkan untuk mengambil gambar dan total waktu yang dibutuhkan untuk menerima gambar dari node pengirim. Selisih waktu ini dikarenakan menggunakan *tools* dari IBR-DTN berupa *dtmoutbox* dan *dtminbox*. Pada *dtmoutbox*, dibutuhkan waktu untuk menemukan file terlebih dahulu lalu dan ada jeda kembali sebelum melakukan pengiriman file, berbeda dengan menggunakan *tools* *dtmsend* yang dimana ketika gambar telah ditangkap oleh kamera modul pada saat itu juga IBR-DTN melakukan pengiriman file dan disimpan terlebih dahulu pada folder bundle yang membuat file yang telah diterima memiliki lifetime penyimpanan di folder bundle. Perbandingan waktu tangkap gambar dan dan terima gambar dapat dilihat pada Tabel 5.4.

Tabel 5.4 Perbandingan Waktu Tangkap Gambar dan Terima

No	Jumlah <i>File</i>	Waktu Penangkapan	Waktu Penerima	Selisih
1	100	53 Detik	01:08	15 detik
2	200	01:43	02:16	33 detik
3	300	02:35	02:52	13 detik
4	500	04:18	04:38	20 detik

5.2.3 Hasil Pengujian Pengiriman *File* selama 30 menit

Pada tahap ini dilakukan pengujian dengan beberapa skenario berbeda dengan mengirimkan *file* selama 30 menit, diberikan waktu tunggu untuk penangkapan gambar selama 30 detik oleh kamera modul, sehingga pada saat *file* telah dikirim dibutuhkan waktu selama 30 detik untuk kamera mengambil gambar kembali dan melakukan pengiriman *file*. Hasil pengujian dapat dilihat pada Gambar 5.7.



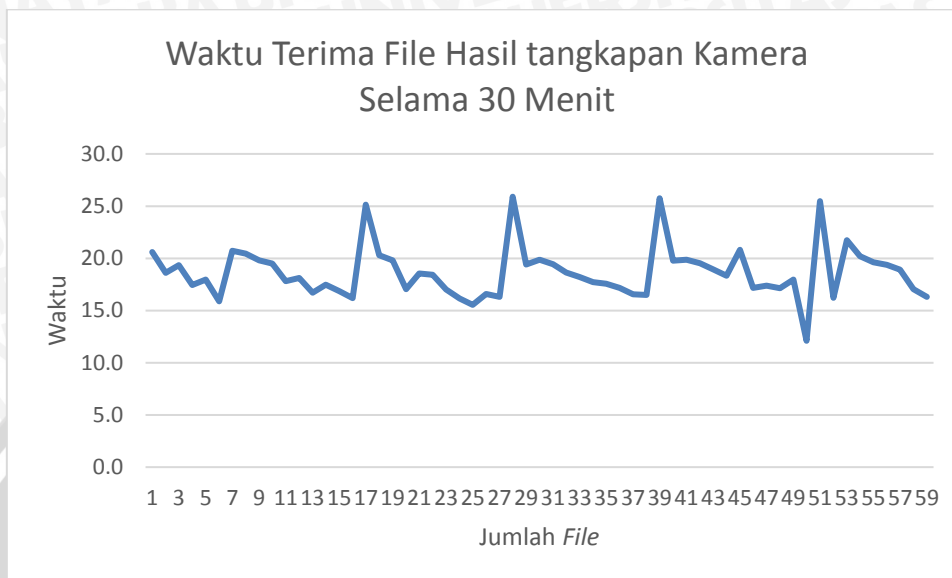
Gambar 5.7 Hasil Pengiriman *File* dengan Lama Waktu Tertentu

Pada Gambar 5.8 pengiriman *file* selama 30 menit, IBR-DTN mampu mengirimkan *file* sebanyak 59 *file* tanpa adanya *file* yang hilang maupun terjadinya duplikasi file. *Tools* IBR-DTN membutuhkan rata-rata waktu pengiriman tiap file gambar hasil tangkap dari kamera sealam 18.7 Detik dengan waktu yang paling tinggi sebesar 25.9 detik dan waktu terendah selama 12.1 detik seperti pada Tabel 5.5.

Tabel 5.5 waktu pengiriman *file*

	Waktu
Rata Rata	18.7 Detik

Max	25.9 Detik
Min	12.1 Detik



Gambar 5.8 Pengiriman File selama 30 Menit



BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil simulasi dan pengujian yang telah dilakukan, maka diambil kesimpulan sebagai berikut:

1. Simulasi infrasturuktur dapat mengatasi masalah ketidak tersediannya jaringan untuk melakukan pertukaran *file*.
2. Terdapat kerusakan file apabila *hardwere* pada node penerima mengalami mati mendadak, berbeda apabila IBR-DTN hanya kehilangan jaringan yang tidak berakibat pada kerusakan *file* yang telah diterima.
3. Untuk pengiriman file tangkap gambar sekaligus tanpa jeda waktu terdapat perbedaan waktu tangkap dambar dengan selisih rata-rata 20.25 detik untuk menerima keseluruhan file tangkap gambar.
4. *Tools* dari ibr DTN membutuhkan waktu untuk mengirimkan file hasil tangkap dengan rata-rata 18.7 detik untuk melakukan pengirim 1 buah file hasil tangkap dari kamera menuju node pengirim.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan infrastruktur ini adalah:

1. Untuk pengembangan lebih lanjut, infrastruktur ini dapat dikembangkan dengan mengkonfigurasi isi dari perangkat lunak agar lebih mudah digunakan.
2. Untuk pengembangan lebih lanjut, dapat ditambahkan beberapa skenario pengujian guna mendapatkan hasil performa dan kinerja IBR-DTN

DAFTAR PUSTAKA

- Doering, dkk. 2008. *IBR-DTN: An Efficient Implementation for Embedded Systems*.
Fall, Kevin. 2003. *A Delay-Tolerant Network Architecture for Challenged Internets*.
Magdalena. 2014. Analisis Penggunaan Protokol Routing Prophet Pada IBR-DTN
untuk Sistem Berabagi Informasi Digital di Daerah Pedalaman. Universitas
Brawijaya.
- Prakista, Evan, Hardyanto. 2013. Perancangan Sistem Pertukaran Informasi di
Pedesaan Berbasis *Delay Tolerant Network*. Univeristas Brawijaya.
- Rahmania, Lidya Amalia. 2013. Penerapan *Delay Tolerant Network* (DTN) untuk
Sistem Konsultasi Kesehatan Jarak Jauh Berbasis Web. Universitas Brawijaya.
- Raspberry Pi Foundation, 2012. Help: What Is A Raspberry Pi. [Online] Tersedia di:
<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/> [Diakses 13 Januari
2017].
- Siswanti, Sri, Desy. 2013. Pengembangan Sistem Aplikasi Pengiriman Data Daerah
Terpencil Berbasis *Delay Tolerant Network*. Universitas Sumatera Selatan.
- Suharsono, Aswin. 2012. Pengertian dan Latar Belakang *Delay Tolerant Network*.
[Online] Tersedia di:
[http://aswinsuharsono.lecture.ub.ac.id/2012/07/pengertian-dan-latar-
belakang-delay-tolerant-network/](http://aswinsuharsono.lecture.ub.ac.id/2012/07/pengertian-dan-latar-belakang-delay-tolerant-network/) [Diakses 12 Januari 2017].
- Wikipedia. 2016. Internet. [Online] Tersedia di:
<https://id.wikipedia.org/wiki/Internet> [Diakses 12 Januari 2017].
- Yuliandoko, dkk. 2015. *Performance of Implementation IBR-DTN and Batman Adv
Routing Protocol in Wireless Mesh Networks*. Institut Politeknik Surabaya.

LAMPIRAN

Lampiran 1 IBRDTN.CONF.

```
#####  
# IBR-DTN daemon                                     #  
#####  
  
#  
# the local eid of the dtn node  
# default is the hostname  
#  
local_uri = dtn://raspy1.dtn  
  
#  
# specifies an additional logfile  
#  
logfile = /var/log/ibrdtn/ibrdtn.log  
  
#  
# Limit the block size of all bundles.  
#  
# The value accepts different multipliers.  
# G = 1,000,000,000 bytes  
# M = 1,000,000 bytes  
# K = 1,000 bytes  
#  
#limit_blocksize = 1.3G  
  
#  
# Limit the block size of foreign bundles.  
# Foreign bundles are not address from or to the  
# local node.  
#  
# The value accepts different multipliers.  
# G = 1,000,000,000 bytes  
# M = 1,000,000 bytes  
# K = 1,000 bytes
```



```
#  
#limit_foreign_blocksize = 500M  
  
#  
# Limit the offset of predated timestamps to a max value.  
# Bundles with an invalid timestamp will be rejected.  
#  
#limit_predated_timestamp = 604800  
  
#  
# Limit the max. lifetime of a bundle.  
# Bundles with a lifetime greater than this value will be  
# rejected.  
#  
#limit_lifetime = 604800  
  
# limit the numbers of bundles in transit (default: 5)  
#limit_bundles_in_transit = 5  
# bind API to a named socket instead of an interface  
#api_socket = /tmp/ibrdtn.sock  
  
# define the interface for the API, choose any to bind on all  
# interfaces  
#api_interface = any  
  
# define the port for the API to bind on  
#api_port = 4550  
  
#  
# enable fragmentation support  
# (default is enabled)  
#  
#fragmentation = no
```

```
#api_port = 4550

#
# enable fragmentation support
# (default is enabled)
#
#fragmentation = no

#
# if fragmentation is enabled, it is possible to split up
# bundles larger than a specific limit into fragments
#
# limit_payload = 500K

#####
# storage configuration          #
#####

#
# define a folder for temporary storage of bundles
# if this is not defined bundles will processed in memory
#
#blob_path = /tmp

#
# define a folder for persistent storage of bundles
# if this is not defined bundles will stored in memory only
#
storage_path = /var/spool/ibrdtn/bundles
# define the port for the API to bind on
#api_port = 4550

#
# enable fragmentation support
# (default is enabled)
#
#fragmentation = no
```

```
#
# if fragmentation is enabled, it is possible to split up
# bundles larger than a specific limit into fragments
#
# limit_payload = 500K

#####
# storage configuration
#####

#
# define a folder for temporary storage of bundles
# if this is not defined bundles will processed in memory
#
#blob_path = /tmp

#
# define a folder for persistent storage of bundles
# if this is not defined bundles will stored in memory only
#
storage_path = /var/spool/ibrdtn/bundles

#
# defines the storage module to use
# default is "simple" using memory or disk (depending on
storage_path)
# storage strategy. if compiled with sqlite support, you could
change
# this to sqlite to use a sql database for bundles.
#
#storage = default

#
# Defines, whether bundleSets are stored persistently in the
storage
# path or the SQLite database. This feature is experimental,
therefore
# the default value is no.
#
```

```
#use_persistent_bundlesets = no
#
# Limit the size of the storage.
# The value accepts different multipliers.
# G = 1,000,000,000 bytes
# M = 1,000,000 bytes
# K = 1,000 bytes
#
limit_storage = 1G
#####
# convergence layer configuration #
#####

#
# discovery over UDP/IP
#
# You can specify an multicast address to listen to for
discovery announcements.
# If no address is specified the multicast equivalent of
broadcast is used.
#
discovery_address = ff02::142 224.0.0.142

# Specify how often discovery beacons are sent. The default is
every 5 seconds.
#discovery_interval = 5

# use short IPND beacons
#discovery_short = 0

# specify the discovery mechanism to use
# 0 = DTN2 compatible discovery
# 1 = IPND version 0
# 2 = IPND version 1 (default)
#discovery_version = 2

# To disable discovery announcements, set this option to zero.
# (default is 1)
"
```

```
#  
# values: default | epidemic | flooding | prophet | none  
#  
# In the "default" the daemon only delivers bundles to neighbors  
# and static  
# available nodes. The alternative module "epidemic" spread all  
# bundles to  
# all available neighbors. Flooding works like epidemic, but  
# do not send the  
# own summary vector to neighbors. Prophet forwards based on  
# the probability  
# to encounter other nodes (see RFC 6693).  
#  
routing = prophet  
#  
# forward bundles to other nodes (yes/no)  
#  
#routing_forwarding = yes  
  
#  
# forward singleton bundles directly if the destination is a  
# neighbor  
#  
#routing_prefer_direct = yes  
  
#  
# Scheduling adds a sorted bundle index to the daemon instance  
# which is used  
# to order the bundles using the priority defined in the  
# SchedulingBlock and  
# several other indicators.  
#  
#scheduling = no  
  
# Interval between two requests on all interfaces in order to  
# determine added or  
# removed addresses. This option is only applicable, if netlink  
# is not supported  
# (in milliseconds, default is 5000)
```

Lampiran 2 Tabel Waktu Terima file pada Pengiriman Selama 30 Menit.

No	Waktu	
1	20.6	25.8
2	18.6	19.8
3	19.4	19.9
4	17.4	19.5
5	18.0	18.9
6	15.9	18.3
7	20.7	20.8
8	20.4	17.2
9	19.8	17.4
10	19.5	17.2
11	17.8	18.0
12	18.1	12.1
13	16.7	25.5
14	17.5	16.2
15	16.9	21.7
16	16.2	20.2
17	25.2	19.6
18	20.3	19.4
19	19.8	18.9
20	17.0	17.0
21	18.6	16.3
22	18.4	
23	17.0	
24	16.2	
25	15.5	
26	16.6	
27	16.3	
28	25.9	
29	19.4	
30	19.9	
31	19.5	
32	18.7	
33	18.2	
34	17.7	
35	17.6	
36	17.2	
37	16.6	
38	16.5	

