

IMPLEMENTASI OFFLINE BROWSING PADA DELAY TOLERANT NETWORK (DTN)

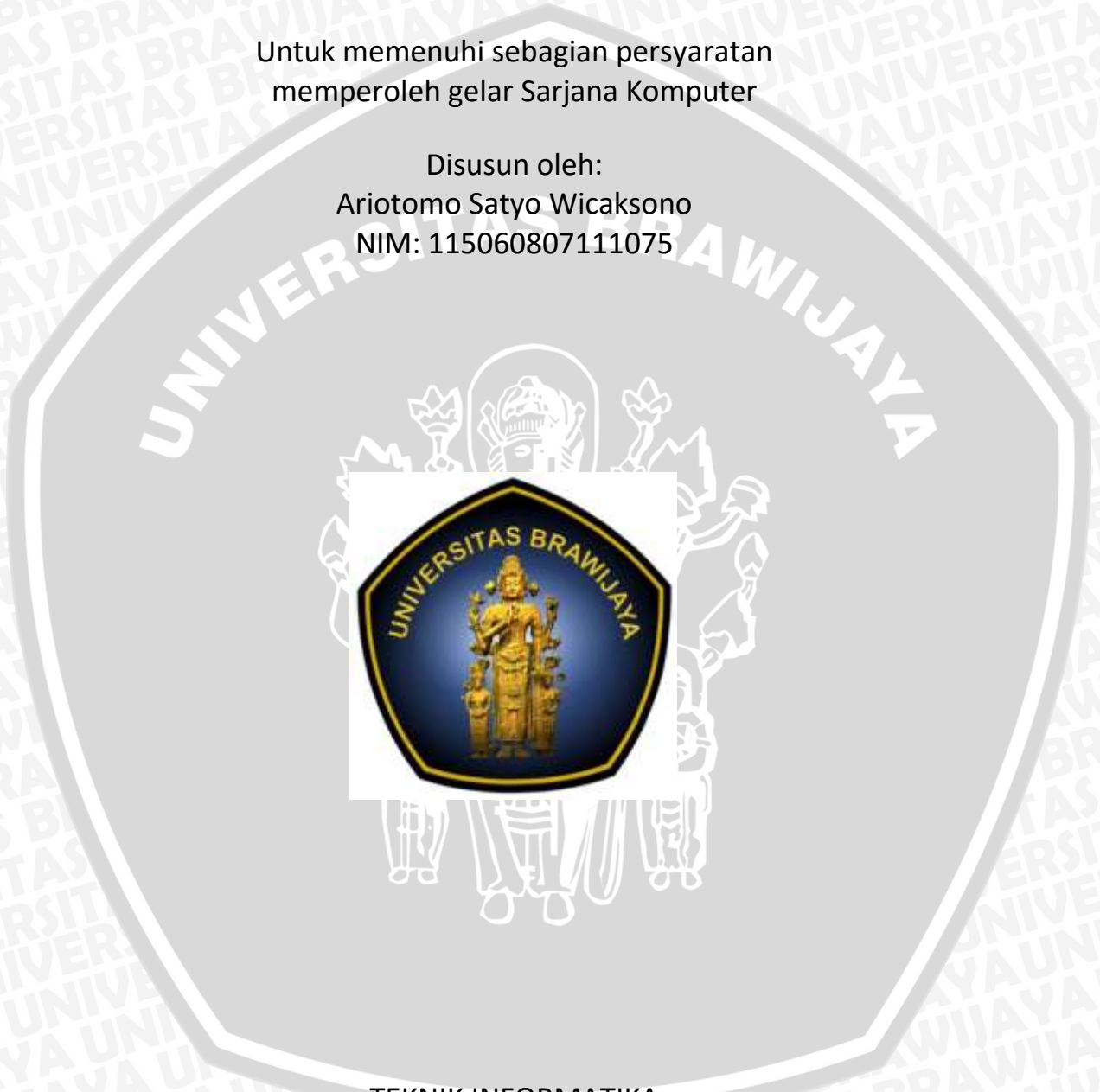
SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ariotomo Satyo Wicaksono

NIM: 115060807111075



TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

IMPLEMENTASI OFFLINE BROWSING PADA DELAY TOLERANT NETWORK (DTN)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ariotomo Satyo Wicaksono
NIM: 115060807111075

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Aswin Suharsono, S.T., M.T.
NIK: 840919 06 1 1 0251

Eko Sakti P., S.Kom., M.Kom.
NIK: 860805 06 1 1 0 252

Mengetahui
Ketua Program Studi NamaProgramStudi

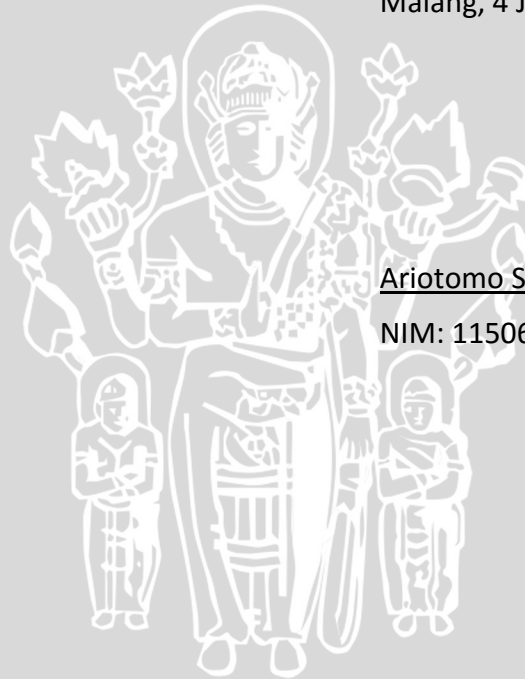
Drs. Marji, M.T.
NIP: 19670801 199203 1 001

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, 4 Januari 2016



Ariotomo Satyo Wicaksono

NIM: 115060807111075

KATA PENGANTAR

Puji syukur kami ucapkan kehadiran Tuhan Yang Maha Esa atas rahmat dan hidayahNya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Implementasi *Offline Browser* pada *Delay Tolerant Network*”.

Penulis menyadari bahwa penyusunan makalah ini tidak akan berjalan lancar tanpa berbagai masukan yang bermanfaat dari dosen pembimbing serta teman-teman penulis. Oleh karena itu penulis mengucapkan terima kasih kepada pihak-pihak tersebut yang telah bersedia untuk memberikan arahan dan saran demi kelancaran penyusunan makalah ini diantaranya:

1. Aswin Suharsono, S.T., M.T., selaku dosen pembimbing I yang telah banyak memberikan ilmu, saran, dan motivasi untuk menyelesaikan penelitian ini.
2. Eko Sakti Pramukantoro S.Kom., M.Kom., selaku dosen pembimbing II yang telah memberikan ilmu, saran dan semangat untuk menyelesaikan penelitian ini.
3. Kedua orang tua penulis, Bambang Suharijadi S.H., M.H., dan Endang Purwani Pontjomurti S.Kom., serta adik penulis Bagas Rayi Prabowo yang telah memberi motivasi, kasih sayang, serta dukungan baik moril maupun materil.
4. Kamilia Latifah A.Md. yang telah memberikan dukungan dan semangat dalam menyelesaikan penelitian ini.
5. Danang Januarko, Sarwo Hadi Wibowo, Nurtria Iman Sari S.Kom, Muhammad Riza Arfiharis, Taufiq Hamidhi, Robi Cahyo Prasetyo, dan seluruh teman-teman TIF-A 2011 yang telah menemani dan memberi saran dan masukan kepada penulis hingga terselesaikannya skripsi ini.
6. Semua teman-teman PTIIK, khususnya Informatika/Illmu Komputer 2011 terima kasih atas segala bantuan dan dukungannya selama ini.
7. Segenap dosen dan karyawan PTIIK Universitas Brawijaya yang telah membantu pelaksanaan skripsi ini.
8. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT Kritik dan saran yang membangun sangat kami harapkan dari para pembaca mengingat makalah yang kami susun ini masih jauh dari kata sempurna.

Akhir kata, kami ucapkan terima kasih dan semoga makalah ini dapat memberikan manfaat bagi para pembaca terutama mahasiswa PTIIK Universitas Brawijaya.

Malang, 4 Januari 2016

Penulis

ariotomosaty@gmail.com

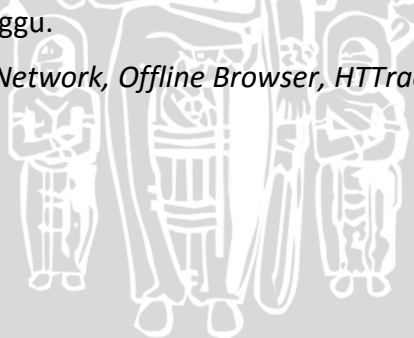


ABSTRAK

Ariotomo Satyo Wicaksono. 2015: Implementasi *Offline Browser* pada *Delay Tolerant Network*. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Aswin Suharsono, S.T., M.T. dan Eko Sakti Pramukantoro, S.Kom., M.Kom.

Internet berbasis TCP/IP yang digunakan secara masal saat ini mungkin merupakan sebuah solusi yang cocok untuk sebagian besar pengguna internet. Namun, terdapat beberapa kasus tertentu yang tidak memungkinkan digunakannya protokol tersebut. Maka salah satu solusinya adalah dikembangkannya arsitektur *Delay Tolerant Network* (DTN). DTN mendukung komunikasi antar node tanpa memperdulikan waktu tunda yang lama, yang dijadikan prioritas adalah pesan sampai walaupun membutuhkan waktu berjam-jam ataupun hitungan hari. Salah satu aspek dari penggunaan internet yang masih lemah dalam DTN adalah *web browsing*, karena tiap melakukan *request* dan mengirim jawaban untuk *request* tersebut membutuhkan waktu yang lama maka pengguna harus menunggu untuk waktu yang panjang untuk membuka tiap halaman, hal ini juga belum memperhitungkan kesulitan melakukan *web browsing* pada DTN karena browser tradisional tidak mendukung protokolnya. Penelitian ini mencoba mengimplementasikan sebuah *offline browser* pada DTN yang memungkinkan pengguna untuk mengirimkan *request* URL yang ingin diunduh oleh sistem dan *tool offline browser* akan mengunduh URL yang diminta beserta *link* yang tercantum pada halaman tersebut sehingga pengguna dapat mengakses semua halaman yang tercantum tanpa harus mengirim *request* untuk setiap halaman dan menunggu.

Kata kunci: *Delay Tolerant Network*, *Offline Browser*, *HTTrack*, *Web Browsing*



ABSTRACT

Ariotomo Satyo Wicaksono. 2015: Implementation of *Offline* Browser on Delay Tolerant Network. Program of Information Technology and Computer Science, Brawijaya University, Malang. Supervisor: Aswin Suharsono, S.T., M.T. and Eko Sakti Pramukantoro, S.Kom., M.Kom.

The widely used TCP/IP based internet may perform poorly in some environment characterized with long delay paths and frequent network partitions. Delay Tolerant Network (DTN) architecture is one of the solution proposed to provide network operation on places with such environment characterization. DTN supports network operation and internet connectivity to a degree without expecting end-to-end connectivity and will relay messages and packets from one node to another regardless of the time it will take to relay them. In using this architecture, user could send non time-sensitive information, but when it comes to conventional web browsing it gets a little tricky because, the user have to send a request for every webpage they want to access and that would take a long time to get an answer from the server, and that is without the convenience of a more traditional web browser that cannot operate on DTN. This research is offering a solution, using an offline browser tool so that user can input the webpage's URL and the offline browser would download the webpage and every page that is linked inside that page, providing the user the webpage they requested along with pages that is linked so the user doesn't have to send a request for each one of them and can browse the site comfortably.

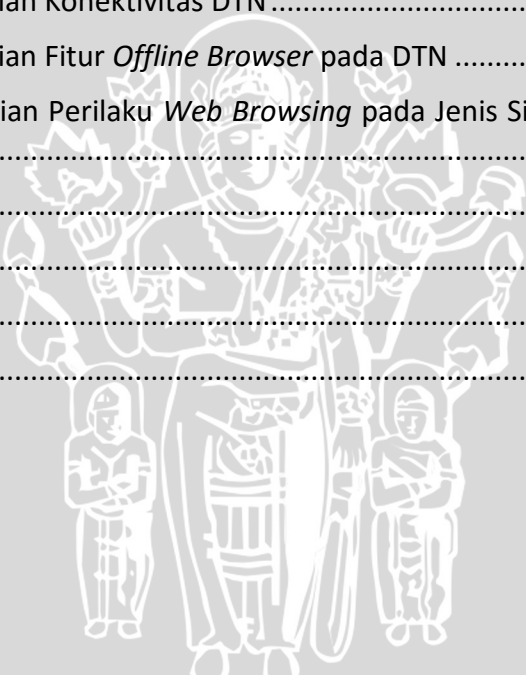
Keywords: Delay Tolerant Network, Offline Browser, HTTrack, Web Browsing



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR SOURCE CODE	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 <i>Delay Tolerant Network (DTN)</i>	4
2.1.1 <i>Bundle Layer</i>	5
2.1.2 <i>Metode Store-and-Forward</i>	5
2.2 DTN2	6
2.2.1 Operasi Pada DTN2	6
2.3 HTTrack	7
BAB 3 METODOLOGI	8
3.1 Jenis Penelitian	8
3.2 Metodologi Penelitian	8
3.2.1 Studi Literatur	10
3.2.2 Analisis Kebutuhan.....	10
3.2.3 Perancangan.....	10
3.2.4 Implementasi Sistem.....	10
3.2.5 Pengujian Sistem dan Analisa Hasil Pengujian.....	10

3.2.6 Pengambilan Kesimpulan dan Saran.....	11
3.3 Perancangan	11
3.3.1 Analisis Kebutuhan.....	11
3.3.2 Penggunaan Perangkat Lunak Pendukung.....	11
3.3.3 Arsitektur Sistem.....	12
BAB 4 Implementasi	14
4.1 Implementasi Simulasi Jaringan.....	14
4.1.1 Konfigurasi Node DTN	14
4.1.2 Implementasi <i>Offline browser</i> HTTrack pada DTN.....	16
BAB 5 PENGUJIAN	21
5.1 Pengujian	21
5.1.1 Pengujian Konektivitas DTN	21
5.1.2 Pengujian Fitur <i>Offline Browser</i> pada DTN	23
5.1.3 Pengujian Perilaku <i>Web Browsing</i> pada Jenis Situs yang Berbeda	25
BAB 6 Penutup	29
6.1 Kesimpulan.....	29
6.2 Saran	29
DAFTAR PUSTAKA.....	30



DAFTAR TABEL

Tabel 4.1 Spesifikasi konfigurasi DTN tiap node	14
Tabel 5.1 Tabel hasil pengujian Perilaku <i>Web Browsing</i> pada Jenis Situs yang Berbeda	28



DAFTAR GAMBAR

Gambar 2.1 <i>Bundle Layer</i>	5
Gambar 2.2 Metode <i>Store-and-Forward</i>	5
Gambar 3.1 Diagram Alir Metodologi Penelitian	9
Gambar 3.2 Arsitektur Sistem	12
Gambar 4.1 Aplikasi DTN2 berhasil dijalankan di node DTN pertama	15
Gambar 4.2 Hasil tes koneksi dengan dtnping	16
Gambar 4.3 Tampilan ketika pengguna selesai memasukkan URL <i>webpage</i> yang <i>direquest</i>	17
Gambar 4.4 Contoh tampilan <i>webpage</i> yang dibuka dari <i>web server</i> node pertama	20
Gambar 5.1 Data berhasil dikirimkan melalui DTN	21
Gambar 5.2 <i>Bundle</i> pada node 1	21
Gambar 5.3 Data berhasil diterima pada node 2	22
Gambar 5.4 Data berhasil dikirimkan melalui DTN	22
Gambar 5.5 <i>Bundle</i> masih tersimpan di node1	22
Gambar 5.6 Situs yang diminta berhasil diunduh dan dapat diakses secara <i>offline</i>	23
Gambar 5.7 <i>Link</i> yang berada pada halaman detik.com dapat diakses secara <i>offline</i>	24
Gambar 5.8 Notifikasi ketersediaan situs	24
Gambar 5.9 Index situs yang sudah tersimpan	25
Gambar 5.10 Situs detik.com dapat dilihat secara <i>offline</i> dengan sempurna	25
Gambar 5.11 Artikel pada detik.com dapat dilihat secara <i>offline</i>	26
Gambar 5.12 Situs ub.ac.id dapat dilihat secara <i>offline</i>	26
Gambar 5.13 <i>Offline browser</i> tidak dapat menduplikasi google.com	27
Gambar 5.14 Google.co.id dapat diakses secara <i>offline</i>	27
Gambar 5.15 <i>Offline browser</i> tidak dapat menduplikasi facebook.com	28

DAFTAR SOURCE CODE

Source Code 4.1 Potongan <i>Source Code Form Request</i> pada node pertama.....	17
Source Code 4.2 <i>Source Code</i> dari fungsi utama pada node DTN terakhir.....	19



BAB 1 PENDAHULUAN

1.1 Latar belakang

Internet berbasis TCP/IP yang digunakan saat ini menyediakan komunikasi *end-to-end* menggunakan rangkaian teknologi *link-layer* yang berbeda-beda. Walaupun tidak disebutkan secara eksplisit, protokol tersebut mengasumsikan beberapa hal mengenai performa koneksi untuk menyediakan jasa tersebut: tersedianya sebuah jalur *end-to-end* antara sumber data dan tujuannya, *round-trip time* antar node pada jaringan tidak berlebihan, dan kemungkinan hilangnya paket saat pengiriman kecil. Namun untuk beberapa situasi dimana karakteristik-karakteristik performa di atas tidak terpenuhi maka dibutuhkan protocol komunikasi yang berbeda, seperti *bundle protocol* pada *delay tolerant network* (DTN).

Salah satu penggunaan utama internet adalah *web browsing*, akan tetapi melakukan browsing menggunakan DTN tidak dapat dilakukan menggunakan *web browser* tradisional karena *web browser* tersebut tidak dapat beroperasi pada DTN. Salah satu penelitian yang sudah dilakukan untuk menjawab permasalahan tersebut adalah **Web Browser Extension for DTN** (Kurnikov, 2011). Penelitian tersebut mencoba membangun sebuah *extension* untuk *web browser* Mozilla Firefox yang bertujuan membuat *browser* tersebut dapat bekerja pada DTN.

Penelitian lain yang juga mencoba menyelesaikan permasalahan ini adalah **Web Access over Delay Tolerant Networks** (Bong, 2012). Penelitian tersebut membangun sebuah aplikasi yang memungkinkan pengguna untuk mengakses internet menggunakan *browser* pilihan pengguna walaupun tidak memiliki akses langsung ke IP networknya. Aplikasi tersebut bekerja dengan menyimpan *cache* dari *request* yang dikirim pengguna lalu dikirim sebagai *bundle* DTN. Aplikasi tersebut juga membangun *thread-thread* untuk mendukung banyak *request* seperti *multi-tab browsing* ataupun *parallel web request*. Namun, kelemahan pada penelitian tersebut adalah melakukan *web browsing* pada DTN tidaklah efisien karena terdapat waktu tunda yang tidak pasti pada saat pengguna mengirimkan *request* dan saat jawaban dari *request* tersebut dikirim kembali.

Offline browser adalah *software* untuk melakukan *web browsing* tanpa menggunakan internet. *Offline browser* bekerja dengan menduplikasi *website* yang diinginkan ke mesin pengguna. *Offline browser* ini juga memberikan fitur tambahan yaitu kemampuan untuk memilih jumlah kedalaman halaman yang ingin diduplikasi serta fitur untuk menduplikasi halaman situs yang di-link pada halaman yang diminta sehingga pengguna dapat membuka halaman-halaman yang terkait dengan yang diminta tanpa harus mengirim *request* untuk masing-masing halaman tersebut. Kemampuan ini cocok dengan karakteristik DTN yang membutuhkan waktu panjang untuk mengirimkan *request* serta mengirimkan balasan dari permintaan tersebut yang menjadikan waktu untuk mengunduh jumlah *webpage* yang banyak tidaklah relevan.

Solusi yang ditawarkan oleh skripsi ini adalah mengimplementasikan sebuah *offline browser* pada DTN sehingga pengguna dapat melakukan *web browsing* secara *offline*. Mengacu pada permasalahan yang telah disampaikan, judul yang diambil dalam skripsi ini adalah "**Implementasi Offline Browsing pada Delay Tolerant Network**". Diharapkan tema skripsi yang diangkat dapat menjadi salah satu solusi agar DTN dapat lebih dirasakan manfaatnya bagi para penggunanya.

1.2 Rumusan masalah

Berdasarkan paparan latar belakang tersebut, maka permasalahan yang akan dibahas dan dijawab pada skripsi ini adalah :

1. Bagaimana merancang sistem yang memungkinkan untuk melakukan *web browsing* tanpa menggunakan internet?
2. Bagaimana perilaku *offline browser* yang diimplementasikan pada DTN untuk *web browsing* jenis situs yang berbeda?

1.3 Tujuan

Mengimplementasikan sebuah *offline browser* pada DTN memungkinkan untuk melakukan *web browsing* pada jaringan dengan konektivitas terbatas ataupun tidak ada koneksi internet sama sekali dengan mengunduh website yang diinginkan pengguna beserta link yang berkaitan sehingga dapat dilihat secara *offline*.

1.4 Manfaat

1. Bagi Peneliti
 - a) Dapat lebih memahami penggunaan DTN sebagai arsitektur jaringan yang cocok digunakan pada daerah dengan konektivitas terbatas.
 - b) Menambah pengetahuan terkait dengan pengimplementasian sebuah *offline browser* pada DTN.
2. Bagi Pengguna
 - a) Dapat melakukan *browsing* website secara *offline* dengan nyaman pada jaringan yang minim koneksi internet.

1.5 Batasan masalah

Untuk menghindari melebarnya permasalahan yang dibahas dalam skripsi ini, maka diterapkan batasan dari masalah yang sudah dirumuskan, sebagai berikut:

1. Pembahasan difokuskan pada cara mengimplementasikan sebuah *offline browser* (HTTrack) pada sebuah jaringan DTN untuk melakukan *web browsing* secara *offline* pada jaringan dengan tanpa koneksi internet.
2. Menggunakan HTTrack sebagai *offline browser* yang diimplementasikan.

1.6 Sistematika pembahasan

Sistematika pembahasan ditunjukkan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab. Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini dijabarkan latar belakang penulis memilih topik skripsi ini, rumusan masalah pengimplementasian sebuah *offline browser* pada DTN, tujuan, manfaat, batasan masalah, sistematika penulisan serta rencana kegiatan.

BAB II LANDASAN KEPUSTAKAAN

Bab ini menjelaskan teori-teori yang dipelajari mengenai DTN, perangkat lunak pendukung DTN yaitu DTN2, kajian mengenai *offline browser* yang digunakan yaitu HTTrack, serta bahan kajian lain yang mendukung

BAB III METODOLOGI PENELITIAN DAN PERANCANGAN

Bab ini membahas tentang metodologi yang digunakan dalam penulisan yang terdiri dari studi literatur, analisa kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, dan analisa hasil pengujian serta pengambilan kesimpulan dan saran. Selain itu, pada bab III juga dijelaskan mengenai rancangan dari pengimplementasian HTTrack pada DTN.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab ini membahas tentang penerapan dari arsitektur sistem secara terperinci dengan langkah-langkah pengerjaan serta menampilkan gambar-gambar dari implementasi yang dilakukan berupa potongan-potongan kode dari hasil implementasi. Implementasi dilakukan berdasarkan sistematika yang dibuat pada bab sebelumnya. Berikutnya pengujian terhadap sistem yang dilakukan menggunakan skenario yang telah dibuat pada bab sebelumnya.

BAB V HASIL PENGUJIAN DAN ANALISIS

Pada bab ini membahas tentang hasil pengujian dan analisis terhadap sistem yang telah dibangun. Hasil pengujian didapatkan dari pengujian yang telah dilakukan sesuai dengan skenario pengujian yang sudah ditentukan. Dari hasil tersebut dapat dilakukan analisis terhadap kinerja sistem, serta mengetahui bagaimana sistem mencapai tujuan.

BAB VI PENUTUP

Memuat kesimpulan dari penelitian yang telah dilakukan serta saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bagian ini akan dibahas dasar dari struktur jaringan yang digunakan serta fungsi-fungsinya. Arsitektur client-server akan dijelaskan, arsitektur ini adalah prinsip utana yang digunakan pada struktur jaringan modern. HTTP pada khususnya dibangun berdasarkan arsitektur tersebut.

Pada bagian pertama bab ini, projek DTN lainnya juga akan dijelaskan.

2.1 Delay Tolerant Network (DTN)

Delay Tolerant Network adalah sebuah arsitektur jaringan komputer yang bertujuan untuk mengatasi masalah dari arsitektur jaringan yang umumnya digunakan saat ini yaitu waktu delay yang tinggi akan menyebabkan transmisi data terganggu. Banyak lokasi di dunia yang tidak dapat dijangkau oleh koneksi tradisional seperti fiber optic, kabel tembaga, gelombang radio atau satelit.

Protokol internet yang ada saat ini belum dapat menjangkau semua karakteristik internet yang dibutuhkan oleh pengguna. Beberapa karakteristik yang belum dapat terpenuhi antara lain (Warthman, 2003) :

- *Continuous, Bidirectional End-to-End Path*: sebuah koneksi dua arah yang selalu tersedia antara sumber dan tujuan untuk mendukung interaksi end-to-end.
- *Short Round-Trips*: Delay yang pendek dalam pengiriman paket data dan jaringan yang konsisten.
- *Symmetric Data Rates*: Kecepatan transmisi data yang konsisten antara sumber dan tujuan.
- *Low Error Rates*: Tingkat kemungkinan error dan korupsi data yang rendah.

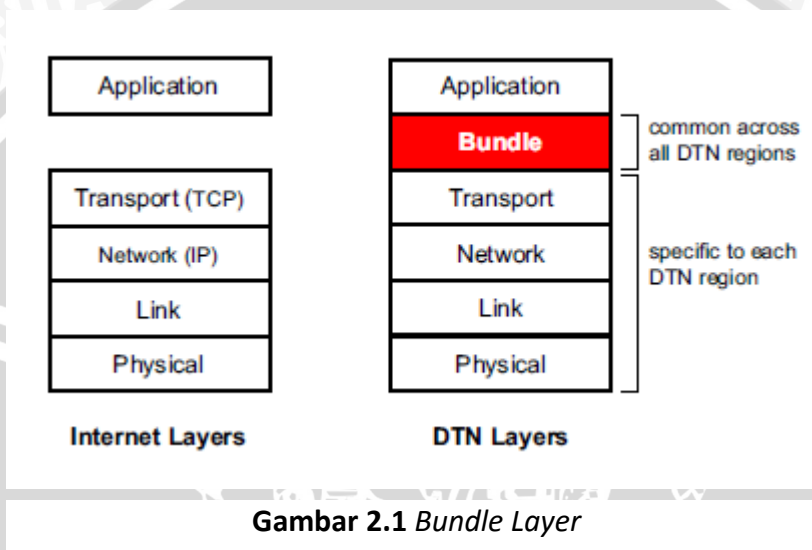
Arsitektur DTN didesain untuk membangun jaringan di daerah yang tidak memiliki konektivitas sesuai dengan karakteristik tersebut dengan prinsip berikut :

- Menggunakan variable long messages (tidak terbatas pada ukuran paket) sebagai abstraksi komunikasi untuk meningkatkan kemampuan jaringan dalam membuat penjadwalan atau pemilihan rute dalam pengiriman data.
- Menggunakan sintaks penamaan yang mendukung banyak penamaan dan pengalamatan untuk meningkatkan operabilitas.
- Menggunakan penyimpanan dimana jaringan mendukung metode store-and-forward pada multiple path dan memiliki rentang waktu yang panjang (misalnya digunakan pada lingkungan dimana terdapat banyak ataupun tidak terdapat jalur end-to-end), serta tidak memperdulikan end-to-end reliability.

- Menyediakan mekanisme keamanan untuk melindungi infrastruktur dari penggunaan tidak sah dengan cara menghilangkan traffic sesegera mungkin.

2.1.1 Bundle Layer

Arsitektur DTN mengimplementasikan *store-and-forward message switching* dengan menambah layer protokol baru yang disebut *bundle layer* diatas *transport layer* dan berada di bawah *application layer*. *Bundle* juga dapat disebut *message*. *Bundle layer* menyimpan dan memforward *bundles* dari satu node ke node lainnya.



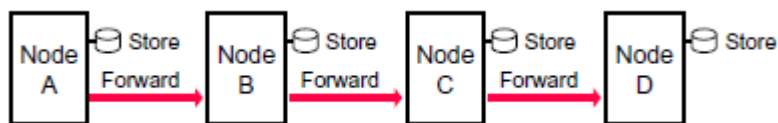
Gambar 2.1 Bundle Layer

Sumber : Warthman, 2003

Bundles terdiri dari tiga hal: (1) user data dari aplikasi sumber, (2) informasi control, didapat dari aplikasi sumber untuk aplikasi tujuan, mendeskripsikan bagaimana cara memproses, menyimpan, membuang, dan handle user data, dan (3) *bundle header*, dimasukkan ke dalam *bundle layer*.

2.1.2 Metode Store-and-Forward

DTN mengatasi permasalahan tidak adanya hubungan end-to-end, delay yang tidak menentu, *data rate* yang tidak simetris, dan kemungkinan error yang tinggi dengan menggunakan *store-and-forward message switching*. Paket data yang dikirim dipindahkan dari satu tempat penyimpanan pada sebuah node ke tempat penyimpanan lainnya pada node yang berbeda, pada jalur yang akan berakhir pada tujuan paket data tersebut dikirim (Warthman, 2003).



Gambar 2.2 Metode Store-and-Forward

Sumber : Warthman, 2003

2.2 DTN2

DTN2 merupakan sebuah perangkat lunak pendukung DTN yang didesain untuk membuktikan bahwa DTN bekerja sesuai dengan yang diharapkan. DTN2 mengimplementasikan The DTN Bundle Protocol yang sesuai dengan spesifikasi pada RFC 5050. DTN2 dapat mengirimkan bundle yang ukurannya lebih besar daripada paket data pada umumnya yang dikirimkan melalui jaringan IP.

DTN2 juga mendukung Bundle Security Protocol untuk menyediakan autentikasi pada saat pengiriman bundle. Bundle bisa dikirimkan melalui (IP) transport layer maupun layer lainnya termasuk Ethernet dan Bluetooth. DTN2 menyediakan mekanisme routing untuk mengirimkan data dari sumber ke tujuan baik itu secara langsung (dari sumber langsung ke tujuan, termasuk skema routing statis) maupun routing yang perlu melewati beberapa node (router) sebelum sampai ke tujuannya (dtnrg.org, 2014).

2.2.1 Operasi Pada DTN2

Berikut beberapa operasi yang dapat dijalankan pada DTN2:

1. **dtnd**, operasi ini berfungsi untuk mengaktifkan dtn daemon. Command yang dilakukan pada saat pertama kali akan menggunakan DTN2 biasanya adalah `sudo dtnd -c dtn.conf`.
2. **dtncping**, operasi ini berfungsi untuk mengirimkan bundle menuju node DTN lain atau mungkin ke localhost, dalam hal ini kita hanya menggunakan daemon lokal.
3. **dtncp**, operasi ini digunakan untuk meng-copy bundle menuju node DTN yang lain. Operasi ini dapat digunakan untuk melakukan pengiriman file. Formatnya adalah `dtncp -expiration <expiration> <file path> <destination>`. Expiration menunjukkan masa aktif bundle tersebut, defaultnya adalah 30 detik. File path menunjukkan lokasi file yang akan dikirim. Destination menunjukkan alamat node tujuan.
4. **dtncpd**, operasi ini digunakan untuk menerima bundle yang dikirimkan, utamanya bundle yang dikirimkan dengan menggunakan dtncp. Formatnya adalah `dtncp <path>`. Path menunjukkan lokasi file yang diterima akan disimpan.
5. **dtncsend**, operasi ini juga digunakan untuk melakukan pengiriman bundle menuju node DTN lain. bundle yang dikirim dapat berasal dari pesan text atau file. Formatnya adalah `dtncsend [opts] -s <source_eid> -d <dest_eid> -t <type> -p <payload>`. `-s source_eid` menunjukkan alamat DTN pengirim. `-d destination_eid` menunjukkan alamat DTN penerima. `-t type` menunjukkan jenis pesan yang dikirimkan, bisa berupa f (file), t (tmpfile), m (pesan), dan d (tanggal)

6. **dtncrcv**, operasi ini digunakan untuk menerima pesan hasil pengiriman dtncsend. Formatnya adalah dtncrcv <endpoint>. Endpoint menunjukkan titik terakhir lokasi pesan diterima.

2.3 HTTrack

HTTrack adalah sebuah software gratis yang berfungsi sebagai *offline browser* yang mudah digunakan. Software ini bekerja dengan mendownloadkan sebuah website dari internet ke direktori local, lalu membangun direktorinya secara rekursif, mengambil halaman HTML, gambar, dan file lainnya dari server ke computer. HTTrack menyusun halaman yang didownloadnya sehingga pengguna dapat menjelajah website tersebut dari link ke link, seperti biasa.

HTTrack menggunakan *web crawler* untuk *memirror* atau mengunduh sebuah website. Keterbatasannya adalah jika protokol pengecualian robot tidak dinonaktifkan maka ada bagian website yang tidak dapat diunduh. Secara *default* HTTrack tidak dapat mengikuti link yang rumit yang dihasilkan oleh fungsi yang rumit. Selain dari itu, software ini dapat dengan mudah mengikuti dan mengunduh link yang dihasilkan oleh *applet* java script sederhana atau flash.

Keuntungan dari menggunakan HTTrack adalah sebagai berikut (Madumathi, 2012):

1. Dapat diakses kapan saja dan dimana saja meskipun tidak ada koneksi internet.
2. Tidak mempedulikan koneksi internet yang tidak stabil dan lambat.
3. Dapat melihat arsipkan dari sebuah website meskipun website tersebut dimatikan.
4. Fleksibel dan mudah digunakan.

BAB 3 METODOLOGI

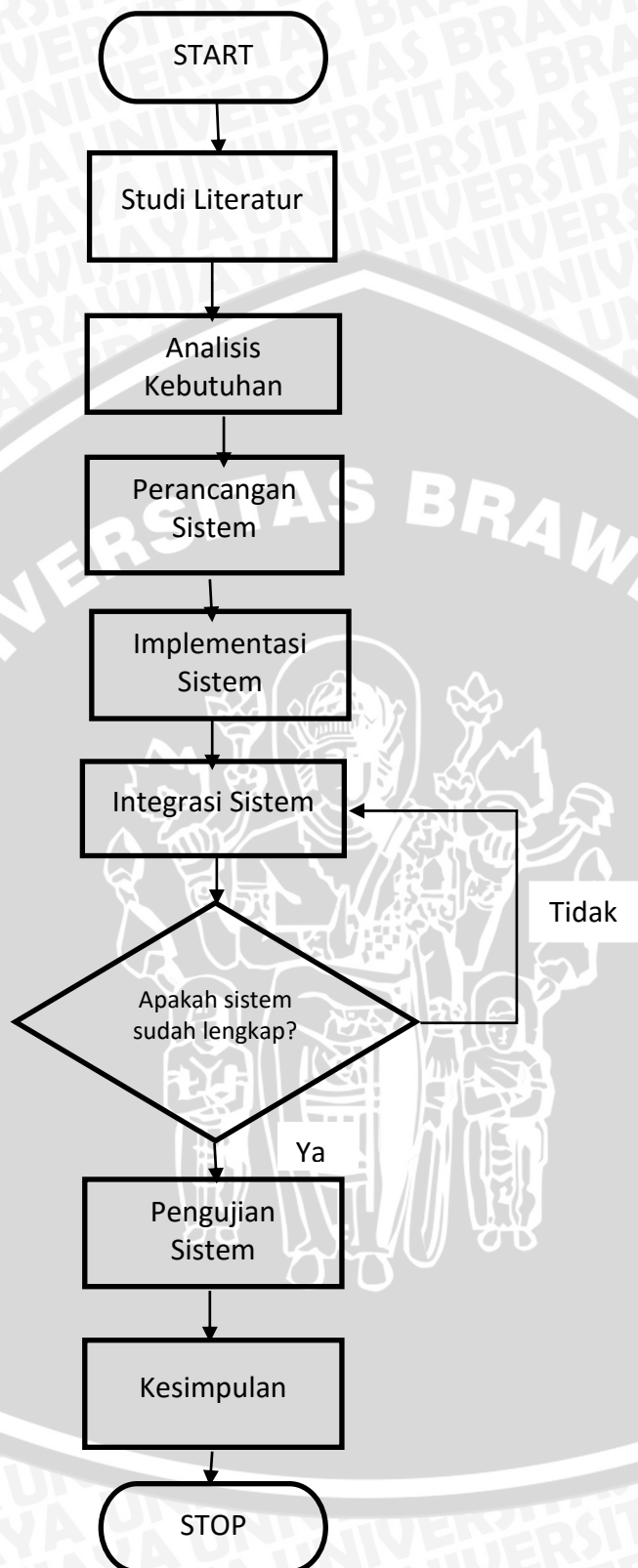
3.1 Jenis Penelitian

Penelitian yang dilakukan merupakan penelitian implementatif pembangunan. Penelitian jenis ini merupakan sebuah kegiatan penelitian dalam rangka membuat sebuah produk perangkat lunak secara sistematis. Penelitian dimulai dari analisis kebutuhan, perancangan perangkat lunak berdasarkan analisis kebutuhan, konstruksi atau pembuatan perangkat lunak, serta pengujian hasil jadi produk perangkat lunak.

3.2 Metodologi Penelitian

Metodologi penelitian yang akan dilakukan pada penelitian ini secara umum ditunjukkan pada diagram alir pada Gambar 3.1 di bawah ini.





Gambar 3.1 Diagram Alir Metodologi Penelitian



3.2.1 Studi Literatur

Studi literatur dilakukan dengan tujuan mempelajari konsep-konsep yang dibutuhkan dalam membangun sistem ini. Literatur yang dipelajari di antaranya adalah jurnal ilmiah, literatur yang dipelajari berisikan pemahaman konsep DTN, perangkat lunak yang digunakan pada DTN yaitu DTN2, serta HTTrack yang digunakan sebagai *offline browser* yang diimplementasikan. Pada tahap ini juga masalah yang dibahas didefinisikan agar pokok bahasan yang diangkat lebih mudah dipahami.

3.2.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menentukan apa saja yang perlu dibangun pada sistem yang dikerjakan. Pada tahap ini dijelaskan mengenai batasan dari sistem serta tujuan dari sistem tersebut. Kebutuhan fungsional yang nantinya akan disediakan oleh sistem ini adalah :

1. Memiliki local web server dimana pengguna dapat memasukkan *request* website yang ingin diunduh untuk dibuka secara *offline*.
2. Menggunakan HTTrack sebagai alat untuk melakukan *download* website yang diinginkan lalu menyusunnya agar mudah dibuka secara *offline*
3. Sesuai dengan batasan *service* yang ditetapkan di batasan masalah, maka fungsi yang ada pada sistem adalah sebagai berikut :

3.2.3 Perancangan

Pada tahap ini dilakukan perancangan sistem, dijelaskan bagaimana proses kerja dari sistem yang diinginkan. Bagaimana sistem mengirimkan request dari node pertama ke node yang terhubung dengan internet dan bagaimana node terakhir mengeksekusi request dari node pertama lalu mengunduh webpage yang diinginkan dan mengirimnya kepada node pertama. Selain itu bagaimana cara menampilkan webpage yang sudah diunduh agar mudah diakses oleh pengguna.

3.2.4 Implementasi Sistem

Sistem dikonstruksi sesuai dengan rancangan yang sudah dibuat pada tahap sebelumnya. Implementasi hasil rancangan sistem diharapkan memenuhi kebutuhan yang dianalisa sebelumnya. Implementasi dilakukan dengan menggunakan bahasa pemrograman PHP, HTTrack sebagai *offline browser*, serta menggunakan perangkat lunak DTN2 sebagai simulator DTN yang sesungguhnya.

3.2.5 Pengujian Sistem dan Analisa Hasil Pengujian

Pengujian dilakukan untuk mengetahui keberhasilan sistem dan memastikan apakah sudah sesuai dengan spesifikasi rancangannya. Pengujian juga dilakukan untuk menguji kemudahan penggunaan untuk user, apakah semua komponen

dalam sistem sudah dapat berjalan secara otomatis atau masih perlu campur tangan manusia. Pengujian dilakukan dengan menjalankan fungsi-fungsi sistem dan membandingkannya dengan kebutuhan fungsionalnya.

3.2.6 Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan hasil analisa sistem. Kesimpulan adalah rangkuman proses penelitian dan jawaban dari rumusan masalah yang sudah ditanyakan sebelumnya. Saran dituliskan sebagai panduan pengembangan sistem ke depannya agar lebih baik lagi.

3.3 Perancangan

3.3.1 Analisis Kebutuhan

Offline browser yang akan diimplementasikan pada DTN adalah HTTrack, sebuah perangkat lunak gratis yang berfungsi untuk mengunduh atau *mirroring* sebuah website agar dapat dibuka secara *offline*. Sesuai dengan penjelasan pada batasan masalah, sistem ini menerima *request* dari pengguna pada web server lokal yang ada pada node pertama lalu mengirimkannya ke node yang memiliki koneksi internet lalu webpage yang sudah didownloadkan dikirim kepada node pertama sehingga dapat dilihat oleh pengguna walaupun tidak ada koneksi internet.

3.3.1.1 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan hal-hal yang harus dapat dilakukan oleh sistem. Berdasarkan uraian servis yang perlu dimiliki oleh sistem pada analisis kebutuhan, berikut kebutuhan fungsional dari *offline browser* :

1. Pengguna dapat melakukan *request website* yang ingin diunduh oleh *offline browser* dan disimpan pada *web server* node1 agar dapat diakses tanpa menggunakan internet.
2. Pengguna dapat membuka halaman yang di-*link* dalam *website* yang diminta tanpa harus melakukan *request* ulang.
3. Terdapat fitur pengecekan apabila situs yang diminta sudah pernah diminta.
4. Pengguna dapat melihat indeks dari situs yang sudah diunduh.

3.3.2 Penggunaan Perangkat Lunak Pendukung

Berikut perangkat lunak pendukung *offline browser* yang diimplementasikan pada DTN :

1. **Sistem Operasi**

Sistem operasi yang digunakan adalah Ubuntu 12.04. Sistem operasi ini diinstal pada node-node DTN. Sistem operasi ini dipilih karena mendukung aplikasi DTN2.

2. **Web Server**

Web server yang digunakan adalah Apache *web server* yang diinstal pada node pertama.

3. Basis Data

Basis data yang digunakan adalah MySQL yang dipasang bersamaan dengan Apache *web server* untuk menyimpan daftar request yang dilakukan oleh pengguna.

4. Oasys

Perangkat lunak yang digunakan untuk menjalankan aplikasi DTN2.

5. DTN2

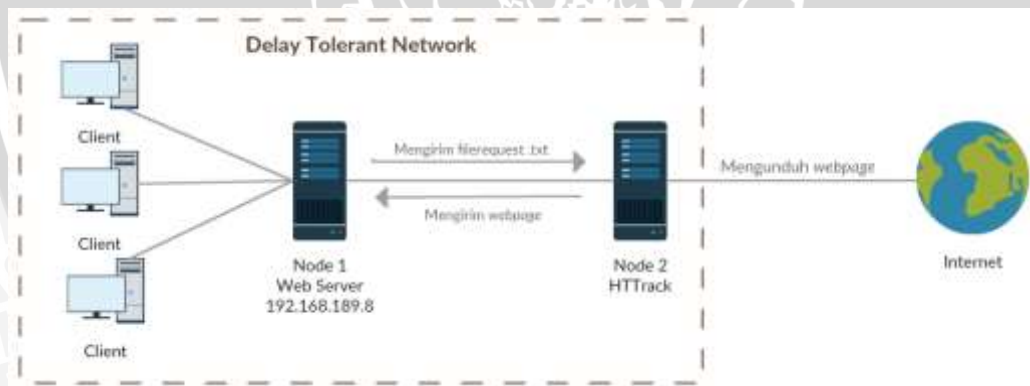
Perangkat lunak utama yang digunakan sebagai bagian dari DTN yang digunakan untuk mengirimkan file dari satu node ke node yang lain.

6. Offline browser

Offline browser yang digunakan pada sistem ini adalah HTTrack untuk mengunduh *webpage* yang diminta oleh pengguna.

3.3.3 Arsitektur Sistem

Rancangan arsitektur sistem ini didasarkan pada tiga fungsi utama, yaitu : fungsi mengirim *request* dari node pertama ke node terakhir menggunakan DTN, fungsi *offline browser* mengunduh *webpage* yang di-*request*, dan fungsi mengirimkan hasil unduhan ke node pertama.



Gambar 3.2 Arsitektur Sistem

Gambar 3.2 menggambarkan arsitektur sistem yang akan dibuat. Dimulai dari *client* yang membuat *request* dari *web server* pada node 1 lalu dikirimkan dengan bentuk file .txt menggunakan DTN ke node 2 yang terhubung dengan internet dan memiliki *offline browser* yang dapat mengunduh *webpage* yang diminta kemudian mengirimkannya kembali ke node 1 sehingga dapat dibuka oleh pengguna.

Terdapat beberapa hal dalam arsitektur ini yang dibutuhkan agar sistem dapat berjalan dengan baik :

1. Form *request* ada di *web server* yang terletak pada node1 dan dapat diakses pengguna pada 192.168.189.8/index.php.
2. *Request* dikirimkan dalam bentuk file .txt dan dikirimkan menggunakan *dtncp*.
3. *Script* untuk menjalankan *offline browser* HTTrack dibuat dengan PHP dan dijalankan setiap jamnya dengan menggunakan *crontab*, sebuah *tool* pada sistem operasi ubuntu.
4. Hasil unduhan dari HTTrack dibungkus dalam arsip zip dan langsung dikirimkan ke node 1 menggunakan *dtncp*.
5. Pada node 1 terdapat *shell script* yang berfungsi untuk memeriksa folder untuk file zip yang dikirimkan oleh node 2 setiap jamnya menggunakan *crontab* dan secara otomatis melakukan *unzip* jika ditemukan.



BAB 4 IMPLEMENTASI

Pada bab ini akan dijelaskan proses pengimplementasian sistem yang dibuat. Penerapan ini sesuai dengan perancangan yang telah dibuat pada bab metodologi penelitian dan perancangan.

4.1 Implementasi Simulasi Jaringan

Yang pertama harus dilakukan adalah instalasi sistem operasi Ubuntu 12.04 pada *virtual machine* untuk node DTN pertama dan node DTN terakhir. Berikutnya adalah melakukan instalasi Oasys sebagai platform untuk aplikasi DTN2. Langkah selanjutnya yaitu menginstalasi aplikasi DTN2 agar DTN dapat berjalan. Kemudian melakukan instalasi Apache *web server*, PHP dan MySQL pada node pertama lalu menginstal HTTrack pada node DTN yang terhubung dengan internet. Simulasi dilakukan pada laptop penulis, yang diinstallkan dua mesin virtual untuk node pertama dan node yang terhubung dengan internet. Web server dijalankan pada mesin virtual node pertama.

4.1.1 Konfigurasi Node DTN

Agar DTN dapat berfungsi perlu dikonfigurasi *routing* dari masing-masing node DTN. Konfigurasi dilakukan pada *dtn.conf*. Konfigurasi yang dilakukan adalah konfigurasi untuk nama node yang dikonfigurasi, *routing table* yang berisi nama node lain beserta alamat IP dari node tersebut, serta jenis protokol yang digunakan. Berikut konfigurasi DTN pada masing-masing node :

Tabel 4.1 Spesifikasi konfigurasi DTN tiap node

	Node pertama	Node terakhir
Nama local_eid	dtn://node1.dtn	Dtn://node2.dtn
Interface	tcp	Tcp
Alamat IP	192.168.198.8	192.168.198.9

Konfigurasi Node DTN Terakhir

```
#konfigurasi untuk memberi nama node
route local_eid "dtn://node2.dtn"

#konfigurasi untuk menambahkan interface
interface add tcp0 tcp

#konfigurasi untuk data alamat IP node berikutnya
link add link1 192.168.198.9 ONDEMAND tcp

#konfigurasi untuk routing table
route add dtn://node1.dtn/* link1
```

Konfigurasi Node DTN Pertama

```
#konfigurasi untuk memberi nama node
route local_eid "dtn://node1.dtn"

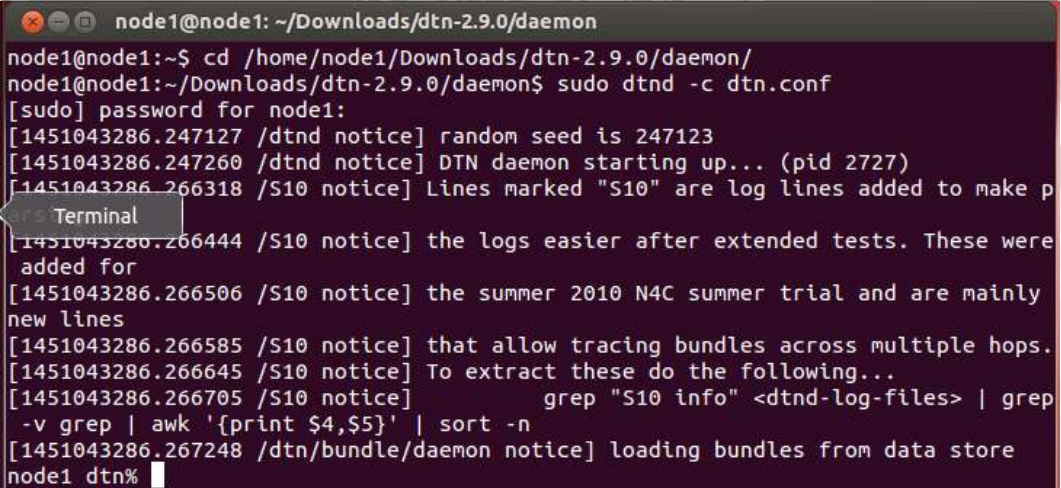
#konfigurasi untuk menambahkan interface
interface add tcp0 tcp

#konfigurasi untuk data alamat IP node berikutnya
link add link1 192.168.198.8 ONDEMAND tcp

#konfigurasi untuk routing table
route add dtn://node2.dtn/* link1
```

Jika semua node berhasil dikonfigurasi maka DTN2 sudah dapat dijalankan. Perintah dijalankan di folder *daemon* pada folder instalasi DTN2. Perintahnya adalah sebagai berikut :

```
# sudo dtnd -c dtn.conf -t
```



```
node1@node1: ~/Downloads/dtn-2.9.0/daemon
node1@node1:~$ cd /home/node1/Downloads/dtn-2.9.0/daemon/
node1@node1:~/Downloads/dtn-2.9.0/daemon$ sudo dtnd -c dtn.conf
[sudo] password for node1:
[1451043286.247127 /dtnd notice] random seed is 247123
[1451043286.247260 /dtnd notice] DTN daemon starting up... (pid 2727)
[1451043286.266318 /S10 notice] Lines marked "S10" are log lines added to make p
Terminal
[1451043286.266444 /S10 notice] the logs easier after extended tests. These were
added for
[1451043286.266506 /S10 notice] the summer 2010 N4C summer trial and are mainly
new lines
[1451043286.266585 /S10 notice] that allow tracing bundles across multiple hops.
[1451043286.266645 /S10 notice] To extract these do the following...
[1451043286.266705 /S10 notice] grep "S10 info" <dtnd-log-files> | grep
-v grep | awk '{print $4,$5}' | sort -n
[1451043286.267248 /dtn/bundle/daemon notice] loading bundles from data store
node1 dtn%
```

Gambar 4.1 Aplikasi DTN2 berhasil dijalankan di node DTN pertama

Setelah aplikasi DTN2 pada semua node dijalankan untuk menguji koneksinya digunakan peritan *dtnping* di folder *apps/dtnping* pada folder instalasi DTN2.

Misalnya kita lakukan tes koneksi dari node DTN pertama ke node DTN terakhir. Perintahnya sebagai berikut :

```
$ dtnping -c 6 dtn://node2.dtn/ping
```



```
node1@node1: ~/Downloads/dtn-2.9.0/apps/dtnping
node1@node1:~/Downloads/dtn-2.9.0/apps/dtnping$ dtnping -c 6 dtn://node2.dtn/ping
source_eid [dtn://node1.dtn/ping.3304]
dtn_register succeeded, regid 10
PING [dtn://node2.dtn/ping] (expiration 30)...
20 bytes from [dtn://node2.dtn/ping]: 'dtnping!' seqno=0, time=71 ms
20 bytes from [dtn://node2.dtn/ping]: 'dtnping!' seqno=1, time=35 ms
20 bytes from [dtn://node2.dtn/ping]: 'dtnping!' seqno=2, time=31 ms
20 bytes from [dtn://node2.dtn/ping]: 'dtnping!' seqno=3, time=33 ms
20 bytes from [dtn://node2.dtn/ping]: 'dtnping!' seqno=4, time=29 ms
20 bytes from [dtn://node2.dtn/ping]: 'dtnping!' seqno=5, time=29 ms
node1@node1:~/Downloads/dtn-2.9.0/apps/dtnping$
```

Gambar 4.2 Hasil tes koneksi dengan dtnping

4.1.2 Implementasi *Offline browser* HTTrack pada DTN

Offline browser yang digunakan pada sistem ini adalah HTTrack yang diinstal pada node terakhir yang terhubung dengan internet. *Package* HTTrack diinstal menggunakan *apt-get* seperti menginstalasi *package* seperti biasa.

Pada sistem ini HTTrack dijalankan dengan script PHP sederhana yang dijalankan secara otomatis setiap jamnya untuk mendeteksi ada atau tidaknya file .txt berisi URL dari *webpage* yang diminta oleh pengguna pada sebuah folder yang sudah ditentukan (/home/node2/node1.dtn).

Setelah *webpage* yang diinginkan oleh pengguna selesai diunduh, kemudian folder dari *webpage* tersebut secara otomatis di-*zip* lalu dikirimkan ke web server pada node pertama dengan *dtncp* yang kemudian dilakukan *unzip* dan diletakkan pada folder /var/www sehingga dapat diakses oleh user pada web server dengan menuliskan URL dari *webpage* yang diinginkan setelah alamat IP dari web server (contoh : <http://192.168.189.8/www.pcgamer.com/>) walaupun tidak ada koneksi internet.

```

if (mysqli_query($conn, $sql)){
    $basedir = '/var/www';
    $time = date("d-m-y-h-i-s");
    $basename = $time.'-request.txt';
    $fp = fopen($basedir.'/'.$basename, 'w');
    fwrite($fp, $data);

    $dtncppath = "/home/node1/Downloads/dtn-
2.9.0/apps/dtncp/dtncp ";
    $tujuan = "dtn://node2.dtn";
    $dtncommand1 = $dtncppath . "/var/www/" . $basename
. " " . $tujuan ;

    $line1 = system($dtncommand1, $returnvalue1);
    fclose($fp);
    unlink($basedir.'/'.$basename);
}
else{
    echo "Insert Gagal!";
}
}
else
{
    print "Masukkan URL web yang ingin anda unduh";
}
?>
<html>
<body>
<form action="" method="post">
<p>Website Address: <input type="text" name="address" />
<input type="submit" value="Download" /></p>
</form>
</body>
</html>

```

Source Code 4.1 Potongan Source Code Form Request pada node pertama

Gambar 4.3 adalah contoh tampilan halaman form untuk melakukan *request webpage* yang ingin diunduh kemudian request tersebut dimasukkan ke dalam file .txt lalu dikirimkan ke node yang terhubung dengan internet menggunakan dtncp. Program 4.1 menunjukkan potongan *source code* fungsi tersebut.

Halaman *request* tersebut dapat diakses dari web server yang ada pada node pertama dengan alamat 192.168.198.8/index.php, web server tersebut dapat diakses oleh semua *client* yang terhubung dengan node pertama.



Gambar 4.3 Tampilan ketika pengguna selesai memasukkan URL *webpage* yang *direquest*

```
<?php
function zip($source, $destination)
{
    if (!extension_loaded('zip') || !file_exists($source))
    {
        return false;
    }

    $zip = new ZipArchive();
    if (!$zip->open($destination, ZIPARCHIVE::CREATE) {
        return false;
    }

    $source = str_replace('\\', '/', realpath($source));

    if (is_dir($source) === true)
    {
        $files = new RecursiveIteratorIterator(new
RecursiveDirectoryIterator($source),
RecursiveIteratorIterator::SELF_FIRST);

        foreach ($files as $file)
        {
            $file = str_replace('\\', '/', $file);

            // Ignore "." and ".." folders
            if( in_array(substr($file, strrpos($file,
'/')+1), array('.', '..')) )
                continue;

            $file = realpath($file);

            if (is_dir($file) === true)
            {
                $zip->addEmptyDir(str_replace($source .
('/', '', $file . '/'));
            }
            else if (is_file($file) === true)
            {
                $zip->addFromString(str_replace($source .
('/', '', $file), file_get_contents($file));
            }
        }
    }
    else if (is_file($source) === true)
    {
        $zip->addFromString(basename($source),
file_get_contents($source));
    }

    return $zip->close();
}

foreach(glob('/home/node2/nodel.dtn/*-request.txt') as
$fileRequest)
{
    $address = file_get_contents($fileRequest);
}
```



```

echo $fileRequest;
echo "<br/>";
$address = preg_replace('/\s+/', '', $address);
echo $address;
echo "<br/>";

$url = escapeshellarg($address);
$host = escapeshellarg(parse_url($address, PHP_URL_HOST));
$url = str_replace("'", "", $url);
$host = str_replace("'", "", $host);
echo $host;
echo "<br/>";

unlink($fileRequest);

    if (empty($address)) {
        echo "gagal";
    }

    else {
        shell_exec("httrack $url -O /var/www/mirror/$host -s0 -r3 -
n");
        zip("/var/www/mirror/$host/",
"/var/www/mirror/$host.zip");

        $dtncppath = "/home/node2/Downloads/dtn-
2.9.0/apps/dtncp/dtncp ";
        $tujuan = "dtn://node1.dtn";
        $dtnccommand1 = "$dtncppath /var/www/mirror/$host.zip
$tujuan";

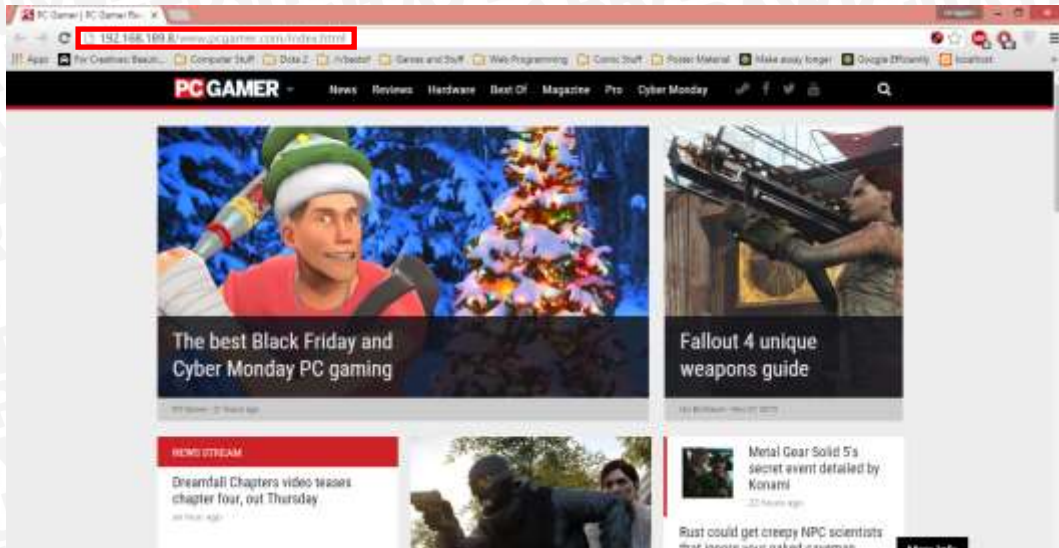
        $line1 = system($dtnccommand1, $returnvalues1);
    }
}
?>

```

Source Code 4.2 Source Code dari fungsi utama pada node DTN terakhir

Script program 4.2 pada node DTN terakhir dijalankan secara otomatis tiap jamnya dengan menggunakan `crontab` untuk mengecek folder `/home/node2/node1.dtn/` untuk mengecek file dengan akhiran `*-request.txt` yang berisi URL yang diminta diunduh oleh pengguna dan kemudian akan otomatis dihapus. Jika ditemukan file sesuai dengan kriteria tersebut maka HTTrack dijalankan untuk mengunduh *webpage* yang diminta lalu hasil file yang diunduh dimasukkan ke dalam *zip* yang dinamakan sesuai dengan URL hostnya dan dikirim kepada node pertama dengan `dtncp`.

Pada node pertama juga dijalankan `crontab` untuk meng-*unzip* *webpage* yang dikirimkan oleh node terakhir kemudian dipindahkan ke folder `/var/www/` agar dapat diakses oleh pengguna dari web server 192.168.198.8.



Gambar 4.4 Contoh tampilan *webpage* yang dibuka dari *web server* node pertama

Gambar 4.4 menunjukkan contoh *webpage* yang sudah diunduh dibuka dari *web server* node pertama oleh pengguna dan dapat diakses oleh pengguna meskipun tidak ada koneksi internet dan pengguna juga dapat membuka *link-link* yang terdapat pada halaman tersebut secara *offline*.



BAB 5 PENGUJIAN

Bab ini akan menjelaskan mengenai hasil dari pengujian yang dilakukan, pengujian konektivitas DTN, dan pengujian fungsionalitas *offline browser*. Dari hasil pengujian dapat dianalisa performa dari sistem dan dapat dibandingkan dengan rancangan sistem awal.

5.1 Pengujian

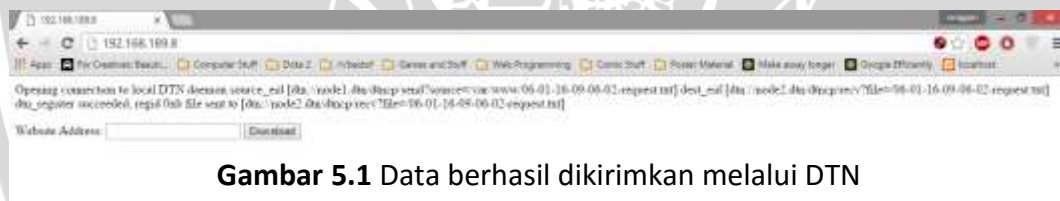
Pengujian yang dilakukan meliputi pengujian fungsionalitas dan pengujian konektivitas DTN.

5.1.1 Pengujian Konektivitas DTN

Pengujian konektivitas DTN dilakukan untuk memastikan DTN berfungsi dengan baik dan dapat mengirim file .txt berisi *request* dari node pertama ke node yang terhubung dengan internet.

5.1.1.1 Skenario Terdapat Dua Node Berdekatan yang Masing-Masing DTN nya Menyalakan

Skenario ini dilakukan dengan *client* mengirimkan data dan terdapat node DTN yang siap menjadi perantara ke node berikutnya. Pengujian ini dilakukan dengan mencoba mengirimkan file antar node sesuai dengan skenario penggunaan sebenarnya.



Gambar 5.1 Data berhasil dikirimkan melalui DTN

Gambar 5.1 menunjukkan pengiriman file berisi URL yang diminta oleh pengguna berhasil dikirimkan ke node tujuan dengan DTN.

```

root@node1: /home/node1/Downloads/dtn-2.3.0/daemon
root@node1: /home/node1/Downloads/dtn-2... root@node1: /home/node1/Downloads/dtn-2...
brslng
[1452085115.262514 /S10 notice] the logs easier after extended tests. These were
added for
[1452085115.262520 /S10 notice] the summer 2010 N4C summer trial and are mainly
new lines
[1452085115.262525 /S10 notice] that allow tracing bundles across multiple hops.
[1452085115.262530 /S10 notice] To extract these do the following...
[1452085115.262535 /S10 notice] grep "S10 info" <dtn-d-log-files> | grep
-v grep | awk '{print $4,$5}' | sort -n
[1452085115.263133 /dtn/bundle/daemon notice] loading bundles from data store
[1452085115.263526 /dtn/bundle/daemon/loop warning] event LINK_CREATED was in que
ue for 3.124106 seconds
[1452085115.264208 /dtn/bundle/daemon/loop warning] event ROUTE_ADD was in queue
for 3.124654 seconds
node1 dtn% [1452088424.699762 /tlnr warning] tlnr thread running slow -- tlnr
is 12317 msec late

node1 dtn%
node1 dtn% bundle list
All Bundles (0):
node1 dtn%

```

Gambar 5.2 Bundle pada node 1

Kemudian pada Gambar 5.2 menunjukkan bahwa file tersebut berhasil dikirimkan dan *bundle* yang berada pada node1 berjumlah 0 karena sudah terkirim

```
root@node2: /home/node2/Downloads/dtn-2.9.0/apps/dtncpd
root@node2:/home/node2/node1.dtn# cd /home/node2/Downloads/dtn-2.9.0/apps/dtncpd
root@node2:/home/node2/Downloads/dtn-2.9.0/apps/dtncpd# dtncpd /home/node2/
opening connection to dtn router...
local_eld [dtn://node2.dtn/dtncp/recv?file=*]
dtn_find_registration succeeded, regid 0xa
dtn_rcv [dtn://node2.dtn/dtncp/recv?file=*]...
*****
File Received at Sat Dec 26 15:45:16 2015
*****
host : node1.dtn
path :
file : 26-12-15-03-45-16-request.txt
loc : /home/node2/node1.dtn//26-12-15-03-45-16-request.txt
size : 25 bytes
Moving payload to final filename: 'mv /tmp/bundlePayload_RKdkGk /home/node2/node1.dtn//26-12-15-03-45-16-request.txt'
*****
dtn_rcv [dtn://node2.dtn/dtncp/recv?file=*]...
```

Gambar 5.3 Data berhasil diterima pada node 2

Dan pada gambar 5.3 ditunjukkan bahwa *file* yang dikirim dari node1 berhasil diterima oleh node2.

5.1.1.2 Skenario Node DTN Tidak Siap Menerima Data

Skenario ini menggambarkan keadaan saat *client* mengirim file *request* tetapi tidak terdapat node DTN yang siap menjadi perantara ke node berikutnya.



Gambar 5.4 Data berhasil dikirimkan melalui DTN

Gambar 5.4 menunjukkan pengiriman file berisi URL yang diminta oleh pengguna berhasil dikirimkan ke node tujuan dengan DTN.

```
root@node1: /home/node1/Downloads/dtn-2.9.0/daemon
root@node1:/home/node1/Downloads/dtn-2... root@node1:/home/node1/Downloads/dtn-2...
[1452085115.262530 /510 notice] To extract these do the following...
[1452085115.262535 /510 notice] grep "510 info" <dtnd-log-files> | grep
-v grep | awk '{print $4,$5}' | sort -n
[1452085115.263133 /dtn/bundle/daemon notice] loading bundles from data store
[1452085115.263526 /dtn/bundle/daemon/loop warning] event LINK_CREATED was in que
ue for 3.124106 seconds
[1452085115.264208 /dtn/bundle/daemon/loop warning] event ROUTE_ADD was in queue
for 3.124654 seconds
node1 dtn% [1452088424.699762 /timer warning] timer thread running slow -- timer
is 12317 nsecs late

node1 dtn%
node1 dtn% bundle list
All Bundles (0):

node1 dtn% bundle list
All Bundles (1):
  2 : dtn://node1.dtn/dtncp/send?source=/var/www/06-01-16-09-25-42-reques
t.txt -> dtn://node2.dtn/dtncp/recv?file=06-01-16-09-25-42-request.txt length 1
9
node1 dtn%
```

Gambar 5.5 Bundle masih tersimpan di node1



Gambar 5.7 Link yang berada pada halaman detik.com dapat diakses secara *offline*

Gambar 5.7 menunjukkan bahwa halaman yang di-link pada situs yang diunduh juga dapat diduplikasi dan dapat diakses pengguna secara *offline*.

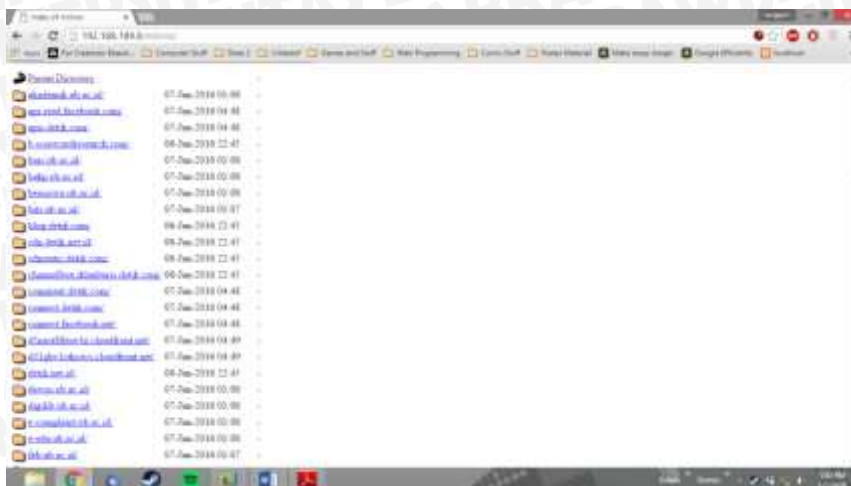
Fitur ketiga yang diuji adalah fitur notifikasi apabila situs yang diminta sudah pernah diminta. Pengujian dilakukan dengan memasukkan URL yang sama dengan yang sudah pernah diminta sebelumnya, yang akan diuji coba adalah <http://www.detik.com>.



Gambar 5.8 Notifikasi ketersediaan situs

Gambar 5.8 menunjukkan skenario saat pengguna memasukkan URL yang sudah pernah diminta maka akan diberikan opsi untuk memeriksa folder yang berisi halaman yang diminta dan pilihan untuk pengguna mengirim ulang permintaan situs.

Fitur keempat yang diuji adalah fitur index dari situs-situs yang sudah diduplikasi dan disimpan, beserta index dari halaman yang diduplikasi dari *link* yang terdapat pada halaman pertama yang diminta oleh pengguna.



Gambar 5.9 Index situs yang sudah tersimpan

Gambar 5.9 menunjukkan index dari situs-situs yang sudah tersimpan pada node1 beserta halaman yang juga diduplikasi dari *link* yang berada pada halaman yang diminta.

5.1.3 Pengujian Perilaku *Web Browsing* pada Jenis Situs yang Berbeda

Pengujian perilaku *web browsing* pada jenis situs yang berbeda dilakukan dengan membuka situs-situs yang berbeda dan melihat kemampuan dari *offline browser* yang digunakan.

Untuk pengujian pertama dilakukan *request* untuk situs berita detik.com dan melihat apakah *offline browser* mampu menduplikasi situs tersebut untuk diakses secara *offline*. Gambar 5.6 menunjukkan detik.com dapat diakses secara *offline*.



Gambar 5.10 Situs detik.com dapat dilihat secara *offline* dengan sempurna

Gambar 5.10 menunjukkan situs detik.com yang sudah berhasil diduplikasi dan dapat diakses secara *offline* karena sudah tersimpan pada web server pada node1.



Gambar 5.11 Artikel pada detik.com dapat dilihat secara *offline*

Begitu juga dengan halaman yang di-link pada halaman utama situs detik.com yang diminta oleh pengguna dapat diakses secara *offline* seperti ditunjukkan pada Gambar 5.11

Untuk pengujian kedua, dilakukan *request* untuk situs statis ub.ac.id untuk melihat kemampuan *offline browser* dalam menduplikasi situs tersebut.



Gambar 5.12 Situs ub.ac.id dapat dilihat secara *offline*

Gambar 5.12 menunjukkan bahwa *offline browser* dapat menduplikasi situs statis seperti ub.ac.id untuk dapat diakses secara *offline*.





Gambar 5.13 Offline browser tidak dapat menduplikasi google.com

Gambar 5.13 menunjukkan bahwa situs google.com tidak dapat diduplikasi karena google me-*redirect* koneksi pengguna kepada CDN yang terdekat dengan pengguna yang mengaksesnya.



Gambar 5.14 Google.co.id dapat diakses secara *offline*

Situs google.com tidak dapat diduplikasi oleh *offline browser* karena google me-*redirect* koneksi pengguna kepada CDN yang terdekat dengan pengguna yang mengaksesnya. Hal ini dapat diatasi dengan melakukan *request* URL tujuan dari *redirect* tersebut, dalam hal ini www.google.co.id/index6c04.html. Seperti yang ditunjukkan pada gambar 5.14, www.google.co.id/index6c04.html berhasil diakses secara *offline*. Akan tetapi, search engine milik google tetap tidak dapat digunakan karena google memasang filter agar HTTrack tidak dapat mengakses hasil pencarian google.



Gambar 5.15 *Offline browser* tidak dapat menduplikasi facebook.com

Gambar 5.15 menunjukkan situs facebook tidak dapat diakses menggunakan *offline browser* HTTrack karena file index.html yang dibuat berisi *redirect* menuju index.html yang sama yang menyebabkan *loop*.

Tabel 5.1 Tabel hasil pengujian Perilaku *Web Browsing* pada Jenis Situs yang Berbeda

Websites	Hasil
www.detik.com	Dapat diunduh dan dibuka secara <i>offline</i> beserta <i>link-link</i> yang terkait.
www.ub.ac.id	Dapat diunduh dan dibuka secara <i>offline</i> beserta <i>link-link</i> yang terkait.
www.google.com	Situs dapat diakses namun fitur <i>search engine</i> tidak dapat digunakan.
www.facebook.com	Tidak dapat diduplikasi dengan sukses. Karena yang dapat diunduh hanya file index.html berisi <i>redirect</i> menuju index.html yang sama dan menciptakan <i>loop</i> .

BAB 6 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap skripsi untuk pengembangan lebih lanjut.

6.1 Kesimpulan

Dari hasil penelitian Implementasi *Offline Browsing* pada *Delay Tolerant Network* dapat disimpulkan bahwa :

1. Sistem yang dibuat dapat membantu pengguna dalam melakukan *web browsing* dengan menggunakan DTN dan meningkatkan efisiensi *web browsing* pada DTN karena sistem juga menduplikasi *link* yang berada pada halaman yang diminta tanpa pengguna harus melakukan *request* untuk masing-masingnya.
2. *Offline browser* yang diimplementasikan tidak dapat digunakan untuk mengakses dan mengunduh situs dinamis seperti facebook, twitter, dan sejenisnya. Dan hanya dapat digunakan untuk mengakses situs statis seperti situs berita detik, dan website resmi ub.ac.id.

6.2 Saran

Saran yang dapat penulis sampaikan untuk pengembangan penelitian ini lebih lanjut adalah :

1. *Offline browser* yang digunakan terbatas tidak dapat menduplikasi *server-side code* seperti PHP atau mengakses *database*. Untuk pengembangannya disarankan mengembangkan sebuah *offline browser* yang dapat melewati keterbatasan tersebut.
2. Untuk pengembangan dari sistem ini disarankan dibangun sistem untuk melakukan *request webpage* tanpa pengguna harus mengisi URL yang diinginkan ke dalam form melainkan menerima *request* dari pengguna dari URL yang dimasukkan ke dalam address bar sehingga lebih mudah digunakan.

DAFTAR PUSTAKA

- Anon., 2012. *HTTrack Software for Self-Access Language Learning*. [Online] Available at: <https://sites.google.com/site/journaloftechnologyforelt/archive/july-2012/4-htrack-software-for-self-access-language-learning> [Accessed 10 October 2015].
- Bong, Z., 2012. *Web Access over Delay Tolerant Networks*, s.l.: International Journal of Future Computer and Communication.
- dtng.org, 2014. *DTN2*. [Online] Available at: <http://www.dtnrg.org/docs/code/DTN2/doc/manual/intro.html> [Accessed 8 April 2015].
- Fall, K., 2003. *A Delay-tolerant Network Architecture for Challenged Internets*, Berkeley: Intel Research.
- Kurnikov, A., 2011. *Web Browser Extension for DTN*, Helsinki: Faculty of Electronics, Communications and Automations, Aalto University, Finland.
- Lindgren, A., 2009. *The Quest for a Killer App for Opportunistic and Delay Tolerant Networks*, s.l.: ACM MobiCom Workshop on Challenged Networks (CHANTS).
- Sari, N. I., 2015. *Implementasi Web Service pada Delay Tolerant Network*, Malang: Implementasi Web Service pada Delay Tolerant Network. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
- Trust, T. I., 2007. *Delay-Tolerant Networking Architecture*. [Online] Available at: <http://tools.ietf.org/pdf/rfc4838.pdf> [Accessed 8 April 2015].
- Warthman, F., 2003. *Delay-Tolerant Networks (DTNs) A Tutorial*, s.l.: dtng.