

SIMULASI DAN ANALISIS JARINGAN KOMPUTER

DENGAN NS-2

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Strata-1
Teknik Informatika**



oleh :

Nama : Mardiansyah

No. Mahasiswa : 05 523 359

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011**

PENGAKUAN

Demi Allah, saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang setiap satunya telah saya jelaskan sumbernya. Jika di kemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, Agustus 2011



Mardiansyah

05523359



LEMBAR PENGESAHAN PEMBIMBING
SIMULASI DAN ANALISIS JARINGAN KOMPUTER
DENGAN NS-2

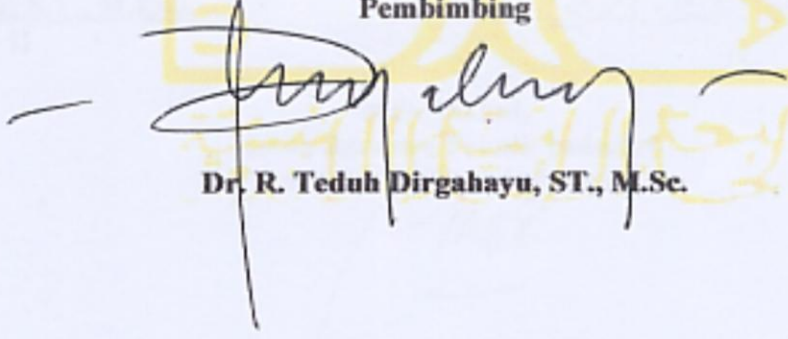
TUGAS AKHIR

oleh :

Nama : Mardiansyah
No. Mahasiswa : 05 523 359

Yogyakarta, Agustus 2011

Pembimbing



Dr. R. Teduh Dirgahayu, ST., M.Sc.

LEMBAR PENGESAHAN PENGUJI

SIMULASI DAN ANALISIS JARINGAN KOMPUTER

DENGAN NS-2

TUGAS AKHIR

oleh :

Nama : Mardiansyah

No. Mahasiswa : 05 523 359

Telah dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, Agustus 2011

Tim Penguji

Dr. R. Teduh Dirgahayu, ST., M.Sc.
Ketua

Syarif Hidayat, S.Kom., M.IT
Penguji I

Hendrik, S.T., M.Eng.
Penguji II

Mengetahui,

Ketua Jurusan Teknik Industri
Universitas Islam Indonesia



Vadi Prayudi, S.Si., M.Kom.

HALAMAN PERSEMBAHAN

*Alhamdulillah, atas izin Allah SWT skripsi ini dapat terselesaikan.
Dengan Hati Yang Tulus dan Ikhlas Kupersembahkan Buah Karyaku ini Kepada:*

*Ayahanda Akhmad Fauzi
Yang Tak Pernah Lelah Memberi Petuah dan Motivasi-Motivasi
Hingga Semangatku Tak Pernah Padam.
Dengan Keringat dan Usahamu Yang Tak Kenal Lelah selama Ini
Akhirnya Kudapat Meraih Semua Ini*

*Ibunda Kartini Tercinta
Yang Senantiasa Membimbing dan Mendoakan
Dengan Penuh Kesabaran dan Keihlasan
Takkan Cukup Kata Tuk Lukiskan Rasa Hormat dan Terima Kasihku Padamu.
Kau Adalah Anugrah Terindah Dalam Hidupku.*

*Adik-adikku Fachrul Rozi dan Yudha Fathur Rachman Tercinta
Terima Kasih Atas Kesempatan Tumbuh, Bermain, Belajar & Hidup Bersama
Engkau Adalah Saudara Terbaik Yang Pernah Ku Miliki*

MOTTO

إِذَا مَاتَ الْإِنْسَانُ انْقَطَعَ عَمَلُهُ إِلَّا مِنْ ثَلَاثٍ؛ صَدَقَةٌ جَارِيَةٍ أَوْ عِلْمٌ يُنْتَفَعُ بِهِ أَوْ وَلَدٌ صَالِحٌ يَدْعُو لَهُ.

"Jika manusia mati terputuslah amalnya kecuali tiga: shadaqah jariyah, atau ilmu yang dia amalkan atau anak shalih yang mendoakannya." **(HR. Muslim)**

مَنْ يُرِدِ اللَّهُ بِهِ خَيْرًا يُفَقِّهْهُ فِي الدِّينِ وَإِنَّمَا أَنَا قَاسِمٌ وَاللَّهُ هُوَ الْمُعْطِي وَلَا تَزَالُ هَذِهِ الْأُمَّةُ قَائِمَةً عَلَى أَمْرِ اللَّهِ لَا يَضُرُّهُمْ مَنْ خَالَفَهُمْ حَتَّى يَأْتِيَ أَمْرُ اللَّهِ.

"Barangsiapa yang Allah kehendaki padanya kebaikan, maka Allah akan fahamkan dia dalam (masalah) dien. Aku adalah Al-Qasim (yang membagi) sedang Allah Azza wa Jalla adalah yang Maha Memberi. Umat ini akan senantiasa tegak di atas perkara Allah, tidak akan memadharatkan kepada mereka, orang-orang yang menyelisihi mereka sampai datang putusan Allah." **(HR. Al-Bukhari)**

مَنْ سَلَكَ طَرِيقًا يَلْتَمِسُ فِيهِ عِلْمًا سَهَّلَ اللَّهُ لَهُ بِهِ طَرِيقًا إِلَى الْجَنَّةِ.

"Barangsiapa menempuh jalan untuk mencari ilmu, maka Allah mudahkan baginya jalan menuju Surga." **(HR. Muslim)**

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji bagi Allah, Tuhan alam semesta. Shalawat dan salam semoga terlimpahkan kepada Rasulullah *Shallallahu Alaihi wa Sallam*, keluarganya, sahabatnya dan pengikutnya hingga akhir zaman.

Sesungguhnya atas petunjuk, pertolongan dan bimbingan-Nya maka Tugas Akhir ini dapat terselesaikan dengan baik. Tugas Akhir ini merupakan salah satu syarat untuk menyelesaikan jenjang studi Strata 1 Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

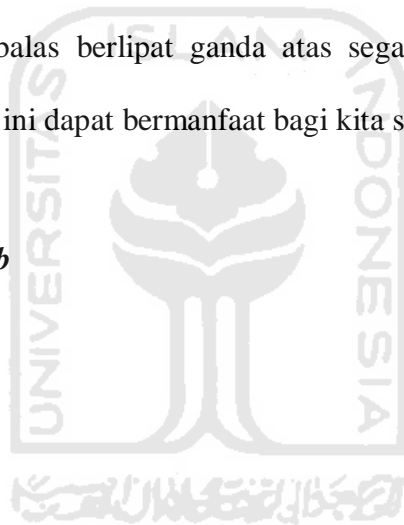
Keberhasilan terselesaikannya Tugas akhir ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu dengan rasa hormat dan terima kasih yang sebesar-besarnya penulis sampaikan kepada :

1. Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Yudi Prayudi, S.Si., M.Kom selaku Ketua Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak DR. R. Teduh Dirgahayu, ST., M.Sc. selaku Dosen Pembimbing yang telah memberikan bantuan dan arahnya dalam penyusunan Tugas Akhir ini.
4. Ayah, Ibu dan keluarga yang selalu memberikan perhatian, do'a dan dorongan.

5. Nova Winta yang selalu memberikan perhatian, do'a, dukungan serta semangat sehingga bisa menyelesaikan tugas akhir ini.
6. Rekan Agung H., Aruka S., Hafni C., Yudha R.H, Jarrot W, dan teman-teman teknik informatika angkatan 2005 yang telah membantu dalam pengerjaan tugas akhir ini.
7. Pihak–pihak lain yang tidak mungkin penulis sebutkan satu per satu, terima kasih atas perhatian dan dukungannya.

Semoga Allah membalas berlipat ganda atas segala amal sholehnya. Harapan penulis semoga Tugas Akhir ini dapat bermanfaat bagi kita semua. Amin.

Wassalamu'alaikum Wr. Wb



Yogyakarta, Agustus 2011

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAKUAN.....	ii
SURAT KETERANGAN SELESAI PENELITIAN	iii
HALAMAN PENGESAHAN PEMBIMBING	iv
HALAMAN PENGESAHAN PENGUJI	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR SCRIPT	xiii
DAFTAR GAMBAR	xiv
DAFTAR DIAGRAM DAN TABEL	xvi
ABSTRAK	xvii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah.....	4
1.5 Metodologi Penelitian.....	4
1.6 Sistematika Penulisan	5

BAB II INSTALASI NETWORK SIMULATOR 2

2.1	Ubuntu	6
2.1.1	Instalasi Ns-2	7
2.2.	Windows	12
2.2.1	Instalasi Cygwin.....	12
2.2.2	Instalasi Ns-2	17

BAB III DASAR_DASAR NS-2

3.1	Dasar Teori.....	22
3.2.1	Network Simulator 2	22
3.2	Elemen Ns-2.....	24
3.2.1	Event Scheduler	24
3.2.2	Network Nodes	25
3.2.3	Agen dan Aplikasi.....	26
3.2.4	Tracing.....	29
3.3	Visualisasi dengan Nam	31

BAB IV CONTOH DAN HASIL SIMULASI

4.1	Pembuatan Simulasi.....	32
4.2	Hasil Simulasi	36
4.3	Analisis Simulasi.....	45

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan	50
5.2 Saran.....	50

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR SCRIPT

Script 3.1 Beberapa Event Scheduler.....	25
Script 3.2 Pembuatan Node	25
Script 3.3 Membuat koneksi TCP.....	27
Script 3.4 Contoh traffic pada TCP.....	28
Script 3.5 Membuat koneksi UDP	28
Script 3.6 Penempatan Node untuk Nam	31
Script 4.1 Isi file tcpudp.nam.....	38
Script 4.2 Isi file tcpud.tr.....	39
Script 4.3 Isi file CBR.tr.....	40
Script 4.4 Isi file VBR.tr.....	41

DAFTAR GAMBAR

Gambar 2.1 Perintah ekstrak file ns-allinone-2.34.tar.gz di terminal.....	7
Gambar 2.2 Perintah gedit untuk mengedit file Makefile.in.....	7
Gambar 2.3 Baris yang harus diubah pada file Makefile.in.....	8
Gambar 2.4 Perintah untuk memulai instalasi Ns-2.34.	8
Gambar 2.5 Proses instalasi ns 2.34 telah selesai.....	8
Gambar 2.6 Perintah gedit untuk mengedit file bashrc	9
Gambar 2.7 File bashrc yang telah ditambahkan path.....	10
Gambar 2.8 Perintah ns untuk menjalankan ns 2.34.....	10
Gambar 2.9 Perintah ./validate untuk melakukan validasi data.	11
Gambar 2.10 Proses validasi yang sedang berjalan.....	11
Gambar 2.11 Pilihan opsi pertama untuk fresh install.....	13
Gambar 2.12 Penentuan direktori root.....	13
Gambar 2.13 Penentuan direktori lokal untuk menyimpan file-file instalasi.	14
Gambar 2.14 Daftar server dan situs untuk menngunduh Cygwin.	14
Gambar 2.15 Install semua paket termasuk paket-paket lama.....	15
Gambar 2.16 Proses download paket file Cygwin.	15
Gambar 2.17 XWin Server di start menu.....	16
Gambar 2.18 Tampilan Cygwin... ..	16
Gambar 2.19 Direktori ns-allinone-2.31.tar.gz.	17
Gambar 2.20 Direktori tk8.4.14.	18

Gambar 2.21 perintah untuk patch tk8.4.14.....	18
Gambar 2.22 Perintah untuk pindah ke direktori ns-allione-2.3.1.	19
Gambar 2.23 Instalasi ns-allinone-2.31 di Cygwin telah selesai.	19
Gambar 2.24 File bashrc di direktori user.....	20
Gambar 3.1 Arsitektur Dasar Ns-2.....	23
Gambar 3.2 Pembuatan Node.....	25
Gambar 3.3 Inserting Trace Object..	30
Gambar 3.4 Format trace file... ..	30
Gambar 3.5 Visualisasi dengan Nam.....	31
Gambar 4.1 Contoh Simulasi.....	33
Gambar 4.2 Network Animator.....	36
Gambar 4.3 Simulasi pada Nam.....	37
Gambar 4.4 Hasil kualitas.awk.....	43
Gambar 4.5 Hasil grafik tcpudp.tr dengan X Graph.....	43
Gambar 4.5 Hasil grafik CBR.tr dan VBR.tr dengan X Graph.....	44

DAFTAR DIAGRAM DAN TABEL

Diagram 4.1 Flow chart simulasi	35
Tabel 4.1 Hasil simulasi skenario 1.....	47
Tabel 4.2 Hasil simulasi skenario 2.....	47
Tabel 4.3 Hasil simulasi skenario 3.....	48
Tabel 4.4 Hasil simulasi skenario awal.	48



Abstraksi

Simulator jaringan adalah suatu perangkat lunak atau keras untuk melakukan simulasi, prediksi terhadap proses-proses yang terjadi dalam suatu jaringan komputer, tanpa menggunakan sistem jaringan komputer yang sesungguhnya. Simulator jaringan ini digunakan oleh industri, akademisi dan pemerintah untuk keperluan tertentu.

Ns-2 (Network Simulator 2) adalah open-source event simulator yang dirancang khusus untuk penelitian dalam jaringan komputer. Simulator ini merupakan salah satu simulator jaringan yang paling banyak digunakan hingga saat ini, memiliki beragam fitur, layak untuk dipelajari dan digunakan oleh mahasiswa.

Pada tugas akhir ini akan diberikan panduan untuk menggunakan Ns-2. Dari cara penginstallan (yang mencakup sistem operasi yang digunakan, versi Ns-2 serta paket-paket yang dibutuhkan dan sebagainya), dasar-dasar Ns-2 hingga studi kasus sederhana dan penerapannya. Sehingga diharapkan dapat memberikan pembelajaran tambahan kepada mahasiswa mengenai Ns-2.

Kata kunci : Network Simulator, Ns-2, Nam, Cygwin, OTcl.

BAB II

INSTALASI NETWORK SIMULATOR 2

Ns-2 bisa dijalankan di semua sistem operasi berbasis UNIX, seperti Linux, BSD dan OSX. Ns-2 juga bisa dijalankan di sistem operasi Windows dengan menggunakan Cygwin. Cygwin merupakan kumpulan dari berbagai piranti perangkat lunak *open source* yang memungkinkan aplikasi berbasis UNIX untuk dikompilasi dan dijalankan di sistem operasi Windows.

Pada bab ini akan diberikan cara instalasi Ns-2 di sistem operasi berbasis UNIX (dalam hal ini adalah Ubuntu) dan sistem operasi Windows.

2.1 Ubuntu

Beberapa hal yang harus diperhatikan pada saat instalasi di Ubuntu yaitu :

1. Kebutuhan minimal :

- a. Prosesor : 1 GHz
- b. Memori : 384 MB
- c. Ruang *harddisk* : 5GB

2. Ada koneksi internet.

Koneksi internet sangat dibutuhkan untuk kelancaran proses instalasi, karena membutuhkan beberapa data/paket yang harus diunduh untuk menjalankannya.

3. Versi Ubuntu yang digunakan.

Pada laporan ini, penulis menggunakan Ubuntu 10.04, namun penulis juga akan memberikan catatan instalasi untuk Ubuntu versi sebelumnya.

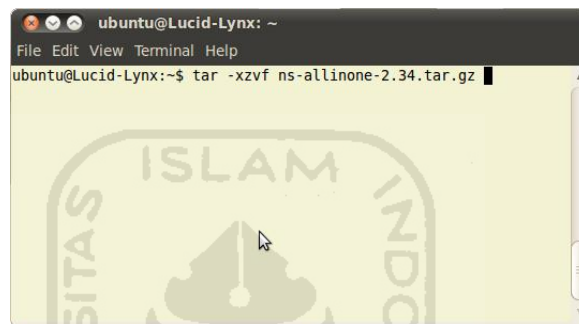
4. Versi Ns-2 yang digunakan.

Pada laporan ini, penulis menggunakan adalah Ns-2.34 allinone (versi terbaru) dengan ukuran file 57 MB, yang diunduh di <http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.34/>.

2.1.1 Instalasi NS-2

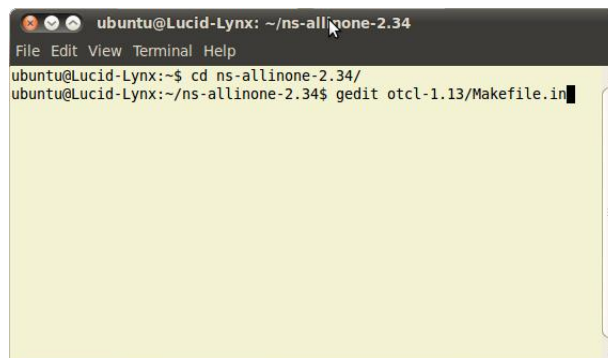
1. Perbarui sistem agar sesuai dengan kebutuhan Ns-2 dengan perintah
`$ sudo apt-get install build-essential autoconf automake libxmu-dev gcc-4.3`
2. Ekstrak file ns-allinone-2.34.tar.gz dengan perintah `$ tar -xzf ns-allinone-2.34.tar.gz` (seperti pada Gambar 2.1)

Pada lokasi dimana instalasi Ns-2 akan dilakukan. Penulis meletakkan file ns-allinone-2.34 di direktori home/ubuntu/



Gambar 2.1 Perintah ekstrak file ns-allinone-2.34.tar.gz di terminal.

3. Masuk ke direktori ns-allinone-2.34 dengan perintah
`$ cd ns-allinone-2.34`
4. Untuk Ubuntu 09.04, 09.10 dan 10.04 dengan program *gedit* (seperti pada Gambar 2.2) ubah variabel sistem di dalam file *Makefile.in* yang berada di dalam direktori `/ns-allinone-2.34/otcl-1.13/` dari `@CC@` menjadi `gcc-4.3` (seperti pada Gambar 2.3), simpan setelah diubah. Untuk Ubuntu 07.04 dan 08.04 bisa langsung melanjutkan ke tahap 5.

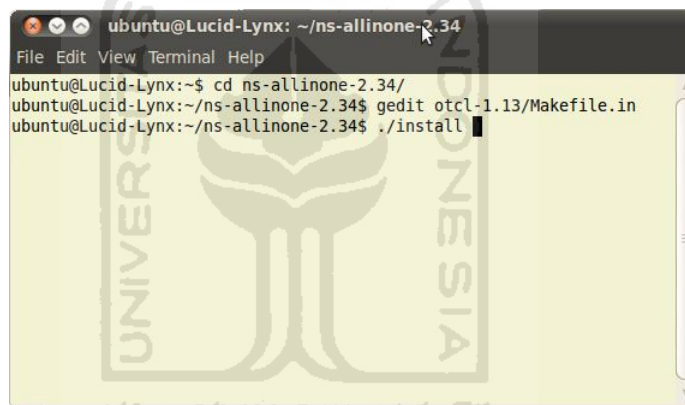


Gambar 2.2 Perintah gedit untuk mengedit file Makefile.in.



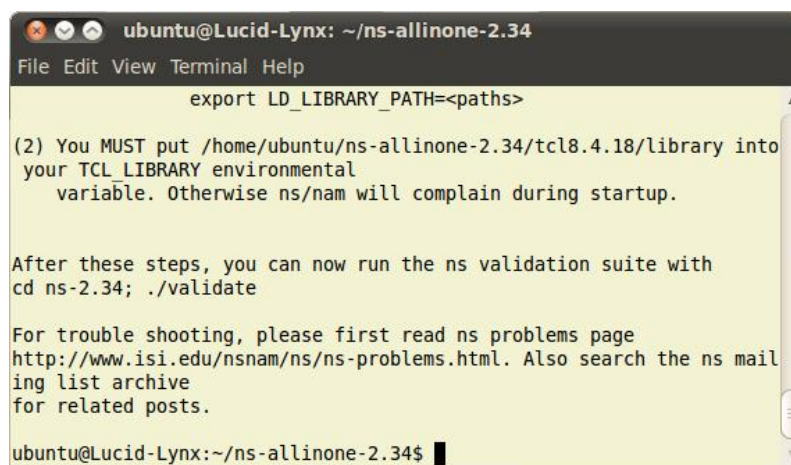
Gambar 2.3 Baris yang harus diubah pada file Makefile.in.

5. Mulai proses instalasi dengan perintah `$./install` (seperti Gambar 2.4)



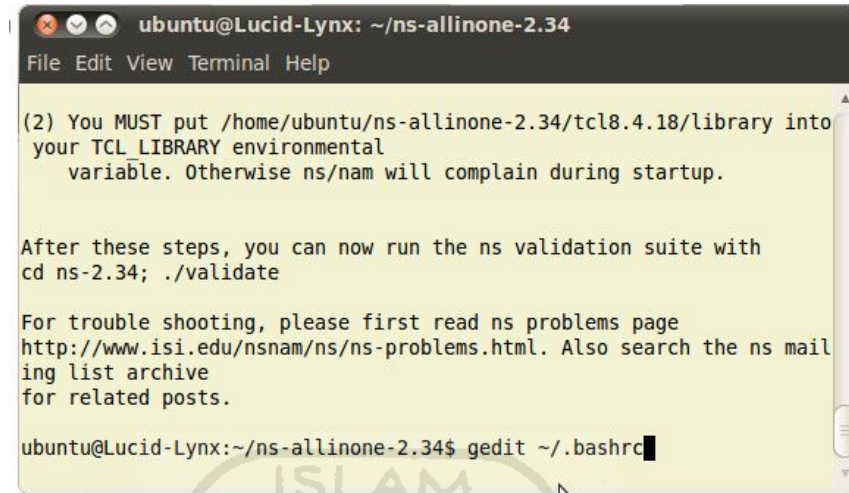
Gambar 2.4 Perintah untuk memulai instalasi Ns-2.34.

6. Tunggu proses instalasi sampai selesai (seperti gambar 2.5).



Gambar 2.5 Proses instalasi ns 2.34 telah selesai.

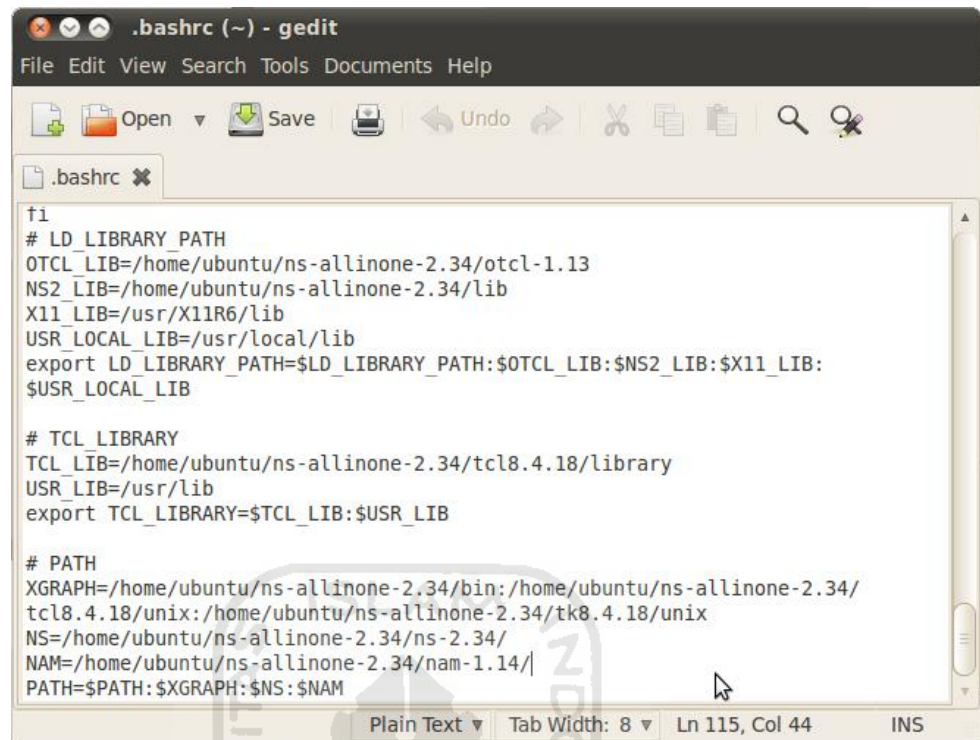
7. Edit variabel sistem dengan program `$ gedit ~/.bashrc` (seperti pada gambar 2.6)



Gambar 2.6 Perintah gedit untuk mengedit file bashrc

Pada baris terakhir, masukkan variabel-variabel di bawah ini dengan catatan ubah `/path` sesuai dengan direktori tempat mengekstrak dan instalasi Ns-2.34. Sebagai contoh penulis mengubah `/path` menjadi `/home/ubuntu/` (seperti pada Gambar 2.7).

```
# LD_LIBRARY_PATH
OTCL_LIB=/path/ns-allinone-2.34/otcl-1.13
NS2_LIB=/path/ns-allinone-2.34/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$U
SR_LOCAL_LIB
# TCL_LIBRARY
TCL_LIB=/path/ns-allinone-2.34/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/path/ns-allinone-2.34/bin:/path/ns-allinone-
2.34/tcl8.4.18/unix:/path/ns-allinone-2.34/tk8.4.18/unix
NS=/path/ns-allinone-2.34/Ns-2.34/
NAM=/path/ns-allinone-2.34/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```



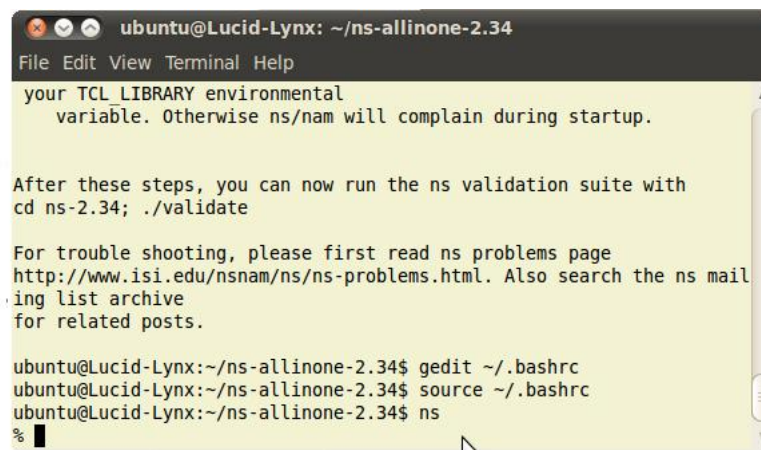
```
# LD_LIBRARY_PATH
OTCL_LIB=/home/ubuntu/ns-allinone-2.34/otcl-1.13
NS2_LIB=/home/ubuntu/ns-allinone-2.34/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:
$USR_LOCAL_LIB

# TCL_LIBRARY
TCL_LIB=/home/ubuntu/ns-allinone-2.34/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
XGRAPH=/home/ubuntu/ns-allinone-2.34/bin:/home/ubuntu/ns-allinone-2.34/
tcl8.4.18/unix:/home/ubuntu/ns-allinone-2.34/tk8.4.18/unix
NS=/home/ubuntu/ns-allinone-2.34/ns-2.34/
NAM=/home/ubuntu/ns-allinone-2.34/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

Gambar 2.7 File bashrc yang telah ditambahkan path.

8. Agar perubahan yang dilakukan langsung diterapkan oleh sistem, pada terminal ketikkan perintah `$ source ~/.bashrc`
9. Uji hasil instalasi Ns-2 dengan menjalankan perintah `$ ns`
Jika muncul tanda %, berarti Ns-2.34 telah berjalan (seperti pada Gambar 2.8).



```
your TCL_LIBRARY environmental
variable. Otherwise ns/nam will complain during startup.

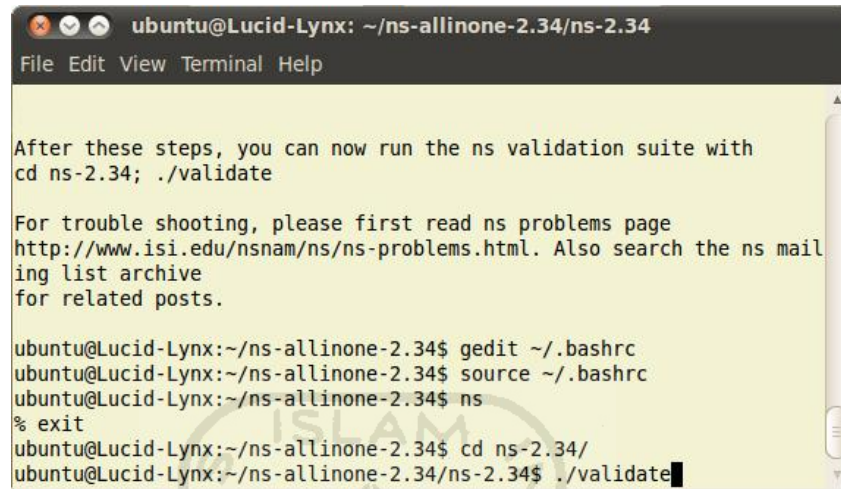
After these steps, you can now run the ns validation suite with
cd ns-2.34; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mail
ing list archive
for related posts.

ubuntu@Lucid-Lynx:~/ns-allinone-2.34$ gedit ~/.bashrc
ubuntu@Lucid-Lynx:~/ns-allinone-2.34$ source ~/.bashrc
ubuntu@Lucid-Lynx:~/ns-allinone-2.34$ ns
%
```

Gambar 2.8 Perintah ns untuk menjalankan ns 2.34

10. Pindah ke direktori Ns-2.34 dengan perintah `$ cd Ns-2.34/`
11. Lakukan validasi apakah ns bisa berjalan dengan baik, dengan perintah `$./validate` (seperti pada Gambar 2.9).



```
ubuntu@Lucid-Lynx: ~/ns-allinone-2.34/ns-2.34
File Edit View Terminal Help

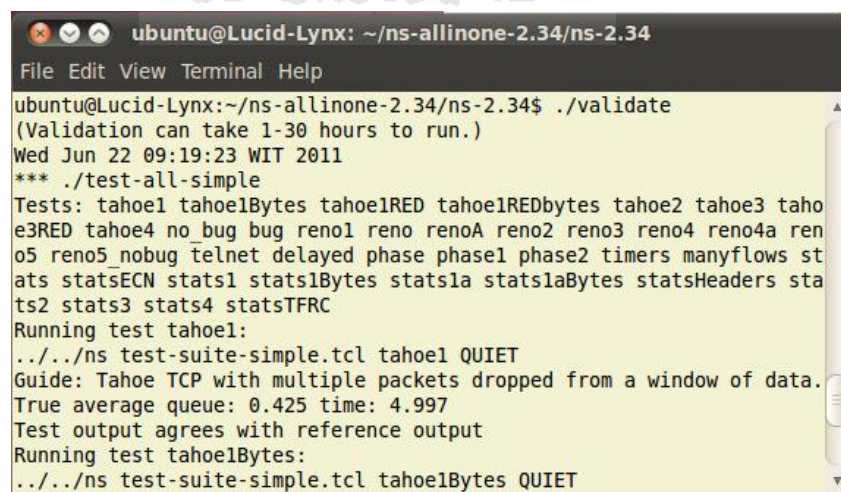
After these steps, you can now run the ns validation suite with
cd ns-2.34; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mail
ing list archive
for related posts.

ubuntu@Lucid-Lynx:~/ns-allinone-2.34$ gedit ~/.bashrc
ubuntu@Lucid-Lynx:~/ns-allinone-2.34$ source ~/.bashrc
ubuntu@Lucid-Lynx:~/ns-allinone-2.34$ ns
% exit
ubuntu@Lucid-Lynx:~/ns-allinone-2.34$ cd ns-2.34/
ubuntu@Lucid-Lynx:~/ns-allinone-2.34/ns-2.34$ ./validate
```

Gambar 2.9 Perintah `./validate` untuk melakukan validasi data.

Proses validasi ini berlangsung antara 1 sampai dengan 30 jam sesuai dengan kemampuan komputer yang digunakan. Pada tahap ini Ns-2.34 akan melakukan pengujian terhadap fungsi-fungsi yang ada. Penulis membutuhkan waktu lebih dari 2 jam untuk menyelesaikan validasi data.



```
ubuntu@Lucid-Lynx: ~/ns-allinone-2.34/ns-2.34
File Edit View Terminal Help

ubuntu@Lucid-Lynx:~/ns-allinone-2.34/ns-2.34$ ./validate
(Validation can take 1-30 hours to run.)
Wed Jun 22 09:19:23 WIT 2011
*** ./test-all-simple
Tests: tahoe1 tahoe1Bytes tahoe1RED tahoe1REDbytes tahoe2 tahoe3 tahoe4 tahoe4Bytes tahoe4RED tahoe4REDbytes tahoe5 tahoe5Bytes tahoe5RED tahoe5REDbytes no_bug bug reno1 reno2 reno3 reno4 reno4a reno5 reno5Bytes reno5RED reno5REDbytes telnet delayed phase phase1 phase2 timers manyflows stats statsECN stats1 stats1Bytes stats1a stats1aBytes statsHeaders stats2 stats3 stats4 statsTFRC
Running test tahoe1:
../ns test-suite-simple.tcl tahoe1 QUIET
Guide: Tahoe TCP with multiple packets dropped from a window of data.
True average queue: 0.425 time: 4.997
Test output agrees with reference output
Running test tahoe1Bytes:
../ns test-suite-simple.tcl tahoe1Bytes QUIET
```

Gambar 2.10 Proses validasi yang sedang berjalan.

2.2 Windows

Proses instalasi Ns-2.34 pada Windows tidak jauh berbeda dengan instalasi pada ubuntu. beberapa hal yang harus diperhatikan, yaitu :

1. Kebutuhan minimal :

- a. Prosesor : 1 GHz
- b. Memori : 384 MB
- c. Ruang *harddisk* : 5GB

2. Versi Windows yang digunakan.

Instalasi Cygwin 1.5 membutuhkan Windows Xp (Home ataupun Professional edition) karena Cygwin 1.5 tidak berjalan lancar di Windows Vista dan Windows 7. Alternatif bagi pengguna Vista dan 7 adalah dengan memasang Ubuntu 10.04 virtual menggunakan perangkat lunak virtualisasi (contoh, Virtual Box dan VMWare).

3. Versi Cygwin yang digunakan dan adanya koneksi internet.

Penulis menggunakan Cygwin versi 1.5 atau yang dikenal juga sebagai Cygwin Legacy, karena paling tinggi kompatibilitasnya terhadap Ns-2 dibanding Cygwin versi terbaru (Cygwin 1.7). Koneksi internet sangat dibutuhkan untuk kelancaran proses instalasi Cygwin karena membutuhkan koneksi internet untuk melakukan instalasi.

4. NS-2 yang digunakan.

Pada instalasi di Cygwin, penulis menggunakan Ns-2.31 allinone, karena Ns-2 yang terbaru belum bisa dipasang di Cygwin baik versi 1.5 maupun 1.7 Ns-2.31 diunduh di

<http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.31/>
dengan ukuran file 70,5 MB.

2.2.1 Instalasi Cygwin

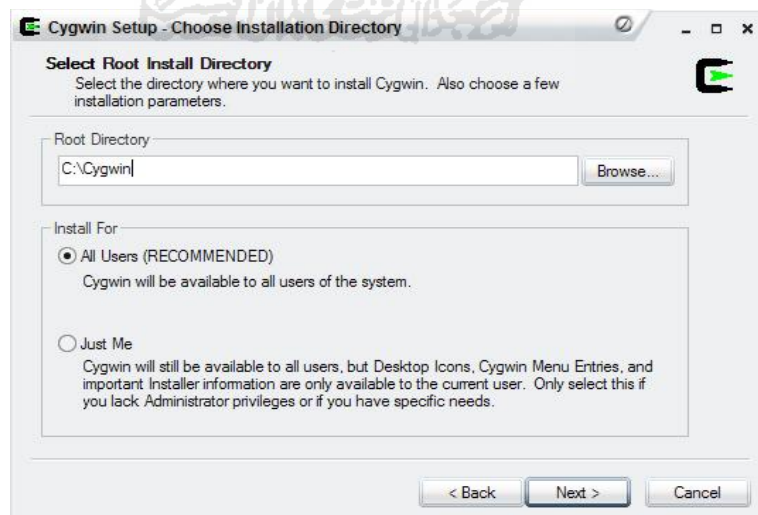
1. Unduh file instalasi Cygwin Legacy di <http://www.cygwin.com>
2. Jalankan *setup-legacy* untuk memulai proses instalasi Cygwin.
3. Pilih opsi pertama: “*Install from Internet*” (seperti pada Gambar 2.11).

Opsi kedua digunakan jika hanya ingin mengunduh file-file instalasi Cygwin. Opsi ketiga digunakan jika telah tersedia file instalasi Cygwin dan melakukan instalasi dari direktori lokal yang berisi file-file instalasi tersebut.



Gambar 2.11 Pilihan opsi pertama untuk *fresh install*.

4. Pilih direktory tempat Cygwin akan dipasang. Penulis menyarankan untuk meletakkan file-file Cygwin di C:\Cygwin. (seperti Gambar 2.12).



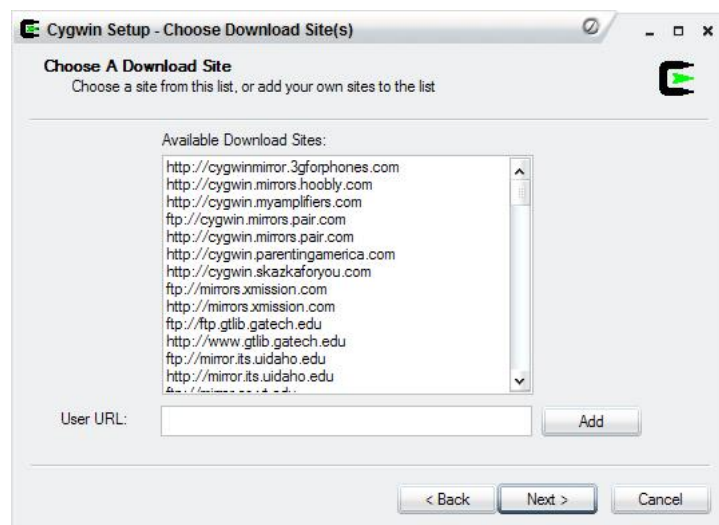
Gambar 2.12 Penentuan direktori root.

5. Tentukan dimana file-file instalasi Cygwin akan disimpan. Penulis meletakkan file-file instalasi tersebut di F:\Master (seperti Gambar 2.13). Hal ini berguna ketika akan melakukan instalasi ulang Cygwin, tidak perlu melakukan instalasi dari internet lagi. Instalasi bisa dilakukan melalui direktori lokal tempat file-file instalasi Cygwin.



Gambar 2.13 Penentuan direktori lokal untuk menyimpan file-file instalasi.

6. Pilih koneksi internet, penulis menyarankan untuk memilih opsi “*Direct Connection*”. Pada laporan ini, penulis melakukan koneksi langsung ke server.
7. Pilih server atau situs dimana Cygwin akan diunduh (seperti Gambar 2.14).



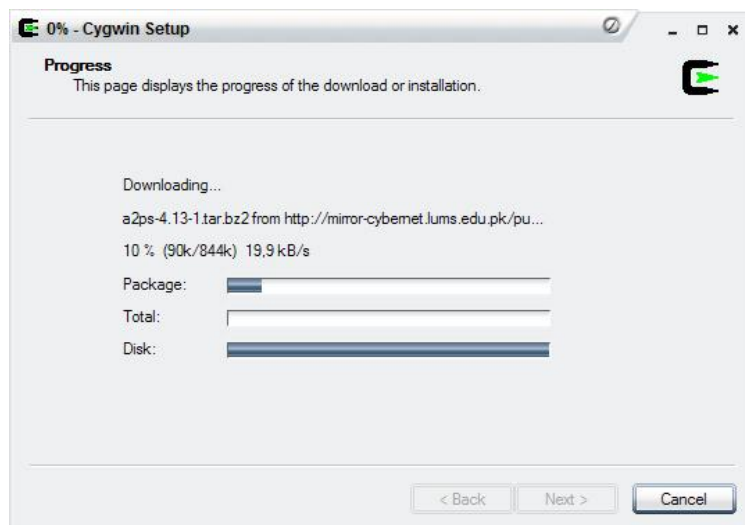
Gambar 2.14 Daftar server dan situs untuk menngunduh Cygwin.

8. Pilih paket-paket yang akan diinstall. Penulis menyarankan untuk memilih semua paket yang ada termasuk paket-paket lama (dengan tidak menconteng opsi “*Hide obsolete packages*”) karena ada beberapa file *obsolete* yang dipakai Cygwin untuk proses instalasi Ns-2.31 allinone (seperti Gambar 2.15).



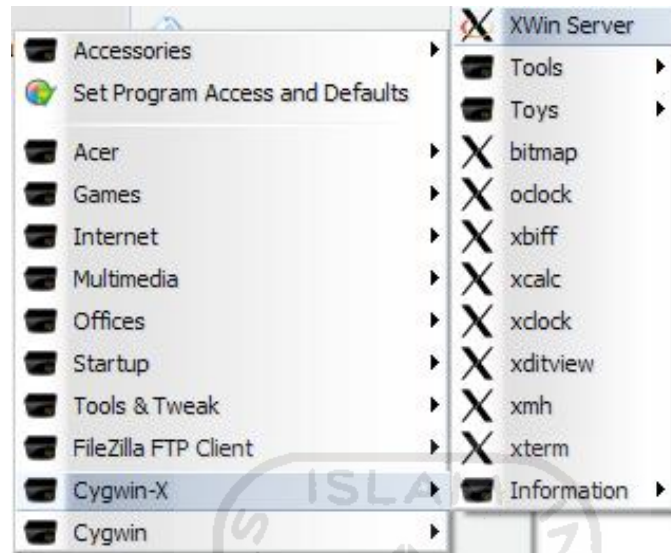
Gambar 2.15 Install semua paket termasuk paket-paket lama.

9. Proses pengunduhan akan dimulai. Setelah proses pengunduhan selesai akan diikuti proses instalasi Cygwin (seperti Gambar 2.16).



Gambar 2.16 Proses download paket file Cygwin.

10. Uji apakah Cygwin yang telah dipasang bisa berjalan dengan baik. Jalankan *XWin Server* yang ada di start menu (seperti Gambar 2.17).



Gambar 2.17 XWin Server di start menu.

11. Jika Cygwin telah terpasang dengan benar maka tampilan seperti Gambar 2.18 yang akan muncul.



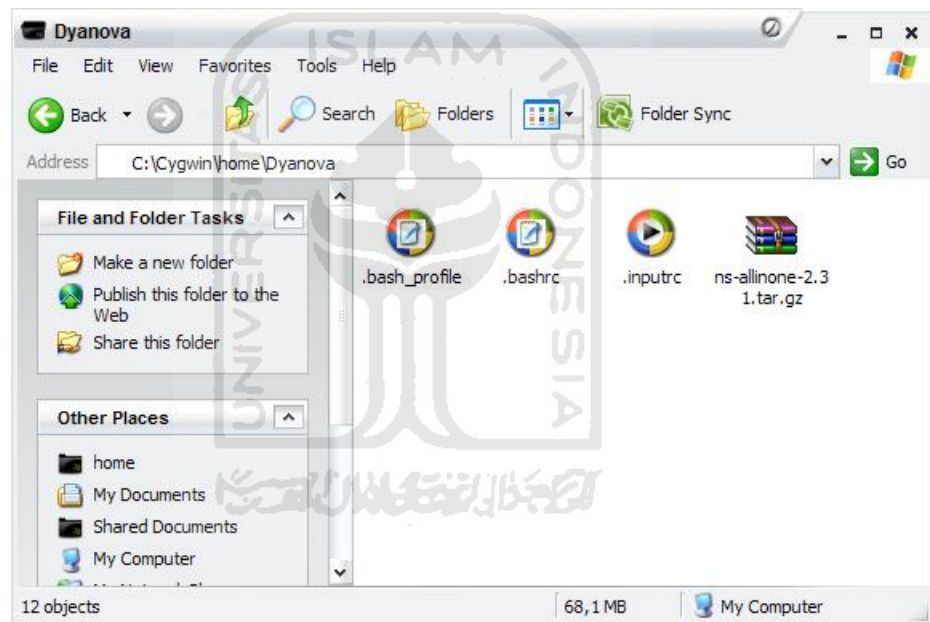
Gambar 2.18 Tampilan Cygwin.

Tahap instalasi Cygwin telah selesai. Berikutnya, instalasi *Ns-2.31 allinone* di Cygwin.

2.2.2 Instalasi Ns-2

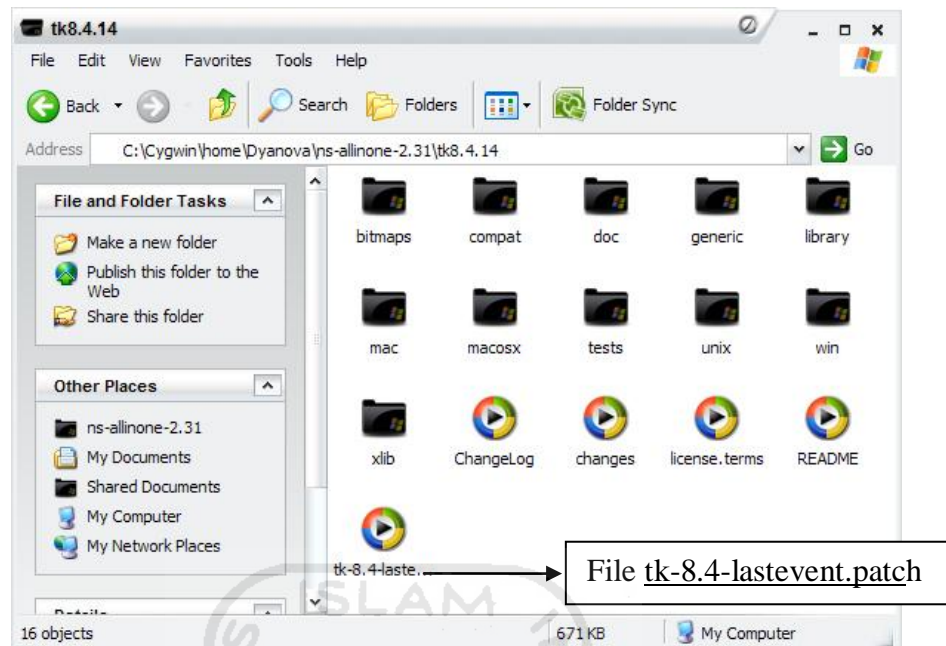
Proses instalasi Ns-2.31 di Cygwin hampir sama dengan instalasi Ns-2.34 di ubuntu. Namun untuk Ns-2.31 yang dipasang di Cygwin, tk-8.414 harus ditambah (*patch*) agar Network Animator (NAM) bisa berjalan dengan benar. Berikut adalah tahapan instalasi Ns-2.31 di Windows Xp melalui Cygwin :

1. Salin dan letakkan file `ns-allinone-2.31.tar.gz` ke direktori `home/namauser` (nama user disesuaikan dengan akun pengguna Windows). Pada laporan ini, penulis menyalin file `ns-allinone-2.31.tar.gz` ke `C:\Cygwin\home\Dyanova` (seperti Gambar 2.19).



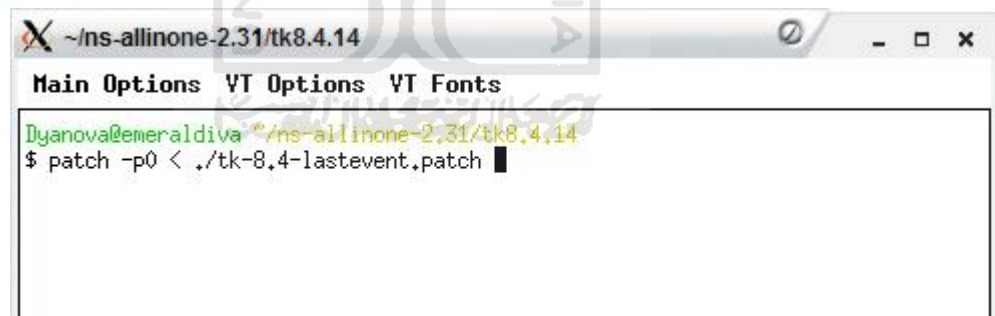
Gambar 2.19 Direktori `ns-allinone-2.31.tar.gz`.

2. Jalankan XWin Server, kemudian ekstrak file `ns-allinone-2.31.tar.gz` dengan perintah `$ tar -xzvf ns-allinone-2.31.tar.gz`
3. Tunggu proses ekstraksiselesai, kemudian unduh patch [tk-8.4-lastevent.patch](http://bugs.gentoo.org/show_bug.cgi?id=225999) di http://bugs.gentoo.org/show_bug.cgi?id=225999.
4. Letakkan patch [tk-8.4-lastevent.patch](#) di direktori `tk8.4.14` yang berada di dalam direktori `ns-allinone-2.31` (seperti gambar 2.20).



Gambar 2.20 Direktori tk8.4.14.

5. Pindah ke direktori tk8.4.14 dengan perintah `$ cd ns-allinone-2.31/tk8.4/`
6. Lakukan penambalاندengan perintah `$ patch -p0 < ./tk-8.4-lastevent.patch`



Gambar 2.21 perintah untuk patch tk8.4.14.

7. Pindah ke direktori ns-allinone-2.31 dengan perintah `$ cd /home/namauser/ns-allinone-2.31`
Ganti nama user sesuai dengan nama direktori pengguna di direktori C:\Cygwin\home\ (seperti Gambar 2.22).


```

X ~/ns-allinone-2.31
Main Options  VT Options  VT Fonts

Dyanova@emeraldiva ~/ns-allinone-2.31/tk8.4.14
$ cd /home/Dyanova/ns-allinone-2.31

Dyanova@emeraldiva ~/ns-allinone-2.31
$

```

Gambar 2.22 Perintah untuk pindah ke direktori ns-allinone-2.31.

8. Lakukan instalasi Ns-2 dengan perintah `$./install` hingga selesai (seperti gambar 2.23).

```

X ~/ns-allinone-2.31
Main Options  VT Options  VT Fonts

otcl:      /home/Dyanova/ns-allinone-2.31/otcl-1.13
tclcl:     /home/Dyanova/ns-allinone-2.31/tclcl-1.19
ns:        /home/Dyanova/ns-allinone-2.31/ns-2.31/ns
nam:       /home/Dyanova/ns-allinone-2.31/nam-1.13/nam
xgraph:    /home/Dyanova/ns-allinone-2.31/xgraph-12.1
gt-itm:    /home/Dyanova/ns-allinone-2.31/itm, edriver, sgb2alt, sgb2ns, sgb2comn
s, sgb2hierns

--

Please put /home/Dyanova/ns-allinone-2.31/bin:/home/Dyanova/ns-allinone-2.31/tcl
8.4.14/unix:/home/Dyanova/ns-allinone-2.31/tk8.4.14/unix
into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /home/Dyanova/ns-allinone-2.31/otcl-1.13, /home/Dyanova/ns-alli
none-2.31/lib,
into your LD_LIBRARY_PATH environment variable.
If it complains about X libraries, add path to your X libraries
into LD_LIBRARY_PATH.
If you are using csh, you can set it like:
    setenv LD_LIBRARY_PATH <paths>
If you are using sh, you can set it like:
    export LD_LIBRARY_PATH=<paths>

(2) You MUST put /home/Dyanova/ns-allinone-2.31/tcl8.4.14/library into your TCL_
LIBRARY environmental
variable. Otherwise ns/nam will complain during startup.

After these steps, you can now run the ns validation suite with
cd ns-2.31; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list ar
chive
for related posts.

Dyanova@emeraldiva ~/ns-allinone-2.31
$

```

Gambar 2.23 Instalasi ns-allinone-2.31 di Cygwin telah selesai

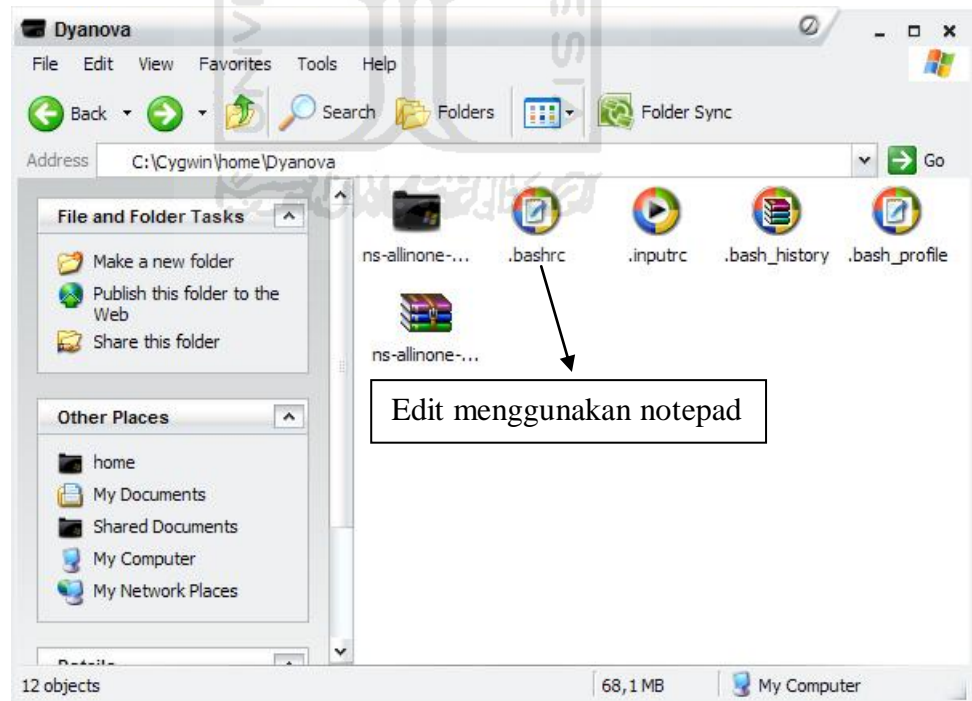
9. Setelah proses instalasi selesai, seperti halnya instalasi Ns-2.34 di Ubuntu 10.04, beberapa variabel system berikut harus ditambahkan di dalam file bashrc dengan menggunakan program notepad.

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/Dyanova/ns-allinone-2.31/otcl-1.13
NS2_LIB=/home/Dyanova/ns-allinone-2.31/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB

# TCL_LIBRARY
TCL_LIB=/home/Dyanova/ns-allinone-2.31/tcl8.4.14/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
XGRAPH=/home/Dyanova/ns-allinone-2.31/bin:/home/Dyanova/ns-allinone-2.31/tcl8.4.14/unix:/home/Dyanova/ns-allinone-2.31/tk8.4.14/unix
NS=/home/Dyanova/ns-allinone-2.31/Ns-2.31/
NAM=/home/Dyanova/ns-allinone-2.31/nam-1.13/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

File bashrc di direktori Cygwin dalam direktori pengguna, seperti pada Gambar 2.24 :



Gambar 2.24 File bashrc di direktori user.

10. Agar perubahan yang dilakukan langsung diterapkan oleh sistem, pada terminal ketikkan perintah `$ source ~/.bashrc`
11. Uji hasil instalasi Ns-2 dengan perintah `$ ns`
Jika muncul tanda %, berarti Ns-2.31 telah berjalan.
12. Pindah ke direktori Ns-2.31 dengan perintah `$ cd Ns-2.31/`
13. Lakukan validasi apakah Ns-2 bisa berjalan dengan baik, dengan perintah `$./validate`
Proses validasi Ns-2.31 pada Cygwin tidak jauh berbeda dari Ns-2.34 pada Ubuntu. Tunggu hingga proses validasi selesai untuk mengetahui apakah Ns-2.31 bisa berjalan dengan baik di Cygwin.



BAB III

DASAR-DASAR NS-2

3.1 Dasar Teori

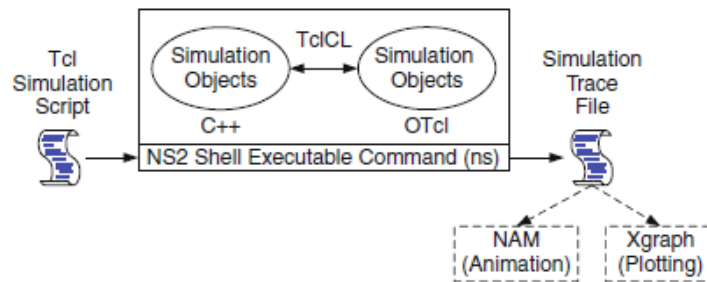
3.1.1 Network Simulator 2

Ns-2 adalah *open-source event simulator* yang dirancang khusus untuk penelitian dalam jaringan komputer. Sejak tahun 1989, NS2 terus mendapatkan perhatian yang luar biasa dari industri, akademisi, dan pemerintah. Setelah diselidiki dan dilakukan peningkatan terus-menerus selama bertahun-tahun, NS2 sekarang berisi modul untuk berbagai komponen jaringan, seperti *routing*, *transport layer protocol*, aplikasi, dll. Untuk meneliti kinerja jaringan, peneliti dapat menggunakan bahasa script untuk mengkonfigurasi jaringan dan mengamati hasil yang dikeluarkan oleh Ns-2.

Ns-2 merupakan simulator berorientasi objek yang ditulis dalam bahasa C++, dengan OTcl, lihat Gambar 3.1 (Issariyakul dan Hossain, 2009) sebagai interpreter. Keduanya saling berhubungan satu sama lain, berikut penjelasannya :

1. C++

C++ adalah bahasa pemrograman terkompilasi untuk meningkatkan efisiensi pada simulasi dan mempersingkat waktu eksekusi. C++ digunakan untuk pendefinisian agen, protokol dan framework, manipulasi *bytes data*, implementasi algoritma, pemrosesan paket serta manajemen waktu pemrosesan paket dan *event*. *Event* merupakan sebuah objek pada hirarki C++ yang memiliki identitas (ID) unik, jadwal waktu, dan pointer menuju objek yang menangani event.



Gambar 3.1 Arsitektur Dasar Ns-2

2. OTcl (object oriented extension of Tcl)

OTcl adalah bahasa pemrograman terinterpretasi untuk mengeksekusi perintah scripts dari pengguna. OTcl berfungsi untuk mendefinisikan topologi jaringan, skenario, aplikasi yang akan disimulasikan, dan bentuk keluaran hasil simulasi.

Alasan penggunaan dua bahasa pemrograman pada Ns-2 adalah karena Ns-2 memiliki banyak implementasi aplikasi yang berbeda. Pertama, karena Ns-2 membutuhkan program yang dapat memanipulasi bit, mengimplementasikan *packet header* dan melakukan algoritma dengan set data yang besar. Kedua, Ns-2 membutuhkan program dengan parameter yang bervariasi atau mampu menjalankan skenario dengan jaringan yang luas.

Program yang dibuat menggunakan Ns-2 biasanya disimpan dengan ekstensi .tcl, yang mengimplementasikan kode Tcl. Setelah program dijalankan, program tersebut akan menghasilkan file dengan ekstensi .tr. File ini dinamakan *trace file* yang memberikan informasi mengenai hasil simulasi. Semua keadaan saat simulasi tersebut dapat dilihat pada file ini.

Hasil simulasi dapat pula dilihat pada file dengan ekstensi .nam. File ini dinamakan *nam file (network animator file)*, yang merupakan paket dari ns-2. Fungsi file ini adalah sebagai penggambaran suatu skenario jaringan yang disimulasikan, seperti topologi, pengiriman paket dan variasi data yang dipakai dalam NAM.

NAM merupakan salah satu modul yang ada pada Ns-2 yang bekerja untuk menjalankan animasi proses simulasi yang dilakukan. File ini juga berisi informasi mengenai semua keadaan pada simulasi tersebut.

3.2 Elemen ns-2

Ns-2 dapat melakukan simulasi jaringan kabel maupun jaringan nirkabel, serta mendukung beberapa protokol seperti TCP, UDP, FTP, dan HTTP. Untuk bisa menggunakan ns-2, pengguna perlu mengetahui elemen-elemen dasar Ns-2. Elemen Ns-2 meliputi *event scheduler*, *network nodes*, *links*, *queues*, topologi, agen dan aplikasi, serta *tracing*. Berikut akan dibahas elemen-elemen dasar Ns-2 yang digunakan pada laporan ini.

3.2.1 Event Scheduler

Event Scheduler adalah perintah yang berfungsi untuk menentukan *event* yang akan terjadi pada simulasi. Misalnya untuk mengawali suatu simulasi ns, membuka trace file serta menjalankan ns. Semuanya membutuhkan event scheduler. Beberapa event scheduler yang lazim digunakan pada Ns-2 bisa dilihat pada Script 3.1.

```
#Membuat Simulasi baru
set ns [new Simulator]
#Membuka trace file
set f [open out.tr w]
$ns trace-all $f
#Membuka NAM trace file
setnf [open out.nam w]
$ns namtrace-all $nf
```

Script 3.1 Beberapa Event Scheduler.

3.2.2 Network Nodes

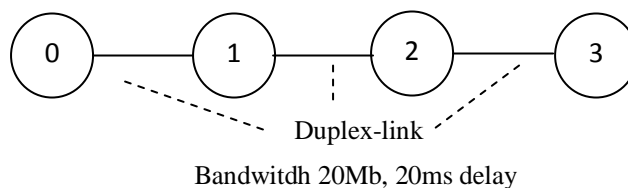
Network node menangani konfigurasi terhadap node seperti tipe node, posisi, komponen jaringan dan sebagainya. Node bisa berupa client, server, router atau perangkat lainnya. Script 3.2 adalah contoh pembuatan empat buah node (node n0, n1, n2 dan n3) dimana node yang saling berhubungan antara lain, node n0 dan n1, node n1 dan n2 serta node n2 dan n3. Masing-masing dihubungkan dengan koneksi duplex dengan bandwidth 2Mb dan *delay* 20 ms (seperti Gambar 3.2).

```
set n0 [$ns node]#pembuatan node
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n1 2Mb 20ms DropTail
$ns duplex-link $n1 $n2 2Mb 20ms DropTail
$ns duplex-link $n2 $n3 2Mb 20ms DropTail

$ns queue-limit $n1 $n2 10#batas tumpukan antrian
```

Script 3.2 Pembuatan Node



Gambar 3.2 Pembuatan Node

DropTail merupakan manajemen antrian yang diterapkan pada pembuatan node-node pada script 3.2. Ada beberapa queue management yang ada pada Ns-2 yaitu :

1. DropTail ini merupakan mekanisme untuk melakukan pengeluaran aliran data yang akan masuk dalam antrian ketika antrian sudah penuh.
2. Random Early Drop, yaitu mekanisme pengeluaran aliran data yang masuk duluan secara acak hingga antrian penuh.
3. Class Based Queueing, yaitu mekanisme antrian yang memungkinkan pembagian bandwidth secara rata setelah dilakukan pengelompokan berdasarkan parameter tertentu.
4. Fair Queueing, yaitu mekanisme antrian yang menggunakan algoritma penjadwalan yang memungkinkan banyak paket membagi kapasitas link secara adil.
5. Stochastic Fair Queuing, salah satu varian dari Fair Queueing yang menggunakan algoritma yang sama dengan fair queueing namun kapasitas yang tersedia telah ditentukan.

Queue-limit adalah perintah pada Ns-2 yang digunakan untuk membatasi maksimal paket yang akan masuk pada antrian antar node. Dalam hal ini penulis membatasi paket yang masuk dan mengalir antara node 1 (n1) dan node 2 (n2) sebanyak 10 antrian.

3.2.3 Agen dan Aplikasi

Setelah mendefinisikan topologi jaringan (node dan link), kemudian dibuat *traffic flow* pada topologi tersebut. *Traffic flow* adalah urutan paket yang dikirimkan dari *source* menuju host tujuan. Untuk itu perlu didefinisikan agen dan aplikasi yang menggunakannya. Agen merupakan routing protocol yang ditetapkan pada protokol dan aplikasi merupakan fungsi yang berjalan pada protokol.

Pada protokol TCP dapat dibuat aplikasi FTP dan Telnet, sedangkan pada protocol UDP dapat dibuat aplikasi CBR (*Constant Bit Rate*) dan *Exponential*. Untuk membuat koneksi TCP, gunakan Script 3.3.

```
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
```

Script 3.3 Membuat koneksi TCP

Berikut adalah penjelasan mengenai script 3.3

1. `set tcp0 [new Agent/TCP]`

Mendeklarasikan agen, dalam script 3.3 adalah TCP. Agen yang dibuat sesuai dengan protokol apa yang akan kita terapkan pada simulasi.

2. `$ns attach-agent $n0 $tcp0`
`$ns attach-agent node agen`

Menentukan node tempat agen yang telah dideklarasikan berada. Dalam script 3.3, agen tcp0 diletakkan pada node n0.

3. `set sink [new Agent/TCPSink]`

Mendeklarasikan agen, dalam hal ini adalah TCPSink. Agen yang akan bertindak sebagai tujuan/penerima.

4. `$ns attach-agent $n3 $sink`
`$ns attach-agent node agent`

Menentukan node tempat agen yang telah dideklarasikan berada, agen sink diletakkan pada node n3, sehingga node n3 akan menjadi node tujuan.

5. `$ns connect $tcp0 $sink`

`$ns connect agen agen`

Menghubungkan agen dengan agen lainnya. Dalam hal ini agen tcp0 dikoneksikan pada agen sink, dengan kata lain node no dan n3 akan terkoneksi dengan menjalankan protokol TCP.

Script 3.4 untuk membuat traffic TCP

```
set ftp [new Application/FTP] #Menggunakan FTP
set ftp [new Application/Telnet] #Atau menggunakan Telnet
$ftp attach-agent $tcp0 #Attach ke agen
```

Script 3.4 Contoh traffic pada TCP

Penjelasan script 3.4 :

1. `set ftp [new Application/FTP]`

Mendeklarasikan traffic yang akan dibuat, dalam hal ini adalah FTP (File Transfer Protocol).

2. `$ftp attach-agent $tcp0`

Parameter yang diterapkan pada traffic ftp, pada script ini traffic ftp dijalankan pada agen tcp0.

Untuk membuat koneksi UDP, gunakan Script 3.5 :

```
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0

set cbr0 [new Application/Traffic/CBR]
$cbr0 set interval_ 0.01
$cbr0 set packetSize_ 280
$cbr0 attach-agent $udp0
```

Script 3.5 Membuat koneksi UDP

Penjelasan dari script 3.5 :

1. `set udp0 [new Agent/UDP]`

Mendeklarasikan agen dalam script ini adalah UDP. Agen yang dibuat sesuai dengan protokol apa yang akan kita terapkan pada simulasi.

2. `$ns attach-agent $n0 $udp0`
`$ns attach-agent node agen`

Menentukan node tempat agen yang telah dideklarasikan berada. Dalam hal ini agen udp0 diletakkan pada node n0.

3. `set cbr0 [new Application/Traffic/CBR]`

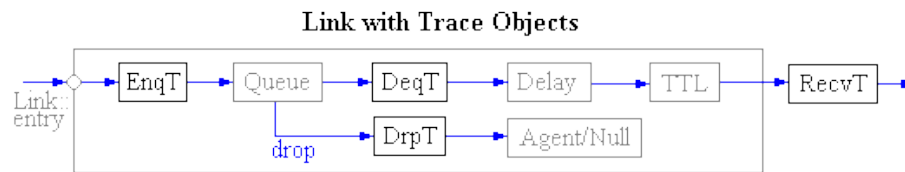
Mendeklarasikan traffic yang akan dibuat, dalam hal ini adalah CBR (Constant Bit Rate).

4. `$cbr0 set interval_ 0.01`
`$cbr0 set packetSize_ 280`
`$cbr0 attach-agent $udp0`

Merupakan parameter yang diterapkan pada traffic cbr0. Yaitu memiliki ukuran paket 280 Kb, dengan interval 0.01 detik dan dijalankan pada agen udp0.

3.2.4 Tracing

Ns-2 dapat membuat jejak berupa visualisasi dan ASCII *trace file* yang merekam semua event yang terjadi pada jaringan saat simulasi dilakukan. Saat tracing digunakan, ns-2 akan memasukkan empat objek yang saling berhubungan, yaitu EnqT, DeqT, RecvT, dan DrpT, yang ditunjukkan pada gambar 3.3 (Chung dan Claypool, 2003).



Gambar 3.3 Inserting Trace Object

EnqT mencatat informasi mengenai paket yang datang dan antrian yang masuk menuju link. Jika terjadi paket *overflow*, maka informasi yang dibuang tersebut akan dicatat oleh DrpT. DeqT mencatat informasi setelah paket keluar dari antrian. RecvT mencatat informasi mengenai paket yang telah diterima pada output link.

Saat tracing ke dalam ASCII trace file, rekaman diorganisasikan menjadi 12 field (seperti Gambar 3.4) yaitu :

Type Identifier	Time	Source Node	Destination Node	Packet Name	Packet Size	Flags	Flow ID	Source Address	Destination Address	Sequence Number	Packet Unique ID
-----------------	------	-------------	------------------	-------------	-------------	-------	---------	----------------	---------------------	-----------------	------------------

Gambar 3.4 Format trace file.

- Field pertama berisi tipe event.
- Field kedua berisi waktu event terjadi.
- Field ketiga berisi input node dari link di mana event terjadi.
- Field keempat berisi output node link di mana event terjadi.
- Field kelima berisi tipe paket.
- Field keenam berisi ukuran paket.
- Field ketujuh berisi *flags*.
- Field kedelapan berisi *flowid* (fid).
- Field kesembilan berisi alamat sumber dalam bentuk *node.port*.
- Field kesepuluh berisi alamat tujuan.
- Field kesebelas berisi *sequence number* paket pada *protocol network layer*.
- Field terakhir berisi id unik paket.

3.3 Visualisasi dengan Nam

Perangkat Nam (network animator) memberikan fitur-fitur untuk mengatur simulasi dalam bentuk visual yang mudah bagi pengguna.

Fitur-fitur tersebut antara lain:

1. Penempatan dan pemberian warna node.
2. Penentuan bentuk node. Bentuk node default adalah bulat, namun dapat ditentukan berupa kotak dan heksagonal.
3. Penambahan label.
4. Pemantauan ukuran queue.

Contoh script berdasarkan fitur Nam (seperti Script 3.8), yaitu :

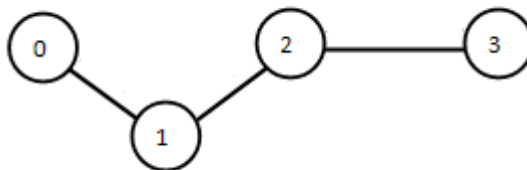
```
#Penempatan node
$ns duplex-link-op $n0 $n1 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
```

Script 3.6 Penempatan Node untuk Nam.

Penjelasan dari script diatas yaitu,

- Posisi node 0 dan 1 berorientasi ke kanan bawah.
- Posisi node 1 dan 2 berorientasi ke kanan atas.
- Posisi node 2 dan 3 berorientasi ke kanan.

Untuk lebih jelas lihat Gambar 3.5



Gambar 3.5 Visualisasi dengan Nam.

BAB IV

CONTOH DAN HASIL SIMULASI

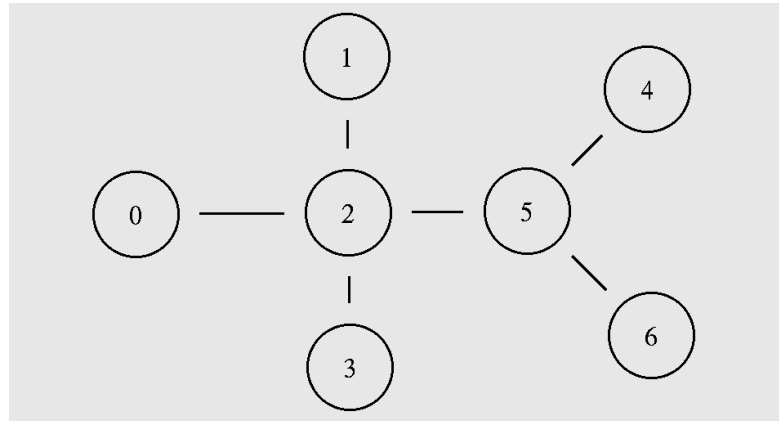
4.1 Pembuatan Simulasi

Simulasi yang akan dibuat adalah sebuah simulasi pengiriman paket data dari node pengirim menuju ke node penerima. Beberapa paket inilah yang akan diukur parameternya seperti jenis paket, throughput dan paket loss. Untuk membuat sebuah simulasi kita harus melakukan beberapa tahapan, agar simulasi yang dibuat sesuai dengan yang kita inginkan.

Model jaringan yang akan dibuat adalah jaringan sederhana dengan 7 node (node 0-node 6). Ukuran antrian antara node 2 dan node 5 adalah terbatas untuk 12 paket, dan semua link lainnya memiliki keterbatasan ukuran antrian 50. Terdapat lalu lintas FTP pada koneksi TCP antara node 1 dan node 6. Dengan ukuran maksimum congestion windows adalah 40 paket dan ukuran paket 280 byte. FTP dimulai pada 0,0 detik dan berakhir pada 15 detik.

Kemudian terdapat lalu lintas CBR pada koneksi UDP antara node 0 dan 6. Layanan CBR menghasilkan dan mengirimkan paket dengan kecepatan 100 paket/detik. Sumber CBR dimulai pada 2,0 detik dan berakhir pada detik ke-7. Kemudian terdapat lalu lintas VBR dengan tingkat 600 kbps antara node 3 dan node 6. Ukuran setiap paket CBR dan VBR adalah 280 byte. VBR dimulai pada 7,0 detik dan berakhir pada 12 detik. Ukuran congestion window harus dipantau dan diplot dari waktu ke waktu. Bandwidth (bandwidth di sini didefinisikan sebagai jumlah byte yang diterima selama suatu interval waktu tertentu, katakanlah 0,5 detik) lalu lintas CBR dan VBR juga harus dipantau dan diplot selama seluruh waktu simulasi. Parameter yang diukur yaitu : jenis paket, throughput dan paket loss.

Berikut ini adalah gambar dari simulasi yang akan dibuat, yaitu :



Gambar 4.1 Contoh Simulasi.

Topologi diatas merupakan jaringan dengan 7 node. Setiap pasang node memiliki koneksi duplex dengan spesifikasi sebagai berikut :

- Node 0 dan node 2 memiliki bandwidth 3 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit 50.
- Node 1 dan 2 memiliki bandwidth 5 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit 50.
- Node 2 dan node 3 memiliki bandwidth 3 Mbps dengan delay 1 ms dengan queue menggunakan SFQ, queue limit 50.
- Node 2 dan node 5 memiliki bandwidth 1 Mbps dengan delay 5 ms dengan queue menggunakan Drop Tail, queue limit 20.
- Node 4 dan node 5 memiliki bandwidth 4 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit 50.
- Node 5 dan node 6 memiliki bandwidth 3 Mbps dengan delay 1 ms dengan queue menggunakan SFQ, queue limit 50.
- Koneksi TCP dengan lalu lintas FTP yang berjalan di atasnya terjadi antara node 1 dan 6.
- Koneksi UDP dengan lalu lintas CBR berjalan di antara node 0 dan 6 dan lalu lintas VBR berjalan di antara node 3 dan 6.

Dari topologi dan parameter-parameter dibangunlah sebuah simulasi yang kita inginkan. Simulasi dibuat dengan menggunakan dasar-dasar yang telah ada pada bab sebelumnya. Script simulasi ini disimpan dengan nama tcpudp.tcl, adapun script lengkap dari simulasi ini bisa dilihat pada bagian lampiran.



Adapun flow chart dari proses simulasi yang berlangsung yaitu :

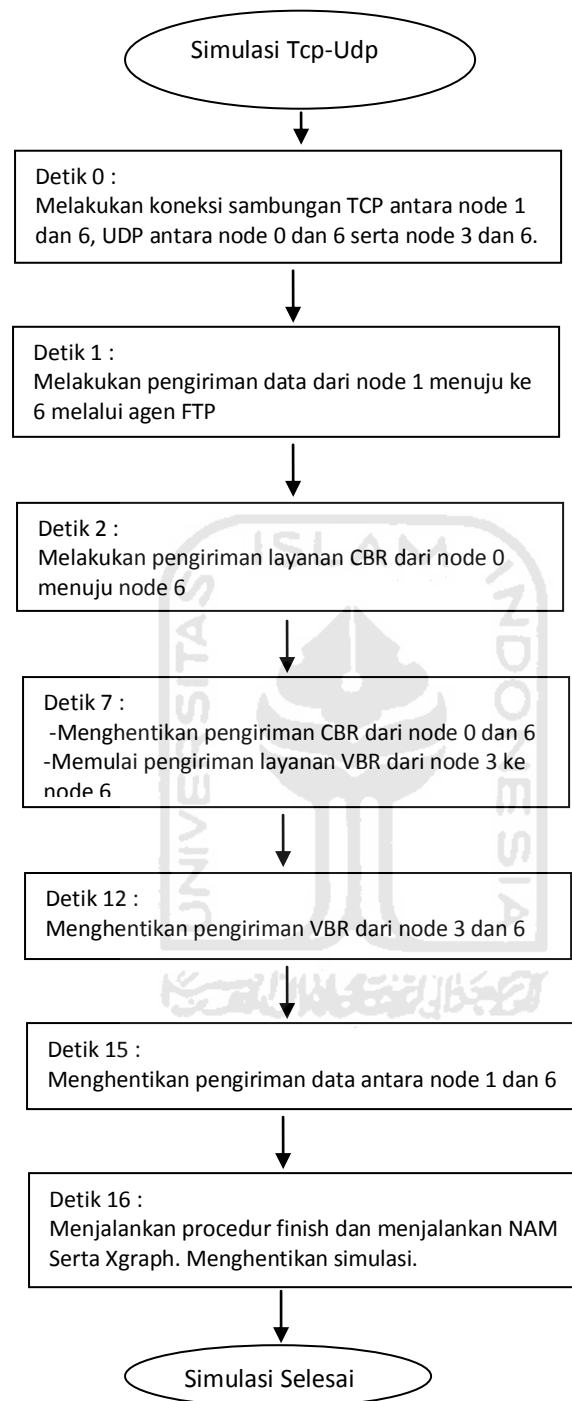
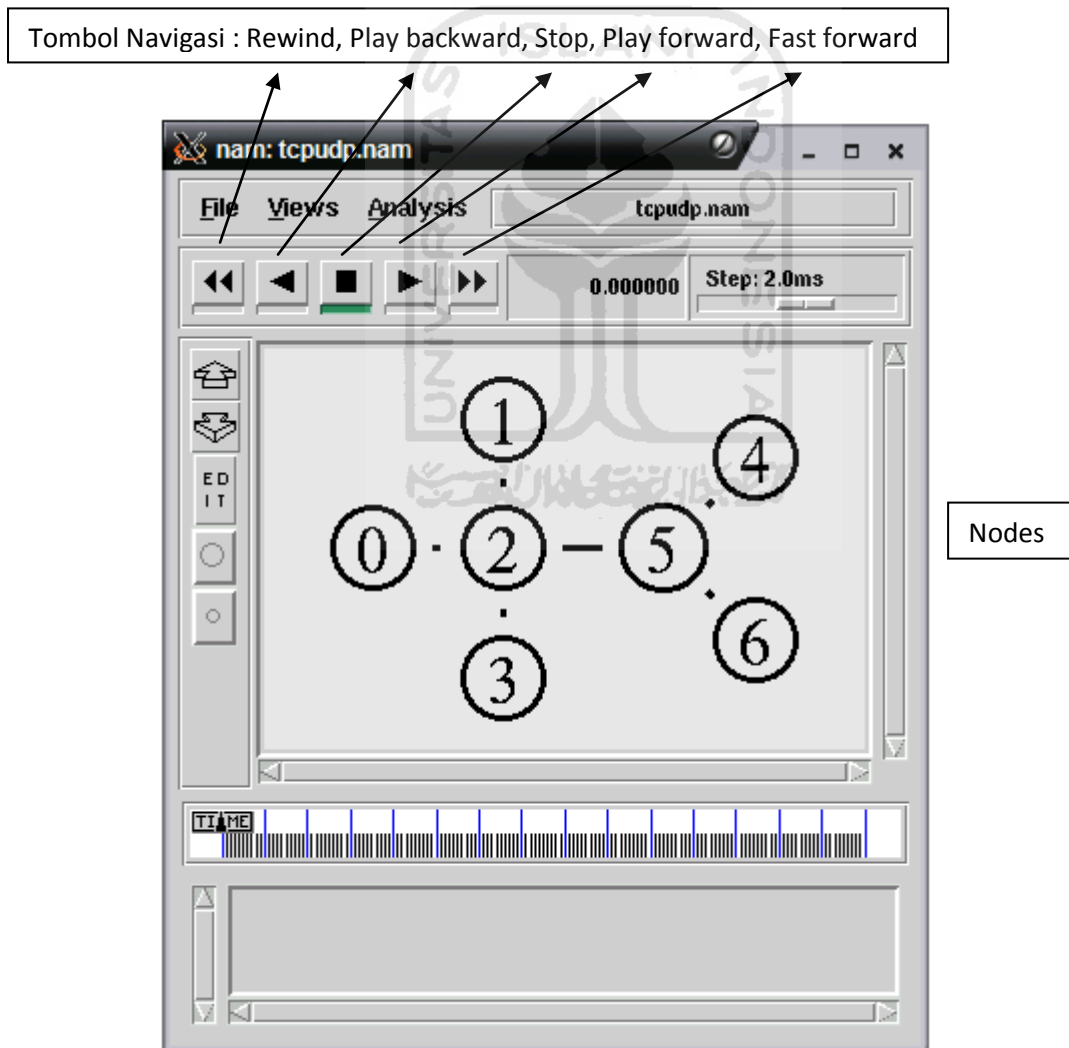


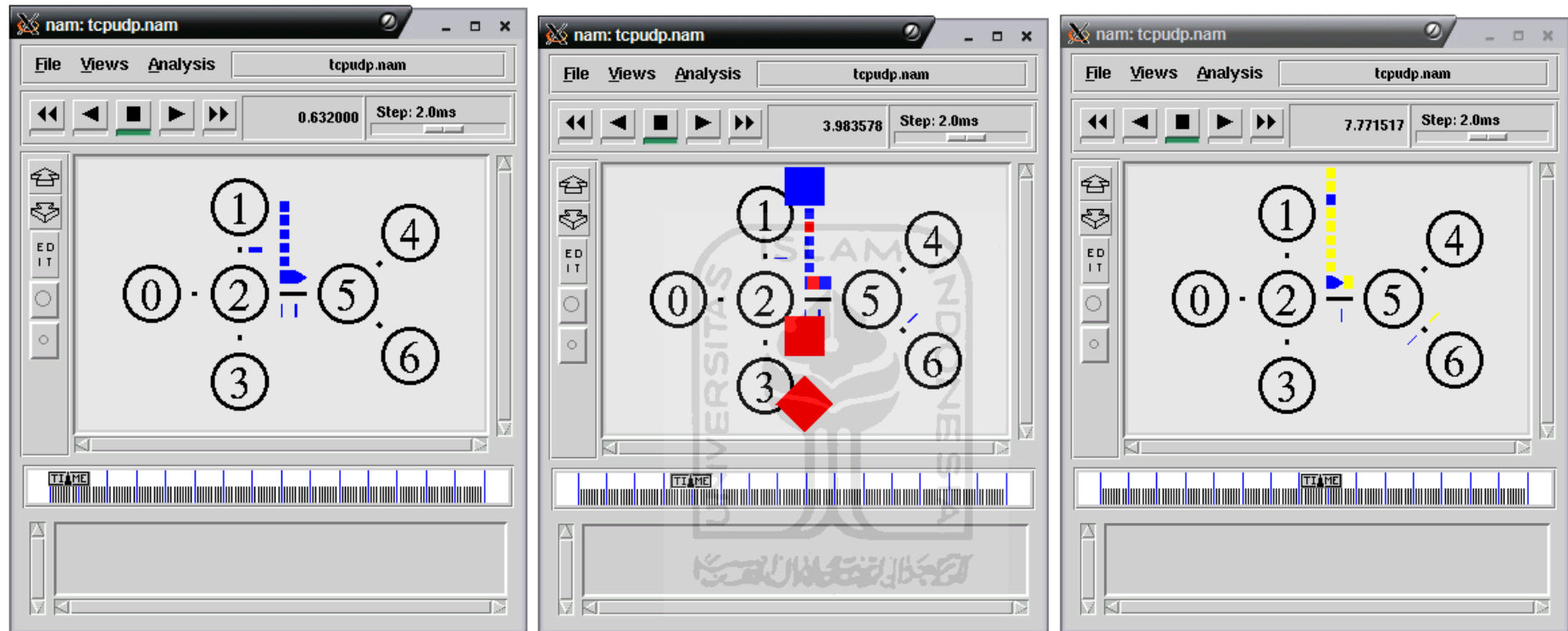
Diagram 4.1 Flow chart simulasi.

4.2 Hasil Simulasi

Untuk menjalankan simulasi, panggil nama file script Tcl yang dibuat, dengan perintah `$ ns namafile.tcl`. File tersebut akan dieksekusi oleh Ns-2 dan menghasilkan *trace file* `tcpudp.tr`, `CBR.tr`, `VBR.tr` dan *nam file* `tcpudp.nam`. File-file inilah yang akan dianalisa untuk mendapatkan parameter yang kita inginkan yaitu jenis paket, paket loss dan throughput. Untuk menjalankan Nam dapat menggunakan tombol navigasi yang ada (seperti pada Gambar 4.2).



Gambar 4.2 Network Animator.



Gambar 4.3 Simulasi pada Nam.

Berikut adalah isi dari file tcpudp.nam :

```
+ -t 0 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
null}
- -t 0 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
null}
h -t 0 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 -1 -----
null}
r -t 0.001064 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
-- null}
+ -t 0.001064 -s 2 -d 5 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
-- null}
- -t 0.001064 -s 2 -d 5 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
-- null}
h -t 0.001064 -s 2 -d 5 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 -1 -----
-- null}
r -t 0.006384 -s 2 -d 5 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
-- null}
+ -t 0.006384 -s 5 -d 6 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
-- null}
- -t 0.006384 -s 5 -d 6 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
-- null}
h -t 0.006384 -s 5 -d 6 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 -1 -----
-- null}
r -t 0.00749066666666667 -s 5 -d 6 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0
6.0 0 ----- null}
+ -t 0.00749066666666667 -s 6 -d 5 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
- -t 0.00749066666666667 -s 6 -d 5 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
h -t 0.00749066666666667 -s 6 -d 5 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 -1 ----- null}
r -t 0.00859733333333333 -s 6 -d 5 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
+ -t 0.00859733333333333 -s 5 -d 2 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
- -t 0.00859733333333333 -s 5 -d 2 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
h -t 0.00859733333333333 -s 5 -d 2 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 -1 ----- null}
r -t 0.01391733333333333 -s 5 -d 2 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
+ -t 0.01391733333333333 -s 2 -d 1 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
- -t 0.01391733333333333 -s 2 -d 1 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
h -t 0.01391733333333333 -s 2 -d 1 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 -1 ----- null}
r -t 0.01498133333333333 -s 2 -d 1 -p ack -e 40 -c 2 -i 1 -a 2 -x {6.0
1.0 0 ----- null}
```

Script 4.1 Isi file tcpudp.nam.

Kemudian hasil dari file tcpudp.tr yaitu :

```
0 1
0.01 1
0.02 2
0.029999999999999999 2
0.040000000000000001 4
0.050000000000000003 4
0.060000000000000005 8
0.070000000000000007 8
0.080000000000000002 12
0.089999999999999997 16
0.099999999999999992 20
0.10999999999999999 24
0.11999999999999998 28
0.12999999999999998 32
0.13999999999999999 36
0.14999999999999999 40
0.16 40.0999
0.170000000000000001 40.1747
0.180000000000000002 40.2741
0.190000000000000003 40.3238
0.200000000000000004 20
0.210000000000000005 20
0.220000000000000006 20
0.230000000000000007 20
0.240000000000000007 20
0.250000000000000006 20
0.260000000000000006 20.05
0.270000000000000007 20.05
0.280000000000000008 20.05
0.290000000000000009 20.05
0.300000000000000001 20.05
0.3100000000000000011 20.05
0.3200000000000000012 20.05
0.3300000000000000013 20.05
0.3400000000000000014 20.05
0.3500000000000000014 20.05
0.3600000000000000015 20.05
0.3700000000000000016 20.05
0.3800000000000000017 20.05
0.3900000000000000018 20.05
0.4000000000000000019 20.05
0.410000000000000002 20.05
0.4200000000000000021 20.05
0.4300000000000000022 20.05
0.4400000000000000022 20.05
0.4500000000000000023 20.05
0.4600000000000000024 1
0.4700000000000000025 1
```

Script 4.2 Isi file tcpudp.tr.

Hasil dari CBR.tr yaitu :

```
2.100000000000000005 0.08959999999999999
2.150000000000000004 0.08959999999999999
2.200000000000000002 0.08959999999999999
2.25 0.26879999999999998
2.2999999999999998 0.224
2.3499999999999996 0.1792
2.3999999999999995 0.1792
2.4499999999999993 0.13439999999999999
2.4999999999999991 0.1792
2.5499999999999989 0.1792
2.5999999999999988 0.13439999999999999
2.6499999999999986 0.13439999999999999
2.6999999999999984 0.1792
2.7499999999999982 0.13439999999999999
2.799999999999998 0.1792
2.8499999999999979 0.13439999999999999
2.8999999999999977 0.1792
2.9499999999999975 0.13439999999999999
2.9999999999999973 0.1792
3.0499999999999972 0.13439999999999999
3.099999999999997 0.1792
3.1499999999999968 0.1792
3.1999999999999966 0.13439999999999999
3.2499999999999964 0.1792
3.2999999999999963 0.1792
3.3499999999999961 0.26879999999999998
3.3999999999999959 0.13439999999999999
3.4499999999999957 0.1792
3.4999999999999956 0.13439999999999999
3.5499999999999954 0.1792
3.5999999999999952 0.13439999999999999
3.649999999999995 0.1792
3.6999999999999948 0.1792
3.7499999999999947 0.13439999999999999
3.7999999999999945 0.13439999999999999
3.8499999999999943 0.1792
3.8999999999999941 0.1792
3.949999999999994 0.1792
3.9999999999999938 0.224
4.0499999999999936 0.1792
4.0999999999999934 0.1792
4.1499999999999932 0.13439999999999999
4.1999999999999931 0.1792
4.2499999999999929 0.13439999999999999
4.2999999999999927 0.1792
4.3499999999999925 0.13439999999999999
4.3999999999999924 0.1792
4.4499999999999922 0.13439999999999999
```

Script 4.3 Isi file CBR.tr.

Dan hasil dari VBR.tr yaitu :

```
7.349999999999819 0.1343999999999999
7.399999999999817 0.58240000000000003
7.449999999999815 0.6271999999999998
7.499999999999813 0.98560000000000003
7.549999999999812 0.6271999999999998
7.59999999999981 0.58240000000000003
7.649999999999808 0.5375999999999997
7.699999999999806 0.44800000000000001
7.749999999999805 0.58240000000000003
7.799999999999803 0.5375999999999997
7.849999999999801 0.5375999999999997
7.899999999999799 0.58240000000000003
7.949999999999797 0.4032
7.999999999999796 0.49280000000000002
8.049999999999794 0.49280000000000002
8.099999999999801 0.49280000000000002
8.149999999999808 0.6271999999999998
8.199999999999815 0.6271999999999998
8.249999999999822 0.6271999999999998
8.299999999999829 0.2687999999999998
8.349999999999837 0.4032
8.399999999999844 0.44800000000000001
8.449999999999851 0.5375999999999997
8.499999999999858 0.49280000000000002
8.549999999999865 0.44800000000000001
8.599999999999872 0.7167999999999999
8.649999999999879 0.76160000000000005
8.699999999999886 0.6271999999999998
8.749999999999893 0.0
8.799999999999901 0.0
8.849999999999908 0.0
8.899999999999915 0.0
8.949999999999922 0.0
8.999999999999929 0.0
9.049999999999936 0.3135999999999999
9.099999999999943 0.4032
9.14999999999995 0.49280000000000002
9.199999999999957 0.5375999999999997
9.249999999999964 0.5375999999999997
9.299999999999972 0.7167999999999999
9.349999999999979 0.08959999999999999
9.399999999999986 0.0
9.449999999999993 0.0
9.5 0.0448
9.5500000000000007 0.58240000000000003
9.6000000000000014 0.1792
9.6500000000000021 0.0
9.7000000000000028 0.2687999999999998
```

Script 4.4 Isi file VBR.tr.

Adapun penjelasan dari masing-masing hasil dari simulasi yang telah kita jalankan yaitu :

File tcpudp.nam merupakan file yang akan digunakan untuk mengetahui aliran data antara node-node yang ada, file inilah yang akan kita gunakan untuk mendapatkan jenis data yang mengalir, paket loss dan throughput.

Berikut ini adalah baris awal dari isi file tcpudp.nam :

```
+ -t 0 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
null}
- -t 0 -ns 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0 -----
- null}
h -t 0 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 -1 -----
- null}
r -t 0.001064 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 6.0 0
----- null}
```

Maksud dari tanda “+” itu berarti ada paket yang ditambahkan pada antrian, tanda “-“ berarti paket yang dikeluarkan dari antrian, “h” berarti ada file yg dirutekan, “r” berarti ada paket yang diterima dan “d” berarti ada paket yg drop. Dengan demikian baris pertama bisa dibaca sebagai berikut :

Paket ditambahkan pada antrian, pada detik ke-0 dari *source* node 1 ke *destination* node 2 menggunakan protokol TCP dengan ukuran paket 40 bytes, flow id (-c) adalah 2, nomor paket (i) adalah 0.

Dengan menggunakan script awk yang telah kita modifikasi bisa kita bisa mendapatkan parameter kualitas layanan yang kita inginkan. Penulis menggunakan script kualitas.awk untuk mendapatkan beberapa parameter yaitu jenis paket, paket loss dan throughput. Script lengkap dari kualitas.awk bisa dilihat pada bagian lampiran.


Untuk menjalankan script awk dengan perintah :

```
awk -f nama-script nama-file-yg-akan-diproses
```

Dalam hal ini script awk dipanggil dengan perintah :

```
Awk -f kualitas.awk tcpudp.nam
```

Dan hasil yang akan didapatkan bisa dilihat pada gambar 4.4 :



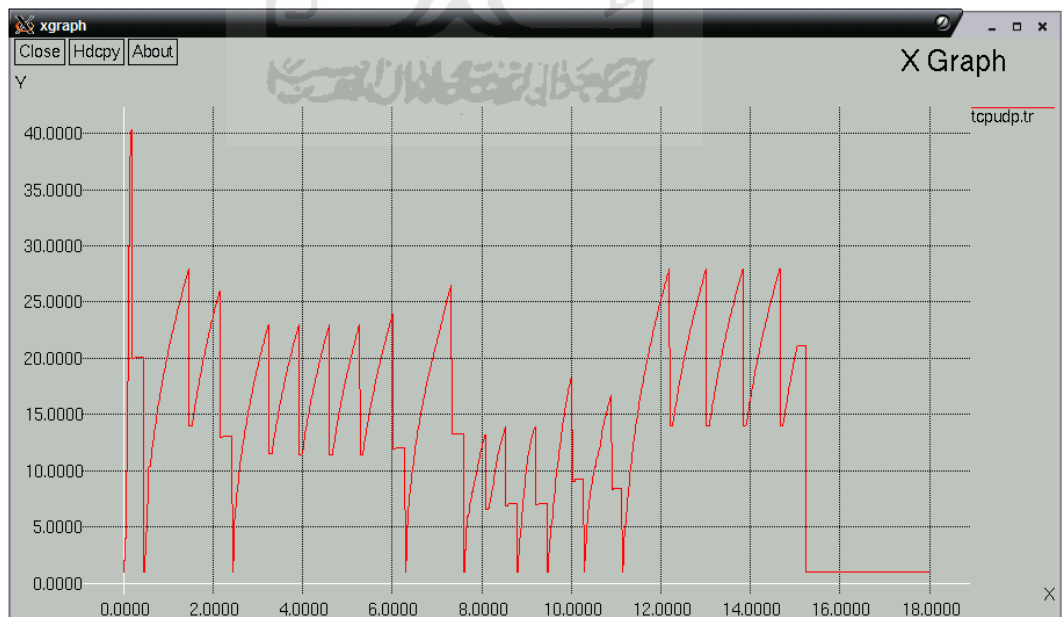
```
Main Options VT Options VT Fonts
Dyanova@emeraldiva ~
$ awk -f kualitas.awk tcpudp.tr

TCP mengirimkan 4347 paket data namun hanya 4303 yang diterima node tujuan (paket loss = 44), dengan throughput : 686,377 Kbps
TCP telah mengirimkan 4303 ack dan 4303 yang terkirim.
CBR mengirimkan 375 paket data namun hanya 369 yang diterima node tujuan (paket loss = 6), dengan throughput : 160,855 Kbps
VBR mengirimkan 836 paket data dan 793 yang terkirim (paket loss = 43) dengan throughput: 366,4 Kbps
System throughput = 854,466 Kbps

Dyanova@emeraldiva ~
$
```

Gambar 4.4 Hasil kualitas.awk.

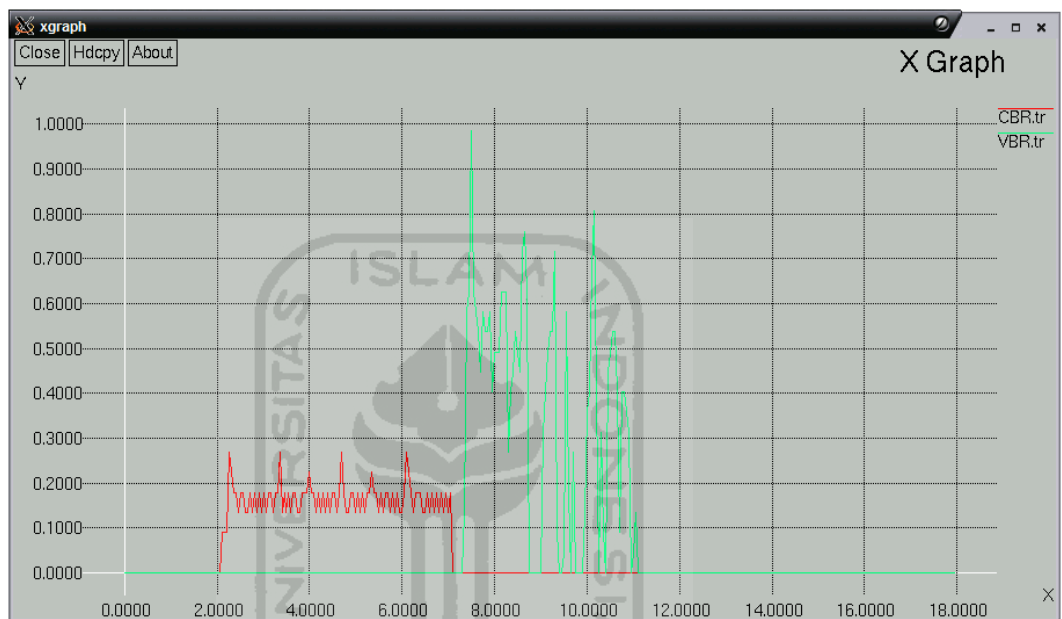
File tcpudp.tr merupakan hasil dari plotting congestion window dari waktu ke waktu yang terjadi pada protokol TCP dengan traffic FTP dan maksimum congestion windows adalah 40. Apabila dijalankan dengan xgraph akan menghasilkan grafik seperti ini :



Gambar 4.5 Hasil grafik tcpudp.tr dengan X Graph.

Kemudian file CBR.tr dan VBR.tr adalah hasil plotting dari bandwidth yang diterima oleh masing-masing traffic. Dengan perhitungan bandwidth yg diterima dalam Mbps.

Kemudian hasil dari CBR.tr dan VBR.tr dalam bentuk grafik adalah sebagai berikut :



Gambar 4.6 Hasil grafik CBR.tr dan VBR.tr dengan X Graph.

4.3 Analisis Simulasi

Dari simulasi yang telah dibuat bisa kita berikan skenario tertentu untuk mendapatkan hasil yang berbeda-beda, sehingga kita bisa melakukan analisis terhadap hasil dari simulasi.

1. Skenario 1

Kita akan membuat simulasi dengan perubahan bandwidth setiap node menjadi lebih besar queue limit dan delay tetap.

- Node 0 dan node 2 memiliki bandwidth 3 Mbps dinaikkan menjadi 9 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit 50.
- Node 1 dan 2 memiliki bandwidth 5 Mbps dinaikkan menjadi 15 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit 50.
- Node 2 dan node 3 memiliki bandwidth 3 Mbps dinaikkan menjadi 9 Mbps dengan delay 1 ms dengan queue menggunakan SFQ, queue limit 50.
- Node 2 dan node 5 memiliki bandwidth 1 Mbps dinaikkan menjadi 3 Mbps dengan delay 5 ms dengan queue menggunakan Drop Tail, queue limit 20.
- Node 4 dan node 5 memiliki bandwidth 4 Mbps dinaikkan menjadi 12 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit 50.

2. Skenario 2

Kita akan membuat simulasi dengan perubahan delay setiap node menjadi lebih besar, bandwidth dan queue limit tetap.

- Node 0 dan node 2 memiliki bandwidth 3 Mbps dengan delay dari 1 ms menjadi 100 ms dengan queue menggunakan Drop Tail, queue limit 50.

- Node 1 dan 2 memiliki bandwidth 5 Mbps dengan delay dari 1 ms menjadi 100 ms dengan queue menggunakan Drop Tail, queue limit 50.
- Node 2 dan node 3 memiliki bandwidth 3 Mbps dengan delay dari 1 ms menjadi 100 ms dengan queue menggunakan SFQ, queue limit 50.
- Node 2 dan node 5 memiliki bandwidth 1 Mbps dengan delay dari 5 ms menjadi 500 ms dengan queue menggunakan Drop Tail, queue limit 20.
- Node 4 dan node 5 memiliki bandwidth 4 Mbps dengan delay dari 1 ms menjadi 100 ms dengan queue menggunakan Drop Tail, queue limit 50.
- Node 5 dan node 6 memiliki bandwidth 3 Mbps dengan delay dari 1 ms menjadi 100 ms dengan queue menggunakan SFQ, queue limit 50.

3. Skenario 3

Kita akan membuat queue limit menjadi lebih kecil , bandwidth dan delay tetap dari setiap node.

- Node 0 dan node 2 memiliki bandwidth 3 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit menjadi 25.
- Node 1 dan 2 memiliki bandwidth 5 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit menjadi 25.
- Node 2 dan node 3 memiliki bandwidth 3 Mbps dengan delay 1 ms dengan queue menggunakan SFQ, queue limit 25.
- Node 2 dan node 5 memiliki bandwidth 1 Mbps dengan delay 5 ms dengan queue menggunakan Drop Tail, queue limit menjadi 10.
- Node 4 dan node 5 memiliki bandwidth 4 Mbps dengan delay 1 ms dengan queue menggunakan Drop Tail, queue limit menjadi 25.
- Node 5 dan node 6 memiliki bandwidth 3 Mbps dengan delay 1 ms dengan queue menggunakan SFQ, queue limit menjadi 25.

Dari ketiga skenario diatas kita mendapatkan hasil sebagai berikut :

Skenario 1

Parameter	Yang terkirim	Yang diterima	Paket loss	Throughput (Kbps)
TCP	14107	14070	37	2248,29
CBR	375	353	22	153,992
VBR	836	819	17	375,609
System	-	-	-	2418,1

Tabel 4.1 Hasil simulasi skenario 1.

Skenario 2

Parameter	Yang terkirim	Yang diterima	Paket loss	Throughput (Kbps)
TCP	131	125	6	19,3644
CBR	375	375	0	163,636
VBR	836	836	0	382,606
System	-	-	-	190,737

Tabel 4.2 Hasil simulasi skenario 2.

Skenario 3

Parameter	Yang terkirim	Yang diterima	Paket loss	Throughput (Kbps)
TCP	4382	4304	78	687,456
CBR	375	375	0	163,776
VBR	836	793	43	363,526
System	-	-	-	856,64

Tabel 4.3 Hasil simulasi skenario 3.

Simulasi Awal

Parameter	Yang terkirim	Yang diterima	Paket loss	Throughput (Kbps)
TCP	4347	4303	44	686,377
CBR	375	369	6	160,855
VBR	836	793	43	366,4
System	-	-	-	854,466

Tabel 4.4 Hasil simulasi skenario awal.

Dari perbandingan hasil dari simulasi awal dan skenario-skenario yang ada dapat diberikan kesimpulan sebagai berikut :

Pada Protokol TCP :

1. Semakin besar bandwidth maka paket yang akan dikirimkan akan memiliki throughput yang lebih besar dan memperkecil kemungkinan packet loss.
2. Semakin besar delay maka semakin kecil jumlah paket yang dikirimkan dan throughput yang dihasilkan namun packet loss semakin besar.
3. Semakin kecil batas antrian yang diberikan maka akan semakin tinggi packet loss yang dihasilkan.

Pada protokol UDP :

CBR

1. Semakin besar bandwidth tidak terlalu mempengaruhi CBR yang dijalankan karena memiliki bit rate tetap yang dikirimkan.
2. Besar atau kecilnya delay maka akan mempengaruhi jumlah packet loss pada CBR. Semakin besar delay maka kemungkinan packet loss akan berkurang.
3. Semakin kecil antrian yang diberikan tidak mempengaruhi CBR karena memiliki bit rate yang tetap.

VBR

1. Semakin besar bandwidth akan mempengaruhi jumlah paket yang dikirimkan dan mengurangi packet loss yang terjadi.
2. Besar atau kecilnya delay maka akan mempengaruhi jumlah packet loss pada VBR. Semakin besar delay maka kemungkinan packet loss akan berkurang.
3. VBR yang memiliki bit rate yang berubah-ubah, semakin kecil antrian maka semakin banyak packet loss yang dihasilkan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diambil dalam Tugas Akhir ini adalah sebagai berikut :

1. Simulator Ns-2 layak untuk dipelajari mendalam, terutama untuk membuat jaringan yang lebih kompleks.
2. Simulasi sederhana memiliki banyak keterbatasan terhadap pengukuran kualitas layanan.
3. Dalam sebuah simulasi semua parameter sangat berpengaruh terhadap hasil yang dilakukan, seperti bandwidth, delay dan queue limit.

5.2 Saran

Saran untuk pengembangan Tugas Akhir ini adalah sebagai berikut :

1. Penggunaan Ns-2 untuk bahan tambahan perkuliahan mahasiswa Teknik Informatika.
2. Penelitian dan pengembangan lebih lanjut mengenai jaringan wireless pada simulator Ns-2.
3. Penggunaan modul tambahan untuk kebutuhan simulasi jaringan yang lebih kompleks.
4. Penelitian lebih lanjut mengenai emulator yang ada pada Ns-2 agar bisa berkomunikasi langsung dengan jaringan baik kabel maupun nirkabel.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Simulator jaringan adalah suatu perangkat lunak atau keras untuk melakukan simulasi, prediksi terhadap proses-proses yang terjadi dalam suatu jaringan komputer, tanpa menggunakan sistem jaringan komputer yang sesungguhnya.

Ada banyak simulator jaringan yang tersedia saat ini antara lain Ns-2 (Network Simulator 2), OPNET IT Guru dan NetSim. Berikut adalah beberapa penjelasan mengenai ketiga simulator jaringan tersebut :

1. Ns-2 adalah *open-source event simulator* yang dirancang khusus untuk penelitian dalam jaringan komputer, merupakan salah satu simulator yang populer dan digunakan untuk kebutuhan penelitian yang dipublikasikan dalam bentuk *paper*, Tugas Akhir maupun *thesis*.
2. OPNET IT Guru (Academic) merupakan sebuah simulator yang dikembangkan berdasarkan OPNET IT Guru 9.1 (Commercial) untuk kepentingan akademis dengan fitur yang disesuaikan berdasarkan pada tingkatan pengenalan dasar jaringan komputer.
3. NetSim adalah sebuah piranti yang komprehensif untuk mempelajari jaringan komputer yang mengajarkan seni dari simulasi jaringan komputer yang memberikan kemudahan dan mempercepat pembelajaran mahasiswa.

Ns-2 memiliki beberapa kelebihan jika dibandingkan simulator jaringan yang lainnya, yaitu :

1. Ns-2 merupakan simulator yang bersifat open source sehingga kita bisa melakukan perubahan terhadap framework Ns-2 sesuai dengan kebutuhan kita.

2. Mudah untuk dikonfigurasi karena menggunakan dua bahasa pemrograman yaitu C++ dan OTcl.
3. Ns-2 memiliki komunitas yang solid, sebagian besar merupakan peneliti dalam jaringan komputer. Dukungan dari para peneliti ini mengakibatkan perkembangan Ns-2 menjadi dinamis.
4. Memiliki banyak penambahan modul yang telah dikembangkan oleh komunitas untuk Ns-2 agar sesuai dengan perkembangan jaringan komputer di dunia.

Selain memiliki kelebihan dibandingkan simulator lainnya, Ns-2 telah menjadi simulator jaringan open source yang paling banyak digunakan dan merupakan salah satu simulator jaringan yang paling banyak digunakan (Issariyakul dan Hossain, 2009).

Beberapa contoh penggunaan Ns-2 untuk keperluan penelitian, pendidikan maupun thesis antara lain adalah :

1. Pengembangan Ns-2 Berbasis Virtual Network Laboratory (Hendi S, 2009). Penelitian ini merupakan pembangunan sistem laboratorium jaringan komputer virtual dengan menggunakan Ns-2 sebagai tool simulasi dan animasinya.
2. Implementation and Performance Analysis of Star-Based Mesh Network, (Muhammad Haq, 2011). Penelitian ini bertujuan untuk mendesain sebuah star-based mesh network yang menggunakan banyak hub/switch untuk meningkatkan kinerja transaksi data, yang biasanya memburuk pada jaringan Mesh konvensional.
3. Implementation and Comparison of Mobility Models in Ns-2 (Sukumar dan Rahul, 2009). Penelitian ini bertujuan mengimplementasi dan membandingkan beberapa model jaringan nirkabel bergerak pada Ns-2.

Ns-2 juga digunakan untuk mengembangkan aplikasi, salah satunya adalah The Active Queue Management and Denial of Service (AQM&DoS). Merupakan platform simulasi yang dibangun berdasarkan Ns-2, untuk melakukan simulasi beberapa skenario yang berhubungan dengan algoritma Distributed Denial-of-Service (DDoS) attacks dan Active Queue Management (AQM).

Platform ini bisa mensimulasikan beberapa serangan Denial-of-Service, seperti Denial-of-Service (DoS) attacks, Distributed Denial-of-Service (DDoS) attacks, Spoofing DoS atau DDoS attacks dan beberapa jenis Algoritma Active Queue Management (AQM).

Karena kelebihan serta berbagai penerapan Ns-2 inilah penulis akan membuat sebuah panduan mengenai Ns-2 yang mencakup dasar-dasar Ns-2, instalasi Ns-2 serta pembuatan simulasi sederhana. Simulasi ini dibuat berdasarkan dasar-dasar Ns-2 yang telah dibahas dalam panduan ini.

1.2 Rumusan Masalah

Jurusan Teknik Informatika UII belum memperkenalkan dan memanfaatkan Ns-2 sebagai perangkat simulasi jaringan karena kurangnya panduan instalasi dan penggunaan Ns-2, adapun rumusan masalah dari Tugas Akhir ini yaitu :

- a. Bagaimana membuat panduan yang mudah, khususnya untuk mahasiswa ?
- b. Bagaimana dasar-dasar Ns-2 ?

1.3 Tujuan

Tujuan dari penugasan ini antara lain :

- a. Membantu mahasiswa dan dosen untuk memahami dan dapat menggunakan simulator ns-2.
- b. Memberikan pemahaman lebih mendalam mengenai Ns-2.

1.4 Batasan Masalah

Batasan masalah untuk penugasan ini adalah :

1. Terbatas pada dasar-dasar Ns-2
2. Tidak memperhitungkan aspek keamanan dalam jaringan, seperti otentikasi dan otorisasi.
3. Terbatas pada simulasi *Local Area Network*.
4. Terbatas pada kemampuan simulator Ns-2 tanpa adanya penambahan modul.

1.5 Metodologi Penelitian

Dalam metodologi penelitian ini ada beberapa tahapan-tahapan yang akan dilakukan dalam melaksanakan Tugas Akhir yaitu :

1. Kajian literatur
Pada tahap ini akan dilakukan studi literatur mengenai Ns-2, perintah-perintah dasar, format penulisan serta instalasinya.
2. Dasar-dasar Ns-2 dan simulasi
Pada tahap ini akan dilakukan pembelajaran dasar-dasar Ns-2 dan perancangan simulasi berdasarkan dasar-dasar yang telah dipelajari.
3. Analisis hasil perancangan dan simulasi
Pada tahap ini akan dilakukan proses analisis terhadap hasil perancangan dan simulasi yang ada.
4. Penarikan kesimpulan dan saran
Pada tahap ini akan ditarik kesimpulan dari hasil penugasan dan saran untuk pengembangan pembelajaran Ns-2.

1.6 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini adalah sebagai berikut:

1. BAB I Pendahuluan, berisi Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan, Metodologi Penulisan dan Sistematika Penulisan.
2. BAB II Instalasi Network Simulator 2, berisi tahapan instalasi Ns-2.
3. BAB III Dasar-dasar Ns-2, berisi dasar-dasar Ns-2 untuk keperluan simulasi pada Bab IV.
4. BAB IV Contoh dan hasil simulasi, merupakan contoh simulasi hasil implementasi dasar-dasar Ns-2.
5. BAB V Kesimpulan dan Saran.



DAFTAR PUSTAKA

NS2 Manual. URL :http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

Gordon, David, Haddad, Ibrahim F."Network Simulator 2: a Simulation tool for Linux".URL :<http://www.linuxjournal.com/article/5929.2002>

Hossain, Ekram & Issariyakul, Teearawat. 2009. Introduction to Network Simulator NS2. New York : Springer Science+Business Media, LLC.

Haq, Muhammad.2011. Implementation and Performance Analysis of Star-Based Mesh Network. Thesis. Tidak diterbitkan. University Höskolan.

Panda, Sukumar & Mohapatra, Rahul. 2009. Implementation and Comparison of Mobility Models in Ns-2.Thesis.Tidak diterbitkan. National Institute of Technology Rourkela.

S, Hendi. 2009. Pengembangan Network Simulator 2 Berbasis Virtual Network Laboratory. Skripsi. Tidak Diterbitkan.Institut Teknologi Bandung.

Chung, Jae., Claypool, Mark., Kinicki, Robert. 2005.MTP: A Streaming-Friendly Transport Protocol. CS Department, Worcester Polytechnic Institute, Tech.

LAMPIRAN

1. Script tcpudp.tcl

```
set ns [new Simulator]

$ns color 1 Red
$ns color 2 Blue
$ns color 3 Yellow

set f0 [open CBR.tr w]
set f1 [open VBR.tr w]
set f3 [open tcpudp.tr w]

set nf [open tcpudp.nam w]
$ns namtrace-all $nf

proc finish {} {
    global ns nf f0 f1 f3
    $ns flush-trace
    close $nf
    close $f0
    close $f1
    close $f3

    exec xgraph CBR.tr VBR.tr -geometry 900x500 &
    exec xgraph tcpudp.tr -geometry 900x500 &

    exec nam tcpudp.nam &
    exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]

$ns duplex-link $n0 $n2 3Mb 1ms DropTail
$ns duplex-link $n1 $n2 5Mb 1ms DropTail
$ns duplex-link $n3 $n2 3Mb 1ms SFQ
$ns duplex-link $n2 $n5 1Mb 5ms DropTail
$ns duplex-link $n5 $n4 4Mb 1ms DropTail
$ns duplex-link $n5 $n6 3Mb 1ms SFQ
$ns queue-limit $n1 $n2 50;
$ns queue-limit $n2 $n5 20;
$ns queue-limit $n0 $n2 50;
$ns queue-limit $n3 $n2 50;
$ns queue-limit $n5 $n4 50;
$ns queue-limit $n5 $n6 50;

$ns duplex-link-op $n0 $n2 orient right
$ns duplex-link-op $n1 $n2 orient down
```

```
$ns duplex-link-op $n3 $n2 orient up
$ns duplex-link-op $n2 $n5 orient right
$ns duplex-link-op $n5 $n4 orient right-up
$ns duplex-link-op $n5 $n6 orient right-down
```

```
$ns duplex-link-op $n2 $n5 queuePos 0.5
$ns duplex-link-op $n1 $n2 queuePos 0.5
$ns duplex-link-op $n0 $n2 queuePos 0.5
$ns duplex-link-op $n5 $n6 queuePos 0.5
$ns duplex-link-op $n3 $n2 queuePos 0.5
```

```
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0
```

```
set cbr0 [new Application/Traffic/CBR]
$cbr0 set interval_ 0.01
$cbr0 set packetSize_ 280
$cbr0 attach-agent $udp0
```

```
set tcp1 [new Agent/TCP/Reno]
$tcp1 set fid_ 2
$tcp1 set window_ 40
$tcp1 set packetSize_ 280
$tcp1 set minrto_ 0.2
set sink3 [new Agent/TCPSink]
$ns attach-agent $n1 $tcp1
$ns attach-agent $n6 $sink3
$ns connect $tcp1 $sink3
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
```

```
set udp1 [new Agent/UDP]
$udp1 set class_ 3
$ns attach-agent $n3 $udp1
```

```
set vbr1 [new Application/Traffic/Exponential]
$vbr1 set packetSize_ 280
$vbr1 set rate_ 600k
$vbr1 set burst_time_ 150ms
$vbr1 set idle_time_ 100ms
$vbr1 attach-agent $udp1
proc plotWindow {tcpsource} {
    global ns f3
    set time 0.01
    set now [$ns now]
    set cwnd [$tcpsource set cwnd_]
    puts $f3 "$now $cwnd"
    $ns at [expr $now+$time] "plotWindow $tcpsource"
}
```

```
proc record {} {
    global sink0 sink1 f0 f1
```

```
set ns [Simulator instance]
```

```
set time 0.05

set bw0 [$sink0 set bytes_]
set bw1 [$sink1 set bytes_]

set now [$ns now]

puts $f0 "$now [expr $bw0/$time*8/1000000]"
puts $f1 "$now [expr $bw1/$time*8/1000000]"

$sink0 set bytes_ 0
$sink1 set bytes_ 0

$ns at [expr $now+$time] "record"
}

set sink0 [new Agent/LossMonitor]
$ns attach-agent $n6 $sink0
set sink1 [new Agent/LossMonitor]
$ns attach-agent $n6 $sink1

$ns connect $udp0 $sink0
$ns connect $udp1 $sink1

$ns at 0.0 "record"
$ns at 0.0 "plotWindow $tcp1"

$ns at 2.0 "$cbr0 start"
$ns at 0.0 "$ftp1 start"
$ns at 7.0 "$vbr1 start"

$ns at 7.0 "$cbr0 stop"
$ns at 15.0 "$ftp1 stop"
$ns at 12.0 "$vbr1 stop"
$ns at 16.0 "finish"

$ns run
```


2. Script kualitas.awk

```
BEGIN { recv=0
tcpreceiveddata=0
tcpatdata=0
tcpack=0
acksize=0
tcpreceivedack=0
cbrreceived=0
cbrat=0
cbrdata=0
vbrreceived=0
vbrdata=0
cbrsimtime=0
vbrsimtime=0
}
{
if ($1 == "r" && $9 == "tcp" && $7==6) {
tcpreceiveddata+=1
pkt_size = $11
recv += pkt_size-20
simtime=$3
}
if ($1 == "r" && $9 == "tcp" && $7==2) {
tcpatdata+=1
}
if ($1 == "r" && $9 == "ack" && $7==5) {
tcpack+=1
}
if ($1 == "r" && $9 == "ack" && $7==1) {
tcpreceivedack+=1
acksize+=$11
}
if ($1 == "r" && $9 == "cbr" && $7==6) {
cbrreceived+=1
cbrdata+=$11-8
if (cbrsimtime==0){cbrstarttime=$3
}
cbrsimtime=$3
}
if ($1 == "r" && $9 == "cbr" && $7==2) {
cbrat+=1
}
if ($1 == "r" && $9 == "exp" && $7==6) {
vbrreceived+=1
vbrdata+=$11-8
if (vbrsimtime==0){vbrstarttime=$3
}
vbrsimtime=$3
}
if ($1 == "r" && $9 == "exp" && $7==2) {
vbrat+=1
}
}
END { printf ("\n")
#printf("TCP simulation ends at %g\n", simtime)
```

```

printf("TCP mengirimkan %g paket data namun hanya %g yang
diterima node tujuan (paket loss = %g), dengan throughput : %g
Kbps \n", tcpatdata, tcpreceiveddata, (tcpatdata-
tcpreceiveddata), (recv/simtime)*(8/1000))
printf("TCP telah mengirimkan %g ack dan %g yang terkirim. \n",
tcpack, tcpreceivedack)
#printf("Actual time for CBR simulation to end: %g\n",
cbrsimtime)
printf("CBR mengirimkan %g paket data namun hanya %g yang
diterima node tujuan (paket loss = %g), dengan throughput : %g
Kbps \n", cbrat, cbrreceived, (cbrat-cbrreceived),
(cbrdata/(cbrsimtime-cbrstarttime))*(8/1000))
#printf("Actual time for VBR simulation end: %g start:%g \n",
vbrsimtime, vbrstarttime)
printf("VBR mengirimkan %g paket data dan %g yang terkirim
(paket loss = %g) dengan throughput: %g Kbps\n", vbrat,
vbrreceived, (vbrat-vbrreceived), (vbrdata/(vbrsimtime-
vbrstarttime))*(8/1000))
printf("System throughput = %g Kbps\n",
((recv+cbrdata+vbrdata)/simtime)*(8/1000))
printf ("\n")
}

```

