BAB 3
METODOLOGI PENELITIAN

3.1 ANALISA KEBUTUHAN

Tahap analisa kebutuhan digunakan untuk menentukan kebutuhan apa saja yang diperlukan untuk menunjang penelitian. Penelitian ini membutuhkan dua jenis perangkat, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Pemilihan perangkat keras dan perangkat lunak didalam penelitian ini diambil berdasarkan penelitian (studi) literatur sebelumnya.

3.1.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan untuk membangun simulasi pada penelitian ini berupa satu unit laptop dengan spesifikasi yang ditunjukkan pada Table 3.1. Di dalam laptop ini kemudian di *install software* VMWare dan *Network Simulator* 2 untuk keperluan simulasi jaringan MANET.

Tabel 3.1 Spesifikasi Perangkat Keras (*Hardware*) yang Digunakan

Processor	Intel® *Core*™ i7-4712MQ CPU @ 2.30 GHz
Memori (RAM)	8 GB
Harddisk	500 GB
VGA	Intel HD 4600 dan NVIDIA *GeForce* GT 740M

3.1.2 Perangkat Lunak (*Software*)

Perangkat lunak sebagai aplikasi yang digunakan untuk membangun simulasi dan pengujian pada penelitian ini ditunjukkan pada Tabel 3.2.

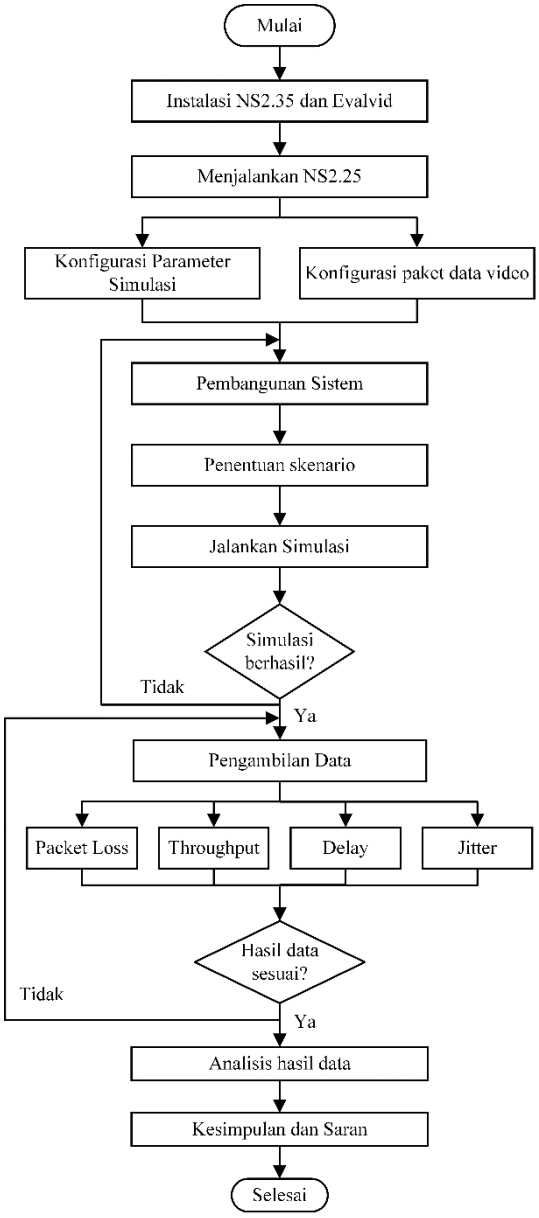
Tabel 3.2 Perangkat Lunak (*Software*) yang Digunakan

Sistem Operasi	Ubuntu 14.04 LTS 64 bit
Virtual Machine	*VMWare Workstation* 16 *Player*
C/C++ *Compiler*	gcc-4.4
Simulator Tools	*Network Simulator* 2.35 dan EvalVid
Analysis Tools	*Microsoft Excel*

3.2 PEMODELAN SISTEM

3.2.1 Alur Penelitian

Pada tahap alur penelitian ini akan mendeskripsikan keseluruhan proses kegiatan selama proses penelitian. Alur penelitian dapat dilihat pada Gambar 3.1 berikut.

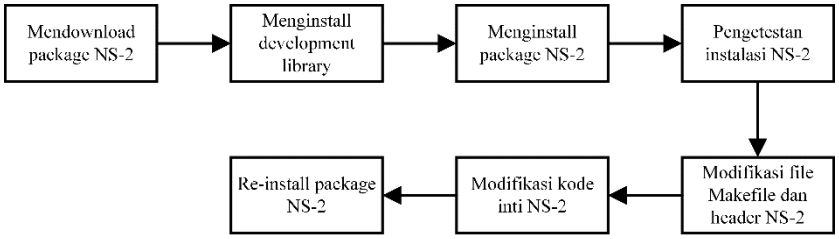


Gambar 3.1 Alur Penelitian

Pada tahap ini, mula-mula simulasi dilakukan dengan instalasi *software Network Simulator* 2 versi 2.35 dan Evalvid, lalu mengkonfigurasi parameter simulasi dan paket data video yang digunakan. Setelah dilakukan konfigurasi, *script* simulasi (sistem) dibangun kedalam script utama guna untuk menjalankan simulasi

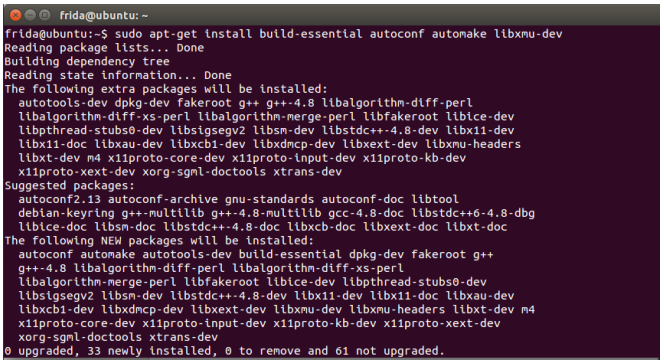
sesuai dengan skenario simulasi yang telah dirancang sebelumnya menggunakan NS2.35. Jika simulasi berhasil, maka tahap berikutnya yaitu pengambilan data. Namun jika hasil simulasi gagal, maka akan dilakukan membangun sistem simulasi ulang hingga hasilnya sesuai.

3.2.1.1 Instalasi NS2.35 dan EvalVid

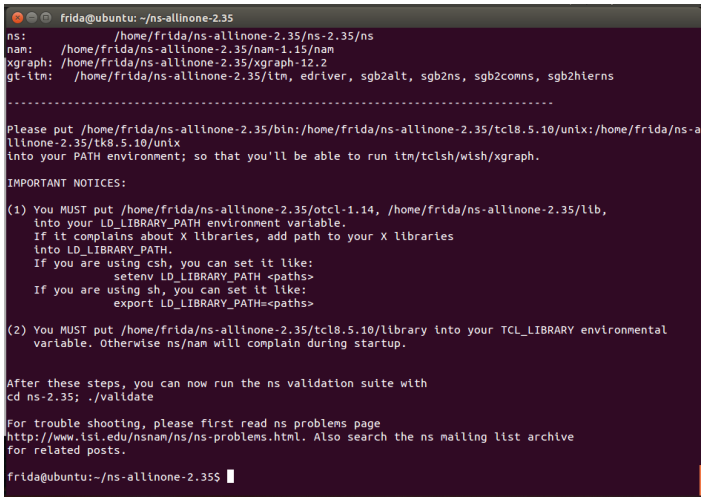


Gambar 3.2 Alur Instalasi NS2.35 dan Evalvid

Program *Network Simulator* 2 versi 2.35 diinstal dan dijalankan secara virtual pada sistem operasi Linux Ubuntu 14.04 LTS 64bit menggunakan mesin virtual *VMWare Workstation* 16 *Player*. Sebelum menjalankan NS2.35, terlebih dahulu melakukan instalasi NS2.35 dengan men*download package*nya di *website* Source Forge dan mengekstrak *file*nya pada *folder* /*home*. Beberapa *development library* diperlukan juga untuk *requirements package* menginstal NS2.35 yang ditunjukkan pada Gambar 3.3.

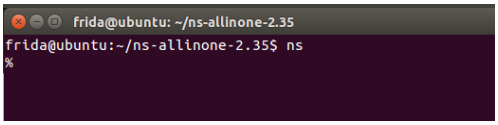


Gambar 3.3 Instalasi *development library*



Gambar 3.4 Instalasi NS2.35

Pada Gambar 3.4 menunjukkan proses instalasi NS2.35 telah selesai. NS2.35 dapat di *test* dengan memasukkan perintah ns jika berhasil terinstal maka akan muncul tanda % seperti pada Gambar 3.5.



Gambar 3.5 Pengetestan NS2.35 telah berhasil terinstal

Tools EvalVid diinstal setelah NS2.35 terinstal. Instalasi terintegrasi NS2 dan EvalVid memiliki beberapa perubahan yang dilakukan, yaitu:

1. Modifikasi yang diterapkan pada file *Makefile* dan *header* NS2 (ns-default.tcl);
2. Penambahan kelas C++ baru ke dalam kode inti NS2 (antara lain packet.h, agent.h, dan agent.cc). Kelas-kelas tersebut menambah NS2 *environment* dengan objek yang memiliki kemampuan untuk mengirimkan video *trace file* yang dihasilkan oleh *tool mp4trace*.

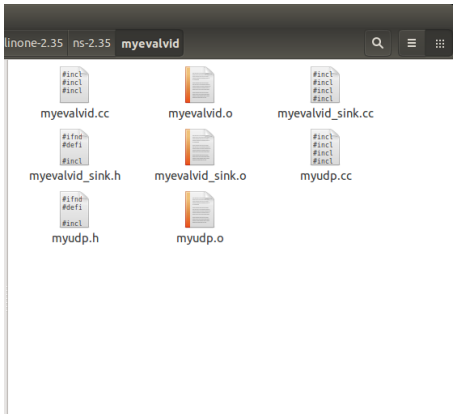
Seperti yang dijelaskan di BAB 2 pada Gambar 2.7, terdapat NS2 Evironment yang didalamnya memiliki tiga agen penghubung yaitu *MyTrafficTrace*, *MyUDP*, dan *MyUDPSink*. Fungsi dari masing-masing agen antara lain:

1. Agen *MyTrafficTrace* mengekstrak ukuran *frame* dan jenis *frame* dari *file trace* video yang dihasilkan dari *traffic trace file*, memecah *frame* video menjadi segmen yang lebih kecil, dan mengirimkan segmen kecil ke lapisan

UDP yang lebih rendah pada waktu yang tepat sesuai dengan pengaturan pengguna yang ditentukan dalam *file* skrip simulasi.

2. Agen *MyUDP* adalah ekstensi dari agen UDP. Agen baru ini memungkinkan pengguna untuk menentukan nama *output file* dari *sender trace file* dan mencatat *timestamp* setiap paket yang ditransmisikan, ID paket, dan ukuran muatan paket (*payload*). Fungsi agen *MyUDP* sesuai dengan fungsi *tool tcp-dump* atau *win-dump* pada lingkungan jaringan *real*.
3. Agen *MyUDPSink* adalah agen penerima untuk paket *frame* video terfragmentasi (terpecah-pecah) yang dikirim oleh agen *MyUDP*. Agen ini juga mencatat *timestamp*, ID paket, dan ukuran muatan (*payload*) dari setiap paket yang diterima dalam *receiver trace file* yang ditentukan pengguna.

Ketiga agen yang sudah berhasil ditambahkan kedalam NS2 terlihat seperti pada Gambar 3.6 dengan nama *file* myevalvid, myevalvid_sink, dan myudp.



Gambar 3.6 Tiga agen penghubung integrasi NS2 dan EvalVid

3.2.1.2 Konfigurasi Parameter Simulasi

Parameter simulasi pada penelitian ini merupakan parameter yang digunakan untuk membangun simulasi menggunakan protokol *routing* AODV dan DSR pada jaringan MANET. Tabel 3.3 adalah tabel parameter yang digunakan pada simulasi.

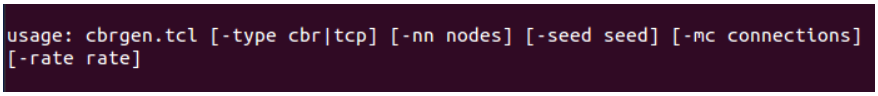
Tabel 3.3 Parameter Simulasi

No.	Parameter Simulasi	Nilai
1	Luas area simulasi	750 x 500 m
2	Max range transmisi	250 m

3	Standar *Wireless*	IEEE 802.11g
4	Pola koneksi	*Peer-to-peer*
5	Tipe *Traffic Application*	CBR (*Constant Bit rate*)
6	Jenis Mobilitas Node	*Random Waypoint*
7	Waktu simulasi	30 s
8	Jumlah node	10, 20, 30
9	Kecepatan perpindahan node	5 m/s, 10 m/s, 15 m/s
10	Protokol *routing*	AODV, DSR

Simulasi dilakukan pada luas area 750 x 500 meter. *Maximum range* transmisi sejauh 250 m dengan menggunakan kanal *wireless* (IEEE 802.11g) dan pola koneksi *peer-to-peer*. Jumlah node yang akan simulasikan sebanyak 10, 20 dan 30 node dengan kecepatan perpindahan node 5 m/s, 10 m/s, dan 15 m/s pada protokol routing AODV dan DSR. Topologi jaringan MANET terdiri atas kumpulan *node* yang bersifat *mobile* sehingga topologinya tidak dapat diprediksi dan bersifat acak. Simulasi dilakukan selama 30 detik.

Konfigurasi *traffic* yaitu *traffic* CBR (*Constant Bit Rate*) dibuat menggunakan *traffic-connection* generator program bernama 'cbrgen.tcl' yang ada pada NS2 dan digunakan untuk membuat koneksi pada node yang sudah dibuat di skrip skenario selama melakukan simulasi pada NS2. Koneksi *traffic* pada 'cbrgen.tcl' memiliki pola koneksi *traffic* random dan dua tipe *traffic* yaitu TCP dan CBR. Format perintah/*syntax* untuk generator *traffic* 'cbrgen.tcl' beserta penjelasannya ditunjukkan pada Gambar 3.7 dan Tabel 3.4.



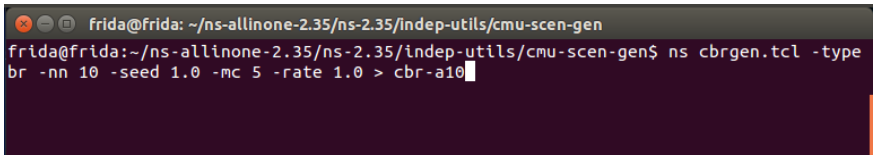
Gambar 3.7 Format perintah/*syntax* generator *traffic* 'cbrgen.tcl'

Tabel 3.4 Keterangan perintah/*syntax* 'cbrgen.tcl'

Parameter	**Keterangan**
-type <cbr|tcp>	Menentukan jenis *traffic* yang digunakan
-nn <nodes>	Menentukan jumlah node
-seed <seed>	Menentukan nilai random *seed*
-mc <max connections>	Menentukan jumlah koneksi antar node

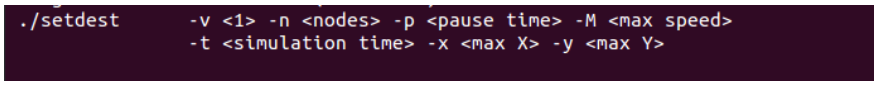
-rate <rate>	Menentukan jumlah paket per detik yang dikirim

Gambar 3.8 merupakan perintah/*syntax* yang digunakan untuk membuat *mobility* menggunakan generator 'setdest'. Pada *option type* menggunakan *traffic* cbr dengan jumlah 10 *node* dan *seed* 1.0, maksimal koneksi yaitu 5 koneksi antar *node* dengan *rate* 1.0. Diakhir *syntax*, file *traffic* akan disimpan dengan nama cbr-a10.



Gambar 3.8 Penggunaan perintah/*syntax* generator *traffic* 'cbrgen.tcl'

Konfigurasi *mobility* dimana *node*nya akan bergerak secara acak dengan jenis mobilitas *Random WayPoint*. Mobility dibuat menggunakan *mobility generator* program bernama 'setdest' yang ada pada NS2 yang digunakan untuk membuat pola pergerakan pada node untuk melakukan simulasi pada NS2. *Mobility* node ini memiliki pola pergerakan yang random. Format perintah/*syntax* untuk *generator mobility* 'setdest' beserta penjelasannya ditunjukkan pada Gambar 3.9 dan Tabel 3.5.

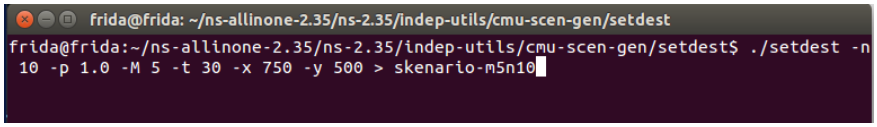


Gambar 3.9 Format perintah/*syntax generator mobility* 'setdest'

Tabel 3.5 Keterangan perintah/*syntax* 'setdest'

Parameter	Keterangan
-v <1>	Menujukkan versi format generator yang digunakan
-n <nodes>	Menentukan jumlah node yang digunakan
-p <pause time>	Menentukan durasi berhenti sebuah paket jika sudah sampai tujuan
-M <max speed>	Menentukan kecepatan maksimum pengiriman paket
-t <simulation time>	Menentukan lamanya waktu simulasi berjalan
-x <max X>	Menentukan panjang maksimum dari area simulasi
-y <max Y>	Menentukan lebar maksimum dari area simulasi

Gambar 3.10 merupakan perintah/*syntax* yang digunakan untuk membuat *mobility* menggunakan *generator* '*setdest'*. Pada *option* versi *syntax* ini menggunakan format generator versi 1. Jumlah *node* yang akan dibuat yaitu untuk 10 *node* dengan waktu *pause* 1.0 *second*, kecepatan maksimal *node* yaitu 5 m/s dengan waktu simulasi 30 detik pada luas area 750 x 500 m. Diakhir *syntax, file traffic* akan disimpan dengan nama skenario-m5n10.



Gambar 3.10 Perintah/*syntax* generator *mobility* 'setdest'

Topologi jaringan MANET pada penelitian ini terdiri dari beberapa node yang sudah ditetapkan, ditempatkan secara acak posisi. Pembuatan node secara acak menggunakan *syntax* pada Gambar 3.11 yang dimasukkan ke dalam *script* simulasi.

```
#Create Random Nodes
for {set i 0} {$i < $val(nn) } { incr i } {
        set node_($i) [$ns_ node]
        $ns_ initial_node_pos $node_($i) 35
}
```

Gambar 3.11 *Code*/*syntax* membuat *random node*

3.2.1.3 Konfigurasi Paket Data Video

Pada Tabel 3.6 dijelaskan bahwa simulasi dilakukan menggunakan *traffic* video H.264 berupa *file raw* video bernama akiyo_cif.yuv yang memiliki resolusi 352 x 288 *pixel* dengan jumlah *frame* 300 *frame* dan berdurasi 10 detik. *Traffic* data video memiliki format CIF (*Common Intermediate Format*) yang merupakan format standar resolusi foto/video yang menunjukan ukuran horizontal dan vertikal dalam besaran *pixel*. Ukuran file YUV yaitu 45.6 MB. *File raw* video dikirimkan melalui kompresi *encode* pada 30 fps (*frame per second*).

Tabel 3.6 Parameter *traffic* video

	Nama video/*traffic*	akiyo_cif.yuv
	Resolusi	352 x 288 *pixel (CIF)*
	Profile	H.264
	Ukuran *file* YUV	45.6 MB
	Paket *frame*	30 *frame*/s

	Durasi	10 detik
	Jumlah *frame*	300 *frame*

Traffic/file video akiyo adalah sebuah video dengan objek seorang Wanita pembaca berita. Video akiyo memiliki latar (*background*) video tidak terlihat ada objek lain yang bergerak, pergerakan hanya terjadi pada bibir dan kepala pembaca berita, sehingga pergerakan *frame* secara temporal dan spatial sangat tinggi.

3.2.1.4 Penentuan Skenario

Pada penelitian ini menggunakan dua macam skenario yaitu penambahan jumlah node dan perubahan kecepatan perpindahan node. Setiap skenario akan menggunakan parameter simulasi jaringan yang sama dan diakhir pengujian yang akan dilakukan menggunakan hasil perbandingan parameter QoS dari protokol AODV dan DSR. Hasil perbandingan parameter dari pengujian protokol routing AODV dan DSR yang diambil adalah *delay*, *jitter*, *throughput*, dan *packet loss*.

Skenario yang digunakan pada penelitian ini adalah penambahan jumlah node dan kecepatan perpindahan node. Skenario penelitian tersebut dimaksudkan untuk mengetahui seberapa jauh pengaruh penambahan jumlah node dan kecepatan perpindahan node terhadap performansi suatu protokol routing AODV dan DSR pada MANET menggunakan aplikasi video *streaming*. Simulasi dilakukan dengan luas area jaringan MANET 750 x 500 meter, jumlah node yang digunakan yaitu 10, 20 dan 30 node dan kecepatan perpindahan node yang digunakan yaitu 5 m/s, 10 m/s dan 15 m/s. Luas area dimaksudkan agar node dapat bergerak bebas dan tidak selalu berada dalam *coverage area* node lain. Adapun skenario penambahan jumlah node yang dilakukan ditunjukkan pada Tabel 3.7 dan skenario perubahan kecepatan perpindahan node ditunjukkan pada Tabel 3.8.

Tabel 3.7 Skenario penambahan jumlah node

Skenario	**Jumlah Node**	**Kecepatan Perpindahan Node**	**Protokol Routing**
1	10	5 m/s	AODV dan DSR
2	20	5 m/s	

3	30	5 m/s	

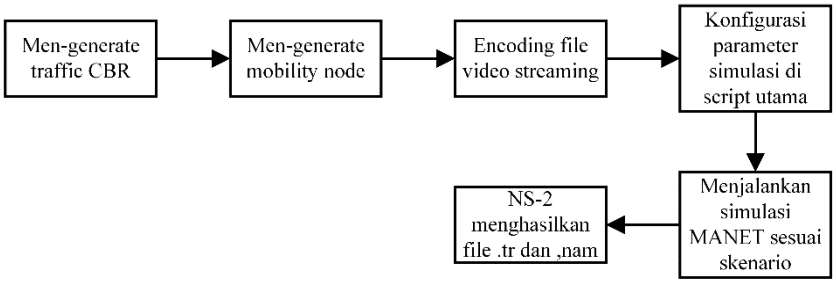
Skenario pertama yaitu skenario penambahan jumlah node pada luas area jaringan 750 x 500 m. Skenario satu menggunakan node sebanyak 10 node dengan kecepatan perpindahan node 5 m/s. Skenario dua menggunakan node sebanyak 20 node dengan kecepatan perpindahan node 5 m/s. Skenario tiga menggunakan node sebanyak 30 node dengan kecepatan perpindahan node 5 m/s. Masing-masing skenario diuji pada protokol routing AODV dan DSR dengan posisi node yang acak dan menggunakan mobilitas *Random WayPoint* dimana node akan bergerak secara acak ke titik tujuan dan kecepatan yang acak.

Tabel 3.8 Skenario perubahan kecepatan perpindahan node

Skenario	**Kecepatan Perpindahan Node**	**Jumlah Node**	**Protokol Routing**
1	5 m/s	10	AODV dan DSR
2	10 m/s	10	
3	15 m/s	10	

Skenario kedua yaitu skenario perubahan kecepatan perpindahan node pada luas area jaringan 750 x 500 m. Skenario satu menggunakan node sebanyak 10 node dengan kecepatan perpindahan node 5 m/s. Skenario dua menggunakan node sebanyak 10 node dengan kecepatan perpindahan node 10 m/s. Skenario tiga menggunakan node sebanyak 10 node dengan kecepatan perpindahan node 15 m/s. Masing-masing skenario diuji pada protokol routing AODV dan DSR dengan posisi node yang acak dan menggunakan mobilitas *Random WayPoint* dimana node akan bergerak secara acak ke titik tujuan dan kecepatan yang acak.

3.2.1.5 Proses Simulasi



Gambar 3.12 Alur Proses Simulasi

Proses simulasi dilakukan secara virtual dengan *software VMWare Workstation* 16 *Player* pada sistem operasi Ubuntu 14.04 LTS menggunakan *software* simulasi yaitu NS2.35 (*Network Simulator* 2) dengan skenario yang telah dibahas pada subbab 3.2.1.4. Sebelum melakukan simulasi, terlebih dahulu membuat *traffic* CBR dan perpindahan *node* (*mobility*) yang sudah dijelaskan pada subbab 3.2.1.2, lalu memindahkan kedua *file* tersebut dalam satu *folder* simulasi agar lebih mudah saat melampirkan *file* dalam *script* simulasi. Dalam *script* simulasi ditambahkan *syntax* untuk dapat menggunakan *file traffic* dan *mobility* seperti *syntax* pada Gambar 3.13.

```
set val(cp)    "cbr-a10"    ;
set val(sc)    "skenario-m5n10"    ;
```

Gambar 3.13 *Syntax* melampirkan *file traffic* CBR dan *mobility node*

Untuk menggunakan paket data video, diperlukan sumber video seperti yang dijelaskan pada subbab 3.2.1.3. *File raw* video yang belum di-*encode* memiliki format YUV. *File* YUV di-*encode* menjadi mp4 dengan dua tahap yaitu dengan menggunakan *tool* x264 dan MP4Box. Perintah/*syntax*-nya sebagai berikut:

```
frida@frida: ~/Documents/node1/aodv
frida@frida:~/Documents/node1/aodv$ x264 -I 30 -B 64 --fps 30 --input-res 352x288
 -o a01.264 akiyo_cif.yuv
```

Gambar 3.14 *Command* untuk encode file video dengan x264

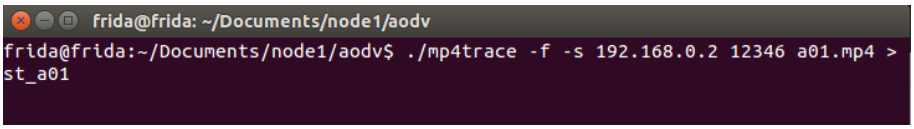
File raw video berformat *.yuv diubah menjadi *coded* video berformat *.264 yang menggunakan pengkodean H.264, dimana *file raw* video akiyo_cif.yuv di-*encode* dengan panjang GOP (*Group of Picture*) 30 *frame*, *bitrate* 64 kbps, dan *framerate* 30 *frame per second* yang memiliki resolusi 325x288 dan *output* disimpan dengan nama a01.264.

```
frida@frida: ~/Documents/node1/aodv
frida@frida:~/Documents/node1/aodv$ MP4Box -hint -mtu 1024 -fps 30 -add a01.264 a
01.mp4
```

Gambar 3.15 *Command* untuk encode file video dengan MP4Box

File berformat *.264 (a01.264) lalu dikonversi menjadi *file* *.mp4 agar dapat ditransmisikan dengan MTU (*Maximum Transmission Unit*) 1024bit dan *framerate* 30 *frame per second* yang disimpan dengan nama a01.mp4. Setiap *frame* akan terfragmentasi menjadi 1024*byte* untuk transmisi. Panjang paket *maximun*

(*packet size*) menjadi 1052 *byte*, termasuk *header* IP (20*bytes*) dan *header* UDP (8*bytes*) yang tercantum pada *file script* tcl simulasi.



Gambar 3.16 *Command* untuk trace file video dengan MP4Trace

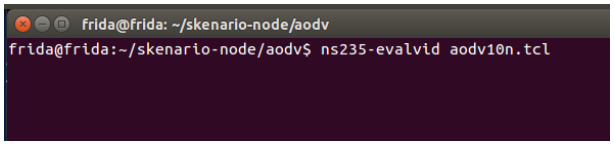
Tools MP4Trace menjadi alat evaluasi yang dapat menghasilkan *trace* dari sebuah video agar dapat dianalisa dengan EvalVid dan NS2. Alat MP4Trace dari Evalvid dapat mengirim *file* MP4 per RTP / UDP ke *host* tujuan yang ditentukan. File video yang diterjemahkan menjadi *trace file* berisi nomor *frame*, jenis *frame* (I, P, B), besar paket, dan waktu yang disimpan dengan nama *trace* st_a01. *Trace file* ini digunakan saat menjalankan proses simulasi pada script tcl simulasi utama.

Pada *script* simulasi dan *file traffic* yang dibuat dengan generator, didalamnya ditambahkan *syntax* untuk *traffic* data video di baris terakhir. Gambar 3.17 merupakan *syntax* untuk menambahkan *traffic* data video yang digunakan dalam simulasi pada *file traffic* CBR dengan jumlah 10 *node*. Melampirkan video ke *server* dengan membuat sumber *traffic* yaitu UDP *agent* dan melampirkannya ke *node*. *Traffic* UDP *agent* dengan nama udp_1 pada *node* 0 menggunakan *traffic* video tambahan pada Evalvid yaitu myUDP yang akan menghasilkan *sender trace file* bernama sd_a10. *Traffic agent* udp_1 dihubungkan dengan *agent null* dengan nama null_1 pada *node* 0 yang akan menghasilkan *receiver trace file* dengan nama rd_a01.

```
#attach video to server
#create source traffic
#Create a UDP agent and attach it to node
#
set udp_1 [new Agent/myUDP]
$ns_ attach-agent $node_(0) $udp_1
$udp_1 set packetSize_ $packetSize
$udp_1 set_filename sd_a10
set null_1 [new Agent/myEvalvid_Sink]
$ns_ attach-agent $node_(9) $null_1
$ns_ connect $udp_1 $null_1
$null_1 set_filename rd_a10
#
```

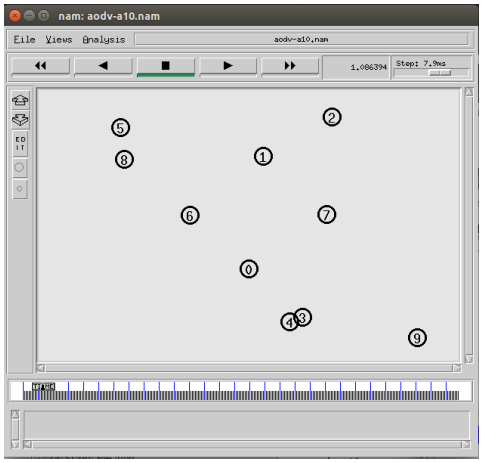
Gambar 3.17 *Syntax* tambahan untuk *traffic* data video

Simulasi dijalankan setelah menambahkan *syntax*/kode *traffic* video dan meng*codec file* video. Untuk melakukan simulasi, perlu masuk ke direktori dimana kumpulan *file* simulasi disimpan. Masing-masing skenario dijalankan secara bergantian sebanyak 3 kali pengujian. Perintah untuk menjalankan *script* simulasi ditunjukkan pada Gambar 3.18.

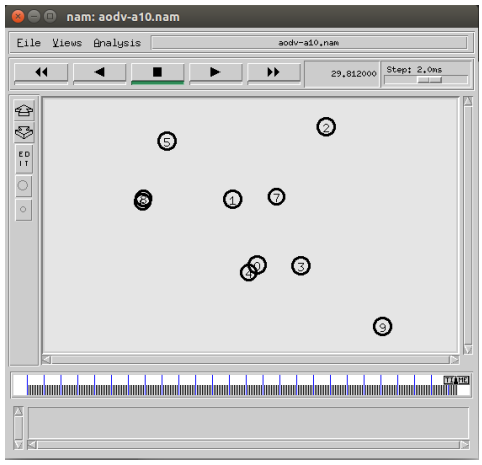


Gambar 3.18 *Command* untuk menjalankan simulasi NS2

Gambar 3.19 merupakan topologi jaringan pada MANET dan juga posisi awal *node* dengan *node* yang digunakan sebanyak 10 *node* pada luas area 750 x 500 m. Gambar 3.20 merupakan posisi *node* pada saat akhir simulasi selesai dimana posisinya berubah dari posisi/topologi awal.

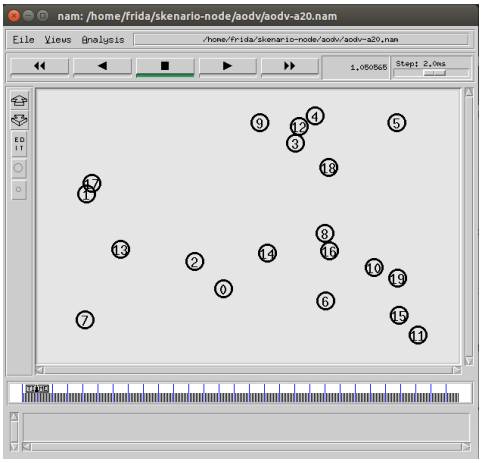


Gambar 3.19 Posisi awal node dengan jumlah 10 node

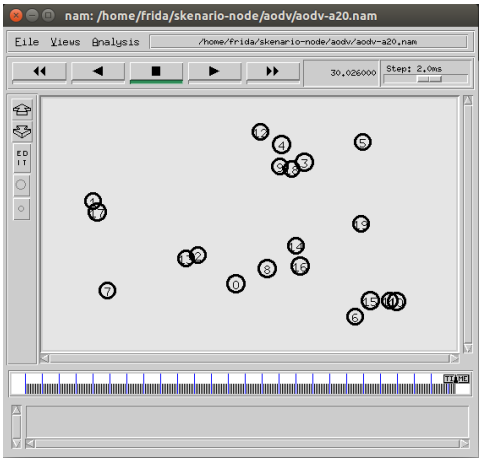


Gambar 3.20 Posisi akhir node dengan jumlah 10 node

Gambar 3.21 merupakan topologi jaringan pada MANET dan juga posisi awal node dengan *node* yang digunakan sebanyak 20 *node* pada luas area 750 x 500 m. Gambar 3.22 merupakan posisi node pada saat akhir simulasi selesai dimana posisinya berubah dari posisi/topologi awal.

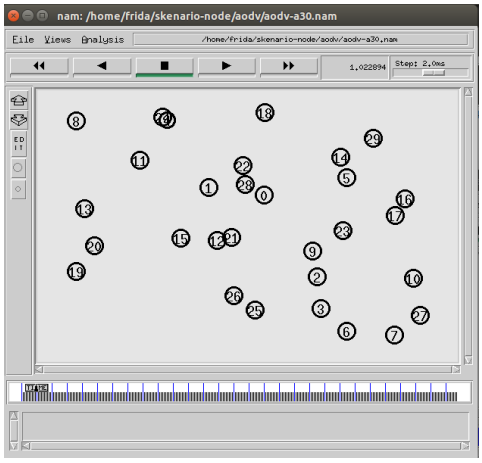


Gambar 3.21 Posisi awal node dengan jumlah 20 node

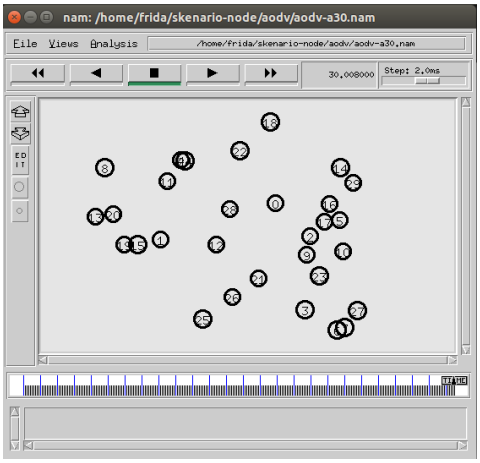


Gambar 3.22 Posisi akhir node dengan jumlah 10 node

Gambar 3.23 merupakan topologi jaringan pada MANET dan juga posisi awal node dengan *node* yang digunakan sebanyak 30 *node* pada luas area 750 x 500 m. Gambar 3.24 merupakan posisi node pada saat akhir simulasi selesai dimana posisinya berubah dari posisi/topologi awal.



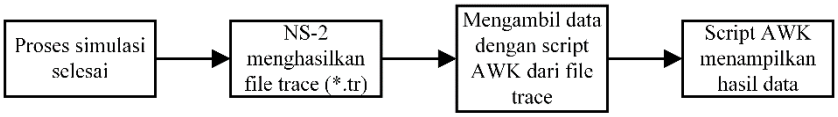
Gambar 3.23 Posisi awal node dengan jumlah 30 node



Gambar 3.24 Posisi akhir node dengan jumlah 30 node

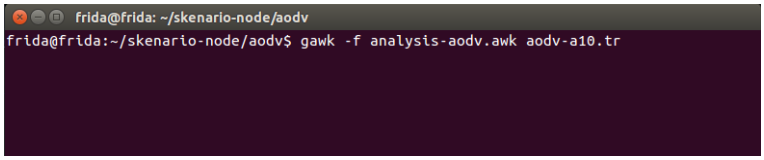
Ketika simulasi berakhir, NS2 akan menghasilkan *trace file* berekstensi *.tr dan *file* NAM berekstensi *.nam yang dapat menampilkan visualisasi *node* pada jaringan.

3.2.1.6 Pengambilan Data



Gambar 3.25 Alur Proses Pengambilan Data

Pengambilan data dilakukan menggunakan *script* AWK dengan mengambil data dari *trace file* yang dihasilkan saat simulasi selesai. Data parameter yang diambil untuk dianalisis yaitu *packet loss*, *throughput*, *delay*, dan *jitter* dari protokol *routing* AODV dan DSR setelah dilakukan simulasi pada NS2. Gambar 3.26 menunjukkan *syntax* yang digunakan untuk mengambil hasil parameter QoS dengan *script* AWK dari *trace file*.



Gambar 3.26 *Syntax* untuk mengambil hasil parameter QoS

3.2.1.7 Analisis Data

Tahap ini dilakukan ketika pengambilan data dari hasil simulasi selesai dan dapat menghasilkan *output* untuk dianalisis. Analisis yang dilakukan yaitu dengan

membandingkan hasil parameter *packet loss*, *throughput*, *delay*, dan *jitter* dari protokol *routing* AODV dan DSR pada jaringan MANET menggunakan aplikasi *video streaming* dengan skenario penambahan jumlah node dan kecepatan perpindahan node. Analisa dilakukan berdasarkan data parameter QoS yang diperoleh dan divisualisasikan ke dalam bentuk grafik menggunakan *Microsoft Excel* agar memudahkan proses analisa, selanjutnya akan dibahas pada BAB 4.