

BAB III METODOLOGI PENELITIAN

Metodologi penelitian dapat berisi penjelasan diagram penelitian. Tahap-tahap penelitian diurai pada alur penelitian. Penelitian ini menggunakan alat yang mendukung untuk proses dilakukannya penelitian. Penelitian ini menggunakan topologi sebagai objek untuk pengambilan data pada tahap penelitian.

3.1. ALAT YANG DIGUNAKAN

3.1.1. PERANGKAT KERAS (*HARDWARE*)

Perangkat keras yang dipergunakan dalam penelitian ini memanfaatkan satu buah laptop berfungsi tempat pembuatan dan pengujian dengan spesifikasi seperti terdapat di tabel 3.1.

Tabel 3. 1 Perangkat Keras Pengujian

No	Nama Perangkat	HP-14s-dk0xxx
1	<i>Processor</i>	AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx (4 CPUs), `2.6GHz
2	RAM	4GB
3	Penyimpanan	128GB

3.1.2. PERANGKAT LUNAK (*SOFTWARE*)

Pada penelitian ini terdapat beberapa *software* yang dipergunakan untuk mendukung penelitian yakni NS2, JOSM, SUMO, VirtualBox, Ubuntu dan Mendeley.

Tabel 3. 2 Perangkat Lunak Pengujian

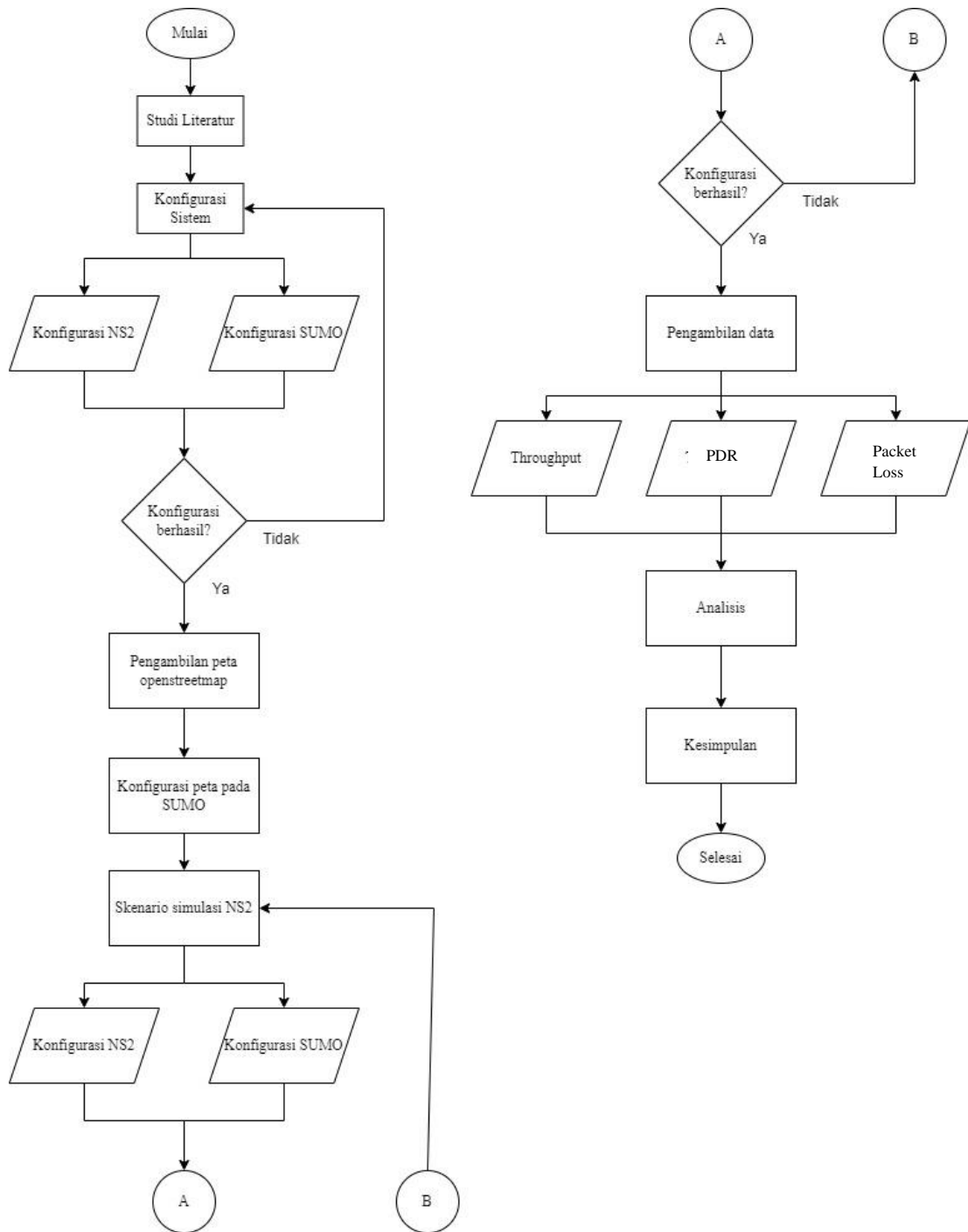
No.	Software	Versi	Fungsi
1	NS2	2.35	Perancangan dan Simulasi
2	SUMO	1.12.0	Membuat topologi jalan
3	OSM	18513	Sumber peta
4	VirtualBox	6.1.30	Mesin virtual
5	Ubuntu	20.04.4	<i>Operating system</i>
6	Mendeley	1.19.8	Membuat citasi pada <i>word</i>

3.2. ALUR PENELITIAN

Pada penelitian ini dipergunakan metode penelitian kuantitatif pada proses pengumpulan data. Data yang dikumpulkan akan dianalisa berdasarkan parameter QoS seperti *packet delivery ratio*, *throughput* dan *packet loss* untuk mengetahui kualitas suatu jaringan. Pada Gambar 3.1 dibawah menunjukkan alur penelitian yang dimulai dari melakukan studi literatur dengan mencari beberapa penelitian terdahulu yang digunakan untuk menentukan judul dan arah penelitian. Kemudian dilanjutkan dengan menentukan judul penelitian, metode penelitian, skenario penelitian dan topologi penelitian.

Selanjutnya melakukan konfigurasi sistem dari beberapa perangkat lunak seperti Network Simulator 2 dan SUMO. Jika pada tahap tersebut terdapat kegagalan maka proses akan diulang Kembali pada tahap konfigurasi sistem. Namun jika berhasil maka akan beralih ke tahap pengambilan peta pada *openstreetmap*. Pengambilan peta pada *openstreetmap* sesuai dengan yang telah ditentukan sebelumnya yakni jalan permukiman desa Sokaraja Tengah. Setelah peta telah diperoleh, tahap selanjutnya adalah *editing* pada aplikasi SUMO untuk membuat topologi jalan. *Editing* pada aplikasi SUMO dapat mengubah batas kecepatan jalan raya atau jalan permukiman serta dapat menghapus beberapa jalan yang tidak terhubung.

Setelah konfigurasi NS2 dan konfigurasi SUMO berhasil, beralih ke pengambilan data. Pada tahap pengambilan data ini dilakukan sesuai skenario seperti kecepatan *node* dan jumlah *node* pada dua routing protokol yakni AOMDV dan AODV. Data tersebut mencakup hasil parameter seperti *packet delivery ratio*, *throughput* dan *packet loss*. Nilai parameter tersebut didapat melalui *script* bertipe *awk* yang dibuat untuk mengetahui nilai-nilai QoS.

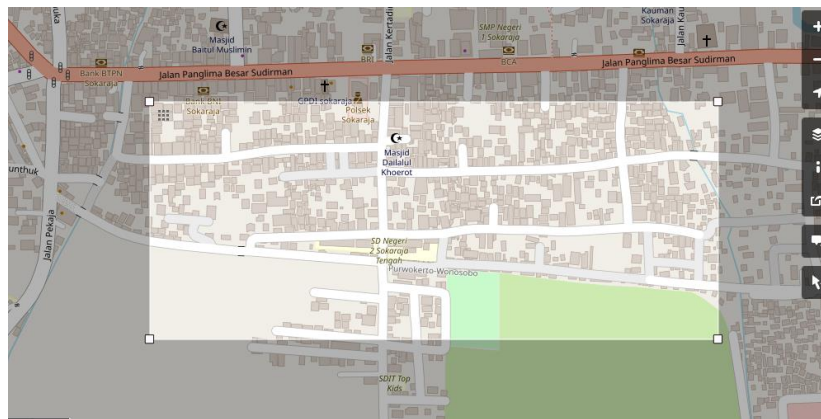


Gambar 3. 1 Alur Diagram Penelitian

Jika proses simulasi pada NS2 mengalami kegagalan maka akan diulang ke tahap skenario simulasi NS2. Selanjutnya dilakukan Analisa dari hasil simulasi dan membandingkan performansi jaringan terhadap parameter dari tiap skenario. Pada tahap akhir penelitian dapat diperoleh beberapa kesimpulan dan saran dari hasil simulasi dan analisis.

3.3. RANCANGAN TOPOLOGI JARINGAN

Topologi jaringan pada VANET bergantung pada mobilitas *node* yang dimodelkan. Oleh karena itu, bentuk dari geografis jalan atau peta yang digunakan sebagai rute pergerakan *node* simulasi akan berpengaruh pada performansi jaringan. Pada penelitian ini digunakan jalan permukiman desa Sokaraja Tengah. Desa Sokaraja Tengah dipilih karena sesuai dengan skenario yang akan diujikan seperti jumlah kendaraan yang bervariasi dari sepi hingga ramai serta kecepatan kendaraan pada lingkungan tersebut memiliki variasi kecepatan rendah dan kecepatan tinggi. *Node-node* yang digunakan antara lain *node source* dan *node destination*.



Gambar 3. 2 Peta Topologi Jaringan

Tabel 3. 3 Parameter Pengujian

PARAMETER	NILAI
Protokol	AOMDV, AODV
Area	1500x1500m
Pergerakan <i>Node</i>	<i>Random Way Point</i>
Standar Teknologi	IEEE.802.11p
Antena Model	<i>Omni Directional</i>
Jumlah <i>Node</i>	6, 20 dan 40 <i>node</i>
Kecepatan <i>Node</i>	8m/s, 20m/s
Traffic Model	CBR
Tipe Kanal	<i>Wireless channel</i>
Tujuan	<i>Random</i>
Waktu Simulasi	100 detik

3.4. KONFIGURASI SIMULASI PENELITIAN

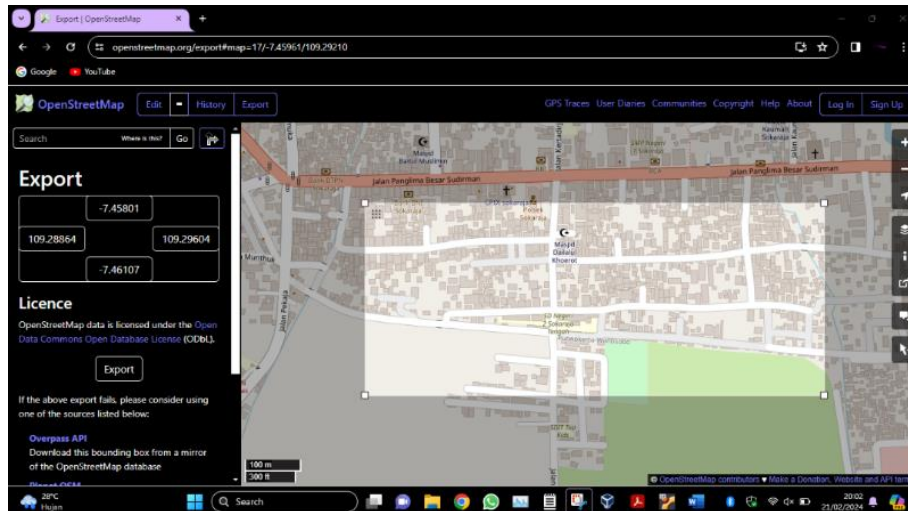
Skenario simulasi dibuat dan disesuaikan untuk mendapatkan tujuan dari penelitian ini yakni mengetahui kinerja jaringan VANET dalam kondisi AOMDV dan AODV. Oleh karena itu akan dibuat dua skenario yaitu skenario kecepatan *node* dan jumlah *node* pada VANET terhadap parameter QoS. Pemilihan jumlah *node* sesuai dengan pengamatan langsung rata-rata pada waktu tertentu seperti pagi, siang dan malam. [19] Sedangkan pemilihan kecepatan *node* sesuai dengan aturan lalu lintas. [20]

Tabel 3. 4 Tabel Skenario

Skenario	Kecepatan (m/s)	Jumlah <i>Node</i>
AOMDV	8	6
		20
		40
	20	6
		20
		40
AODV	8	6
		20
		40
	20	6
		20
		40

3.4.1. KONFIGURASI MOBILITAS

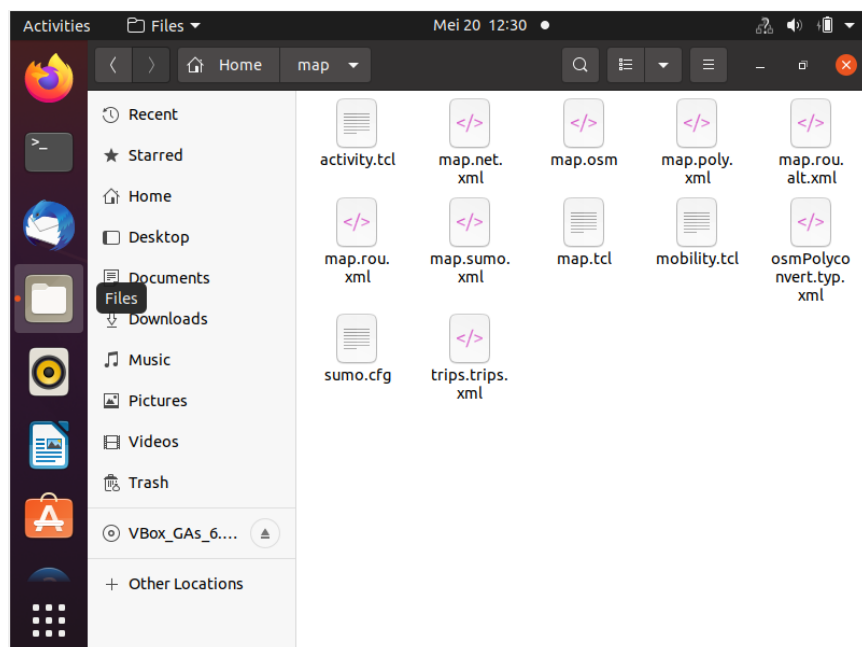
Pada penelitian ini sistem disimulasikan pada jalan permukiman desa Sokaraja Tengah. Koordinat dan peta tersebut diperoleh dari file *openstreetmap* yang bertipe *.osm* lalu akan diubah ke *format* yang bisa dijalankan dengan simulasi lalu lintas berupa SUMO (*Simulation Urban Mobility*). Pengambilan peta dari lokasi yang digunakan bisa dilihat di gambar berikut



Gambar 3. 3 Pengambilan Peta Pada *Open Street Map*

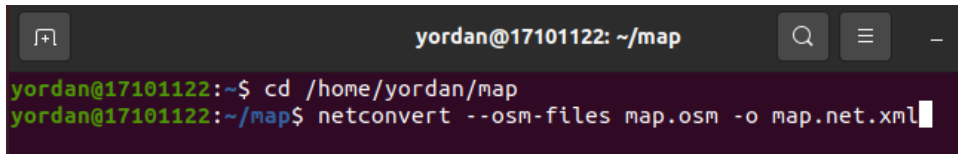
Untuk menggunakan *file .osm* tersebut terdapat beberapa proses konversi dalam *Terminal Emulator* dengan melakukan perintah. Proses tersebut seperti:

1. Memasukan *file map.osm* pada *folder* dan diberi nama *folder map*.
2. Pada *folder map* ditambahkan *file osmPolyconverter.typ.xml* yang didapat dari *directory folder* instalasi SUMO seperti *sumo-1.12.0*. *File osmPolyconverter.typ.xml* digunakan untuk mengkonversi *file .osm* menjadi *file* bertipe SUMO.



Gambar 3. 4 Isi *Folder Map*

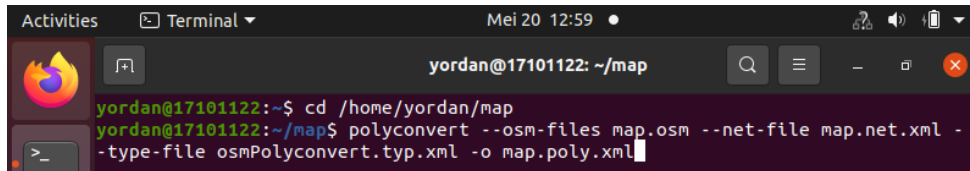
3. Kemudian buka *terminal ubuntu* di kondisi *root* di *directory map* dengan perintah berikut untuk konversi *file .osm* menjadi *net.xml*



```
yordan@17101122: ~/map
yordan@17101122:~$ cd /home/yordan/map
yordan@17101122:~/map$ netconvert --osm-files map.osm -o map.net.xml
```

Gambar 3. 5 Konversi File .osm menjadi net.xml

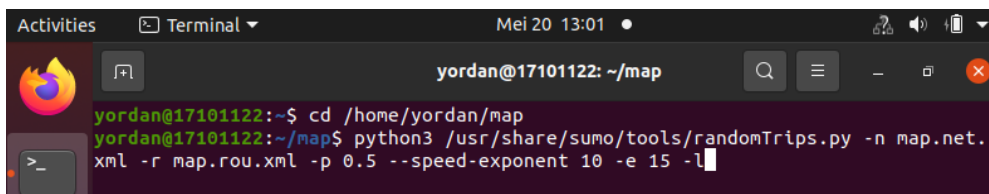
4. Berikutnya setelah file *map.net.xml* didapat dalam folder directory kemudian dilakukan konversi kembali yang menghasilkan file *.poly.xml*



```
Activities Terminal Mei 20 12:59
yordan@17101122: ~/map
yordan@17101122:~$ cd /home/yordan/map
yordan@17101122:~/map$ polyconvert --osm-files map.osm --net-file map.net.xml -
-type-file osmPolyconvert.typ.xml -o map.poly.xml
```

Gambar 3. 6 Konversi File Menjadi .poly.xml

5. Setelah mendapat file *.poly.xml* berikutnya dilakukan konversi memperoleh file *.rou.xml* memanfaatkan file *randomTrips.py* dan *map.net.xml*.



```
Activities Terminal Mei 20 13:01
yordan@17101122: ~/map
yordan@17101122:~$ cd /home/yordan/map
yordan@17101122:~/map$ python3 /usr/share/sumo/tools/randomTrips.py -n map.net.
xml -r map.rou.xml -p 0.5 --speed-exponent 10 -e 15 -l
```

Gambar 3. 7 Konversi File ke .rou.xml

Beberapa perintah berikut digunakan dalam membentuk mobilitas file *randomTrips.py* dengan *random (map.rou.xml)* sesuai rancangan jalur yang telah ditentukan sebelumnya yaitu file *map.net.xml*. Sesuai perintah tersebut sekaligus dibentuk skenario kecepatan *node* dan jumlah *node* yang dipergunakan dalam satuan m/s. Pada gambar 3.9 ditunjukkan jumlah *node* sebesar 30 berasal dari 0.5 dikali dengan 15 sehingga untuk merubah jumlah *node* dapat dilakukan dengan mengganti angka 15 tersebut. Kemudian *speed exponent* 10 menandakan kecepatan setiap *node* maksimum sebesar 10m/s atau sekitar 36km/jam.

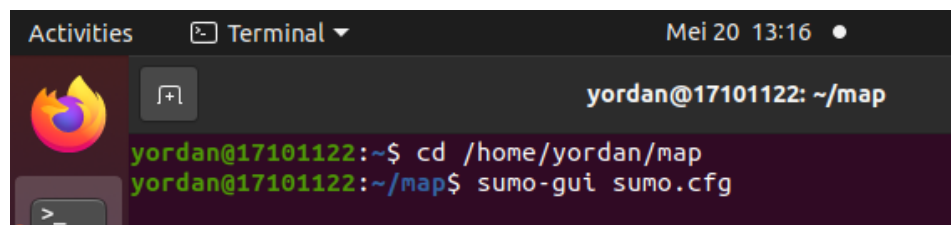
6. Agar SUMO dapat berjalan baik perlu ada perintah tambahan dimana perintah tersebut mengambil *source* data dari *map.net.xml*, *.rou.net.xml* dan *map.poly.xml* yang sudah terbuat oleh tahap konversi sebelumnya. File tersebut disimpan dalam bentuk *.sumo.cfg*. Perintah tersebut dapat diperoleh pada kode program dibawah ini.

```

<Configuration>
  <input>
    <net-file value="map.net.xml"/>
    <route-files value="map.rou.xml"/>
    <additional-files value="map.poly.xml"/>
  </input>
  <time>
    <begin value="0"/>
    <end value="100"/>
    <step-length value="0.1"/>
  </time>
</configuration>

```

7. Hasil konversi pada kode program diatas dijalankan di aplikasi SUMO dengan perintah dalam *file directory* tersebut yaitu



```

Activities Terminal Mei 20 13:16
yordan@17101122: ~/map
yordan@17101122:~$ cd /home/yordan/map
yordan@17101122:~/map$ sumo-gui sumo.cfg

```

Gambar 3. 8 Perintah Menjalankan SUMO



Gambar 3. 9 Peta Hasil Konversi OSM

8. Kemudian dikonversi kembali menjadi *sumo.xml* menggunakan perintah

```

root@17101122:/home/yordan/map# sumo -c sumo.cfg --fcd-output map.sumo.xml
Step #0.00 (1ms ~= 100.00*RT, ~1000.00UPS, vehicles TOT 1 ACT 1 BUF 0)

```

Gambar 3. 10 Perintah Konversi *sumo.xml*

9. *File sumo.xml* berikutnya dikonversi jadi tiga *file* terdiri dari *activity.tcl*, *mobility.tcl* dan *map.tcl*

```
root@17101122:/home/yordan/map# python3 /usr/share/sumo/tools/traceExporter.py
--fcd-input map.sumo.xml --ns2config-output map.tcl --ns2activity-output activi
ty.tcl --ns2mobility-output mobility.tcl
```

Gambar 3. 11 Perintah Konversi tiga file

Software NS2 versi 2.35 *all-in-one* yang dipergunakan dalam melakukan perintah yang telah dibuat di *file .tcl* berisikan *input* dan pengaturan *node* dan perintah yang digunakan selama proses simulasi berlangsung yang menghasilkan *file .nam* dan *.tr*. *Software* NAM digunakan dalam menjalankan simulasi jaringan sesuai *file .nam* yang telah dibuat. *Software* MS. Excel digunakan dalam menampilkan grafik kinerja protokol *routing* sesuai parameter *Packet Delivery Ratio*, *throughput* dan *packet loss* yang dianalisis lebih lanjut.

Setelah memperoleh *file .tcl* pada proses konversi *file .osm* di *folder map*, terdapat beberapa Langkah lanjutan untuk mengatur beberapa skenario, antara lain:

1. Melakukan *copy* isi *file simple-wireless.tcl* di *directory* program NS2 yakni **ns-allinone-2.3.5/ns-2.35/tcl/ex/** ke *file map.tcl*
2. Melakukan pengaturan *file map.tcl* agar sesuai format *wireless* yang bisa dijalankan menggunakan aplikasi NS2.

```
set val(chan) Channel/WirelessChannel ;#Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 1000 ;# max packet in ifq
set val(nn) 6 ;# number of mobilenodes
set val(rp) DSDV ;# routing protocol
#set val(rp) DSR ;# routing protocol
set val(x) 1433.27
set val(y) 1938.53
```

```

# Initialize Global Variables
set ns_          [new Simulator]
set tracefd      [open ZZ.tr w]
$ns_ trace-all $tracefd
set namtrace [open ZZ.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
$ns_ use-newtrace
# set up topography object
set topo    [new Topography]
$topo load_flatgrid $val(x) $val(y)
# Create God
create-god $val(nn)
# Create channel #1 and #2
set chan_1_ [new $val(chan)]
set chan_2_ [new $val(chan)]
# Create node(0) "attached" to channel #1
# configure node, please note the change below.
$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON \
    -channel $chan_1_
set node_(0) [$ns_ node]
# node_(1) can also be created with the same configuration, or with a
different
# channel specified.

```

```

# Uncomment below two lines will create node_(1) with a different channel.
# $ns_ node-config \
#           -channel $chan_2_
set node_(1) [$ns_ node]
set node_(2) [$ns_ node]
set node_(3) [$ns_ node]
set node_(4) [$ns_ node]
set node_(5) [$ns_ node]
set node_(6) [$ns_ node]
$node_(0) random-motion 0
$node_(1) random-motion 0
$node_(2) random-motion 0
$node_(3) random-motion 0
$node_(4) random-motion 0
$node_(5) random-motion 0

for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 20}
# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes
#set up color
$node_(5) color green
$ns_ at 0.0 "$node_(5) color green"
$ns_ at 0.0 "$node_(5) label source"
$node_(1) color blue
$ns_ at 0.0 "$node_(1) color blue"
$ns_ at 0.0 "$node_(1) label destination"
# Setup traffic flow between nodes
# TCP connections between node_(5) and node_(1)
set udp [new Agent/UDP]
$ns_ attach-agent $node_(5) $udp
set cbr [new Application/Traffic/CBR]
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

```

```

$cbr attach-agent $udp
set null [new Agent/Null]
$ns_ attach-agent $node_(1) $null
$ns_ connect $udp $null
$ns_ at 3.0 "$cbr start"
$ns_ at 50.0 "$cbr stop"
# Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 100.0 "$node_($i) reset";}
$ns_ at 100.0 "stop"
$ns_ at 100.01 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd}
puts "Starting Simulation..."
$ns_ run

```

Perintah `map.tcl` berikut merupakan salah satu contoh skenario yang diuji yakni menggunakan protokol *routing* AOMDV. *Node* atau kendaraan yang digunakan berjumlah 20 *node*.

3.5. SKENARIO PENGUJIAN DAN PENGAMBILAN DATA

Pada proses ini yaitu melakukan uji simulasi dengan melakukan pengiriman paket UDP dari *source node* menuju *destination node*. Selain itu dilakukan pengujian jumlah *node* dan kecepatan *node* untuk melihat konsistensi dalam kecepatan *node* dan jumlah *node*. Pengujian dilakukan sebanyak 10 kali dengan mengamati apakah terdapat anomali tertentu.

Tahap berikutnya adalah pengambilan data. Data yang dibutuhkan pada penelitian ini adalah nilai QoS berupa *packet delivery ratio*, *throughput* dan *packet loss*. Pengambilan data QoS ini dilakukan menggunakan *script* berbentuk *file* AWK. Pengambilan data QoS dilakukan ketika *source node* mulai mengirim paket data menuju *destination node*.

3.6. JADWAL PENELITIAN

Penelitian dilakukan selama 3 bulan. Perancangan kegiatan atau jadwal kegiatan selama 3 bulan tersebut diuraikan pada tabel 3.5 sebagai berikut:

Tabel 3. 5 Jadwal Penelitian

No.	Kegiatan	Waktu Penelitian (2024)													
		Februari				Maret				April				Mei	
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II
1	Studi Literatur	■													
2	Menentukan Judul dan Arah Penelitian		■	■											
3	Studi Pustaka			■	■										
4	Penyusunan Proposal				■	■									
5	Revisi Proposal					■	■								
6	Seminar Proposal						■								
7	Revisi Proposal Pasca Seminar						■	■							
8	Pengambilan Data							■	■						
9	Analisa Data								■	■					
10	Penulisan Hasil Data									■	■	■	■	■	
11	Sidang Skripsi														■
12	Revisi Skripsi														■