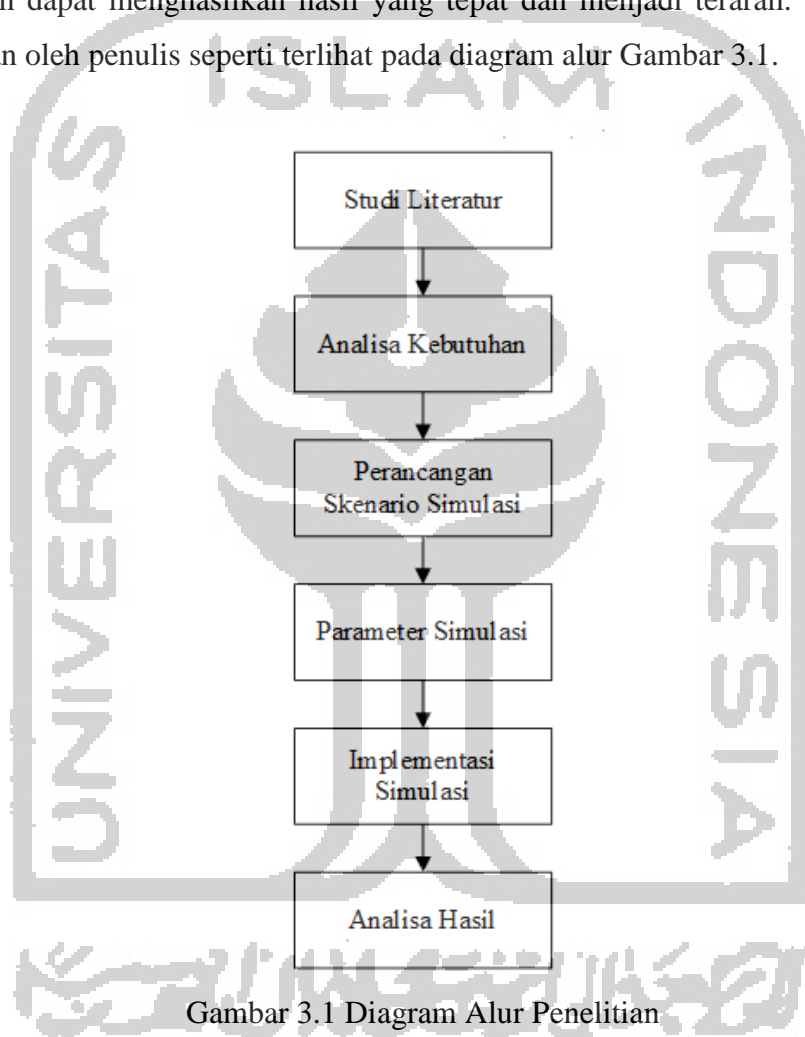


BAB III METODOLOGI

3.1 Alur Penelitian

Dalam menyelesaikan penelitian ini, ada beberapa tahapan yang dilakukan oleh penulis agar penelitian dapat menghasilkan hasil yang tepat dan menjadi terarah. Tahapan-tahapan yang dilakukan oleh penulis seperti terlihat pada diagram alur Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

3.2 Analisa Kebutuhan

Penulis mengklasifikasikan beberapa kebutuhan yang digunakan dalam penelitian ini dengan tujuan untuk menganalisa kebutuhan yang diperlukan untuk membuat simulasi jaringan VANET serta *requirement* untuk menjalankan baris kode NS-3 dan menganalisa data yang dihasilkan.

3.2.1 Analisa Kebutuhan *Hardware*

Pada penelitian ini analisa kebutuhan *hardware* yang digunakan untuk menjalankan simulasi jaringan yaitu sebuah perangkat komputer laptop dengan spesifikasi terlihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Komputer

Deskripsi	Spesifikasi
Merk	Asus
Tipe	A455LB
Processor	Intel core i3-5010U @2.1 GHz
RAM	8GB
Disk	120 GB SSD 1000 GB Harddisk
Display	Intel HD Graphics 5500 NVIDIA GeForce 940M

3.2.2 Analisa Kebutuhan *Software*

Analisa kebutuhan *software* yang digunakan oleh penulis pada penelitian ini adalah sebagai berikut:

- **Ubuntu**

Ubuntu merupakan sistem operasi *open source* yang didistribusikan oleh Linux yang berbasis Debian. Dengan dukungan *software* yang bersifat *open source* serta memiliki *support* jangka panjang akan memudahkan dalam menggunakan sistem operasi tersebut. Dalam penelitian ini penulis menggunakan sistem operasi ubuntu karena menjadi salah satu *platform* yang direkomendasikan untuk membuat simulasi jaringan menggunakan NS-3 serta kemudahan dalam melakukan konfigurasi *library*.

- **Network Simulator 3 (NS-3)**

Network simulator 3 atau NS-3 merupakan simulator jaringan yang berfokus pada bidang penelitian dan pendidikan. Dibawah lisensi GNU GPLv2 perangkat lunak NS-3 bersifat *open source*, tersedia dalam beberapa lintas *platform* yang dapat digunakan secara umum untuk kepentingan penelitian, pengembangan dan penggunaan. Pada infrastruktur simulasi NS-3 juga mendukung pengembangan model simulasi jaringan mendekati kondisi yang realistis dan memiliki karakteristik sebagai emulator jaringan *wireless* ataupun *wired*. Dengan dukungan

modul *library* yang lengkap serta dukungan *protocol* jaringan yang populer, beberapa model radio, MAC dan lainnya, memungkinkan untuk pengembangan model jaringan yang modern. Selain itu penulisan baris kode pada NS-3 menggunakan bahasa pemrograman C++ dan Python sehingga dapat disesuaikan dengan kebutuhan implementasi pada simulasi jaringan.

Adapun langkah-langkah untuk meng-*install Network Simulator 3* (NS3) pada sistem operasi Ubuntu adalah sebagai berikut:

- a. Mengunduh *file master Network Simulation 3* (NS-3) yang terdapat pada situs alamat NS-3 yaitu <https://www.nsnam.org/>.
- b. Meng-*install* beberapa *file library* yang berguna untuk mendukung NS-3 dengan mengetik setiap baris perintah pada konsol *terminal* Ubuntu. Seperti terlihat pada Gambar 3.2.

1	apt-get install gcc g++ python
2	apt-get install gcc g++ python python-dev
3	apt-get install python-setuptools git mercurial
4	apt-get install qt5-default mercurial
5	apt-get install python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev ipython
6	apt-get install gir1.2-gooocanvas-2.0 python-gi python-gi-cairo python-pygraphviz python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython ipython3
7	apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev
8	apt-get install autoconf cvs bzip2 unrar
9	apt-get install gdb valgrind
10	apt-get install uncrustify
11	apt-get install doxygen graphviz imagemagick
12	apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-utils texlive-lang-portuguese dvipng latexmk
13	apt-get install python-sphinx dia
14	apt-get install gsl-bin libgsl-dev libgsl23 libgslcblas0
15	apt-get install tcpdump
16	apt-get install sqlite sqlite3 libsqlite3-dev
17	apt-get install libxml2 libxml2-dev
18	apt-get install cmake libc6-dev libc6-dev-i386 libclang-6.0-dev llvm- 6.0-dev automake pip
19	pip install cxxfilt
20	apt-get install libgtk2.0-0 libgtk2.0-dev
21	apt-get install vtun lxc
22	apt-get install libboost-signals-dev libboost-filesystem-dev

Gambar 3.2 Baris Perintah Untuk Meng-*install File Library* Pendukung

- c. Membuat direktori baru dan lakukan ekstraksi *file master* NS-3.
- d. Masuk ke direktori hasil ekstraksi *file master* NS-3 dengan menggunakan konsol *terminal*.

- e. Lakukan *build file master* NS-3 untuk meng-*install* sistem NS3 dengan mengetik baris perintah pada konsol *terminal*. Seperti terlihat pada Gambar 3.3.

```
1 | ./build.py --enable-examples --enable-tests
```

Gambar 3.3 Baris Perintah *Build File* NS-3

- f. Sebelum meelakukan konfigurasi sistem NS-3, masuk pada direktori folder ns-3.26 dan lakukan konfigurasi dengan mengetik baris perintah pada konsol *terminal*. Seperti terlihat pada Gambar 3.4.

```
1 | ./waf -d debug --enable-examples --enable-tests configure
```

Gambar 3.4 Baris Perintah Konfigurasi *File* NS-3

- g. Selanjutnya mengecek apakah konfigurasi sistem NS-3 sudah berjalan benar dengan mengetik baris perintah pada konsol *terminal*. Seperti terlihat pada Gambar 3.5.

```
1 | ./test.py
```

Gambar 3.5 Baris Perintah Cek Konfigurasi NS-3

- **Sublime Text 3**

Sublime text 3 merupakan text editor lintas *platform* yang menggunakan *interface python api* yang dikembangkan oleh programmer dari Australia, Jon Skinner. Sublime text 3 mendukung banyak bahasa pemrograman dan bahasa *mark up* antara lain C++, C, Java, PHP, dan lain-lain. Selain itu terdapat dukungan fitur *plugin* tambahan untuk menambahkan beberapa fitur fungsi yang memudahkan dalam menulis baris kode seperti *auto correct*, *auto complete*, *command palette*, *multiple selections* dan lainnya.

- **PhyViz**

PhyViz merupakan simulasi visualisasi model mobilitas jaringan terdapat pada NS-3 yang dikembangkan oleh Gustavo Carnerio. Desain *interface* PhyViz menggunakan Gtk+ dan GooCanvas yang bekerja pada Phyton dan C++. PhyViz membuat visualisai simulasi jaringan secara langsung dan tidak menggunakan *trace file* sehingga berguna untuk keperluan *debugging*, seperti: mengetahui perilaku model mobilitas, paket data yang ditransmisikan, mengetahui informasi keadaan simulasi dan lain-lain.

- **Netanim**

Netanim merupakan *network animator offline* terdapat pada NS-3 yang dikembangkan oleh George F Riley dan John Abraham. Desain *interface* netanim dibuat menggunakan Qt *toolkit* yang disusun dengan bahasa C++. Pada netanim *file* animasi menggunakan *trace file*

berbentuk dalam *file Extensible Markup Language (XML)* yang dikumpulkan selama proses simulasi sehingga dapat digunakan untuk melihat hasil data simulasi.

3.3 Perancangan Skenario Simulasi

Perancangan skenario bertujuan guna mendukung simulasi VANET agar terlihat gambaran dari hasil simulasi dan mengetahui pengaruh terhadap hasil simulasi yang akan dibuat. Adapun skenario yang akan dibuat dengan kondisi tertentu, yaitu:

1. Kondisi dimana suatu jaringan VANET mengalami penambahan jumlah node pada suatu wilayah yang tetap dengan kecepatan bervariasi.
2. Kondisi dimana suatu jaringan VANET mengalami peningkatan kecepatan node pada suatu wilayah tetap dengan jumlah node tetap.

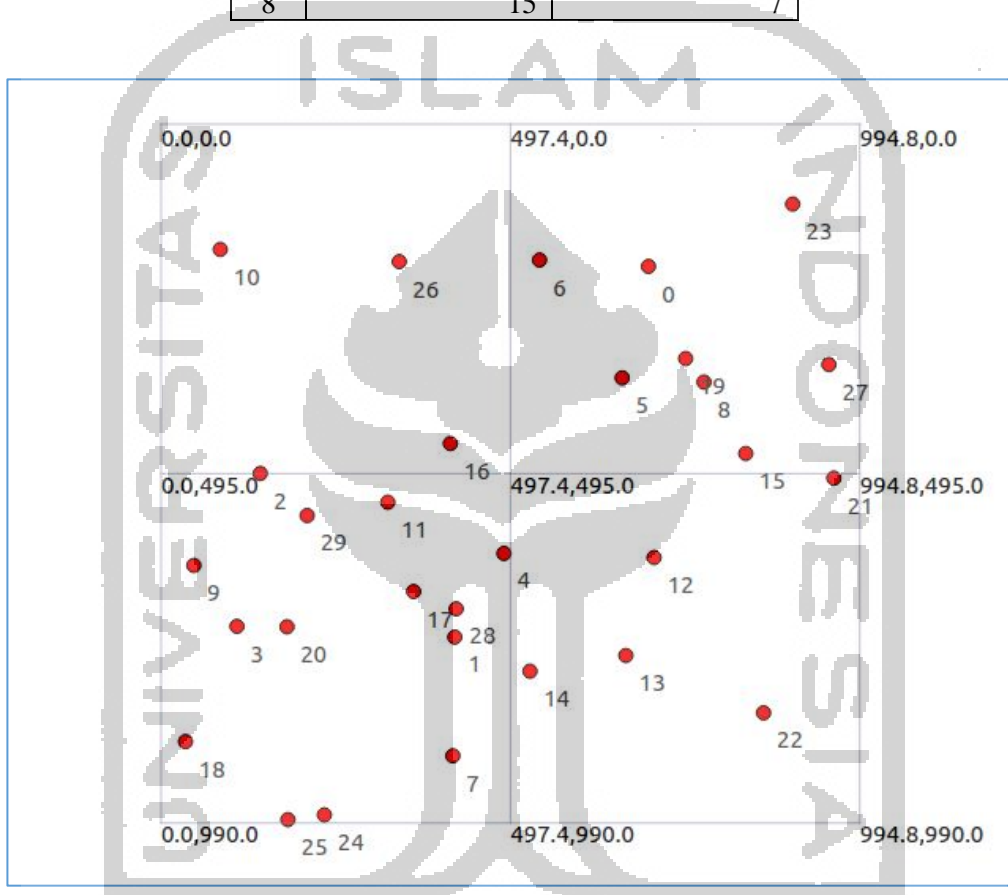
3.3.1 Skenario Penambahan Node

Skenario penambahan jumlah node pada suatu jaringan VANET bertujuan untuk mengetahui tingkat kepadatan node pada suatu wilayah tertentu dalam mentransmisikan paket data dari node sumber ke node tujuan berdasarkan rute yang dilalui. Untuk menganalisa transmisi paket data tersebut yaitu dengan melakukan uji kinerja *routing protocol* OLSR dan AODV berdasarkan parameter QoS yaitu: *Throughput*, *Packet Loss Rate*, *Packet Delivery Ratio*, *Delay* dan *Jitter*. Selanjutnya dari hasil analisa kinerja tersebut dilakukan perbandingan performansi dari kedua *routing protocol*.

Pada skenario penambahan jumlah node, penulis menggunakan node berjumlah 30, 50 dan 80 dengan luas area 1 km² yang ditempatkan secara *random*. Dalam proses komunikasi antara node sumber ke node tujuan dalam mengirimkan paket data dilakukan secara *random* dengan node pengirim dan penerima berjumlah 8 pasang yang dilakukan *one hop* ataupun *multi-hop*. Untuk pergerakan semua node bersifat *random* dengan kecepatan antara 0 m/s sampai 20 m/s. Node pengirim dan node penerima tersebut berlaku untuk semua skenario penambahan jumlah node. Adapun detail node pengirim dan node penerima terlihat seperti Tabel 3.2 dan hasil topologi jaringan VANET dengan penambahan jumlah node terlihat pada Gambar 3.6 dengan jumlah 30 node, Gambar 3.7 dengan jumlah 50 node dan Gambar 3.8 dengan jumlah 80 node.

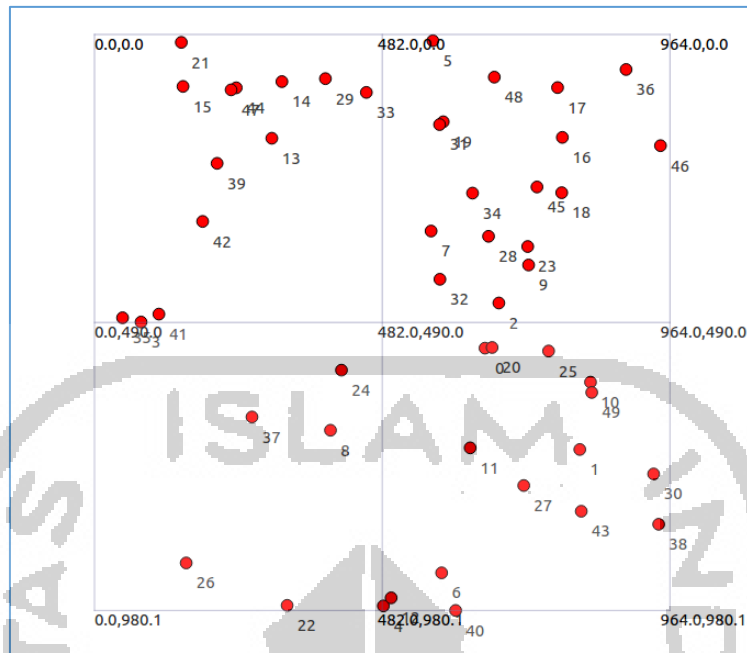
Tabel 3.2 Node Pengirim Dan Node Penerima

No	Node Pengirim	Node Penerima
1	8	0
2	9	1
3	10	2
4	11	3
5	12	4
6	13	5
7	14	6
8	15	7



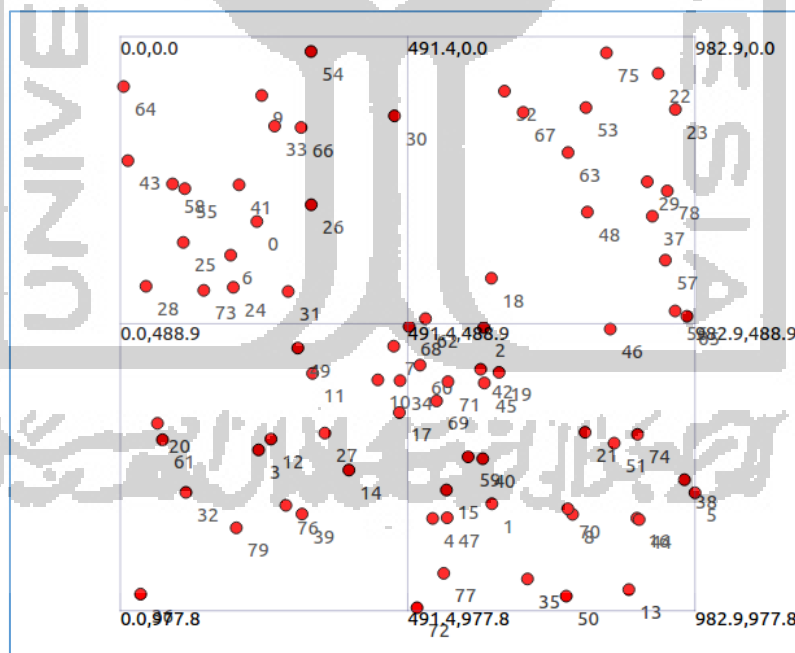
Gambar 3.6 Topologi 30 Node

Dari Gambar 3.6 terlihat topologi node berjumlah 30 yang tersebar secara acak dengan luas wilayah 1 km². Node tersebut memiliki indek nomor dari 0 sampai 29.



Gambar 3.7 Topologi 50 Node

Dari Gambar 3.7 terlihat topologi node berjumlah 50 yang tersebar secara acak dengan luas wilayah 1 km². Node tersebut memiliki indek nomor dari 0 sampai 49.



Gambar 3.8 Topologi 80 Node

Dari Gambar 3.8 terlihat topologi node berjumlah 80 yang tersebar secara acak dengan luas wilayah 1 km². Node tersebut memiliki indek nomor dari 0 sampai 79.

Pada skenario penambahan jumlah node menggambarkan komunikasi yang terjadi antar node sumber ke node tujuan. Hal tersebut untuk mengetahui pengaruh pada perubahan topologi

jaringan ketika mengirimkan paket data yang dilalui secara *one hop* atau *multi-hop*. Apabila jumlah node sedikit proses komunikasi antar node dapat berlangsung secara berkala atau terputus karena rute node yang dilalui tidak tersedia atau diluar jarak jangkauan sinyal transmisi nirkabel sedangkan apabila jumlah node bertambah proses komunikasi antar node dapat berlangsung secara terus menerus karena dapat menemukan rute node untuk sampai ke node tujuan sebagaimana masih dalam jarak jangkauan sinyal transmisi jaringan nirkabel.

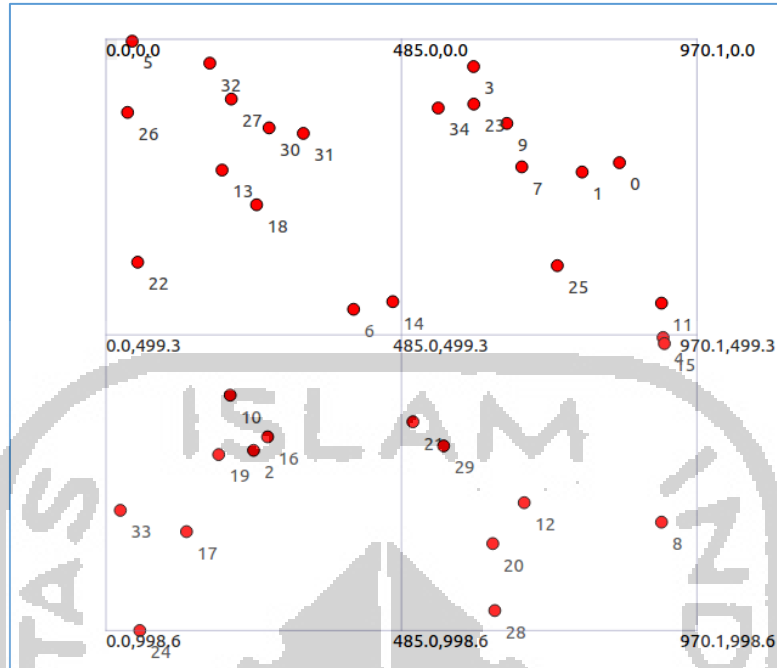
3.3.2 Skenario Peningkatan Kecepatan

Skenario peningkatan kecepatan node pada suatu jaringan VANET bertujuan untuk mengetahui tingkat keberhasilan node dalam menerima paket data pada suatu wilayah tertentu dan mentransmisikan paket data dari node sumber ke node tujuan berdasarkan rute yang dilalui. Untuk menganalisa transmisi paket data tersebut yaitu dengan melakukan uji kinerja *routing protocol* OLSR dan AODV berdasarkan parameter QoS yaitu: *Throughput*, *Packet Loss Rate*, *Packet Delivery Ratio*, *Delay* dan *Jitter*. Selanjutnya dari hasil analisis kinerja tersebut dilakukan perbandingan performansi dari kedua *routing protocol*.

Pada skenario peningkatan kecepatan node, penulis menggunakan node berjumlah 30 dengan luas area 1 km² yang ditempatkan secara *random* untuk semua kecepatan. Dalam proses komunikasi antara node sumber ke node tujuan dalam mengirimkan paket data dilakukan secara *random* dengan node pengirim dan penerima berjumlah 8 pasang yang dilakukan *one hop* ataupun *multi-hop*. Untuk pergerakan node bersifat tetap dengan kecepatan 10 m/s, 15 m/s dan 20 m/s. Node pengirim dan node penerima tersebut berlaku untuk semua skenario peningkatan kecepatan node. Adapun detail node pengirim dan node penerima terlihat seperti tabel 2.1 dan hasil topologi jaringan VANET dengan jumlah node 30 terlihat pada Gambar 3.9. Adapun node pengirim dan node penerima terlihat pada Tabel 3.3.

Tabel 3.3 Node Pengirim Dan Node Penerima

No	Node Pengirim	Node Penerima
1	8	0
2	9	1
3	10	2
4	11	3
5	12	4
6	13	5
7	14	6
8	15	7



Gambar 3.9 Topologi 35 Node

Dari Gambar 3.9 terlihat node berjumlah 35 yang tersebar secara acak dengan luas wilayah 1 km². Node tersebut memiliki indeks nomor dari 0 sampai 34.

Pada skenario peningkatan kecepatan node menggambarkan komunikasi yang terjadi antar node sumber ke node tujuan dapat berlangsung sangat cepat. Hal tersebut akan berpengaruh pada tingkat keberhasilan node dalam menerima dan mengirimkan data yang dilalui secara *one hop* atau *multi-hop*. Apabila kecepatan node rendah proses komunikasi antar node dapat berlangsung secara terus menerus, karena node dapat menemukan rute node untuk sampai ke node tujuan atau dalam jarak jangkauan transmisi sinyal nirkabel. Sedangkan apabila kecepatan node meningkat proses komunikasi dapat terputus karena rute node yang dilalui tidak tersedia atau diluar jarak jangkauan sinyal transmisi nirkabel.

3.4 Parameter Simulasi

Parameter simulasi ini memuat beberapa informasi yang berkaitan dengan bagaimana simulasi dijalankan dan digunakan sebagai indikator perbandingan simulasi. Nilai informasi dari setiap skenario akan berbeda sehingga akan memberikan hasil QoS yang berbeda sehingga dapat menjadi tolak ukur kinerja *routing protocol* OLSR dan AODV. Parameter simulasi yang ditetapkan sesuai skenario terlihat pada Tabel 3.4 dan Tabel 3.5.

Tabel 3.4 Parameter Simulasi Penambahan Node

Parameter	Nilai
Network Simulator	NS3.26
<i>Routing Protocol</i>	OLSR, AODV
Waktu Simulasi	100 detik
Area Simulasi	1 km x 1 km
<i>Mobility Model</i>	GaussMarkov
Jumlah Node	30, 50, 80
Kecepatan	0-20 m/s <i>random</i>
Jumlah Koneksi	8
<i>Transport Protocol</i>	UDP
<i>MAC Protocol</i>	IEEE 802.11p
<i>MAC Rate</i>	6 Mbps
<i>Data Type</i>	CBR
<i>Packet Size</i>	512 bytes
<i>Transmission Range</i>	300 m
<i>Radio Propagation</i>	<i>threeelogdistance</i>

Tabel 3.5 Parameter Simulasi Peningkatan Kecepatan

Parameter	Nilai
Network Simulator	NS3.26
<i>Routing Protocol</i>	OLSR, AODV
Waktu Simulasi	100 detik
Area Simulasi	1 km x 1 km
<i>Mobility Model</i>	GaussMarkov
Jumlah Node	30
Kecepatan	10 m/s, 15 m/s, 20 m/s konstant
Jumlah Koneksi	8
<i>Transport Protocol</i>	UDP
<i>MAC Protocol</i>	IEEE 802.11p
<i>MAC Rate</i>	6 Mbps
<i>Data Type</i>	CBR
<i>Packet Size</i>	512 bytes
<i>Transmission Range</i>	300 m
<i>Radio Propagation</i>	<i>threeelogdistance</i>

1. *Network Simulator*

Network simulator yaitu jenis dan versi simulator jaringan yang digunakan untuk membuat model simulasi jaringan.

2. *Routing Protocol*

Routing protocol yaitu jenis rute untuk berkomunikasi dan mengirimkan paket data dari node sumber ke node tujuan.

3. Waktu Simulasi

Waktu simulasi yaitu waktu yang digunakan selama proses simulasi berlangsung.

4. Area Simulasi

Area simulasi yaitu luas wilayah yang digunakan untuk memuat node-node dalam melakukan pergerakan.

5. *Mobility Model*

Mobility model yaitu jenis dari pergerakan node yang digunakan selama proses simulasi.

6. Jumlah Node

Jumlah node yaitu jumlah node atau kendaraan yang digunakan untuk saling berkomunikasi dan mengirimkan paket data dari node sumber ke node tujuan.

7. Kecepatan

Kecepatan yaitu tingkatan kecepatan node untuk menempuh suatu jarak dalam waktu tertentu.

8. Jumlah koneksi

Jumlah koneksi yaitu jumlah node yang mengirimkan paket data ke node tujuan.

9. *Transport Protocol*

Transport protocol yaitu jenis dari lapisan pembawa paket data dari node sumber ke node tujuan.

10. *MAC Protocol*

MAC protocol yaitu jenis dari standar *wireless* yang digunakan untuk melakukan proses komunikasi.

11. *MAC Rate*

MAC rate yaitu lebar pita yang digunakan selama proses pengiriman paket data dari node sumber ke node tujuan.

12. *Data Type*

Data type yaitu jenis *traffic* yang digunakan untuk mentransmisikan data dalam satuan bit per detik.

13. *Packet Size*

Packet size yaitu ukuran paket data yang dikirimkan dari node sumber ke node tujuan.

14. *Transmission Range*

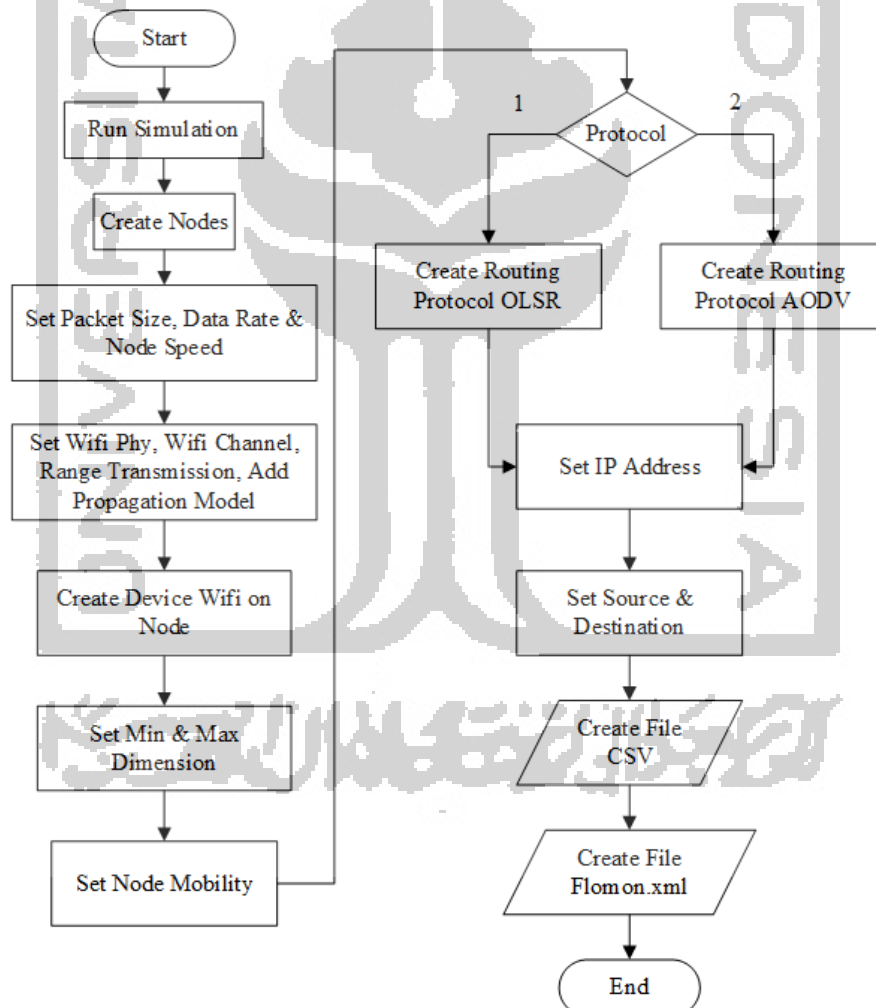
Transmission range yaitu jarak transmisi jangkauan *wireless* yang dapat terhubung antar node.

15. Radio Propagation

Radio propagation yaitu jenis radio propagasi atau proses perambatan gelombang yang digunakan untuk mentransmisikan data dari node sumber ke node tujuan.

3.5 Flowchart Program

Flowchart atau diagram alur merupakan suatu bagan yang menggambarkan rangkaian proses jalannya program dengan menghubungkan suatu proses dengan proses lain. Pada pembuatan *flowchart* ini menggambarkan proses dan jalannya simulasi dari skenario yang telah dibuat sehingga akan digunakan untuk menjalankan simulasi VANET. *Flowchart* dari simulasi program VANET terlihat pada Gambar 3.10.



Gambar 3.10 *Flowchart* Program Simulasi VANET

Pada saat akan menjalankan program simulasi, pengguna harus menuliskan baris perintah pada konsol *terminal* yang terdiri dari jumlah node yang digunakan, jenis *routing protocol*, kecepatan node, jumlah paket data dan memberi nama *file csv*. Kemudian program akan membuat node dan mengatur *packet size*, *data rate* dan kecepatan node yang telah ditentukan.

Selanjutnya node tersebut akan dilakukan konfigurasi jenis *wifi phy*, *wifi channel*, jarak transmisi, dan model radio propagasi yang digunakan untuk mengirimkan dan menerima paket data serta menghubungkan beberapa node. Kemudian program membuat *device wifi card* yang akan dilakukan *install* pada setiap node sehingga dapat saling terhubung.

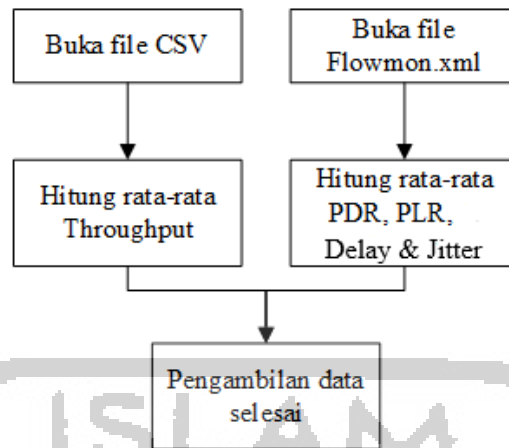
Program akan membuat suatu dimensi luas wilayah yang digunakan untuk menempatkan node untuk melakukan simulasi. Selain itu juga sebagai tempat pergerakan node yang saling terhubung satu sama lain. Selanjutnya program akan menentukan jenis *routing protocol* yang dipilih dan membuat *routing protocol* pada setiap node pada saat simulasi.

Kemudian node-node tersebut akan dilakukan pengaturan pengalamatan IP pada masing-masing node berdasarkan rentang IP yang telah ditentukan. Program akan mengatur node pengirim dan node penerima dalam mengirimkan paket data dan mengumpulkan *traffic data* pada saat simulasi. Selanjutnya dilakukan konfigurasi pada fungsi *FlowMonitor* yang berguna untuk menghitung hasil dari *traffic data* antar node.

Pada tahap terakhir program akan mengumpulkan *traffic data* yang dikelompokkan pada *file csv* yang berisi data *Throughput* dan file *flowmon.xml* yang berisi data *Delay*, paket yang dikirimkan, paket yang diterima dan lainnya pada setiap node pengirim dan penerima. Kemudian program simulasi berhenti dan selesai.

3.6 Tahapan Pengambilan Data

Dalam tahapan pengambilan data setelah simulasi jaringan VANET selesai, terdapat *file* data hasil simulasi, yaitu file dengan ekstensi *.csv* dan *flowmon.xml* yang digunakan untuk menghitung parameter QoS, seperti *Throughput*, *Packet Loss Rate*, *Packet Delivery Ratio*, *Delay* dan *Jitter*. Tahapan tersebut terlihat pada Gambar 3.11.



Gambar 3.11 Tahapan Pengambilan Data

Tahapan pertama yaitu dengan membuka *file* dengan ekstensi *.csv* yang digunakan untuk menganalisa data *Throughput*. Dari data tersebut terdapat nilai-nilai yang dihasilkan selama proses simulasi, selanjutnya dihitung untuk mencari rata-rata *Throughput* sesuai skenario dan *routing protocol*. Sedangkan tahapan kedua yaitu dengan membukan *file* dengan ekstensi *flowmon.xml* yang digunakan untuk menganalisa data dari *Packet Loss Rate*, *Packet Delivery Ratio*, *Delay* dan *Jitter*. Dari data tersebut terdapat nilai-nilai yang dihasilkan selama proses simulasi, selanjutnya dihitung untuk mencari rata-rata *Throughput*, *Packet Loss Rate*, *Packet Delivery Ratio*, *Delay* dan *Jitter* sesuai skenario dan *routing protocol*.