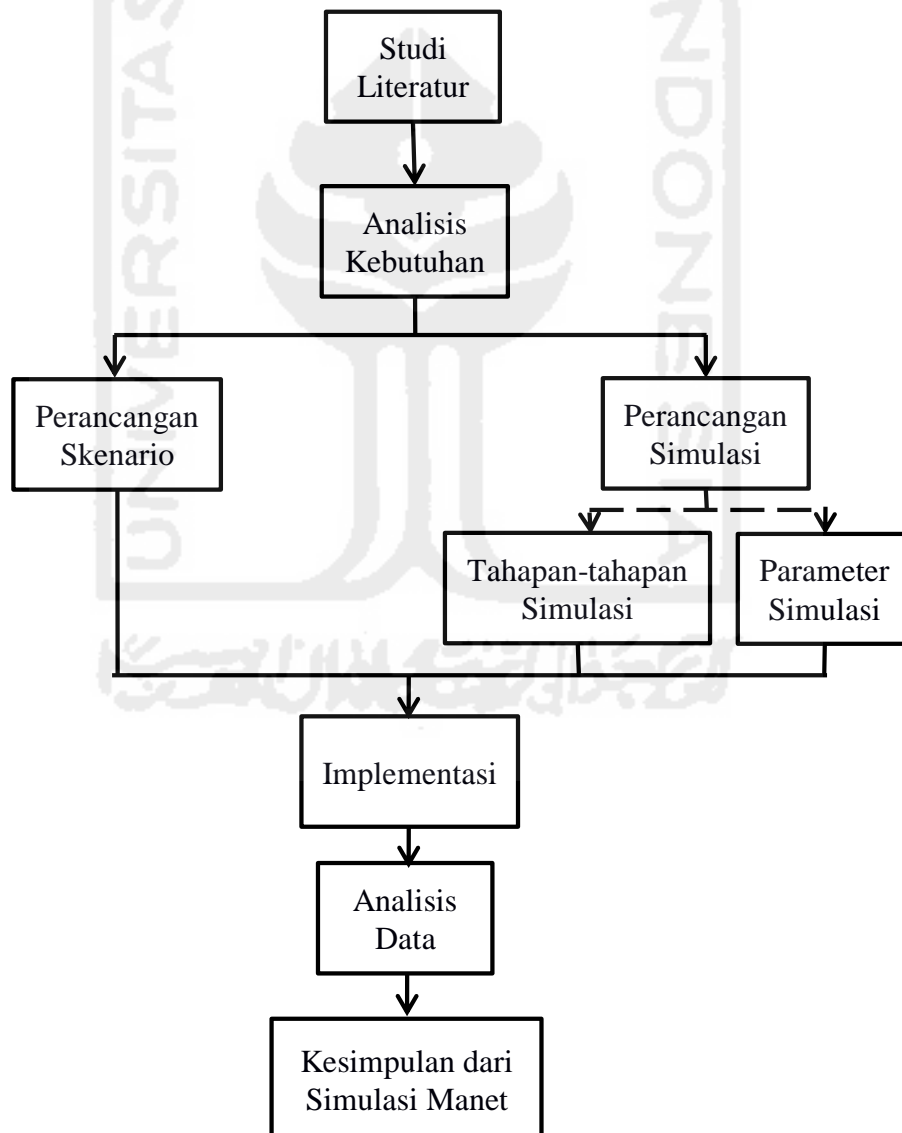


BAB III METODOLOGI

3.1. Alur Penelitian

Dalam pembuatan simulasi *Mobile Ad-Hoc Network* (MANET), sebelumnya dilakukan tahapan-tahapan untuk menyelesaikannya. Berikut gambaran tahapan-tahapan tersebut dapat dilihat pada diagram alur di bawah ini:



Gambar 3.1. Alur penelitian

3.2. Analisis Kebutuhan

Analisis kebutuhan pada simulasi MANET ini dilihat dari *requirement* dasar yang dibutuhkan dalam menggunakan *script* NS3 dan menganalisis data yang dihasilkan. Hasil analisis kebutuhan dapat dilihat sebagai berikut:

3.2.1. Analisis Kebutuhan *Hardware*

1. Laptop

Sebuah laptop Asus EeePc 1225B, dengan spesifikasi *processor* AMD E-450 dan RAM 2048 MB yang digunakan untuk melakukan simulasi MANET.

3.2.2. Analisis Kebutuhan *Software*

1. Ubuntu 16.10

Sistem operasi Ubuntu ini akan digunakan untuk menjalankan beberapa aplikasi pendukung simulasi MANET seperti NS3 dan NetAnim. Alasan penggunaan Linux Ubuntu pada penelitian ini dikarenakan Linux memiliki *gcc compiler* dan menjadi *platform* yang direkomendasikan oleh NS3.

2. *Network Simulator 3* (NS3)

NS3 adalah aplikasi simulasi jaringan yang target utamanya adalah untuk riset di bidang jaringan dan sarana pembelajaran yang bersifat *Open Source*. NS3 pertama kali di publikasikan pada tahun 2006. Simulasi yang dibuat pada NS3 hampir sama dengan implementasi nyatanya yang bersifat *real-time*. Sehingga hasil yang dihasilkan pada simulasi ini nantinya akan hampir sama dengan yang nyata. Pada penelitian ini menggunakan NS3 versi ns-3.26 yang dirilis pada 3 Oktober 2016. Pada NS3 ketika merilis versi terbaru maka dokumentasi dan contoh simulasi yang terdapat pada Doxygen (*Web Documentation*) akan disesuaikan dengan versi yang dirilis. Sehingga ada kemungkinan simulasi yang berjalan pada versi yang lama tidak dapat digunakan pada versi terbaru dan sebaliknya simulasi yang berjalan pada versi terbaru tidak dapat digunakan pada versi yang lama. Hal ini disebabkan adanya pembaruan, perbaikan dan fitur-fitur terbaru pada

module di versi NS3 yang terbaru.

3. Sublime Text 3

Sublime Text 3 adalah sebuah *syntax editor* yang menggunakan *Python* API dikembangkan oleh Jon Skinner seorang programmer dari Australia. Sublime text mempunyai fitur plugin tambahan yang sangat memudahkan. Desain pada sublime text yang simpel dan bagus sehingga terkesan elegan untuk sebuah *syntax editor*. Sublime text juga memiliki kemudahan dalam *auto complete* dan *auto correct* dalam beberapa bahasa pemrograman salah satunya adalah C++.

4. NetAnim

NetAnim adalah aplikasi pembuatan animasi yang dibuat berdasarkan pada *Qt toolkit*. NetAnim membuat animasi dalam bentuk XML berdasarkan pada *trace file* yang dihimpun selama simulasi berjalan. NetAnim dikemas menjadi satu paket dengan NS3 dalam paket instalasi.

3.3. Perancangan Skenario

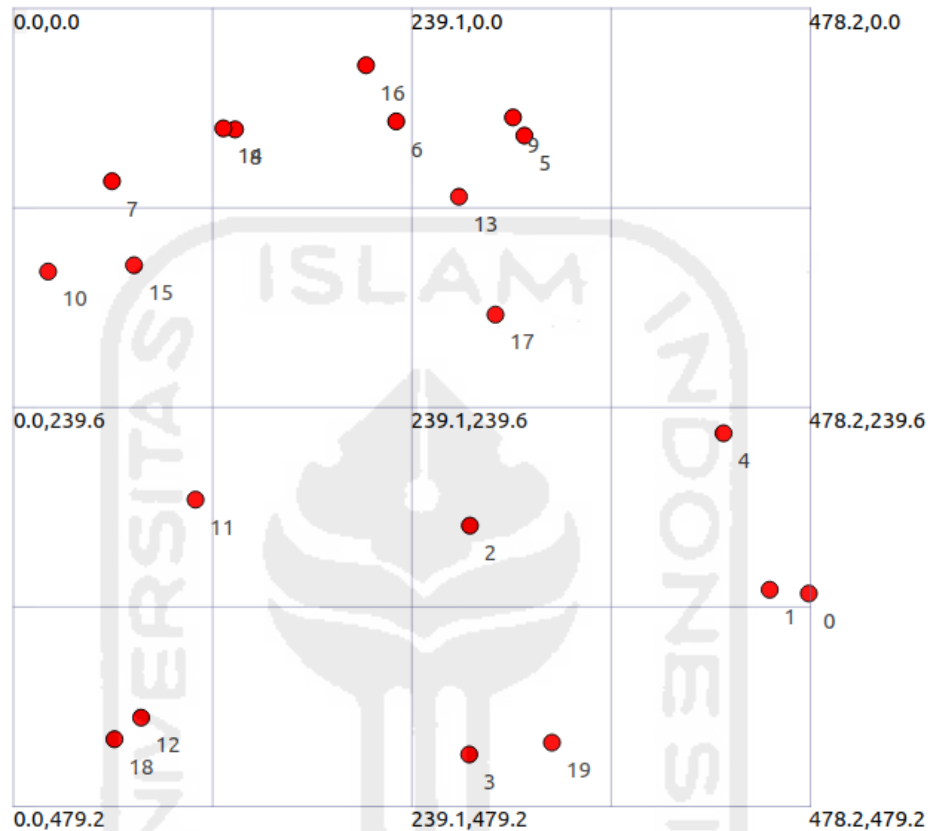
Simulasi pada MANET ini dibutuhkan suatu skenario sebagai pendukung pada saat implementasi. Skenario berguna untuk melihat gambaran dari simulasi yang ingin kita buat dan untuk memberi plot bagaimana pengaruhnya terhadap hasil analisis dari simulasi tersebut. Pada penelitian ini akan dibuat dua buah skenario dengan kondisi tertentu, yaitu:

1. Kondisi dimana luas area yang tetap dengan sebuah jaringan yang mengalami penambahan *node*.
2. Kondisi dimana luas area dan jumlah *node* yang tetap dengan ukuran paket data yang mengalami penambahan.

3.3.1. Skenario Penambahan Node

Pada skenario penambahan node akan dilakukan analisis hasil kerja parameter *QoS* seperti *throughput*, *PDR*, *packet loss* dan *delay* terhadap jaringan yang mengalami penambahan *node*. Kemudian dari hasil analisis parameter *QoS* di kedua protokol routing akan dibandingkan. Berikut contoh gambar yang diambil

dari hasil keluaran NetAnim untuk skenario dengan 20 buah *node*, seperti gambar dibawah:

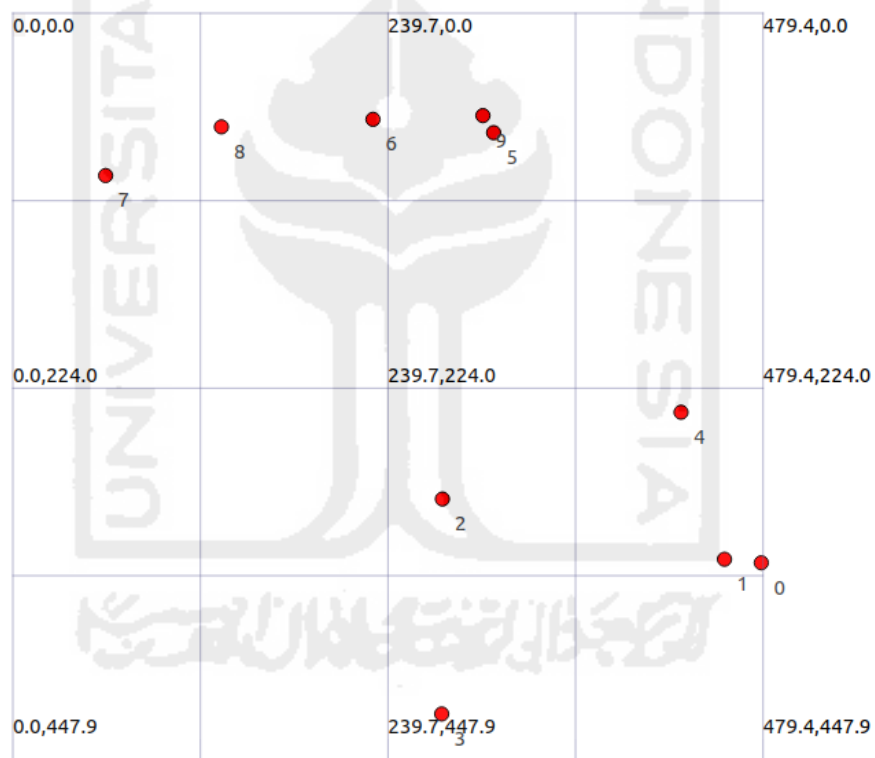


Gambar 3.2. 20 nodes

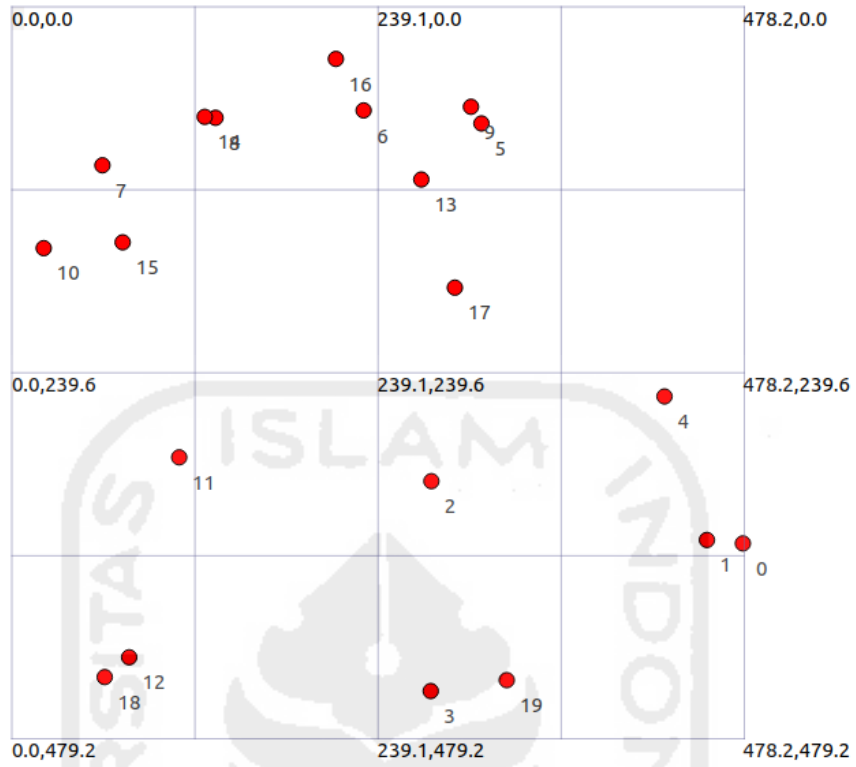
Dari gambar diatas terlihat 20 *node* yang tersebar pada suatu wilayah. *Node* pada wilayah tersebut dimulai dari index ke 0 sampai dengan index ke 19. *Node* index ke 0 memiliki alamat IP 10.1.1.0 dan seterusnya sampai dengan IP 10.1.1.19. Penambahan *node* pada routing protokol OLSR dan DSDV ini sangat mempengaruhi kinerjanya dari protokol tersebut karena membuat padatan pada suatu jaringan. Banyaknya *node* yang ada pada suatu wilayah menjadi salah satu sebab dari terjadinya padatnya suatu jaringan. Dengan luas wilayah yang sama dan semakin banyaknya *node* maka kepadatan jaringan semakin besar. Kepadatan jaringan berpengaruh pada routing tabel yang terbentuk untuk mengirimkan paket data ke suatu *node* dan itu berpengaruh pada proses pengiriman paket data ke semua

node.

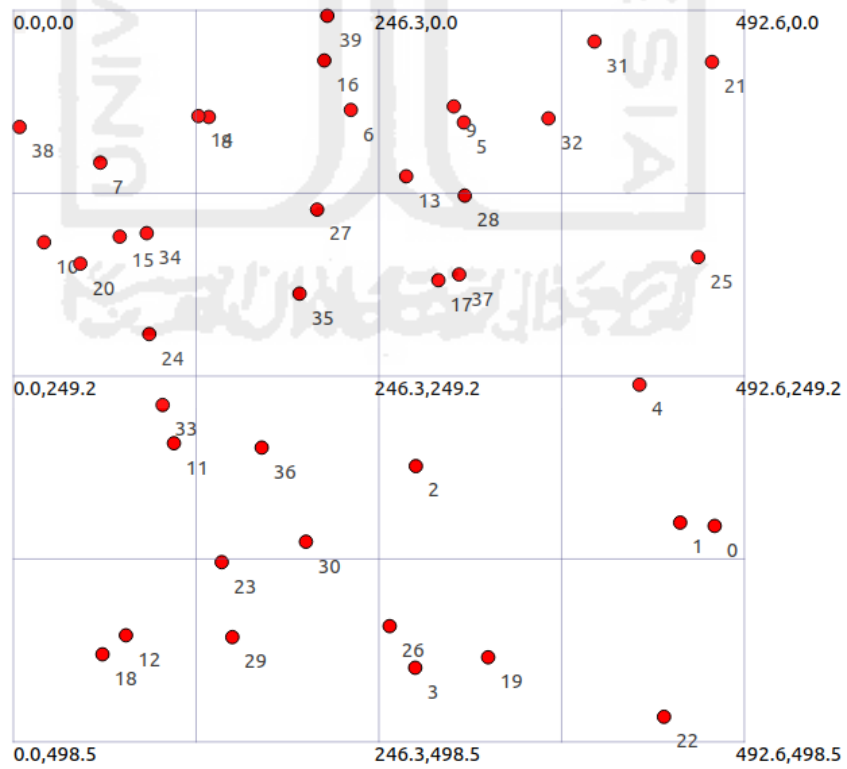
Pada simulasi ini akan dibuat beberapa skenario penambahan *node*, yaitu 10 *node*, 20 *node* dan 40 *node*. Dari penambahan *node* tersebut terbagi 3 *cluster*, yaitu *cluster* kecil untuk 10 *node*, *cluster* sedang untuk 20 *node* dan *cluster* besar untuk 40 *node*. *Node-node* yang berada di suatu wilayah ini akan tersebar secara random. Walaupun tersebar secara *random*, posisi awal *node* antara simulasi satu dengan simulasi berikutnya akan tetap sama. Hal tersebut dikarenakan NS3 memiliki fitur yang nilainya disimpan dalam variabel *streamIndex*. Hal ini dapat dilihat pada gambar dibawah ini:



Gambar 3.3. 10 nodes



Gambar 3.4. 20 nodes



Gambar 3.5. 40 nodes

3.3.2. Skenario Penambahan Ukuran Paket Data

Skenario ini bertujuan untuk menganalisis hasil kerja parameter *QoS* seperti *throughput*, *PDR*, *packet loss* dan *delay* terhadap jaringan yang mengalami penambahan ukuran paket data. Kemudian dari hasil analisis parameter *QoS* di kedua protokol routing akan dibandingkan. Dalam simulasi ini akan digunakan dua ukuran paket data yaitu *64 bytes* dan *128 bytes*. Karena pada skenario ini berfokus pada penambahan ukuran paket data, maka akan dipilih *20 node* yang merupakan jumlah *node* di *cluster* sedang.

3.4. Perancangan Simulasi

Perancangan simulasi yaitu melakukan eksperimen dengan menjalankan beberapa *script* yang tersedia di NS3 untuk membuat rancangan simulasi yang akan digunakan. Pada eksperimen ini dilakukan dengan mengambil referensi dari beberapa penelitian yang telah dilakukan sebelumnya dan dari contoh-contoh yang ada di dalam NS3. Pada beberapa penelitian sebelumnya yang juga membahas tema yang sama dengan penelitian ini, telah menghasilkan beberapa hasil analisis yang dapat menjadi referensi untuk membuat suatu perancangan. Maka beberapa parameter dalam simulasi ini dan skenario yang digunakan berbeda dari penelitian sebelumnya. Terdapat 2 tahapan yang dilakukan pada perancangan ini yaitu menentukan parameter simulasi dan tahapan-tahapan simulasi.

3.4.1. Parameter Simulasi

Parameter simulasi adalah nilai-nilai yang menentukan bagaimana simulasi ini akan berjalan dan nilai-nilainya juga berhubungan dengan beberapa komponen penting di dalam simulasi. Parameter yang berbeda antara simulasi satu dengan yang lainnya akan memberikan hasil keluaran nilai *QoS* yang berbeda. Sehingga nilai parameter pada simulasi ini merupakan variabel yang memiliki nilai konstan selama proses simulasi berjalan dan akan selalu digunakan untuk semua simulasi pada penelitian ini.

Tabel 3.1. Parameter simulasi

| Parameter | Nilai |
|-----------------------|--------------------------------------|
| Waktu simulasi | 100 detik |
| Pengiriman paket data | Dimulai pada detik ke 51 |
| Tipe kanal | <i>Wireless</i> |
| Transport Layer | <i>UDP</i> |
| Model pergerakan node | <i>Random Waypoint</i> |
| Kecepatan node | 10 m/s |
| Ukuran paket data | 64 <i>bytes</i> dan 128 <i>bytes</i> |
| Dimensi ruang | 500 x 500 m |
| <i>TX Power</i> | 5 |
| Koneksi | 4 |
| <i>Pause time</i> | 0 |

1. Waktu simulasi

Waktu simulasi adalah lamanya simulasi yang dijalankan. Waktu simulasi yang dipilih pada penelitian ini adalah 100 detik, lamanya waktu ini dirasa cukup untuk menjalankan simulasi MANET.

2. Pengiriman paket data

Pengiriman paket data adalah waktu dimana node-node melakukan pengiriman paket data. Waktu 50 detik atau setengah dari waktu simulasi dipilih untuk menyediakan waktu bagi routing protokol untuk melakukan *update* routing tabel.

3. Tipe kanal

Tipe kanal adalah media penghubung antar *node*, pada simulasi ini dipilih *wireless* karena OLSR dan DSDV merupakan routing protokol untuk MANET yang bersifat *mobile* atau bergerak secara dinamis.

4. *Traffic sources*

Traffic sources yang digunakan untuk mengirimkan paket data dalam simulasi ini adalah *UDP*.

5. Model pergerakan *node*

Model pergerakan *node* yaitu keadaan *node-node* yang akan bergerak selama simulasi. Dalam simulasi ini digunakan model *random waypoint* yaitu *node-node* yang akan bergerak secara acak. Akan tetapi walaupun *node-node* bergerak secara acak pergerakannya akan tetap sama untuk banyak *node* yang sama pada simulasi yang berbeda.

6. Kecepatan *node*

Kecepatan *node* adalah kecepatan *node* bergerak dari satu posisi ke posisi yang lain. Dalam simulasi ini dipilih kecepatan *node* 10 m/s.

7. Ukuran paket data

Ukuran paket data adalah besar paket yang dikirimkan. Dalam simulasi ini akan menjalankan skenario penambahan ukuran paket data, sehingga akan digunakan 2 ukuran paket data yaitu 64 *bytes* dan 128 *bytes*.

8. Dimensi ruang

Dimensi ruang adalah media tempat berlangsungnya simulasi dan ukurannya dengan satuan m. Dalam simulasi ini digunakan luas 500 x 500 m dirasa cukup untuk menampung 40 *node*.

9. *Tx Power*

Tx power adalah *power* atau daya yang digunakan dalam pengiriman paket data. Dalam simulasi ini dipilih nilai 5 untuk meminimalisir kemungkinan paket data yang hilang pada saat pengiriman.

10. Koneksi

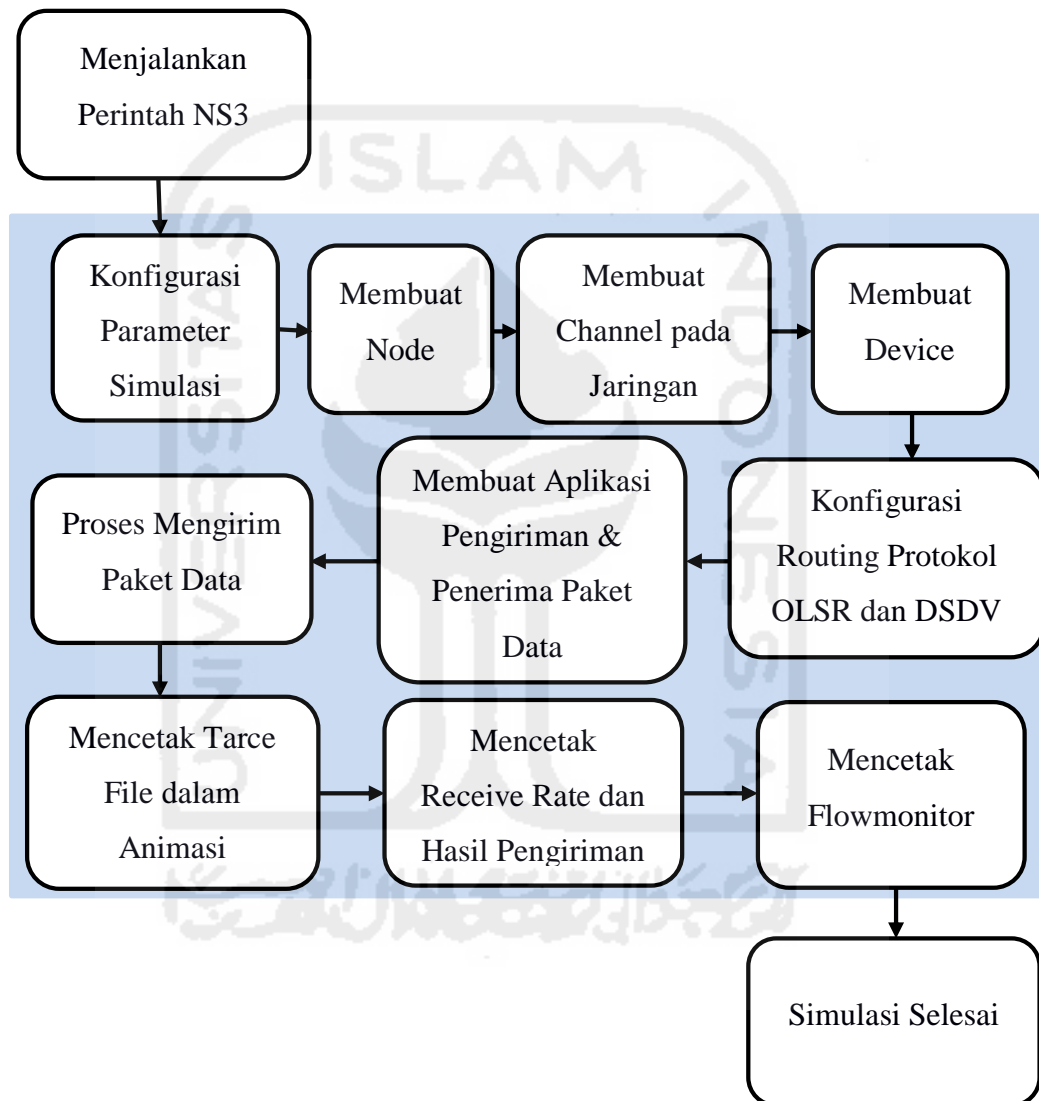
Koneksi adalah hubungan antara 2 *node* atau lebih yang saling berkomunikasi. Dalam penelitian ini koneksi jaringan di set sebanyak 4 koneksi, yang dirasa cukup untuk jaringan MANET 40 *node*.

11. *Pause time*

Pause time adalah waktu dimana *node-node* akan berhenti bergerak secara acak, kemudian *node* bergerak kembali sesuai kecepatan yang telah ditentukan. Dalam simulasi ini *Pause Time* di set 0, agar dari awal *node* terus bergerak dan kecepatan pergerakan *node* tetap sama pada jumlah *node* yang sama dengan simulasi yang berbeda.

3.4.2. Tahapan-Tahapan Simulasi

Pada tahapan-tahapan simulasi ini akan memaparkan poin-poin penting secara garis besar bagaimana langkah-langkah simulasi dijalankan. Berikut ini gambaran tahapan-tahapan simulasi:



Gambar 3.6. Tahapan-tahapan simulasi

1. Menjalankan perintah NS3

Pada tahap pertama, *user* akan menjalankan perintah untuk menjalankan *source script* yang ada di dalam NS3.

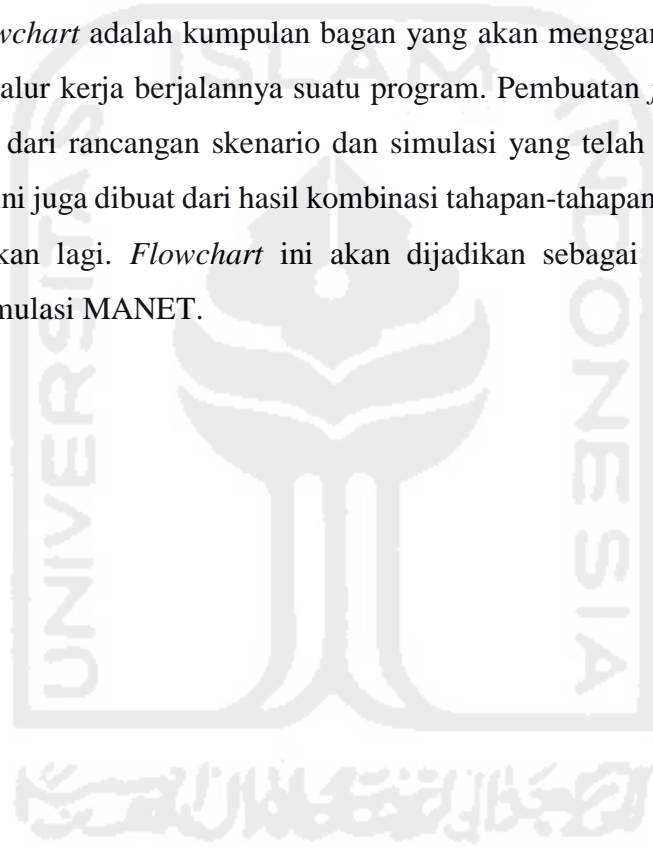
2. Konfigurasi parameter simulasi
Program akan melakukan pengaturan dan memberikan nilai untuk beberapa parameter yang bersifat statis.
3. Membuat *node*
Saat akan menjalankan perintah dengan *script* program terdapat parameter untuk menentukan banyaknya *node* pada simulasi. Setelah *node-node* dibuat pertama kali sesuai dengan yang diinginkan, kemudian dipasangkan aplikasi untuk pengiriman dan penerimaan paket data yang dihubungkan dengan *channel*.
4. Membuat *channel* pada jaringan
Channel dibuat untuk menghubungkan antar *node* dan sebagai media untuk pengiriman, perantaraan, penerimaan paket data.
5. Membuat *device*
Device dibuat untuk menghubungkan *node* dan *channel*. Selain itu, *device* juga berfungsi sebagai tempat *MAC* dan *IP address*.
6. Konfigurasi routing protokol OLSR dan DSDV
Pada tahap ini akan dipilih routing protokol yang akan digunakan dalam simulasi dan juga dilakukan pembuatan *routing table list*.
7. Membuat aplikasi pengiriman dan penerima paket data
Aplikasi berfungsi untuk melakukan pengiriman dan penerimaan paket data selama simulasi berjalan. Setiap *node* yang melakukan pengiriman dan penerimaan paket data akan dipasangkan aplikasi selama simulasi berjalan.
8. Proses mengirim pakat data
Saat simulasi dimulai beberapa *node* akan melakukan *broadcast* untuk membuat routing table kemudian node akan melakukan pengiriman paket data.
9. Mencetak *trace file* dalam animasi
Saat simulasi berjalan, program juga membuat *trace file* yang disimpan dalam bentuk animasi dengan ekstensi *.xml*. *File* animasi tersebut dapat dilihat dan dijalankan menggunakan NetAnim.
10. Mencetak *receive rate* dan detail pengiriman paket

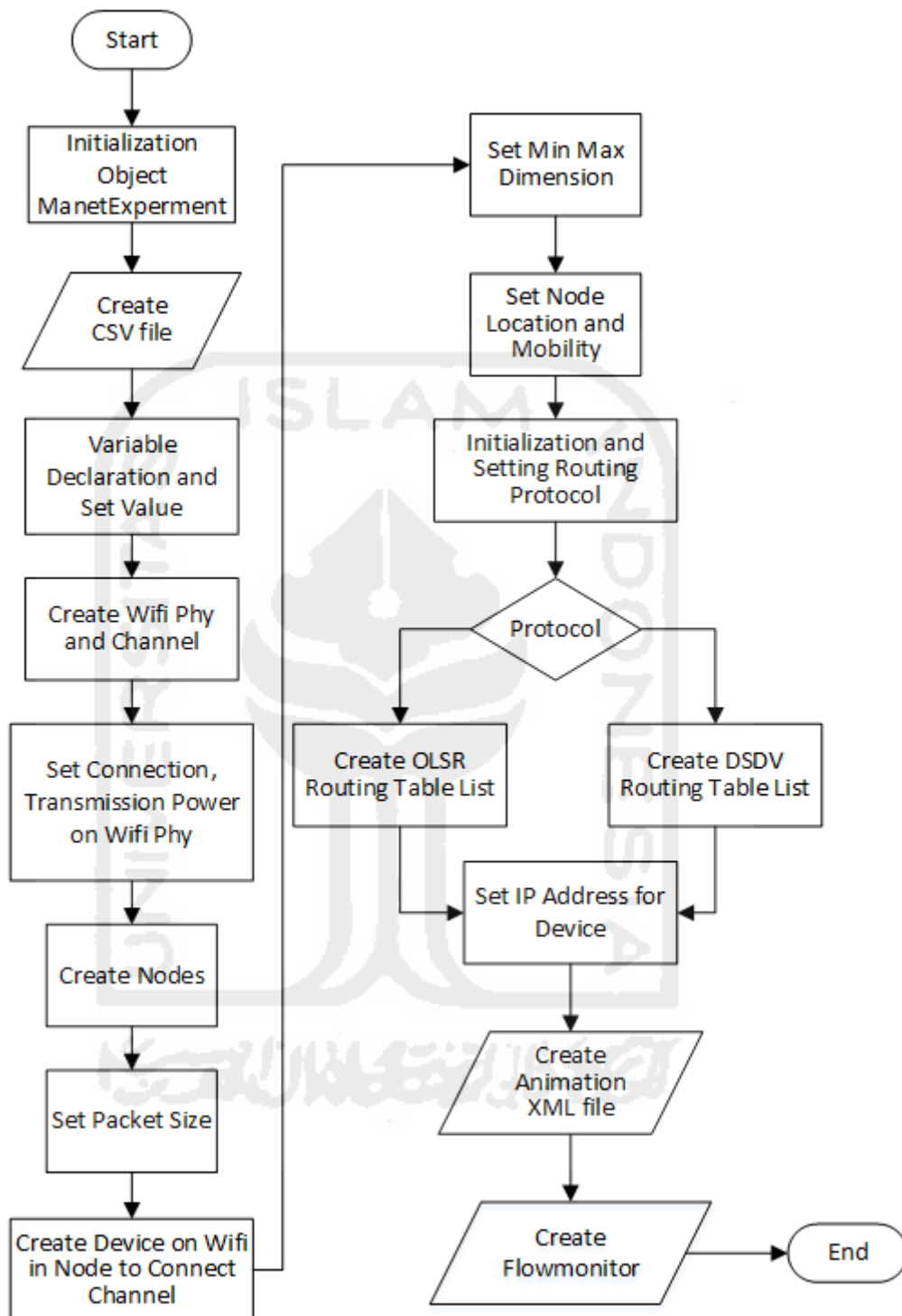
Kemudian, NS3 akan menghasilkan sebuah file berekstensi .csv yang berisi pengiriman data berupa *received rate*. *File* ini akan digunakan untuk menghitung throughput. NS3 juga menyediakan fitur untuk mencatat hasil pengiriman dan *delay* paket data kedalam sebuah *file* berekstensi .flowmon.

11. Simulasi selesai

3.5. **Flowchart Program Simulasi MANET**

Flowchart adalah kumpulan bagan yang akan menggambarkan bagaimana urutan dan alur kerja berjalannya suatu program. Pembuatan *flowchart* ini sangat dipengaruhi dari rancangan skenario dan simulasi yang telah dibuat sebelumnya. *Flowchart* ini juga dibuat dari hasil kombinasi tahapan-tahapan simulasi yang lebih dikembangkan lagi. *Flowchart* ini akan dijadikan sebagai rujukan pembuatan program simulasi MANET.





Gambar 3.7. Flowchart simulasi MANET

Saat ingin menjalankan simulasi maka user harus menuliskan perintah yaitu banyaknya *node* dan jenis routing protokol. Kemudian program akan menginisialisasi sebuah objek dari kelas *ManetExperiment*. Selanjutnya program

akan membuat file .csv dan juga mengisi tabel header di dalamnya. Program kemudian akan menset beberapa nilai default ke dalam beberapa variabel yang ada pada fungsi run.

Program kemudian akan mengkonfigurasi *wifi* yang akan digunakan sebagai media pengiriman dan penerimaan paket data dan juga membuat channel sebagai penghubung antar *node*. Program akan mengatur daya transmisi dan jumlah koneksi pada simulasi. Setelah itu program akan membuat *node* yang akan ditentukan oleh *user*. Selanjutnya program membuat *device*, dimana *device* ini akan di install pada masing-masing *node*, *device* digunakan untuk menghubungkan antara *node* dan *channel*.

Program akan membuat sebuah dimensi wilayah yang akan digunakan sebagai tempat melakukan simulasi. *Node-node* akan ditempatkan di dimensi tersebut dan diatur pergerakannya.

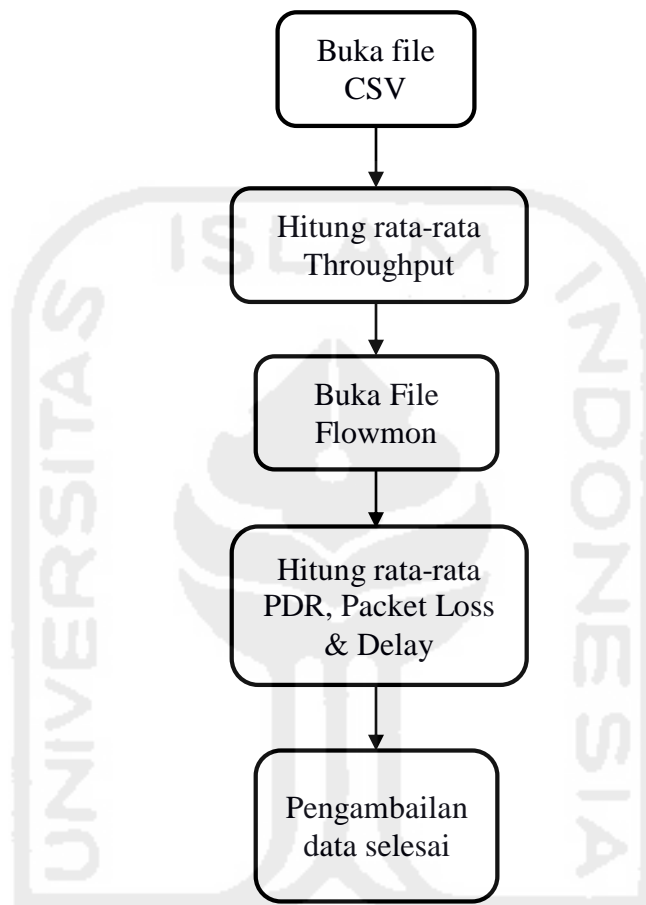
Program akan melakukan inisialisasi routing protokol dan membuat routing tabel berdasarkan protokol yang dipilih saat akan menjalankan perintah simulasi program, kemudian akan melakukan pengaturan range *IP address* dan memberikan *IP* pada masing-masing *node*. Kemudian program akan mengatur kapan aplikasi mulai mengirim dan berhenti mengirim data.

Selanjutnya program akan membuat *trace file* dengan format .tr. Program juga akan mencetak berapa besar bit dan paket data yang diterima oleh *node* tujuan dan dikelompokkan setiap detik. Program akan membuat *file* animasi dalam format .xml yang dihasilkan dari *trace* mobilitas *node*.

Kemudian program akan memasuki bagian pengumpulan data untuk menghitung *delay* dan pengiriman paket. Awalnya program akan menentukan lokasi dan pergerakan *node*, kemudian program akan menentukan *node* tujuan dan sumber yang akan mengirimkan data. Program akan men-generate *traffic* berupa banyaknya paket data yang akan dikirimkan. Program akan melakukan konfigurasi *FlowMonitor* untuk menghitung delay dan pengiriman paket. Kemudian program akan mulai melakukan perhitungan dimulai dari perhitungan paket dikirim, paket diterima, serta minimum, maksimum, total dan rata-rata delay. Lalu program akan mencatat hasil tersebut kedalam *file* yang berekstensi .flowmon. Program selesai

dan simulasi berhenti.

3.6. Tahapan-Taapan Pengambilan Data



Gambar 3.8. Tahapan-tahapan pengambilan data

Untuk melakukan pengambilan data, terdapat 2 *file* yaitu *CSV* dan *flowmon* sebagai rujukan untuk menghitung *throughput*, *PDR*, *packet loss*, dan *delay*. Tahap pertama dalam menganalisis data adalah dengan membuka *file CSV* yang digunakan untuk mencari nilai rata-rata *throughput*.

Kemudian menghitung rata-rata *throughput* dari nilai-nilai yang ada di dalam *file CSV*. Selanjutnya membuka *file flowmon* yang digunakan untuk mencari nilai rata-rata dari *PDR*, *packet loss*, dan *delay*. Kemudian menghitung rata-rata nilai *PDR*, *packet loss* dan *delay*. Setelah menghitung maka akan didapat nilai rata-rata *throughput*, *PDR*, *packet loss* dan *delay*. Pengambilan data selesai.