

BAB III

PEMODELAN SISTEM

3.1 Objek Penelitian

Pada penelitian ini akan dibuat simulasi skenario *load balancing* pada eNB di *network* RAN LTE. Tujuan pembuatan simulasi ini untuk mensimulasikan penyeimbangan beban trafik pada eNB *congested* dengan mendistribusikan beban trafik pada eNB *neighbor* yang berstatus *not congested* . pembuatan simulasi menggunakan software NS-3 dengan OS Ubuntu Xenial Xerus.

3.2 Alat dan Bahan

3.2.1 Perangkat Simulasi

Penulis menggunakan Notebook Acer untuk menjalankan simulasi berspesifikasi sebagai berikut :

Notebook	: Acer E5-471
CPU	: Intel Core i5 1.7 GHz
RAM	: 8GB
Harddisk	: 500 GB
VGA	: Nvidia Geforce 820M
Sistem Operasi	: Ubuntu 16.04 Xenial Xerus 64bit

3.2.2 Software

A. Ubuntu 16.04 Xenial Xerus

Ubuntu mengeluarkan *Operating System* versi terbaru yaitu 16.04 pada tanggal 21 April 2016 dengan *codename Xenial Xerus* . OS versi terbaru ini menggantikan versi lama Ubuntu 14.04 yang di release pada tahun 2014. Pada os ini terjadi beberapa peningkatan dari versi sebelumnya yaitu sudah terinstal langsung *software* Firefox 45, Thunderbird 38, LibreOffice 5.1, File 3.14.2, Video 3.18, Rhythmbox 3.3, Terminal GNOME 3.18, Eye of GNOME 3.18,

Shotwell 0.22, dan masih banyak lagi. Dan, kita dapat dengan mudah memasang aplikasi apa pun yang diinginkan.

Selain itu ini adalah rilis LTS. Itu berarti OS ini mendapat dukungan dan pemeliharaan selama lima tahun oleh Canonical selaku pengembang, agar senantiasa aman, handal, stabil, dan sekaligus berdampak pada penghematan biaya dan waktu, terutama untuk desktop dengan penyebaran skala besar, baik pada pemerintahan, sektor publik dan perusahaan.

B. NS-3.26

NS-3 atau Network Simulator 3, ialah sebuah *software* simulasi yang bersifat *discrete event network* yang banyak digunakan untuk mempelajari *dynamic nature* dari komunikasi jaringan meliputi Simulasi menggunakan kabel maupun *wireless network (routing algorithms, TCP, UDP, dll)* dapat dilakukan menggunakan NS3. Target utama NS-3 adalah untuk mendukung aktifitas pembelajaran dan riset di bidang Jaringan. *Project NS-3* dimulai pada bulan Juli 2006 dan rilis pertama dilakukan pada 30 Juni 2008.

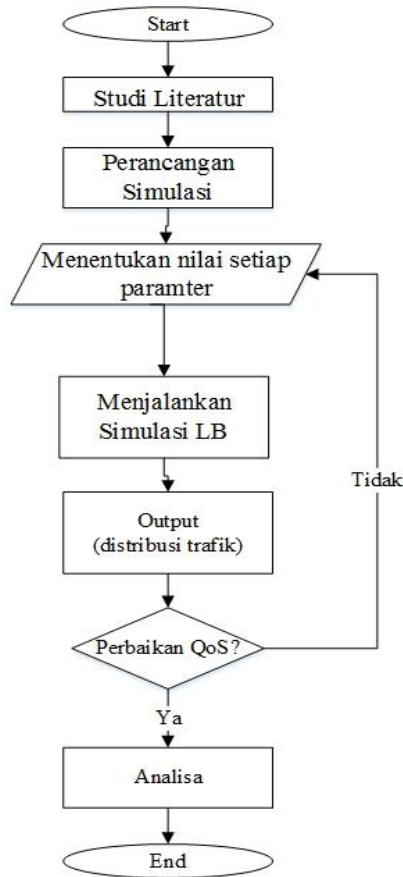
NS3 menggunakan Bahasa pemrograman C++ atau Python. Dilihat dari struktur penulisan bahasa, C++ pada NS3 memang terlihat sedikit lebih rumit daripada Bahasa TCL yang digunakan di NS-2. Akan tetapi NS3 lebih mudah digunakan karena sudah berisi *library* yang dapat di panggil sesuai kebutuhan. Infrastruktur NS-3 dapat mendukung pengembangan Simulasi yang hampir mendekati kondisi sesungguhnya dan memiliki karakteristik mendekati emulator Jaringan. Disamping itu NS-3 dapat digunakan sebagai Simulator untuk Jaringan berbasis IP dan non-IP

3.3 Metodologi Penelitian

Tahapan pertama adalah melakukan studi literatur untuk mempelajari parameter – parameter yang digunakan seperti *RSRP, SINR, RSRQ, Hysteresis, Time to Trigger* dan lainnya. Selain itu penulis juga mempelajari hubungan dan efek perubahan nilai dari suatu parameter yang bisa mempengaruhi skema simulasi yang penulis buat. Tahapan kedua adalah melakukan perancangan simulasi seperti merancang interaksi antar perangkat,

menentukan skenario yang akan digunakan pada simulasi. Simulasi *load balancing* menggunakan skenario *Pareto efficiency*. NS-3 karena menyediakan *tool* yang sangat lengkap untuk LTE sehingga mendukung penulis dalam menjalankan simulasi *load balancing*. Tahapan ketiga adalah menentukan nilai dari parameter *handover* yang akan digunakan untuk dijalankan pada simulasi agar seluruh komponen dapat berinteraksi satu sama lain sehingga proses *load balancing* dapat dilakukan. Tahapan ke empat yaitu menjalankan simulasi yang telah dibuat. Pada tahapan ini penulis melakukan proses *running* simulasi sampai didapat *output* yang diinginkan.

Tahapan kelima yaitu mengamati *output* yang dihasilkan dimana dari *output* yang dihasilkan dapat dilihat beberapa parameter yang akan dijadikan pertimbangan dilakukannya *load balancing* trafik seperti total trafik yang dihasilkan pada tiap -tiap eNB dan sebagai acuan pemilihan sebuah UE (user equipment) apakah layak untuk di *handover* atau tidak. Selanjutnya dilihat trafik yang ada pada sebuah eNB apakah telah berhasil didistribusikan atau belum terdistribusi. Jika belum berhasil terdistribusi maka *throughput* yang didapat oleh setiap UE pada eNB yang *overload* cenderung rendah akibat masih tingginya *load* trafik pada eNB tersebut. Maka dari itu harus mengulang kembali proses *running* simulasi dan mengubah beberapa nilai dari parameter yang digunakan. Sebaliknya apabila trafik telah berhasil terdistribusi maka akan didapat peningkatan rata – rata *throughput* yang didapat dari eNB. Tahapan terakhir adalah melakukan analisa dari hasil *running* simulasi. *Output* yang dianalisa (*penetapan kenaikan power pada eNB target*), peningkatan *throughput*, dan pemerataan *throughput* yang diterima oleh UE. Untuk lebih jelasnya bisa dilihat di flowchart pada gambar 3.1.



Gambar 3.1 Flowchart program skripsi

3.4 Perancangan Sistem

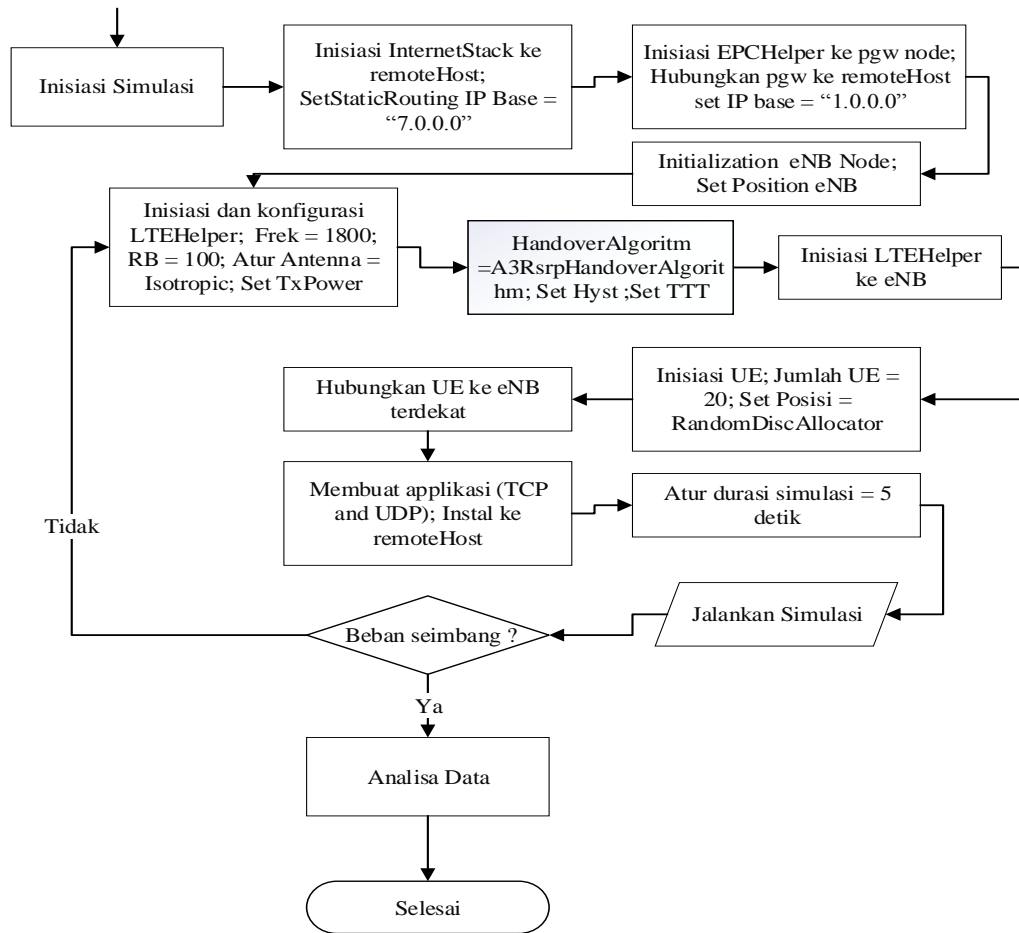
Simulasi yang dirancang oleh penulis terletak di sisi E-UTRAN yang terdiri dari 3 eNodeB yang diberikan daya yang berbeda. eNodeB digunakan sebagai penghubung antara UE ke EPC dan *Internet Host* sebagai penyedia layanan. UE akan disebar secara acak kepada 3 eNodeB tersebut, tujuannya adalah untuk memunculkan ketidak merataan trafik pada 3 eNodeB tersebut, dengan itu pula penulis akan melakukan penyeimbangan trafik atau *load balancing* pada 3 eNode B tersebut agar didapatkan trafik yang optimal. Penuli menggunakan algoritma efisiensi pareto untuk me-*load balancing* trafik yang muncul.

Tabel 3.1 Parameter yang digunakan pada simulasi

Parameter	Keterangan
Jumlah eNodeB	3
Antena	Omni Directional, SISO
Bandwidth	20 Mhz
RB	100
Jumlah UE	20
Jarak Antar eNodeB	2000 Meter
Frekuensi	1800 Mhz
Handover Event	A3
EPC	1
Internet Host	1
Packet Size VoIP	96 Kbps
Trafik data	TCP dan UDP

Perancangan simulasi menggunakan software NS3, yaitu sebuah *discrete event network simulator* yang digunakan untuk mempelajari kondisi real pada komunikasi jaringan baik itu komunikasi yang menggunakan media kabel atau wireless. NS3 menggunakan bahasa pemrograman C++ atau python. Infrastruktur NS3 sendiri mendukung pengembangan simulasi yang mendekati kondisi aslinya dan mempunyai karakteristik mendekati emulator jaringan. Selain itu kelebihan NS3 lainnya dapat digunakan sebagai simulator untuk jaringan berbasis non-IP dan yang berbasis IP.

3.5 Flowchart System



Gambar 3.2 *Flowchart* perancangan Simulasi

Berdasarkan gambar 3.2 *flowchart* perancangan model simulasi yang menggunakan *software* NS-3 yang menggunakan Bahasa pemrograman C++ untuk membuat dan menkonfigurasi sistem *network*-nya. Berikut ini penulis akan mendeskripsikan langkah – langkah konfigurasi simulasi *load balancing* pada *software* NS-3:

1. Langkah pertama adalah mengkonfigurasi bagian EPC (*Evolved Packet Core*) yang terdiri dari P-GW (*Packet Data Network Gateway*) hal ini dikarenakan P-GW berfungsi untuk menghubungkan jaringan LTE ke penyedia layanan seperti internet atau pada NS-3 adalah *Internet Host Container*.
2. Langkah kedua adalah mengkonfigurasi P-GW dengan mengatur *IP base* IPv4 7.0.0.0 lalu hubungkan ke internet host container dengan *IP*

base 1.0.0.0 menggunakan *channel PointToPoint*. Selanjutnya atur *route default* dari *eNodeB* dan UE ke P-GW agar paket yang nantinya dikirim oleh UE ataupun *eNodeB* akan mengarah secara *default* ke P-GW.

3. Langkah ketiga adalah mengkonfigurasi *interface* atau *channel* yang menghubungkan P-GW ke *Internet Host Container* yaitu dengan menggunakan *class PointToPoin*. *Channel Point to Point* adalah *interface logic* yang berfungsi untuk menghubungkan langsung P-GW dengan *Internet Host Container*. Lalu konfigurasi beberapa atribut pada *channel Point to Point* seperti *data rate*, *MTU* dan *delay*. Pada simulasi yang penulis buat *interface PointToPoin* dikonfigurasi dengan maksimum *data rate* mencapai 100Gbps, MTU sebesar 1500, *delay* 0.010 *second* dengan *IP base* 1.0.0.0. Perlu diketahui di NS-3 pemberian IP pada setiap node bersifat dinamis penulis hanya mengatur *IP base*-nya saja.
4. Langkah keempat adalah membuat *eNodeB* beserta sistemnya. Penulis menggunakan 3 *eNodeB* pada skenario yang akan dibuat. *NodeContainer enbNodes* perintah ini digunakan untuk membuat *node* yang diberi nama *enbNodes*. Pada awalnya sebuah *node* tidak memiliki fungsi apa – apa sama seperti sebuah komputer yang tidak di install OS, untuk itu penulis menginstal *LTEHelper* pada *enbNodes* bertujuan agar penulis dapat mengkonfigurasi sistem pada *eNodeB*. Lalu set posisi ketiga eNB tersebut,
5. Langkah kelima membuat *LTEHelper*, *class* ini digunakan untuk mengkonfigurasi sistem yang berada pada *eNodeB*. Pertama yaitu mengkonfigurasi antena, penulis menggunakan antena jenis *isotropic* agar dapat memancarkan gelombang elektromagnetik kesegala arah. Selanjutnya mengkonfigurasi *handoverAlgorithm* penulis menggunakan *event A3* dengan beberapa atribut parameter ditambahkan seperti *Hysteresist* dan TTL. Tujuan penulis menggunakan *HandoverAlgoritm* supaya UE dapat berpindah secara otomatis ke *eNodeB* lain apabila didapatkan RSRP yang lebih tinggi dibandingkan RSRP dari *eNodeB Serving*.

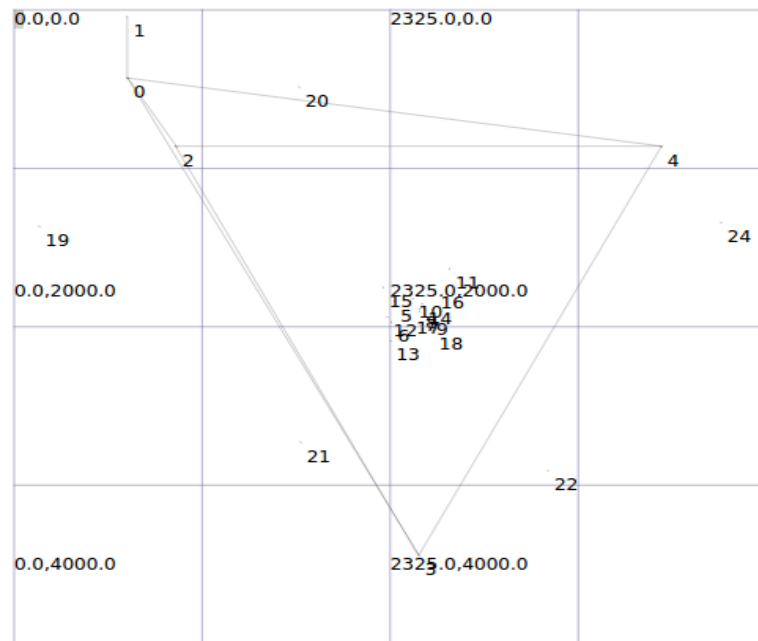
6. Langkah keenam Mendeklarasikan UE menggunakan *Script NodeContainer ueNodes*. Setelah itu install sistem untuk UE dengan menggunakan perintah *NetDeviceContainer ueLteDevs*. Tidak banyak konfigurasi yang dilakukan pada sisi UE hanya menetapkan konfigurasi IP agar UE dapat alokasi IP untuk terhubung ke jaringan.
7. Langkah ketujuh adalah menghubungkan UE dengan *eNodeB* menggunakan *script AttachToClosestEnb* agar UE dapat terhubung ke *eNodeB* dengan jarak terdekat.
8. Langkah kedelapan adalah membuat *application*. Pembuatan *application* bertujuan untuk memberikan layanan pada jaringan yang akan dibuat oleh penulis. Layanan yang digunakan berupa hanya berupa *Voice over IP* yang menggunakan *protocol* TCP dan UDP. *Application* ini nantinya akan di install pada *node Internet Host Container*.
9. Langkah kesembilan tinggal meng-*compile script* yang telah selesai dibuat. Nantinya *output* yang dihasilkan berupa *data trace file* yang masih harus diolah menggunakan *software* pengolah angka seperti *Libre Office* atau *Microsoft Excel* agar didapatkan nilai dari parameter yang dicari.
10. Langkah terakhir adalah menganalisa data *trace file* dari simulasi untuk dihitung. Setelah itu apabila hasil perhitungannya mendapatkan total *throughput* pada setiap eNB *balance* maka tinggal dilakukan Analisa. Akan tetapi kalau data yang didapat ternyata tidak *balance* maka simulasi akan diulang ke langkah 5 yaitu mengatur ulang parameter *TxPower* dan *Hysteresist*.

3.6 Skenario Simulasi

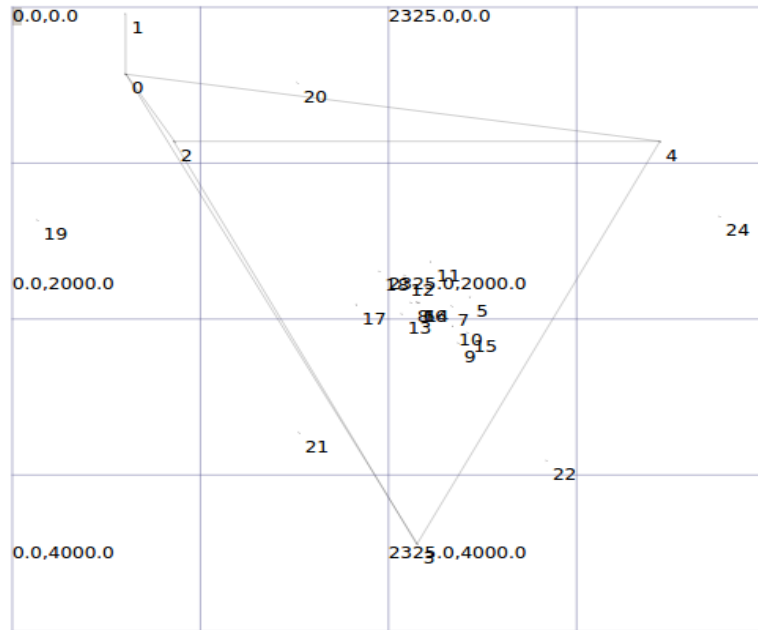
Penulis membuat dua simulasi sebelum *Load Balancing* dan sesudah *Load Balancing* yang dicoba sampai sepuluh kali. Setiap percobaan yang penulis lakukan mempunyai perbedaan pada jarak antar *UE* dengan *eNB* dan beberapa parameter dilakukan penyesuaian untuk mendapatkan total *throughput* pada setiap eNB menjadi seimbang. Untuk membedakan jarak antar *UE* pada sebuah skenario yang dijalankan penulis menggunakan fungsi *RngSeedManager* yaitu

sebuah kelas pada *software* simulasi NS-3 yang berfungsi untuk mengacak jarak antar UE pada suatu skenario yang telah dibuat. Konfigurasi eNB pada masing – masing simulasi memiliki nilai yang berbeda mengacu pada tabel 3.2.

Perbedaan jarak antar UE pada setiap skenario menyebabkan terjadinya perbedaan *throughput* per-UE pada setiap skenario simulasi hal ini juga menyebabkan konfigurasi *TxPower* yang digunakan pada eNB mengalami perbedaan di setiap skenario dengan tujuan untuk mendapatkan keseimbangan beban *peak throughput downlink* pada setiap eNB. Berikut akan ditampilkan efek dari penggunaan kelas *RngSeedManager* pada salah satu skenario yang telah dibuat oleh penulis.



Gambar 3.3 Topologi jaringan pada Skenario 1 dengan nilai *RngSeed* = 1



Gambar 3.4 Topologi jaringan pada Skenario 10 dengan nilai $RngSeed = 80$

Tabel 3.2 Konfigurasi parameter pada skenario *Mobility Load Balancing*

Skenario	eNB	TxPower sebelum LB	TxPower sesudah LB	Hysteresist	Rng Seed	Durasi Simulasi
1	eNB 1	43 dBm	43.5 dBm	-	1	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.57 dBm	-		
2	eNB 1	43 dBm	43.6 dBm	-	5	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.51 dBm	-		
3	eNB 1	43 dBm	43.29 dBm	0.5 dB	10	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.49 dBm	-		
4	eNB 1	43 dBm	43.73 dBm	-	20	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43 dBm	1 dB		
5	eNB 1	43 dBm	43.46 dBm	-	30	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.6 dBm	-		
6	eNB 1	43 dBm	43.71 dBm	-	40	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.9 dBm	-		
7	eNB 1	43 dBm	43 dBm	-	50	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.6 dBm	-		
8	eNB 1	43 dBm	43 dBm	0.5 dB	60	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.49 dBm	-		
9	eNB 1	43 dBm	43.2 dBm	-	70	5 Second

	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.49 dBm	-		
10	eNB 1	43 dBm	43.45 dBm	-	80	5 Second
	eNB 2	43 dBm	43 dBm	-		
	eNB 3	43 dBm	43.5 dBm	-		

3.7 Parameter Pengujian

Parameter yang akan dilakukan pengujian meliputi RSRP, *Hysteresist*, *Time to Trigger* dan *Event A3* sebagai handover algorithm-nya dan parameter pendukung pengambilan keputusan RSRQ dan SINR sedangkan keluaran yang dihasilkan simulasi berupa nilai *throughput downlink*.

3.8 Metode Pengujian

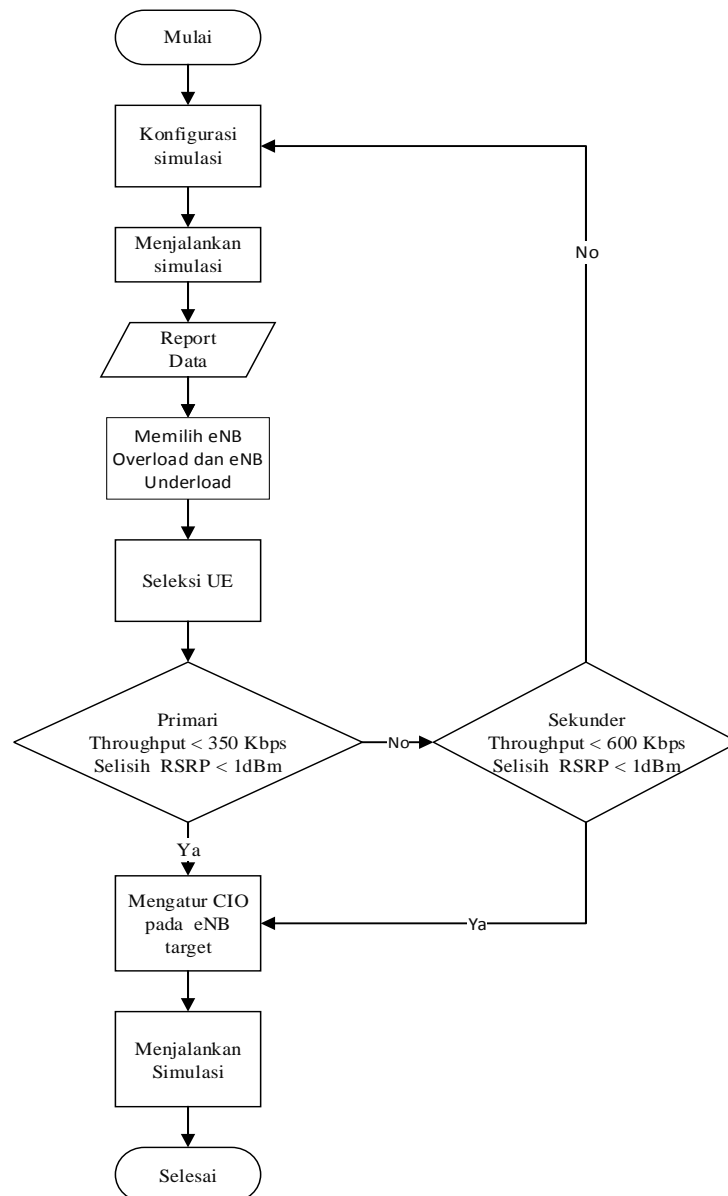
Skenario simulasi menggunakan prinsip *pareto efficiency*. Prinsip *pareto efficiency* banyak digunakan pada bidang ekonomi bertujuan untuk menyeimbangkan suatu keadaan agar menjadi lebih ideal. Kondisi ideal hanya dapat dicapai jika ada salah satu komponen yang dominan diperburuk agar komponen lain terlihat lebih baik. Prinsip *pareto* mengurutkan masalah yang terjadi dimulai dari yang paling penting sampai yang kurang penting, agar nantinya perhatian dapat dipusatkan ke masalah paling penting yang memiliki dampak paling besar daripada memperhatikan masalah yang kurang penting yang tidak menimbulkan dampak terlalu besar.

Untuk menyelesaikan suatu masalah menggunakan prinsip *pareto efficiency*, hal pertama yang harus dilakukan adalah mengidentifikasi elemen – elemen yang dominan kaitanya dengan simulasi yang penulis buat adalah *throughput* pada masing – masing UE. contohnya ketika eNB 2 satu memiliki kepadatan trafik yang tinggi yang disebabkan banyaknya user yang *servicing* pada eNB tersebut menyebabkan rendahnya *throughput downlink* yang didapatkan oleh UE. Maka untuk memperbaiki keadaan pada eNB 2 adalah dengan melakukan *trade off* UE ke eNB *neighbor*. Pada simulasi yang penulis buat cara untuk melakukan *trade off* UE ialah dengan meng-*handoff* UE yang berada pada eNB *servicing* ke eNB *neighbor* sehingga akan didapatkan kondisi yang ideal

yang akan menyebabkan terjadinya pemerataan beban dan peningkatan *throughput* yang didapat oleh UE .

Agar dapat dilakukanya pendistribusian trafik yaitu dengan cara mengkonfigurasi parameter *TxPower* dan *Hysterisis* yang berada pada eNB *neighbors* agar nilai RSRP dari eNB *neighbor* menjadi lebih baik dibandingkan nilai RSRP dari eNB *-serving*, sehingga UE akan melakukan *handover* ke eNB *neighbor*. kejadian tersebut masuk ke *event A3* dimana eNB *neighbors* terlihat lebih baik dibandingkan eNB *-serving*.

3.9 Algoritma Pengujian



Gambar 3.5 Flowchart pengujian

1. **Mulai**

2. **Konfigurasi Simulasi**

Parameter yang dikonfigurasi pada tahapan ini yaitu Txpower disamakan pada setiap eNB dan RNG Seed.

3. **Menjalankan Simulasi**

4. **Output Data**

Pada tahapan keempat dilakukan pengamatan terhadap data hasil simulasi yaitu berupa throughput yang didapat oleh UE ataupun throughput yang dihasilkan oleh eNB.

5. **Cari eNB *underload* dan *Overload***

Pada tahapan ini mengacu pada tahap empat untuk mengidentifikasi eNB *congested* atau *highly loaded* dan eNB *less loaded*. eNB *congested* biasanya memiliki total trafik terendah dikarenakan beban yang tinggi sehingga mempengaruhi QOS eNB tersebut, untuk itu akan di kurangi beban trafiknya agar didapatkan QOS yang lebih baik dengan cara meng-*handover* UE ke eNB *less loaded*. Selanjutnya eNB *less loaded* yaitu eNB yang memiliki total trafik tertinggi dan beban yang rendah sehingga didapatkan QOS yang bagus jika dibandingkan eNB *congested*, oleh karena itu eNB ini akan dijadikan target *handover* meskipun dampaknya akan mengurangi kualitas QOS pada eNB *less loaded* karena adanya penambahan beban baru.

6. **Pemilihan UE**

Pada tahapan ketiga ini dilakukan pengambilan keputusan didasarkan pada prinsip pareto efisiensi yaitu kondisi dimana tidak akan ada kenaikan pada satu variabel tanpa memperburuk variabel lainnya. Kaitanya dengan skripsi ini yaitu dengan memindahkan atau meng-*handover user* yang berada pada ujung eNB *congest (highly loaded)* ke eNB *neighbours (less loaded)* agar didapatkan peningkatan *throughput* pada *user (UE)* yang berada pada eNB *congest*. Selain itu dilakukan pemilihan UE untuk di

handover berdasarkan *report* nilai RSRP yang diterima oleh UE, *Throughput* total pada setiap eNB yang dihasilkan dari *running* simulasi pertama pada software NS-3. Penulis menggolongkan kedalam dua kategori UE yang memenuhi kriteria untuk di *handover* yaitu sebagai berikut;

a. **Primer**

Throughput yang didapatkan UE harus dibawah 350 Kbps.

Margin RSRP antara *eNB underload* dengan *eNB overload* harus dibawah 1dB.

b. **Sekunder**

Throughput yang didapat lebih dari 350 Kbps dan kurang dari 600 Kbps

Margin RSRP antara *eNB underload* dengan *eNB overload* harus dibawah 1dB.

7. **Set Cell Individual Offset (CIO) pada Target eNB**

Pada tahapan keempat dilakukan perubahan nilai CIO pada eNB Target .[10] Penambahan nilai CIO pada eNB Target sebesar 0 sampai maksimal 1 dB, dikarenakan setiap iterasi posisi UE selalu berpindah.[13]

8. **UE Handover**

UE akan *handover* jika didapatkan kondisi RSRP dari target lebih bagus dari eNB serving sehingga memenuhi keadaan (2.8)

A3 handover

1. UE, $RSRP_S > RSRP_T$

If selisih $RSRP_T$ (0 – 1 dBm)

Else kembali ke 1

2. Set CIO in $RSRP_T$ (0 – 1 dBm)

3. If $RSRP_S + Hys_0 + a3Offset_0 < RSRP_T + CIO_1$

Else kembali ke 1

4. UE Handover

9. **Finish**