

BAB III

METODOLOGI PENELITIAN

3.1. Subjek dan Objek Penelitian

Efektivitas protokol routing DYMO dan DSDV adalah subjek dari penelitian ini. Tujuan penelitian ini adalah untuk mencari hasil dari *Throughput* rata-rata, latensi end-to-end rata-rata, rasio pengiriman paket rata-rata, dan pesan kontrol rata-rata yang dihasilkan.

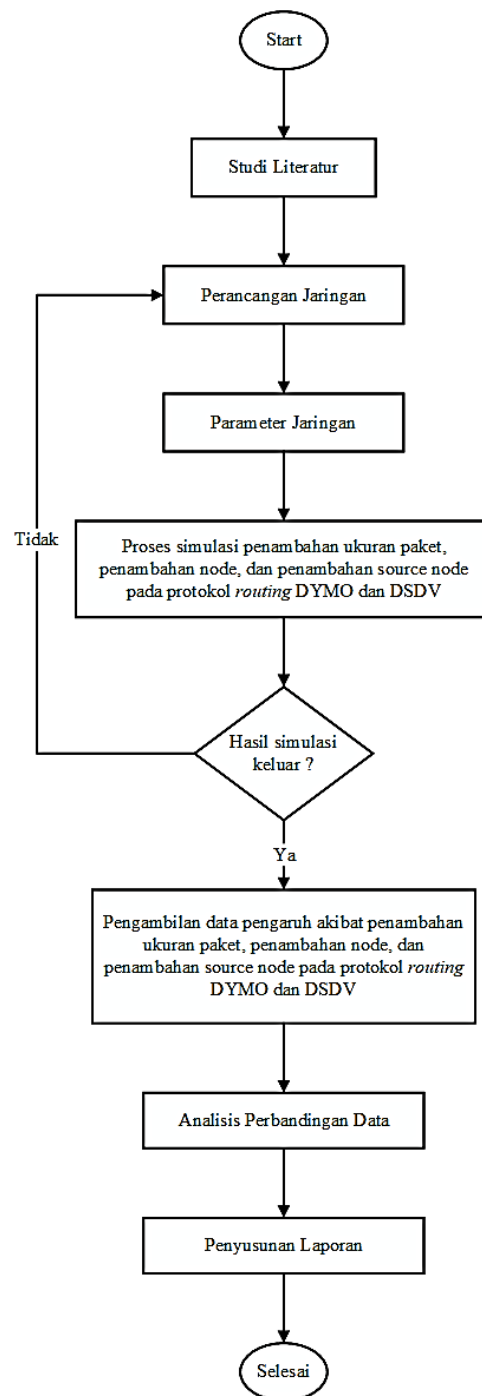
3.2. Alat dan Bahan Penelitian

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut :

1. *Hardware* yang digunakan dalam penelitian ini adalah Laptop yang menggunakan tipe processor AMD Ryzen 3 3200U dengan Radeon Vega Mobile Gfx (4 CPUs) 2.6GHz, RAM 8 Gigabyte yang memiliki ruang penyimpanan 1 Terabyte SATA HDD dan 256 Gigabyte SSD, serta sistem operasi Windows 10.
2. *Software* yang digunakan adalah OMNeT++ 6.0 IDE yang telah diinstal *framework* INET v2.0 di dalamnya.

3.3. Diagram Alir Penelitian

Pelaksanaan tugas akhir ini menggunakan informasi dan data untuk menghasilkan laporan yang dilakukan secara bertahap, dimulai dengan melakukan penelitian literatur, desain jaringan, parameter jaringan, prosedur simulasi, pengambilan data, analisis data, dan penyusunan laporan. Dalam sub-bab desain program, langkah desain jaringan akan dibahas secara lebih rinci.



Gambar 3.1 *Flowchart* Alir Penelitian

3.3.1. Studi Literatur

Tahap pertama peneliti mencari berbagai jurnal terkait judul penelitian yang digunakan sebagai referensi penelitian dan memberikan perspektif peneliti saat merancang skenario.

3.3.2. Perancangan Skenario

Tahap kedua peneliti adalah merancang skenario untuk melakukan simulasi. Perancangan ini menggunakan OMNeT++. Skenario peneliti dalam merancang simulasi adalah menambah jumlah *node*. *Node* yang akan digunakan adalah 50 *node*, 60 *node* dan 70 *node* dengan luas area yang sama yaitu 1500x1500 meter. Dalam penerapan di dunia nyata, kecepatan pergerakan suatu objek tidak dapat dipastikan. Namun pada tabel 3.1 menampilkan rata – rata model kecepatan pergerakan suatu objek dalam beberapa kondisi.

Tabel 3.1 Klasifikasi Kecepatan *Node* [27]

Model Mobility	Kecepatan
HWM (<i>Human Walking Model</i>)	2 mps
RWM (<i>Race Walking Model</i>)	4 mps
SCM (<i>Slow Car Model</i>)	10 mps
FCM (<i>Fast Car Model</i>)	>30mps

Penelitian ini akan menggunakan kondisi RWM dalam penerapan kecepatan pergerakan setiap *node*-nya.

3.3.3. Penentuan Parameter Jaringan

Jaringan MANET adalah salah satu yang digunakan pada peneliti ini. Protokol routing DYMO dan DSDV digunakan sebagai protokol uji. Simulasi dijalankan dengan meningkatkan ukuran paket, jumlah node, dan jumlah node sumber.

3.3.4. Proses Simulasi

Penelitian ini menjalankan program dengan skenario yang dibuat oleh peneliti selama tahap desain skenario sebagai bagian dari proses simulasi. Simulasi dilakukan dengan mengirim paket secara berkala disetiap 0,0666 sekon dengan ukuran awal 512 byte kemudian ditambah menjadi 5000 byte dan jumlah awal *node* adalah 50 yang akan ditambah menjadi 60 dan 70, dan *source node* awal adalah 3 yang akan ditambah menjadi 6.

3.3.5. Hasil Simulasi

Hasil simulasi yang dijadikan sebagai bahan penelitian adalah nilai rata – rata *Throughput*, *End to End Delay*, *Packet Delivery Ratio*, dan *Control Message*. Jika hasil simulasi tidak keluar, maka proses diulang sampai hasil pengujian keluar.

3.3.6. Pengambilan Data

Setelah proses simulasi selesai maka dilanjutkan dengan pengambilan data hasil simulasi yang menjadi sebuah file dengan format *.sca*, *.vci*, dan *.vec* dimana file – file tersebut akan dimasukkan ke dalam file yang berformat *.anf* agar mempermudah proses pengambilan data. Dari data tersebut akan didapatkan total paket yang dikirim, total paket yang diterima, dan total delay yang didapatkan dari *node*. Dari data simulasi tersebut akan dilakukan perhitungan menggunakan rumus yang tertera pada sub-bab 3.5.1 untuk mendapatkan rata – rata *throughput*, *end to end delay* dan *packet delivery ratio*. Selain dari file dengan format yang sudah disebutkan, dilakukan pengambilan data berupa *message/packet traffic* pada tampilan utama Tkenv dari OMNET++ yang kemudian dilakukan perhitungan menggunakan rumus yang tertera pada sub-bab 3.5.2 untuk mendapatkan nilai rata – rata dari *control message*.

3.3.7. Analisis Data

Peneliti melakukan analisis dari rata – rata *throughput*, *end to end delay*, *packet delivery ration*, dan *control message* yang didapatkan untuk melakukan perbandingan dari kinerja protokol *routing* DYMO dan DSDV.

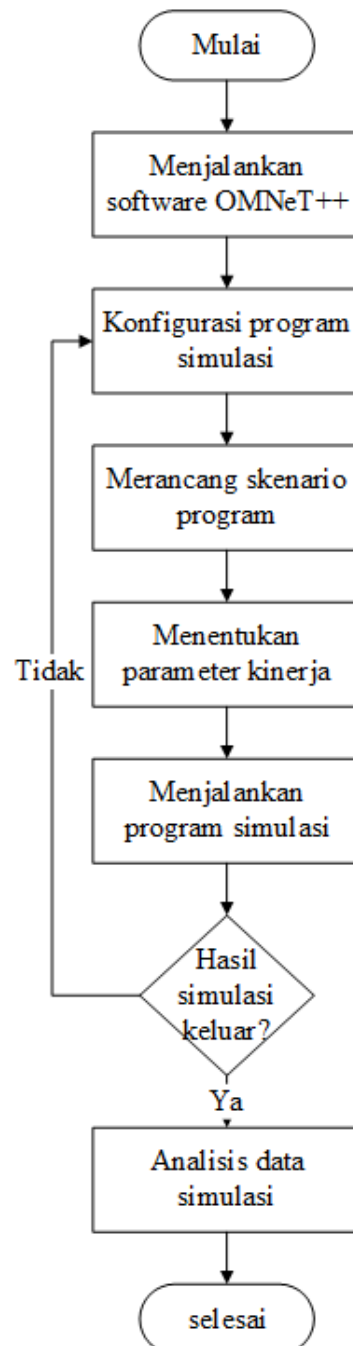
3.3.8. Penyusunan Laporan

Tahap terakhir dari penelitian ini adalah penulisan laporan. Ketika semua tahap proses telah selesai, peneliti menyusun laporan menggunakan informasi yang diperoleh dari tinjauan literatur untuk menarik kesimpulan.

3.4. Perancangan Program

Alur dari perancangan program yang dilakukan oleh peneliti dapat dilihat dari gambar 3.2. Penelitian ini menggunakan simulator OMNeT++ sebagai media uji. Perancangan program mulai dari menjalankan *software* simulator OMNeT++,

konfigurasi parameter simulasi, merancang skenario simulasi, menentukan parameter kinerja, kemudian menjalankan program, hingga lanjut ke tahap akhir yaitu pengambilan data.



Gambar 3.2 *Flowchart* Perancangan Program

3.4.1. Konfigurasi Program

Penelitian ini menggunakan parameter yang bersifat tetap seperti luas area pada simulasi adalah 1500m x 1500m dan waktu simulasi yang dilakukan selama 600s. Parameter simulasi pada protokol *routing* DYMO dan DSDV dapat dilihat pada tabel 3.2.

Tabel 3.2 Parameter Simulasi

Parameter	Nilai
Luas Area Jaringan	1500m x 1500m
Waktu Simulasi	600s
<i>Radio Range</i>	250 m
Jumlah <i>Node</i>	50, 60, dan 70
Kecepatan <i>Node</i>	RWM (4 mps)
Ukuran Paket	512 Byte, 5000 Byte, dan 10000 Byte
<i>Type Mobility</i>	<i>Random Way Point (RWP)</i>
Interval Pengiriman	0,0666s
<i>Traffic Source</i>	UDP
Banyak <i>Source</i>	3 dan 6
Sistem Komunikasi	802.11 g

Gambar 3.3 merupakan tampilan dari menu source pada OMNET++ untuk melakukan konfigurasi program.



```

[General]
network = inet.examples.practice.ManetNetwork
sim-time-limit = 600s
repeat = 30
#num-rngs = 3

cmdenv-express-mode = true
tkenv-plugin-path = ../../../../etc/plugins

description = "DYMO Source Node 3"

# mobility
**.mobilityType = "RandomWPMobility"
**.mobility.speed = 4mps + uniform(0.01mps, -0.01mps)
**.drawCoverage = false
**.constraintAreaMinX = 0m
**.constraintAreaMinY = 0m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxX = 1500m
**.constraintAreaMaxY = 1500m
**.constraintAreaMaxZ = 0m

## node configuration
**.host[0..2].numUdpApps = 1
**.host[0..2].udpApp[0].typename = "UDPBasicBurst"
**.host[0].udpApp[0].destAddresses = "fixhost[0]"
**.host[1].udpApp[0].destAddresses = "fixhost[1]"
**.host[2].udpApp[0].destAddresses = "fixhost[2]"
**.host[0..2].udpApp[0].localPort = 1234 #5000
**.host[0..2].udpApp[0].destPort = 1234 #5000
**.host[0..2].udpApp[0].sendInterval = 0.0666s
**.host[0..2].udpApp[0].burstDuration = 0
**.host[0..2].udpApp[0].chooseDestAddrMode = "perBurst"
**.host[0..2].udpApp[0].sleepDuration = 1s
**.host[0..2].udpApp[0].startTime = 0s

```

Gambar 3.3 Konfigurasi Program

3.4.2. Skenario Program

Skenario pada simulasinya menggunakan luas area tetap namun terdapat penambahan di jumlah *node*, jumlah *source node*, dan ukuran paket. *Node* yang bertugas mengirimkan paket diberi nama host dan *node* yang bertugas sebagai penerima diberi nama fixhost. Dalam simulasi, host[0] yang berarti host ke-0 akan mengirim paket ke fixhost[0] atau fixhost ke-0. Dalam skenario source *node* 3, host yang bertugas melakukan pengiriman paket adalah host[0] sampai host[2], dan dalam skenario source *node* 6 yang bertugas melakukan pengiriman paket adalah host[0] sampai host[5]. Model mobilitas RWP digunakan untuk setiap skenario dan menerapkan model kecepatan RWM (4 mps). Tabel 3.3 dan 3.4 menunjukkan skenario simulasi protokol DYMO dan DSDV. Simulasi akan dijalankan sebanyak 2x pada masing – masing skenario.

Tabel 3.3 Skenario A *Source Node 3*

Parameter	Jumlah <i>Node</i>	Ukuran Paket
A1	50	512 Byte
A2	60	512 Byte
A3	70	512 Byte
A4	50	5000 Byte
A5	60	5000 Byte
A6	70	5000 Byte
A7	50	10000 Byte
A8	60	10000 Byte
A9	70	10000 Byte

Table 3.4 Skenario B *Source Node 6*

Parameter	Jumlah <i>Node</i>	Ukuran Paket
B1	50	512 Byte
B2	60	512 Byte
B3	70	512 Byte
B4	50	5000 Byte
B5	60	5000 Byte
B6	70	5000 Byte
B7	50	10000 Byte
B8	60	10000 Byte
B9	70	10000 Byte

3.4.3. Parameter Kinerja

Parameter kinerja yang diujikan adalah protokol *routing* DYMO dan DSDV, dimana kedua protokol tersebut akan diuji dengan skenario yang dijelaskan pada 3.4.2. Peneliti akan melakukan konfigurasi jaringan pada *software* simulasi jaringan OMNeT++.

3.4.4. Analisis Data

Setelah tahap konfigurasi dan program telah selesai dijalankan, selanjutnya menuju ke tahap analisis data dimana data tersebut akan dilakukan perhitungan untuk mendapatkan nilai dari setiap parameternya. Adapun rumus dari setiap parameter dilihat pada sub-bab 3.5.

Gambar 3.4 merupakan tampilan dari menu Browse Data pada file yang berformat .anf dan gambar 3.5 merupakan tampilan *message/packet traffic* pada Tkenv.

R...	Module	Name	Value
1	ManetNetwork.fixhost[0].networkLayer.arp	initiatedARPResolution:count	2.0
1	ManetNetwork.fixhost[0].networkLayer.arp	sentReply:count	26.0
1	ManetNetwork.fixhost[0].networkLayer.arp	sentReq:count	2.0
1	ManetNetwork.fixhost[0].networkLayer.arp	failedARPResolution:count	0.0
1	ManetNetwork.fixhost[0].networkLayer.arp	completedARPResolution:count	49.0
1	ManetNetwork.fixhost[0].lo0.lo	rcvdPkFromHL:count	0.0
1	ManetNetwork.fixhost[0].lo0.lo	rcvdPkFromHL:sum(packetBytes)	0.0
1	ManetNetwork.fixhost[0].lo0.lo	passedUpPk:count	0.0
1	ManetNetwork.fixhost[0].lo0.lo	passedUpPk:sum(packetBytes)	0.0
1	ManetNetwork.fixhost[0].lo0.lo	droppedPkBitError:count	0.0
1	ManetNetwork.fixhost[0].lo0.lo	droppedPkBitError:sum(packetB...	0.0
1	ManetNetwork.fixhost[0].wlan[0].mgmt	dropPkByQueue:count	0.0
1	ManetNetwork.fixhost[0].wlan[0].mgmt	dropPkByQueue:sum(packetByt...	0.0
1	ManetNetwork.fixhost[0].wlan[0].mgmt	dataQueueLen:max	1.0
1	ManetNetwork.fixhost[0].wlan[0].mgmt	dataQueueLen:timeavg	7.942465828406...
1	ManetNetwork.fixhost[0].wlan[0].mac	number of received packets	5069.0
1	ManetNetwork.fixhost[0].wlan[0].mac	number of collisions	1135.0
1	ManetNetwork.fixhost[0].wlan[0].mac	number of internal collisions	0.0
1	ManetNetwork.fixhost[0].wlan[0].mac	number of retry for AC 0	49.0
1	ManetNetwork.fixhost[0].wlan[0].mac	sent and received bits	4524912.0
1	ManetNetwork.fixhost[0].wlan[0].mac	sent packet within AC 0	384.0

Gambar 3.4 *Browse Data* Pada File .anf

1 - dymo_udp3.ini - C:\omnetpp-4.6\samples\inetmanet-2.0\examples\practice

Event#	Time	Src/Dest	Name	Info
#186	0.001042849441	host[2] --> host[23]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#186	0.001042849441	host[2] --> host[38]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#186	0.001042849441	host[2] --> host[46]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#216	0.001297654692	host[38] --> host[2]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#216	0.001297654692	host[38] --> host[7]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#216	0.001297654692	host[38] --> host[13]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#216	0.001297654692	host[38] --> host[23]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#216	0.001297654692	host[38] --> host[24]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#216	0.001297654692	host[38] --> host[26]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#267	0.001706168017	host[24] --> host[13]	RE_DymoMsg_RREQ	Dymo_RE:99 bytes
#267	0.001706168017	host[24] --> host[26]	RE_DymoMsg_RREQ	Dymo_RE:99 bytes
#267	0.001706168017	host[24] --> host[38]	RE_DymoMsg_RREQ	Dymo_RE:99 bytes
#302	0.002616468328	host[23] --> host[2]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#302	0.002616468328	host[23] --> host[38]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#302	0.002616468328	host[23] --> host[46]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#307	0.002645396209	host[7] --> host[38]	RE_DymoMsg_RREQ	Dymo_RE:99 bytes
#307	0.002645396209	host[7] --> host[46]	RE_DymoMsg_RREQ	Dymo_RE:99 bytes
#341	0.003439004442	host[0] --> host[27]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#341	0.003439004442	host[0] --> host[39]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#341	0.003439004442	host[0] --> host[41]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#341	0.003439004442	host[0] --> host[45]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#347	0.00347343954	host[1] --> host[3]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#347	0.00347343954	host[1] --> host[5]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#347	0.00347343954	host[1] --> host[9]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#347	0.00347343954	host[1] --> host[11]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#347	0.00347343954	host[1] --> host[20]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#347	0.00347343954	host[1] --> host[29]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#347	0.00347343954	host[1] --> host[44]	RE_DymoMsg_RREQ	Dymo_RE:23 bytes
#434	0.003694958498	host[46] --> host[2]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#434	0.003694958498	host[46] --> host[7]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#434	0.003694958498	host[46] --> host[16]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#434	0.003694958498	host[46] --> host[23]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#440	0.003725987922	host[41] --> host[0]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#440	0.003725987922	host[41] --> host[14]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#440	0.003725987922	host[41] --> host[27]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#496	0.004196145261	host[14] --> host[10]	RE_DymoMsg_RREQ	Dymo_RE:99 bytes
#496	0.004196145261	host[14] --> host[41]	RE_DymoMsg_RREQ	Dymo_RE:99 bytes
#522	0.004603834477	host[29] --> host[1]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#522	0.004603834477	host[29] --> host[5]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#545	0.00490552387	host[45] --> host[0]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#545	0.00490552387	host[45] --> host[12]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes
#545	0.00490552387	host[45] --> host[19]	RE_DymoMsg_RREQ	Dymo_RE:56 bytes

Gambar 3.5 Packet Traffic Pada Tkenv

3.5. Parameter Pengujian

3.5.1. Quality of Service

Quality of Service (QoS) adalah upaya untuk menggambarkan kualitas dan karakteristik suatu layanan serta mekanisme untuk mengukur seberapa baik kinerja jaringan.

Parameter yang digunakan sebagai uji kinerja dari protokol *routing* DYMO dan DSDV adalah :

1. Throughput

Jumlah rata-rata data (bit) yang dikirim ke node tujuan dalam jumlah waktu tertentu dikenal sebagai *Throughput*. Persamaan 3.1 menunjukkan rumus perhitungan *throughput*.

$$\text{Throughput (bit)} = \frac{8 \times \text{ukuran paket yang diterima (byte)}}{\text{jumlah waktu simulasi}} \quad (3.1)$$

2. End to End Delay

Jumlah waktu antara saat paket dikirim dan saat diterima oleh node tujuan dikenal sebagai *End to End Delay*. Perhitungan untuk menentukan end-to-end delay ditunjukkan dalam persamaan 3.2.

$$\text{End to End Delay} = \frac{\text{total delay}}{\text{total paket yang diterima}} \quad (3.2)$$

3. Packet Delivery Ratio

Packet Delivery Ratio mengukur berapa banyak paket yang dibuat oleh node sumber dan berapa banyak yang diterima oleh node tujuan. Persamaan 3.3 menunjukkan metode perhitungan untuk *packet delivery ratio*.

$$\text{Packet Delivery Ratio} = \frac{\text{total paket yang diterima}}{\text{total paket yang dikirim}} \times 100 \quad (3.3)$$

3.5.2. Control Message

Control Message adalah pesan kontrol yang digunakan dalam protokol *routing* untuk mengirim informasi tentang topologi jaringan, status *node*, dan informasi lain yang diperlukan untuk mengelola jaringan *mobile ad-hoc* (seperti *hello message*, RREQ, RREP, RERR). Pesan ini tidak mengandung data pengguna dan hanya digunakan untuk mengatur jaringan dan memastikan pengiriman data yang efisien. Dengan menghitung rata-rata bit yang dikirimkan oleh masing-masing *node* sesuai dengan jumlah *node* pada skenario, ditentukan nilai *control message* pada penelitian ini berdasarkan persamaan 3.4 berikut.

$$C = \frac{\sum_1^n b}{n} \times 8 \quad (3.4)$$

Berikut keterangan dari rumus tersebut :

C : *Control Message* (bit)

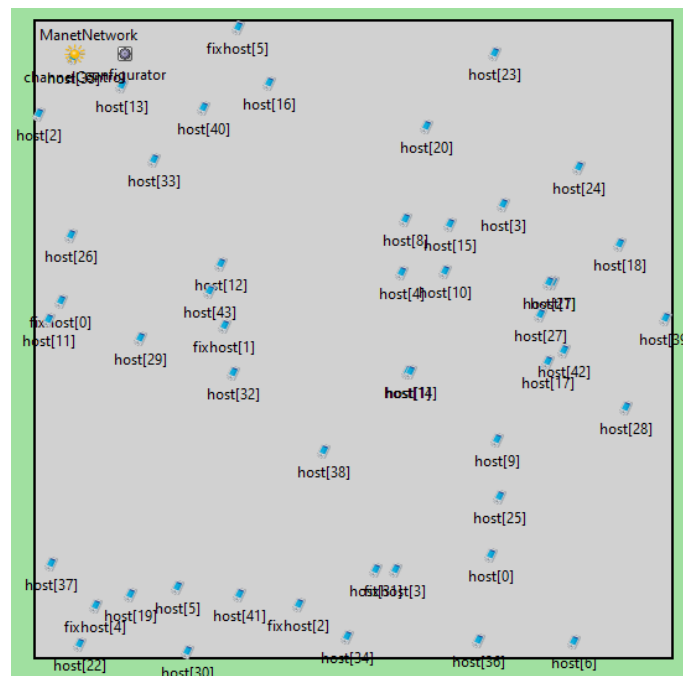
b : Bit yang dihasilkan (byte)

n : Jumlah *node*

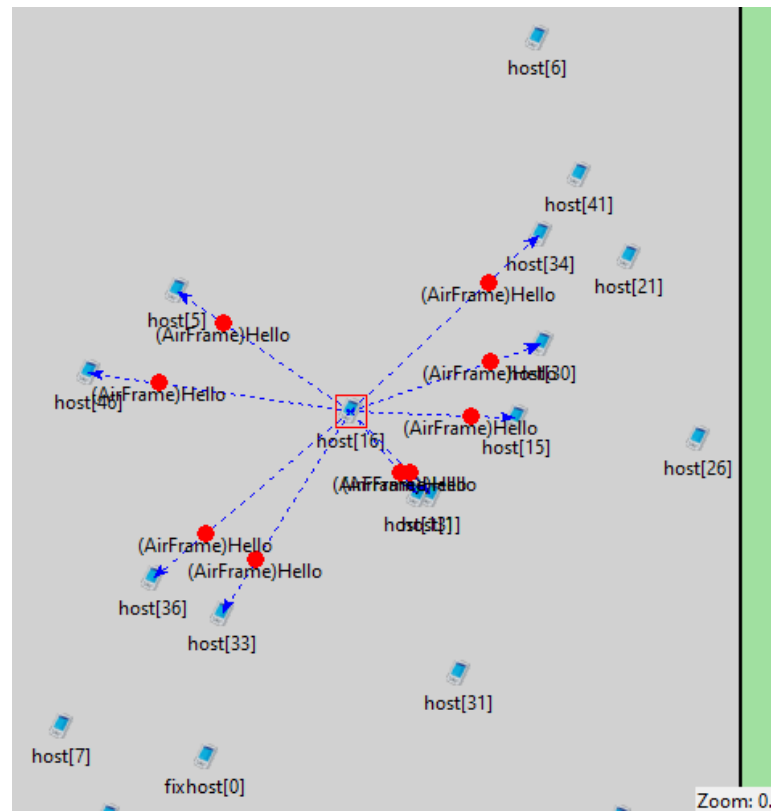
3.6. Topologi Jaringan

Karena perangkat atau node dalam jaringan dapat bergerak bebas, topologi jaringan pada MANET bersifat dinamis dan tidak dapat diprediksi. Setiap node dalam jaringan ini bertindak sebagai *router*, mengirimkan lalu lintas ke *node* lain. Data biasanya dirutekan di banyak *node* pada desain jaringan *multihop* sebelum tiba di lokasi yang dimaksud. Dalam MANET, arsitektur jaringan dapat bergeser secara tiba-tiba dan tidak menentu dari waktu ke waktu, menciptakan tautan yang dapat berjalan dengan cara satu arah atau dua arah. Sampai paket mencapai node target, itu akan diteruskan dari node sumber ke node 250 meter jauhnya di area tersebut menggunakan rute yang sebelumnya ditentukan menggunakan pesan RREQ.

Gambar 3.6 menunjukkan topologi dengan menggunakan pergerakan *Random Way Point* yang membuat lokasi dan pergerakan setiap *node*-nya menjadi acak. Gambar 3.7 menunjukkan proses pengiriman pesan hello berlangsung, dimana *host[16]* mengirimkan pesan ke beberapa *host* yang dapat dijangkau yaitu *host[5]*, *host[34]*, *host[30]*, *host[15]*, *host[13]*, *host[1]*, *host[33]*, *host[36]*, dan *host[46]*.



Gambar 3.6 Topologi Simulasi Jaringan



Gambar 3.7 Proses Pengiriman Pesan