SIMULASI *VEHICULAR AD HOC NETWORK (*VANET) MENGGUNAKAN *OPENSTREETMAP* DENGAN *SIMULATION OF URBAN MOBILITY* (SUMO) DAN OMNET++



Disusun Oleh:

N a m a : Gita Agistira Romanta

NIM : 14523235

PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2017

HALAMAN PENGESAHAN DOSEN PEMBIMBING

SIMULASI *VEHICULAR AD HOC NETWORK (*VANET) MENGGUNAKAN *OPENSTREETMAP* DENGAN *SIMULATION OF URBAN MOBILITY* (SUMO) DAN OMNET++

TUGAS AKHIR

Disusun Oleh:

N a m a	: Gita Agistira Romanta
NIM	: 14523235

Yogyakarta, 19 Agustus 2018

Pembimbing,

(Yudi Prayudi, S.Si, M.Kom)

HALAMAN PENGESAHAN DOSEN PENGUJI

SIMULASI *VEHICULAR AD HOC NETWORK (*VANET) MENGGUNAKAN *OPENSTREETMAP* DENGAN *SIMULATION OF URBAN MOBILITY* (SUMO) DAN OMNET++

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 1 Nopember 2017

Tim Penguji

- _____

Anggota 1

- _____

Anggota 2

- _____

Mengetahui,
Ketua Program Studi Teknik Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dr.. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Gita Agistira Romanta

NIM : 14523235

Tugas akhir dengan judul:

SIMULASI *VEHICULAR AD HOC NETWORK (*VANET) MENGGUNAKAN *OPENSTREETMAP* DENGAN *SIMULATION OF URBAN MOBILITY* (SUMO) DAN OMNET++

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 19 Agustus 2018

(Gita Agistira Romanta)

HALAMAN PERSEMBAHAN

“Dari Abu Zar berkata: Rasulullah saw bersabda kepadaku: “Hai Abu Zar, keluarmu dari rumah pada pagi hari untuk mempelajari satu ayat dari kitab Allah itu lebih baik daripada engkau mengerjakan salat seratus rakaat (H.R. Ibnu Majah: 215)”.

Dengan segala rendah hati kuhaturkan terima kasih kepada

- KepadaMu Ya Allah tempat meminta pertolongan terimalah sebagai amal ibadahku.
- Bapak-Ibuku tercinta yang selalu mencintai dan menyayangi saya, Doa dan pengorbanan kalian yang tak berujung adalah sumber kebahagiaan dalam memaknai hidup.
- Teman-temanku Teknik Informatika 2014 yang tidak henti-hentinya memberikan dorongan, dukungan, bantuan dan dengan kehadiran kalian membuat hidup lebih hidup.
- To My Soulmate Irmawati Firdaus, S.Pd yang sudah menemani dan menyemangati agar terselesainya skripsi ini.
- Teman-temanku dan semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu demi kelancaran penelitian dan dalam menyelesaikan Tugas Akhir ini.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur ke hadirat Allah SWT atas rahmat dan hidayah-Nya sehingga peneliti dapat menyelesaikan skripsi dengan judul "SIMULASI *VEHICULAR AD HOC NETWORK (*VANET) MENGGUNAKAN *OPENSTREETMAP* DENGAN *SIMULATION OF URBAN MOBILITY* (SUMO) DAN OMNET++". Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad SAW, keluarga, sahabat, serta para pengikut beliau hingga akhir zaman.

Skripsi ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Komputer (S.Kom.) pada Program Studi Teknik Informatika, Fakultas Teknik Industri, Universitas Islam Indonesia. Pada kesempatan ini peneliti ingin menyampaikan rasa terima kasih kepada semua pihak yang telah membantu, membimbing dan memberikan arahan serta informasi yang berguna. Ucapan terima kasih tersebut ditujukan kepada:

1) Bapak Nandang Sutrisno, SH., M.Hum., selaku Rektor Universitas Islam Indonesia yang telah memberikan fasilitas dan kemudahan.
2) Bapak Raden Teduh Dirgahayu selaku Ketua Prodi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3) Bapak Yudi Prayudi, S.Si, M.Kom. selaku Dosen Pembimbing I yang telah memberikan masukan, arahan, motivasi dan ide selama penrliti melaksanakan penulisan skripsi ini.
4) Semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat disebutkan satu persatu.

Peneliti menyadari bahwa dalam penyusunan skripsi ini masih banyak terdapat kekurangan serta jauh dari sempurna. Oleh karena itu, peneliti mengharapkan kritik dan saran yang sifatnya membangun. Semoga skripsi ini dapat memberikan manfaat bagi peneliti maupun bagi pihak-pihak yang membutuhkan. Amin.....

Wassalamu'alaikum Wr. Wb

Yogyakarta, 1 Nopember 2017

(Gita Agistira Romanta)

SARI

Meningkatnya jumlah kendaraan per tahun menimbulkan banyak permasalahan bukan saja polusi tapi juga kemacetan hingga kecelakaan lalu lintas yang terus bertambah setiap tahunnya. Sehingga, diperlukan suatu sistem yang dapat membantu mengurangi kemacetan dan kecelakaan seperti Intelligent Transportation System (ITS). ITS adalah transportasi cerdas yang menggabungkan antara sistem transportasi dengan teknologi informasi demi meningkatkan aksesibilitas dan efisiensi serta keamanan transportasi.

Saat ini perkembangan teknologi informasi dan komunikasi menjadi salah satu indikator kemjuan manusia. Salah satu teknologi yang membantu manusia dalam berkomunikasi dengan mudah adalah Vehicular Ad Hoc Network (VANET). VANET termasuk dalam jaringan komunikasi nirkabel dan merupakan subkelas dari Mobile Ad Hoc Network (MANET). Jaringan VANET adalah suatu jaringan Ad Hoc yang digunakan untuk berkomunikasi antara kendaraan satu dengan kendaraan lainnya dengan kata lain disebut juga dengan vehicle to vehicle communication (V2V). SUMO adalah sebuah *framework* simulasi jaringan *discrete-event* yang bertipe *object-oriented*. Simulasi dilakukan dengan menggunakan *Simulation Of Urban Mobility*, salah satu simulator untuk jaringan VANET. Implementasi VANET di dunia nyata membutuhkan biaya yang sangat mahal dan jangkauan luas sehingga tidak memungkinkan untuk diimplementasikan secara langsung.

Dalam sebuah jaringan, protokol routing merupakan hal yang sangat berpengaruh terhadap kinerja jaringan tersebut. Begitu pula dalam jaringan VANET, terdapat banyak protokol routing yang dapat diimplementasikan pada jaringan ini. Salah satunya adalah protokol routing AODV (Ad hoc On demand Distance Vector). Pada penelitin ini, peneliti menggunakan simulator SUMO. Skenario VANET berupa peta hasil generate baik berbentuk grid maupun peta nyata, kecepatan kendaraan, dan traffic light serta berbagai fitur yang mendukung kinerja simulasi VANET. Generator ini banyak digunakan oleh para peneliti untuk menguji skema VANET. Penelitian ini membahas tentang studi kinerja dari skema VANET terhadap salah satu generator, yaitu SUMO dengan hasil akhir yang diharapkan adalah untuk mengetahui kinerja jaringan VANET dalam skenario (peta berbentuk peta riil).

GLOSARIUM

Glosarium memuat daftar kata tertentu yang digunakan dalam laporan dan membutuhkan penjelasan, misalnya kata serapan yang belum lazim digunakan. Contoh penulisannya seperti di bawah ini:

Node	salah satu titik sambungan, titik redistribusi, atau titik akhir komunikasi (beberapa terminal peralatan).
Source code	kumpulan pernyataan atau deklarasi bahasa pemrogramman komputer yang ditulis dan dapat di baca manusia.
Polygon	kerangka dasar pemetaan yang memiliki titik-titik dimana titik tersebut mempunyai sebuah koordinat X dan Y.
Vehicles	bebrapa kendaraan

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
KATA PENGANTAR vi
SARI viii
GLOSARIUM ix
DAFTAR ISI x
DAFTAR TEXT BOX xii
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 2
1.3 Tujuan Penelitian 2
1.4 Manfaat Penelitian 2
1.5 Batasan Masalah 2
1.6 Metodologi Penelitian 3
1.7 Sistematika Penulisan 3
BAB II LANDASAN TEORI 5
2.1 Dasar Teori 5
2.1.1 Simulation of Urban Mobility (SUMO) 5
2.1.2 Protokol AODV 5
2.1.3 OpenStreetMap 6
2.2 Literatur Riview 7
2.3 Bahasa Pemrograman 8
2.3.1 Extensible Markup Language (XML) 10
2.4 Karakteristik Perancangan Protokol VANET 11
2.5 Arsitektur VANET 13
2.5.1 Komponen Utama 13
2.5.2 Arsitektur Komunikasi 14
BAB III METODOLOGI 15
3.1 Kebutuhan Sistem 15
3.1.1 Langkah Menginstall Omnet++ 15
3.2 Perancangan Sistem 17
3.2.1 Pembuatan Peta Pada SUMO 17
3.2.2 Perancangan Simulasi VANET pada Omnet++ yang di Import dari VEINS 23
BAB IV HASIL DAN PEMBAHASAN 24
4.1 Hasil 24
4.1.1 Simulasi VANET pada Generator SUMO 24
4.1.2 Melacak Kendaraan 28
4.1.3 Simulasi VANET Pada Omnet++ 30
4.2 Pembahasan 36
4.2.1 Protokol Routing untuk Jaringan Ad-Hoc 36
4.2.2 AODV (Ad-hoc On-demand Distance Vector) 37

4.3 Pengujian Penelitian.....38
4.4 Analisis Penelitian.....39
BAB V KESIMPULAN DAN SARAN.....41
5.1 Kesimpulan.....41
5.2 Saran.....41
DAFTAR PUSTAKA.....42

DAFTAR TEXT BOX

Text 3.1 *sorce code* file polygon.....21
Text 3.2 *source code* file .cfg.....22

DAFTAR TABEL

Tabel 4.1 Hasil Parameter VANET pada SUMO 26

DAFTAR GAMBAR

Gambar 3.1 Tampilan awal Omnet++ 17
Gambar 3.2 Proses “configure” pada Omnet++ 18
Gambar 3.3 Proses “make” pada Omnet++ 19
Gambar 3.4 Tampilan website openstreetmap.org 20
Gambar 3.5 *source code* dan proses konfersi 23
Gambar 3.6 Proses konfersi file berhasil 24
Gambar 3.7 *source code* dan proses polyconvert 24
Gambar 3.8 *source code* dan proses randomtrips 25
Gambar 3.9 *source code* konfigurasi veins 25
Gambar 3.10 *source code* pemanggilan project veins 25
Gambar 4.1 Tampilan Simulasi VANET 27
Gambar 4.2 Tampilan Simulasi VANET 28
Gambar 4.3 Tampilan *Locate Vehicles* 29
Gambar 4.4 Tampilan *Locate Vehicles* 120 29
Gambar 4.5 Tampilan Simulasi pada Omnet++ 30
Gambar 4.6 Tampilan *Result* 31
Gambar 4.7 *result* Max Speed 31
Gambar 4.8 Tampilan Max Speed Salah Satu Node 32
Gambar 4.9 Tampilan Grafik Pos Y 32
Gambar 4.10 Tampilan Grafik Pos Y Seluruh Node 33
Gambar 4.11 Tampilan Grafik pos X 33
Gambar 4.12 Tampilan Grafik Pos X Seluruh Node 34
Gambar 4.13 Tampilan *Distance* 34
Gambar 4.14 Tampilan *Distance* Salah Satu Node 35
Gambar 4.15 Tampilan *result sentpackets* 36

BAB I
PENDAHULUAN

1.1 Latar Belakang

Meningkatnya jumlah kendaraan per tahun menimbulkan banyak permasalahan bukan saja polusi tapi juga kemacetan hingga kecelakaan lalu lintas yang terus bertambah setiap tahunnya. Sehingga, diperlukan suatu sistem yang dapat membantu mengurangi kemacetan dan kecelakaan seperti *Intelligent Transportation System* (ITS). ITS adalah transportasi cerdas yang menggabungkan antara sistem transportasi dengan teknologi informasi demi meningkatkan aksesibilitas dan efisiensi serta keamanan transportasi. Saat ini perkembangan teknologi informasi dan komunikasi menjadi salah satu indikator kemjuan manusia. Salah satu teknologi yang membantu manusia dalam berkomunikasi dengan mudah adalah *Vehicular Ad Hoc Network* (VANET). VANET termasuk dalam jaringan komunikasi nirkabel dan merupakan subkelas dari *Mobile Ad Hoc Network* (MANET).

Saat ini kepadatan kendaraan di daerah Yogyakarta khusunya perempatan Jl. Kaliurang dengan Ring Road Utara dan Jl. Gejayan dengan Ring Road Utara semakin meningkat sehingga mengkibatkan kemacetan pada jam-jam tertentu khususnya pada jam 17.00 dan tidak sedikit pula terjadi kecelakaan pada daerah tersebut, maka perlu adanya sebuah solusi yang mampu mengatasi kemacetan dan kecelakaan tersebut. Terdapat beberapa solusi, salah satunya sistem jaringan VANET karena sistem jaringan VANET adalah teknologi yang penting untuk dapat meningkatkan keamanan dalam berkendara. Dengan teknologi VANET kendaraan dapat saling berkomunikasi, sehingga jika ada potensi kecelakaan yang bisa terjadi, pengendara akan diberi peringatan agar bisa menghindari kecelakaan tersebut, dengan begitu pencegahan kecelakaan jalan raya dapat dilakukan.

Jaringan VANET adalah suatu jaringan *Ad Hoc* yang digunakan untuk berkomunikasi antara kendaraan satu dengan kendaraan lainnya dengan kata lain disebut juga dengan vehicle to vehicle communication (V2V). SUMO adalah sebuah *framework* simulasi jaringan *discrete-event* yang bertipe *object-oriented*. Simulasi dilakukan dengan menggunakan *Simulation Of Urban Mobility*, salah satu simulator untuk jaringan VANET. Implementasi VANET di dunia nyata membutuhkan biaya yang sangat mahal dan jangkauan luas sehingga tidak memungkinkan untuk diimplementasikan secara langsung. Dalam sebuah jaringan, protokol routing merupakan hal yang sangat berpengaruh terhadap kinerja jaringan tersebut. Begitu pula dalam jaringan VANET, terdapat banyak protokol routing yang dapat diimplementasikan pada jaringan ini. Salah satunya adalah protokol routing AODV (*Ad hoc*

On demand Distance Vector). Pada penelitian ini menggunakan simulator SUMO dan Omnet++. Skenario VANET berupa peta hasil generate baik berbentuk grid maupun peta nyata, kecepatan kendaraan, pelacakan kendaraan dan traffic light serta berbagai fitur yang mendukung kinerja simulasi VANET. Generator ini banyak digunakan oleh para peneliti untuk menguji skema VANET. Penelitian ini membahas tentang studi kinerja dari skema VANET terhadap dua generator, yaitu SUMO dan Omnet++ dengan hasil akhir yang diharapkan adalah untuk mengoptimalkan pergerakan kendaraan di daerah Yogyakarta khusunya Jl. Kaliurang dan Jl. Gejayan agar dapat mengurangi kemacetan dan kecelakaan di daerah tersebut.

1.2 Perumusan Masalah

Berdasarkan penjelasan latar belakang di atas, maka dapat ditetapkan rumusan masalah, sebagai berikut :

1. Bagaimana cara mensimulasikan VANET pada simulator SUMO dan Omnet++ ?
2. Bagaimana cara menampilkan hasil simulasi VANET pada Omnet++ ?
3. Bagaimana cara mengoptimalkan pergerakan kendaraan di Jl. Kaliurang dan Jl. Gejayan untuk mengurangi kemacetan dan kecelakaan ?

1.3 Tujuan Penelitian

Tujuan dari Penelitian ini adalah untuk :

1. Mengoptimalkan pergerakan kendaraan di Jl. Kaliurang dan Gejayan untuk mengurangi kemacetan dan kecelakaan.
2. Melacak kendaraan yang bergerak maupun tidak bergerak.
3. Mendapatkan hasil simulasi yang optimum dengan menggunakan Simulator SUMO dan Omnet++.

1.4 Manfaat Penelitian

Adapun manfaat penelitian ini adalah :

1. Membantu masyarakat untuk mengurangi tingkat kemacetan karena semakin bertambahnya jumlah kendaraan.
2. Membantu masyarakat untuk mencegah terjadinya kecelakaan yang terjadi di jalan.

1.5 Batasan Masalah

Adapun batasan pekerjaan yang ditentukan pada tugas akhir ini adalah sebagai berikut:

1. Tipe pe-*routing*-an yang akan dibahas adalah AODV
2. Penelitian ini tidak membahas tentang pengaruh *mobility movement* terhadap *latency, packetloss, packet dilevery ratio* pada jaringan.
3. Tampilan simulasi menggunakan simulator SUMO.

4. Kecepatan kendaraan yang digunakan pada simulasi 0-40 km/jam.
5. Ukuran data yang dikirimkan sebesar 1 MB, 2 MB, 3 MB, 4 MB.

1.6 Metodologi Penelitian

Penelitian ini dilakukan dengan metode-metode sebagai berikut,

1. Studi Literatur

   Sumber bacaan berupa jurnal, artikel diperoleh dari internet yang berhubungan dengan *Vehicular Adhoc Network, Openstreetmap, Omnet++,* dan SUMO.

2. Perancangan

   Pada tahap ini, dilakukan persiapan kebutuhan dalam proses perancangan yang nantinya akan dipakai untuk pengimplemetasi. Rancangan system yang akan disiapkan adalah kebutuhan *sofware*, sebagai berikut: SUMO dan Omnet++ yang berfungsi sebagai simulator untuk merancang simulasi VANET.

3. Implementasi

   Pada tahap ini, semua rancangan yang telah dibuat akan diimplementasikan. Seperti pembuatan simulasi lengkap dengan rancangan scenario, map, kendaraan, dan luas map dengan menggunakan SUMO.

4. Pengujian

   Pada tahan ini, semua rancangan yang telah dibuat dilakukan pengujian untuk kerja VANET dengan menggunakan *protocol routing Ad Hoc Demand Distance Vector Routing* (AODV). Bertujuan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan kebutuhan yang diinginkan.

1.7 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini terdiri dari beberapa pokok pembahasan yaitu sebagai berikut:

BAB I Pendahuluan

Bab ini berisi latarbelakang tentang system yang akan dikembangkan, rumusan masalah pada penelitian ini, batasan masalah pada pengerjaan penelitian, tujuan pembuatan simulasi, metode penelitian yang digunakan dan sistematika penulisan.

BAB II Landasan Teori

Bab ini berisi tentang teori VANET dan *review* simulator yang digunakan dalam penelitian ini.

BAB III Metodologi

Bab ini berisi perancangan sistem yang akan disimulasikan pada VANET.

BAB IV Implementasi dan pengujian sistem

Bab ini berisi Hasil, Pembahasan dan Implementasi dari system yang dibuat.

BAB V Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan yang bertujuan untuk perbaikan dan pengembangan simulasi di masa mendatang.

BAB II
LANDASAN TEORI

2.1 Dasar Teori

2.1.1 Simulation of Urban Mobility (SUMO)

Simulation of Urban Mobility atau biasa dikenal dengan SUMO merupakan sebuah program *open source simulator* lalu lintas jalan yang memungkinkan pengguna untuk membangun simulasi pergerakan kendaraan pada topologi jaringan VANET yang disesuaikan.

Pengimplementasian SUMO dimulai pada tahun 2001, dengan sebuah rilis *open source* tahun 2002, dikembangkan oleh Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. setiap kendaraan dimodelkan secara eksplisit, memiliki rute sendiri, dan bergerak secara individu melalui jaringan (Nutrihadi, Anggoro, & Ijtihadie, 2016).

2.1.2 Protokol AODV

AODV adalah protokol routing reaktif yang dapat digunakan untuk mendukung komunikasi unicast atau multicast. Protokol ini tidak menyimpan infromasi rute yang lengkap, melainkan hanya informasi node yang harus dituju pada lompatan berikutnya. Sama dengan DSR, protokol ini juga terdiri dari dua mekanisme utama, yaitu: mekanisme pencarian rute dan pemeliharaan rute. Pesan RREQ (*Route Request*) dan RREP (*Route Reply*) digunakan untuk pencarian rute. Node yang akan mengirimkan data akan menyebarkan paket RREP ke node-node di sekitarnya untuk menemukan node tujuan. Jadi, paket RREQ digunakan untuk mengawali pembentukan rute menuju tujuan. Terdapat beberapa informasi yang ada pada paket RREQ, antara lain: alamat node sumber/pengirim dan alamat node penerima, pencacah hop, souce sequence number, destination sequence number, dan broadcast ID.

Pencacah hop atau lompatan berfungsi sebagi pencatat jumlah lompatan setiap dilakukan pengiriman paket RREQ. Broadcast ID adalah identitas dari paket RREQ. Nilai ID ini akan bertambah setiap kali node sumber mengirimkan RREQ Langkah-langkah pencarian rute pada AODV yang lengkap adalah sebagai berikut: (Aziza, Siswipraptini, & Cahyaningtyas, 2017).

1. Jika suatu node sumber (S) membutuhkan suatu rute menuju node tujuan (D), tahap awal yang dilakukan oleh node sumber adalah menyebarkan paket route request (RREQ) menuju node tetangganya
2. Apabila node yang menerima RREQ memiliki informasi rute menuju D, maka node tersebut akan mengirim paket RREP kembali menuju node sumber melalui reverse path yang diciptakan RREQ setiap kali flooding dilakukan. Namun, jika node tersebut tidak memiliki informasi rute menuju D, maka node tersebut akan menyiarkan ulang pesan RREQ ke node di sekitarnya.
3. Node yang menerima RREQ dengan nilai alamat sumber dan nilai broadcast ID yang sama dengan RREQ yang diterima sebelumnya, akan mempertahankan RREQ yang sudah diterima di awal dan membuang RREQ baru.
4. Ketika sebuah node yang memiliki informasi rute menuju D menerima RREQ, maka node tersebut akan melakukan perbandingan antara nilai destination sequence number yang telah dimiliki dengan sequence number yang ada pada RREQ. Jika sequence number pada node lebih besar atau sama dengan yang ada di RREQ, maka node akan mengirim paket RREP menuju S. Jika sequence number pada RREQ lebih kecil, maka pesan RREQ akan diteruskan ke node lain.
5. Jika terdeteksi adanya kerusakan rute, mekanisme route maintenance akan mengirimkan paket route error (RERR) menuju S dan node sumber akan kembali menyebarkan paket RREQ.
6. Informasi masa aktif rute akan diberikan oleh node antara yang menerima RREQ. Informasi untuk perutean dari sumber (S) ke D akan dihapus jika waktu aktifnya telah habis.

2.1.3 OpenStreetMap

OpenStreetMap adalah sebuah proyek kolaboratif berbasis web untuk membuat dan membangun peta geografis seluruh dunia secara gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survei dan mengumpulkan data menggunakan perangkat GPS, fotografi udara, dan sumber bebas lainnnya, dimanfaatkan untuk beragam kebutuhan termasuk navigasi. Siapapun dapat berkontribusi untuk OSM. *OpenStreetMap* dirintis pertama kali oleh Steve Coast di Britania Raya pada tahun 2004. Melalui Open Data Commons Open Database License 1.0, kontributor OSM dapat memiliki, memodifikasi, dan membagikan data peta secara luas. Terdapat beragam jenis peta digital yang tersedia di internet, namun sebagian besar memiliki keterbatasan secara legal maupun teknis.

Hal ini membuat masyarakat, pemerintah, peneliti dan akademisi, inovator, dan banyak pihak lainnya tidak dapat menggunakan data yang tersedia di dalam peta tersebut secara bebas. Di sisi lain, baik peta dasar OSM maupun data yang tersedia di dalamnya dapat diunduh secara gratis dan terbuka, untuk kemudian digunakan dan diredistribusikan kembali. Di banyak tempat di dunia ini, terutama di daerah terpencil dan terbelakang secara ekonomi, tidak terdapat insentif komersil sama sekali bagi perusahaan pemetaan untuk mengembangkan data di tempat ini. OSM dapat menjadi jawaban di banyak tempat seperti ini, baik itu untuk pengembangan ekonomi, tata kota, kontinjensi bencana, maupun untuk berbagai tujuan lainnya.

Data geografis yang berkualitas akan sangat membantu banyak organisasi, komunitas, bahkan masyarakat luas dan pemerintah dalam mengambil keputusan terkait berbagai macam isu, seperti lingkungan, ekonomi, sosial, dan managemen krisis. Di berbagai tempat di dunia, informasi ini belum lengkap atau bahkan tidak ditemukan sama sekali. Relawan pemetaan digital dapat membantu memberikan informasi ini dengan cara mendigitasikan informasi bangunan, jalur transportasi, dan infrastruktur kritis lainnya yang sangat dibutuhkan di berbagai daerah untuk pembangunan berkelanjutan di tempat mereka. Bayangkan bahwa penghidupan di lingkungan kita maupun di kawasan lain di berbagai pelosok dapat berkembang hanya dengan bantuan sederhana kita memetakan wilayah ini.

Anda dapat memberikan kontribusi yang berdampak sangat besar hanya dengan koneksi internet dan kemampuan dasar pengoperasian komputer, bahkan saat Anda sendiri tidak tinggal di daerah yang ingin Anda bantu petakan. Gunakan citra satelit untuk menandai jalanan, bangunan, lahan, dan titik-titik tertentu dan anda bisa menambahkan beragam informasi terkait yang Anda ketahui secara pasti. Data ini akan sangat berguna bagi komunitas dan masyarakat di wilayah tersebut untuk tujuan pengembagan dan kemanusiaan (Nutrihadi et al., 2016).

2.2 Literatur Riview

VANET (*Vehicular Ad Hoc Network*) merupakan bagian dari MANET (*Mobile Ad Hoc Network*). VANET menggunakan jaringan wireless berbasis *Ad Hoc* yang diterapkan pada kendaraan bergerak. VANET menjadi salah satu area riset yang berkembang di beberapa tahun terakhir guna mendukung ITS (Intelligent Transportation System). Tujuan utama VANET adalah membantu kendaraan-kendaraan untuk dapat saling berkomunikasi dan memelihara jaringan komunikasi diantara mereka tanpa menggunakan *central base station* atau *controller*.

VANET bertanggung jawab untuk komunikasi antar kendaraan bergerak pada suatu lingkungan tertentu. Kendaraan dapat berkomunikasi dengan kendaraan lain yang disebut dengan V2V (*Vehicle to Vehicle*) communication atau kendaraan juga dapat berkomunikasi dengan infrastruktur seperti RSU (*Road Side Unit*) yang disebut dengan V2I (*Vehicle to Infrastructure*) *communication* atau V2R (*Vehicle to Roadside*) *communication* (Tonoro, Wardana, & Nugroho, 2015).

2.3 Bahasa Pemrograman

Era Modern saat ini, perangkat lunak sudah terdapat di mana-mana, tidak hanya pada sebuah superkomputer dengan 25 prosesornya, sebuah komputer genggampun telah di lengkapi dengan perangkat lunak yang dapat di sinkronkan dengan PC. Tidak hanya komputer, bahkan peralatan seperti telepon, TV, hingga ke mesin cuci, AC dan microwave, telah di tanamkan perangkat lunak untuk mengatur operasi peralatan itu. Hebatnya lagi adalah setiap peralatan itu akan mengarah pada suatu saat kelak akan dapat saling terhubung.

Pembuatan sebuah perangkat lunak bukan lagi pekerjaan segelentir orang, tetapi telah menjadi pekerjaan banyak orang, dengan beberapa tahapan proses yang melibatkan berbagai disiplin ilmu dalam perancangannya. Tingkat kecerdasan yang di tunjukkan oleh perangkat lunak pun semakin meningkat, selain permasalahan teknis, perangkat lunak sekarang mulai bisa mengenal suara dan gambar. Dengan pengolahan deta dengan komputer sangat bigitu erat dimana suatu sistem komputer adalah merupakan gabungan dari beberapa peralatan komputer yang saling berhubungan dan terorkodinasikan oleh suatu sistem operasi, sistem komputer bekerja sesuai dengan intruksi-intruksi yang diberikan padanya melalui bahasa pemrograman yang dikenalnya. Dalam suatu sistem komputer terdapat bagian/peralatan untuk menerima dan memasukan data dan intruksi. Alat ini kita sebut inpute device, seperi contohnya card-reader. Data yang dimasukan melalui inpute device ini akan ditampung dalam suatu peralatan komputer yang disebut memory atau storage. CPU terdiri dari control Unit dan Arithmatic Logical unit.

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam

berbagai situasi. Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

1. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110
2. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah.Inggris Assembly), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode mnemonic), contohnya MOV, SUB, CMP, JMP, JGE, JL, LOOP, dsb.
3. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&, ||, dsb.
4. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata- kata bahasa manusia, contohnya begin, end, if, for, while, and, or, dsb.

Sebagian besar bahasa pemrograman digolongkan sebagai Bahasa Tingkat Tinggi, hanya bahasa C yang digolongkan sebagai Bahasa Tingkat Menengah dan Assembly yang merupakan Bahasa Tingkat Rendah

Bahasa yang dipakai untuk menginstruksikan computer disebut bahasa pemrograman. Ada 2 jenis bahasa pemrograman terdiri dari bahasa tingkat tinggi & bahasa tingkat rendah. Kita kenal diantaranya: Basic, Algol, Cobol, Pascal, PL-1, RPG, SNOBOL, APL, LISP, GPSS, ADA, DEAL dan sebagainya yang merupakan bahasa tingkat tinggi. Bahasa yang dimengerti oleh mesin computer adalah intruksi dalam bahasa mesin (Machine Language) yang merupakan bahasa tingkat rendah, jadi bahasa tingkat tinggi yang kita sebutkan diatas agar dapat dimengerti oleh computer haruslah diterjemahkan lebih dahulu oleh kompilator. Ciri dari bahasa tingkat rendah adalah bahwa cara penulisanya intruksinya sangat mendekati bentuk intruksi-intruksi dalam bahasa mesin. Bahasa tersebute dalam bentuk kode HEXA yang sulit bagi sebagian untuk mengertinya.

Komputer bekerja secara elektronik, maka ia tidak dapat menerima masukan berupa tulisan kertas begitu saja. Ia baru dapat membaca informasi dalam bentuk kombinasi bit-bit listrik. Kombinasi-kombinasi tersebut didalam computer sering disebut sebagai kode mesin (Machine kode) yang tidak lain adalah bahasa computer itu sendiri, yaitu kode-kode yang dikenal oleh computer, manusia pun dapat mengerti kodekode tersebut hanya tersusun dari angka nol dan satu yang berderet panjang sekali. Jadi computer sudah mengenal istilah bahasa, yaitu serangakaian kombinasi kode yang digunakan untuk menyampaikan rumus. Didalam bahasa program, perintahperintah/rumus-rumus kebanyakan dituliskan dalam bahasa

inggris. Dari bahasa inggris yang sama itu telah dibuat orang berbagai macam cara memberikan perintah pada computer (Wahyudi, Berlian, & Rosdiana, 2013).

2.3.1 Extensible Markup Language (XML)

XML singkatan dari *eXtensible Markup Language*, merupakan bahasa *markup* yang memiliki nilai lebih dibandingkan HTML. XML merupakan penyederhanaan dari SGML *(Standard Generalized Markup Language)* dan direkomendasikan oleh W3C pada 10 Febuari 1998.

XML bukan merupakan pengganti HTML, namun merupakan pelengkap HTML. Masingmasing dikembangkan untuk tujuan yang berbeda. HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri.

Dalam sebuah jaringan internet, terdapat jaringan-jaringan kecil di dalamnya yang memiliki perbedaan sistem dan format data. Salah satu keunggulan XML adalah menyederhanakan pertukaran data dalam sistem dan format data yang berbeda-beda. Fungsi dari bahasa pemrograman XML yaitu (Wahyuningrum, 2012):

1. **Digunakan untuk Pertukaran Data**

   Dengan bahasa pemrograman XML, data dapat dipertukarkan antar suatu system yang tidak compatible. Dalam kenyataan yang sebenarnya sistem komputer dan basis data akan menyimpan suatu data dalam format yang berbeda-beda.

   Satu hal yang akan menyita waktu pengembangan sebuah sistem adalah bagaimana cara mempertukarkan data antar sistem dalam dunia internet. Mengonversi data ke XML dapat mengurangi kompleksitas dan membuat data dapat dibaca oleh berbagai tipe aplikasi.

2. **XML dan B2B**

   Dengan bahasa pemrograman XML, informasi keuangan dapat dipertukarkan dalam lingkungan internet. Dalam jangka waktu yang tidak terlalu lama lagi kita kana melihat bagaimana B2B (business to business) dapat menerapkan XML dan menjadi bahasa utama untuk pertukana informasi kauangan dan bisnis.

3. **XML untuk Pemakaian Data Bersama**

   Dengan bahasa pemrograman XML, file teks dapat digunakan secara bersama. Pada dasarnya XML dapat menyimpan data dalam format teks (*plain text*), karena itu XML dapat memungkinkan suatu cara untuk pemakaian bersama tanpa tergantung

dengan perangkat lunak maupun perangkat keras yang digunakan. Hal ini memudahkan pembuatan data yang sama bagi aplikasi yang berbeda.

Hal ini juga memudahkan untuk mengembangkan suatu sistem walaupun misalnya, kita harus meng'upgrade' sistem opersai yang digunakan, *server,* aplikasi ataupun *browser*.

4. **XML untuk Menyimpan Data**

   Dengan bahasa pemrograman XML, sebuah file teks dapat digunakan untuk menyimpan data. XML juga dapat digunakan untuk menyimpan data dalam format basis data. Aplikasi lain dapat digunakan untuk menulis atau mengambil dari format XML. Setelah itu kita dapat mendesain aplikasi yang bersifat umum (dengan HTML) yang berfungsi untuk menampilakn data saja.

5. **XML Meningkatkan Penggunaan Data**

   Bahasa pemrograman XML didesain secara mandiri terhadap perangkat lunak, perangkat keras dan aplikasi sehinggga memungkinkan data kita dapat diakses oleh semua orang. Kita juga dapat membuat data tertentu dapat diakses oleh berbagai macam standar browser HTML. Klien dan aplikasi lain dapat mengakses file XML sebagai sumber data, sebagaimana akses dilakukan terhadap basis data.

6. **XML untuk Membuat Bahasa Lain**

   Bahasa pemrograman XML adalah 'ibu' dari WAP (*Wireless Application Protocol*) dan juga WML (*Wireless Markup Language*). WAP dan WML digunakan sebagai aplikasi internet untuk peralatan genggaman bergerak seperti *mobile phones* yang ditulis dalam XML.

2.4 Karakteristik Perancangan Protokol VANET

Ada beberapa karakteristik VANET yang mempengaruhi perancangan protokol-protokol yang sesuai untuk VANET, yaitu (Aziza et al., 2017):

1. **Jumlah Node**

   Jumlah dan kepadatan node pada VANET bervariasi. Jumlah node bisa sangat besar di daerah perkotaan pada jam sibuk. Namun jumlah node VANET bisa sangat sedikit di daerah pedesaan dengan jumlah mobil yang dilengkapi kemampuan komunikasi sangat terbatas. Protokol MAC harus mampu bekerja dengan baik pada kedua keadaan tersebut.

2. **Mobilitas Node yang Tinggi**

Kendaraan sebagai node dari VANET dapat bergerak dengan kecepatan tinggi, lebih dari 100 km/jam. Hal ini sering menyebabkan diskoneksi antar node. Jika, dimisalkan, suatu kendaraan yang melaju kencang berkomunikasi dengan kendaraan yang berjalan lambat, maka link komunikasi yang terbentuk akan berumur pendek.

Hubungan akan terputus karena jarak yang memisahkan kedua node tersebut semakin besar. Protokol MAC harus dapat bekerja baik dengan umur link komunikasi yang singkat.

3. **Topologi Jaringan yang Berubah-ubah**

   Karena node-node VANET memiliki mobilitas tinggi, maka topologi jaringan VANET yang terbentuk juga terus-menerus mengalami perubahan.

4. **Topologi Jaringan yang Dapat Diprediksi**

   Walaupun node-node VANET banyak bergerak, namun pergerakannya dibatasi oleh topologi jalan raya yang dilaluinya.

5. **Ketersediaan Informasi Lokasi**

   Informasi lokasi dari node yang diberikan oleh GPS dapat mengurangi waktu pengiriman data dan meningkatkan throughput jaringan. Selain itu, informasi posisi dan kecepatan node-node melaju juga dapat membantu memperkirakan pola mobilitas dari node-node tersebut.

6. **Metode Komunikasi**

   Metode yang digunakan untuk pengiriman pesan dari pengirim ke penerima adalah broadcast.

7. **Lingkungan Komunikasi yang Berbeda**

   VANET memiliki dua tipe lingkungan komunikasi yang berbeda, yaitu: lingkungan trafik jalan raya (highway) dan lingkungan trafik perkotaan. Tipe yang pertama lebih sederhana dibandingankan tipe trafik di perkotaan.

8. **Tidak Memiliki Pusat Pengendali**

   Node-node pada VANET dapat saling terhubung dan berkomunikasi tanpa adanya pengendalian yang terpusat.

9. **Sumber Tenaga/batere yang Tak Terbatas**

   Dibandingkan node-node MANET yang memiliki daya listrik terbatas, node VANET memiliki daya yang tidak terbatas. Batere dari kendaraan yang melaju mampu menyediakan daya terus-menerus untuk mengaktifkan peralatan komunikasi node tersebut.

2.5 Arsitektur VANET

2.5.1 Komponen Utama

Menurut draft dari C2C-CC, komponen penyusun VANET terdiri dari 3 domain berbeda, yaitu: in-vehicle, adhoc, dan infrastruktur. Domain in-vehicle meliputi komponen yang terletak di dalam kendaraan, yaitu on-board unit (OBU) dan application unit (AU). Jumlah AU dalam suatu kendaraan bisa lebih dari satu. AU dapat berupa bagian dari kendaraan yang secara permanen terhubung ke OBU atau berupa laptop, PDA, atau game pad yang terhubung dengan OBU secara dinamis.

Koneksi antara AU dan OBU dapat melalui kabel, bluetooth, WUSB, maupun UWB. Domain kedua adalah domain ad-hoc. Domain ini terdiri dari kendaraan-kendaraan yang telah dilengkapi dengan OBU dan unit-unit komunikasi pendukung di sepanjang jalan (RSU). Tugas utama dari RSU adalah sebagai berikut (Aziza et al., 2017):

1. Memperluas rentang komunikasi dari jaringan dengan mengirimkan kembali informasi ke OBU-OBU yang berada dalam wilayahnya.
2. Bisa menjalankan aplikasi-aplikasi pendukung keselamatan di jalan raya, pengendali persimpangan jalan, lampu lalu lintas virtual, sekaligus bertindak sebagai sumber dan penerima informasi.
3. Menyediakan hubungan ke internet bagi OBU-OBU.
4. Bekerja sama dengan RSU-RSU lain untuk meneruskan informasi yang terkait dengan keselamatan jalan raya.

OBU dari beberapa kendaraan akan membentuk jaringan nirkabel ad-hoc (MANET). OBU-OBU yang terhubung secara nirkabel akan langsung dapat saling berkomunikasi tanpa adanya koordinasi secara terpusat. Jika tidak ada koneksi langsung, protokol routing untuk VANET memungkinkan komunikasi secara multi-hop, yaitu data diteruskan dari satu OBU ke OBU lainnya hingga ke tujuan. OBU dan RSU dapat dianggap sebagai node-node dari VANET. RSU dapat terhubung dengan jaringan infrastruktur komunikasi sehingga RSU dapat terhubung dengan Internet. Hal ini memungkinkan berbagai AU yang terhubung pada suatu OBU untuk berkomunikasi dengan host-host di Internet. OBU dapat juga terhubung ke Internet melalui Hot Spot (HS). Apabila tidak tersedia RSU dan HS, OBU dapat memanfaatkan teknologi komunikasi lain untuk aplikasi yang tidak berhubungan dengan keselamatan lalu lintas, misalnya menggunakan GSM, GPRS, UMTS, 4G, dan lain sebagainya. Domain infrastruktur pada VANET mengacu pada berbagai piranti pendukung terlaksananya komunikasi lainnya yang ada di sepanjang jalan (Aziza et al., 2017)

2.5.2 Arsitektur Komunikasi

Ada empat tipe komunikasi pada VANET, yaitu komunikasi In-vehicle, Vehicle-to-vehicle (V2V), Vehicle-toroad infrastructure (V2I atau V2R), dan Vehicle-to-broadband (V2B) (Liang et al., 2015). Komunikasi In-vehicle menyediakan berbagai informasi yang berhubungan dengan kejadian di dalam suatu kendaraan, misalnya: kecepatan dan posisi kendaraan, informasi apakah pengendara mengantuk atau mengalami kelelahan. Komunikasi V2V mengatur bagaimana kendaraan-kendaraan dapat saling bertukar informasi dan pesan peringatan yang berhubungan dengan lalu lintas. Komunikasi V2I menghubungkan kendaraan dengan berbagai perangkat infrastruktur, termasuk RSU. V2I memungkinkan pengendara menerima informasi tentang kondisi lalu lintas, cuaca, dan informasi lain dari sensor yang tergabung dalam VANET secara real time. Tipe komunikasi V2B menghubungkan kendaran dengan jaringan wireless broadband, seperti 3G/4G. Dari gambar 1 di atas, dapat diketahui bahwa komunikasi V2V dilaksanakan menggunakan standar wireless atau protocol yang khusus dikembangkan untuk VANET, seperti IEEE 802.11p. Sedangkan, komunikasi dari kendaraan ke RSU atau ke perangkat infrastruktur jaringan yang lain dapat dilakukan menggunakan berbagai teknologi wireless lain yang tersedia saat ini (Aziza et al., 2017).

BAB III
METODOLOGI

3.1 Kebutuhan Sistem

Pada tahapan kebutuhan sistem langkah pertama yang dilakukan adalah pengumpulan data dan informasi dari buku-buku referensi, modul, jurnal serta artikel-artikel baik yang disediakan oleh perguruan tinggi ataupun yang di luar dari perguruan tinggi yang sesuai dengan objek penelitian. Adapun *software* yang harus terinstal pada perangkat yang akan dugunakan dalam penelitian ini, yaitu:

1. **Omnet++**

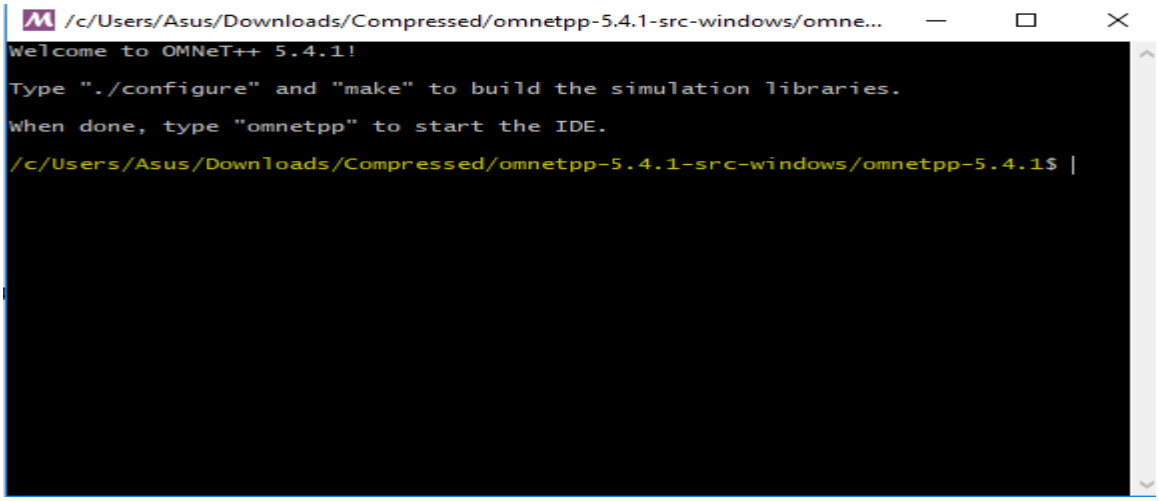
   Omnet++ merupakan seubauh program open source simulator VANET yang memungkinkan pengguna untuk membangun simulasi jaringan dengan tampilan sederhana dengan hasil yang detail.

2. **SUMO**

   Simulation of Urban Mobility atau biasa dikenal dengan SUMO merupakan sebuah program open source simulator lalu lintas jalan yang memungkinkan pengguna untuk membangun simulasi pergerakan kendaraan pada topologi jaringan VANET yang disesuaikan. Penelitian ini menggunakan generator SUMO karena dapat mensimulasikan VANET pada RealMap.

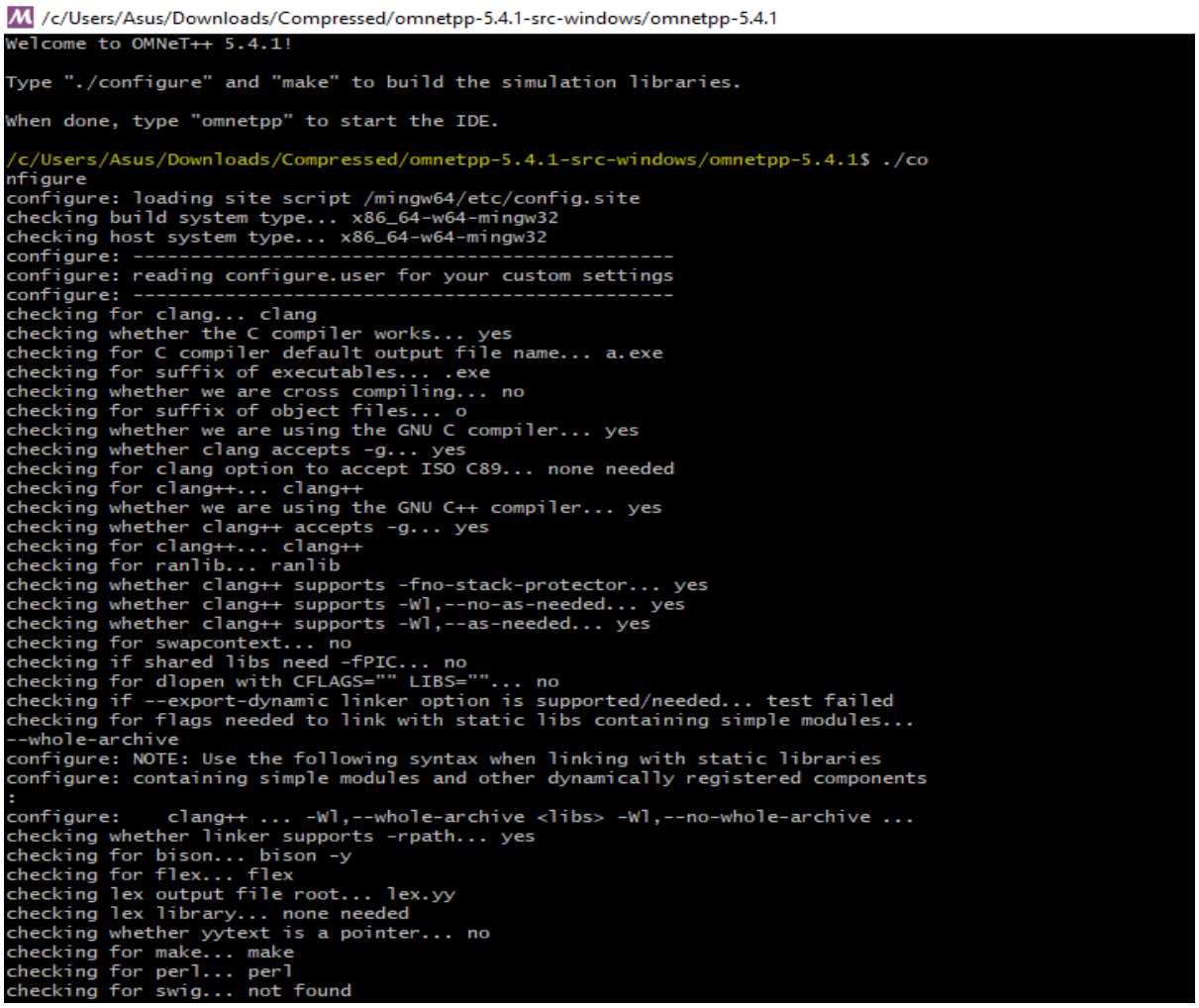
3.1.1 Langkah Menginstall Omnet++

Langkah dalam melakukan instalasi Omnet++ diawali dengan meng-*extract* file Omnet++ yang telah di download, setelah file slesai di-*extract* buka file tersebut dan buka file mingwenv.exe dan akan muncul tapilan seperti pada gambar 3.1.



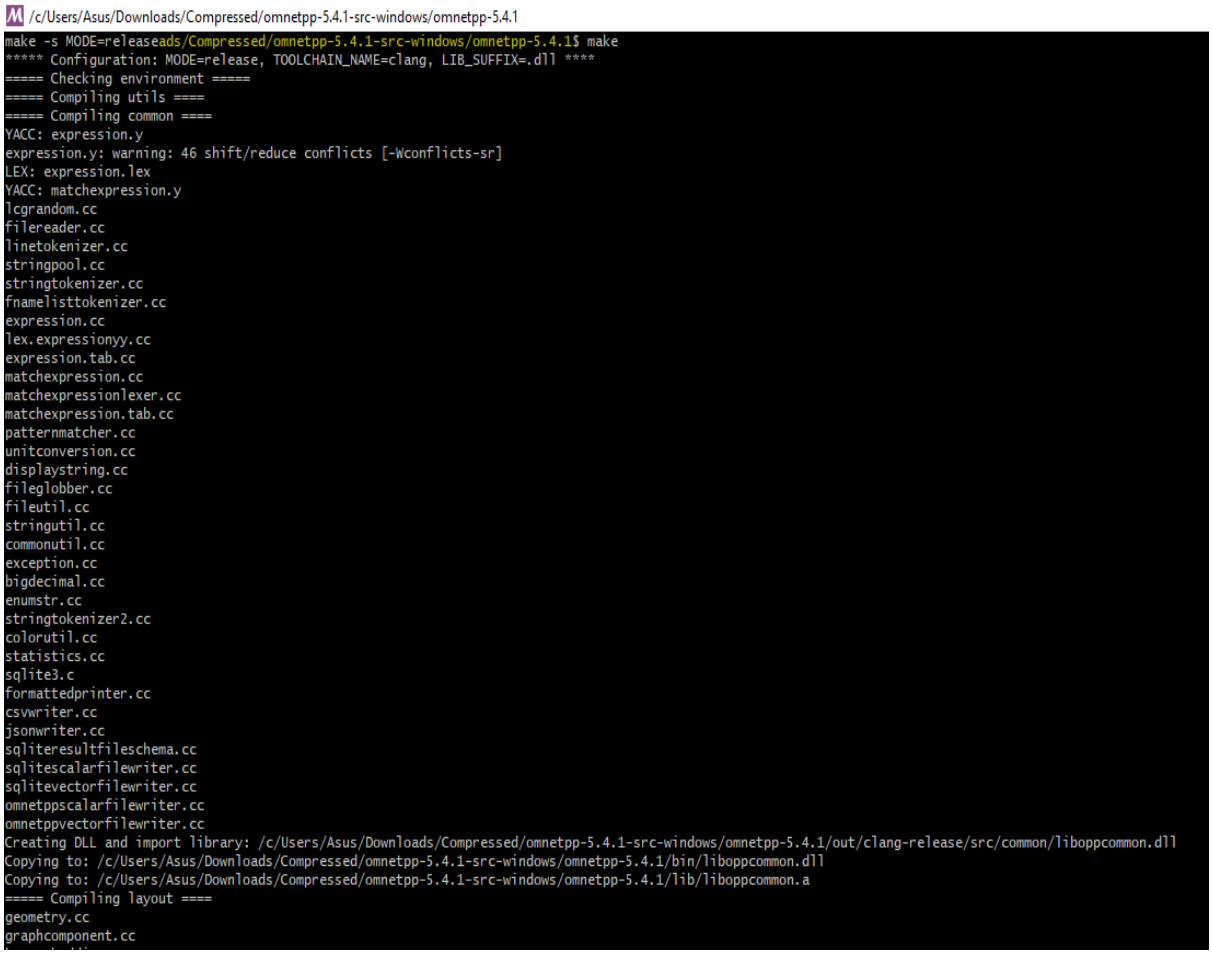
Gambar 3.1 Tampilan awal Omnet++

Tahapan selanjutnya kita diperintahkan untuk melakukan “configure” dan “make” terlebidahulu. Proses dan *Source code* yang digunakan untuk melakukan “configure” seperti pada gambar 3.2.



Gambar 3.2 Proses configure pada Omnet++

Tahapan selanjutnya setelah proses configure selesai adalah melakukan perintah “make” yang dapat dilihat pada gambar 3.3.



Gambar 3.3 Proses “make” pada Omnet++

Proses “make” ini memang terbilang lama dan tahap paling lama dalam instalasi Omnet++, Setelah proses ini slesai maka Omnet++ sudah siap untuk digunakan.

3.2 Perancangan Sistem

Pada penelitian ini akan dilakukan simulasi VANET menggunakan RealMap dengan generator *Simulation Of Urban Mobility* (SUMO) dan Omnet++. Perancangan skenario mobilitas ujicoba VANET akan dimulai dari perancangan peta pergerakan *node,* pembuatan rute lalu lintas yang diambil dari *OpenStreetMap* dan Pengimplementasian pergerakan. Bagian perancangan pada peta pergrekan *node* menggunakan skenario peta riil yang memggunakan sebuah lingkungan lalu lintas yang nyata.

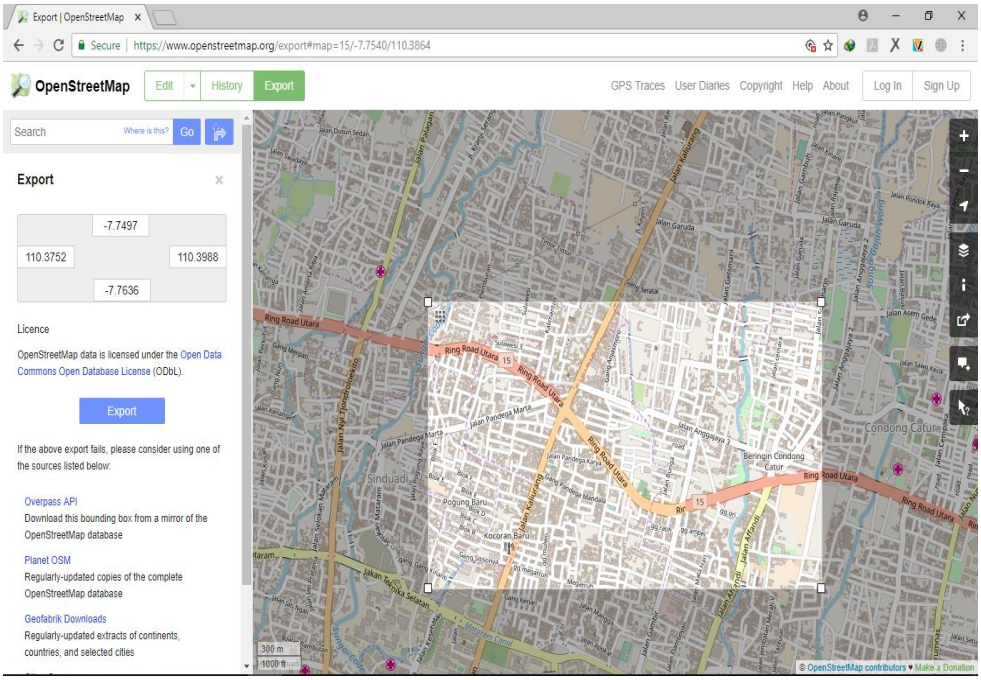
Pada penelitian ini, digunakan lingkungan lalu lintas di daerah Perempatan Jl. Kaliurang dengan Ring Road Utara dan Perempatan Jl. Affandi dengan Ring Road Utara sebagai peta riil. Peta riil yang dimaksud hanya berupa jalur-jalur tanpa adanya infrastruktur.

3.2.1 Pembuatan Peta Pada SUMO

1. Tahap awal pembuatan peta riil adalah dengan mencari daerah *real world* yang akan kita gunakan dalam simulasi VANET dengan mobilitas yang tinggi sehingga cocok

untuk jaringan VANET. Pada penelitian ini, diperoleh daerah Perempatan Jl. Kaliurang dengan Ring Road Utara dan Perempatan Jl. Affandi dengan Ring Road Utara sebagai objek peta riil. Proses pembentukan skenario diawali dengan mengunakan *OpenStreetMap* untuk melakukan *caputre* peta. Berikut beberapa langkah membuat peta *RealMap* pada SUMO:

2. Pertama kita harus mengexport map yang akan kita gunakan untuk simulasi VANET di openstreetmap.org seperti gambar 3.4.



Gambar 3.4 Tampilan website openstreetmap.org

Setelah bagian yang ingin kita gunakan sudah ditetapkan makanya selajutnya kita tekan tombol export dan kita akan mendapatnya file .osm yang akan kita gunakan untuk melakukan simulasi VANET.

3. Kedua kita akan menyiapkan file polygon yang berisi bentuk geometris (polygon atau tempat menarik) dan kita simpan pada satu folder dengan file .osm yang telah kita download tadi. Nama dari file polygon ini bebas dan harus menggunakan .xml untuk menyimpannya agar dapat dibaca oleh generator SUMO. Berikut *source code* file polygon dapat kita lihat pada gambar 3.5.

Judul : Typemap.xml
<polygonTypes>

```
<polygonType id="waterway"              name="water"      color=".71,.82,.82" layer="-4"/>
<polygonType id="natural"               name="natural"    color=".55,.77,.42" layer="-4"/>
<polygonType id="natural.water"         name="water"      color=".71,.82,.82" layer="-4"/>
<polygonType id="natural.wetland"       name="water"      color=".71,.82,.82" layer="-4"/>
<polygonType id="natural.wood"          name="forest"     color=".55,.77,.42" layer="-4"/>
<polygonType id="natural.land"          name="land"       color=".98,.87,.46" layer="-4"/>

<polygonType id="landuse"               name="landuse"    color=".76,.76,.51" layer="-3"/>
<polygonType id="landuse.forest"        name="forest"     color=".55,.77,.42" layer="-3"/>
<polygonType id="landuse.park"          name="park"       color=".81,.96,.79" layer="-3"/>
<polygonType id="landuse.residential"   name="residential" color=".92,.92,.89" layer="-3"/>
<polygonType id="landuse.commercial"    name="commercial"  color=".82,.82,.80" layer="-3"/>
<polygonType id="landuse.industrial"    name="industrial"  color=".82,.82,.80" layer="-3"/>
<polygonType id="landuse.military"      name="military"   color=".60,.60,.36" layer="-3"/>
<polygonType id="landuse.farm"          name="farm"       color=".95,.95,.80" layer="-3"/>
<polygonType id="landuse.greenfield"    name="farm"       color=".95,.95,.80" layer="-3"/>
<polygonType id="landuse.village_green" name="farm"       color=".95,.95,.80" layer="-3"/>

<polygonType id="tourism"               name="tourism"    color=".81,.96,.79" layer="-2"/>
<polygonType id="military"              name="military"   color=".60,.60,.36" layer="-2"/>
<polygonType id="sport"                 name="sport"      color=".31,.90,.49" layer="-2"/>
<polygonType id="leisure"               name="leisure"    color=".81,.96,.79" layer="-2"/>
<polygonType id="leisure.park"          name="tourism"    color=".81,.96,.79" layer="-2"/>
<polygonType id="aeroway"               name="aeroway"    color=".50,.50,.50" layer="-2"/>
<polygonType id="aerialway"             name="aerialway"  color=".20,.20,.20" layer="-2"/>

<polygonType id="shop"                  name="shop"       color=".93,.78,1.0" layer="-1"/>
<polygonType id="historic"              name="historic"   color=".50,1.0,.50" layer="-1"/>
<polygonType id="man_made"              name="building"   color="1.0,.90,.90" layer="-1"/>
<polygonType id="building"              name="building"   color="1.0,.90,.90" layer="-1"/>
<polygonType id="amenity"               name="amenity"    color=".93,.78,.78" layer="-1"/>
<polygonType id="amenity.parking"       name="parking"    color=".72,.72,.70" layer="-1"/>
<polygonType id="power"                 name="power"      color=".10,.10,.30" layer="-1"
```

```
discard="true"/>
<polygonType id="highway"              name="highway"    color=".10,.10,.10" layer="-1"
discard="true"/>

<polygonType id="boundary" name="boundary"    color="1.0,.33,.33" layer="0" fill="false"
discard="true"/>
<polygonType id="admin_level" name="admin_level"    color="1.0,.33,.33" layer="0" fill="false"
discard="true"/>
</polygonTypes>
```

Text 3.1 *sorce code* file polygon

4. Langkah berikutnya harus membuat file configuration, file ini sebagai file inti dari configurasi pada simulasi VANET, file ini yang akan dibuka saat menjalankan simulasi VANET, jadi didalam file ini terdapat *source code* yang dapat memanggil file lain yang diperlukan untuk menjalankan simulasi VANET. File ini bebas diberi nama tetapi harus dibuat menjadi bentuk file .cfg yang dapat dijalankan pada generator SUMO. Berikut *sorce code* file .cfg dapat dilihat pada gambar 3.6

Judul:
map.sumo.cfg

```
<?xml version="1.0" encoding="iso-8859-1"?>

<configuration  xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
xsi:noNamespaceSchemaLocation=”http://sumo.sf.net/xsd/sumoConfiguration.xsd”>

   <input>
      <net-file value=”map.net.xml”/>
      <route-files value=”map.rou.xml”/>
      <additional-files value=”map.poly.xml”/>
   </input>

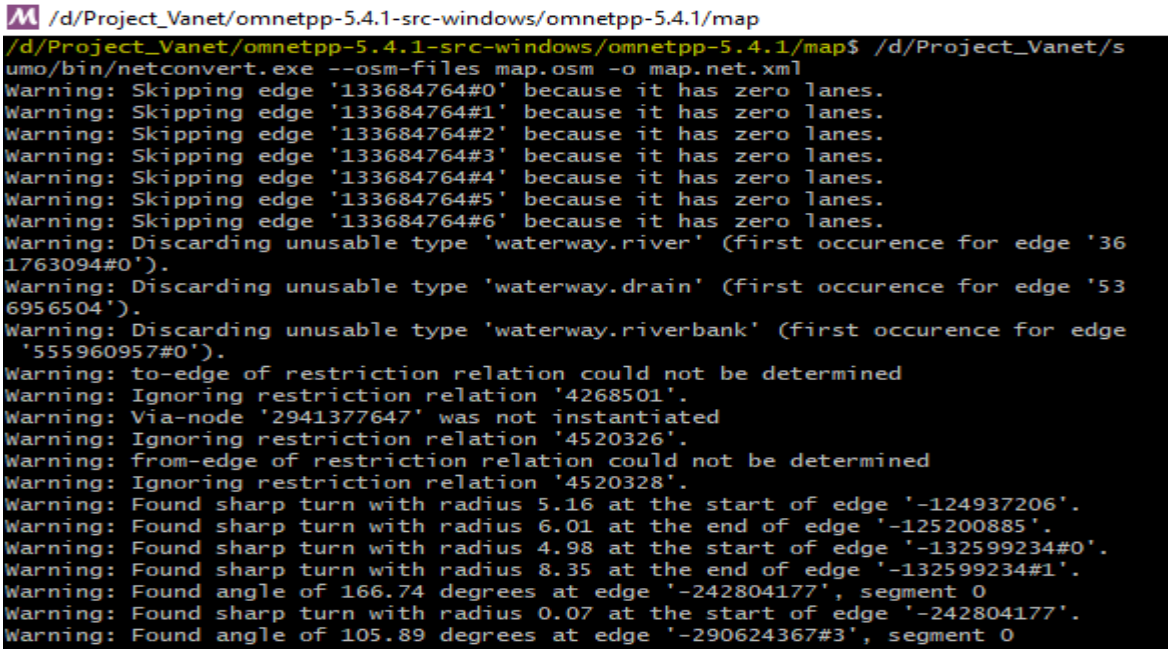
   <time>
      <begin value=”0”/>
      <end value=”1000”/>
   </time>
```

```
<!—
   <report>
      <no-duration-log value=”true”/>
      <no-step-log value=”true”/>
   </report>

</configuration>
```

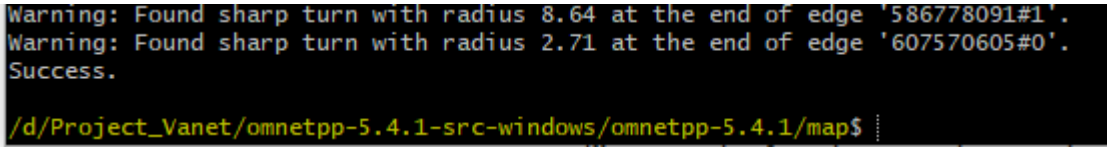
Text 3.2 *source code* file .cfg

5. Langkah selanjutnya adalah mengubah bentuk file .osm yang telah didownload tadi menjdi bentuk file .xml dengan cara menggunakan netconvert.exe pada SUMO agar bentuk file .osm yang didownload tadi dapat dibaca oleh generator SUMO. Adapun langkah melakukan konfersi file .osm ke file .xml dapat dilihat dari gambar dibawah ini.



Gambar 3.5 *source code* dan proses konfersi

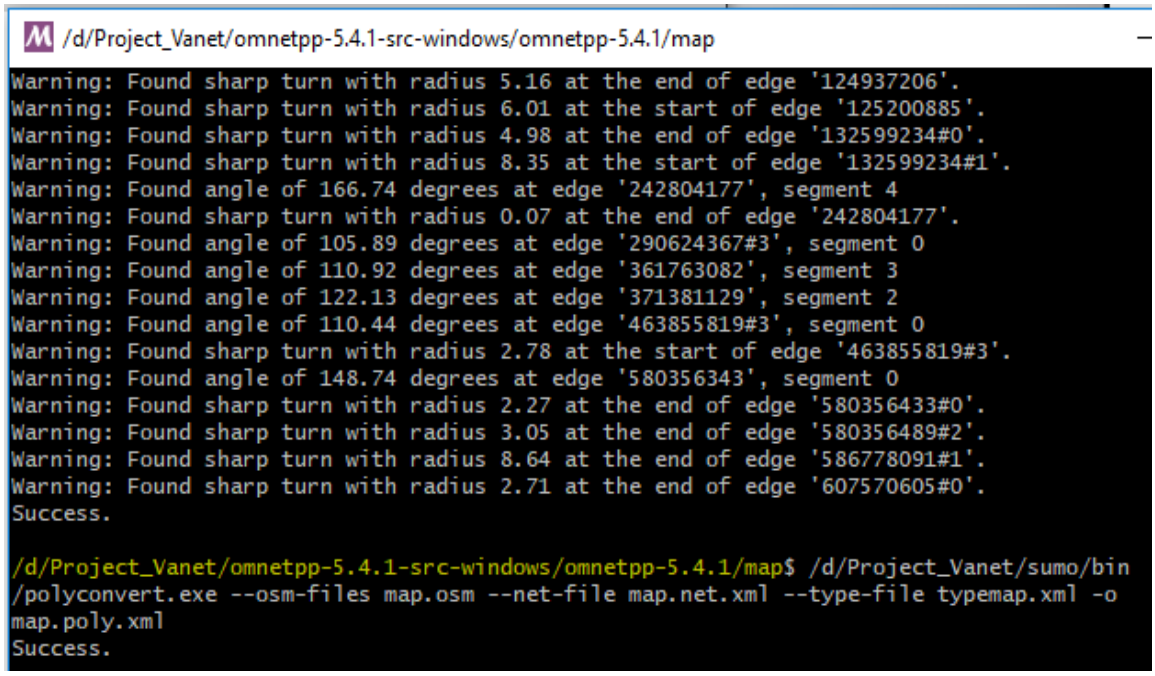
Pada gambar 3.7 terdapat *source code* *“*/d/Projek_Vanet/sumo/bin/netconvert.exe –osm-files map.osm –o map.net.xml” yang berfungsing unutk mengkonfersi file map.osm menjadi file map.net.xml. untuk melihat apakah proses konfersi berhasil atau tidak dapat dilihat pada gambar 3.8.



Gambar 3.6 Proses konfersi file berhasil

Jika terdapat kata "*Success*" diakhir proses maka proses dari konfersi file telah berhasil dan jika diakhir proses terdapat kata "*error*" maka konfersi file telah gagal.

6. Langkah selanjutnya adalah meng-*import* file *type.xml* tadi menjadi bentuk *polygon* yang dapat dibaca secara sempurna oleh generator SUMO dengan menggunakan file *polyconvert.exe*. Langkah ini bertujuan untuk membuat suatu file *polygon* yang sama dengan file .osm agar terlihat nyata saat menjalankan simulasi VANET. Adapun *source code* untuk melakukan polyconvert dapat dilihat pada gambar 3.9.



Gambar 3.7 *source code* dan proses polyconvert

7. Langkah selanjutnya adalah mengkonfersi file *map.net.xml* untuk menentukan sumber dan tujuan node secara acak yang sesuai dengan file .osm agar pergerakan simulasi terlihat nyata pada RealMap yang telah digunakan. Langkah ini mengunakan file *randaomtrip.py* yang telah disediakan oleh SUMO dan akan menghasilkan file *trips.trips.xml* dan *map.rou.xml*. Berikut *source code* dan prosesnya dapat kita lihat pada gambar 3.10.

```
/d/Project_Vanet/omnetpp-5.4.1-src-windows/omnetpp-5.4.1/map$ python /d/Project_Vanet/sumo/tools/rando
mTrips.py -n map.net.xml -r map.rou.xml -e 100 -l
Success.
calling  D:\Project_Vanet\sumo\bin/duarouter -n map.net.xml -r trips.trips.xml -o map.rou.xml --ignore
-errors --begin 0 --end 100.0 --no-step-log --no-warnings
```

Gambar 3.8 *source code* dan proses rendomtrips

8. Jika semua proses di atas berhasil maka akan menghasilkan beberapa file yang akan digunakan untuk melakukan simlasi VANET, file tersebut adalah *map.osm, map.net.xml, map.poly.xml, map.rou.alt.xml, map.rou.xml, trips.trips.xml, typemap.xml dan map.sumo.cfg.*

3.2.2 Perancangan Simulasi VANET pada Omnet++ yang di Import dari VEINS

1. Tahap awal dalam perancangan simulasi pada *Omnet++* adalah dengan mendownload veins dengan versi apapun yang akan digunakan dalam pemanggilan simulasi VANET pada Omnet++.
2. Tahap berikutnya, yaitu mengkonfigurasikan aplikasi sumo dengan project erlangen.sumo.cfg dengan *source code* pada gambar 3.11

```
/d/Project_Vanet/veins-4.7.1/veins-veins-4.7.1/examples/veins$ /d/Project_Vanet/
sumo/bin/sumo.exe -c erlangen.sumo.cfg
Loading configuration... done.
```

Gambar 3.9 *Source code*

3. Tahap berikutnya adalah mengkonfigurasikan veins agar dapat di jalankan pada Omnet++ dengan *source code* pada gambar 3.12

```
/d/Project_Vanet/veins-4.7.1/veins-veins-4.7.1/examples/veins$ /d/Project_Vanet/
veins-4.7.1/veins-veins-4.7.1/sumo-launchd.py -vv -c /d/Project_Vanet/sumo/bin/s
umo.exe
Logging to c:/users/asus/appdata/local/temp/sumo-launchd.log
Listening on port 9999
```

Gambar 3.10 *Source code*

4. Setelah semua berhasil maka project veins dapat dijalankan pada Omnet++.

BAB IV
HASIL DAN PEMBAHASAN

Bab ini membahas tentang hasil sistem jaringan VANET dan pembahasan penelitian yang dilakukan untuk dapat mengetahui manfaat dari penggunaan sistem ini. Pada bab ini dibagi menjadi beberapa subbab sesuai dengan parameter-parameter yang dibutuhkan.

4.1 Hasil

4.1.1 Simulasi VANET pada Generator SUMO

Hasil simulasi VANET pada generator SUMO ini dapat membuat kendaraan-kendaraan dapat berkomunikasi melalui V2V (*Vehicle to Vehicle*) dan V2I (*Vehicle to Infrastructur*), bertujuan untuk meningkatkan keselamatan mengemudi dan manajemen lalu lintas sementara bagi para pengemudi dan penumpang.

Dalam VANET, RSU (*Road Side Unit*) dapat memberikan bantuan dalam menemukan fasilitas seperti restoran dan pom bensin serta mem-*broadcast* pesan yang terkait seperti kecepatan kendaraan kepada pengendara lain.

Hal ini dapat mengurangi terjadinya kecelakaan. Melalui komunikasi V2V, pengendara bisa mendapatkan informasi yang lebih cepat dan lebih baik serta mengambil tindakan awal untuk menghadapi situasi yang abnormal.

Untuk mencapai hal ini, suatu OBU (*On-Board Unit*) secara teratur menyiarkan pesan yang terkait dengan informasi dari posisi pengendara, waktu sekarang, arah pengemudian, kecepatan, lampu sen, percepatan/perlambatan, kondisi jalan.

SUMO terdiri dari banyak *tools* yang memiliki fungsi yang berbeda. Berikut merupakan penjelasan beberapa fungsi tools yang dipakai guna mendukung pembuatan penelitian ini:

1. **netgenerate.exe**

   NETGENERATE menghasilkan jaringan jalan abstrak yang dapat digunakan oleh aplikasi SUMO. Anda dapat menggunakan file definisi skema *XML* untuk menyiapkan konfigurasi *NETGENERATE*.

2. **netconvert.exe**

   NETCONVERT mengimpor jaringan jalan digital dari berbagai sumber dan menghasilkan jaringan jalan yang dapat digunakan oleh alat lain dari paket tersebut. *NETCONVERT* adalah aplikasi baris perintah. Ini mengasumsikan setidaknya satu parameter - kombinasi dari nama jenis file untuk diimpor sebagai nama parameter dan

nama file yang akan diimpor sebagai nilai parameter. Jadi, *netconvert* ini berfungsi untuk mengimpor jaringan dari *OpenStreetMap*.

3. **duarouter.exe**

   DUAROUTER mengimpor definisi permintaan yang berbeda, menghitung rute kendaraan yang dapat digunakan oleh SUMO menggunakan penghitungan jalur terpendek; Ketika dipanggil secara iteratif *DUAROUTER*. *Output* utama *DUAROUTER* adalah file .rou.xml yang namanya ditentukan oleh opsi -o). Selain itu, *.rou.alt.xml* dengan awalan nama yang sama dengan file *.rou.xml* akan dihasilkan. Anda dapat menggunakan file definisi skema *XML* untuk menyiapkan konfigurasi *DUAROUTER*.

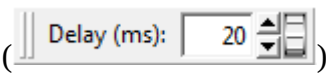
4. **sumo.exe**

   SUMO adalah simulasi itu sendiri; Ini adalah simulasi arus lalu lintas mikroskopik, ruang-kontinu, dan waktu-diskrit. Anda dapat menggunakan file definisi skema *XML* untuk mengatur konfigurasi SUMO.

5. **sumo-gui.exe**

   SUMO-GUI pada dasarnya aplikasi yang sama dengan SUMO, hanya diperpanjang oleh antarmuka pengguna grafis. Menjadi aplikasi berbasis Windows, SUMO-GUI dimulai dengan klik ganda dengan tombol kiri mouse pada Windows, di Linux mungkin dengan satu klik. Setelah ini, tampilan kosong akan muncul. Dengan menggunakan entri menu "*File-> Open Simulation* ..." atau dengan menggunakan "open" -icon (). Anda harus memilik file konfigurasi SUMO yang sudah ada, jika memiliki ekstensi yang tepat " *.sumocfg* ". Jika file konfigurasi SUMO salah, kesalahan dilaporkan, jika tidak, jaringan Anda yang direferensikan dalam file konfigurasi harus ditampilkan.

   Sekarang Anda dapat mulai mensimulasikan dengan menekan tombol "play" (). Simulasi bekerja seolah-olah dimulai pada baris perintah. Simulasi dapat dihentikan menggunakan tombol "stop" () dan dilanjutkan dengan menekan tombol "play" lagi. Ketika dihentikan, juga langkah-langkah tunggal dapat dilakukan dengan menekan tombol "satu langkah" (). Jika simulasi sedang berjalan, detik simulasi saat ini ditampilkan dibagian "digit digital Waktu:" (

). Dengan mengklik "Waktu:", tampilan dapat diubah antara menampilkan <detik> dan <jam: menit: detik>. Di sebelah tampilan waktu adalah kontrol Delay (

) ini

memungkinkan anda untuk memperlambat simulasi dengan menunggu jumlah milidetik yang diberikan di antara langkah-langkah simulasi.

Selain memuat konfigurasi simulasi, juga dimungkinkan untuk memuat jaringan dengan menggunakan entri menu "File-> Open Network ..." atau dengan menggunakan "open network" –icon (). Perlu diketahui, bahwa biasanya SUMO-GUI menganggap jaringan memiliki ekstensi ".net.xml", tetapi juga menerima ekstensi lainnya. simulasi yang dimuat atau jaringan yang dimuat dapat dimuat ulang menggunakan tombol "reload" () atau entri menu "File-> Reload". Jika jaringan atau simulasi dimuat, navigasi melalui jaringan dimungkinkan dengan mouse atau dengan keyboard. Dapat menyeret jaringan dengan tombol kiri mouse ditekan ke semua arah dan zoom baik dengan menggunakan roll mouse atau dengan menekan tombol mouse kanan dan menggerakkan mouse ke atas dan ke bawah. Untuk zoom berbutir halus (setengah kecepatan zoom) tekan tombol "Kontrol" saat menggunakan roda mouse, untuk penggunaan kecepatan ganda "Shift".

6. polyconvert.exe

POLYCONVERT mengimpor bentuk geometris (poligon atau tempat menarik) dari sumber yang berbeda, mengubahnya menjadi representasi yang dapat divisualisasikan menggunakan SUMO-GUI. *POLYCONVERT* dapat mengimpor bentuk dari berbagai jenis file. Biasanya, untuk mengimpor data jenis tertentu, nama jenis digunakan sebagai nama opsi dan nilai menunjukkan posisi file.

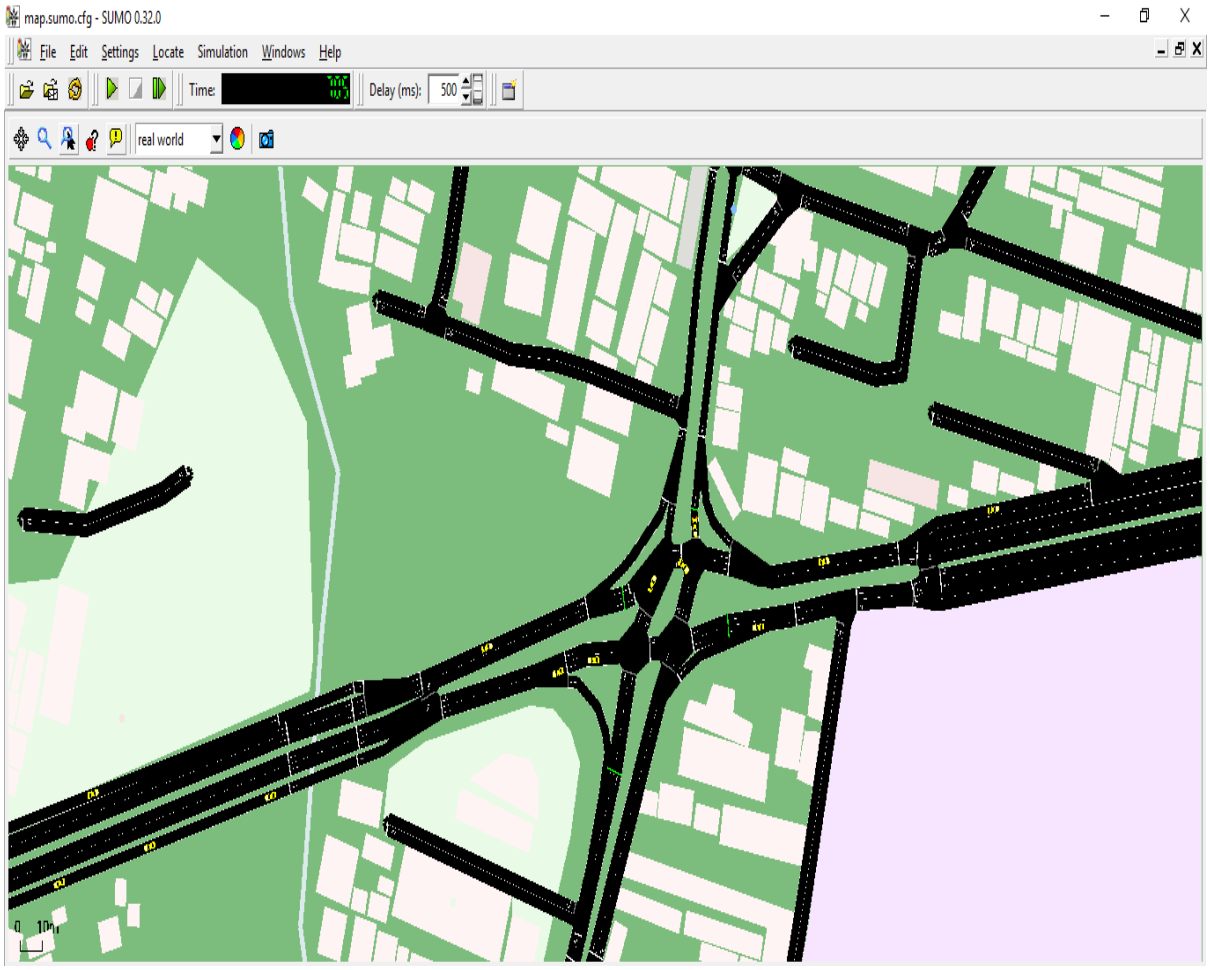
Adapun hasil parameter dari siulasi VANET mengunakan simulator SUMO dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Parameter VANET pada SUMO

Name	Value	Dynamic
Loaded vehicles [#]	958	✓
Departed vehicles [#]	949	✓
Running Vehicles [#]	155	✓
Arrived Vehicles [#]	794	✓
Avg. speed [m/s]	13.22	✓
Avg. relative speed	0.80	✓
End Time [s]	1000	x
Begin Time [s]	0	x
Nodes [#]	1078	x
Edges [#]	2558	x

Total Edge Length[km]	171.68	x
Total Lane Length [km]	182.29	x
Network Version	0.27	x
Area Similasi [m]	1000 x 1000	x
Model Skenario	Peta Riil	x

Tabel diatas menunjukkan hasil parameter dari simulasi VANET dengan menggunakan Generator SUMO, terdapat beberapa parameter yang penting untuk ditampilkan saat simulasi telah selesai dilakukan.

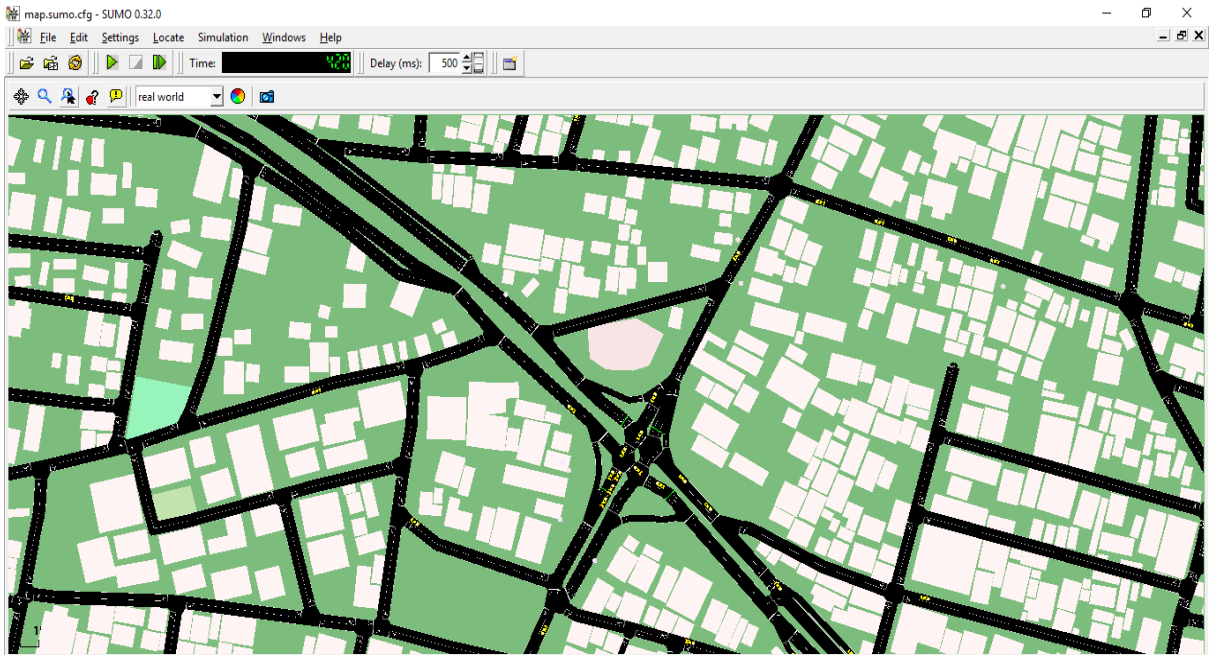


Gambar 4.1 Tampilan Simulasi Vanet

Pada gambar di atas kita dapat melihat simulasi VANET menggunakan generator SUMO yang disimulasikan pada *RealMap* yang menggunakan jalan yang terdapat di Yogyakarta tepatnya pada perempatan Jl. Kaliurang dengan Ring Road Utara dan membuat tampilan jalan, letak gedung dan kendaraan terlihat nyata.

Simulasi ini menggunakan routing protokol AODV yang berfungsi membentuk node/kendaraan yang dapat saling mengirimkan pesan dengan kecepatan tinggi ke kendaraan

yang lainnya dengan tujuan kendaraan yang telah ditentukan. Fungsi menggunakan *Routing Protocol* AODV yaitu agar setiap node/kendaraan yang bergerak tidak mengalami tabrakan dengan node/kendaraan yang lainnya dan mengurangi kemacetan di jalan raya.



Gambar 4.2 Tampilan Simulasi VANET

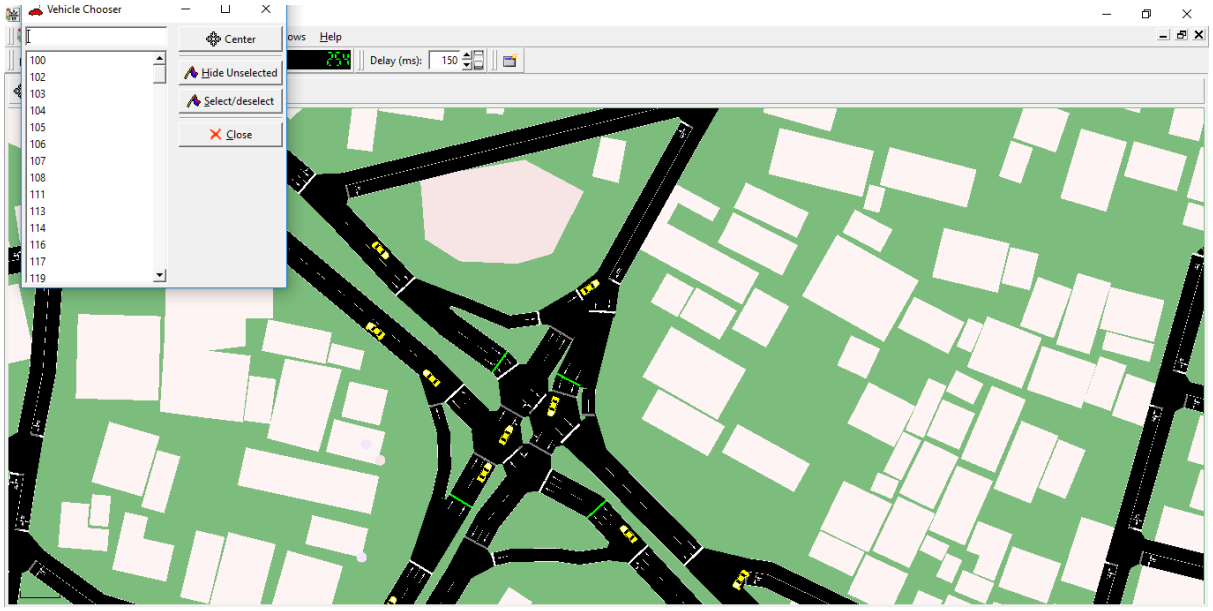
Pada gambar di atas kita dapata melihat simulasi VANET yang hampir sama dengan gambar sebelumnya, yang membedakan adalah letak *RealMap* yang digunakan. *RealMap* yang digunakan pada gambar di atas adalah perempatan Jl. Affandi dengan Ring Road Utara. Simulasi ini menggunakan peta *RealMap* agar lebih mudah dipahami.

Pada generator SUMO ini kita dapat mengatur *Delay* pengiriman pesan pada kendaraan, semakin besar *Delay* yang diberikan maka semakin lambat pergerakan kendaraan pada simulasi ini.

4.1.2 Melacak Kendaraan

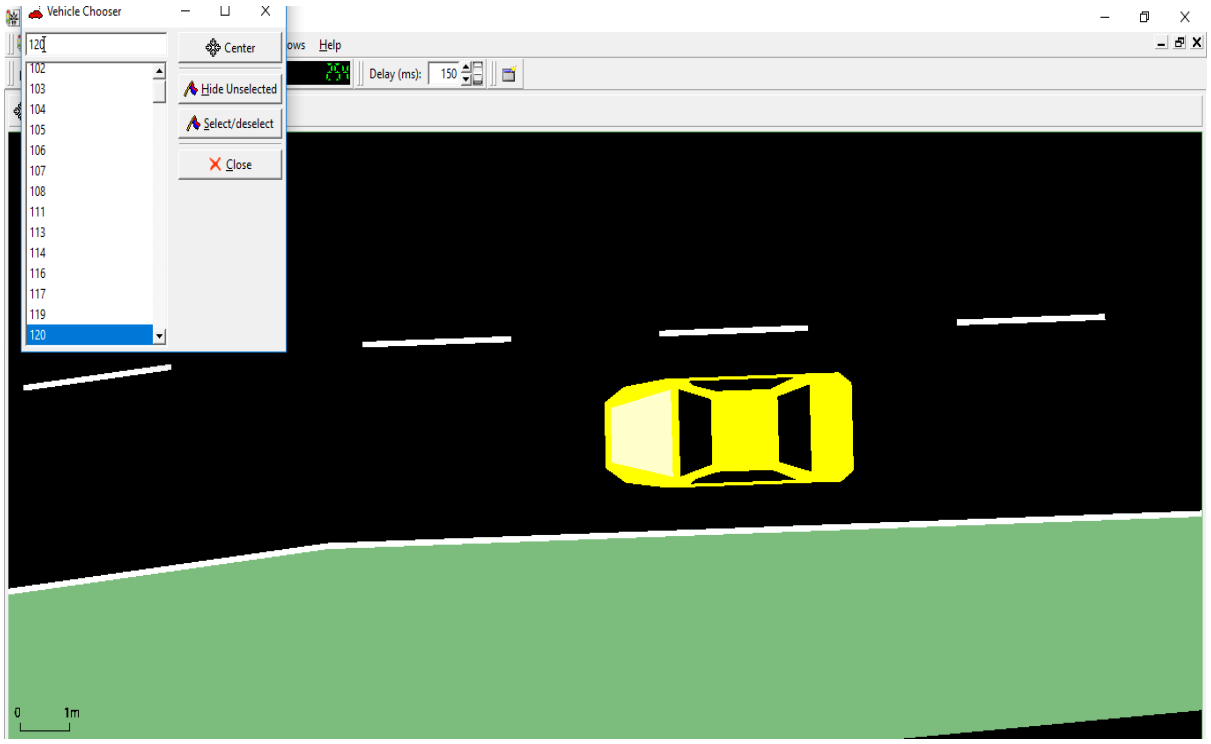
Hasil dari pembuatan simulasi VANET pada generator SUMO tidak hanya dapat membuat kendaraan saling bergerak dan berkomunikasi satu sama lain agar dapat mencegah terjadinya tabrakan data/kendaraan dan kemacetan, tetapi juga dapat melacak kendaraan.

Simulasi ini dapat melacak kendaraan yang sedang bergerak ataupun sedang berhenti. Jadi simulasi ini juga dapat digunakan untuk melakukan pengejaran suatau kendaraan karena dapat melacak kendaraan secara cepat dan tepat.



Gambar 4.3 Tampilan *Locate Vehicles*

Gambar 4.3 adalah tampilan daftar vehicles/kendaraan yang terdapat dalam simulasi VANET ini. Daftar *vehicles* dapat kita temukan dengan cara klik *tools Locate* lalu pilih *Locate Vehicles* maka akan muncul tampilan seperti gambar 4.3.



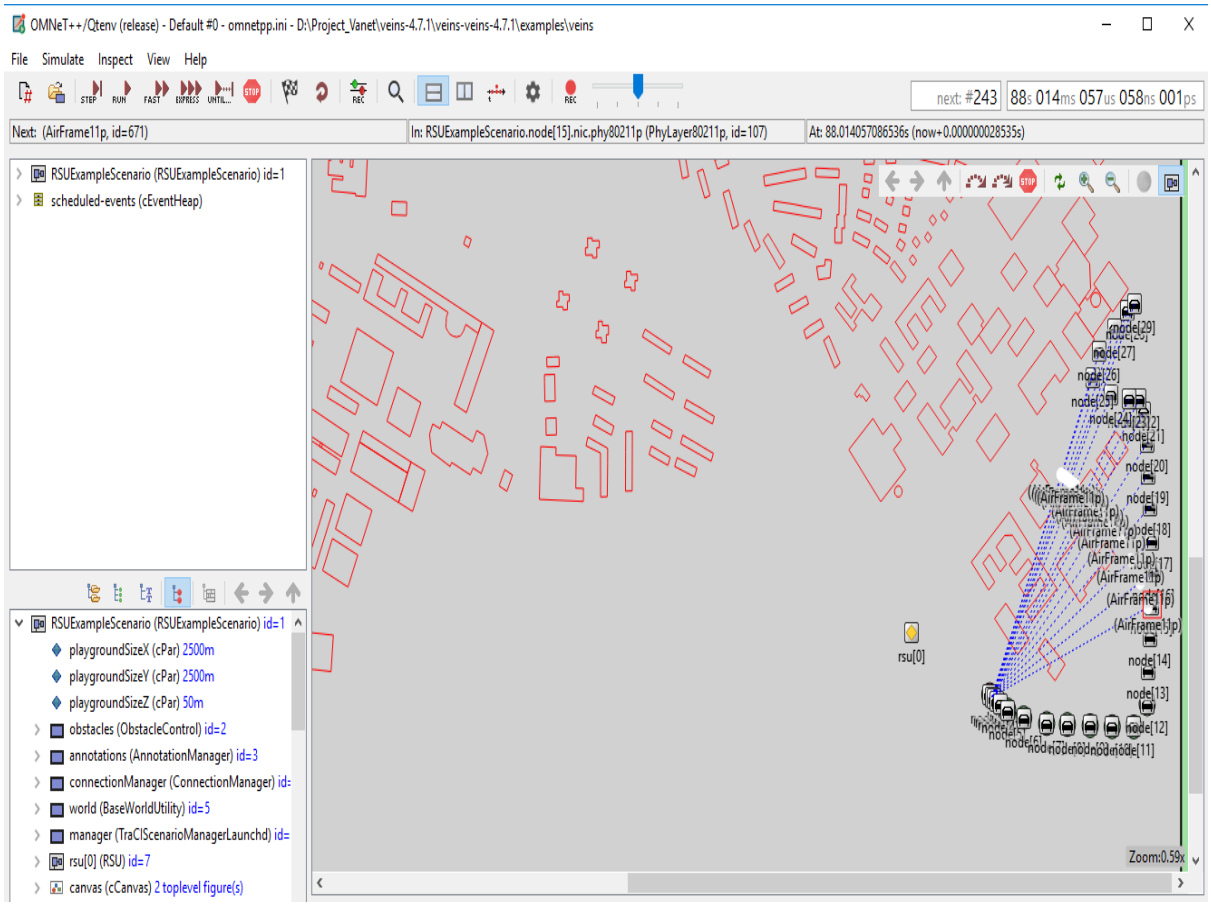
Gambar 4.4 Tampilan *Locate Vehicles* 120

Gambar 4.4 adalah tampilan vehicles/kendaraan yang telah ditentukan pada *Locate Vehicles. Vehicles* ini dapat dicari dengan cara manual ataupun dengan mengetikkan nomer *Vehicles* yang diinginkan, setalah mendapat *Vehicles* yang diinginkan lalu tekan *center* maka akan langsung menampilakn *Vehicles* yang diinginkan.

Selain dapat mencari lokasi kendaraan simulasi ini juga dapat mencari lokasi gedung, jalan, persimpangan jalan dan lainnya. Melacak lokasi gedung, jalan, persimpangan jalan sama dengan mencari lokasi kendaraan, kita hanya perlu memilih *locate polygon* untuk mencari lokasi bangunan, *locate junction* untuk mencari lokasi persimpangan jalan, *locate edge* untuk mencari jalan dan lainnya.

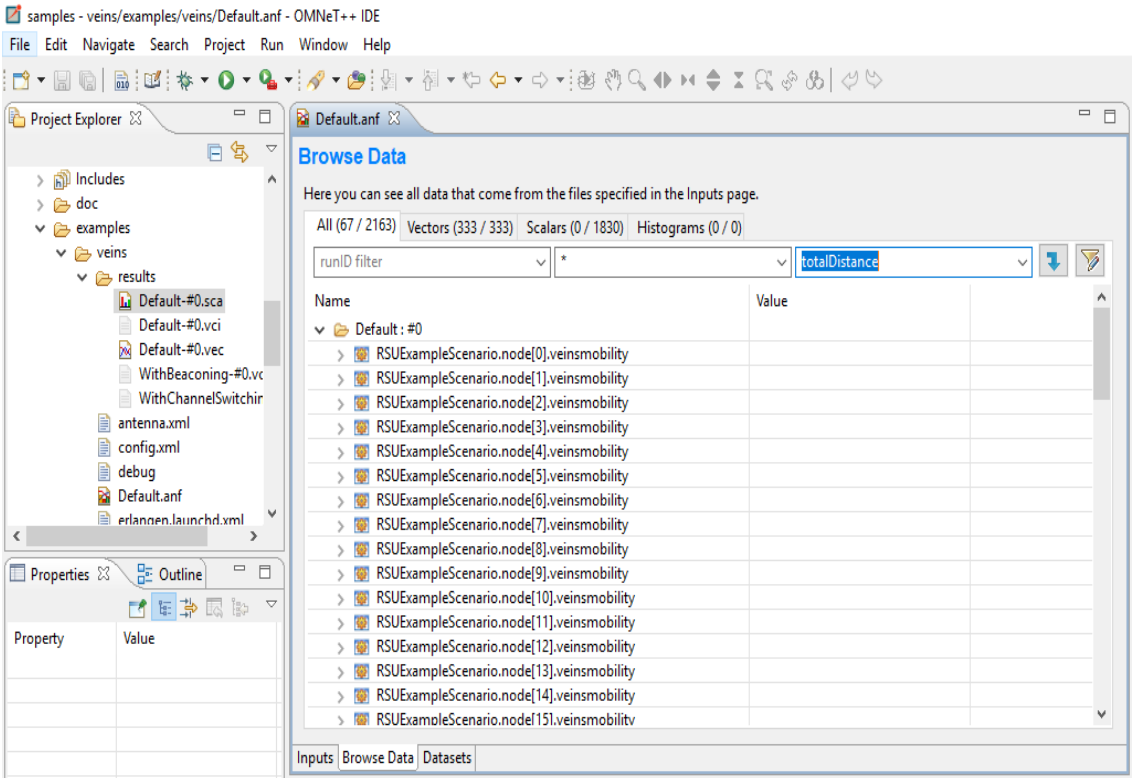
4.1.3 Simulasi VANET Pada Omnet++

Simulasi VANET pada Generator Omnet++ ini memiliki tampilan yang sangat sederhana dan cukup sulit untuk dimengerti tetap Generator ini dapat menampilkan hasil simulasi yang lebih detai dibandingkan dengan Generator SUMO.



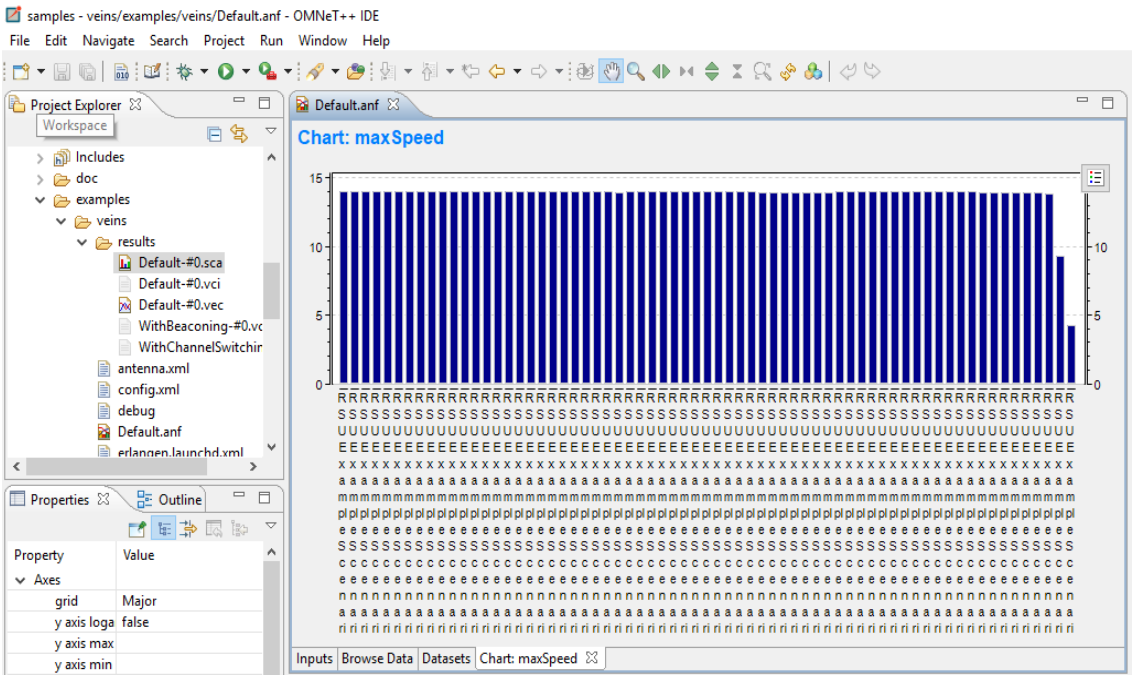
Gambar 4.5 Tampilan Simulasi pada Omnet++

Gambar 4.5 menunjukkan tampilan pada saat simulasi dijalankan, seperti yang kita lihat map pada simulasi ini tidak begitu jelas dan susah dimengeti dan terdapat beberapa node yang saling mengirimkan pesan kepada node lain yang akan bergerak dengan tujuannya masing-masing.



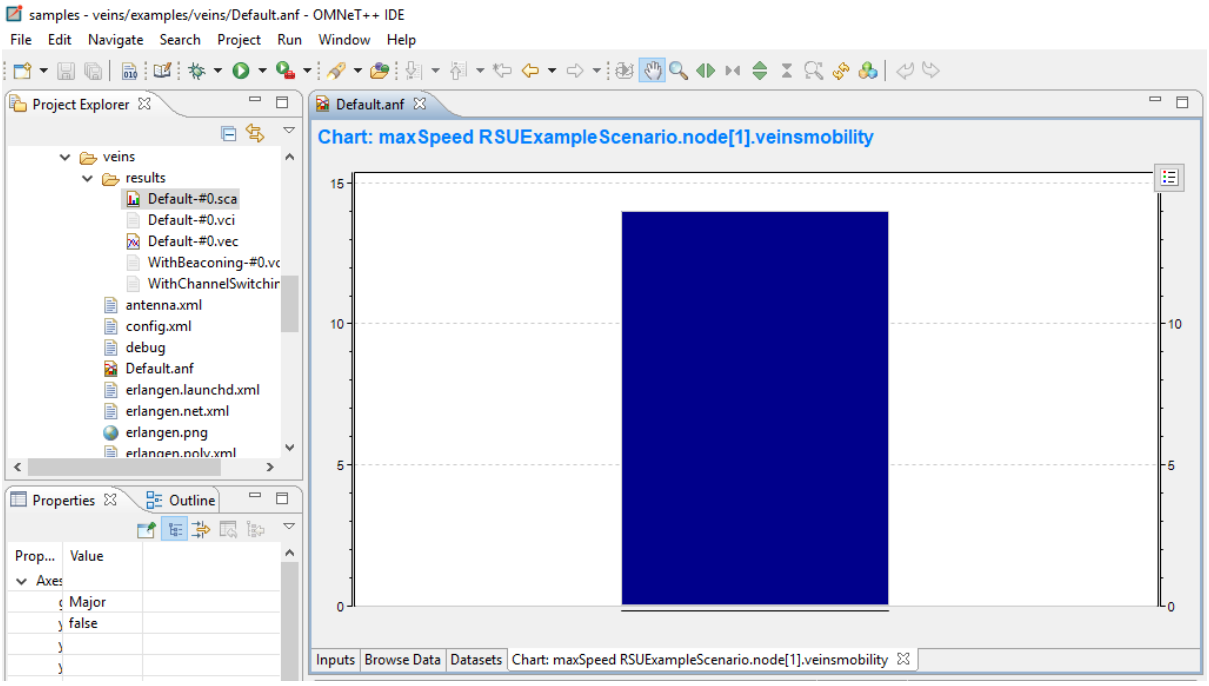
Gambar 4.6 Tampilan *Result*

Gambar 4.6 menunjukkan hasil dari sumilasi VANET yang telah dijalankan tadi. Disini kita dapat menampilakn hasil per-node atau semua node sekaligus, hasil yang dapat ditampilkan antara lain pos x, pos y, speed, acceleration, sentpackets, totalDistance dan lainnya.



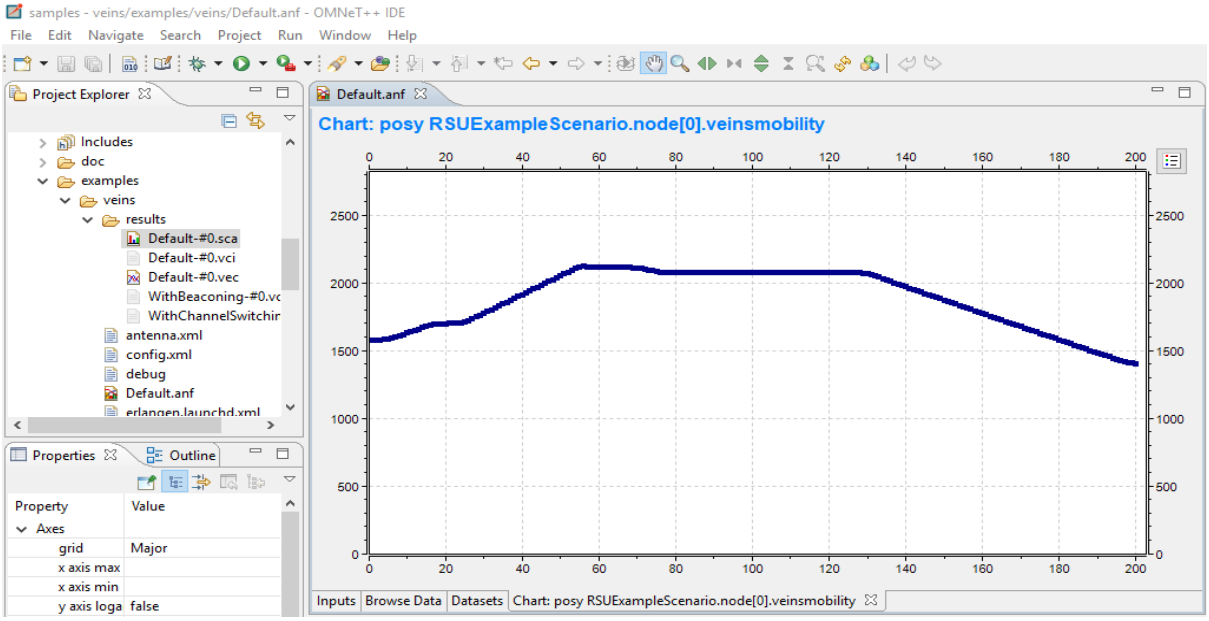
Gambar 4.7 *result* Max Speed seluruh node

Gambar 4.7 menunjukkan tampilan max speed pada seluruh node pada simulasi VANET, pada tampilan ini terlihat max speed dari seluruh node adalah 15 m/s dan min speed dari seluruh node adalah 5 m/s, kita juga bias menampilkan max speed dari setiap node yang kita inginkan seperti pada Gambar 4.8.



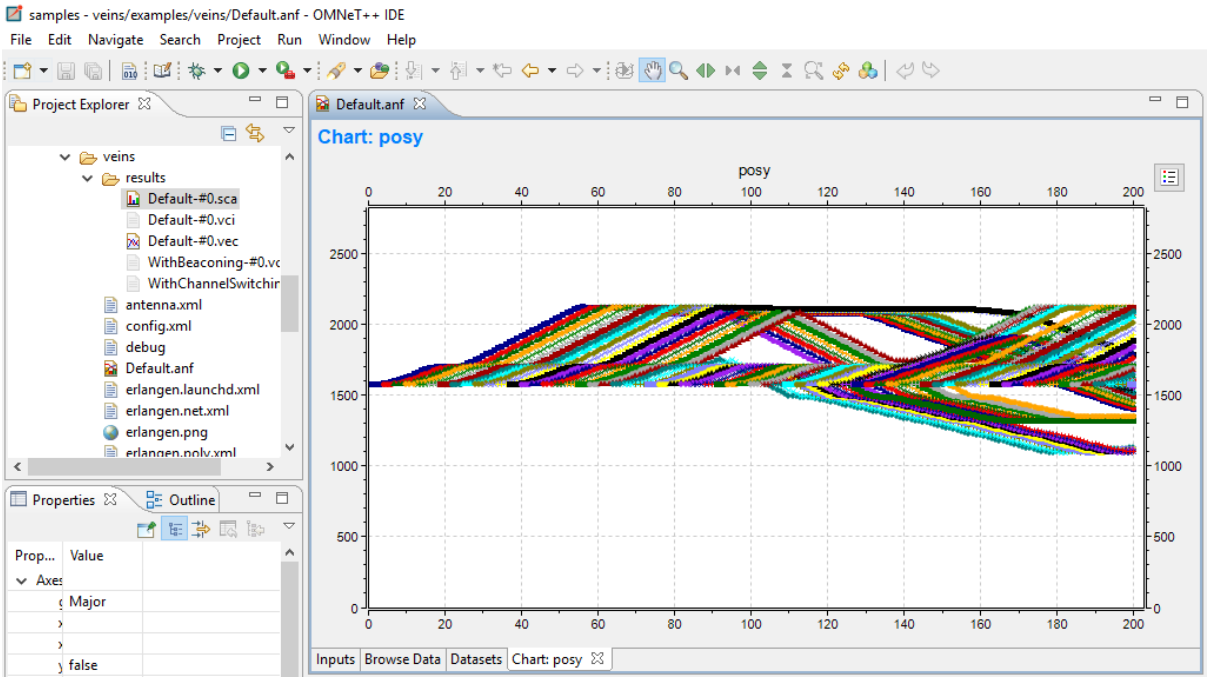
Gambar 4.8 Tampilan Max Speed Salah Satu Node

Pada Gambar 4.8 menampilan tampilan max speed dari salah satu node dari simulasi VANET yang dilakukan sebelumnya, tampilan ini lebih mudah dipahami dari tampilan max speed seluruh node karena kita bias langsung mengetahui node berapa yang ditampilkan.



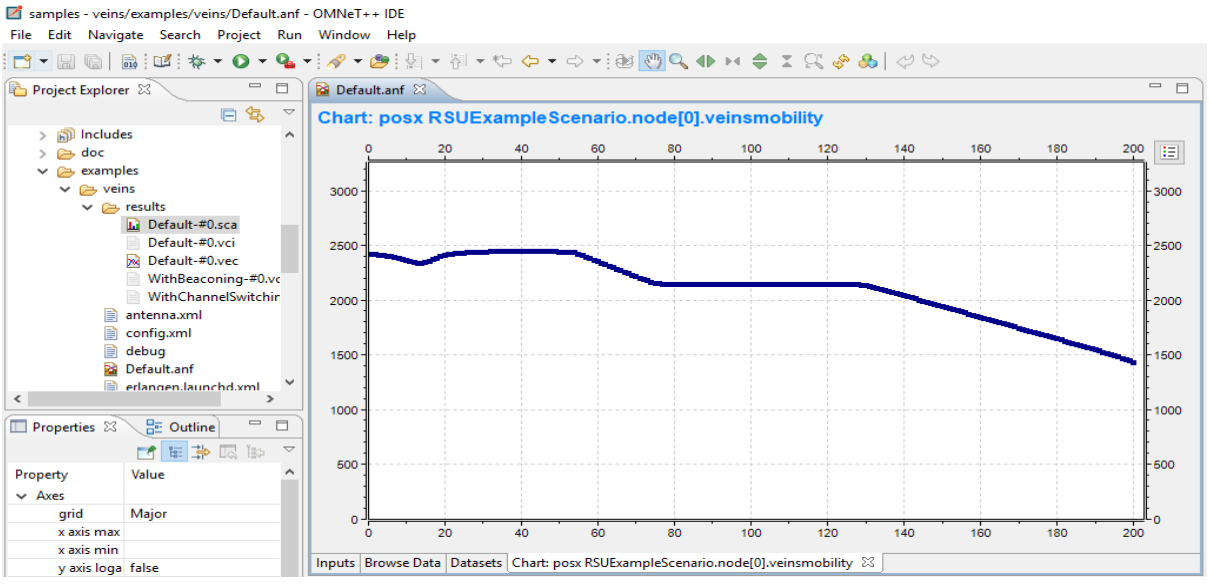
Gambar 4.9 Tampilan Grafik Pos Y

Gambar 4.9 menunjukkan tampilan pos y dari salah satu node pada simulasi VANET, pada tampilan ini terlihat node [0] bergerak dari titk y 1500 sampai melebihi 2000 dan kembali ke 1500, kita juga dapat menampilkan pos y dari seluruh node sekaligus seperti pada Gambar 4.10.



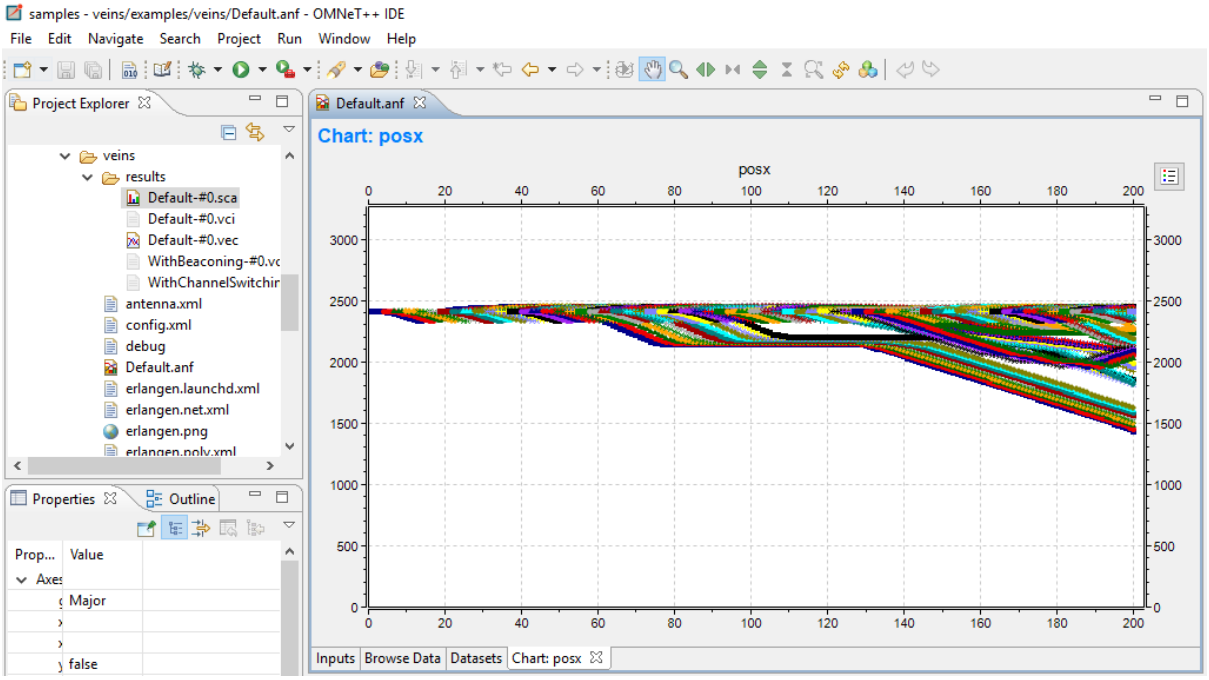
Gambar 4.10 Tampilan Grafik Pos Y Seluruh Node

Pada Gambar 4.10 menampilan tampilan Pos Y dari seluruh node dari simulasi VANET yang dilakukan sebelumnya, tampilan ini lebih sulit untuk dipahami dari tampilan Pos Y ssalah satu node karena kita tidak bias menentukan node berapa yang ingin kita lihat.



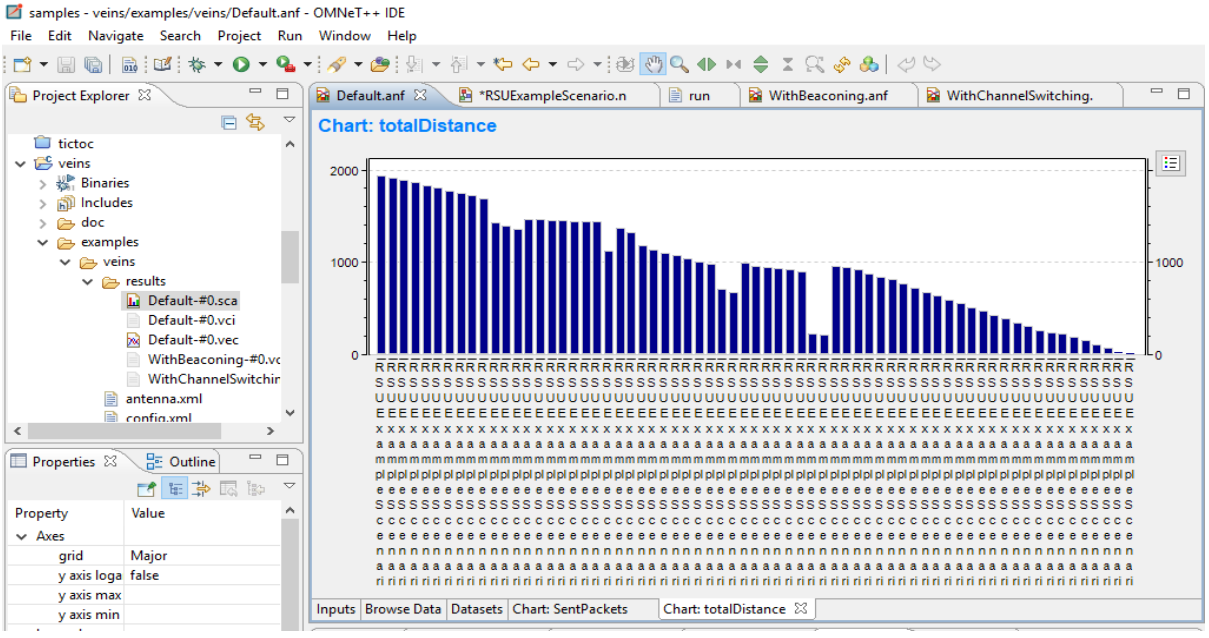
Gambar 4.11 Tampilan Grafik Pos X

Gambar 4.11 menunjukkan tampilan pos x dari salah satu node pada simulasi VANET, pada tampilan ini terlihat node [0] bergerak dari titik x 2500 sampai ke 1500, kita juga dapat menampilkan pos x dari seluruh node sekaligus seperti pada Gambar 4.12.



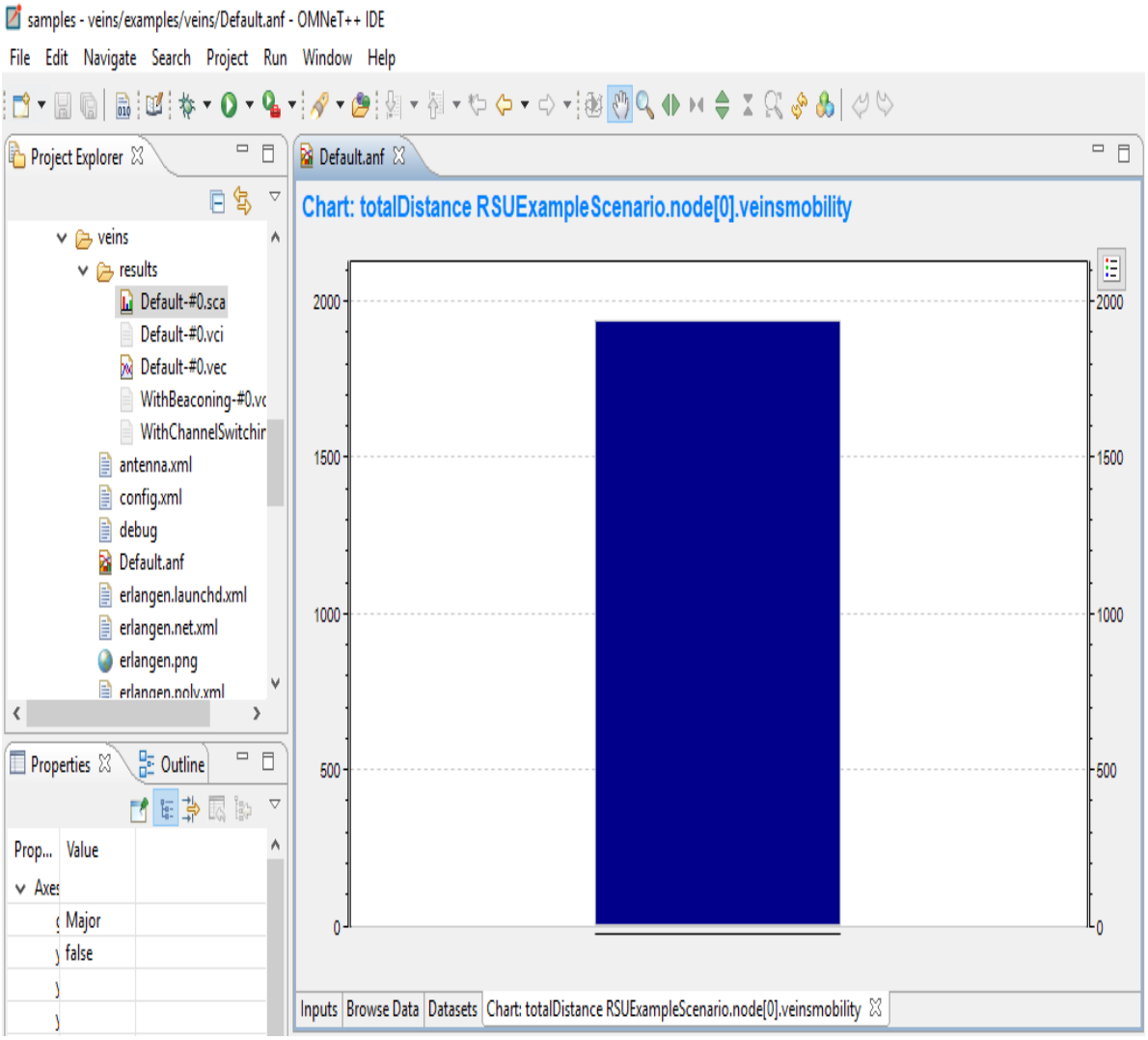
Gambar 4.12 Tampilan Grafik Pos X Seluruh Node

Pada Gambar 4.12 menampilan tampilan Pos X dari seluruh node dari simulasi VANET yang dilakukan sebelumnya, tampilan ini lebih sulit untuk dipahami dari tampilan Pos X ssalah satu node karena kita tidak bias menentukan node berapa yang ingin kita lihat.



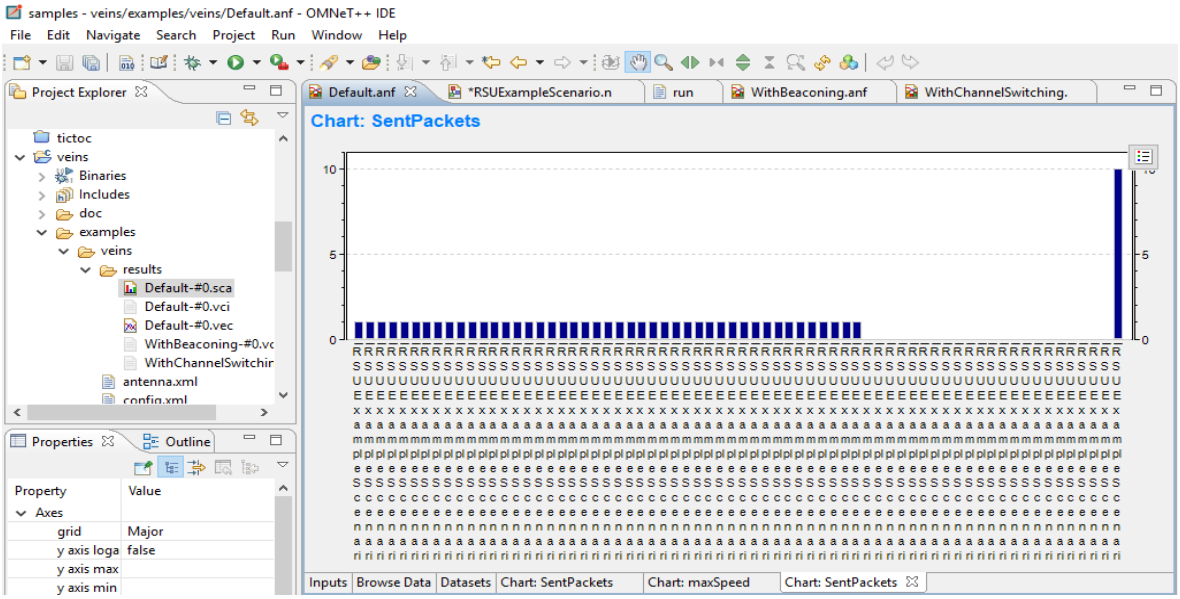
Gambar 4.13 Tampilan *Distance*

Gambar 4.8 menunjukkan tampilan *Distance* pada seluruh node di simulasi VANET, pada tampilan ini terdapat beragam total distance dari setiap node mulai dari yang tertinggi yaitu 2000m sampai yang terendah yaitu 0m. kita juga dapat menampilkan *result Distance* dari setiap node yang kita inginkan, kita juga bias menampilkan distance salah satu node yang diinginkan seperti pada Gambar 4.14.



Gambar 4.14 Tampilan *Distance* Salah Satu Node

Pada Gambar 4.14 menampilan *Distance* dari salah satu node dari simulasi VANET yang dilakukan sebelumnya, tampilan ini lebih mudah dipahami dari tampilan *Distance* seluruh node karena kita bias langsung mengetahui node berapa yang ditampilkan.



Gambar 4.15 Tampilan *result sentpackets*

Gambar 4.9 menunjukkan pengiriman pesan yang dilakukan pada seluruh node di simulasi VANET, terdapat node yang hanya sedikit mengirimkan pesan, node yang sama sekali tidak mengirimkan pesan dan node yang paling banyak mengirimkan pesan, kita juga dapat menampilakn *sentpackets* dari setiap node yang kita inginkan.

4.2 Pembahasan

4.2.1 Protokol Routing untuk Jaringan Ad-Hoc

Mekanisme routing pada jaringan vehicular bertanggung jawab menjaga supaya pertukaran informasi, mengenai topologi jaringan dan kondisi tiap jalur (link state), antar node yang sebagian besar mobile dapat berlangsung terusmenerus. Informasi tersebut diperlukan untuk menentukan rute terbaik dan memelihara jalur komunikasi yang terbentuk antar sepasang node pada jaringan. Jadi, protokol routing di VANET merupakan kunci untuk terbentuknya komunikasi antar kendaraan yang handal, kontinyu, dan aman.

Ada beberapa tantangan yang dihadapi oleh protokol routing di lingkungan VANET. Tantangan tersebut muncul karena beberapa karakteristik khusus yang dimiliki oleh jaringan vehicular ini. Tantangan pertama adalah kerusakan jalur komunikasi akibat node-node yang terus bergerak keluar masuk jaringan.

Adanya kesulitan untuk mempertahankan suatu rute karena perubahan topologi dan kepadatan kendaraan yang berubah-ubah. Adanya masalah hidden terminal, yaitu problem yang muncul jika terdapat terminal atau node yang di luar jangkauan beberapa node yang lain. Hidden terminal dapat menyebabkan pesat penerimaan paket data yang rendah.

Gedung-gedung tinggi di daerah perkotaan juga dapat menimbulkan interferensi, seperti routing loop dan pengiriman data ke arah yang salah, sehingga dapat memperbesar delay. Selain itu, protokol routing juga harus memperhatikan jenis aplikasi pada VANET. Aplikasi yang menyangkut keselamatan berlalu-lintas memerlukan protokol routing yang mampu menjamin waktu pengiriman yang tepat (*strict time delay*).

Bedasarkan penjelasan di atas, maka protokol routing yang dikembangkan untuk lingkungan VANET perlu memiliki kemampuan-kemampuan sebagai berikut:

1. mampu membentuk rute secara dinamis dan mempertahankan rute selama proses komunikasi.
2. mampu menemukan rute alternatif jika ada jalur yang putus.
3. mampu menentukan jalur optimal untuk mengurangi waktu untuk perutean.
4. mampu menyediakan banyak jalur dalam suatu jaringan untuk menghindari kemacetan data (congestion).
5. mampu bekerja dengan baik di lingkungan jalan raya (highway) dan perkotaan (urban) dengan kondisi trafik yang padat, maupun sepi.

4.2.2 AODV (Ad-hoc On-demand Distance Vector)

AODV adalah protokol routing reaktif yang dapat digunakan untuk mendukung komunikasi unicast atau multicast. Protokol ini tidak menyimpan infromasi rute yang lengkap, melainkan hanya informasi node yang harus dituju pada lompatan berikutnya. Sama dengan DSR, protokol ini juga terdiri dari dua mekanisme utama, yaitu: mekanisme pencarian rute dan pemeliharaan rute.

Pesan RREQ (Route Request) dan RREP (Route Reply) digunakan untuk pencarian rute. Node yang akan mengirimkan data akan menyebarkan paket RREP ke node-node di sekitarnya untuk menemukan node tujuan. Jadi, paket RREQ digunakan untuk mengawali pembentukan rute menuju tujuan.

Terdapat beberapa informasi yang ada paket paket RREQ, antara lain: alamat node sumber/pengirim dan alamat node penerima, pencacah hop, *souce sequence* number, *destination sequence* number, dan broadcast ID.

Pencacah hop atau lompatan berfungsi sebagi pencatat jumlah lompatan setiap dilakukan pengiriman paket RREQ. Broadcast ID adalah identitas dari paket RREQ. Nilai ID ini akan bertambah setiap kali node sumber mengirimkan RREQ. Langkah-langkah pencarian rute pada AODV yang lengkap adalah sebagai berikut:

1. Jika suatu node sumber (S) membutuhkan suatu rute menuju node tujuan (D), tahap awal yang dilakukan oleh node sumber adalah menyebarkan paket route request (RREQ) menuju node tetangganya.
2. Apabila node yang menerima RREQ memiliki informasi rute menuju D, maka node tersebut akan mengirim paket RREP kembali menuju node sumber melalui *reverse path* yang diciptakan RREQ setiap kali *flooding* dilakukan. Namun, jika node tersebut tidak memiliki informasi rute menuju D, maka node tersebut akan menyiarkan ulang pesan RREQ ke node di sekitarnya.
3. Node yang menerima RREQ dengan nilai alamat sumber dan nilai broadcast ID yang sama dengan RREQ yang diterima sebelumnya, akan mempertahankan RREQ yang sudah diterima di awal dan membuang RREQ baru.
4. Ketika sebuah node yang memiliki informasi rute menuju D menerima RREQ, maka node tersebut akan melakukan perbandingan antara nilai *destination sequence number* yang telah dimiliki dengan *sequence number* yang ada pada RREQ. Jika *sequence number* pada node lebih besar atau sama dengan yang ada di RREQ, maka node akan mengirim paket RREP menuju S. Jika *sequence number* pada RREQ lebih kecil, maka pesan RREQ akan diteruskan ke node lain.
5. Jika terdeteksi adanya kerusakan rute, mekanisme *route maintenance* akan mengirimkan paket *route error* (RERR) menuju S dan node sumber akan kembali menyebarkan paket RREQ.
6. Informasi masa aktif rute akan diberikan oleh node antara yang menerima RREQ. Informasi untuk perutean dari sumber (S) ke D akan dihapus jika waktu aktifnya telah habis.

4.3 Pengujian Penelitian

Tahap pengujian penelitian dilakukan untuk mengetahui apakah system yang dibuat dapat berjalan dengan baik dan lancar. Dengan dilakukan pengujian ini maka akan dapat ditemukan kekurangan serta kelebihan dari pada system jaringan VANET ini. Pengujian dilakukan dengan berbagai tahap diantaranya:

1. Melakukan uji Protokol AODV dengan Simulais VANET
2. Melakukan uiji pada peta RealMap
3. Melakukan uji pengiriman pesan sederhana pada Omnet++
4. Melakukan uji Simulasi VANET pada generator SUMO dan Omnet++

4.4 Analisis Penelitian

Dengan adanya sistem jaringan VANET daerah kemacetan dan rawan kecelakaan di daerah Yogyakarta khususnya perempatan Jl. Kaliurang dengan Ring Road Utara dan perempatan Jl. Gejayan dengan Ring Road Utara dapat dicegah dan pergerakan kendaraan pada daerah tersebut dapat distabilkan dengan VANET. Adapun komponen penyusun VANET terdiri dari 3 domain berbeda, yaitu: in-vehicle, adhoc, dan infrastruktur. Domain in-vehicle meliputi komponen yang terletak di dalam kendaraan, yaitu On-Board Unit (OBU) dan Application Unit (AU). Jumlah AU dalam suatu kendaraan bisa lebih dari satu. AU dapat berupa bagian dari kendaraan yang secara permanen terhubung ke OBU atau berupa laptop, PDA, atau game pad yang terhubung dengan OBU secara dinamis. Koneksi antara AU dan OBU dapat melalui kabel, bluetooth, WUSB, maupun UWB. Domain kedua adalah domain ad-hoc. Domain ini terdiri dari kendaraan-kendaraan yang telah dilengkapi dengan OBU dan unit-unit komunikasi pendukung di sepanjang jalan (RSU).

OBU dari beberapa kendaraan akan membentuk jaringan nirkabel ad-hoc (MANET). OBU-OBU yang terhubung secara nirkabel akan langsung dapat saling berkomunikasi tanpa adanya koordinasi secara terpusat. Jika tidak ada koneksi langsung, protokol routing untuk VANET memungkinkan komunikasi secara multi-hop, yaitu data diteruskan dari satu OBU ke OBU lainnya hingga ke tujuan. OBU dan RSU dapat dianggap sebagai node-node dari VANET. RSU dapat terhubung dengan jaringan infrastruktur komunikasi sehingga RSU dapat terhubung dengan Internet. Hal ini memungkinkan berbagai AU yang terhubung pada suatu OBU untuk berkomunikasi dengan host-host di Internet. OBU dapat juga terhubung ke Internet melalui Hot Spot (HS). Apabila tidak tersedia RSU dan HS, OBU dapat memanfaatkan teknologi komunikasi lain untuk aplikasi yang tidak berhubungan dengan keselamatan lalu lintas, misalnya menggunakan GSM, GPRS, UMTS, 4G, dan lain sebagainya. Domain infrastruktur pada VANET mengacu pada berbagai piranti pendukung terlaksananya komunikasi lainnya yang ada di sepanjang jalan

Terdapat kelebihan dan kekurangan pada sistem ini, setiap kelebihan dan kekurangan dapat menjadi bahan pertimbangan bagi peneliti dan pembaca untuk menilai keberhasilan dari penelitian ini. Kekurangan dari penelitin ini juga diharapkan akan menjadi pertimbangan untuk pengembangan lebih lanjut demi tercapainya sistem yang sempurna di masa mendatang. Oleh karenanya maka di bawah ini merupakan kelebihan serta kekurangan pada penelitian ini antara lain:

a) Kelebihan Sistem

1. Generator SUMO dapat menampilkan peta nyata dalam simaulasi VANET.
2. Generator Omnet++ dapat menampilkan hasil simulasi VANET yang detai.
3. VANET sebagai jaringan ad-hoc nirkabel dengan node yang memiliki mobilitas tinggi.
4. Tampilan map pada simulasi lebih menarik dan mudah dipahami karena menggunakan peta *RealMap*.
5. Dapat melakukan berbagai macam pencarian seperti kendaraan, jalan, persimpangan jalan dan lainnya.
6. Dapat menampilkan hasil detai setelah melakukan Simulasi.

b) Kekurangan Sistem

1. Perlu dilakukan modifikasi *pre-processing* terhadap mobilitas saat melakukan simulasi di dalam SUMO agar adaptif terhadap perubahan mobilitas tinggi dan densitas kendaraan dalam lingkungan VANET.
2. Perlu dilakukan perubahan dan penambahan skema *traffic light* di dalam generator untuk skenario peta riil Yogyakarta yang dimuat pada *OpenStreetMap* karena masih memiliki kesalahan dalam implementasi skenario nyata Yogyakarta.
3. AODV adalah protokol yang hanya akan membentuk koneksi jika ada node yang akan mengirimkan data.
4. Membutuhkan biaya yang sangat besar untuk diaplikasikan dikehidupan nyata.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis data dan pembahasan dari penelitian serta mengacu pada rumusan masalah seperti yang telah diuraikan pada bab sebelumnya, maka dapat ditarik kesimpulan penelitian sebagai berikut :

1. Kedua Simulator VANET SUMO dan Omnet++ dapat mengoptimalkan pergerakan kendaraan di Jl. Kaliurang dan Jl. Gejayan sehingga dapat mengurangi kecelakaan dan kemacetan di daerah tersebut.
2. Untuk mendapatkan hasil simulasi VANET yang detai dan optimal kita dapat menggunakan Omnet++ tetapi tampilan simulasi pada simulator ini sangat sederhana dan cukup sulit untuk dimengerti. Berbeda dengan simulator SUMO yang memberikan tampilan simulasi yang lebih baik dan mudah dipahami tetapi memberikan hasil yang tidak detail dan sederhana.
3. Pada simulator SUMO kita dapat melacak lokasi kendaraan pada saat simulasi berlangsung, simulator ini dapat melacak kendaraan yang bergerak maupun berhenti.
4. Pada simulator Omnet++ kita dapat menampilkan hasil simulasi VANET dari setiap kendaraan.

5.2 Saran

Bagi para peneliti selanjutnya yang berniat untuk melakukan penelitian terkait dengan judul penelitian ini diharapkan agar dapat melakukan penelitian tentang factor-faktor lain yang juga berpengaruh terhadap simulasi VANET, ada pun yang perlu dilakukan dalam pengembangan selanjutnya yaitu :

1. Perlu dilakukan modifikasi pre-processing terhadap mobilitas saat melakukan simulasi di dalam SUMO dan VanetMobisim agar adaptif terhadap perubahan mobilitas tinggi dan densitas kendaraan dalam lingkungan VANET.
2. Untuk pengembangan selanjutnya dapat dibuat sebuah testbed sehingga hasil analisa yang didapat lebih akurat, selain itu performansi juga dapat diukur dengan parameter QoS yang lain.

DAFTAR PUSTAKA

Aziza, R. N., Siswipraptini, P. C., & Cahyaningtyas, R. (2017). Protokol Routing pada VANET : Taksonomi dan Analisis Perbandingan antara DSR, AODV, dan TORA. *Fifo*, *IX*(2), 98–109.

Hadi, M. Z. S., Amran, H., & R, N. P. (2010). Analisis Performansi Routing AODV pada Jaringan VANet, 64–74.

Nutrihadi, F., Anggoro, R., & Ijtihadie, M. (2016). Studi Kinerja VANET Scenario Generators : SUMO dan VanetMobisim untuk Implementasi Routing Protocol AODV menggunakan Network Simulator 2 (NS-2), *5*(1), 1–6.

Tonoro, A. A., Wardana, H. K., & Nugroho, S. (2015). Simulasi dan Pengkajian Performa Vehicular Ad Hoc Network. *Techné Jurnal Ilmiah Elektroteknika*, *14*, 27–34. https://doi.org/10.4324/9780203017319

Wahyudi, J., Berlian, & Rosdiana. (2013). INSTRUKSI BAHASA PEMROGRAMAN ADT (ABSTRACT DATA TYPE) PADA VIRUS DAN LOOP BATCH, *9*(1).

Wahyuningrum, T. (2012). Implementasi XML Encryption (XML Enc) Menggunakan Java. *JURNAL INFOTEL - Informatika Telekomunikasi Elektronika*, *4*(1), 17. https://doi.org/10.20895/infotel.v4i1.98