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

Gita Agistira Romanta
Departments Of Informatics
Universitas Islam Indonesia
Yogyakarta, Indonesia
14523235@students.uii.ac.id

Abstrak— Vehicular Ad Hoc Network (VANET) merupakan turunan dari MANET (Mobile Ad Hoc Network) sebagai inovasi baru dalam dunia teknologi yang membantu kebutuhan manusia dalam berkomunikasi. VANET dapat mendukung komunikasi langsung antara kendaraan (Vehicle to Vehicle) dan antara kendaraan-infrastruktur (Vehicle to Infrastructure) dengan adanya infrastruktur jaringan nirkabel. Namun, implementasi VANET di dunia masih sulit dilakukan sehingga banyak penelitian dilakukan dengan membuat simulasi menggunakan mobility generator dan network simulator. Pada makalah ini yang diteliti yaitu performa skema VANET yang dihasilkan oleh mobility generator SUMO dan Omnet++. Skenario VANET dengan peta berbentuk grid dan peta riil Perempatan Jl. Kaliurang dengan Ring Road Utara dan Perempatan Jl. Gejayan dengan Ring Road Utara, digunakan pada kedua generator SUMO dan Omnet++ dengan memvariasikan jumlah kendaraan simulasi. 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.

Kata Kunci— VANET, MANET, SUMO, Omnet++

I. PENDAHULUAN

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.

II. REVIEW SISTEM JARINGAN VANET

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).

- *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).

- *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.

- *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 pengembangan dan kemanusiaan (Nutrihadi et al., 2016).

III. METODOLOGI

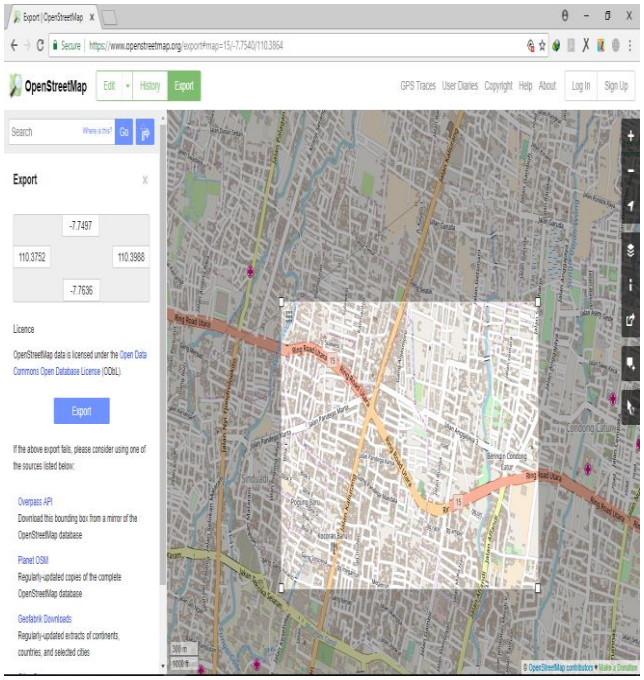
- Perancangan Sistem

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.

a) Pembuatan Peta Pada SUMO

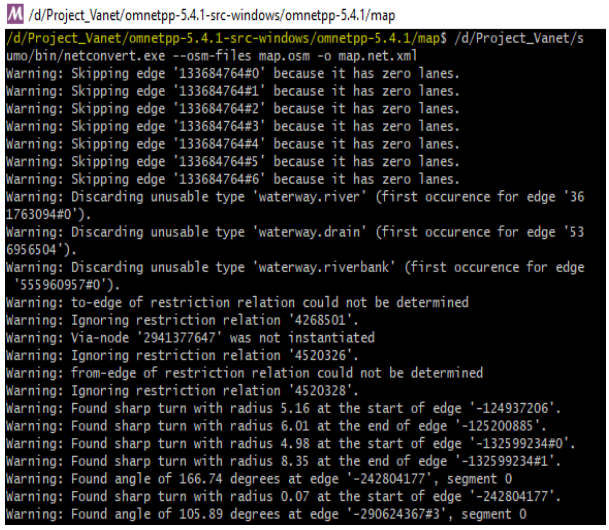
- Tahap awal pembuatan peta riil adalah dengan mencari daerah *real world* yang akan kita gunakan dalam simulasi VANET dengan mobilitas yang tinggi sehingga cocok untuk jaringan VANET. Pada penelitian ini, diperoleh daerah Perempatan Jl. Kaliurang dengan Ring Road Utara dan Perempatan Jl. Affandi dengan 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:
- 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

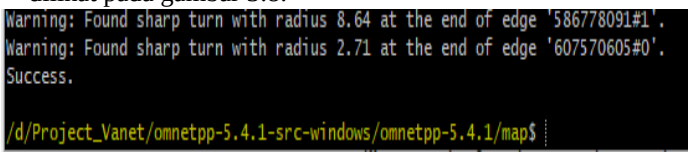
Setelah bagian yang ingin kita gunakan sudah ditetapkan makanya selajutnya kita tekan tombol export dan kita akan mendapatnya file .osm yang akan kita gunakan untuk melakukan simulasi VANET.

- 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.
- 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.
- 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

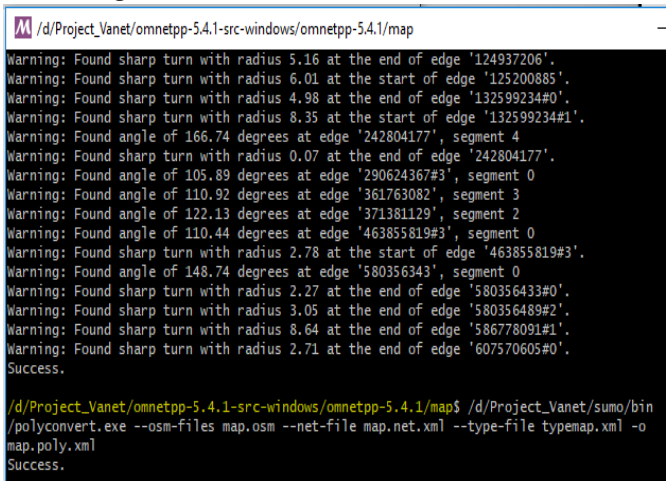
Pada gambar 3.7 terdapat *source code* "/d/Projek_Vanet/sumo/bin/netconvert.exe –osm-files map.osm –o map.net.xml" yang berfungsing unutk mengkonfersi file map.osm menjadi file map.net.xml. untuk melihat apakah proses konfersi berhasil atau tidak dapat dilihat pada gambar 3.8.



Gambar 3.6 Proses konfersi file berhasil

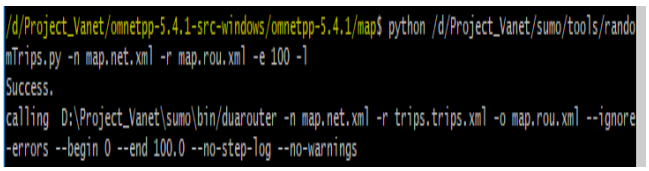
Jika terdapat kata "*Success*" diakhir proses maka proses dari konfersi file telah berhasil dan jika diakhir proses terdapat kata "*error*" maka konfersi file telah gagal.

- 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

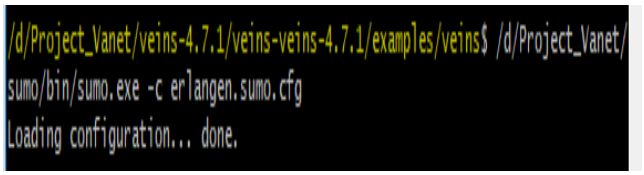
- Langkah selanjutnya adalah mengkonfersi file *map.net.xml* untuk menentukan sumber dan tujuan node secara acak yang sesuai dengan file .osm agar pergerakan simulasi terlihat nyata pada RealMap yang telah digunakan. Langkah ini mengunakan file *randaomtrip.py* yang telah disediakan oleh SUMO dan akan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



Gambar 3.8 *source code* dan proses rendomtrips

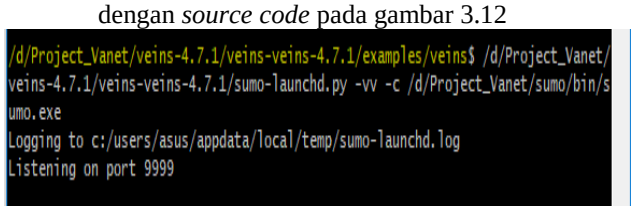
- Jika semua proses di atas berhasil maka akan menghasilkan beberapa file yang akan digunakan untuk melakukan simlasi VANET, file tersebut adalah *map.osm, map.net.xml, map.poly.xml, map.rou.alt.xml, map.rou.xml, trips.trips.xml, typemap.xml dan map.sumo.cfg.*

- **Perancangan Simulasi VANET pada Omnet++ yang di Import dari VEINS**
  - Tahap awal dalam perancangan simulasi pada *Omnet++* adalah dengan mendownload veins dengan versi apapun yang akan digunakan dalam pemanggilan simulasi VANET pada Omnet++.
  - Tahap berikutnya, yaitu mengkonfigurasikan aplikasi sumo dengan project erlangen.sumo.cfg dengan *source code* pada gambar 3.11



Gambar 3.9 *Source code*

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



Gambar 3.10 *Source code*

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

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.

1) Hasil

- 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*.

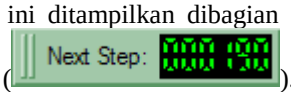
4. **sumo.exe**

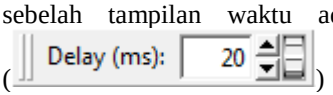
   SUMO adalah simulasi itu sendiri; Ini adalah simulasi arus lalu lintas mikroskopik, ruang-kontinu, dan waktu-diskrit. Anda dapat menggunakan file definisi skema *XML* untuk mengatur konfigurasi SUMO.

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**

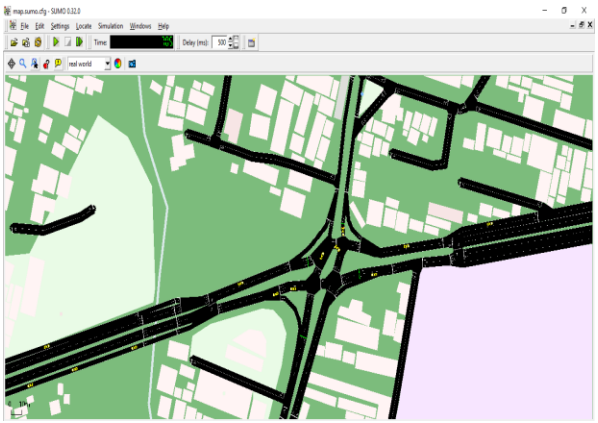
POLYCONVERT mengimpor bentuk geometris (poligon atau tempat menarik) dari sumber yang berbeda, mengubahnya menjadi representasi yang dapat divisualisasikan menggunakan SUMO-GUI. *POLYCONVERT* dapat mengimpor bentuk dari berbagai jenis file. Biasanya, untuk mengimpor data jenis tertentu, nama jenis digunakan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

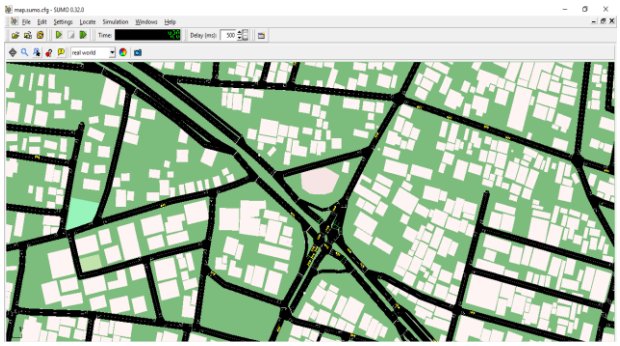
Tabel diatas menunjukkan hasil parameter dari simulasi VANET dengan menggunakan Generator SUMO, terdapat beberapa parameter yang penting untuk ditampilkan saat simulasi telah selesai dilakukan.



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

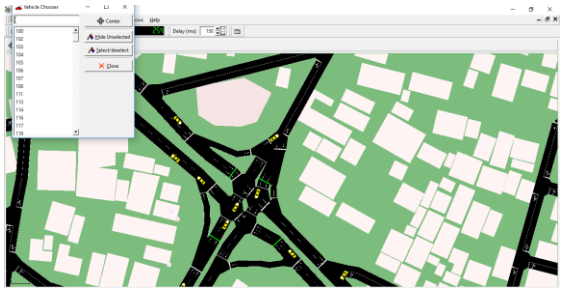
Pada gambar di atas kita dapata melihat simulasi VANET yang hampir sama dengan gambar sebelumnya, yang membedakan adalah letak *RealMap* yang digunakan. *RealMap* yang digunakan pada gambar di atas adalah perempatan Jl. Affandi dengan Ring Road Utara. Simulasi ini menggunakan peta *RealMap* agar lebih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

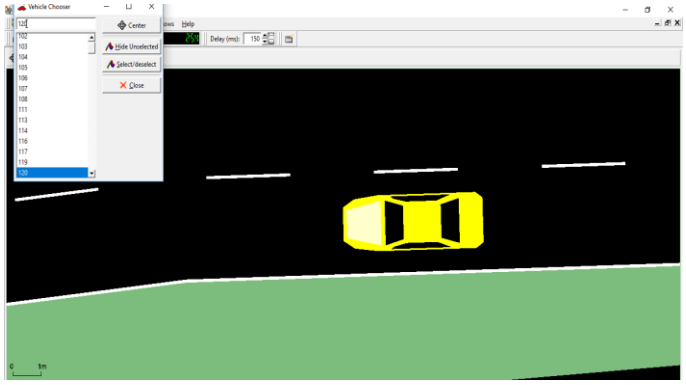
Hasil dari pembuatan simulasi VANET pada generator SUMO tidak hanya dapat membuat kendaraan saling bergerak dan berkomunikasi satu sama lain agar dapat mencegah terjadinya tabrakan data/kendaraan dan kemacetan, tetapi juga dapat melacak kendaraan.

Simulasi ini dapat melacak kendaraan yang sedang bergerak ataupun sedang berhenti. Jadi simulasi ini juga dapat digunakan untuk melakukan pengejaran suatu 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.



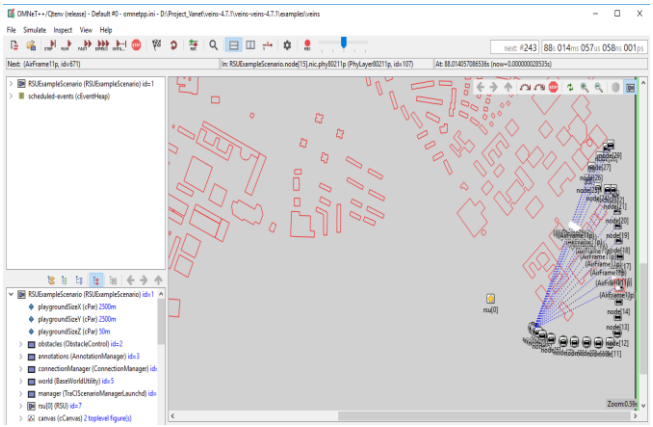
Gambar 4.4 Tampilan *Locate Vehicles* 120

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.

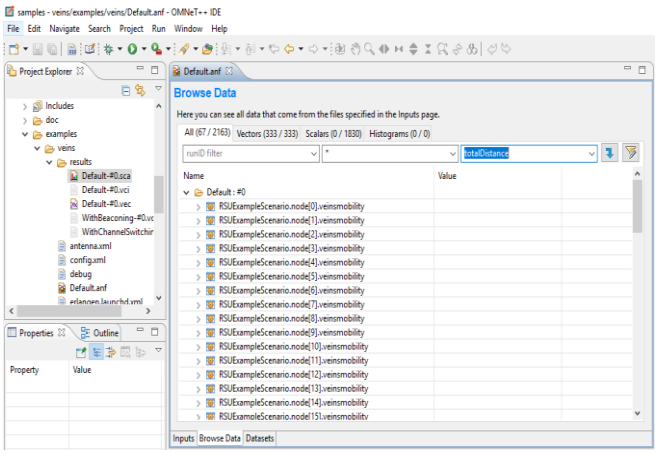
- Simulasi VANET Pada Omnet++

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.



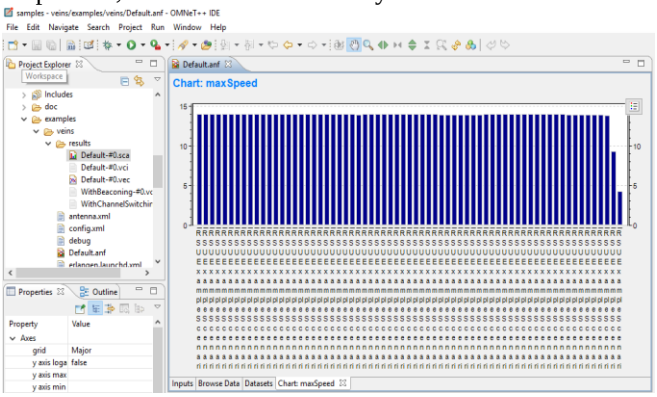
Gambar 4.5 Tampilan Simulasi pada Omnet++

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.



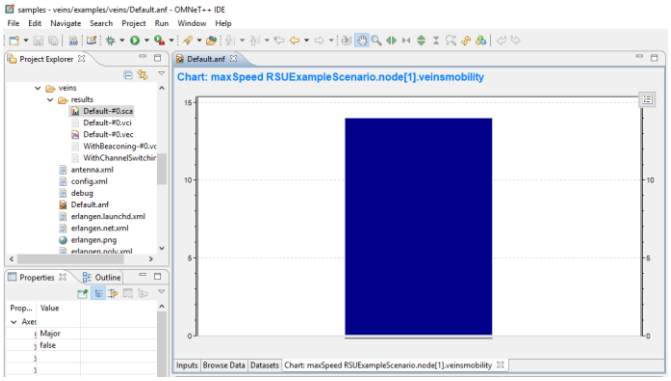
Gambar 4.6 Tampilan *Result*

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

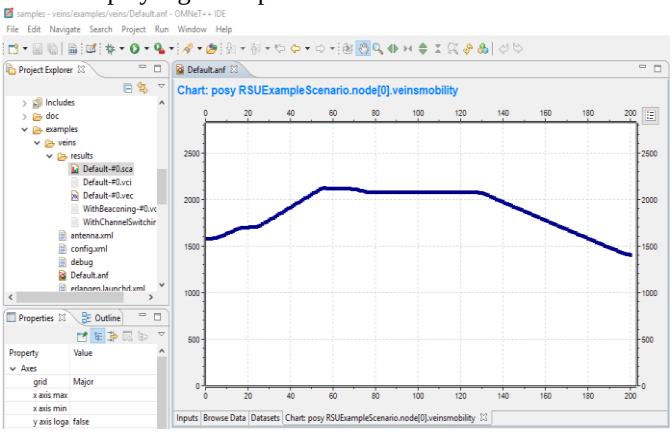
Gambar 4.7 menunjukkan tampilan max speed pada seluruh node pada simulasi VANET, pada tampilan ini terlihat max speed dari seluruh node adalah 15 m/s dan min speed dari seluruh node adalah 5 m/s, kita juga bias menampilkan max speed dari setiap node yang kita inginkan seperti pada Gambar 4.8.



Gambar 4.8 Tampilan Max Speed Salah Satu Node

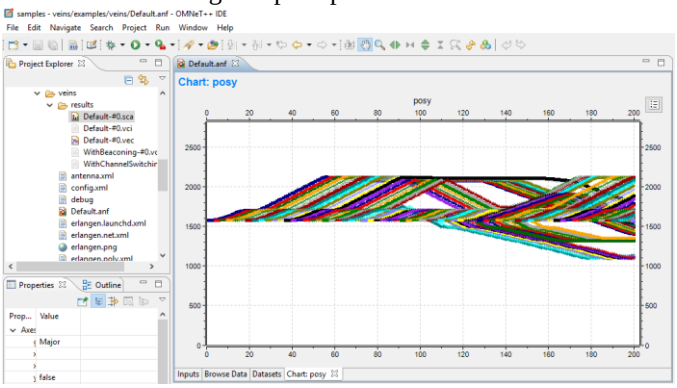
Pada Gambar 4.8 menampilan tampilan max speed dari salah satu node dari simulasi VANET yang dilakukan sebelumnya, tampilan ini lebih mudah dipahami dari tampilan

max speed seluruh node karena kita bias langsung mengetahui node berapa yang ditampilkan.



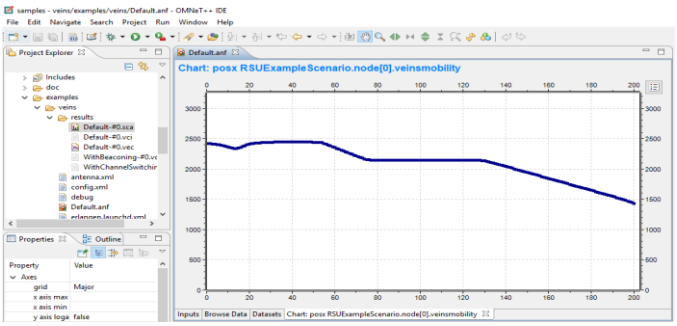
Gambar 4.9 Tampilan Grafik Pos Y

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.



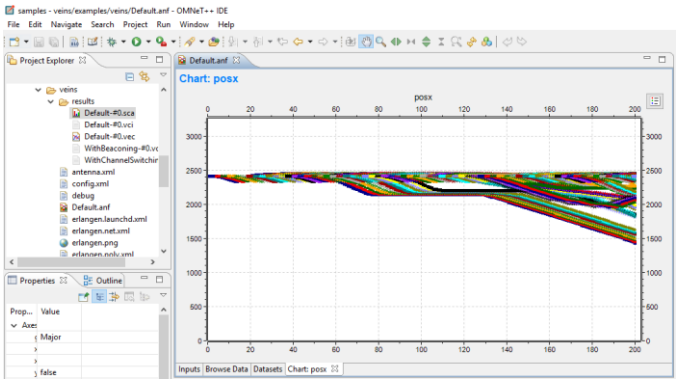
Gambar 4.10 Tampilan Grafik Pos Y Seluruh Node

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.



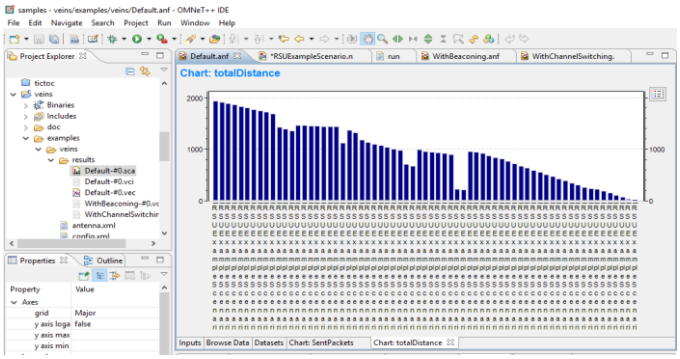
Gambar 4.11 Tampilan Grafik Pos X

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.



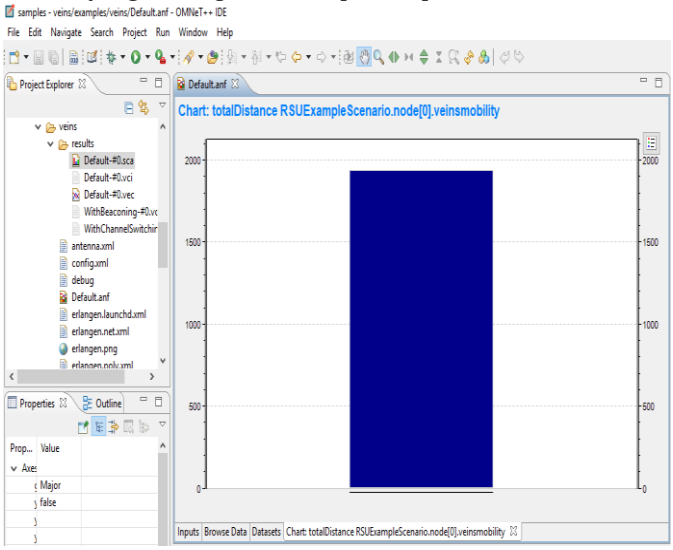
Gambar 4.12 Tampilan Grafik Pos X Seluruh Node

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.



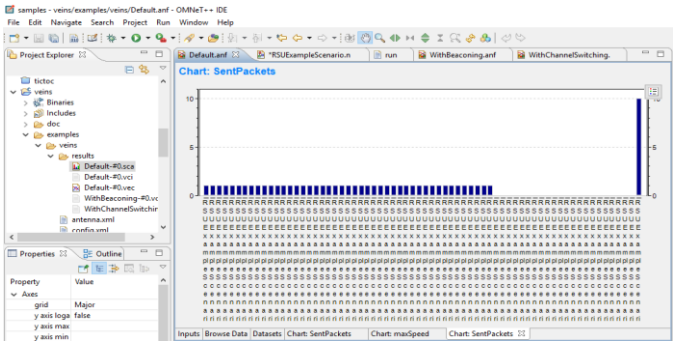
Gambar 4.13 Tampilan *Distance*

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.

2) Pembahasan

- *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.

- *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.

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) 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.

V. 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.

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.

REFERENCES

[1] 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.

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

[3] 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.

[4] 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

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

[6] 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