

DAFTAR PUSTAKA

- Abubakar, Iskandar., dkk. 1998. Pedoman Perencanaan dan Pengoperasian Fasilitas Parkir, Jakarta : Direktorat Bina Sistem lalu Lintas dan Angkutan Kota Direktorat Jenderal Perhubungan Darat.
- Aryandi, Rama Dwi & Ahmad Munawar. 2014. Penggunaan Software Vissim Untuk Analisis Simpang Bersinyal (Studi Kasus Simpang Mirota Kampus Terban Yogyakarta): Universitas Gadjah Mada.
- Direktorat Jendral Perhubungan Darat, 1996. Pedoman Teknis Penyelenggara Fasilitas Parkir. Jakarta.
- Direktorat Jendral Perhubungan Darat, 1998. Pedoman Perencanaan dan Pengoperasian Fasilitas Parkir. Jakarta.
- Manjunatha, Pruthvy. 2013 *Methodology for the Calibration of VISSIM in Mixed Traffic*: IIT Bombay.
- Menneni, S.& Sun C. 2008. *Microsimulation Calibration Using Speed-Flow Relationships*: Research Article.
- Messah, Y.A, dkk. 2012. Analisis Kebutuhan Lahan Parkir di Rumah Sakit Umum Daerah Prof.Dr.W.Z Johannes Kupang. Jurnal Teknik Sipil. Vol 1 No 4 : Fakultas Teknik Sipil Undana.

Palayukan, R.O. 2015. Analisis Karakteristik Parkir Kendaraan pada Area Parkir di Bandara Sultan Hasanuddin di Kota Makassar. Teknik Sipil Fakultas Teknik Universitas Hasanuddin. Makassar.

PTV AG, 2015, PTV Vissim 8 User Manual, Karlsruhe, Germany.

Putra, M.F. 2021. Analisis Karakteristik Parkir Kendaraan di Bandara Hasanuddin. Skripsi. Teknik Sipil Fakultas Teknik Universitas Hasanuddin, Makassar

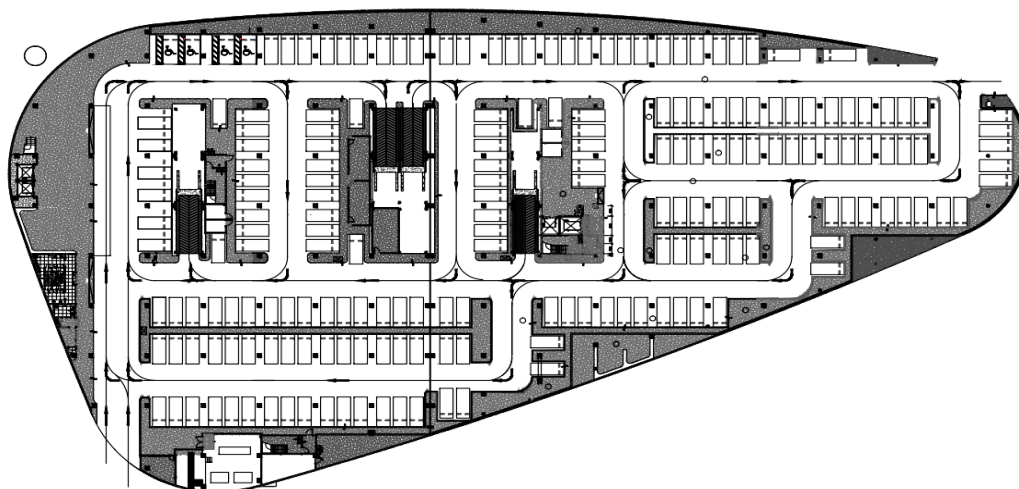
Susetyo, A.D., & Buchari, E. 2016. Permodelan Simulasi Parkir dan Analisis Kebutuhan Parkir Sekolah Islam Terpadu Bina Ilmi, Tanjung Rawa, Palembang. *Proceedings Of The 19th Internasional Symposium Of FSTPT Islamic University Of Indonesia*.

Yulianto, Budi & Setiono. 2013. Kalibrasi Dan Validasi *Mixed Traffic Vissim Model*: Universitas Sebelas Maret.

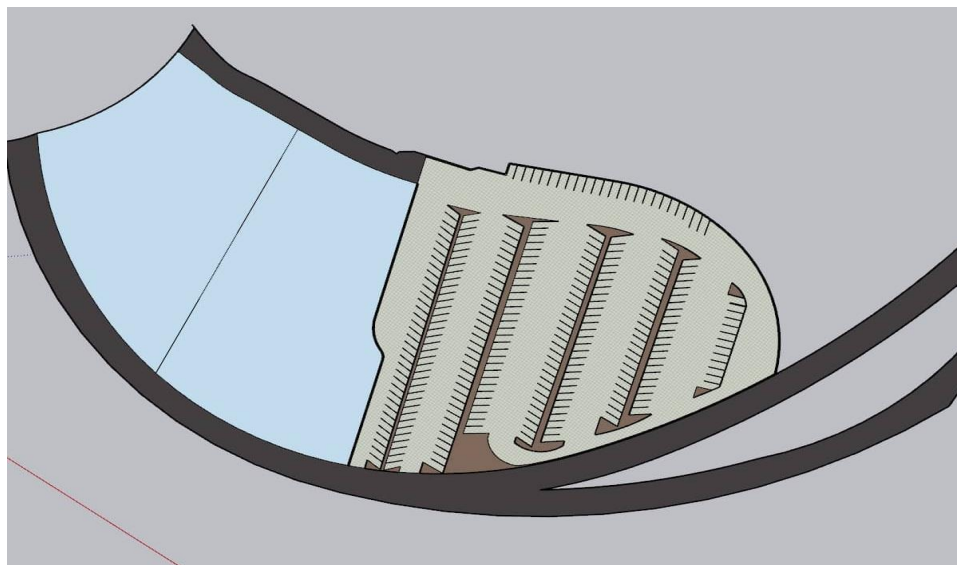
LAMPIRAN



Peta lokasi Studi



Sketsa Kawasan Parkir Basement 196 petak parkir



Dipindai dengan CamScanner

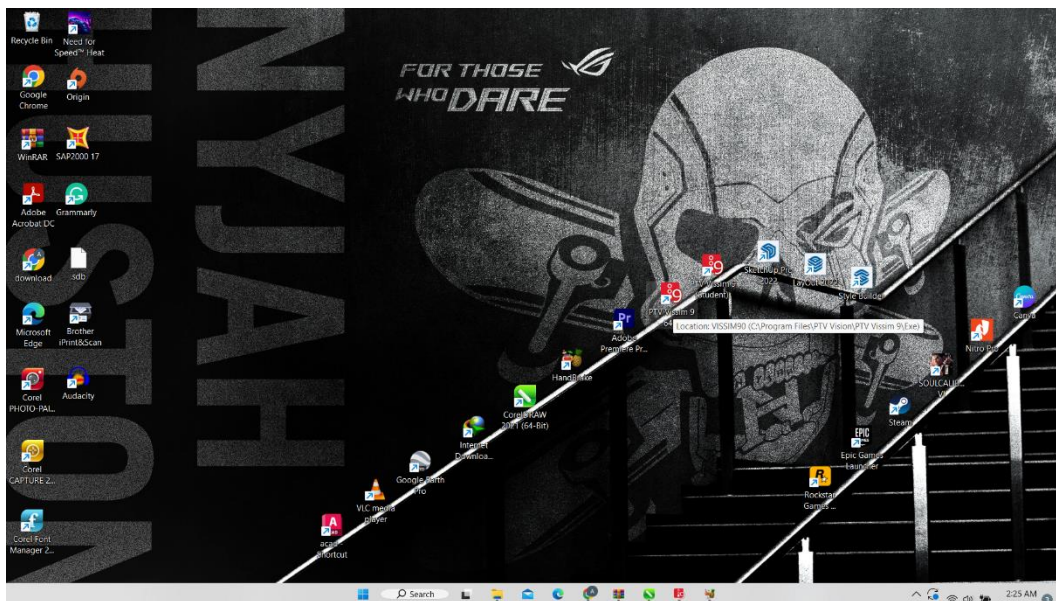
Sketsa Kawasan Parkir C1 224 petak parkir

TUTORIAL PTV VISSIM 9

Langkah-langkah dalam penggunaan aplikasi PTV *vissim* 9.0 :

1) Instalasi Pengenalan Tampilan Layar Program *Vissim*

1. Buka program PTV *Vissim* 9.0 yang ada pada desktop.

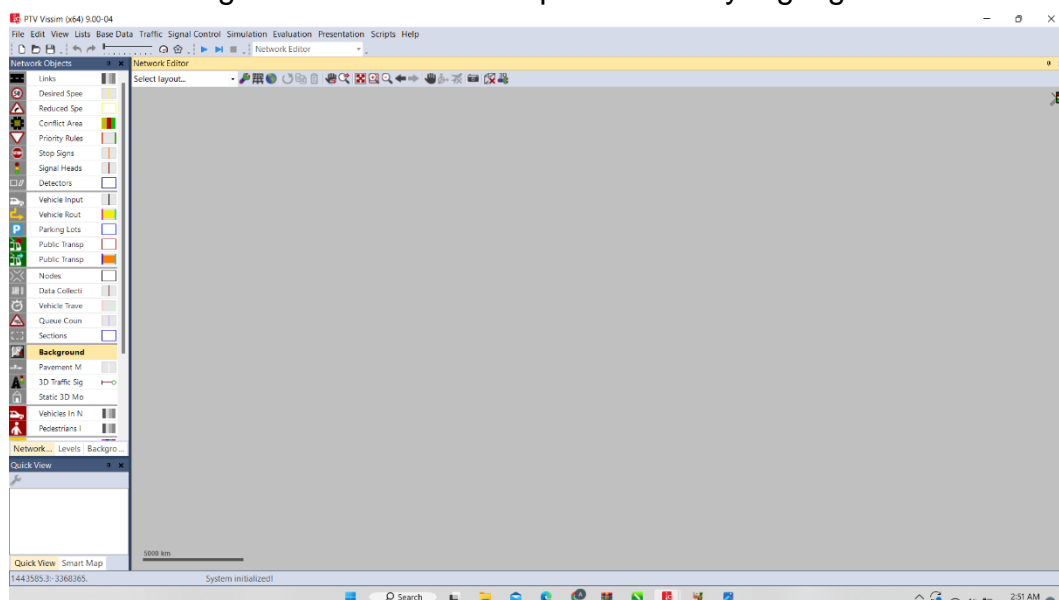


2. Selanjutnya kita menunggu proses *running*. Catatan: *Vissim* hanya bisa dijalankan menggunakan *Windows* bawahan yang original.

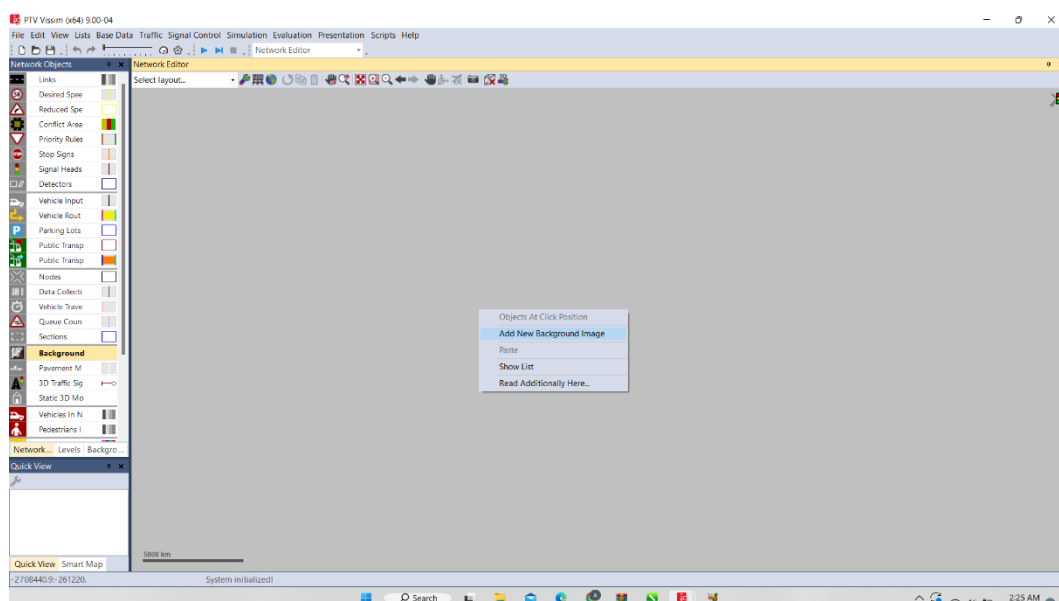


2) Pembangunan Jaringan Jalan

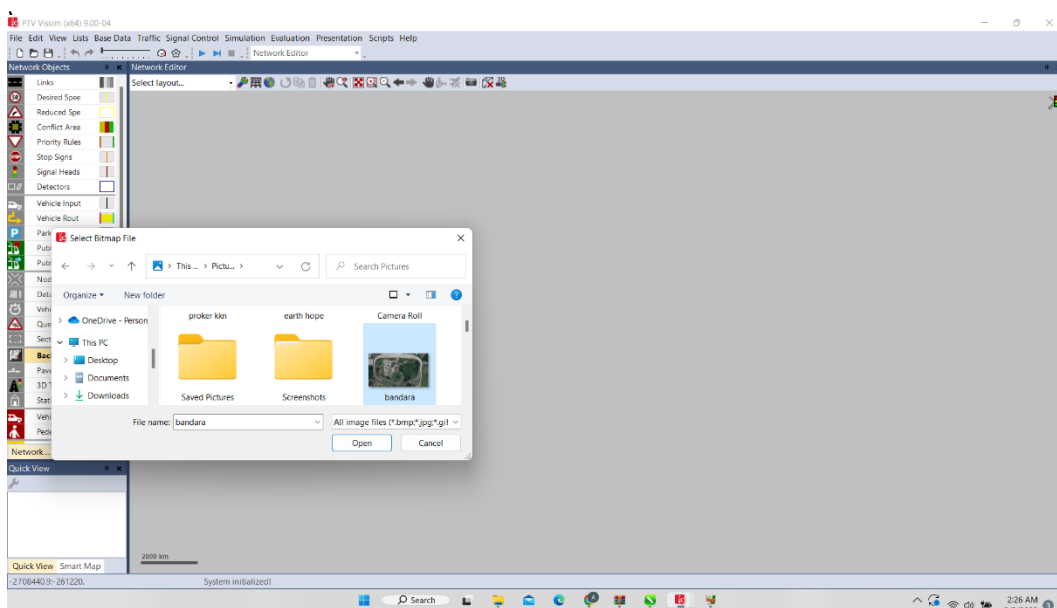
1. Mengatur terlebih dahulu skala display pada *vissim* dengan meng-*scroll* kearah bawah pada mouse yang digunakan.



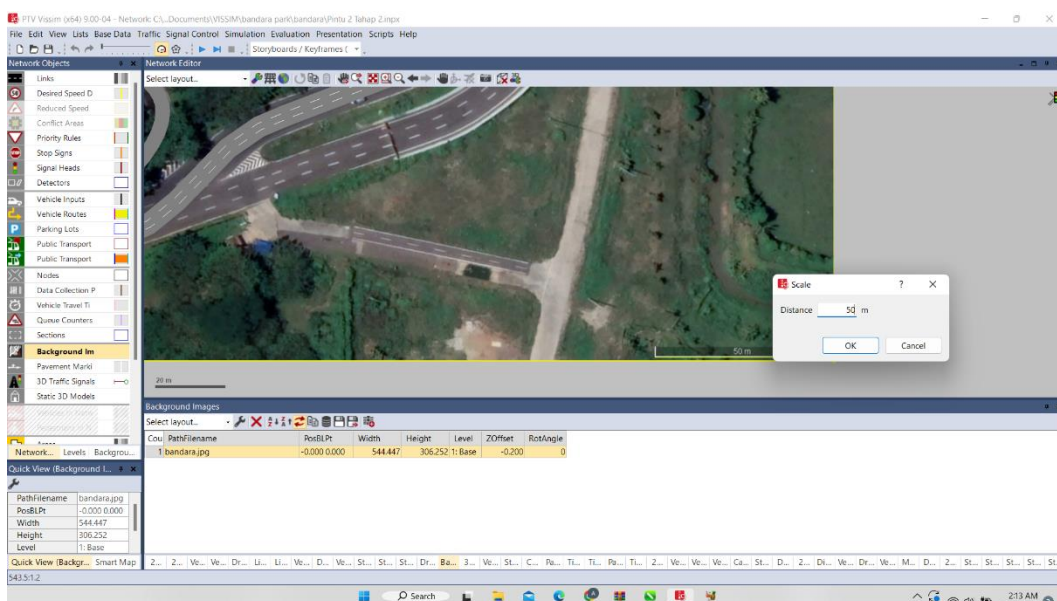
2. Langkah awal untuk membuat model simulasi yaitu memasukkan obyek atau *layout* gambar yang akan menjadi patron dalam membuat jaringan jalan pada *Vissim*, jadi kita insert *backgroundnya* terlebih dahulu setelah itu kita memilih *layout* gambar yang akan dibuat jaringan jalan pada *Vissim*.



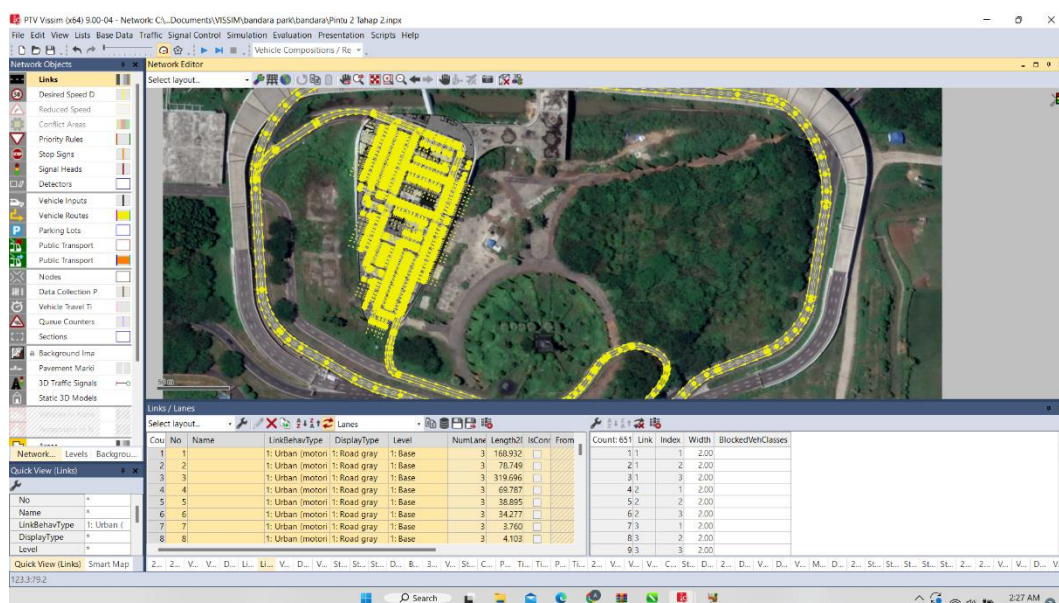
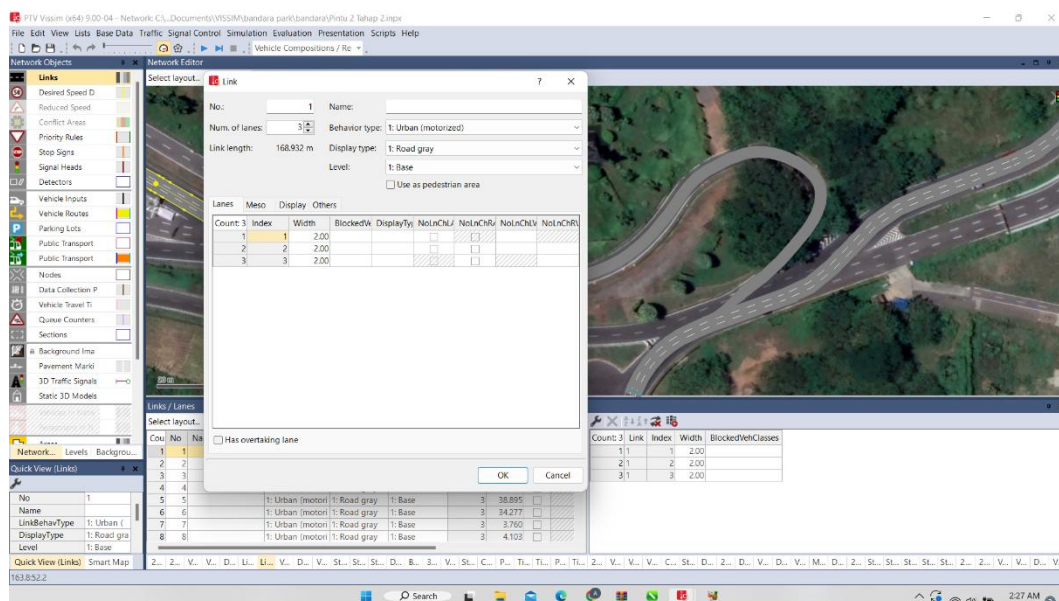
Selanjutnya kita memilih gambar yang kita inginkan yang tersimpan



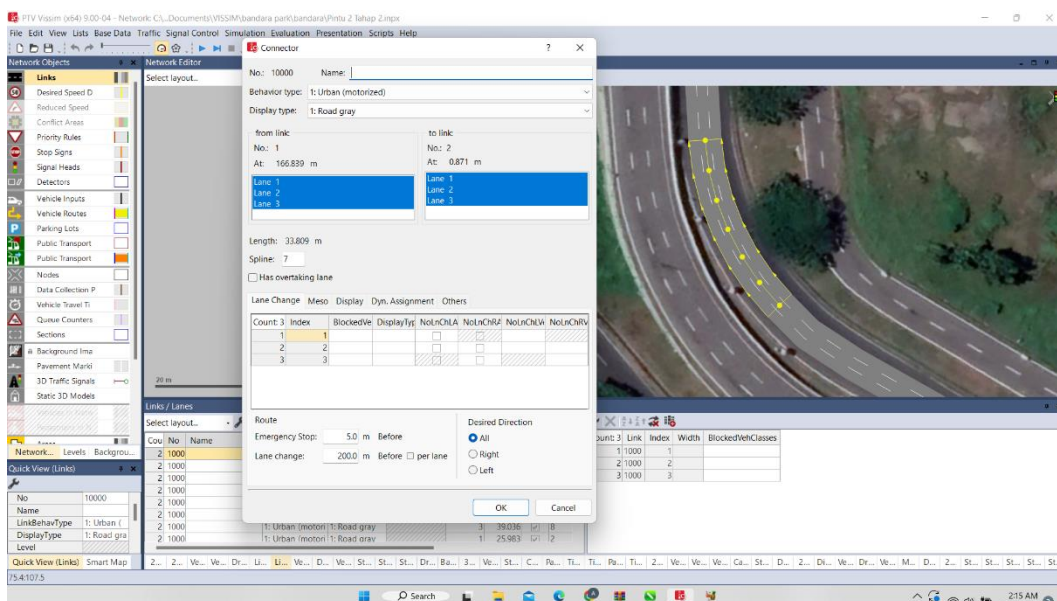
3. Setelah muncul gambar pada monitor *Vissim* kita Kembali mengatur skala pada gambar dengan klik kanan pada gambar-pilih *Set Scale* setelah itu seret garis sesuai pada skala yang diinginkan kemudian masukkan Panjang atau nilai skala tersebut.



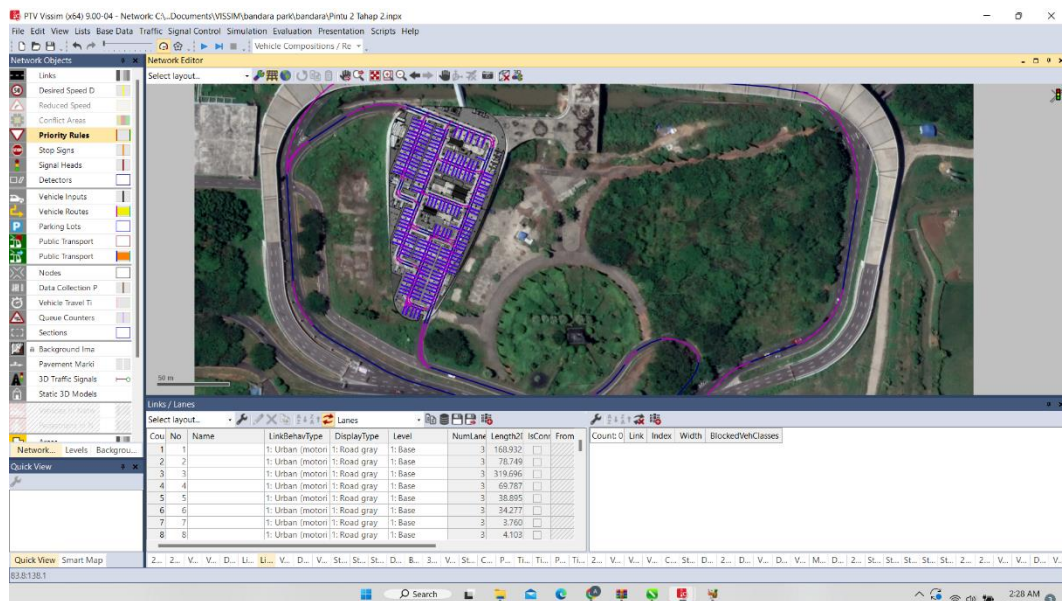
4. Setelah gambar atau *layout* yang menjadi objek terskalakan selanjutnya kita membangun jaringan atau jalan di atas *layout* tersebut dengan memilih *mode insert link* pada *tools* yang ada pada *Vissim*, dengan memasukkan nama jalan, jumlah lajur, dan ukuran/dimensi lajur.



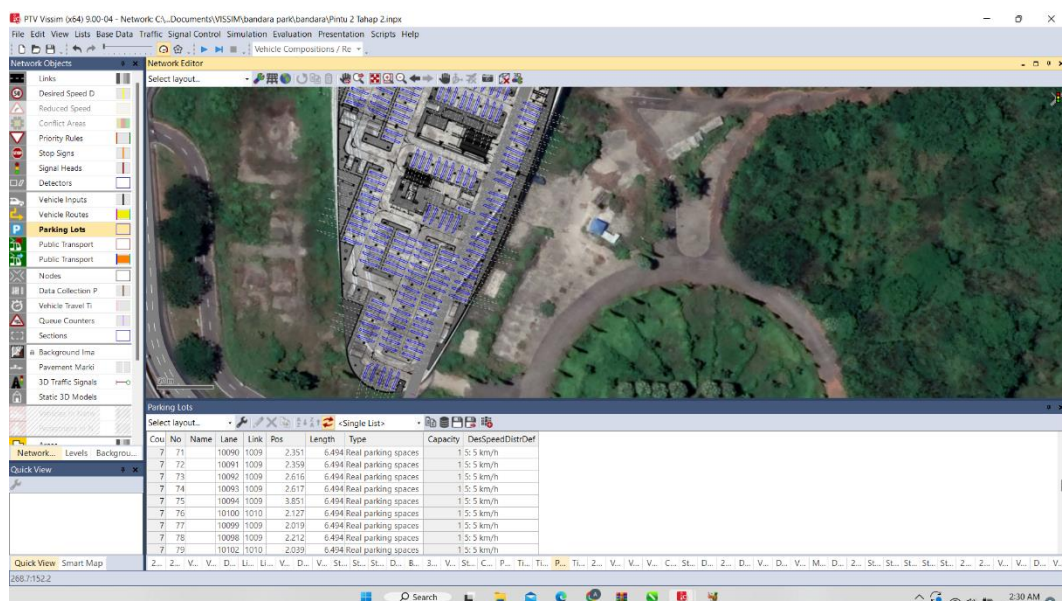
5. Setelah jaringan selesai dibuat seluruhnya, Langkah berikutnya kita menghubungkan jaringan jalan tersebut dengan mode *connector*. Tiap jaringan dihubungkan satu sama lain dengan Click Kanan + SHIFT setelah kita tarik penghubung dari link yang satu ke link yang lainnya sehingga semuanya dapat terhubung.

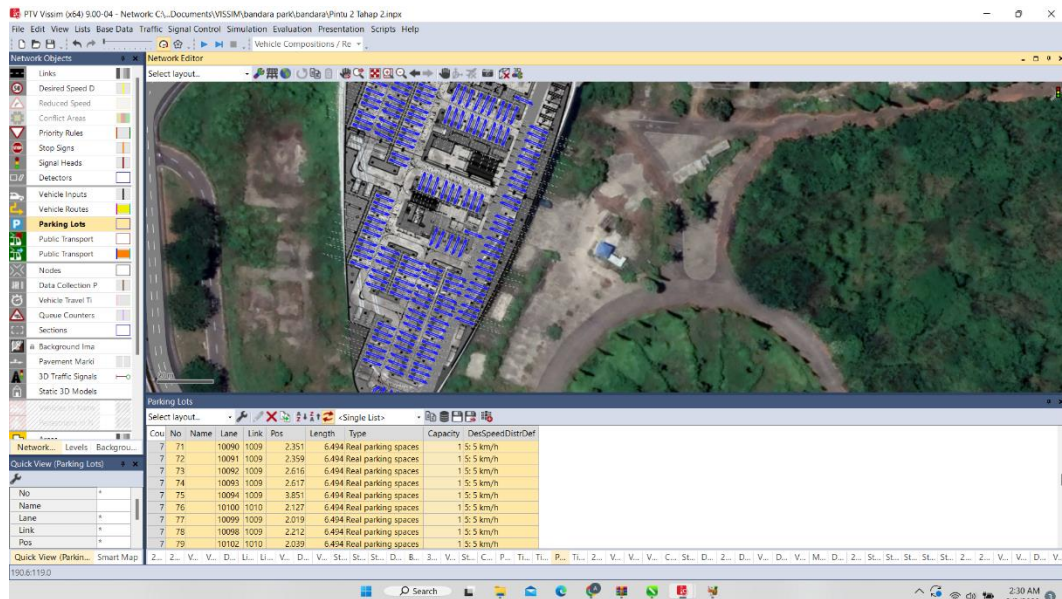


Selanjutnya Ketika semua sudah terhubung kita menekan CTRL + A untuk menunjukkan perbedaan antara jaringan utama dan penghubung pada gambar di bawah ini, garis biru menunjukkan jaringan utama sedangkan garis merah muda menunjukkan jaringan penghubung.

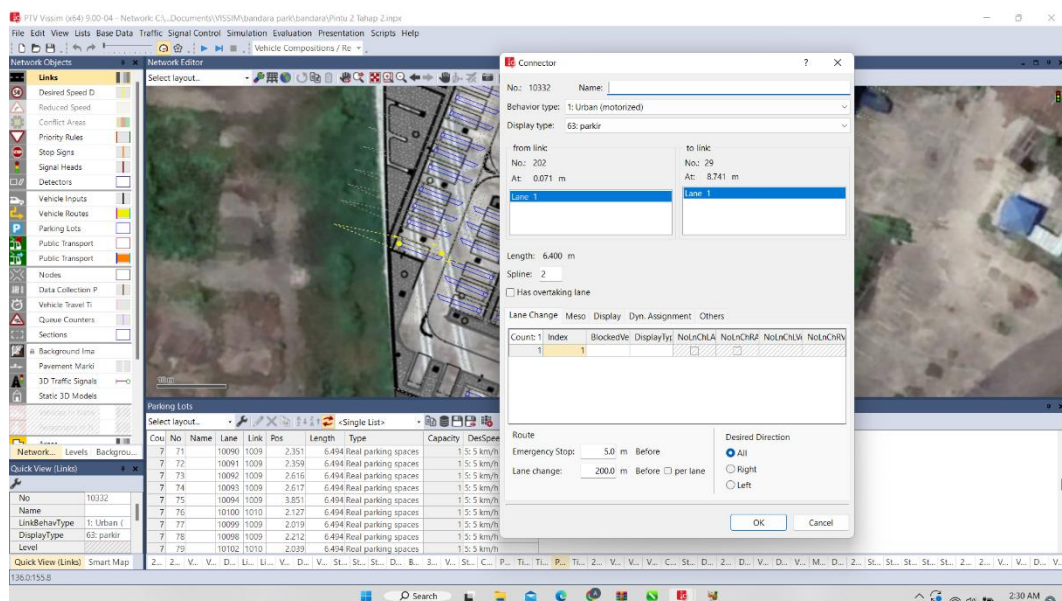


6. Pilih *Parking Lot* diantara Link dan Konektor pada setiap lahan parkir kemudian Tarik sesuai dengan ukuran ruang parkir dan pilih parking spaces untuk menentukan berapa waktu yang digunakan kendaraan untuk mengatur posisi kendaraan saat parkir



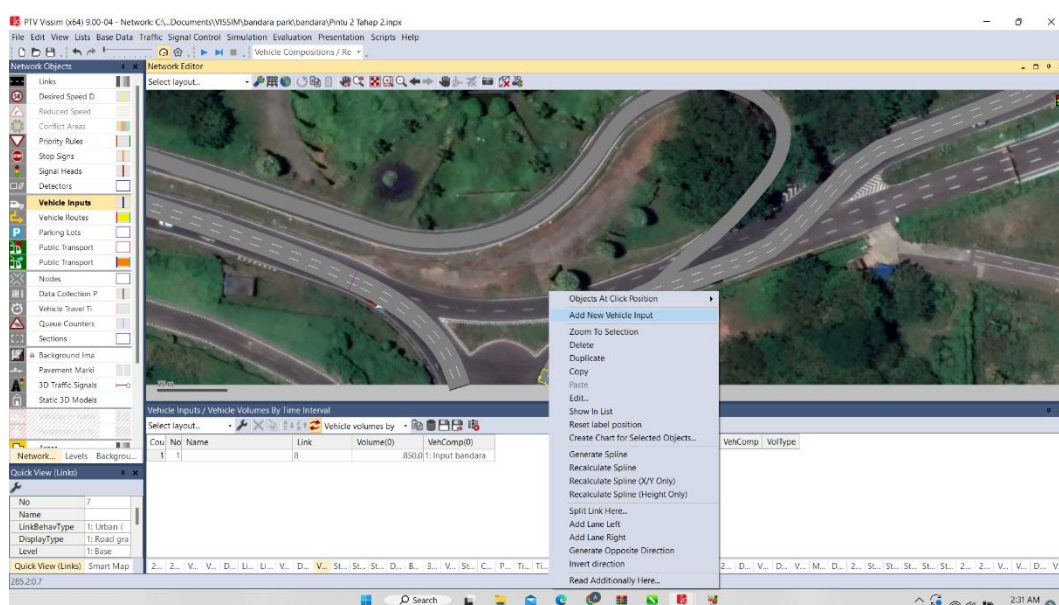


7. Selanjutnya klik tiap link pada lahan parkir lalu Tarik garis tengah kebawah hingga membentuk jalur mundur dan maju seperti pada gambar.

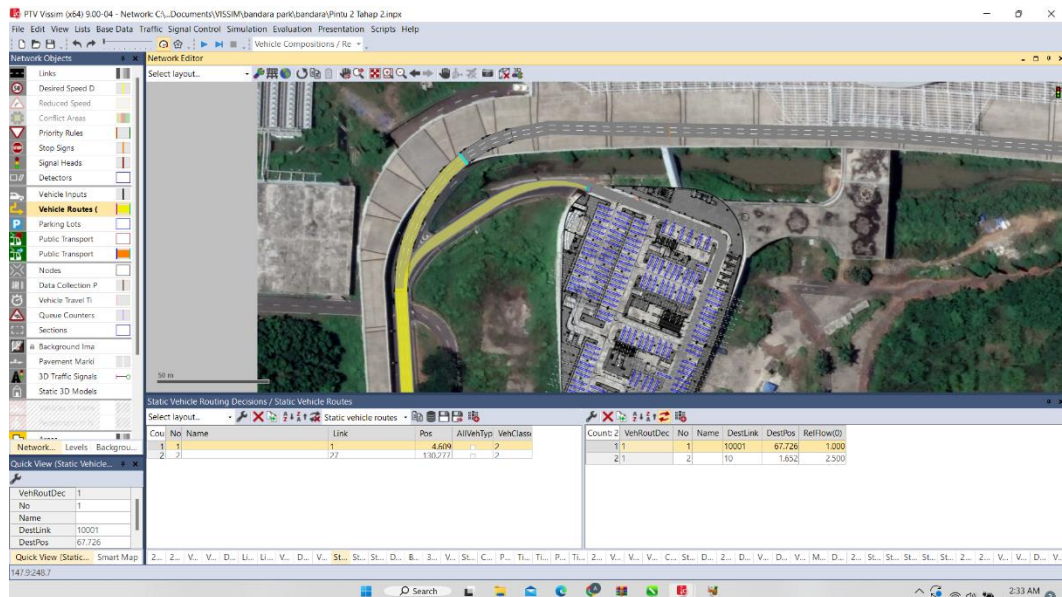


3) Pemasukkan Data

1. Tahap selanjutnya kita menginput jumlah kendaraan serta rute kendaraan yang akan melaju pada jaringan jalan yang kita buat, untuk memasukkan kendaraan kita memilih mode *Vehicle Input*
Catatan : jumlah kendaraan yang di input dalam satuan kendaraan/jam

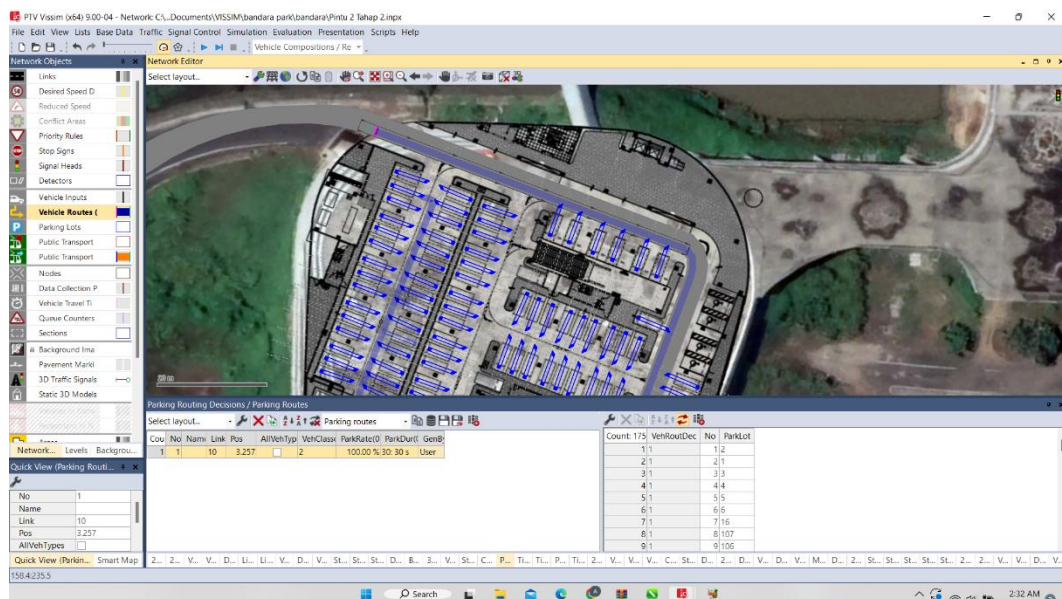


2. Setelah kendaraan selesai di Input, selanjutnya mengatur rute kendaraan, memilih mode *Vehicle Routes*. Kemudian Parking lot, setelah itu memilih link pertama sebagai route awal berikutnya klik kanan pilih *Add New Vehicle Routing Decition* selanjutnya insert route yang diinginkan lalu arahkan kearah pergerakan kendaraan yang sesuai dengan arah pergerakan parkir.

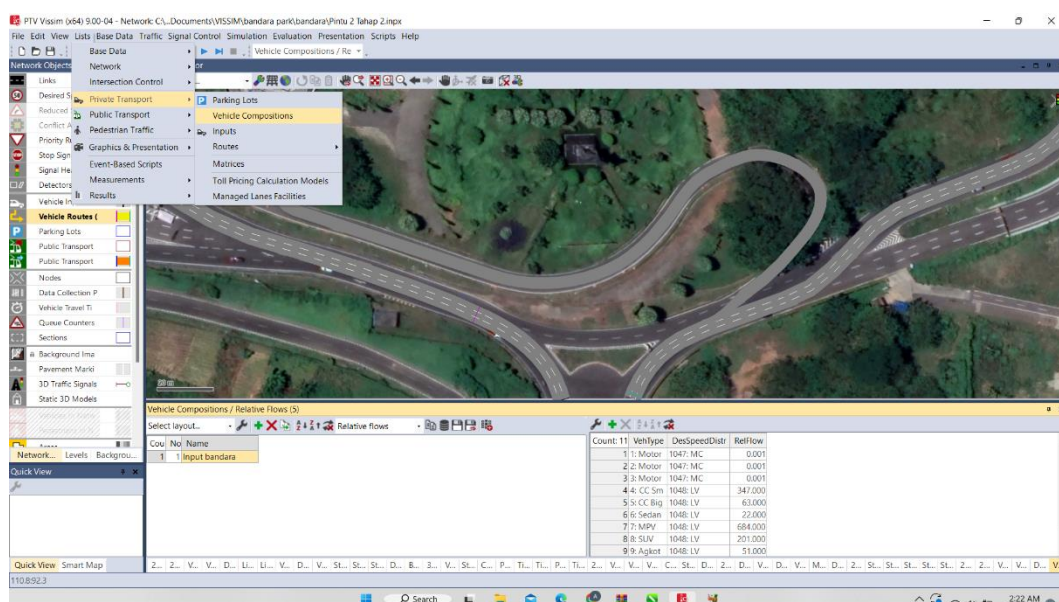


3. Selanjutnya kita akan mengatur komposisi kendaraan yang bergerak lurus dan berbelok sesuai arah rute pergerakan yang di insert masuk pada jaringan jalan.

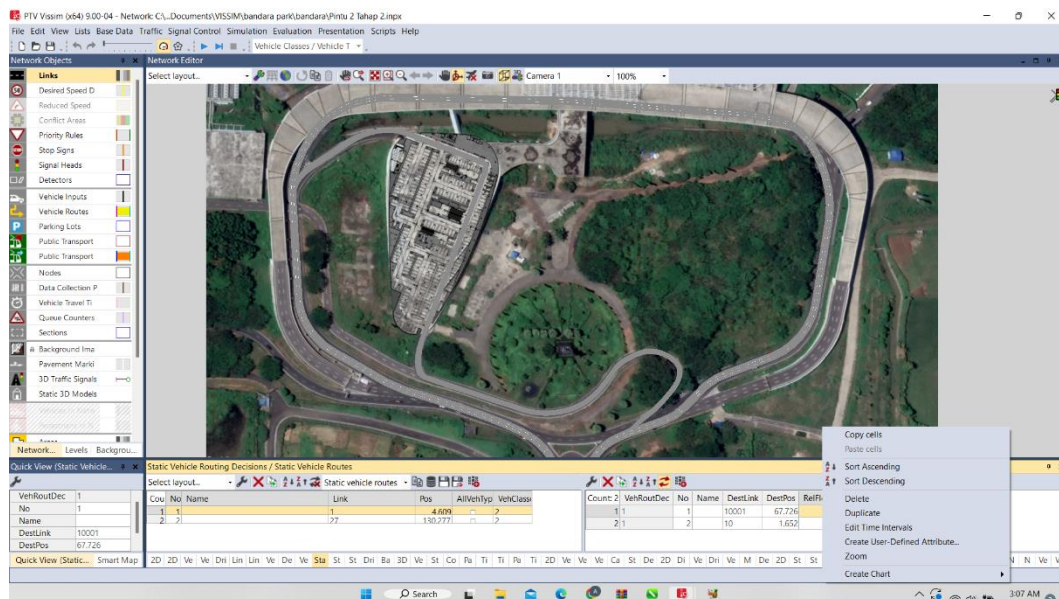
Catatan: komposisi kendaraan bisa dalam satuan % atau satuan jumlah kendaraan



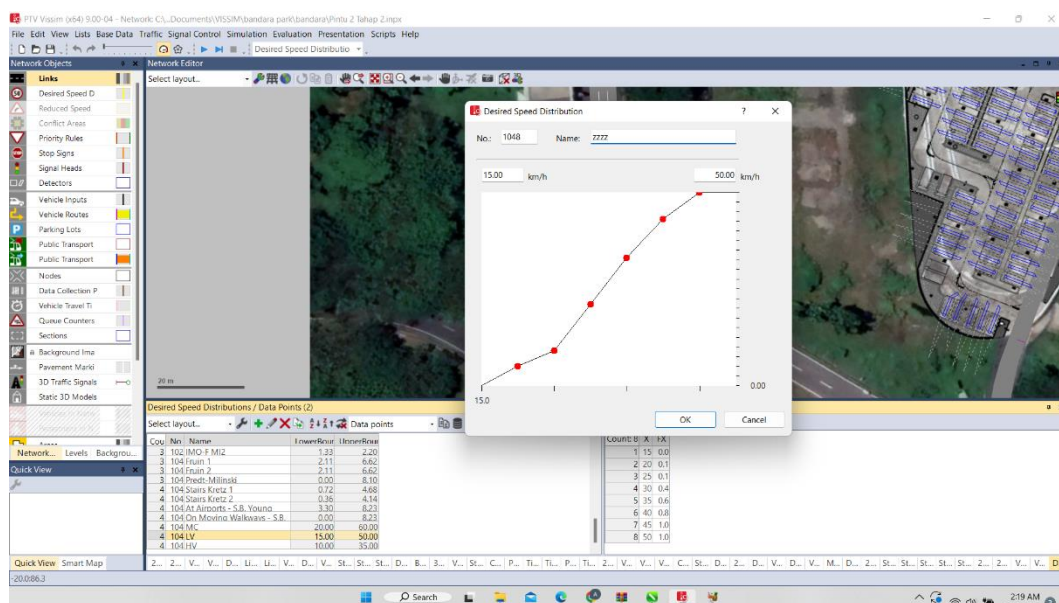
4. Setelah *insert* volume kendaraan dan arah pergerakan kendaraan selesai selanjutnya kita mengatur komposisi jenis kendaraan yang melintas. Jenis-jenis kendaraan pada Vissim dibagi menjadi enam yaitu: Car, HGV, SUV, Bus, Bike dan Tram pada kasus kali ini kita memasukkan City Car, Sedan, MVP, SUV sebagai objek, Langkah awal kita memilih *tools Traffic – Vehicle Composition*.



5. Setelah masuk kedalam text boxnya atau menu kita dapat menambahkan jenis kendaraan yang kita inginkan dengan cara klik kanan kemudian pilih add selanjutnya kita pilih jenis kendaraan yang ingin kita masukkan pada program *Vissim*.

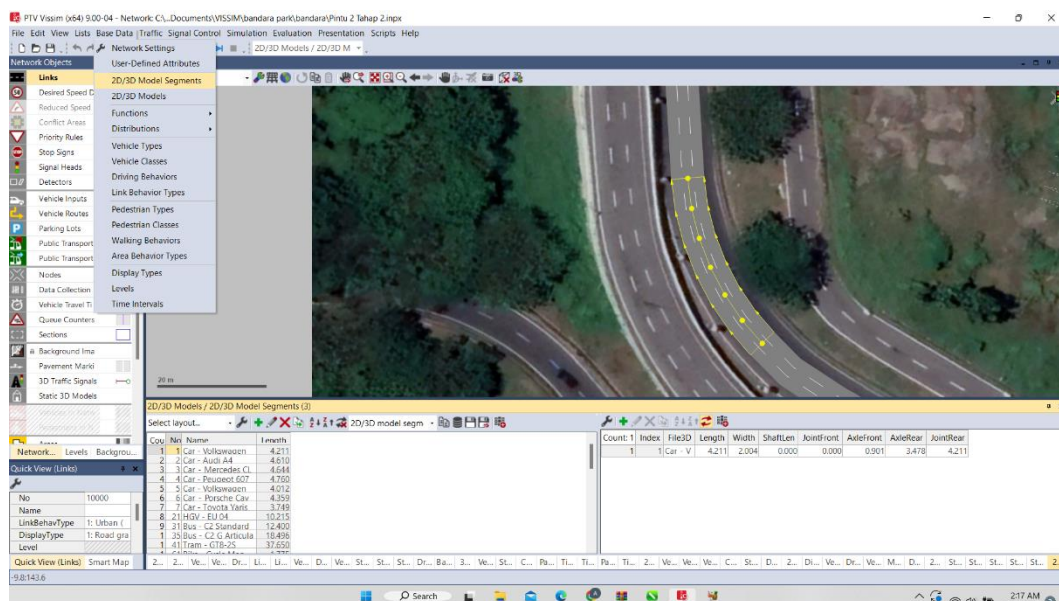


Kemudian kita mengatur kecepatan kendaraan pada *desired speed distribution*

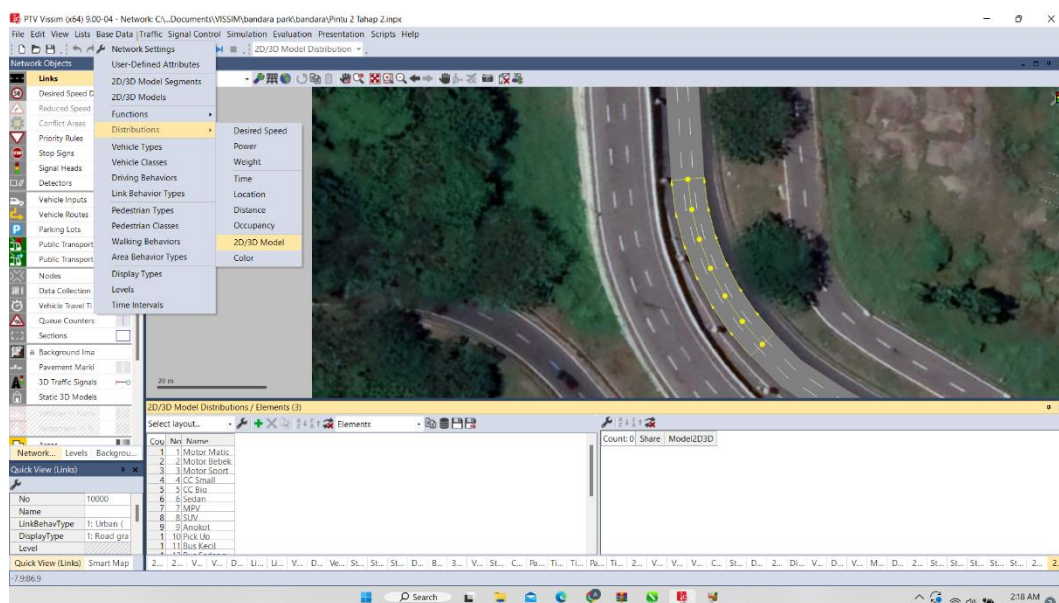


6. Setelah selesai selanjutnya kita mengatur komposisi jenis kendaraan pada *relative flow*.

Catatan: satuan digunakan % kendaraan

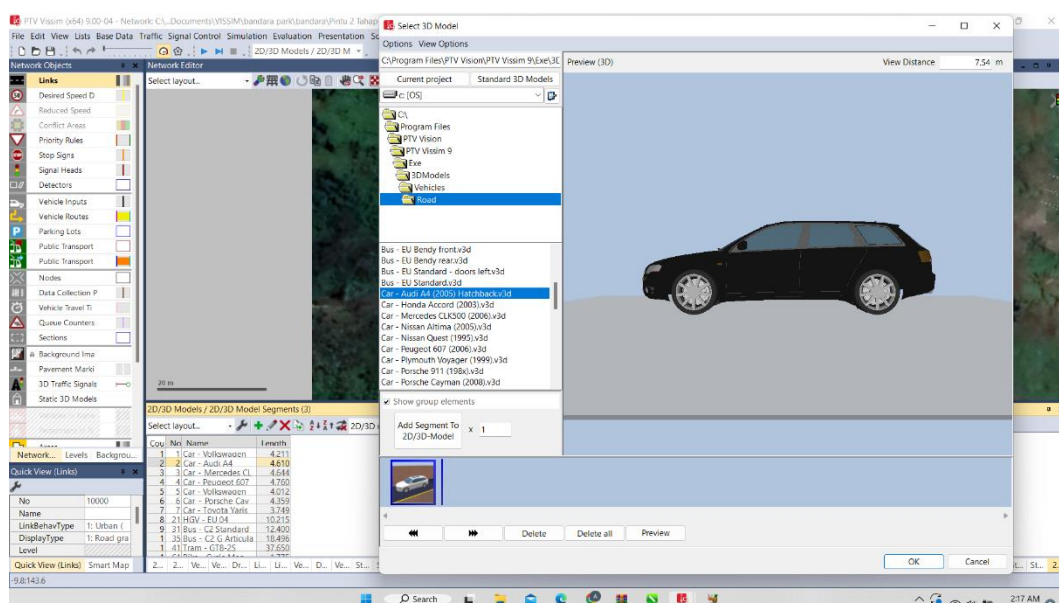


7. Karena *default display* jenis kendaraan bike itu adalah sepeda roda dua jadi kita harus menginput ulang *display* kendaraan bike tersebut dengan *display* sepeda motor dengan masuk ke menu Base data- 2D/3D Models segment



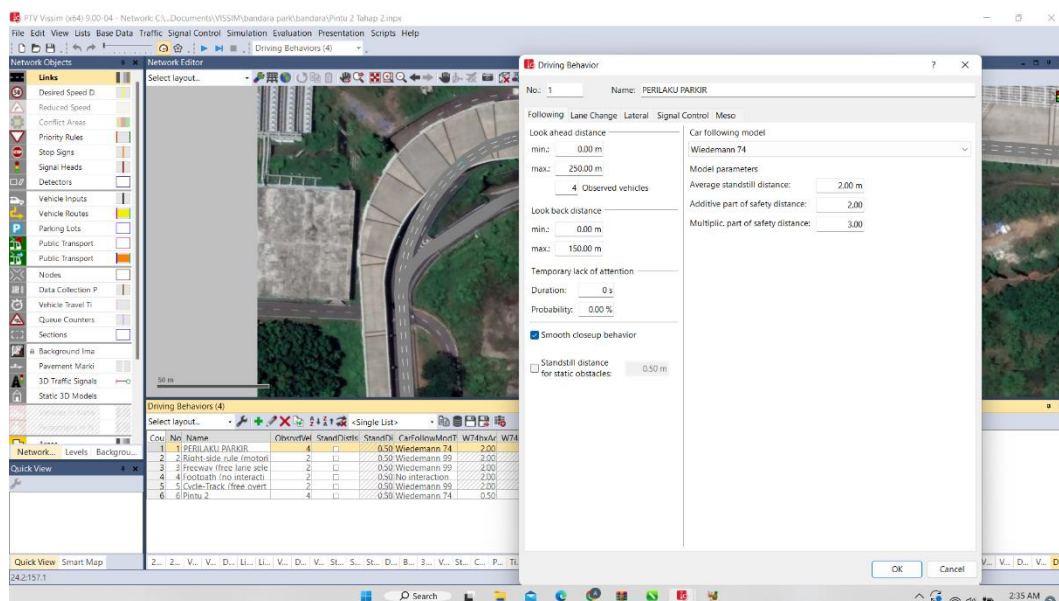
Setelah masuk menu 2D/3D Models Segment kita mencari jenis kendaraan bike kemudian klik kanan + edit selanjutnya kita pilih display kendaraan roda empat kemudian pilih ok.

Catatan: dapat juga ditambahkan display jenis kendaraan roda empat yang lainnya.

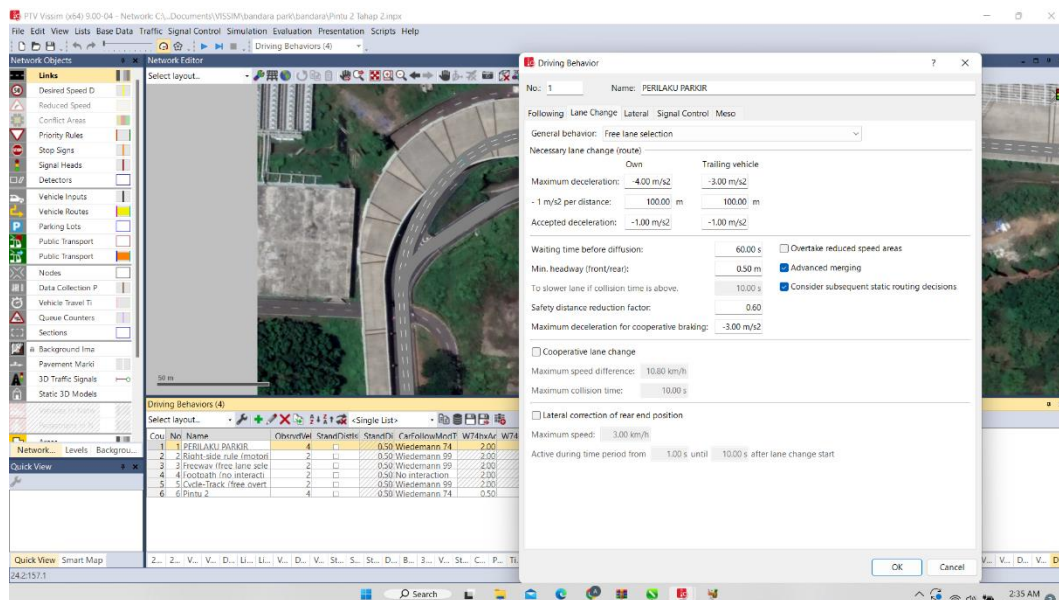


4) Proses Analisa Data

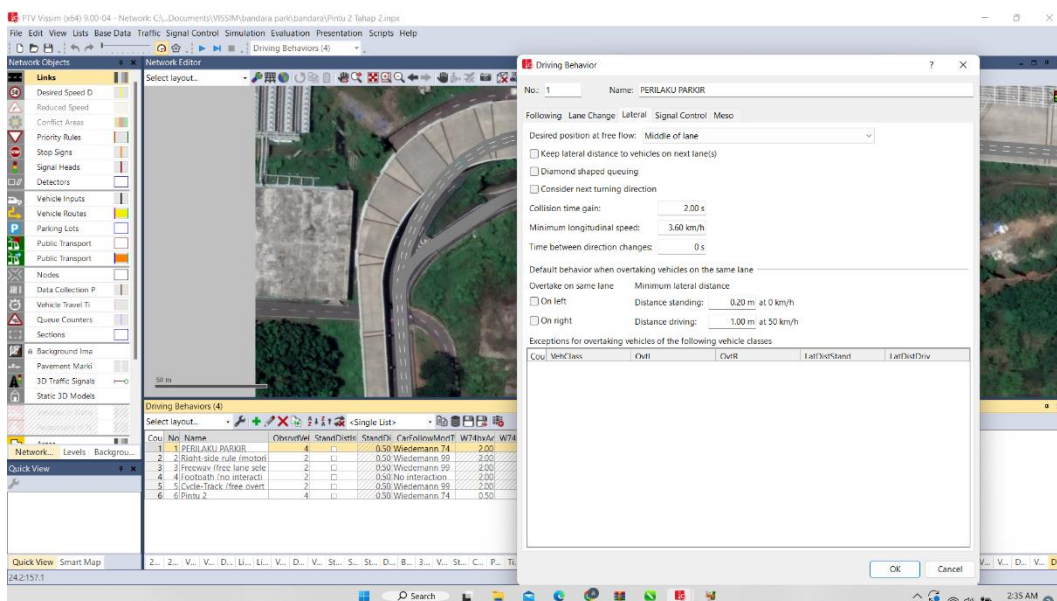
1. Setelah selesai pengimputan data volume dan kecepatan pada studi kasus ini, selanjutnya yaitu pengaturan kalibrasi dengan mengatur *divine behavior* pada simpang tersebut. Mengatur *average standstill distance*, *safety distance*, *standstill distance*, *look ahead distance* dan *look back distance*.



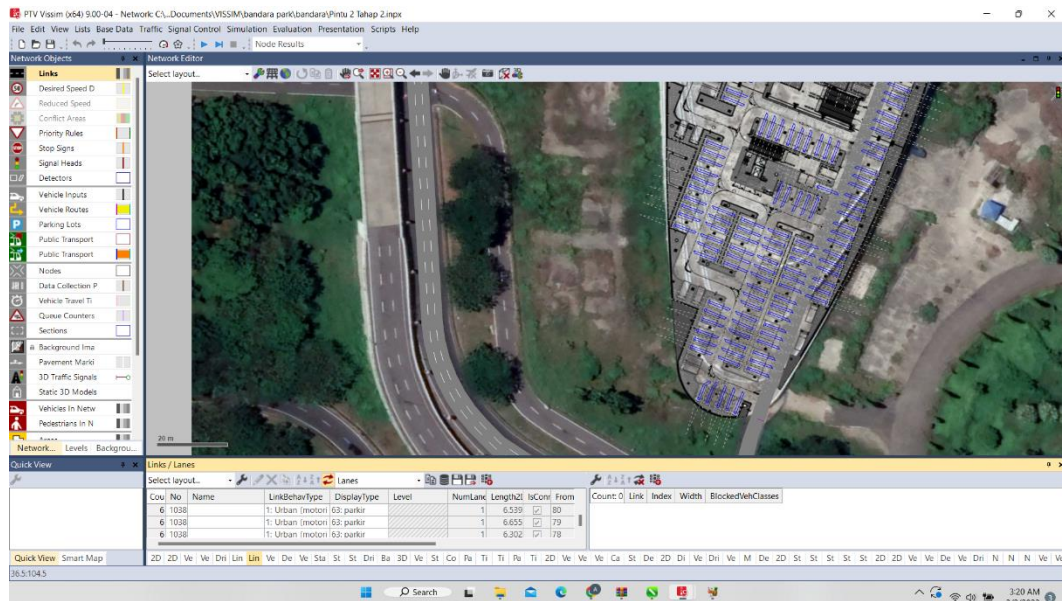
2. Mengatur *Lane Change* pada *Driving Behavior*, dengan mengubah *General Behavior* menjadi *Free Lance Selection*, mengatur jarak antar kendaraan/min. *Headway*



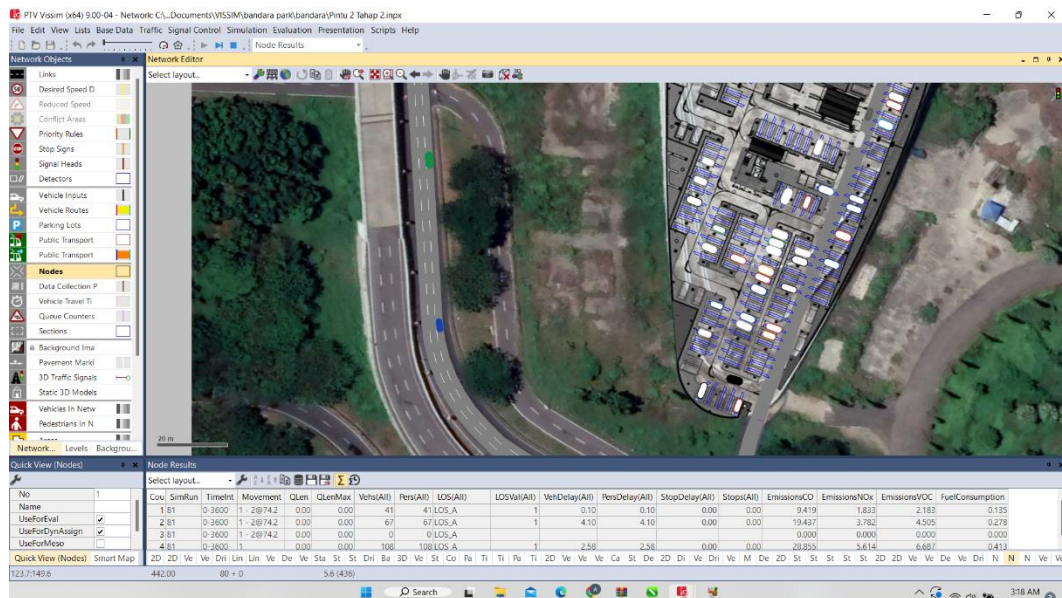
3. Kemudian mengatur Lateral pada *Driving Behavior*, yaitu posisi kendaraan dimana dapat menyiap di lajur mana saja, baik kiri dan kanan, *Desired Position Any*. Setelah itu mengatur Minimum *Lateral Distance*, yaitu *distance standing* dan *distance driving*.



4. Setelah pengaturan *Driving Behavior*, maka ditentukan persimpangan yang akan ditinjau dengan menambahkan *Nodes*. *Nodes* tersebut berfungsi untuk menentukan titik pembacaan hasil *running* yang dapat menghasilkan Panjang antrian, tundaan dan volume kendaraan.



5. Selanjutnya *running* akan menghasilkan *Node Results* dimana kita dapat melihat volume kendaraan yang melewati titik simpang, tundaan kendaraan dan Panjang antrian pada *Nodes* yang telah ditentukan sebelumnya



DOKUMENTASI

