



DATA MINING FOR NETWORK SECURITY

WORKSHOP PENINGKATAN CYBER SITUATIONAL AWARENESS DENGAN
MEMANFAATKAN SISTEM DETEKSI DINI NASIONAL



11 SEPTEMBER 2019

TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS
INDONESIA

1. Instalasi Software

Tools yang dibutuhkan :

1. Wireshark

Wireshark adalah program Network Protocol Analyzer alias penganalisa protokol jaringan yang lengkap. Program ini dapat merakam semua paket yang lewat serta menyeleksi dan menampilkan data tersebut sedetail mungkin.



Gambar 1.1. Logo Wireshark

2. Knime Analytics Platform

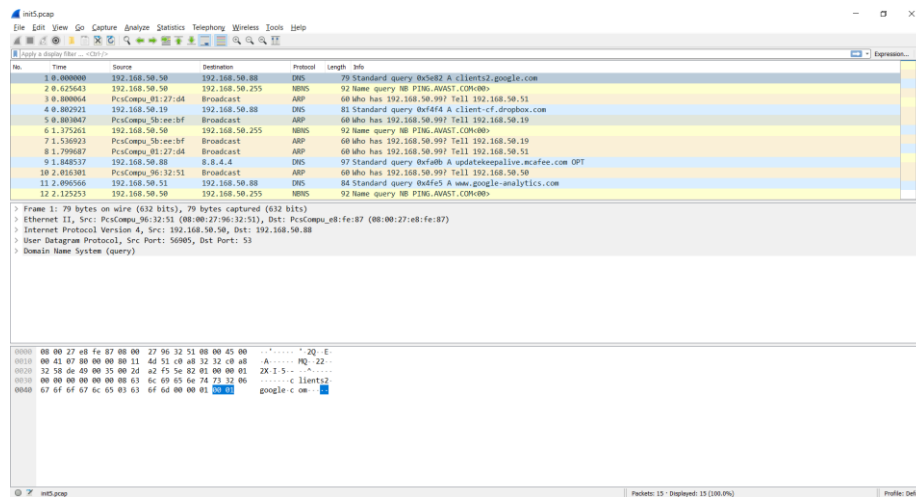
Knime Analytics Platform adalah software open source untuk membuat model data science. Knime membuat pemahaman data dan merancang alur kerja data science dan komponen yang dapat digunakan kembali.



Gambar 1.2. Logo Knime

A. Instalasi Wireshark

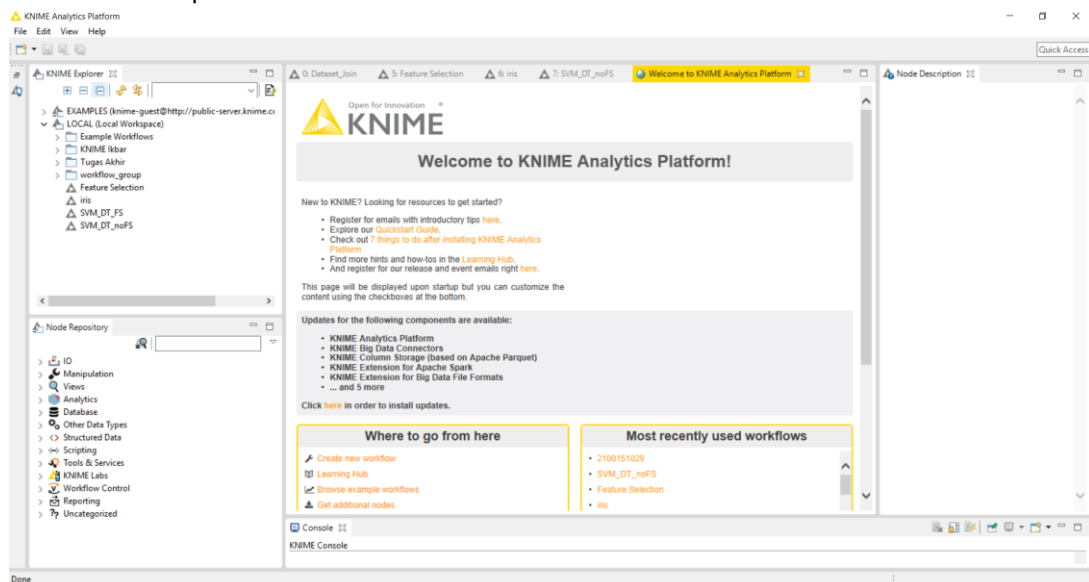
1. Software ini dapat didownload pada halaman <https://www.wireshark.org/>. Kemudian sesuaikan dengan OS pada komputer.
2. Untuk instalasi nya cukup mudah, ikuti saja alur instalasinya dengan pengaturan default.
3. Berikut capture dari software wireshark yang sudah diinstall.



Gambar 1.3. Tampilan Utama Wireshark

B. Instalasi Ktime Analytics Platform

1. Software ini dapat didownload pada halaman <https://www.knime.com/downloads/download-knime> sesuai juga dengan OS pada komputer
2. Untuk instalasi gunakan pengaturan default pada saat proses instalasi
3. Berikut halaman pertama dari software Knime



Gambar 1.4. Tampilan utama KNIME Analytics Platform

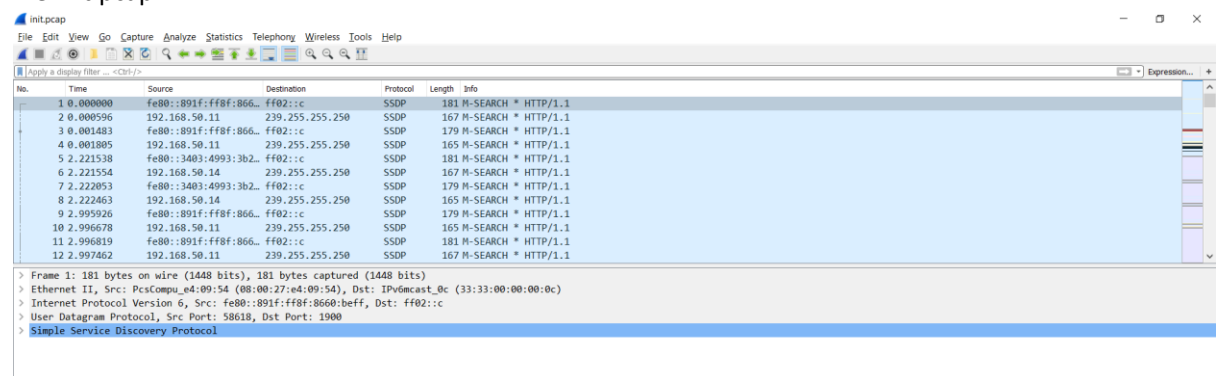
2. Wireshark

1. Proses ini digunakan untuk mengamati file Packet Capture (.pcap). File tersebut berisi lalu lintas jaringan yang ditangkap oleh komputer. Pada kasus kali ini akan digunakan dataset traffic DNS ISOT yang berasal dari University of Victoria karena terdapat simulasi serangan Botnet pada traffic DNS. Untuk memperoleh dataset ini dapat mengunjungi link : <https://www.uvic.ca/engineering/ece/isot/datasets/>
2. File tersebut dibagi menjadi 5 yaitu : init.pcap, init2.pcap, init3.pcap, init4.pcap, init5.pcap

Name	Date modified	Type	Size
init.pcap	11/28/2017 4:52 PM	Wireshark capture ...	163,160 KB
init2.pcap	11/28/2017 4:51 PM	Wireshark capture ...	576,966 KB
init3.pcap	11/28/2017 4:51 PM	Wireshark capture ...	159,862 KB
init4.pcap	11/28/2017 4:52 PM	Wireshark capture ...	656,556 KB
init5.pcap	11/28/2017 4:52 PM	Wireshark capture ...	2 KB

Gambar 2.1. File PCAP dataset ISOT

3. Kemudian buka file tersebut secara bergantian menggunakan Wireshark. Pada langkah ini kita gunakan file init.pcap



Gambar 2.2. Open File init.pcap

4. Untuk mempermudah pada saat proses analisa yang akan dilakukan nantinya, kita akan mengambil data dengan ip versi 4 (ipv4) dan protocol TCP, DNS saja. Untuk proses tersebut dapat dilakukan pada wireshark menggunakan perintah `ip.version==4 && tcp || dns` pada kolom *display filter* tepat dibawah toolbar.

init.pcap

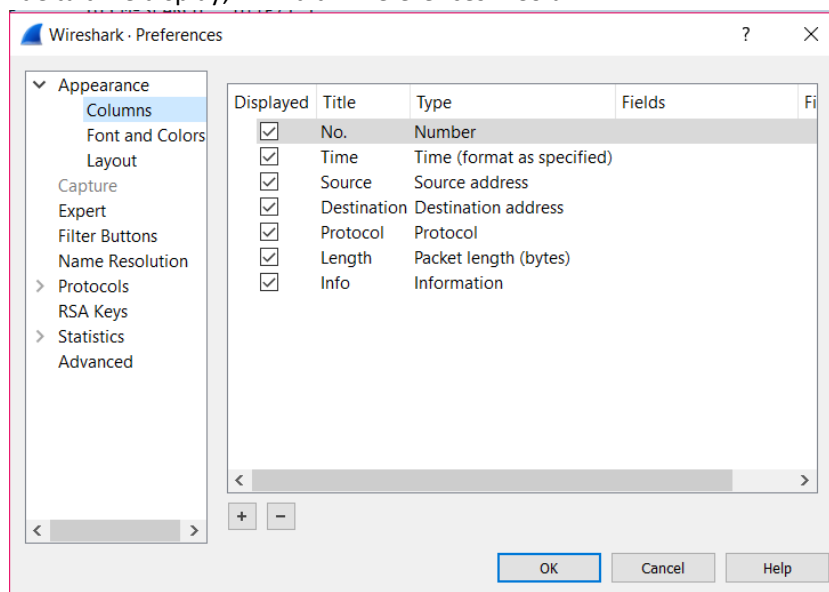
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.version==4 && tcp || dns

No.	Time	Delta_time	Delta_time_disp	Source	Destination	Protocol
79	309.509587	4.427582	0.000000	192.168.50.1	192.168.50.88	TCP
80	309.509587	0.000000	0.000000	192.168.50.88	192.168.50.1	TCP
81	312.263619	2.754032	2.754032	192.168.50.1	192.168.50.88	TCP
82	312.263647	0.000028	0.000028	192.168.50.88	192.168.50.1	TCP
101	462.026672	7.771406	149.763025	192.168.50.88	142.104.64.205	TCP
102	463.024215	0.997543	0.997543	192.168.50.88	142.104.64.205	TCP
106	472.199177	5.162794	9.174962	192.168.50.88	142.104.64.205	TCP

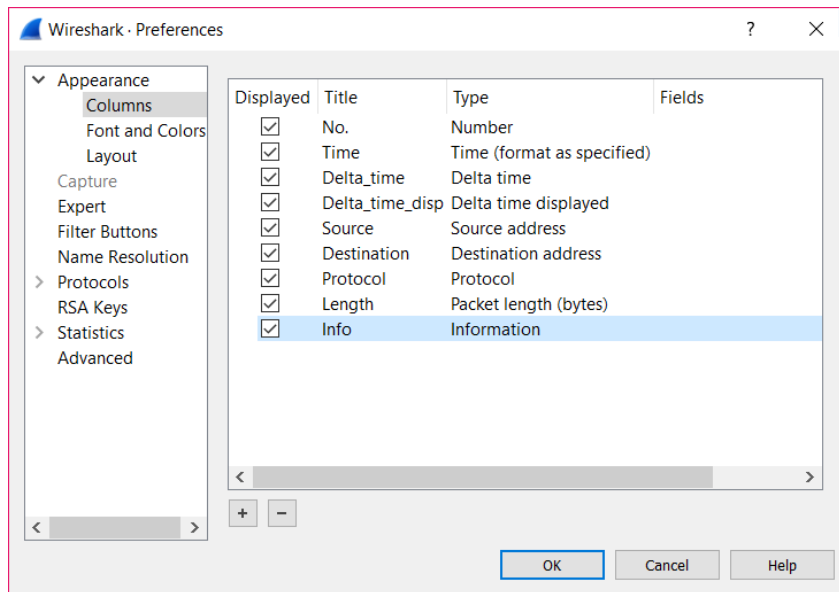
Gambar 2.3. Hasil Filter

- Kemudian kita membutuhkan kolom tambahan yaitu delta time. Untuk mendapatkan delta time dan delta time dan delta time display, klik Edit – Preferences – Column



Gambar 2.4. Menambahkan kolom Delta_time

Kemudain klik pada tanda + untuk menambah kolom baru. Kemudian pada Type, pilih Delta Time. Kemudian lakukan hal yang sama untuk kolom delta time display



Gambar 2.5. Menambahkan Kolom Delta_time_disp

Kemudian Klik OK. Berikut hasilnya.

init.pcap

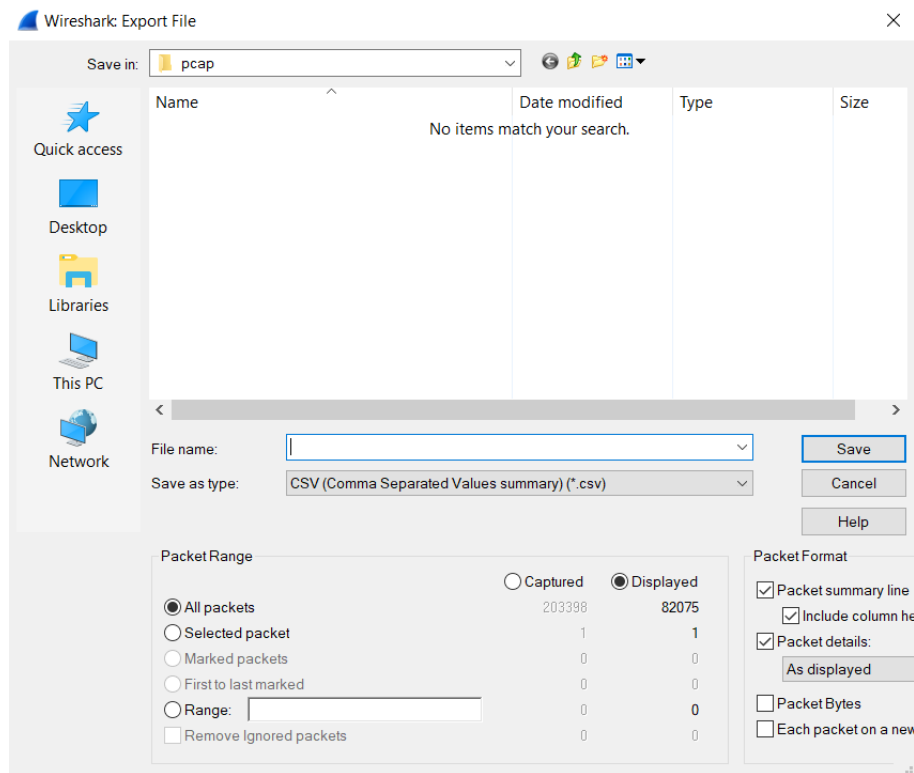
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.version==4 && tcp || dns

No.	Time	Delta_time	Delta_time_disp	Source	Destination	Protocol
79	309.509587	4.427582	0.000000	192.168.50.1	192.168.50.88	TCP
80	309.509587	0.000000	0.000000	192.168.50.88	192.168.50.1	TCP
81	312.263619	2.754032	2.754032	192.168.50.1	192.168.50.88	TCP
82	312.263647	0.000028	0.000028	192.168.50.88	192.168.50.1	TCP

Gambar 2.6. Hasil Penambahan Kolom

- Langkah terakhir yaitu export file pcap tersebut keformat Comma-separated Value (.csv) dengan cara klik File – Export Packet Dissections – As CSV. Yang perlu diperhatikan yaitu pada Pacet Range, pastikan yang terpilih yaitu Displayed, karena data pada Displayed ini sudah terfilter dengan nip version 4.



Gambar 2.7. Menyimpan file .csv

7. Lakukan semua proses diatas pada dataset berikutnya (init2.pcap, init3.pcap, init4.pcap, init5.pcap) hingga seluruh data sudah terkonversi ke dalam format .csv

Name	Date modified	Type	Size
init.csv	8/26/2019 3:25 PM	Microsoft Excel Co...	12,570 KB
init2.csv	8/26/2019 3:26 PM	Microsoft Excel Co...	71,470 KB
init3.csv	8/26/2019 3:27 PM	Microsoft Excel Co...	173,210 KB
init4.csv	8/26/2019 3:29 PM	Microsoft Excel Co...	499,249 KB
init5.csv	8/26/2019 3:29 PM	Microsoft Excel Co...	2 KB

Gambar 2.8. Hasil Konversi PCAP ke CSV

3. Knime Analytics Platform

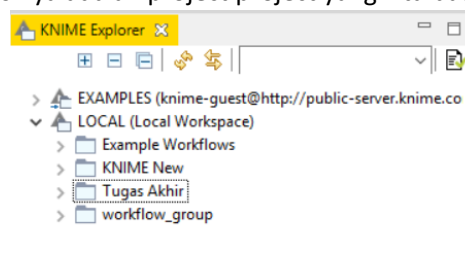
A. Penggabungan Data

1. Setelah semua file tadi telah diexport menjadi file .csv. Buka software Knime Analytics Platform untuk melakukan proses analisa pada traffic DNS. Berikut adalah file yang telah terexport menjadi csv

Name	Date modified	Type	Size
init.csv	8/26/2019 3:25 PM	Microsoft Excel Co...	12,570 KB
init2.csv	8/26/2019 3:26 PM	Microsoft Excel Co...	71,470 KB
init3.csv	8/26/2019 3:27 PM	Microsoft Excel Co...	173,210 KB
init4.csv	8/26/2019 3:29 PM	Microsoft Excel Co...	499,249 KB
init5.csv	8/26/2019 3:29 PM	Microsoft Excel Co...	2 KB

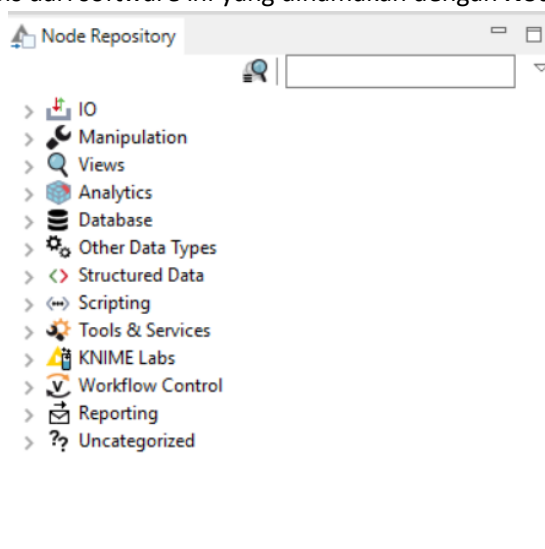
Gambar 3.1. File CSV

2. Setelah software Knime telah terbuka. Terdapat 3 bagian utama dari software ini. Yang pertama yaitu Knime Explorer yang isinya adalah project project yang kita buat pada software ini.



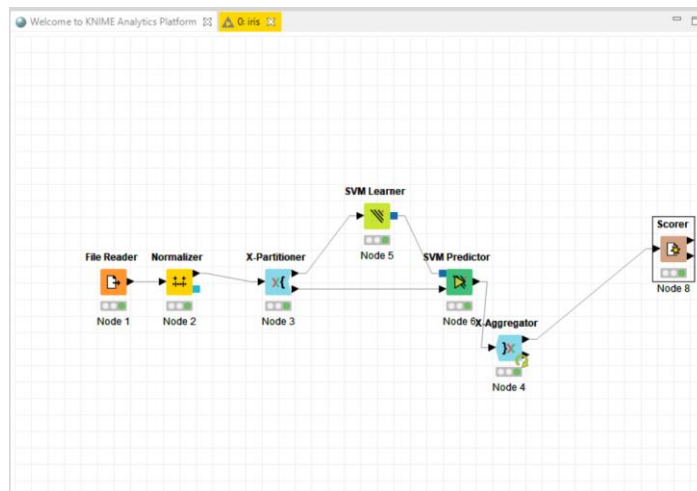
Gambar 3.2. Knime Explorer

Kemudian terdapat Node Repository, bagian ini merupakan bagian yang sangat penting, karena berisi seluruh fungsi tools dari software ini yang dinamakan dengan **Node**.



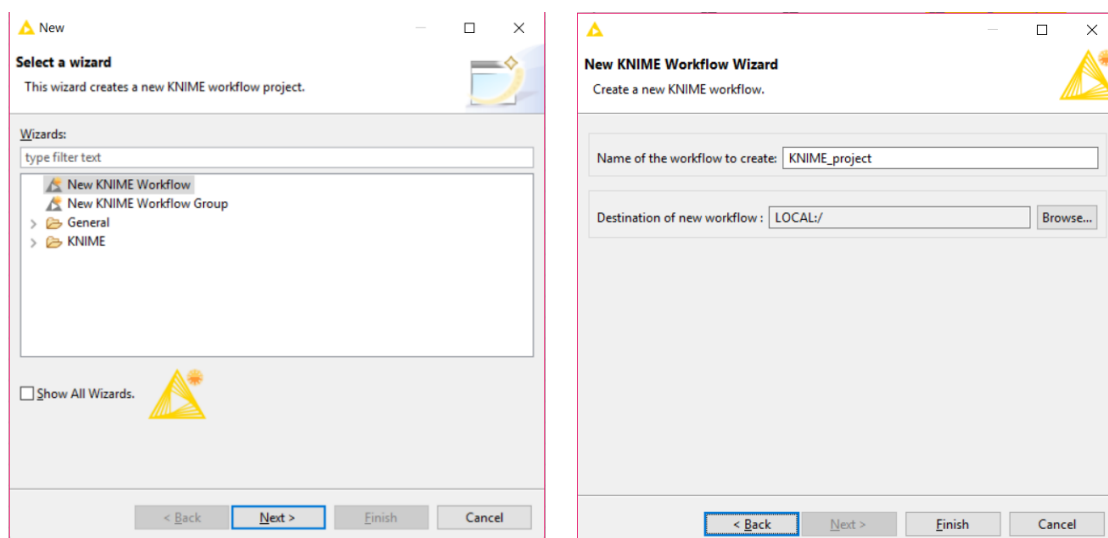
Gambar 3.3. Node Repository

Terakhir yaitu Knime Workflow, bagian ini adalah bagian visual pada Knime, seluruh fungsi yang digunakan akan ditampilkan pada bagian ini. Berikut adalah contoh tampilan pada Knime Workflow



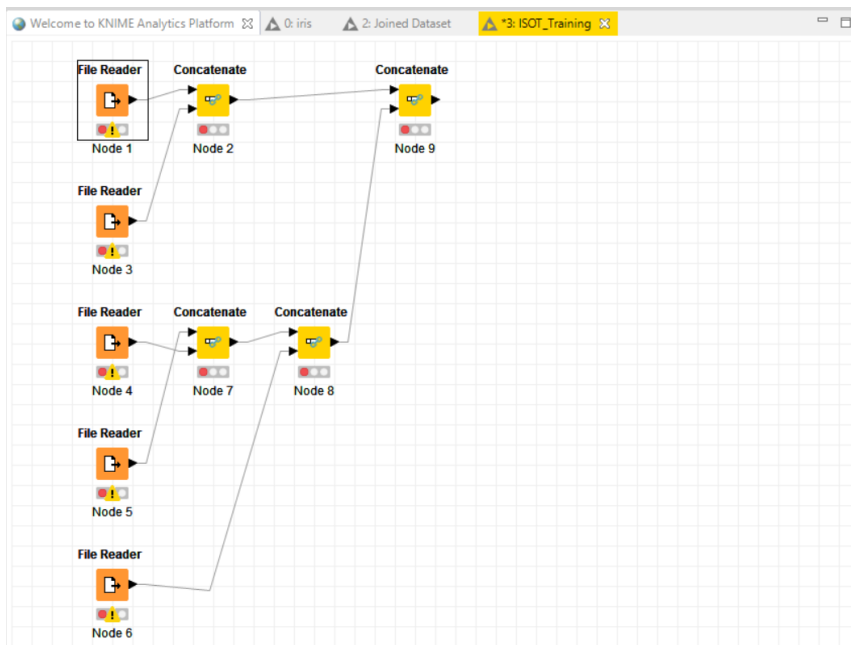
Gambar 3.4. Contoh Knime Workfloe

3. Setelah mengenal semua bagian dari Knime. Setelah itu kita akan membuat workflow/project baru. Dengan cara klik File – New – New Knime Workflow – Tulis Nama workflow dan Lokasi workflow tersebut – Klik Finish



Gambar 3.5. Membuat Workflow Baru

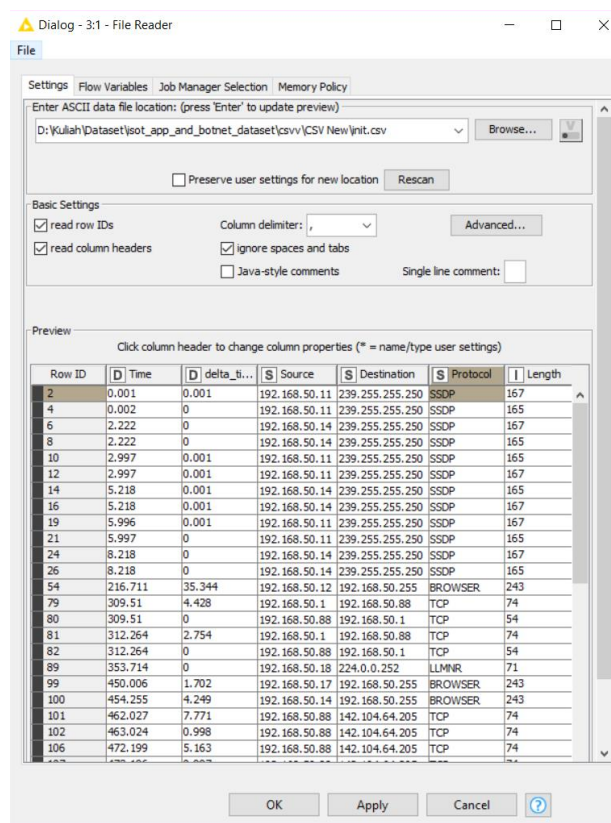
4. Selanjutnya yaitu menggabungkan seluruh data tadi menjadi 1 data. Node yang dibutuhkan untuk proses ini yaitu :
 - a. File Reader : untuk membaca data
 - b. Concatenate : untuk menggabungkan data



Gambar 3.6. Workflow Untuk Menggabungkan Data

Karena node Concatenate hanya dapat menerima input dari 2 data, maka diperlukan lebih dari 1 node Concatenate.

Untuk melihat konfigurasi dari File Reader, dapat digunakan cara klik kanan pada Node, lalu configure



Gambar 3.7. Konfigurasi File Reader

Untuk konfigurasi file reader hanya tinggal memasukkan file csv yang telah diexport pada langkah sebelumnya. Klik Apply – OK. Proses ini belum selesai, karena Node belum di jalankan, untuk

menjalankan Node bisa dengan cara klik kanan pada Node – Execute. Bila berhasil dijalankan, status Node yang berada dibawah Node akan berubah berwarna Hijau.

Concatenated table - 0:9 - Concatenate

File Hilite Navigation View

Table "default" - Rows: 4187432 Spec - Columns: 8 Properties Flow Variables

Row ID	D Time	D Delta_time	D Delta_time_disp	S Source	S Destinat...	S Protocol	I Length	S Info
79	309.51	4.428	0	192.168.50.1	192.168.50.88	TCP	74	48221 > 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=
80	309.51	0	0	192.168.50.88	192.168.50.1	TCP	54	22 > 48221 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
81	312.264	2.754	2.754	192.168.50.1	192.168.50.88	TCP	74	48222 > 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=

Gambar 3.8. Hasil dari File Reader

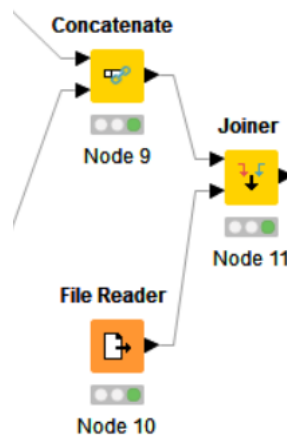
B. Pelabelan Data

- Selanjutnya yaitu proses melabeli data. Pada data ISOT terdapat 2 tipe data, yaitu malicious dan normal. Pertama kita harus menyediakan tabel dengan label **malicious** yang sudah ditentukan seperti berikut.

S Source	S label
192.168.50.14	malicious
192.168.50.15	malicious
192.168.50.16	malicious
192.168.50.17	malicious
192.168.50.18	malicious
192.168.50.30	malicious
192.168.50.31	malicious
192.168.50.32	malicious
192.168.50.34	malicious

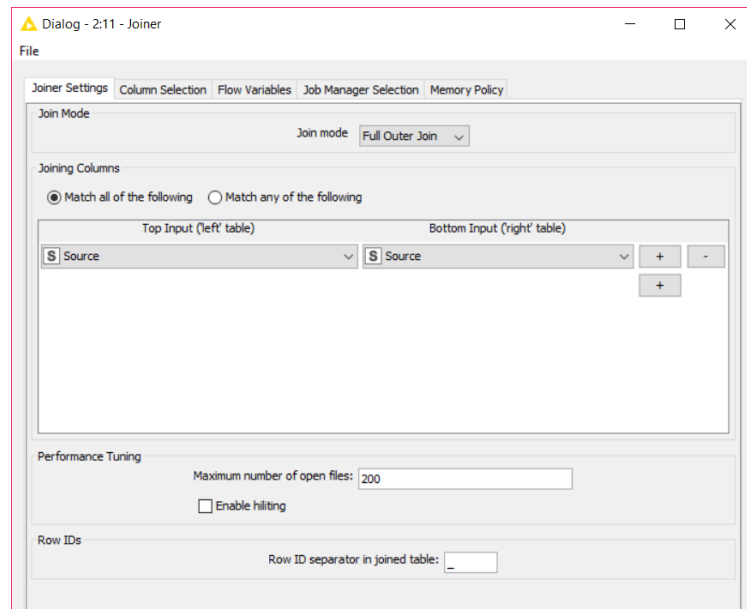
Gambar 3.9. Tabel dengan label Malicious

- Kemudian selanjutnya akan menggabungkan 2 tabel menggunakan node **Joiner**. Node ini membutuhkan 2 input data.



Gambar 3.10. Flow untuk proses penggabungan data

- Untuk konfigurasi Joiner, dapat menggunakan konfigurasi berikut



Gambar 3.11. Konfigurasi Node Joiner

Lalu klik OK dan jalankan. Hasilnya akan seperti berikut.

Joined table - 2:11 - Joiner

File Hilite Navigation View

Table "default" - Rows: 4187432										
Spec - Columns: 9			Properties							
Row ID	D Time	D Delta_ti...	D Delta_ti...	S Source	S Destinat...	S Protocol	I Length	S Info	S label	
27449_Row0	2,415.17	0	1,400.765	192.168.50.14	192.168.50.88	DNS	84	Standard query 0x862c A fe2.update.microsoft.com	malicious	
27826_Row0	3,571.662	0	242.671	192.168.50.14	192.168.50.88	DNS	89	Standard query 0x5fe3 A ds.download.windowsup...	malicious	
28885_Row1	8,095.089	0	2.558	192.168.50.15	192.168.50.88	DNS	76	Standard query 0x7728 A blue.botnet.isot	malicious	

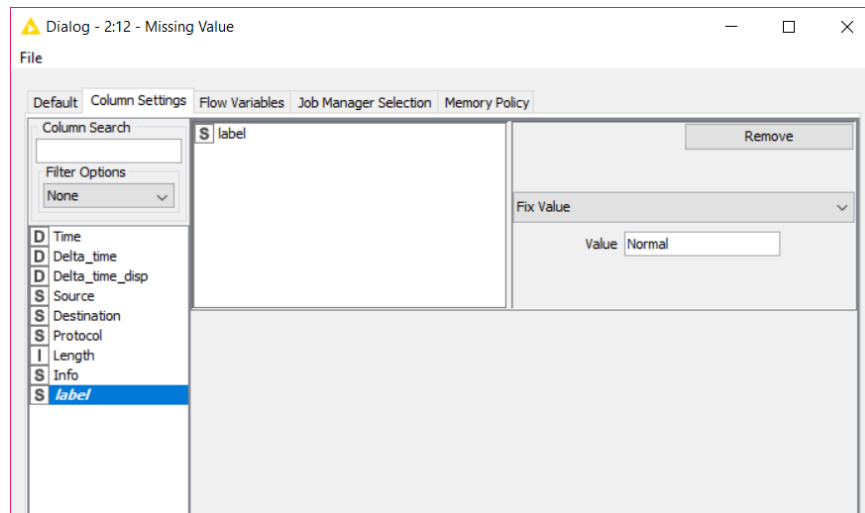
Gambar 3.12. Hasil dari menggabungkan Tabel menggunakan node Joiner

4. Dari langkah 7 menghasilkan output data dengan label malicious tetapi masih terdapat data dengan berlabel "?". Hal ini dikarenakan kita hanya labeling untuk data malicious saja. Untuk melakukan labeling data normal kita akan menggunakan Node **Missing Value**. Node ini digunakan untuk mengisi data kosong.



Gambar 3.13. Node Missing Value

Untuk konfigurasi, dapat menggunakan konfigurasi berikut.



Gambar 3.14. Konfigurasi pada Node Missing Value

Konfigurasi ini nantinya akan mengisi value yang kosong dengan value Normal. Berikut hasil dari proses Missing Value

Output table - 2:12 - Missing Value

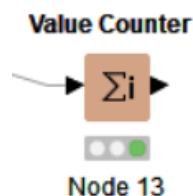
File Hilite Navigation View

Table "default" - Rows: 4187432 Spec - Columns: 9 Properties Flow Variables

Row ID	D Time	D Delta_ti...	D Delta_ti...	S Source	S Destinat...	S Protocol	I Length	S Info	S label
210450_?	509,783.403	0	0	173.254.28.55	192.168.50.88	TCP	86	22 > 49684 [ACK] Seq=21206 Ack=50877378 Wi...	Normal
210451_?	509,783.403	0	0	192.168.50.88	173.254.28.55	SSHv2	42058	Client: Encrypted packet (len=41992)	Normal
210452_?	509,783.403	0	0	173.254.28.55	192.168.50.88	SSHv2	106	Server: Encrypted packet (len=40)	Normal
210453_?	509,783.403	0	0	173.254.28.55	192.168.50.88	TCP	66	22 > 49684 [ACK] Seq=21246 Ack=50896202 Wi...	Normal

Gambar 3.15. Hasil dari proses Missing Value

- Untuk memastikan bahwa kolom label sudah terisi dengan value Malicious atau Normal, dapat menggunakan node **Value Counter**. Node ini berfungsi untuk menghitung jumlah seluruh value pada kolom terpilih.



Gambar 3.16. Node Value Counter

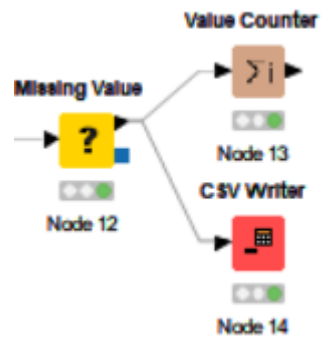
Berikut adalah hasil dari perhitungan value dengan Value Counter. Dalam konfigurasinya tinggal memilih kolom yang ingin dihitung yaitu kolom label.

Row ID	I count
Normal	3135405
malicious	1052027

Gambar 3.17. Hasil dari Value Counter

Dari gambar tersebut bisa dilihat jika sudah tidak ada data yang memiliki label kosong, hanya terdapat 2 label yaitu Normal dan malicious

6. Export file ke dalam format .csv dengan menggunakan node **CSV Writer**



Gambar 3.18. Flow untuk membuat data baru

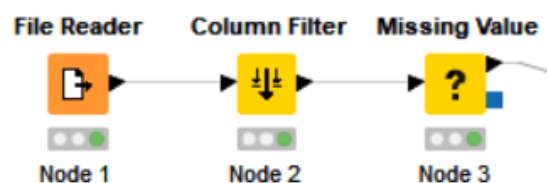
4. Knime For Data Mining

Pada langkah ini, kita akan melakukan sebuah analisa pada data ISOT dengan menggunakan platform Knime. Pada proses sebelumnya kita sudah melakukan penggabungan data dan pelabelan data. Pada proses ini kita akan melakukan analisa data dengan menggunakan Teknik **Data Mining**. Data mining adalah proses penggalian data atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sebuah data. Di dalam Data Mining terdapat beberapa proses yang harus dilakukan terlebih dahulu sebelum dilakukannya data mining. Berikut tahapan-tahapan untuk melakukan data mining, yaitu

1. Data Pre-Processing
2. Data Transformation
3. Data Mining
4. Evaluation

A. Data Pre-processing

1. Proses ini adalah proses dimana data akan dibersihkan (cleaning) karena biasanya didalam suatu data terdapat nilai-nilai yang tidak sempurna atau bahkan terdapat nilai-nilai yang hilang atau kosong yang nantinya akan dapat mempengaruhi proses kedepannya. Pada proses ini kita membutuhkan Node-node berikut : File Reader, Column Filter, Missing Value.



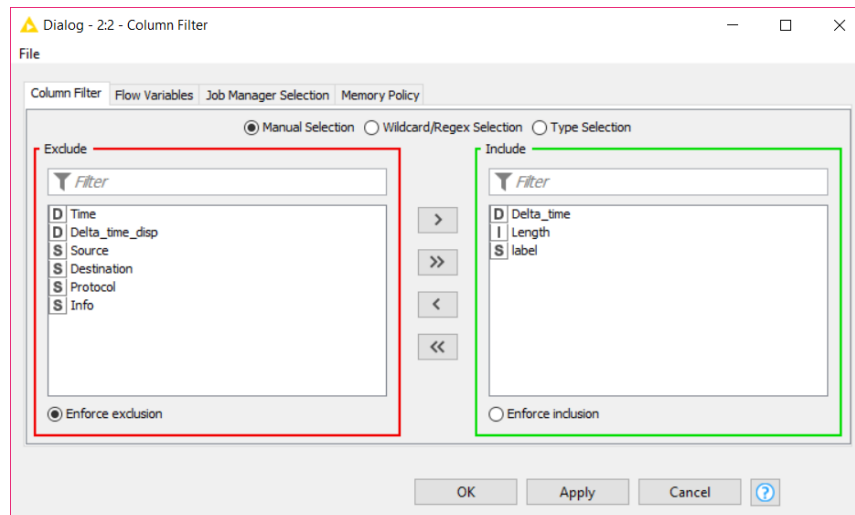
Gambar 4.1. Pre-Processing Process

2. Sebelum memulai data pre-processing, langkah pertama yaitu membaca file data csv yang sudah diexport pada proses sebelumnya menggunakan Node File Reader. Untuk konfigurasiya cari lokasi file data yang diexport tadi.

Row ID	D Time	D Delta_t...	D Delta_t...	S Source	S Destin...	S Protocol	I Length	S Info	S label
Row0	2,415.17	0	1,400.765	192.168.50.14	192.168.50.88	DNS	84	Standard query 0x862c A fe2.update.microsoft.com	malicious
Row1	3,571.662	0	242.671	192.168.50.14	192.168.50.88	DNS	89	Standard query 0x5fe3 A ds.download.windowsup...	malicious
Row2	8,095.089	0	2.558	192.168.50.15	192.168.50.88	DNS	76	Standard query 0x7728 A blue.botnet.isot	malicious
Row3	8,602.268	0	492.736	192.168.50.14	192.168.50.88	DNS	84	Standard query 0x8c7b A fe2.update.microsoft.com	malicious

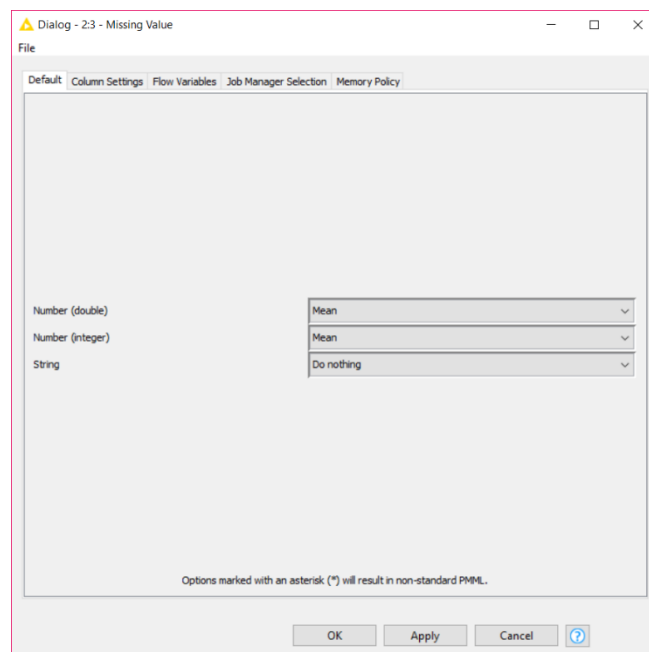
Gambar 4.2. Membaca data menggunakan node File Reader

3. Selanjutnya kita akan menggunakan Node **Column Filter**. Node ini berfungsi untuk mem-filter kolom atau atribut yang tidak digunakan. Disini kita hanya menggunakan 3 atribut, yaitu : Delta_time, length dan label.



Gambar 4.3. Konfigurasi node Column Filter

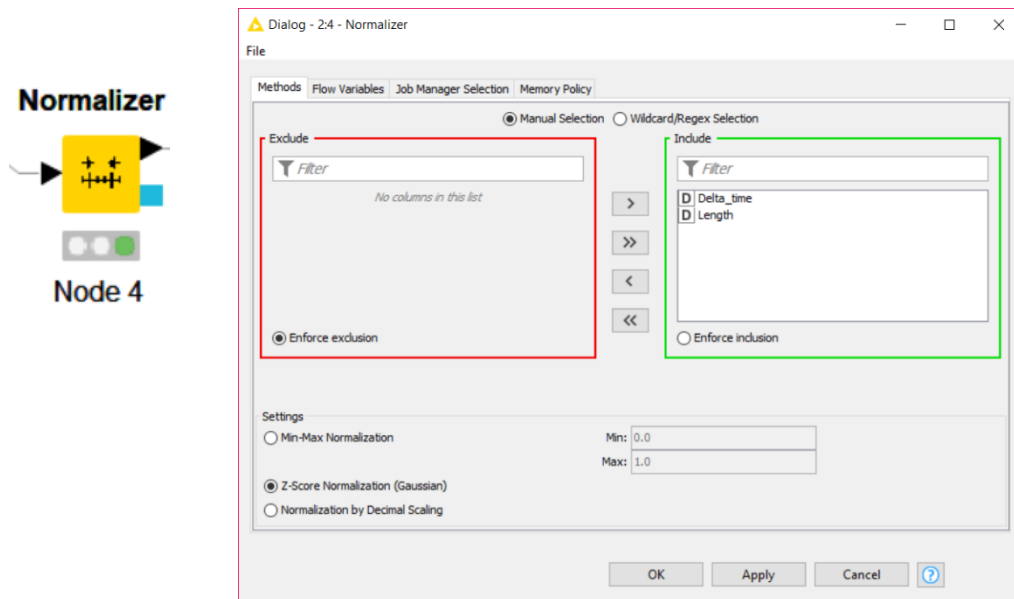
4. Setelah langkah 2 dijalankan, data tersebut menjadi memiliki 3 kolom atau atribut yang sebelumnya terdapat 9 kolom. Selanjutnya kita akan menjalankan Node **Missing Value**. Node ini sudah pernah kita pakai pada proses labeling data. Tetapi pada proses ini kita akan melakukan pembersihan data, karena biasanya didalam suatu data terdapat kolom yang tidak sempurna seperti data yang hilang atau atribut yang tidak relevan, untuk itu Node ini diperlukan untuk mengatasi hal tersebut. Berikut konfigurasinya



Gambar 4.4. Konfigurasi node Missing Value

B. Data Transformation

1. Setelah melakukan data pre-processing, selanjutnya akan menuju ke proses data transformation, pada proses ini data akan diubah ke format yang sesuai untuk proses data mining. Node yang digunakan pada tahap ini yaitu **Normalizer**. Berikut konfigurasiya



Gambar 4.5. Node Normalizer dan Konfigurasiya

2. Setelah dijalankan, kolom dari data tersebut akan berubah menjadi bentuk range. Data inilah yang nantinya akan digunakan dalam pengenalan pola. Berikut adalah hasil dari Node Normalizer

Normalized table - 2:4 - Normalizer

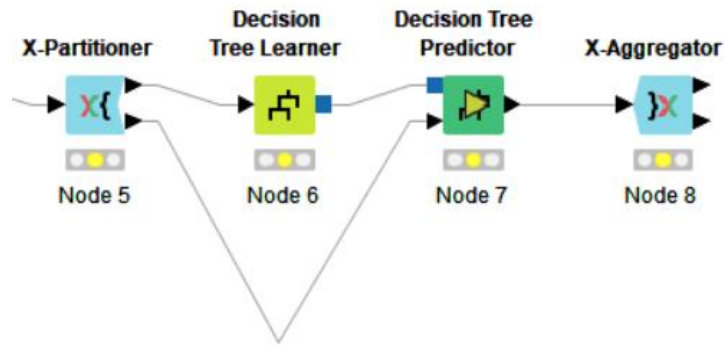
Row ID	D Delta_time	D Length	S label
Row2004603	0.172	-0.087	Normal
Row2004604	0.411	-0.087	Normal
Row2004605	-0.265	-0.08	Normal
Row2004606	1.377	-0.097	Normal
Row2004607	0.697	-0.087	Normal
Row2004608	-0.265	-0.093	Normal
Row2004609	-0.265	-0.086	Normal
Row2004610	-0.266	-0.097	Normal
Row2004611	-0.265	-0.094	Normal
Row2004612	1.068	-0.093	Normal
Row2004613	0.388	-0.085	Normal
Row2004614	-0.266	-0.085	Normal

Gambar 4.6. Hasil dari proses Normalizer

Setelah mendapatkan data ini, baru kita dapat menjalankan proses Data Mining

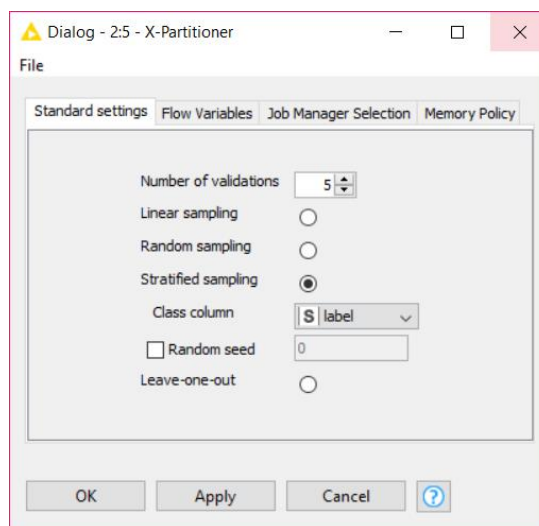
C. Data Mining

1. Setelah menyelesaikan tahap data transformation, kita akan menjalankan proses Data Mining, dalam proses ini kita akan menggunakan Metode Klasifikasi **Decision Tree** dengan teknik **Cross Validation**. Pada proses ini kita membutuhkan Node-node berikut : X-Partitioner, Decision Tree Learner, Decision Tree Predictor, X-Aggregator
Sehingga akan membentuk flow seperti ini



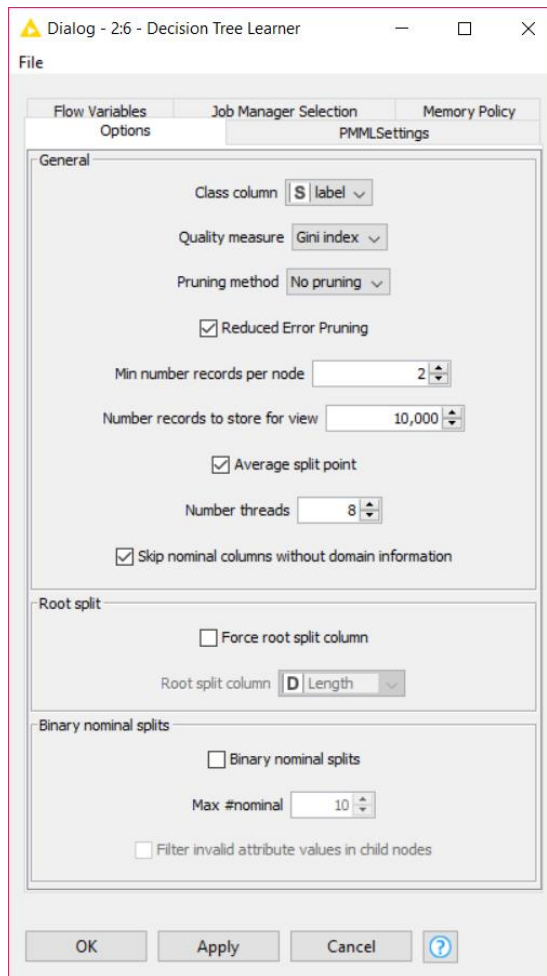
Gambar 4.7. Alur proses dari data mining

2. X-Partitioner berfungsi untuk menentukan jumlah iterasi atau pengulangan pada teknik cross validation, data ini nantinya akan terbagi menjadi 2 yaitu data training dan data testing. Berikut konfigurasi node X-Partitioner



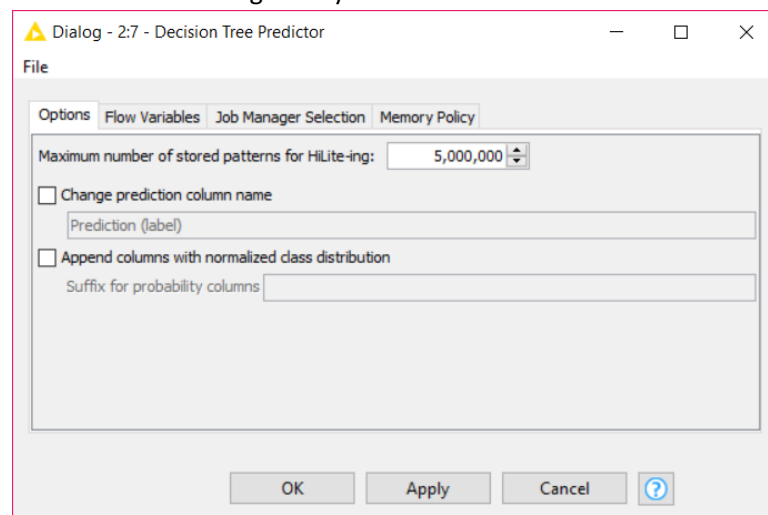
Gambar 4.8. Konfigurasi node X-Partitioner

3. Decision Tree Learner berfungsi sebagai data training, karena metode Decision Tree merupakan supervised learning, sehingga membutuhkan data training untuk mengenali pola dari setiap data. Berikut konfigurasi dari Decision Tree Learner



Gambar 4.9. Konfigurasi node Decision Tree Learner

4. Setelah menjalankan Decision Tree Learner, lalu dilanjutkan dengan Decision Tree Predictor. Node ini berfungsi untuk mengklasifikasi data dengan cara menguji data testing dengan hasil dari proses Decision Tree Learner. Berikut konfigurasinya



Gambar 4.10. Konfigurasi dari Node Decision Tree Predictor

- Node X-Aggregator berfungsi sebagai akhir dari proses cross validation. Node ini akan mengumpulkan hasil dari Node Predictor yang akan menampilkan hasil dari prediksi dari beberapa iterasi yang dilakukan. Tidak ada konfigurasi khusus dari node ini, sehingga bisa langsung dijalankan. Berikut adalah hasil dari node X-Aggregator. Dari hasil ini akan mendapatkan kolom baru yaitu kolom prediksi.

▲ Prediction table - 2:8 - X-Aggregator

File Hilite Navigation View

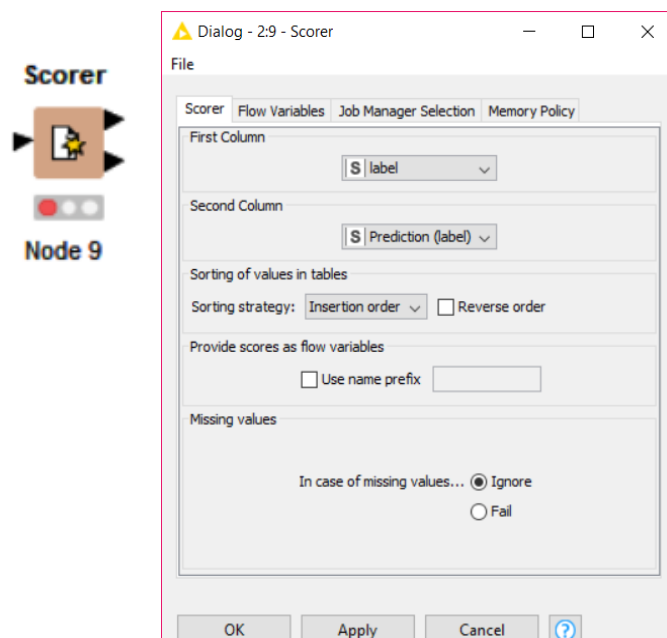
Table "default" - Rows: 4187432 Spec - Columns: 4 Properties Flow Variables

Row ID	D Delta_ti...	D Length	S label	S Prediction (label)
Row0	-0.265	-0.088	malicious	Normal
Row1	-0.265	-0.085	malicious	Normal
Row2	-0.265	-0.093	malicious	malicious
Row4	-0.265	-0.088	malicious	Normal
Row6	-0.264	-0.088	malicious	Normal

Gambar 4.11. Konfigurasi node X-Aggregator


A. Evaluation

- Proses ini merupakan proses terakhir pada tahap data mining yaitu merupakan hasil dari teknik data mining berupa hasil prediksi untuk menilai apakah model ini dapat digunakan untuk mengenali pola serangan pada data ISOT. Untuk langkah ini kita akan menggunakan Node **Scorer** yang didalamnya terdapat perhitungan untuk melihat seberapa baik model ini dengan menggunakan teknik confusion matrix. Berikut konfigurasinya.



Gambar 4.12. Node Scorer dan Konfigurasi

- Setelah node ini dijalankan, kita dapat melihat presentase dari hasil confusion matrix. Hasil inilah yang nantinya akan digunakan untuk menentukan keputusan apakah model ini baik atau tidak dalam menangani kasus data ISOT Botnet untuk mendeteksi serangan.


Confusion Matrix - 2:9 - Scorer

—

□

×

File
Hilite

label \ Pre...	malicious	Normal	
malicious	793857	258170	
Normal	305016	2830389	

Correct classified: 3,624,246

Accuracy: 86.551 %

Cohen's kappa (κ) 0.648

Wrong classified: 563,186

Error: 13.449 %

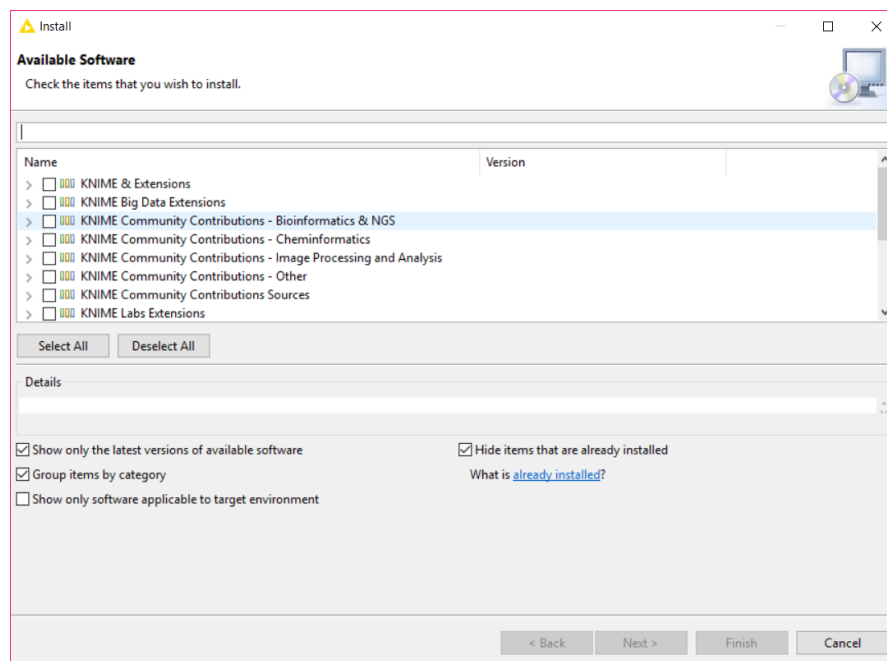
Gambar 4.13. Hasil dari Confusion Matrix

5. Knime For Visualisation Using Open Street Map

Pada bagian ini, kita akan membuat visualisasi serangan menggunakan Open Street Map. Sebelum memulai, terdapat beberapa ekstensi yang harus kita install terlebih dahulu untuk menjalankan beberapa node, yaitu :

1. KNIME OpenStreetMap extension
2. KNIME Web Analytics extension

Untuk mendapatkan ekstensi ini dapat dilakukan dengan cara klik File – Install Knime Extension

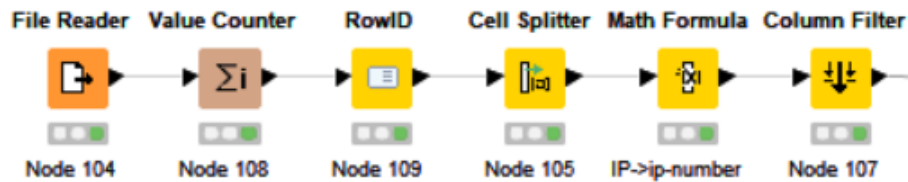


Gambar 5.1. Install Extension pada KNIME

Lalu kita tinggal ketik saja ekstensi yang dibutuhkan, lalu klik Next – klik Next lagi – Pilih Accept terms agreement – Finish. Setelah extension selesai di download dan di install Knime akan melakukan restart. Setelah itu ikut langkah-langkah berikut :

A. Load File IP Address

1. Pertama kita akan membutuhkan 2 tabel, tabel pertama yaitu file dataset kita, yang kedua yaitu file GeoIP database yang bisa didapatkan di link <http://dev.maxmind.com/geoip/> .
2. Pada tabel pertama, kita membutuhkan setidaknya 6 Node pada proses ini, yaitu : **File Reader**, **Value Counter**, **RowID**, **Cell Splitter**, **Math Formula** dan **Column Filter**. Output dari proses ini yaitu berupa data dengan kolom : Source, Count dan Source-number. Berikut gambaran workflow yang akan dibuat.



Gambar 5.2. Alur untuk membaca dan merubah data ISOT

- File Reader merupakan proses untuk membaca data yang ingin diproses. Untuk konfigurasinya hanya mencari file data yang telah tersimpan.

Row ID	Time	Delta	Source	Destination	Protocol	Length	Info	label
Row0	2,415.17	0	192.168.50.14	192.168.50.88	DNS	84	Standard query 0x862c A fe2.update.microsoft.com	malicious
Row1	3,571.662	0	192.168.50.14	192.168.50.88	DNS	89	Standard query 0x5fe3 A ds.download.windowsup...	malicious
Row2	8,095.089	0	192.168.50.15	192.168.50.88	DNS	76	Standard query 0x7728 A blue.botnet.isot	malicious
Row3	8,602.268	0	192.168.50.14	192.168.50.88	DNS	84	Standard query 0x8c7b A fe2.update.microsoft.com	malicious

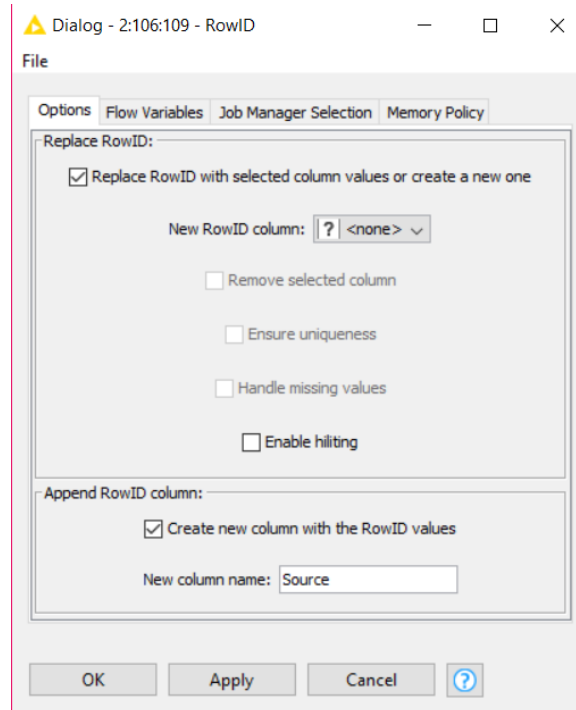
Gambar 5.3. Membaca data menggunakan node File Reader

- Jika file sudah berhasil terbaca, langkah selanjutnya yaitu menghitung jumlah IP pada data tersebut, karena kita hanya membutuhkan data IP (Source). Untuk konfigurasinya kita set "Columns with value to count" ke Source lalu jalankan.

Row ID	count
142.104.64.203	33228
173.254.28.55	13674
192.168.50.1	12
192.168.50.101	541

Gambar 5.4. Hasil perhitungan jumlah IP dengan node Value Counter

- Selanjutnya, karena proses Value Counter tadi merubah kolom RowID menjadi source untuk mengembalikannya kita membutuhkan node RowID dengan konfigurasi.



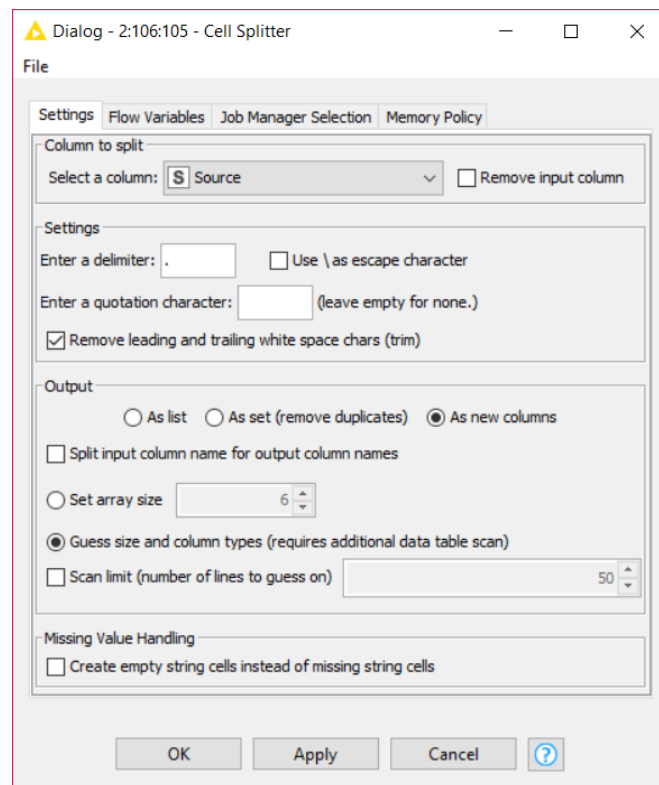
Gambar 5.5. Konfigurasi dari node RowID

Setelah dijalankan, hasil dari proses Node ini yaitu menggeser kolom RowID menjadi kolom baru.

Table "default" - Rows: 41			Spec - Columns: 2	Prop
Row ID	I count	S Source		
Row0	33228	142.104.64.203		
Row1	13674	173.254.28.55		
Row2	12	192.168.50.1		
Row3	541	192.168.50.101		
Row4	540	192.168.50.102		

Gambar 5.6. Hasil dari proses RowID

- Kemudian proses selanjutnya yaitu kita akan memecah IP address menjadi 4 bagian karena kita akan mengubah IP address menjadi IP number. Untuk memecah IP address kita membutuhkan node Cell Splitter dengan konfigurasi



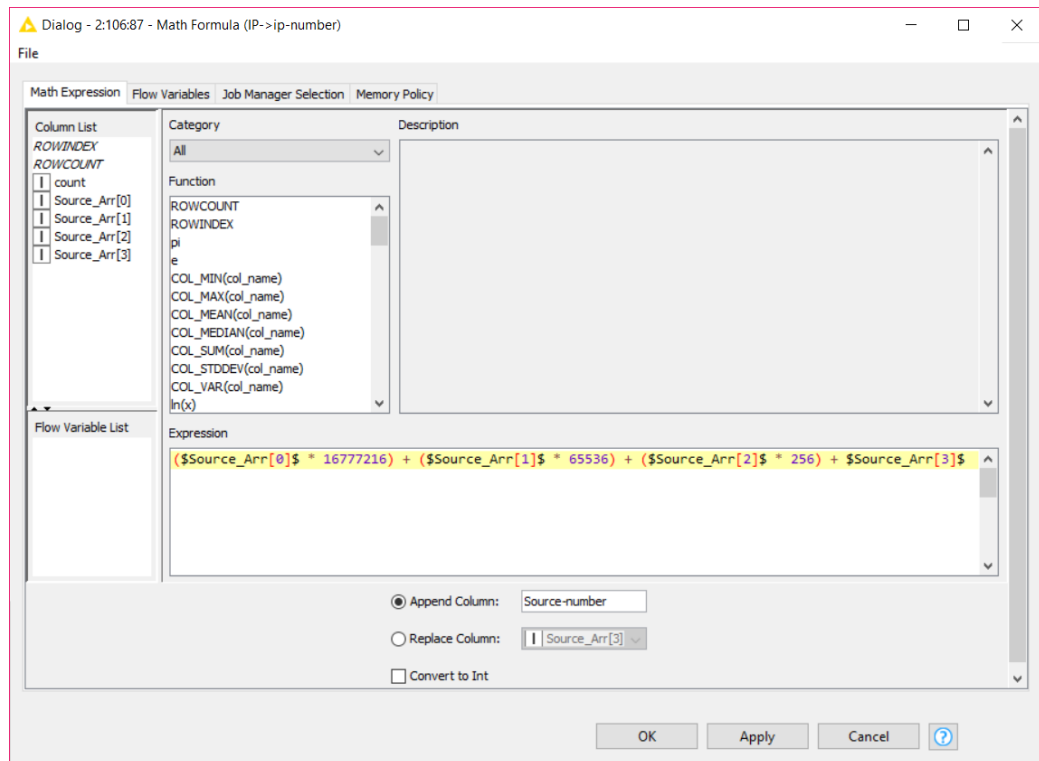
Gambar 5.7. Konfigurasi dari node Cell Splitter

Setelah node ini dijalankan, akan menghasilkan 4 tambahan kolom baru sebagai pecahan dari IP address

Table "default" - Rows: 41 Spec - Columns: 6 Properties Flow Variables						
Row ID	I count	S Source	I Source...	I Source...	I Source...	I Source...
Row0	33228	142.104.64.203	142	104	64	203
Row1	13674	173.254.28.55	173	254	28	55
Row2	12	192.168.50.1	192	168	50	1
Row3	541	192.168.50.101	192	168	50	101

Gambar 5.8. Hasil dari proses node Cell Splitter

7. Jika IP address sudah terpecah menjadi 4 bagian, selanjutnya kita akan mengubah IP address menjadi IP number dengan menggunakan Node **Math Formula** dengan konfigurasi



Gambar 5.9. Konfigurasi node Math Formula untuk mengubah IP addr menjadi IP number

Setelah node ini dijalankan, akan menghasilkan kolom baru yaitu Source-number yang berisi konversi dari IP address ke IP number.

Table "default" - Rows: 41								Spec - Columns: 7	Properties	Flow Variables
Row ID	I count	S Source	I Source...	I Source...	I Source...	I Source...	D Source-number			
Row0	33228	142.104.64.203	142	104	64	203	2,389,197,003			
Row1	13674	173.254.28.55	173	254	28	55	2,919,111,735			
Row2	12	192.168.50.1	192	168	50	1	3,232,248,321			

Gambar 5.10. Hasil Konversi

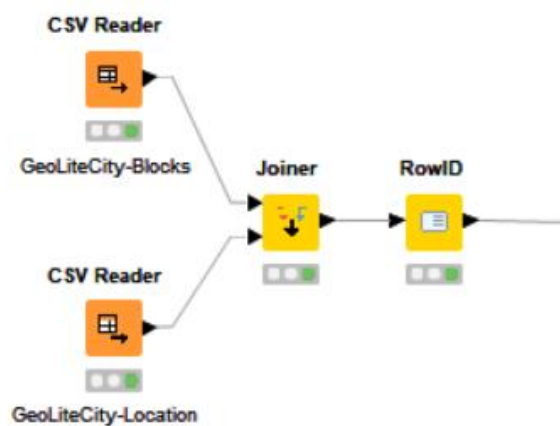
- Setelah berhasil merubah IP address menjadi IP number, selanjutnya kita akan menghilangkan kolom-kolom pecahan dari IP, karena kita sudah tidak membutuhkannya, caranya dengan menggunakan node **Column Filter**, kemudian kita exclude semua kecuali Count, Source dan Source-number

Table "default" - Rows: 41				Spec - Columns: 3	Properties	Flow Variables
Row ID	I count	S Source	D Source-number			
Row0	33228	142.104.64.203	2,389,197,003			
Row1	13674	173.254.28.55	2,919,111,735			
Row2	12	192.168.50.1	3,232,248,321			

Gambar 5.11. Hasil dari Mem-Filter Kolom

B. Load GeoIP Database

1. Setelah kita mendownload data GeoIP database, kita akan mendapatkan dua tipe data, yaitu GeoLiteCity-Blocks dan GeoLiteCity-Location. Dua file ini kita load dengan 2 node Reader, 1 node Joiner, dan 1 Node RowID



Gambar 5.12. Alur dari membaca data GeoIP

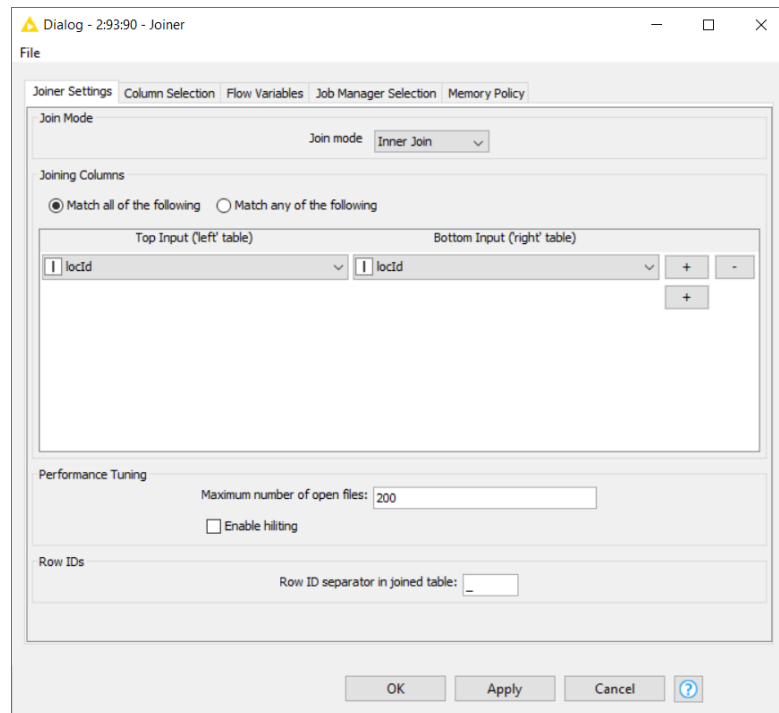
Table "GeoLiteCity-Blocks.csv" - Rows: 1893989				Spec - Columns: 3	Property
Row ID	D startIpNum	D endIpNum	I locId		
Row0	16,777,216	16,777,471	17		
Row1	16,777,472	16,778,239	49		
Row2	16,778,240	16,778,495	14409		

Gambar 5.13. Isi dari File GeoLiteCity-Blocks.csv

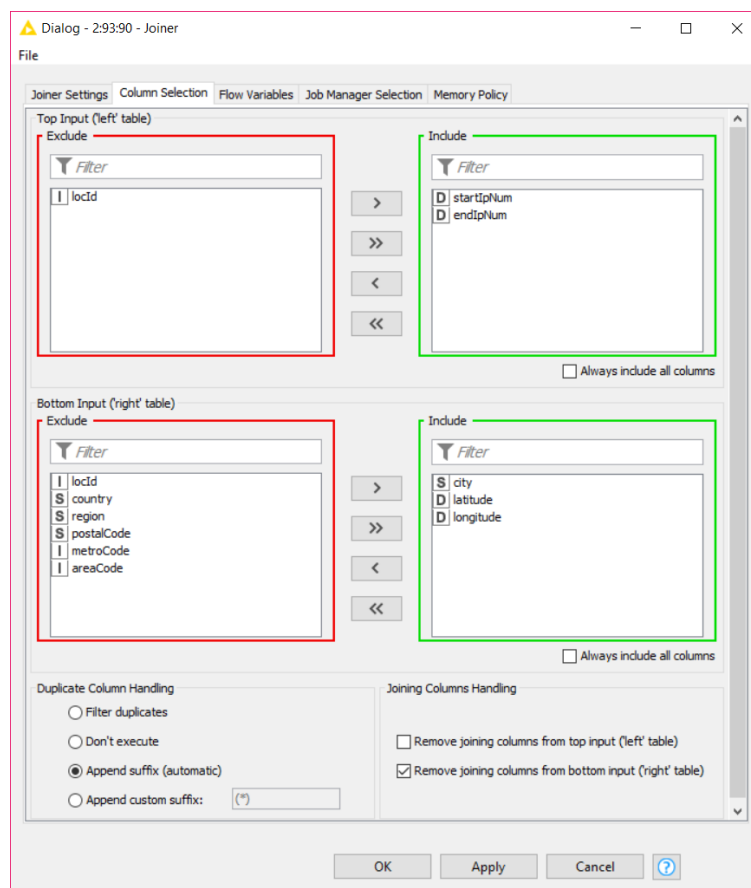
Table "GeoLiteCity-Location.csv" - Rows: 502474											Spec - Columns: 9	Properties	Flow Variables
Row ID	I locId	S country	S region	S city	S postalCode	D latitude	D longitude	I metroCode	I areaCode				
Row0	1	O1				0	0	?	?				
Row1	2	AP				35	105	?	?				

Gambar 5.14. Isi dari File GeoLiteCity-Location.csv

2. Selanjutnya kita akan menggabungkan dua file tersebut menjadi satu menggunakan node **Joiner**. Dari 2 file tersebut kita akan mengambil 5 kolom, yaitu : startIpNum, endIpNum, city, latitude dan longitude.



Gambar 5.15. Konfigurasi dari node Joiner untuk menggabungkan 2 tabel



Gambar 5.16. Konfigurasi dari node Joiner untuk menggabungkan 2 tabel

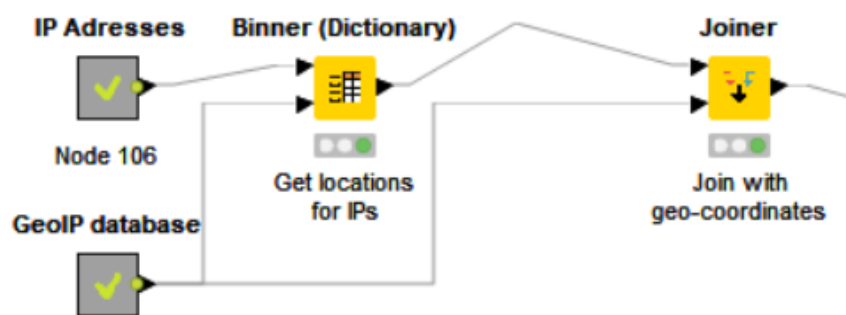
- Setelah data berhasil di jadikan satu dengan **Joiner** terdapat perubahan pada kolom rowID menjadi hasil join rowID dari kedua tabel. Untuk mengatasi ini bisa menggunakan node **RowID** untuk mengembalikan rowID seperti semula

Table "default" - Rows: 1893989 Spec - Columns: 5 Properties Flow Variables					
Row ID	D startIp...	D endIpNum	S city	D latitude	D longitude
Row0_Row16	16,777,216	16,777,471		-27	133
Row1_Row48	16,777,472	16,778,239		35	105

Gambar 5.17. Hasil RowID yang mengalami perubahan

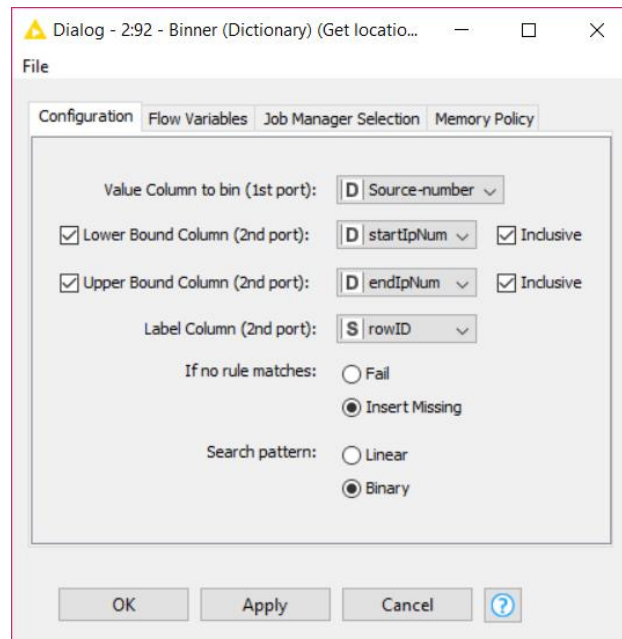
C. Get Location IP

- Setelah dua tabel tersebut sudah di load. Langkah selanjutnya yaitu mendapatkan lokasi IP dari dataset kita yaitu ISOT. Pada langkah ini akan dibutuhkan 2 node yaitu : Binner dan Joiner.



Gambar 5.18. Alur proses untuk mendapatkan Location pada dataset ISOT

- Node Binner ini berfungsi untuk meng-kategorikan data berdasarkan tabel rule/dictionary. Pada kasus ini tabel rule adalah tabel GeoIP database. Peng-kategorian ini berdasarkan IP number atau IP address yang tadi sudah diubah ke format number



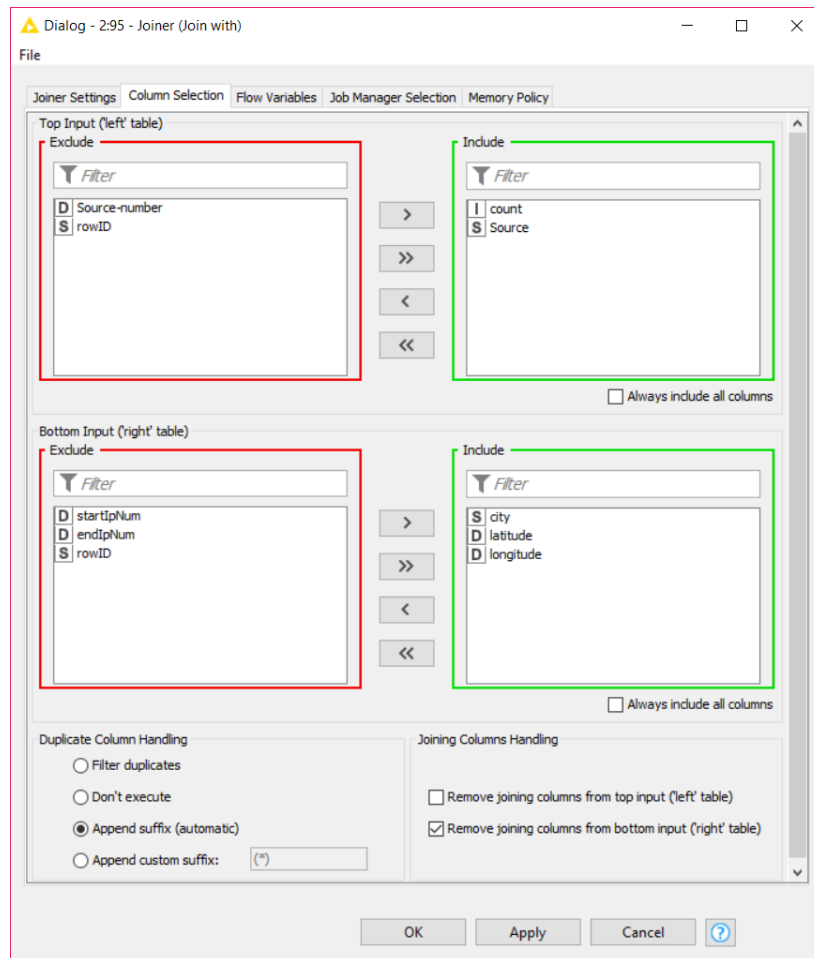
Gambar 5.19. Konfigurasi dari node Binner

Setelah dijalankan, akan menghasilkan data yang telah dikategorikan, kategori ini akan tersimpan pada kolom RowID.

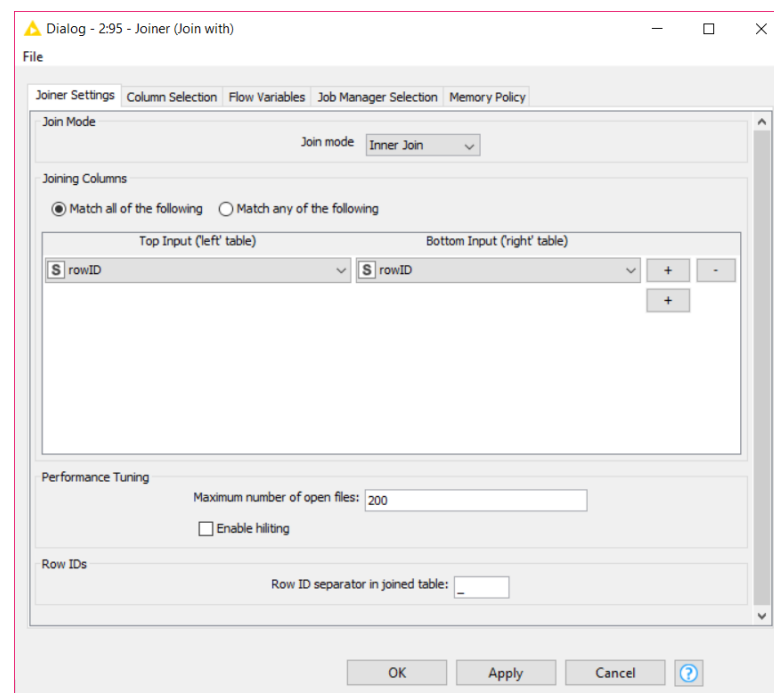
Table "default" - Rows: 41 Spec - Columns: 4 Properties Flow Variables				
Row ID	I count	S Source	D Source-number	S rowID
Row0	33228	142.104.64.203	2,389,197,003	Row1356136_Row371602
Row1	13674	173.254.28.55	2,919,111,735	Row1451119_Row3421
Row2	12	192.168.50.1	3,232,248,321	?
Row3	541	192.168.50.101	3,232,248,421	?

Gambar 5.20. Hasil dari proses node Binner, data sudah dikategorikan

- Setelah pengelompokan kategori, kita akan membutuhkan beberapa kolom yang akan digunakan untuk visualisasi, yaitu : count, source, city, latitude dan longitude. Untuk mendapatkannya kita dapat menggunakan node **Joiner** dengan input dari node **Binner** dan GeoIP Database file kemudian di Join berdasarkan RowID.



Gambar 5.21. Konfigurasi node Joiner untuk menampilkan latitude longitude



Gambar 5.22. Konfigurasi node Joiner untuk menampilkan latitude longitude

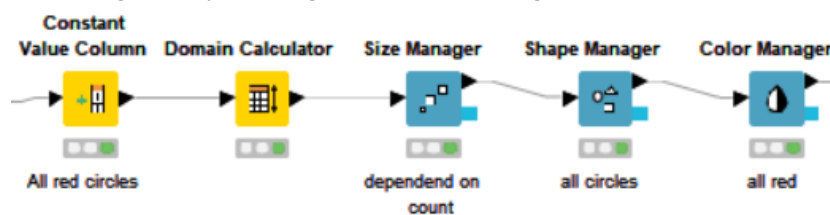
Hasil dari proses ini yaitu data sudah mendapatkan latitude, longitude dan city.

Table "default" - Rows: 15 Spec - Columns: 5 Properties Flow Variables					
Row ID	I count	S Source	S city	D latitude	D longitude
Row0_Row13...	33228	142.104.64.203	Victoria	48.42	-123.367
Row1_Row14...	13674	173.254.28.55	Provo	40.218	-111.613
Row28_Row1...	5	192.35.51.30		38	-97

Gambar 5.23. Data setelah di gabungkan.

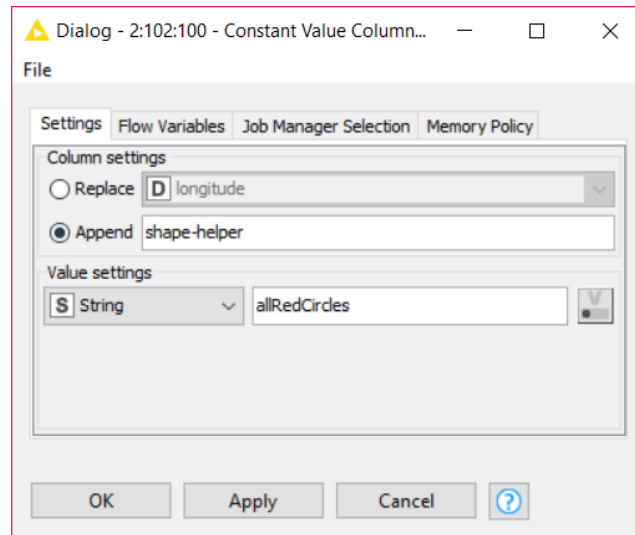
D. Map Marker

- Proses ini digunakan untuk sebagai tanda lokasi terjadinya serangan berdasarkan longitude dan latitude. Pada proses ini dibutuhkan setidaknya 5 node, yaitu : Constant Value Column, Domain Calculator, Size Manager, Shape Manager dan Color Manager.



Gambar 5.24. Alur dari proses pembuatan marker/tanda pada map

- Constant Value Column ini digunakan untuk menambahkan/mengganti kolom yang berisi nilai konstan di setiap baris. Pada proses ini kita akan menambahkan kolom bernama shape-helper dengan value allRedCircles di semua baris.



Gambar 5.25. Konfigurasi dari node Constant Value Column untuk menambahkan kolom

Output table - 2:102:100 - Constant Value Column (All red circles)

File Hilite Navigation View

Table "default" - Rows: 15 Spec - Columns: 6 Properties Flow Variables

Row ID	I count	S Source	S city	D latitude	D longitude	S shape-helper
Row0_Row13...	33228	142.104.64.203	Victoria	48.42	-123.367	allRedCircles
Row1_Row14...	13674	173.254.28.55	Provo	40.218	-111.613	allRedCircles
Row28_Row1...	5	192.35.51.30		38	-97	allRedCircles
Row29_Row1...	5	192.42.93.30		38	-97	allRedCircles
Row30_Row1...	6	192.58.128.30		38	-97	allRedCircles
Row31_Row1...	1	199.7.91.13	College Park	38.834	-76.878	allRedCircles
Row32_Row3...	140377	8.8.4.4		38	-97	allRedCircles
Row33_Row3...	160998	8.8.8.8		38	-97	allRedCircles
Row34_Row1...	1189	91.189.88.149		51.5	-0.13	allRedCircles
Row35_Row1...	1415	91.189.88.152		51.5	-0.13	allRedCircles
Row36_Row1...	136	91.189.88.161		51.5	-0.13	allRedCircles
Row37_Row1...	341	91.189.88.162		51.5	-0.13	allRedCircles
Row38_Row1...	6779	91.189.91.23	Boston	42.358	-71.06	allRedCircles
Row39_Row1...	9249	91.189.91.26	Boston	42.358	-71.06	allRedCircles
Row40_Row1...	10	91.189.95.15		51.5	-0.13	allRedCircles

Gambar 5.26. Hasil dari penambahan kolom menggunakan node Constant Value Column

- Kemudian selanjutnya node **Domain Calculator** ini berfungsi untuk memindai data dan memperbarui daftar nilai yang mungkin dan / atau nilai minimum dan maksimum kolom yang dipilih. Node ini yang nantinya akan mempengaruhi data apa saja yang akan di tampilkan pada Peta.

Dialog - 2:102:99 - Domain Calculator

File

Possible Values Min & Max Values Flow Variables Job Manager Selection Memory Policy

☒ Manual Selection ☐ Wildcard/Regex Selection ☐ Type Selection

Exclude

Filter

No columns in this list

☐ Enforce exclusion

Columns in exclude list:

☒ Retain Min/Max Domain ☐ Drop Min/Max Domain

Include

Filter

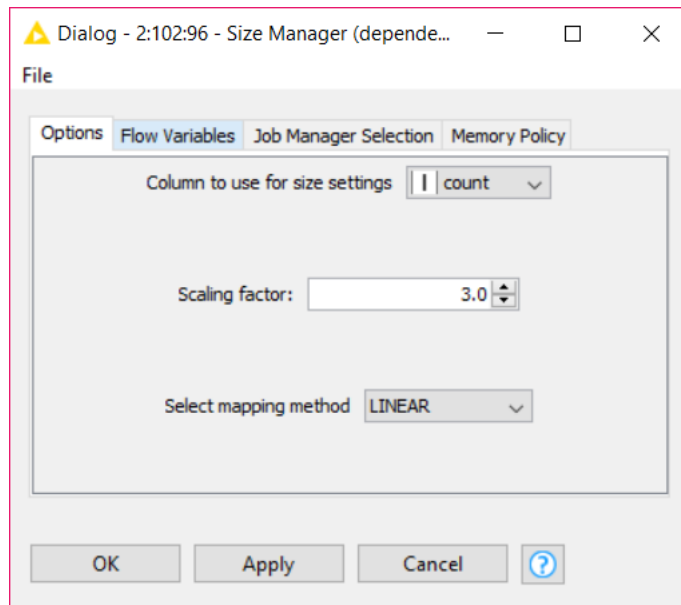
I count
S Source
S city
D latitude
D longitude
S shape-helper

☒ Enforce inclusion

OK Apply Cancel ?

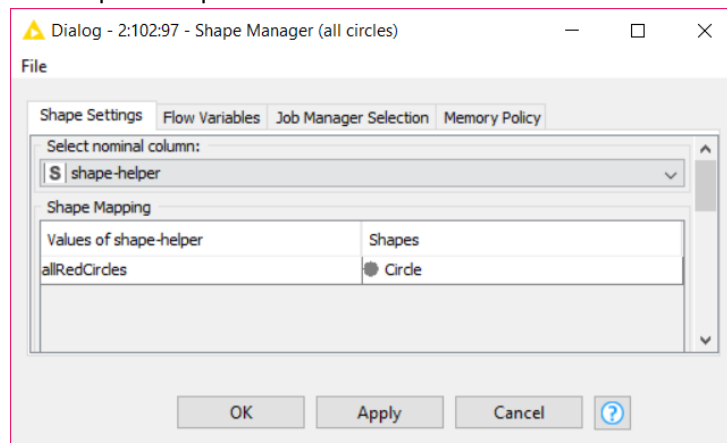
Gambar 5.27. Konfigurasi dari node Domain Calculator

- Selanjutnya yaitu Size Manager, node ini digunakan untuk mengatur ukuran besar tidaknya marker yang ada di peta berdasarkan jumlah dari aktifitas IP tersebut.



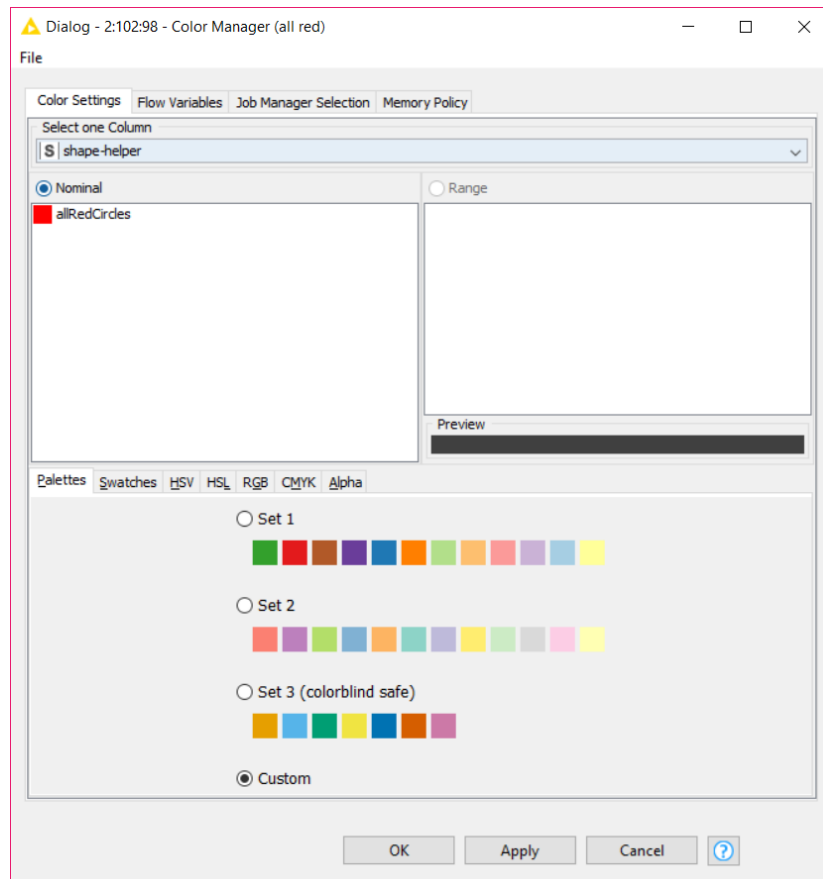
Gambar 5.28. Konfigurasi dari node Size Manager untuk menentukan skala size marker

5. Kemudian setelah itu terdapat node **Shape Manager** yang berfungsi untuk menentukan bentuk Marker yang akan ditampilkan di peta.



Gambar 5.29. Konfigurasi untuk menentukan bentuk marker pada peta

6. Terakhir yaitu terdapat node **Color Manager** yang berfungsi untuk mengatur warna marker yang akan ditampilkan di peta.



Gambar 5.30. Konfigurasi untuk menentukan warna marker pada peta

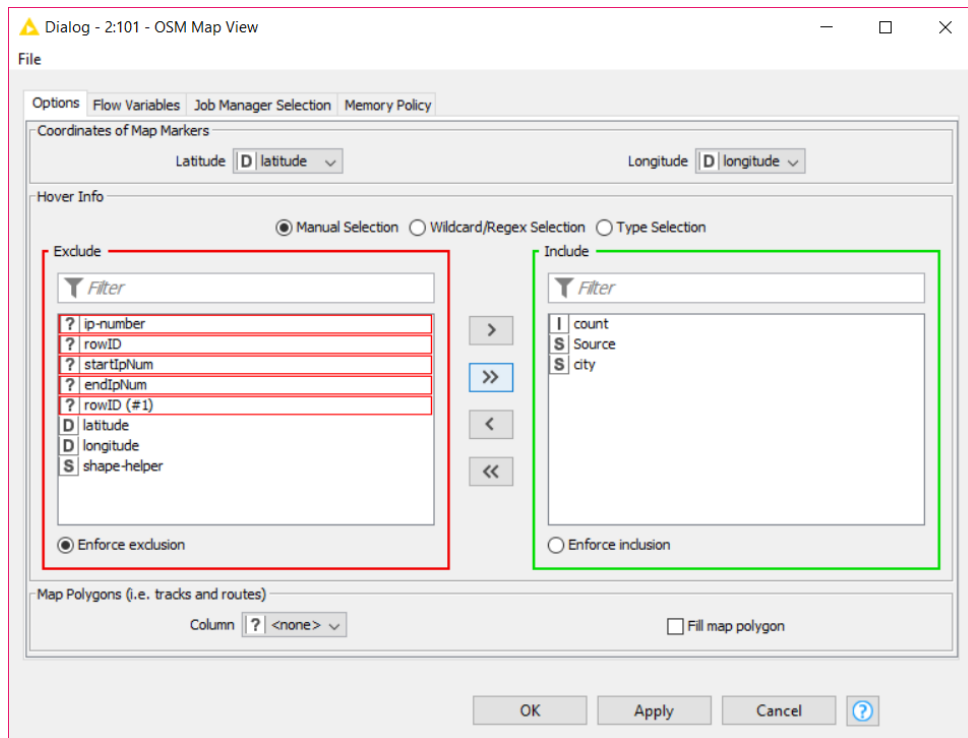
E. Visualisasi Pada Peta

1. Proses terakhir yaitu, mem-visualisasikan dari proses sebelumnya kedalam Peta. Pada proses ini kita akan membutuhkan Node yang bernama OSM Map View.

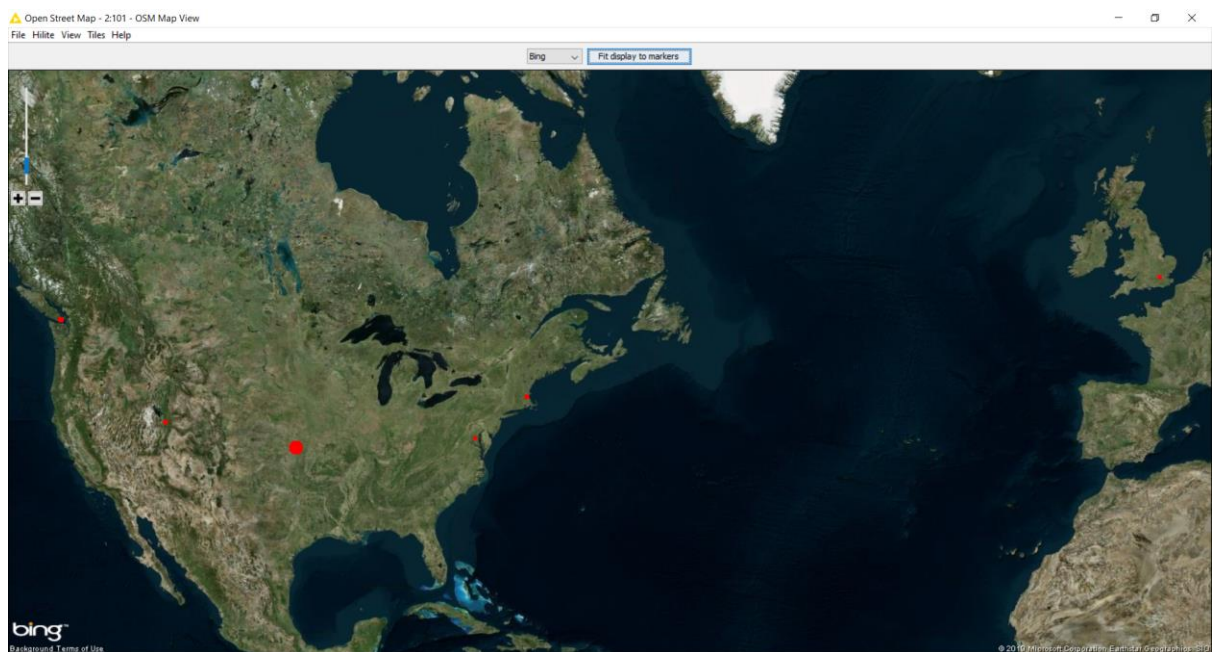


Gambar 5.31. Node OSM Map View untuk menampilkan peta

Node ini berfungsi untuk menampilkan data pada peta. Node ini membutuhkan coordinate yang didapatkan dari Latitude Longitude, serta dapat juga menampilkan info dari marker yang terlihat di peta



Gambar 5.32. Konfigurasi dari node OSM Map View



Gambar 5.33. Hasil dari visualisasi data dengan menggunakan dataset ISOT