

Simulasi Klasifikasi Paket Data Pada Jaringan Named-Data (NDN) Menggunakan *Machine Learning*

Simulation Of Data Package Classification On Named-Data Network (NDN) Using Machine Learning

1st Dzulfiqar Zakiyya Zamaluddin
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
dzulfiqarz@student.telkomuniversity.ac.id

2nd Rendy Munadi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
rendimunadi@telkomuniversity.ac.id

3rd Sri Astuti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sri.astuti@telkomuniversity.ac.id³

Abstrak—Arsitektur jaringan adalah arsitektur jaringan internet masa depan yang dapat merefleksikan dengan jaringan IP saat ini, juga inovasi dalam hal *forwarding*, *routing*, dan *caching* bersamaan dengan permasalahan yang terakumulasi dari *user* hari ini yaitu terkait sulitnya suatu data terakses dikarenakan terlalu tersentralisasinya sebuah data, oleh karena itu NDN hadir untuk memecahkan masalah tersebut. Pada pengujian kali ini, simulasi NDN akan dijalankan untuk mendapatkan dataset untuk diolah menggunakan *Machine Learning* untuk menganalisis jalur rute yang dilalui oleh suatu paket dengan tujuan agar pengembangan NDN dalam hal *routing* serta *forwarding* lebih efisien dan efektif. Hasil dari simulasi NDN untuk menganalisis laju lintas data menghasilkan hasil yang memuaskan terutama pada sisi *forwarding* dan *routing*. Lalu untuk proses *Machine Learning* paket NDN pun dapat dikatakan sukses dengan menghasilkan 8 kluster sebagai kelompok data untuk dianalisis, serta pada proses klasifikasi pun menghasilkan nilai akurasi yang optimal yaitu 0.97 dengan menggunakan nilai parameter data uji dan data latih sebesar 50/50.

Kata Kunci — *Named-data Networking*, *Machine Learning*, *Clustering* paket NDN, *Klasifikasi* paket NDN.

Abstract—*The network architecture is a future internet network architecture that can reflect the current IP network, as well as innovations in forwarding, routing, and caching along with the accumulated problems of users, namely the difficulty of accessing data due to too centralized a data, therefore The NDN is here to solve the problem. In this test, an NDN simulation will be run to obtain a dataset to be processed using Machine Learning*

to analyze the route traversed by a packet with the aim of making NDN development in terms of routing and forwarding more efficient and effective. The results of the NDN simulation to analyze the data traffic rate produce satisfactory results, especially on the forwarding and routing. Then for the Machine Learning, the NDN package can be said to be successful by producing 8 clusters as a group of data to be analyzed, and the classification process also produces an optimal accuracy value of 0.97 using the parameter value of test data and training data of 50/50.

Keywords— *Named-data Networking*, *Machine Learning*, *NDN package Clustering*, *NDN package Classification*.

I. PENDAHULUAN

Lahir pada tahun 1960-an, internet berkembang sangat pesat hingga saat ini, mayoritas penduduk dunia telah merasakan fitur yang disediakan oleh internet. Internet adalah jaringan komunikasi yang dapat menghubungkan antar pengguna dengan media elektronik seperti ponsel, komputer, dan sebagainya[1]. Indonesia adalah salah satu negara pengguna internet yang cukup dominan. Pengguna internet di Indonesia naik 15,5% dibandingkan dengan tahun sebelumnya, angka ini menunjukkan bahwa daya konsumsi masyarakat Indonesia sangatlah besar terhadap internet, dengan rentang usia pengguna antara 16-64 tahun dan rata-rata waktu yang dihabiskan sebesar 3 jam 14 menit

menjadikan daya konsumsi internet di Indonesia sangat besar[2].

Disaat yang bersamaan beberapa user menyadari bahwa arsitektur internet yang digunakan saat ini memiliki kekurangan dalam pengoperasiannya, seperti terlalu tersentralisasinya data yang mengakibatkan data yang akan diakses oleh user menjadi cukup lama dalam pengirimannya bahkan tidak jarang terjadi request time out dikarenakan data yang ingin diakses berjarak terlalu jauh. Karena permasalahan tersebut akhirnya gagasan arsitektur jaringan NDN dirumuskan, dengan mengaplikasikan beberapa aspek fungsional yang dibutuhkan dalam menjalankan interkoneksi global, jaringan NDN yang hadir tidak lama ini dapat mencerminkan segala aspek dari jaringan internet hari ini[3].

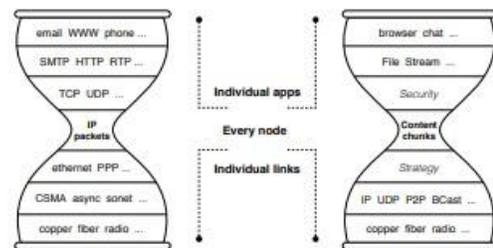
NDN menawarkan salah satu fitur yaitu lintas jalur yang akan dilalui sebuah traffic data dapat diamankan secara terstruktur dan aman, juga tiap paket yang permintaannya diatas rata-rata dapat di-caching pada tiap node dengan tujuan untuk mengurangi beban kerja dari sebuah node dan mempercepat dalam pentransaksian sebuah paket dapat dikatakan jika user akan mengakses sebuah data, paket tersebut tidak perlu mengirimkan permintaan ke server karena data yang user ingin akses telah berada di router atau node terdekatnya[20]. Pada penelitian sebelumnya pun dikatakan bahwa jaringan NDN dapat memilih rute penerusan dengan tepat dikarenakan protokol yang digunakan memiliki inovasi dalam perhitungan biaya untuk melakukan penerusan[8].

Pada simulasi NDN ini setiap data akan direkam oleh Wireshark, juga pada simulasi ini penulis akan menggunakan ndn-traffic-generator untuk instant generator dalam pengiriman paket NDN dari server ke client maupun sebaliknya[5]. Pada tahap selanjutnya data yang telah diperoleh dari hasil simulasi sebelumnya akan diolah menggunakan Machine Learning dengan tujuan agar teknisi jaringan dapat lebih mempertimbangkan strategi penerusan ataupun perutean lebih efisien dan efektif bagi user, dengan diawali pengelompokan data untuk mengelompokan data berdasarkan jalur yang dilalui serta diakhiri klasifikasi untuk lebih memastikan bahwa data tersebut dapat menjadi pertimbangan teknisi jaringan dalam melakukan inovasi pada jaringan NDN.

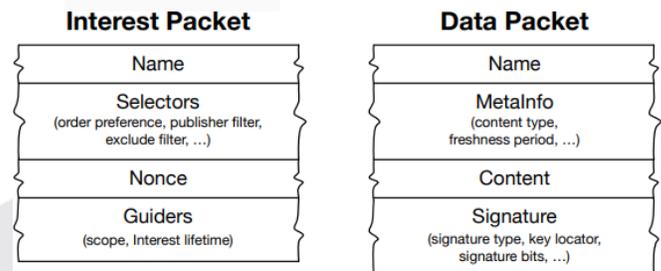
II. KAJIAN TEORI

A. Named-data Network (NDN)

Named Data Network (NDN) adalah salah satu arsitektur jaringan masa depan yang dimana setiap IP *prefix* diganti dengan *Named prefix* sebagai linknya, Pada arsitektur NDN, paket data akan diberi identitas berupa nama konten dari sebuah paket tersebut, bukan alamat sumber ataupun tujuan. Sifat NDN yang mempunyai mekanisme *forwarding* baru dan berbeda dengan arsitektur jaringan *host-to-host* yang berbasis IP[4]. proyek Named Data mengajukan evolusi dari *internet* hari ini yang dianalogikan seperti jam pasir yang berpusat pada IP yang dapat merefleksikan *Hour Glass* IP menjadi *Hour Glass* NDN[6]. Berikut adalah perbandingan *Hour Glass* IP dan NDN yang akan dijelaskan pada gambar 1 :



GAMBAR 1
HOURGLASS NDN DAN IP



GAMBAR 2
STRUKTUR PAKET NDN

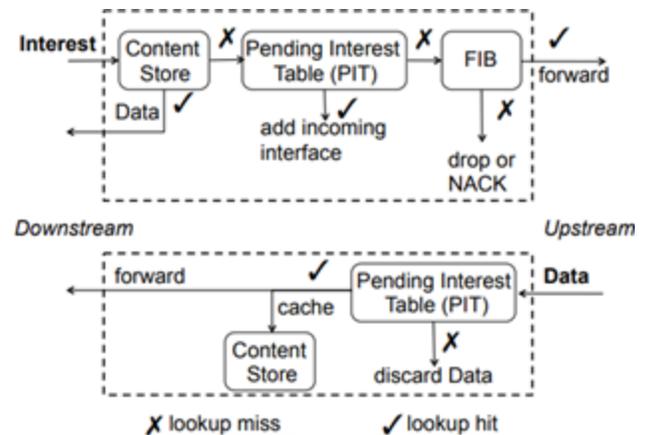
Pada gambar 2 dilampirkan gambaran paket Arsitektur NDN yang berorientasi pada penerima dan konsumen data melalui pertukaran 2 jenis paket yaitu *Interest* dan *Data*[6]. Singkatnya bila konsumen memasukan permintaan data, data tersebut akan diproses oleh sistem hingga pengecekan data dari paket *interest* yang dikirimkan konsumen, dan bila interest cocok maka paket data akan dikirimkan bersamaan dengan nama, konten, dan tanda tangan oleh produsen yang mengikat kedua paket tersebut[6]. Struktur hierarki utama pada NDN terdapat 3 unsur yaitu *Pending Interest Table* (PIT), *Content Store* (CS), *Forwarding Information Base* (FIB). Untuk

penjelasannya akan dijelaskan sebagai berikut :

1. *Pending Interest Table* (PIT), berisikan daftar dari *interest* yang belum terpenuhi. Kondisi tersebut terjadi ketika node belum mendapat respon setelah proses forwarding *Interest*. Jika sampai periode tertentu *interest* konsumen belum terpenuhi, maka akan menjadi *Timeout* terhadap paket *interest* tersebut [7].
2. *Content Store* (CS), merupakan tempat penyimpanan *cache* pada suatu node. *Cache* merupakan bagian konten pada jaringan NDN yang tersimpan pada node dan merupakan hasil duplikasi dari produsen[7].
3. *Forwarding Information Base* (FIB), berguna untuk menyediakan strategi forwarding, serta menentukan hop selanjutnya pada routing. FIB terdiri dari nama prefiks dan tujuan interface paket yang keluar. Pada NDN, FIB dapat dimodifikasi oleh protokol routing dan juga oleh strategi forwarding. FIB pada NDN berbeda dengan FIB pada jaringan IP, yakni pada NDN, FIB dapat berisikan beberapa daftar interface tujuan yang diurutkan berdasarkan rangking tertentu, sedangkan pada IP hanya 1 tujuan single best-hop[8].

B. Named-data Network (NDN)

Forwarding & Routing adalah aspek yang sangat penting dalam sebuah Arsitektur NDN. *Forwarding* yang merupakan proses pemberian paket berupa *interest*, data, dan sebagainya pada antar *face*, sedangkan *Routing* merupakan penyedia layanan pengiriman paket berdasar dari *network layer*, protocol *Routing* pun melakukan kalkulasi biaya pada jalur dan menyimpan pada FIB[6]. NDN merutekan dan meneruskan paket berdasarkan nama, yang menghilangkan tiga masalah yang disebabkan oleh alamat dalam arsitektur IP : kehabisan ruang alamat, traversal NAT, dan manajemen alamat. Tidak ada lagi masalah tentang kehabisan alamat karena NDN sendiri diakses berdasarkan nama yang artinya bisa tidak terbatas untuk merutekan ataupun meneruskan, tidak ada masalah traversal NAT juga karena NDN menghilangkan alamat, publik atau pribadi, yang artinya penetapan alamat dan manajemen tidak lagi diperlukan di jaringan lokal[6]. Berikut adalah proses dari *Forwarding* di NDN pada gambar 2.3 :



GAMBAR 3
PROSES FORWARDING PADA NDN

NFD atau NDN *Forwarding Daemon* adalah salah satu *dependencies* pada NDN, pada NFD ini penulis bisa mengatur strategi penerusan yang penulis mau mulai dari *best route*, *access*, *multipath*, ataupun *adaptive*[9]. Pada NDN pun memungkinkan sebuah paket tersebut dapat di *caching*, *caching* adalah suatu kondisi yang dimana paket data atau *interest* dapat disimpan dalam sebuah *device* untuk contohnya dalam NDN itu suatu paket dapat di-*caching* pada *node* NDN[12], Teknik *caching* pun terdapat banyak jenisnya seperti LRU (*Last Recently Used*) dan FIFO (*First In First Out*), *Caching by Popularity*, dan lainnya[14].

C. Machine Learning

Machine Learning adalah salah satu bidang ilmu dari AI (*Artificial Intelligence*). *Machine Learning* bukanlah hal yang baru, gagasan ini setidaknya ada sejak dikisaran tahun 1970, ketika algoritma terkait pertama kali muncul. *Machine Learning* menjadikan ledakan daya komputasi yang memungkinkan pengguna menggunakan teknologi ini untuk mengatasi masalah yang semakin kompleks, sementara ledakan data yang ditangkap dan disimpan telah memungkinkan pengguna untuk menerapkan teknologi ini ke berbagai domain yang terus berkembang[10]. *Machine Learning* pun memiliki beberapa jenis seperti *supervised*, *unsupervised*, & *semi-supervised* yang dimana setiap jenis ini memiliki karakter masing masing, seperti *supervised* yang mengolah data dengan data yang sudah olah hingga ML jenis ini sangat *massive* digunakan para user IT dalam melakukan hal klasifikasi, serta *unsupervised* yang sering digunakan para user IT untuk melakukan *clustering* serta melakukan pelabelan pada suatu data menggunakan mesin[15].

Dalam pengaplikasiannya terdapat beberapa

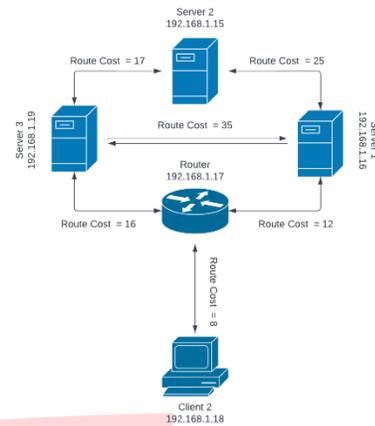
metode dalam pengerjaannya, untuk salah satu contohnya yaitu metode K-Means yang dikategorikan sebagai *Unsupervised Learning* yang dimana *K-Means* ini berperan sebagai pengkluster dari tiap data yang akan diolah berdasarkan dari kemiripan terbaik dari sebuah data dan kemiripan terburuk diantara kluster yang berbeda[18]. oleh karena itu *K-Means* sering digunakan untuk mengelompokan suatu data yang bersifat objektif[16].

Serta dapun model *Machine Learning* : *Supervised* yang salah satu metodenya itu adalah SVM (*Support Vector Machine*). SVM adalah algoritma supervised untuk klasifikasi yang bekerja dengan cara mencari *hyperplane* dengan margin terbesar, sama halnya dengan metode *K-Means* yang berpacu pada jarak sebuah pusat data dengan data yang lainnya, namun yang membedakan disini untuk proses SVM ini diperlukan label dalam pengoperasiannya untuk melatih mesin terlebih dahulu karena pada metode ini dikhususkan untuk klasifikasi data maka dari itu pada model *Machine Learning* ini dapat menghasilkan laporan hasil klasifikasinya[24].

III. METODE

A. Permodelan Sistem

Pada Tugas Akhir ini perancangan dan pemodelan yang dilakukan diawali dengan penginstalan sistem operasi untuk *environment* pengaplikasian dari NDN serta melakukan serangkaian proses instalasi dan konfigurasi dari sistem NDN, dilanjutkan dengan proses preprocessing manual menggunakan excel untuk membersihkan data yang tidak diperlukan. Setelah proses tersebut telah dilaksanakan, maka tiba pada tahap clustering data menggunakan model *K-Means* untuk melabeli tiap data yang diuji, setelah dataset telah di-*preprocessing* maka dataset tersebut akan diklasifikasi menggunakan model SVM. Berikut topologi atau model sistem yang akan digunakan pada simulasi saat ini akan dijelaskan pada gambar 4 :



GAMBAR 4
TOPOLOGI SIMULASI

Untuk hal yang berkaitan dengan *dependencies* NDN pun disini akan diinstal dengan *source* dari GitHub, langkah instalasi ini akan dilakukan di tiap *environment* Ubuntu, lalu khusus untuk Ubuntu yang menjadi peran *router* melakukan instalasi tambahan yaitu *Wireshark* guna untuk merekam lajur paket yang datang maupun yang dikirim. Selanjutnya disini penulis akan melakukan beberapa simulasi terkait pengiriman paket yang terus menerus dan berulang menggunakan salah satu *dependencies* yaitu *ndn-traffic-generator* sebagai pengirim paket *interest* ke *server* pada simulasi yang akan dijalankan[13]. Dengan memanfaatkan salah satu fitur pada *ndn-tools* yaitu *dissect packet with Wireshark*, dengan memanfaatkan fitur ini *ndn-traffic-generator* bukan hanya akan menjadi mesin pengirim data dan melaporkan hasil pengiriman, namun juga dapat merekam semua lalu lintas jalur data[22]. Untuk aksi selanjutnya penulis akan menjalankan program *Wireshark* sebagai *capture* paket data dan *interest* yang sedang dijalankan, setelah itu tiap *traffic* yang terekam pada *Wireshark* akan di *export* menjadi bentuk *csv*. Setelah file *csv* di-*export* maka akan dilakukan proses *preprocessing* secara manual lalu dilanjutkan pada tahap *clustering* oleh mesin.

Setelah melakukan *preprocessing* secara manual, maka akan dilanjutkan pada tahap pengklusteran bagi tiap data yang berasal dari simulasi sebelumnya. Model yang dirancang untuk melakukan *clustering* yaitu model *K-Means* sebagai metode pengolahan datanya dengan tujuan untuk pengelompokan data berdasarkan jalur yang telah dilalui oleh paket-paket tersebut dengan berpacu pada kolom panjang data serta kolom info dalam proses penentuan kelompoknya[19]. Setelah melalui proses clustering

maka dataset yang telah terlabeli akan diolah Kembali dengan proses klasifikasi dengan *Machine Learning* model SVM dengan tujuan seberapa tepatnya *Machine Learning* dalam menentukan berbagai kelompok bagi data yang telah diuji untuk memudahkan para teknisi jaringan dalam melakukan pengembangan lebih lanjut khususnya dalam protokol penerusan serta perutean jalur bagi paket data ataupun *interest*. Alasan penulis menggunakan model SVM ini karena SVM ini sangat berkorelasi dengan model *K-Means* dalam otomasinya yaitu menentukan *hyperplane* dari tiap data untuk memaksimalkan jarak antar label/kluster yang telah

dijalankan pada pengujian sebelumnya[24].

B. Permodelan Sistem

Pada proses ini terdapat 3 tahap pengujian yaitu, simulasi NDN, *preprocessing & clustering K-means*, dan klasifikasi menggunakan model SVM. Adapun protokol atau parameter yang akan dilakukan dalam simulasi NDN ini, beberapa parameter yang akan disertakan sudah menjadi default dari beberapa *dependencies* NDN yang telah diinstal. Untuk lebih jelasnya, berikut akan disertakan pada tabel 1 dibawah ini :

TABEL 1
PARAMETER SIMULASI NDN

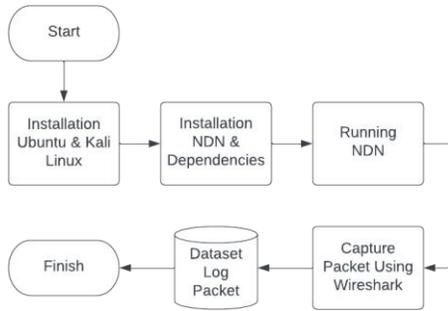
Parameter	Keterangan
Paket Data	Data konfigurasi <i>ndn-traffic-generator</i>
<i>Cache Replacement Policy</i>	<i>Least Recently Used (LRU)</i>
<i>Routing</i>	<i>Named Data Link State Routing Protocol (NLSR)</i>
<i>Forwarding</i>	<i>Best-Route</i>
<i>Interest Generator</i>	<i>Ndn-traffic-generator</i>

1. Tahap 1 : Simulasi NDN

Pada tahap simulasi pengujian untuk NDN, simulasi akan dilakukan dengan cara menjalankan NFD pada tiap *node* NDN dan *router* NDN yang telah terinstall *dependencies* NDN didalamnya. Selanjutnya akan dilakukan pendaftaran tiap alamat udp menggunakan fitur NFD dengan command “*nfdc face create udp://(Alamat IP)*” untuk menghubungkan tiap *node* NDN, serta membuat jalur untuk tiap *node* agar dapat terhubung satu sama lain dengan *router*, dan begitupun sebaliknya. Setelah semua *node* terhubung maka di tiap *node server* dan *node client* akan menjalankan fungsi *ndn-traffic-generator* sesuai dengan perannya kecuali untuk *router*, *router* tidak akan menjalankan apapun cukup hanya sebagai penghubung dari *node server* dan *node client*. Untuk strategi penerusan yang digunakan pada simulasi ini yaitu strategi *best-route*, serta untuk protokol perutean yang digunakan yaitu NLSR.

Fungsi *ndn-traffic-server* ialah sebagai produsen dimana *content store* tersedia, dan sebagai tujuan para *node client* untuk mengirim *interest* paket NDN[21]. Untuk fungsi *ndn-traffic-client* disini sebagai konsumen dari data yang disediakan oleh *node server*

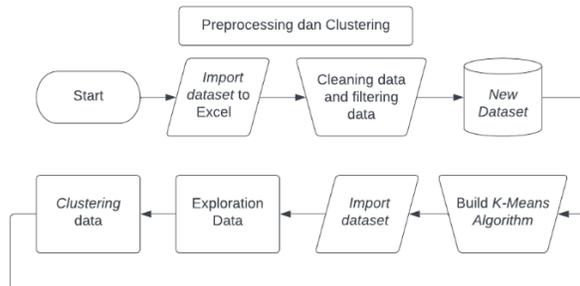
di *content store*, juga para konsumen akan mengirim *interest* juga menerima *packet* data dari server melalui *router*, pada tahap ini lini masa dari laju lintas paket akan direkam oleh *Wireshark* yang memanfaatkan salah satu fitur dari *ndn-tools* yaitu *dissect packet with Wireshark*[22]. Pada simulasi NDN yang akan dijalankan, penulis akan mencoba bagaimana bila suatu client meminta data atau mengirimkan interest hanya mencantumkan “/” saja tanpa ada nama prefix yang disertakan, untuk mengetahui hasil yang dikeluarkan oleh sistem NDN bila client hanya mengirimkan sebuah interest dengan hanya “/” saja, apakah sistem NDN akan mengirimkan data yang di-*caching* atau memberhentikan proses dikarenakan standar pengiriman *interest*-nya tidak lengkap. Berikut diagram alir yang akan digunakan pada simulasi, akan dijelaskan pada gambar 5 dibawah ini :



GAMBAR 5
DIAGRAM ALIR SIMULASI NDN

2. Tahap 2 : Preprocessing Manual & Clustering

Pada tahap ini dataset yang didapatkan pada pengujian sebelumnya akan dibersihkan dari data-data yang tidak diperlukan, setelah dataset bersih dari berbagai data yang tidak diperlukan, maka dilanjutkan dengan tahap *preprocessing* menggunakan *Machine Learning* serta melakukan pelabelan pada tiap data dengan cara pengelompokan data menggunakan metode *K-Means*. Metode *K-Means* ini sendiri yaitu metode pengolahan data dengan cara mengklusterkan tiap data dari jarak terdekat dari jarak titik pusat data yang ditentukan oleh algoritmanya, yang menjadikan metode ini paling cocok untuk *dataset* yang telah penulis dapatkan dari simulasi sebelumnya[17]. simulasi *Machine Learning* ini dilakukan di GoogleColab sebagai *environment* pengaplikasian *Machine Learning*-nya, pada proses ini lebih ditekankan pada penentuan kelompok paket data maupun interest menurut jalur yang telah dilalui dengan berpacu pada kolom panjang data serta kolom info dalam proses penentuan kelompok atau label bagi tiap data yang diuji. Berikut diagram alir dari skenario akhir akan sertakan pada gambar 6 dibawah ini :



GAMBAR 6
DIAGRAM ALIR TAHAP 2

3. Tahap 3 : Klasifikasi Data

Pada tahap ketiga ini, pengujian akan dilanjutkan pada tahap klasifikasi data yang dimana telah di-

preprocessing serta telah di-*clustering* untuk menentukan label dari tiap datanya. Tujuan pada tahap ini yaitu untuk menentukan kolerasi label data pada variabel data yang diujikan untuk mengetahui bahwa label dari suatu data tersebut tepat atau tidak dengan data yang dilabelinya. Pada proses ini dapat disebutkan penting dikarenakan bila data yang dilabeli oleh mesin tidak memiliki akurasi yang cukup baik maka perlu pengujian ulang kembali hingga mendapatkan nilai yang optimal, dikarenakan bila nilai yang dihasilkan telah optimal maka data hasil olahan mesin ini dapat menjadi bahan analisis bagi teknisi jaringan untuk melakukan pengembangan lebih lanjut pada jaringan NDN terutama pada proses penerusan serta perutean agar lebih dapat memudahkan user dalam melakukan proses transaksi sebuah paket. Untuk model klasifikasi yang digunakan disini yaitu model SVM (*Support Vector Machine*) yang dimana algoritma ini akan menentukan garis *hyperplane* dari dataset guna menguji perbandingan margin dari tiap data, untuk lebih jelasnya akan dijelaskan pada gambar 7 berikut :

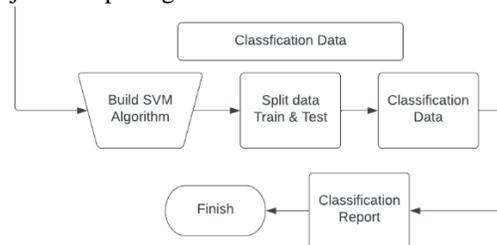


FIGURE 7
DIAGRAM ALIR KLASIFIKASI

C. Parameter Evaluasi

Nilai parameter yang dievaluasikan pada simulasi ini terdapat beberapa hal yang harus diperhatikan karena simulasi ini sangat berfokus pada kinerja NDN dalam melakukan sebuah transaksi paket antara *user* dan *client*, lalu proses *clustering* data menggunakan metode *K-Means*, serta optimasi dari *Machine Learning* sebagai klasifikator data yang telah didapatkan menggunakan metode SVM, berikut parameter yang harus diperhatikan dalam evaluasi kali ini :

1. Label Protokol

Label Protokol disini sangat diperhatikan karena telah ditekankan dari sebelumnya bahwa tiap paket interest maupun data harus terlabeli dengan protokol “UDP (NDN)”, bila hal tersebut tidak tercapai maka data yang tidak terlabeli akan dihapus dari dataset.

2. Jumlah & Nilai pada Kluster

Tiap nilai atau angka kluster akan mewakili identitas dari tiap data itu sendiri maka diperlukan evaluasi dari hal tersebut agar identitas dari tiap data jelas dan terstruktur. Jumlah kluster pun perlu dievaluasi agar tetap optimal dalam pengoperasiannya.

3. Variabel Confussion Matrix

Untuk mengukur hasil dari pengolahan yang dilakukan oleh mesin, salah satu caranya yaitu menggunakan Confusion Matrix sebagai parameter evaluasinya, terdapat parameter seperti f1-score, precision, dan recall[23], namun pada pengujian ini lebih berfokus pada akurasi yang dihasilkan oleh Machine Learning dalam melakukan klasifikasi pada dataset yang telah diinput. Parameter tersebut dipakai untuk menilai efektifnya sebuah kinerja mesin berikut rumus perhitungan yang akan disertakan dibawah ini :

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP\ all + TN\ all}{\text{Total data yang telah diolah}} \times 100 \\
 \text{Recall} &= \frac{TP\ all}{TP\ all + TFNi} \times 100 \\
 f-1\ score &= 2 \times \frac{(\text{Recall} \times \text{Akurasi})}{(\text{Recall} + \text{Akurasi})} \times 100 \\
 \text{precision} &= \frac{TP\ all}{TP\ all + FPI} \times 100
 \end{aligned}$$

IV. HASIL DAN PEMBAHASAN

A. Tahap 1 : Simulasi NDN

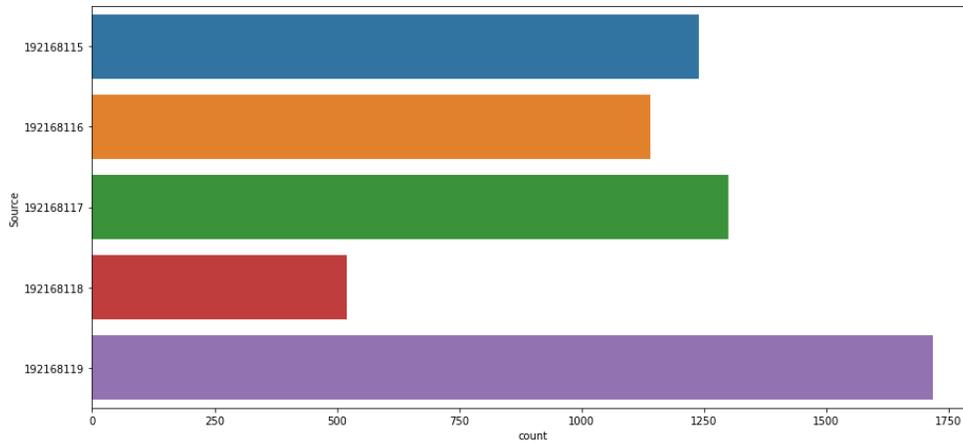
Pada tahap 1 ini akan dilakukannya simulasi jaringan NDN menggunakan 5 node diantaranya 3 node sebagai server, 1 node sebagai client, dan 1 node sebagai router, juga pada simulasi ini tidak akan dilakukannya proses advertise satu persatu dikarenakan menggunakan konfigurasi NLSR yang dimana konfigurasi dijalankan ditiap node. Hasil dari simulasi yang dijalankan mendapatkan hasil yang sukses dikarenakan paket interest dan paket data yang dikirimkan terlabeli oleh "UDP (NDN)", serta mendapatkan rekaman alur penerusan saat proses merekam setiap laju lintas data NDN menggunakan Wireshark. Berikut adalah printscreen dari hasil proses perekaman laju lintas data menggunakan

Wireshark :

Source	Destination	Protocol	Length	Info
192168119	192168115	UDP (NDN)	140	Interest /ndn/Server2/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168116	192168117	UDP (NDN)	187	Data /ndn/Server1/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168116	192168117	UDP (NDN)	140	Interest /ndn/Router1/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168119	192168117	UDP (NDN)	187	Data /ndn/Server3/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168119	192168117	UDP (NDN)	140	Interest /ndn/Router1/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168115	192168119	UDP (NDN)	283	Data /ndn/Server2/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168115	192168119	UDP (NDN)	140	Interest /ndn/Server3/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168117	192168118	UDP (NDN)	187	Data /ndn/Router1/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168119	192168116	UDP (NDN)	187	Data /ndn/Server3/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168117	192168118	UDP (NDN)	140	Interest /ndn/Client1/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168119	192168116	UDP (NDN)	140	Interest /ndn/Server1/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168117	192168118	UDP (NDN)	187	Data /ndn/Router1/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168117	192168116	UDP (NDN)	140	Interest /ndn/Server1/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168119	192168115	UDP (NDN)	187	Data /ndn/Server3/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168119	192168115	UDP (NDN)	140	Interest /ndn/Server2/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168117	192168119	UDP (NDN)	187	Data /ndn/Router1/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168117	192168119	UDP (NDN)	140	Interest /ndn/Server3/%Cl.Router/router/nlsr/INFO/%07%20%08%03
192168116	192168119	UDP (NDN)	187	Data /ndn/Server1/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168116	192168119	UDP (NDN)	187	Data /ndn/Server1/%Cl.Router/router/nlsr/INFO/%07%20%08%03ndr
192168116	192168119	UDP (NDN)	140	Interest /ndn/Server3/%Cl.Router/router/nlsr/INFO/%07%20%08%03

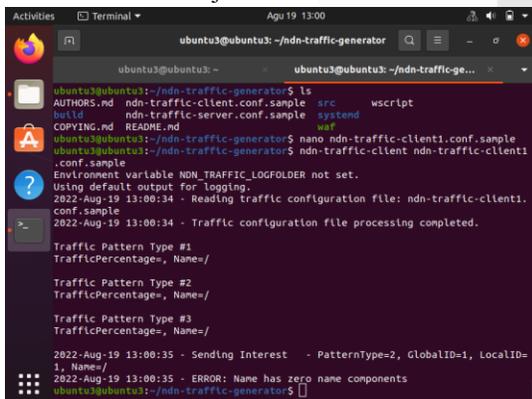
GAMBAR 8
DATA HASIL SIMULASI NDN

Pada gambar 8 ditampilkan bahwa data dan interest yang diminta oleh user atau node client ini melakukan penerusan dan memilih rute efisien dan route cost yang tidak terlalu besar. Bila dilihat pada gambar pemodelan sistem, disana dikatakan bahwa bila interest user mencapai router, disana terdapat 2 jalur penerusan bila interest yang diminta oleh user berada pada server 2, maka mayoritas paket akan memilih jalur terpendek dan memiliki biaya rute paling kecil yaitu jalur melalui server 3 namun tidak sedikit juga yang melakukan penerusan terhadap server 1 terlebih dahulu lalu menuju server 3 atau server 2 walaupun biaya rute yang digunakan cukup besar, hal ini dikarenakan adanya strategi penerusan itu sendiri, juga sifat NDN itu sendiri yang memiliki sifat tracing data, yang dimana suatu interest akan diteruskan bila data yang diminta tidak terdapat pada node tertentu, walaupun ada jalur terpendek yang dapat dilalui yaitu melalui jalur server 3, namun sistem akan tetap mengarahkan penerusan paket ke jalur lain, hal ini dikarenakan traffic pada jalur terpendek sedang sibuk melayani paket-paket yang datang terlebih dahulu dan akibatnya interest akan melalui jalur lain. Namun hal ini masih dapat dikatakan optimal dikarenakan perutean dan penerusan dari paket tetap terstruktur dengan adanya protokol NLSR, jika tidak maka user hanya akan menerima data NACK, karena protokol perutean bersifat statis tidak dinamis, oleh karena itu penulis menggunakan protokol NLSR agar paket data ataupun interest dapat direspon hingga tidak menyebabkan kerugian bagi user. Adapun histogram dari perhitungan kuantitas data yang diterima oleh tiap node berikut akan dijelaskan pada gambar 9 :



GAMBAR 9
HISTOGRAM PAKET NDN DARI NODE

Pada gambar 9 menunjukkan bahwa kuantitas data terbanyak yang diterima oleh node atau node yang sering dilalui yaitu *node server 3*, hal ini sangat berkorelasi dengan pemodelan sistem yang dicantumkan pada BAB 3, bahwa terdapat 2 jalur untuk melakukan penerusan jika paket telah mencapai *router*, yaitu jalur *server 1* dan jalur *server 3*. Simulasi ini dikatakan sangat sukses karena membuktikan bahwa sistem NDN akan mengkalkulasi biaya dari tiap jalur untuk menentukan jalur yang lebih dekat dan efisien. Namun bila user hanya mengirimkan *interest* dengan mencantumkan “/”, maka yang terjadi sistem NDN akan memberhentikan secara paksa permintaan tersebut dikarenakan standar dari pengiriman interest tidak tercapai dan tidak lengkap, oleh karena itu sistem akan melakukan *cancel interest* pada client tersebut untuk memberitahukan bahwa *interest* yang user kirimkan tersebut tidak bisa dijalankan. Berikut *printscreen* dari hasil pengiriman *interest* yang hanya mencantumkan “/” saja :



GAMBAR 10
HASIL SIMULASI PENGIRIMAN INTEREST DENGAN “/” TANPA NAMA

B. Tahap 2 : Preprocessing Manual & Clustering

Pada tahap 2 ini dataset yang telah didapatkan pada tahap 1 akan di *preprocessing* terlebih dahulu agar nantinya data tersebut menjadi data yang siap diolah pada tahap selanjutnya. Pada tahap ini dataset akan melalui *filtering* secara manual untuk memisahkan data yang berprotokol NDN dan non-NDN, serta akan dilakukan juga *clustering* menggunakan metode *K-Means* untuk melabeli tiap data dengan kluster yang telah ditentukan oleh mesin.

1. Tahap 1 : Preprocessing Manual

Pada tahap ini data akan melalui proses *filtering* secara manual untuk memisahkan data yang berprotokol NDN dan yang tidak, dikarenakan pengujian ini berfokus pada NDN maka data yang tidak memiliki identitas sebagai protokol NDN akan disingkirkan agar menjadikan data yang siap diolah pada tahap selanjutnya, Setelah mengetahui data apa saja yang harus disingkirkan maka proses *filtering* akan dilakukan, berikut *printscreen* dari data yang telah diolah akan dijelaskan pada gambar 11 :

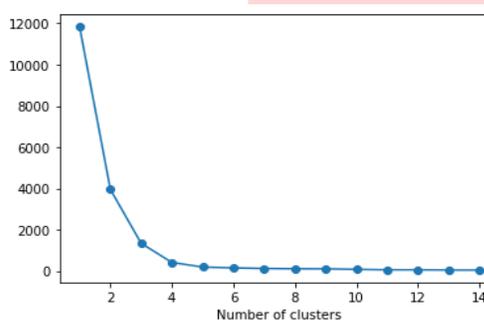
Source	Destination	Protocol	Content	Time
192168119	192168115	UDP (NDN)	140 Interest /ndn/Server2/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168116	192168117	UDP (NDN)	187 Data /ndn/Server1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168116	192168117	UDP (NDN)	140 Interest /ndn/Router1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168119	192168117	UDP (NDN)	140 Interest /ndn/Router1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168115	192168119	UDP (NDN)	283 Data /ndn/Server2/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168115	192168119	UDP (NDN)	140 Interest /ndn/Server3/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168117	192168118	UDP (NDN)	187 Data /ndn/Router1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168119	192168116	UDP (NDN)	187 Data /ndn/Server3/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168117	192168118	UDP (NDN)	140 Interest /ndn/Client1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168119	192168116	UDP (NDN)	140 Interest /ndn/Server1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168117	192168116	UDP (NDN)	187 Data /ndn/Router1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168119	192168115	UDP (NDN)	187 Data /ndn/Server3/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168119	192168115	UDP (NDN)	140 Interest /ndn/Server2/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168117	192168119	UDP (NDN)	187 Data /ndn/Router1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168117	192168119	UDP (NDN)	140 Interest /ndn/Server3/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	
192168116	192168119	UDP (NDN)	187 Data /ndn/Server1/%C1.Router/router/nlsr/INFO/%07%20%08%03ndr	

GAMBAR 11
DATA YANG TELAH DIBERSIHKAN

2. Tahap 2 : Clustering

Pada tahap ini lebih difokuskan kepada eksplorasi data untuk diuji serta proses *clustering* yang

dilakukan oleh *K-Means* untuk melabeli data sebagai syarat untuk melakukan pengujian selanjutnya. Metode *K-Means* ini sangat ideal untuk dijalankan pada proses ini karena sifat dari model ini yaitu menentukan kelompok/kluster dari tiap data berdasarkan jarak dari pusat data yang ditentukan oleh algoritmanya, yang menjadikan algoritma ini sangat tepat untuk proses *labeling* pada data agar label dapat diklasifikasi ketepatannya terhadap data yang diwakilinya. Berikut hasil iterasi dari simulasi *clustering* yang dijalankan oleh metode *K-Means*, untuk lebih jelas nya akan dijelaskan pada gambar 12 :



GAMBAR 12
HASIL ITERASI CLUSTERING

Hasil dari iterasi *clustering* pada gambar 12 menunjukkan bahwa dataset yang diuji dapat dikelompokkan hingga 12 kelompok, prediksi tersebut didapat dari pengujian dengan acuan dari kolom panjang data dan kolom info, namun penulis disini hanya akan mengelompokkan data hanya menjadi 8 kelompok karena menurut penulis kuantitas tersebut sangat ideal bagi data yang telah penulis teliti dan analisis berasal dari *preprocessing* sebelumnya. Pada proses *clustering* ini, penulis memiliki tujuan untuk mengelompok data menjadi berbagai kelompok dikarenakan walaupun panjang data yang disertakan pada *dataset* terdapat kesamaan dengan satu sama lain, namun dengan proses pengelompokan ini penulis maupun *user* dapat mengetahui bahwa setiap data yang telah dikelompokkan memiliki keterangan bagaimana alur paket tersebut berjalan atau melakukan penerusan, maka dari itu proses ini sangat krusial bagi para teknisi untuk mengetahui kelompok data berdasarkan jalur yang telah dilalui. Berikut kuantitas data dari tiap label, untuk lebih jelasnya akan dijelaskan pada gambar 13 :

```

7      2950
3      859
2      618
5      394
4      380
0      361
1      347
6       9
Name: klabel, dtype: int64
    
```

FIGURE 13
KUANTITAS DATA PADA TIAP LABEL

Pada gambar 13 menunjukkan bahwa kuantitas data terbesar terdapat pada label dengan nomor 7, dan kuantitas data terkecil terdapat pada label dengan nomor 6. Untuk lebih jelasnya, kuantitas data akan dijelaskan berpacu pada kolom panjang data sebagai *variable X* nya, berikut penjelasannya :

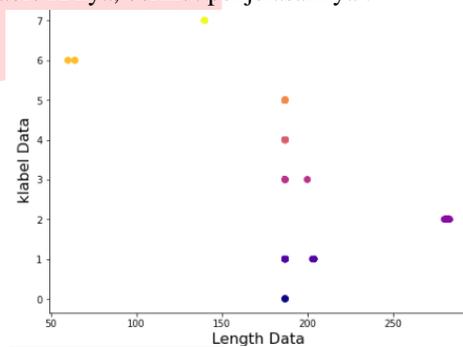


FIGURE 14
PLOTING LABEL

Pada gambar 14 menunjukkan bahwa beberapa nilai memiliki label yang sama namun dengan panjang data yang berbeda, hal ini menunjukkan bahwa data tersebut memiliki kesamaan dengan data yang lain dikarenakan data lain tersebut terdapat pada jalur yang sama pada saat data tersebut diteruskan oleh sistem NDN.

C. Tahap 3 : Klasifikasi Data

Pada tahap 3 ini lebih difokuskan pada penentuan akurasi yang dilakukan oleh *Machine Learning* pada tahap sebelumnya dalam proses pemberian label pada tiap data, dengan melihat akurasi yang dihasilkan pada proses klasifikasi ini, dapat menunjukkan performa mesin dalam mengolah data yang telah diuji. Memilih model SVM sebagai model klasifikasinya, menjadikan pengujian ini berjalan lancar tanpa hambatan dengan didukung dari kolerasi metode sebelumnya menjadikan proses pengelompokan data dan proses pengklasifikasian data berjalan dengan baik. Hasil yang diperoleh dari proses klasifikasi ini mendapatkan hasil yang optimal bahkan dengan jumlah kuantitas data latih dan data tes yang dikonfigurasi hampir sama, berikut hasil yang akan dijelaskan pada gambar

dibawah ini :

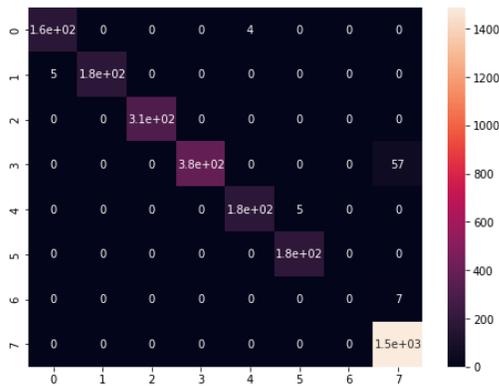


FIGURE 15
CONFUSION MATRIX

classification report

	precision	recall	f1-score	support
0	0.97	0.98	0.97	169
1	1.00	0.97	0.99	180
2	1.00	1.00	1.00	308
3	1.00	0.87	0.93	437
4	0.98	0.97	0.98	185
5	0.97	1.00	0.99	184
6	0.00	0.00	0.00	7
7	0.96	1.00	0.98	1489
accuracy			0.97	2959
macro avg	0.86	0.85	0.85	2959
weighted avg	0.97	0.97	0.97	2959

FIGURE 16
CLASSIFICATION REPORT

Tujuan dari proses klasifikasi ini yaitu bila saat program *Machine Learning* ini adalah untuk mencari akurasi dari label pada data yang telah dikelompokan, mengapa hal ini harus diperhatikan dikarenakan label atau kelompok yang dicantumkan harus memiliki akurasi yang baik agar dapat digunakan sebagai bahan pertimbangan bagi teknisi untuk melakukan pengembangan selanjutnya dari sistem NDN terutama dalam hal penerusan dan perutean. Label yang tertera pun menunjukkan lini masa dari data yang telah melalui jalur tertentu, tentunya hal ini sangat berguna bagi pengembang dalam hal pengembangan selanjutnya agar nantinya perutean dan penerusan bagi data ataupun *interest* yang *user* akses dapat lebih efisien dan lebih efektif.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Pada bagian akhir skripsi ini, penulis akan memaparkan beberapa kesimpulan yang dapat diambil pada pengujian yang dilakukan pada Tugas Akhir ini, yaitu sebagai berikut :

1. Jaringan NDN memiliki sistem penerusan dan perutean yang berbeda dengan jaringan

IP, hal ini dibuktikan pada saat simulasi yang dilakukan, paket yang dikirimkan akan ditentukan jalur mana yang efisien dan efektif. Hal ini didukung dengan banyaknya paket yang melalui node dengan biaya terkecil pada saat proses penerusan suatu data maupun *interest*, juga protokol perutean dari NDN yaitu NLSR menjadi *core* dalam penentuan rute yang akan dilalui oleh suatu paket, begitu juga NFD yang memiliki berkorelasi baik dengan NLSR yaitu strategi penerusan *best-route*.

2. Menurut hasil pengujian yang telah dilakukan, substansi bagi user adalah dapat mengakses data dengan waktu yang efisien dikarenakan protokol perutean dan strategi penerusan dari NDN sangat memperhatikan akan biaya rute serta jalur yang akan diambil. Jika bagi teknisi ataupun pengembang, hasil simulasi NDN yang telah dijalankan serta hasil yang telah dikeluarkan oleh *Machine Learning* sangat bermanfaat untuk tujuan pengembangan lebih lanjut sistem NDN terutama pada hal perutean dan penerusan yang nantinya akan sangat memudahkan user dalam mengakses sebuah data.
3. Model K-Means dan SVM sangat tepat untuk menjadi mekanisme *clustering* dan klasifikasi bagi data NDN dikarenakan algoritma yang dipakai menentukan perbandingan tiap jarak data yang sama dari pusat jarak yang ditentukan oleh mesin, yang menjadikan metode-metode ini sangat tepat untuk mengolah data dari NDN. Hasil *clustering* dan klasifikasi menunjukkan hasil yang optimal, baik dari sisi kualitas data ataupun kuantitas data, hasil yang diperoleh sangat ideal dan optimal, serta performansi dari mesin dapat dikatakan sangat baik karena menghasilkan akurasi pada proses klasifikasi sebesar 0.97.

B. Saran

Adapun saran dari hasil simulasi dari Tugas Akhir ini sebagai berikut :

1. Diperlukan node yang lebih rumit agar mendapatkan data yang cukup memampuni untuk dilakukannya pengujian lebih lanjut.
2. Untuk kedepannya mungkin lebih baik menggunakan emulator sebagai media pengujiannya, karena bila menggunakan tiap *environment* menjadi *node* maka

penyimpanan yang diperlukan menjadi sangat besar yang mengakibatkan tempat penyimpanan akan terpakai habis untuk melakukan pengujian.

3. Diperlukan lebih banyak variabel pada data agar pengujian lebih dapat berjalan baik, juga apabila data yang diambil dari *Wireshark* terakumulasi dengan data *byte* yang tercantum, mungkin simulasi selanjutnya akan menghasilkan nilai yang lebih baik.

REFERENSI

- [1] GIG by Indosat Ooredoo. 2021. "Inilah Sejarah Internet dan Perkembangannya di Indonesia" [online]. Available : <https://gig.id/stories/tech/sejarah-internet-perkembangannya-indonesia>.
- [2] Galuh Putri Riyanto. 2021. Feb 23. Jumlah Pengguna Internet Indonesia 2021 tembus 202 juta [online]. Available : <https://tekno.kompas.com/jumlah-penggunainternet-indonesia-2021-tembus-202-juta>.
- [3] Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thorton, JD., dan Smetter, K., Zhang, B., Tsodik, G., Claffy, kc., Papadopoulos, C., Abdelzaher, T., Wang, L., Crowley, P., Yeh, E. (2010): Named data networking (NDN) project, Technical Report NDN-0001. 3-4.
- [4] Alexander afanasyev, Xiaoke Jiang, Yingdi Yu, Jiewen Tan, Yumin Xia, Allison Makin, Lisia Zhang. "NDNS: A DNS-Like Name Service for NDN". IEEE, 2017.
- [5] named-data/ndn-traffic-generator[online]. Available : <https://github.com/named-data/ndn-traffic-generator> (accessed Nov. 10, 2021).
- [6] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang. Jul. 2014. "Named-data networking (NDN)" ACM SIGCOMM Computer Communication Review Volume 4, Number 3.
- [7] S. H. Ahmed, S. H. Bouk, and D. Kim, Content-Centric Networks An Overview, Applications and Research Challenges. 2016.
- [8] C. Yi, "ADAPTIVE FORWARDING IN NAMED DATA NETWORKING," The University of Arizona, 2014.
- [9] Named-data/NDN Forwarding Daemon[online]. Available : <https://github.com/named-data/nfd> (accessed Nov. 17, 2021).
- [10] Louridas, P. Ebert, C. 2016. Machine Learning. The IEEE Computer Society.
- [11] Get Ubuntu. Available : <https://ubuntu.com/> (accessed Oct. 28, 2021)
- [12] N. M. Ridha, S. R. Nana. May. 2021. Caching and Machine Learning Integration Method on Named Data Network : a Survey. University of Bath. The IEEE Xplore.
- [13] ndn-cxx[online]. Available : <https://github.com/named-data/ndn-cxx> (accessed Jan. 4, 2022).
- [14] N. A. Muhammad, R. U. A. Muhammad, U. Rehmat, K. S. Byung. Mar. 2020. A Comparative Performance Analysis of Popularity-Based Caching Strategies in Named Data Networking. National Research Foundation of Korea (NRF).
- [15] Shobha Gangadhar, Rangaswamy Shanta. Jul. 2018. Machine Learning. R. V. College of Engineering.
- [16] Scikit Learn : K-Means [online]. Available : <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html> (accessed Jul. 2, 2022)
- [17] L. Keunsoo, K. Juhyun, K. H. Ki, H. Younggo, K. Sehun. 2008. DDoS attack detection method using cluster analysis.
- [18] Y. Liu, W. Li, Y. Li. 2007. Network Traffic Classification Using K-means Clustering.
- [19] S. P. Kristina, Y. Miin-shen. Unsupervised K-Means Clustering Algorithm.
- [20] S. Samar, S. Senthilmurugan, R. K. Hemant, P. Bighnaraj, S. Anantha. 2016. Performance Evaluation of Caching Policies in NDN - an ICN Architecture.
- [21] H. Lingling. Jan. 2020. Packet-level-based traffic aggregation to optimize NDN content delivery.
- [22] ndn-tools [online]. Available : <https://github.com/named-data/ndn-tools> (accessed Jun. 23, 2022)
- [23] Scikit Learn : K-Means [online]. Available : https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html
- [24] Y. Ruixi, L. Zhu, G. Xiaohong, X. Li. Jul. 2008. An SVM -based Machine Learning Method for Accurate Internet Traffic Classification.