

KLASIFIKASI DATA DELAY DENGAN LFID STRATEGI FORWARDING MENGUNAKAN MACHINE LEARNING UNTUK MEMAKSIMALKAN KINERJA JARINGAN NDN (NAMED DATA NETWORK)

Sri Astuti¹, Tody Ariefianto Wibowo², Ratna Mayasari³, Ibnu Asror⁴, Gregorius Paradana Satriawan⁵
Fakultas Teknik Elektro^{1,2,3,5}, Fakultas Teknik Informatika⁴, Universitas Telkom^{1,2,3,4,5}
sriastuti@telkomuniversity.ac.id¹, ariefianto@telkomuniversity.ac.id²,
ratnamayasari@telkomuniversity.ac.id³, ibnu@telkomuniversity.ac.id⁴,
gregoriopradanas@student.telkomuniversity.ac.id⁵

Abstract

Named Data Network (NDN) is the future internet network that data-centric and adaptive to consumer requirement. Routing and forwarding systems on the NDN networks are different from IP networks due to the use of cache at each node on the network. The implementation of the Loop Free Inport-Dependent (LFID) routing protocol on NDN networks aims to eliminate loops on the network by eliminating the preferred routes or inefficient next hops. Forwarding strategies that can be implemented are Best Route, Access, Random, and Multicast. Therefore, machine learning technology is needed with various classification methods that can be implemented in machine learning so the output gives the recommendations that can be used to maximize the performance of the NDN network. The final result of this study recommends that the forwarding strategies of Best Route and Access provide good delay values, which in the range of 150 ms to 300 ms. Random forwarding strategy with a payload size ≥ 3072 kbps still provides a good delay value to the network, which in the 150 to 300 ms range. All forwarding strategies of Best Route, Access, Random, and Multicast provide delay values with a very good category of delay values, which is below 150 ms if the type of interest (data) that requested to the network is a popular interest.

Keywords: *Named Data Network , Routing, Forwarding, Machine Learning*

Abstrak

Named Data Network (NDN) adalah jaringan internet masa depan yang bersifat data-centric dan adaptif terhadap kebutuhan konsumen. Sistem routing dan forwarding pada jaringan NDN berbeda dengan jaringan IP dikarenakan adanya penggunaan cache di tiap node pada jaringan. Penggunaan protocol routing Loop Free Inport-Dependent (LFID) pada jaringan NDN bertujuan menghilangkan loop pada jaringan dengan menghilangkan rute pilihan atau next hop yang tidak efisien. Strategi forwarding yang dapat diimplementasikan adalah Best Route, Access, Random, dan Multicast. Oleh karena itu diperlukan teknologi pembelajaran mesin (Machine Learning) dengan berbagai macam metode klasifikasi yang dapat diimplementasikan didalamnya agar keluarannya menghasilkan rekomendasi yang dapat digunakan untuk memaksimalkan kinerja jaringan NDN. Hasil akhir pada penelitian ini merekomendasikan bahwa strategi forwarding Best Route dan Access memberikan nilai delay yang bagus, yakni direntang 150 ms sampai dengan 300 ms. Strategi forwarding Random dengan ukuran payload ≥ 3072 kbps masih memberikan nilai delay yang bagus juga ke jaringan, yakni direntang 150 ms sampai dengan 300 ms. Semua strategi forwarding Best Route, Access, Random, dan Multicast memberikan nilai delay dengan kategori sangat bagus, yakni dibawah 150 ms bila jenis interest (data) yang di-request ke jaringan adalah interest populer.

Kata Kunci: *Named Data Network , Routing, Forwarding, Machine Learning.*

PENDAHULUAN

Perkembangan dunia telekomunikasi digital berdampak pada peningkatan kebutuhan yang harus diakomodasi oleh jaringan. arsitektur jaringan masa depan dituntut mampu mengakomodasi layanan konten multimedia broadband dengan beragam karakteristik. Bahkan menurut ITU-T bahwa jaringan masa depan akan bersifat data-aware, berbasis pada distribusi konten/data[1]. Arsitektur Named

Data Network (NDN) mengubah konsep jaringan yang berbasis pada penyedia (host-centric) menjadi fokus pada informasi tujuan (data-centric). Hal tersebut dapat mempercepat proses pengiriman informasi pada jaringan [2]. Pada jaringan NDN, Penggunaan mekanisme routing dan forwarding berperan penting pada jaringan karena akan berpengaruh terhadap nilai kinerja jaringan. Salah satu protokol routing dikembangkan dalam jaringan NDN, adalah

Loop-Free Inport-Dependent (LFID) [3]. LFID merupakan protokol yang fokus dalam upaya penghilangan 3 looping pada jaringan, dan juga melakukan modifikasi strategi *forwarding* [3]. Pemilihan strategi *forwarding* diperlukan sesuai dengan karakteristik jaringan. Hal ini disebabkan karena adanya beberapa pilihan jalur (*path*), sehingga perlu diputuskan melalui lintasan mana data akan dilewatkan. Pada penelitian sebelumnya[4], telah dilakukan analisis kinerja terhadap protokol *routing* LFID dengan menjalankan strategi *forwarding best route, multicast, random* dan *access*.

Pada penelitian ini, kami mengklasifikasikan penggunaan protokol *routing* LFID dengan strategi *forwarding best route, multicast, random* dan *access* dengan metode klasifikasi yang terdapat pada *Machine Learning*. Hal ini dilakukan dengan tujuan untuk memperoleh rekomendasi strategi *forwarding* yang bagus untuk diterapkan pada jaringan NDN guna memaksimalkan kinerja jaringan. Pada penelitian ini, kami menggunakan data hasil simulasi dari penelitian sebelumnya [4] untuk dilakukan proses *pre-processing* menjadi sebuah dataset yang cocok digunakan pada *Machine Learning*. Data hasil simulasi yang digunakan adalah data delay.

Pada penelitian ini kami menggunakan empat (4) kategori delay, yakni sangat bagus, bagus, sedang, dan jelek untuk menyatakan target kelas dalam pengklasifikasian. Kategori delay tersebut merujuk dari standar Tiphon. Penelitian ini disusun atas tiga bagian, yaitu bagian pertama menjelaskan teori-teori yang mendukung penelitian, bagian kedua menjelaskan metode penelitian, validasi model klasifikasi, ukuran (metrik) performansi, dan hasil evaluasi semua model klasifikasi yang digunakan dalam penelitian ini.

Tabel 1.1 Kategori Delay

Kategori Latency	Besar Delay
Sangat bagus	< 150 ms
Bagus	150 s/d 300 ms
Sedang	300 s/d 450 ms
Jelek	> 450 ms

(Sumber : TIPHON)

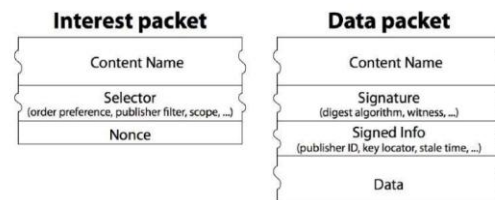
KAJIAN PUSTAKA

A. Named Data Networking

Named Data Networking (NDN) merupakan evolusi dari teknologi internet yang berbasiskan

IP (Internet Protocol) ke basis konten atau data. NDN mengubah paradigma layanan jaringan, yang mana pengiriman paket melalui alamat IP menjadi pengiriman paket dengan identitasnya berdasarkan nama. Adanya perubahan sistem kerja tersebut, NDN memiliki lebih banyak kelebihan yakni penamaan yang lebih mudah, keamanan yang lebih kuat, mobilitas tinggi, dan proses broadcasting yang mudah.

Struktur paket yang digunakan pada komunikasi data NDN, yaitu *interest* dan data. Konsep komunikasi NDN ini dimulai bagian penerima data atau konsumen melalui pertukaran jenis data, yaitu *interest* dan data. Untuk menerima paket data, *consumer* mengirimkan paket *interest* yang sudah ditandai dengan data yang diinginkan. Router akan menyimpan informasi tersebut beserta beberapa informasi jaringan seperti *interface* yang digunakan, kemudian mem-*forward* paket *interest* tersebut. Ketika paket *interest* ini telah menemukan node tujuan yang memiliki data sesuai dengan data yang di-*request* oleh *consumer*, maka paket data tersebut akan dikirimkan kembali ke *consumer* yang me-*request* data menggunakan jalur yang sama beserta *signature key* dari *producer* [5].



Gambar 1.1 Arsitektur Paket di NDN[5]

Zhang, dkk. [5] menyebutkan bahwa ada beberapa komponen penting dari NDN yang salah satunya adalah *routing* dan *forwarding*. Proses dari *routing* dan *forwarding* di NDN sangat tergantung pada nama yang tertera pada paket *interest*. Penggunaan nama ini dapat mengurangi permasalahan yang terjadi pada arsitektur IP seperti: keterbatasan alamat, translasi NAT, mobilitas, dan skalabilitas jaringan. Di sisi lain beberapa hal perbedaan antara jaringan IP dan jaringan NDN dapat dilihat pada tabel 1.2 dibawah ini.

Tabel 1.2 Perbedaan jaringan IP dengan NDN[6]

Pendekatan	Internet	NDN
<i>Routing</i>	FIB <i>IP Prefixes ; single best next hop; Stateless forwarding</i>	<i>Name prefixes; Ranked list interface; Stateful forwarding</i>
Pengalamatan	<i>IP Address</i>	<i>Named Data</i>
<i>Caching</i> pada Jaringan	Tidak Didukung	Dimungkinkan

B. Router NDN

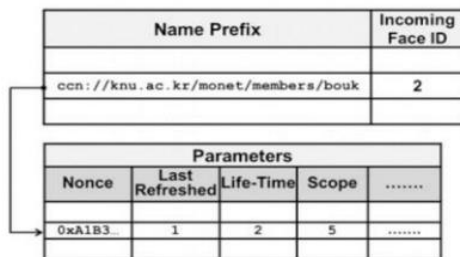
Setiap router di NDN memiliki tiga struktur data, yaitu *Forwarding Information Base (FIB)*, *Pending Interest Table (PIT)*, dan *Content Store (CS)*.

- a) *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 berisikan beberapa daftar *interface* tujuan yang diurutkan berdasarkan rangking tertentu, sedangkan pada IP hanya 1 tujuan *single best-hop*[7].

Tabel 1.3 Struktur FIB[2]

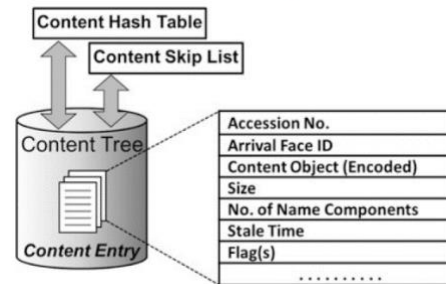
Prefix	Outgoing Face ID
ccn://knu.ac.kr/monet/	1
ccn://comsats.eud.pk/ee/	1,2
.....

- b) *Pending Interest Table (PIT)* berisi daftar *interest* yang belum terpenuhi. Kondisi tersebut terjadi ketika node belum mendapat respon setelah proses *forwarding interest*. Jika sampai periode tertentu *interest* belum terpenuhi, maka akan menjadi *timeout* [2]. Daftar PIT akan dihapus jika node menerima data dari *upstream* atau ketika *interest* menjadi *timeout*.



Gambar 1.2 Struktur PIT[2]

- c) *Content Store (CS)* merupakan tempat penyimpanan *cache* pada suatu *node*. CS mirip dengan alokasi *buffer memory* pada *router* [2]. Pada NDN, konten (*cache*) yang tersimpan dalam *buffer* tersebut dimungkinkan untuk digunakan kembali, yaitu untuk memenuhi permintaan *interest* yang masuk ke *node* tersebut. Ketika paket *interest* sampai ke suatu *node*, maka *node* tersebut akan mencocokkan *interest* dengan *cache* yang ada pada CS *node*. Jika *cache* pada CS sesuai dengan paket *interest*, maka *node* akan mengirimkan data dari CS.



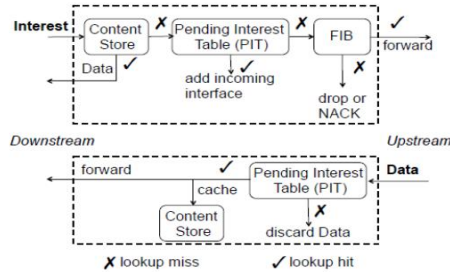
Gambar 1.3 Struktur Content Store[2]

C. Proses Forwarding data di NDN

Konsep *forwarding interest* dan data pada NDN dapat dilihat pada gambar 1.4 di bawah. *Consumer* (pengirim) mengirimkan paket *interest* ke jaringan, *node (router)* yang menerima paket *interest* tersebut akan melakukan pengecekan pada bagian penyimpanan konten (*content store*) dan mencocokkan data, jika datanya cocok dengan paket *interest* yang di *request* oleh *consumer*, maka paket data akan dikirimkan kembali oleh *node (router)* tersebut ke *consumer* yang meminta data termasuk juga informasi *interface* dari asal *interest* tersebut. Sebaliknya jika datanya tidak cocok dengan *interest* yang diminta *consumer*, maka *node* akan mencarinya ke *Pending Interest Table (PIT)*. Jika paket *interest* ditemukan di *Pending Interest Table (PIT)* tersebut, maka *node* akan mencatat daftar sumber *interface* masukan yang baru tersebut. Jika tidak ditemukan maka *node* akan mencarinya ke *Forwarding Information Base (FIB)* dan meneruskan paket *interest* tersebut ke *producer*. Jika pada FIB tidak ditemukan *hop* tujuan selanjutnya untuk meneruskan paket *interest*, maka *NACK* akan dikirimkan ke *consumer* (*node* pengirim paket *interest*), dan di FIB melakukan *drop* pada paket *interest* tersebut[8].

Saat sebuah *node (router)* menerima paket data dan cocok dengan yang tercantum di PIT maka data akan diteruskan ke semua *interface* arah *downstream* yang tercatat dalam PIT. Selanjutnya *node* akan menyimpan data pada bagian penyimpanan konten (*content store*). Paket data selalu menggunakan jalur balik yang sama dengan *interest* sehingga memberikan keseimbangan aliran data. Hal ini dikarenakan satu paket *interest* menghasilkan satu paket data pada tiap *link*. Selain membawa alamat *interface (host)*, paket *interest* diteruskan ke *producer*

berdasarkan nama dan paket data diteruskan ke *consumer* (node pengirim) berdasarkan pada informasi di *Pending Interest Table (PIT)*[8].



Gambar 1.4 Proses *forwarding* di NDN[8]

D. Teori Forwarding

Forwarding adalah proses penerusan paket baik itu paket *interest*, data, ataupun *NACK* melalui antar node. Menurut *NDN Forwarding Daemon (NFD)* terdapat 2 jenis *forwarding* yaitu *forwarding pipeline* dan strategi *forwarding* [9]. *Forwarding pipeline* adalah proses penerusan paket untuk memenuhi suatu kondisi yang cocok, seperti deteksi dan penerimaan *interest*. Strategi *Forwarding* adalah siasat dalam pengambilan keputusan untuk melakukan penerusan paket termasuk sistematisa untuk tujuan pengiriman paket dan waktu pengiriman. Menurut NFD strategi *forwarding* yang digunakan terbagi atas empat [9], yaitu :

- Best Route* merupakan strategi penerusan paket yang mengirimkan *interest* pada arah *upstream* dengan biaya *routing* yang paling rendah di FIB.
- Multicast* merupakan strategi penerusan paket yang mengirimkan *interest* ke semua node arah *upstream* pada FIB. pengecualian untuk node yang melakukan *request* data adalah target *interface* pada FIB.
- Access* merupakan strategi penerusan paket yang mengirimkan paket *interest* menggunakan *next hop* terakhir digunakannya (*last-working*). Jika tidak ada *next hop* terakhir yang digunakannya, maka pada strategi *access* ini akan melakukan pengiriman paket *interest* dengan cara *multicast* ke semua *interface* yang terdapat pada FIB.
- Random* merupakan strategi penerusan paket yang sudah ada pada *routing Loop-Free Inport-Dependent (LFID)*. Strategi ini akan meneruskan paket *interest* ke *hop* selanjutnya secara acak yang terdapat pada daftar FIB.

E. Loop-Free Inport-Dependent (LFID) Routing

Loop-Free Inport-Dependent (LFID) adalah protokol modifikasi *link-state routing* dengan skema *multipath* yang bersifat *hop by hop*. Protokol LFID menggunakan pembacaan topologi seperti yang terdapat di OSPF tetapi menerapkan metode penghitungan jalur *link-state* yang berbeda. Penghitungan jalur *link-state* tersebut memungkinkan jalur lebih pendek dan bebas *loop*, sehingga jalur tersebut dapat digunakan untuk proteksi kegagalan ataupun pengurangan kemacetan[3].

F. Metode Klasifikasi dengan Machine Learning

Metode klasifikasi yang digunakan pada penelitian ini ada 6 model, yaitu *Naïve Bayes*, *CN2 Rule Induction*, *kNN*, *Tree*, dan *Random Forest*. Penggunaan lebih dari satu model klasifikasi pada penelitian ini bertujuan untuk membandingkan hasil evaluasi antar metode klasifikasi tersebut, dan selanjutnya akan dipilih model yang paling sesuai dengan dataset.

1. Naïve Bayes

Naïve Bayes adalah metode klasifikasi yang mudah diterapkan dan menggunakan konsep probabilitas yang mudah. Performansi *Naives Bayes* cukup baik saat digunakan pada kasus dengan data yang sangat besar. Dalam Teorema Bayes, suatu probabilitas bersyarat dinyatakan sebagai berikut[11] :

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (1)$$

Dimana X adalah bukti, H adalah hipotesis, $P(H|X)$ adalah probabilitas bahwa hipotesis H benar untuk bukti X (probabilitas posterior H dengan syarat X), $P(X|H)$ adalah probabilitas bahwa bukti X benar untuk hipotesis H (probabilitas posterior X dengan syarat H), $P(H)$ adalah probabilitas prior hipotesis H, dan $P(X)$ adalah probabilitas prior bukti X.

2. CN2 Rule Induction

CN2 Rule Induction menginduksi aturan dari data menggunakan algoritma CN2. *CN2 Rule Induction* bekerja hanya untuk klasifikasi. Algoritma CN2 adalah teknik klasifikasi yang dirancang untuk induksi sederhana yang efisien, aturan yang dapat dipahami dari bentuk “jika kondisi maka kelas prediksi”, bahkan dalam domain yang kemungkinan terdapat noise[12].

3. k-Nearest Neighbour

k-Nearest Neighbour adalah metode yang bekerja dengan cara mencari sejumlah k pola (di antara semua pola latihan yang ada disemua

kelas) yang terdekat dengan pola masukan, kemudian menentukan kelas keputusan berdasarkan jumlah pola terbanyak diantara k pola tersebut (voting). Pada dasarnya proses pelatihan kNN menghasilkan k yang akurasi tinggi dilakukan dengan observasi terhadap sejumlah k sampai dihasilkan k yang paling optimum. Penggunaan *pseudocode* pada algoritma kNN menunjukkan bahwa kNN sangat sederhana, mudah diimplementasikan hanya dengan mengatur satu parameter k [11].

4. Random Forest

Random forest (RF) adalah metode pembelajaran yang digunakan untuk klasifikasi, regresi, dan lainnya. Random forest membangun sekumpulan pohon keputusan. Masing-masing pohon dikembangkan dari sampel *bootstrap* dari data latih (*train data*). Saat mengembangkan pohon individu, subset sembarang dari atribut diambil (secara random), kemudian atribut terbaik dipisahkan untuk dipilih. Model akhir didasarkan pada *vote* mayoritas dari pohon yang dikembangkan secara individu di *forest*[13].

5. Decision Tree Learning

Decision Tree Learning (DTL) adalah teknik pembelajaran mesin yang membangun representasi aturan klasifikasi berstruktur sekuensial hirarki dengan cara mempartisi himpunan data latih secara rekursif. Pembelajaran DTL menghasilkan pohon keputusan yang berupa *n-ary branching tree* yang mempresentasikan suatu aturan klasifikasi[11].

Beberapa metode pada *Decision Tree Learning* (DTL) adalah *classification and regression trees* (CART), *iterative dichotomizer version 3* (ID3), *C4.5 Quinlan*, *C5.0 Quinlan*, *Cubist Quinlan*, dan lainnya. Metode ID3 banyak digunakan untuk data kategorial, sedangkan metode C4.5 digunakan untuk data kategorial maupun numerik (kontinu). Keduanya mengevaluasi semua atribut menggunakan suatu ukuran ketidakmurnian (*impurity*), seperti *Information Gain* dan *Gain Ratio*, untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan himpunan sampel data. Metode ID3 menggunakan *Information Gain*, sedangkan C4.5 menggunakan *Gain Ratio*[11].

6. Neural Network

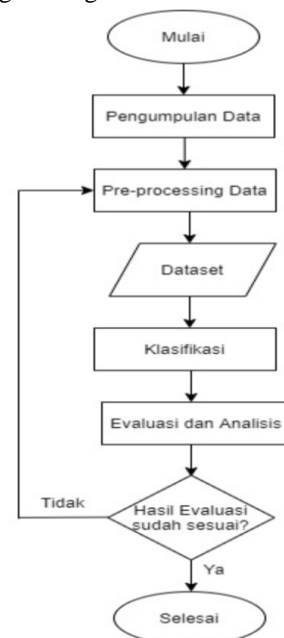
Neural network menggunakan algoritma *Multi Layer Perceptron* (MLP) yang dapat mempelajari model non-linier dan linier. *Multi Layer Perceptron* (MLP) merupakan model ANN yang paling banyak digunakan dalam studi maupun praktis untuk permasalahan klasifikasi

sederhana. Struktur jaringannya yang simple memudahkan untuk mempelajari dan mengaplikasikannya pada masalah klasifikasi [11].

Proses pembelajaran pada MLP adalah menemukan bobot-bobot sinaptik yang paling optimum untuk mengklasifikasikan himpunan data latih dan data validasi. Salah satu algoritma yang paling populer digunakan untuk melatih MLP adalah *Back Propagation*. Algoritma ini melakukan pelatihan MLP dalam dua tahap, yaitu perhitungan maju dan perhitungan mundur. Perhitungan maju untuk menghitung galat (biasa disebut *loss function*) antara keluaran aktual dan target. Perhitungan mundur yang mempropagasikan balik galat tersebut untuk memperbaiki bobot-bobot sinaptik pada semua *neuron* yang ada[11].

METODE PENELITIAN

Sistem klasifikasi yang dibuat pada penelitian ini menggunakan salah satu parameter unjuk kerja jaringan NDN, yakni delay untuk merekomendasikan LFID strategi *forwarding* yang maksimal digunakan pada jaringan NDN. Metode penelitian yang dilakukan mulai dari pengumpulan data, *pre-processing* data menjadi sebuah dataset, sampai hasil evaluasi dari masing-masing metode klasifikasi yang dipakai.



Gambar 3.1 Diagram Alir Penelitian

A. Pre-processing

Pre-processing adalah proses perubahan bentuk data dari tidak terstruktur menjadi data terstruktur. Tahapan pre-processing mengubah

data tekstual menjadi data yang siap digunakan sebagai model teks *mining* [10]. Proses ini sangat penting dilakukan untuk mendapatkan data yang valid dan sesuai dengan model yang akan di buat. Data hasil simulasi yang digunakan pada penelitian ini, tidak bisa digunakan langsung menjadi dataset, sehingga harus dilakukan proses *pre-processing* data menjadi sebuah dataset. Dataset yang terdiri dari beberapa kolom (atribut) dan kelas (target).

B. Dataset

Data delay dengan LFID strategi *forwarding* yang digunakan diambil dari hasil simulasi menggunakan NDN simulator [4]. Terdapat lima kolom dalam dataset yang digunakan, yakni 4 kolom sebagai atribut dan 1 kolom sebagai kelas. Kolom atribut terdiri dari *interest*, jumlah konsumer, strategi *forwarding*, dan ukuran *payload*, sedangkan kolom kelas adalah target yang ingin dicapai.

Tabel 3.1 Dataset

	Target	Interest	strategi Forwarding	Jlh Consumer	Payload size
1	Sedang	umum	Random	5.0	2048.0

Dataset terdiri dari dua bagian, yaitu data train dan data test. Data train digunakan sebagai input data dari model yang telah dibuat, sedangkan data test digunakan sebagai input analisis klasifikasi dan akurasi dari model yang telah dibuat. Dalam penelitian ini digunakan *cross validation* agar semua data pada dataset memiliki peluang yang sama menjadi data train dan data test

C. Validasi Model

Validasi model dilakukan bertujuan untuk mengetahui performansi yang valid dari masing-masing model yang sudah dibangun sebelumnya. Beberapa teknik validasi model adalah *Resubstitution*, *Hold-Out*, *k-Fold Cross Validation*, *Leave One Out Cross Validation*, *Random Subsampling*, *Bootstrapping*[11]. Teknik validasi yang dilakukan pada penelitian ini menggunakan *k-Fold Cross Validation*. Metode *k-Fold Cross Validation* membagi data secara acak menjadi *k* sub-himpunan (disebut *fold*) yang saling bebas. Dalam hal ini menggunakan *k* = 10, artinya masing-masing *fold* berisi 1/10 bagian data. Penggunaan *k-Fold Cross Validation* agar semua data pada dataset memiliki peluang yang sama menjadi data *train* dan data *test*, dan mendapatkan akurasi dengan bias dan variansi yang relatif rendah[11].

D. Metrik Performansi

Ukuran evaluasi model klasifikasi menggunakan beberapa metrik performansi, yaitu akurasi,

presisi, *recall*, *f1-score*. Metrik ini diperoleh dari *confusion matrix* pada tabel 3.2 di bawah untuk masing-masing model klasifikasi. *Confusion Matrix* menjelaskan hasil yang diprediksi dan setiap kolom menunjukkan kondisi aktual (sebenarnya)[14]. *Confusion Matrix* juga berfungsi untuk menganalisis kualitas model klasifikasi dalam mengenali tuple-tuple (kumpulan baris dan kolom) dari kelas yang digunakan[11]. Nilai dari metrik performansi bisa diperoleh dengan menggunakan formula dua sampai lima di bawah.

Tabel 3.2 Confusion Matrix

		Kelas Hasil Prediksi	
		Ya	Tidak
Kelas Aktual	Ya	True Positive	False Negative
	Tidak	False Positive	True Negative

True Positive (TP) adalah jumlah baris yang positif dilabeli dengan benar oleh model klasifikasi. *True Negative* (TN) adalah jumlah baris yang negatif yang dilabeli dengan benar oleh model klasifikasi. *False Positive* (FP) adalah jumlah baris yang negative dilabeli salah oleh model klasifikasi. *False Negative* (FN) adalah jumlah baris yang positif yang dilabeli salah oleh model klasifikasi. Formula yang dipakai untuk ukuran evaluasi model klasifikasi sebagai berikut [11] :

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (2)$$

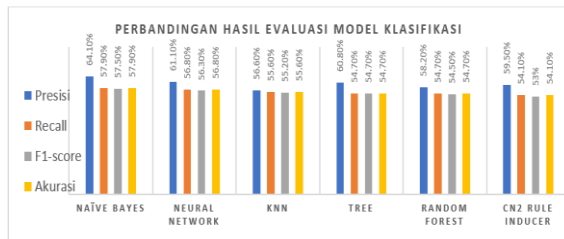
$$Precision \text{ atau ukuran kepastian} = \frac{TP}{TP+FP} \quad (3)$$

$$Recall \text{ atau sensitivity} = \frac{TP}{TP+FN} \quad (4)$$

$$F1-Score = \frac{2 \times precision \times recall}{precision+recall} \quad (5)$$

HASIL UKURAN EVALUASI

Hasil ukuran evaluasi model klasifikasi meliputi tingkat akurasi, presisi, *recall*, *f1-score*. Parameter ukur tersebut diperoleh berdasarkan *confusion matrix* dari masing-masing model klasifikasi. Hasil perbandingan ukuran evaluasi masing-masing model klasifikasi dapat dilihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Hasil Evaluasi Model Klasifikasi

Tabel 4.1 Perbandingan Hasil Evaluasi Model Klasifikasi

Model	Precisi	Recall	F1-score	Akurasi
Naïve Bayes	64,1 %	57,9 %	57,5 %	57,9 %
Neural Network	61,1 %	56,8 %	56,3 %	56,8 %
kNN	56,6 %	55,6 %	55,2 %	55,6 %
Tree	60,8 %	54,7 %	54,7 %	54,7 %
Random Forest	58,2 %	54,7 %	54,5 %	54,7 %
CN2 Rule Inducer	59,5 %	54,1 %	53%	54,1 %

Dari tabel 4.1 di atas terlihat bahwa masing-masing model klasifikasi memiliki rata-rata tingkat akurasi di atas 50%. Artinya tingkat pengenalan atau presentase dari jumlah tuple dalam data uji yang diklasifikasikan dengan benar oleh masing-masing model klasifikasi adalah rata-rata di atas 50%. Tiga model klasifikasi mempunyai tingkat akurasi yang tertinggi dari lainnya, yaitu model *Naïve Bayes*, *Neural Network*, dan *kNN*. Dari ketiga model tersebut, model *Naïve Bayes* memiliki tingkat akurasi tertinggi yaitu 57,9 %, sehingga tingkat kesalahannya (*error rate*) adalah 42,1%. Tingkat presisi (ukuran kepastian) adalah 64,1% untuk model *Naïve Bayes*. Artinya presentase tuple yang dilabeli positif pada kenyataannya adalah benar yakni 64,1%. Ukuran kelengkapan (*recall*) yang menyatakan presentase tuple yang dilabeli sebagai positif pada model *Naïve Bayes* adalah 57,9 %. Secara keseluruhan tingkat akurasi dari semua model klasifikasi yang digunakan pada penelitian ini masih dikatakan bagus karena tingkat kesalahan (*error rate*) dibawah 50%.

Tabel 4.2. Kaitan kondisi dengan kelas

IF "Kondisi"	Then "Kelas"
Interest = Populer	Sangat Bagus
Strategi forwarding = Bestroute	Sangat Bagus
Strategi forwarding = Access	Bagus
Strategi forwarding = Random, dan Ukuran Payload \geq 3072	Bagus
Strategi forwarding = Bestroute atau Access, dan Jumlah Konsumer \geq 5	Bagus
Strategi forwarding = Multicast, dan Ukuran Payload \geq 3072	Sedang
Strategi forwarding = Multicast, dan Ukuran Payload \leq 3072	Sedang
Strategi forwarding = Random, dan Jumlah Konsumer \geq 5	Sedang
Strategi forwarding = Multicast	Jelek

Tabel 4.2 di atas mengenai kaitan kondisi sistem dengan kelasnya yang sesuai. Dari tabel tersebut dapat dilihat bahwa *interest* jenis populer dengan strategi *forwarding* apapun baik itu *bestroute*, *access*, *random*, *multicast* memiliki delay rendah (sangat bagus) dikarenakan seringnya *interest* jenis populer itu di rikues ke jaringan maka memungkinkan data konten yang di minta konsumer dapat ditemukan di *router* terdekat dengan *consumer*. Penggunaan strategi *forwarding bestroute* dan *access* memiliki delay rendah (bagus) dikarenakan strategi *forwarding bestroute* menggunakan hop dengan *cost* terendah dan *access* menggunakan lintasan (hop) terakhir yang digunakan. Penggunaan strategi *forwarding multicast* memiliki delay tinggi (jelek) dikarenakan pada *multicast* mem-broadcast *interest* yang datang kepadanya ke semua node (*router*) di jaringan sehingga memungkinkan pula *packet loss* yang besar.

KESIMPULAN

Dari penelitian yang dilakukan dapat diambil kesimpulan bahwa masing-masing model klasifikasi yang digunakan memiliki rata-rata tingkat akurasi, presisi, dan tingkat sensitivitas yaitu di atas 50% dan tingkat kesalahan (*error rate*) dibawah 50%. Model *Naïve Bayes* memiliki tingkat akurasi tertinggi, yaitu 57,9 %, dan tingkat kesalahan (*error rate*) adalah 42,1%. Strategi *forwarding Best Route* dan *Access* memberikan nilai delay yang bagus, yakni direntang 150 ms sampai dengan 300 ms. Strategi *forwarding Random* dengan ukuran *payload* \geq 3072 kbps masih memberikan nilai delay yang bagus juga ke jaringan, yakni direntang 150 ms sampai dengan 300 ms. Semua strategi *forwarding Best Route*, *Access*, *Random*, dan *Multicast* memberikan nilai delay dengan kategori sangat bagus, yakni dibawah 150 ms bila jenis *interest* (data) yang di-riquest ke jaringan adalah *interest* populer.

REFERENSI

- [1] ITU-T. (2012). Future networks: Objectives and design goals. Recomm. ITU-T Y.3001. Didownload dari : <https://www.itu.int/rec/T-REC-Y.3001-201105-I>.

- [2] S. H. Ahmed, S. H. Bouk, and D. Kim., (2016). Content-Centric Networks An Overview, Applications and Research Challenges.
- [3] K. Schneider., (2019). Hop-by-Hop Multipath Routing: Choosing the Right Nexthop Set.
- [4] Pradana Satriawan, Gregorius., (2020). Analisis Performa Strategi Forwarding pada Protokol Routing Loop-Free Inport-Dependent (LFID) pada Jaringan Named Data Network (NDN). Fakultas Teknik Elektro Universitas Telkom, Bandung.
- [5] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang., (2014). Named data networking. ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 66–73.
- [6] D. Saxena and I. I. T. Roorkee., (2016). Named Data Networking: A Survey. Comput. Sci. Rev. Elsevier, vol. 19, pp. 15–55.
- [7] C. Yi., (2014). Adaptive Forwarding in Named Data Networking. The University of Arizona.
- [8] L. Zhang, V. Jacobson, Dmitri Krioukov, Chirstos Papadopoulos., (2016). Named Data Networking (NDN) Project 2013-2014 Report. Didownload dari : <https://named-data.net/project/annual-progress-summaries/2013-2014/>.
- [9] A. Afanasyev et al., (2015). NFD Developer’s Guide. pp. 1–56.
- [10] V. Srividhya and R. Anitha., (2010). Evaluating Preprocessing Techniques in Text Categorization. International Journal of Computer Science and Application, pp. 49-51.
- [11] Suyanto. (2018). Machine Learning Tingkat Dasar dan Lanjut. Informatika Bandung.
- [12] Clark, Peter and Robin Boswell., (1991). Rule Induction with CN2: Some Recent Improvements. Machine Learning - Proceedings of the 5th European Conference (EWSL-91),151-163.
- [13] Breiman, L., (2001). Random Forests. In Machine Learning. 45(1), 5-32.
- [14] Raouf Boutaba, Mohammad A. Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, Oscar M. Caicedo., (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications.