JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 10
No. 1
April 2024

Pendekatan Metode *Ensemble Learning* untuk Deteksi Serangan DDoS menggunakan *Soft Voting Classifier*

Steven Joses[#1], Stefanie Quinevera[#2], Ricky Mardianto[#3], Donata Yulvida[#4], Ary Mazharuddin Shiddiqi[*5]

[#]*Teknik Informatika,Universitas Widya Dharma Pontianak*
Jl. H.O.S. Cokroaminoto No. 445, Kota Pontianak, Kalimantan Barat
[1]stevenjoses@widyadharma.ac.id
[2]stefani_quinevera@widyadharma.ac.id
[3]ricky_mardianto@widyadharma.ac.id
[4]donata_yulvida@widyadharma.ac.id

[*]*Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember*
Jl. Sukolilo, Surabaya 60111, Indonesia
[5]ary.shiddiqi@if.its.ac.id

***Abstrak*— Serangan Distributed Denial of Service (DDoS) adalah jenis serangan yang kompleks dan sering melibatkan berbagai pola lalu lintas jaringan yang berbeda. Model soft voting classifier dapat menggabungkan hasil dari beberapa model klasifikasi yang berbeda, sehingga meningkatkan kemampuan untuk mendeteksi dan mengatasi serangan DDoS dengan berbagai pola dan skenario yang berbeda. Dengan memanfaatkan model soft voting classifier berdasarkan fitur-fitur yang mendukung, dapat meningkatkan ketahanan sistem terhadap serangan DDoS dengan lebih efektif, mengurangi dampaknya, dan memastikan ketersediaan sumber daya jaringan dan layanan internet bagi pengguna yang mengaksesnya. Data yang digunakan dalam penelitian ini menggunakan dataset DDoS yang diperoleh dari situs kaggle.com. Dataset ini memiliki 23 atribut termasuk satu variabel *output* dengan jumlah data sebanyak 104.245 record. Dilakukan preprocessing pada dataset kemudian diklasifikasi menggunakan lima model machine learning dan sepuluh ensemble learning method untuk mendapatkan hasil akurasi tertinggi. Hasil pengujian menunjukkan bahwa ensemble method sangat optimal dalam mendeteksi serangan DDoS baik menggunakan fitur berdasarkan Information Gain maupun menggunakan fitur berdasarkan Gain Ratio dibandingkan dengan metode machine learning tunggal.**

***Kata kunci*— DDoS, *Soft Voting Classifier*, Deteksi Serangan DDoS, *Machine Learning*, *Information Gain*, *Gain Ratio*, *Feature Selection*.**

I. PENDAHULUAN

Serangan *Distributed Denial of Service* (DDoS) dikenal sebagai serangan dalam dunia maya yang bertujuan untuk membuat sumber daya jaringan atau layanan Internet tidak tersedia bagi pengguna yang mengaksesnya. Serangan DDoS menggunakan sekelompok mesin yang dikelola untuk menyerang mesin korban [1]. Umumnya, serangan ini dilakukan dengan membanjiri sistem target dengan trafik internet berlebihan dari berbagai sumber yang tersebar luas. Serangan ini membutuhkan waktu lebih sedikit untuk mencapai targetnya, menyebabkan server menolak pengguna yang mengaksesnya, dan dalam kasus terburuk, menyebabkan *server crash* [2]. Kasus pertama terjadi pada tanggal 22 Juli 1999, ketika sebuah komputer di Universitas Minnesota terdapat skrip berbahaya bernama *Trin00* yang menyerang 114 komputer lainnya [3]. Pada tahun 2020, serangan terbesar DDoS di *Amazon Web Services* (AWS) mencapai 2,3 Tbps pada momentum tertingginya dan berlangsung selama tiga hari. Hal ini menunjukkan bahwa serangan semacam ini telah menjadi hal yang lazim [4]. Karena keterbatasan perangkat jaringan tradisional, strategi serangan yang berbeda terhadap situs host, serangan DDoS saat ini merupakan masalah keamanan yang paling sulit untuk diidentifikasi dan dilacak.

Serangan DDoS dapat dimitigasi dengan menerapkan standar keamanan yang ketat dan mengambil pendekatan yang tepat seperti *firewall* dan solusi khusus vendor. Untuk melindungi terhadap serangan DDoS, organisasi sering kali mengadopsi berbagai strategi, seperti menggunakan *firewall*, menerapkan sistem deteksi serangan, atau menggunakan layanan perlindungan DDoS yang disediakan oleh vendor. Hingga saat ini, sebagian besar penyedia konten besar seperti YouTube, Facebook, Twitter, dan Amazon mengalami gangguan layanan karena serangan DdoS [5].

Banyak peneliti menggunakan algoritma *machine learning* dan *deep learning* untuk mendeteksi serangan DDoS [6]. *Machine learning* dan *deep learning* lebih efektif dalam mendeteksi ancaman jaringan dibandingkan dengan sistem keamanan konvensional seperti firewall atau proxy. Dengan kemampuannya memahami pola kompleks dalam data, *machine learning* dan *deep learning* mendeteksi ancaman canggih dan berubah. Ini juga lebih fleksibel dalam menyesuaikan diri terhadap perubahan keamanan, mengurangi false positive, dan mendeteksi anomali. Pada penelitian yang menggunakan machine learning untuk deteksi DDoS, digunakan sejumlah data yang diekstraksi oleh para ahli untuk melatih dan menguji model guna meningkatkan kinerja. Sedangkan pada penelitian yang menggunakan *deep learning* untuk penelitian DDoS biasanya memungkinkan model untuk memilih fitur terbaik untuk melatih dan menguji model pada data tersebut [7].

Pemilihan fitur atau seleksi fitur merupakan langkah penting dalam pembelajaran mesin yang melibatkan pemilihan sekelompok fitur yang penting dari sekumpulan fitur yang lebih besar untuk meningkatkan efisiensi model [7]. Proses ini sangat mempengaruhi kinerja model dengan mengurangi kompleksitas dimensi, meningkatkan kemampuan generalisasi, mempercepat proses pembelajaran, dan meningkatkan kemampuan untuk menjelaskan model [7].

Pemilihan fitur yang tepat memiliki peran penting dalam mendeteksi serangan DDoS secara efektif. Penggunaan terlalu banyak fitur yang tidak terkait dapat meningkatkan jumlah komputasi [2]. Penelitian mengenai pengembangan sistem pembelajaran dinamis untuk mengidentifikasi serangan DDoS oleh lli Ko menggunakan *Complete Autoencoder* dengan seleksi fitur yang dinamis untuk memilih fitur-fitur yang penting dalam mengidentifikasi serangan DDoS serta memanfaatkan beberapa *Critical Modules* untuk meningkatkan kinerja sistem [8]. Hasil *F1 Score* yang didapatkan dari penggunaan seleksi fitur ini cukup baik, dengan rata-rata *F1 Score* di atas 97 persen. Penggunaan tiga *Critical Modules* dalam sistem juga meningkatkan *F1 Score* sebesar 6,48 persen. Hasil ini menunjukkan bahwa seleksi fitur yang dinamis dapat membantu meningkatkan kinerja sistem dalam mengidentifikasi serangan DDoS.

Penelitian mengenai "Klasifikasi DDoS yang Efisien dengan Pemilihan Fitur Ensemble" yang dilakukan oleh Singh K. mengusulkan pendekatan baru untuk deteksi serangan DDoS dengan memanfaatkan pemilihan fitur [2]. Penelitian tersebut menggunakan algoritma pemilihan fitur ensembel untuk mengidentifikasi atribut-atribut yang efisien dalam mengklasifikasikan kelas-kelas, mencapai tingkat akurasi 98,3 persen dalam klasifikasi, serta mengurangi beban komputasi dengan nilai *Root Mean Square Error* (RMSE) yang lebih rendah.

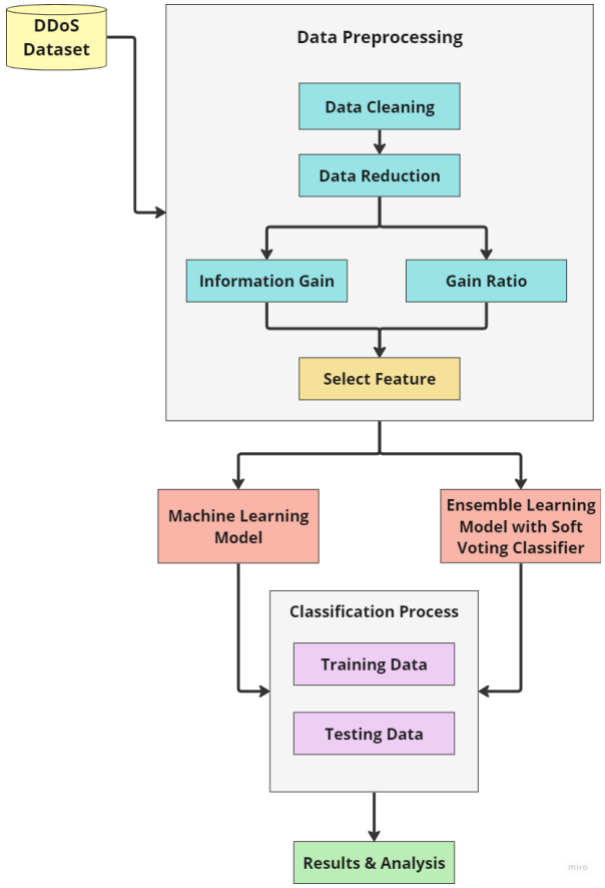
Penelitian mengenai metode deteksi serangan DDoS berdasarkan peningkatan K-Nearest Neighbour (KNN) dengan tingkat serangan DDoS pada *Software-Defined Network* oleh Dong dengan metode yang diusulkan yaitu DDoS *Detection Algorithm based on the Degree of Attack* (DDADA) dan DDoS *Detection Algorithm based on Machine Learning* (DDAML). Diperoleh nilai *True Rate Positive* (TPR) dengan nilai 0.987 dan 0.994 yang memberikan hasil lebih tinggi daripada nilai dari algoritma lain yang digunakan dan juga untuk nilai *False Rate Positive* (FPR) menghasilkan nilai 0.016 dan 0.009 yang lebih rendah dari nilai FPR dari algoritma lain. Nilai dari *Precision, Recall* dan *F-Measure* yang dihasilkan juga memiliki nilai yang lebih tinggi dari algoritma yang dikomparasikan. Peneliti menyimpulkan metode algoritma yang sudah ada untuk mendeteksi serangan DDoS masih memiliki nilai akurasi yang rendah dan rentan terhadap faktor lain. Hasil penelitian yang dilakukan oleh peneliti telah mencapai tingkat deteksi yang lebih tinggi dibandingkan dengan solusi algoritma yang sudah ada [9].

Penelitian mengenai "Pendekatan *Ensemble* untuk klasifikasi dan prediksi diabetes mellitus menggunakan *soft voting classifier*" yang dilakukan oleh Kumari et al. meningkatkan akurasi prediksi diabetes mellitus melalui penggunaan *Ensemble* pada metode *machine learning*, dengan Dataset Pima Indians Diabetes sebagai subjek eksperimen yang mencakup rincian pasien dengan dan tanpa diabetes. Melalui pendekatan *Ensemble* yang menggabungkan algoritma seperti *Random Forest*, *Logistic Regression*, dan *Naive Bayes* dengan *Soft Voting Classifier* menghasilkan hasil tertinggi dalam akurasi, *precision*, *recall*, dan nilai *F1 Score*, mencapai 79,04%, 73,48%, 71,45%, dan 80,6% secara berurutan. Efisiensi pendekatan ini juga terbukti melalui analisis pada dataset kanker payudara, di mana metode *Ensemble* dengan *Soft Voting Classifier* mencapai akurasi sebesar 97,02% [10].

Meskipun penelitian terdahulu telah mengimplementasikan metode *ensemble learning* dan *feature selection* dalam deteksi DDoS, namun masih ada peluang untuk menyelidiki lebih jauh efektivitas model dengan menerapkan teknik *soft voting classifier*. Upaya penelitian ini untuk mengembangkan model deteksi DDoS dengan menggunakan fitur-fitur yang telah diseleksi juga menunjukkan peluang untuk meningkatkan kinerja dan akurasi deteksi. Oleh karena itu, pada penelitian ini diusulkan metode *ensemble learning* dengan menggunakan teknik *soft voting classifier* dan *feature selection* pada penelitian ini.

II. METODE PENELITIAN

Bagian ini membahas data dan metode yang digunakan dalam penelitian ini, yaitu dimulai dari deskripsi data, *preprocessing*, teknik pemilihan fitur, metode klasifikasi, prosedur klasifikasi, dan analisis hasil. Alur kegiatan penelitian yang dilakukan ditunjukkan pada Gambar 1.



Gambar 1. Metode penelitian

A. *Data Preprocessing*

Data preprocessing adalah langkah awal yang penting dalam analisis data yang melibatkan penanganan data yang hilang dan penanggulangan ketidakkonsistenan. *Preprocessing* melibatkan serangkaian langkah berulang yang bertujuan mengubah data mentah menjadi bentuk yang dapat dipahami dan dimanfaatkan lebih lanjut. Data mentah seringkali memiliki karakteristik seperti ketidaklengkapan, ketidakkonsistenan, kurangnya struktur, dan mungkin mengandung kesalahan [11]. Tujuannya adalah untuk mempersiapkan data dengan cermat agar lebih optimal digunakan dalam tahap berikutnya. *Preprocessing* mencakup serangkaian tindakan seperti membersihkan data, mengatasi data yang hilang, normalisasi data, dan mengubah data. Hal ini dilakukan dengan tujuan supaya data yang telah diolah akan menghasilkan performa yang lebih baik dalam pelatihan model. Tahapan preprocessing dilakukan menggunakan bahasa pemrograman Python dengan memanfaatkan library Scikit-Learn. Langkah-langkah ini diawali dengan tahap *Data Cleaning*, *Data Reduction*, *Information Gain*, dan *Gain Ratio* untuk persiapan data sebelum dilakukan analisis lebih lanjut.

1) Data Cleaning: *Data cleaning* atau pembersihan data, merupakan langkah yang dilakukan untuk menghapus gangguan dari data yang tidak konsisten atau tidak memiliki relevansi. Tindakan pembersihan ini akan memiliki dampak pada kinerja teknik/metode data mining karena jumlah dan kompleksitas data yang dikelola akan berkurang [12]. Pada dataset yang digunakan, terdapat data yang bernilai *Not a Number* (NaN) dan nol. *Dataset* dengan nilai yang NaN dan nol dapat mempengaruhi evaluasi hasil. Dalam penelitian ini, kami menghilangkan data bernilai NaN dan nol. Jumlah data awalnya sebanyak 104345 *record* kemudian setelah proses *Data Cleaning* menjadi 103839 *record*.

2) Data Reduction: Pada *dataset* yang digunakan terdapat beberapa data yang tidak digunakan. Masalah yang sering terjadi dengan banyaknya dimensi adalah beban waktu komputasi seiring bertambahnya jumlah variabel [13]. Proses *data reduction* dirancang tidak hanya untuk mengurangi jumlah data tetapi juga untuk mengurangi *noise* [14]. Pada tahap ini dilakukan pengurangan semua data pada fitur yang tidak perlu. Data pada fitur yang dihilangkan pada tahap ini yaitu 'dt','src','dst', dan *'Protocol'*.

3) Information Gain dan Gain Ratio: *Information gain* menggunakan konsep teori informasi dimana jika semakin tinggi entropi, maka semakin banyak informasi pada fitur tersebut. *Information gain* menentukan fitur mana yang lebih berpengaruh dalam *dataset* untuk membedakan kelas yang akan dipelajari [2]. *Information gain* digunakan untuk membedakan kelas-kelas yang sedang dipelajari dengan mengidentifikasi fitur-fitur yang memiliki pengaruh paling signifikan. Proses pemilihan atribut yang akan berfungsi sebagai simpul, baik sebagai akar (*root*) atau simpul internal, menggunakan data atau nilai informasi tertinggi dari setiap atribut yang ada [15]. *Information gain* membantu peneliti dan analis data dalam mengambil keputusan yang lebih cerdas dan tepat dalam pemodelan dan analisis data dengan mengukur sejauh mana suatu fitur dapat memberikan informasi baru dan relevan terkait dengan klasifikasi data.
Rumus menghitung *Information Gain* sebagai berikut:

$$\text{Information Gain} = \text{Nilai Entropi Awal} - \text{Nilai Entropi Akhir}$$

Di mana:

- Entropi Awal: Tingkat ketidakpastian pada dataset sebelum membaginya berdasarkan fitur tertentu.
- Entropi Akhir: Entropi rata-rata dari masing- masing subset yang dihasilkan setelah pemisahan

Entropi(H) dapat dihitung menggunakan rumus seperti:

$$H(X) = -\sum[P(x) * log2(P(x))]$$

Di mana:

- H(X) adalah entropi dari variabel acak X.
- P(x) adalah proporsi dari sampel dalam kelas tertentu terhadap total sampel.

Gain ratio merupakan suatu modifikasi yang bertujuan untuk meningkatkan bias perolehan informasi pada fitur-fitur yang memiliki nilai keragaman signifikan [2]. *Gain ratio* bertujuan untuk menyediakan penilaian yang lebih adil terhadap kepentingan fitur-fitur dengan variasi nilai yang beragam dalam pemrosesan data. Pendekatan ini dimaksudkan untuk mencapai penilaian yang seimbang terhadap kontribusi berbagai fitur dalam analisis data, sehingga menghasilkan hasil yang lebih objektif dan informatif dalam konteks pemrosesan data.
Rumus untuk menghitung *Gain Ratio* adalah sebagai berikut:

$$Gain\ Ratio\ = \frac{Information\ Gain}{Split\ Information}$$

Di mana:

- *Information Gain* dihitung menggunakan rumus yang telah dijelaskan sebelumnya.
- *Split Information* merupakan ukuran banyaknya informasi yang dibutuhkan untuk membagi suatu kumpulan data berdasarkan fitur tertentu.

$$\text{Split Information}\ = \sum_{i=1}^{n}\left[\left(\frac{|D\_v|}{|D|}\right) * log2\left(\frac{|D\_v|}{|D|}\right)\right]$$

Di mana:

- D adalah kumpulan data.
- D_v adalah subset dari D yang sesuai dengan nilai-nilai yang berbeda dari atribut.
- n adalah jumlah cabang atau kategori pada suatu node.

4) Metode Pemilihan Fitur: Pemilihan fitur merupakan salah satu langkah penting dalam proyek pembelajaran mesin, juga dikenal sebagai pemilihan variabel dan atribut karena fokusnya pada atribut yang memiliki dampak paling signifikan terhadap variabel yang diprediksi. Pemilihan fitur yang efektif memastikan bahwa model menjadi lebih sederhana, memudahkan interpretasi oleh peneliti dan pengguna. Selain itu, dengan mengurangi jumlah variabel *input* yang digunakan dalam pengembangan model, proses ini dapat mengurangi biaya komputasi model dan meningkatkan kinerjanya [11]. Pemilihan fitur adalah proses memilih subset fitur yang relevan dan berpengaruh besar dari serangkaian fitur yang tersedia untuk dianalisis. Pemilihan fitur ini mengacu pada variabel atau atribut yang digunakan sebagai masukan pada model *machine learning* atau *ensemble learning*. Teknik pemilihan fitur memainkan peran penting dalam memilih fitur yang paling penting [16]. Pemilihan fitur menunjukkan bahwa 50% fitur *information gain* dan *gain ratio* teratas memberikan akurasi lebih tinggi dibandingkan teknik pemfilteran lainnya [17].

B. Metode

1) K-Nearest Neighbour (KNN): KNN adalah algoritma klasifikasi sederhana namun efektif dalam pembelajaran mesin. Struktur model KNN dibuat berdasarkan himpunan data dan tidak ada asumsi mengenai distribusi data yang mendasarinya sehingga model ini tidak memerlukan poin data pelatihan apa pun [18]. Algoritma ini digunakan untuk memprediksi klasifikasi sampel baru berdasarkan klasifikasi tetangga terdekatnya. KNN adalah jenis pembelajaran berbasis *instance*, juga dikenal sebagai pembelajaran malas, di mana fungsi hanya dievaluasi secara lokal dan semua komputasi ditunda hingga fungsi tersebut dievaluasi [5]. Berikut algoritma KNN:

1. Menentukan jumlah tetangga terdekat (K) yang akan digunakan dalam melakukan pemodelan.
2. Menghitung jarak antara titik data uji dengan seluruh titik data pelatihan menggunakan berbagai macam matrik jarak, seperti *Manhattan*, *Euclidean*, atau *Minkowski distance*.

   Rumus *Minkowski* distance:

$$Md(x,y)\ = (\sum_{i=1}^{n}|x_i - y_i|^p)^{1/p}$$

   Di mana:
   - p adalah parameter yang menentukan jenis *Minkowski distance* yang digunakan. Ketika p =1, Minkowski *distance* menjadi *Manhattan distance*, sementara ketika p=2, *Minkowski distance* menjadi jarak *Euclidean*.
3. Memilih K tetangga yang terdekat berdasarkan jarak yang telah dihitung.
4. Melakukan klasifikasi dengan menentukan kelas mayoritas di antara K tetangga yang terdekat sebagai prediksi kelas untuk titik data uji.

2) Logistic Regression: *Logistic regression* adalah bagian dari metode penambangan data yang digunakan untuk menganalisis data yang menggambarkan suatu variabel respon (terikat) atau beberapa variabel predictor [19]. *Logistic regression* pada dasarnya disebut salah satu algoritma klasifikasi dengan pembelajaran terawasi [20]. Tujuan utama dari *Logistic regression* adalah untuk memprediksi kemungkinan terjadinya suatu peristiwa dengan menghubungkannya dengan satu atau lebih variabel prediktor. Berikut algoritma *Logistic regression*:

1. Memecah data menjadi variabel independen (fitur) dan variabel dependen biner (target).
2. Inisialisasi bobot (koefisien) dan bias (intersep) dengan nilai acak atau nol.
3. Menerapkan model *Logistic regression*.
4. Melatih model dengan penyesuaian bias dan bobot.
5. Melakukan prediksi dari hasil melatih model dan akan menghasilkan probabilitas bahwa variabel dependen dengan nilai 1.
6. Mengatur *Threshold* Untuk mengubah probabilitas prediksi menjadi klasifikasi biner (0 atau 1).

Rumus *Logistic regression*:

$$P(Y=1) = \frac{1}{\left(1 + e\text{^}\left(-(b0 \; + \; b1 * X1 \; b2 * X2 + \ldots + bn * Xn)\right)\right)}$$

Di mana:

- P(Y=1) adalah probabilitas variabel dependen (target) bernilai 1.
- e = bilangan *Eular* (sekitar 2.71828).
- b0 = bias (*intersep*).
- b1, b2, ..., bn adalah bobot (koefisien) untuk masing-masing fitur X1, X2, ..., Xn.

3) Random Forest: *Random forest* adalah salah satu algoritma pembelajaran yang digunakan untuk klasifikasi, regresi, dan tugas lain yang memerlukan analisis prediktif [21]. Ini adalah jenis algoritma agregasi yang menggabungkan beberapa pohon keputusan independen untuk meningkatkan kinerja dan mengurangi *overfitting*. *Decision tree* memiliki kecenderungan untuk menyesuaikan set pelatihannya secara berlebihan, yang dikoreksi oleh *Random forest* [5]. Berikut algoritma *Random forest:*

1. Menetapkan jumlah pohon (*n_estimators*).
2. Melakukan *Bagging,* membuat data latih yang diambil secara acak dari data latih asli dengan penggantian.
3. Membangun *Decision tree* yang menggunakan data latih hasil *Bagging* hingga kriteria penghentian tercapai.
4. Hasil seluruh *Decision tree* di-*ensemble* untuk menghasilkan hasil akhir.

Pada tahap setiap *decision tree* dilakukan perhitungan *GINI impurity* untuk setiap pemisahan nilai atribut. Ini melibatkan menghitung *GINI impurity* untuk setiap nilai atribut yang unik dan kemudian menghitung rata-rata pondered *GINI impurity* dari setiap pemisahan atribut.

Rumus *GINI impurity*:

$$GINI \; = \; 1 - \sum_{i=1}^{C} (Pi)^2$$

Di mana:

- C adalah jumlah kelas target
- pi adalah proporsi contoh dengan kelas tertentu terhadap total contoh.

4) Naive Bayes: *Naive bayes* adalah suatu algoritma untuk melakukan tugas klasifikasi yang mengacu pada teorema probabilitas *Bayes* dengan asumsi sederhana yang dikenal sebagai "asumsi naif" atau "*naive assumption*". Algoritma ini bekerja berdasarkan *teorema Baye*s yang menyatakan sifat mana yang benar. Konstruksinya sederhana karena algoritma tidak mengevaluasi parameter. Hal ini memungkinkannya bekerja pada kumpulan data yang sangat besar [5]. *Naive Bayes Classifier* (NBC) merupakan metode klasifikasi probabilistik yang berdasarkan Teori *Bayes* dan juga algoritma pembelajaran probabilitas yang berasal dari Teori Keputusan Bayesian. Dengan demikian, NBC menjadi suatu pendekatan sederhana namun efektif untuk klasifikasi berdasarkan probabilitas dan pengetahuan dari data yang ada [22]. Berikut algoritma *Naive Bayes*:

1. Inisialisasi probabilitas kelas berdasarkan frekuensi kemunculan kelas dalam data pelatihan.
2. Hitung probabilitas fitur untuk setiap kelas. Dalam kasus variabel berkelanjutan, distribusi fitur dapat diasumsikan sebagai distribusi normal (*Gaussian*).
3. Hitung probabilitas *posterior* menggunakan *teorema Bayes* dengan asumsi naif bahwa semua fitur adalah saling independen.
4. Pilih kelas dengan probabilitas *posterior* tertinggi sebagai prediksi.

Rumus pada algoritma *Naive Bayes*:

$$P(C|X) \; = \; \frac{(P(X|C) \; * \; P(C))}{P(X)}$$

Di mana:

- P(C|X) adalah probabilitas kelas C, diberikan fitur X.
- P(X|C) adalah probabilitas fitur X diberikan kelas C.
- P(C) adalah probabilitas prior kelas C.
- P(X) adalah probabilitas prior fitur X.

5) Multi Layer Perceptron: *Multilayer Perceptron (MLP)* adalah suatu bentuk jaringan saraf tiruan yang memiliki struktur terstruktur dan mengadopsi metode pembelajaran *supervised* [23]. Jaringan ini terdiri dari lapisan-lapisan yang saling terhubung dan dilatih dengan data yang memiliki label, memungkinkannya untuk memahami pola dan hubungan dalam tugas pembelajaran yang diawasi. MLP merupakan jenis arsitektur jaringan saraf tiruan yang terdiri dari setidaknya tiga lapisan: lapisan *input*, satu atau lebih lapisan tersembunyi, dan lapisan *output*. *Perceptron* tunggal hanya dapat menyelesaikan masalah linier, namun MLP lebih cocok untuk contoh non-linier [24].

6) Ensemble *Method dengan Soft Voting Classifier*: *Ensemble method* adalah teknik yang menggabungkan beberapa model *machine learning* untuk meningkatkan performa dan ketahanan satu model. *Ensemble learning* melibatkan penggunaan beberapa algoritma untuk mendapatkan nilai rata-rata akurasi [25]. Salah satu jenis metode agregasi yang umum digunakan adalah *soft voting. Soft voting* merupakan suatu metode yang mana hasil prediksi dari beberapa model yang berbeda diambil sebagai rata-rata atau probabilitas prediksi dari masing-masing model diberi bobot dan prediksi akhir didasarkan pada mayoritas hasil prediksi prediksi model yang berbeda. *Ensemble method* dengan menggunakan *soft voting* merupakan model yang mengintegrasikan sejumlah algoritma klasifikasi yang beragam [16].

Adapun langkah-langkah pada *Ensemble method* dengan *Soft Voting Classifier*:

1. Proses pelatihan: Model pembelajaran mesin yang berbeda dilatih pada kumpulan data yang sama.

2. Proses Pengujian: Setelah model dilatih, setiap model akan digunakan untuk membuat prediksi pada kumpulan data pengujian.
3. Proses *Soft Voting*: Hasil prediksi setiap model di agregasi dengan merata-ratakan hasil prediksi atau probabilitas prediksi setiap model. Prediksi akhir kemudian sebagian besar didasarkan mayoritas dari hasil prediksi model yang berbeda.

III. HASIL DAN PEMBAHASAN

Penelitian ini diawali dengan tahap *data cleaning* dan *data reduction*. *Dataset* yang digunakan dalam penelitian ini menggunakan dataset DDoS yang diperoleh dari situs kaggle.com. *Dataset* ini terdapat 23 atribut dan sudah termasuk label kelas. Jumlah data yang digunakan pada penelitian ini sebanyak 104.345 *record*. Atribut yang terdapat pada dataset dapat ditunjukkan pada Tabel I.

TABEL I
NAMA ATRIBUT

No	Nama Atribut
1	dt
2	switch
3	src
4	dst
5	pktcount
6	bytecount
7	dur
8	dur_nsec
9	tot_dur
10	flows
11	packetins
12	pktperflow
13	byteperflow
14	pktrate
15	pairflow
16	protocol
17	port_no
18	tx_bytes
19	rx_bytes
20	tx_kbps
21	rx_kbps
22	tot_kbps
23	label

Pada tahap ini dilakukan penghilangan *record* yang bernilai *null*. Pada tahap ini data awal sebanyak 104.345 *record* berkurang menjadi 103.839 *record*. Pada tahap yang sama, dilakukan proses pemilihan atribut yang digunakan yaitu sebanyak 19 atribut termasuk label kelas. Terjadinya pengurangan atribut dt, src, dst, dan *protocol* dikarenakan data tidak berpengaruh pada proses *training* dan *testing*.

Pada tahap selanjutnya, dilakukan perhitungan *information gain* dan *gain ratio*. Perhitungan dilakukan dengan tujuan untuk melakukan seleksi fitur 50% tertinggi dari semua atribut data. Data *information gain* dari semua atribut dapat dilihat pada Tabel II.

Setelah data *information gain* telah diperoleh, maka dilakukan proses pengurutan data sesuai nilai *information gain* tertinggi. Pemilihan fitur dilakukan dengan menggunakan hanya 50% fitur yang memiliki *information gain* tertinggi. Proses pemilihan fitur ini dapat mempercepat waktu pelatihan data. Data atribut yang memiliki *information gain* tertinggi ditunjukkan pada Tabel III.

TABEL II
HASIL INFORMATION GAIN PADA SEMUA ATRIBUT

Nama Atribut	Information Gain
switch	0.0041
pktcount	0.6160
bytecount	0.6334
dur	0.1977
dur_nsec	0.0685
tot_dur	0.2668
flows	0.0291
packetins	0.1972
pktperflow	0.5409
byteperflow	0.5567
pktrate	0.3850
Pairflow	0.0119
port_no	0.0052
tx_bytes	0.2090
rx_bytes	0.1810
tx_kbps	0.0714
rx_kbps	0.0746
tot_kbps	0.1071

TABEL III
HASIL INFORMATION GAIN TERTINGGI PADA 9 ATRIBUT

Nama Atribut	Information Gain
bytecount	0.6333
pktcount	0.6160
byteperflow	0.5582
pktrate	0.3868
tot_dur	0.2662
tx_bytes	0.2087
dur	0.1957
packetins	0.1924

Pada tahap yang sama, dilakukan perhitungan *gain ratio*. Perhitungan dilakukan dengan tujuan untuk melakukan seleksi fitur 50% tertinggi dari semua atribut data. Data *gain ratio* dari semua atribut ditunjukkan pada Tabel IV.

Setelah data *gain ratio* telah diperoleh, dilakukan proses yang sama seperti *information gain* yaitu memilih 50% fitur tertinggi dari semua atribut. Data atribut yang memiliki *gain ratio* tertinggi ditunjukkan pada Tabel V.
Pada proses klasifikasi pada *ensemble learning* dilakukan proses kombinasi dari 5 metode yang digunakan. Kombinasi yang dilakukan pada *ensemble learning* ditunjukkan pada Tabel VI.

Setelah tahap pemilihan fitur, langkah berikutnya yaitu melakukan proses klasifikasi pada setiap metode yang dipaparkan pada penelitian ini. Hasil dari proses klasifikasi data berdasarkan fitur yang memiliki *information gain* terbaik ditunjukkan pada Tabel VII. Pada tahap proses yang sama, dilakukan juga proses klasifikasi menggunakan fitur yang memiliki *gain ratio* terbaik ditunjukkan pada Tabel VIII.

TABEL IV
HASIL GAIN RATIO PADA SEMUA ATRIBUT

Nama Atribut	Information Gain
switch	0.0012
pktcount	0.0775
bytecount	0.0787
dur	0.0312
dur_nsec	0.0105
tot_dur	0.0414
flows	0.0123
packetins	0.0461
pktperflow	0.0997
byteperflow	0.0100
pktrate	0.0967
Pairflow	0.0016
port_no	0.0004
tx_bytes	0.0307
rx_bytes	0.0287
tx_kbps	0.0224
rx_kbps	0.0224
tot_kbps	0,0259

TABEL V
HASIL GAIN RATIO TERTINGGI PADA 9 ATRIBUT

Nama Atribut	Information Gain
byteperflow	0.1001
pktperflow	0.0997
pktrate	0.0967
bytecount	0.0787
switch	0.0775
packetins	0.0461
tot_dur	0.0414
tx_bytes	0.0307

Pada tahap yang sama, dalam penelitian ini juga melakukan proses klasifikasi data tanpa menggunakan pemilihan fitur. Hasil dari proses klasifikasi data ditunjukan pada Tabel IX.

Berdasarkan data pada Tabel VIII dan Tabel IX yang diperoleh, dapat diperhatikan bahwa proses pemilihan fitur sangat penting dalam proses klasifikasi. Dengan adanya pemilihan fitur, dapat meningkatkan hasil akurasi dari setiap metode.

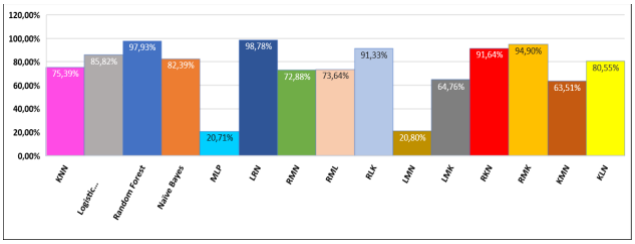
Pada metode KNN mengalami peningkatan akurasi sebesar 9,06% dengan pemilihan fitur menggunakan *information gain* dan *gain ratio*. Metode *logistic regression* mengalami penurunan akurasi sebesar 0,32% dengan pemilihan fitur menggunakan *information gain* dan *gain ratio*. Metode *random forest* juga mengalami penurunan akurasi sebesar 0,05% dengan pemilihan fitur menggunakan *information gain* dan *gain ratio*. Pada metode *naive bayes* mengalami peningkatan akurasi sebesar 2,07% dengan pemilihan fitur menggunakan *information gain* dan *gain ratio*. Pada metode MLP mengalami penurunan akurasi sebesar 0,47% dengan pemilihan fitur menggunakan *information gain* dan mengalami peningkatan yang sangat signifikan sebesar 57,41% pada *gain ratio*.

TABEL VI
KOMBINASI METODE PADA ENSEMBLE LEARNING

Ensemble Method dengan 3 Metode
Logistic Regression, *Random Forest*, *Naive Bayes* (LRN)
Random Forest, MLP, *Naive Bayes* (RMN)
Random Forest, MLP, *Logistic Regression* (RML)
Random Forest, *Logistic Regression*, KNN (RLK)
Logistic Regression, MLP, *Naive Bayes* (LMN)
Logistic Regression, MLP, KNN (LMK)
Random Forest, KNN, *Naive Bayes* (RKN)
Random Forest, MLP, KNN (RMK)
KNN, MLP, *Naive Bayes* (KMN)
KNN, *Logistic Regression*, *Naive Bayes* (KLN)

TABEL VII
HASIL PENGUJIAN METODE DENGAN SELEKSI FITUR BERDASARKAN INFORMATION GAIN

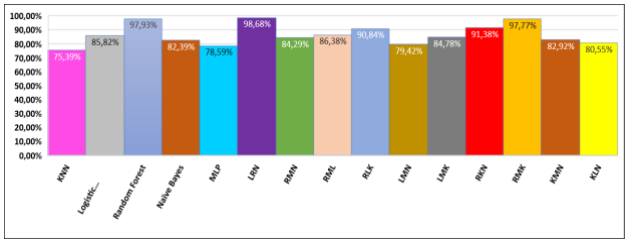
METHOD	ACCURACY
KNN	75,39 %
Logistic Regression	85,82 %
Random Forest	97,93 %
Naïve Bayes	82,39 %
MLP	20,71 %
Soft Voting Classifier using Logistic Regression, RandomForest, NaiveBayes (LRN)	**98,78 %**
Soft Voting Classifier using RandomForest, MLP, NaiveBayes (RMN)	72 88 %
Soft Voting Classifier using RandomForest, MLP, Logistic Regression (RML)	73,64 %
Soft Voting Classifier using RandomForest, Logistic Regression, KNN (RLK)	91,33 %
Soft Voting Classifier using Logistic Regression, MLP, NaiveBayes (LMN)	20,80 %
Soft Voting Classifier using Logistic Regression, MLP, KNN (LMK)	64,76 %
Soft Voting Classifier using RandomForest, KNN, NaiveBayes (RKN)	91,64 %
Soft Voting Classifier using RandomForest, MLP, KNN (RMK)	94,90 %
Soft Voting Classifier using KNN, MLP, Naïve Bayes (KMN)	63,51 %
Soft Voting Classifier using KNN, Logistic Regression, Naïve Bayes (KLN)	80,55 %



Gambar 2. Grafik hasil akurasi metode dengan seleksi fitur berdasarkan information gain

TABEL VIII
HASIL PENGUJIAN METODE DENGAN SELEKSI FITUR BERDASARKAN GAIN RATIO

METHOD	ACCURACY
KNN	75,39 %
Logistic Regression	85,82 %
Random Forest	97,93 %
Naïve Bayes	82,39 %
MLP	78,59 %
Soft Voting Classifier using Logistic Regression, RandomForest, NaiveBayes (LRN)	**98,68 %**
Soft Voting Classifier using RandomForest, MLP, NaiveBayes (RMN)	84,29 %
Soft Voting Classifier using RandomForest, MLP, Logistic Regression (RML)	86,38 %
Soft Voting Classifier using RandomForest, Logistic Regression, KNN (RLK)	90,84 %
Soft Voting Classifier using Logistic Regression, MLP, NaiveBayes (LMN)	79,42 %
Soft Voting Classifier using Logistic Regression, MLP, KNN (LMK)	84,78 %
Soft Voting Classifier using RandomForest, KNN, NaiveBayes (RKN)	91,38 %
Soft Voting Classifier using RandomForest, MLP, KNN (RMK)	97,77 %
Soft Voting Classifier using KNN, MLP, Naïve Bayes (KMN)	82,92 %
Soft Voting Classifier using KNN, Logistic Regression, Naïve Bayes (KLN)	80,55 %

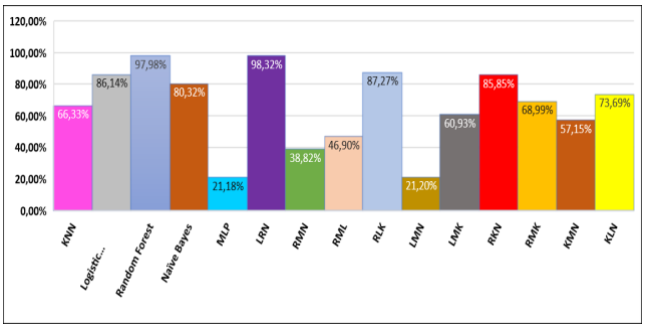


Gambar 3. Grafik hasil akurasi metode dengan seleksi fitur berdasarkan gain ratio

Hasil klasifikasi data menggunakan *ensemble method* dengan lima metode tunggal yang digunakan pada penelitian ini dilakukan kombinasi setiap *ensemble method* menggunakan tiga metode sehingga terdapat sepuluh kombinasi metode. Dari hasil penelitian dapat diperhatikan bahwa perbedaan cukup signifikan antara tanpa pemilihan fitur dan menggunakan pemilihan fitur pada *ensemble method*. Hampir sebagian besar kombinasi metode mengalami peningkatan yang sangat besar baik pada pemilihan fitur yang menggunakan *information gain* maupun pemilihan fitur yang menggunakan *gain ratio*. Selain pemilihan fitur, dapat diperhatikan bahwa *ensemble method* memiliki akurasi yang lebih baik pada hampir sebagian besar kombinasi jika dibandingkan hanya menggunakan metode *machine learning* tunggal dalam proses klasifikasi DDoS.

TABEL IX
HASIL PENGUJIAN METODE TANPA SELEKSI FITUR

METHOD	ACCURACY
KNN	66,33 %
Logistic Regression	86,14 %
Random Forest	97,98 %
Naïve Bayes	80,32 %
MLP	21,18 %
Soft Voting Classifier using Logistic Regression, RandomForest, NaiveBayes (LRN)	**98,32 %**
Soft Voting Classifier using RandomForest, MLP, NaiveBayes (RMN)	38,82 %
Soft Voting Classifier using RandomForest, MLP, Logistic Regression (RML)	46,90 %
Soft Voting Classifier using RandomForest, Logistic Regression, KNN (RLK)	87,27 %
Soft Voting Classifier using Logistic Regression, MLP, NaiveBayes (LMN)	21,20 %
Soft Voting Classifier using Logistic Regression, MLP, KNN (LMK)	60,93 %
Soft Voting Classifier using RandomForest, KNN, NaiveBayes (RKN)	85,85 %
Soft Voting Classifier using RandomForest, MLP, KNN (RMK)	68,99 %
Soft Voting Classifier using KNN, MLP, Naïve Bayes (KMN)	57,15 %
Soft Voting Classifier using KNN, Logistic Regression, Naïve Bayes (KLN)	73,69 %



Gambar 4. Grafik hasil akurasi metode tanpa seleksi fitur

IV. KESIMPULAN

Dari hasil dan pembahasan pada penelitian DDoS dapat disimpulkan bahwa *ensemble method* menjadi sangat optimal dalam proses deteksi DDoS jika dibandingkan dengan metode *machine learning* tunggal. Hal tersebut dapat dilihat dari hasil yang menunjukkan hampir semua kombinasi *ensemble method* memiliki akurasi di atas metode *machine learning* tunggal. Hasil tertinggi yang diperoleh menggunakan *soft voting classifier* dengan kombinasi metode *Logistic Regression*, *Random Forest*, *Naïve Bayes*, yaitu sebesar 98,32%. Hasil terendah yang diperoleh menggunakan *soft voting classifier* dengan kombinasi metode *Logistic Regression*, *Multi Layer Perceptron* dan *Naïve Bayes*, yaitu sebesar 21,20%. Dengan adanya seleksi fitur *information gain* dan *gain ratio* pada *soft voting classifier* dengan kombinasi metode

Logistic Regression, *Random Forest* dan *Naïve Bayes* mengalami peningkatan masing-masing sebesar 0,46% dan 0,36%.

Dengan adanya proses pemilihan fitur dapat meningkatkan akurasi pada beberapa metode *machine learning* tunggal. Akan tetapi, dengan adanya pemilihan fitur pada *ensemble method* menjadikan peningkatan akurasi yang besar untuk hampir semua kombinasi metode baik menggunakan fitur berdasarkan *information gain* maupun menggunakan fitur berdasarkan *gain ratio*.

REFERENSI

[1] A. Maheshwari, B. Mehraj, M. S. Khan, and M. S. Idrisi, "An optimized weighted voting based ensemble model for DDoS attack detection and mitigation in SDN environment," *Microprocess Microsyst*, vol. 89, 2022, doi: 10.1016/j.micpro.2021.104412.

[2] K. J. Singh and T. De, "Efficient Classification of DDoS Attacks Using an Ensemble Feature Selection Algorithm," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 71–83, Jan. 2020, doi: 10.1515/jisys-2017-0472.

[3] T. Khempetch and P. Wuttidittachotti, "Ddos attack detection using deep learning," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, 2021, doi: 10.11591/ijai.v10.i2.pp382-388.

[4] A. Jaszcz and D. Połap, "AIMM: Artificial Intelligence Merged Methods for flood DDoS attacks detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, 2022, doi: 10.1016/j.jksuci.2022.07.021.

[5] M. Aslam *et al.*, "Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT†," *Sensors*, vol. 22, no. 7, 2022, doi: 10.3390/s22072697.

[6] "A Ddos Attack Categorization and Prediction Method Based on Machine Learning," *Journal of Population Therapeutics and Clinical Pharmacology*, vol. 30, no. 9, 2023, doi: 10.47750/jptcp.2023.30.09.030.

[7] J. Zhao, M. Xu, Y. Chen, and G. Xu, "A DNN Architecture Generation Method for DDoS Detection via Genetic Alogrithm," *Future Internet*, vol. 15, no. 4, 2023, doi: 10.3390/fi15040122.

[8] I. Ko, D. Chambers, and E. Barrett, "Adaptable feature-selecting and threshold-moving complete autoencoder for DDoS flood attack mitigation," *Journal of Information Security and Applications*, vol. 55, 2020, doi: 10.1016/j.jisa.2020.102647.

[9] S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN with the Degree of DDoS Attack in Software-Defined Networks," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2019.2963077.

[10] S. Kumari, D. Kumar, and M. Mittal, "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier," *International Journal of Cognitive Computing in Engineering*, vol. 2, 2021, doi: 10.1016/j.ijcce.2021.01.001.

[11] I. D. Acheme and O. R. Vincent, "Machine-learning models for predicting survivability in COVID-19 patients," in *Data Science for COVID-19 Volume 1: Computational Perspectives*, 2021. doi: 10.1016/B978-0-12-824536-1.00011-3.

[12] R. G. Gunawan, Erik Suanda Handika, and Edi Ismanto, "Pendekatan Machine Learning Dengan Menggunakan Algoritma Xgboost (Extreme Gradient Boosting) Untuk Peningkatan Kinerja Klasifikasi Serangan Syn," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 3, pp. 453–463, Dec. 2022, doi: 10.37859/coscitech.v3i3.4356.

[13] D. Uhm, S.-H. Jun, and S.-J. Lee, "A Classification Method Using Data Reduction," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 1, 2012, doi: 10.5391/ijfis.2012.12.1.1.

[14] S. An, Q. Hu, C. Wang, G. Guo, and P. Li, "Data reduction based on NN-kNN measure for NN classification and regression," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 3, 2022, doi: 10.1007/s13042-021-01327-3.

[15] B. Prasojo and E. Haryatmi, "Analisa Prediksi Kelayakan Pemberian Kredit Pinjaman dengan Metode Random Forest," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 7, no. 2, pp. 79–89, Sep. 2021, doi: 10.25077/teknosi.v7i2.2021.79-89.

[16] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble Learning Methods for Anomaly Intrusion Detection System in Smart Grid," in *IEEE International Conference on Electro Information Technology*, 2021. doi: 10.1109/EIT51626.2021.9491891.

[17] P. Nimbalkar and D. Kshirsagar, "Feature selection for intrusion detection system in Internet-of-Things (IoT)," *ICT Express*, vol. 7, no. 2, 2021, doi: 10.1016/j.icte.2021.04.012.

[18] J. Kusuma, Rubianto, R. Rosnelly, Hartono, and B. H. Hayadi, "Klasifikasi Penyakit Daun Pada Tanaman Jagung Menggunakan Algoritma Support Vector Machine, K-Nearest Neighbors dan Multilayer Perceptron," *Journal of Applied Computer Science and Technology*, vol. 4, no. 1, 2023, doi: 10.52158/jacost.v4i1.484.

[19] F. R. Suprihati, "Analisis Klasifikasi SMS Spam Menggunakan Logistic Regression," *Jurnal Sistem Cerdas*, vol. 4, no. 3, 2021, doi: 10.37396/jsc.v4i3.166.

[20] A.- Amrin and O.- Pahlevi, "Implementation of Logistic Regression Classification Algorithm and Support Vector Machine for Credit Eligibility Prediction," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 5, no. 2, 2022, doi: 10.31289/jite.v5i2.6220.

[21] F. Hamami and I. A. Dahlan, "KLASIFIKASI CUACA PROVINSI DKI JAKARTA MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN TEKNIK OVERSAMPLING," *Jurnal Teknoinfo*, vol. 16, no. 1, 2022, doi: 10.33365/jti.v16i1.1533.

[22] H. Mustofa and A. A. Mahfudh, "Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes," *Walisongo Journal of Information Technology*, vol. 1, no. 1, 2019, doi: 10.21580/wjit.2019.1.1.3915.

[23] S. Sudianto, A. D. Sripamuji, I. Ramadhanti, R. R. Amalia, J. Saputra, and B. Prihatnowo, "Penerapan Algoritma Support Vector Machine dan Multi-Layer Perceptron pada Klasisifikasi Topik Berita," *Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI*, vol. 11, no. 2, pp. 84–91, 2022, [Online]. Available: https://ejournal.undiksha.ac.id/index.php/janapati/article/view/44151

[24] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective Heart Disease Prediction Using Machine Learning Techniques," *Algorithms*, vol. 16, no. 2, 2023, doi: 10.3390/a16020088.

[25] A. M. Siregar, "Klasifikasi Untuk Prediksi Cuaca Menggunakan Esemble Learning," *PETIR*, vol. 13, no. 2, 2020, doi: 10.33322/petir.v13i2.998.