BAB II

LANDASAN TEORI

2.1 Intrusion Detection System (IDS)

Intrusion Detection System (IDS) adalah proses pemantauan, mendeteksi, dan menganalisis peristiwa yang dianggap sebagai pelanggaran terhadap kebijakan keamanan lingkungan jaringan memperkenalkan konsep mendeteksi serangan berbasis cyber pada jaringan komputer dengan menyediakan kerangka kerja untuk sistem deteksi intrusi (IDS), yang didasarkan pada hipotesis bahwa pelanggaran keamanan dapat dideteksi dengan memantau catatan audit sistem untuk pola *abnormal system* (Ashfaq, Wang, Huang, Abbas, & He, 2017).

Instrusion detection system (IDS) terdapat beberapa langkah pendekatan menurut (Urvashi & Jain, 2015). Langkah-langkah tersebut adalah sebagai berikut:

1. *Data Collection*, mengumpulkan lalu lintas jaringan menggunakan perangkat lunak tertentu dan dengan demikian membantu untuk mendapatkan informasi tentang lalu lintas seperti jenis paket, host dan rincian protokol.
2. *Feature Selection*, deteksi intrusi berbasis jaringan menghasilkan beberapa informasi header IP, yakni sumber dan alamat IP tujuan, tipe paket, tipe layer 4 protokol dan *flags* lainnya.
3. *Analysis,* data yang terkumpul dianalisi untuk menentukan apakah data tersebut dikategorikan anomali atau tidak.
4. *Actions,* Intrusion Detection System (IDS) memberikan peringatan pada administrator sistem bahwa serangan telah terjadi dan memberitahu sifat

serangan dan mengendalikan serangan dengan menutup port jaringan atau mematikan proses.

2.2 Distributed Denial of Services (DDoS)

Serangan *Denial of Services* (DoS) dan *Distributed Denial of Services* (DDoS) menjadi masalah yang sangat serius untuk keamanan di Internet. Tujuan utama dari serangan tersebut adalah untuk mengganggu layanan dengan membanjiri lalu lintas yang tidak perlu melalui jaringan. DDoS relatif sederhana tetapi menjadi salah satu jenis serangan yang paling kuat. Teknik penyerangan DDoS berupa mengganggu layanan pengguna yang sah dengan menghabiskan sumber daya server seperti: memori, *Central Processing Unit* (CPU), soket dan *bandwidth* (Mehmood, Mukherjee, Ahmed, Song, & Malik, 2018).

2.3 Machine Learning

Machine Learning adalah bidang ilmu komputer yang melibatkan studi dan konstruksi teknik yang memungkinkan komputer untuk belajar mandiri berdasarkan data input untuk memecahkan masalah spesifik (Hoang & Nguyen, 2018).

Jenis-jenis permasalahan yang umumnya diselesaikan dengan pendekatan *Machine learning* adalah klasterisasi dan klasifikasi. Klasterisasi adalah aktivitas yang bertujuan mengelompokan data berdasarkan kedekatan fitur yang dimilikinya, sedangkan klasifikasi bertujuan untuk memisahkan data menjadi kelas-kelas tertentu. Perbedaan yang mendasar antara 2 buah permasalahan ini adalah, pada proses klasterisasi, data-data dikelompokan tanpa pelabelan, sedangkan klasifikasi mengelompokan data-data menjadi label tertentu (Utama, 2018).

2.4 Data Mining

2.4.1 Definisi Data Mining

Data mining adalah proses yang menggunakan teknik statistik, perhitungan, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terakit dari berbagai basis data besar (Zainul Efendy dan Azizel Wanjas Saputra Genda, 2018).

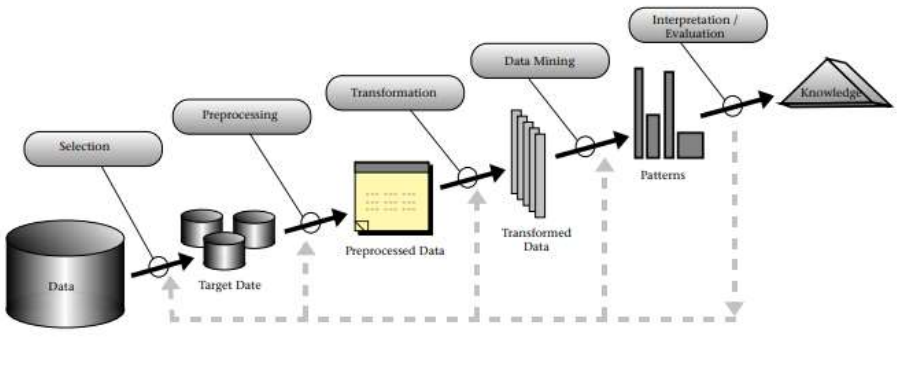
Hasil dari pengolahan data dengan metode data mining ini dapat digunakan untuk mengambil keputusan di masa depan. Data mining ini juga dikenal dengan istilah pattern *recognition* (Heni & Irham Gufroni, 2017). Berikut ini beberapa definisi data mining dari beberapa sumber (T.Larose, 2014) :

- Data mining adalah proses menemukan sesuatu yang bermakna dari suatu korelasi baru, pola dan tren yang ada dengan cara memilah-milah data berukuran besar yang disimpan dalam repositori, menggunakan teknologi pengenalan pola serta teknik matematika dan statistik.
- Data mining adalah analisis pengamatan database untuk menemukan hubungan yang tidak terduga dan untuk meringkas data dengan cara atau metode baru yang dapat dimengerti dan bermanfaat kepada pemilik data.
- Data mining merupakan bidang ilmu interdisipliner yang menyatukan teknik pembelajaran dari mesin (machine learning), pengenalan pola (*pattern recognition*), statistik, *database*, dan visualisasi untuk mengatasi masalah ekstraksi informasi dari basis data yang besar.

- Data mining diartikan sebagai suatu proses ekstraksi informasi berguna dan potensial dari sekumpulan data yang terdapat secara implisit dalam suatu basis data.

2.4.2 Tahapan Data Mining

Tahapan yang dilakukan pada proses data mining diawali dari seleksi data dari data sumber ke data target, tahap preprocessing untuk memperbaiki kualitas data, transformasi, data mining serta tahap interpretasi dan evaluasi yang menghasilkan output berupa pengetahuan baru yang diharapkan memberikan kontribusi yang lebih baik. Secara detail dijelaskan pada gambar 2.1 (Fayyad, Piatetsky-Shapiro, & Smyth, 2015).



Gambar 2.1 Tahapan Data Mining
(Fayyad et al., 2015)

1. Data Selection

   Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing* / Cleaning

Sebelum proses data mining dapat dilaksanakan, perlu dilakukan proses cleaning pada data yang menjadi fokus KDD. Proses cleaning mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.

3. *Tranformation*

Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses coding dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. Data Mining

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation / Evaluation*

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut interpretation. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.5 Lazy Classifier

Lazy Classifier adalah teknik pembelajaran dimana teori melewati data pengaturan ditunda sampai pertanyaan dibuat ke sistem data latih seluruhnya pada informasi persiapan sebelum mendapatkan pertanyaan. Keuntungan menggunakan metodologi *Lazy Classifier* adalah ekstensi target akan diklaim secara lokal, misalnya pada *K-Nearest Neighbour* (KNN) (Kumar, Zinovyev, Verma, & Tiwari, 2018).

2.6 K-Nearest Neighbour

K-Nearest Neighbour (KNN) adalah salah satu pengklasifikasi *Machine Learning* sederhana yang bekerja dengan baik dalam klasifikasi. *K-Nearest Neighbour* (KNN) adalah salah satu dari *Lazy Classifier* (Razak et al., 2018).

KNN membuat prediksi menggunakan dataset pelatihan secara langsung. Prediksi dibuat untuk titik data baru dengan mencari melalui seluruh rangkaian pelatihan untuk *instance K-Nearest Neighbour* dan merangkum variabel output untuk instance k tersebut. Regresi ini mungkin variabel output rata-rata dalam klasifikasi bernilai mode atau paling umum.

Contoh k dalam dataset pelatihan yang paling mirip ditentukan dengan input baru, maka digunakan pengukuran jarak. Variabel input bernilai nyata, ukuran jarak paling populer adalah *Euclidean Distance*. *Euclidean Distance* dihitung sebagai akar kuadrat dari jumlah perbedaan kuadrat antara titik a dan titik b di semua atribut input i yang dijelaskan di rumus 1.

$$Euclidean\ Distance(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$ ..... (Rumus 1)

a adalah data training, b adalah data testing, euclidean distance (a,b) merupakan jarak, i adalah variabel data, n adalah dimensi data (Brownlee, 2017)..

2.7 Naive Bayes

Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Algoritma mengunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas.

Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Keuntungan penggunaan Naive Bayes adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (Training Data) yang kecil untuk menentukan estimasi paremeter yang diperlukan dalam proses pengklasifikasian pada rumus 2 .

$$P(H|X) = \frac{P(X|H).P(H)\ P(X)}{P(X)}$$ .....(Rumus 2)

X adalah data dengan class yang belum diketahui, H adalah hipotesis data suatu class spesifik, P(H|X) adalah pobabilitas hipotesis H berdasar kondisi X (posteriori probabilitas), P(H) adalah probabilitas hipotesis H (prior probabilitas),

P(X|H) adalah probabilitas X berdasarkan kondisi pada hipotesis H, P(X) adalah Probabilitas X (Reza El Akbar, Shofa, Paripurna, & Supratman, 2019).

2.8 Confusion Matrix

Confusion matrix dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari *TruePositive* dan *TrueNegative* memberikan informasi ketika classifier dalam melakukan klasifikasi data bernilai benar, sedangkan *FalsePositive* dan *False-Negative* memberikan informasi ketika classifier salah dalam melakukan klasifikasi data.

TP (*True Positive*) merupakan Jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif, FP (*False Positive*) merupakan Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif, FN (*False Negative*) merupakan Jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif, TN (*True Negative*) merupakan Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif (Fibrianda & Bhawiyuga, 2018).

2.9 Parameter *Metric*

Accuracy merupakan tingkat keterhubungan antara suatu nilai yang diprediksi dengan nilai aktual yang ada (Devita et al., 2018). Berikut merupakan rumus atau formula dari *accuracy* dijelaskan pada rumus 5.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Negative + False\ Positive + True\ Negative}$$.(Rumus 3)

Precision merupakan pengukuran tingkat ketepatan antara informasi yang diminta oleh pemohon dengan jawaban yang diberikan oleh sistem. Rumus atau formula dari *precision* dijelaskan pada rumus 3 (Yunus, Widiastuti, Rasjid, & Chalr, 2019).

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive}$$ ..... (Rumus 4)

Recall merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi dalam suatu pemrosesan data. Berikut merupakan rumus atau formula dari *recall* dijelaskan pada rumus 4 (Yunus et al., 2019)

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative}$$ ..... (Rumus 5)

Sensitivity merupakan proses yang digunakan untuk mengukur pecahan pola positif yang diklasifikasian dengan benar (Pristyanto, Pratama, & Nugraha, 2018).

$$Sensitivity = \frac{True\ Positive}{True\ Positive+False\ Negative}$$ ..... (Rumus 6)

Specificity merupakan proses yang digunakan untuk mengukur pecahan pola negatif yang diklasifikasikan dengan benar (Pristyanto et al., 2018).

$$Specificity = \frac{True\ Negative}{True\ Negative+False\ Positive}$$ ..... (Rumus 7)

Error Rate merupakan kesalahan klasifikasi mengukur rasio prediksi yang salah atas jumlah total contoh yang dievaluasi (Devita et al., 2018).

$$Error\ Rate = \frac{False\ Positive+False\ Negative}{True\ Positive+False\ Positive+True\ Negative+False\ Negative}$$ (Rumus 8)

G-means merupakan indikator evaluasi performa dari algoritma (Pristyanto et al., 2018).

$$G\text{-}means = \frac{False\ Positive + False\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \quad \text{.... (Rumus 9)}$$

2.10 State Of The Art

Penelitian lain tentang klasifikasi *anomaly network traffic* dilakukan perbandingan metode yang dihasilkan dari proses klasifikasi berdasarkan nilai akurasi *confusion matrix, precision, recall*, dan f1 *score*. Naive Bayes, *Support Vector Machine* (SVM) Linear, SVM Polynomial dan SVM Sigmoid menghasilkan persentase akurasi berturut-turut sebesar 85,055%, 99,995%, 99,999%, dan 99,995%. Persentase akurasi tertinggi diperoleh SVM Polynomial, sedangkan Naive Bayes menghasilkan persentase akurasi terendah (Fibrianda & Bhawiyuga, 2018).

Algoritma Naive Bayes dapat menghasilkan akurasi yang maksimal dengan data latih yang sedikit. Metode *K-Nearest Neighbour* dipilih karena metode tersebut tangguh terhadap data noise. Hasil yang didapatkan menunjukkan metode Naive Bayes memiliki kinerja yang lebih baik dengan tingkat akurasi 70%, sedangkan metode K-Nearest Neighbor memiliki tingkat akurasi yang cukup rendah yaitu 40% (Devita, Herwanto, & Wibawa, 2018)

Algoritma *K-Nearest Neighbour* memiliki akurasi yang cukup tinggi dibandingkan dengan algoritma *Support Vector Machine* (SVM) dan Neural Network (NN) untuk kategori *accuracy, precison* dan *recall*. Hasil tersebut menunjukkan bahwa algoritma *K-Nearest Neighbour* dapat memecah data dalam

keadaan *higher-feature space* sehingga dua kelas yang berbeda dapat dikelompokkan dengan baik (Doshi, Apthorpe, & Feamster, 2018)

2.7 Studi Literatur

Tabel 2.1 Studi Literatur

No	Peneliti/ Tahun	Judul	Problem	Metode/Algoritma	Kesimpulan
1	J. Jabez, S. Gowri, S. Vigneshwari, J. Albert Mayan and Senduru Srinivasulu [2018]	*Anomaly Detection by Using CFS Subset and Neural Network with WEKA Tools*	Kurangnya fitur *Intrusion Detection System* (IDS) berbasis jaringan menjadi deteksi yang lebih rendah, terutama dalam kasus serangan frekuensi rendah dan hubungan yang buruk dari deteksi anomali	Algoritma : Neural Network	Sistem IDS (*Instrusion Detective System*) yang diusulkan membutuhkan waktu lebih sedikit untuk eksekusi dan menyimpan tes dalam dataset daripada sistem IDS (*Instrusion Detective System*) yang ada .
2	Yuya Kunugi, Hiroyuki Suzuki, Akio Koyoma [2019]	*IoT Security Viewer System Using Machine Learning*	Alat yang dikembangkan memiliki banyak jenis software untuk diperkenalkan dan visualisasi topologi jaringan tidak dilakukan, sehingga ada masalah bahwa kelainan visual sesaat tidak dapat dikenali.	Algoritma : Random forest	Mengembangkan 12ystem yang dapat mendeteksi kelainan oleh mesin belajar, memvisualisasikan topologi jaringan, dan noti fi es kelainan dengan visualisasi peringatan pada topologi jaringan

Tabel 2.1 Studi Literatur (lanjutan)

No	Peneliti/ Tahun	Judul	Problem	Metode/Algoritma	Kesimpulan
3.	Dominik Breitenbacher, Ivan Homoliak, Yan Lin Aung, Nils Ole Tippenhauer, Yuval Elovici [2019]	*HADES-IoT: A Practical Host-Based Anomaly Detection System for IoT Devices (Extended Version)*	tidak ada metode didirikan atau standar untuk mengukur dan memastikan keamanan perangkat IOT	Metode : Host-Based Anomaly Detection	Mengusulkan HADES-IOT, sistem deteksi anomali berbasis host untuk perangkat IOT, yang menyediakan deteksi proaktif dan ketahanan tamper-proof.
4.	Zhi-Juan Jia, Ning Wang, Yun-Ye Wang, Ming-Sheng Hu [2018]	*The traceability analysis and research of Botnet control center based on ant colony group-dividing algorithm*	Ancaman keamanan jaringan tumbuh seperti. Virus, Worm, Trojan, Phishing dan serangan jaringan serta ancaman keamanan jaringan lainnya telah mendapat perhatian tinggi di seluruh dunia. Di antara ancaman keamanan, Botnet telah menjadi salah satu ancaman paling serius terhadap Internet	Algoritma : ant colony group-dividing, ant colony	Algoritma *ant colony group dividing* dapat meningkatkan efektivitas implementasinya. Oleh karena itu, algoritma ant colony group-dividing dapat menemukan jalur serangan yang tepat dan menemukan pusat kendali botnet.

Tabel 2.1 Studi Literatur (lanjutan)

No	Peneliti/ Tahun	Judul	Problem	Metode/Algoritma	Kesimpulan
5,	Mohd Faizal Ab Razak, · Nor Badrul Anuar, · Fazidah Othman, Ahmad Firdaus, F. Afifi1, Rosli Salleh [2017]	*Bio-inspired for Features Optimization and Malware Detection*	Data sensitif pada perangkat seluler Android menimbulkan ancaman serius bagi pengguna, dan serangan yang berbaahaya	Metode : Bio-inspired Algoritma : Random forest, J48, K-nearest neighbors, multilayer perceptron, AdaBoost	Hasil percobaan menunjukkan tingkat deteksi 95,6% untuk TPR menggunakan classifier AdaBoost pada sampel malware Drebin yang dianalisis menggunakan optimasi fitur PSO.
6.	Ahmad Al-Nawasrah, Ammar Al-Momani, F. Meziane, M. Alauthman [2018]	*Fast Flux Botnet Detection Framework using Adaptive Dynamic Evolving Spiking Neural Network Algorithm*	Botnet sebagai perangkat mesin yang dikendalikan jauh oleh penyerang adalah dasar dari berbagai ancaman keamanan seluruh dunia.	Algoritma : Adaptive Dynamic Evolving Spiking Neural Network	Fast Flux Killer System (FFKS) yang memiliki kemampuan untuk mendeteksi FF-Domains dalam mode online dengan implementasi yang dibangun pada Adaptive Dynamic yang berkembang
7.	Rohan Doshi, Noah Apthorpe, Nick Feamster [2018]	*Machine Learning DDoS Detection for Consumer Internet of Things Devices*	Meningkatnya jumlah perangkat *Internet of Things* (IOT) terhubung ke Internet, namun banyak dari perangkat ini pada dasarnya tidak aman	Metode : Anomali detection Algoritma : Neural Network, KNN, LSVM, Decision tree, Random Forest	Algoritna Neural Network, K-nearest neighbors, Linear support vector machine, Decision tree, Random Forest memiliki set tes akurasi yang lebih tinggi hingga 0.99.

Tabel 2.1 Studi Literatur (lanjutan)

No	Peneliti/ Tahun	Judul	Problem	Metode/Algoritma	Kesimpulan
8.	Xuan Dau Hoang, Quynh Chi Nguyen [2018]	*Botnet Detection Based On Machine Learning Techniques Using DNS Query Data*	Botnet telah menjadi salah satu ancaman utama terhadap keamanan informasi karena mereka terus berkembang dalam ukuran dan kecanggihan.	Metode : DNS query Algoritma : Random forest	Hasil percobaan dengan algoritma forest efektif dalam deteksi botnet dan menghasilkan akurasi deteksi keseluruhan terbaik lebih dari 90%.
9.	Reem Alhajri, Rachid Zagrouba, Fahd Al-Haidari [2019]	*Survey for Anomaly Detection of IoT Botnets Using Machine Learning Auto-Encoders*	Botnet dapat mengembangkan serangan DDoS dan menghadirkan masalah keamanan utama di jaringan IoT, karena tidak ada metode tunggal yang menunjukkan potensi untuk mengatasi ancaman keamanan tersebut	Algoritma : Auto-Encoders	Auto-encoders menawarkan salah satu solusi untuk botnet deteksi.
10.	C. D. McDermott, F. Majdani, Andrei V. Petrovski [2018]	*Botnet Detection in the Internet of Things using Deep Learning Approaches*	Pabrikan IoT lebih mementingkan harga murah dan mengabaikan kemanan sehingga rentan dan mudah dieksploitasi.	Metode : Word Embedding Algoritma : BLSTM-RNN, LSTM-RNN	Hasil untuk mirai, udp, dan dns sangat bagus dengan akurasi validasi 99%, 98%, 98% dan masing-masing 0,000809, 0,125630, 0,116453 metrik kehilangan validasi.

Tabel 2.1 Studi Literatur (lanjutan)

No	Peneliti/ Tahun	Judul	Problem	Metode/Algoritma	Kesimpulan
11.	Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, Yuval Elovici [2018]	*N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders*	Pertumbuhan perangkat IoT yang dibiarkan begitu saja daripada komputer desktop telah menyebabkan peningkatan serangan botnet berbasis IoT.	Metode : Deep autoencoders	Hasil evaluasi menunjukkan kemampuan metode deep autoencoders dapat mendeteksi serangan secara langsung dan akurat saat diluncurkan dari perangkat IoT yang dibiarkan karena merupakan bagian dari botnet.
12.	Hamdija Sinanovic, Sasa Mrdovic [2017]	*Analysis of Mirai Malicious Software*	Mirai adalah salah satu contoh tentang seberapa besar masalah ketidakamanan IoT. Hal tersebut dapat menambah wawasan tentang perangkat lunak berbahaya yang ada dihampir setiap perangkat lunak biasa.	Metode : Analisis statis dan analisis dinamis	Berdasarkan analisis dan kesesuaian perilaku jaringannya yang sederhana, dimungkinkan untuk membuat identitas IDS untuk semua bagian pengoperasian Mirai. Hal tersebut dapat menjadi salah satu cara terbaik dan termudah untuk mendeteksi dan menghentikan Mirai.

Tabel 2.1 Studi Literatur (lanjutan)

No	Peneliti/ Tahun	Judul	Problem	Metode/Algoritma	Kesimpulan
13.	Riri Nada Devita, Heru Wahyu Herwanto, Aji Prasetya Wibawa (2018)	Perbandingan Kinerja Metode Naive Bayes Dan *K-Nearest Neighbour* Untuk Klasifikasi Artikel Berbahasa Indonesia	Kecocokan isi artikel dengan sebuah tema jurnal menjadi faktor utama diterima tidaknya sebuah artikel. Tetapi masih banyak mahasiswa yang bingung untuk menentukan jurnal yang sesuai dengan artikel yang dimilikinya.	Algoritma : K-Nearest Neighbour, Naive Bayes	Hasil yang didapatkan menunjukkan metode Naive Bayes memiliki kinerja yang lebih baik dengan tingkat akurasi 70%, sedangkan metode *K-Nearest Neighbor* memiliki tingkat akurasi yang cukup rendah yaitu 40%..
14.	Vitor Hugo Bezerra, Victor G. Turrisi da Costa, Sylvio Barbon Junior, Rodrigo Sanches Miani, Bruno Bogaz Zarpelao [2018]	*One-class Classification to Detect Botnets in IoT devices*	Dengan meningkatnya jumlah perangkat Internet of Things yang berbeda, ancaman baru terhadap keamanan jaringan muncul karena keamanan yang rendah dari perangkat-perangkat.	Algoritma : *One-class Support Vector Machine* (OSVM), *Support Vector Machine* (SVM)	*One-class Support Vector Machine* (OSVM), hanya memerlukan waktu enam menit dari data yang sah untuk menginduksi model dan mampu mendeteksi semua botnet dalam pengaturan yang berbeda menggunakan jendela waktu 1 detik

Tabel 2.1 Studi Literatur

No	Peneliti/ Tahun	Judul	Problem	Metode/Algoritma	Kesimpulan
15.	Amjad Mehmood, Mithun Mukherjee, Syed Hassan Ahmed, Houbing Song, Khalid Mahmood Malik [2018]	*NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks*	IoT rentan terhadap berbagai jenis ancaman keamanan sama seperti jaringan kabel dan nirkabel lainnya.	Algoritma : Naive Bayes	mekanisme NB-MAIDS yang diusulkan adalah sistem canggih untuk deteksi intrusi dalam jaringan. Algoritma klasifikasi Naïve Bayes dengan praktik beberapa agen untuk deteksi serangan DDoS memberikan kinerja yang lebih baik dibandingkan dengan IDS yang digunakan secara tradisional.
16.	Mercury Fluorida Fibrianda, ,Adhitya Bhawiyuga (2018)	Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan *Support Vector Machine* (SVM)	Serangan Denial of Service (DoS) merupakan suatu tindakan untuk melumpuhkan server komputer pada jaringan internet sehingga komputer tidak dapat menjalankan fungsinya dengan benar.	Algoritma : Naïve Bayes, *Support Vector Machine* (SVM)	Perbandingan yang dihasilkan berdasarkan nilai akurasi confusion matrix, precision, recall, dan f1 score. Naive Bayes, SVM Linear, SVM Polynomial dan SVM Sigmoid menghasilkan sebesar 85,05%, 99,95%, 99,99%, dan 99,95%. Persentase akurasi tertinggi diperoleh SVM Polynomial, sedangkan Naive Bayes menghasilkan persentase akurasi terendah.

Tabel 2.2 Matriks Penelitian

No	Peneliti (Tahun)	Judul	Ruang Lingkup							
			Parameter *Metric*					Algoritma		
			Accuracy	Precision	Recall	Sensitivity	Specificity	KNN	NB	SVM
1.	Mercury Fluorida Fibrianda, ,Adhitya Bhawiyuga (2018)	Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM)	✓	✓	✓	-	-	-	✓	✓
2.	Riri Nada Devita, Heru Wahyu Herwanto, Aji Prasetya Wibawa (2018)	Perbandingan Kinerja Metode Naive Bayes Dan K-Nearest Neighbor Untuk Klasifikasi Artikel Berbahasa Indonesia	✓	✓	✓	-	-	✓	✓	-

Tabel 2.2 Matriks Penelitian (Lanjurtan)

No	Peneliti (Tahun)	Judul	Ruang Lingkup							
			Parameter *Metric*					Algoritma		
			Accuracy	Precision	Recall	Sensitivity	Specificity	KNN	NB	SVM
3.	Rohan Doshi, Noah Apthorpe, Nick Feamster (2018)	Machine Learning DDoS Detection for Consumer Internet of Things Devices	✓	✓	✓	-	-	✓	-	✓
4.	Aditya Dwi Afifaturahman Usulan penelitian	Perbandingan Algoritma K-Nearest Neighbour (Knn) Dan Naive Bayes Menggunakan Parameter Metric Accuracy, Sensitivity Dan Specificity Pada Intrusion Detection System (IDS)	✓	✓	✓	✓	✓	✓	✓	-