BAB II

LANDASAN TEORI

2.1 Analisis *Malware*

Malicous Software atau *Malware* adalah sebuah program yang dirancang khusus untuk melakukan sebuah aktifitas yang dapat membahayakan perangkat lunak pada perangkat korban seperti *Trojan*, *Virus*, *Spyware* dan *Exploit* (Kramer dan Bradfield 2010). *Malware* dapat melakukan aktifitas berbahaya yang memberikan dampak signifikan bagi para korbannya, diantaranya yaitu penyadapan serta pencurian data informasi pribadi, hingga kasus perusakan sistem yang dilakukan oleh penyusup (*Intruder*) terhadap perangkat korban dengan berbagai alasan.

Tipe analisis dalam melakukan analisis pada *malware* terbagi menjadi dua yaitu dengan analisis statis (analisa kode) dan analisis dinamis (analisa perilaku), meskipun dari kedua tipe analisis tersebut mempunyai tujuan yang sama yaitu menjelaskan tentang bagaimana sebuah *malware* bekerja namun peralatan, waktu dan kemampuan yang dibutuhkan dalam menganalisa sangatlah berbeda (Cahyanto, Wahanggara, dan Ramadana 2017). Penjelasan dua tipe analisis tersebut adalah sebagai berikut:

1. Analisis Statis

   Metode analisis statis ini file *malware* tidak akan diaktifkan secara langsung melainkan ditelusuri dan diteliti serta dianalisis terhadap kode sumber yang dituliskan didalam program *malware* dengan melakukan tahapan pembedahan terhadap program *malware*.

Proses memeriksa biner yang diberikan tanpa mengeksekusi sebagian besar dilakukan secara manual. Sebagai contoh jika kode sumber tersedia beberapa informasi menarik, seperti struktur data, fungsi yang digunakan dan grafik panggilan dapat diekstraksi. Informasi ini hilang setelah kode sumber telah dikompilasi menjadi biner yang dapat dieksekusi dan dengan demikian menghambat analisis lebih lanjut. Domain *malware* biasanya yang terakhir adalah kasusnya, karena kode sumber dari biner *malware* saat ini biasanya tidak tersedia. Berbagai teknik digunakan untuk analisis *malware* statis (Syaputra 2020).

2. Analisis Dinamis

File yang diperiksa akan diaktifkan dalam sebuah lingkungan yang safe baik pada sebuah mesin fisik yang telah disediakan sebagai laboratorium *malware* maupun yang berupa virtual (mesin virtual) untuk selanjutnya mampu dikumpulkan infor- masi mengenai dampaknya terhadap komputer ketika file *malware* menjalankan prosesnya, sehingga dapat diketahui kegiatan apa saja yang dilakukan oleh *malware* saat berhasil menginfeksi sebuah komputer. Tahapan dalam analisis dinamis ini akan memeriksa komputer dengan secara keseluruhan seperti proses yang berjalan di komputer, perubahan registry, komunikasi internet dan peristiwa janggal lainnya yang memungkinkan terjadi ketika sebuah komputer telah terinfeksi oleh *malware* (Syaputra 2020).

2.2 *Machine Learning*

Machine Learning (ML) atau Mesin Pembelajar adalah cabang dari *Artificial Intelligence* yang fokus belajar dari data (*learn from data*), yaitu fokus

pada pengembangan sistem yang mampu belajar secara mandiri tanpa harus berulang kali diprogram manusia. ML membutuhkan data yang valid sebagai bahan belajar (ketika proses training) sebelum digunakan ketika testing untuk hasil output yang optimal (Cholissodin dkk., 2020).

Machine learning dapat dilakukan dengan dua cara, yaitu *supervised learning* dan *unsupervised learning*. *Unsupervised learning* yaitu pemrosesan sampel data yang dilakukan tanpa mewajibkan hasil akhir memiliki bentuk yang sesuai dengan bentuk tertentu, dengan menggunakan beberapa sampel data sekaligus. Penerapan *unsupervised* learning dapat ditemukan pada proses visualisasi, atau eksplorasi data. Sebaliknya, dalam *supervised learning*, sampel data x akan diproses sedemikian rupa, sehingga menghasilkan bentuk keluaran yang sesuai dengan hasil akhir y. *Supervised learning* dapat diterapkan pada proses klasifikasi (van Heeswijk, 2015)

2.3 Klasifikasi

Klasifikasi adalah teknik utama dalam memisahkan data dan banyak digunakan di berbagai bidang. Definisi sederhana dari klasifikasi dapat dikatakan sebagai proses menganalisis sekumpulan data dan menghasilkan sekumpulan aturan pengelompokan yang dapat digunakan untuk mengklasifikasikan data yang baru (Sanjaya, Setyati, dan Budianto 2020).

Proses klasifikasi didasarkan pada empat komponen (Gorunescu, 2011).

1. *Kelas*

   Variabel dependen yang berupa kategorikal yang merepresentasikan ‘label’ yang terdapat pada objek.

Contohnya: resiko penyakit jantung, resiko kredit, customer loyalty, jenis gempa.

2. *Predictor*

   Variabel independen yang direpresentasikan oleh karakteristik (atribut) data.

3. *Training dataset*

   Satu set data yang berisi nilai dari kedua komponen di atas yang digunakan untuk menentukan kelas yang cocok berdasarkan predictor.

4. *Testing dataset*

   Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi.

2.4 Algoritma *K-Nearest Neighbour*

KNN adalah algoritma yang meminimalkan jumlah fungsi tujuan kuadrat kesalahan. KNN memiliki konsistensi yang baik untuk proses klasifikasi (Ding dan He 2004). Kinerja pengklasifikasi KNN ditentukan oleh pilihan dari K serta metrik jarak yang diterapkan. Umumnya, nilai K yang lebih besar membuat batas-batas lebih halus antar kelas (Yang dkk., 2007). Algoritma KNN diharapkan dapat digunakan untuk proses klasifikasi ancaman serangan *malware* yang lebih efektif dan akurat.

Algoritma K-Nearest Neighbor bersifat sederhana, bekerja dengan berdasarkan kemiripan dari sampel uji (testing sample) ke sampel latih (training sample) untuk menentukan *K-Nearest Neighbor*-nya (Siringoringo, 2016). K-Nearest Neighbor dilakukan dengan mencari kelompok k objek dalam data training

yang paling dekat (mirip) dengan objek pada data baru atau data testing (Leidiyana, 2010). KNearest Neighbor merupakan teknik klasifikasi yang sederhana, tetapi mempunyai hasil kerja yang cukup bagus (Sibarani, 2015) Secara umum untuk mendefinisikan jarak antara dua objek x dan y, digunakan rumus jarak Euclidean pada persamaan 2.1 (Mustafa, 2014):

$$dxy = \sqrt{\sum} (xi - yi)\ n\ 2\ i=1 \quad (2.1)$$

dimana matriks D(a,b) adalah jarak skalar dari kedua vektor a dan b dari matriks dengan ukuran d dimensi. Algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample pada fase *training*. Fitur-fitur yang sama dihitung untuk testing data (yang klasifikasinya tidak diketahui) pada fase klasifikasi. Jarak dari vektor baru yang ini terhadap seluruh vektor training sample dihitung dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut. Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus dimana klasifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, k = 1) disebut algoritma K-Nearest Neighbor. Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar

performa klasifikasi menjadi lebih baik. KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training data-nya besar, sedangkan kelemahan KNN adalah KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat), training berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample* (Yofianto, 2010).

2.5 Darknet

Darknet adalah ruang IP address yang tidak digunakan yang tidak berspekulasi untuk berinteraksi dengan komputer lain di dunia. Komunikasi dari ruang gelap dianggap skeptis karena sifat pendengarannya yang pasif yang menerima paket masuk, tetapi paket keluar tidak didukung. Lalu lintas apa pun dianggap tidak terpikirkan dan secara khas diperlakukan sebagai *probe*, *backscatter*, atau kesalahan konfigurasi karena tidak adanya *host* yang sah di darknet. Darknet juga dikenal sebagai teleskop jaringan, *sinkholes*, atau *blackholes* (Rathod 2017).

Klasifikasi lalu lintas Darknet sangat penting untuk mengkategorikan aplikasi *real-time*. Menganalisis lalu lintas darknet membantu dalam pemantauan awal *malware* sebelum serangan dan deteksi aktivitas berbahaya setelah wabah. Pekerjaan penelitian Arash dkk mengusulkan teknik baru untuk mendeteksi dan mengkarakterisasi aplikasi VPN dan Tor bersama-sama sebagai perwakilan nyata dari lalu lintas darknet dengan menggabungkan dua kumpulan data publik, yaitu,

ISCXTor2016 dan ISCXVPN2016, untuk membuat kumpulan data darknet lengkap yang masing-masing mencakup lalu lintas Tor dan VPN (Kaur dkk.,, 2020).

2.6 *State of The Art* Penelitian

Tantangan penelitian dijawab dengan memperluas *state-of-the-art* bidang penelitian dengan memasukan unsur-unsur tambahan yang dapat memenuhi kebutuhan Intelligence Forensic. Tabel II.1 menunjukan perbandingan penelitian yang berhubungan dengan fokus pada kontribusi dan batasan dalam proses analisa malware.

Tabel II.1. *State of The Art* Penelitian Terkait

Tabel II. 1 *State of The Art* Penelitian Terkait

No	Nama Pengarang	Tahun	Judul	Masalah	Solusi
1.	Fikri Bahtiar, Nur Widiyasono, Aldy Putra	2018	Memory Volatile Forensik Untuk Deteksi *Malware* Menggunakan Algoritma Machine Learning	Penggunaan volatilitas membutuhkan pengetahuan tentang alat baris perintah (Commdan Line) serta analisis *malware* statis . Sebagian Alat Forensik yang berfungsi mendeteksi *malware* secara otomatis, tetapi harus selalu terhubung dengan internet, dan	1. Pengguna dapat mendeteksi semua proses yang berjalan pada memory volatile dan tidak harus terkoneksi dengan internet dan Pengguna tidak perlu mengingat perintah, sintaknya atau bahkan

				deteksi *malware* yang dilakukan terbatas.	ketika mau menggunakan perintah mana 2. Pengukuran akurasi algoritma GNB 69.952916 % , decision tree 98.913437 %, rdanom forest 99.406012 % , adaptive boss 138.047 98.540384 % , gradient boosting 138.047 98.790293 %
2.	Vinna Rahmayanti Setyaning Nastiti, Denar Regata Akbi, Achmad Rizal Yogaswara	2020	Klasifikasi *Malware* Family Menggunakan Metode K-Nearest Neighbor	Perkembangan *malware* yang semakin pesat, menyebabkan banyak jenis keluarga *malware* baru yang bermunculan. Untuk mendeteksi jenis dari suatu *malware*, salah satu teknik yang dapat digunakan adalah dengan	Klasifikasi varian *malware* dengan menggunakan KNN memperoleh hasil recall 45 % dan precission 41 % dengan menggunakan dataset CICInves&Mal 2019.

				klasifikasi *malware* menggunakan machine learning.	
3.	Muhammad Misbahul Azis	2020	Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan Software Defined Network	Serangan DDoS meningkat jumlahnya dan semakin kompleks pada setiap jenis yang diserangannya. Pada penelitian sebelumnya tentang Detection of Distributed Denial of Service Attack in Sofware Defined Network masih belum ada tindakan apa yang akan dilakukan setelah benar menemukan penyerang.	Menggunakan DDoS dengan tipe TCP Flood Attack pada jaringan Software Defined Network (SDN) untuk dilakukan DDoS mitigasi di jaringan SDN dengan menggunakan K-Nearest Neighbors (KNN) pada controller. Meningkatkan akurasi dalam mendeteksi penyerang sehingga identitas dari penyerang dapat terlihat oleh KNN dengan akurasi 0,999 dan f-1 score 0,999 menggunakan Dataset CICIDS 2017.
4.	Hilmi Hafid	2019	Investigasi Log Jaringan untuk Deteksi Serangan DDOS Menggunakan	Bagaimana memanfaatkan Genreal Regression Neural Network dalam	Memanfaatkan GRNN dalam menginvestigasi log jaringan

			Metode General Regression Neural Network.	menginvestigasi log jaringan untuk mendeteksi serangan DDOS. Seberapa akurat General Regression Neural Network dalam mendeteksi serangan DDoS berdasarkan data latih dan data uji yang digunakan.	untuk mendeteksi serangan DDoS Mengukur tingkat akurasi GRNN dalam mendeteksi serangan DDoS berdasarkan data latih dan data uji yaitu 97,21%, presisi 97,21%, recall 97,19%, f1score 97,2% dengan dataset CICIDS2017.
5.	Erick Lamdompak	2016	Klasifikasi *Malware* Trojan Ransomware Dengan Algoritma Support Vector Machine (SVM).	Perkembangan trojan ransomware yang sulit dideteksi karena menumpang dalam file lain.	Menggunakan Support Vector Machine untuk klasifikasi *malware* dan normal file sebagai pembdaning dataset *malware* dan dataset file normal.
6.	Harsono, Muhammad Chambali, Arif Wirawan Muhammad	2018	Klasifikasi Paket Jaringan Berbasis Analisis Statistik dan Neural Network	Serangan DDoS dengan memfungsikan metode SYN Flood merupakan salah satu contoh semakin berkembangnya teknik serangan DDoS. Dalam aktivitas	Dengan menggunakan algoritma neural network sebagai basis classifier menghasilkan nilai rerata persentase akurasi klasifikasi sebesar 92,99%.

				jaringan Internet, aliran paket data yang memanfaatkan protokol SYN merupakan sebuah paket jaringan yang bersifat legal, karena protokol SYN mutlak diperlukan dalam proses otentikasi komunikasi antar perangkat dalam jaringan Internet.	Dataset DDoS Attack 2007 yang dirilis oleh Center for Applied Internet Data Analysis (CAIDA).
7.	I Komang Kompyang Agus Subrata, I Made Oka Widyantara, Linawati	2017	Klasifikasi Penggunaan Protokol Komunikasi Pada Trafik Jaringan Menggunakan Algoritma K-Nearest Neighbor.	Meningkatnya trafik data yang dapat menyebabkan penurunan performansi jaringan terutama pada jaringan yang memiliki bdanwidth terbatas. Mengetahui penggunaan protokol komunikasi jaringan, sehingga dapat menjadi dasar untuk penentuan prioritas suatu trafik jaringan.	Klasifikasi K-NN memiliki tingkat keakuratan yang sangat tinggi. Hal ini dibuktikan dengan hasil perhitungan yang mencapaiinilai 99,14 % yaitu dengan perhitungan k = 3
8.	Fransiskus Gusti, Ngurah Dwika	2017	Pendeteksian *Malware* pada Lingkungan	Mendeteksi file berbahaya pada serangan terhadap aplikasi web	Membdaningkan algoritma Naïve Baiyes dan Decision Tree

	Royyana Muslim Ijtihadi, Hudan Studiawan		Aplikasi Web dengan Kategorisasi Dokumen.	dengan NaiveBaiyes dan Decision Tree.	untuk mendeteksi malicious file pada aplikasi web.
9.	Kumar	2019	Deep in the Dark: A Novel Threat Detection System using Darknet Traffic.	Bagaimana mendeteksi Ancaman *Malware* dengan menggunakan dataset SURFnet pada traffic darknet dengan ML.	Mengukur akurasi algoritma Random Forest 0,98167 , LightGBM 0,99904, KNN 0,98630 dengan dataset Surfnet
10.	Yan Li dan Yufei Li	2021	ETCC : Encrypted Two-Label Classification Using CNN	Melakukan klasifikasi trafik darknet dengan menggunakan algoritma CNN dan LSTM pada dataset CICDarknet 2020.	Mengukur akurasi CNN yaitu 0,966 dan akurasi LSTM yaitu 0,952

Beragam proses analisa *malware* yang dilakukan dengan berbagai cara dan algoritma yang berbeda memiliki nilai akurasi yang berbeda – beda, namun pada setiap perbdaningan algoritma *Machine Learning* KNN dengan yang lain didapatkan akurasi algoritma KNN lebih baik. Masalah selanjutnya dari penelitian sebelumnya yaitu perlunya dataset yang memiliki lebih banyak fitur dan terbaru agar proses analisis klasifikasi lebih akurat dengan perkembangan *malware* saat ini.

Berdasarkan permasalahan tersebut pada penelitian ini akan digunakan algoritma KNN untuk melakukan klasifikasi dan prediksi ancaman serangan *malware* pada analisis file pcap trafik darknet dengan dataset yang terbaru yaitu CIC Darknet 2020. Hasil

penelitian akan menilai akurasi dan efektifitas penerapan algoritma KNN dengan Dataset CICDarknet 2020 pada file pcap trafik darknet.

2.7 Matriks Penelitian

Tabel II. 2 Tabel Matriks Penelitian

No.	Penulis/Tahun	Judul	Ruang Lingkup									
			Teknik Analisis		Penerapan ML						Tujuan		Objek	
			Realtime	Non-Realtime	KNN	RF	GRNN	NB	SVM	NN	Klasifikasi	Prediksi	*Malware*	Trafik Jaringan
1.	(Bahtiar, Widiyasono, dan Aldya 2018)	Memory Volatile Forensik Untuk Deteksi *Malware* Menggunakan	-	√	-	√	-	-	-	-	√	√	√	-

		Algoritma Machine Learning												
2.	(Rahmayanti dkk., 2020)	Klasifikasi *Malware* Family Menggunakan Metode K-Nearest Neighbor	-	√	√	-	-	-	-	-	√	-	√	-
3.	(Misbahul dan Yufis 2020)	Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan Software Defined Network	√	-	√	-	-	-	-	-	-	√	-	√
4.	(Hafid 2019)	Investigasi Log Jaringan untuk Deteksi Serangan DDOS Menggunakan Metode General Regression Neural Network.	-	√	-	-	√	-	-	-	√	-	-	√

5.	(Lamdompak 2016)	Klasifikasi *Malware* Trojan Ransomware Dengan Algoritma Support Vector Machine (SVM).	-	√	-	-	-	-	√	-	√	-	√	-
6.	(Chambali, Muhammad, dan Harsono 2018)	Klasifikasi Paket Jaringan Berbasis Analisis Statistik dan Neural Network	-	√	-	-	-	-	-	√	√	-	-	√
7.	(Subrata, Widyantara, dan Linawati 2016)	Klasifikasi Penggunaan Protokol Komunikasi Pada Trafik Jaringan Menggunakan Algoritma K-Nearest Neighbor	-	√	√	-	-	-	-	-	√	-	-	√
8.	(Gusti dkk., 2017)	Pendeteksian *Malware* pada Lingkungan Aplikasi Web dengan Kategorisasi Dokumen.	-	√	-	-	-	√	-	-	-	√	√	-

9.	(Kumar dkk., 2019)	Deep in the Dark: A Novel Threat Detection System using Darknet Traffic.	-	√	√	√	-	-	-	-	√	-	-	√
10.	(Analisa Ancaman Serangan *Malware* Pada Trafik *Darknet* Menggunakan Algoritma *K-Nearest Neighbour*, 2021)	Analisa Ancaman Serangan *Malware* Pada Trafik *Darknet* Menggunakan Algoritma *K-Nearest Neighbour*	-	√	√	-	-	-	-	-	√	√	-	√

Keterangan:

KNN = K- Nearest Neighbour

RF = Random Forest

GRNN = General Regression Neural Network

NB = Naive Baiyes

SVM = Support Vector Machine

NN = Neural Network

2.8 Relevansi Penelitian

Tabel II. 3 Tabel Relevansi Penelitian

Peneliti	**(Kumar dkk., 2019)**	**(Yan Li dan Yufei Li, 2021)**	**(Analisa Ancaman Serangan *Malware* Pada Trafik *Darknet* Menggunakan Algoritma *K-Nearest Neighbour*, 2021)**
Judul	Deep in the Dark: A Novel Threat Detection System using Darknet Traffic.	ETCC : Encrypted Two-Label Classification Using CNN	Analisa Ancaman Serangan *Malware* Pada Trafik Darknet Menggunakan Algoritma K-Nearest Neighbour
Masalah Penelitian	Lalu lintas dari pemindaian Internet, penyebaran *malware*, atau peristiwa DDoS yang menyebar Kembali.	Teknologi enkripsi pada trafik darknet dapat membantu peretas untuk menyembunyikan perilaku jahat. Manajer jaringan harus dapat mengidentifikasi lalu lintas terenkripsi secara tepat waktu, sehingga dapat dengan cepat dan akurat menemukan serangan di jaringan, memutus jalur transmisi, dan mengurangi bahaya perilaku berbahaya bagi pengguna.	Penelitan Kumar dkk melakukan pendekatan klasifikasi trafik darknet dengan dataset Surfnet yang masih memerlukan pengembangan dataset. Tingkat akurasi hasil penelitian Yan Li dan Yufei Li masih bisa ditingkatkan.

Objek Penelitian	Klasifikasi trafik daknet dengan dataset Surfnet	Klasifikasi trafik darknet dengan dataset CICDarknet 2020	Klasifikasi dan prediksi trafik darknet dengan dataset CICDarknet 2020
Algoritma / Metode	K-Nearest Neighbor, Random Forest	CNN dan LSTM	K-Nearest Neighbor
Dataset	Surfnet 76 fitur	CICDarknet 2020 86 fitur	CICDarknet 2020 86 fitur
Label	Darknet dan Normal	NonTOR, NonVPN, TOR dan VPN	NonTOR, NonVPN, TOR dan VPN
Seleksi Fitur	-	-	Information Gain