BAB 1
USULAN GAGASAN

1.1 Latar Belakang Masalah

Dalam beberapa dekade terakhir, teknologi terus berkembang pesat di berbagai bidang. Seluruh aspek kehidupan manusia beralih ke sistem berbasis digital. Digitalisasi ini memudahkan kegiatan manusia sehari-hari, seperti administrasi, pengarsipan, komunikasi, hiburan, dan sebagainya. Namun, perkembangan teknologi tidak luput juga dari berbagai ancaman. Ancaman ini datang dalam bentuk jenis kejahatan baru, yang pada umumnya disebut sebagai kejahatan siber atau *cybercrime*. Kejahatan siber memiliki beberapa bentuk seperti *hacking*, *cyberbullying, identity theft, phishing,* dan lainnya. Salah satu jenis kejahatan siber yang paling umum adalah *malware* [1].

Malware merupakan segala jenis perangkatan lunak yang diciptakan dan didesain untuk menyerang sistem dari komputer, jaringan komputer atau *server* [2]. Menurut Statista, jumlah *malware attack* di tahun 2023 mencapai 6,06 miliar serangan. Jenis *malware* yang sering menyerang pengguna adalah *worms, viruses, ransomware, trojans, dan backdoor*. Dua media penyerangan utama *malware* adalah *email* dan situs website [3]. Pada *email, malware* pada umumnya menyebar lewat *file* lampiran. Pengguna yang tidak teliti secara tidak sadar akan menekan *file* lampiran tersebut dan *malware* akan otomatis menginfeksi komputer pengguna [4]. Penyerangan *malware* melalui situs website pada umumnya melalui situs pembajakan aplikasi. Dalam beberapa aplikasi bajakan, kriminal siber biasanya menyisipkan *file-file* yang berisi *malware*. *Malware* yang terdapat pada aplikasi bajakan ini biasanya berbentuk *adware, trojan, virus*, dan jenis *malware* lainnya [5]. Oleh karena itu, diperlukan analisis *malware* untuk mempelajari dan memahami tujuan, jenis, tipe, dan pengaruh dari *malware* untuk mencegah insiden *malware*.

Terdapat tiga penelitian sebelumnya yang membahas permasalahan analisis *malware* menggunakan *machine learning* [6], [7], [8]. Metode utama yang digunakan oleh penelitian sebelumnya adalah penggunaaan Amazon Web Service sebagai media *deployment sandbox* untuk menganalisa *malware* menggunakan *cuckoo sandbox* [6], [8]. Penggunaan AWS ini tentu saja membutuhkan biaya yang tidak sedikit. Berdasarkan perhitungan yang telah dilakukan menggunakan Amazon Pricing Calculator, didapatkan bahwa estimasi penggunaan

AWS selama pengerjaan proyek *capstone* ini berjumlah Rp 11.145.852,00. Hal ini tentu saja menjadi sebuah hambatan terbesar dalam pengerjaan proyek *capstone* ini.

Selain itu, analisis *malware* menggunakan *cloud-based sandbox* memiliki permasalahan tersendiri. *Cloud-based sandbox* membutuhkan jaringan internet agar dapat beroperasi. Hal ini memberikan resiko adanya penyebaran *malware* melalui jaringan internet. *Malware* dapat masuk melalui jaringan, yang kemudian akan masuk dan "menyelipkan" dirinya dalam bentuk *file* terinkripsi dan bentuk ".exe". Pengguna tidak akan menyadari bahwa jaringan tersebut telah terinfeksi oleh *malware* [9]. Penggunaan *cloud-server* juga tidak memberikan perlindungan dari serangan *malware. Malware* dapat menginfeksi melalui berbagai cara, seperti menginfeksi saat proses migrasi *virtual machine,* memanfaatkan konfigurasi *virtual machine*, komunikasi pada server, instalasi *software* yang diperlukan pada *virtual machine,* dan memanfaatkan kerentanan pada *image virtual machine* [10].

1.2 Analisis Masalah

Dalam mengembangkan sebuah solusi yang tepat dan efektif, langkah pertama yang harus dilakukan adalah analisis mendalam terhadap permasalahan yang dihadapi. Berikut adalah aspek-aspek yang menjadi faktor masalah:

1.2.1 Definisi dan Jenis *Malware*

Seperti yang telah dijelaskan sebelumnya, *malware* merupakan segala jenis perangkatan lunak yang diciptakan dan didesain untuk menyerang sistem dari komputer, jaringan komputer atau *server.* Portabilitas *malware* memungkinkan *malware* untuk menyebar ke berbagai *platform* dengan mudah. *Malware* merupakan serangan paling merugikan bagi perusahaan [2]. Adapun jenis-jenis *malware* yang dapat menyebar adalah sebagai berikut:

- Trojan: *malware* yang menyamar sebagai file baik (*benign*) dan dapat mencueri data, atau membuka akses.
- Ransomware: *malware* yang mengunci sistem dan meminta uang tebusan agar diberikan akses.
- Cryptominer: *malware* yang menggunakan sumber daya perangkat korban untuk menambang *cryptocurrency* tanpa sepengetahuan korban [11].

1.2.2 Aspek Teknis

Bagian ini akan membahas permasalahan teknis yang dihadapi dalam pengerjaan sistem analisis ini. Adapun penjelasan yang lebih detail mengenai permasalahan aspek teknis adalah sebagai berikut:

A. Perbandingan Server *Cloud* dengan Server *On-Premises*

Penggunaan *cloud* tentunya membutuhkan biaya yang tidak sedikit, namun apabila dibandingan dengan biaya *on-premises,* tentunya cloud menjadi pilihan yang utama. Hal ini dikarenakan *on-premises* membutuhkan biaya lebih banyak untuk menyewa/membeli infrastruktur, pembelian *hardware* dan *software,* pendinginan, membayar pegawai, biaya perawatan, dan lainnya. Oleh karena itu, *cloud* memiliki keuntungan yang signifikan dalam hal biaya dikarenakan pengguna hanya perlu membayar biaya berlangganan [12], [13].

Salah satu keuntungan penggunaan *cloud* adalah kemudahan skalabilitas. *Cloud* memberikan kemudahan dalam peningkatan beban kerja dan kapasitas. Apabila ingin menambahkan beban kerja dan kapasitas pada *on-premise*, tentu saja akan membutuhkan waktu yang lebih lama seperti waktu pembelian perangkat keras, pemasangan, konfigurasi dan lain-lain. Namun dengan menggunakan *cloud,* skalabilitas dapat dilakukan dengan lebih cepat atau bahkan secara instan. Hal ini memberikan efektivitas yang lebih tinggi. Selain itu, biaya skalabilitas pada *cloud* tentu lebih murah dibanding dengan *on-premise* [12].

Berdasarkan penelitian yang dilakukan sebelumnya, teknis utama dalam pengerjaan analisis *malware* ini adalah penggunaan *cloud-based sandboxing [6], [8].* Penggunaan *cloud-based sandbox* memerlukan layanan *cloud* dari pihak ketiga yang memiliki risiko keamanan jika layanan penyedia *cloud* tidak memiliki kebijakan keamanan yang kuat [10]. Meskipun penggunaan *cloud-server* jaminan keamanan, fleksibilitas dan skalabilitas yang tinggi, penggunaan *server on-premises* memberikan keamanan yang lebih ketat karena dikelola secara langsung [14].

B. Kecepatan Proses Analisis

Pada proses analisis *malware* menggunakan Cuckoo Sandbox yang di-*deploy* pada KVM *virtualization software*, sampel *malware* dikonfigurasi untuk dieksekusi selama 10 menit. Namun, 81% sampel *malware* dari 100.000 data berhasil dihentikan lebih awal sebelum mencapai ambang waktu tersebut, dengan lebih dari setengahnya berhenti dieksekusi pada

menit pertama dan sisanya berhenti pada tiga menit pertama. Walaupun begitu, ada sampel yang dieksekusi selama 13 menit atau maksimal 15 menit [15]. Dengan hasil ini, diharapkan penelitian kami dapat menghasilkan proses analisis yang lebih singkat.

C. Dataset *Malware*

Berdasarkan kajian terhadap penelitian sebelumnya, ditemukan beberapa permasalahan terkait penggunaan dataset *malware*. Ketiga penelitian sebelumnya menggunakan *dataset* dari Virusshare, *repository* sampel *malware*, sebagai dasar pembuatan *machine learning* mereka. Namun, dataset yang digunakan hanyalah sedikit seperti 189 sampel *malware* serta 193 sampel *goodware* [6] dan 382 sampel *malware* serta sampel *goodware* [7], [8]. Padahal, Virusshare sendiri memiliki *dataset* sejumlah 89 juta sampel.

Keterbatasan ukuran dataset ini menimbulkan beberapa masalah signifikan. Pertama, ukuran dataset yang kecil dapat menyebabkan model *machine learning* mengalami overfitting, dimana model cenderung menghafal pola spesifik dari data training daripada mempelajari karakteristik umum *malware*. Kedua, jumlah dataset tidak mampu merepresentasikan keragaman jenis *malware* yang ada di dunia nyata. Ketia, ketidakseimbangan jumlah sampel *malware* dan goodware dapat menyebabkan bias dalam proses pembelajaran model. Keempat, dataset yang berukuran kecil berpotensi menghasilkan model dengan tingkat akurasi yang tidak reliable ketika diaplikasikan pada skala yang lebih besar [16].

D. Akurasi Algoritma Machine Learning

Penelitian *malware analysis* dengan *machine learning* sebelumnya menggunakan tiga algoritma yang berbeda dengan nilai akurasi yang berbeda juga, yaitu *Random Forest*, *K-Nearest Neighbors*, dan *Decision Tree*. *Random forest* dan *K-Nearest Neighbors* (K-NN) memiliki masalah skalabilitas dimana semakin besar dataset yang digunakan, semakin besar sumber daya komputasi yang dipakai [17], [18]. Khususnya bagi K-NN, memiliki kecenderungan melambat dalam waktu pemrosesan data dengan jumlah yang banyak [19] sehingga deteksi *malware* tidak bersifat *real-time* [18]. K-NN dan *decision tree* juga rentan terhadap *overfitting*, dimana model tidak dapat beradaptasi dengan data baru saat diuji dikarenakan model tidak memahami pola dibalik data, karena nilai k yang terlalu kecil atau karena dataset kompleks yang memiliki banyak fitur [20], [21]. Terdapat masalah ketidakseimbangan pada dataset yang membuat algoritma *machine learning* memprediksi secara bias ke kelas mayoritas karena sampel *goodware* lebih banyak dibandingkan dengan

sampel *malware*. Masalah ini biasanya dialami oleh algoritma *random forest* dan *decision tree* karena algoritma tersebut sensitif dengan ketidakseimbangan dataset [22].

E. Tantangan Pendeteksian *Malware*

Beberapa sistem pendeteksi virus pada saat ini masih sulit atau gagal dalam pendeteksian beberapa jenis *malware*. Mayoritas vendor *anti*-virus konvensional telah mengembangkan solusi *anti-virus* yang dapat melindungi perangkat menggunakan metode *signature-based malware detection*, metode mendeteksi *malware* menggunakan fitur unik dari *malware*. Namun, metode ini memiliki kelemahan, yaitu tidak bisa mendeteksi *malware* yang tidak diketahui atau *zero-day malware* [23], [24].

1.2.3 Aspek Ekonomi

Seperti yang telah dijelaskan pada bagian sebelumnya, penggunaan AWS membutuhkan biaya yang tidak sedikit. Estimasi penggunaan *cloud-service* AWS menggunakan Amazon Pricing Calculator selama pengerjaan proyek *capstone* ini membutuhkan biaya kurang lebih Rp 11.145.852,00 untuk penggunaan AWS selama satu tahun. Tingginya biaya ini menjadi tantangan tersendiri dalam pengerjaan proyek *capstone*.

1.2.4 Aspek Keamanan

Dalam pengembangan sistem analisis *malware*, aspek keamanan menjadi elemen yang sangat krusial dan tidak dapat diabaikan. Sistem yang dibangun secara langsung berinteraksi dengan file berbahaya yang berpotensi menyebabkan kerusakan pada sistem apabila tidak ditangani secara aman dan terisolasi.

Salah satu ancaman utama dari sistem sandbox berbasis cloud adalah risiko kebocoran *malware* dari lingkungan virtual ke jaringan luar. Hal ini disebabkan oleh kebutuhan sandbox untuk tetap terhubung ke internet dalam proses deployment, pengambilan data, serta komunikasi antarkomponen sistem. Koneksi ini membuka celah bagi *malware* yang canggih untuk mengeksploitasi celah keamanan dan keluar dari lingkungan analisis melalui jaringan yang digunakan.

1.3 Analisis Solusi yang Ada

Pada penelitian terdahulu dengan judul "Perancangan Sistem Analisis Dinamis Dan Klasifikasi *Malware* Otomatis Dengan Algoritma K-Nearest Neighbors" dirancang sistem otomatis untuk mendeteksi *malware* menggunakan analisis dinamis dengan Cuckoo Sandbox. Data yang dihasilkan kemudian diklasifikasikan menggunakan algoritma K-NN. Hasil pengujian menunjukkan akurasi deteksi *malware* sebesar 95%, dengan K-NN memilih tetangga terdekat untuk klasifikasi file sebagai *malware* atau *goodware* [6]. Namun, pada penelitian ini nilai akurasi yang diambil adalah akurasi dengan nilai k terkecil yang dapat menyebabkan *overfitting*.

Artikel penelitian berjudul "Intelligent Behavior-Based *Malware* Detection System on Cloud Computing Environment" memberikan sebuah metode analisis *malware* menggunakan kecerdasan buatan dan *cloud computing.* Pada penelitian tersebut, diciptakan sebuah sistem yang dapat mendeteksi *malware* berdasarkan sfat dan perilaku dari *malware* yang dianalisis. Dengan menggunakan algoritma *Logistic Model Trees* (LMT), C4.5 (J48), *Random Forest* (RF), *Simple Logistic Regression* (SLR), *Sequential Minimal Optimization* (SMO), dan *K-Nearest Neighbor* (K-NN), sistem dapat membedakan apakah *file* merupakan *malware* atau bukan berdasarkan dari perilaku dari *system calls file* tersebut. Metode ini memiliki nilai tingkat deteksi sebesar 99,8%, tingkat *false positive* 0,4%, dan tingkat akurasi 99,7% [25].

Kedua penelitian sebelumnya dapat dikatakan sebagai analisis dinamis dikarenakan analisis berdasarkan analisis sifat dan perilaku dari *malware*. Selain metode dinamis, terdapat juga metode statis. Metode statis adalah metode analisis *malware* yang tidak perlu menjalankan program *malware*. Namun berdasarkan analisis *source-code* atau kode biner dari *malware* tersebut [26].

Salah satu contoh dari analisis statis adalah penelitian berjudul "Static *Malware* Analysis Using Low-Parameter Machine Learning Models". Penelitian ini menggunakan *machine learning* untuk menganalisis *source-code* dari *malware* yang dianalisis. Penelitian ini menggunakan tiga algoritma utama dalam proses analisis, yakni *Artificial Neural Networks* (ANN), *Support Vector Machines* (SVMs) dan *Gradient Boosting Machines* (GBMs). Alasan utama dalam penggunaan dua algoritma yang berbeda adalah untuk mengurangi adanya *false positive*. Tingkat akurasi tertinggi dicapai saat penggunaan algoritma ANN, yakni senilai 94% [27].

Tabel 1.1 Perbandingan Solusi Terdahulu

Judul Penelitian	**Metode Penelitian**	**Algoritma**	**Tingkat Akurasi**	**Tingkat False Positive**	**Tingkat False Negative**
Perancangan Sistem Analisis Dinamis Dan Klasifikasi *Malware* Otomatis Dengan Algoritma K-Nearest Neighbors	Perancangan sistem otomatis untuk mendeteksi *malware* menggunakan analisis dinamis dengan Cuckoo Sandbox.	K-Nearest Neighbors (K-NN)	95%	N/A	N/A
Intelligent Behavior-Based *Malware* Detection System on Cloud Computing Environment	Menciptakan sebuah sistem yang dapat mendeteksi *malware* berdasarkan sfat dan perilaku dari *malware* yang dianalisis.	• *Logistic Model Trees* (LMT) • C4.5 (J48) • *Random Forest* (RF) • *Simple Logistic Regressio*n (SLR) • *Sequential Minimal Optimization* (SMO) • *K-Nearest Neighbor* (K-NN)	99,7%	0,4%	N/A

Judul Penelitian	Metode Penelitian	Algoritma	Tingkat Akurasi	Tingkat False Positive	Tingkat False Negative
Static *Malware* Analysis Using Low-Parameter Machine Learning Models	Menggunakan *machine learning* untuk menganalisis *source-code* dari *malware* yang dianalisis.	• *Artificial Neural Networks* (ANN) • *Support Vector Machines* (SVMs) • *Gradient Boosting Machines* (GBMs).	94%	ANN: 10.7% SVM: 15.8% GBM: 13.3%	ANN: 4.8% SVM: 5.3% GBM: 5.0%

Kekurangan utama dari solusi terdahulu adalah ketergantungan terhadap koneksi internet. Seperti yang telah dijelaskan sebelumnya, penyambungan dengan internet memberikan resiko adanya penyebaran *malware*. Selain itu, koneksi yang tidak stabil dapat menghambat kinerja sistem secara keseluruhan, menyebabkan gangguan akses atau keterlambatan dalam proses yang seharusnya berjalan cepat.

1.4 Kesimpulan dan Ringkasan CD-1

Metode analisis *malware* dengan menggunakan machine learning memiliki potensi besar dalam mendeteksi *malware* secara efektif. Metode ini terbagi menjadi analisis dinamis, yang mengamati perilaku *malware* saat dijalankan, dan analisis statis, yang memeriksa kode tanpa menjalankannya. Beberapa algoritma seperti K-Nearest Neighbors (K-NN), Random Forest (RF), dan Artificial Neural Networks (ANN) digunakan untuk meningkatkan akurasi deteksi. Meskipun metode-metode ini mampu mencapai akurasi yang tinggi, tantangan seperti ketergantungan pada koneksi internet, risiko keamanan dari penggunaan cloud-based sandboxing, serta potensi overfitting pada algoritma tertentu perlu diatasi. Selain itu, penggunaan server on-premises dapat menjadi solusi alternatif untuk meningkatkan keamanan, meskipun dengan biaya yang lebih besar.