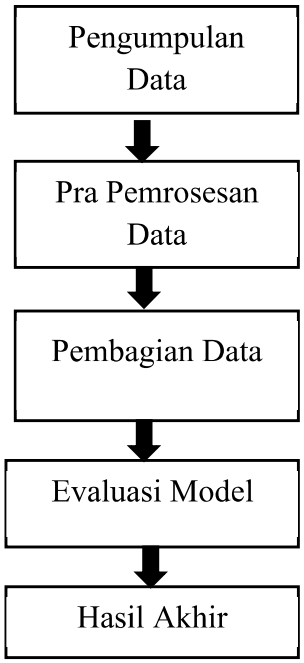
BAB lll

METODE PENELITIAN

3.1 Desain Penelitian

Perencanaan penelitian adalah perencanaan penelitian yang tujuannya untuk memberikan pedoman dalam melakukan proses penelitian. Perencanaan penelitian menyediakan kerangka kerja dan alur kerja untuk seluruh proses penelitian. Berikut adalah desain penelitian dari penelitian ini.



Gambar 3. 1 Desain Penelitian

Berikut penjelasan untuk desain penelitian dari penelitian ini:

1. Pengumpulan Data

Data dikumpulkan dari kumpulan sampel malware dan non-malware yang berasal dari Sumber tertentu Yaitu Kanggle. Data ini digunakan untuk melakukan pelatihan dan pengujian model SVM

2. **Pra-pemrosesan Data (Preprocessing)**

Pada titik ini, data dibersihkan dan disusun. Misalnya, hapus data duplikat atau tidak relevan; normalisasi nilai; menangani data kosong atau tidak lengkap; dan encoding variabel kategori jika diperlukan.

3. Pembagian Data (Data Split)

Data set yang telah diproses terdiri dari dua bagian:

1. Data set pelatihan
2. Data set pengujian

a. Pembuatan Model SVM

Data pelatihan digunakan untuk melatih model Support Vector Machine. SVM memilih hyperplane terbaik untuk membedakan data malware dan non-malware.

b. Pengujian Data

Selanjutnya, data pengujian digunakan untuk menguji model yang telah dilatih. Tujuan dari pengujian ini adalah untuk mengetahui sejauh mana model dapat mengenali serangan malware secara akurat.

4. Evaluasi Model

1. Akurasi
2. Precision
3. Recall, dan
4. Skor F1

Ini adalah metrik yang digunakan untuk mengevaluasi hasil tes.

5. Hasil Akhir

Evaluasi menghasilkan kesimpulan tentang seberapa baik model SVM mendeteksi malware. Keputusan ini membantu menentukan efektivitas metode yang digunakan.

3.2 Metode Pengumpulan data

Dalam penelitian ini, metode pengumpulan data dilakukan melalui Bebrapa metode, yaitu Survei, observasi exprimen, dan studi pustaka untuk mendukung pemahaman terhadap kebutuhan serta tantangan dalam sistem deteksi malware.

3.2.1 Survei

Survei ini dilakukan sebagai bagian dari pengumpulan data sekunder untuk memahami tren, metode, dan masalah dalam mendeteksi serangan malware menggunakan algoritma pengajaran mesin, khususnya Support Vector Machine (SVM). Survei ini juga dilakukan dengan membaca dokumentasi dari platform keamanan seperti VirusTotal dan AV-Test. Data yang dikumpulkan dari survei ini mencakup informasi tentang jenis-jenis malware yang paling umum dan fitur-fitur yang sering digunakan dalam proses deteksi (seperti entropi, persetujuan, panggilan

API, dan ukuran file). Teknik pre-processing dan evaluasi model yang telah digunakan dalam penelitian sebelumnya digunakan sebagai dasar untuk pemilihan fitur dan desain arsitektur model dalam penelitian ini.

3.2.2 Observasi

Peneliti melakukan observasi dengan mengamati aktivitas sistem komputer dan jaringan yang menjadi subjek penelitian, baik dalam kondisi normal maupun saat serangan malware terjadi. Dengan melakukan observasi ini, peneliti dapat memahami pola perilaku yang ditimbulkan oleh malware, seperti perubahan dalam penggunaan CPU, memori, aktivitas jaringan, dan file system. Pola-pola ini kemudian akan dimasukkan ke dalam proses pelatihan model SVM.

3.2.3 Exprimen

Eksperimen dilakukan dengan mensimulasikan serangan malware dalam lingkungan sandbox yang terkontrol. Peneliti menguji sistem uji dengan beberapa sampel malware untuk mendapatkan dataset aktivitas malware secara langsung. Selanjutnya, data diproses dan digunakan untuk pelatihan, pengujian, dan evaluasi model SVM. Tujuan eksperimen ini adalah untuk membuktikan bahwa algoritma SVM berfungsi dengan baik dalam mengklasifikasikan aktivitas malware dan non-malware.

3.2.4 Studi Pustaka

Studi ini mencakup penelusuran dan analisis berbagai artikel dan jurnal akademik yang membahas keamanan siber, deteksi malware, dan penerapan pembelajaran mesin. Tujuan dari penelitian ini adalah untuk membuat fondasi

teoritis dan konseptual yang mendukung penelitian mengenai deteksi serangan malware menggunakan algoritma Support Vector Machine (SVM). Studi ini bertujuan untuk mendapatkan pemahaman yang lebih baik tentang karakteristik malware, teknik ekstraksi fitur (statis maupun dinamis), prinsip kerja algoritma SVM, serta kelebihan dan kekurangan algoritma tersebut dalam konteks klasifikasi malware. Selain itu, penelitian ini juga bertujuan untuk menemukan teknik dan metodologi yang telah digunakan dalam penelitian terdahulu, sehingga dapat digunakan sebagai referensi untuk membangun sistem deteksi malware yang lebih baik. Hasil penelitian ini membentuk fondasi untuk proses pengolahan data, pemilihan fitur, dan pendekatan evaluasi performa model deteksi.

3.3 Metode Deteksi Malware Menggunakan Algoritma Support Vector Machine

Berikut beberapa Metode Deteksi Malware Menggunakan Algoritma Support Vector Machine yang digunakan dalam penelitian ini:

3.3.1 Pengumpulan Dataset

Dalam penelitian ini dataset malware diambil dari sumber yaitu Kanggle. Dataset Yang digunakan dalam proses pre-processing 100.000 data sampel untuk menemukan malware dalam penelitian ini. Jumlah fitur yang terdapat dataset sebanyak 34 fitur.

Adapun fitur yang digunakan untuk mendeteksi serangan malware menggunakan sumber dataset Melalui Kanggle Yang terdiri dari 34 fitur sebagai berikut.

Tabel 3. 1 Tabel Pengumpulan Dataset

no	Fitur	Keterangan
0	prio	Prioritas proses yang dinamis (berubah sesuai dengan perilaku proses).
1	static_prio	Prioritas proses yang statis (tetap, tidak berubah kecuali diubah)
2	normal_prio	Strategi jadwal saat ini memengaruhi prioritas efektif.
3	policy	Jenis kebijakan penjadwalan yang digunakan (misalnya SCHED_FIFO, SCHED_RR, SCHED_NORMAL).
4	vm_pgoff	Offset pada halaman peta memori virtual; digunakan dalam manajemen memori.
5	vm_truncate_count	jumlah pemotongan dan kesalahan dalam area memori virtual (VMA).
6	task_size	ukuran jumlah total ruang alamat virtual yang digunakan oleh proses.
7	cached_hole_size	Ukuran celah dalam peta memori yang telah dicache
8	free_area_cache	Area memori bebas disimpan dalam virtual memory.
9	mm_users	jumlah proses yang berbagi mm_struct, biasanya 1 kecuali clone.
10	map_count	jumlah data yang dimasukkan ke dalam tabel mapping memori (jumlah VMA).
11	hiwater_rss	Puncak dari ukuran set penduduk (jumlah halaman yang paling banyak digunakan)
12	total_vm	total memori virtual yang digunakan proses (dalam halaman).
13	shared_vm	Memori virtual yang digunakan bersama dengan proses lainnya
14	exec_vm	Executable memori virtual yang digunakan untuk menjalankan kode
15	reserved_vm	Memori virtual yang telah dipesan tetapi belum digunakan.
16	nr_ptes	jumlah entri dalam tabel halaman.
17	end_data	penunjuk ke segmen data proses di ujung diagram ELF.
18	last_interval	interval waktu terakhir yang digunakan untuk mengukur atau mengambil sampel
19	nvcsw	jumlah switch konteks sukarela (proses memberikan CPU secara sukarela)

no	Fitur	Keterangan
20	nivcsw	jumlah pergeseran konteks yang tidak diinginkan
21	min_flt	jumlah kesalahan halaman kecil (halaman ditemukan di memori, tidak dibaca di disk).
22	maj_flt	Jumlah kesalahan halaman yang signifikan, yang berarti halaman harus diambil dari disk.
23	fs_excl_counter	counter yang biasanya terkait dengan lock atau sync untuk memberikan akses eksklusif ke filesystem.
24	lock	struktur penting yang melindungi proses dari kondisi ras, seperti spinlock
25	utime	Waktu CPU yang digunakan dalam mode user
26	stime	Waktu CPU yang digunakan di mode kernel (system time).
27	gtime	waktu yang dihabiskan oleh proses grup (dihapus dalam versi kernel baru).
28	cgtime	waktu CPU yang digunakan oleh proses anak-anak (waktu kumulatif turunan).
29	signal_nvcsw	Jumlah context switch sukarela yang disebabkan oleh sinyal.
30	Usage_counter	Jumlah penggunaan atau aktivitas proses; menunjukkan seberapa sering proses beroperasi atau mendapatkan akses ke sumber daya.
31	state	Saat proses diamati, statusnya (misalnya aktif, tidur, zombie, dll.) menunjukkan kondisi proses saat ini.
32	classification	Label klasifikasi target menunjukkan apakah proses tersebut adalah malware atau benign.
33	millisecond	Waktu dalam milidetik sejak awal pengambilan sampel data atau pemantauan; berguna sebagai penanda waktu kronologis
34	Hash	ID proses unik, yang biasanya berupa hash dari file executable, digunakan untuk mengidentifikasi proses.

3.3.2 Data *Preprocessing*

Dalam deteksi serangan *malware*, dataset yang didapatkan dilakukan preprocessing yang dimulai dengan pembersihan data, yang menghilangkan

duplikat data dan menangani nilai yang tidak ada. Selanjutnya, fitur dinamis dan statis, seperti ukuran file, entropi, panggilan API, dan izin, diekstraksi. Encoding mengubah semua fitur menjadi format numerik dan kemudian dinormalisasi untuk memastikan skala yang seragam. Untuk menjaga proporsi kelas, metode sampling stratified digunakan untuk membagi dataset menjadi dua bagian: 80% data latih dan 20% data uji. Untuk memperbaiki ketidakseimbangan kelas, metode SMOTE digunakan. Pre-processing ini memastikan data dalam kondisi terbaik untuk melatih model SVM, yang memungkinkannya mendeteksi malware dengan baik.

Tabel 3. 2 Tabel Data *Processing*

no	Fitur	Keterangan
0	prio	Prioritas proses yang dinamis (berubah sesuai dengan perilaku proses).
1	static_prio	Prioritas proses yang statis (tetap, tidak berubah kecuali diubah)
2	normal_prio	Strategi jadwal saat ini memengaruhi prioritas efektif.
3	policy	Jenis kebijakan penjadwalan yang digunakan (misalnya SCHED_FIFO, SCHED_RR, SCHED_NORMAL).
4	vm_pgoff	Offset pada halaman peta memori virtual; digunakan dalam manajemen memori.
5	vm_truncate_count	jumlah pemotongan dan kesalahan dalam area memori virtual (VMA).
6	task_size	ukuran jumlah total ruang alamat virtual yang digunakan oleh proses.
7	cached_hole_size	Ukuran celah dalam peta memori yang telah dicache
8	free_area_cache	Area memori bebas disimpan dalam virtual memory.
9	mm_users	jumlah proses yang berbagi mm_struct, biasanya 1 kecuali clone.
10	map_count	jumlah data yang dimasukkan ke dalam tabel mapping memori (jumlah VMA).

no	Fitur	Keterangan
11	hiwater_rss	Puncak dari ukuran set penduduk (jumlah halaman yang paling banyak digunakan)
12	total_vm	total memori virtual yang digunakan proses (dalam halaman).
13	shared_vm	Memori virtual yang digunakan bersama dengan proses lainnya
14	exec_vm	Executable memori virtual yang digunakan untuk menjalankan kode
15	reserved_vm	Memori virtual yang telah dipesan tetapi belum digunakan.
16	nr_ptes	jumlah entri dalam tabel halaman.
17	end_data	penunjuk ke segmen data proses di ujung diagram ELF.
18	last_interval	interval waktu terakhir yang digunakan untuk mengukur atau mengambil sampel
19	nvcsw	jumlah switch konteks sukarela (proses memberikan CPU secara sukarela)
20	nivcsw	jumlah pergeseran konteks yang tidak diinginkan
21	min_flt	jumlah kesalahan halaman kecil (halaman ditemukan di memori, tidak dibaca di disk).
22	maj_flt	Jumlah kesalahan halaman yang signifikan, yang berarti halaman harus diambil dari disk.
23	fs_excl_counter	counter yang biasanya terkait dengan lock atau sync untuk memberikan akses eksklusif ke filesystem.
24	lock	struktur penting yang melindungi proses dari kondisi ras, seperti spinlock
25	utime	Waktu CPU yang digunakan dalam mode user
26	stime	Waktu CPU yang digunakan di mode kernel (system time).
27	gtime	waktu yang dihabiskan oleh proses grup (dihapus dalam versi kernel baru).
28	cgtime	waktu CPU yang digunakan oleh proses anak-anak (waktu kumulatif turunan).
29	signal_nvcsw	Jumlah context switch sukarela yang disebabkan oleh sinyal.
30	Usage_counter	Jumlah penggunaan atau aktivitas proses; menunjukkan seberapa sering proses beroperasi atau mendapatkan akses ke sumber daya.

no	Fitur	Keterangan
31	state	Saat proses diamati, statusnya (misalnya aktif, tidur, zombie, dll.) menunjukkan kondisi proses saat ini.
32	classification	Label klasifikasi target menunjukkan apakah proses tersebut adalah malware atau benign.
33	millisecond	Waktu dalam milidetik sejak awal pengambilan sampel data atau pemantauan; berguna sebagai penanda waktu kronologis

Tabel preprocessing di atas menunjukkan bahwa dataset terdiri dari 33 fitur yang telah mengalami proses pembersihan. Nilai-nilai yang tidak relevan dihapus, nilai nol atau konstan ditangani, dan format data disesuaikan agar siap digunakan untuk tahap selanjutnya, pelatihan model klasifikasi. Proses pembersihan data sangat penting untuk memastikan bahwa fitur yang digunakan benar-benar menunjukkan karakteristik data yang penting dan tidak mengandung noise atau outlier yang dapat mengganggu kinerja model.

3.3.3 Explanatory Data Analysis

Sebelum memasuki tahap pemodelan, analisis data eksplisit (EDA) dilakukan untuk mendapatkan pemahaman tentang karakteristik data. Pada titik ini, analisis distribusi data, identifikasi pola, pencarian anomali, dan analisis hubungan antar fitur dilakukan. EDA memastikan data yang digunakan tepat dan berkualitas tinggi. Untuk menunjukkan persebaran data, visualisasi seperti histogram, boxplot, dan scatterplot digunakan. Selain itu, dilakukan analisis korelasi fitur untuk menentukan fitur mana yang memiliki korelasi kuat dengan label malware.

3.3.4 Feature Extraction

Tujuan ekstraksi fitur adalah untuk mengekstrak informasi penting dari data mentah yang terkait dengan aktivitas malware. Studi ini mengekstraksi fitur dinamis seperti jumlah dan jenis panggilan API dan izin yang diminta oleh aplikasi serta fitur statis seperti ukuran file dan entropi. Teknik encoding mengubah fitur menjadi representasi numerik, sehingga dapat digunakan dalam proses pelatihan model pembelajaran mesin.

3.3.5 Data Split

Untuk mendukung proses pelatihan dan evaluasi model deteksi malware, dataset malware yang terdiri dari 100.000 sampel akan dibagi menjadi tiga bagian utama dalam penelitian ini. Set pelatihan akan mencakup 80.000 sampel, atau 80% dari dataset secara keseluruhan. Bagian ini akan digunakan untuk melatih model untuk mengidentifikasi pola malware. Kedua, set validasi terdiri dari 20.000 bentuk testing sampel, atau sekitar 20 persen dari dataset secara keseluruhan, dan berfungsi untuk memvalidasi kinerja model selama proses pelatihan dan membantu dalam penyesuaian parameter model untuk meningkatkan akurasi. Setelah proses pelatihan dan validasi selesai, set uji digunakan untuk menguji kinerja akhir model, memberikan gambaran yang akurat tentang kemampuan model untuk mengidentifikasi malware pada data yang belum pernah dilihat sebelumnya. Pembagian dataset seperti ini dirancang untuk memastikan bahwa model dapat belajar dan dievaluasi dengan baik.

3.3.6 Build Model klasifikasi

Pada tahap ini, algoritma Support Vector Classifier (SVC) digunakan untuk membangun model deteksi malware. Dalam proses pelatihan, data latihan digunakan untuk mencapai hasil yang optimal, parameter SVM seperti kernel, C (parameter regularization), dan gamma (untuk kernel RBF) dioptimasi menggunakan teknik pengaturan hyperparameter seperti pencarian grid atau pencarian acak.

3.3.7 Evaluation Model

Data uji digunakan untuk mengevaluasi model yang telah dibangun dalam mendeteksi serangan malware. Untuk memahami kesalahan klasifikasi yang terjadi, analisis confusion matrix digunakan. Metrik evaluasi yang digunakan termasuk akurasi, ketepatan, recall, skor F1, dan ROC-AUC. Model SVM yang dihasilkan dapat mendeteksi malware dengan tingkat keakuratan yang tinggi dan generalisasi yang baik, menurut evaluasi ini. Tahap terakhir setelah pembentukan model klasifikasi adalah menilainya. Dataset yang telah dipreprocessing dibagi menjadi subset pelatihan dan pengujian. Algoritma svm digunakan untuk melatih model. Hasil klasifikasi untuk setiap model pembelajaran svm dihitung dengan menggunakan matriks kekacauan. Metode ini memudahkan identifikasi hubungan antara hasil pengujian dan kinerja pengklasifikasi.

		Sebenarnya	
		Positif	**Negatif**
Prediksi	**Positif**	**TN** (True Positive)	**FN** (False Positive)
	Negatif	**FP** (False Negative)	**TN** (True Negative)

Tabel 3. 3 Evaluation Model

No	Nama lengkap	Artinya
1.	*True Positive*	Model memprediksi malware, dan malware memang ada.
2.	*False Positive*	Model memprediksi malware, meskipun sebenarnya tidak berbahaya.
3.	*False Negative*	(Berbahaya, malware lolos), tetapi model memprediksi benign.
4.	*True Negative*	Model menunjukkan bahwa itu benign.

Subset pengujian kemudian digunakan untuk menguji performa dan kinerja model. Metrik evaluasi seperti akurasi, presisi, recall, dan skor F1 menunjukkan seberapa baik metode svm dapat mendeteksi malware PE. Berikut adalah ukuran penilaian yang digunakan dalam penelitian ini:

- Nilai akurasi adalah nilai yang menunjukkan seberapa akurat sistem mengklasifikasikan data yang dirumuskan pada formula secara akurat.

$$Accuracy = \frac{Tp+Tn}{Tp+TN+FP+FN}$$ **Rumus 3. 1** Rumus Accuracy

- Nilai precision (presisi) adalah jumlah data positif yang diklasifikasikan secara benar dibagi dengan total data positif yang diklasifikasikan yang dirumuskan pada formula.

$$Precision = \frac{TP}{TP+FP}$$ **Rumus 3. 2** Rumus Precission

- Recall adalah nilai untuk mengetahui berapa persen data kategori positif yang diklasifikasikan dengan benar oleh sistem yang dirumuskan pada formula.

$Recall : \frac{tp}{TP+FN}$ **Rumus 3. 3** Rumus Recall

- Nilai F1-score adalah nilai harmonic mean untuk presisi dan recall, dengan skor terbaik 1.0 dan terburuk 0. Skor yang baik menunjukkan bahwa metode klasifikasi yang dibangun memiliki presisi dan recall yang dirumuskan pada formula.

$F1 - Score = \frac{2 \times Precission \times Recal}{Precission + recal}$ **Rumus 3. 4** Rumus F1-Score

3.4 Lokasi Dan Jadwal Penelitian

Adapun lokasi dan jadwal penelitian yang dilakukan peneliti adalah sebagai berikut.

3.4.1 Lokasi

Penelitian ini dilakukan sendiri oleh peneliti tanpa keterikatan langsung dengan lembaga atau institusi. Penelitian dilakukan di Indonesia, terutama di tempat peneliti tinggal, dengan komputer sebagai alat utama untuk analisis dan pengolahan data. Penelitian dilakukan secara daring (online), mulai dari pengumpulan data, praproses data, pembuatan model klasifikasi, dan evaluasi dan pengambilan kesimpulan. Ini memungkinkan penelitian dilakukan di mana saja tanpa batasan lokasi fisik. Data yang digunakan dalam penelitian ini diperoleh dari platform publik Kanggle (*https://www.kaggle.com/datasets/nsaravana/malware-detection*) yang menyediakan berbagai dataset, termasuk dataset yang berkaitan dengan deteksi malware. Dataset ini dapat diakses secara bebas dan digunakan untuk tujuan penelitian serta pengembangan sistem klasifikasi berbasis pembelajaran mesin. Dalam kasus ini, dataset ini digunakan untuk melatih dan menguji model klasifikasi yang digunakan dalam penelitian ini. Oleh karena itu, penelitian tidak terbatas pada suatu tempat fisik; sebaliknya, fokusnya adalah pada aktivitas analisis yang dilakukan melalui internet, menggunakan data digital yang tersedia secara terbuka.

3.4.2 Jadwal

Untuk memastikan bahwa seluruh proses penelitian berjalan secara sistematis, terarah, dan sesuai dengan waktu yang telah ditentukan, jadwal

penelitian disusun sebagai acuan dalam pelaksanaan setiap tahapan kegiatan, mulai dari persiapan hingga penyusunan laporan akhir.

Tabel 3. 4 Jadwal Penlitian

No	Tahapan Penelitian	Bulan					
		1	2	3	4	5	6
1.	Identifikasi Masalah dan latar belakang	✔					
2.	Tujuan dan Rumusan Masalah	✔					
3.	Manfaat penelitian dan teori dasar	✔	✔				
4.	Metode dan algoritma machine learning		✔	✔			
5.	Kerangka pemikiran dan penelitian terdahulu			✔	✔		
6.	Desain penelitian dan dataset				✔	✔	
7.	Evaluasi model svm					✔	
8.	Penyusunan Laporan Hasil dan Kesimpulan serta Publikasi jurnal					✔	✔
9.	Revisi, Konsultasi dan Finalisasi Laporan						✔


UPB
Universitas Putera Batam