Deteksi *Threat* dan *Vulnerability* pada Twitter menggunakan Algoritma *Support Vector Machine*

1st Raudhatul Rafiqah Assyahiddini
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
dinirasyani@student.telkomuniversity.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
setiacasie@student.telkomuniversity.ac.id

3rdMuhammad Faris Ruriawan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
muhammadfaris@telkomuniversity.ac.id

Abstrak—Di era globalisasi, perkembangan internet di bidang teknologi informasi menimbulkan kejahatan yang merugikan banyak pihak. Hal ini disebabkan adanya *threat* dan *vulnerability* terhadap sistem keamanan. Threat dan *vulnerability* ini dapat ditemukan di Twitter karena banyak pengguna memposting kejahatan sistem secara bebas di Twitter. Maka, pendeteksian *threat* dan *vulnerability* di Twitter dilakukan dengan menggunakan algoritma *Support Vector Machine* (SVM). Proses dalam penelitian ini adalah teks akan dikumpulkan menjadi sebuah dataset, diberi label, teks *pre-processed*, kemudian diberikan metode pembobotan yaitu metode POS *Tagging* dan *Term Frequency – Inverse Document Frequency* (TF-IDF), dan data akan dilatih agar algoritma SVM dapat mengklasifikasikan data untuk mendapatkan nilai akurasi, presisi, *recall* dan F1-Score. Pada Tugas Akhir ini, pengambilan data Twitter dengan total 4270 data, dengan data positif 2135 dan data negatif 2135 dan 90% dataset digunakan untuk data latih dan 10% untuk data pengujian. Dari hasil pengujian performansi didapatkan nilai akurasi sebesar 89%, presisi sebesar 89%, recall sebesar 89% dan F1-Score Pengujian parameter Gamma dan C terbaik sebesar 30000 anis 89%.

Kata kunci— *threat*, *support vector machine*, *text preproessing*, TF-IDF, *vulnerability*, Twitter

I. PENDAHULUAN

Di era globalisasi ini, perkembangan internet di bidang teknologi informasi berkembang pesat. Perkembangan tersebut juga menyebabkan tingkat kejahatan di bidang teknologi informasi yang merugikan individu dan organisasi. Hal ini disebabkan oleh *threat* dan *vulnerability* terhadap keamanan sistem [1]. *Threat* dan *vulnerability* dapat ditemukan pada postingan Twitter, hal ini disebabkan oleh banyaknya pengguna di media sosial. Tidak mengherankan jika hal-hal seperti kejahatan sistem diposting secara bebas di dunia maya. Tidak mengherankan jika hal-hal seperti kejahatan sistem diposting secara bebas di dunia maya. Saat ini, Twitter digunakan untuk memperoleh informasi, termasuk mengeksploitasi *threat* dan *vulnerability* yang disertakan dengan tautan yang mengarah ke web yang juga digunakan oleh penjahat dunia maya untuk melakukan serangan dunia maya, yang tentu saja dapat merusak teknologi milik individu dan organisasi. Penjahat dunia maya menggunakan media sosial untuk memanfaatkan berbagai kejahatan online seperti *phishing*, *spam, malware*, dan serangan *zero-day* [2]. *Threat* dan *vulnerability* yang terkandung dalam posting Twitter sering dimanfaatkan oleh penjahat *cyber* sebelum menangani lubang keamanan *(patch)* organisasi, seperti mengeksploitasi serangan *zero-day* [3]. Sambil menunggu *patch*, penjahat dunia maya dapat mengeksploitasinya dengan menyerang program komputer, gudang data, dan jaringan komputer. Eksploitasi ini mendapat dukungan dari beberapa ahli, bahwa ada penyerang ramah yang dapat menemukan dan melaporkan *vulnerability* daripada mengeksploitasinya sehingga dapat mempercepat suatu organisasi untuk mencegah atau menambal lubang keamanan sistemnya [4]. Oleh karena itu, perlu dilakukan pendeteksian gangguan keamanan sistem berupa *threat* dan *vulnerability* pada postingan Twitter, sehingga dapat terdeteksi postingan mana yang mengandung *threat* dan *vulnerability* dan mana yang tidak serta untuk mengurangi terjadinya eksploitasi di Twitter. Penelitian ini menggunakan algoritma *Support Vector Machine* (SVM).

II. KAJIAN TEORI

A. Twitter

Twitter adalah *micro-blogging* atau media sosial berukuran kecil yang dibuat oleh Jack Dorsey pada Maret 2006 dan dirilis pada Juli tahun itu. Ciri khas Twitter adalah memiliki fitur posting dengan ukuran maksimal 140 karakter [5]. Twitter digunakan sebagai pesan singkat (tweet) dari internet yang dapat digunakan untuk berbagai tujuan termasuk memposting *threat* dan *vulnerability*.

B. *Threat* and *Vulnerability*

Threat merupakan suatu ancaman yang terjadi pada keamanan sistem, yang dilakukan oleh penjahat dunia maya atau *cyber-criminal* dengan melakukan kejahatan seperti pencurian data dan aset teknologi, kekayaan intelektual, perusakan data dan digital lainnya. *Threat* ini dapat berasal antar negara yang bermusuhan, dan peretas yang mengeksploitasi *zero-day attack*, serta dari orang dalam organisasi yang menjadi mata-mata [6].

Vulnerability merupakan kerentanan sistem yang cukup tinggi yang menyerang *hardware, software,* protokol dan pada keamanan sistem komputer [7]. *Vulnerability* dimanfaatkan oleh *cyber-criminal* untuk mengeksploitasi celah keamanan dari suatu sistem untuk membahayakan keamanan sistem tersebut.

TABEL 1
CONTOH TWEET *THREAT* DAN *VULNERABILITY*

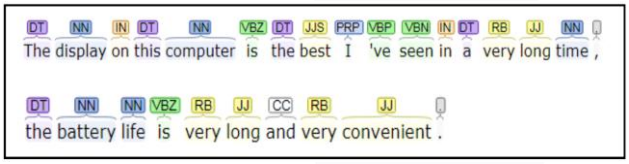
Tweet	Contoh Tweet
Threat	New research! Unpacking CVE-2021-40444: A Deep Technical Analysis of an Office RCE Exploit https://t.co/q2QwkRfrvM
Vulnerability	PoC for CVE-2022-21971 Windows Runtime Remote Code Execution Vulnerability.https://t.co/QtJ7xXtoKH #Pentesting #CVE #RCE #Windows #Vulnerability #Infosec

C. *Scraping Data*

Data *scraping* adalah teknik yang digunakan untuk mengekstrak data dari sebuah situs web. Data yang dikumpulkan berupa informasi yang diperlukan dari situs web dan informasi ini diubah menjadi file .csv. dan berupa data mentah [8]. *Scraping* data dapat menggunakan bahasa pemrograman Python dengan *Snscrape* Library yang berguna untuk mengambil informasi pada platform web seperti media sosial. Data tersebut akan ditambang dan diprogram untuk mendapatkan dataset yang dibutuhkan [9].

D. *Part-Of-Speech (POS) Tagging*

Part-Of-Speech (POS) atau POS *Tagging*, adalah salah satu fitur terpenting dalam susunan kata pada token. Kesalahan dalam *POS Tagging* seringkali menyebabkan kesalahan dalam memahami suatu kalimat [10] dalam proses pelabelan pada setiap token kata. Pada bagian ini, setiap token kata akan diberi label sesuai dengan kelas katanya seperti kata kerja, kata benda, kata keterangan, dan lain-lain.



GAMBAR 1.
ILUSTRASI LABEL KELAS KATA PADA POS TAG

TABEL 2
DAFTAR TAGSET PADA POS *TAGGING*

POS Tag	Nama POS
NN	Kata Benda
VB	Kata Kerja
RB	Kata Keterangan
VBT	Kata kerja Transitif
CC	Konjungsi Koordinatif
JJ	Kata Sifat

E. *Terms Frequency-Inverse Document Frequency (TF-IDF)*

Terms Frequency- Inverse Document Frequency atau TF-IDF adalah metode penghitungan pembobotan antara sebuah kata *(term)* terhadap dokumen atau data yang diberikan. Metode ini merupakan rumus untuk menghitung bobot pada data [11]. Untuk mencari nilai TF, IDF dan *weighted* TF – IDF, didefinisikan dalam persamaan di bawah ini:

$$tf(t,d) = c(t,d) \quad (1)$$

$$idf(t) = log\ \frac{N}{k} \quad (2)$$

$$weight(t,d) = tf(t,d) * idf(t) \quad (3)$$

Diimplementasikan *POS Tagging* ke dalam TF-IDF, kemudian ada parameter baru untuk menghitung nilai TF-IDF, berdasarkan model penelitian [12], *POS Tagging* adalah sebagai berikut:

$$W_{POS(t)} \begin{cases} w1\ , & jika\ t\ kata\ kerja/kata\ benda \\ w2\ , & jika\ t\ kata\ sifat/kata\ keterangan \\ w3\ , & selain\ itu \end{cases} \quad (4)$$

Dimana w1>w2>w3>0. Persamaan ini didasarkan pada penelitian, yang menyatakan bahwa kata kerja dan kata benda biasanya lebih penting daripada kata sifat atau kata keterangan, dan juga lebih penting daripada kata lain seperti kata ganti, preposisi dan konjungsi. Sehingga diperoleh persamaan TF-IDF yang dimodifikasi sebagai berikut:

$$\begin{cases} weightpos\ (t,d) = (\frac{c(t,d)}{I(d)} * wpos(t)) * (log\ \frac{N}{k}) \\ 0, \quad selain\ itu \end{cases} \quad (5)$$

Keterangan:
tf : frekuensi term
t : term
d : dokumen atau data
c(d,t) : frekuensi term pada data
I (d) : panjang data
Idf (t) : IDF
N : panjang semua data yang dianalisis
k : jumlah data yang akan ditemukan
Weightpos (t,d) : bobot keseluruhan untuk t sampai d dengan penambahan POS [13].

F. *Support Vector Machine (SVM)*

SVM adalah metode yang digunakan dalam klasifikasi kelas yang cukup populer di *Machine Learning* dengan mempelajari cara menetapkan label kelas dari *training* pada dataset. SVM digunakan untuk mencari *hyperplane* (batas keputusan) yang optimal sehingga *output* yang diperoleh dapat dipisahkan dari *input*nya. Selain itu, SVM juga berfungsi untuk memprediksi, mengklasifikasikan, dan meregresi data, karena SVM dapat mengabstraksi data sehingga dapat dikenali setelah dataset dilatih [14].

Persamaan Panjang Vektor adalah:

$$\|x\text{-}xi\| = \sqrt{x^2 + y^2} \quad (6)$$

Persamaan kernel RBF adalah:

$$K(x_i, x_j) = \exp\left(-\gamma \left|\left|x_i - x_j\right|\right|^2\right), \gamma > 0 \quad (7)$$

Persamaan *Lagrange Multiplier* adalah:

$$\text{Min } L_D = \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j x_i x_j y_i y_j - \sum_{i,j=1}^{n} \alpha_i \quad (8)$$

Persamaan *hyperplane* adalah:

$$f(x) = W^T.x + b \quad (9)$$

Dimana:

$$W^T.x + b = a^T.y\,K(X_{training}, X_{testing}) + b \quad (10)$$

Sehingga f(x) adalah:

$$f(x)=sgn(a^T.y\,exp(\gamma(X_{training}, X_{testing}) + b) \quad (11)$$

G. K-Fold Cross Validation

K-Fold Cross Validation dapat memberikan bias yang lebih rendah sehingga digunakan untuk mencari nilai akurasi dengan membagi data latih secara adil atau bergantian sesuai ketersediaan data dan membandingkan model untuk prediksi pada data [15].

H. Confusion Matrix

Confusion matrix digunakan untuk mengevaluasi kinerja dari klasifikasi dataset yang telah dilakukan, dengan menghitung *Accuracy, Precision, Recall, dan F1-Score.*

TABEL 3
CONFUSION MATRIX

Data Aktual	Data Predikisi	
	Negative	*Positive*
Negative	*True Negative*	*False Positive*
Positive	*False Negative*	*True Positive*

Pada tabel 3 dapat digambarkan hasil *output* klasifikasi berupa True Negative, False Negative, True Positive, False Positive. Yang dapat diartikan sebagai berikut:

1. *True Positive* (TP), teks yang sudah diklasifikasikan positif sebagai *Threat* dan *Vulnerability* oleh sistem.
2. *False Negative* (FN), teks yang sudah diklasifikasikan positif sebagai *Threat* dan *Vulnerability* namun diklasifikasikan negatif oleh sistem.
3. *True Negative* (TN), teks yang sudah diklasifikasikan negatif sebagai *Threat* dan *Vulnerability* oleh sistem.
4. *False Positive* (FP), teks yang sudah diklasifikasikan sebagai negatif *Threat* dan *Vulnerability* namun diklasifikasikan positif oleh sistem.

Untuk mengevaluasi dataset, dilakukanlah perhitungan sebagai berikut:

1. *Accuracy*: digunakan untuk memprediksi angka yang benar dan salah melalui kumpulan data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \text{ x } 100\% \quad (12)$$

2. *Precision* adalah tingkat keakuratan data yang diinginkan dengan data yang telah dihasilkan oleh sistem.

$$Precision = \frac{TP}{TP+FP} \text{ x } 100\% \quad (13)$$

3. *Recall* adalah tingkat keberhasilan sistem saat melakukan klasifikasi.

$$Recall = \frac{TP}{TP+FP}\ x\ 100\% \quad (14)$$

4. *F1-Score* adalah tingkat evaluasi antara presisi dan recall.

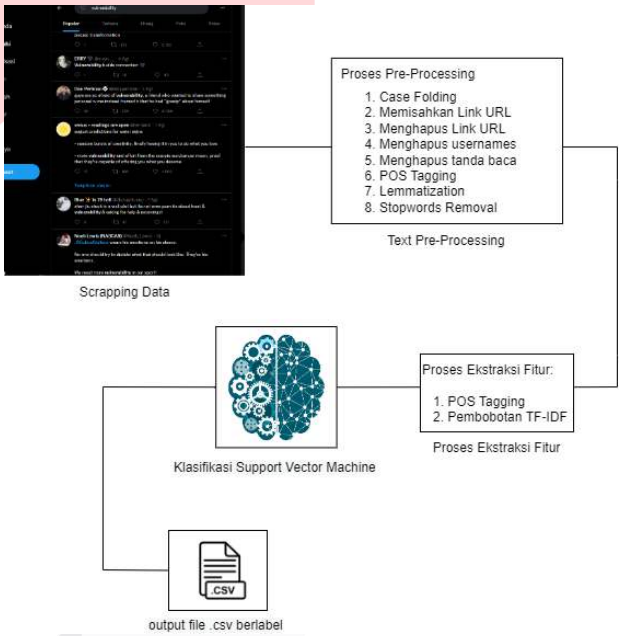
$$F1\ Score\ = 2\ x\ \frac{precision\ x\ recall}{precision+recall} \text{ x } 100\% \quad (15)$$

III. METODE

Konsep utama dari penelitian ini adalah perancangan pendeteksian *threat* dan *vulnerability* pada twitter dengan menggunakan algoritma Support Vector Machine (SVM) untuk mendeteksi dan melakukan uji performansi.

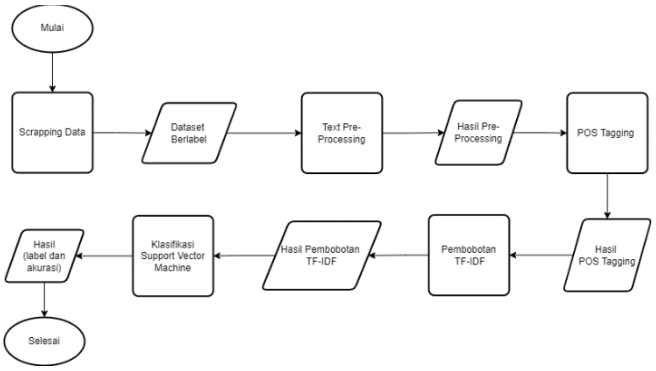
A. Gambaran Umum dan flowchart Sistem

Pada gambar 2 dibawah, menjelaskan mengenai gambaran umum tentang deteksi *threat* dan *vulnerability* dalam posting Twitter menggunakan algoritma SVM. Langkah pertama yaitu dengan mengumpulkan data tweet yang diambil menggunakan teknik web *scraping*, kemudian data tersebut akan diseleksi melalui *text preprocessing*. Setelah data dibersihkan, data akan diberikan pembobotan dengan POS *Tagging* dan TF-IDF. Hasilnya akan digunakan untuk klasifikasi menggunakan algoritma SVM untuk menghasilkan *output* berupa uji performansi dan pelabelan pada data.



GAMBAR 2.
GAMBARAN UMUM ALUR SISTEM

Data yang telah terpilih berjumlah 4270 data dan akan diberi label sebanyak 2 kelas data. Data positif sebanyak 2135 data yang merupakan tweet dengan kategori *threat* dan *vulnerability* diberi label dengan angka 1 dan data negatif dengan 2135 data yang merupakan tweet dengan kategori non-*threat* dan *vulnerability* diberi label dengan angka 0. Kata kunci yang digunakan untuk mengumpulkan data adalah *"cve", "vulnerability", "hack", "eksploitasi", "log4j"*. Gambaran dari penelitian ini adalah sebagai berikut:



GAMBAR 3.
FLOWCHART PERANCANGAN SISTEM

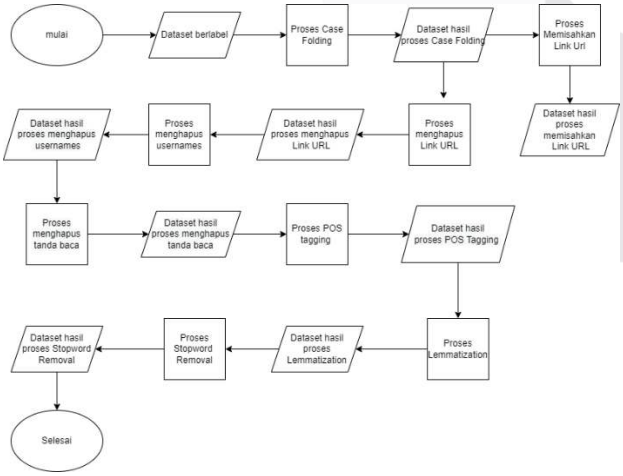
B. *Scraping* Data

Dataset diambil menggunakan *library* Python *Snscrape*. *Library snscrape* digunakan untuk mengambil data dalam format file .csv. Data diambil menggunakan Python dan berkisar dari rentang waktu pengambilan 1 Januari 2021 – 16 Maret 2022. Pertama, sistem akan melakukan pencarian berdasarkan kata kunci yang ditargetkan, seperti *"threat", "cve", "vulnerability", "hack", "exploit ","log4j"*. Dataset telah diberi label secara manual berupa label positif dan negatif. Contoh dataset yang telah di-*scrap* dan diberi label dari media sosial Twitter adalah sebagai berikut:

TABEL 4
DATASET DENGAN LABEL

No	Tweet	Label
1	Google Cloud recommendations for investigating and responding to the Apache "Log4j 2" vulnerability (CVE-2021-44228) https://t.co/LWNtATfnT1 #gcp https://t.co/yUy7nmXX6n	1
2	Log4j ruining my life rn	0
3	Apache Log4j 2 - Remote Code Execution(RCE). https://t.co/Vm54jjMyzn	1
4	Why is log4j doing anything other than printing stuff to files?	0

C. Text Preprocessing



GAMBAR 4.
FLOWCHART PROSES TEXT PREPROCESSING

Pada gambar 4 merupakan flowchart dari text preprocessing. Text Preprocessing adalah proses mengubah teks menjadi teks terstruktur sesuai kebutuhan. Text Preprocessing digunakan untuk memisahkan teks yang berupa pemecahan kata. Berawal dari sebuah paragraf menjadi kalimat dan kalimat yang nantinya akan dipecah menjadi kata-kata dan dari kata akan dapat menghilangkan angka, simbol, dan karakter pada kata tersebut. Hal ini digunakan untuk membersihkan dataset dengan menghilangkan beberapa gangguan yang dapat menyebabkan data sulit untuk diidentifikasi, sehingga data akan menghasilkan hasil berupa teks yang mudah dipahami oleh algoritma[16]. Tabel 5. merupakan hasil akhir dari data preprocessing yang telah dilakukan berdasarkan gambar 4.

TABEL 5 TWEET SEBELUM DAN SESUDAH TEXT PREPROCESSING

Tweet Sebelum text Pre-processing	Tweet Setelah text Pre-processing	Label
@angsuman: DirtyPipe – Linux Local Root Exploit.https://t.co/FsfVy0PWp2	dirtypipe linux local root exploit	1
@D0y0u3v3nl33t: Yea, sure. Well request a CVE.https://t.co/OuOoGG2HqN	yea sure well request cve	0

D. *POS Tagging dan TF-IDF*

Pada proses ini, POS *Tagging* berfungsi untuk mengelompokkan kata berdasarkan kelas kata sesuai tabel 2. Setelah itu, proses perhitungan TF-IDF akan disesuaikan berdasarkan nilai pembobotan menggunakan (4). Pada tabel 6, terdapat 2 contoh tweet yang telah diproses sebelumnya dan diberi label dengan masing-masing kelas kata menggunakan POS *Tagging*.

TABEL 6
CONTOH TWEET DENGAN POS *TAGGING*

Dok	Tweet	POS Tag	Label
D1	dirtypipe linux local root exploit	[(dirtypipe, NN), (linux, VBZ), (local, JJ), (root, NN), (exploit, NN)]	1
D2	yea sure well request cve	[(yea, NN), (sure, RB), (well, RB), (request, VB), (cve, NN)]	0

Pada tabel 6 diatas, tweet diberi label dengan masing-masing kelas kata, dan akan dilakukan perkalian berdasarkan label pada kelas kata sesuai (4). Kelas kata yang berlabel NN atau VB akan dikalikan lima. Untuk RB atau JJ akan dikalikan tiga. Dan untuk selain jenis kelas kata dalam persamaan itu akan bernilai satu.

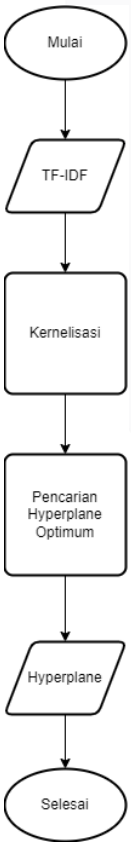
Untuk mendapatkan nilai TF di setiap dokumen dengan membagi 'Jumlah Kata' dengan 'Banyak Kata yang Muncul' di setiap dokumen atau menggunakan (1) dan (5). Untuk menghitung nilai IDF menggunakan (2) dan (5). Dan dihitung menggunakan (5) untuk mendapatkan TD-IDF.

TABEL 7
POS *TAGGING* DAN TF-IDF

Kata	Kelas Kata	POS Tag	Banyak kata muncul		Jumlah Kata		TF		IDF	TF-IDF	
			D1	D2	D1	D2	D1	D2		D1	D2
cve	NN	5	0	1	0	5	0	1	0,301	0	0,301
dirtypipe	NN	5	1	0	5	0	1	0	0,301	0,301	0
exploit	NN	5	1	0	5	0	1	0	0,301	0,301	0
linux	VBZ	1	1	0	1	0	0,2	0	0,301	0,0602	0
local	RB	3	1	0	3	0	0,6	0	0,301	0,1806	0
request	VB	5	0	1	0	5	0	1	0,301	0	0,301
root	NN	5	1	0	5	0	1	0	0,301	0,301	0
sure	RB	3	0	1	0	3	0	0,6	0,301	0	0,1806
well	RB	3	0	1	0	3	0	0,6	0,301	0	0,1806
yea	NN	5	0	1	0	5	0	1	0,301	0	0,301

E. Support Vector Machine

Secara umum, Support Vector Machine hanya dapat mengklasifikasikan data ke dalam dua kelas data. Dalam penelitian ini digunakan Kernel *Radial Basis Function* (RBF) dengan (6). Kernel RBF ini membutuhkan parameter gamma dan C. parameter C bertindak sebagai penalti kesalahan pada klasifikasi. X diambil dari vektor yang merupakan hasil vektorisasi. Exp adalah eksponen dari X dan gamma terhitung. Dan gamma berfungsi untuk pengambilan keputusan batas dan pengambilan keputusan. Misalnya, jika nilai gamma kecil maka batas keputusannya juga kecil, tetapi wilayahnya menjadi lebar dan sebaliknya.



GAMBAR 5.
FLOWCHART KLASIFIKASI SVM

Flowchart pada Gambar 5. dijelaskan bahwa tahap pertama adalah pengumpulan data pada pembobotan TF-IDF yang telah dilakukan. Kemudian dilakukan proses kernelisasi pada SVM yang menghasilkan *hyperplane*. Data yang digunakan adalah data non-linear, sehingga dengan menggunakan SVM non-linear perlu menggunakan kernel *trick* atau Kernel *Radial Basis Function* (RBF) karena kernel ini dapat memberikan performansi yang baik dengan parameter gamma dan C yang menghasilkan *training* yang memiliki nilai *error* yang kecil dibandingkan dengan kernel lainnya.

TABEL 8
HASIL PERHITUNGAN PANJANG VEKTOR

‖x-xi‖	$\sqrt{x^2+y^2}$	Hasil
‖x-x1‖	$\sqrt{(-0.699)^2+(-0.9398)^2+(-0.8194)^2+(-0.699)^2+(-0.699)^2}$	1,7162
‖x-x2‖	$\sqrt{(1.301)^2+(1.1806)^2+(1.1806)^2+(1.301)^2+(1.301)^2}$	7,4738

Setelah mendapatkan panjang vektor dengan menggunakan (6). Kernel RBF untuk perhitungan kernelisasi akan dilakukan dengan menggunakan (7) dengan γ = 0,5.

$$K(x_i, x_j) = \exp\left(-\gamma \left|\left|x_i - x_j\right|\right|^2\right), \gamma > 0$$
$$K(1,1) = \exp(-0.5\ (1{,}7162)^2)$$
$$K(1,1) = exp(-1.4725)$$
$$K(1,1) = -4.002$$
$$K(x_i, x_j) = \exp\left(-\gamma \left|\left|x_i - x_j\right|\right|^2\right), \gamma > 0$$
$$K(1,2) = \exp(-0.5\ (7.4738)^2)$$
$$K(1,2) = exp(-27.9288)$$
$$K(1,2) = -75.91$$

Setelah kernelisasi, cari *hyperplane* optimal dengan menghitung menggunakan (8), (9), (10). Dan hasil akhir untuk mendapatkan nilai f(x) menggunakan (11) untuk mencari *hyperplan*e adalah sebagai berikut:

$$\text{f(x)} = sgn\left(\begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix} exp(-0.5||X_{training}, X_{testing}||^2 + 1)\right)$$

IV. HASIL DAN PEMBAHASAN

A. Pengujian Data

Pada sistem yang dibangun, data akan diolah untuk *testing* dimana data akan dibagi menjadi data *training* dan *testing*. Pada pengujian data ini, dilakukan perbandingan rasio antara data latih dan pengujian yaitu 90% pelatihan dan 10% pengujian, 80% pelatihan dan 20% pengujian, 70% pelatihan dan 30% data pengujian, 60% pelatihan dan 40% pengujian, 50% pelatihan dan 50% pengujian, 40% pelatihan dan 60% pengujian, 30% pelatihan dan 70% pengujian, 20% pelatihan dan 80% pengujian, 10% pelatihan dan 90% data pengujian, dengan random_state= 0. Tujuan dilakukan pengujian ini yaitu untuk mendapatkan akurasi terbaik dari rasio data.

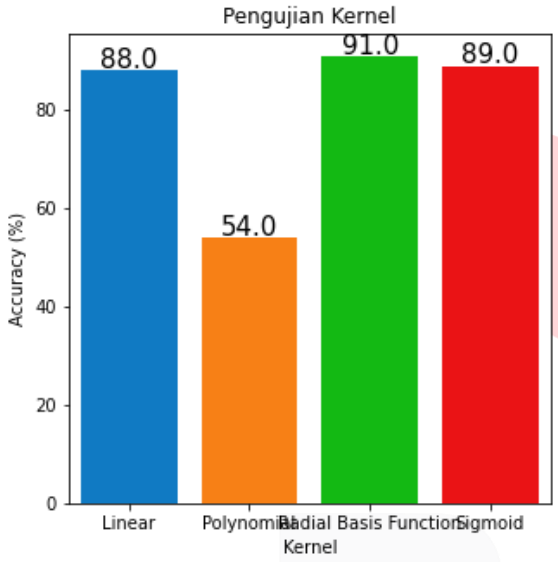
Untuk mengukur kinerja sistem, pengujian sistem yang digunakan berupa akurasi, presisi, recall, F1-Score dan K-*Fold Cross Validation.* Pengukuran dilakukan berdasarkan nilai yang terdapat pada *confusion matrix* pada setiap rasio perbandingan data yang digunakan. Setelah itu, diperoleh perbandingan rasio terbaik dan akan digunakan untuk menghitung parameter gamma dan C.

B. Pengujian Kernelisasi

Pada pengujian kernel ini dilakukan pengujian 4 kernel yang berbeda dengan perbandingan data *testing* dan *training* 10% : 90%. Kernel yang akan diuji meliputi kernel *linier,* kernel *polinomial,* kernel RBF dan kernel *Sigmoid.*

Tujuan dari pengujian kernel pada penelitian ini adalah untuk mengetahui performansi dari algoritma SVM dengan menggunakan 4 pengujian yang berbeda pada kernel yang berbeda. Kernel diuji untuk mengetahui rata-rata akurasi terbaik dari kernel yang akan digunakan.

Grafik dibawah ini adalah hasil dari pengujian kernel:



GAMBAR 6.
GRAFIK PENGUJIAN KERNELISASI

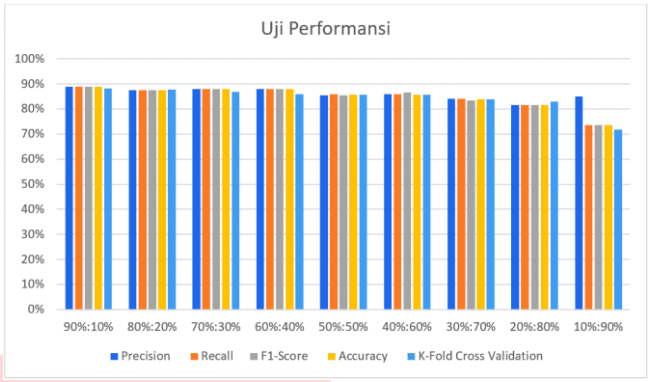
C. Uji Performansi

Berdasarkan Gambar 6, Pengujian Kinerja menggunakan kernel terbaik yaitu kernel RBF dengan rasio 10% : 90% dengan tujuan sistem menampilkan hasil perhitungan dengan menggunakan (13), (14), dan (15) dalam *confusion matrix* dan sekaligus mengetahui nilai parameter *Precision, Recall, dan F1-Score,* seperti terlihat pada Tabel 9 di bawah ini:

TABEL 9
CONFUSION MATRIX PADA PARTISI DATA

Sample	Training Data (%)	Testing Data (%)	Actual Label	Prediction Label		*Precision*	*Recall*	*F1- Score*
				positive	negative			
1	90	10	positive	185	21	**90**	**88**	**89**
			negative	26	195	**88**	**90**	**89**
2	80	20	positive	362	55	88	88	88
			negative	52	385	87	87	87
3	70	30	positive	550	81	88	89	88
			negative	74	576	88	87	88
4	60	40	positive	550	81	88	89	88
			negative	74	576	88	87	88
5	50	50	positive	931	155	85	86	85
			negative	152	897	86	86	86
6	40	60	positive	1099	186	86	86	86
			negative	175	1102	86	86	86
7	30	70	positive	1203	318	80	89	84
			negative	164	1304	88	79	83
8	20	80	positive	1356	358	80	84	82
			negative	271	1431	83	79	81
9	10	90	positive	1245	690	70	83	76
			negative	321	1587	80	64	71

Dari hasil Tabel di atas, maka hasil rata-rata dapat dilihat pada grafik di bawah ini dengan Akurasi, dan *K-Fold Cross Validation.*



GAMBAR 7.
GRAFIK UJI PERFORMANSI

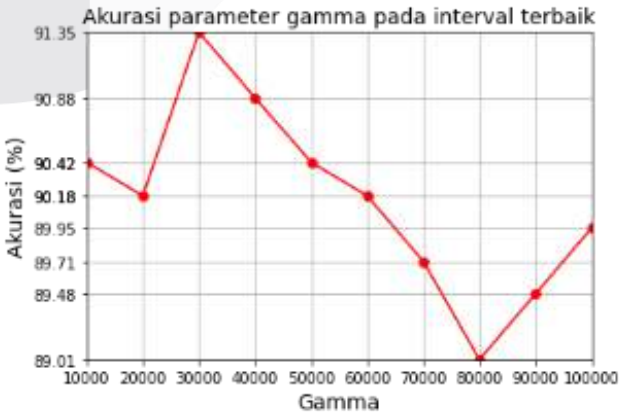
D. Pengujian Parameter Gamma

Pengujian pada parameter gamma ini digunakan sebagai jarak pengaruh pada titik pelatihan, nilai gamma yang didapatkan, akan diuji kembali untuk mencari interval terbaik agar mendapatkan hasil yang lebih baik lagi. Hasilnya sebagai berikut:

TABEL 10
PENGUJIAN PARAMETER GAMMA

Pengujian	Gamma(γ)	Akurasi (%)
1	0,000001	50
2	0,00001	50
3	0,0001	65.65
4	0,001	50
5	0,01	87.61
6	0.1	85.98
7	1	88.08
8	100	88.08
9	1000	88.31
10	**10000**	**90.42**
11	**100000**	**89.95**

Dari Tabel 10 diatas didapatkan hasil yang optimal dengan nilai gamma pada interval 10000 sampai dengan 100000. Tahap selanjutnya dilakukan pengujian gamma pada interval tersebut untuk mengetahui hasil yang optimal. Dapat dijelaskan pada gambar grafik dibawah ini.

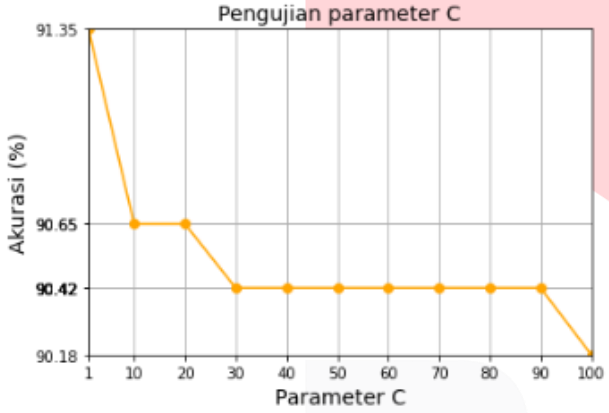


GAMBAR 8.
GRAFIK PENGUJIAN GAMMA PADA INTERVAL TERBAIK

Dari hasil pengujian diatas, didapatkan hasil gamma yang optimal, yaitu nilai gamma terbaik adalah 30000 dengan akurasi 91,35%. Dari akurasi yang didapat, bahwa dataset yang digunakan sebaiknya menggunakan parameter gamma dengan nilai yang besar yang artinya memiliki radius jangkauan yang luas karena Support Vector Machine yang didapat tersebar secara luas.

E. Pengujian Parameter C

Pada hasil pengujian ini terlihat bahwa nilai C terbaik terletak pada C sebesar 1 dengan akurasi 91,35%. Dari hasil yang diperoleh dapat disimpulkan bahwa dataset yang digunakan memiliki *penalty* bias yang rendah karena ketika nilai parameter C rendah maka akan mampu memberikan margin *hyperplane* yang cukup besar. Ini mengorbankan beberapa untuk bias atau kesalahan klasifikasi.



GAMBAR 9.
GRAFIK PENGUJIAN PARAMETER C

Dengan hasil tersebut dapat dinyatakan bahwa gamma dengan nilai 30000 dan C dengan nilai 1 mendapatkan hasil yang paling optimal dalam pengujian ini. Sedangkan parameter C digunakan untuk mengetahui seberapa besar algoritma pada data yang salah diprediksi. Semakin besar nilai parameter C maka akan membantu meminimalkan jumlah kesalahan dalam prediksi, tetapi sebaliknya jika nilai C rendah maka tingkat kesalahan dalam prediksi akan semakin besar.

F. Pengujian Validasi

Pengujian validasi dilakukan dengan melakukan *scraping* pada dataset baru dengan berbagai kata kunci. Dataset baru tersebut kemudian diberi label secara manual, untuk melihat keakuratan sistem. Ada beberapa hasil pengujian keluaran berbagai data baru dengan kata kunci yang berbeda, sebagai berikut:

TABEL 11
UJI VALIDASI DENGAN DATA TESTING BARU

NO	Kata kunci	Jumlah data	Label		Akurasi
			Positive	*Negative*	
1	*Vulnerability*	25 data	13 data	12 data	66,66%
2	CVE	24 data	17 data	7 data	100%
3	Hack_git	25 data	24 data	1 data	100%

V. KESIMPULAN

Berdasarkan pada penelitian dan pengujian yang telah dilakukan, berikut adalah kesimpulan yang dapat diambil:

1. Sistem telah berhasil mendeteksi *threat* dan *vulnerability* pada postingan Twitter menggunakan algoritma Support Vector Machine dengan memberikan label positif dan negatif pada dataset.
2. Uji performansi diperoleh akurasi sebesar 89,0%, nilai presisi sebesar 89%, nilai *recall* sebesar 89%, dan nilai *F1-Score* sebesar 89%, dengan distribusi rasio data pengujian dan data latih 10%: 90%. Parameter C dan Gamma terbaik adalah 1 dan 30.000.

REFERENSI

[1] A. Baccouche, S. Ahmed, D. Sierra-Sosa, and A. Elmaghraby, "Malicious text identification: Deep learning from public comments and emails," Information (Switzerland), vol. 11, no. 6, 2020, doi: 10.3390/info11060312.

[2] S. Samtani, R. Chinn, H. Chen, and J. F. Nunamaker, "Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence," Journal of Management Information Systems, vol. 34, no. 4, pp. 1023–1053, 2017, doi: 10.1080/07421222.2017.1394049.

[3] A. T. Tunggal, "What is a Cyber Threat?," UpGuard, 2021. https://www.upguard.com/blog/cyber-threat (accessed Dec. 16, 2021).

[4] A. T. Tunggal, "What is a Vulnerability?," UpGuard, 2021. https://www.upguard.com/blog/vulnerability (accessed Dec. 16, 2021).

[5] L. Qiu, H. Lin, J. Ramsay, and F. Yang, "You are what you tweet: Personality expression and perception on Twitter," Journal of Research in Personality, vol. 46, pp. 710–718, Dec. 2012, doi: 10.1016/j.jrp.2012.08.008.

[6] L. Liu, C. Chen, J. Zhang, O. De Vel, and Y. Xiang, "Insider Threat Identification Using the Simultaneous Neural Learning of Multi-Source Logs," IEEE Access, vol. 7, pp. 183162–183176, 2019, doi: 10.1109/ACCESS.2019.2957055.

[7] K. Liu, "Vulnerability Severity Prediction With Deep Neural Network," 2019 5th International Conference on Big Data and Information Analytics (BigDIA), pp. 114–119, 2019.

[8] B. Zhao, "Web Scraping," no. May 2017, 2018, doi: 10.1007/978-3-319-32001-4.

[9] A. T. Korkmaz, M. Ulaş, M. Karabatak, Y. Santur, and A. Tunahan Korkmaz, "Data Mining in Finance: Concepts, Trend and Applications", Accessed: Aug. 14, 2022. [Online]. Available: https://www.icaens.com/

[10] Y. Chen, "An english POS tagging approach based on maximum entropy," Proceedings - 2015 International Conference on Intelligent Transportation, Big Data and Smart City, ICITBS 2015, pp. 81–84, Jan. 2016, doi: 10.1109/ICITBS.2015.26.

[11] A. S. Shafie, N. M. Sharef, M. A. A. Murad, and A. Azman, "Aspect Extraction Performance with POS Tag Pattern of Dependency Relation in Aspect-based Sentiment Analysis," Proceedings - 2018 4th International Conference on Information Retrieval and Knowledge Management: Diving into Data Sciences, CAMP 2018, pp. 107–112, Sep. 2018, doi: 10.1109/INFRKM.2018.8464692.

[12] E. Erizal and C. Setianingsih, "Hate Speech Detection in Indonesian Language on Instagram Comment Section Using Maximum Entropy Classification Method," 2019 International Conference on Information and Communications Technology (ICOIACT), pp. 533–538, 2019.

[13] R. Xu, "POS weighted TF-IDF algorithm and its application for an MOOC search engine," ICALIP 2014 - 2014 International Conference on Audio, Language and Image Processing, Proceedings, pp. 868–873, Jan. 2015, doi: 10.1109/ICALIP.2014.7009919.

[14] M. Amjad, N. Ashraf, A. Zhila, G. Sidorov, A. Zubiaga, and A. Gelbukh, "Threatening Language Detection and Target Identification in Urdu Tweets," IEEE Access, vol. 9, pp. 128302–128313, 2021, doi: 10.1109/ACCESS.2021.3112500.

[15] H. Tabrizchi, M. M. Javidi, and V. Amirzadeh, "Estimates of residential building energy consumption using a multi-verse

optimizer-based support vector machine with k-fold cross-validation," Evolving Systems, vol. 12, no. 3, pp. 755–767, Sep. 2021, doi: 10.1007/S12530-019-09283-8.

[16] K. L. Sumathy, "Text Mining : Concepts , Applications , Tools and Issues – An Overview," vol. 80, no. 4, pp. 29–32, 2013.