Deteksi Threat Dan Vulnerability Pada Unggahan Twitter Menggunakan Algoritma Naïve Bayes

1st Paulin Al Imady
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
aalimady@student.telkomuniversity.ac.id

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

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

Abstrak— Twitter merupakan platform media sosial yang menjadi tempat bagi banyak orang untuk dapat mengunggah berbagai hal, tidak terkecuali unggahan yang mengandung unsur ancaman keamanan suatu sistem. Tentunya ini merupakan hal yang berbahaya jika seseorang mengunggah celah keamanan suatu sistem. Ancaman sistem yang dipublikasi dapat disalah gunakan oleh orang lain sehingga merugikan pemilik sistem. Untuk mengantisipasi hal ini, maka dibuat sistem untuk mendeteksi unggahan yang mengandung unsur ancaman (*threat*) dan kerentanan (*vulnerability*) sistem pada media sosial Twitter. Sistem ini menerapkan algoritma *text processing* yang menggunakan *metode* Naïve Bayes dan TF-IDF (*Term Frequency – Inverse Document Frequency*). Metode ini dipilih karena dianggap dapat menghasilkan akurasi yang baik meskipun dengan data training yang sedikit. Pada penelitian Tugas Akhir ini, hasil akhir yang didapatkan adalah sistem dapat membedakan tweet yang mengandung unsur *threat* atau *vulnerability*, dan yang tidak. Dengan rasio pembagian dataset ke dalam data *training* dan data *testing* adalah 70%:30% dan 80%:20%, keduanya mendapatkan nilai akurasi sebesar 88%, nilai presisi sebesar 88%, recall sebesar 88%, dan F1 score sebesar 88%.

Kata Kunci: *text* mining, naïve bayes, TF-IDF, *threat*, *vulnerabilities*, klasifikasi teks.

I. PENDAHULUAN

Menurut statista.com, pengguna internet pada tahun 2021 mencapai 4,6 miliar jiwa, dan diantara jumlah tersebut, 4,2 miliar pengguna internet adalah pengguna media sosial, yang diantaranya yaitu Twitter. Media sosail merupakan platform digital yang digunakan para penggunanya untuk dapat saling berbagi informasi, berkomunikasi, dan membangun relasi antara sesama pengguna media sosial[1].

Seiring dengan kebebasan penggunaan media sosial, para penggunanya dapat mengunggah berbagai hal, semisal hal-hal informatif seperti *update* berita terkini. Namun ditemukan juga beberapa pengguna yang mengunggah hal-hal yang berbahaya[2]. Sebagai contoh, pengguna yang mengunggah unggahan yang mengandung ancaman (*threat*) dan/atau kerentanan (*vulnerability*) terhadap keamanan suatu sistem di media sosial, dalam hal ini khususnya media sosial Twitter.

Pada era digitalisasi seperti saat ini, *therat* dan *vulnerability* terhadap keamanan suatu sistem menjadi perhatian tersendiri[3]. *Threat* dan *vulnerability* yang dipublikasi di media sosial tentunya dapat merugikan pemilik sistem, karena dikhawatirkan akan disalah gunakan oleh orang-orang yang melihat postingan tersebut. Sehingga, dalam rangka mendeteksi publikasi *threat* dan *vulnerability* terhadap keamanan suatu sistem di media sosial, dibuat sistem *text mining* sebagai metode pendeteksi *threat* dan *vulnerability* pada Twitter menggunakan algortima naïve bayes dan TF-IDF (*Term Frequency – Inverse Document Frequency*).

Dengan harapan postingan yang mengandung *threat* dan *vulnerability* terhadap keamanan suatu sistem dapat terdeteksi untuk bisa diambil tindakan lebih lanjut.

Menurut penelitian yang dilakukan oleh Dinda Ayu Muthia[4]. (2018) dengan judul "Komparasi Algoritma Klasifikasi *Text Mining* Untuk Analisis Sentimen Pada *Review* Restoran", mendapati hasil perbandingan antara algoritma Naïve Bayes dan Support Vector Machine menunjukkan hasil akurasi sebesar 87% oleh algoritma Naïve Bayes, sedangkan algoritma Support Vector Machine menunjukkan hasil akurasi sebesar 56%. Dari penelitian tersebut didapati bahwa implementasi algoritma Naïve Bayes untuk melakukan *text mining* menunjukkan hasil baik dan akurasi yang tinggi. Oleh karena itu, sistem pendeteksi *threat* dan *vulnerability* pada twitter dibuat menggunakan algoritma Naïve Bayes.

Tugas akhir ini bertujuan untuk membuat sistem yang dapat membedakan tweet yang mengnadung unsur *threat* atau *vulnerability*, dan yang bukan, dengan menggunakan algoritma pembobotan TF-IDF dan algoritma klasifikasi Naïve Bayes.

II. KAJIAN TEORI

A. Twitter

Sosial media menjadi wadah bagi masyarakat untuk saling berinteraksi, seperti membagikan berita, berbagi pengetahuan, berbagi pengalaman, berdagang, dan lain sebagainya. Salah satu media sosial yang digemari masyarakat yaitu Twitter. Pengguna Twitter merupakan masyarakat dari berbagai kalangan, mulai dari politikus, pelajar, ilmuan, *public figure* dan masih banyak lagi[5]. Hal ini menjadikan hal-hal yang diunggah oleh pengguna Twitter menjadi sangat bervariasi.

Seiring berjalannya waktu, pengguna Twitter semakin bertambah banyak, sehingga semakin banyak pula unggahan pada Twitter (tweet) yang diunggah setiap harinya. Hal ini berarti pada sosial media Twitter tersedia data dalam jumlah yang sangat besar, yang tentunya dapat dimanfaatkan oleh para peneliti untuk keperluan analisis[6].

B. Threat dan *Vulnerability*

Dalam perkembangan dunia modern seperti saat ini, kegiatan manusia tidak bisa dipisahkan dari sistem digital, mulai dari ranah bisnis, belanja online, periklanan, servis, dan masih banyak lagi. Hal ini mengakibatkan semakin luasnya pemanfaatan sistem digital. Seiring berkembangnya dunia digital, secara tidak langsung terjadi juga peningkatan cyber-criminal, atau bisa disebut penjahat dalam dunia maya[7].

Para *cyber-criminal* bekerja dengan melakukan eksplotasi ancaman (*threat*) dan kerentanan (*vulnerability*) pada sistem digital dengan perantara komputer, sehingga menimbulkan kerugian terhadap orang lain[8]. Beberapa jenis *threat* dan *vulberability* yang dieksploitasi oleh para *cyber-criminal* beberapa di antaranya yaitu Cross-site scripting (XSS), Zero-days, Denial-of-service (DOS), SQL injection, dan lain sebagainya.

Contoh *threat* dan *vulnerability* yang diunggah pada sosial media twitter dapat dilihat pada tabel berikut:

TABEL 1
CONTOH *THREAT* DAN *VULNERABILITY* PADA TWEET

Kategori	*Tweet*
Threat	*Clear out the Linux root password using CVE-2022-0847 - https://t.co/QR7QeDqCvE ./exploit https://t.co/gc4LWDUJS5*
Vulnerability	*OPdirect Tool to automate open redirect from a website. OPdirect is a tool to automate open redirect from a website. This tool helps #bugbounty hunter and #penetration tester to get open redirect #vulnerability for the domain they are targeting. OPdire... https://t.co/aA1SEP0txy https://t.co/laH9JmcMR4'*

C. Machine Learning

Machine Learning (pembelajaran mesin) adalah cabang dari *artificial intelligence*, dimana mesin akan dilatih dengan menggunakan algoritma tertentu[9]. Tujuan dari *machine learning* adalah mengajari mesin untuk dapat memahami data, sehingga mesin dapat memberikan *output* yang sesuai dengan keinginan pembuat *machine learning* tersebut[10].

D. Web Scraping

Situs web bisa menjadi sumber informasi yang sangat berguna bagi banyak orang, karena berbagai situs web yang ada menyediakan berbagai informasi bagi orang-orang yang membutuhkan informasi tertentu. Tujuan proses web scraping yaitu mengumpulkan data dari satu atau lebih situs web, dan mengekstraksinya menjadi susunan data yang ter-struktur seperti spreadsheet, database, atau csv[11]. Secara umum, *web scraping* dapat didefinisikan sebagai proses penggalian data dari situs web untuk menghasilkan data terstruktur dari data yang tidak ter sturktur[12]. Selanjutnya data yang telah terstruktur dapat dimanfaatkan lebih lanjut sesuai dengan kebutuhan.

Dalam proses *web scraping*, terdapat beberapa pilihan metode yang dapat digunakan. Cara tradisional yang paling mudah dalam praktiknya adalah dengan secara manual melakukan *copy* dan *paste* terhadap data yang ingin didapatkan. Tetapi tentu saja cara ini akan memakan banyak waktu jika pengguna ingin melakukan *web scraping* dalam jumlah data yang besar. Beberapa opsi lain yang dapat digunakan untuk memudahkan proses *web scraping* dalam jumlah data yang banyak yaitu menggunakan *software web scraping*, HTML parsing, API, extensi *web scraping*, dan *library web scraping* pada suatu bahasa pemrograman[13].

E. Text Mining

Text mining merupakan proses untuk mengekstraksi data dari data yang tidak terstruktur, dalam hal ini data yang dimaksud merupakan data tekstual. Secara garis besar, *text mining* terdiri dari tiga aktivitas utama, yaitu mengumpulkan berbagai teks yang relevan, ekstraksi informasi, dan yang terakhir merupakan data mining atau bisa disebut penambangan data[14].

Perbedaan *data mining* dan *text mining* terletak pada sumber data yang digunakan. Pada *data mining*, data yang digunakan adalah data yang terstruktur dari sebuah sistem, seperti spreadsheet, database, ARP, dan CRM. Sedangkan *text mining*, menggunakan sumber data yang tidak terstruktur, seperti data yang didapat dari email, situs web, dan media sosial[15].

F. Text Pre-Processing

Text Pre-processing merupakan tahap persiapan data tekstual dalam implementasi *text mining*. Secara garis besar, terdapat beberapa tahap dalam melakukan *text pre-processing*, yaitu:

1. Tokenizing

Pada tahap ini dokumen dipecah menjadi kumpulan kata. *Tokenization* dilakukan dengan menghilangkan setiap tanda baca dan memisahkan setiap kata per spasi. Dalam tahap ini juga dilakukan pengubahan setiap kata ke bentuk huruf kecil (*lower case*).

2. Stopword removal

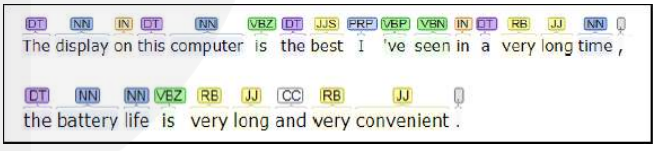
Stopword removal merupakan tahap untuk mengambil kata-kata yang tidak penting dari hasil *tokenizing*, hal ini dilakukan dengan cara menghilangkan *stop word*, semisal *the, was, is*, dan lainnya.

3. Stemming / lemmatization.

Pada tahap *stemming / lemmatization*, semua kata dikembalikan pada kata asalnya, biasanya dilakukan dengan cara menghilangkan kata imbuhan yang ada pada kata dasar[16].

G. Part-of-Speech (POS) Tagging

Part-of-speech atau biasa disebut *pos tagging*, merupakan suatu proses pelabelan setiap token kata. Dalam *POS Tagging*, setiap token kata dilabeli dengan kelas kata dari kata tersebut, semisal kata kerja (*verb*), kata benda (*noun*), kata keterangan (*adverb*), dan lain-lain. Proses ini biasanya diterapkan pada aplikasi Natural Language Processing[17].



GAMBAR 1
ILUSTRASI PELABELAN PADA *POS TAGGING*[18].

Gambar 1 menunjukkan hasil dari proses *POS Tagging* yang diaplikasikan terhadap suatu kalimat. Dalam proses pelabelan, *POS Tagging* tidak hanya mengacu pada kata yang diproses, namun juga mempertimbangkan posisi kata dalam sebuah kalimat, sehingga menghindari ambiguitas makna kata[18].

TABEL 2.
DAFTAR ARTI LABEL PADA POS TAG[19].

Label	Devinisi
UKW	*Unknown word*
CC	Konjungsi koordinatif
CD	Bilangan pokok
DT	Kata sandang
IN	Kata depan atau konjungsi subordinatif
JJ	Kata sifat
MD	Kata kerja bantu

Label	Devinisi
NN	Kata benda
NNP	Kata benda spesifik
PRP	Kata ganti
QT	Pembilang
RB	Kata keterangan
SYM	Simbol, termasuk semua jenis tanda baca
UH	Kta seru
VB	Kata kerja
WH	*what* (apa dalam bahasa inggris)

H. TF-IDF (*Term Frequency – Inverse Document Frequency*)

Algoritma TF-IDF merupakan algoritma yang digunakan dalam melakukan pembobotan hubungan suatu kata (*term*) dalam dokumen. Hal ini dilakukan dengan cara menghintung seberapa banyak frekuensi kemunculan kata tertentu dalam dokumen dan inverse frekuensi kemunculannya. Semakin besar frekuensi kemunculan suatu kata dalam dokumen yang ada, maka kata tersebut akan semakin dianggap penting[20].

Dalam TF-IDF, terdapat dua tahap perhitungan nilai, yaitu perhitungan nilai TF (*Term Frequency*), dan perhitungan nilai IDF (*Inverse Document Frequency*). Selanjutnya, nilai TF-IDF didapat dari perkalian dua nilai yang didapat dari kalkulasi nilai TF dan IDF.

Adapun rumus yang digunakan dalam algoritma TF-IDF adalah sebagai berikut:

$$idf(t,D)=\log(N/(df(t)+1)) \quad (1)$$

N merupakan jumlah total dokumen, dan penambahan satu dilakukan untuk menghindari terjadinya pembagian terhadap nol jika df(t) tidak ditemukan.

I. TF (*Term Frequency*)

Perhitungan TF (*Term frequency*) mengacu pada seberapa banyak suatu kata dalam satu dokumen, dibagi dengan jumlah seluruh kata yang terdapat pada dokumen tersebut. Semisal dokumen x memiliki seratus kata, dan kata "api" muncul lima kali dalam dokumen tersebut, maka kalkulasi untuk menemukan nilai TF dari kata "api" adalah sebagai berikut[21]:

$$TF=5 \div 100 = 0{,}05 \quad (2)$$

J. IDF (*Inverse Document Frequency*)

Dalam IDF (*Inverse Document Frequency*), jumlah suatu kata akan dihitung kemunculanya dalam jumlah dokumen yang ada. Semisal terdapat sepuluh dokumen, dan kata "api" kemunculannya adalah pada sebanyak lima dokumen, maka kalkulasi untuk menemukan nilai IDF dari kata "api" adalah[21]:

$$IDF=\log(10 \div 5)=0.3010 \quad (3)$$

K. Naïve Bayes

Algoritma klasifikasi Naïve Bayes sendiri merupakan metode klasifikasi yang berdasarkan pada teorema Bayes. Algoirtma Naïve Bayes biasanya diimplementasikan dalam sistem text processing. Algoritma ini melakukan klasifikasi probabilitas sederhana, yang dalam pengaplikasiannya menggunakan teorema bayes. Algoritma klasifikasi Naïve Bayes dapat melakukan olah data kuantitatif dan data diskrit yang hanya membutuhkan sedikit data pelatihan untuk memperhitungkan estimasi peluang yang diperlukan untuk klasifikasi. Dibandingkan dengan algortima klasifikasi yang lain, perhitungan algoritma Naïve Bayes tergolong lebih cepat karena hanya melakukan pengujian probabilitas dengan menemukan *class* yang sama dari data training[22].

Persamaan dari torema Byes dapat dijelaskan sebagai berikut[23]:

$$P(H \mid X)=(P(X \mid H).P(H))/(P(X)) \quad (4)$$

Keterangan:

X : Data yang belum diketahui kategori class-nya
H : Hipotesis data X merupakan suatu class yang spesifik
P(H|X) : Probabilitas hipotesis (H) berdasarkan kondisi (X)
P(H) : Probabilitas hipotesis (H)
P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis H
P(X) : Probabilitas X

Pada algoritma Naïve Bayes, proses klasifikasi membutuhkan petunjuk agar dapat menentukan class yang cocok bagi data yang diklasifikasi. Maka dari itu, rumus diatas disesuaikan menjadi seperti berikut:

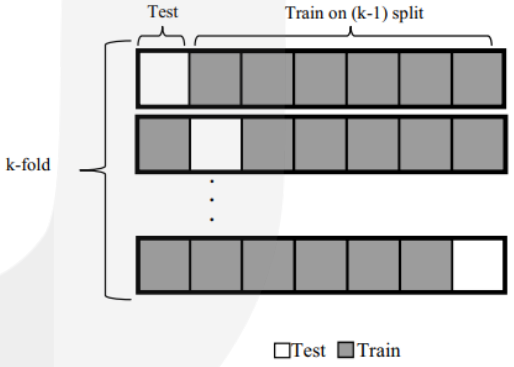
$$P(C \mid F1\ldots Fn)=(P(C)P(F1\ldots Fn|C))/(P(F1\ldots Fn)) \quad (5)$$

Keterangan:

C : Class
F1…Fn: Karakterisitik petunjuk untuk melakukan klasifikasi

L. K-fold Cross Validation

Salah satu langkah yang sangat diperlukan untuk memperkirakan nilai akurasi dari sampel baru adalah klasifikasi. *K-Fold Cross Validation* merupakan salah satu jenis klasifikasi untuk menemukan nilai akurasi dari hasil training suatu *machine learning*. Berbeda dengan algoritma klasifikasi lain yang hanya menggunakan satu pembagian data menjadi *data training* dan *data testing*, *K-Fold Cross Validation* membagi *data training* dan *data testing* secara bergantian, sehingga semua *dataset* memiliki andil dalam peran penjadi *data training* dan *data testing*[24].



GAMBAR 2
ILUSTRASI *K-FOLD CROSS VALIDATION* [24].

Dalam prosesnya, pengguna menentukan jumlah k, yang mana jumlah k mengindikasikan seberapa banyak pembagian data menjadi sejumlah *fold* (lipatan). Selanjutnya salah satu *fold* data akan menjadi *data testing*, sementara *fold* data lainnya digunakan sebagai *data training*. Proses ini akan diulang sejumlah banyaknya k yang telah ditentukan, untuk menguji semua *fold* data sebagai *data testing*. Proses akan berakhir setelah semua *fold* data telah mendapat giliran untuk dijadikan data testing[25].

Hasil akurasi dari *K-Fold Cross Validation* didapat dari nilai rata-rata seluruh hasil pengujian fold.

M. Confusion Matrix

Confusion matrix banyak digunakan dalam *machine learning* sebagai pengukur kinerja dari hasil model klasifikasi yang digunakan. *Confusion matrix* digambarkan dengan sebuah tabel yang merepresentasikan nilai dari empat parameter yang dimiliki *confusion matrix*, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN)[26].

TABEL 3
TABEL REPRESENTASI NILAI PARAMETER *COUNFUSION MATRIX*[27].

		True Values	
		True	False
Prediction	True	TP Correst result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

True positive (TP) merupakan data *positive* yang benar terdeteksi sebagai data *positive*, sedengakan *false positive* (FP) merupakan data *negative* yang terdeteksi oleh model sebagai data *positive*. Begitu pula sebaliknya, *false negative* (FN) merupakan data *positive* yang terseteksi model sebagai data *negative*, dan *true negative* (TN) merupakan data *negative* yang benar terdeteksi sebagai data *negative*[27].

Selanjutnya hasil kalkulasi dari *confusion matrix* diimplementasikan untuk menghitung *accuracy*, *precision*, *F1 Score, dan recall*, dengan rumus yang tertera pada persamaan berikut[28]:

Accuracy

$$accuracy = (TP+TN)/Total \quad (6)$$

Precision

$$precision = TP/(TP+FP) \quad (7)$$

Recall

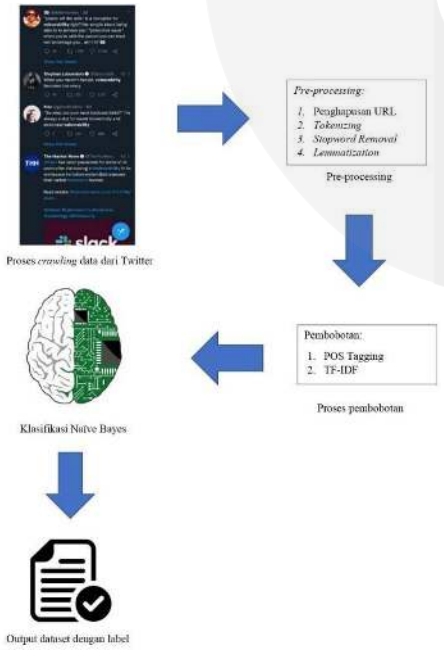
$$recall = TP/(TP+FN) \quad (8)$$

F1 Score

$$F1\ Score = 2\times((recall\times precission))/((recall+precission)) \quad (9)$$

III. METODE

A. Desain Sistem



GAMBAR 3 GAMBARAN UMUM ALUR SISTEM

Seperti yang ditunjukkan pada gambar 3, dalam implementasinya, pada tahap awal sistem akan melakukan scraping data dari Twitter dengan menggunakan *library* Python bernama snscraper. Kemudian data yang didapat dari *web scraping* akan dilanjutkan ke proses *text pre-processing*, agar data dapat diolah untuk dilakukan pembobotan dengan menggunakan algoritma TF-IDF. Setelah dilakukan proses pembobotan dengan TF-IDF, proses selanjutnya yaitu *text classification* menggunakan algoritma klasifikasi Naïve Bayes. Pada tahap akhir, sistem akan mengeluarkan *output* berupa dataset yang sudah memiliki label positif atau negatif.

B. Kebutuhan Data.

Adapun data yang dibutuhkan dalam keberlangsungan berjalannya sistem adalah:

1. *Dataset* berbahasa inggris dari Twitter yang disimpan dalam bentuk .csv dan telah memiliki label positif dan negatif. Data terdiri dari 4274 data, yang teridiri dari 2137 data dengan label positif dan 2137 data dengan label negatif.

TABEL 4
CONTOH DATA TWEET YANG TELAH DIBERI LABEL

Text	Label
I was just talking about the impact of this vulnerability today. Log4j supports MQTT broker.	0
Analysis - Log4j doesnt just blow a hole in your servers, its reopening that can of worms: Is Big Bizexploiting open source? Would more money have prevented this security flaw? Would the cash be useful in other ways anyway? |	1
The Log4J Vulnerability Will Haunt the Internet for Years	0
Incase you missed it Picus have a listed IPS signatures for #log4shell from various vendors in this article.	1

2. *Dataset* berbahasa inggris dari Twitter yang disimpan dalam bentuk .csv dan tidak memiliki label apapun.

C. Kebutuhan Perangkat Lunak.

Perangkat lunak yang digunakan dalam perancangan sistem deteksi *threat* dan *vulnerability* pada Twitter menggunakan algortima Naïve Bayes, dengan spesifikasi:

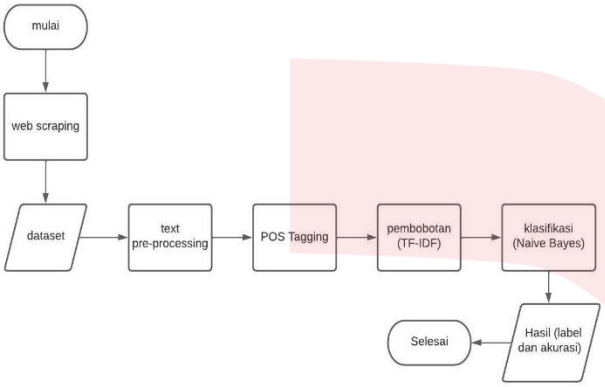
1. Sistem operasi Microsoft Windows 10 64-bit.
2. Bahasa pemrograman Python versi 3.9.6. sebagai bahasa pemrograman yang digunakan dalam pembuatan sistem.
3. Snscrape *library*, merupakan *library* pada python yang digunakan untuk melakukan *web scraping*.
4. Pandas *library*, digunakan sebagai pengolahan data dalam bentuk tabel
5. Natural Language Toolkit (NLTK) module, digunakan dalam melakukan pemrosesan bahasa tekstual
6. Regex library, digunakan untuk mencari karakter dengan pola tertentu pada dataset
7. Numpy library, digunakan dalam kalkulasi data numerik

D. Kebutuhan Pengguna (Brainware)

Untuk dapat mengoperasikan sistem, dibutuhkan pengguna sebagai berikut:

1. Memiliki kemampuan untuk mengoperasikan komputer dengan baik.
2. Memiliki kemampuan untuk mengoperasikan *browser*.
3. Memiliki akun Twitter.
4. Memiliki akun Google.

E. Perancangan Sistem



GAMBAR 4
DIAGRAM ALIR SISTEM PENDETEKSI *THREAT* DAN *VULNERABILITY*

Dari *flowchart* di atas dijelaskan bahwa sistem dimulai dari pengambilan data dari situs web Twitter dengan menggunakan teknik *web scraping*. Data yang didapat dari *web scraping* selanjutnya dilakukan *text pre-processing*.

Sebelum dilakukan proses *lemmatizing*, dilakukan pelabelan pada setiap kata dengan teknik *POS Tagging*. Proses *POS Tagging* bertujuan untuk membantu proses *lemmatizing* agar proses pengembalian kata ke kata dasarnya bisa lebih akurat.

Langkah selanjutnya yaitu pembobotan data dengan menggunakan algortima TF-IDF. Proses ini dilakukan dengan cara menghitung frekuensi kemunculan kata dalam data Dalam penelitian ini, dilakukan sedikit modifikasi dalam proses pembobotan TF-IDF. Proses TF-IDF dikombinasikan dengan teknik pelabelan kata *POS Tagging*. Berdasarkan penelitian[29], proses kombinasi ini dilakukan untuk menekankan kelas kata yang lebih penting.

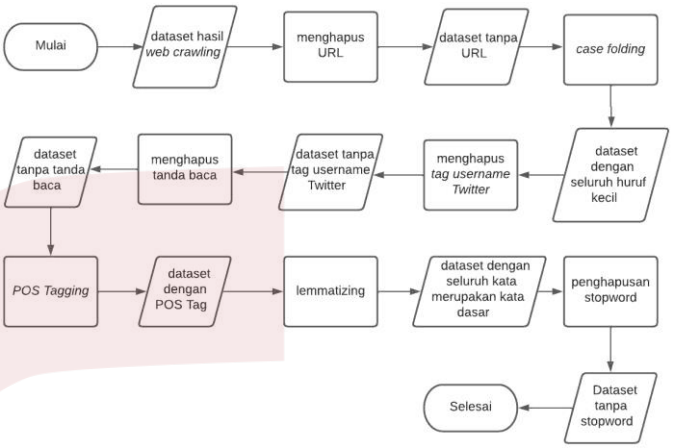
Berikutnya proses klasifikasi dilakukan dengan menggunakan algoritma Naïve Bayes untuk mengklasifikasi data termasuk kedalam kelas *threat* atau *vulnerability*, atau tidak keduanya. Dalam proses kerja algoritma Naïve Bayes, terjadi dua tahap pemrosesan, yaitu proses pelatihan (*training*) dan pengujian (*testing*). Pada tahap *training* dilakukan analisis dan pelatihan menggunakan *dataset* yang ada untuk membangun model. Kemudian akurasi model yang telah dibangun diuji pada tahap *testing*[30].

Sebagai pengukur performa algoritma kalsifikasi Naïve Bayes, digunakan pengukuran dengan menggunakan *Confusion Matrix*. Pada *Confusion Matrix*, nilai kalkulasi yang didapat akan diproses untuk mendapatkan nilai *accuracy, precision, dan recall*.

Selain dengan menggunakna *Confusion Matrix*, nilai akurasi juga dikalkulasi dengan menggunakan algoritma *K-Fold Cross Validation*. Hal ini dilakukan karena proses *K-Fold Cross Validation* membagi *dataset* menjadi data *testing* dan data training dengan secara lebih adil, dimana proses kalkulasi akurasi dilakukan berulang kali hingga seluruh *dataset* mendapat giliran untuk menjadi data *testing* dan data *training*.

F. *Text Pre-Processing*

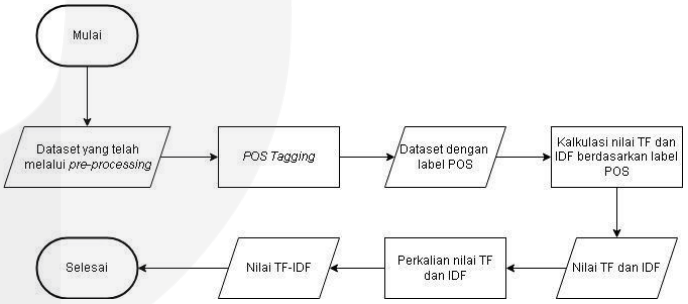
Text pre-processing merupakan tahap pemrosesan data tekstual dengan tujuan agar data text menjadi se-simpel mungkin agar dapat dengan mudah dipahami oleh model *machine learning*. *Proses pre-prosesing* pada penelitian ini akan dijabarkan pada diagram alur dibawah ini.



GAMBAR 5
DIAGRAM ALUR *TEXT PRE-PROCESSING*.

G. Pembobotan

Pada tahap pembobotan, setiap kata atau fitur diberikan diberikan nilai, yang merepresentasikan bobot fitur tersebut. Dalam penelitian ini, proses pembobotan terdiri dari dua tahap, yaitu *POS Tagging* dan TF-IDF. Pada proses *POS Tagging* setiap kata pada dataset diberikan label berdasarkan kelas kata, seperti yang ditunjukkan pada tabel 2. Kemudian proses kalkulasi TF-IDF disesuaikan berdasarkan label yang tertera pada kata.



GAMBAR 6
DIAGRAM ALUR PROSES PEMBOBOTAN

Pada gambar 6, dijelaskan alur proses pembobotan dengan menggunakan kombinasi POS Tagging dan TF-IDF. Sebagai contoh, berikut adalah tweet yang telah melalui tahap *text pre-processing*:

TABEL 5
CONTOH *TWEET* YANG TELAH MELALUI TAHAP *PRE-PROCESSING*

Dokumen	*Tweet*	Label
D1	apache log4j remote code execution rce	1
D2	data security encryption vulnerability unveiled	0

Selanjutnya, dilakukan *POS Tagging* pada data, sehingga data menjadi sebagai berikut:

TABEL 6
CONTOH TWEET YANG TELAH DIBERI LABEL POS

Dokumen	*Tweet*	Label
D1	[('apache', 'NN'), ('log4j', 'JJ'), ('remote', 'JJ'), ('code', 'NN'), ('execution', 'NN'), ('rce','NN')]	1
D2	[('data', 'NNS'), ('security', 'NN'), ('encryption', 'NN'), ('vulnerability', 'NN'), ('unveil', 'NN')]	0

Kemudian tahap selanjutnya yaitu perhitungan nilai TF-IDF berdasarkan label POS dari setiap kata. Sehingga memberikan hasil sebagai berikut:

TABEL 7
NILAI TF-IDF YANG DIHITUNG BERDASARKAN POS TAG

Kata	D2	D1
apache	0	0,2508
code	0	0,2508
data	0,602	0
encryption	0,301	0
execution	0	0,2508
log4j	0	0,1505
rce	0	0,2508
remote	0	0,1505
security	0,301	0
unveil	0,301	0
vulnerability	0,301	0

H. Klasifikasi Naïve Bayes

Selanjutnya, seluruh nilai yang didapat dari proses pembobotan TF-IDF akan dilanjutkan menuju proses klasifikasi. Pada penelitian ini, algoritma yang digunakan yaitu algoritma klasifikasi Naïve Bayes.

Sebagai contoh, diasumsikan akan dimasukkan data uji yaitu "security news today # rce #log4j". selanjutnya dengan naïve bayes akan dihitung probabilitas label dari data uji, dengan Langkah-langkah perhitungan sebagai berikut:

1. *Prior*

   $P(a) = Na/N$

   Diketahui N merupakan jumlah seluruh dokumen dan Na adalah jumlah seluruh dokumen dengan label a. Sehingga jika diimplementasikan kedalam contoh kasus, menghasilkan nilai:

   $P(p) = 1/2$

   $P(n) = 1/2$

   Keterangan:

   p = label positif

   n = label negatif

2. *Conditional probabilities*

   $P(w \mid a)=(count(w,a)+1)/(count(a)+|V|)$

   Dari rumus tersebut dapat diketahui bahwa count(w,a) adalah jumlah kata w dalam data berlabel a. Sedangkan |V|, adalah jumlah kata unique dari seluruh dokumen. Maka jika diimplementasikan kedalam contoh kasus, menghasilkan nilai:

   - $P(security|p) = (0+1) / (6+11) = 1/17$
   - $P(news|p) = (0+1) / (6+11) = 1/17$
   - $P(today|p) = (0+1) / (6+11) = 1/17$
   - $P(rce|p) = (1+1) / (6+11) = 2/17$
   - $P(log4j|p) = (1+1) / (6+11) = 2/17$
   - $P(security|n) = (1+1) / (5+11) = 1/8$
   - $P(news|n) = (0+1) / (5+11) = 1/16$
   - $P(today|n) = (0+1) / (5+11) = 1/16$
   - $P(rce|n) = (0+1) / (5+11) = 1/16$
   - $P(log4j|n) = (0+1) / (5+11) = 1/16$

3. Pemilihan Kelas

   Pada tahap pemilihan kelas, nilai prior dikalikan dengan nilai conditional probabilities dari seluruh kata. Sehingga kalkulasi contoh kasus menjadi:

   $P(p|D3) = P(p) \times P(security|p) \times P(news|p) \times P(today|p) \times P(rce|p) \times P(log4j|p)$

   $P(p|D3) = 1/2 \times 1/17 \times 1/17 \times 1/17 \times 2/17 \times 2/17 = 0{,}00000140895$

   $P(n|D3) = P(n) \times P(security|n) \times P(news|n) \times P(today|n) \times P(rce|n) \times P(log4j|n)$

   $P(n|D3) = 1/2 \times 1/8 \times 1/16 \times 1/16 \times 1/16 \times 1/16 = 0{,}0000000596$

   Dari kalkulasi dengan Naïve Bayes yang telah dilakukan, didapatkan kesimpulan bahwa data uji mendapat nilai lebih tinggi pada kelas positif, sehingga label data uji dinyatakan positif.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Data

Dalam proses pengujian, jumlah data tersebut akan dibagi menjadi data *training* dan data *testing*. Dalam pembagian data menjadi data *training* dan data *testing*, data akan dibagi menjadi beberapa rasio pembagian yang berbeda, yaitu 50% data *training* dan 50% data *testing*, 60% data *training* dan 40% data *testing*, 70% data *training* dan 30% data *testing*, 80% data *training* dan 20% data *testing*, dan yang terakhir 90% data *training* dan 10% data *testing*, dengan menggunakan random_state=0.

Output yang dihasilkan berupa label positif dan negatif yang merepresentasikan kelas data. Label positif bagi data yang terdeteksi sebagai tweet dengan unsur threat atau *vulnerability*, dan label negatif sebagai tweet yang tidak terdeteksi sebagai *threat* atau *vulnerability*.

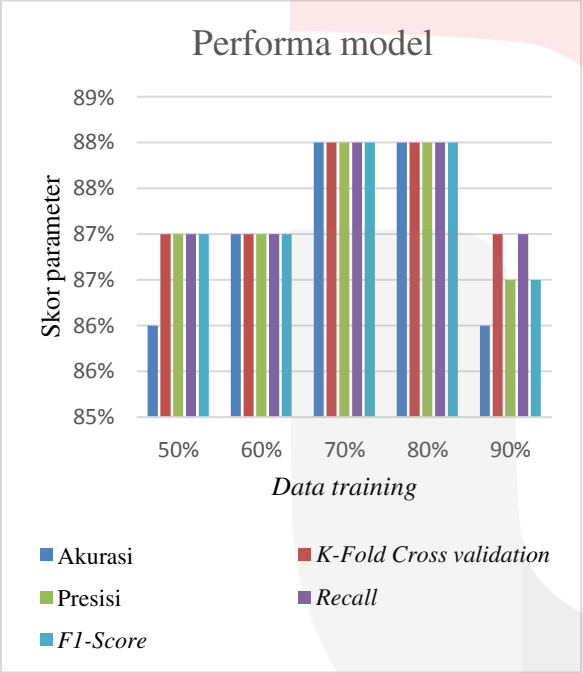
B. Pengujian Partisi Data

Pengujian partisi data dilakukan dengan menjalankan semua perbandingan rasio pembagian data menjadi data *training* dan data *testing*. Data pengujian oleh sistem akan ditampilkan dalam tabel *confusion matrix* dibawah ini:

TABEL 8
CONFUSION MATRIX PARTISI DATA.

Data latih			Label aktual	
			Positif	Negatif
50%	Label prediksi	Positif	953	133
		Negatif	147	902
60%		Positif	749	109
		Negatif	111	739
70%		Positif	553	78
		Negatif	79	571
80%		Positif	371	46
		Negatif	57	380
90%		Positif	181	25
		Negatif	32	189

C. Pengujian Performa



GAMBAR 7
HASIL UJI PERFORMA MODEL

Dalam pengujian kinerja, terdapat beberapa parameter yang digunakan untuk mengukur kinerja model. Parameter yang digunakan adalah akurasi, akurasi *K-Fold Cross Validation*, presisi, *recall*, dan *F1-Score*.

Dari grafik diatas dapat diketahui bahwa rata-rata performansi terbaik terdapat pada data training prosentase 70% dan 80% Dimana keduanya menghasilkan nilai performansi 88%.

D. Skenario Pengujian Validasi

Skenario pengujian validasi dilakukan dengan cara mengunduh data-data baru dengan berbagai kata kunci. Data yang telah diunduh kemudian diberi label secara manual, untuk melihat akurasi sistem. Berikut adalah beberapa hasil pengujian output dari berbagai data baru dengan kata kunci yang berbeda:

1. Hasil kalkulasi dari 24 data yang diunduh dengan kata kunci ‘cve’.

```
Accuracy of the classifier is 0.9583333333333334

Confusion matrix is:
 [[ 7  0]
 [ 1 16]]

classification report is:
              precision    recall  f1-score   support

           0       0.88      1.00      0.93         7
           1       1.00      0.94      0.97        17

    accuracy                           0.96        24
   macro avg       0.94      0.97      0.95        24
weighted avg       0.96      0.96      0.96        24


The value of Precision 1.0

The value of Recall 0.9411764705882353
```

GAMBAR 8
HASIL KALKULASI DARI 24 DATA YANG DIUNDUH DENGAN KATA KUNCI ‘CVE’.

2. Hasil kalkulasi dari 25 data yang diunduh dengan kata kunci ‘*vulnerability’*.

```
Accuracy of the classifier is 0.88

Confusion matrix is:
 [[10  2]
 [ 1 12]]

classification report is:
              precision    recall  f1-score   support

           0       0.91      0.83      0.87        12
           1       0.86      0.92      0.89        13

    accuracy                           0.88        25
   macro avg       0.88      0.88      0.88        25
weighted avg       0.88      0.88      0.88        25


The value of Precision 0.8571428571428571

The value of Recall 0.9230769230769231
```

GAMBAR 9
HASIL KALKULASI DARI 25 DATA YANG DIUNDUH DENGAN KATA KUNCI ‘*VULNERABILITY’*.

V. KESIMPULAN

Berdasarkan sistem yang telah berhasil dibuat, dari hasil pengujian serta analisis, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem dengan algoritma klasifikasi Naïve Bayes berhasil mendeteksi *threat* dan *vulnerability* pada unggahan Twitter, dengan memberi label positif dan negatif pada dataset.
2. Akselerasi terbaik pada sistem pendeteksi *threat* dan *vulnerability* pada unggahan Twitter adalah pada rasio pembagian data latih dan data uji 70%:30%, dan 80%:20%, keduanya mendapatkan nilai akurasi sebesar 88%, nilai presisi sebesar 88%, *F1 Score* sebesar 88%, dan nilai *recall* sebesar 88%.

REFERENSI

[1] L. A. McFarland and R. E. Ployhart, “Social media: A contextual framework to guide research and practice,” J. Appl. Psychol., vol. 100, no. 6, pp. 1653–1677, 2015, doi: 10.1037/a0039244.

[2] W. He, “A review of social media security risks and

mitigation techniques," J. Syst. Inf. Technol., vol. 14, no. 2, pp. 171–180, 2012, doi: 10.1108/13287261211232180.

[3] D. Sgandurra and E. Lupu, "Evolution of attacks, threat models, and solutions for virtualized systems," ACM Comput. Surv., vol. 48, no. 3, pp. 1–38, 2016, doi: 10.1145/2856126.

[4] D. A. Muthia, "Komparasi Algoritma Klasifikasi Text Mining Untuk Analisis Sentimen Pada Review Restoran," J. PILAR Nusa Mandiri, vol. 14, no. 1, pp. 69–74, 2018.

[5] M. S. Saputri, R. Mahendra, and M. Adriani, "Emotion Classification on Indonesian Twitter Dataset," Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018, pp. 90–95, 2019, doi: 10.1109/IALP.2018.8629262.

[6] X. Li, Q. Xie, J. Jiang, Y. Zhou, and L. Huang, "Identifying and monitoring the development trends of emerging technologies using patent analysis and Twitter data mining: The case of perovskite solar cell technology," Technol. Forecast. Soc. Change, vol. 146, no. May, pp. 687–705, 2019, doi: 10.1016/j.techfore.2018.06.004.

[7] M. Humayun, M. Niazi, N. Jhanjhi, M. Alshayeb, and S. Mahmood, "Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study," Arab. J. Sci. Eng., vol. 45, no. 4, pp. 3171–3189, 2020, doi: 10.1007/s13369-019-04319-2.

[8] W. A. Al-Khater, S. Al-Maadeed, A. A. Ahmed, A. S. Sadiq, and M. K. Khan, "Comprehensive review of cybercrime detection techniques," IEEE Access, vol. 8, pp. 137293–137311, 2020, doi: 10.1109/ACCESS.2020.3011259.

[9] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine Learning for industrial applications: A comprehensive literature review," Expert Syst. Appl., vol. 175, no. March, p. 114820, 2021, doi: 10.1016/j.eswa.2021.114820.

[10] M. Batta, "Machine Learning Algorithms - A Review," Int. J. Sci. Res. (IJ, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/ART20203995.

[11] G. Adomavicius and A. Tuzhilin, "Web Scraping:State of the art," IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, 2019.

[12] A. V Saurkar and S. A. Gode, "An Overview On Web Scraping Techniques And Tools," Int. J. Futur. Revolut. Comput. Sci. Commun. Eng., vol. 4, no. 4, pp. 363–367, 2018, [Online]. Available: http://www.ijfrcsce.org/index.php/ijfrcsce/article/view/1529.

[13] M. A. Khder, "Web scraping or web crawling: State of art, techniques, approaches and application," Int. J. Adv. Soft Comput. its Appl., vol. 13, no. 3, pp. 144–168, 2021, doi: 10.15849/ijasca.211128.11.

[14] S. Ananiadou, D. B. Kell, and J. ichi Tsujii, "Text mining and its potential applications in systems biology," Trends Biotechnol., vol. 24, no. 12, pp. 571–579, 2006, doi: 10.1016/j.tibtech.2006.10.002.

[15] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi, "Text mining in big data analytics," Big Data Cogn. Comput., vol. 4, no. 1, pp. 1–34, 2020, doi: 10.3390/bdcc4010001.

[16] K. L.Sumathy and M. Chidambaram, "Text Mining: Concepts, Applications, Tools and Issues An Overview," Int. J. Comput. Appl., vol. 80, no. 4, pp. 29–32, 2013, doi: 10.5120/13851-1685.

[17] M. Pota, F. Marulli, M. Esposito, G. De Pietro, and H. Fujita, "Multilingual POS tagging by a composite deep architecture based on character-level features and on-the-fly enriched Word Embeddings," Knowledge-Based Syst., vol. 164, no. xxxx, pp. 309–323, 2019, doi: 10.1016/j.knosys.2018.11.003.

[18] 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," Proc. - 2018 4th Int. Conf. Inf. Retr. Knowl. Manag. Diving into Data Sci. CAMP 2018, pp. 107–112, 2018, doi: 10.1109/INFRKM.2018.8464692.

[19] IBM Corporation, "Part-of-speech tag sets." https://www.ibm.com/docs/en/wca/3.5.0?topic=analytics-part-speech-tag-sets.

[20] M. Nurjannah and I. Fitri Astuti, "PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman," J. Inform. Mulawarman, vol. 8, no. 3, pp. 110–113, 2013.

[21] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," Int. J. Comput. Appl., vol. 181, no. 1, pp. 25–29, 2018, doi: 10.5120/ijca2018917395.

[22] G. I. Webb, "Naïve Bayes," in Encyclopedia of Machine Learning and Data Mining, Springer US, 2016, pp. 1–2.

[23] Bustami, "Penerapan Algoritma Naive Bayes," J. Inform., vol. 8, no. 1, pp. 884–898, 2014.

[24] 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," Evol. Syst., vol. 12, no. 3, pp. 755–767, 2021, doi: 10.1007/s12530-019-09283-8.

[25] I. K. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation," Int. J. Inf. Technol. Comput. Sci., vol. 13, no. 6, pp. 61–71, 2021, doi: 10.5815/ijitcs.2021.06.05.

[26] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, "Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking," IEEE Access, vol. 8, pp. 90847–90861, 2020, doi: 10.1109/ACCESS.2020.2994222.

[27] F. Rahmad, Y. Suryanto, and K. Ramli, "Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification," IOP Conf. Ser. Mater. Sci. Eng., vol. 879, no. 1, 2020, doi: 10.1088/1757-899X/879/1/012076.

[28] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis

Sentimen Berbasis Teks Pada Twitter," J. Sains Komput. Inform., vol. 5, no. 2, pp. 697–711, 2021.

[29] R. Xu, "POS weighted TF-IDF algorithm and its application for an MOOC search engine," ICALIP 2014 - 2014 Int. Conf. Audio, Lang. Image Process. Proc., pp. 868–873, 2015, doi: 10.1109/ICALIP.2014.7009919.

[30] A. P. Wijaya and H. A. Santoso, "Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government Naïve Bayes Classification on Document Classification to Identify E-Government Content," J. Appl. Intell. Syst., vol. 1, no. 1, pp. 48–55, 2016.