Hate Speech Detection Using Support Vector Machine (SVM) Method [Deteksi Ujaran Kebencian Menggunakan Metode Support Vector Machine (SVM)]

Mohammad Attar Jibran[1)], Ade Eviyanti*[2)]
[1,2)]Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia
*Email Penulis Korespondensi: adeeviyanti@umsida.ac.id

Abstract. *Hate speech is a linguistic phenomenon that deviates from the norms and polite grammar in language and communication ethics. This research is aimed at detecting a word or sentence containing or not containing a hate speech using the SVM method for classification. This research takes data using the Tweepy API and gets a total sample data of 1681. To do word weighting, researchers use TF-IDF to find out the frequency of words that often arise in the dataset. In the classification process, researchers used two methods, namely SVM and XGBoost which then from the best results in SVM with 90% training data and 10% test data obtained a training score of 95.87% and a test score of 87.30% with a gap of 8.57% then from the SVM method was tuned using RSCV and managed to increase the training score by 100% test score of 93.20% with a gap of 6.80%.*

Keywords *– Prediction; Hate Speech; SVM; XGBoost; RSCV*

Abstrak. *Ujaran kebencian merupakan fenomena kebahasaan yang menyimpang dari norma dan tata bahasa yang santun dalam etika berbahasa dan berkomunikasi. Penelitian ini bertujuan untuk mendeteksi sebuah kata atau kalimat yang mengandung atau tidak mengandung ujaran kebencian dengan menggunakan metode SVM untuk klasifikasi. Penelitian ini mengambil data dengan menggunakan Tweepy API dan mendapatkan total sampel data sebanyak 1681. Untuk melakukan pembobotan kata, peneliti menggunakan TF-IDF untuk mengetahui frekuensi kata yang sering muncul dalam dataset. Pada proses klasifikasi, peneliti menggunakan dua metode yaitu SVM dan XGBoost yang kemudian dari hasil terbaik pada SVM dengan data latih 90% dan data uji 10% didapatkan nilai training score 95.87% dan nilai test score 87.30% dengan gap 8.57% kemudian dari metode SVM tersebut di tuning menggunakan RSCV dan berhasil meningkatkan nilai training score 100% test score 93.20% dengan gap 6.80%.*

Kata Kunci *– Prediksi; Ujaran Kebencian; SVM; XGBoost; RSCV*

I. PENDAHULUAN

Ujaran kebencian merupakan suatu fenomena kebahasaan yang menyimpang dari norma tata bahasa yang santun dalam etika berbahasa dan berkomunikasi.[1] Dengan menggunakan ujaran kebencian seseorang dapat melampiaskan hasratnya untuk berkata tidak senonoh dan menjatuhkan martabat orang lain.

Dewasa ini ujaran kebencian semakin marak beredar di internet, khususnya pengguna media sosial. Berawal dari kekecewaan sebuah individu oleh keadaan yang tidak sesuai dengan keinginannya serta dengan mudah dan bebasnya pengguna media sosial dalam mengakses dan mengekspresikan dirinya pada media sosial sehingga kata atau kalimat ujaran kebencian semakin bebas mengudara dan semakin tidak terkendali.

Langkah pemerintah dalam mengatur tindak pidana bagi pelantun ujaran kebencian diatur dalam Pasal 156, Pasal 157, Pasal 310, Pasal 311 KUHP, Pasal 45 ayat (2) UU No 19 Tahun 2016 tentang perubahan atas UU No 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik.[2]

Sebuah kata atau kalimat yang terdeteksi ujaran kebencian akan terdeteksi sebagai sebuah ujaran kebencian dan kata atau kalimat yang tidak mengandung ujaran kebencian akan terdeteksi tidak mengandung ujaran kebencian yang kemudian dalam implementasi penelitian ini nantinya akan dapat membantu penegak hukum menyelesaikan kasus di sebuah tindak pidana yang berhubungan dengan ujaran kebencian.

Bahasa Alami atau *Natural Language Processing* merupakan sebuah teknologi *Machine Learning* dalam wilayah kecerdasan buatan yang mampu mengenali kata, klasifikasi objek kata dan pengenalan pola kata.[3]

Pembobotan kata yang digunakan dalam penelitian ini yaitu *Term Frequency-Inverse Document Frequency* (TF-IDF) yang digunakan untuk mengetahui frekuensi kata yang sering timbul pada *dataset*.

Metode yang digunakan dalam deteksi ujaran kebencian ini adalah SVM dimana Metode *Support Vector Machine* (SVM) merupakan salah satu metode didalam *Supervised Learning* untuk mengklasifikasikan *dataset* yang telah disiapkan sebelumnya. Penelitian ini juga menggunakan metode XGBoost[4] dan SVM *with Randomized Search Cross Validation* (RSCV)[5] sebagai metode bandingan dan untuk lebih mengoptimalkan akurasi sistem.

Berdasarkan latar belakang diatas penulis tertarik untuk mengkaji dan membuat sebuah aplikasi deteksi ujaran kebencian dengan judul "Deteksi Ujaran Kebencian Menggunakan Metode *Support Vector Machine* (SVM)" yang berguna untuk mendeteksi sebuah ujaran kebencian.

II. METODE

A. Tinjauan Pustaka

Penelitian terdahulu yang pertama dari (Wahyuningrum, Rijal Abdulhakim, Yuyun Umaidah, Jajam Haerul Jaman 2021), Mahasiswa program studi Teknik Informatika Universitas Singaperbangsa Karawang. Mahasiswa dari program studi Teknik Informatika ini membuat Jurnal untuk syarat kelulusan S1 dengan judul Optimasi *Support Vector Machine* Berbasis *Particle Swarm Optimization* Untuk Mendeteksi *Hate Speech* Pilkada Karawang, data yang diambil dari jurnal tersebut berasal dari grup facebook Karawang Info dengan kata kunci "Bupati", "Pilkada", dan "Pilkada Karawang" untuk mendeteksi *hate speech* anggota grup dengan menggunakan metode SVM.[6]

Penelitian terdahulu yang kedua dari (Ananda Adhari, Muhammad Nasrun S.Si, M.T., Ratna Astuti Nugrahaeni S.T, M.T. 2021), Mahasiswa program studi S1 Teknik Komputer Universitas Telkom. Mahasiswa dari program studi Teknik Komputer ini membuat Jurnal untuk syarat kelulusan S1 dengan judul DETEKSI UJARAN ANCAMAN BERBASIS WEBSITE PADA MEDIA SOSIAL TWITTER MENGGUNAKAN METODE SUPPORT VECTOR MACHINE, dataset yang diambil dari jurnal tersebut berasal dari twitter untuk kemudian diuji menggunakan metode *Support Vector Machine.*[7]

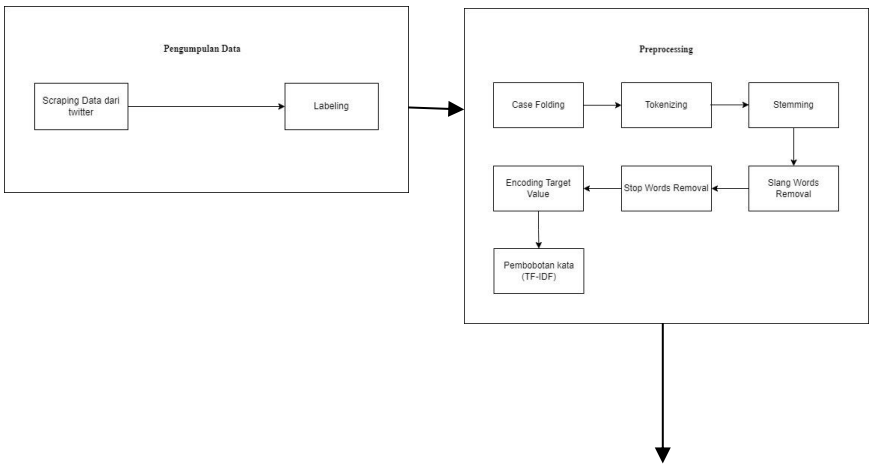
Penelitian terdahulu yang ketiga dari (Dayang Putri Nur Lyrawati 2019), Mahasiswa program studi matematika Universitas Negeri Surabaya. Mahasiswa dari program studi matematika ini membuat Jurnal untuk syarat kelulusan S1 dengan judul DETEKSI UJARAN KEBENCIAN PADA TWITTER MENJELANG PILPRES 2019 DENGAN MACHINE LEARNING, data yang diambil dari jurnal tersebut diambil dariTweepy API dengan kata kunci "#pilpres2019", "#2019gantipresiden", "#debatcapres", "debatpilpres" dan "#jokowi2periode" dari dataset tersebut dilakukan labeling, *preprocessing*, kemudian dilakukan klasifikasi menggunakan metode SVM.[8]

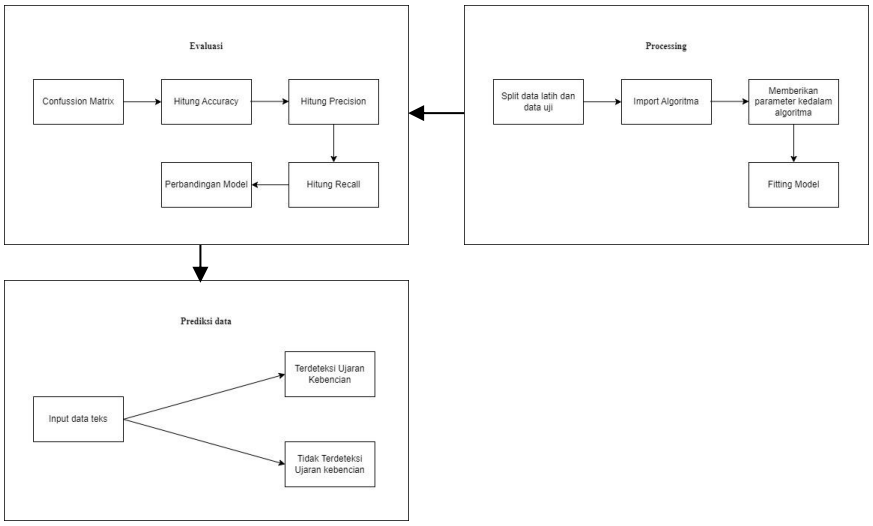
Penelitian terdahulu yang keempat dari (Naufal Azmi Verdikha, Teguh Bharata Adji, Adhistya Erna Permanasari 2018), Mahasiswa dari program studi teknologi informasi Universitas Gadjah Mada. Mahasiswa dari program studi teknologi informasi ini membuat Jurnal untuk syarat kelulusan S1 dengan judul KOMPARASI METODE OVERSAMPLING UNTUK KLASIFIKASI TEKS UJARAN KEBENCIAN, dataset yang digunakan adalah Dataset Tweet Hatepeech yang berasal dari penelitian Waseem, dkk diambil dari salah satu *crawler* tweet Pyton : Tweepy. Dari dataset tersebut dilakukan pra proses, pembobotan, *oversampling*, kemudian diklasifikasikan menggunakan metode SVM.[9]

Penelitian terdahulu yang kelima dari (Luh Putu Ary Sri Tjahyanti 2020), staf edukatif pada FKIP Universitas Panji Sakti Singaraja ini membuat Jurnal dengan judul PENDETEKSIAN BAHASA KASAR (ABUSIVE LANGUAGE) DAN UJARAN KEBENCIAN (HATE SPEECH) DARI KOMENTAR DI JEJARING SOSIAL, dataset yang diambil dari tweet yang ada di twitter menggunakan Twitter Api dan Tweepy Library yang kemudian dilakukan labeling dan berikutnya dilakukan proses klasifikasi dengan metode *Naive Bayes* (NB), *Support Vector Machine* (SVM) dan *Random Forest Decision Tree* (RFDT).[10]

Penelitian ini adalah sebuah penyempurnaan sekaligus pembanding dari penelitian sebelumnya dimana sistem yang dibuat bukan hanya mengoptimalkan *dataset*, melainkan terdapat fitur untuk memprediksi kata atau kalimat yang nantinya akan diketahui buah kata atau kalimat tersebut mengandung ujaran kebencian atau tidak.

B. Desain Umum Sistem





Gambar 1. Desain Umum Sistem

- **Tahap Pengumpulan Data**

Untuk mendapatkan informasi dan data dalam aplikasi deteksi ujaran kebencian ini dibutuhkan pengumpulan data yang diambil dari twitter menggunakan Tweepy API yang kemudian dijadikan satu berkas dalam format .csv.

- **Tahap Preprocessing**

Tahap berikutnya adalah *Preprocessing* untuk mengolah *text* dari dataset, di tahap ini dilakukan *Case Folding*, *Tokenizing*, *Stemming*, *Slang Words Removal*, *Stop Words Removal*, *Encoding Target Value,* Pembobotan Kata menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF).

- **Case Folding**

Pada tahapan awal preprocessing yaitu dilakukan *Case Folding* yang mana berfungsi untuk menyamaratakan kata menjadi huruf kecil (*lowercase*) semua, dan dapat menghapus kata yang tidak diperlukan seperti RT, @, 8, http dan lain sebagainya.

- **Tokenizing**

Selanjutnya tahapan *Tokenizing* yang digunakan untuk memecah kalimat-kalimat menjadi kata. Dengan tokenizing dapat membedakan antara pemisah kata atau bukan, juga mencakup proses penghapusan nomor, simbol, tanda baca yanng tidak penting.

- **Stemming**

Stemming digunakan untuk menghilangkan kata imbuhan menjadi kata dasar. Selain itu juga dapat mengelompokkan kata-kata yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda dikarenakan pemakaian imbuhan yang berbeda.

- **Slang Words Removal**

Slang Words merupakan kata atau istilah gaul yang sudah menjadi budaya atau kebiasaan dalam percakapan sehari-hari. Oleh karena itu diperlukan adanya *Slang Words Removal* untuk menyempurnakan kata singkatan menjadi kata yang harfiah.

- **Stop Words Removal**

Stop Words Removal digunakan untuk memilih kata-kata penting dan membuang kata yang kurang begitu penting. Harapan dari penghapusan kata tidak penting ini adalah untuk meningkatkan performa model.

- **Encoding Target Value**

Encoding Target Value digunakan untuk mengubah data kategoris yang ada dalam target menjadi data nominal. Ini dikarenakan mesin hanya bisa membaca data angka.

- **Pembobotan Kata menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF)**

Term weighting adalah pembobotan tiap kata syang dapat menaikkan analisa sentimen dalam proses *text mining*. *Term Frequency - Inverse Document Frequency* (TF-IDF) merupakan metode yang seringkali digunakan untuk pembobotan kata. Untuk mencari nilai dari kata yang muncul dalam

suatu dokumen digunakan TF. kemudian untuk mencari nilai dari kata yang muncul dalam keseluruhan dokumen digunakan IDF, nilai TF berbanding terbalik dengan IDF, semakin banyak kata yang muncul maka nilai IDF akan semakin kecil. Perhitungan bobot kata dari sebuah dokumen dalam TF-IDF dilakukan dengan menghitung setiap nilai dari masing-masing TF dan IDF.[11]

- **Tahap Processing**

Pada tahap *Processing* ini digunakan 3 metode yang berbeda untuk klasifikasi data, yakni menggunakan metode *Support Vector Machine* (SVM), XGBOOST, dan *Hyperparameter* SVM menggunakan *Randomized Search Cross Validation*. Pada penggunaan SVM dan XGBoost data dibagi menjadi 2 bagian yaitu data latih dan data uji dengan persentase sebagai berikut :

Tabel 1. Data Latih dan Data Uji

Data Latih	Data Uji
90%	10%
80%	20%
75%	25%

- **Tahap Evaluasi**

Setelah dilakukan tahapan klasifikasi, selanjutnya dilakukan tahapan evaluasi dimana tahapan ini digunakan untuk mengukur hasil dari *Machine Learning* yang telah dibuat. Matrik evaluasi yang dibuat adalah *Confusion Matrix*. Dengan adanya perhitungan dari *Confusion Matrix* dapat diperoleh hasil *accuracy*, *sensitivity* dan *specificity*.[12]

- **Tahap Prediksi Data**

Tahapan terakhir ini adalah Tahap Prediksi Data dimana tahapan ini adalah inti dari penelitian, berupa prediksi teks atau kata yang di masukkan dan kemudian data masukan akan dideteksi sebagai sebuah ujaran kebencian atau tidak.

III. HASIL DAN PEMBAHASAN

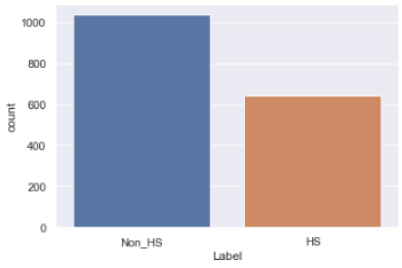
A. Pengumpulan Data

Penelitian ini menggunakan dataset yang diambil dari cuitan warganet yang ada di twitter menggunakan *Tweepy* Api dalam pengumpulan data dan kemudian dari data tersebut dilakukan proses labeling untuk mengkurasi *tweet* (sebutan cuitan warganet twitter) berupa HS untuk data yang mengandung ujaran kebencian dan Non_HS untuk data yang tidak mengandung ujaran kebencian.

Tabel 2. Data Label

	Label	0
0	HS	643
1	Non_HS	1038

Dari tabel diatas data didominasi oleh Non HS. Jumlah data HS adalah 643 dan data Non_HS berjumlah 1038 jumlah keseluruhan data total HS dan Non_HS adalah 1681.



Gambar 2. Perbandingan Jumlah Data Label

B. Hasil Preprocessing

Setelah tahapan pengumpulan data, dimulailah tahapan *preprocessing* berupa *Case Folding*, *Tokenizing*, *Stemming*, *Slang Words Removal*, *Stop Words Removal*, *Encoding Target Value* dan TF-IDF didapatkan hasil seperti berikut :

	Label	Tweet\t	length	Case_folded	Tokenized	Stemmed	No_Slang	No_Stop	Ready
0	0	RT @spardaxyz: Fadli Zon Minta Mendagri Segera...	111	fadli zon minta mendagri segera menonaktifkan ...	[fadli, zon, minta, mendagri, segera, menonakt...	[fadli, zon, minta, mendagri, segera, nonaktif...	[fadli, zon, minta, mendagri, segera, nonaktif...	[fadli, zon, mendagri, nonaktif, ahok, gubernu...	fadli zon mendagri nonaktif ahok gubernur dki
1	0	RT @baguscondromowo: Mereka terus melukai aksi...	109	mereka terus melukai aksi dalam rangka memenja...	[mereka, terus, melukai, aksi, dalam, rangka, ...	[mereka, terus, luka, aksi, dalam, rangka, pen...	[mereka, terus, luka, aksi, dalam, rangka, pen...	[luka, aksi, rangka, penjara, ahok, ahok, gaga...	luka aksi rangka penjara ahok ahok gagal pemil...
2	0	Sylvi: bagaimana gurbernur melakukan kekerasan...	117	sylvi bagaimana gurbernur melakukan kekerasan ...	[sylvi, bagaimana, gurbernur, melakukan, keker...	[sylvi, bagaimana, gurbernur, laku, keras, per...	[sylvi, bagaimana, gurbernur, laku, keras, per...	[sylvi, gurbernur, laku, keras, perempuan, buk...	sylvi gurbernur laku keras perempuan bukti fot...
3	0	Ahmad Dhani Tak Puas Debat Pilkada, Masalah Ja...	116	ahmad dhani tak puas debat pilkada masalah jal...	[ahmad, dhani, tak, puas, debat, pilkada, masa...	[ahmad, dhani, tidak, puas, debat, pemilihan k...	[ahmad, dhani, tidak, puas, debat, pemilihan k...	[ahmad, dhani, puas, debat, pemilihan kepala d...	ahmad dhani puas debat pemilihan kepala daerah...
4	0	RT @lisdaulay28: Waspada KTP palsu.....kawal P...	80	waspada ktp palsu kawal pilkada	[waspada, ktp, palsu, kawal, pilkada]	[waspada, ktp, palsu, kawal, pemilihan kepala ...	[waspada, ktp, palsu, kawal, pemilihan kepala ...	[waspada, ktp, palsu, kawal, pemilihan kepala ...	waspada ktp palsu kawal pemilihan kepala daerah

Gambar 3. Hasil Akhir *Preprocessing*

C. Hasil Processing

Pada tahapan ini dilakukan uji split data latih dan data uji. Data latih digunakan untuk melatih model sedangkan data uji digunakan untuk mengevaluasi model. Pada tahapan uji split data ini, peneliti menggunakan fungsi dari library scikit learn yaitu train_test_split. Berikut adalah kode dari split data latih dan data uji.

```
from sklearn.model_selection import train_test_split
X_train1, X_test1, y_train1, y_test1 = train_test_split(tfidf_vector, label, test_size=0.1, shuffle=True, random_state=42)
```

Gambar 4. Pembagian data latih 90% dan data uji 10%

```
from sklearn.model_selection import train_test_split
X_train2, X_test2, y_train2, y_test2 = train_test_split(tfidf_vector, label, test_size=0.2, shuffle=True, random_state=42)
```

Gambar 5. Pembagian data latih 80% dan data uji 20%

```
from sklearn.model_selection import train_test_split
X_train3, X_test3, y_train3, y_test3 = train_test_split(tfidf_vector, label, test_size=0.25, shuffle=True, random_state=42)
```

Gambar 6. Pembagian data latih 75% dan data uji 25%

Dari kode diatas didapatkan jumlah data latih dan data uji. Berikut adalah hasil dari pembagian data.

Tabel 4. Hasil Pembagian Data

Persentase Data Latih	Persentase Data Uji	Jumlah Data Latih		Jumlah Data Uji	
		X train	y train	X test	y test
90%	10%	1868	1868	208	208
80%	20%	1660	1660	416	416
75%	25%	1557	1557	519	519

D. Klasifikasi dengan Support Vector Machine (SVM)

Berdasarkan dari hasil data latih dan uji diatas. Dilakukan tahapan modeling dengan metode *Support Vector Machine* (SVM) dengan hasil sebagai berikut:

Tabel 5. Hasil Modeling Metode *Support Vector Machine* (SVM)

Data Latih	Data Uji	Metode	Estimator	Kernel	Skor Latih	Skor Uji
90%	10%				98,44%	89,90%
80%	20%	SVM	SVC	Linear	98,73%	89,90%
75%	25%				98,84%	89,59%

E. Klasifikasi dengan XGBoost

Sebagai model pembanding di penelitian ini digunakan model dengan metode XGBoost untuk membandingkan dari metode SVM yang digunakan dalam penelitian ini, dan menghasilkan data sebagai berikut:

Tabel 6. Hasil Modeling Metode XGBoost

Data Latih	Data Uji	Metode	Estimator	Skor Latih	Skor Uji
90%	10%			95,87%	87,30%
80%	20%	XGBOOST	XGBClasiffier	96,38%	87,01%
75%	25%			96,40%	87,09%

Dari hasil metode XGBoost diatas bahwasanya dalam penelitian ini penggunaan metode SVM lebih unggul daripada metode XGBoost. Dari hasil data diatas perolehan skor uji tertinggi diperoleh dalam pengujian dua metode SVM dengan data latih 90% dan 80% dan data uji 10% dan 20% dengan skor uji 89,90%. Tetapi terdapat perbedaan overfitting dari hasil tersebut, dari data latih 90% dan data uji 10% mendapatkan gap terendah yaitu 8,54%. Sehingga lebih efektif menggunakan metode SVM dengan data latih 90% dan data uji 10%.

F. Hyperparameter SVM dengan Randomized Search Cross Validation (RSCV)

Hyperparameter digunakan untuk meningkatkan kinerja dari metode yang efektif digunakan, dan *Randomized Search Cross Validation* adalah salah satu metode yang diaplikasikan dalam hyperparameter. dari data penelitian diatas penggunaan metode SVM lebih unggul dalam penelitian ini sehingga SVM dipilih untuk ditingkatkan kinerjanya dengan *Randomized Search Cross Validation* dan diperoleh hasil sebagai berikut:

Tabel 7. Hasil Modeling Metode SVM dengan RSCV

Data Latih	Data Uji	Model	Estimator	Kernel	Hyperparameter	Skor Latih	Skor Uji
90%	10%	SVM	SVC	Linear	RSCV	100,00%	93,20%

Hasil dari penelitian diatas ditunjukkan bahwa *Randomized Search Cross Validation* (RSCV) mendapatkan hasil yang sangat efektif yaitu mendapatkan skor latih 100% dan skor uji 93,20% dengan gap 6,80%.

G. Evaluasi

Setelah tahapan klasifikasi dengan metode yang dilakukan diatas, maka tahap selanjutnya adalah evaluasi untuk mengukur kerja dari model yang telah dibuat. Didalam penelitian ini peneliti menggunakan *Confussion Matrix* (CM). Terdapat empat kombinasi nilai prediksi dan nilai aktual dalam *Confusion Matrix*.

Tabel 8. *Confusion Matrix*

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

Empat istilah nilai tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN). Untuk memudahkan dalam memahami kegunaannya, peneliti membuat studi kasus untuk memberikan deteksi sebuah kata yang merupakan ujaran kebencian atau tidak.

- **True Positive (TP)**

Merupakan data positif yang diprediksi benar, contohnya terdapat kata yang mengandung ujaran kebencian dan dari model yang dibuat terdeteksi mengandung ujaran kebencian.

- **True Negative (TN)**

Merupakan data negatif yang diprediksi benar, contohnya terdapat kata yang tidak mengandung ujaran kebencian dan dari model yang dibuat tidak terdeteksi mengandung ujaran kebencian.

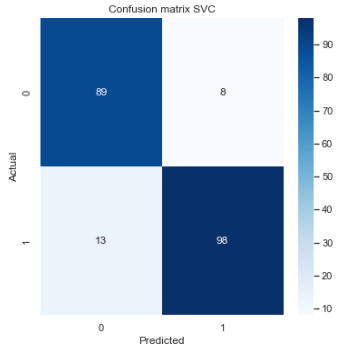
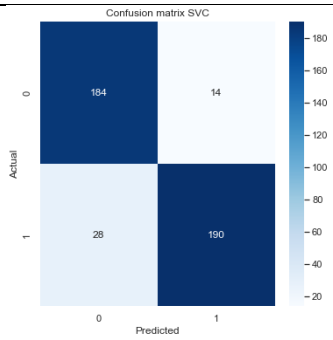
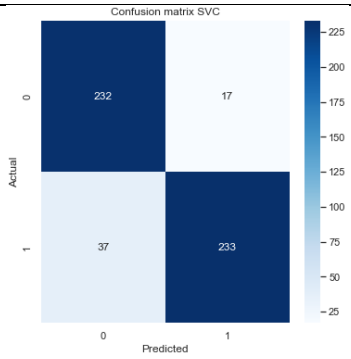
- **False Positive (FP)**

Merupakan data negatif akan tetapi diprediksi sebagai data positif, contohnya sebuah kata yang tidak mengandung ujaran kebencian, akan tetapi dari model yang dibuat terdeteksi mengandung ujaran kebencian.

- **False Negative (FN)**

Merupakan data positif akan tetapi diprediksi sebagai data negatif, contohnya sebuah kata yang mengandung ujaran kebencian, akan tetapi dari model yang dibuat terdeteksi tidak mengandung ujaran kebencian.

Tabel 9. Hasil *Confusion Matrix*

Data Latih	Data Uji	*Confusion Matrix*
90%	10%	
80%	20%	
75%	25%	

Dari hasil tabel confusion matrix diatas, kemudian dilakukan perhitungan nilai *accuracy*, *precision*, dan *recall*. Nilai *accuracy* adalah metode perhitungan berdasarkan kedekatan antara nilai prediksi dengan nilai aktual dengan mengetahui jumlah data yang diklasifikasikan secara benar. Berikut adalah rumus perhitungan *accuracy*:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} X\ 100\%$$

(1)

Precision adalah metode perhitungan dengan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi yang terambil dari sistem baik yang relevan ataupun tidak. Berikut adalah rumus perhitungan *precision*:

$$Precision = \frac{TP}{TP + FP}$$

(2)

Recall adalah metode perhitungan yang membandingkan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi relevan yang ada dalam dataset. Berikut adalah rumus perhitungan *recall*:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Berikut merupakan tabel hasil perhitungan *accuracy, precision* dan *recall*.

Tabel 10. Hasil *Accuracy, Precision dan Recall*

Data Latih	Data Uji	Accuracy	Precision	Recall
90%	10%	90%	87%	92%
80%	20%	90%	87%	93%
75%	25%	90%	86%	93%

Dari hasil tabel diatas menunjukkan bahwa nilai akurasi tertinggi mendapatkan hasil 90% dengan nilai presisi 87%, recall 93%, namun skor latih dan skor uji memiliki gap yang cukup tinggi yaitu 8,83%, sedangkan yang memiliki gap terendah adalah skor latih 98,44% dan skor uji 89,90%. Dari model tersebut didapatkan nilai akurasi 90% dengan presisi 87% dan recall 92%. Kemudian dari model SVM tersebut di tingkatkan dengan *hyperparameter Randomized Search Cross Validation* dan berhasil menaikkan skor latih sebesar 100% skor uji sebesar 93,20% dengan gap 6,80% mendapatkan nilai akurasi 93% dengan presisi 91% dan recall 95%.

H. Hasil Prediksi Data

Dari semua tahapan diatas kemudian dilakukan prediksi data yang didalamnya terdapat masukan untuk teks berupa kata atau kalimat yang kemudian diprediksi merupakan sebuah ujaran kebencian atau tidak merupakan sebuah ujaran kebencian dengan kode sumber sebagai berikut:

```
input_tweet = ['perbuatannya membuat orang lain menjadi percaya bahwa dia adalah orang baik',
        'toni bajingan']
def preProcessText(tweer):
  new_tweets = []

  for tw in texts:
    tw = case_folding(tw)
    tw = tokenized(tw)
    tw = stemming(tw)
    tw = removeSlang(tw)
    tw = removeStopWords(tw)
    tw = ' '.join(tw)
    new_tweets.append(tw)
  return new_tweets

def predictNewData(tweets):
  saved_model = joblib.load('Hate Speech Classifier.joblib')
  saved_tfidf = joblib.load('Hate Speech TF-IDF Vectorizer.joblib')
  vectorized_tweets = saved_tfidf.transform(tweets)
  input_prediction = saved_model.predict(vectorized_tweets)
  for i in range(len(input_tweet)):
    if input_prediction[i]==1:
      print('Input text:\n',
        input_tweet[i],
        '\nPrediction: \nHate Speech!\n')
    else:
      print('Input text:\n',
        input_tweet[i],
        "\nPrediction: \nNot a Hate Speech.\n")

predictNewData(input_tweet)
```

Gambar 7. Kode sumber prediksi teks ujaran kebencian atau tidak

```
Input text:
perbuatannya membuat orang lain menjadi percaya bahwa dia adalah ora
ng baik
Prediction:
Not a Hate Speech.

Input text:
toni bajingan
Prediction:
Hate Speech!
```

Gambar 8. Hasil Prediksi teks ujaran kebencian atau tidak

VII. SIMPULAN

A. Kesimpulan

Penelitian ini telah berhasil menghasilkan sebuah sistem deteksi ujaran kebencian dengan menggunakan metode *Support Vector Machine* (SVM) dengan penggunaan data latih 90% dan data uji 10% berhasil mendapatkan hasil skor latih 95,87% dan skor uji 87,30% hasil tersebut memiliki gap terendah di 8,57%. Metode pembanding yang digunakan dalam penelitian ini adalah metode *XGBoost* yang mendapatkan hasil kurang baik jika dibandingkan dengan metode *Support Vector Machine* (SVM). Kemudian dari metode SVM tersebut kemudian dilakukan *hyperparameter Randomized Search Cross Validation* (RSCV) dan berhasil menaikkan skor latih sebesar 100% skor uji sebesar 93,20% dengan gap 6,80% mendapatkan nilai akurasi 93% dengan presisi 91% dan recall 95%. Tahapan terakhir dari penelitian ini adalah memprediksi suatu kata atau kalimat yang mengandung atau tidak mengandung sebuah ujaran kebencian. Hasil akurasi dari sistem yang dibangun ini masihlah tergolong overfit sehingga pada penelitian selanjutnya diharapkan dapat ditemukan hasil gap akurasi yang lebih rendah dengan menggunakan metode dalam machine learning yang lain untuk didapatkan sebuah pembanding.

UCAPAN TERIMA KASIH

Terimakasih saya haturkan kepada Allah SWT. Berkat rahmat dan hidayah-Nya saya bisa menyelesaikan penelitian ini dengan baik. Terimakasih pula kepada kawan-kawan saya yang selalu memberikan semangat moril dan moral untuk segera menyelesaikan penelitian. Terimakasih kepada warga warung kopi slank, jeplak dan bejo yang selalu menemani dan memberikan solusi yang solutif dalam masa pengerjaan penelitian ini. Serta terimakasih kepada janji yang tidak dapat ditepati lagi. Semoga penelitian ini dapat berguna dan disempurnakan pada kemudian hari.

REFERENSI

[1] dan D. E. C. W. Dian Junita Ningrum, Suryadi, "KAJIAN UJARAN KEBENCIAN DI MEDIA SOSIAL," *Dian Junita Ningrum, Suryadi, dan Dian Eka Chandra Wardhana*, vol. 2, no. 3, pp. 241–252, 2018.

[2] F. A. S. Awaluddin, Afif Khalid, "Analisis Yuridis Tentang Pertanggungjawaban Pidana Pelaku Ujaran Kebencian (Hate Speech)," *Univ. Islam Kalimantan*, no. 19, pp. 1–14, 2022, [Online]. Available: http://eprints.uniska-bjm.ac.id/9294/.

[3] M. K. Kelviandy, I. Komputer, and U. Gunadarma, "Kajian Penelitian Pembelajaran Mesin Untuk Pemrosesan Bahasa Alami Dalam Kalimat Perundungan Di Media Sosial," vol. 03, no. 02, pp. 104–108, 2022.

[4] I. Muslim Karo Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *J. Softw. Eng. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 11–18, 2020.

[5] K. Akyol, "Coronary artery disease classification with support vector machines tuned via randomized search," pp. 1–15, 2022.

[6] W. Ayu, R. Abdulhakim, Y. Umaidah, and J. H. Jaman, "Optimasi Support Vector Machine Berbasis Particle Swarm Optimization Untuk Mendeteksi Hate Speech Pilkada Karawang," *J. Appl. Informatics Comput.*, vol. 5, no. 2, pp. 190–201, 2021, doi: 10.30871/jaic.v5i2.3473.

[7] A. Adhari, M. Nasrun, and ..., "Deteksi Ujaran Ancaman Berbasis Website Pada Media Sosial Twitter Menggunakan Metode Support Vector Machine," *eProceedings …*, vol. 8, no. 2, pp. 1920–1925, 2021, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/viewFile/14602/14381.

[8] D. P. N. Lyrawati, "Deteksi Ujaran Kebencian Pada Twitter Menjelang Pilpres 2019 Dengan Machine Learning," *J. Ilm. Mat.*, vol. 7, no. 2, pp. 104–110, 2019.

[9] N. A. Verdikha, T. B. Adji, and A. E. Permanasari, "Komparasi Metode Oversampling Untuk Klasifikasi Teks Ujaran Kebencian," *Semin. Nas. Teknol. Inf. dan Multimed. 2018*, pp. 85–90, 2018.

[10] L. P. A. S. Tjahyanti, "Pendeteksian Bahasa Kasar (Abusive Language) Dan Ujaran Kebencian (Hate Speech) Dari Komentar Di Jejaring Sosial," *J. Chem. Inf. Model.*, vol. 07, no. 9, pp. 1689–1699, 2020.

[11] J. A. Septian, T. M. Fachrudin, and A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *J. Intell. Syst. Comput.*, vol. 1, no. 1, pp. 43–49, 2019, doi: 10.52985/insyst.v1i1.36.

[12] 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.

Conflict of Interest Statement:
The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.