

DETEKSI UJARAN KEBENCIAN MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)

Oleh:

Mohammad Attar Jibrán

Ade Eviyanti

Informatika

Universitas Muhammadiyah Sidoarjo

Agustus, 2023

Latar Belakang

- Ujaran kebencian merupakan suatu fenomena kebahasaan yang menyimpang dari norma tata bahasa yang santun dalam etika berbahasa dan berkomunikasi. 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.
- 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.

Latar Belakang

- 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.
- 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 dan SVM with *Randomized Search Cross Validation* (RSCV) 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.

Metodologi Penelitian

Tinjauan Pustaka

(Wahyuningrum, Rijal Abdulhakim, Yuyun Umaidah, Jajam Haerul Jaman, 2021)

Judul : Optimasi Support Vector Machine Berbasis Particle Swarm Optimization Untuk Mendeteksi Hate Speech Pilkada Karawang.

Penelitian tersebut bertujuan untuk membuat sistem yang dibangun akan mempermudah untuk mendeteksi hate speech yang digunakan oleh pengguna sosial media.

(Ananda Adhari, Muhammad Nasrun S.Si, M.T., Ratna Astuti Nugrahaeni S.T, M.T., 2021)

Judul : DETEKSI UJARAN ANCAMAN BERBASIS WEBSITE PADA MEDIA SOSIAL TWITTER MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Penelitian tersebut bertujuan untuk menguji tingkat akurasi dari dataset menggunakan metode SVM untuk mendeteksi ujaran ancaman.

Metodologi Penelitian

(Dayang Putri Nur Lyrawati, 2019)

Judul : DETEKSI UJARAN KEBENCIAN PADA TWITTER MENJELANG PILPRES 2019 DENGAN MACHINE LEARNING

Penelitian tersebut bertujuan untuk membuat sistem deteksi ujaran kebencian yang menggunakan algoritme SVM-RBF kernel.

(Naufal Azmi Verdikha, Teguh Bharata Adji, Adhistya Erna Permanasari, 2018)

Judul : KOMPARASI METODE OVERSAMPLING UNTUK KLASIFIKASI TEKS UJARAN KEBENCIAN

Penelitian tersebut bertujuan untuk membandingkan penggunaan oversampling SMOTE-SVM dengan SMOTE yang hasilnya lebih cepat SMOTE daripada SMOTE-SVM dengan kinerja klasifikasi yang tidak jauh berbeda.

Metodologi Penelitian

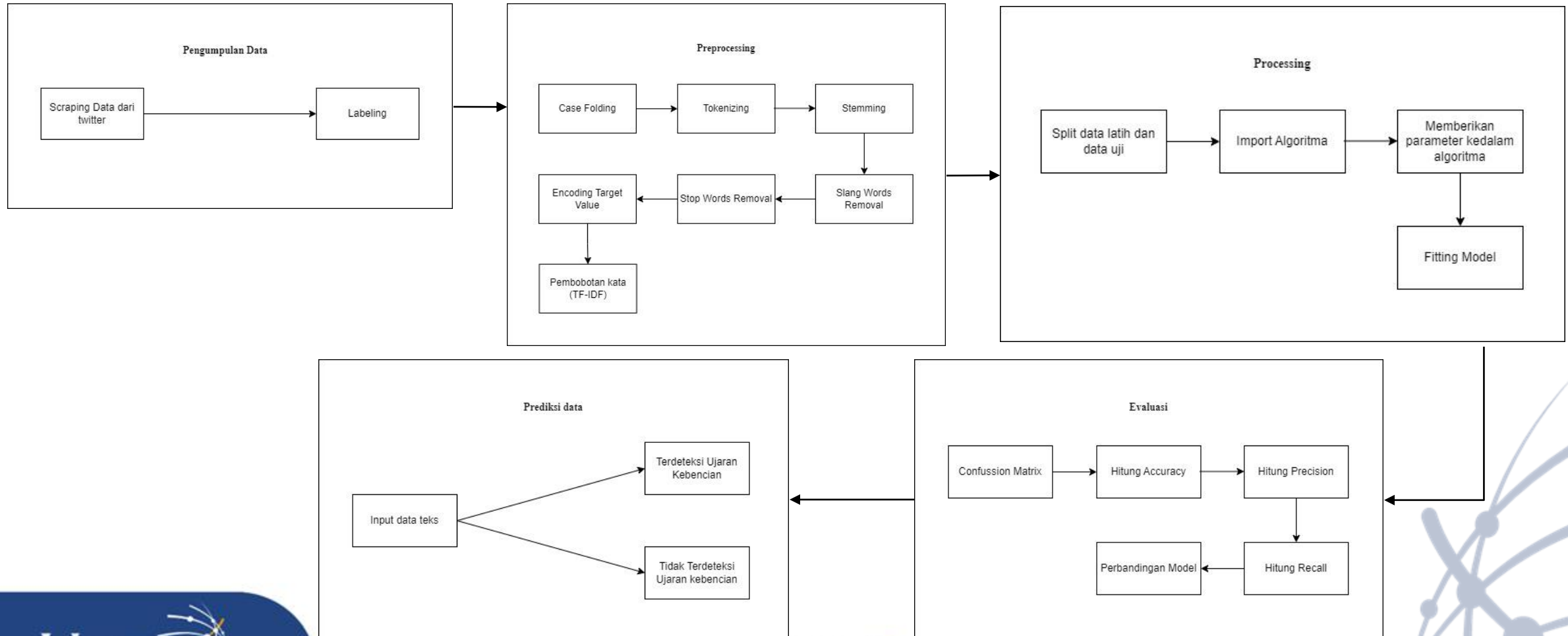
(Luh Putu Ary Sri Tjahyanti, 2020)

Judul : PENDETEKSIAN BAHASA KASAR (ABUSIVE LANGUAGE) DAN UJARAN KEBENCIAN (HATE SPEECH) DARI KOMENTAR DI JEJARING SOSIAL

Penelitian tersebut bertujuan untuk mendeteksi bahasa kasar dan ujaran kebencian sekaligus membandingkan penggunaan 2 label dan 3 label yang hasilnya lebih efektif apabila menggunakan 2 label dengan menyarankan penggunaan kamus huruf besar, tanda baca dan tertawa untuk meningkatkan hasil klasifikasi.

Metodologi Penelitian

Desain Umum Sistem



Metodologi Penelitian

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).

a. 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.

Metodologi Penelitian

b. 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 yang tidak penting.

c. 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.

d. 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.

Metodologi Penelitian

e. 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.

f. 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.

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

- *Term weighting* adalah pembobotan tiap kata yang 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.

Metodologi Penelitian

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. 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*.

Metodologi Penelitian

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.

Hasil dan Pembahasan

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. 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.

Hasil dan Pembahasan

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/it	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, 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...	[luka, aksi, rangka, penjara, ahok, ahok, gaga...	luka aksi rangka penjara ahok ahok gagal pemil...
2	0	Sylvi: bagaimana gubernur melakukan kekerasan...	117	sylvi bagaimana gubernur melakukan kekerasan ...	[sylvi, bagaimana, gubernur, melakukan, keker...	[sylvi, bagaimana, gubernur, laku, keras, per...	[sylvi, gubernur, laku, keras, perempuan, buk...	sylvi gubernur 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, 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 daerah

Hasil dan Pembahasan

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.

Pembagian data latih 90% dan data uji 10%

```
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)
```

Pembagian data latih 80% dan data uji 20%

```
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)
```

Pembagian data latih 75% dan data uji 25%

```
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)
```

Hasil dan Pembahasan

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

Tabel. 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

Hasil dan Pembahasan

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

Tabel. 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

Hasil dan Pembahasan

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

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

Hasil dan Pembahasan

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. Hasil Modeling Metode XGBoost

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

Hasil dan Pembahasan

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. 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 dan Pembahasan

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*:

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

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*:

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

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*:

$$Recall = \frac{TP}{TP + FN}$$

Hasil dan Pembahasan

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

Tabel. 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%.

Hasil dan Pembahasan

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(tweeter):
    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)
```

Hasil dan Pembahasan

Hasil Prediksi Data

- Dari kode sumber diatas mendapatkan hasil seperti berikut:

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

Kesimpulan

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 pembandingan 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 pembandingan.

