KLASIFIKASI DATA TWEET UJARAN KEBENCIAN DI MEDIA SOSIAL MENGGUNAKAN NAIVE BAYES CLASSIFIER

Noor Aliyah Susanti, Miftahul Walid, Hoiriyah
Universitas Islam Madura
Noorkholifah5551@gmail.com

ABSTRAK

Ujaran kebencian banyak dilihat dan sering terjadi di dunia maya, terutama media sosial *Twitter*. Semenjak adanya pemilihan presiden di tahun 2014, masyarakat mulai mengenal *bullying* di dunia maya. Ujaran kebencian, berita-berita *hoax*, bahkan ancaman terhadap pemerintah dan tokoh publik kerap dilakukan. Untuk mengukur sentimen masyarakat terhadap suatu berita maka perlu dilakukan analisis sentimen, khususnya komentar pengguna *Twitter*. Pada penelitian ini, pengujian metode menggunakan *Multinomial Naive Bayes* (MNB) untuk mengukur akurasi klasifikasi ujaran kebencian dalam data *tweet*. Sebelum melakukan perhitungan nilai akurasi, data *tweet* harus diolah melalui teks *preprocessing* agar kata (*term*) dapat dikonversikan ke dalam bentuk matriks. Untuk kemudian diolah sebagai data numerik. Pengujian dilakukan pada dua kondisi pembobotan *n-gram*, yakni *unigram* dan *bigram*. Mulai menghitung nilai akurasi masing - masing pembobotan *Unigram* dan *Bigram* sehingga didapat hasilnya bahwa model perhitungan algoritma *Naive Bayes Classifier* memiliki nilai akurasi yang sama untuk masing - masing pembobotan *n-gram*, yakni 69,23076923076923.

Kata kunci : *analisis,bullying,n-gram,sentimen,twitter,ujaran*

1. PENDAHULUAN

Perundungan (*bullying*) merupakan salah satu kasus yang sering terjadi di Indonesia. Perundungan ini mulai marak dilakukan masyarakat Indonesia semenjak adanya pemilihan presiden (pilpress) di tahun 2014. Berbagai kebijakan pemerintah menjadi sorotan masyarakat dengan berbagai ragam komentar mereka di sosial media. Menyusul dengan berbagai komentar yang mengarah pada ujaran kebencian. Komentar mengejek, mengumpat dan menebarkan kebencian kepada tokoh tertentu.

Pengumpulan data menggunakan *dataset tweet* yang diambil dari situs *www.kaggle.com*. Data merupakan kumpulan data tweet mengenai ujaran kebencian terhadap pemerintah dan tokoh tertentu. Dataset yang berhasil diunduh terdapat sebanyak 5.221 dengan data *tweet* telah berlabelkan sentimen positif dan sentimen negatif. Sentimen positif bernilai 1 dan sentimen negatif bernilai 0.

Dataset yang didapat berbentuk data berupa teks acak dalam file berekstensi *.csv*. Untuk menganalisa data teks yang masih acak menggunakan *text preprocessing* yang berfungsi untuk memilah dan memberi label pada teks yang akan diklasifikasi menggunakan metode *Naive Bayes Classification*.

Menurut surat edaran Kapolri Nomor: SE/06/X/2015 tentang ujaran kebencian (*Hate Speech*) menjelaskan tentang ujaran kebencian yang berupa tindak pidana diatur dalam KUHP dan ketentuan pidana lainnya di luar KUHP yang berbentuk antara lain: 1) Penghinaan, 2) Pencemaran nama baik, 3) Penistaan, 4) Perbuatan tidak menyenangkan, 5) Memprovokasi, 6) Menghasut, 7) Menyebarkan berita bohong.

Penelitian ini bertujuan untuk mengklasifikasikan ujaran kebencian dengan menghitung nilai akurasi pada setiap pembobotan fitur n-gram.

2. TINJAUAN PUSTAKA

Adapun penelitian mengenai ujaran kebencian sebelumnya dilakukan oleh Syafyahya [1] dalam penelitiannya yang berjudul "Ujaran Kebencian dalam Bahasa Indonesia: kajian bentuk dan makna". Dalam penelitian ini, Syafyahya menganalisa mengenai makna yang terdapat dalam ujaran kebencian terbagi menjadi dua, yakni makna konseptual dan makna kontekstual. Dimana makna konseptual merupakan makna bentuk kebahasaan yang bebas konteks. Sedangkan untuk makna kontekstual merupakan makna kebahasaan yang terikat dengan konteks. Dengan penelitian menggunakan data sentimen publik terhadap pemberitaan kasus ujaran kebencian di media sosial dan media massa elektronik.

Selanjutnya. Ningrum dkk [2] melakukan sebuah penelitian dengan judul "Kajian Ujaran Kebencian di Media Sosial". Dalam penelitian Ningrum dkk ini, didapat kesimpulan sebagai berikut: 1) Pada konteks tuturan paling banyak ditemukan ujaran kebencian penistaan agama, dan pada kolom komentar paling banyak ditemukan komentar yang bersifat mencela. 2) Jenis Tindak Tutur Ilokusi (TTI) paling banyak ditemukan tuturan *netizen* di kolom komentar adalah TTI Ekspresif kategori mengkritik.

Menurut Klein [3], ujaran kebencian merupakan segala bentuk tindak tutur (tuturan, tulisan, simbol, gambar, bahasa tubuh, dll) yang mengekspresikan kebencian dalam bentuk verbal.

2.1. *Twitter*

Twitter merupakan layanan jejaring sosial dan mikroblog daring yang memungkinkan penggunanya

untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter akan tetapi pada tanggal 07 November 2017 bertambah hingga 280 karakter yang dikenal dengan sebutan kicauan (*tweet*) [4].

Twitter memiliki banyak pengguna karena beberapa fitur yang dimiliki salah satunya adalah komunitas. Sehingga memungkinkan bisa *sharing* pengetahuan lebih banyak. Informasi yang ada juga bersifat *real time.* Banyak yang menjadikan *Twitter* sebagai personal branding bagi para motivator maupun pelaku usaha. Karena berbagai kemudahan untuk mendapatkan jaringan pertemanan dan komunitas.

2.2. *Data Mining*

Data mining umumnya digunakan untuk menemukan informasi dari sejumlah data yang besar [5]. Teknik data mining digunakan untuk mengimplementasikan dan memecahkan berbagai macam masalah penelitian. Jenis penelitian yang memiliki ruang lingkup *data mining* antara lain: *text mining*, *web mining*, *image mining*, *sequential pattern mining*, *spatial mining*, *medical mining*, *multimedia mining*, *structure mining* dan *graph mining*.

2.3. *Text Mining*

Text mining merupakan salah satu jenis penambangan data (*data mining*) yang menambang informasi yang berguna dari dokumen teks. Penambangan teks juga disebut teknik yang mengekstraksi data teks terstruktur ataupun data teks tidak terstruktur (acak) untuk kemudian menemukan sebuah pola di dalamnya. Teknik penambangan teks banyak digunakan dalam berbagai jenis penelitian, misalnya: *Natural Language Processing*, *Information Retrieval*, *Text Classification* dan *Text Clustering* [6].

Tahapan - tahapan proses dalam *text mining* yang paling umum antara lain: *Tokenizing*, *Stemming*, *Filtering*, *tagging* dan *Analyzing*. Tahapan ini dikenal dengan *text preprocessing*, yaitu tahapan awal sebelum data teks diolah sebagai *training set* dan *test set*.

2.4. *Algoritma Bayes dan Naive Bayes Classifier*

Metode penelitian terbaik untuk klasifikasi data yang menerapkan algoritma pada teorema *Bayes* adalah *Naive Bayes Classfier*. Metode klasifikasi *Naive Bayes* telah banyak digunakan dalam domain penelitian yang bertujuan untuk mengklasifikasi jumlah data yang besar.

Suatu jurnal internasional menyebutkan bahwa *Teorema Bayes* adalah suatu konsep aturan kemungkinan yang benar dan salah untuk dapat diolah menjadi informasi atau pengetahuan tambahan [7].

Berikut rumus teorema bayes:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (1)$$

Klasifikasi pada penelitian ini menggunakan Metode *Naive Bayes Classifier* yang berdasarkan pada rumus teorema *Bayes* untuk probabilitas bersyarat.

2.5. *Bahasa Pemrograman Python*

Python merupakan bahasa pemrograman *interpretatif* yang dapat digunakan di berbagai *platform* dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode dan merupakan salah satu bahasa populer yang berkaitan dengan *Data Science, Machine Learning,* dan *Internet of Thing (IoT)* [8].

Python banyak digunakan untuk *data science* dan *machine learning*. Bahasa pemrograman ini banyak digunakan karena kemudahannya. Manfaat *Python* antara lain: (1) Membersihkan data, (2) Membuat visualisasi dan (3) Membangun model.

Python banyak digunakan karena memiliki *library* yang lengkap, *open source* dan fleksibel sehingga mampu meningkatkan produktivitas *developer*.

2.6. *Anaconda Navigator dan Jupyter Notebook*

Anaconda Navigator adalah antarmuka pengguna grafis desktop (GUI) yang termasuk dalam distribusi *Anaconda* yang memungkinkan pengguna untuk meluncurkan aplikasi dan pengelola paket, lingkungan, dan saluran *Conda* tanpa menggunakan baris perintah [9].

Jupyter berasal singkatan dari tiga bahasa pemrograman yakni *Julia* (Ju), *Python* (Py), dan R. Jupyter notebook merupakan aplikasi web gratis yang paling banyak digunakan oleh *data scientist*. Aplikasi ini dipakai untuk membuat dan membagikan dokumen yang memiliki kode, hasil hitungan, visualisasi dan teks [10].

Dalam penelitian ini menggunakan *Jupyter notebook* untuk menjalankan baris perintah bahasa *Python* dalam menganalisa data teks dan uji model *Naive Bayes Classifier*. *Jupyter notebook* lebih mudah penggunaannya (*user friendly*).

3. METODE PENELITIAN

3.1. Sumber Data

Data yang digunakan dalam penelitian ini bersumber dari situs *www.kaggle.com*. Data sentimen yang diunduh merupakan sekumpulan *tweet* dari pengguna *Twiter* mengenai ujaran kebencian kepada pemerintah. Data telah diberi label 0 dan 1. Nilai 0 disini berarti sentimen negatif dan nilai 1 merupakan sentimen positif. Kemudian data yang berupa sekumpulan teks dari *tweet* pengguna *Twitter* diproses pada tahapan *text preprocessing*.

3.2. Struktur Data

Data yang berjumlah 5.221 data *tweet* tentang ujaran kebencian akan diuji dengan pembagian dataset yakni dengan membagi dataset menjadi 80% *training set* dan 20% *test set*. Sebelumnya dilakukan *text preprocessing* terlebih dahulu pada data *tweet*

sehingga dokumen menjadi sekumpulan kata dasar yang sudah dibersihkan dari kata, huruf maupun simbol dan angka yang tidak dibutuhkan saat pembobotan. Berikut ini contoh struktur data penelitian:

Tabel 1. Contoh struktur data penelitian

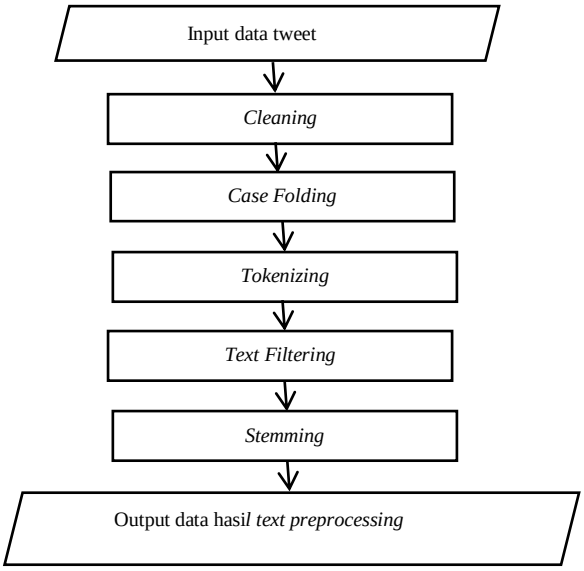
No	Tweet	Sentimen
1	tangkap ahok rakyatbersamafpi indonesia akan hancur tanpa ulama	Negatif
2	benar bebel ini cebong	Negatif
3	pak kiai bela dong banyak umat islam lagi di pojokan bela islam juga ibadah	Positif
4	anies piknik bersama anak yatim dan dhuafa cuma diliput media	Positif
…	…	…

Sentimen negatif merupakan sentimen yang berisi ujaran kebencian terhadap pemerintah.

3.3. Tahapan Analisis Data

Adapun tahapan - tahapan analisis data pada penelitian ini, antara lain:

1) Mengambil data *tweet* di situs *www.kaggle.com.* dan menyimpannya dalam database dengan nama file ujaran_benci .*csv*.
2) Menyiapkan data *tweet* untuk diproses melalui *preprocessing* teks. Dalam tahapan ini, data teks akan dirapikan, dihilangkan kata yang tidak penting, dirapikan apabila ada spasi ganda, serta menghilangkan huruf, angka ataupun simbol-simbol tidak penting melalui proses *tokenizing*.
3) Melakukan pembobotan kata untuk menemukan kata yang sering muncul sehingga nantinya memudahkan untuk menentukan *training set* dan *test set*.
4) Menguji akurasi model dengan *Naive Bayes Classifier.*



Gambar 1. Tahapan proses *preprocessing* teks

4. HASIL DAN PEMBAHASAN

4.1. Menyiapkan data *tweet*

Data *tweet* ujaran kebencian kepada pemerintah di unduh dari situs *www.kaggle.com* dan disimpan ke dalam database *ujaran_benci.csv*.

```
data=pd.read_csv('ujaran_benci.csv')
data.sort_index(inplace=True)
data.head()
```

	tweet,labels
0	tangkap ahok rakyatbersamafpi indonesia akan h...
1	status dukung gerakan matikan lilin untuk ahok...
2	aki amien rais bilang prabowo dengan bung karn...
3	belajar dari negara tetanga vietnam terbukti k...
4	sih cebong sudah kebanyakan makan kotoran joko...

```
data.axes
```

```
[RangeIndex(start=0, stop=5522, step=1),
 Index(['tweet,labels'], dtype='object')]
```

```
data.tail()
```

	tweet,labels
5517	eanjink awas sampai sih biru itu bawa orang ke...
5518	pendahulu memengusir cina kota turun dan turun...
5519	jelas nya ahok atau jokowi diangap bukan mengh...
5520	antena kamu panjang juga hut aku akui mulai ji...
5521	ayo jaga keutuhan nkri tangkap penista agama s...

Gambar 2. Data *tweet* ujaran_benci.csv

Data *Tweet* yang telah di unduh, masih harus dirapikan agar dapat terbaca dan diproses untuk diklasifikasikan. Terdapat 5.521 data *tweet* yang akan dibagi menjadi *data train* dan *data test*. Namun, data *tweet* tersebut terbaca dalam satu kolom. sehingga perlu dilakukan proses split data kolom dengan menggunakan perintah *Python* sehingga didapat hasil data sebagai berikut:

	tweet	label
0	tangkap ahok rakyatbersamafpi indonesia akan h...	0
1	status dukung gerakan matikan lilin untuk ahok...	0
2	aki amien rais bilang prabowo dengan bung karn...	0
3	belajar dari negara tetanga vietnam terbukti k...	0
4	sih cebong sudah kebanyakan makan kotoran joko...	0

Gambar 3. Data *tweet* dengan label

4.2. Mengelompokkan *tweet* berlabel positif dan negatif.

Data tweet yang diunduh merupakan kumpulan sentimen masyarakat terhadap pemerintah dan tokoh publik. Dari sentimen ini dibagi antara yang berlabel positif dan negatif.

Sentimen negatif berisi ujaran yang mengarah kepada ujaran kebencian yang merupakan salah satu jenis *bullying* yang umum dilakukan di media sosial,

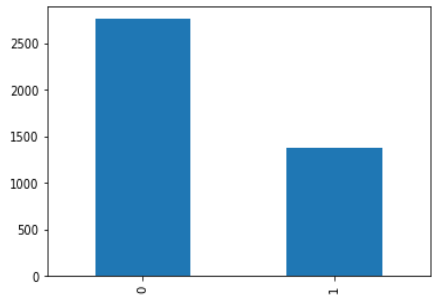
terutama *Twitter*. Berikut ini perintah untuk menampilkan berapa jumlah sentimen positif dan sentimen negatif: *y_train.value_counts().*

```
In [5]: y_train.value_counts()
Out[5]: 0    2945
        1    1472
        Name: label, dtype: int64
```

Gambar 4. Data tweet berlabel positif dan negatif

```
In [7]: # Plot the target label and notice that it is imbalanced

        y_train.value_counts().plot(kind='bar')
Out[7]: <AxesSubplot:>
```



Gambar 5. Grafik *data tweet* berlabel positif dan negatif

4.3. *Text Preprocessing*

Teks yang akan dicari nilai vektornya harus dirapikan dan dibersihkan sehingga bisa dilakukan pembobotan pada tiap kata. Pada *text preprocessing* ini, data *tweet* akan diolah untuk kemudian dihasilkan suatu dokumen yang disusun dari kumpulan kata dasar.

```
Tokenizing Result :

0    [tangkap, ahok, rakyatbersamafpi, indonesia, a...
1    [status, dukung, gerakan, matikan, lilin, untu...
2    [aki, amien, rais, bilang, prabowo, dengan, bu...
3    [belajar, dari, negara, tetanga, vietnam, terb...
4    [sih, cebong, sudah, kebanyakan, makan, kotora...
Name: tweet_tokens, dtype: object
```

Gambar 6. Data *tweet* hasil *tokenizing*

Selanjutnya data akan disimpan ke dalam database "*Teks_pre.csv*" dengan menggunakan perintah: *data.to_csv("Teks_pre.csv").*

Untuk menampilkan kembali data *tweet* yang telah melalui *text preprocessing*, dengan mengetikkan perintah:

```
data_baru = pd.read_csv("Teks_pre.csv")
data_baru.head()
```

Berikut ini tampilan data sebelum dan sesudah text *preprocesing:*

	Unnamed: 0	tweet	label	tweet_tokens
0	0	tangkap ahok rakyatbersamafpi indonesia akan h...	0	['tangkap', 'ahok', 'rakyatbersamafpi', 'indon...
1	1	status dukung gerakan matikan lilin untuk ahok...	0	['status', 'dukung', 'gerakan', 'matikan', 'li...
2	2	aki amien rais bilang prabowo dengan bung karn...	0	['aki', 'amien', 'rais', 'bilang', 'prabowo', ...
3	3	belajar dari negara tetanga vietnam terbukti k...	0	['belajar', 'dari', 'negara', 'tetanga', 'viet...
4	4	sih cebong sudah kebanyakan makan kotoran joko...	0	['sih', 'cebong', 'sudah', 'kebanyakan', 'maka...

Gambar 7. Data *tweet* sebelum dan sesudah *tokenizing*

4.4. Pembobotan kata dengan TF-IDF

Tujuan pembobotan kata ini untuk mengubah kata (*string*) ke dalam bentuk vektor. Sehingga dapat digunakan oleh *machine learning* untuk klasifikasi data pada proses selanjutnya.

Data *tweet* hasil *text preprocessing* akan dilakukan pembobotan pada setiap kata dalam dokumen. Pembobotan TF-IDF akan dilakukan dengan dua perhitungan pembobotan *n-gram*, yaitu: *Unigram* dan *Bigram* untuk melihat perbandingan akurasi pengujian model *Multinomial Naive Bayes* pada ketiga *term weighting* tersebut.

Berikut hasil pembobotan *TF-IDF Unigram* dengan contoh index data ke-0 :

```
Show TFIDF sample ke-0

tangkap ahok rakyatbersamafpi indonesia akan hancur tanpa ulama

                        TF          IDF         TF-IDF      Term

array position 13       0.142857    4.522745    0.646106    ahok
array position 15       0.142857    4.131698    0.590243    akan
array position 302      0.142857    6.090135    0.870019    hancur
array position 331      0.142857    3.295727    0.470818    indonesia
array position 893      0.142857    6.672056    0.953151    tangkap
array position 895      0.142857    5.368000    0.766857    tanpa
array position 966      0.142857    5.556052    0.793722    ulama
```

Gambar 8. Data hasil pembobotan TF-IDF

Kemudian data hasil pembobotan *n-gram* akan disimpan dalam database dengan perintah sebagai berikut:

```
data_baru[["tweet", "label","TF_UNIGRAM",
"IDF_UNIGRAM",
"TFIDF_UNIGRAM"]].to_excel("TFIDF_Unigram.xlsx")
```

TF_UNIGRAM	IDF_UNIGRAM	TFIDF_UNIGRAM
[0.14285714285714285, 0.14285714285714285, 0.1...	[4.522745191683338, 4.131698458999445, 6.09013...	[0.646106459594762 6, 0.5902426369999206, 0.870...
[0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.1...	[4.522745191683338, 5.083895899336844, 5.80983...	[0.5653431489604173, 0.6354869874171055, 0.726...
[0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.1...	[4.605860098393844, 4.845810768024435, 3.35309...	[0.5757325122992305, 0.6057263460030544, 0.419...
[0.07142857142857142, 0.07142857142857142, 0.0...	[6.5254529391317835, 5.665251673908672 5, 3.353...	[0.4661037813665559 5, 0.4046608338506194 6, 0.2...
[0.06666666666666667, 0.06666666666666667, 0.0...	[4.369471320329614, 3.5319959794149285, 3.7642...	[0.2912980880219742 7, 0.2354663986276619, 0.25...

Gambar 9. Hasil perhitungan TF-IDF *Unigram*

Kemudian data hasil pembobotan *n-gram* akan disimpan dalam database dengan perintah sebagai berikut:

```
data_baru[["tweet", "label","TF_UNIGRAM",
"IDF_UNIGRAM",
"TFIDF_UNIGRAM"]].to_excel("TFIDF_Unigram.xlsx")
```

	TF_UNIGRAM	IDF_UNIGRAM	TFIDF_UNIGRAM
	[0.14285714285714285, 0.14285714285714285, 0.1...	[4.522745191683338, 4.131698458999445, 6.09013...	[0.6461064559547626, 0.5902426369999206, 0.870...
	[0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.1...	[4.522745191683338, 5.083895899336844, 5.80983...	[0.5653431489604173, 0.6354869874171055, 0.726...
	[0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.1...	[4.605860098393844, 4.845810768024435, 3.35309...	[0.5757325122992305, 0.6057263460030544, 0.419...
	[0.07142857142857142, 0.07142857142857142, 0.0...	[6.5254529391317835, 5.665251673908725, 3.353...	[0.46610378136655595, 0.40466083385061946, 0.2...
	[0.06666666666666667, 0.06666666666666667, 0.0...	[4.369471320329614, 3.5319959794149285, 3.7642...	[0.29129808802197427, 0.2354663986276619, 0.25...

Gambar 10. Hasil perhitungan TF-IDF *Unigram*

Pada pembangkitan kata metode n-gram digunakan untuk mengambil potongan kata sejumlah n dari sekumpulan kata. Begitupula untuk TF-IDF *Bigram* kata yang dibobot berdasarkan per dua kata, berikut ini baris perintah untuk fitur *bigram:*

```
def get_TF_bigram(row):
    idx = row.name
    return [tf for tf in tf_mat_bigram[idx] if tf != 0.0]

data_baru["TF_BIGRAM"] =
data_baru.apply(get_TF_bigram, axis=1)

def get_IDF_bigram(row):
    idx = row.name
    return [item[1] for item in zip(tf_mat_bigram[idx],
idf_mat_bigram) if item[0] != 0.0]

data_baru["IDF_BIGRAM"] =
data_baru.apply(get_IDF_bigram, axis=1)

def get_TFIDF_bigram(row):
    idx = row.name
    return [tfidf for tfidf in tfidf_mat_bigram[idx] if tfidf !=
0.0]

data_baru["TFIDF_BIGRAM"] =
data_baru.apply(get_TFIDF_bigram, axis=1)

def get_Term_bigram(row):
   idx = row.name
   return [item[1] for item in zip(tf_mat_bigram[idx],
terms_bigram) if item[0] != 0.0]

data_baru["TWEET_BIGRAM"] =
data_baru.apply(get_Term_bigram, axis=1)

data_baru[["TWEET_BIGRAM", "TF_BIGRAM",
"IDF_BIGRAM", "TFIDF_BIGRAM"]].head()
```

Kemudian menyimpan hasil pembobotan *Bigram* dengan perintah:

```
data_baru[["tweet","label","TWEET_BIGRAM",
"TF_BIGRAM", "IDF_BIGRAM",
"TFIDF_BIGRAM"]].to_excel("TFIDF_Bigram.xlsx")
```

Sehingga hasil dari perhitungan TF-IDF *Bigram* dapat dilihat pada gambar berikut ini :

	TWEET_BIGRAM	TF_BIGRAM	IDF_BIGRAM	TFIDF_BIGRAM
	['indonesia akan']	[1.0]	[7.41927081515388]	[7.41927081515388]
	['akun facebok']	[1.0]	[5.878825774206732]	[5.878825774206732]
	['bung karno']	[1.0]	[8.007057480055998]	[8.007057480055998]
	['belajar dari']	[1.0]	[7.824735923262045]	[7.824735923262045]
	['cebong sudah', 'jokowi sudah', 'makan kotoran']	[0.3333333333333333, 0.3333333333333333, 0.333...	[7.824735923262045, 7.670585243434786, 7.67058...	[2.608245307754015, 2.5568617478115954, 2.5568...

Gambar 11. Hasil hitung TF-IDF *Bigram*

4.5. Pengujian metode dengan *Naive Bayes*

Pada penelitian ini, pengujian metode menggunakan *Multinomial Naive Bayes* (MNB) untuk mengukur akurasi klasifikasi ujaran kebencian dalam data *tweet*.

Pengujian dilakukan pada dua kondisi pembobotan *n-gram*, yakni *unigram* dan *bigram*. Mulai menghitung nilai akurasi masing - masing pembobotan *Unigram* dan *Bigram*. Dengan baris perintah sebagai berikut:

```
from sklearn.naive_bayes import MultinomialNB
# fit the training dataset on the NB classifier
Naive1 = MultinomialNB()
Naive1.fit(Train_X1_Tfidf,Train_Y1)

Naive2 = MultinomialNB()
Naive2.fit(Train_X2_Tfidf,Train_Y2)

# predict the labels on validation dataset
predictions_NB1 = Naive1.predict(Test_X1_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score ->
",accuracy_score(predictions_NB1, Test_Y1)*100)
predictions_NB2 = Naive2.predict(Test_X2_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score ->
",accuracy_score(predictions_NB2, Test_Y2)*100)
```

Pengujian dengan membagi dataset menjadi 80 % *training set* dan 20% *test set*. Maka akan didapat hasil perhitungan akurasi *Naive Bayes* untuk *Unigram* sebesar 69,23076923076923 dan *Bigram* sebesar 69,230762307623. Sebagaimana gambar berikut ini:

Pembobotan TF-IDF ---------	UNIGRAM	BIGRAM
Naive Bayes Accuracy Score	69.23076923076923	69.23076923076923

Gambar 12. Nilai akurasi *Naive Bayes*

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian prediksi dengan model perhitungan algoritma *Naive Bayes Classifier* dengan nilai akurasi mendekati 70% termasuk dalam kategori *good*. Sebelum melakukan perhitungan nilai akurasi, data *tweet* harus diolah melalui teks *preprocessing* agar kata (*term*) dapat dikonversikan ke dalam bentuk matriks. Untuk kemudian diolah sebagai data numerikDengan perhitungan *matrix* dan split *data twee*t menggunakan model *Multinomial Naive Bayes* dapat disimpulkan bahwa perhitungan pembobotan TF-IDF memiliki nilai akurasi yang sama untuk masing - masing pembobotan *n-gram*, yakni 69,23076923076923. Disarankan untuk penelitian selanjutnya menguji nilai akurasi dengan metode klasifikasi yang lain sebagai pembanding.

DAFTAR PUSTAKA

[1] Syafyahya, Leni. "Ujaran Kebencian Dalam Bahasa Indonesia: Kajian Bentuk Dan Makna". Makalah Kongres KBI, 2018.

[2] Ningrum, Dian Junita. Suryadi, dan Dian Eka C W. "Kajian Ujaran Kebencian Di Media Sosial", Jurnal Ilmiah Korpus, Volume II, Nomor III, Desember 2018.

[3] Klein, G. B. "*Applied Linguistics to Identify and Contrast Racist 'Hate Speech': Cases from the English and Italian Language"*. Applied Linguistics Research Journal,2(3), 1–16. 2018. . https://doi.org/10.14744/alrj.2018.36855

[4] Wikipedia." *Twitter*". Diakses tanggal 15 Mei 2022, dari wikipedia: https://id.wikipedia.org/wiki/Twitter.

[5] Vijayarani et al, "*Preprocessing Techniques for Text Mining - An Overview*", International Journal of Computer Science & Communication Networks, Vol5(1),7-16. 2015.

[6] Kalra, Vaishali. Aggarwal, Rashmi. "*Importance of Text Data Preprocessing & Implementation in Rapidminer*". In proc. FIC on Information and Knowledge Management, vol.14, pp.71-75. 2018. DOI: 10.15439/2018KM46.

[7] Hadini, F.M. "Detection *System Milkfish Formalin Android-Based Method Based on Image Eye Using Naive Bayes Classifier"*.9(1),2-5. 2017.

[8] Dicoding. "Deskripsi *Python*". diakses tanggal 20 Mei 2022, dari dicoding: https://www.dicoding.com/academies/86.

[9] Wikipedia." *Anaconda* (*Python Distribution*)". Diakses tanggal 20 Mei 2022, dari wikipedia: https://en.wikipedia.org/wiki/Anaconda_(Python_distribution).

[10] Algorit.ma. "*Jupyter: Pengertian, Fitur dan Fungsi*". diakses tanggal 20 Mei 2022, dari algorit.ma: https://algorit.ma/blog/cara-menggunakan-jupyter-notebook-2022/