

# IMPLEMENTASI TEXT MINING UNTUK DETEKSI UJARAN KEBENCIAN TERHADAP IBU KOTA NUSANTARA MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBORS* PADA PLATFORM X

Ilham Zakariya<sup>1</sup>, M. Syafrullah<sup>2\*</sup>

<sup>1,2</sup>Teknik Informatika, Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: <sup>1</sup>2011500622@budiluhur.ac.id, <sup>2\*</sup>mohammad.syafrullah@budiluhur.ac.id  
(\* : corresponding author)

**Abstrak-** Ibu Kota Nusantara (IKN) telah menimbulkan perdebatan sengit di media sosial, dengan beberapa diskusi berkembang menjadi ujaran kebencian. Penelitian ini bertujuan mendeteksi ujaran kebencian terhadap IKN di media sosial X, memberikan wawasan terbaru dalam menghadapi isu sensitif secara digital. Data pada penelitian ini menggunakan 1800 *posts* di media sosial X yang relevan dengan topik IKN atau Ibu Kota Nusantara dengan rentang 24 Mei 2024 hingga 14 Juni 2024. Penelitian ini menggunakan metode *K-Nearest Neighbors* (KNN), sebuah algoritma yang tidak memerlukan model pelatihan sebelumnya dan beroperasi dengan mengklasifikasikan dan mendeteksi data berdasarkan kedekatan jarak *k*. Data yang telah dikumpulkan dari media sosial X dianalisis menggunakan pendekatan *text mining* dengan teknik ekstraksi fitur TF-IDF sebagai vektorisasi untuk mengubah data teks menjadi format numerik. Pengujian dilakukan dengan berbagai rasio split data dan berbagai jarak *k* untuk mengevaluasi konsistensi model. Hasil penelitian menunjukkan bahwa KNN mampu mendeteksi ujaran kebencian dengan akurasi yang memuaskan, hasil Akurasi terbaik adalah pada rasio 50:30 dan jarak *k* bernilai 3 serta tanpa *balance* data yaitu 88%, performa *Precision* terbaik adalah pada rasio 70:30 dan jarak *k* bernilai 5 serta dengan *balance* data yaitu 95%, performa *Recall* terbaik adalah pada rasio 60:40 dan jarak *k* bernilai 9 serta tanpa *balance* data yaitu 99%, dan terakhir performa *F1-Score* terbaik adalah pada rasio 50:30 dan jarak *k* bernilai 7 serta tanpa *balance* data yaitu 95%.

**Kata Kunci:** ujaran kebencian, *ikn*, deteksi, *k-nearest neighbors*

## IMPLEMENTATION OF TEXT MINING FOR DETECTING HATE SPEECH TOWARDS THE NEW CAPITAL OF NUSANTARA USING THE *K-NEAREST NEIGHBORS* ALGORITHM ON PLATFORM X

**Abstract -** The new capital of Nusantara (IKN) has sparked intense debates on social media, with some discussions escalating into hate speech. This research aims to detect hate speech towards IKN on social media platform X, providing the latest insights into addressing sensitive issues digitally. The data for this research consists of 1800 posts on social media platform X relevant to the topic of IKN or the new capital of Nusantara, collected from May 24, 2024, to June 14, 2024. This study uses the *K-Nearest Neighbors* (KNN) method, an algorithm that does not require prior training and operates by classifying and detecting data based on the proximity of *k*-nearest neighbors. The collected data from social media platform X was analyzed using a text mining approach with TF-IDF feature extraction technique for vectorization to convert text data into numerical format. Testing was conducted with various data split ratios and different values of *k* to evaluate the consistency of the model. The results show that KNN can detect hate speech with satisfactory accuracy. The best accuracy result was 88% with a 50:30 split ratio and *k* value of 3 without balanced data. The best precision performance was 95% with a 70:30 split ratio and *k* value of 5 with balanced data. The best recall performance was 99% with a 60:40 split ratio and *k* value of 9 without balanced data. Finally, the best *F1-score* performance was 95% with a 50:30 split ratio and *k* value of 7 without balanced data.

**Keywords:** hate speech, *IKN*, detection, *k-nearest neighbors*

## 1. PENDAHULUAN

Ibu Kota Negara Nusantara (IKN) adalah proyek pemerintah Indonesia untuk menggantikan DKI Jakarta sebagai ibu kota, yang telah menjadi pusat pemerintahan selama 63 tahun. IKN dibangun untuk mencapai visi Indonesia sebagai negara maju pada tahun 2045, menjadi identitas nasional, dan mendorong pembangunan yang Indonesia-sentris serta pertumbuhan ekonomi. Namun, keputusan ini memicu perdebatan di media sosial X (sebelumnya Twitter), dengan opini publik yang terpecah. Banyak tanggapan negatif yang mengandung ujaran kebencian terhadap pemerintah, terutama setelah diberlakukannya Undang-Undang No. 3 Tahun 2022 tentang Materi Pokok Peraturan IKN, yang semakin meningkatkan volume ujaran kebencian di platform tersebut.

Ujaran kebencian atau *Hatespeech* dapat dilakukan secara lisan, tulisan, ataupun gambar yang menyerang seseorang atau instansi, yang dapat merugikan seseorang atau instansi tersebut [1]. Lalu Christianto berpendapat bahwa memprovokasi, penyebaran berita bohong, dan pencemaran nama baik di jejaring sosial media itu adalah termasuk Ujaran Kebencian [2].

Menurut Kominfo, sejak 2018 terdapat 3.640 konten ujaran kebencian berbasis SARA dan politik di *platform* digital. Selama kampanye pemilu 2024, Katadata mencatat 182.118 unggahan ujaran kebencian di media sosial, dengan jenis terbanyak berupa serangan terhadap identitas, hinaan, dan kata-kata kotor. Salah satu contoh terbaru adalah kasus Edy Mulyadi, yang menghina IKN di Kalimantan Timur melalui YouTube dan dijatuhi hukuman 7,5 bulan penjara.

Penelitian terdahulu telah memberikan kontribusi signifikan dalam upaya klasifikasi dan deteksi ujaran kebencian di media sosial menggunakan berbagai metode *K-Nearest Neighbors*. Misalnya, penelitian oleh [3], menggunakan metode Naïve Bayes dan N-Gram pada *dataset* multi-label Twitter berbahasa Indonesia. Hasil terbaik mereka diperoleh dengan kombinasi fitur *word* unigram, *word* bigram, dan *recall*, mencapai *F-score* sebesar 64,957%. Selain itu, [4] dalam skripsinya mengeksplorasi penggunaan algoritma *K-Nearest Neighbors* (KNN) untuk mengklasifikasikan atau mendeteksi ujaran kebencian dengan kelas *hate speech* dan *non hate speech* yang berhubungan dengan penggemar K-pop di Twitter. Hasil penelitiannya menunjukkan akurasi tertinggi 78.38% dengan menggunakan 1106 data *posts* yang terdiri dari 251 data ujaran kebencian dan 855 data *non*-ujaran kebencian, dan juga terdiri dari 884 jumlah *posts* pada data latih dan 222 jumlah *posts* pada data uji.

Oleh karena itu, penelitian ini diharapkan tidak hanya melanjutkan upaya sebelumnya dalam mendeteksi ujaran kebencian di media sosial, tetapi juga mengadaptasikannya dalam konteks yang sangat relevan dan aktual di Indonesia, yaitu IKN. Dengan mengkombinasikan metode *text mining*, perbedaan rasio dan algoritma *K-Nearest Neighbors* (KNN), diharapkan penelitian ini dapat memberikan wawasan baru dan solusi praktis dalam menangani ujaran kebencian terkait IKN di media sosial X.

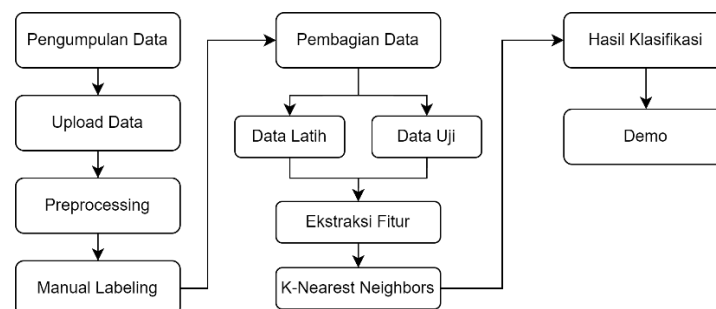
## 2. METODE PENELITIAN

### 2.1 Pengumpulan Data

*Dataset* atau data yang digunakan pada penelitian ini adalah *posts* yang bersumber dari media sosial X. Data yang digunakan adalah merupakan data *posts* yang dikumpulkan sejak tanggal 24 Mei 2024 hingga 14 Juni 2024 yang berjumlah 1800 *posts*. Data tersebut diperoleh dengan sebuah script atau program bernama *twscrape*. Kata kunci yang digunakan untuk pengumpulan *dataset posts* merupakan *ikn*, dan ibu kota nusantara.

### 2.2 Penerapan Metode

Dalam membangun program deteksi ujaran kebencian menggunakan *K-Nearest Neighbors*, diperlukan beberapa langkah. Langkah-langkah tersebut kemudian dilakukan untuk menjalankan program dari awal hingga akhir secara sempurna. Berikut langkah-langkah tersebut pada gambar 1.



Gambar 1. Tahapan Metode

Hal yang pertama dilakukan adalah tahapan pengumpulan data, data dikumpulkan dari media sosial X menggunakan program *twscrape* dengan kata kunci "*ikn*" dan "*ibu kota nusantara*." Data tersebut disimpan dalam format CSV. Setelah dilakukan proses pengumpulan data, lalu menuju tahap *Upload Data* yaitu data diseleksi oleh di Program Excel dan diunggah ke sistem untuk diperlihatkan rincian *dataset* tersebut.

Setelah dilakukan *Upload Data*, tahap selanjutnya adalah *Preprocessing* untuk membersihkan data *posts* dari kata-kata yang tidak bermakna. Tahap ini dilakukan dengan data yang didapatkan pada tahap pengumpulan data *posts*. Penelitian ini menggunakan beberapa teknik *Preprocessing* yaitu teknik *Decode HTML Entities*, teknik *Case*

*Folding*, teknik Menghilangkan Simbol, teknik mengganti *Word Elongation*, teknik *Normalize Slang*, Teknik menghapus *Stopword*, dan yang terakhir teknik *Stemming* [5].

- a. *Decode HTML Entities*: Proses ini melibatkan perubahan simbol dan emoji pada html menjadi teks atau simbol yang dapat diproses saat proses analisis, agar melancarkan proses klasifikasi.
- b. *Case Folding*: Proses ini melibatkan pengubahan semua huruf dalam *dataset* menjadi huruf kecil untuk mengeliminasi perbedaan antara huruf besar dan kecil, sehingga memudahkan proses klasifikasi.
- c. Menghilangkan Simbol: Tahap ini fokus pada penghapusan simbol-simbol yang tidak relevan seperti #, ., /, ? yang tidak memberikan nilai tambah dalam analisis, memastikan data lebih bersih dan siap untuk klasifikasi.
- d. Mengganti *Word Elongation*: Proses ini menangani kata-kata dengan huruf berulang, seperti "kenapaa" atau "bagaimanaa", dengan menggantinya menjadi bentuk standar. Hal ini membantu menyederhanakan teks dan meningkatkan akurasi klasifikasi.
- e. *Normalize Slang*: Proses ini mengubah kata *non baku* menjadi bentuk baku sesuai Kamus Besar Bahasa Indonesia (KBBI), contohnya mengganti "emg" menjadi "emang" dan "knp" menjadi "kenapa", yang membantu menyelaraskan data dengan standar bahasa yang lebih formal.
- f. Menghapus *Stopword*: Proses ini melibatkan penghapusan kata-kata umum yang sering muncul tetapi tidak memiliki signifikansi semantik yang kuat, seperti "di", "ke", dan "yang", untuk fokus analisis pada kata-kata yang lebih informatif.
- g. *Stemming*: *Stemming* mengurangi kata-kata berimbuhan ke bentuk dasar mereka, seperti mengubah "menahan" menjadi "tahan" dan "berbalas-balasan" menjadi "balas". Ini membantu dalam menyederhanakan kata-kata sehingga mudah dianalisis dan diklasifikasikan.

Setelah tahap *preprocessing*, selanjutnya adalah tahap *Manual Labeling* yang menandakan *posts* kedalam kelas-kelas spesifik berdasarkan karakteristik tertentu [6]. Pelabelan dalam penelitian ini dibagi menjadi dua kategori: ujaran kebencian dan *non-ujaran* kebencian, dilakukan secara manual setelah tahap *preprocessing*. Proses ini tidak hanya fokus pada konten eksplisit tetapi juga pada analisis mendalam makna linguistik dari setiap kata dan frasa menggunakan pendekatan makna konseptual. Pendekatan ini memahami kata berdasarkan arti leksikal standar tanpa mempertimbangkan konteks. Meskipun data telah dibersihkan, analisis linguistik tetap memungkinkan penilaian yang objektif dan akurat. Proses *Manual Labeling* ini dibimbing oleh ibu Saskia Lydiani, S.Pd., M.Si sebagai ahli pakar.

Setelah tahap *Manual Labeling*, selanjutnya adalah tahap Pembagian data yang dilakukan dengan rasio 50:50, 60:40, 70:30, 80:20, dan 90:10 untuk menghindari bias. Contohnya, pada rasio 70:30, 70% data digunakan untuk pelatihan (420 data latih) dan 30% untuk pengujian (180 data uji). Tahap ini bertujuan menyediakan data yang cukup untuk pelatihan model sambil mempertahankan data untuk validasi dan pengujian kinerja algoritma secara adil dan efektif.

Lalu selanjutnya adalah tahap *Balancing* yang bersifat opsional. Tahap ini menggunakan *Random Undersampling* untuk *balancing* dengan tujuan untuk meningkatkan performa prediksi dengan mengurangi bias terhadap kelas mayoritas [7]. Tahap ini menurunkan kelas mayoritas pada data latih antara kelas *Hatespeech* dan *Non Hatespeech*.

Setelah tahapan *Balancing*, tahapan selanjutnya adalah Ekstraksi Fitur dimana menggunakan pembobotan TF-IDF untuk mengubah teks menjadi representasi vektor pada data latih. *Term Frequency (TF)* mengukur frekuensi kemunculan kata dalam dokumen, sementara *Inverse Document Frequency (IDF)* menghitung kebalikan dari jumlah dokumen yang mengandung kata tersebut. Teknik ini sering digunakan sebagai pencarian informasi untuk meningkatkan hasil klasifikasi [8].

Rumus Pembobotan *TF-IDF* dapat dilihat pada persamaan 1 dan persamaan 2 berikut:

$$TF(w_i, d_j) = \frac{\text{Frekuensi kata } w_i \text{ dalam dokumen } d_j}{\text{Total kata dalam dokumen } d_j} \quad (1)$$

$$IDF(N, df_i) = \log\left(\frac{N}{df_i}\right) \quad (2)$$

$$TF - IDF = TF(w_i, d_j) \times IDF(N, df_i)$$

Setelah vektor didapatkan dari Pembobotan *TF-IDF*, data vektor disimpan dalam *database* dalam bentuk JSON sehingga dapat diproses ketika pada tahap penerapan metode *K-Nearest Neighbors*.

Setelah tahapan ekstraksi fitur dengan pembobotan *TF-IDF* dan telah didapatkan vektor dari teks. Selanjutnya adalah tahapan perhitungan *K-Nearest Neighbors (KNN)*. *KNN* adalah metode atau algoritma yang digunakan untuk klasifikasi berdasarkan jarak k tetangga terdekat, *KNN* termasuk algoritma *Lazy Learning* yang dimana

tidak memiliki model [9]. Metode ini mengukur jarak antar titik data dan mengidentifikasi k sampel terdekat untuk menentukan kelasnya, menggunakan *Euclidean distance* atau *Manhattan distance*, tergantung pada struktur data [10].

Rumus *Euclidean Distance* dapat dilihat pada persamaan 3 berikut:

$$Euclidean\ Distance(A, B) = \sum_{i=1}^t \sqrt{(A - B)^2} \quad (3)$$

Data latih yang sudah terbobot digunakan untuk mengklasifikasikan data uji dengan mencari k tetangga terdekat berdasarkan jarak vektor menggunakan *Euclidean Distance*. KNN mengategorikan data uji berdasarkan mayoritas label dari tetangga terdekatnya, melibatkan pembuatan vektor uji, perhitungan jarak, penentuan tetangga terdekat, dan penentuan klasifikasi akhir.

Untuk Menguji performa dari metode KNN, peneliti menggunakan teknik pengujian *Confusion Matrix* Tabel 1. *Confusion Matrix* adalah metode untuk mengevaluasi performa model klasifikasi dengan menampilkan frekuensi *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN). Matriks ini membantu menghitung akurasi serta metrik lain seperti *precision* dan *recall*, memberikan wawasan lebih dalam tentang kinerja model pada kelas yang berbeda [11].

**Tabel 1.** *Confusion Matrix*

	Prediksi Kelas <i>Hatespeech</i>	Prediksi Kelas <i>Non Hatespeech</i>
Aktual Kelas <i>Hatespeech</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
Aktual Kelas <i>Non Hatespeech</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Pada pengujian kali ini akan ditemukan Akurasi, Presisi, *Recall*, dan *F1-Score* dari metode yang telah diimplementasi guna mengevaluasi hasil dari metode tersebut. Persamaan 4, Akurasi adalah mengukur keefektifan model dalam mengklasifikasikan kasus dengan benar.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Persamaan 5, Presisi adalah mengukur ketepatan model dalam mengidentifikasi label positif.

$$Presisi = \frac{TP}{TP + FP} \quad (5)$$

Persamaan 6, *Recall* adalah mengukur ketepatan model untuk mendeteksi semua kasus positif aktual dari data.

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

Persamaan 7, *F1-Score* adalah mengukur rata-rata keseimbangan dari presisi dan *recall*.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{precision + Recall} \quad (7)$$

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengumpulan Data

Data untuk penelitian ini dikumpulkan dari media sosial X menggunakan program twscrape dengan kata kunci "ikn" dan "ibu kota nusantara" sejak 24 Mei 2024 hingga 14 Juni 2024, menghasilkan 1800 *posts*. Data disimpan dalam format CSV. Contoh data dapat dilihat pada Tabel 2.

**Tabel 2.** Contoh *Dataset Posts*

No	Posts
1	uang ikn itu utang dari china
2	ikn punya rezim china hutang belum lunas
3	ikn untuk ekonomi indonesia maju
4	indonesia punya ikn bagaikan negara maju sangat keren
5	Taksi Terbang IKN Diuji Coba Mulai Juni

### 3.2 Preprocessing dan Manual Labeling

Di tahap ini, data yang telah dikumpulkan dilakukan proses *preprocessing* untuk menghilangkan informasi yang tidak diperlukan ketika proses pengklasifikasian teks.

**Tabel 3.** Sebelum dan Sesudah *Preprocessing*

No	<i>Posts Sebelum Preprocessing</i>	<i>Posts Sesudah Preprocessing</i>
1	uang ikn itu utang dari china	uang ikn hutang dari china
2	ikn punya rezim china hutang belum lunas	ikn punya rezim china hutang belum lunas
3	ikn untuk ekonomi indonesia maju	ikn ekonomi indonesia maju
4	indonesia punya ikn bagaikan negara maju sangat keren	indonesia punya ikn bagai negara maju keren
5	Taksi Terbang IKN Diuji Coba Mulai Juni	taksi terbang ikn uji coba juni

Setelah *dataset* dilakukan tahapan *preprocessing* seperti contoh Tabel 3, selanjutnya adalah tahap *Manual Labeling* dua kelas yaitu *Hatespeech* dan *Non Hatespeech* oleh rekomendasi pakar yang bersangkutan terkait dengan ilmu teks dan psikologi tersebut, dan didapatkanlah *dataset* yang sudah di lakukan tahapan *Preprocessing* dan *Manual Labeling*.

### 3.3 Split Data dan Balance Data

Tahap selanjutnya adalah tahapan *Split Data* lalu *Balancing Data*, data yang sudah di-*preprocessing* akan di bagi menjadi beberapa rasio data uji dan data latih yaitu 50:50, 60:40, 70:30, 80:20, dan 90:10. Sebagai contoh pada Tabel 4, jika dilakukan *split data* dengan rasio 70:30 dimana 70% dari *dataset* tersebut ialah data latih, sedangkan 30% dari data tersebut ialah data uji. Lalu setelah tahapan *Split Data* tersebut jika kelas jumlah kelas pada data latih tidak seimbang secara jauh, maka akan dilakukan *Balance Data* menggunakan teknik *Random Undersampling*.

**Tabel 4.** *Split Data* dan *Balance Data*

1800 Data	Tanpa <i>Balance Data</i>			Dengan <i>Balance Data</i>	
	Rasio	Data Latih	Data Uji	Data Latih	Data Uji
	50:50	900	900	730	900
	60:40	1080	720	850	720
	70:30	1260	540	1004	540
	80:20	1440	360	1136	360
	90:10	1620	180	1292	180

Sebagai contoh jika ketika melakukan *Balance Data* pada rasio pembagian data 70:30 yang memiliki jumlah data kelas *Hatespeech* 758 dan jumlah data kelas *Non Hatespeech* 502, maka kelas mayoritas akan diturunkan menggunakan teknik *Random Undersampling* yaitu kelas *Hatespeech* menjadi berjumlah 502.

### 3.4 Ekstraksi Fitur Pembobotan TF-IDF

Setelah membagi data, tahap berikutnya adalah Ekstraksi Fitur menggunakan TF-IDF untuk mengubah teks menjadi format vektor. Teknik ini menonjolkan kata-kata penting dalam dokumen dan memberikan bobot pada informasi unik dan relevan. Data latih Tabel 5 yang telah diolah digunakan dalam pengujian dengan *K-Nearest Neighbors* (KNN) untuk menilai efektivitas model dalam mengklasifikasikan data uji, dengan tujuan membangun model yang akurat dan efisien untuk mendeteksi ujaran kebencian.

**Tabel 5.** Contoh Data Latih

Dokumen	Konten	Label
D1	uang ikn hutang dari china	<i>Hatespeech</i>
D2	ikn punya rezim china hutang belum lunas	<i>Hatespeech</i>
D3	ikn ekonomi indonesia maju	<i>Non hatespeech</i>
D4	indonesia punya ikn bagai negara maju keren	<i>Non hatespeech</i>

Lalu dilakukan Perhitungan *TF* dan *IDF* seperti Tabel 6:

**Tabel 6.** Contoh Perhitungan *TF* dan *IDF*

Term	<i>TF</i>	<i>TF</i>	<i>TF</i>	<i>TF</i>	dFi	<i>IDF</i>
	D1	D2	D3	D4		
uang	0.2	0	0	0	1	1.386

Term	TF D1	TF D2	TF D3	TF D4	dFi	IDF
ikn	0.2	0.143	0.250	0.143	4	0
hutang	0.2	0.143	0	0	2	0.693
dari	0.2	0	0	0	1	1.386
china	0.2	0.143	0	0	2	0.693
punya	0	0.143	0	0.143	2	0.693
rezim	0	0.143	0	0	1	1.386
belum	0	0.143	0	0	1	1.386
lunas	0	0.143	0	0	1	1.386
ekonomi	0	0	0.250	0	1	1.386
indonesia	0	0	0.250	0.143	2	0.693
maju	0	0	0.250	0.143	2	0.693
bagai	0	0	0	0.143	1	1.386
negara	0	0	0	0.143	1	1.386
keren	0	0	0	0.143	1	1.386

Selanjutnya dilakukan pembobotan *TF-IDF* Tabel 7:

**Tabel 7.** Perhitungan pembobotan *TF-IDF*

Term	$TF - IDF = TF(w_i, d_j) \times IDF(N, df_i)$			
	D1	D2	D3	D4
uang	0.277	0	0	0
ikn	0	0	0	0
hutang	0.139	0.099	0	0
dari	0.277	0	0	0
china	0.139	0.099	0	0
punya	0	0.099	0	0.99
rezim	0	0.198	0	0
belum	0	0.198	0	0
lunas	0	0.198	0	0
ekonomi	0	0	0.347	0
indonesia	0	0	0.173	0.99
maju	0	0	0.173	0.99
bagai	0	0	0	0.198
negara	0	0	0	0.198
keren	0	0	0	0.198

Didapatkanlah Tabel 7, vektor dari dokumen 1 hingga 4 yaitu skor dari *TF-IDF* tersebut. Lalu data disimpan pada *database* dan akan digunakan ketika metode *K-Nearest Neighbors*.

### 3.5 Klasifikasi *K-Nearest Neighbors*

Setelah tahap Ekstraksi Fitur dengan *TF-IDF* dilakukan dan mendapatkan model vektor data latih, selanjutnya adalah tahap klasifikasi dengan algoritma *K-Nearest Neighbors* menggunakan perhitungan jarak *euclidean distance*. *Euclidean distance* berguna untuk mencari jarak antara data uji dengan semua data latih sehingga didapatkan jarak terdekatnya.

**Tabel 8.** Sampel Vektor

Data	Vektor
Data Latih 1	[0.277, 0, 0.139, 0.277, 0.139, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Data Latih 2	[0, 0, 0.099, 0, 0.099, 0.099, 0.198, 0.198, 0.198, 0, 0, 0, 0, 0, 0]
Data Latih 3	[0, 0, 0, 0, 0, 0, 0, 0, 0.347, 0.173, 0.173, 0, 0, 0, 0]
Data Latih 4	[0, 0, 0, 0, 0, 0.99, 0, 0, 0, 0, 0.99, 0.99, 0.198, 0.198, 0.198]
Data Uji	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0.173, 0.087, 0.087, 0, 0, 0.173]

Jarak antara data uji dengan data latih 1 dihitung dengan persamaan 3 berikut ini:

$$d(\text{uji}, \text{latih 1}) = \sqrt{(0-0.277)^2 + (0-0)^2 + (0-0.139)^2 + (0-0.277)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2}$$

$$d(uji, latih1) = \sqrt{0.076729 + 0 + 0.019321 + 0.07629 + 0.019321 + 0 + 0 + 0 + 0 + 0.29929 + 0.007569 + 0.007569 + 0 + 0 + 0.029929}$$

$$d(uji, latih1) = \sqrt{0.267096}$$

$$d(uji, latih1) = 0.517$$

Berikut Tabel 9 merupakan jarak data uji dengan masing-masing data latih:

**Tabel 9.** Hasil Perhitungan *Euclidean Distance*

	Jarak Data Uji	Label
Data Latih 1	0.517	<i>Hatespeech</i>
Data Latih 2	0.471	<i>Hatespeech</i>
Data Latih 3	0.274	<i>Non Hatespeech</i>
Data Latih 4	0.345	<i>Non Hatespeech</i>

Dalam perhitungan *Euclidean distance*, label dengan jarak terkecil menjadi *e* label. Jika *k* adalah 3, tiga jarak terdekat dipilih, dan jika dua di antaranya adalah *non-hatespeech*, maka data uji diprediksi sebagai *non-hatespeech*.

### 3.6 Pengujian

Pengujian Tabel 10 dan Tabel 11 mengevaluasi efektivitas algoritma *K-Nearest Neighbors* dengan menggunakan berbagai rasio data uji (50%, 40%, 30%, 20%, dan 10%) serta dengan atau tanpa *balance data*. *Confusion Matrix* digunakan untuk menganalisis kesalahan klasifikasi, sementara metrik seperti akurasi, *precision*, *recall*, dan *F1-score* menilai kinerja algoritma secara menyeluruh.

**Tabel 10.** *Confusion Matrix* Tanpa *Balance Data*

Jarak k	50:50				60:40				70:30				80:20				90:10			
	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN
3	472	65	65	298	415	90	13	202	301	87	11	141	183	37	24	37	101	22	4	53
5	446	60	91	303	419	107	9	185	303	97	9	131	183	42	24	111	99	22	6	53
7	510	85	27	278	420	116	8	176	306	111	6	117	182	38	25	115	97	25	8	50
9	525	118	12	245	424	127	4	165	306	117	6	111	196	42	11	111	102	28	3	47

**Tabel 11.** *Confusion Matrix* Dengan *Balance Data*

Jarak k	50:50				60:40				70:30				80:20				90:10			
	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN
3	465	46	79	310	352	24	92	252	208	30	103	199	165	19	35	141	90	10	17	63
5	428	43	116	313	344	24	100	252	146	8	165	221	122	12	78	148	73	9	34	64
7	421	44	123	312	358	28	86	248	254	22	57	207	89	6	111	154	51	9	56	64
9	458	47	86	309	352	30	92	246	235	13	91	169	137	10	63	150	71	10	36	63

Lalu setelah didapatkan skor *confusion matrix*, didapatkan nilai metrik Tabel 12 dan Tabel 13 dari performa *confusion matrix*.

**Tabel 12.** Hasil Performa *Confusion Matrix* Tanpa *Balance Data*

Jarak k	50:50				60:40				70:30				80:20				90:10			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
3	85%	88%	88%	88%	86%	82%	97%	89%	82%	78%	96%	86%	83%	83%	88%	86%	86%	82%	96%	89%
5	83%	88%	83%	86%	84%	80%	98%	88%	80%	76%	97%	85%	82%	81%	88%	85%	84%	82%	94%	88%
7	88%	86%	95%	90%	83%	78%	98%	87%	78%	73%	98%	84%	83%	83%	88%	85%	82%	80%	92%	85%
9	86%	82%	98%	89%	82%	77%	99%	87%	77%	72%	98%	83%	85%	82%	95%	88%	83%	78%	97%	87%

**Tabel 13.** Hasil Performa *Confusion Matrix* Dengan *Balance Data*

Jarak k	50:50				60:40				70:30				80:20				90:10			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
3	86%	91%	85%	88%	84%	94%	79%	86%	86%	75%	87%	67%	76%	85%	90%	83%	86%	85%	90%	84%
5	82%	91%	79%	84%	83%	93%	77%	85%	82%	68%	95%	47%	63%	75%	91%	61%	73%	76%	89%	68%
7	81%	91%	77%	83%	84%	93%	81%	86%	81%	85%	92%	82%	87%	67%	94%	45%	60%	64%	85%	48%
9	85%	91%	84%	87%	83%	92%	79%	85%	85%	81%	94%	72%	82%	80%	93%	69%	79%	74%	88%	66%

Perhitungan *confusion matrix* menunjukkan performa terbaik adalah Akurasi tertinggi 88% pada rasio 50:30 dengan k=3 tanpa *balance data*, *Precision* tertinggi 95% pada rasio 70:30 dengan k=5 dengan *balance data*, *Recall* tertinggi 99% pada rasio 60:40 dengan k=9 tanpa *balance data*, dan *F1-Score* tertinggi 95% pada rasio 50:30 dengan k=7 tanpa *balance data*.

#### 4. KESIMPULAN

Berdasarkan analisis klasifikasi dan deteksi pada 1800 *posts* yang relevan dengan Ibu Kota Nusantara (IKN) dari 24 Mei 2024 hingga 14 Juni 2024, hasil menunjukkan bahwa dari kelima rasio yang digunakan, empat nilai k yang ditentukan, serta dengan atau tanpa *balance data*, performa terbaik adalah sebagai berikut: Akurasi tertinggi 88% pada rasio 50:30 dengan k=3 tanpa *balance data*, *Precision* tertinggi 95% pada rasio 70:30 dengan k=5 dengan *balance data*, *Recall* tertinggi 99% pada rasio 60:40 dengan k=9 tanpa *balance data*, dan *F1-Score* tertinggi 95% pada rasio 50:30 dengan k=7 tanpa *balance data*. Pada metode *K-Nearest Neighbors*, penentuan nilai k sangat berpengaruh terhadap prediksi, sementara penentuan rasio split data juga mempengaruhi akurasi metode tersebut. Untuk meningkatkan kualitas dan akurasi sistem deteksi, beberapa langkah yang dapat diambil meliputi: memperbaiki tahap *preprocessing* dengan menambahkan langkah-langkah tambahan, menambah jumlah data, memasukkan media sosial selain X untuk memperluas jangkauan pengklasifikasian ujaran kebencian, dan berkolaborasi dengan ahli bahasa dan politik untuk meningkatkan kualitas label.

#### DAFTAR PUSTAKA

- [1] L. Suryani Widayati, "Ujaran Kebencian Batasan Pengertian Dan Larangannya," *Kajian Singkat Terhadap Isu Aktual Dan Strategis*, vol. x, no. 6, 2018.
- [2] H. Christianto, *Perbuatan Pidana Ujaran Kebencian Ragam Dan Studi Kasus*, Pertama. Yogyakarta: Graha Ilmu, 2018.
- [3] R. M. Yazid, F. R. Umbara, and P. N. Sabrina, "Deteksi Ujaran Kebencian dengan Metode Klasifikasi Naïve Bayes dan Metode N-Gram pada *Dataset* Multi-Label Twitter Berbahasa Indonesia," *Informatics and Digital Expert (INDEX)*, vol. 4, no. 2, pp. 46–52, 2023, doi: 10.36423/index.v4i2.894.
- [4] L. Handayani, "Implementasi K-Nearest Neighbor Dalam Mengklasifikasikan Hate Tweet K-Popers Pada Twitter," *Skripsi*, Universitas Islam Negeri Syarif Hidayatullah, Jakarta, 2023. [Online]. Available: <https://repository.uinjkt.ac.id/dspace/handle/123456789/73418>
- [5] D. Rifaldi, Abdul Fadlil, and Herman, "Teknik *Preprocessing* Pada Text Mining Menggunakan Data Tweet 'Mental Health,'" *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 3, no. 2, pp. 161–171, 2023, doi: 10.51454/decode.v3i2.131.
- [6] O. H. Rahman, A. Gunawan, and K. Agus, "Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 17–23, 2021, doi: 10.29207/resti.v5i1.2700.
- [7] R. Ilmi Hidayati, "Analisis *Imbalanced* Multiclass Pada Status Kepemilikan Asuransi Dengan Metode Multinomial Logistic Regression," Institut Teknologi Sepuluh Nopember, Surabaya, 2019. Accessed: Jul. 05, 2024. [Online]. Available: <http://repository.its.ac.id/id/eprint/64492>
- [8] M. F. Luthfi and K. M. Lhaksamana, "Implementation of TF-IDF Method and Support Vector Machine Algorithm for Job Applicants Text Classification," *Jurnal Media Informatika Budidarma*, vol. 4, no. 4, pp. 1181–1186, 2020, doi: <http://dx.doi.org/10.30865/mib.v4i4.2276>.
- [9] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.
- [10] R. K. Dinata, H. Akbar, and N. Hasdyna, "Algoritma K-Nearest Neighbor dengan *Euclidean* Distance dan Manhattan Distance untuk Klasifikasi Transportasi Bus," *ILKOM Jurnal Ilmiah*, vol. 12, no. 2, pp. 104–111, 2020, doi: 10.33096/ilkom.v12i2.539.104-111.
- [11] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan *Confusion Matrix* Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 5, no. 2, pp. 697–711, 2021, [Online]. Available: <https://ejournal.tunasbangsa.ac.id/index.php/jsakti/article/view/369>