

IMPLEMENTASI MAJORITY VOTE PADA METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE (STUDI KASUS : KENAIKAN PAJAK HIBURAN)

Bayu Samodera¹, Kartini², Muhammad Muharrom Al Haromainy³

^{1,2,3} Universitas Pembangunan Nasional Veteran Jawa Timur, Jl. Rungkut Madya No.1, Gn. Anyar, Kec. Gn. Anyar, Surabaya, Jawa Timur 60294, Telp. (031) 8706369

Received: 9 Juli 2024

Accepted: 31 Juli 2024

Published: 7 Agustus 2024

Keywords:

Majority Vote, Naive Bayes, Pajak Hiburan, Support Vector Machine

Corespondent Email:

kartini.if@upnjatim.ac.id

Abstrak. Pajak adalah kontribusi wajib yang diambil oleh pemerintah dari warga negara atau penduduk yang tinggal di dalam wilayah hukumnya. Salah satu jenis pajak yang menarik perhatian adalah pajak untuk hiburan. Kenaikan tarif pajak hiburan dapat memiliki dampak signifikan terhadap industri hiburan dan juga persepsi masyarakat terhadap kebijakan pemerintah. Penelitian ini bertujuan untuk menganalisis persepsi masyarakat terhadap kenaikan pajak hiburan kedalam kategori positif, netral, dan negatif dengan menggunakan data dari media sosial Twitter. Penelitian ini menggunakan metode Naive Bayes dan Support Vector Machine dan menggabungkannya melalui metode Majority Vote. Hasil Skenario pertama dengan menggunakan 90% data latih dan 10% data uji, Naive Bayes mencapai akurasi 71.82%, sementara SVM mencapai 78.11%, dan Majority Vote mendapat akurasi 80.55%. Lalu pada skenario kedua dengan membagi data menjadi 80% untuk pelatihan dan 20% untuk pengujian, Naive Bayes mencapai akurasi 70.32%, sementara SVM mencapai 74.35%, dan Majority Vote mendapat akurasi 77.80%. Terakhir skenario ketiga dengan membagi data menjadi 70% untuk pelatihan dan 30% untuk pengujian, Naive Bayes mencapai akurasi 68.16%, sementara SVM mencapai 71.18%, dan Majority Vote mendapat akurasi 75.50%. Hasil sentimen akhir dengan menggunakan Majority Vote menunjukkan bahwa mayoritas masyarakat menyikapi kenaikan pajak hiburan sebagai negatif.

Abstract. Tax is a mandatory contribution collected by the government from citizens or residents within its jurisdiction. One type of tax that is of concern is entertainment tax. An increase in entertainment tax rates could have a significant impact on the entertainment industry and also public perception of government policy. This research aims to analyze public sentiment towards the increase in entertainment tax into positive, neutral and negative categories using data from social media Twitter. This research applies the Naive Bayes and Support Vector Machine methods both separately and combined via Majority Vote. Results of the first scenario using 90% training data and 10% test data, Naive Bayes achieved 71.82% accuracy, while SVM achieved 78.11%, and Majority Vote achieved 80.55% accuracy. Then in the second scenario using 80% training data and 20% test data, Naive Bayes achieved 70.32% accuracy, while SVM achieved 74.35%, and Majority Vote achieved 77.80% accuracy. Finally, in the third scenario, using 70% training data and 30% test data, Naive Bayes achieved 68.16% accuracy, while SVM achieved 71.18%, and Majority Vote achieved 75.50% accuracy. The final sentiment results using Majority Vote show that the majority of people view the increase in entertainment tax as negative.

1. PENDAHULUAN

Pajak merupakan salah satu sumber pendapatan utama negara selain dari sektor migas dan nonmigas [1]. Pajak berperan sebagai alat fiskal yang digunakan oleh pemerintah untuk menghimpun dana guna pembiayaan berbagai program pembangunan dan kebijakan. Di Indonesia, sistem perpajakan terus mengalami penyesuaian sebagai respons terhadap dinamika ekonomi dan kebutuhan keuangan negara. memainkan peran penting dalam pembangunan karena berfungsi sebagai sumber keuangan untuk mendanai pengeluaran negara, termasuk biaya pembangunan [2].

Perubahan peraturan pajak di Indonesia sering kali dipengaruhi oleh kondisi ekonomi dan kebutuhan negara. Salah satu kebijakan yang menjadi perhatian adalah kenaikan pajak hiburan, yang dikenakan pada kegiatan yang bersifat menghibur seperti tontonan, pertunjukan, permainan, atau acara yang menarik kerumunan orang [3]. Kenaikan pajak ini diusulkan sebagai sumber tambahan pendapatan daerah dan sebagai langkah strategis untuk mengatasi citra negatif tempat hiburan.

Untuk memahami sentimen masyarakat terhadap kenaikan pajak hiburan, analisis data dari platform media sosial seperti *twitter* menjadi penting. *Twitter* adalah platform populer yang digunakan masyarakat untuk mengekspresikan pandangan mereka terhadap kebijakan pemerintah [4].

Penelitian ini menerapkan metode *Naive Bayes* dan *Support Vector Machine*, dan dikombinasikan dengan metode *Majority Vote*, untuk mengklasifikasikan sentimen masyarakat terhadap kenaikan pajak hiburan. *Naive Bayes* adalah metode klasifikasi yang sederhana dan efektif, yang didasarkan pada prinsip probabilitas [5], sedangkan *Support Vector Machine* menggunakan konsep hyperplane untuk memisahkan kelas data secara optimal [6]. Kombinasi kedua metode dengan menggunakan *Majority Vote* diharapkan memberikan hasil klasifikasi yang lebih akurat dan dapat diandalkan [7].

Penelitian ini bertujuan untuk mengevaluasi opini publik mengenai kenaikan pajak hiburan dengan menggunakan data dari platform *twitter* dan membandingkan akurasi metode *Naive Bayes*, *Support Vector Machine*, dan *Majority Vote* dalam klasifikasi sentimen,

serta memahami dan menganalisis sentimen yang diungkapkan oleh masyarakat di media sosial *twitter* terkait dengan kenaikan pajak hiburan di Indonesia.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian yang menjadi rujukan dalam kajian ini adalah penelitian yang telah dilakukan oleh Mochammad Alfi Rizky Reynaldhi dan Yuliant Sibaroni dengan judul penelitian “Analisis Sentimen Review Film pada Twitter menggunakan Metode Klasifikasi Hybrid Support Vector Machine, Naïve Bayes, dan Decision Tree”. Pendekatan klasifikasi ini menggabungkan algoritma *Support Vector Machine*, *Naïve Bayes*, dan *Decision Tree* untuk mengelola *volume* data besar dengan tingkat akurasi yang tinggi. Ketiga metode ini digabungkan melalui teknik *ensemble voting*. Evaluasi penelitian menunjukkan bahwa model ini mencapai akurasi sebesar 0.8192 dan F1-score sebesar 0.78 [8].

Penelitian lainnya dilakukan oleh I Putu Ananda Miarta Utama, dkk. Dengan judul “Multi-Aspect Sentiment Analysis Hotel Review Using RF, SVM, and Naïve Bayes based Hybrid Classifier”. Penelitian ini mengimplementasikan klasifikasi *hybrid* dengan menerapkan metode *Random Forest*, *Support Vector Machine*, dan *Naive Bayes* dalam sebuah *Hybrid Classifier* menggunakan teknik *ensemble voting*, didapatkan hasil rata-rata akurasi 84%. *Naïve Bayes* mencapai akurasi rata-rata 82.4%, sedangkan *Random Forest* dan *Support Vector Machine* masing-masing mencatatkan akurasi rata-rata 82.2% dan 81% [9].

2.2 Analisis Sentimen

Analisis sentimen merupakan metode untuk mengolah teks guna memperoleh informasi mengenai evaluasi yang positif, netral, atau negatif. Teknik ini digunakan oleh pengguna internet di platform media sosial untuk menyampaikan pendapat atau penilaian pribadi terhadap suatu peristiwa atau fenomena [10].

2.3 Twitter

Twitter adalah platform media sosial dimana pengguna dapat membagikan pesan pendek yang disebut "tweet". Saat ini, kemajuan teknologi di Indonesia telah mempermudah akses masyarakat terhadap layanan media sosial. Hal ini menjadikan platform-platform tersebut lebih terbuka dan mudah dijangkau oleh pengguna [11]. *Twitter* memiliki kelebihan sebagai sarana untuk menyebarkan informasi, memfasilitasi pembelajaran, meningkatkan aksesibilitas komunikasi, memotivasi, meningkatkan keterlibatan, berperan sebagai alat komunikasi, serta membangun komunitas [12]. *Twitter* sering digunakan sebagai sumber data untuk text mining. Hal ini disebabkan oleh kemampuan *twitter* untuk melakukan pencarian tidak hanya berdasarkan nama akun, tetapi juga berdasarkan isi teks dalam status.

2.4 Naïve Bayes

Naive Bayes adalah metode klasifikasi yang sederhana dan efektif. Metode ini menggunakan teorema probabilitas untuk menentukan peluang terbaik, dengan memperkirakan probabilitas masa depan berdasarkan informasi yang telah tersedia di masa lalu [13]. Klasifikasi *Bayes* sering diandalkan karena dapat menghasilkan hasil akurasi yang baik dengan waktu yang bersaing apabila diberi dataset yang banyak. Teorema dasar metode *Bayes* dapat dinotasikan pada persamaan berikut:

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad (2.1)$$

Keterangan:

- X : Data fitur yang akan diprediksi kelasnya
- H : Hipotesis data merupakan suatu kelas spesifik.
- $P(H|X)$: *posterior* probabilitas
- $P(H)$: *prior* probabilitas
- $P(X|H)$: *Likelihood*
- $P(X)$: Probabilitas X

Berikut adalah persamaan yang dapat digunakan untuk menghitung *prior* dalam proses klasifikasi.

$$\begin{aligned} P(y=\text{positif}) &= \frac{N_p}{N} \\ P(y=\text{negatif}) &= \frac{N_n}{N} \end{aligned} \quad (2.2)$$

Keterangan :

- N_p : Jumlah data dalam dataset kelas positif.
- N_n : Jumlah data dalam dataset kelas negatif.
- N : Total jumlah data dalam dataset.

Proses selanjutnya menghitung probabilitas *likelihood* dapat dilakukan dengan menggunakan persamaan berikut.

$$P(x|y = \text{positif/negatif}) = \frac{N_{word}}{N_{total word} + V} \quad (2.3)$$

Keterangan :

- N_{word} : Jumlah kemunculan kata tertentu x dalam kelas y.
- $N_{total word}$: jumlah total kata dalam kelas y.
- V : jumlah total seluruh kata dalam data uji.

Proses terakhir menghitung probabilitas *posterior* dapat dilakukan dengan menggunakan persamaan berikut.

$$P(y = c|x) = P(x_1|y = c) \times P(x_2|y = c) \times \dots \times P(x_n|y = c) \times P(y = c) \quad (2.4)$$

Keterangan :

- $P(y = c|x)$: Probabilitas *posterior* dari kelas c setelah mengamati fitur x (kata-kata dalam kalimat uji).
- $P(x_1|y = c)$: Probabilitas bahwa fitur x1 muncul dalam kelas c (probabilitas likelihood).
- $P(y = c)$: Probabilitas *prior* dari kelas c.

2.5 Support Vector Machine

Support Vector Machine merupakan sebuah metode yang membandingkan suatu seleksi parameter standart nilai diskrit yang disebut kandidat set [14]. Metode ini dipilih karena telah berhasil diterapkan dalam

beberapa penelitian sebelumnya. *Support Vector Machine* menciptakan *hyperplane* optimal yang memisahkan kedua kategori. Pada penelitian ini, kernel yang dipilih adalah kernel polinomial. *Support Vector Machine kernel Polynominal* dapat dinotasikan pada persamaan berikut:

$$K(x_i, x_j) = (\gamma (x_i, x_j) + r)^d \quad (2.5)$$

Keterangan:

- x_i, x_j : Vektor fitur dari dua sampel data yang ingin dibandingkan.
- γ : Parameter *Kernel*.
- r : Probabilitas hipotesis H berdasar kondisi X (*posterior* probabilitas)
- d : derajat dari polinomial.

2.6 Ensemble Majority Voting

Metode *ensemble* merupakan teknik pembelajaran yang menggabungkan beberapa model dasar untuk menghasilkan satu model prediksi yang lebih optimal. Inti dari pendekatan ini adalah menggabungkan kekuatan beberapa model untuk meningkatkan akurasi prediksi secara keseluruhan, dengan tujuan mendapatkan satu model yang paling efektif dalam memprediksi hasil akhir [8]. *Ensemble learning* memanfaatkan kekuatan gabungan dari berbagai model untuk menghasilkan prediksi yang lebih akurat dan stabil daripada menggunakan metode klasifikasi tunggal. Teknik *majority voting* adalah salah satu metode *ensemble learning* yang umum digunakan, di mana keputusan akhir diambil berdasarkan mayoritas suara dari semua model yang terlibat.

2.7 Evaluasi Akhir

Evaluasi performa model klasifikasi sentimen dilakukan menggunakan *Confusion Matrix*. *Confusion matrix* adalah tabel yang menunjukkan perbandingan antara prediksi yang benar dan yang salah yang dihasilkan oleh model klasifikasi [15]. *Confusion matrix* mencakup empat istilah yang merepresentasikan hasil proses klasifikasi: *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Di bawah ini adalah contoh tabel *confusion matrix*.

Tabel 1. Confusion Matrix

		Kelas Prediksi	
		1	0
Kelas	1	TP	FN
	0	FP	TN

Selanjutnya rumus persamaan *confusion matrix* untuk menghitung *accuracy*, *precision*, *recall*, dan *F1-Score* adalah seperti berikut.

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

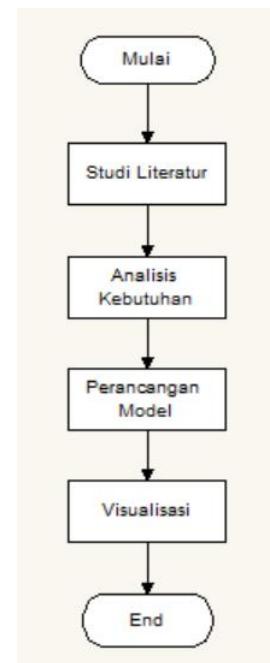
$$\text{Precision} : \frac{\text{TP}}{\text{TP}+\text{FP}} \times 100\%$$

$$\text{Recall} : \frac{\text{TP}}{\text{TP}+\text{FN}} \times 100\%$$

$$\text{F1-Score} : \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision}+\text{Recall})} \times 100\%$$

3. METODE PENELITIAN

Penelitian ini terfokus pada studi kasus tertentu dan dilakukan melalui beberapa tahapan. Berikut adalah diagram alur yang menggambarkan proses perancangan model.



Gambar 1. Alur Metode Penelitian

a. Studi Literatur

Studi literatur merupakan langkah penelitian yang mencakup pengumpulan, pemilihan, analisis, serta sintesis materi-materi literatur yang relevan dengan topik atau permasalahan penelitian yang diteliti.

b. Analisis Kebutuhan

Pada bagian ini, akan dibahas dua aspek utama agar tujuan penelitian ini dapat tercapai, yaitu kebutuhan data dan kebutuhan *hardware* dan *software*.

c. Perancangan Model

Dalam merancang model pembangunan, berbagai aspek harus dipertimbangkan untuk memastikan model yang dihasilkan mampu memberikan klasifikasi yang akurat dan dapat dipercaya. Beberapa tahap yang ada didalam perancangan model diantaranya pengumpulan data, penyaringan data, *pre-processing* data, pelabelan data, ekstraksi fitur TF-IDF, klasifikasi, dan evaluasi hasil.

d. Visualisasi

Visualisasi digunakan untuk memberikan pemahaman yang lebih intuitif terhadap performa model. Visualisasi dapat dilakukan dengan menggunakan berbagai jenis grafik dan diagram. Salah satu bentuk visualisasi yang umum digunakan adalah *word cloud*.

4. HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Pengumpulan data diambil dari media sosial *twitter* dengan menggunakan *Tweets Harvest*. Penelitian ini menggunakan data dari platform twitter dengan melakukan pencarian menggunakan kata kunci “pajak hiburan”. *Tweet* yang dikumpulkan berupa *tweet* berbahasa Indonesia. Total dari hasil pengumpulan data menggunakan kata kunci tersebut pada rentang tanggal 1 Januari 2024 – 2 Mei 2024 didapat sebanyak 4021 data tweets. Data hasil pengambilan menggunakan *Tweets Harvest* kemudian disimpan ke dalam file dengan format CSV. Berikut merupakan gambar hasil pengumpulan data dengan *Tweets Harvest*.

id_str	tweet
1,75E+18	Pajak ini merupakan pajak daerah yang menjadi sum
1,75E+18	Hai, #KawanPajak dan #MitraDJPK ternyata banyak ji
1,75E+18	Polemik Kenaikan Pajak Hiburan 40-75 Persen, Inul D
1,75E+18	HORE !! Sandiaga Uno Pastikan Kenaikan Tarif Pajak
1,75E+18	@YRadianto Maaf ikutan nambah dikit, Visi misi : D
1,75E+18	@andthenudo Betul... Pasti dia jd mikir pajak hibura
1,75E+18	Sandiaga Uno Minta Para Kepala Daerah Beri Insenti
1,75E+18	@andthenudo Gara-gara pajak tempat hiburan naik
1,75E+18	Pajak hiburan merupakan pajak daerah! Pemungutan

Gambar 2. Hasil Pengumpulan Data

4.2 Penyaringan Data

Penyaringan data dilakukan untuk menghapus data yang tidak relevan atau tidak diperlukan dari kumpulan data. Proses penyaringan data dilakukan secara manual dengan mencari data *tweets* yang mengandung kata kunci “pajak hiburan”. Setelah dilakukan penyaringan data, didapat data sebanyak 4012 *tweets*. Selanjutnya dilakukan penghapusan kolom yang tidak dibutuhkan dan hanya mengambil kolom *tweet*.

full_text
Seperinya tidak serta merta 13 juta menanggung seluruhnya. Jangan lupa masih ada PPN pajak kendaraan pajak bumi bi
lagi ada ops biar gaslighting pembayar pajak sama non pembayar pajak. This what makes bapak cengegesan itu different
balance brow, apalagi bisnis hiburan pajak daerahnya gede, jadi Pemda untung
@Faraz_kepri @Aditakap @inorii_yuzuriha Pajak hiburan 30% hoax.. Taunya pemda yg ambil
Di Venom ada pajak hiburan gak ya
Komisi III DPRD Medan Kritik Bapenda PAD dari Pajak Tempat Hiburan Belum Maksimal #BNetwork https://t.co/5dA1f1t
@Hasbil_Lbs @AgusMagelangan Yg gk paham itu anda. Pejabat pakai duit pajak rakyat. Dunia hiburan gk pakai. Wkwkwk
Fedi Nuril pakai hak dia sebagai rakyat yang bayar pajak Bung. Politisi kalau sesuai kapasitas mau 'ikut campur' dunia hibi
@Hasbil_Lbs Lho politik itu mayoritas kegiatannya dibayai dari uang rakyat ngurusi hajat hidup orang banyak masa uda
@panjiracob sih bioskop2 begitu + pajak hiburan dsb nya.

Gambar 3. Hasil Penyaringan Data

4.3 Pre-Processing Data

Pre-processing bertujuan untuk membersihkan, mengubah, atau mengatur ulang data agar sesuai dengan kebutuhan analisis. Terdapat beberapa langkah yang dilakukan dalam proses *pre-processing*, yaitu:

a. Case Folding

Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil. Proses ini bertujuan untuk memastikan keseragaman dalam analisis teks dengan mengabaikan variasi huruf besar dan kecil. Berikut adalah gambar hasil *case folding*.

0	sepertinya tidak serta merta 13 juta menanggun...
1	lagi ada ops biar gaslighting pembayar pajak s...
2	balance brow, apalagi bisnis hiburan pajak dae...
3	@faraz_kepri @Aditakap @inorii_yuzuriha pajak ...
4	di venom ada pajak hiburan gak ya
	...
4007	horang kaya teriak pajak hiburan dinaikan jadi...
4008	@vivacoid gabung pajak hiburan minta turun...
4009	pemerintah menaikkan pajak hiburan termasuk di...
4010	@ganjarpranowo pak ganjar tolong turunin pajak...
4011	pemprov bali akan kaji ulang soal spa yang mas...

Gambar 4. Hasil Case Folding

b. Cleansing

Tujuan dari *cleansing* adalah untuk menyederhanakan teks dan menghilangkan elemen yang tidak memberikan informasi penting. Langkah-langkah dalam proses pembersihan data atau *cleaning* mencakup beberapa tahap yang melibatkan penanganan singkatan, penghapusan tanda baca, eliminasi angka, penghapusan *URL*, penghilangan nama pengguna atau *mention*, eliminasi tagar atau *hashtag*, serta penghapusan huruf tunggal. Berikut adalah gambar hasil *cleansing*.

```

0  sepertinya tidak serta merta juta menanggung s...
1  lagi ada ops biar gaslighting pembayar pajak s...
2  balance brow apalagi bisnis hiburan pajak daer...
3  pajak hiburan hoax taunya pemda yg ambil
4  di venom ada pajak hiburan gak ya
...
4007 horang kaya teriak pajak hiburan dinaiakan jadi...
4008 gabung pajak hiburan minta turun
4009 pemerintah menaikkan pajak hiburan termasuk di...
4010 pak ganjar tolong turunin pajak satu isu yg pa...
4011 pemprov bali akan kaji ulang soal spa yang mas...

```

Gambar 5. Hasil *Cleansing*

c. Tokenizing

Tokenizing adalah proses memecah teks menjadi token atau kata-kata individual. Setiap kata atau frasa menjadi entitas terpisah yang dapat diolah lebih lanjut. Hasilnya adalah daftar token yang merepresentasikan kata-kata dalam teks tersebut. Setiap kata diidentifikasi dengan tanda kutip dan dipisahkan dengan tanda koma. Berikut adalah gambar hasil *tokenizing*.

```

0  ['sepertinya', 'tidak', 'serta', 'merta', 'jut...
1  ['lagi', 'ada', 'ops', 'biar', 'gaslighting', ...
2  ['balance', 'brow', 'apalagi', 'bisnis', 'hibu...
3  ['pajak', 'hiburan', 'hoax', 'taunya', 'pemda'...
4  ['di', 'venom', 'ada', 'pajak', 'hiburan', 'ga...
...
4007 ['horang', 'kaya', 'teriak', 'pajak', 'hiburan...
4008 ['gabung', 'pajak', 'hiburan', 'minta', 'turun']
4009 ['pemerintah', 'menaikkan', 'pajak', 'hiburan'...
4010 ['pak', 'ganjar', 'tolong', 'turunin', 'pajak'...
4011 ['pemprov', 'bali', 'akan', 'kaji', 'ulang', '...

```

Gambar 6. Hasil *Tokenizing*

d. Normalisasi

Proses normalisasi mengubah bahasa tidak baku menjadi baku. Normalisasi dilakukan menggunakan daftar kata baku atau kata dasar yang terdapat pada kamus-alay.csv dari nasalsabila. Proses ini bertujuan untuk menggantikan kata-kata alay atau bahasa gaul dengan bentuk formal atau standar dari kata tersebut. Berikut adalah gambar hasil normalisasi.

```

0  ['sepertinya', 'tidak', 'serta', 'merta', 'juta', 'menan...
1  ['lagi', 'ada', 'ops', 'biar', 'gaslighting', 'pembayar', ...
2  ['balance', 'brow', 'apalagi', 'bisnis', 'hiburan', 'paj...
3  ['pajak', 'hiburan', 'hoax', 'taunya', 'pemda', 'yang', 'a...
4  ['di', 'venom', 'ada', 'pajak', 'hiburan', 'enggak', 'ya']
...
4007 ['horang', 'kayak', 'teriak', 'pajak', 'hiburan', 'dinai...
4008 ['gabung', 'pajak', 'hiburan', 'meminta', 'turun']
4009 ['pemerintah', 'menaikkan', 'pajak', 'hiburan', 'termasuk'...
4010 ['pak', 'ganjar', 'tolong', 'turunin', 'pajak', 'satu', 'i...
4011 ['pemprov', 'bali', 'akan', 'kaji', 'ulang', 'soal', 'spa', ...

```

Gambar 7. Hasil Normalisasi

e. Stopword Removal

Tahap ini penting untuk mengeluarkan kata-kata umum yang sering muncul tetapi kurang memberikan informasi penting dalam analisis teks, seperti "yang", "dan", atau "atau". Dalam proses menggunakan *stopwords* bahasa Indonesia dari *library* NLTK. Berikut adalah gambar hasil *stopword removal*.

```

0  [' ', ' ', ' ', ' ', 'merta', 'juta', '...
1  [' ', ' ', ' ', 'ops', 'biar', 'gaslight...
2  [' ', 'balance', 'brow', ' ', 'bisnis'...
3  [' ', 'pajak', 'hiburan', 'hoax', 'tau...
4  [' ', 'venom', 'ada', 'pajak', 'hibur...
...
4007 [' ', 'horang', 'kayak', 'teriak', 'pa...
4008 [' ', 'gabung', 'pajak', 'hiburan', '...
4009 [' ', 'pemerintah', 'menaikkan', 'pajak',...
4010 [' ', 'ganjar', 'tolong', 'turunin', '...
4011 [' ', 'pemprov', 'bali', 'akan', 'kaji',...

```

Gambar 8. Hasil *Stopword Removal*

f. Stemming

Tahap ini menghapus akhiran kata sehingga kata-kata dengan akhiran yang sama dianggap setara. Contohnya, kata-kata seperti "berlari", "berlari-lari", dan "berlari-lah" akan diganti menjadi bentuk dasar "lari". Proses *stemming* menggunakan metode Nazief and Adriani. Berikut adalah gambar hasil *Stemming*.

```

0  merta juta tanggung lupa ppn pajak kendara paj...
1  ops biar gaslighting bayar pajak non bayar paj...
2  balance brow bisnis hibur pajak daerah gede pe...
3  pajak hibur hoax tau pemda ambil
4  venom pajak hibur ya
...
4007 horang kayak teriak pajak hibur naik kemarin d...
4008 gabung pajak hibur turun
4009 perintah naik pajak hibur diskotik bar karaoke...
4010 ganjar tolong turunin pajak isu pilih presiden...
4011 pemprov bal kaji ulang spa masuk kategori hibu...

```

Gambar 9. Hasil *Stemming*

4.4 Pelabelan Data

Proses labelisasi data pada penelitian ini dilakukan menggunakan *InSet* atau *Indonesian Lexicon of Sentiment*. *InSet* yang digunakan dalam penelitian ini berasal dari studi sebelumnya yang dilakukan oleh Fajri Koto dan Gemala Y. *InSet* terdiri dari 3.609 kata positif dan 6.609 kata negatif. Jika jumlah kata positif

lebih banyak daripada kata negatif, teks akan diklasifikasikan sebagai "Positif". Sebaliknya, jika jumlah kata negatif lebih banyak, teks akan diklasifikasikan sebagai "Negatif". Jika keduanya sama banyak, teks akan diberi label "Netral". Berikut adalah gambar hasil dari pelabelan data.

Jumlah Positif:	830
Jumlah Netral:	837
Jumlah Negatif:	2345
Total Data:	4012
nazief_adriani_stemming sentimen	
0	merta juta tanggung lupa ppn pajak kendara paj... Negatif
1	ops biar gaslighting bayar pajak non bayar paj... Negatif
2	balance brow bisnis hibur pajak daerah gede pe... Netral
3	pajak hibur hoax tau pemda ambil Negatif
4	venom pajak hibur ya Netral

Gambar 10. Hasil Pelabelan Data

4.5 Ekstrasi Fitur TF-IDF

Ekstrasi Fitur *TF-IDF* dilakukan untuk menilai pentingnya bobot sebuah kata dalam sebuah dokumen terkait dengan kumpulan dokumen lainnya. Berikut adalah adalah implementasi proses dalam menghitung *TF*, *IDF*, dan *TF-IDF*.

a. TF

TF (*Term Frequency*) adalah metode untuk mengukur frekuensi kemunculan suatu kata dalam sebuah dokumen. Berikut adalah adalah implementasi kode pada proses menghitung *TF*.

calc Term Frequency		
1	def calc_TF(document):	
2	# Menghitung berapa kali kata muncul dalam ulasan	
3	TF_dict = {}	
4	for term in document:	
5	if term in TF_dict:	
6	TF_dict[term] += 1	
7	else:	
8	TF_dict[term] = 1	
9	# Menghitung TF untuk setiap kata	
10	for term in TF_dict:	
11	TF_dict[term] = TF_dict[term] / len(document)	
12	return TF_dict	
13		
14	# Menerapkan fungsi calc_TF	
15	data["TF dict"] = data["tweet list"].apply(calc_TF)	

Gambar 11. Source Code Term Frequency

Berikut adalah gambar hasil dari proses menghitung *Term Frequency*.

0	{'merta': 0.0666666666666667, 'juta': 0.06666...
1	{'ops': 0.0303030303030304, 'bian': 0.03030...
2	{'balance': 0.1111111111111111, 'brow': 0.1111...
3	{'pajak': 0.1666666666666666, 'hibur': 0.1666...
4	{'venom': 0.25, 'pajak': 0.25, 'hibur': 0.25, ...

Gambar 12. Hasil Term Frequency

b. IDF

IDF (*Inverse Document Frequency*) mengukur seberapa penting suatu kata

dengan memperhitungkan seberapa sering kata muncul di seluruh koleksi dokumen, bukan hanya di satu dokumen. Berikut adalah adalah implementasi kode pada proses menghitung *IDF*.

calc Inverse Document Frequency	
1	import math
2	
3	# Menghitung IDF untuk setiap term
4	def calculate_IDF(DF, total_docs):
5	idf_dict = {}
6	for term, freq in DF.items():
7	idf_dict[term] = math.log(total_docs / (freq + 1))
8	return idf_dict
9	
10	# Menghitung DF untuk setiap term
11	def calc_DF(tfDict):
12	count_DF = {}
13	for document in tfDict:
14	for term in document:
15	if term in count_DF:
16	count_DF[term] += 1
17	else:
18	count_DF[term] = 1
19	return count_DF
20	
21	DF = calc_DF(data["TF dict"])
22	idf_dict = calculate_IDF(DF, len(data))
23	
24	# Konversi kamus IDF menjadi DataFrame
25	idf_df = pd.DataFrame(idf_dict.items(), columns=['Term',
26	'IDF'])

Gambar 13. Source Code Inverse Document Frequency

Berikut adalah gambar hasil dari proses menghitung *Inverse Document Frequency*.

⤤	Hasil IDF untuk setiap term:
0	Term IDF
0	merta 7.603898
1	juta 5.118991
2	tanggung 5.657988
3	lupa 5.252523
4	ppn 4.405225
...	...
5690	hambur 7.603898
5691	anjgggggggggg 7.603898
5692	horang 7.603898
5693	sengsaraayo 7.603898
5694	sarana 7.603898

[5695 rows x 2 columns]

Gambar 14. Hasil Inverse Document Frequency

c. TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) mengukur pentingnya sebuah term dalam sebuah dokumen dalam konteks keseluruhan koleksi dokumen. Berikut adalah adalah implementasi kode pada proses menghitung *TF-IDF*.

```

calc Term Frequency - Inverse Document Frequency

1 #calc TF-IDF
2 def calc_TF_IDF(TF):
3     TF_IDF_Dict = {}
4     #For each word in the review, we multiply its tf and its
5     idf.
6     for key in TF:
7         TF_IDF_Dict[key] = TF[key] * IDF[key]
8     return TF_IDF_Dict
9
10 #Stores the TF-IDF Series
11 data["TF-IDF_dict"] = data["TF_dict"].apply(calc_TF_IDF)
12
13 # Membuat DataFrame dari data
14 df = pd.DataFrame({
15     "Term": list(data["TF_dict"][index].keys()),
16     "TF": list(data["TF_dict"][index].values()),
17     "IDF": [idf_dict[term] for term in
18             data["TF_dict"][index]],
19     "TF-IDF": list(data["TF-IDF_dict"][index].values())
20 })

```

Gambar 15. *Source Code TF-IDF*

Berikut adalah gambar hasil dari proses menghitung *TF-IDF*.

	Term	TF	IDF	TF-IDF
0	bahagia	0.10	6.099821	0.609982
1	kena	0.10	2.885399	0.288540
2	pajak	0.10	0.018617	0.001862
3	hibur	0.10	0.020905	0.002090
4	bayar	0.10	3.578546	0.357855
5	malam	0.05	3.362571	0.168129
6	rp	0.10	5.899150	0.589915
7	juta	0.15	5.118991	0.767849
8	ya	0.10	2.603313	0.260331
9	rupiah	0.05	6.687607	0.334380
10	total	0.05	6.505286	0.325264

Gambar 16. Hasil *TF-IDF*

4.6 Klasifikasi Naive Bayes

Proses klasifikasi akan dilakukan sebanyak tiga kali menggunakan perbandingan yang berbeda antara data latih dan data uji. Pertama, dengan perbandingan 90% data latih dan 10% data uji. Kedua, dengan perbandingan 80% data latih dan 20% data uji. Ketiga, dengan perbandingan 70% data latih dan 30% data uji. Penggunaan tiga skenario berbeda bertujuan untuk mencari tingkat akurasi tertinggi. Berikut adalah implementasi kode pada proses klasifikasi menggunakan *Multinomial Naive Bayes*.

```

47 def predict_naive_bayes(X_test, class_priors, likelihoods):
48     predictions = []
49     for text in X_test:
50         log_probs = defaultdict(float)
51         for cls, prior in class_priors.items():
52             log_prob = np.log(prior)
53             for word, count in text.items():
54                 if word in likelihoods[cls]:
55                     log_prob += log_likelihoods[cls][word] * count
56             log_probs[cls] = log_prob
57         predicted_class = max(log_probs, key=log_probs.get)
58         predictions.append(predicted_class)
59     return predictions
60
61
62 # Train Naive Bayes
63 X_train_counts, nb_vocabulary = vectorize_text(X_train_texts)
64 X_test_counts, _ = vectorize_text(X_test_texts)
65 class_priors, likelihoods = fit_naive_bayes(X_train_counts,
66 y_train, nb_vocabulary)
67
68 # Make Predictions
69 nb_predictions = predict_naive_bayes(X_test_counts,
70 class_priors, likelihoods)

```

Gambar 17. Source Code Multinomial Naïve Bayes

Hasil dari penerapan kode diatas dengan menggunakan 3 skenario berbeda dapat dilihat pada tabel 2 dibawah ini.

Tabel 2. Hasil Akurasi *Naive Bayes*

Pembagian Data	Accuracy	F-1 Score
90% data latih 10% data uji	71.82%	57%
80% data latih 20% data uji	70.32%	54%
70% data latih 30% data uji	68.16%	51%

Berdasarkan hasil dari penggunaan ketiga data *train* dan *test* yang berbeda pada pengujian *Multinomial Naive Bayes*, tingkat akurasi tertinggi terjadi saat menggunakan pembandingan 90% data latih dan 10% data uji. Hal ini menunjukkan bahwa model yang dilatih dengan proporsi data *train* yang lebih besar cenderung memberikan kinerja yang lebih baik saat diuji pada data *test* yang terpisah.

4.7 Klasifikasi *Support Vector Machine*

Klasifikasi juga akan dilakukan sebanyak tiga kali menggunakan perbandingan yang berbeda antara data latih dan data uji. Pertama, dengan perbandingan 90% data latih dan 10% data uji. Kedua, dengan perbandingan 80% data latih dan 20% data uji. Ketiga, dengan perbandingan 70% data latih dan 30% data uji. Proses klasifikasi ini menggunakan *kernel polynomial* dengan parameter yaitu $\gamma = 0,001$, $\lambda = 0,5$, $C = 1$, $d = 2$, iterasi maksimum = 10. Berikut adalah implementasi kode pada proses klasifikasi menggunakan *Support Vector Machine*.

```

SVM using kernel polynomial

1 # Polynomial kernel function
2 def polynomial_kernel(x, y, d=2, coef=1):
3     return (np.dot(x, y) + coef) ** d
4
5 # Train SVM
6 def train_svm(X, y, C=1, max_iter=10, epsilon=0.0001,
7 kernel=polynomial_kernel):
8     n = len(y)
9     alpha = np.zeros(n)
10    b = 0
11
12    for _ in range(max_iter):
13        for i in range(n):
14            xi = X[i]
15            yi = y[i]
16            ui = sum(alpha[j] * y[j] * kernel(xi, X[j]) for j in range(n))
17
18            Ei = ui - yi
19
20            if (yi * Ei < -epsilon and alpha[i] < C) or (yi * Ei > epsilon
21 and alpha[i] > 0):
22                j = np.random.choice([l for l in range(n) if l != i])
23
24                xj = X[j]
25                yj = y[j]
26                uj = sum(alpha[k] * y[k] * kernel(xj, X[k]) for k in
27 range(n)) + b
28
29                Ej = uj - yj
30
31                alpha[i] -= Ej / (C * len(X))
32                alpha[j] += Ej / (C * len(X))
33
34                b -= Ej / len(X)

```

Gambar 18. Source Code SVM Kernel Polynomial

Hasil dari penerapan kode diatas dengan menggunakan 3 skenario berbeda dapat dilihat pada tabel 3 dibawah ini.

Tabel 3. Hasil Akurasi SVM

Pembagian Data	Accuracy	F1 Score
90% data latih 10% data uji	78.11%	72%
80% data latih 20% data uji	74.35%	64%
70% data latih 30% data uji	71.18%	64%

Berdasarkan hasil dari penggunaan ketiga data *train* dan *test* yang berbeda pada pengujian *Support Vector Machine* dengan menggunakan *kernel polynomial*, tingkat akurasi tertinggi terjadi saat menggunakan pembagian 90% data latih dan 10% data uji. Peningkatan kinerja pada data uji terpisah cenderung terjadi saat model dilatih dengan proporsi data latih yang lebih besar.

4.7 Klasifikasi Majority Voting

Setelah menghitung kedua klasifikasi dengan menggunakan metode *Multinomial Naive Bayes* dan *Support Vector Machine*, langkah selanjutnya adalah menentukan hasil akhir dengan menggabungkan kedua metode melalui teknik *majority voting*. Dalam teknik *majority voting*, label kelas dari sebuah data uji ditentukan berdasarkan label yang paling sering diprediksi oleh semua model yang digunakan. Berikut adalah implementasi kode pada proses *majority voting*.

```
Majority vote (accuracy)
1 # Ensemble Voting Classifier
2 voting_clf =
3     VotingClassifier(estimators=[('mnb',
4         mnb.BestEstimator_), ('svm',
5             svm.BestEstimator_)],
6         voting='hard')
7 voting_clf.fit(X_train_vec, y_train)
8 ensemble_pred = voting_clf.predict(X_test_vec)
9
10 # Evaluation
11 accuracy = accuracy_score(y_test, ensemble_pred)
12 print(f'Ensemble Majority Vote Accuracy: {accuracy}')
```

Gambar 19. Source Code Majority Voting

Hasil dari penerapan kode diatas dengan menggabungkan kedua metode menggunakan 3 skenario berbeda dapat dilihat pada tabel dibawah ini.

Tabel 4. Hasil Akurasi *Majority Vote*

Pembagian Data	Accuracy	F1 Score
90% data latih 10% data uji	80.55%	75%
80% data latih 20% data uji	77.80%	72%

70% data latih 30% data uji	75.50%	64%
--	--------	-----

Langkah selanjutnya adalah menentukan sentimen akhir. Berikut adalah implementasi kode untuk menentukan sentimen akhir menggunakan *majority voting*.

```
18 # Jumlahkan suara untuk setiap kategori sentimen
19 total_counts = {
20     'positive': nb_counts[1] + svm_counts[1],
21     'neutral': nb_counts[0] + svm_counts[0],
22     'negative': nb_counts[-1] + svm_counts[-1]
23 }
24
25 # Tentukan sentimen akhir berdasarkan suara terbanyak
26 final_sentiment = max(total_counts, key=total_counts.get)
27
28 # Cetak jumlah suara untuk setiap kategori
29 print("\nTotal Prediksi Sentimen Gabungan :")
30 print(f"positive = {total_counts['positive']}, neutral = {total_counts['neutral']}, negatif = {total_counts['negative']}")
```

Gambar 20. Source Code Sentimen Akhir

Proses ini menentukan keputusan akhir berdasarkan mayoritas prediksi yang dibuat oleh setiap model. Penentuan akurasi dari *majority vote* dilakukan dengan membandingkan prediksi akhir (*hasil dari majority vote*) dengan label sebenarnya. Hasil akhir sentimen menggunakan skenario 1 (90:10) adalah sebagai berikut.

```
Naive Bayes Sentimen :
positif = 64
neutra = 26
negatif = 311

SVM Sentimen :
positif = 54
neutra = 111
negatif = 237

Total Prediksi Sentimen Gabungan :
positif = 118, neutra = 137, negatif = 548

Final Sentiment = negatif
```

Gambar 21. Hasil Akhir Sentimen

4.8 Tabel Perbandingan

Tabel 5 Hasil Akurasi *Majority Vote*

Pembagian Data	Tingkat Akurasi		
	Naïve Bayes	SVM	Majority Voting
90:10	71.82%	78.11%	80.55%
80:20	70.32%	74.35%	77.80%
70:10	68.16%	71.18%	75.50%

Berdasarkan tabel diatas, pada metode *majority vote* menunjukkan tingkat akurasi

tertinggi sebesar 80.55% ketika menggunakan 90% data latih dan 10% data uji.

Alasan utama saat menggunakan *Majority Vote* bisa menghasilkan tingkat akurasi tinggi adalah karena kombinasi dari dua metode analisis yang saling melengkapi. *Naïve Bayes*, meskipun sederhana dan cepat, cenderung memiliki bias tinggi jika ada fitur yang saling bergantung, sedangkan *Support Vector Machine* efektif dalam menemukan batas keputusan yang optimal di antara kategori tetapi bisa menjadi rumit dan membutuhkan waktu lebih lama. Dengan menggabungkan keduanya, *Majority Vote* memanfaatkan kekuatan dari kedua model untuk menghasilkan prediksi yang lebih akurat dan stabil, terutama ketika didukung oleh *volume* data latih yang besar, memungkinkan model untuk belajar dari variasi data yang lebih luas dan menghasilkan keputusan yang lebih generalisasi terhadap data yang belum terlihat.

4.9 Visualisasi

Setelah mendapatkan hasil dari klasifikasi menggunakan metode *majority voting*, langkah selanjutnya adalah melakukan evaluasi lebih lanjut terhadap model yang dihasilkan. Evaluasi dilakukan dengan visualisasi menggunakan *word cloud*. Berikut adalah kode untuk membuat visualisasi *word cloud*.

```
Word Cloud

1 positif_texts = ' '.join(data[data['sentimen_numeric'] == 1]['nazief_adriani_stemming'])
2 negatif_texts = ' '.join(data[data['sentimen_numeric'] == -1]['nazief_adriani_stemming'])
3 netral_texts = ' '.join(data[data['sentimen_numeric'] == 0]['nazief_adriani_stemming'])
4
5 # Buat WordCloud untuk setiap sentimen
6 def generate_wordcloud(text, title):
7     wordcloud = WordCloud(width=800, height=400,
8     background_color='white').generate(text)
9     plt.figure(figsize=(10, 5))
10    plt.imshow(wordcloud, interpolation='bilinear')
11    plt.axis('off')
12    plt.title(title)
13    plt.show()
```

Gambar 22. Source Code Word Cloud

Berikut adalah gambar hasil implementasi kode diatas.



Gambar 23. Word Cloud Positif



Gambar 23. Word Cloud Negatif



Gambar 23. *Word Cloud Netral*

5. KESIMPULAN

Berdasarkan hasil studi menggunakan metode *Majority Voting* pada *Naive Bayes* dan *Support Vector Machine* untuk menganalisis respons masyarakat terhadap kenaikan pajak hiburan di *Twitter*, beberapa temuan penting dapat disimpulkan sebagai berikut:

1. Hasil penelitian menunjukkan bahwa tingkat akurasi terbesar dengan metode *Support Vector Machine*, *Naive Bayes*, dan *Majority Voting* ada pada saat pengujian menggunakan 90% data latih dan 10% data uji dengan akurasi *Support Vector Machine* sebesar 78.11%. Lalu, pada *Naive Bayes* akurasi sebesar 71.81%. Selanjutnya, pada metode *Majority Voting* dengan menggabungkan kedua metode tingkat akurasi sebesar 80.55%.
 2. Penggunaan Metode *Majority Voting* yang menggabungkan *Naive Bayes* dan *Support Vector Machine* terbukti efektif dalam mengklasifikasikan sentimen positif, negatif dan negatif terhadap kenaikan pajak hiburan. Metode ini menunjukkan tingkat akurasi yang lebih tinggi dibandingkan penggunaan metode tunggal.
 3. Berdasarkan analisis, mayoritas masyarakat menunjukkan sentimen negatif terhadap kenaikan pajak hiburan. Kekhawatiran utama berkisar pada dampak ekonomi yang dirasakan oleh konsumen akibat kenaikan tarif pajak.

UCAPAN TERIMA KASIH

Penulis Saya mengucapkan terima kasih yang sebesar-besarnya atas bimbingan dan dukungan yang diberikan dalam penulisan jurnal ini. Semua kontribusi, baik langsung maupun tidak langsung, sangat berarti dalam proses penelitian ini. Terima kasih atas panduan, saran, dan dorongan yang telah diberikan.

DAFTAR PUSTAKA

- [1] F. Majid, H. S. Sholikhah, and S. Lintang, “Dampak Kenaikan Tarif Pajak Pertambahan Nilai pada Masyarakat Indonesia,” *J. Mhs. Akunt. UNITA*, vol. 2, no. 2, pp. 92–97, 2023, [Online]. Available: https://doi.org/10.36563/jamanta_unita.v2i2.701.
- [2] E. Ratnawati and Rachmad Gesah Mukti Prabowo, “MEKANISME PEMUNGUTAN PPN dan PPnBM,” p. 5, 2019.
- [3] Y. P. Lumentah, “Sistem Pemungutan Pajak Hiburan,” *J. Ris. Ekon. Manajemen, Bisnis dan Akunt.*, vol. 1, no. 3, pp. 1049–1059, 2013.
- [4] K. N. Widyatnyana, I. W. Rasna, and I. B. Putrayasa, “Analisis Jenis dan Makna Pragmatik Ujaran Kebencian di dalam Media Sosial Twitter,” *J. Pendidik. dan Pembelajaran Bhs. Indones.*, vol. 12, no. 1, pp. 68–78, 2023.
- [5] D. Sepri, P. Algoritma, N. Bayes, U. Analisis, K. Penggunaan, and A. Bank, “Media Cetak,” *J. Comput. Syst. Informatics (JoSYC)*, vol. 2, no. 1, pp. 135–139, 2020.
- [6] M. Nur Akbar, N. A. S. Yusuf, N. Nasrullah, and M. Mubarak, “Analisis Sentimen Pengguna Indihome dengan Metode Klasifikasi Support Vector Machine (SVM),” *J. Software, Hardw. Inf. Technol.*, vol. 2, no. 1, pp. 13–21, 2022, doi: 10.24252/shift.v2i1.18.
- [7] R. Putra Pradana, D. Arifianto, H. Azizah, and A. Faruq, “Sentiment Analysis Komentar Youtube Samsung S20 Menggunakan Metode Majority Vote Samsung S20 Youtube Comments Sentiment Analysis Using Majority Vote Method,” *J. Smart Teknol.*, vol. 3, no. 1, pp. 2774–1702, 2021, [Online]. Available: http://jurnal.unmuhammadiyah.ac.id/index.php/JST_PENERBIT:UNIVERSITASMUHAMMADIYAHJEMBER.
- [8] M. A. R. Reynaldhi and Y. Sibaroni, “Analisis Sentimen Review Film pada Twitter menggunakan Metode Klasifikasi Hybrid Naïve Bayes dan Decision Tree,” *e-Proceeding Eng.*, vol. 8, no. 5, pp. 10127–10137, 2021.
- [9] I. P. A. M. Utama, S. S. Prasetyowati, and Y. Sibaroni, “Multi-Aspect Sentiment Analysis Hotel Review Using RF, SVM, and Naïve Bayes based Hybrid Classifier,” *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 630, 2021, doi: 10.30865/mib.v5i2.2959.
- [10] F. V. Sari and A. Wibowo, “Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi,” *J. SIMETRIS*, vol. 10, no. 2, pp. 681–686, 2019.
- [11] R. I. Agustin, “Komparasi Algoritma Naïve Bayes Dan Svm Untuk Analisis Sentimen Twitter Korupsi Bansos Beras Masa Pandemi,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 912–918, 2024, doi: 10.23960/jitet.v12i2.4020.
- [12] Y. P. Dewi, Y. Abidin, and K. Kurniawan, “Penggunaan Twitter terhadap Pembelajaran Bahasa Indonesia,” *Riksa Bhs.*, vol. XV, pp. 307–312, 2021.
- [13] A. D. Wibisono, S. Dadi Rizkiono, and A. Wantoro, “Filtering Spam Email Menggunakan Metode Naive Bayes,” *TELEFORTECH J. Telemat. Inf. Technol.*, vol. 1, no. 1, 2020, doi: 10.33365/tft.v1i1.685.
- [14] I. Riadi, R. Umar, and F. D. Aini, “Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (Svm),” *Ilk. J. Ilm.*, vol. 11, no. 1, pp. 17–24, 2019, doi: 10.33096/ilkom.v11i1.361.17-24.
- [15] A. Y. Alfajr, K. Kartini, and A. P. Sari, “Classification of Distracted Driver Using Support Vector Machine Based on Principal Component Analysis Feature Reduction and Convolutional Neural Network,” *J. Komput. dan Inform.*, vol. 11, no. 2, pp. 237–245, 2023, doi: 10.35508/jicon.v11i2.12658.