

# Analisis DistilBERT dengan Support Vector Machine (SVM) untuk Klasifikasi Ujaran Kebencian pada Sosial Media Twitter

Naufal Azmi Verdikha<sup>1</sup>, Reza Habid<sup>2</sup>, Asslia Johar Latipah<sup>3</sup>

<sup>1,2,3</sup> Teknik Informatika, Universitas Muhammadiyah Kalimantan Timur, Indonesia

\*nav651@umkt.ac.id

## Abstract

*Hate speech is a significant issue in content management on social media platforms. Effective classification of hate speech plays a crucial role in maintaining a safe social media environment, combating discrimination, and protecting users. This study evaluates a hate speech classification model using SVM with linear and polynomial kernels. The dataset used consists of labeled Indonesian-language tweets. The importance of developing an effective classification model to address hate speech has led to the utilization of DistilBERT as a feature extraction method. However, DistilBERT has high-dimensional features, necessitating dimensionality reduction to reduce model complexity. Therefore, in this study, the PCA dimensionality reduction method is implemented with various scenarios of dimensionality, namely 10, 20, 30, 40, and 50. Evaluation is performed using F1-Score, and the entire study is evaluated using 10-fold cross-validation. The evaluation results indicate that in the scenario with a linear kernel, the model achieves the highest F1-Score of 0.75 in the 50-dimensional scenario. Meanwhile, in the scenario with a polynomial kernel, the model achieves the highest F1-Score of 0.7857 in the 50-dimensional scenario. These findings demonstrate that the use of a polynomial kernel with 50 dimensions yields the best performance in classifying hate speech.*

*Keywords: hate speech, SVM classification, PCA dimension reduction, F1-Score.*

## Abstrak

Ujaran kebencian merupakan isu yang signifikan dalam pengelolaan konten di platform media sosial. Klasifikasi ujaran kebencian yang efektif memainkan peran penting dalam menjaga lingkungan media sosial yang aman, memerangi diskriminasi, dan melindungi pengguna. Penelitian ini mengevaluasi model klasifikasi ujaran kebencian menggunakan SVM dengan kernel linier dan polinomial. Dataset yang digunakan adalah kumpulan data tweet berbahasa Indonesia yang telah diberi label. Pentingnya pengembangan model klasifikasi yang efektif untuk mengatasi ujaran kebencian mendorong penggunaan metode ekstraksi fitur DistilBERT. Namun, DistilBERT memiliki dimensi fitur yang tinggi, sehingga diperlukan reduksi dimensi untuk mengurangi kompleksitas model. Oleh karena itu, dalam penelitian ini, metode reduksi dimensi PCA diimplementasikan dengan berbagai skenario dimensi yang berbeda, yaitu 10, 20, 30, 40, dan 50. Evaluasi menggunakan nilai F1-Score, dan seluruh penelitian dievaluasi menggunakan metode 10-fold cross validation. Hasil evaluasi menunjukkan bahwa pada skenario dengan kernel linier, model mencapai F1-Score tertinggi sebesar 0.75 pada skenario dengan dimensi 50. Sementara itu, pada skenario dengan kernel polinomial, model mencapai F1-Score tertinggi sebesar 0.7857 pada skenario dengan dimensi 50. Hasil ini menunjukkan bahwa penggunaan kernel polinomial dengan dimensi 50 memberikan kinerja terbaik dalam mengklasifikasikan ujaran kebencian.

*Kata kunci: ujaran kebencian, klasifikasi, DistilBERT, reduksi dimensi PCA, F1-Score*

## 1. Pendahuluan

Kemunculan media sosial membawa perubahan perilaku masyarakat pada beberapa aspek seperti budaya, moralitas dan norma-norma yang ada, sehingga seluruh masyarakat Indonesia dari segala usia dan lapisan masyarakat memiliki dan menggunakan media sosial untuk mengumpulkan informasi dan

membagikannya kepada masyarakat[1]. Saat ini, media sosial sebagai sarana komunikasi telah menjadi bagian penting dalam kehidupan sehari-hari masyarakat luas dan keberadaannya telah mengubah cara masyarakat dalam mengutarakan pandangannya. Selain itu, teknologi internet juga berperan aktif agar masyarakat dapat menerima informasi dari mana saja [2].

Media sosial menyediakan pengguna Internet, atau lebih dikenal sebagai netizen, sebuah alat untuk komunikasi online. Misalnya, di Twitter, pengguna internet dapat berkomunikasi melalui tweet yang dikirimkan ke aplikasi. Tweet ini bisa positif dan ada pula yang negatif. Komentar negatif menjadi masalah karena seringkali mengandung ujaran kebencian dan dapat berujung pada tindakan hukum terhadap penciptanya [3].

Ujaran kebencian atau ujaran kebencian merupakan ujaran yang menyinggung perasaan orang lain dan seringkali dianggap sebagai “kejahatan”. Dapat digunakan untuk menargetkan individu atau kelompok orang. Dijelaskan bahwa ujaran kebencian merupakan suatu bentuk perbuatan atau perbuatan yang dilakukan atas dasar supremasi hukum. Dalam hal ujaran kebencian tidak dilakukan, maka peraturan perundang-undangan yang mengatur ujaran kebencian harus dipatuhi. Selain itu, terdapat sanksi yang dikenal dengan ancaman pidana dalam peraturan perundang-undangan tersebut. Dijelaskan bahwa ujaran kebencian merupakan suatu bentuk perbuatan atau perbuatan yang dilakukan atas dasar supremasi hukum. Dalam hal ujaran kebencian tidak dilakukan, maka peraturan perundang-undangan yang mengatur ujaran kebencian harus dipatuhi. Selain itu, undang-undang dan peraturan ini memberikan hukuman yang disebut intimidasi pidana [4].

Salah satu tindakan yang dapat dilakukan untuk mengetahui apakah suatu kata atau frasa merupakan ujaran kebencian adalah dengan menggunakan *Natural Language Processing* (NLP). Pemrosesan bahasa alami (NLP) adalah bagian dari kecerdasan buatan yang memungkinkan komputer memahami bahasa manusia dan memberikan hasil dalam bahasa tersebut. Pada dasarnya *Natural Language Processing* (NLP) terdiri dari tiga komponen utama yaitu database, mesin inferensi, dan antarmuka pengguna. Basis data berisi domain dan peristiwa heuristik. Semua informasi yang dikumpulkan akan disimpan dalam database dalam bentuk model yang sesuai [5].

DistilBERT, yang merupakan singkatan dari "Distilled Bidirectional Encoder Representations from Transformers," adalah

sebuah inovasi terkini dalam domain pengolahan bahasa alami (NLP). Dalam konteks NLP, DistilBERT adalah sebuah model bahasa yang telah dirombak menjadi versi yang lebih ringkas dan efisien, tetapi tetap mempertahankan kemampuan utamanya dalam pemahaman bahasa manusia. Dikembangkan berdasarkan arsitektur Transformer yang terkenal, DistilBERT membawa manfaat signifikan dalam percepatan pelatihan dan penghematan sumber daya komputasi tanpa mengorbankan kualitas pemrosesan bahasa [6]. Versi BERT saat ini yang dikenal sebagai DistilBERT memiliki fungsionalitas dan tampilan yang sama. DistilBERT juga digunakan dengan cara yang sama seperti BERT untuk membantu *Nature Language Processing*. Akan tetapi, DistilBERT memiliki hasil yang lebih rendah daripada BERT, yaitu 71% dari total hasil BERT [7].

Penelitian sebelumnya oleh [8] melakukan penelitian dengan judul “*Online New Sentiment Clasification Using DistilBERT*” menggunakan pengklasifikasi pembelajaran mesin NBSVM (Naive Bayes - Support Vector Machines) dan pengklasifikasi pembelajaran mendalam BiGRU. Dengan menggunakan dua data berbeda yaitu data asli seimbang dan data asli tidak seimbang dengan nilai skor F1 pada data asli seimbang, NBVSM memperoleh nilai akurasi F1-Score sebesar 0,85 dan 0,86 pada data asli tidak seimbang dan nilai F1-Score pada klasifikasi deep learning data asli tidak seimbang 0,89 dan data asli seimbang 0,91.

Penelitian lain yang menggunakan DistilBERT yang dilakukan oleh [9] dengan judul “*Natural language processing analysis applied to COVID 19 open text opinions using a distilBERT model for sentiment categorization*” menggunakan tiga pendekatan data. Nilai akurasi pengklasifikasian kalimat dengan sentimen negatif lebih tinggi untuk pendekatan ketiga dengan nilai akurasi presisi 0,826 dan nilai F1-score 0,803.

Penelitian sebelumnya oleh [10] melakukan penelitian dengan judul “*Dimensionality reduction using Principal Component Analysis for cancer detection based on microarray data classification*”.

Penelitian ini membandingkan dua metode klasifikasi: Support Vector Machine (SVM) dan algoritma Levenberg-Marquardt Backpropagation (LMBP). PCA digunakan untuk mereduksi dimensi data dengan hasil yang baik, menghasilkan data yang hanya memiliki error kurang dari 2% dari data asli. Jumlah vektor eigen tidak signifikan meningkatkan akurasi klasifikasi. SVM menunjukkan bahwa penggunaan kernel linier memberikan akurasi rata-rata sekitar 92,26%, sementara dengan kernel RBF akurasi meningkat menjadi 94,46%. Pada LMBP, jumlah neuron mempengaruhi kestabilan model; dengan 15 neuron, akurasi tertinggi mencapai 96,07%, sementara SVM mencapai akurasi 94,98%. LMBP lebih baik dalam menggeneralisasi data baru dari data microarray.

Dari penelitian [11] dengan judul “Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter”, mendeteksi ujaran kebencian dan kata-kata kasar dengan banyak label, dimana kata-kata yang digunakan dikategorikan dalam beberapa kategori. Label atau kelas yang digunakan terdiri dari 12 kategori berdasarkan kategori antara lain berdasarkan kategori yaitu ujaran kebencian, ujaran kasar, ujaran kebencian individu, ujaran kebencian kelompok, ujaran kebencian agama, ujaran kebencian rasial, ujaran kebencian fisik, ujaran kebencian gender, ujaran kebencian pidato. lainnya, ujaran kebencian yang lemah, ujaran kebencian yang moderat, dan ujaran kebencian yang kuat. Jumlah tweet ujaran kebencian sebanyak 5.561 tweet, 5.043 tweet ofensif, dan 7.604 tweet tanpa ujaran kebencian.

Penelitian ini akan menguji model pembelajaran klasifikasi ujaran kebencian menggunakan data yang berasal dari penelitian[11]. Penelitian ini akan menggunakan DistilBERT sebagai metode ekstraksi fitur, dan selanjutnya akan menerapkan teknik reduksi dimensi menggunakan PCA pada fitur yang diekstraksi. Fitur yang telah direduksi dimensinya akan dipelajari model pendeteksian menggunakan algoritma SVM dan nilai evaluasi F1-score.

## 2. Metode Penelitian

Alur penelitian ini digambarkan seperti pada Gambar 1 berikut.



Gambar 1. Metodologi Penelitian

Penelitian ini dilakukan dalam beberapa tahapan, diantaranya mulai, menginput dataset penelitian [11], pre-processing data, ekstraksi DistilBERT, reduksi dimensi PCA, klasifikasi SVM, dan Evaluasi F1 Score.

### A. Dataset

Tahapan ini dilakukan dengan menginput dataset dari penelitian [11] menguji algoritma dalam mencari model yang sesuai kelas klasifikasinya. Data set ini berasal dari open source github.

### B. Pre-Processing Data

Pada tahapan ini dataset akan melalui beberapa tahap pembersihan data, tahapan pre-processing sebagai berikut:

1. Memberikan id disetiap kolom  
Fungsi id untuk mempermudah mengidentifikasi, melacak dan memproses data agar lebih efisien dan akurat.
2. Lower Case  
Tahapan ini mengubah semua teks menjadi huruf kecil agar memudahkan dalam pemrosesan di tahap berikutnya.
3. Remove Attribute

Tahapan ini digunakan untuk menghapus atribut-atribut spesifik dalam teks tweet, seperti URL, simbol retweet, username, emoji, dan karakter berulang.

#### 4. *Non Alphanumeric*

Tahapan ini untuk menghapus semua karakter alphanumeric atau karakter yang tidak diinginkan, seperti tanda baca, simbol, atau karakter khusus lainnya.

#### 5. *Input kamus alay dan stopword*

Pada tahapan ini 2 dataset kamus alay dan 1 *dataset stopword* di input kedalam pre-processing, berikut ini merupakan isi dari kamus alay dan *stopword*.

#### 6. *Spell Checker*

Dalam tahapan spell checker ini, digunakan dua kamus alay untuk melakukan koreksi ejaan dalam sebuah teks, sehingga ejaannya sesuai dengan standar yang terdapat dalam Kamus Besar Bahasa Indonesia (KBBI). Dalam kamus alay yang digunakan, yaitu "alay\_dict 2", mengandung banyak kata-kata bahasa daerah. Digunakan untuk memperbaiki kata-kata yang tidak tepat, konsultasi dilakukan dengan para penutur asli dari suku yang bersangkutan. Penggunaan kamus alay bertujuan untuk menjernihkan dan memperbaiki penggunaan bahasa agar sesuai dengan standar resmi yang terdapat dalam KBBI.

#### 7. *Stemming*

Tahapan ini digunakan untuk mengubah teks menjadi kata dasar dengan menggunakan library open source Sastrawi.

#### 8. *Stopword*

Tahapan ini digunakan untuk menghapus kata-kata yang umum dan tidak memberikan banyak informasi. Dalam proses ini menggunakan dataset *stopword* yang sudah di-input sebelumnya

#### 9. *Remove Data Kosong*

Pada tahapan ini menghapus data kosong, data yang hanya berisi spasi serta data yang hanya berisi 1 kata.

#### C. *Fitur Ekstraksi DistilBERT*

Tahap ini digunakan untuk menerapkan model DistilBERT pada dataset yang telah melalui proses tokenisasi, padding, dan masking. Pada tahap ini, dataset yang telah

diolah tersebut disesuaikan agar sesuai dengan struktur masukan yang diterima oleh model DistilBERT.

#### D. *Reduksi Dimensi PCA*

Pada tahap ini, fitur-fitur hasil ekstraksi menggunakan model DistilBERT akan digunakan sebagai input untuk PCA. Analisis komponen utama (PCA) adalah metode analisis statistik yang digunakan untuk mengurangi ukuran kumpulan data yang kompleks [12]. Tujuan utama PCA adalah untuk mengidentifikasi pola dan struktur yang ada dalam kumpulan data dengan mengurangi dimensi variabel sambil mempertahankan informasi sebanyak mungkin. PCA mencapai hal ini dengan menemukan kombinasi linier baru dari variabel asli, yang disebut komponen utama. Komponen utama ini merupakan vektor arah dalam ruang fitur, dimana proyeksi data ke komponen utama pertama menghasilkan varian tertinggi. Tujuan dari PCA adalah untuk mengurangi dimensi fitur-fitur tersebut dengan tetap mempertahankan sebanyak mungkin informasi yang terkandung dalam fitur-fitur tersebut. Nilai komponen atau dimensi yang digunakan 10, 20, 30, 40, 50.

Penggunaan maksimal 50 nilai komponen atau dimensi dalam reduksi dimensi PCA pada penelitian ini didasarkan pada pertimbangan kapasitas perangkat keras yang tersedia. Proses PCA yang melibatkan perhitungan matriks kovariansi dan vektor eigen dapat menjadi komputasi yang intensif, membutuhkan sumber daya komputasional yang cukup besar. Keterbatasan kapasitas perangkat keras yang tersedia dalam penelitian ini membatasi jumlah komponen utama yang dapat digunakan untuk memastikan kinerja dan keefisienan komputasi yang optimal.

#### E. *Cross Validation*

Pada tahap ini, pembagian dataset menjadi beberapa subset (fold), di mana setiap subset digunakan secara bergantian sebagai data uji (*train\_index*) dan yang lainnya sebagai data latih (*test\_index*), menggunakan nilai K sebanyak 10 fold. Dalam metode validasi silang k-fold, kumpulan data dibagi secara acak menjadi beberapa partisi. Kemudian dilakukan sejumlah pengujian sesuai dengan jumlah



diharapkan dataset yang semula berisi kata-kata atau kalimat dalam bahasa sehari-hari pengguna Twitter dapat disesuaikan menjadi bahasa Indonesia yang sesuai dengan norma bahasa dalam KBBI.

Proses *preprocessing* ini menjadi langkah kritis dalam mempersiapkan data sebelum dilakukan analisis lebih lanjut dan membantu memastikan keberhasilan dari model yang digunakan pada penelitian ini. Hasil dari *preprocessing* ini dapat dilihat pada gambar di bawah ini.

Gambar 3 Isi Dataset Setelah *Preprocessing*

Gambar diatas memperlihatkan isi dari dataset yang sudah melalui tahap *preprocessing*. Dapat dilihat perbedaan isi kolom *tweet\_clean* dan isi kolom *tweet\_clean* ada perubahan isi. Sebelum di *pre-processing* dataset memiliki 13169 baris dan 13 kolom setelah di *preprocessing* dataset memiliki 13112 baris dan 4 kolom.

### 3.3. Fitur Ekstraksi Distilbert

Setelah melalui tahap *preprocessing* pada dataset Twitter, langkah selanjutnya adalah proses fitur ekstraksi menggunakan model DistilBERT. Proses ini melibatkan tiga kolom pada dataset, yaitu kolom ID, kolom "tweet clean" yang merupakan hasil dari proses *preprocessing*, dan kolom HS yang berisi label atau informasi tentang sentimen dari setiap tweet, lebel 0 menyatakan kelas Non Hate speech dan lebel 1 menyatakan kelas Hate speech.

ID	Tweet_Clean	HS
0	d00001 cowok usaha lacak perhati lantas remeh perhati...	1
1	d00002 telat tau edan sarap gaul cigax jiffa cal licew	0
2	d00003 41 kadang pikir percaya tuhan jatuh kali kali ...	0
3	d00004 ku tau mata sipit lihat	0
4	d00005 kaum cebong kafir lihat dongok dungu haha	1
...	...	...
13107	d13165 bicara ndasmu congor sekata anjing	1
13108	d13166 kasur enak kunyuk	0
13109	d13167 hati hati bisu bosan duh	0
13110	d13168 bom real mudah deteksi bom kubur dahsyat ledak...	0
13111	d13169 situ foto kutil onta	1

Gambar 4. Isi Dataset Proses Ekstraksi Fitur

Pada tahap fitur ekstraksi ini, setiap baris dataset dijalankan melalui model DistilBERT untuk mendapatkan representasi numerik yang disebut "token." Token adalah unit dasar dari teks yang telah diubah menjadi bilangan numerik agar dapat diolah oleh model. Dengan menggunakan DistilBERT, kita dapat mengonversi teks dari setiap baris dataset menjadi vektor numerik yang mencerminkan makna dan informasi di dalamnya.

Selanjutnya, dalam proses ekstraksi fitur ini, dataset yang telah diolah oleh DistilBERT diambil tokennya dengan memperhatikan panjang maksimum yang telah ditetapkan sebesar 128 token. Artinya, jika teks pada suatu baris dataset melebihi panjang maksimum tersebut, maka teks tersebut akan dipotong atau dipendekkan menjadi 128 token sesuai aturan yang telah ditetapkan.

Hasil dari proses fitur ekstraksi menggunakan DistilBERT adalah vektor numerik yang merepresentasikan setiap tweet dalam dataset. Vektor-vektor ini akan digunakan sebagai representasi fitur dari teks pada tweet tersebut dan dapat menjadi input untuk model machine learning, seperti klasifikasi sentimen atau tugas lainnya.

token_ids \	feature_distilbert
0 [3, 24656, 3641, 12612, 1500, 3543, 1702, 1852...	[[0.14643915, 0.1713648, -0.5270182, -0.202883...
1 [3, 2128, 1494, 8296, 2645, 1487, 5506, 1029, ...	[[[-0.20156366, -0.22168754, -0.22396314, 0.333...
2 [3, 8217, 3362, 11808, 4823, 4379, 3763, 1859, ...	[[[0.21471566, 0.41791952, 0.050950672, 0.37290...
3 [3, 2030, 8296, 3122, 3908, 1531, 5927, 1, 2, ...	[[[0.036371592, -0.14293818, -0.21324247, 0.035...
4 [3, 3605, 12370, 12147, 19075, 5927, 6479, 160...	[[[-0.14531858, 0.24509943, -0.2311413, -0.1876...
...	...
13107 [3, 10272, 13858, 21656, 1005, 12014, 1528, 31...	[[[0.36976746, 0.34318095, -0.1703101, 0.409832...
13108 [3, 26879, 16093, 2763, 4727, 1028, 1, 2, 2, 2...	[[[0.13579059, -0.09266545, -0.3739355, 0.30271...
13109 [3, 4006, 4006, 14619, 26376, 3343, 1015, 1, 2...	[[[0.10735784, 0.08104091, -0.41293126, 0.34848...
13110 [3, 6223, 6447, 3947, 20592, 6223, 17443, 1603...	[[[0.6207894, -0.21312533, -0.68504727, 0.73445...
13111 [3, 4534, 5007, 13058, 1514, 14974, 1007, 1, 2...	[[[0.17882411, 0.16073936, 0.14147611, 0.444890...

Gambar 5. Hasil Ekstraksi Fitur

### 3.4. Reduksi Dimensi PCA

Setelah selesai melakukan ekstraksi fitur menggunakan model DistilBERT pada dataset, langkah selanjutnya adalah menerapkan Principal Component Analysis (PCA). Pada tahap ini, fitur-fitur yang dihasilkan dari ekstraksi menggunakan DistilBERT akan dijadikan sebagai input



untuk PCA. Tujuan utama dari PCA adalah untuk mengurangi dimensi fitur-fitur tersebut, sambil mempertahankan sebanyak mungkin informasi yang terkandung dalam fitur-fitur tersebut.

Pada tahap reduksi dimensi PCA ini, telah ditetapkan beberapa nilai komponen atau dimensi yang akan diuji, yaitu 10, 20, 30, 40, dan 50. Artinya, kita akan mencoba mengurangi dimensi fitur-fitur tersebut menjadi berbagai jumlah komponen utama tersebut. Penggunaan nilai maksimal 50 komponen dalam PCA pada penelitian ini dipilih berdasarkan pertimbangan kapasitas perangkat keras yang tersedia. Proses PCA melibatkan perhitungan matriks kovariansi dan vektor eigen, yang dapat menjadi komputasi yang intensif dan memerlukan sumber daya komputasional yang cukup besar.

Keterbatasan kapasitas perangkat keras yang tersedia dalam penelitian ini menyebabkan jumlah komponen utama yang dapat digunakan dalam PCA menjadi terbatas. Meskipun demikian, pemilihan jumlah komponen utama yang optimal akan tetap diupayakan untuk memastikan kinerja dan efisiensi komputasi yang optimal dalam analisis dataset ini. Dengan menerapkan PCA sebagai langkah reduksi dimensi, diharapkan bahwa dataset yang semula memiliki fitur-fitur dengan dimensi tinggi dapat diwakili dengan jumlah komponen yang lebih rendah namun tetap mempertahankan sebagian besar informasi penting dari dataset asli. Hal ini akan membantu dalam mempercepat proses analisis dan meningkatkan efisiensi dalam penggunaan model machine learning pada penelitian ini.

Tabel 1. Waktu Reduksi Dimensi

Nilai Komponen	Waktu
10	2 m 26.4s
20	2 m 36.4s
30	2 m 53.5s
40	3 m 22.3s
50	6 m 53.4s

Waktu yang diperlukan dalam proses reduksi dimensi dapat dipengaruhi oleh besarnya nilai komponen yang direduksi. Semakin besar nilai komponen yang

diinginkan, semakin banyak informasi yang harus dipertahankan, dan akibatnya, semakin lama waktu yang dibutuhkan untuk memproses dataset. Dapat dilihat pada tabel dibawah ini.

### 3.5. Cross Validation

Proses *cross validation* menggunakan KFold 10 berdasarkan parameter yang sudah ditentukan. Tahap ini dilakukan sebelum masuk dalam klasifikasi SVM dan evaluasi F1-score dengan memisahkan data hasil reduksi dimensi dengan nilai komponen dimensi 50 dan kolom HS pada data hasil pre processing. Bagian ini menampilkan hasil pembagian data train dan data test dari fold 1 dan fold 10 untuk mengetahui pembagian data. Hasil proses *cross validation* terdapat pada gambar 6 dan gambar 7 dibawah ini.

```

100%|██████████| 10/10 [00:00<00:00, 101.27it/s]
Fold 1:
- Train data: 11800 samples
- Test data: 1312 samples
Fold 1:
- Train Index: [ 1287 1288 1292 ... 13109 13110 13111]
- Test Index: [ 0 1 2 ... 1323 1324 1325]

```

Gambar 6. Hasil Fold 1

Dari hasil diatas bahwa pada fold 1 terdapat data train dengan jumlah 11800 sample dengan indeks dari 1287 sampai 13111 dan data tes sebanyak 1312 dengan indeks dari 0 sampai 1325.

```

Fold 10:
- Train data: 11801 samples
- Test data: 1311 samples
Fold 10:
- Train Index: [ 0 1 2 ... 11808 11809 11811]
- Test Index: [11796 11797 11798 ... 13109 13110 13111]

```

Gambar 7. Hasil Fold 10

Dari hasil diatas dapat dilihat bawah pada fold 10 terdapat data train dengan jumlah 11801 sample dengan indeks dari 0 sampai 11811 dan data test sebanyak 1311 sample dengan indeks dari 11796 sampai 13111. Waktu yang digunakan untuk menjalankan proses cross validatin KFold 1 sampai 10 dibawah 1 detik.

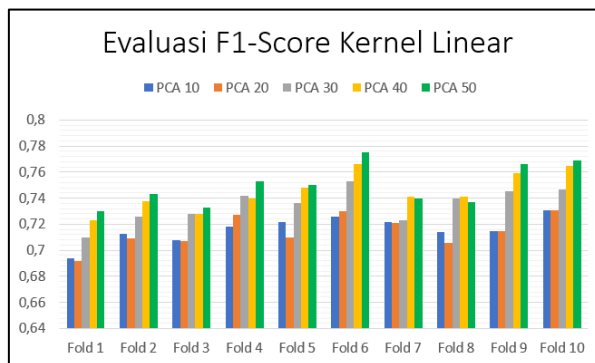
### 3.6. Klasifikasi SVM & Evaluasi F1 Score

Dari tahapan ini hasil cross validation memasuki klasifikasi dengan algoritma SVM dan menghitung evaluasinya untuk mengetahui keefektifan metode dari hasil evaluasinya. Pada klasifikasi ini menggunakan Kernel linear dan polynomial

dan menggunakan 5 nilai komponen dimensi pada setiap Kernelnya. Pada Kernel linear hasil yang diperoleh sebagai berikut.

Tabel 2. Hasil Evaluasi dari kernel Linear

Fold	Nilai Komponen Dimensi & F1-Score				
	10	20	30	40	50
Fold 1	0,694	0,692	0,71	0,723	0,730
Fold 2	0,713	0,709	0,726	0,738	0,743
Fold 3	0,708	0,707	0,728	0,728	0,733
Fold 4	0,718	0,727	0,742	0,74	0,753
Fold 5	0,722	0,710	0,736	0,748	0,750
Fold 6	0,726	0,730	0,753	0,766	0,775
Fold 7	0,722	0,721	0,723	0,741	0,740
Fold 8	0,714	0,706	0,740	0,741	0,737
Fold 9	0,715	0,715	0,745	0,759	0,766
Fold 10	0,731	0,731	0,747	0,765	0,769



Gambar 8. Grafik Evaluasi F1 Score Kernel Linear

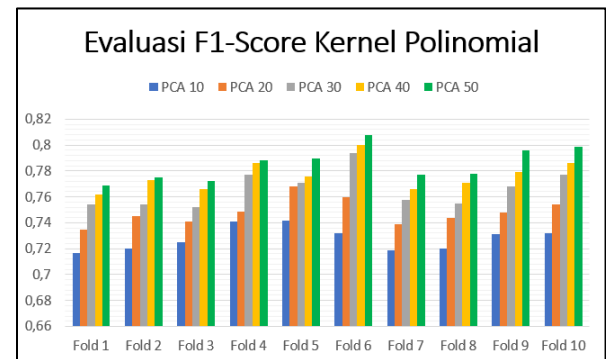
Berdasarkan hasil dari tabel dan grafik diatas pada kernel linear nilai komponen dimensi 50 mendapatkan hasil yang lebih tinggi dari setiap foldnya daripada nilai komponen dimensi lainnya, dan nilai terendah berada pada komponen dimensi 10.

Dalam percobaan menggunakan kernel polinomial evaluasi f1-score memperoleh hasil seperti tabel dibawah ini.

Tabel 3. Hasil Evaluasi dari kernel Polinomial

Fold	Nilai Komponen Dimensi & F1-Score				
	10	20	30	40	50
Fold 1	0,717	0,735	0,754	0,762	0,769
Fold 2	0,720	0,745	0,754	0,773	0,775
Fold 3	0,725	0,741	0,752	0,766	0,772
Fold 4	0,741	0,749	0,777	0,786	0,788
Fold 5	0,742	0,768	0,771	0,776	0,790
Fold 6	0,732	0,760	0,794	0,800	0,808
Fold 7	0,719	0,739	0,758	0,766	0,777
Fold 8	0,720	0,744	0,755	0,771	0,778
Fold 9	0,731	0,748	0,768	0,779	0,796

Fold	Nilai Komponen Dimensi & F1-Score				
	10	20	30	40	50
Fold 10	0,732	0,754	0,777	0,786	0,799



Gambar 9. Grafik Evaluasi F1 Score Kernel Polinomial

Berdasarkan hasil dari tabel dan grafik diatas pada kernel polinomial nilai komponen dimensi 50 mendapatkan hasil yang lebih tinggi dari setiap foldnya daripada nilai komponen dimensi lainnya, dan nilai terendah berada pada komponen dimensi 10.

Jika dilihat dari konfusi matriks makan akan diperoleh hasil seperti pada tabel dibawah ini dari setiap kernel SVM dan nilai komponen dimensi dapat dilihat pada tabel dibawah ini.

Tabel 4. Confusion Matrix (Average F1-Score Kernel Linear)

Confusion Matrix (Average F1-Score Kernel Linear)				
Nilai PCA	TP	FP	TN	FN
PCA 50	390	114	642	165
PCA 40	358	133	623	198
PCA 30	351	139	617	204
PCA 20	343	158	598	212
PCA 10	344	157	599	211

Tabel 5. Confusion Matrix (Average F1-Score Kernel Polinomial)

Confusion Matrix (Average F1-Score Kernel Polinomial)				
Nilai PCA	TP	FP	TN	FN
PCA 50	390	114	642	165
PCA 40	384	119	637	171
PCA 30	374	123	634	181
PCA 20	363	135	621	192
PCA 10	336	131	625	219



Sehingga berdasarkan TP, TN, FP, dan FN dari setiap kernel dan nilai komponen dimensi berbeda serta perhitungan rata-rata F1-Score dari setiap Fold dan komponen dimensi dalam fungsi program diperoleh hasil dan waktu yang diperlukan dapat dilihat pada tabel dibawah ini.

Tabel 6. Hasil Rata-rata F1-score

Rata-rata F1-Score			
Kernel Linear		Polinomial	
Nilai Komponen	F1-Score	Nilai Komponen	F1-Score
10	0,716	10	0,728
20	0,715	20	0,748
30	0,735	30	0,766
40	0,745	40	0,777
50	0,749	50	0,785

Berdasarkan hasil rata-rata F1-Score yang diperoleh nilai tertinggi menggunakan kernel polinomial dengan nilai komponen dimensi 50 sebesar 0.785 dan terendah berada dikernel linear dengan nilai komponen dimensi 20 sebesar 0.715. Waktu terlalu lama dimasing-masing kernel yang diperlukan pada kernel linear dengan nilai komponen dimensi 50 selama 5 jam 16 menit 28 detik dan pada kernel polinomial dengan nilai komponen dimensi 50 selama 55 detik.

#### 4. Kesimpulan

Dalam penelitian ini, dilakukan evaluasi metode ekstraksi fitur distilbert menggunakan reduksi dimensi PCA dan klasifikasi SVM dengan metrik F1-Score pada dua jenis kernel yang digunakan (Linear dan Polinomial) pada masing-masing nilai komponen dimensi 10, 20, 30, 40, 50 dengan 10-fold cross validation. Hasil evaluasi menunjukan bahwa rata-rata F1-score tertinggi terdapat pada kernel polinomial dengan nilai komponen dimensi 50 sebesar 0,785, sedangkan F1-Score terendah terdapat pada kernel linear dengan nilai komponen dimensi 20 sebesar 0,715. Dalam hal waktu komputasi, kernel polinomial membutuhkan waktu yang lebih sedikit dari pada kernel linear, namun menunjukkan hasil F1 score yang cukup baik dibandingkan kernel linear. Selain itu reduksi dimensi menggunakan PCA cukup efisien

karena dapat mempercepat waktu komputasi di masing-masing kernel dengan nilai komponen dimensi berbeda-beda. Dapat disimpulkan bahwa dalam konteks penelitian ini metode ekstraksi fitur DistilBERT dengan nilai komponen dimensi 50 menggunakan kernel polinomial memiliki kinerja yang cukup baik dengan nilai F1-score 0,785 atau 78,5%.

#### 5. Saran

Berdasarkan hasil penelitian ini, adapun saran yang dapat penulis berikan yaitu:

1. Memperhatikan metode pembersihan kata-kata pada tahapan pre-processing seperti dalam kata-kata yang tidak baku dan kata-kata yang tidak terdeteksi dengan baik untuk memperbaiki keakuratan pada proses analisis.
2. Mempertimbangkan metode reduksi dimensi dalam penentuan nilai komponen untuk menghasilkan nilai evaluasi yang baik.

#### 6. Daftar Pustaka

- [1] A. S. Cahyono, "Pengaruh Media Sosial Terhadap Perubahan Sosial Masyarakat di Indonesia," *J. Publiciana*, vol. 9, no. 1, pp. 140–157, 2018, [Online]. Available: <https://journal.unita.ac.id/index.php/publiciana/article/view/79>
- [2] A. R. Isnain, A. I. Sakti, D. Alita, and N. S. Marga, "Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma Svm," *Jdmsi*, vol. 2, no. 1, pp. 31–37, 2021, [Online]. Available: <https://t.co/NfhnmJtXw>
- [3] A. P. J. Dwitama, "Deteksi Ujaran Kebencian Pada Twitter Bahasa Indonesia Menggunakan Machine Learning : Reviu Literatur," *J. SNATi*, vol. 1, no. 1, pp. 31–39, 2021.
- [4] I. M. Kardiyasa, A. A. S. L. Dewi, and N. M. S. Karma, "Sanksi Pidana Terhadap Ujaran Kebencian (Hate Speech)," *J. Analog. Huk.*, vol. 2, no. 1, pp. 78–82, 2020, doi: 10.22225/ah.2.1.1627.78-82.
- [5] I. Iswandi, I. S. Suwardi, and N. U. Maulidevi, "Penelitian Awal :

- Otomatisasi Interpretasi Data Akuntansi Berbasis Natural Language Processing mudah untuk digunakan , sehingga manusia berharap akan dapat berbicara kepada karena lambatnya pemahaman terhadap transaksi yang terjadi . menampilkan informasi,” *J. Sist. Inf.*, vol. 5, no. 2, pp. 622–628, 2013.
- [6] F. Fajri, B. Tutuko, and S. Sukemi, “Membandingkan Nilai Akurasi BERT dan DistilBERT pada Dataset Twitter Tahapan Penelitian,” vol. 8, no. 2, 2022.
- [7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” pp. 2–6, 2019, [Online]. Available: <http://arxiv.org/abs/1910.01108>
- [8] S. K. Akpatsa *et al.*, “Online News Sentiment Classification Using DistilBERT,” *J. Quantum Comput.*, vol. 4, no. 1, pp. 1–11, 2022, doi: 10.32604/jqc.2022.026658.
- [9] M. Jojoa, P. Eftekhar, B. Nowrouzi-Kia, and B. Garcia-Zapirain, “Natural language processing analysis applied to COVID-19 open-text opinions using a distilBERT model for sentiment categorization,” *AI Soc.*, no. 0123456789, 2022, doi: 10.1007/s00146-022-01594-w.
- [10] Adiwijaya, U. N. Wisesty, E. Lisnawati, A. Aditsania, and D. S. Kusumo, “Dimensionality reduction using Principal Component Analysis for cancer detection based on microarray data classification,” *J. Comput. Sci.*, vol. 14, no. 11, pp. 1521–1530, 2018, doi: 10.3844/jcssp.2018.1521.1530.
- [11] M. O. Ibrohim and I. Budi, “Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter,” pp. 46–57, 2019, doi: 10.18653/v1/w19-3506.
- [12] J. Shlens, “A Tutorial on Principal Component Analysis,” 2014, [Online]. Available: <http://arxiv.org/abs/1404.1100>
- [13] I. A. M. Supartini, I. K. G. SUKARSA, and I. G. A. M. SRINADI, “Analisis Diskriminan Pada Klasifikasi Desa Di Kabupaten Tabanan Menggunakan Metode K-Fold Cross Validation,” *E-Jurnal Mat.*, vol. 6, no. 2, p. 106, 2017, doi: 10.24843/mtk.2017.v06.i02.p154.
- [14] F. S. Jumeilah, “Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 1, pp. 19–25, 2017, doi: 10.29207/resti.v1i1.11.
- [15] N. M. Patil and M. U. Nemade, “Music Genre Classification Using MFCC, K-NN and SVM Classifier,” *Int. J. Comput. Eng. Res. Trends*, vol. 4, no. 2, pp. 2349–7084, 2017, [Online]. Available: [www.ijcert.org](http://www.ijcert.org)