KOMPARASI PERFORMA MODEL KLASIFIKASI EMOSI DENGAN WORD EMBEDDING MENGGUNAKAN ALGORITMA SVM DAN RANDOM FOREST

Muhammad Adam Rachman, Agussalim, Eka Dyar Wahyuni
Sistem Informasi, Universitas Pembangunan Nasional "Veteran" Jawa Timur
Jl. Raya Rungkut Madya, Gunung Anyar, Surabaya, Indonesia
muhammadadam1608@gmail.com

ABSTRAK

Beragamnya emosi dari masyarakat dalam merespon dinamika pemerintahan yang sedang berlangsung saat ini sudah berlangsung dari tahun ke tahun. Banyak informasi yang tidak tersampaikan dengan baik karena kurangnya pemahaman masyarakat terkait konteks emosional yang terkandung dalam informasi tersebut. Salah satu pendekatan yang dapat dilakukan adalah dengan melakukan sebuah pengelompokkan emosi. Penelitian ini bertujuan untuk membandingkan performa algoritma Support Vector Machine (SVM) dan Random Forest yang diintegrasikan dengan teknik *word embedding* dalam klasifikasi emosi pada komentar YouTube berbahasa Indonesia terkait kebijakan pemerintah. Data pada penelitian ini sebanyak 3074 data, dan akan dibangun total 16 skenario pemodelan. Proses klasifikasi terdiri dari dua tahap yaitu klasifikasi emosi dan klasifikasi jenis emosi. Penggunaan teknik sampling dan rasio pembagian data memberikan hasil yang bervariasi di setiap model. Model dengan performa paling optimal untuk klasifikasi emosi adalah Random Forest dengan Word2Vec dengan hasil akurasi 86%, sedangkan model klasifikasi jenis emosi dengan performa paling optimal adalah Support Vector Machine dengan FastText dengan nilai akurasi sebesar 77%. Word2Vec mampu menangkap hubungan semantis kata cukup baik walaupun jumlah dataset yang digunakan relatif kecil. Di satu sisi lain, FastText juga mampu mengimbangi performa Word2Vec karena kemampuannya memanfaatkan representasi berbasis subkata untuk menangani kata-kata yang tidak terdapat dalam korpus.

Kata kunci : *klasifikasi, emosi, machine learning, word embedding, FastText, Word2Vec*

1. PENDAHULUAN

Pada era perkembangan teknologi yang semakin canggih saat ini, gaya hidup manusia diiringi dengan media yang serba digital. Adanya media digital tersebut memungkinkan penyebaran opini publik secara masif dan lebih terbuka [1]. Setiap kebijakan yang dikeluarkan pemerintah selalu menjadi perbincangan hangat oleh masyarakat melalui media digital tersebut. Beragam respons muncul terhadap kebijakan pemerintah, mulai dari rasa senang, marah, hingga emosi lainnya. Beragamnya emosi dari masyarakat dalam merespon dinamika pemerintahan yang sedang berlangsung saat ini sudah berlangsung dari tahun ke tahun [2].

Emosi memiliki peran penting dalam proses komunikasi [3]. Emosi dapat diungkapkan melalui berbagai macam bentuk komunikasi seperti ucapan/lisan, ekspresi wajah, dan tulisan. Sering kali orang-orang masih menyalahartikan sebuah informasi karena mereka tidak memahami konteks emosi yang terkandung di dalamnya. Kurangnya pemahaman ini dapat menyebabkan kesalahpahaman dan konflik. Memahami dan menafsirkan emosi orang lain dapat menghasilkan interaksi sosial yang lebih baik, mengurangi konflik, dan meningkatkan kualitas hubungan sosial [4]. Terdapat banyak metode untuk dapat mengenali emosi dari sebuah teks, salah satunya adalah menggunakan *machine learning*.

Machine learning adalah salah satu cabang dari kecerdasan buatan yang berfokus pada pengembangan model yang memungkinkan komputer untuk membuat sebuah keputusan berdasarkan data yang diberikan [5]. Dalam penerapan *machine learning*, khususnya *Natural Language Processing*, memahami hubungan antarkata dianggap penting karena dapat meningkatkan performa model [6]. Banyak teknik yang dapat digunakan untuk memahami hubungan semantis antar kata dalam NLP, salah satu teknik yang cukup populer yaitu *word embedding*. Penerapan *word embedding* pada *text mining* dapat merepresentasikan mengenai hubungan antar kata dalam bentuk ruang vektor berkelanjutan [6].

Berdasarkan latar belakang yang telah diuraikan sebelumnya, penelitian ini akan mengaplikasikan algoritma *machine learning* Support Vector Machine dan Random Forest dengan tujuan untuk mengetahui perbandingan hasil performa kedua algoritma dengan teknik *word embedding* untuk mengklasifikasikan emosi pada komentar YouTube berbahasa indonesia mengenai topik kebijakan pemerintah. Perbandingan kedua metode ini dilakukan karena masing-masing algoritma memiliki karakteristik yang berbeda dalam menangani data teks. Support Vector Machine dikenal unggul dalam memisahkan data pada ruang dimensi tinggi dengan memanfaatkan margin optimal dan secara konsisten mendapatkan performa terbaik dalam mengklasifikasikan teks [7]. Di sisi lain, Random Forest memiliki keunggulan dalam menangani data yang kompleks dan berdimensi tinggi [8]. SVM dan RF menghasilkan performa yang cukup baik untuk melakukan tugas klasifikasi teks dengan data yang tidak seimbang [9]. Dengan melakukan komparasi, penelitian ini bertujuan untuk menentukan algoritma dan ekstraksi fitur *word embedding* mana yang paling

optimal dalam menangkap hubungan semantis teks serta menghasilkan model klasifikasi emosi yang lebih akurat.

2. TINJAUAN PUSTAKA

2.1. *Text Mining*

Text mining adalah sebuah proses untuk menemukan dan mengekstrak informasi penting dan menarik dari teks yang tidak terstruktur seperti dokumen dan *website*. Proses ini mencakup berbagai teknik, mulai dari pengambilan informasi, klasifikasi teks, clustering, hingga ekstraksi entitas, relasi, dan peristiwa [10]. Text mining sering digunakan untuk mengkategorikan dan menganalisis sentimen dari data teks yang besar. Teknik ini berguna untuk mengidentifikasi wawasan dan pola tren dalam data tersebut. Beberapa perusahaan menggunakan text mining untuk menganalisis pendapat, emosi, dan sentimen masyarakat terhadap merk dan produk mereka [11].

2.2. Word2vec

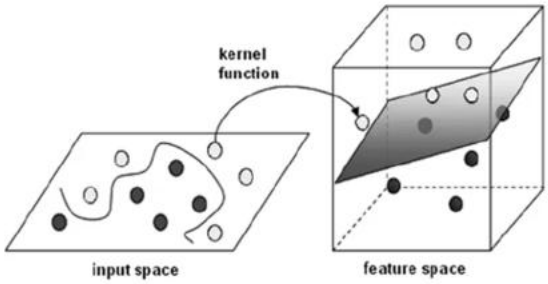
Teknik *word vector* menjadi solusi baru dalam merepresentasikan kata-kata di bidang *Natural Language Processing* (NLP). *Word vector* dibangun menggunakan bantuan jaringan saraf tiruan. Teknik ini diusulkan dengan nama Word2Vec [12]. Word2Vec adalah jaringan saraf yang mempelajari pengubahan bentuk kata dari teks menjadi bentuk vektor. Model Word2vec mempelajari representasi vektor dari kata-kata, yang membawa makna semantik dan berguna dalam berbagai tugas NLP [13].

2.3. Fastext

FastText adalah metode *word embedding* yang merupakan pengembangan dari Word2Vec. Metode ini mewakili setiap kata sebagai sekumpulan karakter *n-gram*, sehingga efektif untuk menangani kata-kata yang jarang dan di luar kosa kata [14]. FastText terbukti mampu mengungguli model TF-IDF tradisional dengan memerlukan lebih sedikit langkah praproses sambil mempertahankan performa yang tinggi [15].

2.4. Support Vector Machine

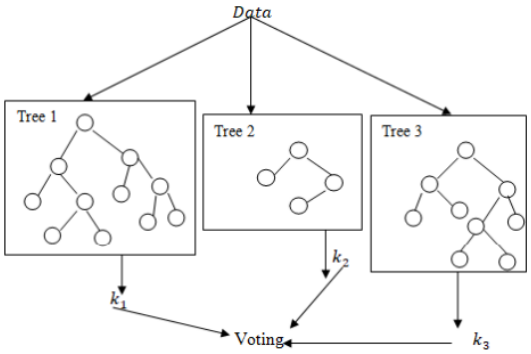
Support Vector Machine (SVM) adalah bagian dari pembelajaran mesin yang bekerja dengan cara menemukan *hyperplane* terbaik yang berguna untuk memisahkan dua buah kelas pada input. Metode SVM dibagi menjadi dua, yaitu SVM linier dan SVM nonlinier. SVM linier merupakan data yang dipisahkan secara linier, yaitu memisahkan kedua class pada *hyperplane* dengan margin maksimum, sedangkan SVM nonlinier yaitu menerapkan fungsi dari *kernel* terhadap ruang yang berdimensi lebih tinggi [16].



Gambar 1. Ilustrasi *hyperplane* SVM

2.5. Random Forest

Random Forest merupakan metode klasifikasi yang berdasarkan oleh pohon keputusan atau *decision tree*. Setiap pohon keputusan dibuat dengan memilih variabel secara acak untuk melakukan pada setiap cabang-cabangnya. Metode ini membantu meningkatkan akurasi dan mengurangi kemungkinan *overfitting* karena keputusan akhir didasarkan pada berbagai pohon, bukan hanya satu [17].



Gambar 2. Ilustrasi struktur Random Forest

2.6. *Confusion Matrix*

Confusion matrix atau bisa disebut juga matriks klasifikasi merupakan sebuah alat yang penting dalam menganalisis performa dari *machine learning*. *Confusion matrix* digunakan untuk mengevaluasi performa dari sebuah model klasifikasi [18]. *Confusion matrix* membandingkan hasil prediksi model dengan nilai sebenarnya dari data uji. Tabel ini terdiri dari empat bagian utama: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Dengan *confusion matrix*, kita dapat menghitung berbagai metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score* [19].

$$Accuracy = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN})} \quad (1)$$

Accuracy merepresentasikan sejauh mana model dapat mengklasifikasikan dengan tepat.

$$Precision = \frac{(\mathrm{TP})}{(\mathrm{TP} + \mathrm{FP})} \quad (2)$$

Precision merepresentasikan tingkat ketepatan antara data yang diminta dan hasil prediksi model.

$$Recall = \frac{(TP)}{(TP + FN)} \quad (3)$$

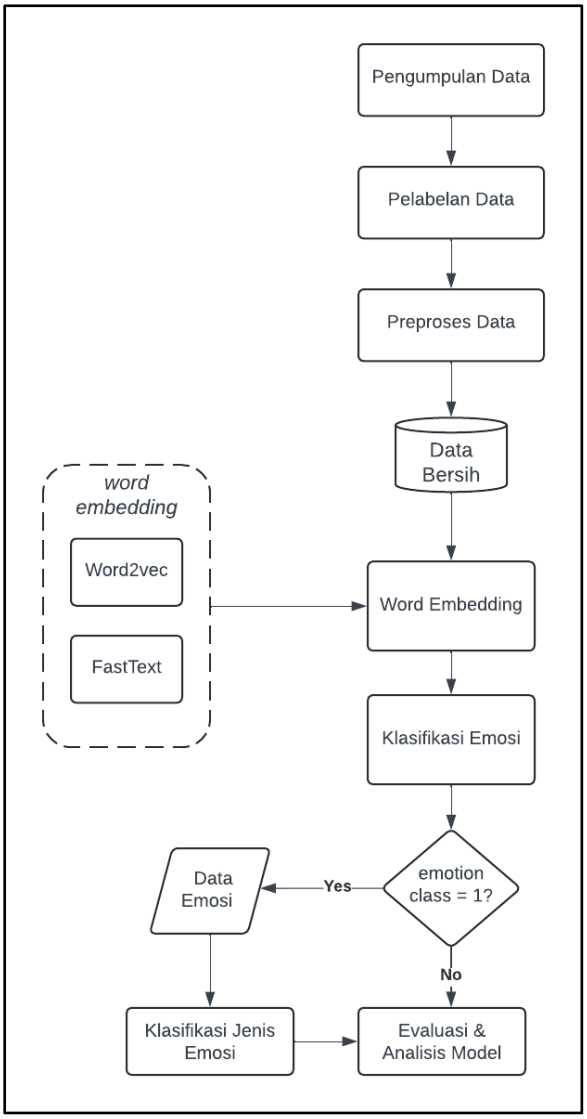
Recall merepresentasikan keberhasilan model dalam menemukan kembali informasi yang relevan.

$$F1\text{-}score = \frac{(2 \times Recall \times Precision)}{(Recall + Precision)} \quad (4)$$

F1-score merepresentasikan nilai rata-rata harmonik dari *Precision* dan *Recall*.

3. METODE PENELITIAN

Alur penelitian yang dilakukan ditunjukkan pada gambar diagram alir penelitian di bawah.



Gambar 3. Diagram alir penelitian

Pada setiap proses dalam Gambar 3 akan dijelaskan dalam poin-poin sebagai berikut

3.1. Pengumpulan Data

Data komentar pengguna YouTube pada konten video dengan topik kebijakan pemerintah diambil menggunakan bantuan YouTube Data API dan kemudian disimpan dalam bentuk *csv*.

3.2. Pelabelan Data

Pelabelan data pada penelitian ini akan digunakan lima jenis pelabel dengan bantuan *pre-trained* model klasifikasi emosi. Setelah didapatkan hasil label akan diambil modus label dari setiap class untuk digunakan sebagai label akhir. Jika tidak didapatkan modus dari proses tersebut maka akan dilakukan pelabelan ulang secara manual. Lima pelabel *pretrained* model tersebut adalah *michellejieli* [20], *finiteautomata* [21], *vamossyd* [22], *jhartman* [23], dan GPT-3.5 [24].

3.3. Praproses Data

Setelah melakukan pelabelan, proses berikutnya adalah *preprocessing* atau praproses data yang bertujuan untuk memfilter data sehingga menghasilkan data dalam bentuk yang lebih optimal dan terorganisir dengan baik. Tahapan pada praproses data ini yaitu *cleaning*, *case folding*, *normalization*, dan *stemming*.

3.4. Proses *Word Embedding*

Tahap selanjutnya akan dilakukan *word embedding* untuk mengubah setiap kata pada data latih menjadi nilai vektor agar dapat dipelajari oleh model. Ada dua metode *word embedding* yang akan digunakan yaitu Word2Vec dan FastText.

3.5. Klasifikasi Emosi

Pada tahap ini akan dilakukan proses klasifikasi teks yang mengandung emosi atau tidak. Pelatihan model menggunakan data latih yang sebelumnya telah dilabeli. Hasil pada proses ini yaitu *class* yang mengandung emosi nantinya akan digunakan sebagai data model pada tahap klasifikasi kedua.

3.6. Klasifikasi Jenis Emosi

Pada tahap ini akan dilakukan pengelompokkan emosi berdasarkan enam *class* yaitu *Joy*, *Sadness*, *Anger*, *Fear*, *Surprise*, dan *Disgust*. Data yang digunakan adalah data yang mengandung emosi dari tahap klasifikasi emosi sebelumnya.

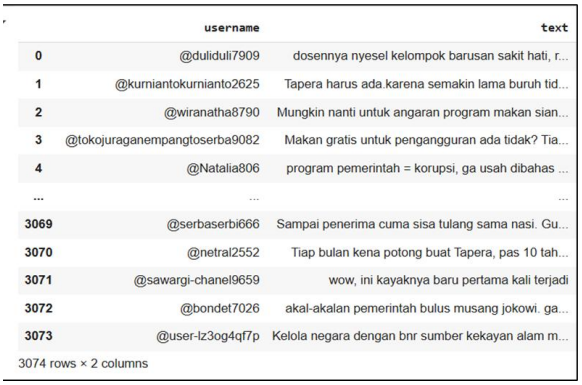
3.7. Evaluasi Model

Hasil pengujian dari data uji akan ditampilkan dalam bentuk nilai tertentu yang merepresentasikan performa model yang telah dilatih terhadap atribut yang telah ditetapkan sebelumnya. Tahap evaluasi model ini melibatkan *confusion matrix*, CPU *usage*, dan *processing time*.

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Pengumpulan data komentar ini yang dibutuhkan adalah *id* dari video YouTube yang mengandung topik kebijakan pemerintah, dan API *key* dari Youtube Data API v3. Data yang diambil dari proses *crawling* tersebut meliputi nama akun yang bertipe string untuk menunjukkan siapa akun yang membuat komentar, dan komentar dari akun tersebut.



	username	text
0	@duliduli7909	dosennya nyesel kelompok barusan sakit hati, r...
1	@kurniantokurnianto2625	Tapera harus ada karena semakin lama buruh tid...
2	@wiranatha8790	Mungkin nanti untuk angaran program makan sian...
3	@tokojuraganempangtoserba9082	Makan gratis untuk pengangguran ada tidak? Tia...
4	@Natalia806	program pemerintah = korupsi, ga usah dibahas ...
...	...	...
3069	@serbaserbi666	Sampai penerima cuma sisa tulang sama nasi. Gu...
3070	@netral2552	Tiap bulan kena potong buat Tapera, pas 10 tah...
3071	@sawargi-chanel9659	wow, ini kayaknya baru pertama kali terjadi
3072	@bondet7026	akal-akalan pemerintah bulus musang jokowi. ga...
3073	@user-lz3og4qf7p	Kelola negara dengan bnr sumber kekayan alam m...

3074 rows × 2 columns

Gambar 4. Hasil pengumpulan data

Proses pengumpulan data pada Gambar 4, data diambil berdasarkan *video_id* pada masing-masing video sehingga dilakukan pengambilan beberapa pengumpulan *video_id* untuk memperoleh dataset penelitian. Sebanyak 3074 data akan digunakan dalam penelitian ini.

4.2. Pelabelan Data

Data komentar sebelumnya akan diterjemahkan ke dalam bahasa Inggris terlebih dahulu sebagai bentuk perantara sementara karena pelabel yang digunakan berbasis bahasa Inggris. Kemudian akan dilakukan pelabelan dengan *pretrained* model untuk menghasilkan lima jenis label yang dapat dilihat pada gambar 5 di bawah.

	text	michelle	vamos	finite	jhartman_distilr	gpt3.5
0	dosennya nyesel kelompok barusan sakit hati, r...	sadness	Sad	sadness	sadness	Sadness
1	Tapera harus ada karena semakin lama buruh tid...	neutral	Neutral	others	neutral	Neutral
2	Mungkin nanti untuk angaran program makan sian...	neutral	Neutral	others	neutral	Neutral
3	Makan gratis untuk pengangguran ada tidak? Tia...	fear	Fear	others	fear	Sadness
4	program pemerintah = korupsi, ga usah dibahas ...	neutral	Disgust	others	neutral	Disgust
...	...	...	...	...	...	...

Gambar 5. Hasil pelabelan data

Setiap pelabel akan menghasilkan tujuh label yaitu "*Joy*", "*Sadness*", "*Fear*", "*Disgust*", "*Anger*", "*Surprise*", dan "*Neutral*". Hasil fiksasi label pada setiap tahap klasifikasi akan ditentukan secara objektif dengan diambilnya modus dari label yang diberikan pelabel. Jika tidak didapatkan modus atau terjadi *tie-breaking* maka model akan ditimbang dengan label penulis secara manual. Label pada klasifikasi pertama yaitu "*Yes*" dan "*No*". Label "*Yes*" didapatkan dari proses *mapping* label yang ditentukan berdasarkan label "*Joy*", "*Sadness*", "*Fear*", "*Disgust*", "*Anger*", "*Surprise*", sedangkan label "No" didapatkan dari proses *mapping* label yang ditentukan berdasarkan label "*Neutral*".

4.3. Praproses Data

Setelah data dilabeli, selanjutnya akan dilakukan tahap praproses data atau menyiapkan data sehingga akan menghasilkan data bersih yang siap digunakan untuk proses berikutnya.

Tahapan praproses data dalam penelitian ini yaitu *cleaning*, *case folding*, *normalization*, dan *stemming*. Tahap praproses data diawali dengan melakukan pembersihan data atau cleaning seperti penanganan *missing value*, menghapus emoji, menghapus elemen html, menghapus *mention* akun, menghapus *url link*, dan lainnya. Selanjutnya akan dilakukan proses *case folding* atau menyeragamkan setiap data menjadi huruf kecil semua. Setelah itu, akan dilakukan normalisasi data seperti menangani permasalahan kata yang typo atau kurang benar penulisannya dan juga penanganan terhadap kata yang disingkat. Terakhir *stemming* atau mengubah bentuk kata yang memiliki imbuhan baik imbuhan awalan, imbuhan akhiran, imbuhan sisipan, maupun imbuhan kombinasi awal dan akhir menjadi bentuk kata dasarnya. Contoh praposes data dapat dilihat pada pada tabel 1.

Tabel 1. Contoh Hasil dari tahap praproses

Text	Cleaned-text
Mudah-mudahan ada demo besar buat gagalin proyek korupsi lg udh muak ga tahan lagi sama pemerintahan jokowi ini	mudah mudah ada demo besar buat gagal proyek korupsi lagi sudah muak tidak tahan lagi sama perintah jokowi ini

4.4. Proses Word Embedding

Tahap selanjutnya akan dilakukan proses *word embedding* agar data dapat dipelajari model ketika proses pemodelan nanti. Proses ini menggunakan metode Word2vec dan FastText untuk mengubah kata menjadi vektor numerik yang merepresentasikan kata. Proses *word embedding* tersebut akan diterapkan pada setiap data yang telah dibagi untuk beberapa macam skenario pemodelan. Detail setiap skenario pemodelan dapat dilihat pada tabel 2 di bawah

Tabel 2. Jenis skenario pemodelan

Nama Model	Deskripsi
Model 1	SVM dengan Word2Vec
Model 2	SVM dengan FastText
Model 3	Random Forest dengan Word2Vec
Model 5	Random Forest dengan FastText

Tabel 2 berisi detail empat pemodelan yang akan dibangun. Total empat model tersebut akan dijalankan pada setiap tahap klasifikasi. Setiap model akan menjalankan skenario data normal dan RUS, dengan proporsi 80:20 dan 70:30, sehingga didapatkan total sebanyak 16 skenario pemodelan untuk setiap tahap klasifikasi.

4.5. Klasifikasi Emosi

Proses klasifikasi ini akan mendeteksi apakah sebuah teks mengandung emosi atau tidak. Pembangunan model menggunakan konfigurasi parameter yang seragam dan telah disesuaikan untuk menghasilkan kinerja yang paling optimal. Lima model dengan hasil performa terbaik akan diambil untuk memudahkan proses seleksi dalam menentukan satu model terbaik secara keseluruhan.

Tabel 3. Lima model terbaik klasifikasi emosi

Klasifikasi Emosi					
Indikator	**Model 1 Normal 80:20**	**Model 1 RUS 80:20**	**Model 3 Normal 80:20**	**Model 3 RUS 80:20**	**Model 4 RUS 80:20**
Train Accuracy	0.89%	0.89%	0.96%	0.96%	0.96%
Test Accuracy	0.85%	0.85%	0.86%	0.85%	0.86%
Precision	0.85%	0.85%	0.86%	0.85%	0.85%
Recall	0.85%	0.86%	0.85%	0.85%	0.85%
F1-score	0.86%	0.86%	0.86%	0.86%	0.85%
CPU *usage*	0.8%	0.8%	0.8%	0.8%	0.8%
Processing time	4.02 detik	3.70 detik	2.31 detik	5.48 detik	1.26 detik
Model status	*balanced*	*balanced*	*balanced*	*balanced*	*balanced*

Berdasarkan tabel 3 di atas, didapatkan beberapa kesimpulan berikut. Lima skenario model terbaik untuk proses klasifikasi emosi yaitu Model 1 Normal 80:20, Model 1 RUS 80:20, Model 3 Normal 80:20, Model 3 RUS 80:20, dan Model 4 RUS 80:20. Kelima model tersebut menghasilkan akurasi tertinggi dari 16 skenario yang dilakukan yaitu sebesar 85% hingga 86%. Waktu pemrosesan untuk lima model terbaik di atas relatif cepat karena hanya berkisar antara 1 detik hingga 6 detik, sedangkan penggunaan CPU juga menunjukkan hasil yang cukup rendah yaitu sebesar 0.8%. Kelima model yang dihasilkan tidak mengindikasikan terjadinya *overfitting/underfitting* karena perbedaan akurasi latihan dan pengujian cukup rendah sehingga dapat dikatakan kelima model tersebut memiliki status *balanced*.

4.6. Klasifikasi Jenis Emosi

Proses klasifikasi jenis emosi ini menggunakan data dari proses klasifikasi sebelumnya yang memiliki *class* emosi. Proses klasifikasi ini akan mengelompokkan emosi berdasarkan jenisnya ke dalam enam *class* yaitu *Joy*, *Sadness*, *Anger*, *Fear*, *Surprise*, dan *Disgust*. Lima model dengan hasil performa terbaik akan diambil untuk memudahkan proses seleksi dalam menentukan satu model terbaik secara keseluruhan.

Tabel 4. Lima model terbaik klasifikasi jenis emosi

Klasifikasi Jenis Emosi					
Indikator	**Model 1 Normal 80:20**	**Model 2 Normal 80:20**	**Model 2 Normal 70:30**	**Model 2 RUS 80:20**	**Model 4 Normal 80:20**
Train Accuracy	0.97%	0.84%	0.84%	0.84%	1.00%
Test Accuracy	0.76%	0.77%	0.75%	0.75%	0.76%
Precision	0.76%	0.78%	0.76%	0.76%	0.77%
Recall	0.76%	0.77%	0.75%	0.75%	0.72%
F1-score	0.76%	0.77%	0.75%	0.75%	0.72%
CPU *usage*	0.8%	0.8%	0.4%	0.8%	0.8%
Processing time	2.22 detik	0.68 detik	0.97 detik	0.15 detik	3.28 detik
Model status	*overfit*	*balanced*	*balanced*	*balanced*	*overfit*

Berdasarkan tabel 4 di atas, didapatkan beberapa kesimpulan berikut. Lima skenario model terbaik untuk proses klasifikasi jenis emosi yaitu Model 1 Normal 80:20, Model 2 Normal 80:20, Model 2 Normal 70:30, Model 2 RUS 80:20, dan Model 4 Normal 80:20. Kelima model tersebut menghasilkan akurasi tertinggi dari 16 skenario yang dilakukan yaitu sebesar 75% hingga 77%. Waktu pemrosesan untuk lima model terbaik di atas relatif cepat karena hanya berkisar antara 1 detik hingga 4 detik, sedangkan penggunaan CPU juga menunjukkan hasil yang relatif rendah yaitu sebesar 0.8%. Dari lima model tersebut, Model 1 Normal 80:20 dan Model 4 Normal 80:20 mengindikasikan terjadi *overfitting* karena nilai akurasi pelatihan dan pengujian memiliki selisih yang cukup signifikan. Pada sisi lain, Model 2 Normal 80:20, Model 2 Normal 70:30, dan Model 2 RUS 80:20 mengindikasikan status model *balanced* karena perbedaan akurasi latihan dan pengujian cukup rendah.

4.7. Evaluasi Model

Pada tahap klasifikasi emosi, Model 3 Normal 80:20 dengan waktu pemrosesan selama 2.31 detik dan Model 4 RUS 80:20 dengan waktu pemrosesan selama 1.26 detik memiliki nilai akurasi yang sama yaitu sebesar 86%. Akan tetapi, nilai *F1-score* untuk Model 3 Normal 80:20 tersebut lebih tinggi dibandingkan Model 4 RUS 80:20 yaitu sebesar 86%. Sehingga dapat disimpulkan bahwa model terbaik untuk klasifikasi emosi adalah Random Forest dengan ekstraksi fitur Word2Vec rasio pembagian data 80:20.

Pada tahap klasifikasi jenis emosi, Model 2 Normal 80:20 memiliki hasil akurasi dan F1-score yang paling tinggi diantara model lain yaitu sebesar 77%, serta model tersebut juga terindikasi model yang seimbang. Oleh karena itu, dapat disimpulkan bahwa model terbaik untuk klasifikasi jenis emosi adalah Support Vector Machine dengan ekstraksi fitur FastText rasio pembagian data 80:20.

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa pembuatan model klasifikasi emosi dengan menggunakan teknik *word embedding* menghasilkan performa yang cukup bagus. Secara keseluruhan, terdapat beberapa model yang terindikasi mengalami *overfitting*, namun mayoritas model yang dibangun memiliki status *balanced*. Hasil performa yang ditunjukkan melalui metrik evaluasi yang digunakan menunjukkan hasil dengan selisih yang cukup dekat satu sama lain pada setiap proses klasifikasi. Model dengan performa paling optimal

untuk klasifikasi emosi adalah Random Forest Normal 80:20 dengan nilai akurasi sebesar 0.86, sedangkan model dengan performa paling optimal untuk klasifikasi jenis emosi adalah SVM Normal 80:20 dengan nilai akurasi sebesar 0.77. Selain itu, penerapan teknik word embedding Word2Vec dan FastText menghasilkan performa yang saling kompetitif dengan dibuktikannya hasil yang tidak terlalu signifikan diantara keduanya. Word2Vec mampu menangkap hubungan semantis kata cukup baik walaupun jumlah dataset yang digunakan relatif kecil. Di satu sisi lain, FastText juga mampu mengimbangi performa Word2Vec karena kemampuannya memanfaatkan representasi berbasis subkata untuk menangani kata-kata yang tidak terdapat dalam korpus. Saran yang dapat ditambahkan untuk penelitian selanjutnya adalah karena penelitian ini hanya menggunakan teknik *word embedding* langsung pada teks tanpa kombinasi dengan metode ekstraksi fitur lainnya, penambahan kombinasi ekstraksi fitur teknik *word embedding* seperti teknik *n-grams*, POS *tagging*, dan semacamnya dapat dipertimbangkan untuk memperoleh dimensi fitur yang lebih besar. Selain itu, juga dapat dipertimbangkan untuk mengambil makna semantis dari *emoticon*, penggunaan karakter yang berulang seperti "kesalll!!!" yang menandakan adanya intensitas emosi sehingga dapat menghasilkan model klasifikasi emosi yang lebih akurat.

DAFTAR PUSTAKA

[1] E. Dubois, A. Gruzd, and J. Jacobson, "Journalists' Use of Social Media to Infer Public Opinion: The Citizens' Perspective," *Soc Sci Comput Rev*, vol. 38, no. 1, pp. 57–74, Feb. 2020, doi: 10.1177/0894439318791527.

[2] K. Arifin and S. I. Al-Idrus, "Klasifikasi Emosi Pengguna Twitter Terhadap Bakal Calon Presiden Pada Pemilu 2024 Menggunakan Algoritma Naïve Bayes," *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, vol. 23, no. 1, p. 37, Feb. 2024, doi: 10.53513/jis.v23i1.9558.

[3] M. N. Shiota, *Basic and Discrete Emotion Theories*, 1st Edition. Routledge, 2024.

[4] M. Khosla, "Understanding Micro-momentary Emotional Expressions: A Life-Span Perspective," 2024, pp. 109–127. doi: 10.1007/978-3-031-46349-5_7.

[5] S. U. Hassan, J. Ahamed, and K. Ahmad, "Analytics of machine learning-based algorithms for text classification," *Sustainable Operations and Computers*, vol. 3, pp. 238–248, 2022, doi: 10.1016/j.susoc.2022.03.001.

[6] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurr Comput*, vol. 33, no. 23, Dec. 2021, doi: 10.1002/cpe.5909.

[7] D. E. Cahyani and I. Patasik, "Performance comparison of tf-idf and word2vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021, doi: 10.11591/eei.v10i5.3157.

[8] N. Jalal, A. Mehmood, G. S. Choi, and I. Ashraf, "A novel improved random forest for text classification using feature ranking and optimal number of trees," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2733–2742, Jun. 2022, doi: 10.1016/j.jksuci.2022.03.012.

[9] A. H. Salman and W. A. M. Al-Jawher, "Performance Comparison of Support Vector Machines, AdaBoost, and Random Forest for Sentiment Text Analysis and Classification," *Journal Port Science Research*, vol. 7, no. 3, pp. 300–311, Sep. 2024, doi: 10.36371/port.2024.3.8.

[10] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *J Big Data*, vol. 8, no. 1, p. 140, Oct. 2021, doi: 10.1186/s40537-021-00516-9.

[11] M. S. A. Rifky, A. A. Arifiyanti, and R. Permatasari, "Analisis Sentimen Pengguna Youtube Mengenai Analog Switch Off Menggunakan Word Embedding Dan Metode Long Short-Term," *Jurnal Teknik Mesin, Industri, Elektro Dan Informatika (JTMEI)*, vol. 2, no. 3, pp. 195–202, 2023, [Online]. Available: https://doi.org/10.55606/jtmei.v2i3.2273

[12] W. X. Zhao *et al.*, "A Survey of Large Language Models," Mar. 2023.

[13] E. Rodman, "A Timely Intervention: Tracking the Changing Meanings of Political Concepts with Word Vectors," *Political Analysis*, vol. 28, no. 1, pp. 87–111, Jan. 2020, doi: 10.1017/pan.2019.23.

[14] S. Z. Salsabiila, H. Nurramdhani Irmanda, and A. Arista, "Comparison of Fasttext and Word2Vec Weighting Techniques for Classification of Multiclass Emotions Using the Conv-LSTM Method," in *2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS)*, IEEE, Nov. 2023, pp. 125–130. doi: 10.1109/ICIMCIS60089.2023.10349034.

[15] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, "An Efficient Text Classification Using fastText for Bahasa Indonesia Documents Classification," in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, IEEE, Jul. 2020, pp. 69–75. doi: 10.1109/DATABIA50434.2020.9190447.

[16] A. F. Rochim, K. Widyaningrum, and D. Eridani, "Performance Comparison of Support Vector Machine Kernel Functions in Classifying COVID-19 Sentiment," in *2021 4th International Seminar on Research of Information Technology*

and Intelligent Systems (ISRITI), IEEE, Dec. 2021, pp. 224–228. doi: 10.1109/ISRITI54043.2021.9702845.

[17] Y. Sun, Y. Li, Q. Zeng, and Y. Bian, "Application Research of Text Classification Based on Random Forest Algorithm," in *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, IEEE, Apr. 2020, pp. 370–374. doi: 10.1109/AEMCSE50948.2020.00086.

[18] K. Riehl, M. Neunteufel, and M. Hemberg, "Hierarchical confusion matrix for classification performance evaluation," *J R Stat Soc Ser C Appl Stat*, vol. 72, no. 5, pp. 1394–1412, Dec. 2023, doi: 10.1093/jrsssc/qlad057.

[19] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput Sci*, vol. 2, no. 3, p. 160, May 2021, doi: 10.1007/s42979-021-00592-x.

[20] M. Jieli, "Fine-tuned DistilRoBERTa-base for Emotion Classification," https://huggingface.co/michellejieli/emotion_text_classifier.

[21] J. M. Pérez *et al.*, "pysentimiento: A Python Toolkit for Opinion Mining and Social NLP tasks," Jun. 2021.

[22] D. F. Vamossy and R. P. Skog, "EmTract: Extracting emotions from social media," *International Review of Financial Analysis*, vol. 97, p. 103769, Jan. 2025, doi: 10.1016/j.irfa.2024.103769.

[23] J. Hartman, "Emotion English DistilRoBERTa-base," https://huggingface.co/j-hartmann/emotion-english-distilroberta-base.

[24] OpenAI, "GPT-3.5 Model Overview," OpenAI API