Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik2024118010
Vol. 11, No. 5, Oktober 2024, hlm. 1067-1076 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

KLASIFIKASI TINGKAT STRES DARI DATA BERBENTUK TEKS DENGAN MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE* (SVM) DAN *RANDOM FOREST*

Naufal Fathirachman Mahing[*1], Alifi Lazuardi Gunawan[2], Ahmad Foresta Azhar Zen[3], Fitra Abdurrachman Bachtiar[4], Satrio Agung Wicaksono[5]

[1,2,3,4,5]Universitas Brawijaya, Malang
Email: [1] naufalmahing@student.ub.ac.id, [2]alifilazuardi@student.ub.ac.id, [3]ahmadforesta@student.ub.ac.id, [4]fitra.bachtiar@ub.ac.id, [5]satrio@ub.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 09 November 2023, diterima untuk diterbitkan: 30 Oktober 2024)

Abstrak

Stres merupakan keadaan dimana seseorang merasakan adanya tekanan yang berlebih pada dirinya. Pemantauan tingkat stres menjadi hal yang penting bagi manusia. Tingkat stres yang tinggi dapat menimbulkan dampak negatif terhadap kesehatan manusia. Deteksi dini stres menjadi sesuatu yang sangat penting untuk dilakukan. Salah satu cara mengetahui tingkat stres seseorang adalah melalui analisis teks. Penelitian ini dilakukan untuk melakukan klasifikasi tingkat stres berdasarkan data berupa teks menggunakan algoritma Support Vector Machine (SVM) dan Random Forest. Pada penelitian ini melakukan perbandingan beberapa metode transformasi. Transformasi yang dilakukan pada penelitian ini menggunakan TF-IDF, CountVectorizer, NRCLex, dan Word Affect Intensities. Data yang digunakan dalam penelitian ini berupa sebuat teks berbahasa Inggris yang diambil dari media sosial Twitter. Total data yang digunakan yaitu 8439 data. Pelatihan model baik untuk Support Vector Machine dan Random Forest menggunakan 6751 data. Sedangkan untuk pengujian menggunakan 1688 data. Hasil penelitian menunjukkan bahwa algoritma SVM dengan pembobotan menggunakan TF-IDF memiliki performa yang paling baik dibandingkan dengan algoritma Random Forest dan metode transformasi lainnya yang digunakan dalam penelitian. Model algoritma SVM dengan transformasi TF-IDF yang dibangun berhasil mendapatkan akurasi sebesar 84%. Model ini mendapatkan akurasi yang lebih tinggi dibanding model Random Forest yang memperoleh akurasi tinggi sebesar 80% dengan menggunakan transformasi CountVectorizer.

Kata kunci: *stres, SVM, random forest, klasifikasi teks, word affect intensities*

CLASSIFICATION OF STRESS LEVELS FROM TEXT DATA USING SUPPORT VECTOR MACHINE (SVM) ALGORITHM AND RANDOM FOREST

Abstract

Stress is a condition where a person feels excessive pressure on himself. Monitoring stress levels is important for humans. High levels of stress can have a negative impact on human health. Early detection of stress is something that is very important to do. One way to find out someone's stress level is through text analysis.This research was conducted to classify stress levels based on text data using the Support Vector Machine (SVM) and Random Forest algorithms. This research compares several transformation methods. The transformation performed in this study uses TF-IDF, CountVectorizer, NRCLex, and Word Affect Intensities. The data used in this research is an English text taken from Twitter social media. The total data used is 8439 data. Model training for both Support Vector Machine and Random Forest uses 6751 data. While for testing using 1688 data. The results showed that the SVM algorithm with weighting using TF-IDF had the best performance compared to the Random Forest algorithm and other transformation methods used in the study. The SVM algorithm model with TF-IDF transformation that was built managed to get an accuracy of 84%. This model obtained a higher accuracy than the Random Forest model which obtained a high accuracy of 80% using the CountVectorizer transformation.

Keywords: *stress, SVM, Random Forest, text classification, word affect intensities*

1. PENDAHULUAN

Tekanan psikologis pada manusia baik yang sadar atau tidak sadar, dari dalam maupun dari luar disebut dengan stres (Perangin-Angin dan Bachtiar, 2021). Tingkat stres merupakan hal yang penting bagi makhluk hidup, tidak terkecuali untuk manusia, jika

tingkat stres manusia tinggi dapat mengakibatkan penyakit. Persoalan tersebut sudah menunjukkan pentingnya untuk selalu sadar akan tingkat stres supaya selalu sehat. Deteksi dini stres menjadi sesuatu yang sangat penting untuk mencegah dampak lebih buruk yang dapat ditimbulkan. Dengan perkembangan ilmu komputer, deteksi dini stres dapat dilakukan dengan menggunakan berbagai metode. Metode tersebut antara lain dengan menggunakan analisis dari data *smart device*, data penggunaan *gadget*, ekspresi wajah, teks, dan berbagai metode lain (Munoz & Iglesias, 2022).

Cara lain untuk mengetahui tingkat stres adalah melalui teks. Deteksi stres dengan menggunakan teks dapat dilakukan dengan analisis sintaks dan fitur linguistik pada teks dengan menggunakan pendekatan *machine learning* atau pendekatan leksikal. Pendekatan leksikal dilakukan dengan mencocokkan teks yang akan diklasifikasikan dengan kamus yang telah dibuat sebelumnya untuk memberikan skor pada tiap kata berdasarkan bobot yang telah ditetapkan dan frekuensi kemunculannya. Pendekatan ini mampu menjangkau beberapa fitur dari teks seperti sentimen, emosi, dan topik yang terkandung (Munoz & Iglesias, 2022). Pendekatan lain yang digunakan adalah dengan melakukan *word embedding.* Beberapa teknik dari *word embedding* adalah *Word2Vec*, *GloVe*, dan *FastText*. Pendekatan *word embedding* mampu menangkap sintaksis dan semantik dari teks dengan baik.

Penggunaan metode pembobotan kata yang berbeda akan memberikan hasil yang berbeda. Penelitian yang dilakukan, membandingkan model *Random Forest* dan SVM dengan beberapa metode pembobotan kata, yaitu TF-IDF, *CountVectorizer*, NRCLex, dan *Word Affect Intensities*.

Berdasarkan permasalahan yang telah diuraikan, pada penelitian ini akan melakukan klasifikasi tingkat stres pada data teks dari aplikasi Twitter menggunakan metode *Support Vector Machine*(SVM) dan *Random Forest* dalam. Alasan dari metode SVM dipilih adalah karena pada penelitian sebelumnya (Pillai, Thelwall, dan Orasan, 2018) (Jadhav, dkk., 2019) metode tersebut diketahui memiliki performa yang baik dalam melakukan deteksi stres berdasar teks terutama pada dataset yang memiliki label. Untuk metode Random Forest dipilih karena pada penelitian lain Nijhawan, Attigeri, Ananthakrishna (2022) berpendapat metode *Random Forest* juga memiliki performa yang baik dalam klasifikasi sentimen.

Tujuan penelitian ini adalah membandingkan metode-metode preproses yang ada dengan fitur nilai afektif, dan melakukan analisis terhadap hasil yang didapatkan dari model.

1.1 Kajian Pustaka

Penelitian yang dilakukan oleh Rastogi, Liu, dan Cambria (2022), deteksi stres dilakukan pada teks yang berasal dari tweet pada aplikasi Reddit dan Twitter. Penelitian ini bertujuan untuk melakukan analisis deteksi stres dari teks sosial media. Dalam penelitian ini dilakukan perbandingan performa dari model *transformer-based*, *lexical-based*, dan *embedding-based*. Hasil dari eksperimen yang dilakukan merupakan model *transformer-based* yang mendapatkan akurasi yang paling tinggi.

Penelitian lain yang dikaji di penelitian ini adalah penelitian oleh Jadhav dkk. (2019), yang melakukan kajian dan analisis mengenai pendekatan yang digunakan dalam deteksi stres di sosial media dan berfokus pada data berupa teks. Analisis komparasi dilakukan pada tiga metode yaitu *Long Short Term Memory* (LSTM), *Bidirectional Long Short Term Memory* (BLSTM), dan *Support Vector Machine* (SVM). Dari analisis yang dilakukan, ditemukan bahwa pada dataset yang memiliki label, metode SVM memiliki hasil yang paling baik. Akan tetapi, pada kasus nyata dataset seringkali tidak memiliki label. Pada dataset tanpa label SVM tidak bisa melakukan deteksi dengan baik. LSTM dan BLSTM dapat melakukan deteksi dengan baik, baik dengan dataset berlabel maupun dengan dataset tanpa label. BLSTM memiliki performa yang lebih baik dibandingkan dengan LSTM karena bekerja secara dua arah.

Penelitian lain yang dilakukan oleh Nijhawan, Attigeri, dan Ananthakrishna (2022) bertujuan untuk melakukan analisis sentimen dan emosi pada teks dari berbagai sosial media. Analisis yang dilakukan menggunakan LDA (*latent dirichlet allocation*) untuk menentukan topik yang dibahas. Analisis emosi juga dilakukan dengan BERT model yang telah dilatih dengan *library* ktrain. Selanjutnya dilakukan klasifikasi menjadi lima kelas yakni *joy*, *sad*, *neutral*, *angry* dan *fear.* Hasilnya model dapat memperoleh akurasi sebesar 94%. Selanjutnya juga dilakukan analisis sentimen menjadi dua kelas yakni 0 yang menyatakan sentimen positif dan 1 yang berarti sentimen negatif. Dari beberapa algoritma yang digunakan, *Random Forest Classifier* memberikan hasil yang terbaik dengan akurasi mencapai 97,78%. Dengan keberhasilan klasifikasi sentimen ini dapat digunakan untuk melakukan kolaborasi dengan berbagai parameter kesehatan lain untuk digunakan sebagai deteksi dini kesehatan metal berbasiskan teks.

Pada penelitian sebelumnya (Risa, Pradana dan Bachtiar, 2021) telah dilakukan deteksi stres siswa berdasarkan tweet menggunakan algoritma *naive bayes.* Penelitian tersebut mengklasifikasikan tingkat stres menjadi tiga kelas yaitu kelas stres ringan, stres sedang dan stres berat. Penelitian tersebut mendapatkan hasil akurasi 75% dari 90 data latih dan 4 data uji. Deteksi stres tersebut kemudian diimplementasikan pada sistem monitoring berbasis web.

Pembeda dari penelitian ini penelitian yang sudah dikaji adalah metode preproses yang digunakan dari penelitian-penelitian tersebut, penelitian ini

memasukkan metode *Word Affect Intensities* yang diajukan oleh Saputra dkk. (2022) dalam perbandingan metode preproses. Pada penelitian ini juga dilakukan analisis pengaruh nilai *affective intensities* dengan model *machine learning* yang digunakan.

1.2 Stres

Stres dapat didefinisikan sebagai tekanan mental atau emosional yang timbul akibat keadaan yang tidak dapat dihindari atau biasa disebut sebagai *stressor*. Stres juga dapat merujuk pada ketegangan tertentu yang terjadi pada tubuh manusia akibat dari berbagai *stressor*. *Stressor* ini akan memicu pelepasan hormon stres dalam tubuh manusia (Strimpel, 1997).

2. METODE PENELITIAN

2.1 Diagram Alir

Diagram alir dari penelitian ini ditunjukkan pada Gambar 1. Pertama dimulai dari proses Pencarian dataset yang sesuai dengan topik yaitu dataset teks yang memiliki fitur *text*, *hashtags*, dan label dari Twitter. Tahap pertama yang pengumpulan data yang didapatkan dari github, dilanjutkan dengan *preprocessing* yang terdiri dari beberapa langkah, dilanjutkan dengan transformasi dari beberapa metode, dilanjutkan dengan *modeling* menggunakan SVM dan RF, model tersebut kemudian dievaluasi dengan akurasi, r*ecall*, *precision*, dan *F1-score*.

2.2 Dataset

Data yang digunakan merupakan data sekunder yang diambil dari Github. Dataset dapat dilihat pada link ini, Stress-Detection_Social-Media-Articles.

Tabel 1. Lima data pertama dataset

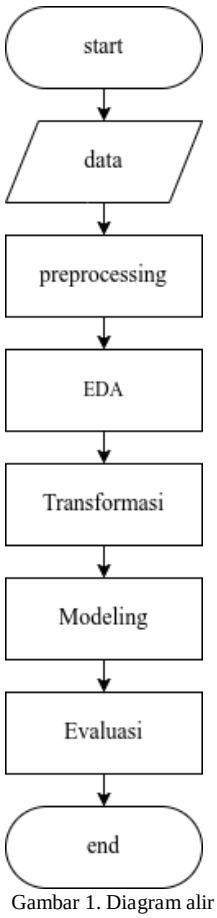
text	Hashtags	labels
Being s mom is cleaning 24/7 the …	['momlife', 'kids', 'tired']	1
And now we have been given…	['walkthru']	0
Wishing YOU Peace Joy…	['Peace', 'Joy', …	0
speak-no-evil monkey…	['therapy', 'help'...	1
Psy Do u hv any regrets?...	[]	0

2.3 Data Preprocessing

Proses *preprocessing* yang dilakukan adalah menghapus data duplikat yang berjumlah 461 dan melakukan drop terhadap fitur yang tidak digunakan yaitu *hashtags*. Fitur *hashtags* tidak digunakan karena pada penelitian ini berfokus untuk mendapatkan wawasan jika model hanya menggunakan fitur teks saja. *Preprocessing* terhadap teks juga dilakukan yang meliputi hal-hal berikut:

1. Mengubah teks menjadi *lowercase*
2. Menghapus kurung kotak ("[" dan "]") termasuk teks di dalamnya
3. Menghapus seluruh link
4. Menghapus tag html
5. Menghapus tanda baca
6. Menghapus *new line* ('\n')
7. Menghapus kata yang terdapat angka
8. Menghapus s*topword* seperti kata '*the*', '*and*', dan '*a*'
9. *Stemming*

Library yang digunakan untuk melakukan hal ini adalah *library* re dan *library* NLTK.



Gambar 1. Diagram alir

2.4 Exploratory Data Analysis (EDA)

Pada tahap *Exploratory data analysis* (EDA), dilakukan penggunaan metode *Word Cloud* untuk mengidentifikasi kata-kata yang sering muncul pada teks dengan kelas non stres maupun pada kelas stres. Langkah ini dilakukan untuk memberikan wawasan awal yang informatif, yang membantu identifikasi topik utama yang terkait dengan masing-masing kelas.

2.5 Modeling dan Pembobotan Kata

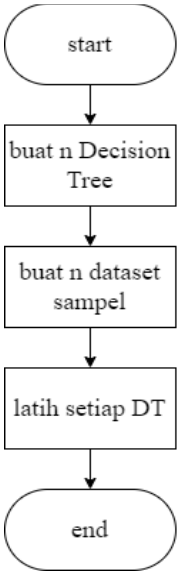
Terdapat empat metode transformasi yang digunakan di dalam penelitian ini, yaitu TF-IDF, CountVectorizer, NRCLex, dan *Word Affect Intensities*. TF-IDF atau *term frequency–inverse document frequency* melakukan transformasi dengan membandingkan kemunculan term dalam suatu dokumen. *CountVectorizer* menggunakan kemunculan sebuah term dalam kalimat untuk nilai

dari matriks. NRCLex menggunakan kamus National Research Council Canada (NRC) *affect lexicon* untuk menentukan emosi yang terkandung dalam suatu teks. *Word Affect Intensities* menggunakan transformasi dari NRCLex untuk membentuk fitur baru berupa emosi positif dan negatif. Model yang digunakan pada penelitian ini adalah Random Forest dan SVM. Random Forest sebagai metode utama, dan metode SVM sebagai metode pembanding. Data yang digunakan berjumlah 8439, dimana untuk pelatihan digunakan 6751 data dan pengujian menggunakan 1688 data. Pelatihan dari kedua model dilakukan dengan strategi K-Fold Cross Validation dengan jumlah *fold* 10.

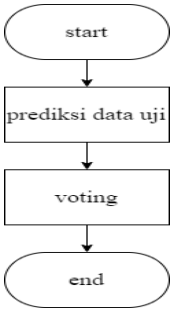
Model Random Forest

Random forest merupakan metode s*upervised learning* yang mengkombinasikan beberapa *decolcision tree acak* dan menggabungkan seluruh hasil prediksi tersebut (Biau & Scornet, 2016). Pada permasalahan regresi, hasil prediksi digabungkan dengan menghitung rata-rata, sedangkan pada permasalahan klasifikasi, hasil prediksi digabungkan dengan melakukan *majority vote*. Random forest memperbaiki karakter dari *decision tree* yang memiliki kecenderungan untuk *overfit* dengan data latih (Hastie, Tibshirani, & Friedman, 2009).

Random Forest merupakan salah satu metode *bagging*. Metode tersebut bekerja dengan membuat banyak model dari satu metode, yang kemudian dilatih dengan sampel-sampel dataset yang berbeda dari dataset yang sama. Setelah setiap model selesai dilatih, untuk memprediksi sebuah data, metode *bagging* menggunakan semua model untuk membuat prediksi masing-masing yang kemudian melakukan voting hasil prediksi yang paling banyak jumlahnya sebagai hasil prediksi akhir. Berdasarkan penjelasan oleh Liaw dan Wiener (2002), maka algoritma dari *Random Forest* dapat digambarkan pada Gambar 2 dan Gambar 3.



Gambar 2. *Flowchart* latihan *Random Forest*



Gambar 3. *Flowchart* prediksi *Random Forest*

Hyperparameter yang digunakan pada model *Random Forest* didapat melalui proses *GridSearch CV* untuk mencari *hyperparameter* terbaik. *Hyperparameter* yang dieksplorasi yaitu *n_estimators*: 200, 300, 400; *max_features*: *sqrt*, *log2*; *max_depth*: 5, 30, *None*; *criterion*: *gini*, *entropy*.

Model SVM

Support Vector Machine (SVM) merupakan metode pembelajaran *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan untuk menemukan *hyperplane* terbaik yang dapat memisahkan dua kelas pada ruang input (Nugroho, Witarto, Handoko, 2003). SVM merupakan salah satu metode yang relatif baru yang diusulkan oleh Vapnik dkk., (1995). SVM memiliki performa yang baik pada klasifikasi data dengan sampel besar, terutama untuk klasifikasi teks (Kwok,1998).

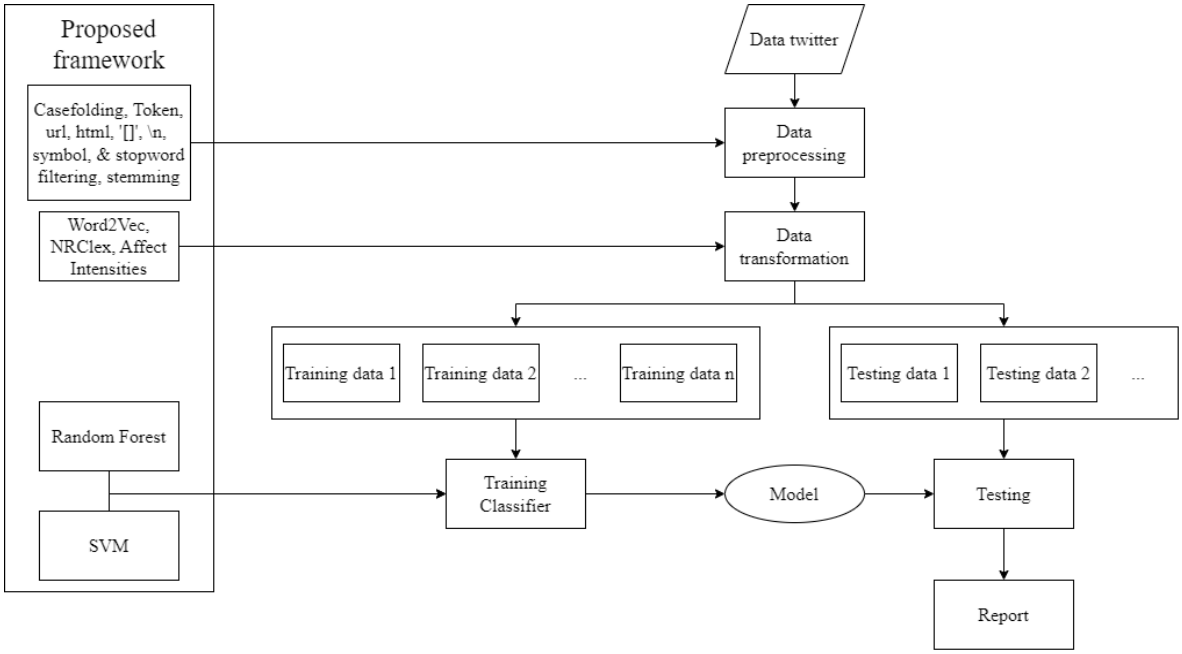
Tidak seperti *neural network* yang berusaha untuk mencari *hyperplane* pemisah antar class, SVM berusaha untuk mencari *hyperplane* terbaik pada ruang input. Prinsip dasar pada SVM adalah linear *classifier*. Kemudian dikembangkan lebih lanjut agar dapat menyelesaikan permasalahan non-linear dengan memanfaatkan konsep *kernel trick* pada ruang yang berdimensi tinggi.

SVM bekerja dengan memosisikan setiap data dalam sebuah peta berdimensi tinggi. Dengan memetakan setiap data dalam peta tersebut, diharapkan algoritma dapat memisahkan data-data tersebut menjadi dua kelas atau lebih dengan *hyperplane*. *Hyperplane* adalah sebuah garis yang memiliki satu dimensi lebih rendah dari dimensi peta yang digunakan untuk memetakan data. Untuk membuat *hyperplane* dapat digunakan beberapa fungsi yang biasa disebut dengan fungsi *kernel*. Garis besar dari algoritma SVM dapat dipahami seperti pada Gambar 5 dan Gambar 6.

Hyperparameter yang digunakan pada model SVM didapat melalui proses GridSearch CV untuk mencari *hyperparameter* terbaik. *Hyperparameter* yang dieksplorasi yaitu *kernel*: *linear*, *poly*, *rbf*, *sigmoid* dan C: 1, 2, 3.

2.6 Framework

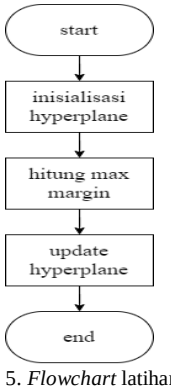
Framework yang kami gunakan dapat dilihat pada Gambar 4.



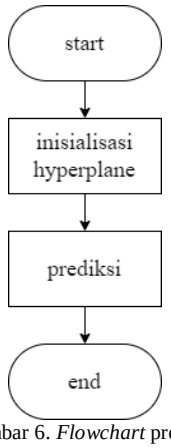
Gambar 4. *Framework* model

Framework model pada penelitian membagi data yang sudah dilakukan *preprocessing*, transformasi membagi data menjadi data *training* dan *testing* dengan ukuran data *testing* adalah 10% dari seluruh data. Data yang digunakan untuk *training* dibagi menjadi 10 bagian, dari setiap bagian tersebut dibagi menjadi data *training* dan *testing*, dengan ukuran data *testing* yang sama dari split pertama yaitu 20%. Data *training* digunakan untuk melatih *Random Forest* dan SVM menghasilkan model yang sudah di-*tuning* dan siap dievaluasi. Data *testing* dari hasil *split* pertama digunakan untuk tahap *testing*. Hasil prediksi kemudian digunakan untuk menghasilkan *report* dari performa model yang digunakan untuk evaluasi.

Library yang digunakan untuk *Framework* ini tersedia pada Sklearn, *Gensim*, dan NRCLex. Sklearn menyediakan fungsi yaitu *GridSearchCV*, *train_test_split*, *classification_report*, SVM, dan *RandomForestClassifier*. *Gensim* digunakan untuk menyediakan model Word2Vec. NRCLex digunakan untuk menyediakan fungsi untuk menghubungkan kata dengan emosi dari kata tersebut.



Gambar 5. *Flowchart* latihan *SVM*



Gambar 6. *Flowchart* prediksi *SVM*

2.7 Evaluasi Model

Evaluasi yang digunakan untuk mengukur model adalah akurasi, r*ecall*, *precision*, dan *F1-score*.

Recall adalah rumus yang digunakan untuk menghitung perbandingan hasil dari prediksi yang benar untuk label *positive* dan prediksi yang salah untuk label *negative*. Rumus untuk menghitung *recall* dapat dilihat di persamaan (1).

$$Recall = \frac{TP}{(TN + FN)} \times 100\% \quad (1)$$

$$TP = true\ positive$$
$$TN = true\ negative$$
$$FN = false\ negative$$

Precision sendiri mirip dengan *Recall*, tapi digunakan untuk menghitung perbandingan terhadap prediksi yang salah untuk label *positive*. Rumus untuk menghitung *precision* dapat dilihat di persamaan (2).

$$Precision = \frac{TP}{(TN + FP)} \times 100\% \quad (2)$$

$$TP = true\ positive$$
$$TN = true\ negative$$
$$FP = false\ positive$$

F1-score merupakan rata-rata harmonis dari *recall* dan *precision* yang sudah dijelaskan diatas. Pengukuran ini digunakan untuk menyederhanakan ukuran *recall* dan *precision* menjadi satu. Rumus untuk menghitung *F1-score* dapat dilihat di persamaan (3).

$$F1\text{-}score = \frac{2 \times Recall \times Precision}{(Recall + Precision)} \times 100\% \quad (3)$$

Akurasi memiliki rumus seperti pada persamaan (4) karena dengan menambahkan TP dan TN didapatkan jumlah hasil prediksi yang benar, dan menjumlahkan TP, TN, FP, dan FN merupakan jumlah dari seluruh prediksi yang dilakukan. Penjumlahan dari penyebut membuat persamaan membandingkan jumlah yang benar dengan jumlah seluruh prediksi.

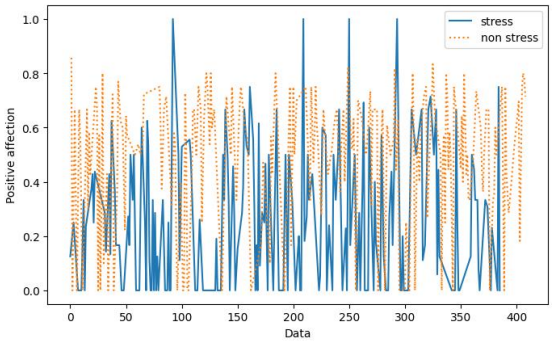
$$akurasi = \frac{TP + TN}{(TP+TN+FP+FN)} \times 100\% \quad (4)$$

$$TP = true\ positive$$
$$TN = true\ negative$$
$$FP = false\ positive$$
$$FN = false\ negative$$

3. HASIL EKSPERIMEN

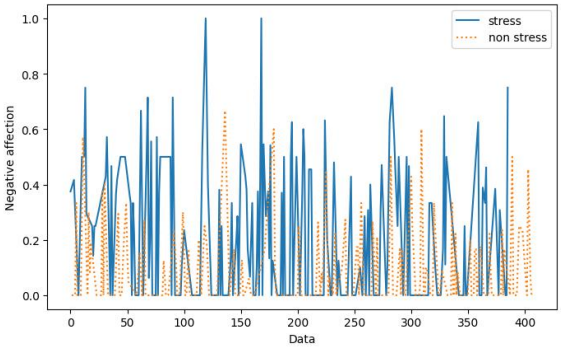
Hasil eksperimen akan dijabarkan menjadi beberapa pembahasan. Pertama hasil dari ekstraksi fitur WAI akan dilakukan, kemudian pembahasan mengenai perbandingan dari hasil setiap pembobotan kata pada model *Random Forest* dan SVM, dilanjutkan dengan visualisasi hasil TF-IDF, pembahasan *confusion matrix*, dan diakhiri dengan pembahasan data yang salah diklasifikasi baik dari vektor dan teks data tersebut.

Gambar 7 menunjukkan fluktuasi dari nilai *positive affection*. Hasil dari visualisasi tersebut dapat disimpulkan bahwa nilai *positive affection* memiliki hasil yang mirip untuk label stres dan non stres.



Gambar 7. Nilai *Positive affection*

Gambar 8, seperti Gambar 7 merupakan visualisasi tapi merupakan nilai *negative affection*. Visualisasi tersebut menampilkan bahwa nilai *negative affection* cenderung lebih tinggi untuk data dengan label stres.



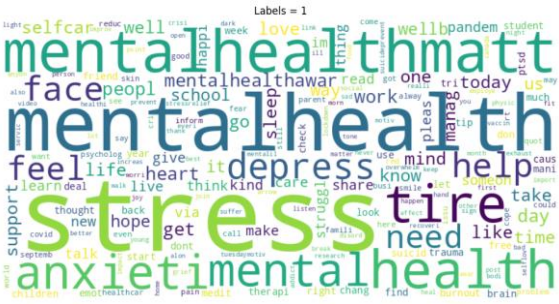
Gambar 8. Nilai *Negative affection*

Gambar 9 menunjukkan visualisasi dari perbandingan model SVM dengan setiap transformasi yang dilakukan pada penelitian. Hasil dari transformasi tersebut menampilkan hasil dari TF-IDF, *CountVectorizer*, NRCLex, dan *Word Affect Intensities* (WAI). Dari setiap hasil tersebut didapatkan hasil yang paling tinggi TF-IDF dan hasil paling buruk adalah WAI.

Gambar 10 menunjukkan visualisasi dari perbandingan model *Random Forest*. Didapatkan hasil yang paling baik adalah dengan transformasi *CountVectorizer*, dan hasil paling buruk adalah transformasi WAI.

Gambar 11 menunjukkan kata-kata yang sering muncul pada teks dengan kelas non stres. Kata-kata tersebut seperti *happi*, *smile*, *excit*, *face*, *stres*, dan lain-lain. Sementara itu Gambar 12 menunjukkan kata-kata yang sering muncul pada teks dengan kelas stres. Pada kelas stres kata-kata yang muncul menggambarkan bagaimana perasaan penulis teks seperti *tired*, *stress*, *depress*, dan lain-lain. Kata-kata ini dapat digunakan sebagai penciri dari masing-masing kelas.

Gambar 11. *Word cloud* kelas 0

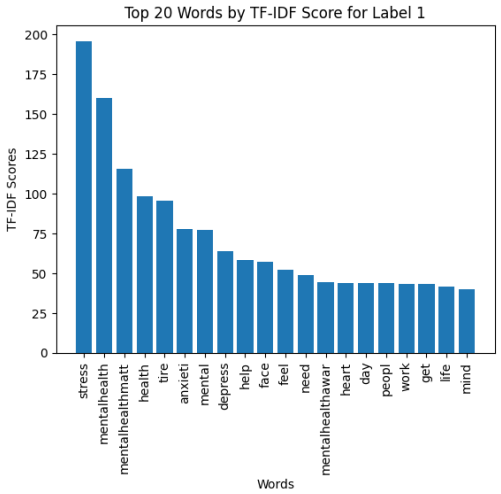


Gambar 12. *Word cloud* kelas 1

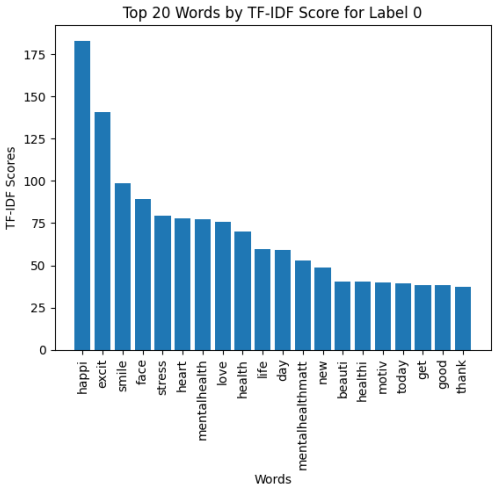
Kata-kata yang sering muncul pada setiap teks masing-masing kelas ini akan berguna untuk melakukan klasifikasi. Transformasi data yang dilakukan menggunakan TF-IDF akan membuat term/kata yang sering muncul mempunyai nilai yang besar. Nilai yang besar ini menandakan term tersebut mempunyai bobot yang tinggi dalam model pada proses klasifikasi. Gambar 13 adalah 20 kata dengan nilai TF-IDF tertinggi pada kelas stres dan Gambar 14 adalah 20 kata dengan nilai TF-IDF tertinggi untuk kelas non stres. Kata-kata ini sesuai dengan plot pada *word cloud*.

Hasil dari pencarian parameter terbaik dengan menggunakan *GridSearch* CV untuk kedua model adalah untuk model SVM nilai parameter C adalah 2 dan parameter *kernel* adalah RBF. Sementara itu, untuk model *Random Forest* parameter *criterion* diperoleh dengan nilai *entropy*, *max_depth* adalah *none*, *max_feature* adalah sqrt dan *n_estimator* adalah 400.

Tabel 2 merupakan *confusion matrix* dari hasil klasifikasi SVM dan Tabel 3 merupakan *confusion matrix* dari hasil klasifikasi *random forest*.



Gambar 13. 20 kata dengan nilai TF IDF tertinggi kelas 1(stres)



Gambar 14. 20 kata dengan nilai TF IDF tertinggi kelas 0(non stres)

Tabel 2. *Confusion matrix* SVM

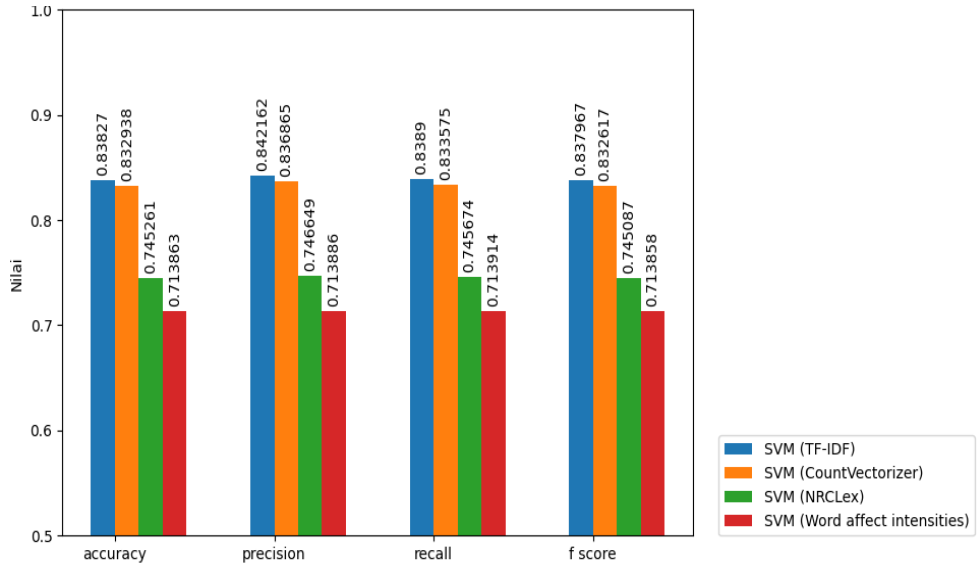
		Prediksi	
		Positive	***Negative***
Aktual	***Positive***	744	90
	Negative	183	671

Tabel 3. *Confusion matrix Random forest*

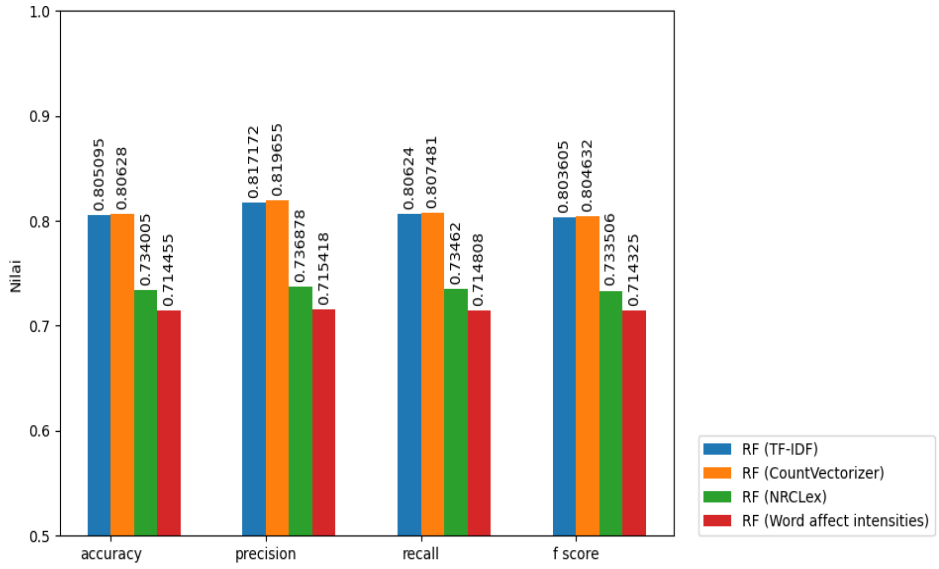
		Prediksi	
		Positive	***Negative***
Aktual	***Positive***	753	81
	Negative	248	606

Dari hasil evaluasi, dapat dilihat bahwa hasil model SVM memiliki akurasi sebesar 84%. Artinya 84% dari data uji diprediksi dengan benar. *Precision* dari model tersebut yaitu 80%, yang berarti 80% dari prediksi berlabel "1" adalah benar berlabel "1". *Recall* dari model tersebut yaitu 89%, yang berarti model dapat mengklasifikasi 89% dari seluruh data berlabel "1" dengan benar. F1-score dari model tersebut yaitu 84% yang merupakan rata-rata harmonis dari *precision* dan *recall*.

Dari evaluasi yang telah dilakukan, baik model SVM maupun *random forest* memiliki performa yang cukup baik dalam melakukan klasifikasi stres pada dataset twitter_full. Model SVM memiliki akurasi 84%, *precision* 80%, recall 89%, dan F1-score 84%, sedangkan model *random forest* memiliki akurasi 81%, *precision* 75%, recall 90%, dan F1-score 82%. Model SVM memiliki performa sedikit lebih unggul dibanding dengan model *random forest*.



Gambar 9. Perbandingan transformasi SVM



Gambar 10. Nilai transformasi RF

Contoh kesalahan klasifikasi dari hasil prediksi model *random forest* ditunjukkan pada Tabel 4. Tabel tersebut menampilkan jumlah dari masing-masing nilai.

Kelima data tersebut memiliki jumlah nilai (*value count*) yang ditunjukkan pada Gambar 15. Masih perlu adanya eksplorasi lebih lanjut untuk menyimpulkan ciri-ciri dari kesalahan klasifikasi model.

Tabel 4. Vektor data salah klasifikasi

id	**aa**	**aaa**	**aaaa**	...	□□□□ □□	□□□□ □	□□	□□□□ □
1	0	0	0	...	0	0	0	0
2	0	0	0	...	0	0	0	0
17	0	0	0	...	0	0	0	0
19	0	0	0	...	0	0	0	0
22	0	0	0	...	0	0	0	0

Isi dari teks dengan kesalahan klasifikasi ditunjukkan pada Tabel 5. Label dari setiap teks dengan kesalahan klasifikasi merupakan non stres dan prediksi model *random forest* adalah stres.

```
0    19890
1       20
2        1
Name: 1, dtype: int64
0    19899
1       12
Name: 17, dtype: int64
0    19890
1       19
2        2
Name: 19, dtype: int64
0    19908
1        3
Name: 22, dtype: int64
0    19896
1       15
Name: 23, dtype: int64
```

Gambar 15. *Value count* dari lima data salah klasifikasi

Tabel 5. Kesalahan klasifikasi model

id	text	prediksi	label
1	*mentalhealthmatters and the well-being of all our pupils. Letters regarding our health checks where issued today. You're MentalHealthMatters sparkles we have a variety of 16 programmes to support our young people ranging from equine therapy to mtb! Valued*	1	0
2	*ACCT CONTEMPLATIVE LAB SERIES WORLD SCRIPTURE Dear friends, I want to introduce you to the Patreon page of the Asian Centre for Creative Theology—the second wing of my missional ministry. mentalhealth meditation Singapore*	1	0
17	*My own Inferno Sugar Maple. Augustwish is for a extraordinary SeptemberToRemember FallColours mentalhealth Mindfulness DBT IntegrativePsychiatry coping*	1	0
19	*The major risk factors for heart disease are high blood cholesterol, high blood pressure and smoking. Contact us at +91-9818403954/ 9818391954 Visit to know more. cardiachealth healthyheart cardiacarrest heartpatient hearttips stress cardiologist*	1	0
22	*gblackys stress free*	1	0

Tabel 5 menunjukkan pada teks id satu, membahas tentang kesehatan dan menyebutkan mengenai hasil dari pada teks tersebut disebutkan kata-kata yang memiliki skor TF-IDF tinggi untuk tweet kelas stres yaitu '*mental health*'. Dipahami dari konteks teks id satu, teks dari tweet tersebut merupakan promosi yang membahas pentingnya kesehatan mental. Teks id 2 dan 17 juga sama dengan teks id 1, dimana konteks tidak sesuai dengan kata yang digunakan. Teks id 19, dibahas mengenai penyakit hati, pada teks tersebut disebut kata stres yang merujuk pada tipe kardiologi. Teks id 22, disebutkan kata stres namun hal tersebut bermakna bahwa objek 'glackys' bebas dari stres. Dari beberapa teks *misklasifikasi* tersebut menunjukkan kekurangan dari menggunakan TF-IDF sebagai transformasi untuk klasifikasi teks berlabel stres. Pemahaman konteks untuk transformasi dan tidak hanya dari kata yang digunakan saja sangat diperlukan. Transformasi dengan menghitung ukuran sentimen positif dan negatif juga belum cukup untuk memberikan evaluasi model diatas 90%.

4. KESIMPULAN DAN SARAN

Proses *preprocessing* data teks dapat dilakukan dengan menggunakan berbagai metode. Transformasi dengan *CountVectorizer* menggunakan kemunculan sebuah term dalam kalimat untuk nilai dari matriks. TF-IDF melakukan transformasi dengan membandingkan kemunculan term tersebut dalam suatu dokumen. NRCLex menggunakan kamus untuk menentukan emosi yang terkandung dalam suatu teks. *Word Affect Intensities* menggunakan transformasi dari NRCLex untuk membentuk fitur baru berupa emosi positif dan negatif. Masing-masing metode ini mempunyai kelebihan dan kekurangan.

Berdasarkan penelitian yang telah dilakukan didapatkan bahwa dari beberapa metode yang dilakukan, model SVM dengan metode transformasi TF-IDF memiliki performa yang paling baik dengan akurasi 84%, *precision* 80%, *recall* 89%, dan *F1-score* 84%. Di sisi lain, pada model *Random Forest* didapatkan bahwa model *Random Forest* memiliki performa paling baik dengan metode transformasi *CountVectorizer*, akurasi 80%, *precision* 81%, *recall* 80%, dan *F1-score* 80%. Oleh karena itu, dapat disimpulkan bahwa model SVM lebih baik dibanding dengan model *Random Forest* dengan selisih akurasi sekitar 3%.

Penelitian ini menggunakan algoritma SVM dan *Random Forest* serta menggunakan metode transformasi TF-IDF, *CountVectorizer*, NRCLex, dan *word affect intensities* untuk melakukan deteksi stres dari data berbentuk teks. Penelitian seterusnya diharapkan dapat menggunakan algoritma lain seperti *deep learning* yang lebih kompleks serta menggunakan *word embedding* sebagai metode transformasi.

DAFTAR PUSTAKA

BIAU, G. & SCORNET, E., 2016. A Random Forest Guided Tour. TEST, p. 197–227.

HASTIE, T., TIBSHIRANI, R. & FRIEDMAN, J. H., 2009. The Elements Of Statistical Learning: Data Mining, Inference, and Prediction. New York, Springer series in statistics.

JADHAV, S. et al., 2019. Text Based Stress Detection Techniques Analysis Using Social Media. 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), pp. 1-5.

KWOK, T.-Y., 1996. Automatic Text Categorization Using Support Vector Machine. s.l., s.n.

LIAW, A. AND WIENER, M., 2002. Classification and Regression by Random Forest. The Newsletter of the R Project, 2, pp.18–22.

MUÑOZ, S. & IGLESIAS, C. A., 2022. A text classification approach to detect

psychological stress combining a lexicon-based feature framework with distributional representations. Information Processing & Managemen, 59(5).

N, V. VAPNIK, 1995. The Nature of Statistical Learning Theory. New York: Springer Verlag.

NIJHAWAN, T., ATTIGERI, G. & ANANTHAKRISHNA, T., 2022. Stress Detection Using Natural Language Processing and Machine Learning Over Social Interactions. Journal of Big Data, IX(1), p. 33.

NUGROHO, A. S., WITARTO, A. B. & HANDOKO, D., 2003. Support Vector Machine, Teori dan Aplikasinya dalam Bioinformatika1. [Online].

PERANGIN-ANGIN, D.J. & BACHTIAR, F.A., 2021. Classification of Stress in Office Work Activities Using Extreme Learning Machine Algorithm and One-way ANOVA F-Test Feature Selection. 2021 4th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2021, pp.503–508. https://doi.org/10.1109/ISRITI54043.2021.9702802.

PILLAI, R. G., THELWALL, M. & ORASAN, C., 2018. Detection of Stress and Relaxation Magnitudes for Tweets. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, p. 1677–1684.

RASTOGI, A., LIU, Q. & CAMBRIA, E., 2022. Stress detection from social media articles: New dataset benchmark and analytical study. 2022 International Joint Conference on Neural Networks (IJCNN). IEEE. pp.1–8.

RISA, D.F., PRADANA, F. AND BACHTIAR, F.A., 2021. Implementasi Metode Naive Bayes untuk Mendeteksi Stres Siswa Berdasarkan Tweet pada Sistem Monitoring Stres. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 8(6).

SAPUTRA, R., ISTANTO, H., BACHTIAR, F. A., RIDOK, A., (2022). Pengaruh Word Affect Intensities Terhadap Deteksi Ulasan Palsu. Jurnal Teknologi Informasi Dan Ilmu Komputer, 9(2), 427–434. https://doi.org/10.25126/JTIIK.2022925652

STRIMPEL, O. B. R., 1997. Computer graphics. McGraw-Hill Encyclopedia of Science and Technology, Volume IV, pp. 279-283.