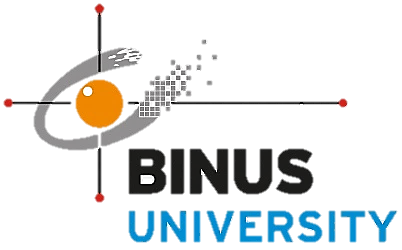
**Klasifikasi Tweet dengan Menggunakan Metode *Long Short-Term Memory (LSTM), Decision Tree,* dan *Random Forest***



**Dosen Pembimbing:**

Lili Ayu Wulandhari, S.Si., M.Sc., Ph.D.

**Anggota Kelompok:**

|  |  |
| --- | --- |
| Gisela Elviany | 2501972120 |
| Nayla Anandhita | 2501979726 |
| Sherryl Kurniawan | 2540119570 |

**PROGRAM STUDI COMPUTER SCIENCE**

**FAKULTAS SCHOOL OF COMPUTER SCIENCE**

# **UNIVERSITAS BINA NUSANTARA**

**Jl. Kebon Jeruk Raya No. 27, Kebon Jeruk, Jakarta Barat 11530**

**2022/2023**

**BAB 1**

**PENDAHULUAN**

Di tengah era digital ini, manusia tidak dapat dipisahkan dengan media sosial. Media sosial menjadi wadah bagi setiap orang untuk mengomunikasikan diri di internet. Lewat media sosial, orang bebas mengespresikan perasaan, termasuk argumen dan opini mereka tentang berbagai topik. Perasaan yang disalurkan pada suatu teks mengandung emosi yang berperan penting dalam berbagai aspek kehidupan. Misalnya untuk memberikan opini terhadap suatu keputusan bisnis. Transisi digital mengakibatkan perusahaan untuk mengubah data tidak terstruktur menjadi suatu wawasan bermakna seperti mengukur kepuasaan pelanggan dengan menganalisis komentar, saran, serta ulasan yang diberikan [1]. Dengan begitu, perusahaan dapat memahami perspektif pelanggan sehingga mereka dapat melakukan perubahan untuk meningkatkan kualitas sistem, barang, dan pelayanan mereka. Di bidang kesehatan, analisis emosi dapat membantu dalam pemantauan kondisi psikologis seseorang. Pesan yang diungkapkan dalam media sosial dapat menunjukkan emosi dan keadaan seseorang. Sehingga jika orang tersebut terlihat depresi, bantuan profesional dapat disalurkan dan mencegak terjadinya bunuh diri.[2]

Salah satu media sosial berbasis teks yang populer saat ini adalah X (dulu bernama twitter). Menurut internetlivestats, rata-rata 6,000 *tweet* diunggah pada setiap detiknya, berarti bahwa ada 500 juta tweet yang dikirim dalam satu hari.[3] Setiap teks yang diunggah mengandung berbagai macam emosi yang dirasakan oleh penulisnya. Walaupun begitu, memahami maksud dari emosi pada suatu teks bukanlah hal yang mudah. Terkadang kita salah memahami emosi dari teks yang ada. Oleh karena itu, penelitian dalam analisis emosi dari teks memiliki potensi aplikasi yang luas dan penting serta perlu dikembangkan metode yang secara otomatis dapat menganalisis emosi dari suatu teks.

Penelitian ini akan berfokus pada tugas klasifikasi emosi berbasis teks. Teks akan diklasifikasikan ke dalam empat kategori emosi utama, yaitu kemarahan *(anger)*, kebahagiaan *(joy),* kesedihan *(sadness)*, dan ketakutan *(fear)*. Dengan menggunakan dataset yang diperoleh dari Kaggle, penelitian ini akan menggunakan metode *Long Short-Term Memory* (LSTM), *Decision Tree*, dan *Random Forest* untuk memperoleh model yang secara optimal dapat mengkategorikan emosi dalam teks. Untuk mencapai tujuan tersebut, penelitian akan melewati beberapa langkah, yaitu *text preprocessing, text representation, modelling,* serta *evaluation*. Melalui langkah-langkah tersebut, penelitian ini diharapkan dapat memberikan kontribusi dalam bidang analisis emosi berbasis teks dengan menyediakan solusi praktis untuk memahami emosi yang terkandung secara otomatis.

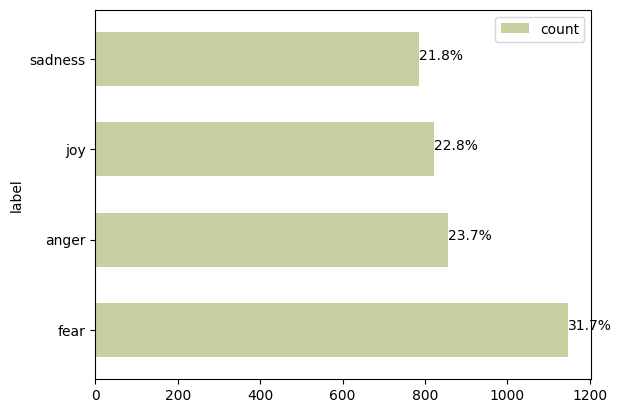
**BAB 2**

**METODOLOGI**

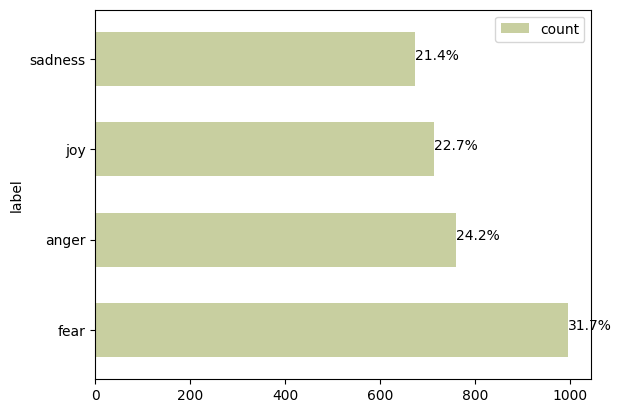
**2.1 Dataset**

Penelitian ini menitikberatkan pada analisis emosi berbasis teks yang diambil dari data terbuka dari website Kaggle. Data yang telah dikumpulkan terbagi secara langsung berdasarkan data *training* dan data *testing*, dimana data *testing* berjumlah 3142 baris dan data *training* berjumlah 3613 baris. Dataset yang digunakan, baik pada data *training* dan data *testing* berisi 2 kolom, diantaranya kolom “*text*” yang berisi kalimat *tweet* yang diunggah pemiliki akun X dan kolom “label” yang terdiri ada empat klasifikasi berdasarkan *tweet* unggahan, yaitu *sadness, joy, anger, dan fear*. Kolom “label” pada dataset ini akan digunakan untuk melakukan klasifikasi text berdasarkan keempat kategori emosi seseorang.

Gambar 2.1.1. Data *Training*



Gambar 2.1.2 Data *Testing*



**2.2 Tahapan Penelitian**

**2.2.1 *Exploratory Data Analysis* (EDA)**

Tahapan dalam proses analisis klasifikasi berbasis teks diawali dengan EDA *(Explanatory Data Analysis).* EDA adalah proses awal dalam tahapan analisis untuk mengetahui karakteristik utama dalam dataset sebelum melakukan permodelan yang sesuai. EDA bertujuan untuk memahami dataset secara mendalam serta distribusinya, pengambilan keputusan model terbaik, mengurangi risiko kesalahan dalam analisis, dan menangani *missing value* ataupun *outlier* [4]. EDA dalam analisis ini meliputi beberapa tahapan sebagai berikut:

**2.2.1.1 *Unique Data***

Tahapan EDA yang pertama dalam analisis ini, dengan melakukan pengecekan nilai unik dalam data. Dalam hal ini nilai unik dalam data training dan data testing terdiri atas empat kategori utama berdasarkan kolom “label” untuk klasifikasi emosi, yaitu *sadness, joy, anger, dan fear*. Pengecekan nilai unik dalam data dilakukan untuk memastikan bahwa semua label atau kategori ada dalam dataset dan mengetahui cakupan kategori dalam dataset yang selanjutnya digunakan untuk mengetahui distribusi data berdasarkan plot ataupun grafik.

**2.2.1.2 Distribusi Data**

Setelah mengetahui kategori apa saja yang terdapat dalam dataset, proses EDA ini dilanjutkan dengan mengetahui distribusi apa saja dalam dataset, baik dalam data training maupun dalam data testing berdasarkan nilai unik. Dalam data *training*, nilai unik terdiri atas “*fear*” berjumlah 1147 data, “*anger*” berjumlah 857 data, “*joy*” berjumlah 823 data, dan “*sadness*” berjumlah 786 data, sedangkan pada data *testing*, nilai unik terdiri atas “*fear*” berjumlah 995 data, “*anger*” berjumlah 760 data, “*joy*” berjumlah 714 data, dan “*sadness*” berjumlah 673 data. Visualisasi data berdasarkan nilai unik dapat dilihat pada gambar 2.1.1 untuk data *training* dan gambar 2.1.2 untuk data *testing*.

**2.2.1.3 Pengecekan *Missing Value***

Setelah mengetahui penyebaran data *training* dan data *testing*, selanjutkan dilakukan pengecekan *missing value* untuk memastikan bahwa tidak ada nilai yang hilang atau kosong dalam dataset. Hal ini bertujuan untuk mencegah terjadinya error dalam proses analisis klasifikasi selanjutnya. Dalam dataset, baik pada data *training* dan data *testing* tidak terdapat nilai yang hilang sehingga tidak perlu penanganan lebih lanjut dalam mengatasi nilai hilang.

**2.2.2 *Text Preprocessing***

Tahapan dalam proses analisis klasifikasi berbasis teks selanjutnya adalah *preprocessing*. *Text preprocessing* dilakukan untuk menyiapkan data mentah menjadi data yang dapat diolah lebih lanjut. Text preprocessing meliputi *case folding, cleaning,* dan *filtering*.[5] Tahapan dalam *text preprocessing* ini biasanya dilakukan dengan tujuan agar meningkatkan keterbacaan dalam data, mengurangi kata-kata yang tidak diperlukan dalam analisis, meningkatkan akurasi, dan meningkatkan kualitas dari data. Tahapan preprocessing dalam data ini, yaitu:

**2.2.2.1 *Case Folding***

*Case folding* dilakukan untuk standarisasi teks. Dalam hal ini, pada dataset, tidak semua kalimat memiliki struktur kapitalisasi yang sama, masih terdapat huruf besar dan huruf kecil, sehingga dalam case folding dilakukan pengubahan struktur kapitalisasi kalimat agar setiap struktur kalimat memiliki kesamaan yaitu menggunakan huruf kecil dalam struktur kalimatnya. [5]

**2.2.2.2 *Cleaning***

Setelah memastikan bahwa struktur kapitalisasi dalam kalimat sudah huruf kecil semua dengan *case folding*, selanjutnya dilakukan *cleansing* pada setiap kalimat dalam data. Dalam dataset ini, masih terdapat beberapa elemen yang tidak diinginkan, seperti:

* + - * Terdapat simbol karakter
      * Terdapat tanda baca
      * Terdapat simbol emoji dan stiker
      * Terdapat angka

Dikarenakan dalam dataset tersebut masih terdapat elemen yang tidak diinginkan, maka analisis ini memerlukan proses *cleaning* untuk menghilangkan elemen tersebut. Tahapan cleansing sangat diperlukan dalam proses analisis ini agar menghindari risiko error atau ketidaktepatan dalam penentuan model serta hasil akurasinya.

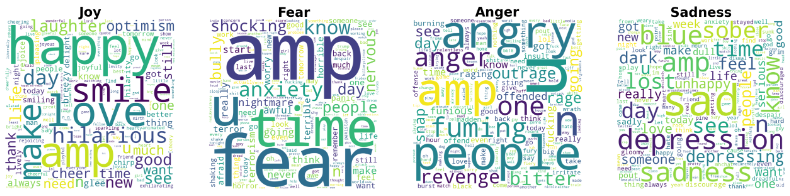
**2.2.2.3 *Filtering***

Setelah proses *cleaning* selesai dilakukan, untuk teknik *preprocessing* data selanjutnya adalah filtering. *Filtering* dapat berupa menghilangkan *stopword* di dalam data. *Stopword* biasanya berisi kata yang sering muncul dan tidak penting dalam sebuah kalimat,[6] sehingga untuk membersihkan duplikasi kata dalam teks, analisis ini melakukan *filtering* untuk menghapus kata tersebut.

**2.2.2.4 Visualisasi *WordCloud***

Selanjutnya, untuk melihat frekuensi data yang muncul dalam data dapat dilihat dengan visualisasi data. Dalam analisis ini, menggunakan WorkCloud untuk melihat visualisasi berdasarkan dengan frekuensi katanya,[7] dimana setiap kata yang sering muncul akan dimunculkan dalam visualisasi dengan kata yang hurufnya lebih besar dibandingkan kata lain.

Gambar 2.2.2.4 Visualisasi



**2.2.2.5 Tokenisasi**

Setelah tahapan proses *cleaning* dan *filtering* selesai, selanjutkan tahapan terakhir dalam preprocessing data yaitu tokenisasi. Tokenisasi adalah proses pemecahan kalimat menjadi kata-kata individual atau biasa disebut sebagai token.[8] Tujuan dari tokenisasi untuk memudahkan dalam melakukan analisis karena dengan memecah kalimat menjadi struktur yang lebih kecil dan mempermudahkan dalam memanipulasi teks dalam tahapan teks representation.

**2.2.3 *Text representation***

Setelah *Text Preprocessing* telah selesai dilakukan, langkah selanjutnya untuk melakukan pengolahan data yaitu pada tahap *text representation*. *Text Representation* adalah tahapan mengubah format data mentah menjadi data yang nantinya dapat diolah dan dianalisis lebih lanjut.[9] Ada dua metode text represntation yang digunakan pada analisis ini, yaitu TF-IDF dan Word2Vec – *Skipgram*.

**2.2.3.1 TF-IDF**

*TF (Term Frequency) - IDF (Inverse Document Frequency)* merupakan salah satu metode *text representation,* dimana melakukan pembobotan data untuk mengukur seberapa penting kata (token) tersebut dalam teks. Pada Teknik IDF, teknik ini akan mengurangi bobot dengan kata yang sudah umum dan menambahkan bobot nilai pada kata yang jarang muncul. Pada Teknik TF, dilakukan perhitungan frekuensi terhadap kata yang sering muncul. [10]

Dimana,

* = bobot term terhadap dokumen
* = jumlah kemunculan term terhadap dokumen
* D = jumlah semua dokumen dalam data
* = jumlah dokumen yang mengandung term

**2.2.3.1 Word2Vec – *Skipgram***

Word2Vec merupakan teknik representasi data dengan bantuan neural network, dimana data akan diolah menjadi representasi vector dalam bentuk numerik. Terdapat dua arsitektur model dalam Word2Vec, yaitu CBOW dan *Skipgram*. Perbedaan keduanya adalah, yaitu CBOW melakukan prediksi kata target berdasarkan kata-kata yang ada di sekitarnya, sedangkan *Skipgram* melakukan prediksi kata sekitarnya berdasarkan kata target.[11] Pada analisis sentimen emosi, paper ini memanfaatkan Word2Vec model Skipgram dalam metode *text representation* yang kedua.

**2.2.4 *Classification Model***

Dalam penelitian ini, beberapa model klasifikasi digunakan untuk menganalisis teks dan mengkategorikan emosi yanh terkandung dalam teks dalam data. Model-model yang digunakan adalah LSTM (*Long Short-Term Memory*) dengan metode *Skipgram* dan TF-IDF, dan Decison Tree dengan metode TF-IDF.

**2.2.4.1 LSTM**

LSTM *neural network* adalah salah satu jenis RNN (*recurrent neural network) yang* berarti *neural network* menggunakan *output* dari *layer* langkah waktu sebelumnya sebagai bagian untuk *input layer* tersebut. Karena *layer* RNN memiliki koneksi berbentuk sirkular dengan dirinya sendiri, RNN dapat melakukan *feedback* dengan dirinya sendiri dan *neuron* pada beberapa *layer* pada langkah waktu sebelumnya.[12] Bentuk koneksi *neuron* seperti ini menyebabkan LSTM-RNN dapat melakukan *backpropagation* sehingga dapat belajar dan menggunakan bobot dari *layer* sebelumnya.

**2.2.4.2 Decision Tree**

Decision trees adalah algoritma *machine learning* yang digunakan untuk membuat model prediktif atau klasifikasi. Sebuah *decision tree* berawal dari *root node, internal nodes* yang merupakan simpul dalam tree yang memiliki anak, *leaf node* di akhir *decision tree* yang tidak memiliki anak simpul lain di mana solusi telah ditemukan dan keputusan sudah dibuat.[13]

**2.2.4.3 Random Forest**

*Random forest* adalah algoritma *machine learning* yang menggabungkan beberapa *decision trees* untuk mencapai suatu hasil. Algoritma ini bekerja dengan membandingkan keputusan yang diambil setiap *decision tree* dan melakukan pemungutan suara berdasarkan keputusan akhir yang paling populer. Dengan begitu, algoritma *random forest akan* mengambil keputusan dengan akurasi yang tinggi karena mengambil keputusan berdasarkan beberapa *decision trees.*

**2.2.5 Metrik evaluasi**

Evaluasi metrik adalah untuk mengukur kinerja dari hasil analisis dalam data yang dilakukan menggunakan berbagai teknik. Beberapa contoh metode evaluasi metrik adalah:

**2.2.5.1 *Accuracy***

*Accuracy* adalah metrik yang mengukur sejauh mana model dapat mengklasifikasikan data dengan benar secara keseluruhan. *Accuracy* dihitung dari jumlah prediksi benar dibagi dengan total data yang diuji.[14]

**2.2.5.2 *Precision***

*Precision* adalah metrik yang mengukur sejauh mana prediksi positif yang benar atau ketepatan model. *Precision* dihitung dengan membagi jumlah benar debgan jumlah total prediksi.[15]

**2.2.5.3 *Recall***

*Recall* adalah metrik yang mengukur sejauh mana model dapat mendeteksi dengan benar kelas positif atau melihat presentase data yang benar dari suatu kelas dibandingkan dengan total data aktual.[16]

**2.2.5.4 F1-*Score***

FI-*Score* adalah metrik yang digunakan untuk mengukur kinerja model untuk mengukur seberapa akurat model memprediksi kelas yang benar dari rata-rata harmonik *presicion dan recall*.[17]

**BAB 3**

**HASIL DAN ANALISIS**

**3.1 Hasil Model LSTM**

**3.1.1 TF-IDF**

**Tabel 1.** Arsitektur LSTM (TF-IDF)

|  |  |  |
| --- | --- | --- |
| Layer (Type) | Output Shape | Param # |
| Embedding (Embedding) | (None, 21, 100) | 2000000 |
| Lstm (LSTM) | (None, 256) | 365568 |
| Batch\_normalization (BatchNormalization) | (None, 256) | 1024 |
| Dropout (Dropout) | (None, 256) | 0 |
| Dense (Dense) | (None, 4) | 1028 |
| Total parameters: 2367620 (9.03 MB)   * Trainable params: 2367108 * Non-trainable params: 512 | | |

Model LSTM melalui TF-IDF ini mempunyai arsitektur dengan lapisan *embedding* untuk menginisialisasi bobot. Setelah lapisan *embedding*, model ini menggabungkan lapisan atau *layer* LSTM dengan 256 unit. Untuk memastikan training yang stabil, diperlukan *Batch Normalization* untuk aktivasi gradien. Diikuti dengan *Dropout* 0.7 untuk pembaruan selama proses training. Hal ini untuk mencegah adanya *overfitting*.

**Tabel 2.** Classification Report LSTM (TF-IDF) Training

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.46 | 0.01 | 0.03 |
| 1 | 0.24 | 0.96 | 0.39 |
| 2 | 0.48 | 0.10 | 0.17 |
| 3 | 0.00 | 0.00 | 0.00 |
|  | | | |
| Accuracy |  |  | 0.26 |
| Macro avg | 0.29 | 0.27 | 0.15 |
| Weighted avg | 0.31 | 0.26 | 0.14 |

Model LSTM menggunakan TF-IDF *Training* menunjukkan kinerja yang bervariasi. Kelas 1: Recall tinggi & kelas 2: Precision tinggi dan *Recall* rendah, dan kelas 0 & 3: gagal klasifikasi. Nilai akurasi pada data training adalah 0.26. Hal ini berarti hanya sekitar 26% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat buruk dalam mengklasifikasikan kelas-kelas pada data training yang kurang terwakili dan membutuhkan perbaikan untuk tugas klasifikasi yang diinginkan.

**Tabel 3.** Classification Report LSTM (TF-IDF) Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.35 | 0.02 | 0.03 |
| 1 | 0.24 | 0.95 | 0.39 |
| 2 | 0.27 | 0.04 | 0.07 |
| 3 | 0.00 | 0.00 | 0.00 |
|  | | | |
| Accuracy |  |  | 0.24 |
| Macro avg | 0.22 | 0.25 | 0.12 |
| Weighted avg | 0.23 | 0.24 | 0.12 |

Model LSTM menggunakan TF-IDF *Testing* menunjukkan kinerja yang bervariasi. Kelas 1: Recall tinggi & kelas 2: Precision dan *Recall* rendah, dan kelas 0 & 3: gagal klasifikasi. Nilai akurasi pada data testing adalah 0.24. Hal ini berarti hanya sekitar 24% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat buruk dalam mengklasifikasikan kelas-kelas pada data testing yang kurang terwakili dan membutuhkan perbaikan untuk tugas klasifikasi yang diinginkan.

Perbandingan nilai akurasi pada data training dan data testing terjadi underfit, dimana nilai akurasi pada data training dan data testing rendah. Oleh karena hal tersebut, model prediksi analisis emosi berbasis text tidak dapat menggunakan LSTM menggunakan TFIDF.

**3.1.2 Skipgram**

**Tabel 4.** Arsitektur LSTM (Skipgram)

|  |  |  |
| --- | --- | --- |
| Layer (Type) | Output Shape | Param # |
| Embedding\_1 (Embedding) | (None, 21, 100) | 20000000 |
| Lstm\_1 (LSTM) | (None, 256) | 365568 |
| Batch\_normalization\_1 (BatchNormalization) | (None, 256) | 1024 |
| Dropout\_1(Dropout) | (None, 256) | 0 |
| Dense\_1(Dense) | (None, 4) | 1028 |
| Total parameters: 2367620 (9.03 MB)   * Trainable params: 2367108   Non-trainable params: 512 | | |

Arsitektur ini sama seperti LSTM untuk metode Word2Vec - Skipgram, mencakup lapisan Dense dengan 4-unit dan fungsi aktivasi softmax. Struktur ini menggabungkan lapisan Embedding, LSTM, Normalisasi Batch, Dropout, dan Dense untuk memberikan kinerja yang baik untuk tugas klasifikasi teks.

**Tabel 5.** Classification Report LSTM (Skipgram) Training

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.99 | 0.99 | 0.99 |
| 1 | 0.98 | 0.98 | 0.98 |
| 2 | 0.98 | 1.00 | 0.99 |
| 3 | 0.98 | 0.96 | 0.97 |
|  | | | |
| Accuracy |  |  | 0.98 |
| Macro avg | 0.98 | 0.98 | 0.98 |
| Weighted avg | 0.98 | 0.98 | 0.98 |

Model LSTM menggunakan Word2Vec-Skipgram *Training* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3 : Recall dan precision tinggi, Nilai akurasi pada data training adalah 0.98. Hal ini berarti hanya sekitar 98% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat baik dalam mengklasifikasikan kelas-kelas pada data training.

**Tabel 6.** Classification Report LSTM (Skipgram) Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.35 | 0.40 | 0.37 |
| 1 | 0.28 | 0.20 | 0.23 |
| 2 | 0.24 | 0.38 | 0.29 |
| 3 | 0.26 | 0.12 | 0.17 |
|  | | | |
| Accuracy |  |  | 0.29 |
| Macro avg | 0.28 | 0.27 | 0.27 |
| Weighted avg | 0.29 | 0.29 | 0.28 |

Model LSTM menggunakan Word2Vec-Skipgram *Testing* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision rendah. Nilai akurasi pada data testing adalah 0.29. Hal ini berarti hanya sekitar 29% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat buruk dalam mengklasifikasikan kelas-kelas pada data testing yang kurang terwakili dan membutuhkan perbaikan untuk tugas klasifikasi yang diinginkan.

Perbandingan nilai akurasi pada data training dan data testing terjadi overfit, dimana nilai akurasi pada data training tinggi dan data testing rendah. Oleh karena hal tersebut, model prediksi analisis emosi berbasis text kurang optimal dengan menggunakan LSTM dengan Word2Vec – Skipgram.

**3.2 Hasil Model Decision Tree**

**3.2.1 TF-IDF**

**Tabel 7.** Classification Report Decision Tree (TF-IDF) Training

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.98 | 1.00 | 0.99 |
| 1 | 0.97 | 1.00 | 0.98 |
| 2 | 1.00 | 1.00 | 1.00 |
| 3 | 1.00 | 0.94 | 0.97 |
|  | | | |
| Accuracy |  |  | 0.99 |
| Macro avg | 0.99 | 0.98 | 0.99 |
| Weighted avg | 0.99 | 0.99 | 0.99 |

Model Decision Tree menggunakan TF-IDF *Training* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision tinggi, Nilai akurasi pada data training adalah 0.99. Hal ini berarti hanya sekitar 99% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat baik dalam mengklasifikasikan kelas-kelas pada data training.

**Tabel 8.** Classification Report Decision Tree (TF-IDF) Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.78 | 0.83 | 0.80 |
| 1 | 0.84 | 0.79 | 0.81 |
| 2 | 0.85 | 0.85 | 0.85 |
| 3 | 0.80 | 0.77 | 0.79 |
|  | | | |
| Accuracy |  |  | 0.81 |
| Macro avg | 0.82 | 0.81 | 0.81 |
| Weighted avg | 0/82 | 0.81 | 0.81 |

Model Decision Tree menggunakan TF-IDF *Testing* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision cukup tinggi. Nilai akurasi pada data training adalah 0.81. Hal ini berarti hanya sekitar 81% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang cukup baik dalam mengklasifikasikan kelas-kelas pada data testing.

Dari perbandingan nilai akurasi pada data training dan data testing, dapat terlihat bahwa dengan menggunakan model Decision Tree dengan TF-IDF cukup optimal dalam melakukan prediksi analisis emosi berbasis teks.

**3.2.2 Skipgram**

**Tabel 9.** Classification Report Decision Tree (Skipgram) Training

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.98 | 1.00 | 0.99 |
| 1 | 0.97 | 1.00 | 0.98 |
| 2 | 1.00 | 1.00 | 1.00 |
| 3 | 1.00 | 0.94 | 0.97 |
|  | | | |
| Accuracy |  |  | 0.99 |
| Macro avg | 0.99 | 0.98 | 0.99 |
| Weighted avg | 0.99 | 0.99 | 0.99 |

Model Decision Tree menggunakan Word2Vec-Skipgram *Training* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision sangat tinggi. Nilai akurasi pada data training adalah 0.99. Hal ini berarti hanya sekitar 99% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat baik dalam mengklasifikasikan kelas-kelas pada data training.

**Tabel 10.** Classification Report Decision Tree (Skipgram) Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.37 | 0.42 | 0.39 |
| 1 | 0.22 | 0.05 | 0.08 |
| 2 | 0.18 | 0.10 | 0.13 |
| 3 | 0.25 | 0.55 | 0.35 |
|  | | | |
| Accuracy |  |  | 0.28 |
| Macro avg | 0.26 | 0.28 | 0.24 |
| Weighted avg | 0.27 | 0.28 | 0.25 |

Model Desicion Tree menggunakan Word2Vec-Skipgram *Testing* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision rendah. Nilai akurasi pada data testing adalah 0.28. Hal ini berarti hanya sekitar 28% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat buruk dalam mengklasifikasikan kelas-kelas pada data testing yang kurang terwakili dan membutuhkan perbaikan untuk tugas klasifikasi yang diinginkan.

Perbandingan nilai akurasi pada data training dan data testing terjadi overfit, dimana nilai akurasi pada data training tinggi dan data testing rendah. Oleh karena hal tersebut, model prediksi analisis emosi berbasis text kurang optimal dengan menggunakan Desicion Tree dengan Word2Vec - Skipgram

**3.2 Hasil Model Random Forest**

**3.2.1 TF-IDF**

**Tabel 11.** Classification Report Random Forest (TF-IDF) Training

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.99 | 0.99 | 0.99 |
| 1 | 0.98 | 0.99 | 0.98 |
| 2 | 1.00 | 1.00 | 1.00 |
| 3 | 0.97 | 0.97 | 0.97 |
|  | | | |
| Accuracy |  |  | 0.99 |
| Macro avg | 0.99 | 0.99 | 0.99 |
| Weighted avg | 0.99 | 0.99 | 0.99 |

Model Random Forest menggunakan TF-IDF *Training* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision sangat tinggi. Nilai akurasi pada data training adalah 0.99. Hal ini berarti hanya sekitar 99% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat baik dalam mengklasifikasikan kelas-kelas pada data training.

**Tabel 12.** Classification Report Random Forest (TF-IDF) Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.74 | 0.90 | 0.81 |
| 1 | 0.90 | 0.76 | 0.82 |
| 2 | 00.92 | 0.85 | 0.88 |
| 3 | 0.83 | 0.77 | 0.80 |
|  | | | |
| Accuracy |  |  | 0.83 |
| Macro avg | 0.85 | 0.82 | 0.83 |
| Weighted avg | 0.84 | 0.83 | 0.83 |

Model Random Forest menggunakan TF-IDF *Testing* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision cukup tinggi. Nilai akurasi pada data training adalah 0.83. Hal ini berarti hanya sekitar 83% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang cukup baik dalam mengklasifikasikan kelas-kelas pada data testing.

Dari perbandingan nilai akurasi pada data training dan data testing, dapat terlihat bahwa dengan menggunakan model Random Forest dengan TF-IDF cukup optimal dalam melakukan prediksi analisis emosi berbasis teks.

**3.2.2 Skipgram**

**Tabel 11.** Classification Report Random Forest (Skipgram) Training

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.99 | 1.00 | 0.99 |
| 1 | 0.98 | 0.99 | 0.98 |
| 2 | 1.00 | 1.00 | 1.00 |
| 3 | 0.98 | 0.96 | 0.97 |
|  | | | |
| Accuracy |  |  | 0.99 |
| Macro avg | 0.99 | 0.99 | 0.99 |
| Weighted avg | 0.99 | 0.99 | 0.99 |

Model Random Forest menggunakan Word2Vec-Skipgram *Training* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision sangat tinggi. Nilai akurasi pada data training adalah 0.99. Hal ini berarti hanya sekitar 99% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat baik dalam mengklasifikasikan kelas-kelas pada data training.

**Tabel 11.** Classification Report Random Forest (Skipgram) Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Kelas | Precision | Recall | F1-Score |
| 0 | 0.36 | 0.46 | 0.42 |
| 1 | 0.13 | 0.02 | 0.03 |
| 2 | 0.18 | 0.07 | 0.10 |
| 3 | 0.25 | 0.58 | 0.34 |
|  | | | |
| Accuracy |  |  | 0.29 |
| Macro avg | 0.24 | 0.28 | 0.22 |
| Weighted avg | 0.25 | 0.29 | 0.24 |

Model Random Forest menggunakan Word2Vec-Skipgram *Testing* menunjukkan kinerja yang bervariasi. Kelas 0, kelas 1, kelas 2, dan kelas 3: Recall dan precision rendah. Nilai akurasi pada data testing adalah 0.29. Hal ini berarti hanya sekitar 29% keberhasilan model dalam memprediksi secara benar. Evaluasi model ini menunjukkan performa yang sangat buruk dalam mengklasifikasikan kelas-kelas pada data testing yang kurang terwakili dan membutuhkan perbaikan untuk tugas klasifikasi yang diinginkan.

Perbandingan nilai akurasi pada data training dan data testing terjadi overfit, dimana nilai akurasi pada data training tinggi dan data testing rendah. Oleh karena hal tersebut, model prediksi analisis emosi berbasis text kurang optimal dengan menggunakan Random Forest dengan Word2Vec - Skipgram

**3.3 Perbandingan Model**

Analisis emosi berbasis teks menggunakan beberapa permodelan Text Mining dengan memanfaatkan dua metode text representation, yaitu TF-IDF dan Word2Vec – Skipgram. Keduanya memiliki nilai akurasi berbeda-beda setiap metodenya. Terdapat tiga model yang digunakan untuk melakukan analisis emosi, yaitu LSTM, Decision Tree, dan Random Forest. Dari seluruh permodelan, dalam analisis ini hanya terdapat dua model optimal untuk melakukan analisis emosi berbasis teks, yaitu model Decision Tree dan Random Forest dengan metode TF-IDF, dengan nilai akurasi pada Tabel 3.3.1

**Tabel 12**. Model Optimal

|  |  |  |
| --- | --- | --- |
| **Model** | **Akurasi** | |
| **Training** | **Testing** |
| **Decision Tree – TF-IDF** | **0.99** | **0.81** |
| **Random Forest – TF-IDF** | **0.99** | **0.83** |

**BAB 4**

**KESIMPULAN**

Analisis emosi berbasis text merupakan analisis yang memanfaatkan permodelan algortima text mining. Beberapa tahapan pengolahan data dengan EDA, data preprocessing, data representation untuk meningkatkan akurasi dan prediksi permodelan. Pada analisis ini, hanya dua model optimal yang digunakan untuk analisis emosi berbasis text, yaitu Model Decision Tree dengan metode representasi TF-IDF dan Model Random Forest dengab metode representasi TF-IDF. Ketidakseimbangan data dapat menyebabkan model bias terhadap kelas yang lebih dominan, hal ini dapat terlihat pada akurasi nilai training model decision tree – TF-IDF dan model random forest – TF-IDF memiliki nilai akurasi sebesar 0.99 yang berarti nilai ini masih overfit, tetapi masih bisa digunakan karena nilai akurasi pada data testing diatas 70%. Berdasarkan tabel 3.3.1 dapat terlihat bahwa kedua model memiliki nilai akurasi pada data training sama, oleh karena itu, untuk menentukan model terbaik dilihat dari perbandingan nilai akurasi pada data testing., dimana model Random Forest dengan metode text representation TF-IDF memiliki nilai akurasi lebih tinggi sebesar 0.83 atau 83% dibandingkan model Decision Tree dengan metode text representation TF-IDF. Kesimpulannya, untuk melakukan analisis emosi berdasarkan empat label, yaitu “anger”, “joy”, “fear”, dan “sadness” menggunakan model Random Forest dengan metode text representation TF-IDF.

**REFERENSI**

1. Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social network analysis and mining*, *11*(1), 81.
2. Hakak, N. M., Mohd, M., Kirmani, M., & Mohd, M. (2017, July). Emotion analysis: A survey. In *2017 international conference on computer, communications and electronics (COMPTELIX)* (pp. 397-402). IEEE.
3. Internet Live Stats. (n.d.-b). Twitter usage statistics. Retrieved June 21, 2024, from <https://www.internetlivestats.com/twitter-statistics/#:~:text=Every%20second%2C%20on%20average%2C%20around%206%2C000%20tweets%20are%20sent%20on%20Twitter%2C%20which%20corresponds%20to%20over%20200%20billion%20tweets%20per%20year>.
4. Merdiansah, R., Siska, S., & Ridha, A. A. (2024). Analisis Sentimen Pengguna X Indonesia Terkait Kendaraan Listrik Menggunakan IndoBERT. *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, *7*(1), 221-228.
5. Mustofa, A., & Novita, R. (2022). Klasifikasi Sentimen Masyarakat Terhadap Pemberlakuan Pembatasan Kegiatan Masyarakat Menggunakan Text Mining Pada Twitter. *Building of Informatics, Technology and Science (BITS)*, *4*(1), 200-208.
6. Gata, W. (2017). Akurasi text mining menggunakan algoritma k-nearest neighbour pada data content berita SMS. *Jurnal Teknologi Informasi*, 6, 1-13.
7. Asyhari, M. Y., Juwari, J., Hapsari, E. D., & Yulianto, S. (2023). Pendekatan Metode Kolokasi untuk Text Processing Ulasan Aplikasi Android Surveilans Penyebaran Covid-19 di Indonesia. *Journal Information System Development (ISD)*, *8*(1), 33-42.
8. Fikri, M. I., Sabrila, T. S., & Azhar, Y. (2020). Perbandingan metode Naïve Bayes dan Support Vector Machine pada analisis sentimen Twitter. *Jurnal Teknologi Informasi*, 10, 71-76.
9. Suryawan, I. W. B., Utami, N. W., & Fredlina, K. Q. (2023). Analisis Sentimen Review Wisatawan Pada Objek Wisata Ubud Menggunakan Algoritma Support Vector Machine. *Jurnal Informatika Teknologi dan Sains (Jinteks)*, *5*(1), 133-140.
10. Nur Akbar, Muhammad, and Nur Annisa Safitri Yusuf. 2022. “Analisis Sentimen Pengguna Indihome Dengan Metode Klasifikasi Support Vector Machine (SVM).” Journal Shift Vol 2(1):13 –21.
11. Dani, A. H., Puspaningrum, E. Y., & Mumpuni, R. (2024). Studi Performa TF-IDF dan Word2Vec Pada Analisis Sentimen Cyberbullying. *Router: Jurnal Teknik Informatika dan Terapan*, *2*(2), 94-106.
12. Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.
13. De Ville, B. (2013). Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, *5*(6), 448-455.
14. Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), 37-63.
15. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437. <https://doi.org/10.1016/j.ipm.2009.03.002>
16. Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), 37-63.
17. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002