**PROGRAM STUDI SARJANA SISTEM INFORMASI**

**LAPORAN PROYEK PEMROSESAN BAHASA ALAMI**



***Analysis Emotion Of Comment Topic Reddit With Multi-label Classification Using SVM Algorithm***

Oleh:

|  |  |
| --- | --- |
| 12S18027 | Christopher A Hutabarat |
| 12S18028 | Alda G M Lumban Gaol |
| 12S18032 | Sarah H M Siahaan |

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO**

**INSTITUT TEKNOLOGI DEL**

**LAGUBOTI**

**NOVEMBER 2021**

**DAFTAR ISI**

[DAFTAR TABEL 3](#_heading=h.30j0zll)

[BAB 1 PENDAHULUAN 1](#_heading=h.1ksv4uv)

[1.1](#_heading=h.44sinio) Latar Belakang 1

[1.2](#_heading=h.2jxsxqh) Tujuan 2

[1.3](#_heading=h.z337ya) Manfaat 2

[1.4 Ruang Lingkup 2](#_heading=h.3j2qqm3)

[BAB 2 RENCANA KERJA 3](#_heading=h.1fob9te)

[2.1](#_heading=h.2et92p0) Pembagian Tugas 3

[2.2](#_heading=h.tyjcwt) Jadwal Kerja 3

[REFERENSI 4](#_heading=h.3znysh7)

DAFTAR TABEL

[Tabel 1 Pembagian Tugas 3](#_heading=h.3dy6vkm)

[Tabel 2 Jadwal Kerja 3](#_heading=h.1t3h5sf)

# 

# BAB 1 PENDAHULUAN

## Latar Belakang

Reddit (reddit.com) adalah sebuah situs yang dibangun oleh Alexis Ohanian dan Steve Huffman pada tahun 2005. Reddit merupakan sebuah situs penyedia konten, dimana para penggunanya yang disebut sebagai *redditors* dapat membagikan maupun memberikan komentar terhadap konten-konten yang ada pada reddit. Konten dan komentar tersebut dapat disukai (*upvote*) atau tidak disukai (*downvote*) oleh *redditors* lain. Konten dan komentar *redditors* yang mendapat *downvote* akan diberikan poin karma, dan yang mendapat *upvote* akan mendapatkan penghargaan. Hal ini menjadi pendorong bagi para *redditors* untuk memposting konten yang baik, membuat komentar yang bermanfaat dan juga memberikan umpan balik yang relevan [1].

Salah satu cara yang dapat dilakukan untuk menilai kualitas komentar yang di*posting* pada reddit adalah dengan menggunakan *emotion analysis*. *Emotion analysis* adalah proses mengidentifikasi dan menganalisis emosi yang diungkapkan dalam suatu data tekstual. *Emotion analysis* dalam teks dapat membantu menentukan opini dan maksud afektif dari penulis, serta sikap, evaluasi, dan kecenderungan mereka terhadap berbagai topik [2].

Memahami opini publik dapat membantu kreator dalam mengambil keputusan [3]. Sama halnya dengan *redditors* pada *platform* reddit yang sebaiknya juga memahami opini *redditors* lainnya terhadap konten yang dihasilkan dan komentar yang diberikan. Sehingga peluang *redditors* akansemakin besar dalam menghasilkan konten dan memberikan tanggapan yang ditanggapi dengan aksi *upvote*.

*Multilabel classification* merupakan sebuah *task* prediktif pada *data mining* yang memungkinkan untuk mengasosiasikan sebuah data kedalam beberapa label. Setiap konten dan komentar yang ada pada reddit akan diberikan label dan bisa saja setiap konten dan komentar memiliki label lebih dari satu. Adapun label pada *dataset* tersebut digunakan untuk mempermudah proses klasifikasi. Terdapat beberapa metode klasifikasi yang dapat digunakan dalam klasifikasi multilabel data, namun pada proyek ini akan digunakan metode klasifikasi *Support Vector Machine* (SVM). Pada dasarnya SVM digunakan untuk menentukan *decision boundary* atau *hyperplane* antar dua kelas, sehingga akan menjadi tantangan apabila diimplementasikan pada *multilabel classification* karena secara native cara kerja pengklasifikasian SVM berbeda dengan klasifikasi multilabel. Sehingga, untuk mengatasi tantangan tersebut dapat digunakan *MultiOutputClassifier* SVM yang disediakan oleh *machine learning open source library* berbasis *python* yaitu Scikit-learn*.* Strategi ini merupakan pengklasifikasian multi target yang dapat digunakan untuk mendukung klasifikasi multilabel dengan SVM.

Oleh karena itu, tim proyek memutuskan untuk mengangkat proyek dengan judul “*Analysis Emotion Of Comment Topic Reddit With Multi-label Classification Using SVM Algorithm*”. Proyek diharapkan dapat berkontribusi memecahkan tantangan *multilabel classification* menggunakan SVM, menganalisis emosi pada komentar reddit dengan *multilabel classification* menggunakan metode *Support Vector Machine* (SVM) *algorithm,* dan melakukan pengujian akurasi terhadap metode tersebut.

## Tujuan

Tujuan dari pelaksanaan proyek ini adalah:

1. Berkontribusi memecahkan tantangan *multilabel classification* menggunakan metode SVM.
2. Menerapkan metode SVM dalam menganalisis emosi pada komentar di reddit dengan *multilabel classification.*
3. Untuk mengetahui bagaimana tingkat akurasi menggunakan metode SVM dalam melakukan *emotion analysis*.

## Manfaat

Manfaat dari pelaksanaan proyek ini adalah:

* + - 1. Bagi *redditors*

Proyek ini diharapkan dapat menjadi bahan evaluasi bagi para *redditors* dalam membagikan konten yang lebih baik.

1. Bagi Mahasiswa

Proyek ini diharapkan dapat memberi wawasan bagi tim proyek dalam menerapkan pemrosesan bahasa alami tepatnya SVM *algorithm* pada data teks reddit.

## 1.4 Ruang Lingkup

Ruang lingkup dari pelaksanaan proyek ini adalah:

1. Dataset yang digunakan adalah dataset g0\_emotions yang berasal dari Hugging Face yang berisi 58000 komentar reddit dengan label sebanyak 27 kategori.
2. Metode yang digunakan dalam pengerjaan proyek *multilabel classification* adalah *Support Vector Machine* (SVM).

# BAB 2 ISI

Bab ini berisikan penjelasan mengenai hasil analisis tim proyek terhadap data dan metode yang digunakan, desain pemrosesan bahasa alami yang disajikan dalam bentuk *flowchart,* implementasi kode program beserta keluarannya, dan hasil evaluasi yang dilakukan.

## Analisis

Sub bab ini berisikan penjelasan mengenai hasil tim proyek terhadap data dan metode yang digunakan.

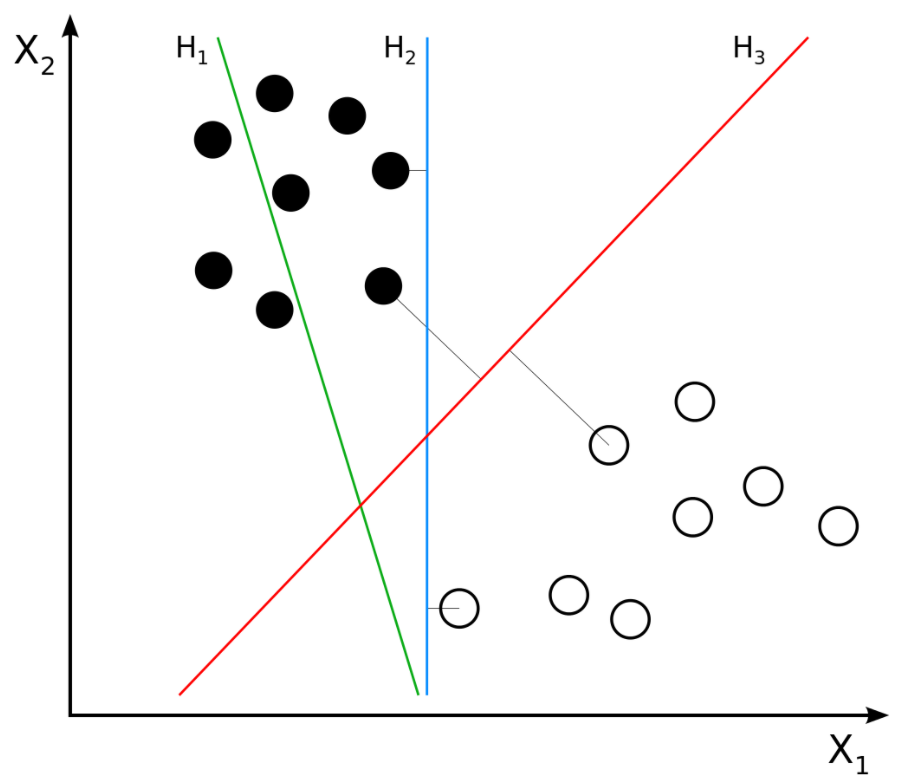
1. **Analisis Data**

Proyek ini menggunakan *dataset* yang bersumber dari tautan berikut: <https://huggingface.co/datasets/go_emotions>. *Dataset* GoEmotions ini berisikan kumpulan data beranotasi manual terbesar yaitu terdiri dari 58.000 *Reddit English comments,* yang dikategorikan kedalam 27 label kategori emosi dan sebuah label netral. *Dataset* GoEmotions ditujukan untuk keperluan pendalaman *multi-class, multi-label emotion classification.* Pada proyek ini data akan dibagi menjadi --- *data set* dan --- *data train.*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Atribut** | **Tipe Atribut** | **Deskripsi** |
| 1. | text | string | Teks komentar yang diposting |
| 2. | id | string | Id komentar yang diposting |
| 3. | author | string | Penulis komentar |
| 4. | subreddit | string | Topik khusus dalam komentar |
| 5. | link\_id | string | Id link dari komentar |
| 6. | parent\_id | string | Id parent dari komentar |
| 7. | created\_utc | float | label timestamp dari komentar |
| 8. | rater\_id | bool | Id unik dari pembuat catatan |
| 9. | example\_very\_unclear | boolean | Label yang menandakan bahwa komentar sangat sulit untuk diberi label oleh annotator (pembuat catatan) |
| 10. | admiration | integer | Kategori ekspresi yang menunjukkan kekaguman |
| 11. | amusement | integer | Kategori ekspresi yang menunjukkan sesuatu yang lucu dan menghibur |
| 12. | anger | integer | Kategori ekspresi yang menunjukkan perasaan tidak senang dan antagonisme |
| 13. | annoyance | integer | Kategori ekspresi yang menunjukkan perasaan terganggu dan jengkel |
| 14. | approval | integer | Kategori ekspresi yang mengungkapkan pendapat setuju |
| 15. | caring | integer | Kategori ekspresi yang mengungkapkan kepedulian terhadap orang lain |
| 16. | confusion | integer | Kategori ekspresi yang menunjukkan kebingungan dan ketidakpahaman |
| 17. | curiosity | integer | Kategori ekspresi yang menunjukkan keingintahuan terhadap sesuatu |
| 18. | desire | integer | Kategori ekspresi yang menunjukkan keinginan untuk mengharapkan sesuatu terjadi |
| 19. | disappointment | integer | Kategori ekspresi yang menunjukkan kekecewaan oleh tidak terpenuhinya suatu harapan |
| 20. | disapproval | integer | Kategori ekspresi yang mengungkapkan pendapat yang tidak setuju |
| 21. | disgust | integer | Kategori ekspresi yang menunjukkan rasa jijik oleh sesuatu yang tidak menyenangkan atau menyinggung |
| 22. | embarrassment | integer | Kategori ekspresi yang menunjukkan rasa sadar diri, malu dan canggung |
| 23. | excitement | integer | Kategori ekspresi yang menunjukkan rasa kegembiraan dan antusiasme yang besar |
| 24. | fear | integer | Kategori ekspresi yang menunjukkan rasa takut/khawatir |
| 25. | gratitude | integer | Kategori ekspresi yang menunjukkan rasa bersyukur |
| 25. | grief | integer | Kategori ekspresi yang menunjukkan rasa kesedihan mendalam |
| 27. | joy | integer | Kategori ekspresi yang menunjukkan rasa suka cita |
| 28. | love | integer | Kategori ekspresi yang menunjukkan rasa kasih sayang |
| 29. | nervousness | integer | Kategori ekspresi yang menunjukkan rasa gugup/khawatir/cemas |
| 30. | optimism | integer | Kategori ekspresi yang menunjukkan rasa berharap pada masa depan/keberhasilan sesuatu |
| 31. | pride | integer | Kategori ekspresi yang menunjukkan rasa puas atas pencapaian prestasi |
| 32. | realization | integer | Kategori ekspresi yang menunjukkan rasa menyadari sesuatu |
| 33. | relief | integer | Kategori ekspresi yang menunjukkan rasa lega dari kecemasan |
| 34. | remorse | integer | Kategori ekspresi yang menunjukkan rasa bersalah dan menyesal |
| 35. | sadness | integer | Kategori ekspresi yang menunjukkan sakit, sedih |
| 36. | surprise | integer | Kategori ekspresi yang menunjukkan rasa heran, terkejut oleh sesuatu yang tidak terduga |
| 37. | neutral | integer | Kategori ekspresi yang menunjukkan rasa netral atau biasa saja |

1. **Analisis Metode**

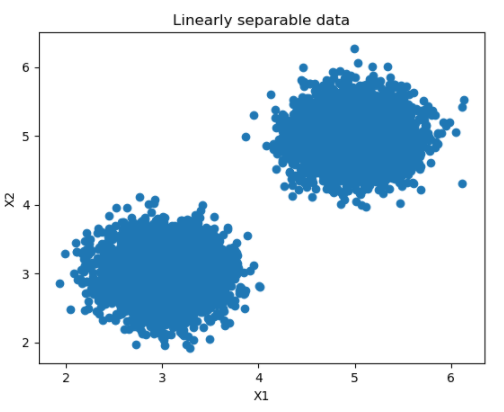
Support Vector Machine (SVM) adalah kelas algoritma machine learning yang menggunakan fungsi kernel untuk mempelajari decision boundary atau batas keputusan antara dua kelas. Decision boundary ini adalah margin maksimum antara dua kelas, artinya batasan ini akan berjarak sama dari kelas satu dan dua. Penentuan *boundary* yang disebut juga sebagai *hyperplane* akan dilakukan dengan bantuan dari *support vectors*.



Gambar diatas menunjukkan fitur dengan dua dimensi yaitu sumbu X1 dan X2 serta tiga buah garis (*hyperplane*) satu dimensi yaitu H1, H2, dan H3. Selanjutnya H2 dan H3 adalah *decision boundary* karena kedua garis ini dapat memisahkan satu kelas dengan kelas lainnya. Melalui gambar tersebut juga dapat diketahui bahwa H3 adalah *hyperplane* terbaik diantara *hyperplane* yang lainnya, karena H3 memisahkan kedua kelas dengan jarak/margin maksimum yang sama sehingga H3 memiliki margin tertinggi antara kelas dibandingkan dengan *hyperplane* lainnya.

Sayangnya, SVM kurang mampu mengatasi klasifikasi multilabel karena decision boundary pada SVM hanya mampu memisahkan dua kelas untuk memastikan margin maksimumnya. Jika kelas lain ditambahkan pada klasifikasi SVM maka jarak antara kelas menjadi tidak maksimal. Namin, keterbatasan tersebut dapat diatasi dengan menggunakan *MultiOutputClassifier* SVM yang disediakan oleh *machine learning open source library* berbasis *python* yaitu Scikit-learn*.* Penerapan SVM *MultiOutputClassifier* dengan Scikit-learn Scikit-learn menyediakan fungsionalitas *MultiOutputClassifier,* yang mampu mengimplementasikan pengklasifikasi multilabel untuk pengklasifikasi reguler apa pun. Strategi ini merupakan pengklasifikasian multi target yang dapat digunakan untuk mendukung klasifikasi multilabel dengan SVM. Berikut uraian *line scenario-*nya:

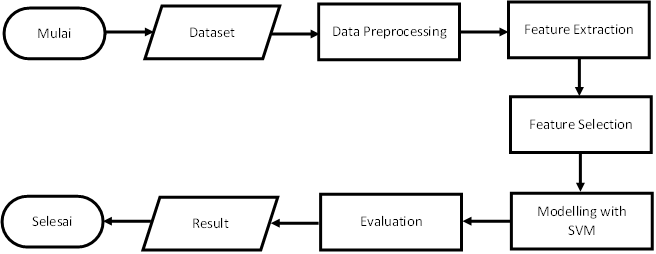
1. Bagi dataset secara acak menjadi dua bagian. Asumsikan data dapat dibagi secara linear.



1. Lalu, pastikan bahwa semua dependensi diimpor seperti pyplot API dari Matplotlib untuk memvisualisasikan hasil. Dalam hal ini Numpy akan digunakan pada beberapa pemrosesan angka, dan beberapa dependensi sklearn akan diimpor lagi sesuai kebutuhan. Secara khusus *make\_blobs* dapat digunakan dalam pembuatan data, *MultiOutputClassifier* untuk pengklasifikasi multilabel, LinearSVC untuk SVM , *train\_test\_split* untuk membagi data menjadi set pelatihan dan pengujian, hingga *multilabel\_confusion\_matrix* dan *ConfusionMatrixDisplay* untuk menghasilkan dan memvisualisasikan matriks konfusi.
2. Tentukan opsi-opsi konfigurasi, seperti jumlah sampel yang akan dihasilkan, pusat *cluster*, dan jumlah kelas.
3. Pada tahap ini, telah dihasilkan data yang sesuai dengan spesifikasi diatas. Buatlah array dengan bentuk yang sama untuk label kedua – warna. Inisialisasi secara acak demi kesederhanaan.
4. Gabungkan label pelatihan menjadi satu larik sehingga menghasilkan pemisahan antara data pelatihan dan pengujian.
5. Inisialisasi classifier SVM dan ubah menjadi multilabel.
6. .fit data ke classifier, yang artinya proses pelatihan dimulai. Setelah pemasangan selesai, classifier terlatih tersedia di multilabel\_classifier. Kini .predict telah dapat dipanggil untuk menghasilkan prediksi untuk data uji.
7. Bandingkan y\_test (label kebenaran dasar yang sebenarnya) dan y\_test\_pred (label yang diprediksi) menggunakan matriks konfusi (mengikuti langsung setelah segmen kode).
8. Buatlah matriks konfusi untuk setiap label dengan multilabel\_confusion\_matrix, dan kemudian memplotnya dengan ConfusionMatrixDisplay menggunakan Matplotlib.

## Design

Pada sub bab ini dijelaskan desain pemrosesan bahasa alami dari *Analysis Emotion Of Comment Topic Reddit With Multi-label Classification Using SVM Algorithm* yang dapat dilihat pada gambar [nomor gambar].



1. Data Preprocessing

Pada tahap ini dilakukan pra-pemrosesan data dengan mengolah data pada dataset yang masih terdapat noise, tidak lengkap dan tidak konsisten serta menghilangkan kata kata yang tidak diperlukan.Teknik pra-pemrosesan data dapat meningkatkan kualitas data, sehingga membantu meningkatkan akurasi dan efisiensi[buku].

1. Data Cleaning

Pada tahap ini dilakukan pembersihan terhadap data yang tidak lengkap, *noise,* dan data yang tidak konsisten. *Data cleaning* akanmencoba untuk mengisi data-data yang tidak lengkap, menghaluskan *noise* saat mengidentifikasi *outlier,* dan melakukan perbaikan pada data-data yang tidak konsisten[buku].

1. Case Folding

*Case folding* merupakan proses yang dilakukan untuk menyeragamkan karakter pada data. Proses *case folding* adalah proses mengubah seluruh huruf menjadi huruf kecil. Pada proses ini karakter-karakter ‘A’-‘Z’ yang terdapat pada data diubah kedalam karakter ‘a’-‘z’. Karakter-karakter selain huruf ‘a’ sampai ‘z’ (tanda baca dan angka-angka) akan dihilangkan dari data dan dianggap sebagai delimiter [Fithri Selva Jumeilah (2017)].

1. Tokenization

Pada tahap ini dilakukan pemecahan kalimat menjadi kata-kata sehingga membentuk token yang bermakna. Strategi umum yang dilakukan pada tahap tokenizing adalah memotong kata pada white space /spasi dan membuang karakter tanda baca. Tahap tokenizing membagi urutan karakter menjadi kalimat dan kalimat menjadi token.

1. Stopwords Removal

Setelah tahap tokenization, maka dilakukan tahap filtering yaitu dengan menghapus kata-kata yang sangat umum (stopwords removal). Adapun kata yang termasuk dalam stopword contohnya adalah “this”, “there”, “the”, “a”, “of” dan lain lain.

1. Stemming

Pada tahap ini dilakukan pembersihan terhadap imbuhan awalan dan akhiran dari sebuah kata[Uma(2019)]. Adapun imbuhan yang dibersihkan yaitu “ss”, “ies”, “ing”, “ed”, “ate”, “ize”, “able”, “al”, dan lain-lain. Contohnya “automates”, “automatic”, “automation” di stemming menjadi “automat”.

1. Lemmatization

Pada tahap ini mirip dengan stemming, namun dilakukan pencocokan dengan lemma-lemma yang terdapat pada kamus sehingga hasil akan lebih akurat [Uma(2019)]. Adapun imbuhan yang dibersihkan yaitu “-s”, “-‘s”, “am”, “are”, “is” dan lain-lain.

1. Feature Extraction

Feature Extraction adalah proses mengubah dokumen teks dari format apa pun menjadi daftar fitur yang dapat dengan mudah diproses dengan teknik klasifikasi teks. Feature extraction akan sangat mendukung dalam menyediakan fitur yang baik bagi training model. Cara yang akan digunakan untuk melatih model tekstual adalah dengan TF IDF Vectorization dan Word2Vec.

1. Feature Selection

*Feature selection* adalah proses ekstraksi terhadap *optimal subset* dari atribut yang menyimpulkan arti yang sama dengan *complete attribute set* [Neha (2017)]*.* Dengan melakukan ekstraksi, akan diketahui bagaimana pola dan urutan kata dalam teks komentar. *Feature selection* dalam proyek ini akan dilakukan dengan *unigram, bigram,* dan *trigram* yang menghasilkan *document term matrix* dengan *N-gram.*

1. Modelling with SVM

Setelah *feature* berhasil diseleksi, maka langkah selanjutnya adalah melakukan *train* pada dataset dengan menggunakan algoritma *Support Vector Machine* (SVM). SVM adalah salah satu metode pada *supervised learning*, modeling dengan algoritma SVM bekerja dengan membangun *hyperplane* yaitu fungsi pemisah antar kelas dengan margin maksimal. Pengklasifikasian dengan SVM menghasilkan data yang akurat meskipun data yang digunakan kurang seimbang (*imbalance*) [Yuchun Tang (2009)].

Diagram, schematic

Description automatically generated

Teknik pemodelan dengan SVM akan menggunakan 4 cara yaitu:

* SVM Model with TF IDF
* SVM Model with Word2Vec
* SVM Model with K-Fold Cross Validation
* Linear SVM Classifier

1. Evaluation and Result

Setelah *modelling with SVM* selesai, selanjutkan akan dilakukan evaluasi terhadap model yang dibangun menggunakan *confusion matrix. Confusion matrix* akan memberikan perbandingan dari hasil klasifikasi model dengan hasil aktual klasifikasi. Berikut adalah tabel *confusion matrix:*

Table

Description automatically generated

Hasil klasifikasi dipresentasikan dalam 4 istilah yaitu:

* TP adalah *true positive* atau data positif yang diprediksi benar
* FN adalah *false negatif* atau data positif yang diprediksi sebagai data negatif
* FP adalah *false positif* atau data negatif yang diprediksi sebagai data positif
* TN adalah *true negative* atau data negatif yang diprediksi benar

Pengukuran *performance metrics* dari *confusion matrix* akan dilakukan mulai dari melakukan perhitungan terhadap *accuracy score, precision, recall* dan *F1-Score.* berikut penjelasannya:

1. *Accuracy score* untuk mengetahui tingkat keakuratan klasifikasi yang dihasilkan model

Text

Description automatically generated with medium confidence

1. *Precision* untuk mengetahui tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model

*Text

Description automatically generated*

1. *Recall* untuk mengetahui tingkat keberhasilan model dalam menemukan kembali suatu informasi.



## Implementasi

Sub bab ini berisikan penjelasan mengenai implementasi tim proyek terhadap data dan metode yang digunakan.

1. Data Preprocessing

*Data preprocessing* akan meningkatkan kualitas data, sehingga membantu meningkatkan akurasi dan efisiensi. Pada tahap ini akan dilakukan *data cleaning, case folding, tokenization, stopwords removal, stemming,* dan *lemmatization.* Namun sebelumnya, dilakukan *import library, import* dataset, pemeriksaan deskripsi dataset, dan pemeriksaan jumlah *row* dataset. Berikut adalah kode program *import library:*

|  |
| --- |
| import pandas as pd  from pandas import DataFrame as df |

Berikut adalah kode program *import dataset:*

|  |
| --- |
| df\_go1 = pd.read\_csv("goemotions\_1.csv")  df\_go2 = pd.read\_csv("goemotions\_2.csv")  df\_go3 = pd.read\_csv("goemotions\_3.csv")  df\_go\_emotion = pd.concat([df\_go1, df\_go2, df\_go3]) |

Berikut adalah kode program pemeriksaan deskripsi dataset:

|  |
| --- |
| df.describe(df\_go\_emotion) |

Berikut adalah kode program pemeriksaan jumlah *row* dataset:

|  |
| --- |
| index = df\_go\_emotion.index  number\_of\_rows = len(index)  print(number\_of\_rows) |

1. Data Cleaning

*Data cleaning* akan mencoba untuk mengisi data-data yang tidak lengkap, menghaluskan *noise* saat mengidentifikasi *outlier,* dan melakukan perbaikan pada data-data yang tidak konsisten. Pada tahap ini akan dilakukan *drop column* terhadap kolom yang tidak dibutuhkan*,* pengecekan terhadap terhadap *data null,* dan identifikasi terhadap id yang unik. Berikut adalah kode program untuk melakukan *drop column* terhadap kolom yang tidak dibutuhkan:

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

Berikut adalah kode program untuk melakukan pengecekan terhadap terhadap *data null:*

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

Berikut adalah kode program untuk melakukan *drop column* terhadap kolom yang tidak dibutuhkan:

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Case Folding

*Case folding* akan mengubah semua teks komentar menjadi *lowercase format.* Berikut adalah kode program untuk melakukan *lowercase formatting*:

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Tokenization

*Tokenization* akan memecah teks komentar menjadi unit yang relevan, seperti *words, symbol, phares,* ataupun bentuk token lainnya. Berikut adalah kode program untuk melakukan *tokenization*:

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Stopwords Removal

*Stopwords removal* akan menghilangkan kata-kata seperti “this”, “there”, “the”, “a”, “of”, “with”, “have”, dan lain sebagainya. Berikut adalah kode program untuk melakukan *stopwords removal:*

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Stemming

*Stemming* akan menghilangkan imbuhan pada setiap kata untuk memperoleh *stem* atau kata dasar pada teks komentar. Berikut adalah kode program untuk melakukan *stemming*:

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Lemmatization

*Lemmatization* akan mengubah kata menjadi bentuk dasarnya sesuai dengan kata-kata yang terdapat dalam kamus. Berikut adalah kode program untuk melakukan *lemmatization*:

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Feature Extraction

*Feature extraction* akan dilakukan terhadap teks komentar menggunakan TF IDF dan Word2Vec. Berikut adalah kode program untuk melakukan *feature extraction*:

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Feature Selection

*Feature selection* dapat dilakukan dengan unigram, bigram, dan trigram. Berikut adalah kode program untuk melakukan *feature selection*:

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

*Output* kode program diatas adalah sebagai berikut:

1. Modeling with SVM

*Modeling with SVM,* merupakan tahap pemodelan terhadap *sentiment analysis* yang akan dilakukan. Dalam menerapkan SVM *algorithm,* akan dilakukan 4 tahap berikut ini:

1. *SVM Model with TF IDF*

Berikut adalah kode program untuk membangun model SVM dengan TD IDF:

|  |
| --- |
|  |

1. *SVM Model with Word2Vec*

Berikut adalah kode program untuk membangun model SVM dengan Word2Vec:

|  |
| --- |
|  |

1. *SVM Model with K-Fold Cross Validation*

Berikut adalah kode program untuk membangun model SVM dengan *K-Fold Cross Validation:*

|  |
| --- |
|  |

1. *Linear SVM Classifier*

*Linear SVM Classifier* akan dilakukan setelah label yang digunakan dalam *sentiment analysis* didefinisikan terlebih dahulu. Berikut adalah kode program untuk mendefinisikan label yang digunakan dalam *sentiment analysis:*

|  |
| --- |
|  |

Dilanjutkan dengan memisahkan data menjadi *train data* dan *test data.* Berikut adalah kode program untuk memisahkan *train data* dan *test data:*

|  |
| --- |
|  |

Berikut adalah kode program untuk membangun model SVM dengan *Linear SVM Classifier:*

|  |
| --- |
|  |

## Hasil

1. Evaluation SVM Model – TF IDF
2. Evaluation SVM Model – Skipgram Word2Vec
3. Evaluation SVM Model with K-Folds Cross Validation
4. SVM Model with Misclassify
5. Accuracy SVM Model with OneVsRestClassifier

# BAB 3 PENUTUP

## Kesimpulan

## Saran

## Rencana Kerja

1. **Pembagian Tugas**

Tabel 1 Pembagian Tugas

|  |  |  |
| --- | --- | --- |
| NIM | NAMA | KONTRIBUSI PEMBAGIAN TUGAS |
| 12S18027 | Christopher A Hutabarat | 33,33% |
| 12S18028 | Alda G M Lumban Gaol | 33,33% |
| 12S18032 | Sarah H M Siahaan | 33,33% |

Berikut tugas yang dilakukan oleh tim proyek:

* Data Analyst

Mengidentifikasi dan menafsirkan data model kemudian melakukan analisis terhadap strategi yang efisien untuk diimplementasikan pada proyek.

* Programmer

Merancang dan mengimplementasikan *code* untuk membangun sistem sesuai hasil analisis *data analyst.*

1. **Jadwal Kerja**

Tabel 2 Jadwal Kerja

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Task | Minggu ke- | | | |
| 1 | 2 | 3 | 4 |
| 1. | Membuat Proposal Proyek  - Analisis Latar Belakang  - Pemilihan Metode  - Presentasi Proposal |  |  |  |  |
| 2. | Analisis   * Analisis Data * Analisis Metode |  |  |  |  |
| Desain  - Data Preprocessing  - Feature Extraction  - Feature Selection  - Modelling with SVM |
| 3. | Implementasi  - Exploratory Data Analysis (EDA)  - Data Preprocessing  - Feature Extraction  - Feature Selection  - Modelling with SVM |  |  |  |  |
| 4. | Evaluasi Akurasi Model |  |  |  |  |
| 5. | Laporan Akhir dan Presentasi Akhir |  |  |  |  |

# REFERENSI

|  |  |
| --- | --- |
| [1] | Anderson, "Ask me anything: what is Reddit?," *Library Hi Tech News,* vol. 32, pp. 8-11, 2015. |
| [2] | S. Aman and S. S. , "Identifying expressions of emotion in text," in *International Conference on Text, Speech and Dialogue*, Springer, Berlin, Heidelberg, 2007. |
| [3] | J. K. Rout, "A model for sentiment and emotion analysis of unstructured social media text," *Electron Commer Res ,* no. 18, pp. 181-199, 2017. |
| [4] | K. I. Gunawan and J. Santoso, "Multilabel Text Classification Menggunakan SVM dan Doc2Vec Classification pada Dokumen Berita Bahasa Indonesia," *Journal of Information System, Graphics, Hospitality and Technology,* vol. 3, no. 1, pp. 29-38, 2021. |