Depression Detection in Social Media by Analyzing User's Sentiment (Naive Bayes)

Darren Vernon Riota[1], Jericho Cristofel Siahaya[2], Muhammad Rizky Azzakky[3], Ricky Ng[4]
Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1]darren.vernon@student.umn.ac.id, [2]jericho.cristofel@student.umn.ac.id, [3]rizky.azzakky@student.umn.ac.id, [4]ricky.ng@student.umn.ac.id

***Abstract*—The rise of technology and the internet, along with the advent of social media has presented a promising new methodology for the early detection of depression. These methods, mostly based on machine learning techniques using a sort of statistical formula to calculate input and produce insight. These days, social media has become a part of human life. This is due to the fact that social media companies are always growing every year. From teenagers to adults are highly attached to social media and this habit creates a new era in the field of data processing, especially with textual data. In early 2020, it was reported that Twitter, one of the largest social media platforms, produces approximately 500 million tweets every day. It's a huge streaming of data from a single company. However, this new era also opens a new opportunity for building a system that can solve a complex problem such as detecting depression using machine learning models. In this work, we propose the use of Naive Bayes method to classify depressed and not depressed tweets (text) based on the tweet's sentiment. We evaluated our model using recall and precision to see how many tweets that are correctly corrected by our classifier. Experimental results show that our model was able to outperform previous work with different method and algorithm.**

Keywords—depression detection, twitter, naive bayes, text classification

I. PENDAHULUAN

Setiap tahunnya, depresi menyebabkan sekitar 800.000 kasus bunuh diri di seluruh dunia [1]. Depresi merupakan salah satu tipe penyakit mental (*mental illness*) yang menyebabkan gangguan pada gaya hidup sehari-hari, seperti ketidakmampuan untuk berinteraksi sosial maupun keputusasaan dan perasaan sedih terus menerus; tidak mampu merasakan kebahagiaan. Beberapa faktor yang berperan dalam menyebabkan depresi antara lain mencakup faktor genetik, biologis, sosial dan stres [2][3].

Dewasa ini, teknologi telah berkembang dengan cukup pesat, menciptakan kemudahan untuk saling berkomunikasi hingga bertukar informasi, terutama di dalam media sosial. Twitter memiliki kontribusi yang besar dalam kemudahan pertukaran informasi ini, khususnya dalam media berupa teks. Dikutip dari sebuah sumber, menyatakan bahwa setidaknya ada 500 juta teks (tweet) yang diunggah di Twitter setiap harinya [4]. Setiap tweet yang diunggah memiliki konteks yang bervariasi dan tidak jarang tweet yang diunggah mencerminkan perasaan si pengguna media sosial itu sendiri.

Melalui unggahan-unggahan di media sosial, perasaan atau emosi seseorang dibalik suatu unggahan spesifik dapat dicerminkan dengan jelas. Dengan ini, perasaan pengguna yang seringkali terikat atau terkait dengan kesehatan mental dapat diprediksi dengan menggunakan teknik klasifikasi teks (*text classification*).

Dalam penelitian ini, tim peneliti akan membangun sebuah NLP Tool yang bisa diimplementasikan dalam bentuk Web Application ataupun API, yang menggunakan *machine learning* untuk mengklasifikasi kesehatan mental seseorang berdasarkan unggahan-unggahannya (tweet*)* pada sosial media Twitter ke dalam 2 kategori, yaitu *depressed* atau *not depressed*. Peneliti memilih algoritma Naive Bayes untuk membangun alat klasifikasi teks (*text classification)* yang memiliki tingkat akurasi yang tinggi. Dengan ini, pengguna yang cenderung menunjukkan tanda-tanda depresi dapat dideteksi dengan awal, sehingga terdapat semakin banyak waktu untuk ditangani atau diobati oleh para psikologis, yang mana akan berdampak pada menurunnya tingkat depresi secara *general* di seluruh dunia.

II. KAJIAN TEORI

A. Sentiment Analysis

Sentimen adalah pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu (bertentangan dengan pertimbangan pikiran)[5]. Setiap manusia umumnya akan mengungkapkan respon terkait suatu kejadian/keadaan berdasarkan pengalamannya yang merupakan hasil kumulatif dari proses berpikir terhadap *input* yang pernah diterima olehnya selama ini. Respon tersebut umumnya dapat dikategorisasikan ke dalam tiga kelompok besar yaitu positif, negatif dan netral.

Sentiment Analysis sendiri adalah salah satu teknik dalam mengekstrak informasi yang berupa pandangan (Sentimen) seseorang terhadap suatu isu atau kejadian [6]. *Sentiment analysis* dapat digunakan untuk mengungkap opini publik terhadap suatu isu, kepuasan pelayanan, kebijakan, prediksi harga saham dan analisis pesaing berdasarkan data tekstual [6]. Namun, fenomena pertumbuhan data yang terjadi secara eksponensial menjadi tantangan baru dalam proses analisis sentimen. Pendekatan konvensional bukan lagi jawaban tepat untuk mengungkap dan menentukan jenis sentimen dalam data tekstual. Mempekerjakan manusia untuk mengklasifikasikan jenis sentimen dari suatu kumpulan data tekstual yang sangat besar dan beragam tentu akan membutuhkan waktu dan biaya yang tidak sedikit. Bayangkan jika kita memiliki 100.000 ribu tweet per hari yang harus ditentukan satu per satu jenis sentimen nya, pasti bukan hanya membutuhkan waktu yang lama, tapi juga membutuhkan sumber daya yang sangat besar. Oleh karena itu, dibutuhkan sebuah teknik baru dalam analisis sentimen yang dapat secara otomatis mengekstrak informasi dari data secara cepat dan bertanggung jawab.

B. Naive Bayes

Algoritma Naive Bayes merupakan sebuah teorema yang bekerja pada *conditional probability*. Artinya, Naive Bayes menghitung suatu probabilitas *event* yang akan terjadi berdasarkan perhitungan probabilitas kejadian-kejadian yang sudah pernah terjadi sebelumnya [7]. Naive Bayes merupakan algoritma yang paling populer untuk digunakan dalam klasifikasi teks.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Gambar 1.0 - Naive Bayes *Formula*

Cara Naive Bayes bekerja:

1. Menghitung frekuensi kata dari kedua kelas (untuk *binary classification*) dan membuat tabel frekuensi sebagai model awal.
2. Input (kalimat) baru yang masuk lalu di-tokenisasi untuk kemudian dihitung masing-masing *probability*-nya dari kedua kelas.
3. Dalam *frequency table*, setiap kata diharuskan untuk memiliki skor > 0 (tidak boleh ada 0). Untuk menghilangkan 0, dilakukan *Laplace Smoothing*.

- *Laplace Smoothing*

Dengan menerapkan Naive Bayes pada sebuah data terkadang juga menyebabkan *misclassification* jika data *training* hanya sedikit atau data-nya tidak kaya sehingga data *testing* tidak ditemukan pada data *training*, dan menyebabkan hasil probabilitas bernilai 0 (zero) dan menyebabkan *error* pada proses klasifikasi [8].

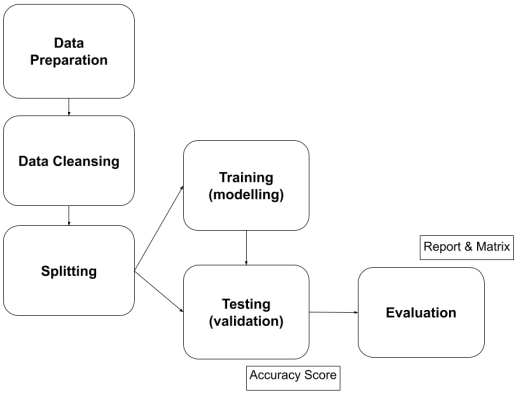
Kekurangan dari algoritma Naive Bayes ini dapat diminimalisir dengan melakukan *smoothing* menggunakan *laplace smoothing* atau juga dikenal dengan *additive smoothing*. *Laplace smoothing* berfungsi untuk mencari probabilitas terkecil dari sebuah *event* yang sama sekali belum pernah terjadi ataupun belum ada pada data *training*.

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d),$$

Gambar 2.0 - *Laplace Smoothing Formula*

III. METODOLOGI PENELITIAN

A. Framework (Kerangka Kerja)



Gambar 3.0 - *Framework* Penelitian

B. Preparation

Pada penelitian ini, peneliti menggunakan dataset sekunder Sentiment140 yang bersumber dari Kaggle [9]. Agar model yang dibuat dapat memiliki performa yang bagus, maka peneliti membutuhkan dataset yang cukup banyak sehingga model dapat melakukan klasifikasi yang sesuai. Dataset Sentiment 140 dengan jumlah data yang seimbang ini dirasa cukup untuk membuat model yang bagus. Pada tahapan implementasi, peneliti menggunakan bahasa pemrograman Python 3 dan Jupyter Notebook untuk melakukan pra-pengolahan data, kalkulasi, serta membuat model klasifikasi. Adapun beberapa *libraries* yang digunakan untuk membantu proses implementasi yaitu:

- Numpy
- Pandas
- scikit-learn
- Matplotlib, Seaborn
- Flask (untuk *deployment*)

C. Tahap Implementasi

- Data Preparation

Dataset Sentiment140 berisi 1,600,000 data *Tweets* yang seimbang untuk kedua kelas (800,000:800,000).

Tabel 1.0 - Distribusi Data

DATASET	
POSITIF	800,000
NEGATIF	800,000
TOTAL	**1,600,000**

Dataset ini berisikan beberapa *fields: Target, ID, Date, Flag, User, Text. Fields* yang akan peneliti gunakan adalah Target dan Text.

- *Data Cleansing*

Pada tahapan ini dilakukan pra-pengolahan data yang bertujuan untuk membuat data yang berupa teks menjadi lebih dapat dipahami oleh komputer. Beberapa proses yang dilakukan meliputi menghilangkan tanda baca, menghilangkan emoji atau *emoticon*, menghilangkan tautan atau kata yang memiliki format berupa tautan, mengubah teks menjadi huruf kecil (*lowercase*), menghilangkan *stopwords* atau kata-kata umum yang sering digunakan (*I. you, they,* dll.), dan yang terakhir adalah melakukan tokenisasi untuk untuk memisahkan data yang berupa tweet tadi menjadi unit yang lebih kecil yaitu kata per kata .

Perbandingan teks yang sebelum di-*cleansing* sudah di-*cleansing* adalah seperti ini :

target	text
0	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
0	is upset that he can't update his Facebook by texting it... and might cry as a result School

	today also. Blah!
0	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds

Tabel 4 - Teks sebelum *cleansing*

Text	Label
switchfoot awww thats bummer shoulda got david carr third day	0
upset cant update facebook texting might cry result school today also blah	0
kenichan dived many times ball managed save 50 rest go bounds	0

Tabel 5 - Teks sesudah *cleansing*

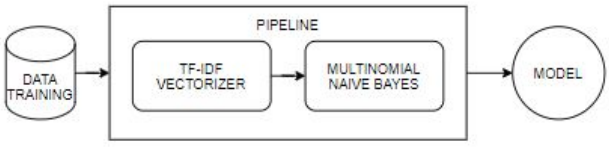
- *Model classification*

  Setelah data telah selesai di-*cleansing*, maka data siap dipakai untuk membuat model klasifikasi menggunakan Naive Bayes. Pada penelitian ini, peneliti membagi (*split*) dataset ke dalam dua kelompok, yaitu *training* dan *testing*.

  Untuk membuat model dengan performa klasifikasi yang bagus maka dibutuhkan data *training* yang banyak (variasi) sehingga model Naive Bayes mampu melakukan prediksi dengan baik. Peneliti menggunakan 95% data pada *training* dan 5% data pada *testing*.

  - **Training**

    Pada tahapan *training* peneliti menggunakan *pipeline* sehingga mempermudah proses kalkulasi. *Pipeline* terdiri dari dua proses, yaitu vektorisasi dan kalkulasi naive bayes. Di sini peneliti menggunakan vektorisasi Tf-idf untuk menghitung frekuensi dari tiap kata yang ada pada data *training*.



Gambar 6 - *Pipeline*

Setelah vektorisasi dilakukan, maka kemudian dilakukan kalkulasi naive bayes untuk menghitung *probability* tiap kata dari kedua kelas. Pada tahapan kalkulasi, peneliti menggunakan Multinomial Naive Bayes (MultinomialNB).

Beberapa parameter yang digunakan peneliti pada *modelling* data *training* adalah sebagai berikut:

- **N-gram** = pada rentang satu sampai tiga kata (1, 3).
- **α smoothing** = 10, untuk lebih meningkatkan performa model sekaligus meminimalisir misklasifikasi.

  - **Testing**

    Setelah melakukan *modelling* pada data *training* kemudian dilakukan validasi menggunakan 5% data *testing*. Performa dari model diukur berdasarkan validasi skor dari akurasi (*accuracy score*), dengan rumus sebagai berikut:

$$ACC = \frac{\text{TRUE POSITIVES + TRUE NEGATIVES}}{\text{ALL SAMPLES}}$$

Gambar 7 - *Accuracy Score Formula*

Namun, peneliti tidak hanya menggunakan skor dari akurasi tetapi juga sensitivitas (*recall*) dan presisi (*precision*) dari hasil model klasifikasi.

$$REC = \frac{\text{TRUE POSITIVES}}{\text{TRUE POSITIVES + FALSE NEGATIVES}}$$

$$PRE = \frac{\text{TRUE POSITIVES}}{\text{TRUE POSITIVES + FALSE POSITIVES}}$$

Gambar 8 - *Recall & Precision Formula*

IV. EVALUASI

Untuk menguji performa dari model klasifikasi yang telah dibuat, peneliti menggunakan perhitungan *accuracy*, *recall* serta *precision* dari *confusion matrix*.

Tabel 2.0 - *Classification Report*

	precision	recall	f1-score	support
1	0.85	0.72	0.78	40001
0	0.76	0.87	0.81	39999
accuracy			0.80	

Accuracy digunakan untuk melihat rasio data yang diprediksi dengan benar oleh model dari total keseluruhan data yang ada. Namun, di sini peneliti tidak menggunakan *accuracy* sebagai acuan utama melainkan *recall* dan *precision*. *Recall* digunakan untuk melihat sensitivitas pada rasio prediksi benar positif (*true positive*) dari keseluruhan data yang memang benar positif. Sedangkan, *precision* digunakan untuk melihat rasio prediksi benar negatif (*true negative*) dari keseluruhan data yang diprediksi negatif.

Alasan utama peneliti menggunakan kedua acuan metrik tersebut adalah dikarenakan pada model klasifikasi depresi

ini metrik utama yang harus lebih difokuskan adalah prediksi orang-orang yang tidak depresi namun sebenarnya mengalami depresi (*false negative*) yang mana memiliki resiko atau *cost* yang lebih besar dibandingkan orang-orang yang diprediksi depresi namun sebenarnya tidak mengalami depresi (*false positive*). Maka, *recall* digunakan untuk melihat seberapa bagus performa model untuk memprediksi orang yang tidak depresi dari keseluruhan data yang memang benar tidak depresi. Sedangkan, *precision* digunakan untuk melihat seberapa bagus performa model untuk memprediksi orang yang depresi dari keseluruhan data orang yang memang benar mengalami depresi.

Pada penelitian ini hasil dari *recall* pada kategori negatif (orang yang tidak depresi) adalah 0.87 atau 87% yang mana lebih dari sama dengan 0.85 (≥ 85%) untuk membuktikan bahwa performa model sudah cukup bagus. Sedangkan, hasil dari *precision* pada kategori positif (orang yang depresi) adalah 0.85 atau 85%.

Tabel 3.0 - *Confusion Matrix*

	1	0
1	28802	11199
0	5094	34905

Dari hasil *confusion matrix*, bisa terlihat bahwa jumlah *false negative* atau orang yang diprediksi tidak depresi namun sebenarnya depresi sangat sedikit dengan jumlah dari hasil validasi 5049 yang membuktikan bahwa performa model sudah cukup bagus untuk tidak salah mengklasifikasi orang-orang yang sebenarnya depresi namun diprediksi/dinyatakan tidak depresi. Sedangkan, pada *false positive* atau orang yang diprediksi depresi namun sebenarnya tidak depresi cukup banyak dan ini tidak menjadi masalah dikarenakan resikonya tidak terlalu besar dibanding salah memprediksi orang yang tidak depresi namun sebenarnya mengalami depresi.

V. KESIMPULAN

A. Kesimpulan

Pada penelitian ini, peneliti berhasil membuat model klasifikasi untuk memprediksi depresi berdasarkan teks tweet, dengan hasil akurasi (*accuracy*): 80%, sensitivitas (*recall*): 87%, dan presisi (*precision*): 85%. Sehingga, performa dari model ini dapat dikatakan sudah cukup bagus untuk mengklasifikasi orang yang depresi dan tidak depresi.

Peneliti juga membandingkan hasil dengan algoritma lain, yaitu *Support Vector Machine*. Metode SVM dapat digunakan untuk membangun sistem deteksi depresi lewat sentiment *Twitter* dengan hasil akurasi 61.54%[10].

B. Limitasi

Algoritma Naive Bayes bekerja menggunakan teorema bayes yang diimplementasikan secara naive, sehingga mengasumsikan setiap kata saling independen dan tidak melihat konteks dari satu kalimat tersebut. Sehingga algoritma ini tidak dapat menginterpretasikan konteks sebuah kalimat dengan artinya sesungguhnya di dunia nyata. Selain itu, naive bayes juga memiliki kekurangan ketika jumlah data training terlalu sedikit sehingga akan ditemukan banyak probabilitas null (0).

C. Saran

Penelitian selanjutnya bisa menggunakan dataset primer yang lebih signifikan kepada target dengan konteks depresi serta menambahkan fitur *tuning* pada parameter model untuk membandingkan tiap parameter mana yang paling bagus untuk dipakai.

Akurasi pada penggunaan algoritma *Naive Bayes* lebih tinggi daripada menggunakan *Support Vector Machine* karena dataset yang digunakan lebih banyak. Disamping itu, kesalahan klasifikasi terjadi pada *Support Vector Machine* akibat nilai bobot pada data uji memiliki kemiripan dengan nilai bobot data latih positif. Hal tersebut mengakibatkan data yang seharusnya diklasifikasi sebagai data negatif disalah tafsirkan oleh *Support Vector Machine*. Tingkat akurasi ini juga dapat ditingkatkan dengan menambah data latih dan juga menyeimbangkan rasio antara kelas positif dan negatif, dengan minimal rasio perbedaan 3:7.[10] Terakhir, kelompok anda juga boleh menambahkan saran terkait dengan penelitian lanjutan yang dapat dilakukan.

DAFTAR PUSTAKA

[1] https://www.who.int/news-room/fact-sheets/detail/depression retrieved on 02/12/2020

[2] Bembnowski, Marta & Jośko-Ochojska, Jadwiga. (2015). What causes depression in adults?. Polish Journal of Public Health. 125. 10.1515/pjph-2015-0037.

[3] C. Hammen. (2018). Risk Factors for Depression: An Autobiography Review. Annual Review of Clinical Psychology.

[4] https://www.dsayce.com/social-media/tweets-day/ retrieved on 04/12/2020

[5] Kamus. 2016. Pada KBBI Daring. Diambil 4 November 2020, dari https://kbbi.kemdikbud.go.id/entri/sentimen

[6] Stone, Philip J., Dexter C. Dunphy, and Marshall S. Smith. "The general inquirer: A computer approach to content analysis." MIT Press, Cambridge, MA (1966).

[7] https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html retrieved on 04/12/2020

[8] Listiowarni, I. (2019). Implementasi Naïve Bayesian dengan Laplacian Smoothing untuk Peminatan dan Lintas Minat Siswa SMAN 5 Pamekasan.

[9] KazAnova, &. &. (2017, September 13). Sentiment140 dataset with 1.6 million tweets. Retrieved from Sentiment140 dataset with 1.6 million tweets on 06/12/2020

[10] Kurniasari, P. (2015). *SISTEM PENDETEKSI GEJALA DEPRESI PENGGUNA MEDIA SOSIAL TWITTER MENGGUNAKAN METODE SUPPORT VECTOR MACHINE*. Retrieved December 6, 2020.