

Analisis Model-Model *Machine Learning Interpretable* Pada *Emotional Tweets* Berbahasa Indonesia

Joseph Setiawan Hardadi¹, Hendra Bunyamin^{1,*}

¹ Program Studi Teknik Informatika; Universitas Kristen Maranatha; Jl. Prof. drg. Surya Sumantri, M.P.H. No. 65, telp +6222-2012186/fax 022-2015154; e-mail: if1972007@student.it.maranatha.edu, hendra.bunyamin@it.maranatha.edu

* Korespondensi: e-mail: hendra.bunyamin@it.maranatha.edu

Diterima: 08 Agustus 2023; Review: 20 November 2023; Disetujui: 11 Desember 2023

Cara sitasi: Hardadi JS, Bunyamin H. 2023. Analisis Model-Model *Machine Learning Interpretable* Pada *Emotional Tweets* Berbahasa Indonesia. Bina Insani ICT Journal. Vol 10(2): 200-211

Abstrak: Penelitian ini membahas analisis model *machine learning* yang *interpretable* terhadap prediksi emosi *tweets* berbahasa Indonesia. Banyaknya pengguna *Twitter* di Indonesia membuat analisis emosi menjadi penting, karena hal ini dapat digunakan untuk memprediksi respon orang terhadap sesuatu, seperti keluhan karyawan terhadap perusahaannya dan respon siswa terhadap . Tujuan utama penelitian ini adalah menilai bagaimana algoritma-algoritma *interpretable* yang digunakan memprediksi emosi dari *tweets* tersebut. Metode yang digunakan dalam penelitian ini adalah normalisasi data, ekstraksi fitur, *training*, evaluasi dan interpretasi. Setelah normalisasi data, fitur-fitur yang diekstraksi adalah *Part-of-Speech Tagging*, *Stemming*, perhitungan jumlah kata emosi, *TF-IDF*, *Word Embedding*, nilai sentimen, dan ortografi yang terdiri dari jumlah huruf kapital, tanda seru, tanda baca, jumlah kata, dan jumlah huruf masing-masing *tweet*. Kemudian, model dilatih dan dievaluasi berdasarkan metrik F_1 score. Kemudian interpretasi dengan menggunakan *partial dependence plot* dilakukan untuk melihat efek data terhadap model. Hasil penelitian menunjukkan bahwa *Logistic Regression* memiliki F_1 score terbaik dibanding ketiga algoritma lainnya, yaitu di atas 60%. *Partial dependence plot* dari *Logistic Regression* juga dapat menggambarkan relasi kata-kata dalam *TF-IDF* terhadap model, dengan komposisi sebagian kata berdampak positif dan sebagian pula berdampak negatif atau berdampak tidak besar. Dengan demikian, dibandingkan semua algoritma, *Logistic Regression* adalah pilihan algoritma *interpretable* terbaik dalam kasus ini. Kode program dari penelitian dapat diakses di https://github.com/1972007/Kode-Jurnal_Joseph.

Kata kunci: algoritma machine learning, emosi, media sosial, interpretasi

Abstract: This research discusses the analysis of interpretable machine learning models for predicting the emotions of Indonesian language tweets. The large number of Twitter users in Indonesia makes emotional analysis important, because this can be used to predict people's responses to something, such as employee complaints about their company and students' responses to social media. The main aim of this research is to assess how the interpretable algorithms used predict emotions from these tweets. The methods used in this research are data normalization, feature extraction, training, evaluation and interpretation. After data normalization, the features extracted are Part-of-Speech Tagging, Stemming, calculation of the number of emotional words, TF-IDF, Word Embedding, sentiment value, and orthography consisting of the number of capital letters, exclamation marks, punctuation marks, number of words, and the number of letters of each tweet. Then, the model is trained and evaluated based on the F_1 score metric. Then interpretation using a partial dependence plot is carried out to see the effect of the data on the model. The research results show that Logistic Regression has the best F_1 score compared to the other three algorithms, namely above 60%. The partial dependence plot from Logistic Regression can also describe the relationship of words in TF-IDF to the model, with the composition of some words having a positive impact and some also having a negative impact or not having a big impact. Thus, compared to all algorithms, Logistic Regression is the best

interpretable algorithm choice in this case. The program code from the research can be accessed at https://github.com/1972007/Kode-Jurnal_Joseph.

Keywords: *machine learning algorithms, emotions, social media, interpretation*

1. Pendahuluan

Emosi telah menjadi bagian dari kehidupan manusia. Emosi dapat muncul dari cara bicara atau pesan seseorang. Ciri-ciri emosi yang muncul dapat digunakan untuk memprediksi keluhan konsumen, karyawan, ataupun siswa. Selain dalam kasus-kasus yang disebutkan, keluhan-keluhan atau pesan dari seseorang dapat ditemukan dalam media sosial [1].

Salah satu media sosial yang sering digunakan adalah *Twitter*. Banyak orang yang mengekspresikan emosinya dengan mengetik pesan dalam *Twitter*. Pesan tersebut umumnya disebut dengan kata *tweets*. Pemakaian *Twitter* yang sangat aktif dapat dilihat di negara Indonesia, yang menduduki urutan ketiga terbanyak dalam hal keaktifan pemakaian *Twitter* pada tahun 2012 – 2018 [1]. Fakta-fakta tersebut menjadikan *Twitter* sebagai tempat dengan kumpulan data yang berlimpah untuk kasus-kasus *Natural Language Processing* seperti prediksi emosi dari teks.

Masalah mengidentifikasi emosi yang dilakukan secara manual dapat dilakukan dengan bantuan *Machine Learning*. *Machine Learning* adalah metode-metode yang digunakan komputer untuk meningkatkan kemampuan prediksi atau perilaku berdasarkan data [2]. Penelitian ini akan berfokus terhadap algoritma *Machine Learning* yang *Interpretable*. Secara singkat, algoritma *Machine learning interpretable* adalah algoritma yang dapat membantu manusia memahami cara atau proses pembelajaran dan prediksi algoritma tersebut, hal ini untuk menghindari pemahaman algoritma sebagai kotak hitam (*black box*) [2].

Algoritma-algoritma *interpretable* yang digunakan adalah 4 algoritma *interpretable* yang sudah dibahas oleh Molnar [2]. Algoritma-algoritma tersebut digunakan karena algoritma-algoritma ini memiliki kemampuan memberikan gambaran masalah lebih luas dibandingkan algoritma lainnya. Algoritma-algoritma ini dapat dikatakan sebagai algoritma dasar *interpretable* yang digunakan sebagai studi awal. Selain itu, penggunaan *tweet* berbahasa Indonesia memerlukan penyesuaian proses pemrograman dan *library* yang digunakan pula sesuai Bahasa Indonesia.

Dengan demikian, tujuan penelitian ini adalah melihat bagaimana keempat algoritma tersebut memprediksi emosi dari *tweet* berbahasa Indonesia yang disediakan. Tujuan tersebut akan dicapai dengan melakukan pemrosesan data, ekstraksi fitur, pelatihan (*training*), evaluasi dan interpretasi. Proses-proses tersebut akan dibahas di bagian selanjutnya.

2. Metode Penelitian

Metode penelitian dalam artikel ini mencakup:

- a. Koleksi data, yaitu pengambilan data dari *twitter*, yang telah dilakukan Saputri et. al.[1],
- b. Pemrosesan data, yaitu pengolahan data dari kata menjadi angka yang dapat dibaca komputer,
- c. Proses training, yang terbagi menjadi dua bagian
 1. Persiapan pembagian data, misalnya dengan *Stratified K-Fold*,
 2. Pelatihan model pada data yang sudah diproses dengan algoritma-algoritma *interpretable*,
- d. Evaluasi model dengan menggunakan F_1 score, dan
- e. Interpretasi model yang menggunakan *partial dependence plot*.

Adapun bahasa pemrograman yang digunakan adalah *python* dengan mode *jupyter notebook*.

2.1. Koleksi

Tweet-tweet diambil dari *Twitter* dengan *Twitter Streaming API* selama dua minggu, dari 1 Juni sampai 14 Juni 2018 [1]. Terdapat 5 label emosi yang tersedia, yaitu *love*, *joy*, *surprise*, *anger*, *sadness*, dan *fear*. *Tweet-tweet* tersebut telah dipilah dan dipisahkan dari *tweet* yang tidak tergolong *personal*.

Tabel 1. Contoh Koleksi Dataset

Label	Tweet
anger	Soal jln Jatibaru,polisi tdk bs GERTAK gubernur .Emangny polisi tdk ikut pmbhasan? Jgn berpolitik. Pengaturan wilayah,hak gubernur. Persoalan Tn Abang soal turun temurun.Pelik.Perlu kesabaran. [USERNAME] [USERNAME] [URL]
anger	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.
happy	Kepingin gudeg mbarek Bu hj. Amad Foto dari google, sengaja, biar teman-teman jg membayangkannya. Berbagi itu indah.
anger	Jln Jatibaru,bagian dari wilayah Tn Abang.Pengaturan wilayah tgg jwb dan wwnang gub.Tng Abng soal rumit,sejak gub2 , trdahulu.Skrng sedng dibenahi,agr bermnfaat semua pihak.Mohon yg punya otak,berpikirlah dgn wajar,kecuali otaknya butek.Ya kamu. [URL]
happy	Sharing pengalaman aja, kemarin jam 18.00 batalin tiket di stasiun pasar senen, lancar, antrian tidak terlalu rame,15 menitan dan beress semua! Mungkin bisa dicoba twips, di jam-jam segitu cc [USERNAME]

Sumber: Hasil Penelitian (2023)

Tweet yang dihilangkan adalah *tweet* dari pemerintah, berita atau iklan. Selain itu, privasi akun yang ditulis didalam *tweet* tersebut dijaga dengan cara mengganti setiap nama akun menjadi "*username*". Perlu diingat bahwa data yang disediakan kurang seimbang, dengan jumlah data dengan label emosi *anger* 1.101 dan jumlah data berlabel *love* 637. Jumlah data berlabel *fear* adalah 649, *joy* atau *happy* 1.017, dan *sadness* 997.

2.2. Pemrosesan Data

Pemrosesan data dilakukan untuk mengubah *tweet* menjadi data yang lebih terstruktur. Pemrosesan tersebut dilakukan karena *tweet* tidak memiliki aturan yang tertulis untuk mengirim kata-kata. Selain itu, suatu *tweet* dapat memiliki kata-kata singkatan, kesalahan penulisan, huruf kapital yang tidak digunakan pada tempatnya atau tidak digunakan sama sekali. Hal-hal tersebut menyebabkan data *tweet* tidak terstruktur sehingga perlu dilakukan pemrosesan. Berikut adalah tahap pemrosesan data :

2.2.1. Normalisasi Data

Normalisasi data adalah metode penyesuaian data untuk mengurangi kemungkinan suatu data memiliki hasil duplikat. Apabila data yang digunakan adalah kata-kata atau *tweet*, kata-kata dalam *tweet* harus diubah menjadi huruf kecil. Selain itu, kata-kata tidak bermakna seperti nama panggilan, link dan kata sambung akan dihapus. Pengejaan juga perlu diperbaiki agar suatu kata yang maknanya sama tidak dianggap berbeda karena salah ketik. Proses perbaikan pengejaan dapat dibantu dengan kamus dari Saputri et. al. [1].

Tabel 1. Contoh Hasil Normalisasi

label	Tweet	Tweet_normalisasi
anger	Soal jln Jatibaru,polisi tdk bs GERTAK gubernur .Emangny polisi tdk ikut pmbhasan? Jgn berpolitik. Pengaturan wilayah,hak gubernur. Persoalan Tn Abang soal turun temurun.Pelik.Perlu kesabaran. [USERNAME] [USERNAME] [URL]	jalan jatibaru,polisi gertak gubernur .emangny polisi pmbhasan? berpolitik. pengaturan wilayah,hak gubernur. tn abang turun temurun.pelik. kesabaran. [username] [username] [url]
anger	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.	cewek lho (kayaknya), rasain sibuk jaga , rasain sakitnya haid, paniknya pulang malam . gimana orang asing? wajarlah korban takut curhat, dibela dihujat.
happy	Kepingin gudeg mbarek Bu hj. Amad Foto dari google, sengaja, biar teman-teman jg membayangkannya. Berbagi itu indah.	kepingin gudeg mbarek bu hj. amad foto google, sengaja, biar teman-teman membayangkannya. berbagi indah.
anger	Jln Jatibaru,bagian dari wilayah Tn Abang.Pengaturan wilayah tgg jwb dan wwnang gub.Tng Abng soal rumit,sejak gub2 , trdahulu.Skrng sedng dibenahi,agr bermnfaat semua pihak.Mohon yg punya otak,berpikirlah dgn wajar,kecuali otaknya butek.Ya kamu. [URL]	jalan jatibaru, wilayah tn abang.pengaturan wilayah tgg wwnang gub.tng abng soal rumit, gub , trdahulu. sedng dibenahi,agr bermnfaat .mohon otak,berpikirlah wajar,kecuali otaknya butek.ya . [url]
happy	Sharing pengalaman aja, kemarin jam 18.00 batalin tiket di stasiun pasar senen, lancar, antrian tidak terlalu rame,15 menitan dan beress semua!	sharing pengalaman , kemarin jam . batalin tiket stasiun pasar senen, lancar, antrian ramai, menitan beress! dicoba twips, jam-jam segitu cc [username]

label	Tweet	Tweet_normalisasi
	Mungkin bisa dicoba twips, di jam-jam segitu cc [USERNAME]	

Sumber: Hasil Penelitian (2023)

Normalisasi data akan dilakukan dengan *scripting* dari Python dan *library pandas*. *Pandas* memiliki fungsi untuk mengubah semua teks dalam satu kolom secara bersamaan. Hal ini memudahkan pencarian pola terhadap semua tweet dengan fungsi *regular expression*. Fungsi tersebut digunakan untuk mengubah huruf kapital menjadi huruf kecil, menghapus daftar kata tidak bermakna [3], dan mengubah kata-kata salah ketik menjadi kata yang benar.

2.2.2. Part-of-Speech Tagging

Part-of-Speech Tagging adalah proses pemasangan kata-kata dengan kategori gramatikalnya. Proses ini akan menggolongkan apakah suatu kata itu tergolong kata benda, sifat, kerja atau jenis kata lainnya.

Tabel 2. Contoh Hasil Part-of-Speech Tagging dari Tweet yang Telah Dinormalisasi

Label	Tweet	Part-of-Speech Tag
anger	jalan jatibaru.polisi gertak gubernur .emangny polisi pmbhasan? berpolitik. pengaturan wilayah,hak gubernur. tn abang turun temurun.pelik. kesabaran. [username] [username] [url]	[('jalan', 'NN'), ('jatibaru', 'JJ'), (',', 'Z'), ('polisi', 'NN'), ('gertak', 'NN'), ('gubernur', 'NN'), ('.emangny', 'FW'), ('polisi', 'NN'), ('pmbhasan', 'NN'), ('?', 'Z'), ('berpolitik', 'NN'), (',', 'Z'), ('pengaturan', 'NN'), ('wilayah', 'NN'), (',', 'Z'), ('hak', 'NN'), ('gubernur', 'NN'), (',', 'Z'), ('tn', 'NN'), ('abang', 'NN'), ('turun', 'VB'), ('temurun.pelik', 'NN'), (',', 'Z'), ('kesabaran', 'NN'), (',', 'Z'), ('[', 'Z'), ('username', 'NN'), (']', 'Z'), ('[', 'Z'), ('username', 'NN'), (']', 'Z'), ('[', 'Z'), ('url', 'NN'), (']', 'Z)']
anger	cewek lho (kayaknya), rasain sibuk jaga , rasain sakitnya haid, paniknya pulang malam . gimana orang asing? wajarlah korban takut curhat, dibela dihuajat.	[('cewek', 'NN'), ('lho', 'NN'), ('(', 'Z'), ('kayaknya', 'RB'), (')', 'Z'), (',', 'Z'), ('rasain', 'IN'), ('sibuk', 'NN'), ('jaga', 'NN'), (',', 'Z'), ('rasain', 'IN'), ('sakitnya', 'NN'), ('haid', 'NN'), (',', 'Z'), ('paniknya', 'RB'), ('pulang', 'VB'), ('malam', 'NN'), (',', 'Z'), ('gimana', 'NN'), ('orang', 'NN'), ('asing', 'JJ'), ('?', 'Z'), ('wajarlah', 'NN'), ('korban', 'NN'), ('takut', 'JJ'), ('curhat', 'VB'), (',', 'Z'), ('dibela', 'NN'), ('dihujat', 'NN'), (',', 'Z)']
happy	kepingin gudeg mbarek bu hj. amad foto google, sengaja, biar teman-teman membayangkannya. berbagi indah.	[('kepingin', 'NN'), ('gudeg', 'FW'), ('mbarek', 'FW'), ('bu', 'FW'), ('hj', 'FW'), (',', 'Z'), ('amad', 'FW'), ('foto', 'FW'), ('google', 'FW'), (',', 'Z'), ('sengaja', 'VB'), (',', 'Z'), ('biar', 'CD'), ('teman-teman', 'NN'), ('membayangkannya', 'RB'), (',', 'Z'), ('berbagi', 'RB'), ('indah', 'JJ'), (',', 'Z)']
anger	jalan jatibaru, wilayah tn abang.pengaturan wilayah tgg wwnang gub.tng abng rumit, gub , trdahulu. sedng dibenahi,agr bermnfaat .mohon otak,berpikirlah wajar,kecuali otaknya butek.ya . [url]	[('jalan', 'NN'), ('jatibaru', 'JJ'), (',', 'Z'), ('wilayah', 'NN'), ('tn', 'NN'), ('abang.pengaturan', 'NN'), ('wilayah', 'NN'), ('tgg', 'NN'), ('wwnang', 'NN'), ('gub.tng', 'NN'), ('abng', 'NN'), ('rumit', 'NN'), (',', 'Z'), ('gub', 'NN'), (',', 'Z'), ('trdahulu', 'NN'), (',', 'Z'), ('sedng', 'NN'), ('dibenahi', 'VB'), (',', 'Z'), ('agr', 'NN'), ('bermnfaat', 'NN'), ('mohon', 'NN'), ('otak', 'NN'), (',', 'Z'), ('berpikirlah', 'NN'), ('wajar', 'JJ'), (',', 'Z'), ('kecuali', 'CC'), ('otaknya', 'RB'), ('butek.ya', 'JJ'), (',', 'Z'), ('[', 'Z'), ('url', 'NN'), (']', 'Z)']
happy	sharing pengalaman , kemarin jam . batalin tiket stasiun pasar senin, lancar, antrian ramai,menitan beress ! dicoba twips, jam-jam segitu cc [username]	[('sharing', 'NN'), ('pengalaman', 'NN'), (',', 'Z'), ('kemarin', 'NN'), ('jam', 'NN'), (',', 'Z'), ('batalin', 'VB'), ('tiket', 'NN'), ('stasiun', 'NN'), ('pasar', 'NN'), ('senin', 'NNP'), (',', 'Z'), ('lancar', 'JJ'), (',', 'Z'), ('antrian', 'NN'), ('ramai', 'JJ'), (',', 'Z'), ('menitan', 'NN'), ('beress', 'FW'), (',', 'Z'), ('dicoba', 'VB'), ('twips', 'NN'), (',', 'Z'), ('jam-jam', 'FW'), ('segitu', 'FW'), ('cc', 'FW'), ('[', 'Z'), ('username', 'NN'), (']', 'Z)']

Sumber: Hasil Penelitian (2023)

Dalam *Part-of-Speech* kali ini, akan digunakan algoritma *Conditional Random Field*. *Conditional Random Field* adalah algoritma yang dapat menangkap 2 output yang mendekati output yang dipelajari dengan input asalnya. Algoritma ini akan digunakan untuk *Part-of-speech Tagging* dengan asumsi *output* berupa *tag* dan *input* adalah kata [4]. *Part-of-Speech Tagging* akan dibantu dengan daftar *Part-of-Speech Tag* dari Dinakaramani et al. [5]. Daftar tersebut akan dipelajari model *Conditional Random Field* dari *library nltk*. Model *Conditional Random Field* yang telah dihasilkan dapat digunakan untuk memberikan tag terhadap kata-kata.

2.2.3. Stemming

Stemming adalah perubahan suatu kata menjadi kata dasarnya. *Stemming* akan menyamakan kata-kata yang kata dasarnya sama, meskipun memiliki imbuhan lainnya.

Stemming dapat dilakukan dengan berbagai cara, meskipun tetap harus memperhatikan tata bahasa dari data yang digunakan [6].

Tabel 3. Contoh Tweet Hasil *Stemming*

Label	Tweet	Tweet_stem
anger	jalan jatibaru, polisi gertak gubernur. emangny polisi pmbhasan? berpolitik. pengaturan wilayah, hak gubernur. tn abang turun temurun, pelik. kesabaran. [username] [username] [url]	jalan jatibaru polisi gertak gubernur emangny polisi pmbhasan politik atur wilayah hak gubernur tn abang turun turun pelik sabar username username url
anger	cewek lho (kayaknya), rasain sibuk jaga, rasain sakitnya haid, paniknya pulang malam. gimana orang asing? wajarlah korban takut curhat, dibela dihujat.	cewek lho kayak rasain sibuk jaga rasain sakit haid panik pulang malam gimana orang asing wajar korban takut curhat bela hujat
happy	kepingin gudeg mbarek bu hj. amad foto google, sengaja, biar teman-teman membayangkannya. berbagi indah.	kepingin gudeg mbarek bu hj amad foto google sengaja biar teman teman bayang bagi indah
anger	jalan jatibaru, wilayah tn abang. pengaturan wilayah tgg wwnang gub. tng abng rumit, gub, trdahulu. sedng dibenahi, agr bermnfaat. mohon otak, berpikirlah wajar, kecuali otaknya butek. ya. [url]	jalan jatibaru wilayah tn abang atur wilayah tgg wwnang gub tng abng rumit gub trdahulu sedng benah agr bermnfaat mohon otak pikir wajar kecuali otak butek ya url
happy	sharing pengalaman, kemarin jam. batalin tiket stasiun pasar senin, lancar, antrian ramai, menitan beress! dicoba twips, jam-jam segitu cc [username]	sharing alam kemarin jam batalin tiket stasiun pasar senin lancar antri ramai menit beress coba twips jam jam segitu cc username

Sumber: Hasil Penelitian (2023)

Berikut adalah contoh dari *stemming*. Kata-kata yang masih utuh akan berubah menjadi kata-kata dasarnya. *Stemming* akan disesuaikan dengan bahasa yang digunakan teks, yaitu bahasa Indonesia. *Stemmer* yang umum untuk bahasa Indonesia adalah *sastrawi*, yang dikembangkan dari jurnal-jurnal dan modifikasi dari pengembang [7]–[9].

2.3. Ekstraksi Fitur

Ekstraksi fitur mengubah data dari teks menjadi data numerik yang dapat dibaca oleh algoritma *machine learning*. Ekstraksi fitur diperlukan karena algoritma yang digunakan hanya membaca data numerik, bukan teks. Selain fitur berupa *tweet* itu sendiri, fitur-fitur lain akan digunakan untuk menunjang pelatihan algoritma. Proses ekstraksi fitur didasari pada penelitian dari Saputri et al. [1]. Berikut adalah fitur-fitur yang digunakan.

2.3.1. Daftar Kata-kata Emosi Bahasa Indonesia

Daftar Kata-kata Emosi Bahasa Indonesia: daftar ini berisi jumlah kata-kata yang melambangkan masing-masing emosi untuk setiap *tweet*. Daftar kata yang digunakan adalah daftar kata emosi yang terdaftar dalam jurnal yang ditulis Shaver et. al. [10]. Secara konsep, satu teks akan diwakilkan oleh lima fitur atau kolom, yang diberi nama sesuai label yang digunakan. Kata-kata dalam suatu teks akan dipisahkan dengan *library*. Kata-kata tersebut akan dicari didalam daftar kata emosi. Apabila kata tersebut tergolong dalam suatu emosi, nilai emosi tersebut akan ditambah satu. Hal ini akan terus berlanjut sampai kata terakhir dalam suatu teks. Proses yang sama akan dimulai dari awal untuk teks yang berbeda. Daftar kata tersebut akan diimplementasikan dengan *scripting* dalam bahasa *python*, dibantu oleh *library pandas* dan kata-kata yang telah dipisahkan oleh *nlk*.

2.3.2. Bag-of-Words

Bag-of-words adalah vektor kata-kata yang berisi jumlah kata tersebut dalam suatu teks. Jumlah kata dapat memberi gambaran terhadap topik yang diangkat [11] Sebagai contoh, banyaknya kata makanan seperti apel, martabak, udang, coklat, dan lain sebagainya mengindikasikan bahwa topik tersebut adalah tentang makanan., sedangkan angka sebagai barisnya. Konsep Bag-of-Words mudah dipahami, tetapi konsep ini memiliki masalah dimensi yang sesuai dengan banyak kata yang ada dalam dataset. Dimensi *bag-of-words* akan menjadi membengkak ketika fitur ini mewakili semua kata dalam dataset. Perlu diingat pula, bahwa kata yang muncul dalam satu teks tidak selalu muncul dalam teks lainnya. Dalam penelitian ini, *bag-of-words* digunakan untuk menunjang perhitungan *TF-IDF* di bagian selanjutnya.

2.3.3 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency adalah perhitungan untuk menilai penting tidaknya suatu kata. *TF-IDF* memiliki kelebihan dibandingkan *Bag-of-Words* dalam hal menilai kata-kata yang lebih penting dengan nilai yang lebih tinggi. Kata-kata yang kurang penting seperti kata sambung akan dinilai rendah oleh *TF-IDF*, sedangkan kata-kata yang memiliki peran penting dalam dokumen umumnya dinilai tinggi [12]. Akan tetapi, *TF-IDF* tetap memiliki masalah dimensi seperti *Bag-of-Words*. *Bag-of-Words* dan *TF-IDF* dalam penelitian ini akan diproses dengan memanfaatkan library *gensim* [13]. Meskipun *gensim* dapat memproses kedua fitur tersebut, fitur *TF-IDF* akan lebih difokuskan daripada *Bag-of-Words* karena penilaian *TF-IDF* yang lebih lengkap. Dalam penelitian ini, *TF-IDF* akan berbentuk tabel dengan kata-kata yang ada dalam semua *tweet* sebagai kolomnya

2.3.4. Word Embeddings

Word Embeddings adalah representasi kata dalam bentuk vektor angka yang padat. *Word Embeddings* memiliki jumlah dimensi di antara 50-1.000; jumlah yang cukup kecil untuk representasi kata, tidak seperti *TF-IDF* dan *Bag-of-Words*. Jumlah dimensi *Word Embeddings* yang lebih kecil dibandingkan *TF-IDF* dan *Bag-of-Words* menjadi nilai tambah tersendiri untuk fitur ini. Hanya saja, *Word Embeddings* merepresentasikan satu kata menjadi vektor, sehingga untuk suatu teks atau kalimat, perlu melakukan perubahan seperti nilai vektor semua teks yang harus dirata-ratakan. Salah satu implementasi *Word Embeddings* adalah *word2vec* [6]. *Word Embeddings* yang akan digunakan adalah *word2vec*, dengan library *gensim* seperti fitur sebelumnya. Model *word2vec* yang digunakan dengan data teks training mencapai 1.026.484 *tweet* [1].

2.3.5. Kamus Sentimen

Kamus Sentimen adalah fitur yang berisi kata-kata yang dikelompokkan berdasarkan sentimennya. Kelompok sentimen umumnya dibagi menjadi sentimen positif dan negatif. Kamus sentiment yang digunakan kali ini juga memberikan nilai sebagai ukuran untuk tingkat sentimen suatu kata. Kamus Sentimen yang digunakan adalah InSet Lexicon. Kelebihan leksikon tersebut adalah pengelompokan kata berdasarkan sentimen serta nilai kata tersebut, dimulai dari -5 untuk nilai sentimen negatif sampai +5 untuk nilai sentimen positif [14]. Dalam penelitian ini, bentuk dari fitur ini adalah jumlah nilai positif dan jumlah nilai negatif per *tweet*. Nilai tersebut diambil dari kata per kata *tweet* yang dicocokkan dengan kamus sentiment InSet Lexicon.

2.3.6. Ortografi

Ortografi adalah daftar aturan tentang penggunaan huruf kapital, tanda baca, kata sambung, dan ejaan suatu kata. Ortografi tertentu dapat menangkap emosi tergantung penggunaan keempat hal tersebut. Salah satu contohnya adalah penggunaan tanda seru untuk emosi *anger*. Ortografi yang difokuskan adalah penghitungan kapitalisasi, tanda baca, tanda seru, total huruf, dan total karakter setiap *tweet*, sesuai penyedia dataset [1]. Sebagai contoh singkat, *tweet* "Kepingin gudeg mbarek Bu hj. Amad Foto dari google, sengaja, biar teman-teman jg membayangkannya. Berbagi itu indah." akan dicatat sebagai 94 total huruf, 0 tanda seru, 22 tanda baca, 18 kata, dan 0 kata dengan huruf kapital. Proses tersebut akan dilakukan dengan pemrograman *python* dan library *pandas* untuk fungsi akses teks.

2.4. Training

Training dilakukan untuk mengenalkan algoritma kepada data. Data tersebut harus sudah diproses dengan tahap-tahap sebelumnya agar proses *training* berjalan dengan lancar. Algoritma dalam pelatihan ini pun harus memiliki kemampuan untuk melakukan klasifikasi banyak fitur. Syarat tersebut perlu diperhatikan karena tidak sedikit algoritma yang berfokus terhadap regresi dan klasifikasi dua label atau biner.

2.4.1. Pembagian data

Data akan dibagi menjadi data *training* dan *testing*. Akan tetapi, pembagian data secara langsung bergantung pada hasil data yang akan dibagi. Apabila tidak diperhatikan bagian data *training* dan *testing* akan memiliki proporsi yang tidak seimbang atau bahkan kehilangan fitur. Algoritma *K-Fold* digunakan untuk membagi data menjadi bagian-bagian yang beragam, dengan

nilai K yang umum digunakan adalah 5 atau 10, bergantung dari jumlah data yang diproses. Pembagian K -Fold memudahkan evaluasi model karena akan ada beragam nilai untuk beragam pemotongan data. Hal ini membantu melihat performa model terlepas dari pembagian modelnya [15]. Meskipun begitu, proporsi data tetap perlu diperhatikan. Maka dari itu, algoritma *Stratified K-Fold*, yang dapat dikatakan sebagai ekstensi dari K -Fold, digunakan untuk membagi data dengan proporsi yang seimbang [16].

2.4.2. Algoritma

Algoritma yang digunakan harus dianggap *interpretable* dan dapat melakukan prediksi banyak kelas, sesuai lingkup penelitian. Prediksi banyak kelas diperlukan karena ada 5 kelas atau label emosi yang harus diprediksi. *Library* utama *training* dalam penelitian ini adalah *scikit learn*. *Library* tersebut dapat membantu pembagian data, implementasi algoritma, evaluasi dan interpretasi. Algoritma *Logistic Regression*, *Decision Tree*, *Naive Bayes*, dan *K-Nearest Neighbor* juga sudah disediakan oleh *scikit learn* [17]. Algoritma-algoritma tersebut digunakan mengingat dalam *python* dan *library scikit-learn*, algoritma tersebut lah yang dapat melakukan prediksi banyak kelas dan tergolong *interpretable* menurut Molnar [2]. Berikut algoritma yang digunakan :

2.4.2.1. Logistic Regression

Logistic Regression adalah algoritma yang memodelkan masalah menjadi hasil-hasil prediksi. Hasil *Logistic Regression* adalah probabilitas suatu data *input* adalah bagian dari kelas tertentu. Dengan demikian, data yang diprediksi memiliki kelas dan juga probabilitas data tersebut benar sesuai dengan kelas hasil prediksinya. *Logistic Regression* umum digunakan untuk klasifikasi biner dan multi label dalam *library* atau bahasa pemrograman tertentu [2].

2.4.2.2. Decision Trees

Decision Trees memodelkan data *training* menjadi banyak cabang yang terus bercabang. *Decision Trees* juga menentukan akhir dari cabang tersebut, yang disebut dengan *leaf*. Cabang-cabang tersebut berisi kondisi-kondisi yang dipelajari dari data *training* yang mengelompokkan data *input* kepada suatu kelas. *Decision Tree* umum digunakan untuk klasifikasi, baik klasifikasi berlabel banyak maupun berjumlah label biner [2].

2.4.2.3. Naïve Bayes

Naïve Bayes adalah algoritma klasifikasi yang menggunakan teori *Bayes* tentang probabilitas bersyarat. Setiap fitur yang ada akan dianggap independen oleh *Naïve Bayes*, meskipun pada nyatanya tidak independen. Suatu data akan dihitung berdasarkan peluang fitur data tersebut berada di salah satu kelas. Hasil peluang semua fitur akan dikalikan dan dinormalisasikan untuk menjadi hasil prediksi [18].

2.4.2.4. K-Nearest Neighbor

K-Nearest Neighbor adalah algoritma yang memetakan data *training* dan memprediksi data *input* berdasarkan titik terdekat data *input* terhadap data *training*. Nilai K adalah jumlah data terdekat yang akan digunakan untuk menentukan kelas suatu data. *K-Nearest Neighbor* tidak melibatkan penilaian fitur yang rumit, melainkan memetakan data dan melihat data-data yang merupakan tetangga dari data *input* [18].

2.5.3. Evaluasi

Algoritma yang telah dilatih akan dievaluasi dengan nilai F_1 . Nilai F_1 menggabungkan kedua nilai *precision* dan *recall*. Kedua nilai evaluasi tersebut memperhatikan prediksi algoritma yang sesuai dengan kelasnya, atau disebut dengan *true positives*. Keunggulan tersebut membuat F_1 umum digunakan untuk data yang tidak rata proporsinya antar kelas [18]. Proporsi data yang tidak rata telah terlihat sebelumnya, dengan dataset yang digunakan memiliki jumlah data dengan label emosi *anger* 1.101, jumlah data berlabel *love* 637, jumlah data berlabel *fear* adalah 649, *joy* atau *happy* 1.017, dan *sadness* 997. Fungsi F_1 tersebut telah disediakan oleh *scikit learn* [17].

2.5.4. Interpretasi

Dalam penelitian ini, tahap interpretasi adalah tahap untuk mencoba melihat penyebab model memprediksi sesuatu. Interpretasi model dalam penelitian ini dilakukan dengan menggunakan *partial dependence plot*. Plot tersebut tidak memandang model yang digunakan,

tetapi lebih memperhatikan efek fitur terhadap model. Selain itu, plot tersebut menggambarkan hubungan model dan fitur secara keseluruhan [2]. Dengan demikian, *partial dependence plot* dapat memudahkan interpretasi fitur terhadap prediksi model. *Library partial dependence plot* juga telah disediakan oleh *scikit learn*. Perlu diperhatikan bahwa, *partial dependence plot* akan melakukan *plotting* terhadap semua label, sehingga akan ada banyak plot sebanyak jumlah label yang ada [17]. Meskipun begitu, *plot* yang ditampilkan adalah *plot* dari algoritma yang memiliki performa terbaik untuk memudahkan interpretasi.

3. Hasil dan Pembahasan

Berikut hasil pengujian dari metode yang telah dijabarkan. Pelatihan dan pengujian dilakukan dengan algoritma *logistic regression*, *decision tree*, *Naïve Bayes*, dan *K-Nearest Neighbor* (dengan $n_neighbors = 15$). Data yang telah diolah telah terbagi berdasarkan jenis fitur yang ingin diekstraksi. Data *training* dan *testing* dipisahkan menggunakan metode *Stratified K-Fold* dengan K bernilai 5. Kolom-kolom berikut disesuaikan dengan fitur-fitur yang dibahas pada bagian sebelumnya.

Keterangan dari setiap tabel kolom, yang berupa fitur, adalah :

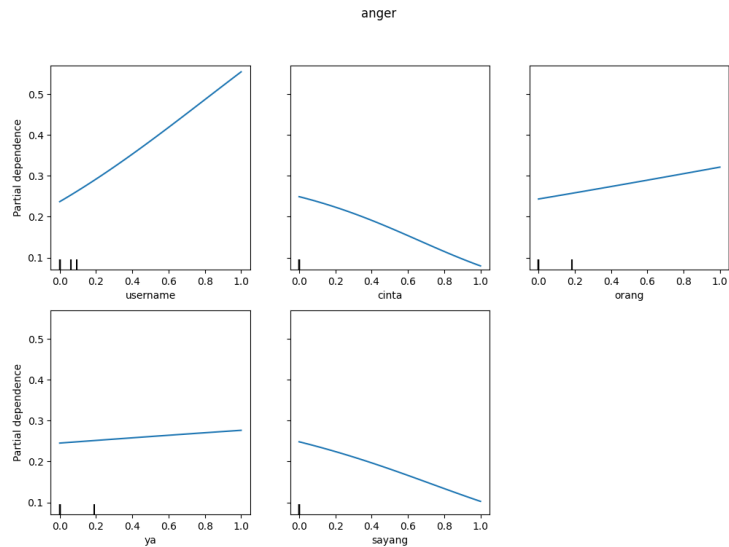
- Emotion Words* adalah fitur hitungan kata-kata emosi yang telah dihitung berdasarkan definisi kata emosi berikut [1].
- TF-IDF* adalah fitur *TF-IDF* yang telah disiapkan berdasarkan *tweet*.
- word2vec* adalah fitur *Word Embeddings* yang berasal dari kalimat yang dinilai oleh model yang telah disiapkan [1]
- Inset* adalah fitur nilai sentimen yang didasari oleh kamus sentimen *InSet* [14].
- Orth* adalah fitur jumlah ortografi yang dihitung dari data *tweet*
- POS* adalah fitur *POS Tagging* yang disiapkan berdasarkan data *tweet*, daftar *POS Tag* dari Dinakaramani et al. [5] dan *CRF Tagger*.
- All* adalah fitur gabungan dari semua fitur sebelumnya.
- Penilaian akan difokuskan terhadap nilai F_1 yang memiliki fungsi rata-rata *weighted*, untuk memastikan bahwa perhitungan nilai F_1 memperhatikan jumlah data per label.

Tabel 5. Hasil Nilai F1 dari Uji Algoritma

Fitur/Algoritma	<i>Logistic Regression</i>	<i>Decision Tree</i>	<i>Naïve Bayes</i>	<i>K-Nearest-Neighbor</i>
<i>Emotion Words</i>	44.07%	48.98%	40.23%	42.92%
<i>TF-IDF</i>	61.07%	55.31%	51.62%	25.54%
<i>Word2vec</i>	59.77%	35.73%	51.77%	54.76%
<i>InSet</i>	36.29%	32.31%	40.01%	29.26%
<i>Orth</i>	37.31%	30.63%	40.02%	26.73%
<i>Pos</i>	34.59%	32.26%	34.80%	27.60%
<i>All</i>	64.09%	52.06%	55.22%	57.09%

Sumber: Hasil Penelitian (2023)

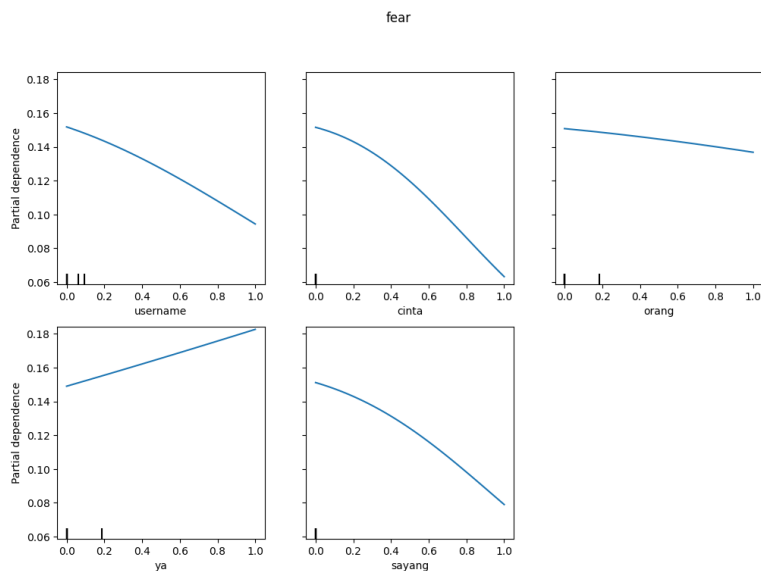
Dapat dilihat dalam tabel ini bahwa algoritma *Logistic Regression* memiliki nilai terbaik di angka 61.07% dengan fitur *TF-IDF* dan 64.09% dengan semua fitur yang digabungkan. Fitur *Word2Vec* menyusul dengan nilai F_1 paling besar 59.77% dengan algoritma *Logistic Regression* juga. Algoritma lain memiliki nilai yang lebih bervariasi dibandingkan *Logistic Regression*. Pengujian ini menunjukkan bahwa algoritma *Logistic Regression* mampu menangkap nilai-nilai setiap kata dalam *TF-IDF* untuk memprediksi emosi suatu teks. Dengan hasil tersebut, algoritma *Logistic Regression* dan fitur *TF-IDF* akan dijadikan patokan untuk interpretasi dengan *Partial Dependency Plot*.



Sumber: Hasil Penelitian (2023)

Gambar 1. Hasil *Partial Dependence Plot Logistic Regression* dan fitur *TF-IDF* terhadap label *Anger*.

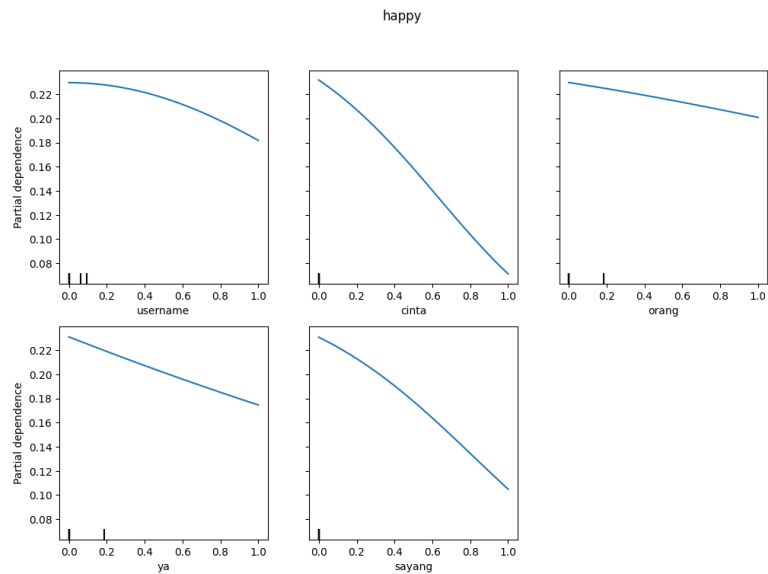
Plot pada Gambar 1 menggambarkan dampak suatu fitur terhadap kemampuan prediksi *logistic regression* dalam memprediksi suatu emosi. Karena fitur yang difokuskan adalah *TF-IDF*, fitur yang digunakan adalah kata-kata yang ada dalam *TF-IDF*. 5 kata dengan rata-rata *TF-IDF* tertinggi, yang artinya kata-kata tersebut penting dalam satu dataset tersebut, diambil untuk memudahkan penelitian. Dari gambar tersebut, dapat disimpulkan bahwa *Logistic Regression* menilai kata *username* sebagai kata yang berdampak positif untuk label *anger*, yang artinya kata tersebut, apabila jumlahnya banyak dalam satu kalimat, seringkali berada di *tweet* seseorang yang sedang marah, seperti pengguna *twitter* yang melakukan *tag* berkali-kali dalam suatu *tweet* karena kesal akan suatu hal, seperti “sSoal jln Jatibaru, polisi tdk bs GERTAK gubernur .Emangny polisi tdk ikut pmbhasan? Jgn berpolutik. Pengaturan wilayah, hak gubernur. Persoalan Tn Abang soal turun temurun. Pelik. Perlu kesabaran. [USERNAME] [USERNAME] [URL]”.



Sumber: Hasil Penelitian (2023)

Gambar 2. Hasil *Partial Dependence Plot Logistic Regression* dan fitur *TF-IDF* terhadap label *Fear*.

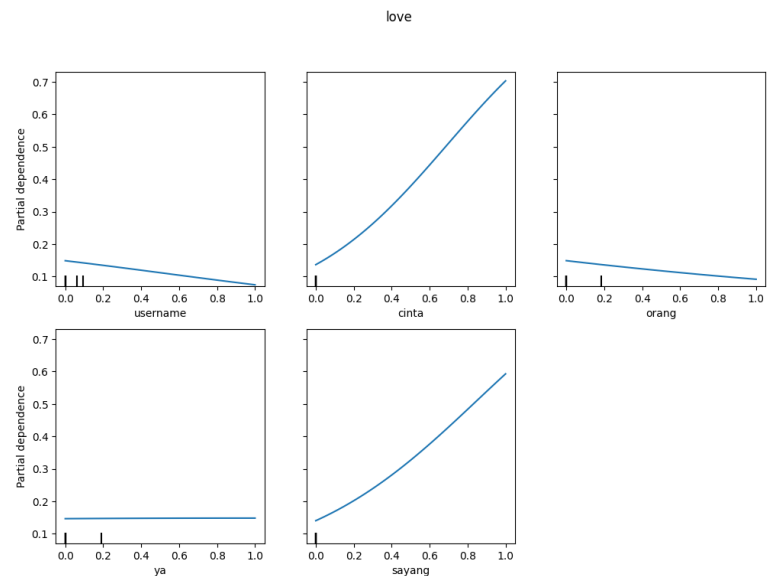
Logistic Regression juga menilai kata *ya* sebagai kata yang berdampak positif terhadap label *fear*. Dapat disimpulkan *Logistic Regression* melihat bahwa kata tersebut sering muncul di *tweet* berlabel *fear*.



Sumber: Hasil Penelitian (2023)

Gambar 3. Hasil *Partial Dependence Plot Logistic Regression* dan fitur *TF-IDF* terhadap label *Happy*.

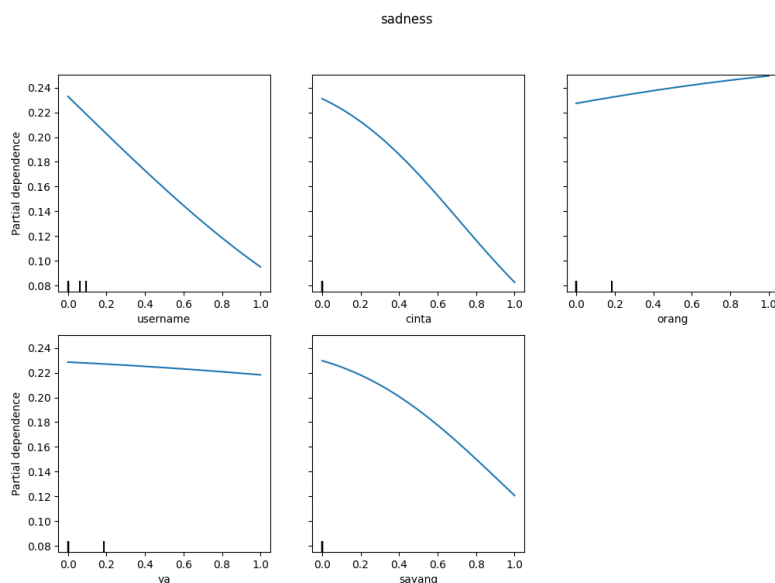
Kata-kata yang telah dipilih sebelumnya cenderung berdampak negatif terhadap label *happy*. Kata-kata tersebut bisa saja jarang ditemukan di *tweet* berlabel *happy*, sehingga *Logistic Regression* menyimpulkan demikian.



Sumber: Hasil Penelitian (2023)

Gambar 4. Hasil *Partial Dependence Plot Logistic Regression* dan fitur *TF-IDF* terhadap label *Love*.

Berdasarkan gambar diatas, kata-kata seperti “sayang” dan “cinta” berdampak positif terhadap *love*.



Sumber: Hasil Penelitian (2023)

Gambar 5. Hasil *Partial Dependence Plot Logistic Regression* dan fitur *TF-IDF* terhadap label *Sadness*.

1 Kata yang telah dipilih sebelumnya dan berdampak positif terhadap label *sadness* adalah kata “orang”. *Logistic Regression* terlihat jarang melihat keempat kata lainnya dalam *tweet* berlabel *sadness*.

Berikut adalah hasil *partial dependence plot* yang didapat. Plot dikelompokkan berdasarkan label dan kelas yang dilihat. Menurut plot-plot yang telah dibuat, sebagian kata berdampak positif terhadap suatu kelas dan sebagian kata tidak berdampak banyak atau berdampak negatif. Hal itu terlihat pada penggunaan kata “username” yang berdampak positif terhadap kelas *anger*. Kata “username” sendiri ditujukan sebagai ganti kata dari *username* dalam *twitter*, seperti pada saat teks tersebut menyebut nama akun orang lain. Dapat diduga, bahwa penggunaan kata *username* dalam kasus ini berupa penyebutan nama orang secara berlebihan di saat penulis *tweet* merasa sangat marah. Sebaliknya, kata-kata yang diwakili oleh kata “username” berdampak negatif terhadap label *sadness*, seolah-olah kata tersebut mengarah kepada emosi lain selain *sadness*.

4. Kesimpulan

Hasil penelitian ini menunjukkan bahwa *Logistic Regression* menjadi algoritma dengan performa terbaik dibandingkan algoritma lain yang diuji. *TF-IDF* juga menjadi fitur yang memberikan hasil yang baik, selain fitur gabungan dan *Word embeddings*. Walaupun demikian, tidak ada algoritma yang memiliki nilai evaluasi F_1 melebihi 70% sehingga penggunaan algoritma *interpretable* terhadap kasus prediksi emosi dari *tweet* kurang meyakinkan. Meskipun begitu, dapat disimpulkan bahwa fitur seperti *TF-IDF* dan *Word Embedding* dapat mewakili suatu teks secara lengkap dan dapat digunakan untuk prediksi emosi dengan teks selanjutnya. *Logistic Regression* dapat pula membaca kata-kata yang sesuai dengan emosi yang ingin diprediksi, sehingga algoritma tersebut dapat menjadi langkah pertama untuk deteksi emosi. Diharapkan ke depannya algoritma *interpretable* lainnya dapat digunakan untuk klasifikasi dan melampaui kemampuan prediksi keempat algoritma yang diuji.

Penelitian ini juga dapat memberikan masukan ke industri, yaitu fitur *TF-IDF* dapat difokuskan untuk memprediksi *tweet* bermuatan emosi. Algoritma-algoritma lain juga dapat digunakan sebagai pembanding terhadap algoritma sebelumnya.

Ucapan Terima Kasih

Terima kasih kepada Universitas Kristen Maranatha atas dukungannya untuk penyelesaian penelitian ini.

Referensi

- [1] M. S. Saputri, R. Mahendra, dan M. Adriani, "Emotion Classification on Indonesian Twitter Dataset," dalam *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, Institute of Electrical and Electronics Engineers Inc., Jan 2019, hlm. 90–95. doi: 10.1109/IALP.2018.8629262.
- [2] C. Molnar, *Interpretable Machine Learning*, 2 ed. 2022. [Daring]. Tersedia pada: <https://christophm.github.io/interpretable-ml-book>
- [3] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," University of Amsterdam, 2003
- [4] U. Kamath, J. Liu, dan J. Whitaker, *Deep learning for NLP and speech recognition*. Springer International Publishing, 2019. doi: 10.1007/978-3-030-14596-5.
- [5] A. Dinakaramani, F. Rashel, A. Luthfi, dan R. Manurung, "Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus," dalam *2014 International Conference on Asian Language Processing (IALP)*, 2014, hlm. 66–69. doi: 10.1109/IALP.2014.6973519.
- [6] D. Jurafsky dan J. H. Martin, *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Second Edition*. New York: Pearson, 2014.
- [7] A. Z. Arifin, P. Adhi, K. Mahendra, dan H. T. Ciptaningtyas, "ENHANCED CONFIX STRIPPING STEMMER AND ANTS ALGORITHM FOR CLASSIFYING NEWS DOCUMENT IN INDONESIAN LANGUAGE," 2009, [Daring]. Tersedia pada: <https://www.researchgate.net/publication/228416000>
- [8] J. Asian, "Effective Techniques for Indonesian Text Retrieval," RMIT University, 2007
- [9] A. D. Tahitoe dan D. Purwitasari, "Implementasi modifikasi enhanced confix stripping stemmer untuk bahasa indonesia dengan metode corpus based stemming".
- [10] P. R. Shaver, U. Murdaya, dan R. C. Fraley, "Structure of the Indonesian emotion lexicon."
- [11] J. Eisenstein, "Natural Language Processing," 2018.
- [12] G. Bonaccorso, "Machine Learning Algorithms Popular algorithms for data science and machine learning".
- [13] R. Řehůřek dan P. Sojka, "Software Framework for Topic Modelling with Large Corpora," Nov 2010, hlm. 45–50. doi: 10.13140/2.1.2393.1847.
- [14] F. Koto dan G. Y. Rahmaningtyas, "Inset lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs," dalam *Proceedings of the 2017 International Conference on Asian Language Processing, IALP 2017*, Institute of Electrical and Electronics Engineers Inc., Feb 2018, hlm. 391–394. doi: 10.1109/IALP.2017.8300625.
- [15] S. Raschka dan V. Mirjalili, *Python machine learning : machine learning and deep learning with python, scikit-learn, and tensorflow 2*.
- [16] S. Widodo, H. Brawijaya, dan S. Samudi, "Stratified K-fold cross validation optimization on machine learning for prediction," *Sinkron*, vol. 7, no. 4, hlm. 2407–2414, Okt 2022, doi: 10.33395/sinkron.v7i4.11792.
- [17] F. Pedregosa *dkk.*, "Scikit-learn: Machine Learning in Python," Jan 2012, [Daring]. Tersedia pada: <http://arxiv.org/abs/1201.0490>
- [18] S. Masís, *Interpretable Machine Learning with Python*. 2021.