



Klasifikasi Argument Pada Teks dengan Menggunakan Metode Multinomial Logistic Regression Terhadap Kasus Pemindahan Ibu Kota Indonesia di Twitter

Mochammad Naufal Rizaldi*, Adiwijaya, Said Al Faraby

Fakultas Informatika, Universitas Telkom, Bandung 40257, Indonesia

Email: ^{1,*}naufalrzd@student.telkomuniversity.ac.id, ²adiwijaya@telkomuniversity.ac.id,

³saidalfaraby@telkomuniversity.ac.id

Email Penulis Korespondensi: naufalrzd@student.telkomuniversity.ac.id

Abstrak—Informasi pemindahan Ibu Kota Indonesia dari Jakarta ke Kalimantan Timur tentunya menimbulkan pro dan kontra yang disampaikan oleh masyarakat Indonesia melalui jejaring sosial twitter. Akan tetapi, komentar pro dan kontra tersebut tentunya beragam, disertai maupun tidak disertai dengan argumen atau bahkan tidak berhubungan sama sekali dengan topik yang sedang dibahas. Keterbatasan pengguna dalam menyaring informasi tersebut tentunya akan mempersulit masyarakat atau bahkan pemerintah dalam menganalisis informasi yang terkandung dalam tweet. Oleh karena itu, dibangun sebuah sistem yang dapat melakukan klasifikasi terhadap tweet secara otomatis ke dalam tiga kelas, yaitu non argumen, argumen dan unknown. Metode yang digunakan dalam penelitian ini yaitu Multinomial Logistic Regression (MLR). MLR merupakan metode generalisasi dari Logistic Regression dan digunakan untuk melakukan klasifikasi 3 kelas atau lebih. Sebelum dilakukan proses klasifikasi, tweet harus dilakukan preprocessing terlebih dahulu agar tweet bersih dari segala noise yang ada. Feature extraction yang digunakan dalam penelitian ini yaitu diantaranya unigram, bigram dan trigram. Dalam penelitian ini, terdapat 12 skenario pengujian dan metode pembandingan yaitu Artificial Neural Network (ANN). Dari semua skenario pengujian hasil yang paling baik untuk metode MLR yaitu dengan skenario SRU dengan akurasi sebesar 41,30%, sedangkan untuk metode ANN yaitu dengan skenario RU dengan akurasi sebesar 45,10%.

Kata Kunci: Multinomial Logistic Regression, Artificial Neural Network, Backpropagation

Abstract—Information on moving the Indonesian capital from Jakarta to East Kalimantan certainly raises the pros and cons conveyed by the Indonesian people through the Twitter social network. However, the pros and cons comments are of course varied, accompanied or not accompanied by arguments or even completely unrelated to the topic under discussion. User limitations in filtering out that information will certainly make it difficult for the public or even the government to analyze the information contained in the tweet. Therefore, a system was built that could classify tweets automatically into three classes, namely non-arbitration, argument and unknown. The method used in this research is Multinomial Logistic Regression (MLR). MLR is a generalization method of Logistic Regression and is used to classify 3 or more classes. Before the classification process is carried out, the tweet must be preprocessed in order to make the tweet clear of all existing noise. Feature extractions used in this study include unigram, bigram and trigram. In this study, there are 12 test scenarios and comparison methods, namely Artificial Neural Network (ANN). Of all the test scenarios the best results for the MLR method are SRU with an accuracy of 41,30%, while for the ANN method namely the RU scenario with an accuracy of 45,10%.

Keywords: Multinomial Logistic Regression, Artificial Neural Network, Backpropagation

1. PENDAHULUAN

Belakangan ini Indonesia sedang diramaikan dengan informasi pemindahan Ibu Kota dari DKI Jakarta ke Kalimantan Timur. Tentunya hal ini menimbulkan pernyataan pro dan kontra dari masyarakat terhadap informasi tersebut. Penyampaian pro dan kontra tersebut banyak disampaikan melalui media sosial, salah satunya jejaring sosial *twitter*. Namun tidak semua pernyataan tersebut dilengkapi dengan argument atau bahkan menyatakan hal diluar konteks pemindahan Ibu Kota Indonesia.

Keterbatasan pengguna dalam menyaring informasi tersebut tentunya akan mempersulit dalam pengambilan keputusan bagi pemerintah. Oleh karena itu, perlu adanya penyaringan informasi apakah mengandung argument atau tidak, atau bahkan diluar konteks pembahasan. Berdasarkan permasalahan tersebut, penulis melakukan klasifikasi argument pada teks berbahasa Indonesia dari *twitter* dengan topik Pemindahan Ibu Kota Indonesia dengan menggunakan metode *Multinomial Logistic Regression* (MLR), kemudian hasilnya akan dilakukan perbandingan dengan menggunakan metode *Artificial Neural Network* (ANN).

Multinomial Logistic Regression (MLR) merupakan metode generalisasi dari *sigmoid function* yang dinamakan dengan *softmax function*. *Sigmoid function* digunakan dalam *Logistic Regression* yang mana merupakan metode baseline dalam melakukan klasifikasi text dalam *natural language processing* [1]. *Multinomial Logistic Regression* digunakan untuk klasifikasi lebih dari 2 kelas, oleh karena itu dalam penelitian ini menggunakan metode *Multinomial Logistic Regression*. Selain itu, *Multinomial Logistic Regression* memiliki persamaan dengan metode neural network, yang mana dalam melakukan proses klasifikasi model harus melakukan learning terlebih dahulu untuk mendapatkan hasil yang maksimal.

Penelitian yang dilakukan oleh membahas [2] tentang penerapan *argument mining* dengan menggunakan metode klasifikasi *Logistic Regression* dan *Random Forest*. Pada penelitian [2] terdapat beberapa kategori ekstraksi fitur, diantaranya:

a. *Lexical (L)*: unigram, bigram dan *WordNet verb synsets*;



- b. *Twitter-specific (T): punctuation, emoticons;*
- c. *Syntactic/Semantic (S): POS Tagging*
- d. *Sentiment (SE)*

Terdapat tiga pendekatan pengujian untuk melakukan *argument mining*. Pertama, pengujian dilakukan dengan menggunakan metode klasifikasi *Random Forest* dan fitur ekstraksi yang digunakan yaitu *Lexical*. Pengujian kedua dilakukan dengan menggunakan fitur level *Lexical* dan dimasukkan ke dalam model klasifikasi *Logistic Regression*. Pengujian terakhir dengan menggunakan metode klasifikasi *Logistic Regression* dan menggunakan seluruh fitur ekstraksi yang ada. Dari ketiga pendekatan pengujian tersebut didapatkan bahwa metode klasifikasi *Logistic Regression* dan semua fitur ekstraksi memiliki nilai F1 Score paling baik yaitu sebesar 78%.

2. METODOLOGI PENELITIAN

2.1 Preprocessing

Preprocessing merupakan tahap awal dalam pengolahan data dengan tujuan agar data lebih mudah untuk digunakan dan dibaca setelah masuk ke dalam proses *mining* [3]. Terdapat beberapa teknik yang umum digunakan pada proses *preprocessing*, diantaranya *Stemming*, *Stopwords*, dan *Tokenization*, dan *POSTaging* [4]. Tahap *preprocessing* yang dilakukan pada penelitian ini yaitu sebagai berikut:

1. *Casefolding*, merupakan proses untuk merubah semua huruf yang terdapat pada setiap dokumen menjadi huruf kecil (*lowercase*).
2. *Remove URL*, merupakan proses untuk menghapus URL yang terdapat pada dokumen.
3. *Tokenization*, merupakan proses untuk melakukan pemisahan kata demi kata dari suatu kalimat.
4. *Remove Username and Hashtag*, merupakan proses untuk menghapus username dan hastag yang terdapat pada setiap dokumen.
5. *Remove Punctuation*, merupakan proses untuk menghilangkan tanda baca yang terdapat pada setiap dokumen.
6. *Remove Emoticon*, merupakan proses untuk menghapus *emoticon* yang terdapat pada setiap dokumen.
7. *Remove Digit/Number*, merupakan proses untuk menghapus digit atau angka yang terdapat pada setiap dokumen.
8. *Remove Duplicate*, merupakan proses untuk menghapus salah satu dokumen yang ganda.
9. *Replace Abbreviation Word*, merupakan proses untuk me-*replace*/menimpa kata-kata singkatan dalam bahasa Indonesia.
10. *Stemming*, merupakan proses untuk mengubah kata yang mengandung imbuhan menjadi kata dasar.

2.2 TF-IDF

Term Frequency-Invers Document Frequency (TF-IDF) merupakan sebuah teknik pembobotan yang paling sering digunakan [5]. Teknik ini digunakan untuk menghilangkan istilah yang paling umum dan hanya mengekstrak istilah yang paling relevan dari corpus [6]. Dalam TF-IDF, TF (*Term Frequency*) menyatakan jumlah kata yang sering muncul dalam suatu dokumen, sedangkan IDF (*Invers Document Frequency*) untuk menentukan apakah kata tersebut merupakan kata umum atau bukan. Jika kata tersebut muncul di banyak dokumen, maka kata tersebut merupakan kata umum dan nilai IDF akan semakin kecil. Pada umumnya TF-IDF dirumuskan sebagai berikut:

$$TFIDF_a = TF_a \times \log \left(\frac{N}{DF_a} \right) \quad (1)$$

Term frequency (TF) adalah jumlah kemunculan kata dalam sebuah dokumen. N adalah jumlah total dokumen/dataset. DF adalah jumlah dokumen yang mengandung fitur terkait dan *a* adalah atribut yang akan dilakukan pembobotan. Hasil dari proses TFIDF ini yaitu kumpulan kata unik yang sudah memiliki nilai bobot [7].

2.3 N-Gram

N-Gram merupakan sebuah *language model* dalam *Natural Language Processing* yang terbentuk dari suatu urutan N kata [8]. N-Gram dapat dicontohkan dalam kalimat "pindah ibu kota" sebagai berikut:

- a. 1-Gram (*Unigram*), merupakan urutan kata yang terdiri dari 1 kata, maka N-Gram yang akan terbentuk dari contoh kalimat tersebut yaitu "pindah" "ibu" "kota"
- b. 2-Gram (*Bigram*), merupakan urutan kata yang terdiri dari 2 kata, maka N-Gram yang akan terbentuk dari contoh kalimat tersebut yaitu "pindah ibu" "ibu kota"
- c. 3-Gram (*Trigram*), merupakan urutan kata yang terdiri dari 3 kata, maka N-Gram yang akan terbentuk dari contoh kalimat tersebut yaitu "pindah ibu kota".

2.4 Multinomial Logistic Regression

Multinomial Logistic Regression juga dikenal sebagai *Softmax Regression* merupakan sebuah pendekatan pembelajaran mesin untuk mengklasifikasikan satu kelas vektor fitur dari beberapa kelas [9]. *Multinomial Logistic*



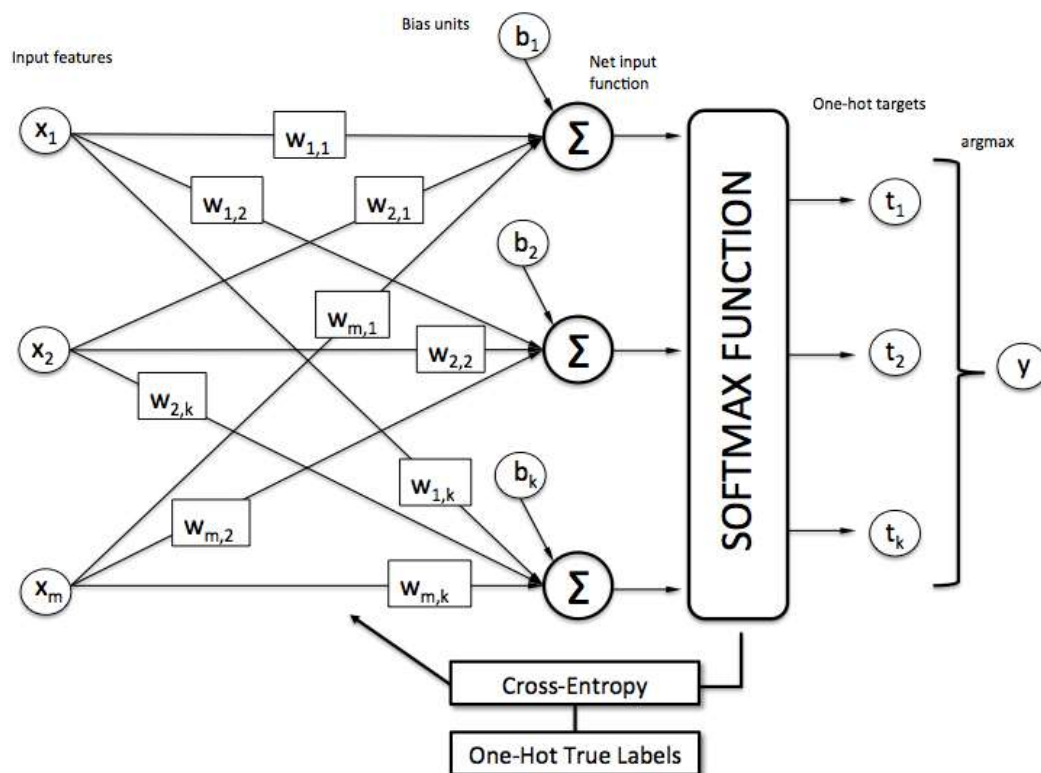
Regression merupakan sebuah metode klasifikasi generalisasi dari *Logistic Regression* untuk menyelesaikan masalah *multiclass*. Perbedaan yang mendasar pada kedua metode tersebut terdapat pada fungsi aktifiasinya. Metode *Logistic Regression* menggunakan fungsi aktifasi *Sigmoidi*, sedangkan *Multinomial Logistic Regression* menggunakan fungsi aktifasi *Softmax*. Rumus fungsi aktifasi *Softmax* dituliskan sebagai berikut:

$$softmax(z^{(i)}) = \frac{e^{z^{(i)}}}{\sum_{j=0}^k e^{z_j^{(i)}}} \tag{2}$$

Dimana parameter z didefinisikan sebagai berikut:

$$z = X \cdot W^T + b \tag{3}$$

Rumus diatas dinamakan dengan *net input fuction* atau *forward function*, dimana X merupakan vektor fitur, W merupakan vektor bobot, dan b merupakan vektor bias. Pada **Gambar 1** berikut dapat dilihat model arsitektur *Multinomial Logistic Regression*.



Gambar 1. Model Arsitektur *Multinomial Logistic Regression*

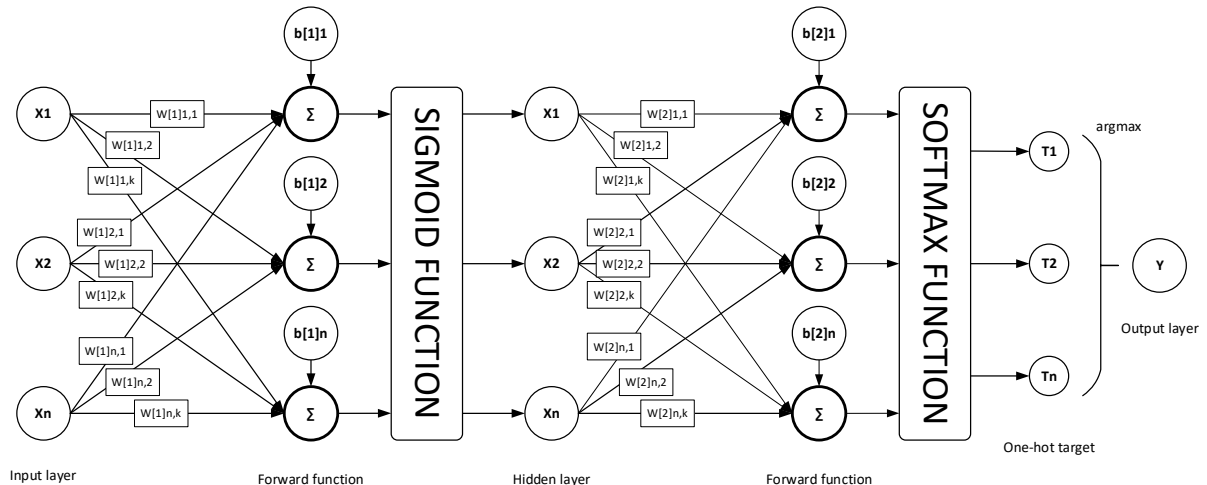
2.5 Artificial Neural Network

Artificial Neuran Network (ANN) merupakan sebuah teknik dalam *machine learning* yang paling banyak digunakan yang terinspirasi dari anatomi otak manusia [10]. ANN terdiri dari banyak neuron yang saling terhubung dan bekerja bersamaan secara sistematis untuk menyelesaikan masalah klasifikasi. Dalam ANN terdapat algoritma *back-propagation*, yang mana secara iteratif digunakan untuk mengubah bobot yang ada pada setiap neuron sehingga dapat meminimalkan kesalahan.

Arsitektur dalam ANN terdiri dari tiga macam layer, yaitu input layer, hidden layer dan output layer. Ketiga layer tersebut saling terhubung satu sama lain. Pada setiap layer terdapat beberapa neuron. Input layer digunakan untuk menerima input dari luar sistem, dalam penelitian ini vektor TF-IDF akan menjadi input. Di dalam hidden layer akan dilakukan proses perhitungan bobot dan bias terhadap input yang diberikan (*forward function*), kemudian hasilnya dimasukkan ke dalam fungsi aktifasi sigmoid dengan rumus sebagai berikut.

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{4}$$

Hasil dari fungsi aktifasi sigmoid akan dilakukan *forward function* kembali, kemudian hasilnya dimasukkan ke dalam fungsi aktifasi *softmax*. Fungsi aktifasi *softmax* akan menghasilkan output berupa kelas prediksi. Setelah mendapatkan kelas prediksinya kemudian dilakukan *back-propagation* untuk menghitung *error* dan melakukan update bobot sehingga dapat meminimalkan *error*.



Gambar 2. Arsitektur Artificial Neural Network

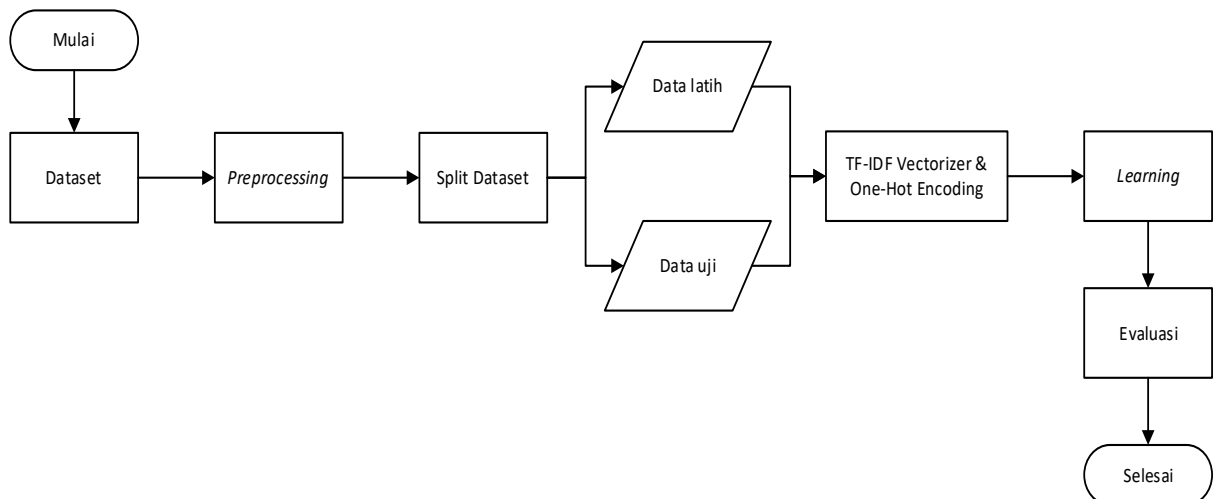
2.6 One-Hot Encoding

One-Hot Encoding merupakan prosedur yang paling umum untuk mengatasi *multi-class classification task* [11]. Dalam versi yang paling sederhana dari *One-Hot Encoding*, setiap kelas harus dikonversi menjadi vektor yang ukurannya sama dengan jumlah himpunan kelas yang mungkin, dan semua nilai dalam vektor ini adalah 0 kecuali untuk posisi indeks yang sesuai dengan kelasnya [12]. Oleh karena itu setiap kelas yang mungkin harus dikonversi menjadi angka terlebih dahulu sebelum dilakukan proses *One-Hot Encoding*. Dalam penelitian ini terdapat 3 kelas yaitu non argumen, argumen dan *unknown*. Masing-masing kelas tersebut dikonversi menjadi angka 0 untuk kelas non argumen, angka 1 untuk kelas argumen dan angka 2 untuk kelas *unknown*. Sehingga *One-Hot Encoding* dari ketiga kelas tersebut menjadi [1, 0, 0] untuk kelas 0, [0, 1, 0] untuk kelas 1 dan [0, 0, 1] untuk kelas 2.

3. HASIL DAN PEMBAHASAN

3.1 Rancangan Sistem

Sistem yang dibangun pada penelitian ini bertujuan untuk melakukan klasifikasi teks berupa *tweet* ke dalam tiga kelas yaitu non argumen, argumen dan *unknown*. Secara garis besar sistem yang dibangun terdiri dari lima tahapan, yaitu pengumpulan dataset, preprocessing, *vectorizing*, *learning*, dan evaluasi.



Gambar 3. Flowchart Sistem

a. Pengumpulan Dataset

Proses pengumpulan dataset dilakukan dengan cara *crawling* menggunakan *Application Programming Interface* (API) yang sudah disediakan oleh *twitter*. Kata kunci yang digunakan saat *crawling* yaitu “pindah ibu kota” dan data yang diambil ketika *crawling* hanya data *text/tweet* saja. Data *tweet* yang di-*crawling* yaitu mulai dari bulan Juni 2019 hingga September 2019 dan mendapatkan 5046 *tweet*. Setelah dataset terkumpul, setiap *tweet* yang ada diberi label/kelas ke dalam tiga kelas yaitu non argumen, argumen dan *unknown* secara manual.



b. Preprocessing

Setelah data *tweet* sudah terkumpul, selanjutnya yaitu tahap *preprocessing*. Terdapat 10 tahap preprocessing yang dilakukan dalam penelitian ini. Setiap tahap preprocessing sudah dijelaskan pada bagian 2.2. Pada bagian ini akan dijelaskan contoh dari setiap tahap *preprocessing* pada Tabel 1 berikut.

Tabel 1. Preprocessing

Proses	Masukkan	Keluaran
Remove URL	Mungkin Ini Cara JKW untuk Mengurangi Macet di Jkt dan Mengurangi Manusia2 Yg Di Jkt untk Pindah Ke Kalimantan https://t.co/4LiSTHs9of	Mungkin Ini Cara JKW untuk Mengurangi Macet di Jkt dan Mengurangi Manusia2 Yg Di Jkt untk Pindah Ke Kalimantan
Casefolding	Mungkin Ini Cara JKW untuk Mengurangi Macet di Jkt dan Mengurangi Manusia2 Yg Di Jkt untk Pindah Ke Kalimantan	mungkin ini cara jkw untuk mengurangi macet di jkt dan mengurangi manusia2 yg di jkt untk pindah ke kalimantan
Tokenization	mungkin ini cara jkw untuk mengurangi macet di jkt dan mengurangi manusia2 yg di jkt untk pindah ke kalimantan	["mungkin", "ini", "cara", "jkw", "untuk", "mengurangi", "macet", "di", "jkt", "dan", "mengurangi", "manusia2", "yg", "di", "jkt", "untk", "pindah", "ke", "kalimantan"]
Remove Username and Hashtag	pemerintah berencana untuk pindah ibu kota. jika berhasil, indonesia bisa menjadi negara yang pertama yang menerapkan skema tersebut dalam sejarah pemindahan ibu kota suatu negara. #ibukotabaru via @detikfinance	pemerintah berencana untuk pindah ibu kota. jika berhasil, indonesia bisa menjadi negara yang pertama yang menerapkan skema tersebut dalam sejarah pemindahan ibu kota suatu negara. via
Remove Punctuation	pemerintah berencana untuk pindah ibu kota. jika berhasil, indonesia bisa menjadi negara yang pertama yang menerapkan skema tersebut dalam sejarah pemindahan ibu kota suatu negara. via	pemerintah berencana untuk pindah ibu kota jika berhasil indonesia bisa menjadi negara yang pertama yang menerapkan skema tersebut dalam sejarah pemindahan ibu kota suatu negara via
Remove Emoticon	alhamdulillah pak anies jadi lumayan kelihatann nanti 🤔🤔🤔	alhamdulillah pak anies jadi lumayan kelihatann nanti
Remove Digit/Number	perkenalkan nama saya adalah jaya winata singkatannya jawin saya lahir di kota bekasi, tanggal 17 mei 1994 tapi saat saya umur 10 bulan saya pindah ke kota bandung kebetulan saya adalah sepupu dari pihak ayah kiandre ibu saya adalah adik dari beliau	perkenalkan nama saya adalah jaya winata singkatannya jawin saya lahir di kota bekasi, tanggal mei tapi saat saya umur bulan saya pindah ke kota bandung kebetulan saya adalah sepupu dari pihak ayah kiandre ibu saya adalah adik dari beliau
Remove Duplicate	pindah ibu kota efektif kah pindah ibu kota efektif kah pindah ibu kota efektif kah	pindah ibu kota efektif kah
Replace Abbreviation	mungkin ini cara jkw untuk mengurangi macet di jkt dan mengurangi manusia yg di jkt untk pindah ke kalimantan	mungkin ini cara jokowi untuk mengurangi macet di jakarta dan mengurangi manusia yang di jakarta untuk pindah ke kalimantan
Stemming	mungkin ini cara jokowi untuk mengurangi macet di jakarta dan mengurangi manusia yang di jakarta untuk pindah ke kalimantan	mungkin ini cara jokowi untuk kurang macet di jakarta dan kurang manusia yang di jakarta untuk pindah ke kalimantan

Setelah dilakukan *preprocessing* didapatkan sebanyak 1811 data *tweet* yang sudah bersih dan dapat diproses ke tahap selanjutnya yaitu tahap *learning*.

c. Split Dataset

Dataset yang sudah bersih akan dilakukan proses pemisahan data menjadi 2 bagian, yaitu data test dan data train. Proses pengambilan data untuk data test dilakukan secara acak namun memperhatikan jumlah kelas yang ada sehingga tidak ada kelas yang dominan. Penulis menetapkan jumlah data untuk data test sebanyak 450 data (150 data kelas 0, 150 data kelas 1 dan 150 data kelas 2). Sedangkan untuk data train berisi data-data yang tidak terpilih sebagai data test.

Tabel 2. Jumlah Data Train dan Data Test

Jenis	Jumlah Kelas		
	0	1	2
Data Train	117	261	983



Jenis	Jumlah Kelas		
	0	1	2
Data Test	150	150	150

d. Learning

Setelah dataset sudah terbagi menjadi 2 bagian (data train dan data test), data tersebut dapat dilakukan proses *learning*. Sebelum masuk proses *learning*, data train dan data test harus diubah terlebih dahulu ke dalam bentuk vektor. Dalam penelitian ini proses pengubahan teks ke dalam bentuk vektor menggunakan metode TF-IDF. Selain itu, label atau kelas yang terdapat pada data train dan data test harus diubah ke dalam bentuk *One-Hot Encoding* yang sudah dibahas pada bagian 2.6.

Pada tahap *learning*, metode yang digunakan yaitu *Multinomial Logistic Regression* untuk melakukan klasifikasi berdasarkan data yang sudah tersedia. Dalam penelitian ini, terdapat beberapa parameter yang harus dimasukkan ke dalam model *Multinomial Logistic Regression*, yaitu diantaranya *learning rate*, total *epoch*, data train yang sudah divektorisasi, kelas data latih yang sudah di-*encode*, dan data validasi berupa data test yang sudah divektorisasi dan kelas data test yang sudah di-*encode*. Selain metode *Multinomial Logistic Regression*, dalam penelitian ini juga terdapat metode pembandingan yaitu *Artificial Neural Network (ANN)*.

Dalam penelitian ini, terdapat beberapa metrics evaluasi, diantaranya *macro precision average*, *macro recall average* dan *macro F1-Score average*. *Macro Precision Average* digunakan untuk mengetahui proporsi rata-rata identifikasi positif yang benar pada keseluruhan data [13]. *Macro Precision Average* dapat dirumuskan sebagai berikut:

$$precision = \frac{\sum_{i=1}^n \frac{TP_i}{TP_i + FP_i}}{n} \tag{5}$$

Macro Recall Average digunakan untuk mengetahui proporsi rata-rata kelas positif aktual yang diidentifikasi dengan benar [13]. *Macro Recall Average* dapat dirumuskan sebagai berikut:

$$recall = \frac{\sum_{i=1}^n \frac{TP_i}{TP_i + FN_i}}{n} \tag{6}$$

Macro F1-Score Average digunakan untuk mengetahui hubungan antara kelas positif dan kelas yang sudah diklasifikasikan berdasarkan rata-rata per-kelas [13]. *Macro F1-Score Average* dapat dirumuskan sebagai berikut:

$$F1 = \frac{\sum_{i=1}^n (2 * \frac{precision_i * recall_i}{precision_i + recall_i})}{n} \tag{7}$$

Akurasi merupakan tingkat keberhasilan suatu model dalam melakukan klasifikasi. Akurasi dalam penelitian ini dirumuskan sebagai berikut:

$$acc = \frac{jumlah\ keseluruhan\ data\ yang\ diidentifikasi\ benar}{jumlah\ keseluruhan\ data} \tag{8}$$

Dalam penelitian ini terdapat beberapa skenario pengujian yang dilakukan untuk mengetahui performansi yang dihasilkan dari metode *Multinomial Logistic Regression (MLR)* dan *Artificial Neural Network (ANN)*. Dataset yang digunakan dalam penelitian ini bersifat *imbalanced* (tidak seimbang). Setelah dilakukan pelabelan secara manual, pendapat atau komentar masyarakat Indonesia terhadap isu pemindahan Ibu Kota Indonesia melalui *tweet* di jejaring sosial *twitter* cenderung tidak berhubungan dengan topiknya atau masuk ke dalam kelas *unknwon*. Dengan data yang seperti itu, artinya masyarakat Indonesia yang menggunakan *twitter* tidak terlalu menganggap serius akan isu pemindahan Ibu Kota Indonesia. Namun pada posisi kedua, *tweet* masuk ke dalam kelas argumen, artinya masih banyak juga pengguna *twitter* di Indonesia yang menyampaikan pendapatnya secara jelas dan berargumen. Pada posisi terakhir, *tweet* masuke dalam kelas non argument, artinya ada beberapa pengguna *twitter* di Indonesia menyampaikan pendapatnya tanpa disertai argumen.

Tabel 3. Dataset *Tweet* Pemindahan Ibu Kota

Kelas	Jumlah
Non Argumen (0)	267
Argumen (1)	411
Unknown (2)	1133
Total	1811

Pada Tabel 4 berikut terdapat 12 skenario pengujian yang sudah dilakukan dalam penelitian ini. Skenario tersebut dilakukan pengujian terhadap 2 metode, yaitu MLR dan ANN sebagai metode pembandingan.



Tabel 4. Skenario Pengujian

Skenario	Preprocessing		Fitur		
	Stemming (S)	Replace Abbreviation Word (R)	Unigram (U)	Bigram (B)	Trigram (T)
U	X	X	✓	X	X
B	X	X	X	✓	X
T	X	X	X	X	✓
SU	✓	X	✓	X	X
SB	✓	X	X	✓	X
ST	✓	X	X	X	✓
RU	X	✓	✓	X	X
RB	X	✓	X	✓	X
RT	X	✓	X	X	✓
SRU	✓	✓	✓	X	X
SRB	✓	✓	X	✓	X
SRT	✓	✓	X	X	✓

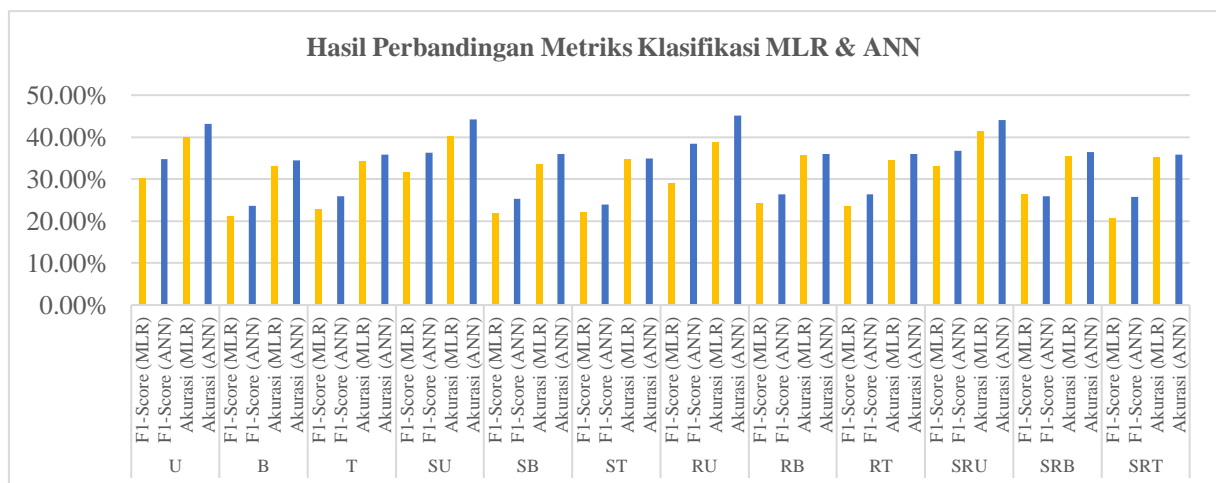
3.2 Pengaruh Kombinasi Preprocessing Dan Fitur Terhadap Klasifikasi

Pengujian yang sudah dilakukan dalam penelitian ini terdapat 12 skenario pengujian dengan kombinasi preprocessing dan fitur yang tertera pada **Table 4**. Pada setiap skenario dilakukan proses klasifikasi dengan menggunakan metode MLR. Terdapat tiga parameter yang harus diisi untuk melakukan proses klasifikasi dengan metode MLR, yaitu diantaranya sebagai berikut:

1. Data train, dimana data train tersebut harus sudah dilakukan proses vektorisasi (dalam penelitian ini menggunakan TF-IDF) sebelum dilakukan proses *learning*.
2. *Learning rate* sebesar 0,5.
3. *Epoch* sebesar 2000 *epoch*.

Sedangkan untuk metode ANN, terdapat 4 parameter yang harus diisi untuk melakukan proses klasifikasi, yaitu diantaranya sebagai berikut:

1. Data train, dimana data train tersebut harus sudah dilakukan proses vektorisasi (dalam penelitian ini menggunakan TF-IDF) sebelum dilakukan proses *learning*.
2. *Learning rate* sebesar 0,004.
3. *Epoch* sebesar 2000 *epoch*.
4. *Hidden layer* sebesar 50.



Gambar 4. Hasil Metriks Klasifikasi MLR

Pada **Gambar 4**, hasil metriks yang paling baik yaitu pada skenario SRU (*Stemming, Replace Abbreviation Word* dan *Unigram*) untuk model MLR. Sedangkan untuk model ANN, hasil skenario yang paling baik yaitu RU (*Replace Abbreviation Word* dan *Unigram*). Jika dilihat dari keseluruhan matriks evaluasi tersebut, terdapat peningkatan nilai matriks evaluasi pada model ANN. Hal ini dikarenakan pada model ANN terdapat 2 layer dan 2 fungsi aktivasi sebelum hasilnya dikeluarkan. Fungsi aktivasi pada kedua layer tersebut yaitu *sigmoid* pada layer pertama dan *softmax* pada layer kedua. Fungsi aktivasi *sigmoid* akan menghitung nilai probabilitas dari hasil *forward function* antara input dengan bobot dan bias. Kemudian hasilnya dilakukan *forward function* kembali dan hasil dimasukkan ke dalam fungsi aktivasi softmax. Fungsi aktivasi softmax akan menghitung probabilitas dari semua kelas yang ada dengan kelas target yang memiliki nilai probabilitas tertinggi [14].



Jika dilihat dari hasil keseluruhan metrik evaluasi terhadap semua skenario pengujian baik MLR maupun ANN, penggunaan fitur *Unigram* menghasilkan performansi yang lebih baik daripada *Bigram* dan *Trigram*. Hal ini terjadi dikarenakan terdapat banyak sekali kata-kata yang ada di data test tidak tersedia di dalam data train atau dinamakan dengan *out of vocabulary* (oov). Berikut jumlah serta contoh kata oov pada setiap skenario.

Tabel 5. Out of Vocabulary (OOV)

Skenario	Jumlah	Contoh
U	944	terkontrol, terlihat, terancam, popularitas, nasionalisme, kebingungan, dll.
B	4632	ada ancaman, ada benarnya, apa adanya, berapa kerugian, keadaan rakyatnya, dll.
T	5782	ada di negeri, ada didepan mata, gak becus kerja, pindah dikira murah, dll.
SU	1245	ancaman, anggaran, berencana, besanan, seandainya, sebelum, jamannya, dll.
SB	4806	zaman belanda, untuk pemindahan, tidak cukup, ibu kotanya, jakarta dipindah, dll.
ST	5932	jakarta dalam kesehariannya, kalangan asn pns, nyatanya pindah ibu, dll.
RU	1020	zaman, pendatang, mematikan, didepan, bercanda, besuk, dangkal, dll.
RB	4722	ada ancaman, akan terealisasi, babat hutan, bagaimana konsepnya, dll.
RT	5801	cepat pindah biar, ibu kota mereka, ikutan pindah gak, melihat potensi tol, dll.
SRU	1320	alasan, bangunan, kedaulatan, kepemimpinan, rencananya, sebaiknya, dll.
SRB	4895	tunggu waktunya, tidak cukup, setiap negara, pindah sekalian, kota sejarah, dll.
SRT	5950	yg pindah kota, tidak di jawa, pindah sekedar usulan, pindah kenapa tidak, dll.

Selain oov pada data test, peneliti juga menemukan beberapa *tweet* yang salah label, namun *classifier* berhasil memprediksi *tweet* tersebut dengan benar. Akibatnya akurasi yang dihasilkan oleh *classifier* berkurang dikarenakan *miss prediction*. Berikut contoh *tweet* yang salah label namun diprediksi secara benar oleh *classifier*.

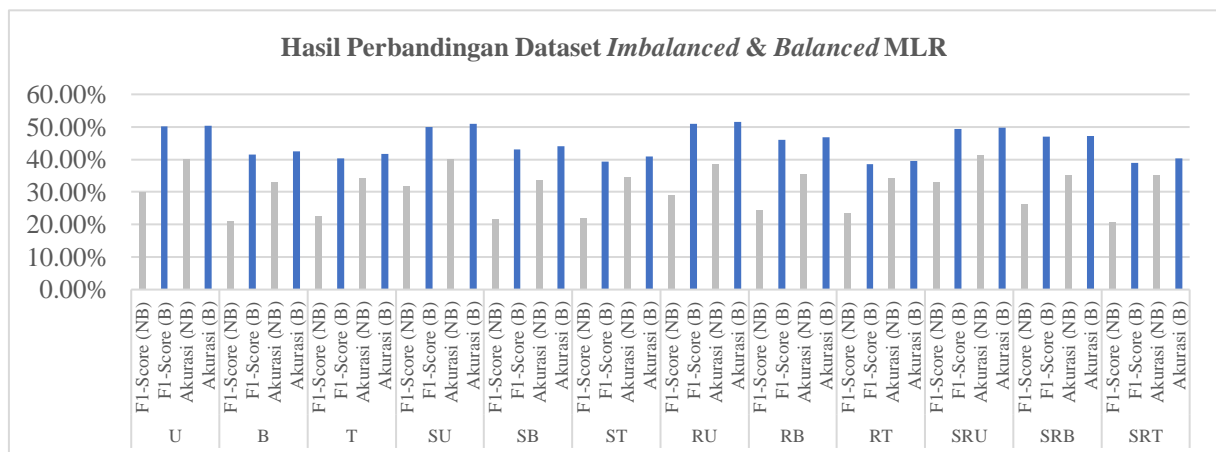
3.3 Pengaruh Balancing Data Terhadap Klasifikasi

Berdasarkan hasil pengujian sebelumnya yang sudah dipaparkan pada bagian 4.1, dataset yang digunakan bersifat *imbalanced* akibatnya model akan cenderung melakukan klasifikasi data baru ke kelas yang dominan dalam dataset tersebut. Oleh karena itu, penulis menjoba melakukan pengujian terhadap dataset yang sudah dilakukan penyesuaian sehingga dataset menjadi *balanced*. Proses penyesuaian data dilakukan dengan cara *oversampling* pada data train (menduplikasi *tweet* yang kelasnya minoritas). Dalam penelitian ini, *tweet* dengan kelas yang paling banyak yaitu kelas 2, oleh karena itu kelas 0 dan 1 dilakukan duplikasi *tweet* sehingga jumlahnya sama dengan *tweet* kelas 2.

Tabel 6. Dataset Balanced

Kelas	Jumlah
Non Argumen (0)	983
Argumen (1)	983
Unknown (2)	983
Total	2949

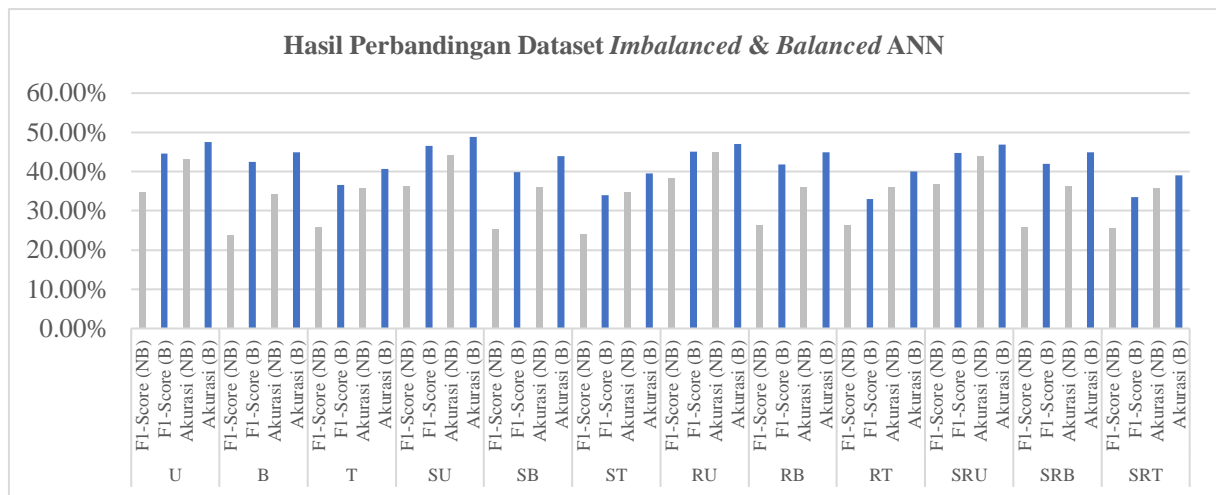
Pengujian yang dilakukan terhadap dataset yang sudah *balanced* sama dengan pengujian sebelumnya yang tertera pada **Tabel 4**. Mengingat dataset sudah dilakukan *oversampling* untuk *balancing* dataset, maka dari itu terdapat perbedaan parameter *epoch* pada model MLR. *Epoch* yang digunakan untuk dataset *balanced* sebesar 4000 epoch. Hasil semua pengujian terhadap dataset yang sudah *balanced* dapat dilihat pada gambar grafik berikut.



Gambar 5. Hasil Balancing Dataset MLR

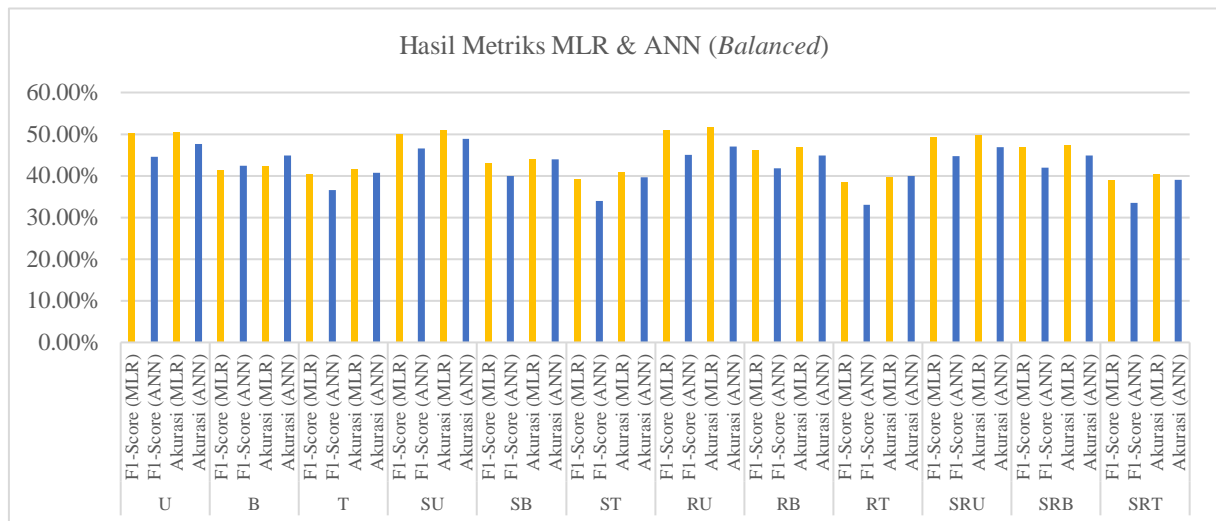


Pada Gambar 5, setelah dilakukan *oversample* pada dataset, hasil pengujian menggunakan model MLR menunjukkan adanya sedikit peningkatan performansi dibandingkan dengan dataset yang *imbalanced*. Hal ini terjadi karena kelas yang terdapat pada data train seimbang, tidak ada kelas yang dominan sehingga model akan memberikan performansi yang lebih baik dalam melakukan klasifikasi. Selain itu, faktor lain yang mempengaruhi performansi *classifier* yaitu persebaran kosa kata yang terdapat di data train dan data test. Berdasarkan **Table 5**, terdapat banyak sekali kosa kata yang terdapat di data test namun tidak tersedia di dalam data train, akibatnya kata-kata oov tersebut tidak masuk ke dalam perhitungan. Tidak menutup kemungkinan bahwa kata-kata oov tersebut merupakan kata-kata yang berpengaruh besar terhadap klasifikasi.



Gambar 6. Hasil Balancing Dataset Model ANN

Pada Gambar 6, setelah dilakukan *oversample* pada dataset, hasil pengujian menggunakan model ANN menunjukkan adanya sedikit peningkatan performansi dibandingkan dengan dataset yang *imbalanced*. Namun jika dibandingkan dengan model MLR, model MLR menghasilkan performansi yang lebih baik untuk dataset yang *balanced*.



Gambar 7. Hasil Metriks MLR & ANN (Balanced)

4. KESIMPULAN

Berdasarkan hasil pengujian yang sudah dilakukan dengan menerapkan metode *Multinomial Logistic Regression* (MLR) dan *Artificial Neural Network* (ANN), dapat disimpulkan bahwa model ANN bekerja lebih baik dibandingkan dengan model MLR untuk dataset yang *imbalanced*. Sedangkan untuk dataset yang *balanced* model MLR menghasilkan performansi yang lebih baik. Hal ini dikarenakan model MLR tidak memiliki *hidden layer* dan hanya memiliki satu fungsi aktivasi yaitu *softmax*, sehingga setelah dilakukan proses *forward* dan mendapatkan hasil dari fungsi aktivasi *softmax* model akan langsung mendapatkan hasil klasifikasinya. Sedangkan pada model



ANN memiliki *hidden layer* dan memiliki 2 fungsi aktivasi yaitu *sigmoid* dan *softmax*, sehingga setelah dilakukan proses *forward* model akan memasukkan hasilnya ke dalam fungsi aktivasi *sigmoid*. Setelah itu model akan melakukan proses *forward* dan memasukkan hasilnya ke dalam fungsi aktivasi *softmax*, kemudian akan mendapatkan hasil klasifikasinya. Selain itu, dataset yang *imbalanced* akan berpengaruh juga terhadap hasil klasifikasi dan metrik. Dengan menggunakan dataset yang *imbalanced*, model akan cenderung mengklasifikasikan data baru dengan kelas yang paling dominan pada dataset tersebut (dalam penelitian ini hasil klasifikasi akan cenderung ke kelas 2). Setelah dilakukan *balancing* pada dataset, hasil klasifikasi akan menjadi lebih baik dan seimbang. Selain itu, *out of vocabulary* (oov) juga berpengaruh terhadap performansi kedua model tersebut. Semakin banyak kata-kata yang terdapat di data test namun tidak tersedia di data train, maka model akan sulit untuk menghitung dan mengklasifikasikan data tersebut karena terdapat kata/informasi yang hilang. Dalam sebuah penelitian, dataset sangatlah berpengaruh pada hasil klasifikasi, terutama dataset yang bersifat text berupa *tweet*. Oleh karena itu, diharapkan untuk penelitian selanjutnya dapat mengenali sifat dari dataset tersebut. Dataset harus seimbang agar proses klasifikasi mendapatkan hasil yang maksimal. Perlu diperhatikan juga, keberagaman kata mempengaruhi proses klasifikasi, semakin beragam kata-kata maka *classifier* akan semakin sulit untuk mengenali data tersebut. Selain itu, dalam proses pelabelan yang dilakukan lebih dari 1 orang harus diperhatikan dan disetujui kelas datanya agar tidak ada data yang ambigu.

REFERENCES

- [1] D. Jurafsky and J. Martin, "Logistic Regression," *Speech Lang. Process. An Introd. to Nat. Lang. Process. Comput. Linguist. Speech Recognit.*, p. 20, 2019.
- [2] M. Dusmanu, E. Cabrio, and S. Villata, "Argument Mining on Twitter: Arguments, Facts and Sources," pp. 2317–2322, 2018.
- [3] M. S. Mubarak, A. Adiwijaya, and M. D. Aldhi, "Aspect-based sentiment analysis to review products using Naïve Bayes," *AIP Conf. Proc.*, vol. 1867, 2017.
- [4] H. N. Rohman and I. Asror, "Automatic Detection of Argument Components in Text Using Multinomial Nave Bayes Clasiffier," in *Journal of Physics: Conference Series*, 2019.
- [5] T. Akhir, "Perbandingan Metode Naïve Bayes dan Metode K-Nearest Neighbor pada Klasifikasi Teks Pelamar Kerja PT . Telkom Indonesia Program Studi Sarjana Teknik Informatika Fakultas Informatika Universitas Telkom Bandung," 2020.
- [6] P. Bafna, D. Pramod, and A. Vaidya, "Document clustering: TF-IDF approach," *Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016*, pp. 61–66, 2016.
- [7] Daeli, N.O.F. and Adiwijaya, A., 2020. Sentiment Analysis on Movie Reviews using Information Gain and K-Nearest Neighbor. *Journal of Data Science and Its Applications*, 3(1), pp.1-7.
- [8] D. Jurafsky and J. H. Martin, "Chapter 3: N-Gram Language Models N-Gram Language Models," *Speech Lang. Process.*, 2019.
- [9] T. Kim and S. J. Wright, "PMU Placement for Line Outage Identification via Multinomial Logistic Regression," vol. 3053, no. c, 2016.
- [10] A. Bhardwaj, A. Tiwari, H. Bhardwaj, and A. Bhardwaj, "A genetically optimized neural network model for multi-class classification," *Expert Syst. Appl.*, vol. 60, pp. 211–221, 2016.
- [11] P. Rodríguez, M. A. Bautista, J. González, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image Vis. Comput.*, vol. 75, pp. 21–31, 2018.
- [12] Zhang, X., & LeCun, Y. (2017). Which encoding is the best for text classification in chinese, english, japanese and korean?. arXiv preprint arXiv:1708.02657.
- [13] Sokolova, M. and Lapalme, G., (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4), pp.427-437.
- [14] Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., (2018). Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378.