

Klasifikasi Keberpihakan *tweet* menggunakan *Multinomial Naïve Bayes* (Studi Kasus : Pemilihan Presiden 2019)

Alfian Yulianto¹, Anisa Herdiani², Indra Lukmana Sardi³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹fahfian@students.telkomuniversity.ac.id, ²anisaherdiani@telkomuniversity.ac.id,

³indraluk@telkomuniversity.ac.id

Abstrak

Tweet yang tersebar di media sosial twitter dapat menunjukkan keberpihakan *tweet* terhadap calon presiden dan wakil presiden yang akan menjabat. Banyak tweet yang dapat menunjukkan keberpihakan terhadap suatu paslon mulai dari tweet yang memuji paslon yang didukungnya hingga menjelekkan paslon yang tidak didukungnya. Penelitian ini menganalisis dari banyaknya tweet yang beredar untuk menilai keberpihakannya berdasarkan data *tweet* pada sosial media twitter. Melalui tugas akhir peneliti memberikan solusi dengan klasifikasi otomatis yang dilakukan komputer dengan mengklasifikasikan *tweet* tersebut apakah memihak Jokowi atau Prabowo. klasifikasi yang dibangun menggunakan metode multinomial naïve bayes classifier, Hasil pengujian dengan *10-folds cross validation* pada penelitian ini memiliki rata-rata hasil *F1-Score* 0,71 dan akurasi 0,72.

Kata kunci : Klasifikasi, Pilpres, Pemilihan Presiden, *Multinomial Naïve Bayes*

Abstract

Tweets that are spread on social media twitter can show the tweet alignments of candidates for president and vice president who will take office. Many tweets can show partiality towards a paslon, starting from tweets that compliment a paslon that it supports to vilify a paslon that it does not support. This study analyzes the number of tweets in circulation to assess its alignments based on tweet data on twitter social media. Through this final project the researcher provides a solution with an automatic classification by a computer by classifying the tweet whether it is in favor of Jokowi or Prabowo. the classification was built using the multinomial naïve bayes classifier method. The test results with 10-fold cross validation in this study had an average *F1-Score* of 0.71 and an accuracy of 0.72.

Keywords: Classification, *Multinomial Naïve Bayes*, Presidential Election.

1. Pendahuluan

Latar Belakang

Berdasarkan Pasal 22E Undang-Undang Dasar Negara Republik Indonesia 1945 diperkenalkan Pemilu Legislatif, Pemilu Presiden dan Wakil Presiden [1]. Seiring berjalannya pemilihan presiden tahun 2019 banyak berita perihal pemilihan presiden diberbagai media sosial termasuk twitter. Twitter merupakan aplikasi media sosial yang dikembangkan oleh Jack Dorsey di San Farnscisco pada tahun 2006. Setiap tahun jumlah pengguna twitter berkembang dari sejak pertama kali dibuat. Twitter digunakan untuk menampilkan pesan dari para penggunanya yang disebut sebagai *tweet*, seiring berkembang dan banyaknya pengguna twitter, twitter dapat dimanfaatkan dalam berbagai aspek mulai dari untuk komunikasi, pembelajaran hingga untuk kampanye politik [2].

Twitter biasa digunakan politisi dan partai politik untuk menggalang simpatisan maupun dukungan karena dapat menekan pengeluaran biaya kampanye [2]. Twitter digunakan secara luas untuk pertimbangan politik. *tweet* dapat memberi tahu tentang ikatan ideologis antara partai dan potensi koalisi atau keberpihakan politik dalam pemilihan. Twitter tidak hanya digunakan untuk menyebarkan pendapat politik, tetapi juga untuk membahas pendapatnya dengan pengguna lain termasuk perihal pihak yang didukungnya. Politisi biasa berkomunikasi dengan pemilih yang menggunakan media sosial dan berusaha memobilisasi pendukung [3]. Semakin banyak pengikut yang dimiliki maka semakin besar yang berpihak pada mereka. Banyak dijumpai *tweet* pada linimasa pengguna media sosia yang memaparkan kelebihan dan kekurang setiap pihak dari pasangan calon untuk membuat orang lain agar beralih pihak. Orang akan belajar dari apa yang diposting orang lain, karena mereka terlibat dan terkena berita secara berakaitan sesuai tema yang mereka ikuti [4]. Banyaknya *tweet* yang tersebar pada linimasa dapat diklasifikasikan berdasarkan sentimennya, dari klasifikasi sentimen keberpihakan tersebut dapat memberi informasi mengenai pandangan masyarakat terhadap suatu paslon tertentu yang dapat menjadi acuan keberpihakan seseorang terhadap suatu paslon. Namun, untuk mengklasifikasikan *tweet* tersebut berpihak terhadap paslon tertentu yang jumlahnya dapat mencapai ribuan tentu akan membutuhkan waktu yang lama.

Berdasarkan hal tersebut, terdapat solusi yang dapat ditawarkan adalah menggunakan *sentimen classification*. *Sentimen classification* merupakan klasifikasi teks yang dilakukan oleh komputer dengan tujuan mengkategorikan

teks berdasarkan polaritas seperti positif atau negatif, suka ataupun tidak suka atau dengan kategori polaritas yang lainnya [5]. Metode yang digunakan pada penelitian ini adalah *Multinomial Naive Bayes*, metode ini dipilih karena memiliki rata-rata akurasi yang cukup tinggi antara 85%-90% untuk klasifikasi sentimen teks dengan data *tweet* [6], metode alasan lainnya metode ini dipilih yaitu karena menggunakan pendekatan probabilistik yang berarti lebih cocok digunakan untuk mengklasifikasikan suatu kelas yang belum dapat diprediksi secara pasti dikarenakan sentimen yang terdapat dalam suatu *tweet* memiliki sifat yang berbeda-beda bisa implisit atau bahkan eksplisit, *tweet* tersebut juga terkadang tidak memiliki atribut yang saling berkorelasi dalam penyusunan suatu *tweet* dengan kelas sentimennya [7].

Topik dan Batasannya

Berdasarkan latar belakang yang sudah dijelaskan sebelumnya topik yang dibahas pada penelitian ini adalah sebagai berikut.

- Bagaimana mengklasifikasi keberpihakan suatu *tweet* dalam Pilpres 2019 ?
- Bagaimana performansi dari sistem yang dibangun dalam mengklasifikasikan keberpihakan dengan menggunakan *Multinomial Naive Bayes Classifier*?

Adapun beberapa batasan masalah terhadap sistem yang dibangun dalam penelitian ini adalah sebagai berikut.

- Dataset* yang digunakan berasal dari media sosial twitter.
- Dataset* yang diambil bertopik seputar pemilihan presiden 2019.
- Dataset* berbahasa Indonesia.
- Pelabelan *dataset* dengan kategori kelas “jokowi” dan “prabowo”.
- Tidak mengidentifikasi simbol, *link*, *emoticon*, gambar maupun video.

Tujuan

Pada penelitian ini bertujuan untuk mengimplementasikan metode *Multinomial Naive Bayes Classifier* pada *tweet* untuk mengklasifikasikan keberpihakan dan menganalisis hasil performansi dari klasifikasi yang telah dilakukan oleh sistem dengan menggunakan metode *Multinomial Naive Bayes Classifier*.

Organisasi Tulisan

Organisasi penulisan pada penelitian ini memiliki urutan sebagai berikut : Bagian 2 menjelaskan studi literatur dari landasan teori yang digunakan. Bagian 3 menjelaskan sistem yang dibangun yaitu sistem klasifikasi keberpihakan *tweet* dengan menggunakan metode *Multinomial Naive Bayes Classifier*. Bagian 4 menjelaskan mengenai hasil pengujian dan analisis hasil pengujian dari sistem yang telah dibangun, dan yang terakhir pada bagian 5 disampaikan kesimpulan dan saran dari penelitian.

2. Studi Terkait

2.1 Klasifikasi

Klasifikasi atau yang pada umumnya pada data mining bisa juga disebut sebagai *supervised learning* merupakan suatu bagian pada *data mining* yang terdapat pada *analytical processing* dari data. Klasifikasi merupakan teknik yang berfokus terhadap proses pembelajaran untuk kumpulan data (*dataset*) yang telah terpisah kedalam bagian label atau grup tertentu. Dalam proses pembelajaran akan diperoleh model yang mampu memberi prediksi atau estimasi bagian data mana yang tidak memiliki grup atau label tertentu, oleh karena itu diperlukan data yang telah terbagi dan memiliki grup ataupun label kelasnya masing-masing [8].

2.2 *Multinomial Naive Bayes Classifier*

Multinomial Naive Bayes adalah salah satu dari metode *supervised learning* [9]. *Multinomial NBC* merupakan pengembangan model dari algoritma *bayes* yang biasa digunakan dalam klasifikasi teks. Pada rumus *Multinomial Naive Bayes Classifier*, kelas dokumen tidak ditentukan hanya berdasarkan kata yang muncul, namun berdasarkan jumlah kemunculannya juga [10]. Adapun Probabilitas dokumen d yang terletak dikelas c memiliki perhitungan [9]:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (1)$$

Dimana $P(t_k|c)$ adalah probabilitas kondisional dari t_k yang berada pada dokumen yang dimiliki kelas c . Dari persamaan diatas dapat diketahui bahwasannya $P(t_k|c)$ merupakan *likelihood probability* dari t_k yang terdapat pada kelas c , di sisi lain $P(c)$ merupakan *prior probability* dokumen yang berada pada kelas c . Hasil dari *posterior probability* akan dibandingkan untuk menentukan kelas, kelas yang memiliki nilai *posterior probability* terbesar merupakan kelas yang dipilih sebagai hasil prediksi [9]. Berikut formula dari *Prior probability* :

$$P(c) = \frac{N_c}{N} \quad (2)$$

N_c merupakan jumlah dari kategori dari c . N jumlah kategori keseluruhan. Untuk *likelihood probability* akan dihitung kata atau fitur t_k pada seluruh dokumen latih pada c , dengan menggunakan *LaPlace Smoothing* memiliki Rumus [11]:

$$P(t_k|c) = \frac{N_k + 1}{|V| + N'} \quad (3)$$

Dimana N_k adalah jumlah kemunculan t_k dalam dokumen latih pada suatu kelas c dan N adalah jumlah total kata atau fitur yang terdapat pada c dokumen latih. Penambahan angka 1 pada formula 3 berfungsi sebagai *LaPlace Smoothing* yang dilakukan agar terhindar dari zero probability pada ekstraksi fitur, sehingga hasil akhir yang diperoleh tidak bernilai nol [9].

2.3 Evaluasi Performansi

Evaluasi performansi ialah parameter yang berguna untuk mengukur performa dari model yang telah dibangun seberapa akurat model sistem yang telah dibangun tersebut. pengukuran parameter akan dapat diperoleh dari representasi *confusion matrix* yang merupakan salah satu dari metode untuk mengukur kinerja suatu sistem klasifikasi yang dibangun [12], dijelaskan sebagai berikut :

- TP yaitu *true positive*, artinya kondisi saat suatu kelas *true* dan berhasil diklasifikasikan sebagai kelas *true*.
- TN yaitu *true negative*, artinya kondisi saat suatu kelas *false* dan berhasil diklasifikasikan sebagai kelas *false*.
- FP yaitu *false positive*, artinya kondisi saat suatu kelas *false* diklasifikasikan sebagai kelas *true*.
- FN yaitu *false negative*, artinya kondisi saat suatu kelas *true* diklasifikasikan sebagai kelas *false*.

Berdasarkan *confusion matrix* parameter yang akan digunakan untuk mengukur performansi diantaranya :

1. *Precision* adalah suatu bagian dari dokumen yang diambil oleh sistem yang memiliki nilai relevan [9].

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

2. *Recall* adalah suatu bagian dari dokumen yang relevan yang diambil oleh sistem [9].

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

3. *F1-Score* adalah ukuran tunggal yang merangkum *precision* dan *recall*, *F1* merupakan rata-rata harmonik antara *precision* dan *recall* yang digunakan untuk melihat keseimbangan antara *precision* dan *recall* [8].

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (6)$$

4. *Accuracy* adalah kedekatan antara nilai prediksi dengan nilai aktual, akurasi memberikan bobot yang sama untuk kesalahan label dari kedua jenis, yang dimana *FP* dan *FN* keduanya masuk dalam formula 7 [8].

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (7)$$

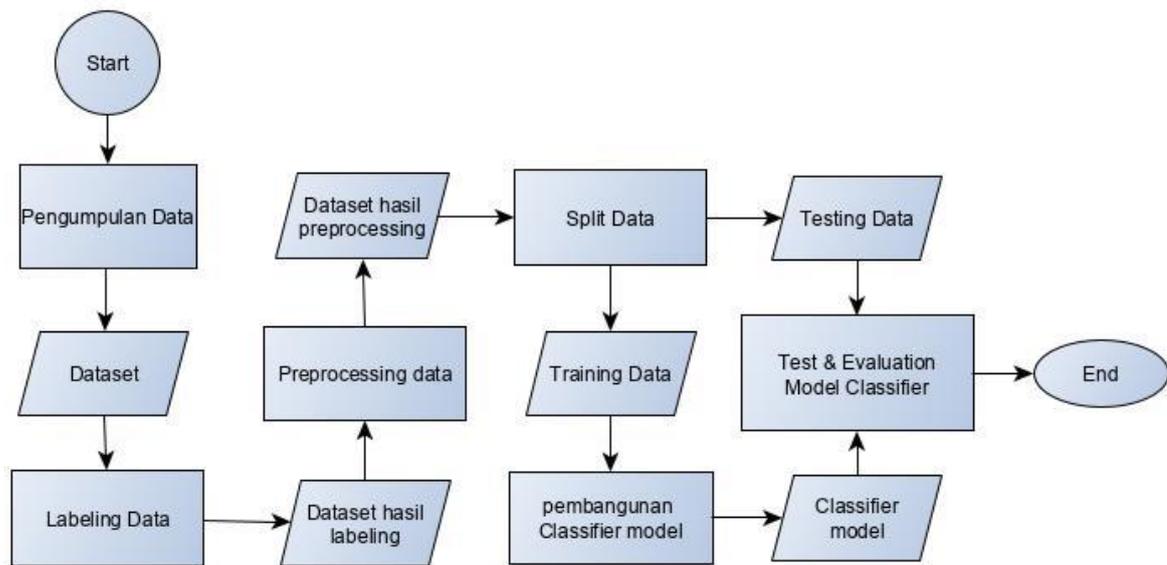
2.4 K-folds Cross Validation

Model yang telah dibangun selanjutnya akan dievaluasi dengan suatu pendekatan. pada penelitian ini digunakan *K-folds cross validation* untuk mengevaluasi model karena *K-folds cross validation* merupakan salah satu pendekatan untuk evaluasi model dari suatu classifier dengan cara data yang telah memiliki kelas akan dibagi kedalam K bagian.

Pada umumnya nilai K yang sering digunakan adalah sekitar 10 [8]. Salah satu dari K akan digunakan sebagai data *testing*, sementara data $(K-1)$ digunakan sebagai data *training*. Pendekatan akan diulangi dengan memilih K berbeda untuk data *testing* kemudian rata-rata yang diperoleh dari pengujian dijadikan sebagai nilai evaluasi terakhir [8].

3. Sistem yang Dibangun

Setelah bagian Gambar berikut merupakan alur metodologi penyelesaian yang dilakukan pada penelitian ini.



Gambar 1 Alur Proses Penelitian

Berikut penjelasan dari setiap proses pada gambar:

1. Pengumpulan Data

Pengguna di twitter menghasilkan banyak *tweet* setiap hari. Beberapa *tweet* ini tersedia untuk peneliti dan pengembang melalui API publik twitter. Pada penelitian ini digunakan sumber pustaka terbuka pada bahas python untuk pembuat *crawler data* yang biasa disebut *tweepy*.

Pada tahap ini akan dilakukan *crawling* atau pengumpulan data dari twitter untuk mendapatkan *dataset* dengan kata kunci “jokowi”, “prabowo”, “ma’ruf”, “sandi” yang selanjutnya dilakukan pelabelan data untuk kelas keberpihakannya.

Adapun orang yang layak dalam pelabelan data penelitian ini agar diperoleh keluaran data yang valid dan tidak subjektif peneliti menggunakan tiga orang dengan kriteria aktif pada media sosial twitter dan mengikuti berita seputar pilpres 2019, dengan ketentuan:

- a. Minimal sehari membuka twitter dan membaca berita politik seputar pilpres 2019 selama masa kampanye berlangsung.
- b. Mengikuti *trending topic* perihal pilpres 2019 setiap harinya

Untuk menghindari kecenderungan memihak suatu kubu ketika pelabelan data, maka pelabelan data dilakukan dengan beberapa kriteria sebagai berikut :

- Jika mengejek suatu paslon maka dinyatakan memihak pada paslon yang tidak diejek. (termasuk jika mengejek ditujukan pada orang-orang yang ada dalam kubu tersebut)
- Jika memuji atau membela suatu paslon maka dinyatakan memihak paslon yang dipuji atau dibelanya (termasuk jika memuji atau membela ditujukan pada orang-orang yang ada dalam kubu tersebut)
- Jika memberi saran atau nasihat suatu paslon, maka memihak pada paslon tersebut (termasuk jika ditujukan pada orang-orang yang ada dalam kubu tersebut)

Berikut adalah contoh data yang diperoleh dari hasil *crawling*. Contoh pada tabel 2.

Tabel 1 *Dataset* hasil *crawling*

<i>username dan id user</i>	<i>id_tweet</i>	<i>isi tweet</i>	<i>tanggal dibuat</i>
_denBagus/295460652	1087682320209858560	b'@Ary0977 @jokowi @wiranto1947 Presiden Wiranto mengkritik petugas partai. Seumur hidup ya baru kali ini ada ndan_cuk di "langkah" anak buah. :)'	22-01-19 12:04

2. *Preprocessing data*

Pada tahap ini terdiri dari beberapa bagian, yakni terdapat *case fold*, *remove character*, *tokenizing* dan *stopword*. Berikut penjelasannya.

a. *Case fold*

Pada proses ini sistem akan merubah semua abjad yang kapital menjadi non-kapital, Contoh pada tabel 2 :

Tabel 2 *Case folding*

Input	Output
Karena elektabilitas 01 tetap nyungsep	karena elektabilitas 01 tetap nyungsep

b. *Remove character*

Pada proses ini karakter yang tidak digunakan atau berpengaruh pada pembangunan *classifier* nanti akan dihilangkan. Contoh pada tabel 3:

Tabel 3 *Remove character*

Input	Output
b'@Ferdinand_Haeen @jokowi Karena elektabilitas 01 tetap nyungsep...\nDipoles apapun masyrkt sdh tahu...blio hanya boneka...\n#2019GantiPresiden'	jokowi Karena elektabilitas 01 tetap nyungsep Dipoles apapun masyrkt sdh tahu blio hanya boneka #2019GantiPresiden

c. *Tokenizing*

Tokenisasi merupakan proses memecah suatu konten tekstual menjadi beberapa bagian semisal dipecah kedalam kategori kata, istilah, simbol atau elemen yang memiliki makna yang disebut token [13].

Pada proses ini sistem merubah susunan kalimat menjadi terpisah-pisah menjadi per bagian token atau per kata yang berdiri sendiri. Contoh pada tabel 4 :

Tabel 4 *Tokenisasi*

Input	Output
karena elektabilitas 01 tetap nyungsep	karena Elektabilitas 01 Tetap nyungsep

d. *Stopword removal*

Stop word removal adalah kata yang disaring sebelum maupun sesudah pemrosesan teks bahasa alami atau kata yang disaring bisa diartikan sebagai penghilangan kata yang tidak diperlukan dan hanya menyisakan kata yang penting [14].

Pada proses ini sistem akan menghilangkan kata yang tidak diperlukan seperti dan menyisakan kata penting. Contoh pada tabel 5 :

Tabel 5 *Stopword removal*

Input	Output
Karena elektabilitas 01 tetap nyungsep Dipoles apapun masyrkt sdh tahu blio hanya boneka .#2019GantiPresiden'	elektabilitas 01 nyungsep dipoles apa masyrkt sdh blio boneka #2019gantipresiden

e. *Stemming*

kata yang terdapat pada *tweet* memiliki banyak imbuhan awalan dan akhiran, oleh karena itu akan dilakukan pengubahan kata ke bentuk dasar dengan proses *stemming*, yang dimana dapat menghilangkan imbuhan awalan dan akhiran yang terdapat pada suatu kata yang ditunjukkan pada tabel 6.

Tabel 6 Stemming

Input	Output
Karena elektabilitas 01 tetap nyungsep Dipoles apapun masyrkt sdh tahu blio hanya boneka #2019GantiPresiden'	elektabilitas 01 nyungsep poles apa masyrkt sdh blio boneka #2019gantipresiden

3. Pembangunan model Classifier

Pada tahap ini pembangunan classifier akan dimulai dengan menghitung nilai *prior probability* dari masing-masing kelas, *Prior probability* didapatkan dari jumlah suatu kelas tertentu dibagi dengan total seluruh kelas. Berikut ditampilkan pada tabel 7.

Tabel 7 Contoh data latih dan data uji

	Doc	Tweet	Kelas
Data Latih	1	twips rezim doyan korupsi yaa	Prabowo
	2	karena elektabilitas 01 tetap nyungsep	Prabowo
	3	indonesia kembang #pilihnyangpastijokowi #01Indonesiamaju	Jokowi
Data Uji	4	Rezim doyan nyungsep gimana indonesia maju	?

Langkah pertama, menghitung *prior probability* dari setiap kelas yang ada yaitu jokowi dan prabowo.

$$P(\text{jokowi}) = \frac{1}{3} = 0,33$$

$$P(\text{prabowo}) = \frac{2}{3} = 0,67$$

Setelah mendapat nilai *prior probability* setiap kelas, selanjutnya menghitung *conditional probability*.

$$P(\text{rezim}|\text{jokowi}) = \frac{0+1}{4+14} = \frac{1}{18} = 0,05$$

$$P(\text{rezim}|\text{prabowo}) = \frac{1+1}{10+14} = \frac{2}{24} = 0,08$$

$$P(\text{doyan}|\text{jokowi}) = \frac{0+1}{4+14} = \frac{1}{18} = 0,05$$

$$P(\text{doyan}|\text{prabowo}) = \frac{1+1}{10+14} = \frac{2}{24} = 0,08$$

$$P(\text{nyungsep}|\text{jokowi}) = \frac{0+1}{4+14} = \frac{1}{18} = 0,05$$

$$P(\text{nyungsep}|\text{prabowo}) = \frac{1+1}{10+14} = \frac{2}{24} = 0,08$$

$$P(\text{gimana}|\text{jokowi}) = \frac{0+1}{4+14} = \frac{1}{18} = 0,05$$

$$P(\text{gimana}|\text{prabowo}) = \frac{0+1}{10+14} = \frac{1}{24} = 0,04$$

$$P(\text{indonesia}|\text{jokowi}) = \frac{1+1}{4+14} = \frac{2}{18} = 0,11$$

$$P(\text{indonesia}|\text{prabowo}) = \frac{0+1}{10+14} = \frac{1}{24} = 0,04$$

$$P(\text{maju}|\text{jokowi}) = \frac{0+1}{4+14} = \frac{1}{18} = 0,05$$

$$P(\text{maju}|\text{prabowo}) = \frac{0+1}{10+14} = \frac{1}{24} = 0,04$$

Setelah mendapatkan nilai *conditional probability* selanjutnya dilakukan perhitungan untuk menentukan *tweet* tersebut masuk kedalam kelas jokowi atau prabowo. Berikut perhitungan dari *tweet* untuk penentuan kelas.

$$P(\text{jokowi}|\text{Doc4}) = 0,33 \times 0,05 \times 0,05 \times 0,05 \times 0,05 \times 0,11 \times 0,05 = 1,134 \times 10^{-8}$$

$$P(\text{prabowo}|\text{Doc4}) = 0,67 \times 0,08 \times 0,08 \times 0,08 \times 0,04 \times 0,04 \times 0,04 = 2,195 \times 10^{-8}$$

Setelah diperoleh hasil probabilitas dari *tweet* terhadap setiap kelas, maka hasil probabilitas dibandingkan dari kedua kelasnya, dilihat dari hasil diatas nilai probabilitas kelas prabowo lebih besar dibandingkan jokowi sehingga Doc 4 masuk ke dalam kelas "**prabowo**".

4. Evaluasi

4.1 Hasil Pengujian

Pada hasil pengujian dari penelitian ini dengan menggunakan *dataset* sejumlah 1370 *tweet* yang menjadi data latih sekaligus data uji secara bergantian dengan proses *K-fold cross validation* dengan jumlah iterasi atau $K=10$ yang berarti 90% dataset digunakan untuk proses *training* dan 10% sisanya akan digunakan sebagai *testing*. Data terbagi menjadi 10 iterasi saat melakukan *cross validation* dengan isi masing-masing setiap iterasi sejumlah 137 data *tweet* yang digunakan sebagai *testing* dan sisanya sebagai *training*. Hasil dari 10 iterasi pengujian akan menampilkan *precision*, *recall*, *f1-score*, serta *accuracy*, dari hasil pengujian akan diambil nilai rata-rata *f1-score* karena juga merupakan rata-rata harmonik dari *precision* dan *recall*, selain itu juga akan dihitung rata-rata *accuracy*-nya. Hasil dari pengujian diperlihatkan pada tabel 8.

Tabel 8 Hasil pengujian

<i>Fold</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>	<i>Accuracy</i>
1	0.70903	0.741094	0.707659	0.722628
2	0.770896	0.792069	0.768795	0.773723
3	0.717849	0.757064	0.724704	0.751825
4	0.682479	0.684848	0.678404	0.678832
5	0.707419	0.719904	0.691441	0.693431
6	0.678171	0.676496	0.676748	0.678832
7	0.757491	0.761408	0.751494	0.751825
8	0.710716	0.706657	0.700155	0.70073
9	0.724367	0.722933	0.722258	0.722628
10	0.716977	0.723046	0.719588	0.742647
Rata-rata			0.714125	0.72171

4.2 Analisis Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan, hasil rata-rata *F1-Score* dan Akurasi dengan penggunaan *stemming* dan tanpa penggunaan *stemming* adalah 0,71 dan 0,72. Adapun beberapa hal yang dapat mempengaruhi hasil dari pengujian yang dilakukan yaitu :

1. *Preprocessing*
2. *Data training* yang digunakan baik dari segi jumlah maupun isinya

Preprocessing memiliki peran penting karena ketika pengumpulan data, banyak *tweet* yang isinya tidak rapi dan tidak beraturan dengan banyaknya kata yang disingkat, tanda baca, *link*, dan *emoticon*.

Penggunaan data *tarining* juga memiliki pengaruh dan peran yang penting, karena semakin banyak jumlah data *training* yang digunakan maka semakin tinggi akurasi yang diperoleh karena banyak kata yang dapat menggambarkan dan memiliki korelasi dengan kelasnya atau malah bisa jadi akurasi semakin menurun karena banyak data yang tidak jelas dan tidak memiliki korelasi yang jelas. Pertama peneliti menggunakan beberapa fitur *preprocessing* yang menjadi *baseline* atau yang bisa disebut sebagai informasi dasar yang digunakan sebagai pembandingan untuk memperlihatkan dampak program yang dijalankan, adapun *baseline*-nya yaitu *case fold*, *tokenization*, dan *stopword removal*. Dari penggunaan *baseline* tersebut diperoleh rata-rata *f1-score* dan akurasi yang ditampilkan pada tabel 9.

Tabel 9 *Baseline* Analisis

No	Penggunaan Data	Rata-rata <i>f1-score</i>	Rata-rata akurasi
1	<i>case fold+tokenization + stopwords removal</i>	0,61	0.62

Setelah diperoleh nilai *baselien* hasil pengujian yang ditampilkan pada tabel 9, maka selanjutnya ditambahkan beberapa skenario pengujian yang membuat hasil dari rata-rata *f1-score* dan akurasi berbeda-beda yang ditampilkan pada tabel 10.

Tabel 10 Hasil Analisis Pengujian

No	Penggunaan Data	Selisih	Rata-rata <i>f1-score</i>	Rata-rata Akurasi
1	<i>Data Cleaning</i>	- 0,01	0.60	0,61
2	<i>Data Cleaning +Penggunaan Hashtag</i>	+ 0,1	0.70	0,71
3	<i>Data Cleaning + Penggunaan Hashtag+Stemming</i>	+ 0,01	0.71	0,72

Dari tabel 10 memperlihatkan pembersihan data dari karakter-karakter yang tidak diperlukan memang membuat data menjadi lebih mudah ketika diproses dalam pembangunan model klasifikasi, namun ada bagian yang hilang ketika proses pembersihan data tersebut yang menjadi penyebab turunnya nilai rata-rata *f1-score* dan akurasinya.

Selanjutnya disertakanlah penggunaan *hashtag* yang sebelumnya hilang pada saat pembersihan data, sehingga pada saat pembersihan data yang dapat meningkatkan akurasi klasifikasinya yaitu dengan catatan menyertakan *hashtag* dari setiap *tweet*-nya. Begitu juga pada bagian *preprocessing*, fitur *stemming* pada *preprocessing* sedikit mampu meningkatkan akurasi dari hasil klasifikasi.

Selain itu penyebab turunnya nilai akurasi bisa disebabkan karena banyak kata yang seharusnya masuk kedalam kategori *stopword removal* namun karena pada isi *tweet* disingkat sehingga tidak terhapus dengan baik dan menjadikan nilai pengali pada *likelihood* semakin kecil yang membuatnya semakin jauh tidak berkorelasi dengan kelasnya, berikut contohnya : kata "tetap" disingkat menjadi "ttp", "yang" disingkat menjadi "yg", "saya" menjadi "sy" dan masih banyak lagi kata-kata tidak penting yang perlu dihilangkan dari *dataset*. Juga terdapat kata yang seharusnya masuk kedalam satu jenis fitur yang sama yang sama seperti kata "masyarakat" dengan "msyrkt" namun karena perbedaan dalam penulisan menyebabkan kata tersebut dihitung secara terpisah, begitu juga persebaran kelas pada data *training* dapat mempengaruhi performansi dari MNB.

5. Kesimpulan

Berdasarkan hasil pengujian serta analisis sebelumnya, maka diperoleh kesimpulan sebagai berikut :

1. Metode *Multinomial Naive Bayes Classifier* mampu mengklasifikasikan keberpihakan *tweet* pada pilpres 2019 dengan rata-rata nilai *F1-Score* 0,71 dan dengan *accuracy* 0,72.

Adapun saran yang ingin disampaikan guna mengembangkan Tugas Akhir ini kedepannya adalah :

1. Penambahan kamus kata untuk *slang word* dan kata singkatan yang lebih banyak pada *corpora* untuk menangani *slang word* dan kata singkatan yang sering beredar di media maya khususnya pada media sosial twitter yang memiliki aturan karakter terbatas untuk satu *tweet*.

Daftar Pustaka

- [1] R. Indonesia, "Undang-undang pasal 22E tentang Pemilihan Umum," in *Lembaran Negara RI*, Jakarta, Sekretariat Negara, 2001.
- [2] B. Rahardjo, *Twitter Baik dan Benar*, Bandung: PT Insan Infonesia, 2013.
- [3] A. Tumasjan, T. O. Sprenger, P. G. Sandner and I. M. Welp, "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment," *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, p. 178, 2010.
- [4] L. Bode, "Facebooking It to the Polls: A Study in Online Social Networking and Political Behavior," *Journal of Information Technology & Politics*, vol. 9, no. 4, pp. 352-369, 2012.
- [5] B. Pang, L. Lee dan S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," dalam *Empirical Methods on Natural Language Processing*, New York, 2002.
- [6] P. Bhumika Gupta, M. Negi, K. Vishwakarma, G. Rawat and P. Badhani, "Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python," *International Journal of Computer Applications*, vol. 165, no. 9, p. 0975 – 8887, 2017.
- [7] P. A. Agustina, T. Matulatan, M. Tech and M. Martaleli Bettiza S.Si, "KLASIFIKASI TRENDING TOPIC TWITTER DENGAN PENERAPAN METODE NAÏVE BAYES".
- [8] C. C. Aggarwal, *Data Mining*, New York: Springer International Publishing, 2015.
- [9] C. D. Manning, P. Raghavan and H. Schütze, *An Introduction to Information Retrieval*, Cambridge: Cambridge University Press, 2009.
- [10] I. Witten, E. Frank and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques 3rd Edition*, USA: Elsevier, 2011.
- [11] M. W. Berry and J. Kogan, *TEXT MINING: APPLICATIONS AND THEORY*, United Kingdom: John Wiley & Sons, 2010.
- [12] E. Prasetyo, *Data Mining: Konsep dan Aplikasi menggunakan MATLAB*, Yogyakarta: ANDI, 2012.
- [13] D. S. Vijayarani and M. R. Janani, "TEXT MINING: OPEN SOURCE TOKENIZATION," *Advanced Computational Intelligence: An International Journal (ACII)*, vol. 3, 2016.
- [14] S. Jusoh and H. M. Alfawareh2, "Techniques, Applications and Challenging Issue in Text Mining," *International Journal of Computer Science Issues*, vol. 9, no. 6 no 12, November -2012.