

Klasifikasi Spam Tweet Pada Twitter Menggunakan Metode Naïve Bayes (Studi Kasus: Pemilihan Presiden 2019)

Muhammad Hanafiah¹, Anisa Herdiani, S.T., M.T.², Widi Astuti, S.T., M.Kom.³

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

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

³widiwdu@telkomuniversity.ac.id

Abstrak

Pemilu merupakan wujud dari kebebasan berpendapat dan kebebasan berserikat. Sedangkan arti dari pemilu itu sendiri adalah proses pemilihan orang untuk mengisi jabatan-jabatan politik tertentu. Pada tahun 2019 Indonesia telah menyelenggarakan pesta demokrasi 5 tahunan yang telah dilaksanakan pada tanggal 17 april 2019. Pembahasan tentang pilpres 2019 begitu ramai diperbincangkan termasuk di media sosial *twitter*, *Twitter* telah memperkenalkan fitur *Trending Topics* sejak tahun 2018. Fitur ini memudahkan pengguna untuk mengetahui berita atau informasi terbaru di seluruh dunia secara real time. Bagi pengguna yang ingin melakukan promosi produk tentunya hal ini akan sangat bermanfaat bagi mereka, apalagi disertai dengan *hashtag* yang berkaitan dengan pilpres 2019 yang sedang ramai diperbincangkan. Dengan cara ini, mereka dapat mempromosikan konten dagangan mereka ke calon pembeli dengan lebih cepat. Hal ini menyebabkan para pengguna biasa membuang banyak waktu dalam memilih dan memilah informasi atau berita yang mereka inginkan karena konten yang tidak perlu muncul secara terus menerus atau dapat juga disebut dengan *spam*. Dalam menghadapi permasalahan tersebut, maka dari itu pada tugas akhir telah dibangun suatu sistem menggunakan metode *Naive Bayes* untuk mengklasifikasi sekumpulan tweet kedalam dua kelas yaitu *spam* dan *non spam*. Sebelum memasuki proses pengklasifikasian, sistem melakukan *preprocessing* terhadap *dataset* yang digunakan, hal tersebut bertujuan untuk meningkatkan performansi dari ke dua metode yang digunakan. Dalam penelitian ini *Naive Bayes* dengan *preprocessing* dan *Naive Bayes* tanpa *preprocessing* masing-masing menghasilkan nilai akurasi 76,34% dan 74,14%.

Kata Kunci: *Pemilu, Twitter, Tweet, Spam, Naive Bayes.*

Abstract

Elections are a manifestation from freedom of opinion and freedom of association. Whereas the meaning of the election itself is the process of selecting people to fulfill certain political positions. In 2019 Indonesia has held a 5 years democratic party which will be held on 17 April 2019. Discussions about the 2019 presidential election are so widely discussed including on social media twitter, Twitter has introduced the Trending Topics feature since 2018. This feature makes it easy for users to know the news or the latest information around the world in real time. For users who want to carry out product promotions surely this will be very beneficial for them, especially accompanied by the 2019 replace the president election hashtag which is widely being discussed. In this way, they can promote their merchandise to potential buyers faster. This causes ordinary users to waste a lot of time in choosing and sorting information or news they want due to content that does not need to appear continuously or can also be called spam. In dealing with these problems, therefore in this final project will be built a system using the Naive Bayes method to classify a set of tweets into two classes namely spam and non spam. Before entering the classification process, the system preprocesses the data set used, it aims to improve performance from the two methods used. In this study Naive Bayes with preprocessing and Naive Bayes without preprocessing each produced an accuracy value of 76,34% and 74,14%.

Keywords: *Election, Twitter, Tweet, Spam, Naive Bayes*

1. Pendahuluan

1.1. Latar Belakang

Pergantian presiden di Indonesia sampai saat ini sudah enam kali terjadi mulai dari Presiden pertama Soekarno sampai yang ketujuh saat ini Joko Widodo. Pada tahun 2019 Indonesia telah menyelenggarakan pesta demokrasi 5 tahunan yang telah dilaksanakan pada tanggal 17 april 2019, tentunya hal ini menuai opini masyarakat tentang siapa yang lebih pantas untuk menjabat sebagai presiden. Pembahasan 2019 ganti presiden dan tetap jokowi begitu ramai diperbincangkan termasuk di media sosial *twitter*, banyak yang mendukung jokowi untuk menjabat 2 periode tapi banyak pula yang mendukung prabowo untuk menjadi presiden indonesia

yang ke-8.

Twitter telah memperkenalkan fitur *Trending Topics* sejak tahun 2018. Fitur ini memudahkan pengguna untuk mengetahui berita atau informasi terbaru di seluruh dunia secara *real time*. *Twitter* sekarang telah menjadi salah satu sumber konten tercepat, dikarenakan berita dan informasi terbaru akan muncul terlebih dahulu di *Twitter* sebelum muncul di media sosial lain [1]. Bagi pengguna yang ingin melakukan promosi produk tentunya hal ini akan sangat bermanfaat bagi mereka [2], apalagi dibarengi dengan *hashtag* yang berkaitan dengan pilpres 2019 yang sedang ramai diperbincangkan. Dengan cara ini, mereka dapat mempromosikan konten dagangan mereka ke calon pembeli dengan lebih cepat. Hal ini menyebabkan para pengguna biasa membuang banyak waktu dalam memilih dan memilah informasi atau berita yang mereka inginkan. Konten yang tidak diperlukan namun secara terus menerus muncul dan memberikan dampak negatif ke pengguna secara umum dapat dikategorikan sebagai *spam* [3]. Maka dari itu dibutuhkan suatu sistem yang dapat mengklasifikasikan *tweet* tersebut ke dalam 2 kategori yakni *spam* dan *non spam*.

Faisal Aji Prayoga telah melakukan klasifikasi *tweet spam* dengan metode *Naïve Bayes* dan *K-Nearest Neighbor*(KNN) dan menghasilkan akurasi yang cukup baik yaitu 82% dan 71% [4], *Naïve Bayes* memiliki akurasi dan kecepatan pengklasifikasian yang lebih baik dibandingkan *K-Nearest Neighbor* [4], namun pada penelitian tersebut tidak dilakukan *preprocessing* terhadap *dataset* yang digunakan. Menurut Datta H.Deshmukh, Tushar Ghorpade dan Puja Padiya pada penelitiannya, *preprocessing* yang dilakukan sebelum pengklasifikasian dapat meningkatkan akurasi *classifier*, dimana hal tersebut telah diujikan terhadap *classifier Naïve Bayes*, *Hidden Naïve Bayes* dan *NBTree* [5].

Pada penelitian ini metode *Naïve Bayes* diujikan dengan melakukan *preprocessing* terhadap *dataset* untuk meningkatkan performansi metode *Naïve Bayes* dalam melakukan klasifikasi data yang berupa *tweet* berbahasa Indonesia.

1.2. Topik dan Batasan

Berdasarkan latar belakang yang sudah dijelaskan sebelumnya topik yang dibahas pada penelitian ini yaitu bagaimana perbedaan nilai akurasi yang didapat dari klasifikasi *tweet spam* dan *non spam* menggunakan metode *Naïve Bayes* dengan *preprocessing* dan tanpa *preprocessing*?

Adapun batasan masalah pada penelitian ini adalah sebagai berikut.

- a. Data set yang digunakan berasal dari *tweet* media sosial *Twitter*.
- b. Data yang digunakan hanya *tweet* yang berbahasa Indonesia.
- c. Tidak mengidentifikasi simbol, *link*, gambar, *emoticon*, gambar maupun video.
- d. Pelabelan pada data set hanya *spam* dan *non spam*.

1.3. Tujuan

Penelitian ini bertujuan mengimplementasikan metode *Naïve Bayes Classifier* dalam klasifikasi *spam tweet* berbahasa indonesia tentang pilpres 2019 dan menganalisis nilai akurasi yang didapatkan dengan menggunakan metode *Naïve Bayes Classifier* tanpa *preprocessing* dan *Naïve Bayes Classifier* dengan *preprocessing*, karna dengan melakukan *preprocessing* sebelum pengklasifikasian dapat meningkat akurasi *classifier* [5].

1.4. Sistematika Penulisan

Urutan penulisan dalam penelitian ini adalah sebagai berikut : bagian 2 menunjukkan studi literatur mengenai teori yang digunakan, bagian 3 menjelaskan sistem yang akan diajukan yaitu klasifikasi *spam tweet* pada *twitter* menggunakan metode *Naïve Bayes*. Bagian 4 akan menjelaskan mengenai hasil pengujian dan evaluasi sistem. Terakhir bagian 5 akan menjelaskan kesimpulan dan saran.

2. Studi Terkait

2.1 Twitter

Twitter adalah sebuah layanan jejaring *social* yang didirikan pada tahun 2006 oleh Jack Dorsey, disebut juga *microbloging* karena user hanya dapat memposting sebuah pesan teks dengan panjang maksimal adalah sebanyak 140 karakter. *Twitter* pertama kali digunakan sebagai layanan internal bagi karyawan Odeo, dan versi lengkapnya diperkenalkan kepada publik pada tanggal 15 Juli 2006 [6].

Post user disebut juga dengan *tweet*, dimana *tweet* tersebut akan ditampilkan dalam *timeline*. *Twitter* memiliki sistem *follow* dan *follower*. *Follow* berfungsi untuk mengikuti sebuah akun, sehingga posting dari akun tersebut dapat muncul di *timeline* bagi yang *follow* akun tersebut tanpa harus menunggu konfirmasi dari *user* yang di *follow*, kecuali status user tersebut *private*. Sedangkan *follower* adalah akun *twitter user* lain yang dapat mengikuti sebuah *update posting* dari akun yang di *follow*. Untuk melakukan komunikasi dengan *user* lain, *user* tinggal melakukan *mention username* yang diinginkan. *Twitter* juga menyediakan fitur *hashtag* bagi *user* untuk mengetahui hal paling populer yang sedang dibicarakan banyak orang, atau juga disebut dengan *trending topic*.

2.2 Spam

Spam adalah suatu bentuk kegiatan penggunaan perangkat elektronik untuk mengirimkan pesan secara terus-menerus tanpa diinginkan oleh penerimanya. Orang yang melakukan *spam* disebut *spammer*. Kegiatan *spam* dikenal dengan sebutan *spamming* [7]. Adanya *spam* dapat memberikan kerugian bagi pengguna biasa. Disamping *spam* mengganggu, *spam* juga akan memakan *storage* pada mail *server* dan menyumbat Internet dengan sampah [3]. Lain halnya dengan *spammer*, *spam* justru memberikan keuntungan bagi mereka. M. Mangalindan dari *The 7 Wall Street Journal Europe*, mengklaim bahwa jika *spammer* mendapat sedikitnya 100 respon untuk setiap 10.000.000 email, mereka masih bisa mendapat keuntungan. Media sosial seperti Facebook dan Twitter tak dapat lepas dari *spam*. Pada umumnya kegiatan *spamming* dilakukan dengan cara membuat akun palsu untuk mendapatkan informasi pribadi dari akun yang ditargetkan atau mempromosikan iklan komersial untuk mendapatkan keuntungan pribadi [2].

2.3 Preprocessing

Preprocessing merupakan langkah yang dilakukan setelah membaca *text* dan sebelum dilakukan pembobotan. Tujuan dilakukan teks *preprocessing* adalah meningkatkan kualitas *term* dan *frase* untuk digunakan dalam pembobotan dan mengurangi dimensi fitur dalam proses penambangan [8]. Langkah-langkah *preprocessing* yang dilakukan berupa:

1. *Case folding* adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil.
2. *Tokenizing* adalah tahap pemecahan kalimat berdasarkan tiap kata yang menyusunnya.
3. *Stopword removal* adalah tahap menghapus kata-kata yang berupa *stopword* seperti ini, itu, di, dan sebagainya.
4. *Stemming* adalah proses yang mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan-imbuhan pada kata dalam dokumen atau mengubah kata kerja menjadi kata benda [9].

2.4 Naïve Bayes

Naïve Bayes Classifier adalah konsep probabilitas penentuan kelompok. Algoritma klasifikasi ini dapat mengolah data dalam jumlah besar dengan hasil akurasi yang tinggi [10]. Dalam metode *Naïve Bayes Classifier* dilakukan proses pengklasifikasian teks berdasarkan data latih yang sebelumnya sudah disimpan. Proses yang terdapat pada algoritma ini ada dua tahap yaitu proses pelatihan (*training*) dan proses klasifikasi. Untuk persamaan umum metode *Naïve Bayes* dalam persamaan 1 [10].

$$P(C_j | W_i) = \frac{P(C_j) \times P(W_i | C_j)}{P(W_i)} \quad (2.1)$$

Keterangan :

$P(C_j | W_i)$: *posterior*, Peluang kategori j ketika terdapat kemunculan kata i

$P(C_j)$: *prior*, Peluang kemunculan sebuah kategori j

$P(W_i | C_j)$: *likelihood*, Peluang sebuah kata i masuk ke dalam kategori j

$P(W_i)$: *evidence*, Peluang kemunculan sebuah kata

i : indeks kategori yang berawal dari 1 sampai dengan kata ke-k

j : indeks kategori yang berawal dari 1 sampai dengan kategori ke-n

Persamaan untuk menghitung peluang kemunculan dokumen ada kategori tertentu (*prior*), ditunjukkan dalam persamaan 2 [10].

$$P(C_j) = \frac{NC_j}{N} \quad (2.2)$$

Keterangan :

NC_j : Banyak dokumen berkategori cj pada dokumen latih

N : Jumlah keseluruhan dokumen latih yang digunakan.

2.5 K-Fold Cross Validation

K-Fold cross validation adalah salah satu metode yang digunakan untuk mencari tahu rata-rata keberhasilan dari suatu sistem yang dibangun dengan cara melakukan *looping* dengan mengacak atribut masukan sehingga suatu sistem teruji untuk beberapa atribut input yang acak. *K-Fold cross validation* dimulai dengan membagi data sejumlah *nfold* yang diinginkan. Dalam proses *cross validation* nantinya data akan dibagi dalam n buah partisi yang memiliki ukuran yang sama D1, D2, D3...Dn. Selanjutnya proses *testing* dan *training* dilakukan sebanyak n kali. Dalam iterasi ke-i partisi Di akan menjadi data *testing* dan selebihnya akan menjadi data

training. Untuk penggunaan jumlah *fold* terbaik untuk uji validitas, disarankan menggunakan *10-Fold Cross Validation* dalam model [11]. Skenario pengujian merupakan tahap penentuan pengujian yang akan dilakukan. Pengujian akan dilakukan menggunakan metode *k-cross validation* dengan nilai *k* sebanyak *10 Fold*, pengujian ini memiliki tujuan untuk mendapatkan akurasi metode *Naive Bayes* yang diimplementasikan pada analisis *spam* jika diuji dengan data *training* dan data *testing* yang berbeda. Penggunaan *10 Fold* ini dianjurkan karena merupakan jumlah *Fold* terbaik untuk uji validitas [11].

2.6 Evaluasi Klasifikasi

Pengujian pada sistem yang akan dibangun dilakukan dengan melakukan perhitungan untuk mendapatkan akurasi dari performansi klasifikasi sistem. Pada penelitian ini nantinya perhitungan performansi lebih difokuskan untuk kelas negatif, karena penelitian ini memiliki tujuan untuk mencari ketepatan sistem dalam mendeteksi *spam tweet*. Dalam proses evaluasi klasifikasi, terdapat beberapa komponen yang akan dilakukan perhitungan [12], yaitu sebagai berikut.

1. *True Positive* (TP) adalah kondisi saat suatu kelas *true* dan berhasil diklasifikasikan sebagai kelas *true*.
2. *True Negative* (TN) adalah kondisi saat suatu kelas *false* dan berhasil diklasifikasikan sebagai kelas *false*.
3. *False Negative* (FN) adalah merupakan kondisi saat suatu kelas *true* diklasifikasikan sebagai kelas *false*.
4. *False Positive* (FP) adalah kondisi saat suatu kelas *false* diklasifikasikan sebagai kelas *true*.

Berikut keempat komponen jika digambarkan dalam tabel, dapat dilihat pada tabel 2.1.

Tabel 2.1 Evaluasi Klasifikasi [12]

| | Kelas Klasifikasi (True) | Kelas Klasifikasi (False) |
|-------------------------|--------------------------|---------------------------|
| Kelas Sebenarnya(True) | TP | FN |
| Kelas Sebenarnya(False) | FP | TN |

Empat parameter tersebut akan digunakan untuk perhitungan *precision* dan *recall*. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Perhitungan *precision* dapat dilihat pada persamaan 2.3 [12].

$$precision = \frac{TP}{TP + FP} \quad (2.3)$$

Recall merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Perhitungan *recall* dapat dilihat pada persamaan 2.4 [12].

$$recall = \frac{TP}{TP + FN} \quad (2.4)$$

F-Measure adalah hasil rata-rata antara nilai *precision* dan *recall*. *F-measure* merupakan pengukuran performansi sistem yang menggabungkan nilai *precision* dan *recall* [12]. Perhitungan *F-measure* dilihat pada persamaan 2.5.

$$Fmeasure = \frac{2 \times precision \times recall}{precision + recall} \quad (2.5)$$

Accuracy merupakan perhitungan data yang bernilai *true* dibanding dengan keseluruhan jumlah data [12]. Nilai *accuracy* didefinisikan sebagai tingkat kedekatan antara nilai aktual dan prediksi dari sistem. Perhitungan Rumus *Accuracy* dapat dilihat pada persamaan 2.6.

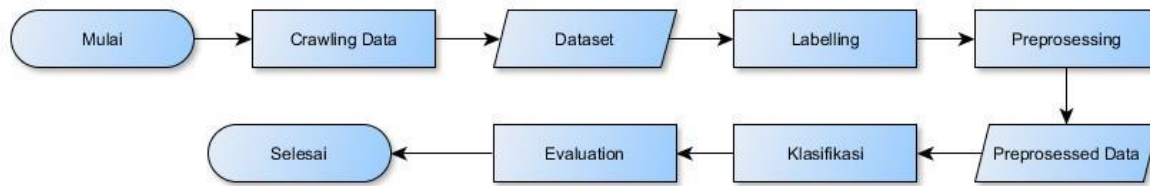
$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (2.6)$$

3. Metodologi

Pada bab ini akan dijelaskan bagaimana perancangan sistem yang akan dibangun.

3.1 Tahapan Perancangan Sistem

Dalam penelitian ini akan dibangun sistem yang menggunakan metode *Naive Bayes classifier* untuk mengklasifikasi *tweet* ke dalam kelas *spam* atau *non-spam*. Perancangannya dapat dilihat pada gambar 3.1:



Gambar 3.1 Rancangan Sistem

Tabel 3.1 Deskripsi Proses Naive Bayes

| No | Tahap | Input | Proses | Output |
|----|----------------------|--------------------------------------|---|--------------------------|
| 1 | <i>Crawling data</i> | <i>Keyword</i> | Pengambilan data dilakukan dengan menggunakan <i>tweepy</i> untuk mengakses <i>Twitter API</i> . | <i>Dataset</i> |
| 2 | <i>Labelling</i> | <i>Data Set</i> | Melakukan pelabelan untuk setiap <i>tweet</i> , apakah <i>tweet</i> tersebut termasuk ke kelas <i>spam</i> atau <i>non spam</i> | <i>Data Training</i> |
| 3 | <i>Preprocessing</i> | <i>Data training</i> | Membersihkan dataset yang telah diambil agar sistem dapat lebih mudah dalam melakukan kumputerisasi data teks. Proses ini terdiri dari <i>case folding</i> , <i>tokenizing</i> , <i>stopword removal</i> , dan <i>stemming</i> . | <i>Preprocessed Data</i> |
| 4 | Klasifikasi | <i>Preprocessed Data(Data Train)</i> | Melakukan klasifikasi berdasarkan metode <i>Naïve Bayes</i> . | Hasil klasifikasi |
| 5 | <i>Evaluating</i> | Hasil klasifikasi | Mengevaluasi sistem dengan menghitung nilai <i>precision</i> , <i>recall</i> , <i>f-measure</i> dan akurasi. <i>Precision</i> digunakan untuk mengetahui performansi dari sistem yang dihasilkan. <i>Recall</i> digunakan untuk mengetahui tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. <i>F-measure</i> digunakan untuk mendapatkan nilai rata-rata dari <i>precision</i> dan <i>recall</i> , nilai ini sangat dibutuhkan untuk perbedaan nilai <i>precision</i> dan <i>recall</i> yang cukup signifikan. Akurasi digunakan untuk menilai tingkat kedekatan antara nilai aktual dan prediksi dari sistem. | Nilai evaluasi sistem |

3.2 Pengumpulan Data

Dataset yang digunakan dalam pengerjaan tugas akhir ini adalah dokumen berbahasa Indonesia. Data diambil dari *Twitter* dengan cara mengakses *Twitter API* dengan memanfaatkan *library* dari *python* yaitu *tweepy*. Jumlah total dataset yang digunakan adalah 1000 *tweet* yang memiliki *hashtag* yang berkaitan dengan pilpres 2019, dimana *hashtag* tersebut ialah #pilpres2019, #2019gantipresiden, #2019tetapjokowi, #2019prabowosandi, #2019prabowosandimenang, #2019jokowilanjut, #2019jokowidodopresidenri, #2019jokowimarufamin dan #2019jokowimaruf. *Tweet* yang termasuk *spam* adalah *tweet* yang tidak berkaitan atau tidak ada hubungannya dengan pilpres 2019. Untuk pelabelan data dilakukan oleh 3 orang, kemudian setiap *tweet* ditarik kesimpulan apakah *tweet* tersebut termasuk *spam* atau *non spam* berdasarkan jumlah *vote* terbanyak. Dari 1000 data latih diperoleh jumlah *tweet* untuk kelas *non spam* sebanyak 622 dan untuk *spam* sebanyak 378.

3.3 Fungsionalitas Sistem

3.3.1. Preprocessing

Pada tahap ini, data dibersihkan dengan cara *case folding*, *tokenizing*, *stopword removal*, dan *stemming*. Tahap *preprocessing* ini dilakukan karna data yang diambil memiliki kemungkinan huruf besar dan mengandung kata-kata *stopword*.. Berikut adalah gambaran dari proses preprocessing pada sistem penelitian ini, dapat dilihat pada gambar 3.2:



Gambar 3.2 Tahapan Preprocessing

Berikut penjelasan dari masing-masing tahapan *preprocessing*:

3.3.1.1. Case Folding

Dalam tahap ini dilakukan mengubah semua huruf dalam dokumen menjadi huruf kecil, karna data *tweet* yang diambil dari *twitter* memungkinkan memiliki huruf besar, dengan mengubah data *tweet* menjadi huruf kecil dapat mempermudah dalam proses pengklasifikasian. Contoh prosesnya ditunjukkan di Tabel 3.2

Tabel 3.2 Contoh Case Folding

| Sebelum | Setelah |
|--|--|
| Presiden indonesia ini ga tau isi surat yang ia tandatangani sungguh memalukan #2019GantiPresiden Jokowi akui tak tahu isi Perpres kenaikan uang mobil pejabat. | presiden indonesia ini ga tau isi surat yang ia tandatangani sungguh memalukan #2019GantiPresiden jokowi akui tak tahu isi perpres kenaikan uang mobil pejabat. |

3.3.1.2. Tokenizing

Dalam tahap ini dilakukan pemecahan kalimat berdasarkan tiap kata yang menyusunnya, dengan adanya tahap ini mempermudah dalam melakukan *preprocessing* selanjutnya. Contoh prosesnya ditunjukkan di Tabel 3.3.

Tabel 3.3 Contoh Tokenizing

| Sebelum | Setelah |
|--|---|
| presiden indonesia ini ga tau isi surat yang ia tandatangani sungguh memalukan #2019GantiPresiden jokowi akui tak tahu isi perpres kenaikan uang mobil pejabat. | presiden, indonesia, ini, ga, tau, isi, surat, yang, ia, tandatangani, sungguh, memalukan, #2019GantiPresiden, jokowi, akui, tak, tahu, isi, perpres, kenaikan, uang, mobil, pejabat. |

3.3.1.3. Stopword Removal

Dalam tahap ini dilakukan penghapusan kata-kata yang berupa *stopword* seperti ini, itu, di, dan sebagainya. Kata-kata yang termasuk *stopword* nantinya disimpan dalam sebuah file kemudian diakses untuk penghapusan data *tweet* yang mengandung kata *stopword*, karna kata *stopword* tidak dibutuhkan dalam proses pengklasifikasian. Contoh prosesnya ditunjukkan di Tabel 3.4.

Tabel 3.4 Contoh Stopword Removal

| Sebelum | Setelah |
|---|---|
| presiden, indonesia, ini, ga, tau, isi, surat, yang, ia, tandatangani, sungguh, memalukan, #2019GantiPresiden, jokowi, akui, tak, tahu, isi, perpres, kenaikan, uang, mobil, pejabat. | presiden, indonesia, tau, isi, surat, tandatangani, sungguh, memalukan, #2019GantiPresiden, jokowi, akui, tahu, isi, perpres, kenaikan, uang, mobil, pejabat. |

3.3.1.4. Stemming

Dalam tahap ini dilakukan dilakukan *stemming* yang mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan-imbuhan pada kata dalam dokumen atau mengubah kata kerja menjadi kata benda di Tabel 3.5.

Tabel 3.5 Contoh stemming

| Sebelum | Setelah |
|---|--|
| presiden, indonesia, tau, isi, surat, tandatangani, sungguh, memalukan, #2019GantiPresiden, jokowi, akui, tahu, isi, perpres, kenaikan, uang, mobil, pejabat. | presiden, indonesia, tau, isi, surat, tandatangani, sungguh, malu, #2019GantiPresiden, jokowi, akui, tahu, isi, perpres, naik, uang, mobil, pejabat. |

3.4. Naïve Bayes

Pada implementasi *Naïve Bayes Classifier* menggunakan data set hasil *preprocessing*. *Naïve Bayes Classifier* menggunakan prior probability yaitu nilai probabilitas yang diyakini benar sebelum melakukan pengujian. Prior probability diambil dari jumlah kata yang memiliki kelas *spam* atau *non-spam* kemudian dibagi dengan jumlah seluruh kata. Berikut adalah contoh perhitungan probabilitas prior yang data di ambil dari Tabel 3.7.

Tabel 3.7. Contoh Data Latih dan Data Uji

| | Doc. | Tweet | Kelas |
|------------|------|------------------------------------|----------|
| Data Latih | 1 | presiden jokowi jadwal hadir acara | Non spam |
| | 2 | kubu prabowo kampanye jakarta | Non spam |
| | 3 | mari cek diskon murah | Spam |
| Data Uji | 4 | pesta rakyat diskon baju murah | ? |

Berikut adalah contoh probabilitas *prior* pada kelas *non spam* dan kelas *spam* yang dilihat pada Tabel 4.

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

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

Setelah mendapatkan nilai *prior probability*, lalu menghitung nilai *conditional probability* dengan cara jumlah kata kemunculan di kelas *spam* atau *non-spam* kata ditambah satu dibagi dengan jumlah seluruh kata pada kelas *spam* atau *non-spam* ditambah dengan jumlah seluruh kata. Berikut adalah contoh *conditional probability* pada setiap kata di satu *tweet* dan *tweet* dilihat pada Tabel 4.

$$P(\text{pesta}|\text{non spam}) = \frac{0+1}{9+13} = \frac{1}{22} = 0,04$$

$$P(\text{pesta}|\text{spam}) = \frac{0+1}{4+13} = \frac{1}{17} = 0,05$$

$$P(\text{rakyat}|\text{non spam}) = \frac{0+1}{9+13} = \frac{1}{22} = 0,04$$

$$P(\text{rakyat}|\text{spam}) = \frac{0+1}{4+13} = \frac{1}{17} = 0,05$$

$$P(\text{diskon}|\text{non spam}) = \frac{1+1}{9+13} = \frac{2}{22} = 0,09$$

$$P(\text{diskon}|\text{spam}) = \frac{1+1}{4+13} = \frac{2}{17} = 0,11$$

$$P(\text{baju}|\text{non spam}) = \frac{1+1}{9+13} = \frac{2}{22} = 0,09$$

$$P(\text{baju}|\text{spam}) = \frac{0+1}{4+13} = \frac{1}{17} = 0,05$$

$$P(\text{murah}|\text{non spam}) = \frac{0+1}{9+13} = \frac{1}{22} = 0,04$$

$$P(\text{murah}|\text{spam}) = \frac{1+1}{4+13} = \frac{2}{17} = 0,11$$

Setelah mendapatkan nilai *conditional probability*, dilakukan perhitungan untuk memilih kelas di mana satu *tweet* atau kalimat tersebut masuk ke dalam kelas non spam atau kelas spam. Berikut adalah contoh perhitungan dari pemilihan kelas pada satu *tweet* atau kalimat.

$$P(\text{non spam}|\text{Doc4}) = \frac{2}{3} \times \frac{1}{22} \times \frac{1}{22} \times \frac{1}{22} \times \frac{1}{22} \times \frac{1}{22} = 1.29358609 \times 10^{-7}$$

$$P(\text{spam}|\text{Doc4}) = \frac{1}{3} \times \frac{1}{17} \times \frac{1}{17} \times \frac{2}{17} \times \frac{2}{17} \times \frac{1}{17} = 9.39061704 \times 10^{-7}$$

Dari perhitungan di atas, maka kalimat pada data uji Tabel 3.7 masuk ke dalam kelas *spam* karena hasil probabilitasnya lebih besar pada kelas *non spam*.

4. Evaluasi dan Hasil Pengujian

Pada pengujian ini data yang digunakan berjumlah 1000 *tweet*. Selanjutnya data dibagi menggunakan k-fold cross validation dengan nilai k=10 sehingga menjadi 10 lipatan dengan masing-masing isinya adalah 100 data.

Skenario pengujian yang dilakukan dibagi menjadi 2 yaitu analisis dan hasil pengujian pada tahap *preprocessing* dan parameter *k-fold*.

4.1 Analisis dan Hasil Pengujian Tahap Preprocessing

Pada tahap ini dilakukan analisis pengujian terhadap tahap-tahap *preprocessing*. Berikut skenario tahap *preprocessing* yang dilakukan, dapat dilihat pada tabel 4.1

Tabel 4.1. Skenario tahap preprocessing

| Skenario ke- | Preprocessing yang digunakan |
|--------------|--|
| 1 | Tanpa <i>preprocessing</i> |
| 2 | <i>Case folding, Tokenizing dan Stopword removal.</i> |
| 3 | <i>Case folding, Tokenizing, Stopword removal dan Stemming</i> |

Dari masing-masing skenario didapatkan akurasi dengan menggunakan metode Naïve Bayes, dilakukan tanpa *preprocessing* dan dengan *preprocessing* terlebih dahulu sebelum masuk ke tahap pengklasifikasian.

4.1.1 Tanpa Preprocessing

Pada tahap ini tidak dilakukan *preprocessing* terhadap data set yang dimiliki, dataset yang sudah dilabelling langsung dimasukkan ke dalam proses pengklasifikasian. Berikut hasil yang diperoleh, dapat dilihat pada tabel 4.1

Tabel 4.2. Skenario tanpa preprocessing

| Skenario | Precision | Recall | F1 | Akurasi |
|---------------------|-----------|--------|--------|---------|
| Tanpa Preprocessing | 78,35% | 74,78% | 78,35% | 74,14% |

4.1.2 Case folding, Tokenizing dan Stopword Removal

Pada tahap ini *preprocessing* yang dilakukan *casefolding, tokenizing dan stopwords removal* terhadap data set yang dimiliki. dataset yang sudah dilabelling dan di *preprocessing* kemudian dimasukkan ke dalam proses pengklasifikasian. Berikut hasil yang diperoleh, dapat dilihat pada tabel 4.2

Tabel 4.3. Skenario tahap case folding, tokenizing dan stopwords removal

| Skenario | Precision | Recall | F1 | Akurasi |
|------------------|-----------|--------|--------|---------|
| Stopword Removal | 83,86% | 75,07% | 78,81% | 74,94% |

4.1.3 Case folding, Tokenizing, Stopword Removal dan Stemming

Pada tahap ini *preprocessing* yang dilakukan *casefolding, tokenizing, stopwords removal, stemming* terhadap data set yang dimiliki. dataset yang sudah dilabelling dan di *preprocessing* kemudian dimasukkan ke dalam proses pengklasifikasian. Berikut hasil yang diperoleh, dapat dilihat pada tabel 4.4

Tabel 4.4. Skenario tahap case folding, tokenizing, stopwords removal dan stemming

| Skenario | Precision | Recall | F1 | Akurasi |
|----------|-----------|--------|--------|---------|
| Stemming | 84,70% | 76,13% | 79,78% | 76,34% |

Akurasi terbaik dimiliki oleh *preprocessing* yang lengkap menggunakan tahap *preprocessing* yang dilakukan *casefolding, tokenizing, stopwords removal, stemming* yaitu sebesar 76,34%, karna *stemming* mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan-imbuhan pada kata dalam dokumen atau mengubah kata kerja menjadi kata benda, seperti kata 'mengembangkan' berubah menjadi 'kembang sehingga akan mempengaruhi perhitungan pada proses pengklasifikasian *Naïve Bayes*, akan tetapi waktu yang dibutuhkan untuk proses *stemming* sendiri cenderung lebih lama dari tahap *preprocessing* lainnya

Dari analisis prediksi sistem, terdapat kesalahan dalam memprediksi kata spam menjadi bukan spam. Hal tersebut terjadi karena banyaknya variasi data pada dataset yang digunakan. Variasi *tweet spam* yang dipelajari oleh system terdiri dari 4 jenis, yaitu *tweet* motivasi, nasehat agama, promosi produk dan pantun. Berikut beberapa contoh *tweet spam* yang salah diprediksi oleh *system*, dapat dilihat pada tabel 4.5:

Tabel 4.5. contoh tweet spam

| Kalimat tweet | Label | Prediksi |
|---|-------|----------|
| Tidak dikatakan hebat seorang pasangan yang tidak mampu setia terhadap pasangannya. | Spam | Non Spam |
| Tanding bola hasil seri tak ada kesempatan lawan menyerang. | Spam | Non Spam |
| Teruslah percaya dan bekerja keraslah. | Spam | Non Spam |

4.2 Pengujian K-fold

Perbandingan hasil akurasi pada *Naïve Bayes Classifier* (NBC) dengan *preprocessing* lengkap dengan tahap *casefolding*, *tokenizing*, *stopword removal*, *stemming* mendapatkan nilai akurasi yang paling baik. Berikut diujikan untuk beberapa nilai K-fold pada *cross validation*, dapat dilihat pada tabel 4.6.

Tabel 4.6. pengujian k-fold

| <i>K-fold</i> | Precision | Recall | F1 | Akurasi |
|---------------|-----------|--------|--------|---------|
| 2 | 84,70% | 76,13% | 79,78% | 76,34% |
| 3 | 84,38% | 61,51% | 69,37% | 66,31% |
| 4 | 78,79% | 50,00% | 58,78% | 58,46% |
| 5 | 76,10% | 40,90% | 49,89% | 53,42% |

Dari tabel 4.4 dapat disimpulkan K-fold yang memiliki nilai k=2 mempunyai akurasi yang paling baik dan dapat disimpulkan bahwa semakin besar parameter K-fold semakin rendah akurasi yang didapatkan. Ketika k=2 dengan pembagian data di fold pertama 263 non spam dan 273 spam, kemudian di fold kedua 358 non spam dan 142 spam masing-masing menghasilkan akurasi pada tiap foldnya 89,94% dan 62,69%. Ketika k=3 dengan pembagian data di fold pertama 128 non spam dan 205 spam, kemudian di fold kedua 263 non spam dan 70 spam, selanjutnya di fold ketiga 231 non spam dan 103 spam masing-masing menghasilkan akurasi pada tiap foldnya 83,89%, 55,45% dan 60,44. Akurasi yang dihasilkan oleh k=2 lebih tinggi dibandingkan dengan k yang memiliki nilai lebih tinggi, karna ketika k=2 dari 1000 data yang digunakan akan dibagi menjadi 500:500 untuk dijadikan sebagai data test dan data train, sehingga variasi data yang dipelajari sistem pada data train tidak terlalu beragam dan menghasilkan akurasi yang tinggi. ketika nilai k lebih besar maka variasi data yang dipelajari juga makin beragam sehingga menyebabkan akurasinya lebih kecil.

5. Kesimpulan dan Saran

Berdasarkan hasil pengujian dan analisis yang dilakukan sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut:

- Metode *Naïve Bayes Classifier* dengan *preprocessing* dapat mengklasifikasikan kelas spam dan non-spam yang terdapat pada suatu *tweet* dengan rata-rata akurasi berkisar 76,34%, sedangkan rata-rata akurasi metode *Naïve Bayes Classifier* tanpa *preprocessing* adalah 74,14%.

Saran yang ingin disampaikan guna mengembangkan Tugas Akhir ini adalah sebagai berikut :

- Banyaknya variasi data yang digunakan pada penelitian ini, menyebabkan metode *Naïve Bayes* sulit dalam mempelajari pola *tweet spam*, untuk penelitian selanjutnya disarankan lebih memperkecil lingkup *spam* yang dimaksud sehingga akurasi yang dihasilkan lebih baik.

Daftar Pustaka

- [1] T. India, "blog.twitter.com," Twitter, 5 December 2018. [Online]. Available: https://blog.twitter.com/en_in/topics/events/2018/Year-on-Twitter-2018-India.html. [Accessed Thursday August 2019].
- [2] Y. Zhu, X. Wang, E. Zong, N. L. N, H. Li and Q. Yang, "Discovering Spammers in Social Networks," *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [3] H. T. W, "SPAM-Different Approach to Fighting," 2004.
- [4] F. A. Prayoga, A. Pinandito and R. S. Perdana, "Rancang Bangun Aplikasi Deteksi Spam Twitter menggunakan Metode Naive Bayes dan KNN pada Perangkat Bergerak Android," *Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, pp. 554-564, 2018.
- [5] D. H. Deshmukh, T. Ghorpade and P. Padiya, "Classification Using Preprocessing and Machine Learning Algorithms on NSL- KDD Dataset," *International Conference on Communication, Information & Computing Technology (ICCICT)*, pp. 16-17, 2015.
- [6] A. Strange, "Flickr Document Reveals Origin Of Twitter," *Wired News*, 20 April 2007. [Online]. Available: <https://www.wired.com/2007/04/flickr-document/>. [Accessed 20 Oktober 2018].
- [7] M. G. H, "Web spam taxonomy," New York, ACM Press, 2005.
- [8] A. I. Khadim, Y. -N. Cheah and N. H. Ahamed, "Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering," *2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*, pp. 69-73, 2014.
- [9] Z. Ceska and C. Fox, "The Influence of Text Pre-processing on Plagiarism Detection," 2011.

- [10] R. S. P. M. A. F. A. R. T. Lestari, "Analisis Sentimen Tentang Opini Pilkada Dki 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan N ive Bayes dan Pembobotan Emoji," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 12, pp. 1718-1724, 2017.
- [11] Kohavi, "A study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," 2014, p. 167.
- [12] M. F. A. Azka, "Pemanfaatan Ontologi dengan Paradigma Pembangunan Combined Hierarchy dalam Pengukuran Indeks Kebahagiaan Masyarakat Kota Bandung," 2016, pp. 10-15.