Klasifikasi Posting Tweet mengenai Kebijakan Pemerintah Menggunakan Naive Bayesian Classification

Garnis Berliana[1], Shaufiah, S.T, M.T.[2], Siti Sa'adah, S.T., M.T.[3]
[1,2,3]Prodi S1 Teknik Informatika. Fakultas Informatika, Universitas Telkom
[1]garnis31@gmail.com, [2]shaufiah@gmail.com, [3] tisataz@gmail.com

Abstrak

Twitter merupakan media sosial yang populer di kalangan masyarakat dalam memberikan informasi karena lebih mudah dan cepat. Dengan adanya media sosial, masyarakat menjadi lebih mudah menyampaikan aspirasi dan pendapat mengenai kebijakan yang telah dibuat oleh pemerintah. Salah satu kebijakan tersebut adalah amnesti pajak. Pada penelitian ini menggunakan algoritma *Naïve Bayesian Classification* untuk mengklasifikasikan *tweet* yang berisi informasi tentang amnesti pajak. *Naïve Bayesian Classification* merupakan salah satu teknik klasifikasi dalam data mining yang sederhana. Ekstraksi fitur yang digunakan pada pengklasifikasian amnesti pajak menggunakan naive bayesian classification adalah unigram dan frekuensi kata dimana hasil akurasi tertinggi yang didapat sebesar 53,45% dengan data training sebesar 80% dari 578 data *tweet* amnesti pajak. Metode naive bayes dengan fitur unigram kurang tepat untuk digunakan dalam pengklasifikasian *tweet* mengenai amnesti pajak.

Kata kunci: Twitter, *tweet*, data mining, analisis sentimen, klasifikasi, *naïve Bayesian classification.*

Abstract

Twitter is a popular social media among people in providing information because it is easier and faster. With the existence of social media, the community becomes easier to convey the aspirations and opinions about the policies that have been made by the government. One of the policy is tax amnesty. In this research use Naïve Bayesian Classification algorithm to classify tweets that contain information about tax amnesty. Naïve Bayesian Classification is one of the classification techniques in simple data mining. Feature extraction used in tax amnesty classification using naive bayesian classification is unigram and word frequency where the highest accuracy obtained is 53.45% with training data of 80% of 578 data tweet of tax amnesty. Naive bayes classification with unigram feature is not appropriate for tweet classification about tax amnesty.

***Keywords**: Twitter, tweet, data mining, sentiment analysis, classification, naïve Bayesian classification..*

1. Pendahuluan

Pada era globalisasi ini, pertukaran informasi terjadi dengan begitu mudah melalui media sosial, salah satunya adalah twitter. Menurut eBizMBA, twitter berada di urutan kedua sebagai media sosial terpopuler setelah Facebook di dalam Top 15 Most Popular Social Networking Sites dengan perkiraan pengunjung bulanan yang unik berjumlah 310.000.000 [18]. Twitter menjadi tempat sebagian besar masyarakat untuk mengemukakan opini mereka terkait isu yang sedang hangat dibicarakan pada saat tertentu dengan bebas. Opini-opini yang ada di twitter dapat digunakan untuk menilai sentimen atas suatu topik tertentu, seperti produk, film, jasa, tokoh publik, kebijakan pemerintah dan sebagainya. Salah satu kebijakan pemerintah yang banyak menarik perhatian masyarakat adalah amnesti pajak. Amnesti pajak merupakan kebijakan pemerintah di bidang perpajakan yang berlaku hingga 31 Maret 2017 [2].

Pada Tugas Akhir ini, tweet yang mengandung opini masyarakat terhadap amnesti pajak diklasifikasikan menggunakan metode klasifikasi naïve Bayesian classification. Tweet yang berisi amnesti pajak diambil dengan menggunakan NodeXL. Namun, data yang telah dikumpulkan masih belum dapat digunakan untuk klasifikasi karena terdapat kata-kata yang tidak baku dan terdapat banyak noise di dalam data tersebut sehingga perlu dilakukan preprocessing. Tahapan preprocessing dilakukan untuk menghilangkan hal yang tidak dibutuhkan (url, mention), tokenization, stopword removal, dan stemming. Berdasarkan penelitian yang dilakukan Ledy Agusta, stemming dengan algoritma nazief dan adriani memiliki hasil akurasi yang lebih baik dibandingkan dengan stemming menggunakan algoritma porter [5]. Setelah itu, data tersebut diklasifikasikan menggunakan naïve Bayesian classification. Pendekatan naïve Bayesian classification merupakan pendekatan yang mengacu pada teorema Bayes yang menggunakan prinsip peluang statistika untuk mengkombinasikan pengetahuan sebelumnya dengan pengetahuan baru untuk menyelesaikan masalah klasifikasi [6]. Pada penelitian ini, data twitter yang digunakan adalah data yang berisi opini sehingga data yang telah dikumpulkan terlebih dahulu melalu filtering manual dimana tweet dengan username kemenkeuri, dirjenpajakri, akun kantor pajak daerah, dan

media-media berita dihapuskan. Kata-kata yang tidak baku pada data yang telah di-filter diubah menjadi kata baku secara manual. Pada penelitian tugas akhir ini, sistem mampu mengklasifisikan data twitter ke dalam tiga kelas yaitu positif, negatif, dan netral dengan ekstraksi fitur menggunakan unigram dan frekuensi kata (term frequency). Tetapi akurasi yang dihasilkan hanya mencapai 53,45% dengan data pembelajaran sebesar 80% dari 578 data twitter.

2. Dasar Teori

2.1 Text Mining

Data mining merupakan suatu proses penemuan pengetahuan (*Knowledge Discovery*) dari sejumlah data yang besar. Langkah-langkah untuk melakukan penemuan pengetahuan (*Knowledge Discovery*) adalah sebagai berikut [4] :

1. *Data cleaning* yang berfungsi untuk menghapus *noise* dan data yang tidak konsisten.
2. *Data integration* merupakan suatu proses dimana beberapa sumber data dapat dikombinasikan.
3. *Data selection* merupakan suatu proses dimana data yang relevan dengan *analysis task* akan diambil dari *database*.
4. *Data transformation* merupakan suatu proses yang mengubah atau menggabungkan data ke bentuk yang tepat untuk *mining*.
5. *Data mining* merupakan suatu proses penting dimana metode *intelligent* diterapkan untuk mengambil pola data.
6. *Pattern evaluation* berfungsi untuk mengidentifikasi pola-pola yang menarik untuk mewakili *knowledge* (pengetahuan) berdasarkan beberapa tindakan *interestingness*.
7. *Knowledge presentation* merupakan suatu proses dimana visualisasi dan teknik representasi *knowledge* (pengetahuan) digunakan untuk mempresentasikan *knowledge* ke pengguna.

2.2 Analisis Sentimen

Analisis Sentimen atau opinion mining merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap suatu masalah atau objek oleh seseorang, apakah cenderung beropini positif atau negatif. Besarnya pengaruh dan manfaat dari analisis sentimen menyebabkan penelitian atau aplikasi mengenai analisis sentimen berkembang pesat, bahkan di Amerika kurang lebih 20-30 perusahaan yang memfokuskan pada layanan analisis sentimen. Pada dasarnya analisis sentimen merupakan klasifikasi, tetapi tidak semudah proses klasifikasi biasa karena terkait penggunaan bahasa dimana terdapat ambigu dalam penggunaan kata, tidak adanya intonasi dalam sebuah teks, dan perkembangan dari bahasa itu sendiri [1].

Skripsi analisis sentimen pada skripsi ini dilakukan dengan menggunakan pendekatan dalam *machine learning* yang dikenal dengan metode naive bayes dan dikhususkan pada dokumen teks berbahasa Indonesia yang diambil dari Twitter.

2.3 Klasifikasi : Naive Bayes

Klasifikasi adalah fungsi pembelajaran yang mengklasifikasikan sebuah unsur data ke dalam salah satu dari beberapa kelas yang telah didefinisikan [9]. Salah satu metode klasifikasi yang dapat digunakan adalah metode naïve bayes yang sering disebut *naïve Bayesian classification* (NBC). Naive Bayes merupakan sebuah algoritma pembelajaran yang berbasis pada teori Bayes dengan menggunakan asumsi yang kuat (*naive*). Teori Bayes merupakan suatu teori tentang mencari suatu probabilitas sesuatu berdasarkan data yang telah ada sebelumnya. Metode ini juga bisa digunakan untuk mengklasifikasikan opini berdasarkan data yang telah dilatih sebelumnya. Inti dari naive bayes adalah mencari probabilitas tertinggi dari suatu data. Rumus bayes dapat ditulis sebagai berikut:

$$\mathrm{Pcd} = \frac{\mathrm{P(c)} \times \mathrm{Pdc}}{\mathrm{P(d)}} \quad (2.1)$$

Berikut adalah keterangan dari rumus (2.1):

- Pcd adalah probabilitas kelas c setelah d dimasukkan ke kelas c.
- $P(c)$ adalah probabilitas kelas c sebelumnya
- Pdc adalah probabilitas d pada kelas c
- Pd adalah probabilitas d

Naive Bayes telah dipelajari sejak tahun 1950. Naive bayes diperkenalkan dengan nama yang berbeda ke dalam komunitas *text retrieval* pada awal tahun 1960 dan tetap menjadi metode yang populer untuk

kategorisasi teks dimana menilai dokumen ke dalam satu kategori atau kategori lainnya dengan frekuensi kata sebagai fitur. Dengan *preprocessing* yang tepat, metode ini dapat menjadi lebih baik daripada metode SVM. Naive Bayes ada tigas jenis menurut distribusi fitur, yaitu gaussian naive bayes, multinomial naive bayes, dan bernoulli naive bayes. Tetapi algoritma naive bayes yang sering digunakan untuk *text mining* adalah multinomial naive bayes. Multinomial Naïve Bayes merupakan salah satu metode spesifik dari metode Naive Bayes. Multinomial naïve bayes ini juga merupakan salah satu mechine learning dalam *supervised learning* pada proses pengklasifikasian teks dengan menggunakan nilai probabilitas suatu kelas dalam suatu dokumen. Menurut Multinomial Naïve Bayes, secara umum probabilitas suatu dokumen d, sebagai bagian dari anggota kelas c. Probabilitas dari suatu dokumen d terhadap kelas c dapat dihitung dengan rumus sebagai berikut [19].

$$P(c|d)\ \alpha\ P(c)\ \prod_{1}^{n} P(t_k|c) \qquad (2.2)$$

Dimana:

- $P(\ tk\ |c)$ adalah probabilitas kemunculan suatu *term* tk dalam dokumen pada kelas c dimana tk adalah *term* dalam dokumen d.
- P(c) adalah prior probabilitas suatu dokumen pada kelas c.

Perhitungan nilai P(c) dan P(tk|c) dilakukan pada saat melatih data. Probabilitas suatu kelas dapat dilakukan dengan jumlah suatu kelas dokumen dalam kelas latih atau Nc dibagi dengan jumlah total dokumen kelas yang ada atau *N* dalam dokumen latih [19], sebagai berikut:

$$P(c) = \frac{N_c}{N} \qquad (2.3)$$

Perhitungan *conditional* probabilitas dilakukan untuk menghitung probabilitas kemunculan suatu kata dalam setiap kelas. *Conditional probability* dapat dilakukan dengan menggunakan frekuensi kemunculan suatu kata pada suatu kelas.

$$P(t|c) = \frac{T_{ct} + 1}{\sum_{t' \epsilon V}(T_{ct'} + 1)} \qquad (2.4)$$

Maximum a posterior (MAP) digunakan untuk menentukan kelas suatu dokumen testing dengan mengambil nilai maksimum probabilitas setiap dokumen. Adapun rumus untuk MAP adalah sebagai berikut :

$$C_{map} = argmax_{c \in C} \hat{P}(c|d)$$
$$= argmax_{c \in C} \hat{P}(c) \prod_{1}^{n_k} \hat{P}(t_k|c) \quad (2.5)$$

Pada rumus MAP diatas setiap *conditional probability* atau setiap probabilitas suatu kata dikalikan. Perkalian tersebut menghasilkan floating point underflow. Dalam hal ini untuk menghindari floating point underflow maka akan dilakukan proses penjumlahan setiap probabilitas kata dengan menggunakan logaritma dimana log(x,y) =log(x)+ log(y). Untuk mencari MAP pada Multinomial Naïve Bayes adalah sebagai berikut:

$$C_{map}$$
$$= argmax_{c \in C} [log \hat{P}(c)$$
$$+ \sum_{1 \le k \le nd} log \hat{P}(t_k|c) \quad (2.6)$$

2.4 Pembobotan fitur

Pembobotan fitur merupakan sebuah proses pemberian nilai pada setiap fitur berdasarkan relevansi dan pengaruhnya terhadap hasil klasifikasi. Nilai tersebut nantinya dapat digunakan sebagai dasar untuk melakukan seleksi fitur berdasarkan minimum bobot yang telah dihitung dari setiap fitur. Pembobotan dilakukan dengan menggunakan metode TF-IDF.

Algoritma TF-IDF pertama kali dicetuskan oleh Salton dan Buckley pada tahun 1988 dan digunakan untuk kepentingan information retrieval, yang kemudian turut dimanfaatkan sebagai salah satu algoritma dalam metode feature weighting dalam text mining. TF-IDF memiliki formula sebagai berikut:

$$\text{TF} - \text{IDF} = \text{TF} \times \text{IDF} \quad (2.7)$$

Rumus tersebut dapat dijabarkan menjadi term frequency dari fitur i pada dokumen j dikalikan dengan IDF dari fitur i pada dokumen j, dimana IDF sendiri merupakan kepanjangan dari Inverse Document Frequency.

Berikut adalah rumus untuk TF dan IDF :

$$\text{TF} = \frac{\text{jumlah kemunculan term pada satu dokumen}}{\text{jumlah seluruh term dalam satu dokumen}} \quad (2.8)$$

$$\text{IDF} = \log \frac{\text{jumlah seluruh dokumen}}{\text{jumlah dokumen suatu term muncul}} \quad (2.9)$$

Semakin sering sebuah fitur muncul dalam sebuah teks, maka semakin besar pula bobot yang akan didapat, yang artinya maka akan semakin penting pula fitur tersebut. Metode ini dianggap efektif untuk information retrieval.

2.5 Pengukuran Kinerja Sistem

Pengukuran kinerja sistem klasifikasi umumnya dilakukan dengan cara menggunakan matriks confusion. Matriks confusion merupakan tabel yang mencatat hasil kerja klasifikasi.

f_{ij}		Kelas hasil prediksi (j)	
		Kelas = 1	Kelas = 0
Kelas asli (i)	Kelas = 1	f_{11}	f_{10}
	Kelas = 0	f_{01}	f_{00}

Setiap sel f_{ij} dalam matriks menyatakan jumlah record/ data dari kelas i yang hasil prediksinya masuk ke kelas j. Misalnya sel f_{11} adalah jumlah data dalam kelas 1 yang secara benar dipetakan ke kelas 1, dan f_{10} adalah data dalam kelas 1 yang dipetakan secara salah ke kelas 0. Berdasarkan isi matriks confusion, maka dapat diketahui jumlah data dari masing-masing kelas yang diprediksi secara benar yaitu (f_{11}+f_{00}) dan data yang diklasifikasikan secara salah yaitu (f_{10}+f_{01}). Kuantitas matriks confusion dapat diringkas menjadi dua nilai, yaitu akurasi. Dengan mengetahui jumlah data yang diklasifikasikan secara benar maka dapat diketahui akurasi hasil prediksi. Untuk menghitung akurasi digunakan formula sebagai berikut :

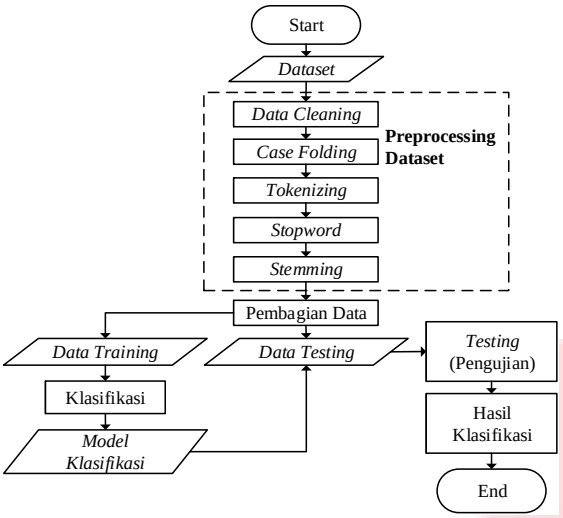
$$\text{Akurasi} = \frac{\text{Jumlah data yang diprediksi secara benar}}{\text{Jumlah prediksi yang dilakukan}} \quad (2.10)$$

Semua algoritma klasifikasi berusaha untuk membentuk model yang mempunyai akurasi yang tinggi (laju error yang rendah). Umumnya model yang dibangun dapat memprediksi dengan benar pada semua data yang menjadi data latihnya, tetapi ketika model berhadapan dengan data uji barulah kinerja model dari sebuah algoritma klasifikasi ditentukan [11].

3. Pembahasan

3.1 Gambaran Umum Sistem

Gambaran dari proses yang terjadi dalam sistem adalah sebagai berikut :



Gambar 1 - Gambaran umum sistem

Berdasarkan gambar 1 mengenai gambaran sistem, langkah langkah yang diterapkan dalam skema sistem adalah :

1. Pengambilan data dengan teknik crawling pada media sosial Twitter dengan topik amnesti pajak menggunakan NodeXL. Kata kunci yang digunakan adalah #amnestipajak, #pengampunanpajak, #taxamnesty, amnesti pajak, pengampunan pajak, dan tax amnesty. Setelah mendapatkan data yang diperoleh dari proses *crawling*, selanjutnya data yang berisi *tweet* dipindahkan ke Microsoft Excel secara manual. Karena data tersebut banyak yang tidak mengandung opini, maka penulis menyaring secara manual untuk mendapatkan data yang berisi opini. Data *tweet* yang berisi opini tersebut kemudian diberikan label secara manual. Pemberian label dibagi menjadi tiga, yaitu positif, negatif, dan netral.
2. Sebagian besar data yang didapat belum memenuhi penulisan tata Bahasa Indonesia yang baku. Kata-kata yang tidak baku diubah menjadi baku secara manual.
3. Data melalui tahap *case folding* dimana data tweet yang telah terkumpul diubah menjadi huruf kecil semua.
4. Username pada data yang telah terkumpul dihilangkan karena tidak berpengaruh pada klasifikasi. *Username* pada twitter biasanya diawali dengan simbol "@" yang diikuti dengan nama *user* dan diakhiri dengan spasi, contohnya @garnis_berliana. Tahap ini dilakukan dengan melakuan pencocokan string dengan pola *username* kemudian menghapusnya apabila sesuai dengan pola *username.*.
5. Selanjutnya, menghilangkan url karena tidak dibutuhkan dalam proses klasifikasi.
6. Tahap selanjutnya melakukan tokenizing untuk memisahkan string berdasarkan kata yang menyusunnya.
7. Melakukan tahap *stopword removal*. Stopword merupakan kata umum yang sering muncul dalam jumlah besar dan biasanya dianggap tidak memiliki makna, seperti kata penghubung, kata pengganti, dan lain sebagainya. Contoh kata yang termasuk stopword adalah saya, aku, yang, dan, sebagai, pak, bu, dan sebagainya. Karena tidak memiliki makna dan agar jumlah kata yang diproses berkurang, maka kata yang termasuk ke dalam stopword perlu dihapus. Penulis menggunakan daftar stopword dari Tala dan menambahkan kata-kata lain yang termasuk stopword.
8. Tahap selanjutnya yaitu proses stemming yang merupakan proses mengubah kata berimbuhan menjadi kata dasar sesuai kamus KBBI. Algoritma stemming yang digunakan yaitu algoritma nazief dan adriani yang dibangun oleh Sastrawi.
9. Setelah tahap *preprocessing*, data tweet dibagi menjadi data training dan data testing menggunakan sistem dengan presentasi data training 80% dan data testing sebesar 20%.
10. Klasifikasi naive bayes dilakukan dengan menggunakan fitur kata (unigram) dari data training yang kemudian dihitung frekuensi kemunculan kata dan dihitung probabilitasnya untuk digunakan dalam klasifikasi data testing.
11. Dari tahapan yang sudah dilakukan diatas, sistem akan mengklasifikasikan data testing dan menghasilkan nilai akurasi dari klasifikasi yang telah dilakukan untuk melihat kinerja sistem dalam pengklasifikasian tweet tentang amnesti pajak.

3.2 Hasil dan Analisis Pengujian

Pengujian yang dilakukan oleh penulis dilakukan dengan confusion matriks dengan melihat nilai akurasi dari berbagai skenario uji yang telah dirancang. Ada 3 skenario uji yang akan dijalankan, yaitu :

i. Klasifikasi dilakukan dengan tahapan *pre-processing* dengan stemming Sastrawi dan pembagian data training dan testing dengan presentase berbeda.
ii. Klasifikasi dilakukan dengan tahapan *pre-processing* tanpa proses stemming dan pembagian data training dan testing dengan presentase berbeda.
iii. Klasifikasi dilakukan dengan tahapan *pre-processing* tanpa *stopword removal* dan pembagian data training dan testing dengan presentase berbeda.

Pada skenario uji yang pertama, didapat akurasi sebagai berikut:

```
============================================================
Hasil akurasi dari skenario perbedaan presentase data training
Akurasi dengan data training sebesar 80%= 53.45%
Akurasi dengan data training sebesar 70%= 52.3%
Akurasi dengan data training sebesar 60%= 49.78%
Akurasi dengan data training sebesar 50%= 50.17%
Akurasi dengan data training sebesar 90%= 50.0%
Akurasi dengan data training sebesar 40%= 49.28%
Akurasi dengan data training sebesar 30%= 49.26%
Akurasi dengan data training sebesar 20%= 46.32%
Akurasi dengan data training sebesar 10%= 46.15%
============================================================
```

Pada skenario pertama dapat dilihat jika besarnya data pembelajaran dan data testing mempengaruhi nilai akurasi. Akurasi tertinggi terjadi pada saat data pembelajaran sebesar 80% dengan nilai 53,45%. Tetapi karena pembagian data dilakukan menggunakan sistem dimana data yang diacak sebanyak 100 setiap kali dijalankan sehingga memungkinkan perbandingan jumlah data dari tiap kelas berbeda pada data training. Perbedaan data tiap kelas pada data training dapat menyebabkan sistem kurang mempelajari suatu kelas tertentu sehingga mempengaruhi kinerja sistem dalam mengklasifikasikan. Karena jumlah data kelas netral yang lebih banyak dibandingkan kedua kelas lainnya, maka pada saat pembagian data yang dilakukan oleh sistem kemungkinan sistem untuk lebih mempelajari data kelas netral semakin besar yang terlihat dimana hasil pengklasifikasian pada data testing yang dilakukan sistem semuanya termasuk ke kelas netral.

Pada skenario uji yang kedua, didapat akurasi sebagai berikut:

```
Hasil akurasi dari skenario perbedaan presentase data training dan data
Akurasi dengan data training sebesar 80%= 53.45%
Akurasi dengan data training sebesar 70%= 52.3%
Akurasi dengan data training sebesar 60%= 49.78%
Akurasi dengan data training sebesar 50%= 50.17%
Akurasi dengan data training sebesar 90%= 50.0%
Akurasi dengan data training sebesar 40%= 49.28%
Akurasi dengan data training sebesar 30%= 49.26%
Akurasi dengan data training sebesar 20%= 46.32%
Akurasi dengan data training sebesar 10%= 46.15%
============================================================
```

Akurasi dari klasifikasi tanpa melakukan stemming dibandingkan dengan menggunakan stemming hasilnya sama saja pada presentase data training dan testing yang berbeda. Dapat disimpulkan bahwa, proses stemming pada sistem ini tidak berpengaruh terhadap hasil akurasi.

Pada skenario uji yang ketiga, didapat akurasi sebagai berikut:

```
Hasil akurasi dari skenario perbedaan presentase data training dan data t
Akurasi dengan data training sebesar 80%= 44.83%
Akurasi dengan data training sebesar 70%= 48.85%
Akurasi dengan data training sebesar 60%= 46.55%
Akurasi dengan data training sebesar 50%= 47.24%
Akurasi dengan data training sebesar 90%= 43.1%
Akurasi dengan data training sebesar 40%= 46.84%
Akurasi dengan data training sebesar 30%= 48.28%
Akurasi dengan data training sebesar 20%= 46.34%
Akurasi dengan data training sebesar 10%= 46.17%
============================================================
```

Dibandingkan dengan hasil skenario pertama, hasil skenario lebih kecil jika sistem tidak melakukan tahapan *stopword removal*. Hal tersebut menunjukkan bahwa tahapan *stopword removal* sangat berpengaruh dalam pengklasifikasian dikarenakan kata-kata yang tidak berguna untuk proses klasifikasi dihilangkan. Dengan penghapusan kata-kata yang tidak berguna tersebut, maka fitur kata yang digunakan untuk proses klasifikasi pun berkurang. Untuk tahapan ini, daftar kata *stopword* dapat disesuaikan sesuai dengan data yang digunakan.

4. Kesimpulan

Kesimpulan yang diperoleh dari tugas akhir ini adalah sebagai berikut :

- Pembagian dataset menjadi data training dan data testing mempengaruhi kinerja sistem dalam mengklasfikasikan data. Namun karena pada penelitian ini pembagian dilakukan dengan sistem dimana kemungkinan data training setiap kelas tidak seimbang sehingga mempengaruhi kinerja sistem yang dapat dilihat pada hasil akurasi pada skenario pembagian data training dan data testing dengan presentase yang berbeda. Jika dilihat dari hasil akurasi, pembagian dataset yang dapat menghasilkan akurasi tinggi adalah data training sebesar 80% dan data training sebesar 20%.

- Proses stemming pada sistem yang dibuat tidak berpengaruh terhadap kinerja sistem dalam mengklasifikasikan data yang dapat dilihat dari hasil akurasi dari sistem yang menggunakan stemming dengan hasil akurasi dari sistem yang tidak menggunakan stemming adalah sama besar hasilnya.
- Proses *stopword removal* mempengaruhi kinerja sistem dalam mengklasifikasikan data. Hal tersebut dapat dilihat pada skenario yang tidak menggunakan proses *stopword removal* dimana hasil akurasi yang dihasilkan lebih kecil dibandingkan dengan yang menggunakan *stopword removal*.
- Ekstraksi fitur unigram dan frekuensi kata dalam klasifikasi amnesti pajak ini menghasilkan akurasi tertinggi sebesar 53,45% dan terendah sebesar 46,15%. Klasifikasi tweet mengenai amnesti pajak menggunakan metode naive bayes kurang maksimal hasilnya jika dibandingkan dengan penelitian-penelitian dari referensi yang dapat dilihat dari hasil akurasi sistem yang terbesar pada 53,45% sedangkan penelitian lain hasil akurasinya dapat mencapai 70% ke atas. Hal tersebut dapat terjadi karena penggunaan ekstrasi fitur yang kurang tepat atau metode naive bayes tidak tepat untuk digunakan pada klasifikasi amnesti pajak.

Daftar Pustaka

[1] Muhamad Yusuf Nur dan Diaz D.Santika, "Analisis Sentimen pada Dokumen Berbahasa Indonesia dengan Pendekatan Support Vector Machine," *Konferensi Nasional Sistem dan Informatika 2011*, pp. 9-14, 2011.

[2] "Amnesti Pajak," 28 Desember 2017. [Online]. Available: http://www.pajak.go.id/content/amnesti-pajak.

[3] Ismail Sunni, Dwi Hendratmo Widyantoro, "Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini terhadap Tokoh Publik," *Jurnal Sarjana Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika*, pp. 200-206, 2012.

[4] Dyarsa Singgih Pamungkas, Noor Ageng Setiyanto, Erlin Dolphina, "Analisis Sentiment pada Sosial Media Twitter menggunakan Naive Bayes Classifier terhadap Kata Kunci "Kurikulum 2013"," *Techno.COM*, pp. 299-314, 2015.

[5] L. Agusta, "Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia," *Konferensi Nasional Sistem dan Informatika 2009*, pp. 196-201, 2009.

[6] Sandi Fajar Rodiyansyah, E.W, "Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayesian Classification," 2012.

[7] Ronen Feldman, J.S, The Text Mining Handbook, Advanced Approaches in Analyzing Unstructured Data, New York: Cambridge University Press, 2006.

[8] Courtney D.Corley, Diane J.Cook, Armin R.Miller, Karan P.Singh, "Text and Structural Data Mining of Influenza Mentions in Web and Social Media," 2010.

[9] P. B. Batrinca, "Social Media analytics; a survey of techniques, tools and platforms," 2014.

[10] V.S.Moertini, "Data Mining Sebagai Solusi Bisnis," 2002.

[11] E.Prasetyo, Data Mining, Mengolah Data Menjadi Informasi Menggunakan Matlab, Penerbit Andi, 2014.

[12] "About Twitter," Twitter, 2015. [Online]. Available: https://about.twitter.com/. [Accessed 19 Maret 2015].

[13] "The Search API," Twitter, 2015. [Online]. Available: https://dev.twitter.com/rest/public/search. [Accessed 19 Maret 2015].

[14] Tan P.N., Steinbach M., Kumar V., Introduction to Data Mining, Boston: Pearson Education, 2006.

[15] Han Jiawei, Kamber Micheline, Data Mining:Concepts and Techniques, San Fransisco: Morgan Kaufmann Publisher, 2006.

[16] I. Rish, An Empirical study of the Naive Bayes Classifier, California: International Joint Conference on Artificial Intelligence, 2006.

[17] I. Witten, Text Mining: Practical Handbook of Internet Computing, Florida: Chapman & Hall/CRC Press, 2005.

[18] "Top 15 Most Popular Social Networking Sites," 1 Maret 2015. [Online]. Available: http://www.ebizmba.com/articles/social-networking-websites. [Accessed 18 Maret 2015].

[19] UP, O.e., "Naive Bayes text classification," [Online]. Available: https://nlp.standford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html. [Accessed 5 Januari 2018].