SISTEM KLASIFIKASI *FEEDBACK* PELANGGAN DAN REKOMENDASI SOLUSI ATAS KELUHAN DI UPT PUSKOM UNS DENGAN ALGORITMA NAÏVE BAYES *CLASSIFIER* DAN *COSINE SIMILARITY*

Aisha Alfiani Mahardhika
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A
Surakarta
aish.alfiani@gmail.com

Ristu Saptono
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A
Surakarta
r_saptono@uns.ac.id

Rini Anggrainingsih
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
rinianggra@gmail.com

***Abstrak*—Saat ini, konsumen dapat menyampaikan keluhan terhadap UPT Puskom UNS melalui *mentions* terhadap akun Twitter. *Mentions* yang diberikan oleh konsumen kemudian diklasifikasikan apakah mentions tersebut termasuk keluhan, berita atau *spam*. Klasifikasi *mentions* dilakukan menggunakan algoritma Naïve Bayes *Classifier* berdasarkan *supervised learning*. Peningkatan akurasi untuk algoritma Naïve Bayes *Classifier* dilakukan dengan menggunakan teknik *Laplacian Smoothing*. Algoritma *Cosine Similarity* digunakan untuk mengelompokkan *mentions* keluhan yang memiliki *term* yang sama. Dari kelompok *mentions* tersebut, administrator akan memberikan solusi yang relevan terhadap keluhan. Hasil penelitian menunjukkan bahwa proses klasifikasi dengan algoritma Naïve Bayes *Classifier* untuk proses pelatihan memiliki tingkat akurasi terendah 86.67% dengan data pelatihan sebanyak 30 *mentions* dan tingkat akurasi tertinggi 100% dengan data pelatihan sebanyak 20 *mentions*. Proses pengujian dilakukan secara bertahap dengan tingkat akurasi terendah adalah 60% yang dicapai pada pengujian pertama dan kedua, sedangkan tingkat akurasi tertinggi dicapai pada pengujian kelima dan keenam yakni 90%. *Mentions* keluhan tidak dapat terkelompokkan dengan algoritma *Cosine Similarity* karena jumlah data yang sangat terbatas yakni 29 data dan tidak ada *mentions* yang memiliki *term* sama. Namun setelah dilakukan *self-test, mentions* keluhan yang memiliki *term* sama dapat terkelompokkan dengan baik.**

Kata kunci—Klasifikasi, Naïve Bayes Classifier, Cosine Similarity, rekomendasi solusi.

1. PENDAHULUAN

Keluhan merupakan salah satu sinyal ketidakpuasan yang diberikan konsumen terhadap sebuah perusahaan. Perbedaan antara harapan dan kemampuan sesungguhnya dari produk atau jasa yang diterima oleh konsumen akan menyebabkan ketidakpuasan, dimana hal ini dapat menimbulkan negative effect yang diyakini akan berpengaruh terhadap loyalitas konsumen [1]. Oleh sebab itu, penanganan terhadap keluhan konsumen pun menjadi hal yang mutlak dilaksanakan oleh perusahaan.

Beragam cara digunakan perusahaan agar konsumen dapat menyampaikan keluhannya dengan baik. Saat ini, cara lain yang mulai dilirik oleh perusahaan agar konsumen dapat menyampaikan keluhannya adalah melalui *social media*. Data menunjukkan bahwa dari 63 juta masyarakat pengguna internet di Indonesia, sebanyak 95% diantaranya menggunakan internet untuk mengakses *social media* yakni Facebook dan Twitter [2].

Salah satu *social media* yang dapat digunakan untuk menyampaikan keluhan konsumen adalah Twitter, yang beralamat di http://www.twitter.com. Twitter dapat dilihat sebagai alat komunikasi interaktif dari mulut ke mulut melalui media elektronik [3]. Oleh sebab itu, Twitter merupakan *social media* yang tepat untuk menjalin hubungan dengan pelanggan [3].

Sebagai sebuah institusi pelayanan, UPT. Puskom UNS memiliki cara untuk menghimpun keluhan dari konsumen, salah satunya adalah melalui Twitter yang dapat diakses di http://www.twitter.com/UPTPuskomUNS. Konsumen dapat menyampaikan keluhan terhadap UPT. Puskom UNS dengan melakukan *mentions* terhadap akun Twitter tersebut.

Salah satu kelemahan penyampaian keluhan melalui Twitter adalah *mentions* berbentuk teks digital tidak terstruktur. Selain itu, tidak semua *mentions* yang diberikan kepada UPT. Puskom UNS adalah berupa keluhan. Hal tersebut menyulitkan administrator Twitter, karena administrator harus memilih terlebih dahulu *mentions* mana yang dianggap sebagai keluhan, dan selanjutnya menjawab keluhan dari konsumen secara manual. Selain itu, *mentions* yang tidak terbaca ataupun adanya *mentions* berisi keluhan yang sama dari dua atau lebih *user* berbeda menyebabkan administrator tidak dapat menjawab keluhan secara maksimal.

Oleh sebab itu dibutuhkan sebuah sistem yang dapat memudahkan administrator untuk mengelola keluhan pelanggan dengan memilah *tweet* berupa keluhan dan bukan keluhan, dan selanjutnya mengelompokkan *tweet* keluhan yang dianggap memiliki makna sama. Salah satu cara yang dapat digunakan adalah dengan metode analisis *text mining*. Analisis *text mining* diperlukan dalam menangani masalah teks digital tidak terstruktur. Salah satu kegiatan penting dalam *text mining* adalah klasifikasi atau kategorisasi teks. Kategorisasi teks sendiri saat ini memiliki berbagai cara pendekatan antara lain pendekatan *probabilistic*, *support vector machine* (SVM), dan *artificial neural network*, atau *decision tree classification*.

Penelitian untuk melihat performa Naïve Bayes dalam pengklasifikasian dokumen telah dilakukan oleh Ting [4]. Dalam penelitian tersebut, didapatkan hasil bahwa metode Naïve Bayes merupakan metode klasifikasi paling baik jika dibandingkan dengan metode lain seperti *decision tree*, *neural network* dan *support vector machines* dalam hal akurasi dan efisiensi komputasi. Penelitian yang dilakukan oleh Hamzah [5] yang mengkaji kinerja metode Naïve Bayes *Classifier* untuk kategorisasi teks berita dan teks akademik memberikan hasil bahwa algoritma Naïve Bayes *Classifier* memiliki kinerja yang baik untuk klasifikasi dokumen teks, baik dokumen berita maupun dokumen akademik.

Penelitian lain dilakukan oleh Isa dan Abidin [6] untuk mengukur tingkat kesamaan antar dokumen menggunakan algoritma *Vector Space Model* untuk mendeteksi plagiarisme. Hasilnya, algoritma *Vector Space Model* dapat mendeteksi dengan baik kesamaan dokumen melalui kesamaan paragraf dalam dokumen.

Penelitian ini akan mengklasifikasikan *feedback* pelanggan UPT Puskom UNS yang disampaikan melalui *mentions* Twitter dengan pendekatan *probabilistic* menggunakan algoritma Naïve Bayes *Classifier*. *Tweet* akan diklasifikasikan menjadi keluhan, berita dan spam. Algoritma Naïve Bayes *Classifier* telah digunakan secara luas karena kemudahannya, baik dalam proses pelatihan maupun klasifikasi [7]. Algoritma Naïve Bayes *Classifier* menjadi sangat efisien jika dilakukan berdasarkan *supervised learning* [8].

Dalam perhitungan menggunakan algoritma Naïve Bayes *Classifier*, sering ditemukan adanya perhitungan yang mengandung nilai peluang sama dengan 0, menyebabkan hasil perhitungan menjadi kurang akurat. Untuk menghindari munculnya peluang bernilai 0 pada algoritma ini, digunakan teknik Laplacian Smoothing.

Algoritma Cosine Similarity digunakan untuk mengelompokkan mentions keluhan yang memiliki term sama. Dari kelompok mentions tersebut, administrator akan memberikan solusi yang relevan terhadap keluhan. Selanjutnya ketika ada mentions baru berupa keluhan, sistem akan menampilkan rekomendasi solusi untuk kemudian dimoderasi oleh administrator. Dengan cara ini administrator tidak perlu memberikan jawaban yang sama secara berulang-ulang kepada akun yang berbeda. Hal ini akan mempercepat kinerja administrator dalam memberikan tanggapan terhadap keluhan konsumen.

2. LANDASAN TEORI

2.1. *TWITTER API*

Twitter API terdiri dari dua komponen yang berbeda, REST dan SEARCH API. REST API memungkinkan *developer* Twitter untuk mengakses data *core* Twitter (*tweet, timeline, user data*), sedangkan SEARCH API digunakan untuk membuat *query tweet* [9].

2.2. *TEXT MINING*

Text mining merupakan variasi dari *data mining* yang digunakan untuk menemukan pola tertentu dari sekumpulan besar data tekstual [10].

Langkah yang dilakukan dalam *text mining* adalah sebagai proses *text preprocessing*. Tindakan yang dilakukan pada tahap *text preprocessing* adalah *toLowerCase*, yaitu mengubah semua karakter huruf menjadi huruf kecil serta *tokenizing*, yaitu proses pemecahan kalimat menjadi token berupa kata atau *term*, dimana setiap *term* dipisahkan oleh delimiter. Tanda titik (.), koma (,), spasi ( ) dan karakter angka yang ada pada kalimat dapat dianggap sebagai delimiter [11].

2.3. JARO-WINKLER *DISTANCE*

Salah satu metode similaritas yang digunakan untuk mendeteksi kesamaan dua dokumen adalah Jaro *metric*. Dalam penelitian persamaan dokumen, didapatkan hasil yang baik dengan menggunakan metode Jaro, yang didasarkan pada jumlah dan urutan karakter yang sama antara dua dokumen [12].

Algoritma Jaro mendefinisikan 'karakter yang sama' sebagai karakter pada kedua *string* yang sama dan memenuhi ketentuan jarak teoritis [12]. Jarak teoritis dua buah karakter yang disamakan dapat dibenarkan jika tidak melebihi nilai persamaan di bawah ini.

$$\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1$$

Persamaan di bawah ini menunjukkan rumus untuk menghitung jarak (d_j) antara dua *string* yaitu s_1 dan s_2 pada algoritma Jaro.

$$d_j = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)$$

dimana:

m = jumlah karakter yang sama dan memenuhi kriteria
$|s_1|$ = panjang *string* 1
$|s_2|$ = panjang *string* 2
t = jumlah transposisi

Pengembangan dari algoritma Jaro berdasarkan Winkler menggunakan nilai panjang *prefix* yang sama di awal *string* dengan nilai maksimal adalah 4 (*l*) [13].

Persamaan di bawah ini menunjukkan nilai Jaro-Winkler *distance* (d_w) bila *string* s_1 dan s_2 yang diperbandingkan.

$$d_w = d_j + (lp(1-d_j))$$

dimana:

d_j = Jaro *distance* untuk *string* s1 dan s2
l = panjang *prefix* umum di awal *string* (panjang karakter yang sama sebelum ditemukan ketidaksamaan, maksimal 4)
p = konstanta *scaling factor*. Nilai standar untuk konstanta ini menurut Winkler adalah p = 0.1.

Semakin tinggi Jaro-Winkler *distance* untuk dua *string* berarti semakin mirip kedua *string* tersebut. Nilai terendah Jaro-Winkler *distance* adalah 0 yang menandakan tidak ada kesamaan antara kedua *string*. Nilai tertingginya adalah 1 yang menunjukkan kedua *string* sama persis [14].

2.4. NAÏVE BAYES *CLASSIFIER*

Naive Bayes Classifier merupakan klasifikasi yang berdasarkan pada teorema Bayes. *Naive Bayesian Classifier* mengasumsikan bahwa setiap atribut dalam

sebuah kelas merupakan atribut independen yang tidak terkait dengan atribut lain. Asumsi ini disebut *class conditional independence*. Naive Bayesian classifier memiliki tingkat akurasi dan kecepatan yang tinggi ketika diaplikasikan kepada data berjumlah besar [15].

Persamaan di bawah ini merupakan persamaan Naïve Bayes untuk klasifikasi dokumen.

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

dimana:

P(c|d) = *posterior*, yakni probabilitas dokumen d berada di kelas *c*,

P(c) = *prior*, yaitu probabilitas kelas *c* sebelum masuknya dokumen *d*,

$P(t_k|c)$ = *likelihood*, yaitu probabilitas kemunculan token t_k dalam kelas *c*,

n_d = jumlah token dalam dokumen *d* [15].

Dalam Naïve Bayes *Classifier*, dokumen *d* akan masuk ke dalam kelas *c* yang memiliki *maximum a posteriori* (MAP) atau kelas c_{map}, dihitung dengan persamaan sebagai berikut:

$$c_{map} = \arg\max_{c \in C} P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

dimana c adalah variabel kelas yang tergabung dalam himpunan kelas C [15].

2.5. *LAPLACIAN SMOOTHING*

Teknik *Laplacian Smoothing* digunakan untuk mengatasi nilai probabilitas kondisional pada Naïve Bayes *Classifier* yang dapat bernilai 0. Cara yang digunakan pada teknik ini adalah dengan menambahkan angka 1 pada perhitungan *Likelihood* [16].

Persamaan di bawah ini menunjukkan perhitungan nilai *Likelihood* untuk algoritma Naïve Bayes *Classifier*.

$$P(F_i|C) = \frac{1+n(F_i,C)}{|W| + n(C)}$$

dimana:

$n(F_i,C)$ = jumlah term F_i yang ditemukan di seluruh data pelatihan dengan kategori C

$n(C)$ = jumlah term di seluruh data pelatihan dengan kategori C

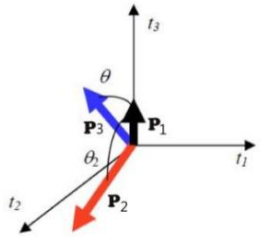
$|W|$ = jumlah seluruh term dari seluruh data pelatihan [16].

2.6. *VECTOR SPACE MODEL*

Representasi satu set dokumen sebagai vector dalam ruang vektor dikenal sebagai *Vector Space Model* (VSM) dan merupakan dasar untuk sejumlah operasi pengambilan informasi seperti penilaian dokumen dalam *query*, klasifikasi dan *clustering* dokumen [17].

Vector Space Model digunakan untuk mengukur kemiripan antara dua buah dokumen. Dokumen merupakan *vector* berdimensi *n*, sedangkan *t* adalah seluruh *term* yang ditemukan dalam *library* tanpa duplikasi [6].

Gambar 1 memperlihatkan tiga buah vector pada ruang dimensi 3. Nilai *cosinus* digunakan untuk mengukur kesamaan antara dua *vector*. Pada Gambar 1, P_1 adalah vektor dari dokumen pembanding, sementara P_2 dan P_3 adalah vektor dari dokumen yang dibandingkan.



Gambar 1. Vector Space Model [6]

2.7. PEMBOBOTAN TF x IDF

Term Frequency (TF) adalah jumlah kemunculan *term t* pada dokumen *d*, yang dirumuskan sebagai *freq(d, t)*. Matriks bobot *term frequency* atau TF(d,t) menunjukkan hubungan antara *term* t dengan dokumen d, dimana jika dokumen d tidak mengandung *term* t maka bobotnya bernilai 0, dan sebaliknya. Fungsi di bawah ini menunjukkan perhitungan nilai TF [15].

$$TF(d, t) = freq\ (d, t)$$

Document Frequency (DF) merupakan jumlah dokumen yang mengandung *term t. Inverse Document Frequency* (IDF) menunjukkan pembobotan dari *term* t. *Term* yang jarang muncul dalam dokumen memiliki nilai IDF yang tinggi, sementara *term* yang sering muncul dalam dokumen memiliki nilai IDF yang lebih rendah. Fungsi di bawah ini menunjukkan perhitungan nilai IDF [17].

$$IDF\ (t) = \log \frac{N}{df(t)}$$

Nilai TF-IDF dalam *Vector Space Model* dihitung dengan fungsi sebagai berikut **[*15*]**:

$$TF\text{-}IDF(d, t) = TF\ (d, t) \times IDF(t)$$

2.8. *COSINE SIMILARITY*

Untuk menghitung kesamaan antara kedua dokumen dalam *vector space*, maka akan dihitung nilai *cosine similarity* dari representasi vektor kedua dokumen [17].

$$Sim(P_1,P_2) = \cos\theta = \frac{P_1 \cdot P_2}{|P_1||P_2|}$$

Pada fungsi diatas, pembilang merepresentasikan nilai *dot product* dari kedua vektor, sedangkan penyebut merepresentasikan nilai perkalian dari Euclidean *length* kedua vektor. Nilai *dot product* dari kedua vektor dapat dicari dengan fungsi sebagai berikut [17]:

$$P_1 \cdot P_2 = \sum_{i=1}^{M} P_{1_i} P_{2_i}$$

Sedangkan nilai Euclidean *length* dari vector P dapat dicari dengan fungsi di bawah ini [17]:

$$|P| = \sqrt{\sum_{i=1}^{M} P_i^2}$$

Jika nilai *cosine similarity* dari kedua vector adalah 1 maka kedua dokumen adalah sama persis. Jika nilai *cosine similarity* adalah 0 maka dapat dikatakan bahwa kedua dokumen tidak sama.

2.9. *CONFUSION MATRIX*

Confusion matrix merupakan matriks yang menampilkan prediksi klasifikasi dan klasifikasi yang aktual. *Confusion matrix* berukuran LxL, dimana L adalah jumlah label klasifikasi yang berbeda. Tabel 1 menunjukkan *confusion matrix* untuk L=2 [18].

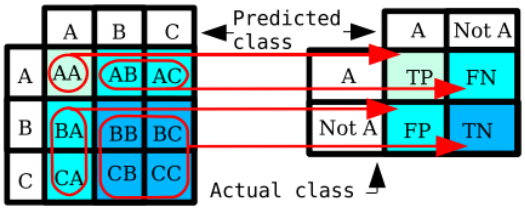
Tabel 1. Confusion Matrix untuk L = 2 [18]

Aktual \ Prediksi	Negatif	Positif
Negatif	a	b
Positif	c	d

Nilai akurasi didapatkan dari rumus di bawah ini:

$$\text{Akurasi} = \frac{a+d}{a+b+c+d} \text{ [18]}$$

Gambar 2 menunjukkan perubahan dari *extended confusion matrix* berukuran 3x3 menjadi berukuran 2x2, dengan kelas 'A' sebagai kelas positif dan kelas 'Not A' sebagai kelas negatif.



Gambar 2. *Extended confusion matrix* 3x3 [19]

3. METODOLOGI

Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

3.1. STUDI LITERATUR

Pada tahap ini, dilakukan studi literatur untuk mempelajari *text mining*, metode *Jaro-Winkler distance*, algoritma Naïve Bayes *Classifier* dan *Cosine Similarity.*

3.2. PENGUMPULAN DATA

Data diambil dari *mentions* yang dilakukan pihak luar terhadap Twitter milik UPT Puskom UNS sejak tanggal 17 Februari 2014 hingga 4 September 2014. Jumlah data yang diambil sebanyak 90 data.

3.3. *TEXT MINING*

Tahap *text mining* yang dilakukan pada penelitian ini adalah *text preprocessing*. Pada tahap ini, *mentions* diolah dengan cara mengubah seluruh abjad dalam *mentions* menjadi huruf kecil, menghapus kalimat *retweet* (RT), menghapus *username* dan *hashtag*, serta menghapus seluruh karakter selain abjad pada *mentions*. Selanjutnya *mentions* yang berupa *string* diubah menjadi bentuk token/*term*, yang dipisahkan oleh delimiter berupa spasi ( ).

3.4. KLASIFIKASI DENGAN NAÏVE BAYES *CLASSIFIER*

Data yang ada diklasifikasikan sebagai keluhan, berita dan spam menggunakan algoritma Naïve Bayes *Classifier*. Pertanyaan, keluhan dan sarkasme akan dimasukkan dalam kategori keluhan. Sementara itu, *mentions* kosong atau yang memiliki konten hanya berupa username akan diklasifikasikan sebagai spam.

Algoritma Naïve Bayes *Classifier* dilakukan berdasarkan proses *supervised learning*. Proses *supervised learning* pertama dilakukan terhadap 20 *mentions* awal. Selanjutnya ditambahkan 10 *mentions* untuk digunakan dalam proses *supervised learning* berikutnya hingga 80 *mentions* digunakan sebagai dokumen pelatihan.

Langkah-langkah yang dilakukan dalam *supervised learning* menggunakan algoritma Naïve Bayes *Classifier* adalah sebagai berikut:

1. Menghitung nilai *prior* setiap kategori
2. Menghitung frekuensi setiap *term* pada *mentions* untuk setiap kategori
3. Menghitung nilai *likelihood* setiap term pada *mentions* untuk setiap kategori
4. Menghitung nilai *posterior* setiap *mentions* untuk setiap kategori
5. Menentukan klasifikasi *mentions* berdasarkan nilai *posterior* tertinggi

3.5. PEMBAHARUAN *LIBRARY*

Setelah proses klasifikasi selesai dilakukan, *administrator* akan memberikan tinjauan terhadap hasil klasifikasi, dan mengubah hasil klasifikasi jika dianggap tidak sesuai.

Term dari seluruh *mentions* akan dibandingkan dengan seluruh *stop word* dengan menggunakan algoritma *Jaro-Winkler*. *Stop word* yang digunakan berasal dari penelitian yang telah dilakukan oleh Fadilla Z. Tala [20]. Jika *term* dari *mentions* memiliki kesamaan minimal 95% dengan *term* dari *stop word*, maka *term* tersebut dianggap sebagai *stop word*. Selanjutnya, sistem secara otomatis akan memasukkan seluruh *term* dari *mentions* selain *stop word* ke dalam *library* sesuai dengan kategori *mentions*.

3.6. PENGELOMPOKAN *MENTIONS* KELUHAN DENGAN *COSINE SIMILARITY*

Berikut ini merupakan langkah-langkah pengelompokan *mentions* keluhan yang bermakna sama dengan algoritma *Cosine Similarity*.

1. Menghitung nilai *term frequency* (TF) untuk setiap *term*
2. Menghitung nilai *index document frequency* (IDF) untuk setiap term
3. Menghitung bobot setiap *term*
4. Melakukan normalisasi
5. Menghitung panjang vektor
6. Menghitung nilai similaritas setiap *mentions*

3.7. PENENTUAN SOLUSI TERHADAP *MENTIONS* KELUHAN

Pada tahap ini, *administrator* akan memberikan solusi terhadap setiap kelompok *mentions* keluhan.

3.8. ANALISIS HASIL

Pada tahap ini dilakukan pengujian terhadap proses klasifikasi oleh algoritma Naïve Bayes *Classifier*. Setelah dilakukan *supervised learning* terhadap 20 dokumen, 10 dokumen selanjutnya digunakan sebagai proses pengujian. Selanjutnya setelah dilakukan *supervised learning* terhadap 30 dokumen, 10 dokumen selanjutnya digunakan sebagai proses pengujian. Proses pengujian dilakukan hingga 10 dokumen terakhir. Pengujian dilakukan dengan cara membandingkan hasil klasifikasi dokumen menggunakan algoritma Naïve Bayes *Classifier* dengan klasifikasi dokumen secara manual. Pengujian dilakukan untuk mengetahui tingkat akurasi hasil klasifikasi.

Pengujian hasil klasifikasi dalam penelitian ini menggunakan *extended confusion matrix*. Dokumen yang termasuk *False Positive* atau *error* tipe I adalah dokumen berita dan *spam* yang diklasifikasikan oleh sistem sebagai dokumen keluhan serta dokumen keluhan dan *spam* yang diklasifikasikan oleh sistem sebagai berita. Klasifikasi ini tidak menyebabkan kesalahan yang fatal karena dokumen keluhan dan berita akan cenderung dibaca oleh *administrator*.

Sedangkan dokumen keluhan dan berita yang diklasifikasikan oleh sistem sebagai *spam* dikategorikan sebagai *False Negative* atau *error* tipe II. *Error* tipe II menyebabkan kesalahan yang fatal karena dokumen keluhan dan berita tidak terbaca oleh *administrator* dikarenakan dokumen dengan klasifikasi *spam* cenderung diabaikan.

Gambar 3 menunjukkan pengaplikasian *extended confusion matrix* dalam penelitian ini.

Gambar 3. Aplikasi *extended confusion matrix* dalam penelitian

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="3">Sistem</td></tr>
<tr><td>Keluhan</td><td>Berita</td><td>Spam</td></tr>
<tr><td rowspan="3">Realita</td><td>Keluhan</td><td>True Positive</td><td>False Positive</td><td rowspan="2">False Negative (Type II Error)</td></tr>
<tr><td>Berita</td><td colspan="2" rowspan="2">False Positive (Type I Error)</td><td>True Positive</td></tr>
<tr><td>Spam</td><td>True Negative</td></tr>
</table>

4. HASIL DAN PEMBAHASAN

4.1. IMPLEMENTASI

Pada penelitian ini, berhasil dikumpulkan sebanyak 90 data yang berasal dari *mentions* terhadap Twitter milik UPT Puskom UNS. Selanjutnya, dilakukan proses *text preprocessing* terhadap data tersebut. Setelah proses *text preprocessing* selesai, dilakukan proses klasifikasi terhadap dokumen dengan algoritma Naïve Bayes *Classifier*.

Selanjutnya, *administrator* melakukan peninjauan terhadap hasil klasifikasi. Jika ada kategori yang tidak sesuai, *administrator* berhak mengubah kategori *mentions*.

Pada saat proses pengubahan kategori dilakukan, sistem akan mencocokkan *term* dari seluruh *mentions* dengan *term* yang termasuk *stop word* menggunakan algoritma Jaro-Winkler *distance*. Jika *term* dari *mentions* mirip dengan *term* dari *stop word*, dengan batas nilai minimal adalah 0.95, maka *term* tersebut akan dianggap sebagai *stop word*. Selanjutnya, sistem akan memasukkan *term* yang bukan merupakan *stop word* ke dalam *library* sesuai dengan kategori yang telah ditinjau oleh *administrator*.

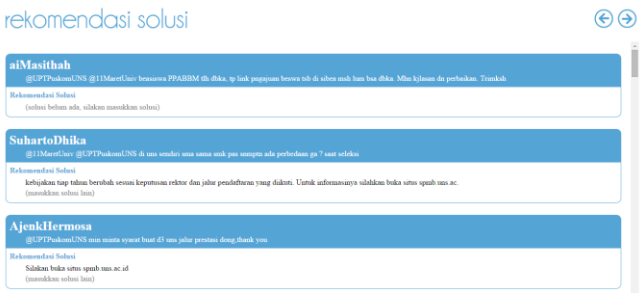
Setelah proses klasifikasi dengan algoritma Naïve Bayes *Classifier* selesai, maka dilakukan pengelompokan dokumen keluhan yang dianggap memiliki makna sama dengan algoritma *Cosine Similarity*. Hasil pengelompokan dokumen dengan algoritma *Cosine Similarity* menunjukkan bahwa dokumen keluhan tidak dapat terkelompokkan karena diantara data yang sangat terbatas yakni 29 data, tidak ada *mentions* yang memiliki *term* sama sehingga tidak ada pasangan *mentions* yang memiliki batas minimal nilai cosinus adalah 0.8. Karena alasan tersebut, maka dilakukan proses *self-test* untuk menguji algoritma *Cosine Similarity*, yaitu menguji dokumen dengan dokumen itu sendiri. Dokumen yang digunakan berjumlah 10 dokumen yang sama persis (dipilih secara acak) dan 5 dokumen yang mirip (dipilih secara acak). Dokumen yang mirip memiliki makna yang serupa namun sebagian kata telah diubah.

Tabel 3 menunjukkan hasil proses *Cosine Similarity* untuk *self-test*. ID_Dokumen dengan nilai awalan '0_' menunjukkan 10 dokumen yang sama persis, sedangkan ID_Dokumen dengan nilai awalan '1_' menunjukkan 5 dokumen yang mirip.

Tabel 3. Hasil proses *Cosine Similarity* untuk *self-test*

No	ID_ Dokumen 1	ID_ Dokumen 2	Nilai Cosinus	Ket.
1	0010	0_0010	1	Mirip
2	0017	0_0017	1	Mirip
3	0019	0_0019	1	Mirip
4	0025	0_0025	1	Mirip
5	0028	0_0028	1	Mirip
6	0030	0_0030	1	Mirip
7	0033	0_0033	1	Mirip
8	0034	0_0034	1	Mirip
9	0035	0_0034	1	Mirip
10	0036	0_0036	1	Mirip
11	1_45199758 9743734784	451997589 743734784	0.84	Mirip
12	1_49753811 4123206658	497538114 123206658	0.80	Mirip

Dari Tabel 2 dapat dilihat bahwa untuk dokumen yang sama persis, seluruh dokumen dapat dideteksi dengan nilai cosinus adalah 1. Sedangkan untuk dokumen yang mirip, terdapat 2 pasang dokumen yang terdeteksi karena batas minimal nilai cosinus dua dokumen yang dianggap mirip adalah 0.8. Sehingga untuk selanjutnya, rekomendasi solusi dapat diberikan pada dokumen yang mirip. Gambar 3 menunjukkan halaman pemberian rekomendasi solusi.



Gambar 3. Halaman pemberian rekomendasi solusi

Karena sifatnya yang berupa rekomendasi, maka administrator dapat melakukan perubahan atau update terhadap solusi yang diberikan untuk selanjutnya dikirimkan kepada user. Rekomendasi solusi yang diberikan dapat berjumlah lebih dari satu.

4.2. ANALISIS HASIL

Pada data klasifikasi dokumen secara manual, terdapat 29 dokumen dengan kategori keluhan, 57 dokumen dengan kategori berita dan 4 dokumen dengan kategori *spam*. Sedangkan setelah dilakukan proses klasifikasi dengan algoritma Naïve Bayes *Classifier*, didapatkan data klasifikasi dari setiap proses *supervised learning*. Dalam penelitian ini diasumsikan bahwa *administrator* memusatkan perhatian pada dokumen yang diklasifikasikan sebagai keluhan serta membaca dokumen yang diklasifikasikan sebagai berita. Sedangkan dokumen dengan klasifikasi *spam* cenderung untuk diabaikan, dimana dokumen ini termasuk dalam *False Negative*. *False Negative Rate* menunjukkan tingkat kesalahan fatal oleh sistem, dihitung dari jumlah dokumen yang termasuk *False Negative* dibagi dengan jumlah total seluruh dokumen.

Selanjutnya dilakukan perhitungan jumlah dokumen yang hasil klasifikasinya akurat, yakni dokumen yang termasuk *True Positive* dan *True Negative.* Nilai persentase akurasi dari setiap proses didapatkan dari persamaan di bawah ini.

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{N}} \times 100\% \text{ [18]}$$

Tingkat akurasi kategori dari setiap proses *training* dan *testing* ditunjukkan oleh Tabel 4.

Tabel 4. Tingkat akurasi dokumen pelatihan

Jumlah Data *Training*	Fase *Training*		Jumlah Data *Testing*	Fase *Testing*		Akurasi (%)	
	TP + TN	FP + FN		TP + TN	FP + FN	*Training*	*Testing*
20 data	20	0	10 data	6	4	100%	60%
30 data	26	4	10 data	6	4	86.67%	60%
40 data	36	4	10 data	7	3	90%	70%
50 data	46	4	10 data	8	2	92%	80%
60 data	58	2	10 data	9	1	96.67%	90%
70 data	69	1	10 data	9	1	98.57%	90%
80 data	79	1	10 data	8	2	98.75%	80%

10 data pertama yang digunakan untuk proses *testing* merupakan data yang muncul setelah dilakukan *supervised learning* terhadap 20 dokumen. 10 data kedua yang digunakan untuk proses pengujian merupakan data yang muncul setelah dilakukan *supervised learning* terhadap 30 dokumen. Sementara 10 data terakhir pada proses pengujian merupakan data yang muncul setelah dilakukan *supervised learning* terhadap 80 dokumen.

Pada proses awal pelatihan, tingkat akurasi cenderung tinggi, kemudian menurun. Selanjutnya, tingkat akurasi naik secara bertahap dan cenderung stabil. Hal ini mengindikasikan bahwa sistem dapat menerima proses pembelajaran yang diberikan.

Sedangkan pada proses pengujian, tingkat akurasi naik secara bertahap dan cenderung stabil. Namun demikian, pada proses pengujian setelah pelatihan sebanyak 80 data, tingkat akurasi mengalami penurunan sebesar 10%. Hal ini disebabkan *term* pada *mentions* belum muncul pada *mentions* sebelumnya yang digunakan untuk proses pembelajaran, sehingga sistem tidak dapat mengkategorikan *mentions* dengan baik. Sementara untuk pengujian dengan tingkat akurasi yang tinggi menunjukkan bahwa *term* pada *mentions* sudah terdapat pada *mentions* sebelumnya yang digunakan untuk proses pembelajaran, sehingga sistem dapat mengkategorikan *mentions* tersebut dengan baik.

5. KESIMPULAN DAN SARAN

Hasil penelitian menunjukkan bahwa dapat dilakukan proses klasifikasi pada mentions Twitter dengan algoritma Naïve Bayes *Classifier*. Pada proses pengujian, tingkat akurasi tertinggi yang berhasil dicapai adalah 90%, sedangkan tingkat akurasi terendah adalah 60%.

Mentions keluhan tidak dapat terkelompokkan dengan algoritma *Cosine Similarity* dikarenakan jumlah data yang sangat terbatas yakni 29 data dan tidak ada *mentions* yang memiliki *term* yang sama. Namun setelah dilakukan proses *self-test* dengan jumlah total 15 *mentions*, didapatkan hasil bahwa *mentions* yang mirip dapat terkelompokkan dengan algoritma *Cosine Similarity*.

Saran yang dapat dipertimbangkan untuk penelitian lebih lanjut antara lain:

1. Perlu dilakukan pengelompokan *mentions* dengan *Cosine Similarity* jika jumlah *mentions* cukup banyak dan *term* dalam *mentions* terbatas pada bidang tertentu saja.
2. Dapat mempertimbangkan *semantik* atau makna kata agar hasil klasifikasi yang didapatkan lebih akurat.

6. DAFTAR PUSTAKA

[1] Wijaya, T. 2008. "Pengaruh Kepuasan Pada Penanganan Keluhan dan Citra Perusahaan Terhadap Loyalitas Konsumen Natasha Skin Care". , vol. XIV, no. 1.

[2] Pitakasari, A. R. (2013, Oktober) Republika Online. [Online]. http://www.republika.co.id/berita/trendtek/internet/13/10/30/mvh7rm-penggunaan-internet-di-indonesia-95-persen-untuk-sosmed

[3] Devi, M. F.S. and Sanaji, 2013. "Penerapan Bauran Pemasaran Dalam Jaringan Melalui Media Sosial Untuk Membangun Hubungan Pelanggan". *Jurnal Ilmu Manajemen Volume 1 Nomor 5*, pp. 1314-1326.

[4] Ting, S. L. , Ip, W. H. , and Tsang, H. C. July 2011. "Is Naïve Bayes Classifier a Good Classifier for Document Classification?" *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 37-46.

[5] Hamzah, A. 2012. *Klasifikasi Teks Dengan Naive Bayes Classifier (NBC) Untuk Pengelompokan Teks Berita dan Abstract Akademis* in *Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III*, Yogyakarta, pp. 269-277.

[6] Isa, T. M. and Abidin, T. F. 2013. "Mengukur Tingkat Kesamaan Paragraf Menggunakan Vector Space Model untuk Mendeteksi Plagiarisme". *Seminar Nasional dan Expo Teknik Elektro*, pp. 229-234.

[7] Chakrabarti, S. , Roy, S. , and Soundalgekar, M.V. 2003. "Fast and Accurate Text Classification Via Multiple Linear Discriminant Projection". *The International Journal on Very Large Data Bases*, pp. 170-185.

[8] Aribowo, T. 2010. "Aplikasi Inferensi Bayes pada Data Mining terutama Pattern Recognition". Bandung,.

[9] Wardhani, Y. 2012. "Deteksi Spammer di Twitter dengan Mempelajari Tweet-Based Features". Surakarta,.

[10] Feldman, R. and Sanger, J. 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*.. New York: Cambridge University Press.

[11] Weiss, S. M. et al. 2005. *Text Mining: Predictive Methods for Analyzing Unstructured Information*.. New York: Springer.

[12] Jaro, M. A. 1989. "Advances In Record-Linkage Methodology As Applied To Matching The 1985 Censuf of Tampa, Florida". *Journal of The American Statistical Association*, pp. 414-420.

[13] Winkler, W. E. 1999. "The State of Record Linkage and Current Research Problems"..

[14] Kurniawati, A. , Puspitodjati, S. , and Rahman, S. 2010. "Implementasi Algoritma Jaro-Winkler Distance Untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia"..

[15] Han, J. and Kamber, M. 2006. *Data Mining: Concepts and Techniques, Second Edition*.. San Fransisco, United States of America: Morgan Kaufmann Publishers.

[16] Dai, W. et al. 2007. "Transferring Naive Bayes Classifiers for Text Classification"..

[17] Manning, C. D. , Raghavan, P. , and Schutze, H. 2009. *An Introduction to Information Retrieval*.. Cambridge, England: Cambridge University Press.

[18] Kohavi, R. and Provost, F. 1998. "Glossary of Terms". *Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, vol. 30, no. 2/3, pp. 271-274.

[19] Felkin, M. 2007. "Comparing Classification Results between N-ary and Binary Problems" in *Quality Measures in Data Mining*, GuilletFabrice J. and HamiltonHoward J. , Eds.: Springer Berlin Heidelberg, pp. 277-301.

[20] Tala, F. Z. 2003. "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia". Netherlands,.