Pengkategorian Pesan Singkat Berbahasa Indonesia pada Jejaring Sosial Twitter dengan Metode Klasifikasi *Naïve Bayes*

Rizal Setya Perdana[1], Suprapto, ST., MT.[2], Rekyan Regasari MP, ST., MT.[3]
Program Studi Teknik Informatika
Program Teknologi Informasi dan Ilmu Komputer
Universitas Brawijaya Malang
[1]rizalespe@ub.ac.id, [2]prapto_te@ub.ac.id, [3]rekyan.rmp@ub.ac.id

Abstract

Categorization system of short messages on Twitter (tweet) social network is the application of text mining to automatically classify a tweet into a particular category. The categorization of short messages on Twitter (tweet) help users to not get overwhelmed by reading the information in a state that is still random. The method used in the making of the application of categorizing tweets is Naive Bayes classification method. Naive Bayes is a method of probabilistic approaches in making inferences based on the Bayesian theorem in general. The data used in the training process of categorization derived from RSS (Really Simple Syndication) documents provided by certain websites and already have the previous category. The categories in this system are news, sports, finance, technology, entertainment, and automotive. The application will focus on the tweet in Bahasa which has certain morphological processing categorization. The application does some processing stages in the conduct, such as in the form of case folding preprocessing and parsing, transformation including removing stop words and stemming, calculating the frequency and probability and the calculation of Naive Bayes. Stemming method that is used specifically with the Indonesian language morphological results are used to obtain the frequency calculation of Naive Bayes classification. Categorization generated by the application compared to manual categorization has an average of 80% precision, 79% recall and F1 measure of 78%. The process of categorization stemming also affects the results in terms of both effectiveness and efficiency.

Keywords : Categorization, Classification, Naive Bayes, Twitter, Stemming, Bahasa Indonesia

Abstrak

Sistem pengkategorian pesan singkat pada jejaring sosial Twitter (*tweet*) merupakan penerapan dari *text mining* yang berusaha mengelompokkan secara otomatis sebuah *tweet* kedalam suatu kategori tertentu. Tujuan pengkategorian pesan singkat pada Twitter (*tweet*) membantu pengguna agar tidak kewalahan dengan membaca informasi *tweet* dalam kondisi yang masih acak. Metode yang digunakan dalam pembuatan aplikasi pengkategorian *tweet* adalah metode klasifikasi Naïve Bayes. Metode ini melakukan pendekatan probabilistik dalam melakukan inferensi yakni berbasis teorema Bayes secara umum. Data latih yang digunakan pada proses pengkategorian didapat dari dokumen RSS (*Really Simple Syndication*) yang disediakan oleh website tertentu dan sudah memiliki kategori sebelumnya. Kategori-kategori yang terdapat pada sistem adalah berita, olahraga, keuangan, teknologi, hiburan, dan otomotif. Aplikasi akan fokus pada *tweet* berbahasa Indonesia, dimana bahasa Indonesia mempunyai morfologi tertentu dalam pemrosesan pengkategorian. Aplikasi melakukan beberapa tahapan dalam melakukan pemrosesan diantaranya adalah *preprocessing* berupa *case folding*, dan *parsing*, *transformation* berupa penghapusan *stopwords* dan *stemming*, penghitungan frekuensi dan probabilitas dan perhitungan Naïve Bayes. Metode *stemming* yang digunakan khusus menangani morfologi bahasa Indonesia yang hasilnya digunakan dalam mendapatkan frekuensi dalam perhitungan klasifikasi Naïve Bayes. Pengkategorian yang dihasilkan oleh aplikasi dibandingkan dengan pengkategorian manual mempunyai rata-rata *precision* sebesar 80%, *recall* 79% dan F_1 *measure* sebesar 78%. Proses *stemming* juga mempengaruhi hasil pengkategorian baik dari segi efektifitas maupun efisiensi.

Kata kunci : Pengkategorian, Klasifikasi, *Naive Bayes*, Twitter, *Stemming*, Bahasa Indonesia

1. PENDAHULUAN

Twitter merupakan aplikasi jejaring sosial yang memungkinkan bagi pengguna untuk berbagi informasi dalam bentuk teks-teks pendek (140 karakter) [1:120]. Informasi yang datang secara acak pada pengguna dengan jumlah yang banyak dalam waktu yang singkat membuat pengguna mengalami kewalahan dalam mendapatkan informasi yang dibutuhkannya. Tujuan dari penelitian adalah untuk mengkategorikan *tweet* atau pesan singkat secara otomatis ke dalam kategori yang berbeda sehingga pengguna tidak kewalahan dengan membaca informasi *tweet* dalam kondisi yang masih acak [2:841].

Berbagai metode telah diterapkan dan masih terus dikembangkan oleh para peneliti di seluruh dunia. Salah satu metode yang banyak digunakan untuk melakukan pengkategorian dokumen dengan metode klasifikasi adalah *Naive Bayes.* Algoritma *Naive Bayes* tidak rumit dan efektif dalam klasifikasi teks [4:699].

Pada pengkategorian teks singkat otomatis pada jejaring sosial Twitter diberikan beberapa kategori yaitu berita umum, keuangan, olahraga, hiburan, teknologi, dan otomotif yang berbahasa Indonesia. Bahasa Indonesia merupakan bahasa yang mempunyai morfologi yang berbeda dengan bahasa Inggris sehingga dalam proses ekstraksi kata menggunakan metode *stemming* untuk bahasa Indonesia.

2. DASAR TEORI

Twitter

Twitter adalah layanan jejaring sosial yang memungkinkan para pengguna untuk berbagi informasi dalam bentuk pesan teks singkat sejumlah 140 karakter [1:120]. Dengan Twitter seoarang pengguna dapat ber-*microblog* tentang berbagai macam topik [2:1]. Elemen-elemen yang dimiliki oleh Twitter mirip dengan karakter yang dimiliki email, IM, *texting*, *blogging*, RSS dan lain-lain [5:7].

Beberapa faktor yang menyebabkan Twitter berbeda dan memiliki keunikan adalah :

1. Pesan yang dikirim dan diterima pada Twitter tidak lebih dari 140 karakter atau setara dengan panjang dari kepala berita.
2. Pesan yang ada pada Twitter bersifat publik, seperti tulisan yang ada pada sebuah *blog* dimana tidak ada batasan siapa saja yang akan membaca tulisan yang ada di dalamnya.
3. Pesan yang ada pada *timeline* Twitter akan berbeda antara pengguna satu dengan pengguna yang lainnya, hal ini karena seorang pengguna dapat memilih pengguna mana yang hendak diikuti pesan yang ditulis. Twitter memberi istilah untuk hal tersebut dengan sebutan *following.*
4. Pesan singkat pada Twitter dapat dikirim dan diterima melalui banyak media dan juga mekanisme, seperti menggunakan perangkat *mobile,* komputer (PC), aplikasi web dan *desktop.*

Text Mining

Klasifikasi dokumen pesan singkat pada jejaring sosial Twitter menjadi beberapa kategori adalah satu contoh aplikasi dari *text mining.* Beberapa contoh lain aplikasi dari *text mining* adalah *text summarization, text categorization, document clustering, language identification, ascribing authorship* [8].

Text mining atau dengan sebutan lain seperti *intelligent text analysis, text data mining,* atau *knowledge discovery in text* secara sederhana dapat diartikan sebagai proses penemuan pola yang sebelumnya tidak terlihat pada dokumen teks atau sumber tertentu [6:183].

Tahapan *Text Mining*

Tahapan yang dilakukan di dalam *text mining* adalah proses awal terhadap teks (*text preprocessing*), transformasi teks ke dalam bentuk perantara (*text transformation/future generation*), dan penemuan pola (*pattern discovery*) [7:26]. Pada sistem akan melakukan pengolahan terhadap data masukan berupa data teks dari pesan singkat Twitter dan menghasilkan keluaran berupa pola sebagai hasil interpretasi.

Text Preprocessing

Tahapan awal dari proses *text mining* adalah *text preprocessing* yang bertujuan untuk mempersiapkan dokumen-dokumen teks menjadi data yang siap untuk mengalami proses pengolahan pada tahapan selanjutnya. Tindakan yang dilakukan pada tahap *preprocessing text* atau *text normalization* adalah [8:28] :

1. *Part Of Speech Tagging* atau *Grammatical Tagging* adalah proses penandaan pada kalimat dengan membagi menjadi kategori-kategori dalam tata bahasa seperti kata kerja, kata benda, kata sifat, kata depan, dan sebagainya [10:1].
2. *Parsing* atau pemecahan kalimat menjadi sekumpulan kata-kata.
3. *Case Folding* atau pengubahan karakter huruf menjadi huruf kecil [11:2].

Text Transformation (feature generation)

Pada tahapan ini akan mengolah teks yang sudah melalui tahapan sebelumnya yaitu tahap *text preprocessing* yang akan dilakukan proses transformasi. Tindakan yang dilakukan pada proses transformasi ini adalah [7:29] :

1. *Stopword removal* atau penghapusan kata-kata yang sering muncul dan kata-kata tersebut tidak mempengaruhi makna dari keseluruhan kalimat [9:1].
2. *Stemming* adalah proses mengubah kata-kata ke dalam bentuk atau morfologi dasarnya [12:1]. Pengertian lain *stemming* adalah suatu proses yang menyediakan suatu pemetaan anatara berbagai kata dengan morfologi yang berbeda menjadi satu bentuk dasar (*stem*) [13:1].

Pattern Discovery

Tahap inti dari seluruh proses *text mining* adalah tahap penemuan pola atau *pattern discovery*. Tahap *pattern discovery* berusaha menemukan pola atau pengetahuan yang terkandung dari keseluruhan teks. *Knowledge discovery* adalah proses ekstraksi terhadap informasi implisit pada sejumlah kumpulan data besar yang sebelumnya tidak diketahui dan berguna bagi pengguna [14:2].

Terdapat dua teknik pembelajaran pada tahap *pattern discovery* dalam *data mining* atau *text mining* yaitu *unsupervised* dan *supervised learning*. Pada metode *unsupervised* tidak ada variabel target yang dituju, sedangkan pada metode *supervised* terdapat variabel target dan algoritma diberikan beberapa contoh nilai data yang digunakan sebagai pembelajaran [15:90].

Perbedaan diantara kedua metode yaitu pada *supervised learning* terdapat label atau nama kelas pada data latih (supervisi) dan data baru diklasifikasikan berdasarkan data latih. Sedangkan pada *unsupervised learning* tidak terdapat label atau nama kelas pada data latih, data latih dikelompokkan berdasarkan ukuran kemiripan pada suatu kelas.

Supervised machine learning adalah algoritma pencarian yang berasal dari contoh data yang diolah untuk menghasilkan hipotesis secara umum, kemudian hipotesis yang dihasilkan digunakan untuk mempresiksi kasus baru [16:249].

Ada beberapa metode untuk melakukan proses *supervised learning* sebagai tahapan dalam *pattern discovery,* namun yang akan dibahas pada bagian selanjutnya dalam tugas akhir ini hanya metode *naive bayes classifier*. Penggunaan algoritma *Naive Bayes* dipilih karena tidak rumit dan efektif dalam klasifikasi teks [4:699].

***Stemming* pada Bahasa Indonesia**

Struktur morfologi kata bahasa Indonesia

Secara etimologi kata morfologi berasal dari kata *morf* yang berarti bentuk dan kata *logi* yang berarti ilmu. Jadi secara harafiah kata morfologi bebrarti ilmu bentuk. Di dalam kajian linguistik, morfologi berarti ilmu mengenai bentuk-bentuk dan pembentukan kata [18:3].

Morfologi adalah bagian dari ilmu bahasa yang membicarakan atau mempelajari seluk-beluk bentuk kata serta pengaruh perubahan-perubahan bentuk kata terhadap golongan dan arti kata, atau dengan kata lain dapat dikatakan bahwa morfologi mempelajari seluk-beluk bentuk kata serta fungsi perubahan-perubahan bentuk kata itu, baik fungsi gramatik maupun semantik [17:19].

Metode *Stemming* Arifin dan Setiono

Algoritma *stemming* akan mengolah tiap kata yang dimasukkan untuk diproses dan menghasilkan kata dalam bentuk dasarnya. Langkah-langkah yang dilakukan dalam algoritma *stemming* bahasa Indonesia Arifin dan Setiono adalah sebagai berikut [19:2] :

1. Pemeriksaan semua kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 Awalan (prefiks) dan 3 Akhiran (sufiks) [19:2]. Sehingga bentuknya menjadi :

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Seandainya kata tersebut tidak memiliki imbuhan sebanyak imbuhan di atas, maka imbuhan yang kosong diberi tanda x untuk prefiks dan diberi tanda xx untuk sufiks [19:3].

2. Pemotongan dilakukan secara berurutan sebagai berikut :

AW : AW (Awalan)
AK : AK (Akhiran)
KD : KD (Kata Dasar)

a. AW I, hasilnya disimpan pada p1 (prefiks 1)
b. AW II, hasilnya disimpan pada p2 (prefiks 2)
c. AK I, hasilnya disimpan pada s1 (sufiks 1)
d. AK II, hasilnya disimpan pada s2 (sufiks 2)
e. AK III, hasilnya disimpan pada s3 (sufiks 3)

Pada setiap tahap pemotongan di atas diikuti dengan pemeriksaan di kamus apakah hasil pemotongan itu sudah berada dalam bentuk dasar. Kalau pemeriksaan ini berhasil maka proses dinyatakan selesai dan tidak perlu melanjutkan proses pemotongan imbuhan lainnya [19:3]. Contoh pemenggalan kata “mempermainkannya” .

a. Langkah 1 :
Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AW I
Kata = permainkannya

b. Langkah 2 :
Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AW II
Kata = mainkannya

c. Langkah 3 :
Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AK I
Kata = mainkan

d. Langkah 4 :
Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AK II
Kata = main

e. Langkah 5 :
Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AK III. Dalam hal ini AK III tidak ada, sehingga kata tidak diubah.
Kata = main

f. Langkah 6
Cek apakah kata ada dalam kamus
Ya : Success
Tidak : "Kata tidak ditemukan"

3. Namun jika sampai pada pemotongan AK III, belum juga ditemukan di kamus, maka dilakukan proses kombinasi[19:3]. KD yang dihasilkan dikombinasikan dengan imbuhan-imbuhannya dalam 12 konfigurasi berikut:

a. KD
b. KD + AK III
c. KD + AK III + AK II
d. KD + AK III + AK II + AK I
e. AW I + AW II + KD
f. AW I + AW II + KD + AK III
g. AW I + AW II + KD + AK III + AK II
h. AW I + AW II + KD + AK III + AK II + AK I
i. AW II + KD
j. AW II + KD + AK III
k. AW II + KD + AK III + AK II
l. AW II + KD + AK III + AK II + AK I

Sebenarnya kombinasi a, b, c, d, h, dan l sudah diperiksa pada tahap sebelumnya, karena kombinasi ini adalah hasil pemotongan bertahap tersebut[ARI-01]. Dengan demikian, kombinasi yang masih perlu dilakukan tinggal 6 yakni pada kombinasi-kombinasi yang belum dilakukan (e, f, g, i, j, dan k). Tentunya bila hasil pemeriksaan suatu kombinasi adalah 'ada', maka pemeriksaan pada kombinasi lainnya sudah tidak diperlukan lagi [19].

Pemeriksaan 12 kombinasi ini diperlukan, karena adanya fenomena *overstemming* pada algoritma pemotongan imbuhan. Kelemahan ini berakibat pada pemotongan bagian kata yang sebenarnya adalah milik kata dasar itu sendiri yang kebetulan mirip dengan salah satu jenis imbuhan yang ada. Dengan 12 kombinasi itu, pemotongan yang sudah terlanjur tersebut dapat dikembalikan sesuai posisinya [19:3].

Naive Bayes Classifier

Teori keputusan Bayes adalah pendekatan statistik yang fundamental dalam pengenalan pola (*pattern recognition*) [20:74]. *Naive bayes classifier* termasuk ke dalam algoritma pembelajaran bayes yang dibangun oleh data pelatihan untuk memperkirakan probabilitas dari setiap kategori yang terdapat pada ciri dokumen yang diuji [21:3]. Sistem akan dilatih dengan menggunakan data latih lengkap berupa pasangan nilai-nilai atribut dan nilai target kemudian sistem akan diberikan data baru dan sistem diberi tugas untuk menebak nilai fungsi target dari data tersebut [23:177].

Persamaan pengkategorian dokumen menggunakan *Naive bayes* adalah sebagai berikut [22:251] :

$$P(kategori \mid kata) = \frac{P(kata \mid kategori)P(kategori)}{P(kata)} \quad \text{..... (1)}$$

keterangan : P (peluang)

Naive bayes classifier memberi nilai target kepada data baru menggunakan nilai V_{MAP}, yaitu nilai kemungkinan tertinggi dari seluruh anggota himpunan set domain V [23:177].

$$V_{MAP} = \underset{v_j \in V}{\text{argmax}}\, P(v_j \mid a_1,a_2,a_3,...,a_n) \quad \text{.....(2)}$$

Teorema bayes kemudian digunakan untuk menulis ulang persamaannya ditulis menjadi 2 menjadi persamaan 3 sebagai berikut :

$$V_{MAP} = \underset{v_j \in V}{\text{argmax}} \frac{P(a_1,a_2,a_3,...,a_n \mid V_j)P(V_j)}{P(a_1,a_2,a_3,...,a_n)} \quad \text{.....(3)}$$

Pada pengklasifikasian teks, perhitungan rumus dapan didefinisikan [23:182] :

$$P(v_j) = \frac{docs_j}{examples} \quad \text{.....(4)}$$

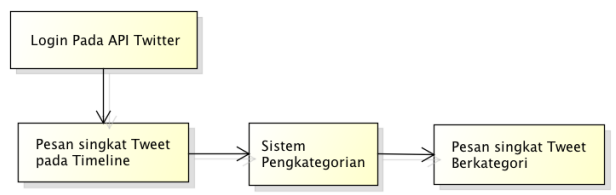
$$P(w_k \mid v_j) = \frac{n_k+1}{n+|kata|} \quad \text{.....(5)}$$

Keterangan :
1. $docs_j$: kumpulan dokumen yang memiliki nilai target v_j

2. examples : jumlah dokumen yang digunakan dalam pelatihan (kumpulan data latih)
3. n : jumlah total kata yang terdapat di dalam data tekstual yang memiliki nilai fungsi target yang sesuai.
4. n_k : jumlah kemunculan kata w_k pada semua data tekstual yang memiliki nilai fungsi target yang sesuai.
5. |kata| : jumlah kata yang berbeda yang muncul dalam seluruh data tekstual yang digunakan.

3. METODOLOGI

Perancangan dan desain sistem



Gambar 1. Diagram Blok Sistem Pengkategorian

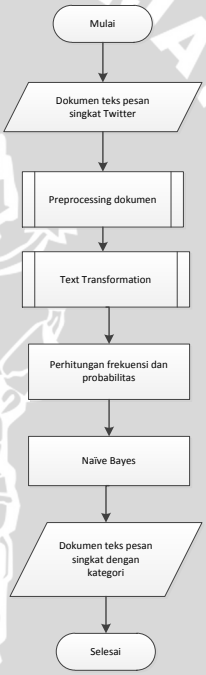
a. Pengguna Twitter *login* masuk pada aplikasi menggunakan *username* dan *password* akun Twitter pada halaman login OAuth Twitter, apabila proses *login* sukses maka pengguna akan masuk pada sistem aplikasi pengkategorian.
b. Pengguna yang sudah berhasil masuk ke dalam sistem akan melihat *timeline* yang masih belum terkategori. Pesan singkat atau Tweet yang ditampilkan berasal dari *request* API yang disediakan oleh Twitter.
c. Pada sistem terdapat menu untuk melakukan pemilihan kategori mana yang akan ditampilkan, pada proses ini pesan singkat akan diolah oleh sistem pengkategorian menggunakan algoritma Naive Bayes.
d. Keluaran dari sistem aplikasi ini adalah Tweet yang sudah memiliki kategori tertentu yang sudah memiliki label kategori tertentu.

Pembangunan Arsitektur Aplikasi

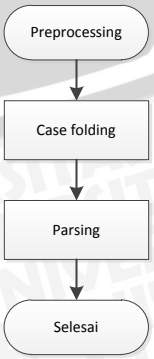
Adapun arsitektur yang didapat adalah sebagai berikut :

1. Pengguna akan masuk pada aplikasi Twitter *client* yang terhubung secara langsung melalui API yang disediakan oleh Twitter dnegan *user interface* berupa aplikasi web, pada tampilan awal pengguna akan melihat *timeline* yang berisi pesan singkat pada halaman beranda.
2. Sistem melakukan tahapan-tahapan *preprocessing* dimana pada proses ini dilakukan *parsing, case folding, stopword removal* dan *stemming*. Proses akan dilanjutkan pada tahap berikutnya dengan menggunakan kata-kata yang dianggap penting saja.
3. Pada proses yang terpisah dari aplikasi Twitter *client* yang dijelaskan sebelumnya, data latih yang bersumber dari dokumen RSS diproses sama seperti tahap ke dua. Dokumen RSS yang telah memiliki label berupa kategori yang berasal dari sumber berita diproses *text preprocessing* untuk selanjutnya dibuat tabel frekuensi kemunculan kata disertai perhitungan probabilitas.
4. Aplikasi Twitter *client* yang menampilkan pesan singkat melakukan perhitungan berdasarkan pada data hasil perhitungan oleh data latih, sehingga pesan singkat yang ada pada Twitter akan memiliki kategori sesuai dengan perhitungan menggunakan metode *naive bayes*.

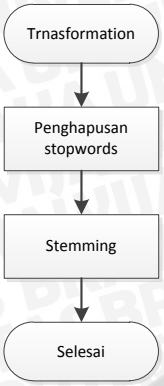
Untuk mempermudah alur jalan dari sistem, maka semua proses yang dijelaskan pada tahap perancangan ini digambarkan dalam diagram alir berikut :



Gambar 2. Diagram Alir Keseluruhan Sistem



Gambar 3. Diagram Alir *Preprocessing*



Gambar 4. Diagram Alir *Transformation*

a. *Case folding* dan *Parsing*

Case folding adalah mengubah semua huruf dalam teks menjadi huruf kecil. Sedangkan proses *parsing* sederhana yang dilakukan adalah memecah sebuah teks menjadi kumpulan kata-kata tanpa memperhatikan keterkaitan antar kata dan peran atau kedudukannya dalam kalimat.

b. Penghapusan *Stopword*

Penghapusan *stopword* dilakukan dengan menghapus kata-kata yang dianggap tidak mempengaruhi maksud dari isi atau makna dari keseluruhan kalimat.

c. *Stemming*

Dalam bahasa Indonesia imbuhan terdiri dari sufiks (akhiran), infiks(sisipan), dan prefiks (awalan). *Stemming* merupakan proses untuk mendapatkan kata dasar dari setiap kata yang ada pada dokumen. Proses ini bertujuan untuk mendapatkan kata penting pada kalimat. Proses *stemming* melakukan proses perubahan dari kata-kata yang ada menjadi kata dasar yang nantinya digunakan dalam pembobotan kalimat

d. Perhitungan frekuensi dan probabilitas

Pada proses ini akan menghitung jumlah kemunculan kata secara komulatif, sehingga kata yang mempunyai frekuensi tinggi pada sebuah dokumen akan dianggap sebagai kata penting pada dokumen tersebut.

e. *Naive Bayes*

- *Learn naive bayes*

  Pada tahap *learn naive bayes* sistem akan melakukan proses pembentukan pengetahuan yang berasal dari data latih. Pengetahuan yang nantinya dihasilkan dari proses akan digunakan pada proses *classify naive bayes.*

- *Classify naive bayes*

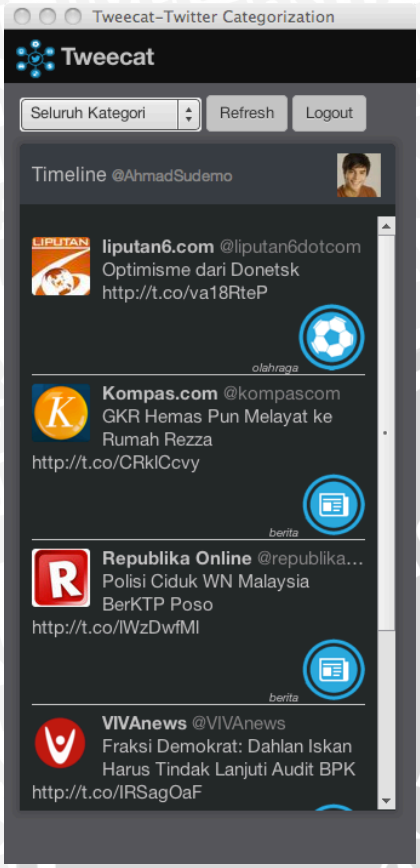
  Pada tahap ini dilakukan proses perhitungan data uji berdasarkan pengetahuan yang akan menghasilkan perkiraan atau estimasi kecenderungan kategori yang dimiliki oleh data uji. *Classify naive bayes* berusaha mencari nilai probabilitas tertinggi untuk mengklasifikasikan data uji pada kategori yang paling tepat.

4. Implementasi

Implementasi yang diberikan memiliki batasan-batasan sebagai berikut :

1. Dokumen yang diproses adalah pesan singkat pada jejaring sosial *Twitter* berbahasa Indonesia.
2. Kategori yang dipakai adalah berita, keuangan, olahraga, hiburan, teknologi, dna otomotif.
3. Data pelatihan yang digunakan berasal dari dokumen RSS.
4. Metode yang digunakan untuk proses *stemming* adah Metode Arifin.
5. Metode yang digunakan untuk pengklasifikasian dokumen adalah metode Naive Bayes.
6. Sistem tidak memperhatikan sinonim, kesalahan ejaan kata, dan sisipan.
7. Implementasi dilakukan dengan bahasa pemrograman PHP sebagai pemroses, DBMS menggunakan MySQL, dan aplikasi *client* menggunakan Adobe AIR.
8. Sistem pemroses menggunakan *framework* PHP yaitu Codeigniter.
9. Aplikasi dapat berjalan baik diimplementasikan pada perangkat keras dengan spesifikasi prosesor Intel Core 2 Duo, RAM 2GB, Hardisk 160 GB, Sistem Operasi Mac OSX Snow Leopard, serta dengan bantuan *tools* pemrograman Aptana dan Squel Pro.

Pengguna akan melakukan proses *login* pada halaman dialog *login,* selanjutnya pengguna akan melihat halaman *home timeline* yang terdiri dari beberapa dokumen *tweet* yang sudah memiliki kategori tertentu. Pengguna juga dapat memilih kategori apa saja yang dapat dilihat pada *home timeline.*



Gambar 5. Tampilan Aplikasi *Client* pada saat Mengkategorikan

5. Pengujian dan Analisa

Pengujian dilakukan dengan melihat hasil pengkategorian sistem dengan pengkategorian secara manual, pengujian efektifitas terhadap sistem ini akan menggunakan metode *precision, recall,* dan F_1 *measure*. Performa dari identifikasi topik sebuah dokumen biasanya diukur menggunakan *precision and recall scores* [26]. F_1 *measure* merupakan gabungan antara *recall* dan *precision* [27:43]. Pengujian dilakukan dengan dua macam pengolahan yaitu dengan melalui proses *stemming* dan tanpa menggunakan proses *stemming*.

Untuk mempelajari pengaruh jumlah data latih terhadap efektifitas sistem klasifikasi maka dilakukan 10 kali uji coba dengan jumlah data latih yang berbeda dengan beragam proporsi yang proporsional. Pembagian proporsi dilakuan dilakukan dengan menggunakan metode *proportionate stratified sampling* pada pengujian sistem klasifikasi, dari 37555 akan dibagi menjadi beberapa bagian. Pembagian jumlah data latih akan tampak seperti pada tabel 1 :

Tabel 1. Tabel Pembagian Jumlah Data Latih

Stage	Olahraga	Teknologi	Hiburan	Keuangan	Berita	Otomotif	Jumlah
1	53	27	49	61	162	23	375
2	265	137	246	303	810	116	1877
3	530	270	490	610	1620	234	3754
4	753	381	713	850	2332	335	5364
5	1506	762	1426	1700	4664	670	10728
6	2259	1143	2139	2550	6996	1005	16092
7	3012	1524	2852	3400	9329	1340	21457
8	3765	1905	3565	4250	11662	1675	26822
9	4518	2286	4278	5100	13995	2010	32187
10	5271	2667	4991	5950	16328	2345	37552

Selain untuk menguji efektifitas sistem, juga akan dilakukan evaluasi terhadap efisiensi dalam pengkategorian. Pada evaluasi pengklasifikasian atau pengkategorian teks lebih mengutamakan hasil keefektifannya daripada efisiensinya [24:32]. Pengujian pengkategorian akan dilakukan dengan melalui proses *stemming* dan pengujian pengkategorian tanpa menggunakan *stemming*. Pengujian pengkategorian sistem ini diharapkan bisa mengetahui pengaruh metode *stemming* yang diterapkan pada sistem. Selain itu pengujian dilakukan untuk melihat efisiensi dari segi kecepatan saat melakukan pengkategorian. Pengujian ini akan menguji 20 pesan singkat pada jejaring sosial Twitter.

Pada perhitungan hasil pengujian waktu eksekusi pengkategorian yang dapat dilihat adalah bahwa waktu yang digunakan untuk melakukan pengkategorian cenderung meningkat sebanding dengan jumlah data latih yang ada. Hal ini mengindikasikan semakin banyak jumlah data latih yang kita gunakan maka akan semakin lama pula waktu yang dipakai. Selain itu pada pengujian juga menunjukkan bahwa dengan menggunakan proses *stemming* dan tanpa menggunakan *stemming* juga mempengaruhi lama waktu yang dibutuhkan. Dengan menggunakan proses *stemming* maka waktu yang dibutuhkan juga lebih besar dibandingkan dengan tanpa menggunakan proses *stemming*.

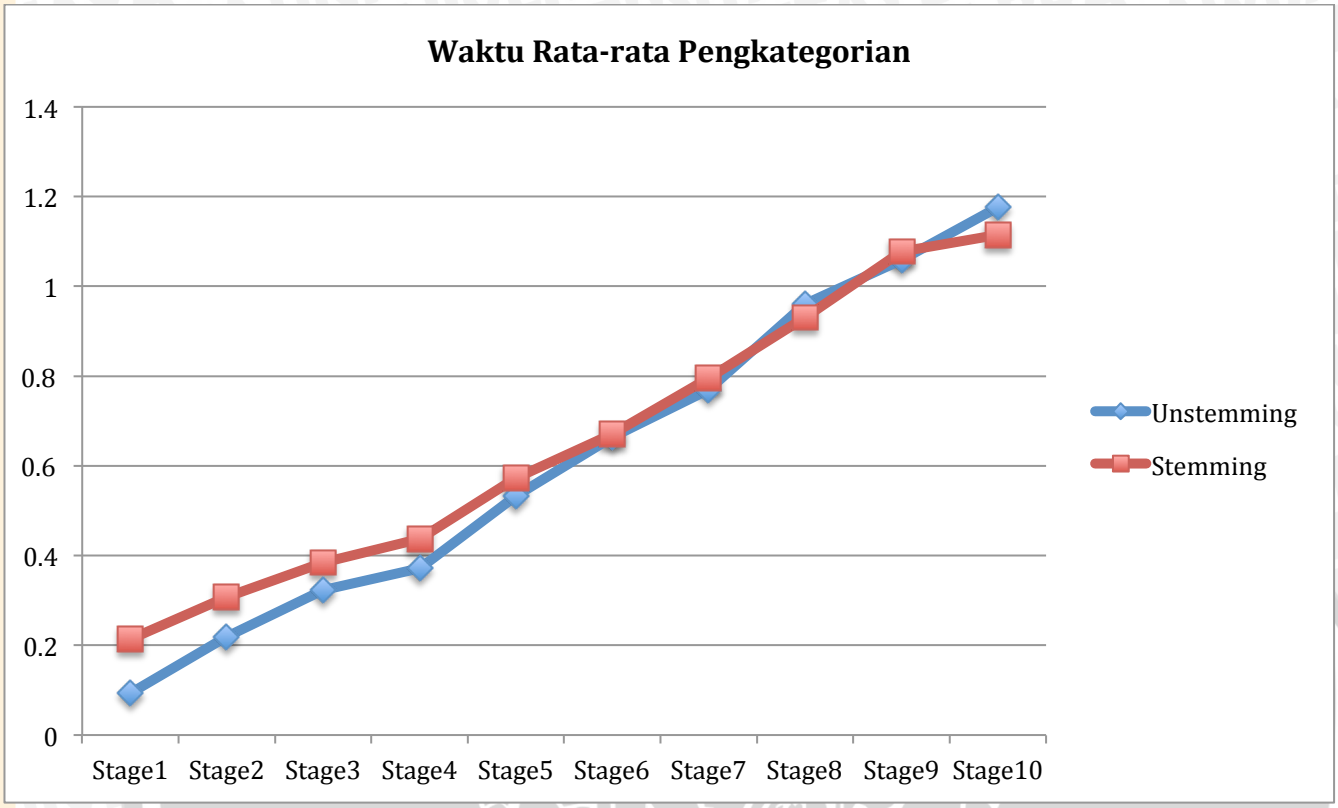
Tabel 2. Waktu Eksekusi Proses Pengkategorian dengan Melalui Proses *Stemming* dalam *Microtime*

No	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6	Stage 7	Stage 8	Stage 9	Stage 10
1	0.380875826	0.495334864	0.486974955	0.80457902	0.890140057	1.349081993	1.994889975	1.080045938	2.587732077	1.595799208
2	0.158480883	0.302911043	0.43373704	0.453613043	0.631949186	0.736500025	0.867233038	1.29795289	1.208107948	1.236124039
3	0.195596933	0.282857895	0.34947896	0.378453016	0.479984999	0.573885918	0.819530964	0.92310214	1.106094122	1.145862103
4	0.111557007	0.203722	0.256303072	0.296905994	0.384220839	0.466771841	0.551025867	0.750558138	0.791178942	0.85367918
5	0.389162779	0.573404074	0.7116642	0.772683859	1.251354933	1.172656059	1.314715862	1.595240116	1.727721214	1.822994947
6	0.205226898	0.306568861	0.382608891	0.42048502	0.551079035	0.620584011	0.693581104	0.876487017	0.944509029	1.064821959
7	0.220536947	0.290689945	0.369382143	0.431576967	0.52304697	0.606446028	0.67989707	0.873771906	0.979186058	1.064378977
8	0.106671095	0.174672127	0.248872042	0.251471043	0.330665827	0.410394907	0.462754011	0.619591951	0.685060978	0.763442993
9	0.246085882	0.319749117	0.390027046	0.446881056	0.55072093	0.629063845	0.70930481	0.91237402	0.95373702	1.090848923
10	0.108963966	0.194656134	0.254520893	0.295423985	0.382590055	0.470691919	0.533288956	0.69203496	0.775178909	0.855981827
11	0.334915161	0.233720064	0.306277037	0.35472393	0.487961054	0.587561131	0.691640139	0.844572067	0.93995595	1.032112122
12	0.234207869	0.328935862	0.441586018	0.487590075	0.621288061	0.717251062	0.811202049	1.008692026	1.08319211	1.08319211
13	0.0873909	0.192695141	0.289079189	0.295114994	0.401074886	0.498782158	0.586681128	0.744896889	0.836349964	0.918282986
14	0.215817213	0.344927073	0.430009127	0.465379	0.596786022	0.689025879	0.798864126	0.955535889	1.042270899	1.158414125
15	0.216673136	0.30616498	0.394946814	0.427552938	0.536221027	0.640594006	0.698172092	0.895210028	0.895210028	1.105308056
16	0.184789896	0.295418024	0.363331079	0.405123949	0.624977112	0.660414934	0.763304949	0.917860985	1.026893139	1.108390093
17	0.368249893	0.485473156	0.581197023	0.616317034	0.747631073	0.859932899	0.967638016	1.117954016	1.248214006	1.357317925
18	0.246326923	0.395425081	0.461127996	0.51794219	0.641763926	0.759381056	0.856395006	1.049167156	1.124095917	1.233060122
19	0.116266966	0.189893961	0.243769169	0.291394949	0.394301891	0.461683035	0.540357828	0.695259094	0.768259048	0.883510113
20	0.149952888	0.2367239	0.294172049	0.334887981	0.41989398	0.504966021	0.571146011	0.75497508	0.838707924	0.907320023

Tabel 3. Waktu Eksekusi Proses Pengkategorian Tanpa Melalui Proses *Stemming* dalam *Microtime*

No	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6	Stage7	Stage8	Stage9	Stage10
1	0.2382359505	0.3993890285	0.511906862	0.3651959896	0.7278900146	0.863339186	1.152242899	1.097204924	1.116427898	1.132164955
2	0.1055860519	0.2975490093	0.458884001	0.5113329887	0.7723481655	0.899333954	1.012017965	1.269929886	1.360245943	1.510792017
3	0.1286361218	0.1842100620	0.258597851	0.3134720325	0.4545550346	0.711207867	0.718458176	0.958343029	1.058772802	1.282052994
4	0.0559861660	0.1764061451	0.250726938	0.3166048527	0.4360058308	0.541713953	0.613070965	0.781561136	0.898020983	1.049050093
5	0.1715209484	0.3953528404	0.565453053	0.6684889793	0.9226291180	1.167444944	1.353520155	1.608170033	1.76691699	1.903487921
6	0.0775899887	0.1962010860	0.291208029	0.3375639915	0.4680669308	0.562766075	0.681311131	0.84697485	0.929783106	1.061769962
7	0.0572791100	0.1472671032	0.23011899	0.3085439205	0.4097509384	0.529805183	0.600269079	0.79897213	0.930126905	1.027723789
8	0.0531151295	0.1478481293	0.209905148	0.2562599182	0.3545019627	0.45373702	0.52839303	0.689681053	0.757035017	0.841529131
9	0.0730638504	0.1705400944	0.262266874	0.3496220112	0.4664850235	0.592260122	0.66552496	0.848428965	0.938343048	1.110394955
10	0.0687777996	0.1666049957	0.239670992	0.2867901325	0.4266169071	0.535316944	0.607321978	0.796504021	0.880991936	0.966241121
11	0.0876610279	0.2280149460	0.334915161	0.3973209858	0.5406138897	0.664748192	0.76022315	0.966477871	1.051064014	1.158355951
12	0.0969998837	0.2270510197	0.382358074	0.4426219463	0.5894069672	0.761658907	0.850543976	1.059273958	1.161270142	1.282813072
13	0.0732359886	0.2023320198	0.278548956	0.3366079330	0.5103590488	0.613507032	0.696471214	0.894232988	0.991608143	1.124103069
14	0.0794479847	0.2107989788	0.300765991	0.3906948566	0.5355150700	0.652369022	0.786794186	0.986914873	1.088977098	1.150426865
15	0.0834529400	0.1998019218	0.298913002	0.3622031212	0.4909319878	0.60890913	0.716061831	0.895690203	1.02690196	1.095124006
16	0.0779490471	0.2109420300	0.308918953	0.3681709766	0.5390369892	0.673781872	0.794202089	0.969779015	1.047739983	1.193173885
17	0.1009931564	0.2475419044	0.388508081	0.4572699070	0.6205439568	0.746541977	0.85955596	1.103724957	1.19315505	1.34600091
18	0.0969948769	0.2476830482	0.417104959	0.4181439877	0.5751140118	0.695471048	0.83933115	1.062905073	1.22151494	1.353791952
19	0.0671648979	0.1626141071	0.240062952	0.3020830154	0.4139051437	0.515087843	0.599238157	0.794836998	0.874492884	0.966006041
20	0.0717878342	0.1639089584	0.232921839	0.2548701763	0.3911001682	0.483520031	0.558172941	0.798171043	0.887814045	0.970895052

Data waktu eksekusi pada tabel 2 dan 3 dapat ditampilkan dalam bentuk grafik perbandingan seperti pada gambar dibawah ini :



Gambar 6. Grafik waktu rata-rata pengkategorian

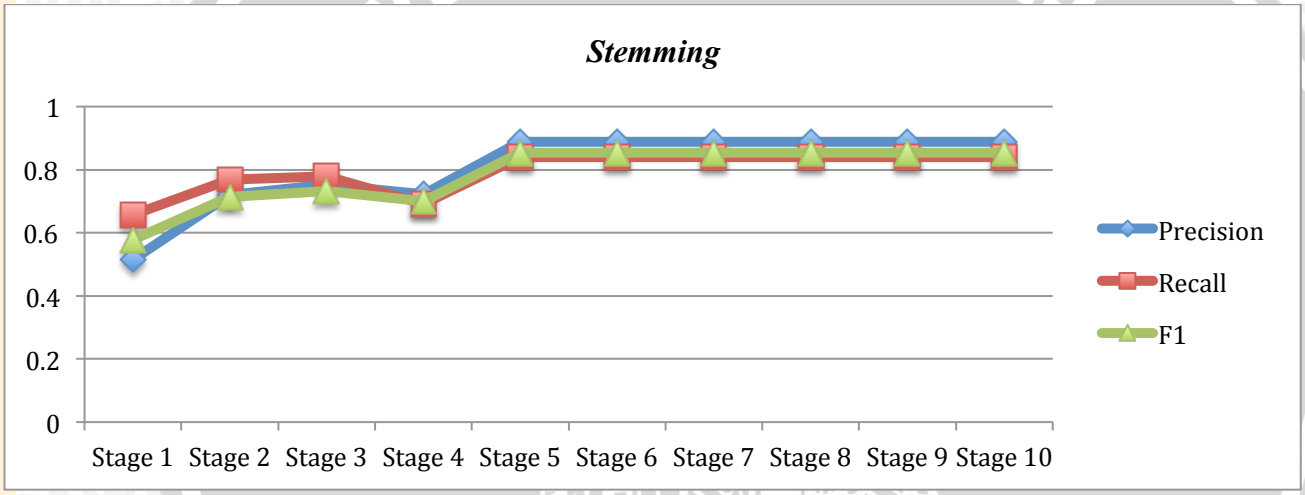
Berikut adalah tabel nilai *recall, precision* dan F_1 *measure* dari uji coba efektifitas. Pembagian jumlah data latih sesuai dengan tabel 1. Proses evaluasi performa efektivitas dari sistem klasifikasi teks menggunakan suatu standar yang disebut matriks *confusion.* Matriks *confusion* berisi informasi mengenai klasifikasi yang sebenarnya dam prediksi klasifikasi yang dilakukan oleh sistem [25:121]. Komponen dalam melakukan perhitungan *recall, precision* dan F_1 *measure* adalah menggunakan nilai TP (*true positive*), FP (*false positive*), TN (*true negative*). TP merupakan nilai pengkategorian oleh sistem ya dan secara manual ya, FP merupakan nilai pengkategorian oleh sistem ya dan secara manual tidak, dan TN merupakan pengklasifikasian sistem tidak dan secara manual ya.

Tabel 4. Evaluasi rata-rata efektifitas sistem menggunakan proses *stemming*

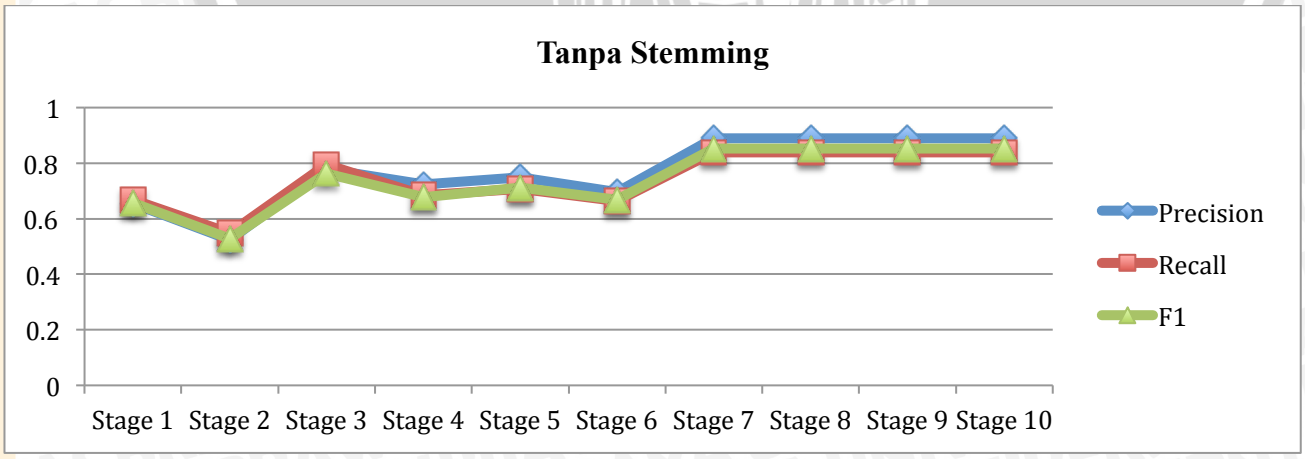
Stage	Precision	Recall	F_1 *measure*
Stage 1	0.515082956	0.656565657	0.577281918
Stage 2	0.720050905	0.76875	0.714815955
Stage 3	0.752856335	0.779166667	0.732251082
Stage 4	0.722087104	0.690972222	0.700430264
Stage 5	0.888753771	0.839583333	0.85314248
Stage 6	0.888753771	0.839583333	0.85314248
Stage 7	0.888753771	0.839583333	0.85314248
Stage 8	0.888753771	0.839583333	0.85314248
Stage 9	0.888753771	0.839583333	0.85314248
Stage 10	0.888753771	0.839583333	0.85314248

Tabel 5. Evaluasi rata-rata efektifitas sistem tanpa proses *stemming*

Stage	Precision	Recall	F_1 *measure*
Stage 1	0.653657617	0.668402778	0.656092949
Stage 2	0.523897059	0.548611111	0.528283599
Stage 3	0.77849736	0.795833333	0.761831164
Stage 4	0.723369155	0.684027778	0.677810139
Stage 5	0.749010181	0.70625	0.711585968
Stage 6	0.696446078	0.664583333	0.667624616
Stage 7	0.888753771	0.839583333	0.85314248
Stage 8	0.888753771	0.839583333	0.85314248
Stage 9	0.888753771	0.839583333	0.85314248
Stage 10	0.888753771	0.839583333	0.85314248



Gambar 7. Grafik nilai *precision, recall,* dan *F1* dengan *stemming*



Gambar 8. Grafik nilai *precision, recall,* dan *F1* tanpa *stemming*

Dari hasil evaluasi efektifitas, didapatkan nilai rata-rata *recall, precision,* dan F_1 *measure* yang berbeda untuk jumlah data latih yang berbeda untuk jumlah data latih yang berbeda. Dengan semakin banyak jumlah data latih yang digunakan dalam tahap pembelajaran maka semakin meningkat nilai rata-rata *recall, precision,* dan F_1 *measure*. Walaupun pada kategori tertentu nilai *recall, precision,* atau F_1 *measure* justru menurun dengan meningkatnya jumlah data latih tapi secara rata-rata nilai evaluasi efektifitas sistem semakin meningkat. Tetapi peningkatan jumlah data latih juga disertai dengan meningkatnya waktu pengkategorian, dengan kata lain menurunnya efisiensi sistem. Oleh karena itu dibutuhkan kecermatan dalam memilih besarnya jumlah data latih, agar sistem dapat berjalan dengan baik dan seimbang. Pada pengujian terlihat bahwa peningkatan terjadi pada *stage 4* dan *stage 5*. Sehingga dapat disimpulkan bahwa pada jumlah data latih pada *stage* 5 sudah cukup untuk dapat mengklasifikasikan dokumen dengan baik. Dengan demikian dapat dikatakan bahwa pada jumlah data latih pada *stage* 5 merupakan solusi terhadap pengaruh besarnya data latih terhadap nilai efektifitas dan efisiensi sistem yang berbanding terbalik. Dari sepuluh kali uji coba menggunakan data latih, nampak pada hasil pengkategorian adalah nilai rata-rata *recall* sebesar 0.793295455, *precision* sebesar 0.804259992, dan F_1 *measure* sebesar 0.78436341, sehingga dapat disimpulkan bahwa efektifitas sistem secara rata-rata sudah berjalan dengan baik.

7. KESIMPULAN DAN SARAN

Pada bagian ini akan dipaparkan kesimpulan dan saran yang dapat diambil setelah melakukan pengujian terhadap sistem.

Kesimpulan

Kesimpulan yang dapat diambil setelah melakukan pengujian terhadap hasil pengkategorian dan performa sistem antara lain:

1. Pengkategorian pesan singkat pada jejaring sosial Twitter dengan metode *naive bayes* dapat diterapkan dengan menggunakan data latih yang bersumber pada dokumen RSS menghasilkan nilai rata-rata *recall* (ukuran dari jumlah dokumen benar dari suatu kategori yang berhasil diklasifikasikan oleh sistem) sebesar 0.793295455 (79%), *precision* (ukuran dari jumlah dokumen yang diklasifikasikan oleh sistem dan dokumen tersebut benar) sebesar 0.804259992 (80%), dan F_1 *measure* (gabungan dari nilai *precision* dan *recall*) sebesar 0.78436341 (78%).
2. Efektifitas sistem semakin meningkat dengan meningkatnya jumlah data latih yang digunakan dalam pembelajaran, sebaliknya efisiensi sistem semakin menurun dengan meningkatnya jumlah data latih dalam pembelajaran.
3. Pada percobaan yang dilakukan, pada data latih RSS sebesar 10728 data atau *stage* 5 sudah cukup untuk dapat mengklasifikasikan dokumen dengan baik.
4. Stemming berpengaruh dalam mendapatkan frekuensi dari setiap kata, dimana frekuensi setiap kata yang akan menjadi dasar sebagai penentuan kata penting. Penggunaan metode *stemming* mempunyai dampak terhadap pengkategorian karena secara rata-rata dengan *stemming* akan meningkatkan prosentase akurasi pengkategorian.

Saran

Pembuatan aplikasi pengkategorian pesan singkat (*tweet*) pada jejaring sosial Twitter berbahasa Indonesia menggunakan metode Naive Bayes masih memiliki beberapa kekurangan yang mungkin berguna untuk penelitian selanjutnya. Saran yang dapat diberikan setelah pengerjaan Tugas Akhir ini adalah :

1. Memberikan batasan terhadap frekuensi kemunculan kata yang layak untuk dilakukan pemrosesan.
2. Proses *stemming* yang dibangun lebih kompleks lagi, dengan mempertimbangkan sisipan dan imbuhan yang diserap dari bahasa asing.
3. Evaluasi dapat dilakukan dengan membandingkan dengan algoritma pengklasifikasian teks lainnya.
4. Pemberian bobot yang berbeda dari kata-kata yang berupa istilah khas dalam suatu kategori tertentu.

DAFTAR PUSTAKA

[1] Phuvipadawat, Swit, dan Tsuyoshi, Murata . 2010. Breaking News Detection and Tracking in Twitter. International Conference on Web Intelligence and Intelligent Agent Technology.

[2] Sriram,Bharath, Fuhry,David, Demir,Engin, Ferhatosmanoglu,Hakan, Demirbas,Murat. 2010 . Short Text Classification in Twitter to Improve Information Filtering. International ACM SIGIR Conference on Research and Development in Information Retrieval.

[3] Manning,D Christopher, Raghavan,Prabhakar, Schütze,Hinrich. 2009 . An Introduction to Information Retrieval. Cambridge University Press.

[4] GuoQiang. 2010 . AnEffective Algorithm for Improving the Performance of Naive Bayes for Text Classification. Cambridge University Press.

[5] O'Reilly,Tim, Milstein Sarah. 2012 . The Twitter Book . O'Reilly.

[6] Mustafa,Atika, Akbar,Ali, dan Sultan, Ahmer. 2009 . Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization. International Journal of Multimedia and Ubiquitous Engineering Vol. 4, No. 2, April, 2009.

[7] Even, Yahir dan Zohar. 2002. Introduction to Text Mining. Automated Learning Group National Center For Supercomputing Applications. University of Illionis. http://www.docstoc.com/docs/25443990/Introduction-to-Text-Mining Diakses pada tanggal 6 Juli 2012.

[8] Witten ,Ian H. Text Mining. Computer Science, University of Waikato, Hamilton.

[9] Zhu, Xiaojin. 2010. Basic Text Process. The University of Wisconsin Madison.

[10] Pisceldo, Femphy, Adriani, Mirna, dan Manurung, Ruli. 2009. Probabilistic Part Of Speech Tagging for Bahasa Indonesia. Faculty of Computer Science University of Indonesia.

[11] Langgeni, Diah Pudi, Baizal, ZK Abdurahman, dan Firdaus, Yanuar. 2010. Clustering Artikel Berita Berbahasa Indonesia Menggunakan *Unsupervised Feature Selection.* Seminar Nasional Informatika 2010.

[12] Allan, James ,dan Kumaran, Giridhar . 2003 . Stemming in the Language Modeling Framework. Department of Computer Science University of Massachusetts Amherst.

[13] Tala, F. Z. 2003. A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia. M.S thesis.

[14] Devedzic, Vladan. 2001. Knowledge Discovery and Data Mining in Databases. FON-School of Business Administration, University of Belgrade, Yugoslavia.

[15] Larose, Daniel T. 2005. Discovering Knowledge in Data An Introduction to Data Mining. Wiley-Interscience.

[16] Kotsiantis, S.B, 2007. Supervised Machine Learning: A Review of Classification Techniques. Informatica 31 (2007) 249-268.

[17] Ramlan, M. 1986. Ilmu Bahasa Indonesia : Morfologi Suatu Tinjauan Deskriptif. CV Karyono. Yogyakarta.

[18] Chaer, Abdul. 2008. Morfologi Bahasa Indonesia (Pendekatan Proses). Rineka Cipta. Jakarta.

[19] Arifin,Agus Zainal , Setiono,Ari Novan . 2001 . Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering.

[20] Santosa, Budi. 2007 . Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis. Graha Ilmu .Yogyakarta.

[21] Dumais,Susan, Platt,John, dam Hackerman,David. 2008. Inductive Learning Algorithms and Representations for Text Categorization.

[22] Joshi,Shweta, dan Nigam,Bhawna. 2011. Categorizing the Document using Muli Class Classification in Data Mining. 2011 International Conference on Computational Intelligence and Communication Systems.

[23] Mitchell,Tom M. 1997. Machine Learning. T.M. Mitchell, McGraw Hill.

[24] Sebastiani, Fabrizio. 2002. Machine Learning in Automated Text Categorization.ACM Computing Surveys, Vol. 34, No. 1, March 2002.

[25] Visa,Sofia, Ramsya,Brian, Ralescu,Anca, dan Knap,Esther. 2011. Confusion Matrix-based Feature Selection. Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference 2011

[26] Hovy ,Eduard . 2003 . Text Summarization chapter 32.

[27] Yang,Yiming, dan Liu,Xin. 1999. A re-examination of text categorization methods. Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval.