Penerapan *Text Mining* dalam *Spam Filtering* untuk Aplikasi *Chat*

Ni Luh Ratniasih[1], Made Sudarma [2], Nyoman Gunantara [3]

Abstract — The Internet has become something important in the communication development. One communication facilities on the Internet is the Internet relay chat or known as chat. Chat applications in real time is often misused for the purpose of spreading the virus, promotions, and other interests known as spam. Spamming is the sending of unwanted messages by someone who has a chat account. This causes the chat account feel uncomfortable with the condition. Based on these problems this research create a chat application that can filter messages or spam filtering by applying text mining. Spam filtering process can be done in two phases: text pre-processing and analyzing. These two phases are carried out to calculate the weight (W) of connectedness with the word spam messages. Based on the results of tests performed on chat applications by applying text mining to perform filtering on spam messages generate the level of accuracy of 91.41%.

Intisari — Internet telah menjadi sesuatu hal yang penting dalam perkembangan sarana komunikasi. Salah satu fasilitas komunikasi yang terdapat pada internet adalah internet relay chat atau yang sering dikenal dengan istilah chat. Aplikasi chat yang bersifat real time sering disalahgunakan untuk keperluan penyebaran virus, promosi, dan kepentingan lain yang dikenal dengan istilah spam. Tindakan spamming adalah pengiriman pesan yang tidak diinginkan oleh seseorang yang memiliki sebuah akun chat. Hal ini menyebabkan pemilik akun merasa tidak nyaman dengan kondisi tersebut. Berdasarkan permasalah tersebut maka dalam penelitian ini membuat sebuah aplikasi chat yang dapat menyaring pesan atau spam filtering dengan menerapkan text mining. Proses spam filtering dilakukan dengan dua tahap yaitu tahap text pre-processing dan analyzing. Kedua tahap ini dilakukan untuk menghitung bobot (W) keterhubungan kata spam dengan pesan. Berdasarkan hasil pengujian yang dilakukan pada aplikasi chat dengan menerapkan text mining untuk melakukan filtering terhadap pesan spam menghasilkan tingkat akurasi sebesar 91.41%.

***Kata Kunci*— Aplikasi *Chat*, *Text Mining*, *Spam filtering*..**

I. PENDAHULUAN

Internet Relay Chat atau yang sering dikenal dengan istilah *chat*, merupakan sumber daya dalam internet yang memungkinkan dua orang atau lebih (*group*) melakukan dialog secara langsung dan *real time* dalam bentuk komunikasi yang tertulis [1]. Berbagai penelitian pun dilakukan untuk mengembangkan aplikasi *chat* itu sendiri seperti penelitian yang dilakukan oleh Diny Wahyuni yaitu pengembangan aplikasi pertukaran pesan berbasis teks melalui jaringan lokal (LAN) menggunakan *Microsoft Visual C++ 6.0* [2]. Aplikasi *chat* yang bersifat *real time* sering disalahgunakan untuk keperluan *spam* oleh beberapa orang, dimana mereka akan mengirimkan pesan yang tidak diinginkan oleh pemilik akun *chat* untuk berbagai tujuan mulai dari *marketing* sampai dengan merusak sistem dan mencuri informasi dari komputer korban yang memilih sebuah *link* dalam pesan sampah tersebut.

Berbagai cara dan aplikasi telah digunakan untuk mengatasi masalah *spam* yang bermula dari masalah *email spam* sampai dengan *SMS spam*. Jose Maria Gomez, dkk., melakukan penelitian mengenai *spam filtering* pada SMS dengan menggunakan metode *Bayessian Filtering* yang dipublikasikan pada sebuah jurnal yang berjudul *Content Based SMS Spam Filtering* [3]. Mereka mencoba menggunakan *Bayesian Filters* yang terbukti efektif dalam penanganan *spam email*. Penelitian dilakukan dengan menggunakan koleksi *spam* untuk SMS dalam dua bahasa yaitu Inggris dan Spanyol. Hasilnya, *Bayesian Filtering* juga efektif jika diterapkan untuk *SMS spam.*

Pada penelitian ini akan dirancang sebuah aplikasi *chat* yang mampu menyaring *spam* dengan menggunakan *text mining* dan teknik *Challenge-response filtering*. Rumusan masalah utama dalam penelitian ini adalah mengetahui pola kalimat pesan yang dinyatakan *spam* serta tingkat akurasi dan *response time* dari sistem *filtering spam* dalam mengklasifikasikan pesan ke dalam kelompok *spam* dan *non spam*. Bahasa teks pesan yang digunakan sebagai data *training* dan data *testing* adalah bahasa *Inggris*.

II. METODE PENELITIAN

A. *Asitektur Umum Sistem.*

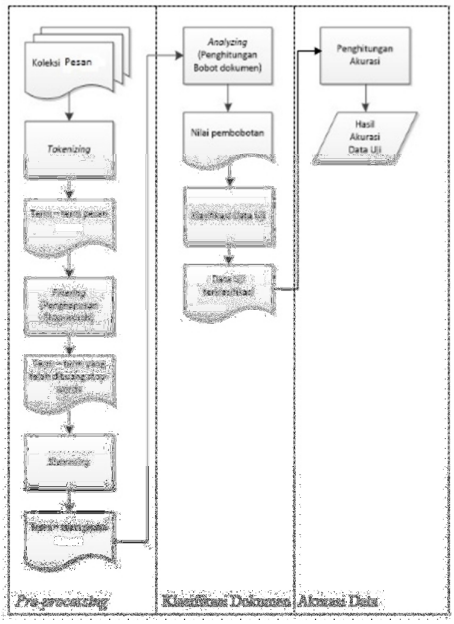
Arsitektur sistem menggambarkan kerja sistem pada proses analisa dan implementasi, dimana arsitektur sistem dari penelitian ini ditunjukkan pada Gambar 1. Tahap pertama dimulai dari *pre-processing* koleksi pesan yang dimulai dari proses *tokenizing*, penghapusan *stop-words*, diakhiri dengan proses *stemming*.

Tahap kedua adalah pengklasifikasian dokumen yang dimulai dari *analyzing* yaitu penghitungan bobot dari proses *pre-processing* sehingga diperoleh nilai pembobotan. Dari nilai pembobotan dilakukan klasifikasi terhadap data *testing*. Tahap terakhir dari sistem ini yaitu menghitung prosentase ketepatan (akurasi) sistem. Untuk menghitung akurasi sistem dalam mengklasifikasikan data *testing* digunakan teori *confusion matrix* yaitu dengan menentukan berapa nilai TP, TN, FP dan FN kemudian dihitung dengan menggunakan persamaan (1) berikut ini [4] :

[1]*Mahasiswa Magister Teknik Elektro Universitas Udayana Kampus Jl. PB Sudirman Denpasar; e-mail ratni.3112@yahoo.com*

[2,3] *Dosen Teknik Elektro Fakultas Teknik Universitas Udayana, Jalan Kampus Bukit Jimbaran 80361 INDONESIA (telp: 0361-703315; fax: 0361-4321; e-mail: msudarma@unud..ac.id, gunantara@unud.ac.id*

$$Akurasi = \frac{TP+TN}{TP+FN+FP+TN} * 100\% \quad (1)$$



Gambar 1: Arsitektur Sistem

B. *Data Penelitian*

Data yang digunakan dalam penelitian ini adalah data primer dan data sekunder. Data sekunder diperoleh atau dikumpulkan dari berbagai sumber yang telah ada yaitu dari halaman *website* yang menyediakan s*pam archive* yaitu *http://untroubled.org/spam/* dan *http://www.dt.fee.unicamp.br*. *Link* tersebut merupakan situs yang khusus mengarsip kalimat *spam* dan *non spam*. Sedangkan data primer diperoleh atau dikumpulkan secara langsung atau mandiri.

Jumlah data pesan yang digunakan dalam penelitian adalah sebanyak 1748 data (dalam bentuk kalimat pesan). Terdapat 5 contoh data pesan seperti pada Tabel 1. Di dalam tabel terdapat 4 pesan yang merupakan pesan *non spam* yaitu pesan nomer 1, 2, 4, dan 5 pesan yang merupakan pesan *spam* yaitu pesan nomer 3. Pesan *spam* adalah pesan yang memberikan informasi tertentu yang bersifat komersil atau pesan yang menyampaikan hadiah.

TABEL I
CONTOH KATEGORI PESAN

No	Pesan	Kategori
1.	*Ok lar... Joking wif u oni...*	*Non Spam*
2.	*U dun say so early hor... U c already then say...*	*Non Spam*
3.	*Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...*	*Non Spam*
4.	*Nah I don't think he goes to usf, he lives around here though*	*Non Spam*
5.	*Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's*	*Spam*

Dari seluruh data yang digunakan, data akan dibagi menjadi dua yaitu 70 % sebagai *data training* (data latih) dan 30% sebagai *data testing* (data uji). *Data training* merupakan data yang digunakan dalam melakukan pembelajaran sedangkan *data testing* adalah data yang tidak pernah dipakai sebagai pembelajaran dan akan berfungsi sebagai data pengujian kebenaran atau keakurasian hasil pembelajaran [5]. Berdasarkan hasil penelitian yang dilakukan oleh Yushintia Pramitarini, dkk dalam mengklasifkasikan status gizi anal balita menggunakan *Naive Bayes Classifier* (NBC) membuktikan bahwa persentase (%) akurasi sistem paling tinggi dihasilkan dengan perbandingan jumlah *data training* dan *data testing* adalah 70% dan 30% [6]. Sehingga jumlah data yang digunakan sebagai *data training* adalah 1224 kalimat pesan dan *data testing* adalah 524 kalimat pesan.

C. *Text Mining*

Text mining merupakan salah satu teknik yang dapat digunakan untuk melakukan klasifikasi, dimana *text mining* merupakan variasi dari *data mining* yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar [7]. Dalam penerapan *text mining*, terdapat beberapa langkah yang perlu dilakukan antara lain :

a) *Tokenizing*

Tokenizing merupakan proses penguraian deskripsi yang semula berupa kalimat menjadi kata [8]. Contoh proses *tokenizing* pada sebuah kalimat dapat dilihat pada Tabel 2.

TABEL II
TAHAP *TOKENIZING*

Text Input	**Hasil *Token***
Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...	*go* *until* *jurong* *point* *crazy* *available* … *wat*

Semua kata yang menyusun kalimat pada kolom "*Text Input*" dipotong berdasarkan kata yang menyusunnya seperti terlihat di kolom "Hasil *Token*" pada Tabel 2.

b) *Filtering*

Filtering adalah tahap mengambil kata penting dari hasil proses *token*. Bisa menggunakan algoritma *stop list* atau *word list* [9]. *Filtering* dapat juga diartikan sebagai proses mengambil kata – kata penting dari hasil proses *token* atau penghapusan *stopwords*. *Stopwords* merupakan kosa kata yang bukan merupakan ciri (kata unik) dari suatu dokumen [10]. Untuk contoh tahap *filtering* terlihat pada Tabel 3.

Kolom "Hasil *Token"* pada Tabel 3 adalah kata – kata yang berasal dari proses *tokenizing* sedangkan kata yang berada pada kolom "Hasil *Filtering"* adalah hasil setelah proses *filtering* yaitu menghilangkan kata yang tidak penting seperti kata penghubung "*go*", "*in*", dan "*there*".

TABEL III
TAHAP *FILTERING*

Hasil *Token*	Hasil *Filtering*
go	*until*
until	*jurong*
jurong	*point*
point	*crazy*
crazy	*available*
available	*bugis*
in	
bugis	
…	…
wat	*wat*

c) *Stemming*

S*temming* merupakan tahap untuk mencari *root* kata dari hasil *filtering*. *Stemming* adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya *(stem)* [11]. Hasil proses *stemming* data *training* dapat dilihat pada Gambar 2.

Gambar 2: Hasil Proses *Stemming*

Pada kalimat dengan ID 5 terdapat kalimat hasil proses *filtering* yaitu "*Nah don't think goes usf, lives here*", setelah adanya proses *stemming* kata "*goes*" berubah menjadi "*go*".

Gambar 3: Hasil Proses *Tagging*

d). *Tagging*

Tagging merupakan tahap untuk mencari bentuk awal/*root* dari tiap kata lampau atau hasil dari proses *stemming*. Hasil proses *tagging* sistem terhadap pesan terlihat pada Gambar 3. Terdapat beberapa kata lampau yang dikembalikan ke bentuk awal, misalkan pada data pesan dengan ID 13 terdapat kata "*won*" dirubah ke bentuk awal menjadi "*win*".

e). Tahap *Analysing*

Analyzing merupakan tahap penentuan seberapa jauh keterhubungan antar suatu kata atau term terhadap suatu dokumen atau kalimat dengan menghitung nilai/bobot keterhubungan. Algoritma *TF/IDF* digunakan dalam proses perhitungan bobot (W) terminologi kata. Algoritma ini digunakan untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval*. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat [12].

Persamaan yang digunakan untuk menghitung bobot (W) masing – masing dokumen terhadap kata kunci adalah [12] :

$$W_{d,t} = tf_{d,t} * IDF \quad (2)$$

dimana : W = bobot dokumen ke – n
d = dokumen
t = kata kunci
tf = *terms frequency* (jumlah kemunculan kata)
IDF = *Inverse Document Frequency*

Nilai tf diperoleh dari [12] :

$$tf_d = \frac{\text{Jumlah munculnya kata } t \text{ dalam dokumen}}{\text{Total jumlah seluruh kata dalam dokumen}} \quad (3)$$

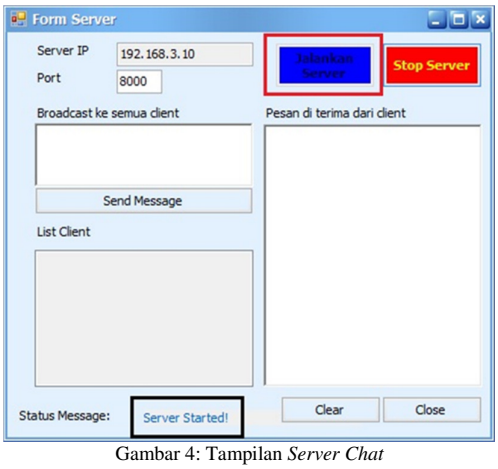
Nilai IDF didapatkan dari [12]:

$$IDF = log_2 \left(\frac{D}{df}\right) \quad (4)$$

dimana : D = total dokumen, dalam hal ini total kalimat yang ada.
df = jumlah dokumen yang mengandung kata kunci.

III. HASIL DAN PEMBAHASAN

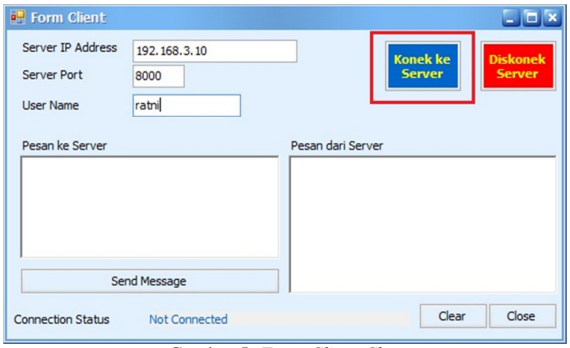
Untuk melakukan kegiatan *chatting* melalui sistem aplikasi ini, hal pertama yang harus dilakukan adalah menjalankan *server* agar *client* atau *user* dapat terkoneksi ke *server* dan *user* dapat melakukan kegiatan *chatting*.



Gambar 4: Tampilan *Server Chat*

Jika server terkoneksi maka statusnya akan menjadi "*Server Started!"* seperti yang terlihat pada Gambar 4. Perlu dicatat adalah *IP Address server*, *LAN card* pada komputer harus berisi *IP Address* terlebih dahulu, dan *Port* yang digunakan adalah *port* bebas dan pastikan tidak sedang digunakan oleh aplikasi yang lain.

Setelah *server* dijalankan, langkah selanjutnya adalah melakukan kegiatan *chatting*. Untuk memulai kegiatan *chatting*, *user* dapat memilih dan membuka *form* menu "*Client*" sehingga akan tampil *form client* seperti Gambar 5.



Gambar 5: *Form Client Chat*

Gambar 6: *Tampilan Form Challenge Response*

Untuk mengirimkan pesan ke *server*, atau ke *client* yang lainnya, cukup mengisi pesan kemudian menekan tombol ***send message*.** Jika proses dan koneksi sudah benar, pesan yang dikirimkan oleh *client* akan diterima oleh *server*. Apabila pesan *chat* yang dikirimkan oleh *client* memiliki nilai IDF lebih kecil dari nilai *threshold* maka pesan akan diklasifikasikan sebagai pesan *spam*, maka sistem akan langsung memumculkan *form Challenge Response*. Tampilan untuk *form Challenge Response* seperti pada Gambar 6.

Data *training* yang digunakan sebanyak 1224 kalimat pesan di-*load* melalui sistem. Tahap awal akan dihitung jumlah kata yang menyusun masing – masing kalimat (kolom "Jumlah Kata") dan jumlah kata *spam* yang terdapat pada masing – masing kalimat (kolom "Jumlah Kata *Spam*"). Selanjutnya menghitung frekuensi kemunculan kata *spam* di dalam kalimat atau tf untuk masing – masing kalimat dengan menggunakan persamaan (3) sehingga diperoleh hasil seperti Tabel 4:

TABEL IV
HASIL PERHITUNGAN DATA LATIH

ID	Kalimat Asli	Jml Kata	Filter Verb Dasar	Jml Kata *Spam*	tf
1	*It will stop on itself. I however suggest she stays with someone that will be able to give ors for every stool.*	22.00	*will suggest stays will able give ors stool*	0.00	0.00
2	*You have 1 new message. Please call 08718738034.*	8.00	*1 new message call 08718738034*	2.00	0.25
3	*U r too much close to my heart. If u go away i will be shattered. Plz stay with me.*	20.00	*U r close heart u away will shattered Plz stay*	0.00	0.00
4	*URGENT! We are trying to contact U. Todays draw shows that you have won a £800 prize GUARANTEED. Call 09050001808 from land line. Claim M95. Valid12hrs only*	27.00	*URGENT trying contact U Todays draw shows win £800 prize GUARANTEED Call 09050001808 land line Claim M95 Valid12hrs only*	5.00	0.19
5	*I got lousy sleep. I kept waking up every 2 hours to see if my cat wanted to come in. I worry about him when its cold :(*	28.00	*lousy sleep keep waking 2 hours see cat wanted come worry cold*	2.00	0.07
…	…	...	...	…	…
1224	*Ah poop. Looks like ill prob have to send in my laptop to get fixed cuz it has a gpu problem*	21.00	*Ah poop Looks ill prob laptop fixed cuz gpu problem*	0.00	0.00
			Nilai Rata - Rata		**0.10**

Selanjutnya menghitung inverse frekuensi dokumen (dalam hal ini kalimat) yang mengandung kata *spam* atau nilai *IDF* dengan menggunakan persamaan (4) :

$$IDF = log_2 (\frac{D}{df})$$

$$= log_2 (\frac{1224}{921})$$

$$= log_2 (1.33)$$
$$= 0.12$$

Setelah memperoleh nilai tf dan IDF selanjutnya akan dihitung nilai ambang bobot (W) dengan menggunakan persamaan (2), namun untuk memperoleh nilai ambang bobot (W) yang akan dihitung adalah nilai rata – rata bobot (W) sebagai berikut :

$$\overline{W_{d,t}} = \overline{tf_{d,t}} * IDF$$
$$= 0.10 * 0.12$$
$$= 0.012$$

Jadi nilai ambang bobot (W) yang digunakan untuk *filtering* pesan *chat* adalah adalah 0.012. Pesan *chat* dapat dikatakan *spam* jika bobot (W) keterhubungan kata *spam* dengan kalimat dari pesan *chat* tersebut lebih besar dari nilai ambang (*threshold*) batas demikian sebaliknya. Sehingga dapat dibuatkan tabel klasifikasi *spam* dan *non spam* seperti Tabel 5 berikut ini :

TABEL V
KLASIFIKASI PESAN

Nilai Bobot (W)	Klasifikasi
≥ 0.012	*Spam*
< 0.012	*Non Spam*

Hasil penghitungan bobot (W) pada sistem dapat dilihat pada Gambar 7.

Gambar 7: Hasil Penghitungan Bobot Sistem

Jumlah data *testing* yang digunakan adalah sebanyak 524 kalimat pesan. Pengujian ini bertujuan untuk mendapatkan akurasi sistem dan *response time*.

TABEL VI
HASIL PENGUJIAN

ID	Jumlah Kata	Jumlah Kata *Spam*	tf	Bobot (W)	Label	Hasil
1	12.00	0.00	0.00	0.000	*Spam*	*Non Spam*
2	13.00	0.00	0.00	0.000	*Spam*	*Non Spam*
3	18.00	2.00	0.11	0.299	*Non Spam*	*Spam*
4	19.00	2.00	0.11	0.266	*Non Spam*	*Spam*
5	4.00	0.00	0.00	0.000	*Non Spam*	*Non Spam*
6	22.00	1.00	0.05	0.112	*Spam*	*Spam*
7	25.00	3.00	0.12	0.254	*Spam*	*Spam*
8	14.00	1.00	0.07	0.141	*Non Spam*	*Spam*
9	22.00	1.00	0.05	0.097	*Non Spam*	*Spam*
10	15.00	3.00	0.20	0.374	*Spam*	*Spam*
11	21.00	3.00	0.14	0.255	*Non Spam*	*Spam*
12	13.00	2.00	0.15	0.266	*Non Spam*	*Spam*
13	23.00	3.00	0.13	0.224	*Spam*	*Spam*
14	7.00	2.00	0.29	0.487	*Non Spam*	*Spam*
15	17.00	0.00	0.00	0.000	*Non Spam*	*Non Spam*
…	…	…	…	….	…	….
524	8.00	0.00	0.00	0.000	*Non Spam*	*Non Spam*

Kalimat pesan dengan ID 1 dengan kalimat "*Latest News! Police station toilet stolen, cops have nothing to go on!*" menunjukkan hasil bahwa jumlah kata yang terhitung sebanyak 12 kata, jumlah kata *spam* yang muncul pada kalimat tersebut adalah 0, nilai tf yang dihasilkan 0, nilai bobot (W) adalah 0. Karena nilai bobot (W) lebih kecil dari 0.012 maka hasil yang muncul pada kolom "Hasil" adalah *Non Spam*.

Confucion matrix digunakan untuk pengukuran efektifitas klasifikasi atau tingkat akuasi dengan menggunakan persamaan (1). C*onfucion matrix* untuk membandingkan hasil proses sistem dengan label *data testing* ditunjukkan pada Tabel 4. Dari tabel diperoleh nilai *TP* = 315, *TN* = 164, *FP* = 40, *FN* = 5.

TABEL VII
CONFUSION MATRIX

		Kelas Prediksi	
		Spam	*Non Spam*
Kelas Sebenarnya	*Spam*	315	5
	Non Spam	40	164

Tingkat akurasi sesuai Table 7 c*onfucion matrix* yang dinyatakan dalam persamaan (1) sebagai berikut :

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} * 100\%$$
$$= \frac{315 + 164}{315 + 5 + 40 + 164} * 100\%$$
$$= \frac{479}{524} * 100\%$$
$$= 91.41\ \%$$

Dari 524 *data testing* diperoleh akurasi kecocokan antara hasil proses sistem menggunakan *text mining* dengan label *data testing* sebesar 91.41%. Hasil akurasi sistem dapat juga dihitung langsung oleh sistem dengan hasil seperti Gambar 8.

Gambar 8: Hasil Akurasi

Jumlah data yang di-*load* sebanyak 524 kalimat pesan. Dari 524 kalimat pesan tersebut, terdapat kalimat pesan yang dinyatakan *spam* yaitu sebanyak 355 sedangkan *non spam* sebanyak 169. Jumlah kecocokan hasil proses sistem menggunakan *text mining* dengan label *data testing* sebanyak 479 kalimat pesan. Persentase kecocokan antara hasil proses sistem menggunakan *text mining* dengan label *data testing* atau dengan istilah tingkat akurasi sistem sebesar 91.41%. Sedangkan r*espons time* (waktu tanggap) yang dalam sistem disebut durasi adalah 277 detik.

IV. KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan beberapa hal sebagai berikut:

a. Penerapan metode *text mining* dan teknik *challenge-response filtering* pada proses *filtering* pesan *spam* dilakukan dengan membangun sistem aplikasi *chat*. Proses *filtering* dilakukan dengan tahap *text pre-processing* dan *analyzing* sehingga diperoleh kalimat yang dinyatakan sebagai kalimat *spam* adalah berdasarkan kemunculan kata *spam* dalam kalimat pesan tersebut, dimana jika nilai W sebuah kalimat pesan lebih besar dari 0.012 (*threshold*). Pola kalimat yang dinyatakan *spam* adalah kalimat yang mengandung unsur seksual, kalimat yang mengandung angka yang panjang (seperti nomor telepon dan alamat), serta kalimat yang mengandung kata – kata yang tidak umum atau berbentuk singkatan.
b. Dari hasil pengujian data *testing* sejumlah 524 kalimat pesan diperoleh tingkat akurasi sistem sebesar 91.41%. Sedangkan *response time* dari sistem *filtering spam* dalam mengklasifikasikan pesan ke dalam kelompok *spam* dan *non spam* adalah sebesar 277 detik. Faktor yang mempengaruhi tingkat akurasi adalah pertama, jumlah kata (*list spam*) di dalam *database* yang digunakan sebagai acuan kata *spam*, faktor kedua adalah kurangnya *filtering* terhadap data *testing* menggunakan pola kalimat *spam* yang telah diperoleh.

REFERENSI

[1] Abdul Kadir & Terra CH Triwahyuni. 2003. *Pengenalan Sistem Informasi*. Yogyakarta: Penerbit Andi Yogyakarta.
[2] Wahyuni Diny & Susetyo Hadi. 2008. Pengembangan Aplikasi Pertukaran Pesan Berbasis Teks Melalui Jaringan Lokal (LAN) Menggunakan Microsoft Visual C++ 6.0. Jurnal Komputasi. 2008; 07.
[3] Gomez Jose Maria, Guillermo Cajigas Bringas, Enrique Puertas Sanz. 2007. Content Based SMS Spam Filtering.
[4] Kristina Paskianti. 2011. Klasifikasi Dokumen Tumbuhan Obat menggunakan Algoritma KNN Fuzzy. Thesis Fakultas Matematika dan Ilmu Pengetahuan Alam IPB. Bogor.
[5] I. H. Witten, E. Frank, and M. A. Hall. 2011. Data Mining Practical Machine Learning Tools and Technique. Burlington: Morgan Kaufmann Publisher.
[6] Pramitarini, Y., Purnama I.K.E., Purnomo, M., 2005. *Analisa Rekam Medis Untuk Menentukan Status Gizi Anak Balita Menggunakan Naive Bayes Classifier*. Seminar Nasional Manajemen Teknologi XVII. Surabaya. 2 Februari 2013; ISBN:978-602-97491-6-8.
[7] Feldman, R & Sanger, J. 2007. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press: New York.
[8] Weiss, S.M., Indurkhya, N., Zhang, T., Damerau, F.J. 2005. Text Mining : Predictive Methods fo Analyzing Unstructered Information. Springer: New York.
[9] Berry, M.W. & Kogan, J. 2010. Text Mining Aplication and theory. WILEY : United Kingdom.
[10] Dragut, E., Fang, F., Sistla, P., Yu, S. & Meng, W. 2009. Stop Word and Related Problems in Web Interface Integration. http://www.vldb.org/pvldb/2/vldb09-384.pdf. Diakses tanggal 8 Desember 2013.
[11] Tala, Fadillah Z. 2003. A Study of Stemming Efects on Information Retrieval in Bahasa Indonesia. Institute for Logic, Language and ComputationUniversiteit van Amsterdam The Netherlands. http://www.illc.uva.nl/Research/Reports/MoL-2003-02.text.pdf. Diakses tanggal 29 September 2014.
[12] Robertson, Stephen, Understanding Inverse Document Frequency: On theoretical arguments for IDF, Journal of Documentation, Vol. 60, pp. 502–520