KLASIFIKASI EMAIL DENGAN MENGGUNAKAN METODE NAÏVE BAYESIAN STUDI KASUS : MAILING LIST www.tux.org

Tantiny [1], Budi Susanto[2], Widi Hapsari[3]

Abstrak: Pada jaman *modern* ini, komunikasi dan penyebaran informasi merupakan hal yang sangat penting. Salah satu bentuk komunikasi yaitu surat-menyurat, tidak lagi dilakukan secara tradisional menggunakan kertas, amplop dan perangko. Surat-menyurat secara global sekarang dilakukan menggunakan teknologi *email*. *Email* saat ini menjadi salah satu alat komunikasi yang pertumbuhannya kian pesat dari hari ke hari. Hal ini dapat dicermati melalui banyaknya komunitas *mailing list* yang bermunculan di Internet. Namun ada kendala yang muncul dari penggunaan *email*, yakni jumlah *email* yang banyak dan diterima dalam waktu yang bersamaan. Hal ini berakibat informasi-informasi yang ada dalam *email* menjadi terkubur dalam tumpukan informasi yang lain. Untuk mengatasi masalah tersebut, maka telah dikembangkan beberapa aplikasi untuk mengklasifikasikan *email* menurut kriteria tertentu, seperti kategori, pengirim atau pun *subject email.* Tugas Akhir ini bertujuan membangun sebuah sistem klasifikasi *email* dengan menggunakan Metode Naive Bayesian dengan mengambil studi kasus dari *mailing list* www.tux.org . Sistem yang dibangun mampu mengklasifikasikan *email* kedalam 3 kategori yang sudah ditentukan dengan pengetahuan ia miliki dan mengubah pengetahuan tersebut jika terjadi kesalahan klasifikasi (pembelajaran bertahap).

Kata Kunci: *text mining, email, klasifikasi, Naive Bayesian.*

PENDAHULUAN

Di jaman era globalisasi informasi menjadi komoditi yang sangat bernilai. Oleh karena itu alat-alat komunikasi kian hari kian berkembang dengan tujuan mempercepat sampainya informasi ke tangan pengguna. Kini alat-alat komunikasi dapat ditemui dalam berbagai macam media seperti media cetak, *audio*, *visual* maupun digital.

Internet sebagai salah satu media digital menawarkan bentuk teknologi komunikasi yang murah dan cepat, yakni *email*. Kemudahan yang ditawarkan *email* menarik banyak orang untuk menggunakannya. Hal ini ditambah dengan tersedianya layanan *email* gratis seperti Yahoo! Mail ataupun Gmail. Selain itu komunitas *mailing list* yang tumbuh di dunia maya juga ikut ambil bagian menumbuhkan kebiasaan penggunaan *email* sebagai salah satu alat komunikasi.

Tetapi pada praktik di lapangan *email* yang seharusnya sebagai sumber informasi bisa kehilangan fungsinya. Hal ini terjadi, karena banyaknya *email* yang diterima dalam waktu yang bersamaan. Contohnya bagi pengguna *email* yang berlangganan *mailing list* dalam satu hari bisa menerima puluhan bahkan ratusan *email*. Tentunya sangat disayangkan, karena informasi yang terkandung dalam *email* terkubur oleh tumpukan informasi yang lain.

[1] *Tantiny , Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[2] *Budi Susanto, S.Kom., M.T., Dosen Teknik Inforrnatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[3] *Dra. Widi Hapsari, M.T., Dosen Teknik Inforrnatika, Fakultas Teknik, Universitas Kristen Duta Wacana*

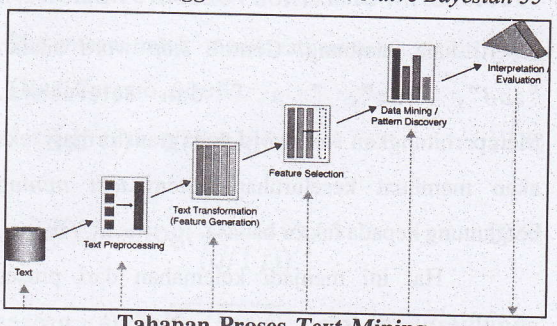
DASAR TEORI

TEXT MINING

Text mining dapat diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, dengan secara otomatis mengekstrak informasi dari sumber-sumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber (Marti Hearst, 2002). Sebagai bentuk aplikasi dari *text mining*, sistem klasifikasi *email* menggunakan *email* sebagai sumber informasi dan informasi klasifikasi sebagai informasi yang akan diekstrak dari sumber informasi. Informasi klasifikasi dapat berbentuk angka-angka probabilitas, *set* aturan atau bentuk lainnya.

Text mining adalah varian dari *data mining*, yang berusaha mencari informasi yang tersimpan dalam suatu data terstruktur seperti basis data. Perbedaan antara *text mining* dan *data mining* terletak pada sumber data yang digunakan. *Text mining* melakukan ekstraksi informasi terhadap data tesktual (*natural language*) yang tidak terstruktur, sedangkan *data mining* melakukan ekstraksi informasi dari data yang terstruktur .

Tahapan proses proses *text mining* dibagi menjadi 4 tahap utama, seperti pada gambar dibawah ini, yaitu proses awal terhadap teks (*text preprocessing*), transformasi teks ke dalam bentuk antara (*text transformation*), pemilihan fitur-fitur yang sesuai (*feature selection*) dan penemuan pola (*pattern discovery*) (Loretta AuvilLoretta & Searsmith, 2003). Masukan awal dari proses ini adalah suatu data teks dan menghasilkan keluaran berupa pola sebagai hasil interpretasi.



Tahapan Proses *Text Mining*.

TEXT PREPROCESSING

Tahap proses awal terhadap teks bertujuan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Tahap ini diawali dengan melakukan pemecahan sekumpulan karakter ke dalam kata-kata (token). Setiap token adalah objek dari suatu tipe, sehingga jumlah token akan lebih banyak daripada tipenya (Budi Susanto, 2006).

Pada tahap ini hal yang perlu diperhatikan adalah terdapatnya karakter-karakter tertentu seperti petik tunggal ('), titik (.), semikolon (;), titik dua (:) serta angka (0-9) atau lainnya yang muncul dalam sebuah *email*. Dalam memperlakukan karakter-karakter tersebut sangat tergantung sekali pada kontek aplikasi yang dikembangkan. Sehingga diperoleh kumpulan kata-kata yang dikandung oleh suatu teks atau kumpulan teks, yang kemudian akan dibawa sebagai *input* untuk tahap berikutnya.

Text Transformation

Pada tahap ini hasil yang diperoleh dari tahap *text preprocessing* akan melalui proses transformasi yang dilakukan dengan mengubah kata-kata ke dalam bentuk dasar sekaligus mengurangi jumlah kata-kata tersebut. Salah satu jenis tindakan yang dapat dilakukan yaitu penghapusan *stopword* .

Stop Word

Stop word adalah kata-kata yang bukan merupakan ciri (kata unik) dari suatu dokumen

seperti kata sambung. Contoh *stop word* adalah "*and*", "*the*", " *a* " dan seterusnya. Memperhitungkan *stop word* pada transformasi teks akan membuat keseluruhan sistem *text mining* bergantung kepada faktor bahasa.

Hal ini menjadi kelemahan dari proses penghilangan *stop word*. Namun proses penghilangan *stop word* tetap digunakan karena proses ini akan sangat mengurangi beban kerja sistem. Dengan menghilangkan *stop word* dari suatu teks maka sistem hanya akan memperhitungkan kata-kata yang dianggap penting.

Feature Selection

Walaupun teks sudah melalui proses penghapusan *stop word* namun tidak semua kata yang tersisa menggambarkan isi dari dokumen. Tahap seleksi fitur (*feature selection*) bertujuan untuk mengurangi dimensi dari suatu kumpulan teks, atau dengan kata lain menghapus kata-kata yang dianggap tidak penting atau tidak menggambarkan isi dokumen berdasarkan frekuensi dari kata tersebut.

Terdapat dua jenis kata yang dapat dianggap "tidak menggambarkan isi dokumen" yaitu kata yang muncul terlalu sedikit atau terlalu banyak dalam dokumen dan kata yang muncul dalam banyak dokumen. Kata yang muncul terlalu sedikit dianggap bukanlah kata yang begitu penting sedangkan kata yang muncul terlalu banyak dianggap sebagai kata umum.

Pattern Discovery

Tahap penemuan pola atau *pattern discovery* adalah tahap terpenting dari seluruh proses *text mining*. Tahap ini berusaha menemukan pola atau pengetahuan dari keseluruhan teks.

Classification

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri bisa berupa aturan "jika-maka", berbentuk pohon pengambilan keputusan (*decision tree*), formula matematis seperti *Naive Bayesian* dan *Support Vector Maching* atau bisa juga berupa jaringan seperti *Neural Network*.

Proses klasifikasi biasanya dibagi menjadi dua fase : *learning* dan *test*. Pada fase *learning*, sebagian data yang telah diketahui kelas datanya diumpankan untuk membentuk model prediksi. Karena menggunakan data yang telah diberikan label terlebih dulu oleh ahli di bidang itu sebagai contoh data yang benar maka klasifikasi sering juga disebut sebagai metoda diawasi (*supervised method*). Kemudian pada fase *test*-nya model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya mencukupi model ini dapat dipakai untuk prediksi kelas data yang belum diketahui.

Supervised Learning

Supervised learning adalah salah satu teknik dalam pembelajaran mesin, untuk membentuk sebuah fungsi dari data latihan. Suatu data latih pada *supervised learning* terdiri dari beberapa pasangan nilai-nilai masukan dan nilai keluaran (nilai dari atribut tujuan).

Berdasarkan keluaran dari fungsi, *supervised learning* dibagi menjadi 2, regresi dan klasifikasi. Regresi terjadi jika *output* dari fungsi merupakan nilai yang kontinyu sedangkan klasifikasi terjadi jika keluaran dari fungsi adalah nilai tertentu dari suatu atribut tujuan (tidak kontinyu). Tujuan dari *supervised learning* adalah untuk memprediksi nilai dari fungsi untuk sebuah data masukan yang sah setelah hanya melihat sejumlah data latih.

Berikut adalah tahapan umum yang biasanya dilakukan pada *supervised learning* :

1. Menentukan tipe dari data latih
2. Mengumpulkan data latih. Data latih yang digunakan seharusnya memiliki karakterisktik dunia nyata. Karena itu data latih dapat berasal baik dari hasil pengukuran atau dari pakar.
3. Menentukan representasi fitur masukan dari fungsi yang ingin dibentuk karena tingkat akurasi dari fungsi dapat dipengaruhi oleh representasi dari masukan (contoh : jumlah fitur tidak boleh terlalu banyak tetapi juga tidak boleh terlalu sedikit, cukup untuk memprediksi keluaran secara akurat).
4. Menentukan struktur dari pengetahuan (fungsi) dan algoritma yang akan digunakan.
5. Jalankan algoritma terhadap data latih.

Metode TF-IDF (*Terms Frequency-Inverse Document Frequency*)

Metode TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap *email*. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata didalam sebuah *email* tertentu dan *inverse* frekuensi *email* yang mengandung kata tersebut. Frekuensi kemunculan kata didalam *email* yang diberikan menunjukkan seberapa penting kata tersebut didalam *email* tersebut. Frekuensi *email* yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut.

Sehingga bobot hubungan antara sebuah kata dan sebuah *email* akan tinggi apabila frekuensi kata tersebut tinggi didalam *email* dan frekuensi keseluruhan email yang mengandung kata tersebut yang rendah pada kumpulan *email* (*database*). Persamaan untuk mendapatkan nilai tf-idf, yakni (Budi Susanto, 2006).

$$\text{tfidf (j)} = \text{tf(j)} * \text{idf (j)} \qquad [2.1]$$

$$idf(j) = \log(\frac{N}{(df(j))}) \qquad [2.2]$$

$$Norm(D)\,\Sigma\sqrt{\Sigma(w(j)^2)} \quad ; \quad w(j)\,\Sigma tfdif(j) \qquad [2.3]$$

$$w_{ij} = tfidf(t_j, e_j) = tf_{ij} * idf(j) \qquad [2.4]$$

Pada persamaan [2.1], kita melihat bahwa bobot TF-IDF merupakan frekuensi kemunculan kata j dimodifikasi dengan sebuah factor skala (idf(j)). Persamaan idf(j) secara sederhana menghitung jumlah dokumen yang berisi kata j (df(j)) dan membalik skalanya. Sehingga ketika suatu kata muncul di beberapa di beberapa dokumen, maka kata tersebut dipertimbangkan sebagai kata yang tidak penting dan nilai faktor skala akan rendah (bahkan mendekati nol). Demikian juga sebaliknya, jika kata bersifat unik dan muncul hanya di beberapa dokumen, faktor skala akan membesarkan karena kata tersebut bersifat penting. Jika diinginkan bentuk normalisasi dari nilai tf-idf(j) menjadi nilai diantara (0...1), maka kita dapat menerapkan persamaan [2.3].Untuk pemmbentukan vector masing-masing dokumen, dapat pula diberikan nilai bobot masing-masing kata dalam dokumen tersebut dengan persamaan [2.4].

Metode Naïve Bayesian

Metode Naive Bayesian adalah klasifikasi model statistik. *Naive Bayesian* dapat memprediksikan kemungkinan-kemungkinan kelas anggota, seperti kemungkinan yang menempatkan sampel baru pada kelas khususnya. *Naive Bayesian* berlandaskan pada teorema *Bayesian* yang selalu memperlihatkan

performa yang cepat dan tepat sekalipun diterapkan pada *database* yang sangat besar.

Berikut ini akan disajikan garis besar Metode *Naive Bayesian* untuk klasifikasi teks Sistem dilatih menggunakan data latih lengkap berupa pasangan nilai-nilai atribut dan nilai target kemudian sistem akan diberikan sebuah data baru dalam bentuk $<a_1,a_2,a_3,...a_n>$ dan sistem diberi tugas untuk menebak nilai fungsi target dari data tersebut (Mitchell, Tom, 1997).

Metode Naïve bayes memberi nilai target kepada data baru menggunakan nilai V_{MAP}, yaitu nilai kemungkinan tertinggi dari seluruh anggota himpunan *set* domain V.

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1 a_2 ... a_n) \qquad [2.5]$$

Terorema Bayes kemudian digunakan untuk menulis ulang rumus tersebut menjadi :

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1 a_2 ... a_n \mid v_j) P(v_j)}{P(a_1 a_2 ... a_n)} \qquad [2.6]$$

$$v_{MAP} = \arg\max_{V_j \in V} P(a_1 a_2 ... a_n \mid v_j) P(v_j) \qquad [2.7]$$

Metode Naïve Bayes bekerja dengan dasar asumsi bahwa atribut-atribut yang digunakan bersifat *conditionally independent* antara satu dan yang lainnya, terhadap nilai fungsi target atau dengan kata lain rumus [2.5] dapat kita tulis ulang menjadi :

$$v_{MAP} = \arg\max_{V_j \in V} P(v_j) \prod_i P(a_i \mid v_j) \qquad [2.8]$$

V_{MAP} adalah nilai probabilitas hasil perhitungan *Metode Naïve Bayes* untuk nilai fungsi target yang bersangkutan. Frekuensi kemunculan kata pada data latih menjadi dasar perhitungan nilai dari $P(v_j)$ dan $P(a_i|v_j)$. Himpunan *set* dari nilai-nilai probabilitas ini berkorespondensi dengan hipotesa yang ingin dipelajari. Hipotesa kemudian digunakan untuk mengklasifikasikan data-data baru dengan menggunakan rumus [2.6]. Mekanisme penghitungan P(ailvj) yang digunakan adalah sesuai dengan rumus [2.7]

$$P(w_k \mid v_j) = \frac{n_k + 1}{n + |kosakata|} \qquad [2.9]$$

Keterangan :

1. n : adalah jumlah total kata berbeda yang terdapat di dalam semua data tekstual yang memiliki nilai fungsi target yang sesuai.
2. n_k : adalah jumlah kemunculan kata w_k pada semua data tekstual yang memiliki nilai fungsi target yang sesuai.
3. |kosakata| : adalah jumlah kata yang berbeda yang muncul dalam seluruh data tekstual yang digunakan.

Ringkasan algoritma untuk Metode *Naïve Bayes* adalah :

A. Proses pelatihan. Input adalah *email* contoh yang telah diketahui kategorinya:

{*Examples* adalah kumpulan data latih yang mencakup seluruh kemungkinan nilai fungsi target. Fungsi ini mempelajari probabilitas $P(w_k|v_j)$ dan $P(v_j)$.}

1. Kumpulkan semua kata, *puctuation* dan *token* yang muncul pada Examples.

♦ *Vocabulary* ← kumpulan kata-kata yang berbeda (*distinct*) yang muncul pada *Examples*.

2. Hitung $P(v_j)$ dan $P(w_k|v_j)$

Untuk setiap nilai target v_j dari V.

♦ $Docs_j$ ← kumpulan dokumen yang memiliki nilai target v_j.

♦ $P(v_j)$ ← $|docs_j|/|Examples|$.

♦ $Text_j$ ← hasil konkatenasi seluruh dokumen pada $docs_j$.

♦ N ← jumlah kata yang berbeda yang muncul pada $Text_j$.

♦ Untuk setiap kata w_k yang ada dalam *Vocabulary*.

$N_k \leftarrow$ jumlah kemunculan kata w_k dalam $Text_j$.

$P(w_k|v_j) \leftarrow (n_k+1)/(n+|Vocabulary|)$.

B. Proses klasifikasi. Input adalah *email* yang belum diketahui kategorinya:

{Mengembalikan estimasi nilai tar*get* dari *Doc*.}

♦ *Positions* ← seluruh kata dari *Doc* yang ditemukan dalam *Vocabulary*

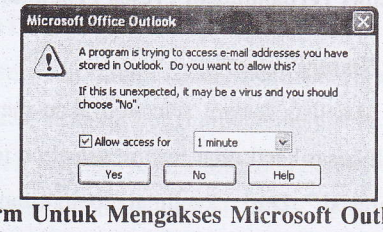
♦ *Return* Vnb , dimana

$$Vnb = \text{argmax } P(vj) \prod P(a_i|v_j)$$

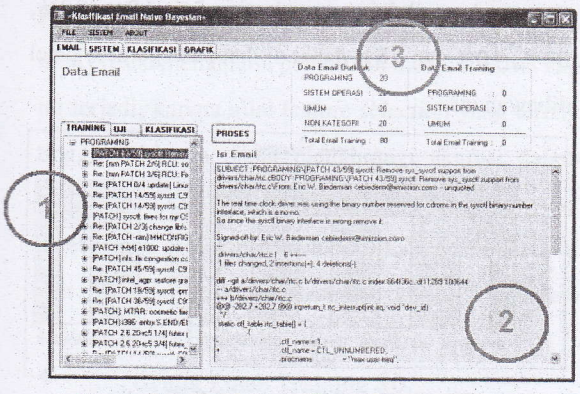
CARA KERJA SISTEM

Proses Migrasi *Email* dari Microsoft Outlook ke *Database*

Proses migrasi dari Microsoft Outlook ke *database* akan menghasilkan daftar *list* untuk *tab email* training, *email* uji dan informasi jumlah *email* dari Microsoft Outlook yang terdapat dalam *database*.



Form Untuk Mengakses Microsoft Outlook.



Form Email Dengan *Email* Hasil Migrasi.

Keterangan :

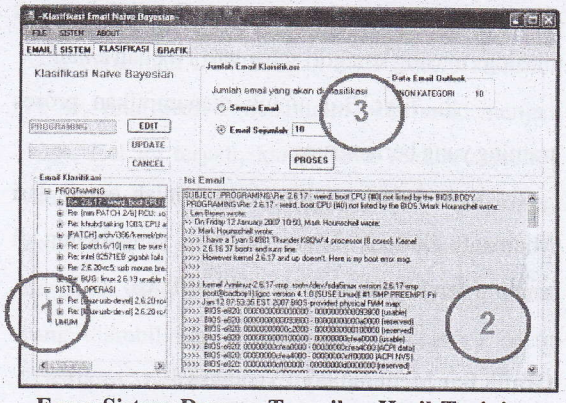
1. *Tab page* yang menampilkan *email* hasil migrasi, *email* ditampilkan menurut kategori dan dibagi berdasarkan tipenya, yakni *email* TRAINING, *email* UJI dan *email* hasil KLASIFIKASI.

2. *Rich text box* untuk menampilkan isi *email* yang dipilih.

3. Informasi mengenai jumlah *email* dari Microsoft Outlook dalam *database* yang belum di training dan jumlah *email* hasil klasifikasi.

Proses Pembangunan Pembelajaran Metode Naïve Bayesian

Proses pembangunan pembelajaran Metode Naïve Bayesian untuk masing-masing kategori, yakni kategori PROGRAMING, kategori SISTEM OPERASI dan kategori UMUM. Proses pembelajaran ini terdiri dari beberapa sub proses yakni tokenisasi, pembuangan *stop word*, pembobotan nilai TF-IDF, *feature selection* dan pembangunan vektor pembelajaran Naïve Bayesian.



Form Sistem Dengan Tampilan Hasil Training.

Keterangan :

1. *Tab page* yang menampilkan *email* hasil klasifikasi, *email* ditampilkan menurut kategori.

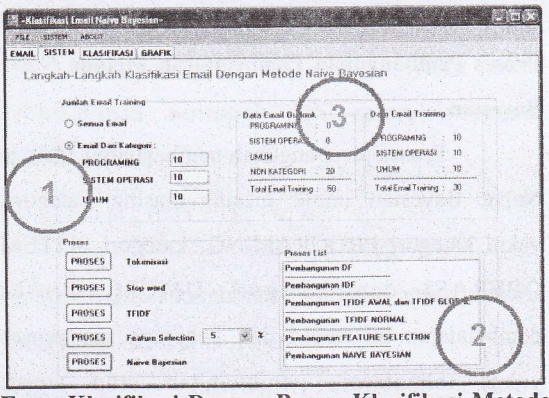
2. *Rich text box* untuk menampilkan isi *email* yang dipilih.

3. Informasi mengenai jumlah *email* dari

Microsoft Outlook dalam *database* yang belum di training.

Proses Pengujian Klasifikasi Metode Naïve Bayesian

Proses pengujian klasifikasi Metode Naïve Bayesian bertujuan untuk menguji *email* termasuk dalam klasifikasi kategori yang mana. Pada proses ini akan dilakukan perhitungan klasifikasi dengan Metode Naïve Bayesian serta perhitungan nilai presisi hasil klasifikasi.



Form Klasifikasi Dengan Proses Klasifikasi Metode Naïve Bayesian.

Keterangan :

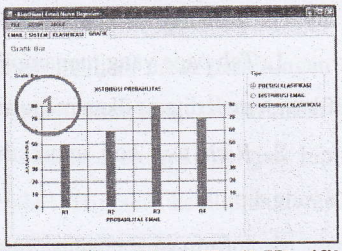
1. Jumlah *input*-an *email* training untuk masing-masing kategori sebanyak 10 buah *email.*

2. *Text Box* untuk menampilkan proses training yang berlangsung.

3. Informasi mengenai jumlah *email* dari Microsoft Outlook dalam *database* yang belum di training dan jumlah *email* hasil klasifikasi.

Grafik Bar Menampilkan Visualisasi

Grafik bar akan menampilkan hasil presisis klasifikasi.



Grafik Bar Dengan Visualisasi Hasil Klasifikasi Metode Naïve Bayesian.

Keterangan :

1. Visualisasi hasil klasifikasi dalam grafik bar.

ANALISIS SISTEM

Untuk analisis sistem akan dilakukan pengujian dengan menggunakan jumlah *email* sebanyak 15, 30, 45 dan 120 buah *email* training untuk masing-masing kategori yang ada dan 10 buah *email* uji yang belum memiliki kategori.

Hasil Pengujian

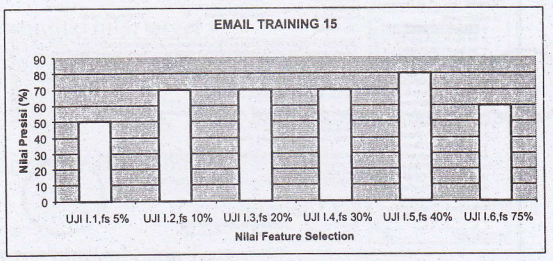
Dari keempat hasil pengujian akan dianalisis pengaruh dari nilai *feature selection* dan jumlah *email* terhadap hasil dari nilai presisi klasifikasi. Hasil keseluruhan perhitungan presisi dari keempat pengujian dapat dilihat pada tabel dibawah ini.

Feature Selection	Perbandingan Hasil Presisi (%) Pengujian			
	I	II	III	IV
	15 email	*30 email*	*45 email*	*120 email*
5%	50	80	80	80
10%	70	80	80	90
20%	70	80	80	80
30%	70	90	80	70
40%	80	70	90	40
75%	60	50	40	50

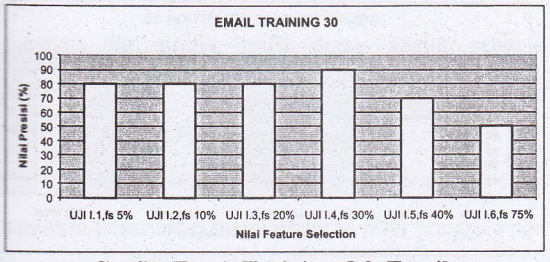
Tabel Perbandingan Presisi Pengujian.

Analisis *Feature Selection*

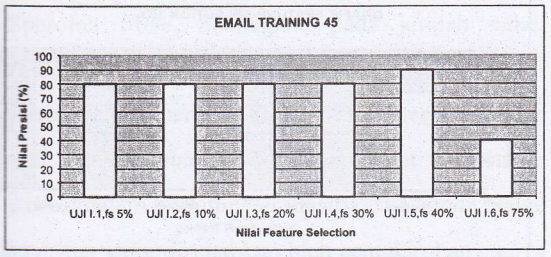
Analisis *feature selection* akan membahas pengaruh besar kecil nilai *feature selection* terhadap nilai presisi. Dari keempat pengujian yang mengunakan nilai *feature selection* 5%, 10%, 20%, 30%, 40% dan 75% dapat dilihat pada gambar tabel dibawah ini.



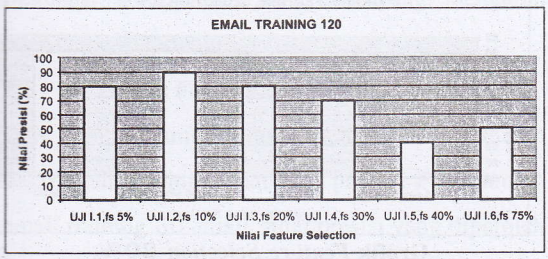
Grafik *Email* Training 15 *Email.*



Grafik *Email* Training 30 *Email*.



Grafik *Email* Training 45 *Email*.



Grafik *Email* Training 120 *Email*.

Analisis dari grafik-grafik diatas adalah sebagai berikut :

♦ **Grafik Email Training 15.**

Dari gambar untuk pengujian I.1, I.2, I.3, I.4 dan I.5 dapat dilihat bahwa, nilai *feature selection* yang semakin tinggi akan membuat nilai presisi klasifikasi semakin tinggi pula. Nilai presisi tertinggi diperoleh saat nilai *feature selection* berada pada nilai 40% dengan nilai presisi 80%. Namun pada Pengujian I.6 dengan nilai *feature selection* 75 %, nilai presisi mengalami penurunan menjadi 60%. Hal ini terjadi, karena nilai *feature selection* yang terlalu tinggi, sehingga dapat mengurangi akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan semakin banyaknya kata yang diambil dan dihitung probabilitasnya dalam klasifikasi Metode Naïve Bayesian.

♦ **Grafik Email Training 30.**

Dari gambar untuk pengujian II.1, II.2, II.3, dan II.4 dapat dilihat bahwa, nilai *feature selection* yang semakin tinggi akan membuat nilai presisi klasifikasi semakin tinggi pula. Nilai presisi tertinggi diperoleh saat nilai *feature selection* berada pada nilai 30% dengan nilai presisi 90%. Namun pada Pengujian II.5 dan II.6 dengan nilai *feature selection* 40% dan 75 %, nilai presisi mengalami penurunan menjadi 70% dan 50%. Hal ini terjadi, karena nilai *feature selection* yang terlalu tinggi, sehingga dapat mengurangi akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan semakin banyaknya kata yang diambil dan dihitung probabilitasnya dalam klasifikasi Metode Naïve Bayesian.

♦ **Grafik Email Training 45.**

Dari gambar untuk pengujian III.1, III.2, III.3, III.4 dan III.5 dapat dilihat bahwa, nilai *feature selection* yang semakin tinggi akan membuat nilai presisi klasifikasi semakin tinggi pula. Nilai presisi tertinggi diperoleh saat nilai *feature selection* berada pada nilai 40% dengan nilai presisi 90%. Namun pada Pengujian III.6 dengan nilai *feature selection* 75 %, nilai presisi mengalami penurunan menjadi 40%. Hal ini terjadi, karena nilai *feature selection* yang terlalu tinggi, sehingga dapat mengurangi akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan semakin banyaknya kata yang diambil dan dihitung probabilitasnya dalam klasifikasi Metode Naïve Bayesian.

♦ **Grafik Email Training 120.**

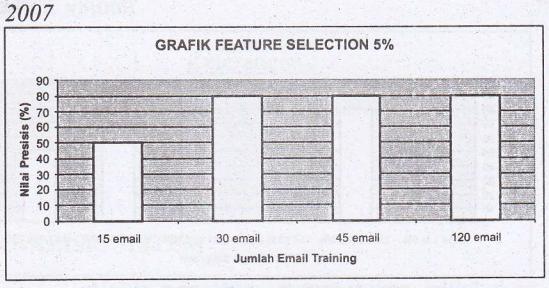
Dari gambar untuk pengujian IV.1 dan IV.2, dapat dilihat bahwa, nilai *feature selection* yang semakin tinggi dan jumlah *email* training yang banyak akan membuat nilai presisi klasifikasi semakin tinggi pula. Nilai presisi tertinggi diperoleh saat nilai *feature selection* berada pada nilai 10% dengan nilai presisi

90%, karena jumlah *email* training yang banyak sehingga saat nilai *feature selection* 10% kata-kata yang terambil sudah cukup banyak jumlahnya dan tinggi frekuensi kemunculan katanya. Sedangkan pada Pengujian IV.3 dan IV.4 nilai presisi mengalami penurunan menjadi 80% dan 70% pada saat nilai *feature selection* bernilai 20% dan 30%. Hal ini terjadi dikarenakan jumlah *email* training yang banyak, sehingga kata-kata yang dihasilkan dari pengambilan *feature selection* berjumlah lebih banyak dari pengujian sebelumnya (untuk nilai *feature selection* yang sama) dan berakibat turunnya nilai akurasi perhitungan klasifikasi.

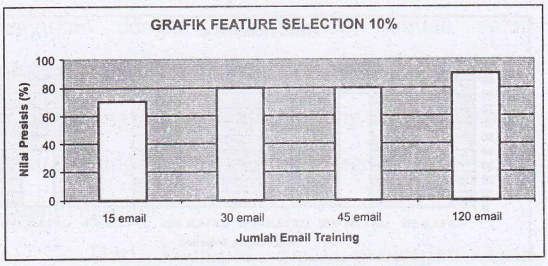
Sedangkan pada Pengujian IV.5 dan IV.6 dengan nilai *feature selection* 40% dan 75 %, nilai presisi semakin mengalami penurunan menjadi 40% dan 50%. Hal ini terjadi, karena nilai *feature selection* yang terlalu tinggi, sehingga dapat mengurangi akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan semakin banyaknya kata yang diambil dan dihitung probabilitasnya dalam klasifikasi Metode Naïve Bayesian.

Analisis Jumlah *Email* Training

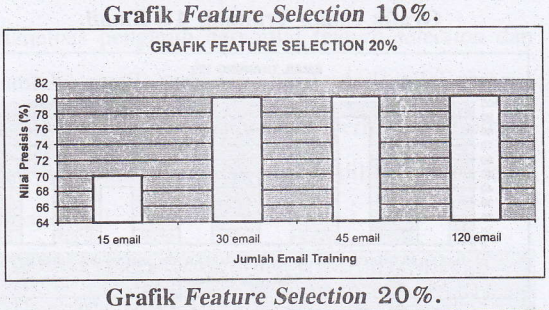
Analisis jumlah *email* training akan membahas pengaruh besar kecil jumlah *email* training terhadap nilai presisi. Dari keempat pengujian yang mengunakan jumlah *email* training sebesar 15 *email*, 30 *email*, 45 *email* dan 120 *email* akan disajikan dalam grafik dengan pengelompokkan berdasarkan nilai *feature selection* yang digunakan. Grafik dapat dilihat pada gambar.



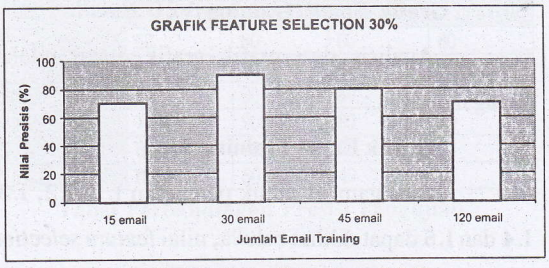
Grafik *Feature Selection* 5%.



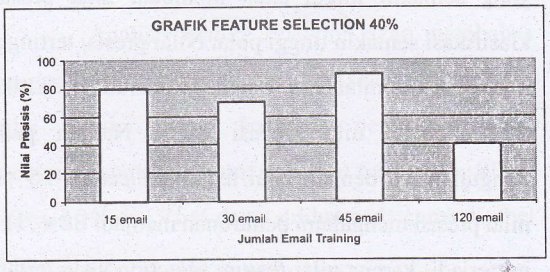
Grafik *Feature Selection* 10%.



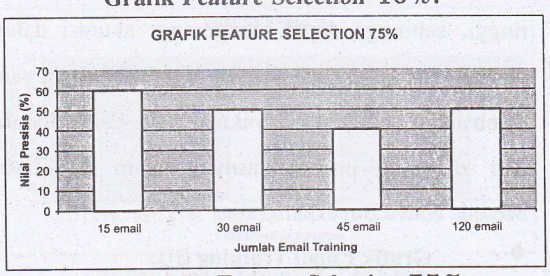
Grafik *Feature Selection* 20%.



Grafik *Feature Selection* 30%.



Grafik *Feature Selection* 40%.



Grafik *Feature Selection* 75%.

Analisis dari grafik-grafik diatas adalah sebagai berikut :

♦ **Grafik *Feature Selection* 5%.**

Dari gambar jumlah *email* training yang semakin besar akan meningkatkan nilai presisi. Pada jumlah *email* training 15 *email*, nilai presisi yang diperoleh 50%. Sedangkan pada jumlah *email* training 30 *email*, 45 *email* dan 120 *email*, nilai presisi yang diperoleh 80%. Hal ini terjadi karena nilai jumlah *email* yang besar dapat menambah akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan oleh semakin banyaknya frekuensi kemunculan kata-kata yang mengarah pada kategori tertentu.

♦ **Grafik *Feature Selection* 10%.**

Dari gambar jumlah *email* training yang semakin besar akan meningkatkan nilai presisi. Pada jumlah *email* training 30 *email*, nilai presisi yang diperoleh 70%. Sedangkan pada jumlah *email* training 30 *email*, dan 45 *email* , nilai presisi yang diperoleh 80%, sedangkan pada saat jumlah *email* training 120 *email* diperoleh nilai presisi 90%. Hal ini terjadi karena nilai jumlah *email* yang besar dapat menambah akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan oleh semakin banyaknya frekuensi kemunculan kata-kata yang mengarah pada kategori tertentu.

♦ **Grafik *Feature Selection* 20%.**

Dari gambar jumlah *email* training yang semakin besar akan meningkatkan nilai presisi. Pada jumlah *email* training 15 *email*, nilai presisi yang diperoleh 70%. Sedangkan pada jumlah *email* training 30 *email*, 45 *email* dan 120 *email*, nilai presisi yang diperoleh 80%. Hal ini terjadi karena nilai jumlah *email* yang besar dapat menambah akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan oleh semakin banyaknya frekuensi kemunculan kata-kata yang mengarah pada kategori tertentu.

♦ **Grafik *Feature Selection* 30%.**

Dari gambar nilai presisi tidak lagi hanya dipengaruhi jumlah *email* training, namun juga nilai *feature selection* ikut berperan. Pada jumlah *email* training 15 *email*, nilai presisi yang diperoleh 70%. Sedangkan pada jumlah *email* training 30 *email*, nilai presisi yang diperoleh 90%. Untuk *email* training 45 dan 120 nilai presisi mengalami penurunan dari nilai sebelumnya menjadi 80% dan 70%. Hal ini dikarena nilai *feature selection* yang tinggi, sehingga dapat mengurangi akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan semakin banyaknya kata yang diambil dan dihitung probabilitasnya dalam klasifikasi Metode Naïve Bayesian.

♦ **Grafik *Feature Selection* 40%.**

Dari gambar nilai presisi tidak lagi hanya dipengaruhi jumlah *email* training, namun juga nilai *feature selection* ikut berperan. Pada jumlah *email* training 15 *email*, nilai presisi yang diperoleh 80%. Sedangkan pada jumlah *email* training 30 *email*, nilai presisi yang diperoleh mengalami penurunan menjadi 70%. Untuk *email* training 45 nilai presisi yang diperoleh mengalami kenaikan dari nilai sebelumnya menjadi 90% yang kemudian mengalami penurunan lagi menjadi 40% pada saat jumlah *email* training berada pada 120 *email*. Hal ini dikarena nilai *feature selection* yang tinggi, sehingga dapat mengurangi akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan semakin banyaknya kata yang diambil dan dihitung probabilitasnya dalam klasifikasi Metode Naïve Bayesian.

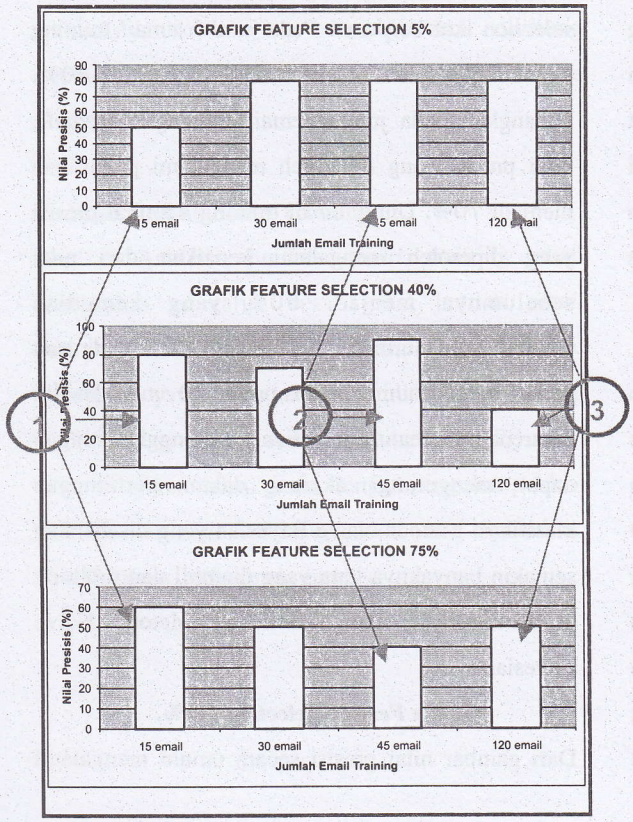
♦ **Grafik *Feature Selection* 75%.**

Dari gambar nilai presisi secara umum mengalami

penurunan. Pada jumlah *email* training 15 *email*, nilai presisi yang diperoleh 60%. Sedangkan pada jumlah *email* training 30 *email*, nilai presisi yang diperoleh mengalami penurunan menjadi 50%. Untuk *email* training 45 nilai presisi mengalami penurunan lagi menjadi 40%. Sedangkan pada saat nilai jumlah *email* training 120 *email*, nilai presisi kembali mengalami kenaikkan menjadi 50%. Hal ini dikarena nilai *feature selection* yang tinggi, sehingga dapat mengurangi akurasi dalam perhitungan klasifikasi Metode Naïve Bayesian yang disebabkan semakin banyaknya kata yang diambil dan dihitung probabilitasnya dalam klasifikasi Metode Naïve Bayesian.

Analisis Hubungan Antara Nilai *Feature Selection* Dengan Jumlah *Email* Training

Dari analisis *feature selection* dan analisis jumlah *email* diperoleh hubungan antara keduanya pada gambar dibawah ini.



Analisis dari grafik-grafik diatas adalah sebagai berikut :

1. Untuk jumlah *email* 15 dengan *feature selection* 5%, saat nilai *feature selection* 5% nilai presisi yang diperoleh 50%. Hal ini disebabkan nilai *feature selection* yang rendah dan jumlah *email* training yang sedikit, sehingga kata-kata yang diambil untuk perhitungan klasifikasi juga sedikit (akibat nilai *feature selection* yang rendah) dan frekuensi kemunculan kata-kata tersebut juga rendah (akibat jumlah *email* training yang sedikit). Kedua faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab kata-kata yang akan dihitung probabilitasnya terlalu sedikit dan nilai frekuensi kemunculan katanya juga rendah, sehingga akurasi perhitungan untuk masing-masing kategori menjadi kurang akurat.

Untuk jumlah *email* 15 dengan *feature selection* 40%, saat nilai *feature selection* 40% nilai presisi yang diperoleh 80%. Hal ini disebabkan nilai *feature selection* yang tinggi. Faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab kata-kata yang akan dihitung probabilitasnya cukup banyak, sehingga akurasi perhitungan untuk masing-masing kategori menjadi lebih akurat.

Untuk jumlah *email* 15 dengan *feature selection* 75%, saat nilai *feature selection* 75% nilai presisi yang diperoleh 60%. Hal ini disebabkan nilai *feature selection* yang sangat tinggi, sehingga kata-kata yang diambil untuk perhitungan klasifikasi jumlahnya lebih banyak (akibat nilai *feature selection* yang tinggi). Faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab kata-kata yang akan dihitung probabilitasnya sangat banyak,

sehingga kata-kata yang umum (kata-kata yang belum tersaring saat proses pembuangan *stop word*) yang berada disemua kategori ikut terambil dan berakibat turunnya akurasi klasifikasi menjadi turun, karena akan ikut dihitung probabilitasnya.

2. Untuk jumlah *email* 45 dengan *feature selection* 5%, saat nilai *feature selection* 5% nilai presisi yang diperoleh 80%. Hal ini disebabkan jumlah *email* training yang banyak. Faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab frekuensi kemunculan kata menjadi lebih banyak walaupun jumlah kata yang diambil sedikit (akibat nilai *feature selection* yang rendah), sehingga akurasi perhitungan untuk masing-masing kategori menjadi lebih akurat.

Untuk jumlah *email* 45 dengan *feature selection* 40%, saat nilai *feature selection* 40% nilai presisi yang diperoleh 90%. Hal ini disebabkan nilai *feature selection* yang cukup tinggi dan jumlah *email* training yang banyak, sehingga kata-kata yang diambil untuk perhitungan klasifikasi juga banyak (akibat nilai *feature selection* yang tinggi) dan frekuensi kemunculan kata-kata tersebut juga tinggi (akibat jumlah *email* training yang banyak). Kedua faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab kata-kata yang akan dihitung probabilitasnya cukup banyak dan nilai frekuensi kemunculan katanya juga tinggi, sehingga akurasi perhitungan untuk masing-masing kategori menjadi lebih akurat.

Untuk jumlah *email* 45 dengan *feature selection* 75%, saat nilai *feature selection* 75% nilai presisi yang diperoleh 40%. Hal ini disebabkan nilai *feature selection* yang sangat tinggi, sehingga kata-kata yang diambil untuk perhitungan klasifikasi jumlahnya lebih banyak (akibat nilai *feature selection* yang tinggi). Faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab kata-kata yang akan dihitung probabilitasnya sangat banyak, sehingga kata-kata yang umum (kata-kata yang belum tersaring saat proses pembuangan *stop word*) yang berada disemua kategori ikut terambil dan berakibat turunnya akurasi klasifikasi menjadi turun, karena akan ikut dihitung probabilitasnya.

3. Untuk jumlah *email* 120 dengan *feature selection* 5%, saat nilai *feature selection* 5% nilai presisi yang diperoleh 80%. Hal ini disebabkan jumlah *email* training yang banyak. Faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab frekuensi kemunculan kata menjadi lebih banyak walaupun jumlah kata yang diambil sedikit (akibat nilai *feature selection* yang rendah), sehingga akurasi perhitungan untuk masing-masing kategori menjadi lebih akurat.

Untuk jumlah *email* 120 dengan *feature selection* 40%, saat nilai *feature selection* 40% nilai presisi yang diperoleh 40%. Hal ini disebabkan oleh pengambilan kata-kata yang terlalu umum, karena jumlah *email* training yang banyak berarti menghasilkan *bag of word* yang besar dan kata-kata yang umum frekuensi kemunculannya lebih tinggi dibandingkan kata-kata yang bersifat khusus untuk suatu kategori. Oleh karena tingginya frekuensi kemunculan kata-kata yang sifatnya umum, maka saat pembobotan nilai TF-IDF kata-kata inilah yang akan memiliki nilai bobot TF-IDF normal yang besar dan sesuai dengan aturan *feature selection* akan mengambil kata-kata yang memiliki bobot TF-IDF normal yang besar. Faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab kata-kata yang akan dihitung probabilitasnya adalah kata-kata yang bersifat umum yang berada disetiap kategori, sehingga akurasi

perhitungannya rendah.

Untuk jumlah *email* 120 dengan *feature selection* 75%, saat nilai *feature selection* 75% nilai presisi yang diperoleh 50%. Hal ini disebabkan nilai *feature selection* yang sangat tinggi, sehingga kata-kata yang diambil untuk perhitungan klasifikasi jumlahnya lebih banyak (akibat nilai *feature selection* yang tinggi). Faktor tersebut yang berpengaruh pada perhitungan probabilitas di metode klasifikasi Naïve Bayesian, sebab kata-kata yang akan dihitung probabilitasnya sangat banyak, sehingga kata-kata yang umum (kata-kata yang belum tersaring saat proses pembuangan *stop word*) yang berada disemua kategori ikut dihitung probabilitasnya.

KESIMPULAN

Dari sistem klasifikasi *email* dengan Metode Naïve Bayesian yang dikembangkan untuk menerapkan konsep *data mining* pada dokumenen *email* dapat ditarik beberapa kesimpulan :

- Sistem klasifikasi *email* dengan Metode Naïve Bayesian yang mengambil studi kasus *mailing list* www.tux.org, telah berhasil menerapkan tahap-tahap sebuah proses *text mining* terhadap kumpulan suatu dokumen teks dan juga berhasil menerapkan proses klasifikasi dengan Metode Naïve Bayesian.
- Keakuratan klasifikasi yang diuji tidak menggunakan proses *stemming* (pengembalian kata menjadi kata dasar).
- Keakuratan klasifikasi tergantung pada tinggi rendahnya nilai *feature selection*. Semakin besar nilai *feature selection*, maka akurasi dari hasil klasifikasi semakin tinggi. Namun nilai *feature selection* yang sangat tinggi dapat menurunkan akurasi dari hasil klasifikasi. Berdasarkan hasil pengujian nilai *feature selection* yang cocok untuk menghasilkan akurasi klasifikasi yang tepat adalah nilai *feature selection* yang berkisar antara 10% sampai 40%.
- Keakuratan klasifikasi juga dipengaruhi pada jumlah *email* training. Semakin banyak jumlah *email* training akurasi klasifikasi yang dihasilkan makin akurat.

DAFTAR PUSTAKA

Auvil, Loretta & Searsmith, Duane. ***Using Text Mining for Spam Filtering***, Automated Learned Group National Center for Supercomputing Applications University of Illinois, http://algdocs.ncsa.uiuc.edu/PR-20031116-3.ppt, Diakses pada tanggal : 3 Febuari 2007.

Hearst, Marti. 17 Oktober 2003. ***What is text mining?***. http://www.sims.berkeley.edu/~hearst/text-mining.html

Kusumo, Ario Suryo. (2002). **Visual Basic.NET versi 2002 dan 2003**, Jakarta : PT Elex Media Komputindo.

Mitchell, Tom M. (1997). ***Machine Learning***. Singapore: McGraw Hill

Rickyanto, Isak. (2003). **Membuat Aplikasi Windows Dengan Visual Basic.NET**, Jakarta : PT Elex Media Komputindo.

Susanto, Budi. (2006). **Studi Email Mining : Email Clustering**, Institut Teknologi Bandung, 2006, Hal : 12.

Stopword list, http://web.inet-tr.org.tr/Online/Waishelp/stopemail.html, Diakses pada tanggal : 31 Januari 2007.

Tala, Fadillah Z. (2003). ***A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia***, Amsterdam : Universiteit van Amsterdam.

Tanenbaum, Adrew S. (2003). ***Computer Networks***, New Jersey : Pearson Education Inc.

Yung, Kok. (2005). **Membangun Aplikasi Database Dengan Visual Basic. NET 2005 dan Perintah SQL**, Jakarta : PT Elex Media Komputindo.

Weiss, Sholom M.; Indurkhya, Nitin; Zhang, Tong; and Dameru, Fred J.. (2005). ***Text Mining: Predictive Methods for Analyzing Untructured Information***, Springer Science+Business Media, Inc.