

## FILTERING SPAM EMAIL MENGGUNAKAN METODE NAIVE BAYES

Aria Wibisono<sup>1)</sup>, Sampurna Dadi Rizkiono<sup>2)</sup>, Agus Wantoro<sup>3)</sup>

Program Studi Informatika, Universitas Teknokrat Indonesia

Jl. H. ZA. Pagar Alam No. 9-11, Labuhan Ratu, Bandar Lampung

Email : ariawibisono7@gmail.com<sup>1)</sup>, sampurna.go@gmail.com<sup>2)</sup>, aguswantoro@teknokrat.ac.id<sup>3)</sup>

### Abstrak

*Spam* adalah email yang tidak diminta yang berisi promosi produk, pornografi, virus dan *content-content* yang tidak penting, yang dikirim ke banyak orang. Masalah *spam* dapat diatasi dengan adanya aplikasi *filtering email*, yaitu aplikasi yang secara otomatis mendeteksi sebuah email, apakah email tersebut *spam* atau bukan. *Naive Bayes* merupakan salah satu metode klasifikasi sederhana yang dapat digunakan untuk mengelompokkan data berdasarkan kriteria tertentu. Metode ini memanfaatkan teorema probabilitas yaitu mencari peluang terbaik, dengan memprediksi probabilitas di masa depan berdasarkan informasi di masa sebelumnya. Tujuan utama penelitian ini adalah mengkaji penerapan metode Naive Bayes untuk menentukan email spam dan email ham. Hasil pengujian aplikasi terhadap 5 email yang terdiri dari 2 email *spam* dan 3 email *ham* menunjukkan bahwa algoritma naive bayes mampu mengklasifikasikan email dengan akurasi 60%.

**Kata Kunci :** *spam, naive bayes*, klasifikasi, email.

### 1. Pendahuluan

Penggunaan internet telah menjadi kebutuhan yang penting dalam kehidupan berkomunikasi. Dari sekian banyak fasilitas di internet, salah satunya adalah surat elektronik atau yang lebih dikenal sebagai email. Electronic mail (email) merupakan media komunikasi di internet seperti untuk berdiskusi (maillist), transfer informasi berupa file (mail attachment) bahkan dapat digunakan untuk media iklan suatu perusahaan. Mengingat fasilitas email yang murah dan kemudahan untuk mengirimkan ke berapapun jumlah penerimanya maka beberapa pihak tertentu memanfaatkannya dengan mengirimkan email berisi promosi produk atau jasa, pornografi, virus, dan content-content yang tidak penting ke ribuan pengguna email. Email-email inilah yang biasanya disebut dengan spam. Untuk mengatasi masalah ini diperlukan suatu filter[1], salah satunya adalah klasifikasi, yang dapat memisahkan *spam e-mail* dan *bukan spam e-mail* [2]. Terdapat beberapa metode yang dapat digunakan dalam klasifikasi spam e-mail seperti Decision Tree, K-Nearest Neighbor (KNN), Naive Bayes, ID3 dan C4.5. Dari metode-metode tersebut, naive bayes merupakan metode statistik yang sederhana dan memiliki akurasi

yang baik serta error rate yang minimum dalam proses pengklasifikasian [3].

Naive bayes disebut juga multinomial naive bayes merupakan metode klasifikasi yang memanfaatkan metode probabilitas dan statistik. Metode ini merupakan model penyederhanaan dari algoritma bayes yang cocok dalam pengklasifikasian teks atau dokumen. Dalam klasifikasi menggunakan naive bayes, nilai kategori dari suatu dokumen akan ditentukan berdasarkan fitur/kata yang muncul dalam dokumen yang akan diklasifikasikan. Masalah tersebut dapat diatasi dengan adanya aplikasi email filtering [4][5]. Tetapi meskipun berbagai perangkat lunak email filtering banyak tersedia, masalah spam juga semakin berkembang, sehingga melahirkan beberapa metode email filtering. Salah satu metode email filtering yang paling populer yaitu metode Bayes (*Bayesian filtering*) [6]. Metode ini memanfaatkan teorema probabilitas yaitu teorema Bayes, dengan memprediksi probabilitas di masa depan berdasarkan informasi di masa sebelumnya.

Oleh karena itu penulis mencoba mengkaji penerapan metode Bayes pada proses email filtering dan kemudian menganalisis tingkat akurasi dari Bayesian filtering tersebut. Dengan melakukan penelitian judul "*Filtering Spam E-mail menggunakan metode Naive Bayes*".

### 2. Landasan Teori

#### 2.1. Klasifikasi Teks

Kategorisasi teks (atau Klasifikasi teks) adalah suatu proses yang mengelompokkan suatu teks ke dalam suatu kategori tertentu [7][8]. Dalam *text mining*, klasifikasi mengacu kepada aktifitas menganalisis atau mempelajari himpunan dokumen teks *pre-classified* untuk memperoleh suatu model atau fungsi yang dapat digunakan untuk mengelompokkan dokumen teks lain yang belum diketahui *class*-nya ke dalam satu atau lebih *class pre-defined* tersebut.

Dokumen yang digunakan untuk pembelajaran dinamakan contoh (*sample* atau *training dataset*) yang dideskripsikan oleh himpunan atribut atau variabel. Klasifikasi termasuk pembelajaran jenis *supervised learning*. Jenis lain adalah *unsupervised learning* atau

dikenal sebagai *clustering*. Pada *supervised learning*, data latihan mengandung pasangan data *input* (biasanya vektor) dan *output* yang diharapkan, sedangkan pada *unsupervised learning* belum ditentukan target *output* yang harus diperoleh.

Proses klasifikasi teks dapat dibagi ke dalam dua fase, yaitu fase *information retrieval* (IR) yakni untuk mendapatkan data numerik dari dokumen teks dan fase klasifikasi utama yakni dimana suatu algoritma memroses data numerik di atas untuk memutuskan ke kategori mana teks baru (bukan contoh) ditempatkan [9].

## 2.2. Pengertian Corpus

*Corpus* didefinisikan sebagai kumpulan sistematis teks alami termasuk bahasa lisan maupun tertulis dimana struktur dan isi dari *corpus* mengikuti prinsip-prinsip linguistik tertentu (Nesselhauf, 2005). *Corpus* yang digunakan dalam penelitian ini adalah *corpus* Lingspam.

## 2.3. Spam

*Spam* atau *junk mail* adalah penyalahgunaan dalam pengiriman berita elektronik untuk menampilkan berita iklan dan keperluan lainnya yang mengakibatkan ketidaknyamanan bagi para pengguna web [2]. Menurut Rahardjo (2006), *spam* adalah *unsolicited email* (email yang tidak diminta) yang dikirim ke banyak orang.

Berikut ini definisi dari *spam* menurut Lambert (2003).

1. Isi atau konten dari email tidak relevan dengan minat penerima.
2. Penerima tidak dapat menolak datangnya email yang tidak diminta tersebut dengan cara-cara lazim.
3. Dari sisi penerima, pengiriman dan penerimaan pesan tersebut memberikan keuntungan bagi pengirimnya.

Bentuk berita *spam* yang umum dikenal meliputi: *spam* pos-el, *spam* pesan instan, *spam* *usenet news-group*, *spam* mesin pencari informasi web (*web search engine spam*), *spam* blog, *spam* berita pada telepon genggam, *spam* forum internet, dan lain lain. *Spam* ini biasanya datang bertubi-tubi tanpa diminta dan sering kali tidak dikehendaki oleh penerimanya.

Menurut Rahardjo (2006), *spam* terjadi akibat murahnya biaya untuk mengirimkan *spam*. Biaya untuk mengirimkan satu email sama dengan seribu email, atau bahkan satu juta email.

*Spam* dapat dikategorikan sebagai berikut.

1. *Junk mail* yaitu email yang dikirimkan secara besar-besaran dari suatu perusahaan bisnis, yang sebenarnya tidak kita inginkan.

2. *Non-commercial spam*, misalnya surat berantai atau cerita humor yang dikirimkan secara masal tanpa tujuan komersial tertentu.
3. *Pornographic spam* yaitu email yang dikirimkan secara masal untuk mengirimkan gambar-gambar pornografi.
4. *Virus spam* yaitu email yang dikirimkan secara massal, dan mengandung virus atau Trojans.

## 2.4. Email Filtering

*Email filtering* adalah suatu proses yang otomatis akan mendeteksi sebuah email, apakah email tersebut sebuah *spam* atau bukan (*ham email*). Beberapa metode yang dapat digunakan untuk *email filtering* antara lain *Keyword filtering*, *Black listing* dan *White listing*, *Signature-Based filtering*, *Naive Bayesian (statistical) filtering*.

Beberapa karakteristik *email filtering* yaitu.

1. *Binary class*  
*email filtering* hanya mengklasifikasikan email ke dalam kelas *spam* dan *legitimate email*.
2. Prediksi  
*email filtering* mampu melakukan prediksi kelas dari suatu email.
3. Komputasi mudah  
mengingat sifat data email yang memiliki dimensi tinggi maka dibutuhkan sebuah *email filter* yang mampu melakukan komputasi dengan mudah.
4. *Learning*  
mampu melakukan *learning* dari email-email yang sudah ada sebelumnya.
5. Kinerja yang bagus  
memiliki akurasi yang tinggi, meminimalkan nilai *false positive* dan mentolerir nilai *false negative* yang cukup tinggi.

## 2.5. Naive Bayes

*Naive Bayes* atau multinomial *naive bayes* merupakan metode yang digunakan untuk mengklasifikasikan sekumpulan dokumen. Algoritma ini memanfaatkan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. Metode NB menempuh dua tahap dalam proses klasifikasi teks, yaitu tahap pelatihan dan tahap pengujian (klasifikasi). Pada tahap pelatihan dilakukan proses analisis terhadap sampel dokumen berupa pemilihan vocabulary, yaitu kata yang mungkin muncul dalam koleksi dokumen sampel yang sedapat mungkin dapat menjadi representasi dokumen. Selanjutnya adalah penentuan probabilitas prior bagi tiap kategori berdasarkan sampel dokumen. Pada tahap klasifikasi ditentukan nilai kategori dari suatu dokumen berdasarkan *term* yang muncul dalam dokumen yang diklasifikasi.

Dalam *naïve bayes*, kemungkinan dokumen  $d$  berada di *class*  $C$  dihitung sebagai berikut:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq nd} P(tk|c) \quad (2.1)$$

dimana  $P(tk|c)$  adalah *conditional probability* dari fitur  $tk$  yang terdapat dalam dokumen dari *class*  $C$ . Dapat diartikan,  $P(tk|c)$  adalah ukuran berapa banyak kemunculan fitur  $tk$  memberikan kontribusi bahwa  $C$  adalah *class* yang benar.  $P(c)$  adalah *prior probability* dari dokumen yang terdapat di *class*  $C$ . Jika fitur dari sebuah dokumen tidak memberikan *evidence* yang jelas untuk sebuah *class* dibandingkan dengan *class* lainnya, maka fitur dengan *prior probability* tertinggi yang akan dipilih. Token dalam  $d$  ( $t_1, t_2, \dots, t_{nd}$ ) merupakan bagian dari *vocabulary* yang digunakan untuk klasifikasi dan  $nd$  adalah jumlah token tersebut dalam  $d$ .

Tujuan utama dalam klasifikasi teks adalah menemukan *best class* untuk sebuah dokumen. *Best class* dalam *naïve bayes* adalah yang paling mungkin atau *maximum a posteriori* (MAP) *class*  $C_{map}$ :

$$C_{map} = \arg \max_{C \in C} P(c|d) = \arg \max_{C \in C}$$

$$P(c) \prod_{1 \leq k \leq nd} P(tk|c) \quad (2.2)$$

dimana  $\arg \max$  adalah *argument maximum* dan untuk  $P$  ditulis  $\hat{P}$  karena tidak diketahui nilai sebenarnya dari parameter  $P(c)$  dan  $P(tk|c)$ .

Pada Persamaan (2.2), banyak *conditional probability* yang dikalikan, satu untuk masing-masing posisi  $1 \leq k \leq nd$ . Hal ini dapat mengakibatkan masalah *underflow*. Masalah *underflow* adalah masalah yang muncul pada dokumen yang memiliki jumlah kata yang sangat besar. Hasil perkalian dari nilai-nilai *conditional probability* dari seluruh kata yang berjumlah sangat besar akan membuat variabel *score* bernilai sangat kecil. Nilai *score* yang sangat kecil dapat menimbulkan kesalahan saat dilakukan proses perbandingan. Oleh karena itu, lebih baik untuk melakukan perhitungan dengan menambahkan logaritma probabilitas daripada mengalikan probabilitas. *Class* dengan nilai probabilitas tertinggi masih yang paling mungkin. Oleh karena itu maksimalisasi yang sebenarnya dilakukan dalam kebanyakan implementasi dari *naïve bayes* adalah:

$$C_{map} = \arg \max_{C \in C} [\log \hat{P}(c) + \sum_{1 \leq k \leq nd} \log \hat{P}(tk|c)] \quad (2.3)$$

untuk menghitung nilai dari  $\hat{P}(c)$  adalah sebagai berikut:

$$\hat{P}(c) = \frac{N_c}{N} \quad (2.4)$$

( $N_c$  adalah dokumen yang berada di *class*  $C$  dan  $N$  adalah jumlah dokumen) Diperkirakan *conditional probability*  $\hat{P}(t|c)$  sebagai frekuensi relatif dari fitur  $t$  dalam dokumen-dokumen di *class*  $C$  dapat dihitung dengan persamaan:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}} \quad (2.5)$$

( $T_{ct}$  adalah jumlah kemunculan fitur  $t$  dalam *training* dokumen dari *class*  $C$ ).

Persamaan (2.3) memiliki interpretasi yang sederhana. Setiap kondisi parameter  $\log P(tk|c)$  adalah bobot yang menunjukkan seberapa baik indikator  $tk$  untuk  $C$ . Demikian pula  $\hat{P}(c)$  adalah bobot yang menunjukkan frekuensi relatif  $C$ . Hasil penjumlahan *log prior probability* dan bobot fitur adalah ukuran tentang berapa banyak kemunculan yang ada untuk dokumen di *class*  $C$  dan Persamaan (2.3) memilih *class* dengan *evidence* terbanyak.

Persamaan (2.5) akan menimbulkan masalah baru apabila fitur tidak ditemukan dalam *training set*. Fitur yang tidak ditemukan menyebabkan masalah pembagian dengan nol (*division by zero*). Untuk mengatasi hal tersebut maka digunakan *add-one* atau *Laplace Smoothing*, seperti tampak pada Persamaan (2.6).

$$\hat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V} (T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum_{t' \in V} T_{ct'})+B} \quad (2.6)$$

dimana  $B = |V|$  adalah jumlah fitur dalam *vocabulary*.

## 2.6. Klasifikasi Teks Menggunakan Naive Bayes

Di dalam proses klasifikasi menggunakan multinomial *naïve bayes*, akan dilakukan terlebih dahulu *pre-processing e-mail*.

### 1. Preprocessing

Tahap *pre-processing e-mail* dilakukan untuk menghilangkan fitur-fitur yang tidak penting dan membersihkan seluruh dokumen *e-mail* dari *markup tag* sehingga saat digunakan akan meningkatkan akurasi dari *classifier*. Berikut ini merupakan langkah-langkah umum dalam *pre-processing e-mail*:

1. Tokenisasi  
Langkah awal adalah mengekstrak isi dokumen *e-mail* ke dalam bentuk kata/fitur atau lebih dikenal

dengan nama token. Proses ini dinamakan tokenisasi. Pada tahap ini juga dilakukan *case folding*, yaitu penyeragaman bentuk huruf dimana seluruh huruf diseragamkan ke bentuk huruf kecil yaitu antara huruf “a” sampai huruf “z” dan juga menghilangkan karakter-karakter tertentu seperti tanda baca dan angka.

## 2. Eliminasi *stopword*

*Stopwords* adalah kata-kata yang memiliki frekuensi kemunculan yang tinggi dalam dokumen namun tidak memiliki nilai informasi yang tinggi. Contoh *stopword* misalnya kata “the”, “and”, “a”, “of”, “in”, “is”, “this”, “that”. Dalam penelitian yang dilakukan oleh Silvatt dan Ribeiro (2003), mereka menyimpulkan bahwa penghapusan *stopwords* memiliki efek yang signifikan untuk meningkatkan akurasi *classifier*.

## 3. Lemmatisasi fitur

Proses lemmatisasi (*lemmatization*) akan mereduksi kata ke dalam bentuk kata dasar atau sering disebut lemma dimana kata dasar (lemma) tersebut memiliki arti dan ditemukan dalam kamus. Misalnya, kata “walked”, “walks”, dan “walking” jika dilakukan proses lemmatisasi maka lemmanya adalah “walk” (kata “walk” ada di dalam kamus Bahasa Inggris).

## 2.7. Klasifikasi Email

Penggunaan metode *naïve bayes* terdiri dari dua fase, yaitu fase pelatihan dan fase pengujian. Berikut ini adalah tahap-tahap pada fase pelatihan dan fase pengujian:

### a. Fase Pelatihan (*training*):

Fase pelatihan adalah sebagai berikut:

1. Ekstrak seluruh kata dari seluruh dokumen dalam dokumen latih kemudian buat tabel representasi dokumen latih.
2. Hitung *prior probability P class* untuk setiap *class* dengan menggunakan rumus pada persamaan (2.4).
3. Hitung *conditional probability* dari semua kata dan *class* dengan menggunakan rumus pada persamaan (2.6).

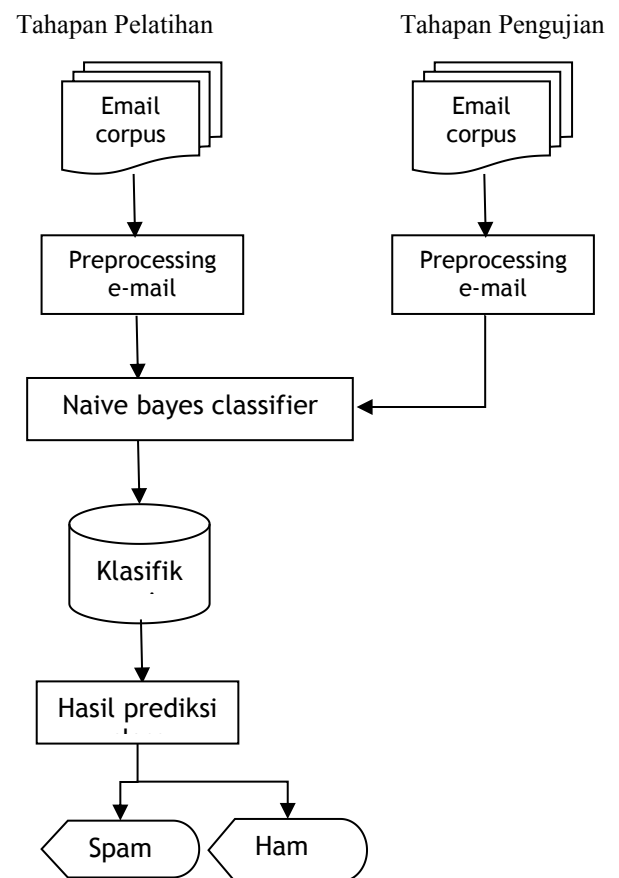
### b. Fase Pengujian (*testing*):

Fase pengujian adalah sebagai berikut:

1. Ekstrak seluruh kata dari seluruh dokumen dalam dokumen latih.
2. Hitung bobot (*score*) dari dokumen *d* yang termasuk dalam *class C* dengan menggunakan rumus pada persamaan (2.2).
3. Prediksi *class* dokumen uji dengan cara memilih *class* yang memiliki skor terbesar berdasarkan persamaan (2.3).

## 3. Metode Penelitian

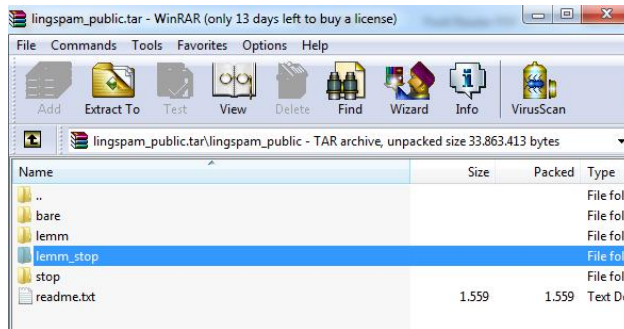
Tahapan yang digunakan untuk perhitungan menggunakan algoritma *Naive Bayes Classification* sebagai berikut:



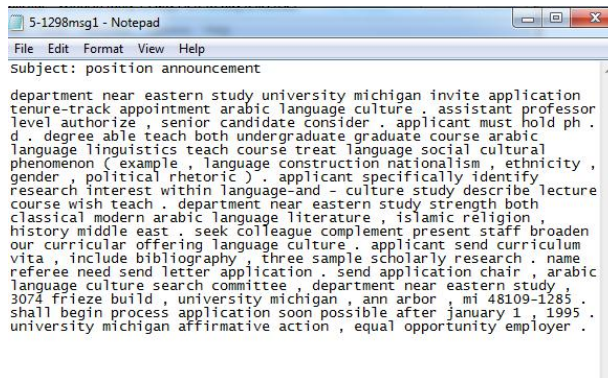
Gambar 3.1 Tahapan Penelitian

### 3.1. Corpus

*Corpus* Lingspam yang terdiri dari 2893 *e-mail* yang terdiri dari 2412 *ham e-mail* (*e-mail* yang bukan *spam*) dan 481 *spam e-mail* yang terbagi dalam 10 *part*. *E-mail* yang digunakan dalam tahap pelatihan diambil dari *part 2*, sedangkan *e-mail* dari *part 1* akan digunakan dalam tahap pengujian, tetapi tidak digunakan semua dikhawatirkan akan memboroskan memori pada saat digunakan dalam aplikasi. Gambar 3.2 merupakan folder dalam *corpus* Lingspam dan bagian yang digunakan yang berada di dalam folder “*lemm\_stop*” yang berarti sudah dilakukan lemmatisasi dan eliminasi *stopwords*. Gambar 3.2 merupakan contoh *e-mail* yang digunakan.



Gambar 3.2 Corpus Lingspam



Gambar 3.3 Contoh e-mail ham

### 3.2. Prosedur dan Pengumpulan Data

Pengumpulan data merupakan tahapan persiapan yang harus dilaksanakan terlebih dahulu sebelum dilakukan penelitian [10]. Berikut merupakan aktivitas yang dilaksanakan dalam pengumpulan sumber teori:

#### 1. Studi Literatur

Berfungsi sebagai pendukung dari penelitian yang akan dilaksanakan. Teori-teori yang digunakan bersumber dari buku, jurnal dan penelitian-penelitian sejenis yang dapat mendukung pemecahan masalah dalam penelitian yang dilakukan.

#### 2. Dokumentasi (Document)

Merupakan metode pengumpulan data dengan cara membaca, mencatat, mengutip, dan mengumpulkan data-data secara teoritis dari buku-buku dan internet sebagai landasan penyusunan penelitian [11]. Peneliti meminjam buku di perpustakaan, mencari data dari internet juga dilakukan untuk referensi laporan ini, dimana teori tersebut diletakkan pada landasan teori.

### 3.3. Preprocessing E-mail

Dalam penelitian ini pada tahap *preprocessing*, penulis tidak melakukan proses lemmatisasi fitur dan eliminasi *stopwords* karena dataset Lingspam yang digunakan sudah terlebih dahulu melakukan proses-proses

tersebut. Pada tahap ini, *input* berupa dokumen *e-mail* akan diekstrak ke dalam bentuk kata/fitur. Dalam tahap ini juga dilakukan *case folding* dan menghilangkan karakter “-” karakter tertentu seperti tanda baca dan angka.

Subject	track
position	appointment
announcement	arabic
department	language
near	culture
eastern	assistant
study	professor
university	level
michigan	authorize
invite	senior
application	candidate
tenure	consider

Gambar 3.4 Tokenisasi e-mail

### 3.4. Ilustrasi Klasifikasi Email Menggunakan Naive Bayes

Pada tahap ini akan dilakukan proses klasifikasi dimana akan didapatkan model klasifikasi berdasarkan *input* yang diberikan. Tabel 3.1 merupakan contoh dokumen latih dan uji yang kemudian akan dicari *class* yang paling cocok untuk dokumen uji (Dok 6).

Tabel 1 Tabel dokumen latih dan dokumen uji

	Dok	Daftar kata dalam email	class
Dokumen latih	1	Position, announcement, department	Ham
	2	Job, announcement, department	Ham
	3	Job, posting, announcement	Ham
	4	Adult, nature, sexual	Spam
	5	Social, security, number, criminal	Spam
Dokumen uji	6	Tautologous, name, Mcnamara, level	?

Untuk menyelesaikan kasus di atas, tahapan yang dilakukan adalah:

1. Buat tabel representasi dokumen latih seperti tampak pada Tabel 2.

Tabel 2 Tabel Representasi Dokumen Latih

	Ham	Ham	Ham	Spam	Spam
Position	1				
Announcement	1	1	1		
Deparment	1	1			
Job		1	1		
Posting			1		
Adult				1	
Nature				1	
Sexual				1	
Social					1
Security					1

Number					1
Criminal					1
Tautologous					
Name					
Mcnamara					
Level					

Pada model multinomial, kata-kata yang muncul berulang diikutsertakan di dalam perhitungan probabilitas. Pada Tabel 3.2, kalimat pada Dok 1 yaitu "Position, announcement, department" akan direpresentasikan dengan cara menghitung frekuensi kemunculan kata "Position", yaitu 1 kali, kata "Announcement dan Department" muncul 1 kali, dan kata-kata lain memiliki frekuensi kemunculan nol.

2. Hitung *prior probability* dari dokumen latih. Merujuk pada Tabel 3.1 maka dengan menggunakan rumus pada Persamaan (2.5), diperoleh hasil  $P(ham) = 3/5 = 0,6$  dan  $P(spam) = 2/5 = 0,4$
3. Hitung *conditional probability* menggunakan Persamaan (2.6). Detail perhitungan conditional probability adalah :  
 $P(position|ham) = P(posting|ham) = (1+1)/(9+16) = 0,088$

$$P(announcement|ham) = (3+1)/(9+16) = 0,16$$

$$P(department|ham) = P(job|ham) = (2+1)/(9+16) = 0,12$$

$$P(adult|ham) = P(nature|ham) = P(sexual|ham) = P(social|ham) = P(severity|ham) = P(number|ham) = P(criminal|ham) = P(tautologous |ham) = P(name|ham) = P(mcnamara|ham) = P(level|ham) = (0+1)/(9+16) = 0,04$$

$$P(position|spam) = P(posting|spam) = P(announcement|spam) = P(department|spam) = P(job|spam) = P(tautologous |spam) = P(name|spam) = P(mcnamara|spam) = P(level|spam) = (0+1)/(7+16) = 0,043$$

$$P(adult|spam) = P(nature|spam) = P(sexual|spam) = P(social|spam) = P(severity|spam) = P(number|spam) = P(criminal|spam) = (1+1)/(7+16) = 0,086$$

4. Hitung *score* dari setiap class pada dokumen uji menggunakan Persamaan (2.2).  
 $Score(ham) = P(ham) \times P(tautologous |ham) \times P(name|ham) \times P(mcnamara|ham) \times P(level|ham) = (3/5) \times (0,04)4 = 0,096$

Sedangkan score untuk class "spam" adalah:  
 $Score(spam) = P(spam) \times P(tautologous |spam) \times P(name|spam) \times P(mcnamara|spam) \times P(level|spam) = (2/5) \times (0,043)4 = 0,0688$

Prediksi *class* dokumen uji dengan cara memilih class yang memiliki skor terbesar. Dengan menggunakan Persamaan (2.3) tampak bahwa  $Ham > Spam$ , maka Dok 6 masuk dalam class "Ham".

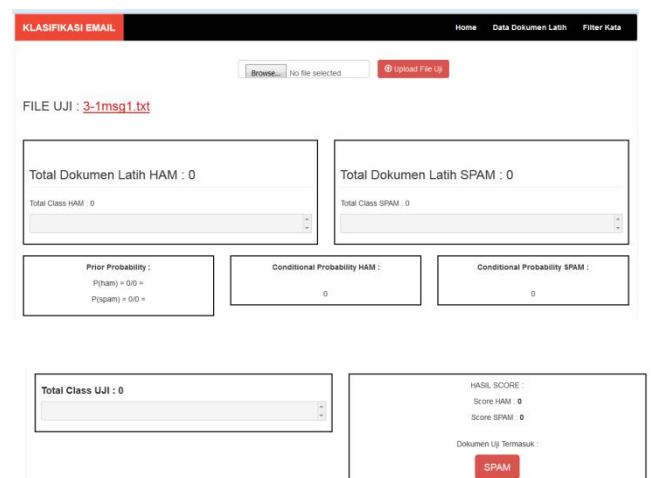
## 4. Hasil Dan Pembahasan

### 4.1.1. Implementasi Perangkat Lunak

Implementasi dari metode naive bayes yang diterapkan pada dreamweaver dapat dilihat sebagai berikut:

### 4.1.2. Tampilan Awal

Desain fisiknya dapat dilihat pada Gambar 4.1



Gambar 4.1 Tampilan Awal

Pada sistem ini ada 3 buah menu bar yaitu, menu home, menu data dokumen latih dan menu filter kata. Pada menu home terdapat button browse untuk memilih file yang akan diuji lalu tekan button upload untuk mengupload file ujinya, dengan nama file (contoh : 3-1msg1.txt).

### 4.1.3. Menu Data Dokumen Latih

Tampilan menu dokumen latih dapat diliha pada gambar 4.2 berikut ini.

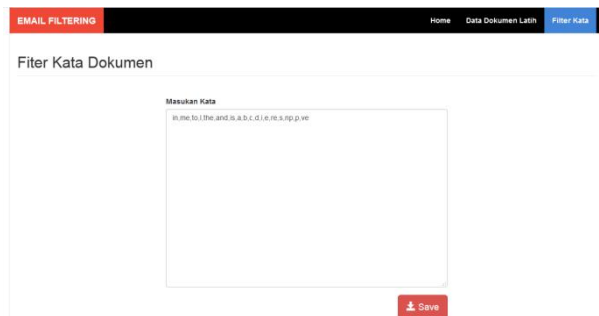
No.	Nama File	Jenis
1	5-129msg1.txt	HAM
2	5-129msg2.txt	HAM
3	5-129msg3.txt	HAM
4	spmsg3.txt	SPAM
5	spmsg15.txt	SPAM

Gambar 4.2 Menu Data Dokumen Latih

Pada menu data dokumen latihan diatas berfungsi untuk mengupload file ham dan file spam sebagai dokumen latihan, dengan cara klik browse pilih file yang akan di upload lalu klik button upload (sebagai contoh terlampir 3 file ham dan 2 file spam), kemudian ada button hapus ham dan hapus spam untuk menghapus file yang sudah di upload.

**4.1.4. Menu Filter Kata**

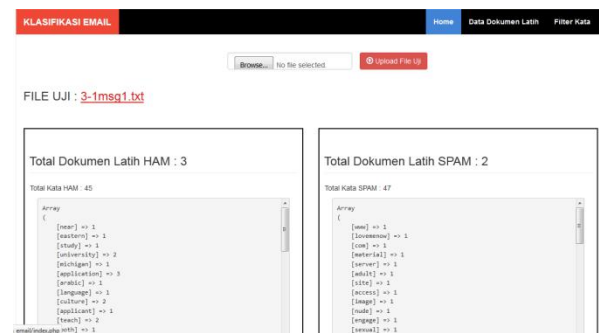
Kemudian di menu filter kata terdapat kolom untuk menginputkan kata kata yang memiliki frekuensi kemunculan yang tinggi dalam dokumen tetapi tidak memiliki nilai informasi yang tinggi (contoh : the, and, a, of, in, is, that, dll), seperti pada Gambar 4.3 dibawah ini.



Gambar 4.3 Menu Filter Kata

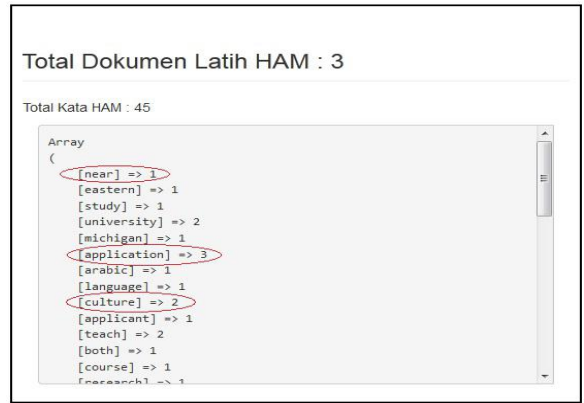
**4.1.5. Menu Home**

Pada menu home terdapat kotak bagian total dokumen latihan ham dan kotak bagian total dokumen latihan spam yang berfungsi untuk melihat total class/kata yang sering muncul pada 3 file ham dan 2 file spam yang di upload seperti gambar 4.4 ini.



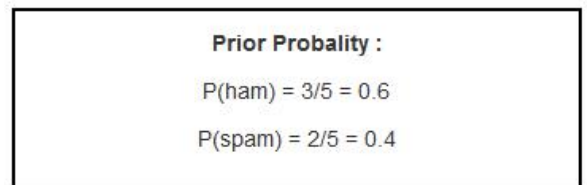
Gambar4.4 Class dokumen ham & spam

Lalu pada gambar 4.5Maksudnya, kata near muncul paling sedikit 2 kali pada 1 file tertentu, kata application muncul paling sedikit 2 kali pada 3 file tertentu, kata culture muncul paling sedikit 2 kali pada 2 file tertentu.



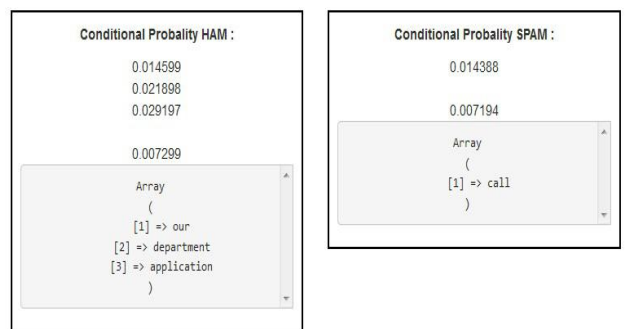
Gambar 4.5 Class Dokumen Latihan

Dari gambar 4.6 di bawah ini contoh  $P(\text{ham}) = 3/5 = 0,6$ , dimana 3 adalah dokumen latihan yang berada di class ham, 5 adalah jumlah total dokumen latihan, begitu juga dengan class spam.



Gambar 4.6 Prior Probability

Conditional probability yaitu untuk menghitung kata dari setiap class untuk menentukan nilai yang digunakan dalam dokumen uji nantinya. Dalam contoh ini nilai yang digunakan adalah 0.007874 untuk class ham dan 0.007463 untuk class spam.



Gambar 4.7 Conditional Probability

$$P = \frac{T_n+1}{T_c+B'}$$

$T_n$  = jumlah kemunculan kata dalam dokumen latihan pada class.

$T_c$ = jumlah kata yang muncul dalam setiap class.

$B'$ = total jumlah kata pada class.

## Perhitungan HAM

$$1. P = \frac{1+1}{45+92} = 0,014599$$

$$2. P = \frac{2+1}{45+92} = 0,021898$$

$$3. P = \frac{3+1}{45+92} = 0,029197$$

$$4. P = \frac{0+1}{45+92} = 0,007299$$

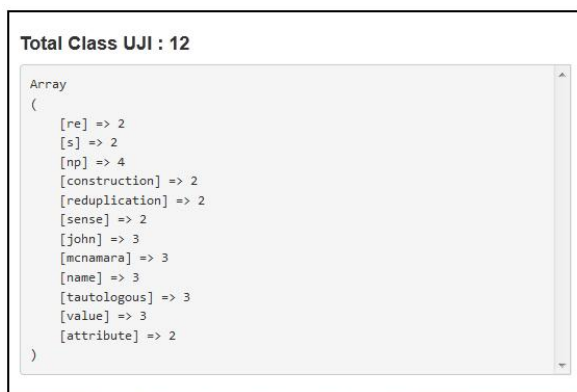
## Perhitungan SPAM

$$1. P = \frac{1+1}{47+92} = 0,014388$$

$$2. P = \frac{0+1}{47+92} = 0,007194$$

Maka yang digunakan adalah nilai nomer 4 untuk ham dan nilai nomer 2 untuk spam karena, kata/fitur yang terdapat pada dokumen uji tidak ada yang sama dengan di dokumen latih.

Pada dokumen uji ada 12 kata yang paling sering muncul, kata mcnamara, name, toutologous, value muncul sebanyak 3 kali, kata construction, reduplication dan sense muncul sebanyak 2 kali dan lain-lain. 12 kata ini yang nantinya akan di kalikan dengan nilai conditional probability.



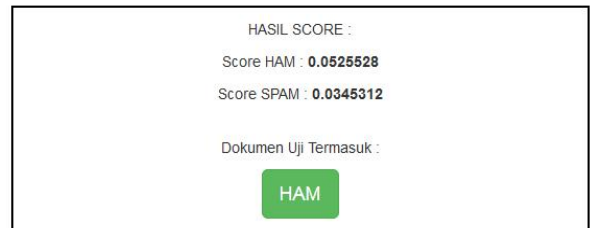
Gambar 4.8 Class Dokumen Uji

Hasil score didapat dari prior probability dikali (conditional probability dikali jumlah kata dokumen uji).

$$\text{HAM} : 0,6 \times (0,007299) 12 = 0,0525528$$

$$\text{SPAM} : 0,4 \times (0,007194) 12 = 0,0345312$$

Jadi, hasilnya jika score ham lebih besar daripada score spam maka, dokumen uji tersebut masuk pada class Ham.



Gambar 4.9 Hasil Score

Dari contoh percobaan diatas dimana dokumen yang digunakan adalah 5 dokumen, dengan 3 dokumen Ham (5-1298msg1.txt, 5-1298msg2.txt dan 5-1298msg3.txt) dan 2 dokumen Spam (spmsga3.txt, dan spmsga15.txt) didapat nilai akurasi :

$$\text{Akurasi} = \frac{\text{jumlahprediksibenar}}{\text{jumlahdatauji}} \times 100\%$$

$$\text{Akurasi} = \frac{3}{5} \times 100\%$$

$$\text{Akurasi} = 0,6 \times 100\%$$

$$\text{Akurasi} = 60\%$$

## 5. Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Metode *Naïve Bayes Classifier* dapat digunakan untuk melakukan klasifikasi email spam berbahasa inggris.
2. Jumlah data latih sangat berpengaruh dalam proses klasifikasi, karena jika kata yang sering muncul pada setiap class nya lebih banyak dan dokumen latih pada setiap class nya juga lebih banyak maka akan meningkatkan nilai akurasi pada class tersebut.
3. Hasil klasifikasi tergantung dari kata yang ada pada data latih.

## 6. Saran

1. Diperlukan adanya penelitian lanjutan pada *naïve bayes classifier* khususnya dalam hal menangani *dataset* yang memiliki jumlah data yang tidak seimbang di dalam *class (unbalanced data)* serta penggunaan dataset berbahasa indonesia agar lebih mudah digunakan oleh pembaca.
2. Aplikasi dapat dikembangkan lebih lanjut sehingga mampu melakukan analisis terhadap *body* email bertipe html atau *mime* dan *attachment* yang dianalisa dari *contentnya*



## Daftar Pustaka

- [1] M. Sharma, "A Survey of Email Spam Filtering Methods," vol. 7, pp. 14–21, 2018.
- [2] B. Kurniawan, S. Effendi, and O. S. Sitompul, "Klasifikasi Konten Berita Dengan Metode Text Mining," *J. Dunia Teknol. Inf.*, vol. 1, no. 1, pp. 14–19, 2012, [Online]. Available: <http://download.portalgaruda.org/article.php?article=58993&val=4123>.
- [3] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for Spam filtering," *J. Comput. Sci.*, vol. 12, no. 2, pp. 66–73, 2012.
- [4] M. Bakri, "Penerapan Data Mining untuk Clustering Kualitas Batu Bara dalam Proses Pembakaran di PLTU Sebalang Menggunakan Metode K-Means," *J. Teknoinfo*, vol. 11, no. 1, p. 6, 2017, doi: 10.33365/jti.v11i1.3.
- [5] A. S. Rajput, V. Athavale, and S. Mittal, "Intelligent model for classification of SPAM and HAM," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 6, pp. 773–777, 2019.
- [6] A. Hamzah, "Klasifikasi Teks Dengan Naïve Bayes Classifier (NBC) Untuk Pengelompokan Teks Berita Dan Abstract Akademis," *Pros. Semin. Nas. Apl. Sains Teknol. Periode III*, no. 2011, pp. 269–277, 2012, doi: 1979-911X.
- [7] R. Kaur and P. Luthra, "Load Balancing in Cloud Computing," pp. 1–8, 2014.
- [8] D. Damayanti and H. Sulistiani, "Sistem Informasi Pembayaran Biaya Sekolah Pada SD Ar-Raudah Bandar Lampung," *J. Teknoinfo*, vol. 11, no. 2, p. 25, 2017, doi: 10.33365/jti.v11i2.23.
- [9] A. S. Patil and B. V. Pawar, "Automated classification of web sites using Naive Bayesian algorithm," *Lect. Notes Eng. Comput. Sci.*, vol. 2195, pp. 519–523, 2012.
- [11] Jupriyadi, J. (2018). *Implementasi Seleksi Fitur Menggunakan Algoritma Fvbrm Untuk Klasifikasi Serangan Pada Intrusion Detection System (Ids)*. Prosiding Semnastek.
- [12] E. R. Susanto and A. S. Puspaningrum, "Model Prioritas Program Pemerataan Ipm Di Provinsi Lampung Menggunakan Metode Analytic Hierarchy Process," *J. Teknoinfo*, vol. 14, no. 1, p. 9, 2020, doi: 10.33365/jti.v14i1.543.