

# ANALISIS METODE NAIVE BAYESAN PADA APLIKASI SPAM FILTERING BERBASIS JAVA DESKTOP UNTUK MENGLASIFIKASIKAN SUREL

Subandi<sup>1</sup>, Ahmad Luthfi<sup>2</sup>, Zaniel Mazalisa<sup>3</sup>  
Dosen Universitas Bina Darma<sup>1</sup>, Mahasiswa Universitas Bina Darma<sup>2</sup>  
Jalan Jenderal Ahmad Yani No.12 Palembang  
[s.subandi@rocketmail.com](mailto:s.subandi@rocketmail.com)<sup>1</sup>, [luthfi.ubd@gmail.com](mailto:luthfi.ubd@gmail.com)<sup>2</sup>, [zaniel@gmail.com](mailto:zaniel@gmail.com)<sup>3</sup>

---

**Abstract :** Spam mail is an email sent to thousands of email users continuously and typically contain promotional product / service / business, pornography, viruses and emails do not want. Until now, the problem of spam mails continue to evolve as the development of email filtering software by using various methods such as Naive Bayesian classification, address block, association rules and various other methods. Of the many methods of email filtering, klasifikasi naive bayesian have a high degree of accuracy. In this study, I will attempt to explain the email filtering using naive Bayesian classification method. Naive Bayesian filter is built from a set of emails that have been classified as spam mail and legitimate mail. The contents of the email in tokenizer, selected features are significant and thus counted as spam probability value token using the rules naive bayes. From the results of this classification filter built a database that is used to identify the email as spam or legitimate mail. Naive bayes filter email by calculating the probability of classifying email based on the probability value token database filters that have been built earlier.

**Keywords :** email filtering, Bayesian filters, naive bayes, spam mail, legitimate mail.

**Abstrak :** *Spam mail* merupakan surel yang dikirimkan ke ribuan pengguna surel secara terus menerus dan biasanya berisi promosi produk/jasa/usaha, pornografi, virus dan surel yang tidak kita inginkan. Hingga saat ini permasalahan spam mail terus berkembang seiring berkembangnya perangkat lunak *email filtering* dengan menggunakan berbagai metode seperti klasifikasi *naive bayesian*, *address block*, aturan asosiasi dan berbagai metode yang lain. Dari sekian banyak metode *email filtering*, klasifikasi *naive bayesian* memiliki tingkat akurasi yang tinggi. Pada penelitian ini penulis akan mencoba untuk menjelaskan *email filtering* dengan menggunakan metode klasifikasi *naive bayesian*. *Naive bayesian filter* dibangun dari sekumpulan surel yang telah diklasifikasikan ke dalam *spam mail* dan *legitimate mail*. Isi dari *email* ditokenisasi, dipilih *feature* yang signifikan dan kemudian dihitung nilai probabilitas token sebagai *spam* dengan menggunakan aturan *naive bayes*. Dari hasil klasifikasi tersebut dibangun sebuah basis data *filter* yang digunakan untuk mengidentifikasi surel sebagai *spam* atau *legitimate mail*. *Naive bayes filter* mengklasifikasikan surel dengan menghitung probabilitas surel berdasarkan nilai probabilitas token pada basis data *filter* yang telah dibangun tadi.

**Kata Kunci :** *email filtering*, *bayesian filter*, *naive bayes*, *spam mail*, *legitimate mail*.

---

## 1. PENDAHULUAN

### 1.1. Latar Belakang

Surel merupakan media komunikasi di *internet* seperti untuk berdiskusi (*mailing list*), transfer informasi berupa *file* (*mail attachment*) bahkan dapat digunakan untuk media iklan suatu

perusahaan atau produk tertentu. Mengingat fasilitas surel yang murah dan kemudahan untuk mengirimkan ke berapapun jumlah penerimanya maka beberapa pihak tertentu memanfaatkannya dengan mengirimkan surel berisi promosi produk atau jasa, pornografi, virus, dan konten-konten yang tidak penting ke ribuan pengguna surel.

Surel inilah yang biasanya disebut dengan *spam mail*. Dampak buruk yang paling utama dari adanya *spam mail* adalah terbuangnya waktu dengan percuma untuk menghapus *spam mail* dari kotak masuk satu persatu. Meskipun berbagai perangkat lunak surel *filtering* banyak tersedia, namun masalah *spam mail* juga semakin berkembang.

Berbagai upaya telah dilakukan untuk mengatasi permasalahan *spam*. Pada mulanya proses penyaringan surel *spam* dilakukan dengan pendekatan *rule-based*. surel akan dikategorikan sebagai *spam* menurut aturan-aturan tertentu seperti kemunculan kata, alamat pengirim, dan struktur *header*. Pendekatan ini secara praktik kurang efektif dan memiliki tingkat *false positive* yang tinggi. Pendekatan berbasis konten *statistic* (menggunakan metode *Naive Bayes Classifier*) untuk menyaring pesan *spam* pertama kali diteliti oleh Pantel dan Lin (1998) dan berhasil mencapai tingkat akurasi 92% dengan tingkat *false positive* 1,16%. Teknik serupa juga digunakan oleh Sahami et al (1998) meskipun kinerjanya tidak setinggi *spam-filter* yang dirancang oleh Pantel dan Lin (1998). Graham (2002) membahas teknik *spam-filtering* menggunakan metode *Naive Bayes Classifier* (NB) dengan pendekatan yang cukup berbeda jika dibandingkan dengan metode *Naive Bayes Classifier* pada umumnya. Metode ini diklaim berhasil mencapai tingkat akurasi sebesar 99,95% dengan *false positive* sebesar 0,05%. Kinerja yang cukup tinggi ini membuat metode *content-based filtering* semakin banyak digunakan dalam aplikasi *spam-filter* (Yerazunis 2004).

Oleh karena itu pada penelitian ini, penulis mencoba menjelaskan surel *filtering* untuk mengotomatisasikan proses pemilahan *spam mail* dan *legitimate mail* (bukan *spam mail*). Salah satu metode *email filtering* yang paling populer yaitu *naive bayesian filtering*. Metode ini memanfaatkan teorema probabilitas yaitu teorema *bayes* dan fungsionalitas *data mining* yaitu klasifikasi *naive bayesian*. Kelebihan *naive bayesian filtering* diantaranya adalah tingkat akurasi yang tinggi dan kesalahan yang minimum. Hal inilah yang melatarbelakangi penulis untuk menjelaskan metode ini.

## 1.2. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membuat sebuah aplikasi yang dapat mengklasifikasikan surel yang *spam* dan bukan *spam* dengan tingkat akurasi yang tinggi menggunakan metode *Naive Bayesian*.

## 1.3. Manfaat Penelitian

Penelitian ini diharapkan bisa memberikan manfaat bagi penulis sendiri maupun Universitas Bina Darma Palembang. Manfaat tersebut antara lain:

1. Untuk Universitas Bina Darma Palembang. Hasil penelitian ini supaya dapat menambah bacaan ilmiah bagi para mahasiswa dan mahasiswi Universitas Bina Darma Palembang.
2. Penulis. Dapat berbagai pengetahuan dan pengalaman berdasarkan perbandingan

teoritis dan praktis dilapangan terutama pada *spam filtering java desktop* serta membuka pemikiran yang baru sebagai pelengkap teori yang diterima dibangku kuliah.

## 2. METODOLOGI PENELITIAN

### 2.1 Metode Pengumpulan Data

Dalam melakukan penelitian untuk mendapatkan data dan informasi, maka metode yang digunakan dalam proses pengumpulan data yang akan dilakukan adalah Studi Kepustakaan, yaitu studi data yang dilakukan melalui penelusuran *literatur* atau buku-buku referensi pendukung sebagai landasan berfikir atau teori dan dari data-data statistik yang di dapat sebagai input sistem.

### 2.2 Metode Pengembangan Sistem

Menurut Al Fatta (2007) Beberapa ahli membagi proses-proses pengembangan sistem ke dalam sejumlah urutan yang berbeda-beda. Tetapi semuanya akan mengacu pada proses-proses standar berikut:

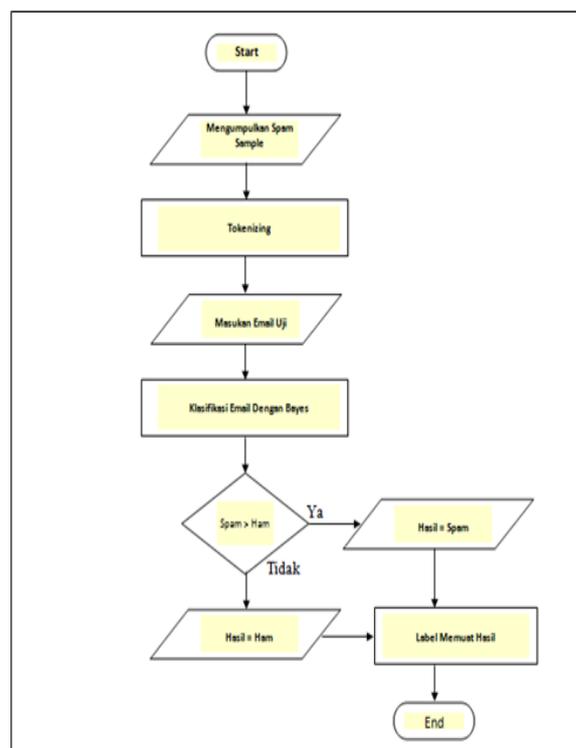
1. Analisis
2. Desain
3. Implementasi
4. Pemeliharaan

Pada perkembangannya, proses-proses standar tadi dituangkan dalam satu metode yang terkenal dengan nama *System Development Life Cycle* (SDLC) yang merupakan metodologi

umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain .

### 2.3 Algoritma Sistem Klasifikasi Email

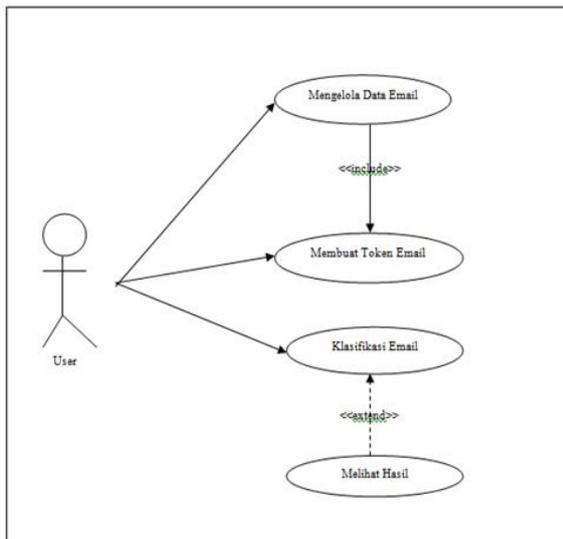
Algoritma sistem merupakan gambaran umum dari proses kategori spam dan non spam, disini akan dilakukan proses pencocokan antara kata-kata penting yang mewakili kata kunci yang ada dalam tabel. Pembuatan tabel kata kunci dilakukan untuk membuat program kategori spam dan non spam ini dilakukan agar mendapatkan hasil yang baik. Dimulai dari start dengan cara mengumpulkan spam sample terus melakukan tokenizing lalu memasukan email uji, setelah itu lakukan klasifikasi email dengan menggunakan metode Naive Bayes. Kemudian bila spam besar dari ham maka hasilnya spam dan jika tidak hasilnya Ham, itu semua tertera di label memuat hasil dan selesai.



Gambar 3.1 Algoritma Sistem Klasifikasi Email

## 2.4 Perancangan Use Case Diagram

*Use case diagram* adalah gambaran umum serta keseluruhan sistem dimana pengguna yaitu *user*. *User* bisa mengolah data *email* dan melakukan relasi tambahan untuk menjalankan fungsinya, setelah itu *user* bisa membuat *token* email. Kemudian *user* bisa melakukan klasifikasi *email* dan bisa melihat hasil dari pemilahan *email* tersebut.

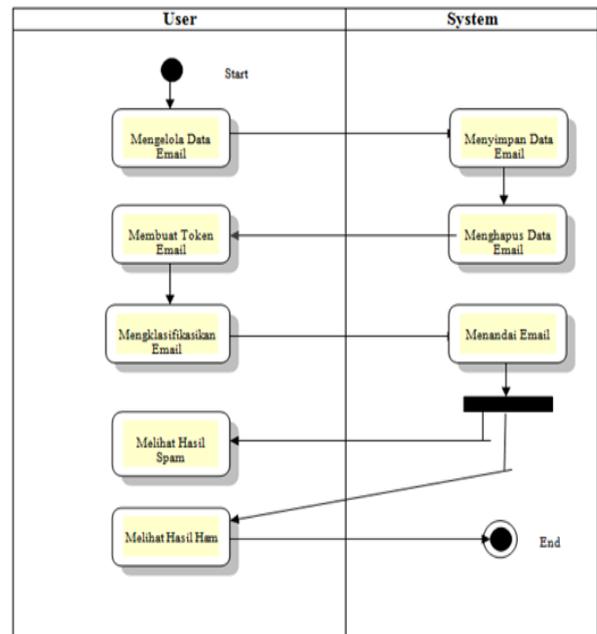


**Gambar 3.2 Use Case Diagram**

## 2.5 Perancangan Activity Diagram

*Activity diagram* menggambarkan sebuah keadaan dimulai (*start state*) yang menampilkan halaman menu dengan beberapa pilihan seperti gambar diatas. Dan sebagai pengguna bisa memilih tampilan menu kemudian melihat proses yang terjadi, setelah itu selesai.

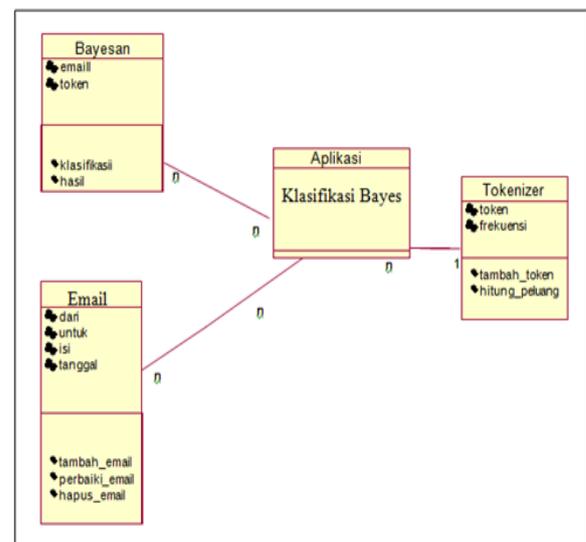
Pada *activity diagram* pertama *user* melakukan *start* dan masuk ke sistem dan melakukan penyimpanan data lalu sistem juga bisa menghapus data *email*. *User* bisa membuat *token email* dan bisa melakukan klasifikasi *email* terus menandai *email* dalam sistem, disitu *user* bisa melihat hasil *spam* dan melihat hasil *ham* setelah itu selesai.



**Gambar 3.3 Activity Diagram**

## 2.6 Perancangan Class Diagram

*Diagram* ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kalaborasi-kalaborasi. *Class Diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan dan asosiasi. *Class diagram* ini akan menunjukkan objek-objek yang terdapat pada aplikasi ini.



**Gambar 3.4 Class Diagram**

## 2.7 Perancangan Tabel

Data yang digunakan dalam pembangunan perangkat lunak ini disimpan dalam sebuah basis data *MySQL* yang terdiri dari 2 tabel yaitu tabel yang berfungsi untuk menyimpan data contoh *spam* Struktur tabel yang terdapat dalam basis data aplikasi *spam filtering* adalah sebagai berikut :

### 1. Tabel Sampel Surel

- a. Nama Tabel : Sampel Surel
- b. Fungsi : simpan data *spam*
- c. *Primary Index* : *id*

**Tabel 3.1 Tabel Sampel Surel**

No	Nama Field	Jenis	Ukuran	Keterangan
1	Id	Int	11	Identitas Email Sampe
2	Pengirim	Varchar	100	Nama Pengirim
3	Penerima	Varchar	100	Nama Penerima
4	Isi Email	Text	-	Isi Email
5	Tipe	Varchar	1	Tipe Email

### 2. Tabel *knowledge\_base*

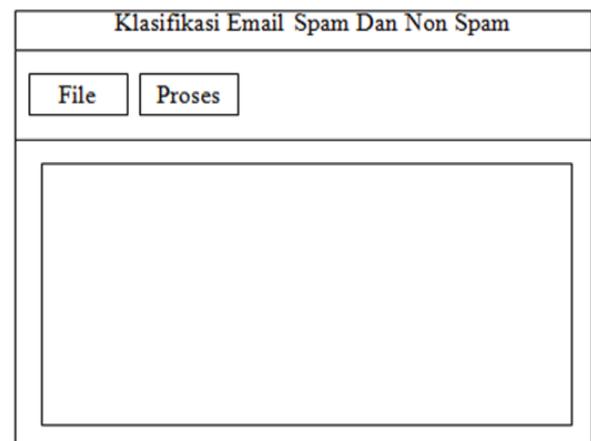
- a. Nama Tabel : *tokens*
- b. Fungsi : komentar spam
- c. *Primary Index* : *id*

**Tabel 3.2 Tabel *Tokens***

No	Nama Field	Jenis	Ukuran	Keterangan
1	Id	Int	11	Identitas Token
2	Token	Varchar	50	Token
3	Jumlah	Int	11	Jumlah Token
4	Tipe	Tinyint	3	Tipe Token

## 2.8 Antarmuka *Klasifikasi Email*

Pada *form* proses klasifikasi terdapat menu *file* dan proses, ini merupakan menu untuk melakukan klasifikasi surel *spam* dan *non spam* dengan algoritma *bayesian*. Pada menu *file* terdapat menu sampel surel dan keluar, sedangkan di menu proses terdapat menu *classifying email* dan *tokenizer*. Pada *form* proses klasifikasi terdapat menu *file* dan proses, ini merupakan menu untuk melakukan klasifikasi surel *spam* dan *non spam* dengan algoritma *bayesian*. Pada menu *file* terdapat menu sampel surel dan keluar, sedangkan di menu proses terdapat menu *classifying email* dan *tokenizer*.



**Gambar 3.5 Proses Klasifikasi Surel**

## 2.9 *Tokenizer*

Didalam *tokenizer* kita bisa menyimpan *token* dan *ham spam*, disini dalam proses *tokenizer* kita akan mencari isi *token spam* dan *ham*. Setelah proses selesai kita akan mengetahui jumlah *statistic token ham* dan *token spam*. Jadi secara singkat disini kita mengisi *ham* dan *spam*, lalu melakukan proses *tokenizing* dan menyimpannya di menu simpan.

**Gambar 3.6 Tokenizer**

### 2.10 Sampel Surel

Dalam penelitian ini kita koleksi sampel surel yang telah kita pindahkan ke *notepad* untuk di klasifikasikan, didalam *form* sampel surel ada beberapa menu utama, disini pertama kita bisa memilih *tipe file*, apakah *spam*, *ham* atau bukan. Setelah itu kita bisa menjalankan aplikasi tersebut dengan perintah yang kita inginkan yang telah terdapat didalam aplikasi tersebut, seperti: menu tambah, ubah tak terklasifikasi, ubah terklasifikasi dan hapus *email*. Menu-menu tersebut dapat kita jalankan sesuai dengan fungsinya masing-masing.

**Gambar 3.7 Sampel Surel**

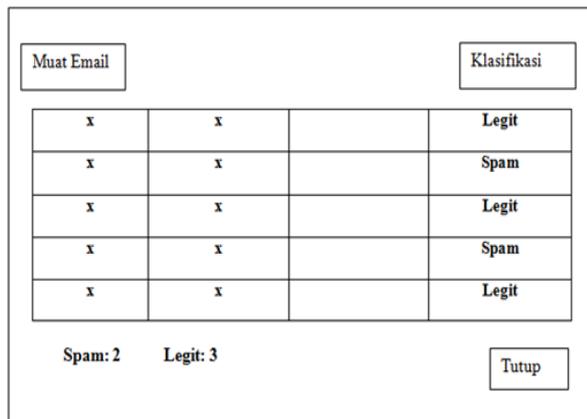
### 2.11 Form Tambah Sampel

Pada rancangan *form* tambah sampel terdapat beberapa menu di antaranya menu pengirim, menu penerima, isi, simpan dan *reset*. Menu – menu tersebut bisa digunakan oleh *user* untuk menambahkan sampel surel sesuai dengan fungsinya masing - masing. Pada menu pengirim di situ kita isikan alamat pengirim dan pada menu penerima kita isikan alamat penerima, setelah itu pada isi kita isikan contoh surel yang telah kita cari sampelnya, lalu kita simpan data tersebut.

**Gambar 3.8 Form Tambah Sampel**

### 2.12 Klasifikasi Antarmuka

Klasifikasi antarmuka ini adalah tampilan akhir dari aplikasi yang penulis buat, disini terdapat menu muat surel, klasifikasi dan tutup. Pada tampilan akhir ini kita bisa melihat proses hasil dari klasifikasi surel, di sini terdapat jumlah berapa surel tersebut yang termasuk ke dalam *spam* dan *non spam* setelah melakukan uji coba pada aplikasi *filter spam* tersebut.



**Gambar 3.9 Interface Klasifikasi**

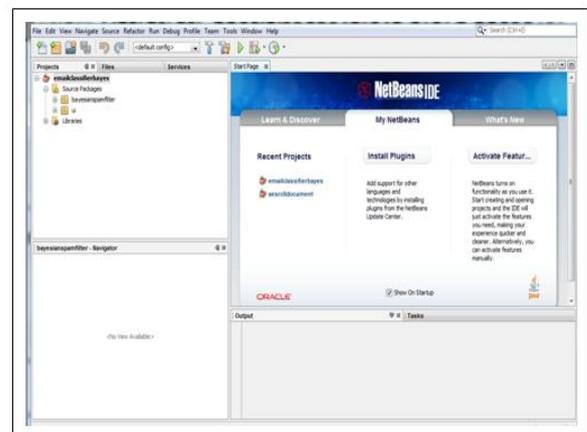
### 3. HASIL

Setelah melakukan analisis sistem dan perancangan sistem maka hasil yang dicapai oleh penulis adalah sebuah aplikasi *spam filtering* dengan menggunakan *metode Naive Bayes* yang dibuat dan di jalankan menggunakan bahasa pemrograman *Java Desktop* serta *Netbeans* sebagai *IDE* untuk menjalankannya. Aplikasi tersebut bermanfaat untuk mengklasifikasikan surel yang *spam* dan *non spam*.

Dengan adanya aplikasi *spam filtering* tersebut bisa mempercepat dan mempermudah kita dalam menentukan surel yang *spam* dan *non spam*. Aplikasi *spam filtering* ini telah dibuat dengan menggunakan bahasa pemrograman *Java Desktop* dan dijalankan dengan *Netbeans* sebagai *IDE*, untuk menjalankan aplikasi tersebut hal pertama yang harus dilakukan yaitu pastikan perangkat lunak (*software*) seperti: *Java*, *Netbeans IDE* dan *MySQL Front* sudah terpasang di komputer atau *laptop*. Disini penulis menggunakan *Windows 7* sebagai sistem operasi. Setelah semua perangkat lunak terpasang, jalankan aplikasi tersebut dengan cara memilih aplikasi *Netbeans IDE* dan *MySQL Front* yang

berada di *desktop*. Setelah data dari *MySQL Front* sudah tersambung maka kita bisa menjalankan aplikasi *spam filtering* ini dari *Netbeans IDE* dengan cara memilih *open project* dan cari basis data *MySQL Front* yang telah dibuat dengan nama *emailclassifierbayes*. Cara menjalankannya pilih pada *emailclassifierbayes* dan pilih *Run*. Tampilan dari aplikasi *Netbeans IDE* dan *MySQL Front* seperti gambar 4.1.

### 3.1 Tampilan Netbeans IDE



**Gambar 4.1 Tampilan Netbeans IDE**

Aplikasi program *Netbeans IDE* pada *spam filtering* ini berfungsi untuk menjalankannya, dengan catatan, *Netbeans IDE* bisa dijalankan bila basis data pada *MySQL Front* telah terkoneksi. Segala suatu yang dijalankan pada aplikasi ini yaitu berada di aplikasi *Netbeans IDE*. Pada aplikasi *Netbeans IDE* terdapat banyak menu antara lain, *file*, *edit*, *view*, *navigate*, *source*, *refactor*, *run*, *debug*, *profile* dan sebagainya. Dibawahnya juga terdapat menu-menu penting seperti: *new file*, *new project*, *open project*, *save all file* dan lain-lain.

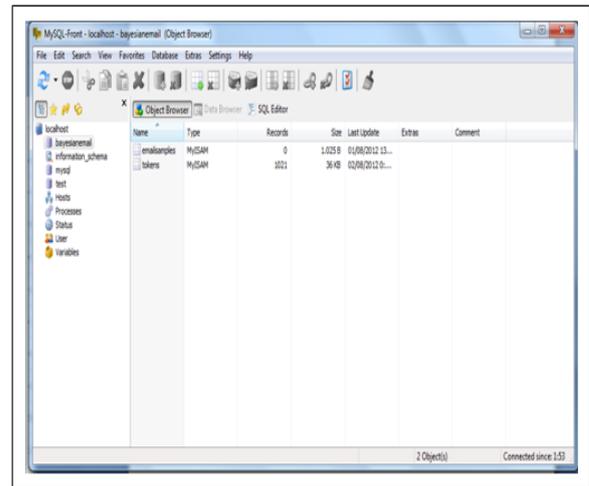
Untuk menjalankan aplikasi ini pilih *open project* dan cari *file name database MySQL* yang telah dibuat dengan nama *emailclassifierbayes*,

lalu klik *open project* maka akan muncul folder *emailclassfierbayes* yang berisi seperti: *source packages* (*Bayesian spam filter* dan *Ui*), serta terdapat juga folder *libraries*.

### 3.2 Tampilan Netbeans IDE Tampilan MySQL Front

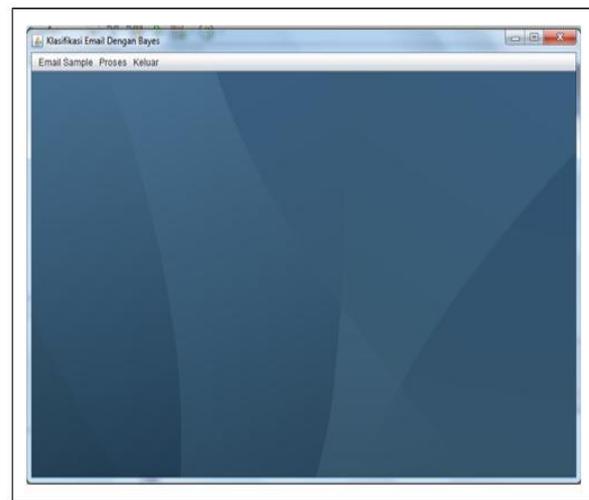
Pada aplikasi *MySQL Front* ketika memilihnya maka akan muncul seperti gambar diatas, *MySQL Front* ini berfungsi sebagai penyimpanan basis data untuk *Netbeans IDE*. Sebelum muncul seperti gambar diatas terlebih dahulu muncul aplikasi *open session*, *open session* ini merupakan sebuah nama aplikasi untuk membuka program. Disitu terdapat *session* nama dan *last login*, untuk membuka basis data pilih *open* pada *open sessions*. *MySQL Front* memiliki menu di antaranya: *file*, *edit*, *search*, *view*, *favorit*, *database* dan sebagainya.

Basis data dalam aplikasi ini ada yang namanya *localhost*, *localhost* ini merupakan tempat penyimpanan basis data *MySQL*, disitu terdapat beberapa menu *Bayesian email* yaitu sampel surel dan *tokens*. Sampel surel merupakan tempat penyimpanan basis data seperti rancangan table sampel yang terletak di menu *object browser* dan menu data *browser*, sedangkan *token* sama halnya sampel surel yang menyimpan data rancangan juga di *object browser* dan menu data *browser*.



Gambar 4.2 Tampilan MySQL Front

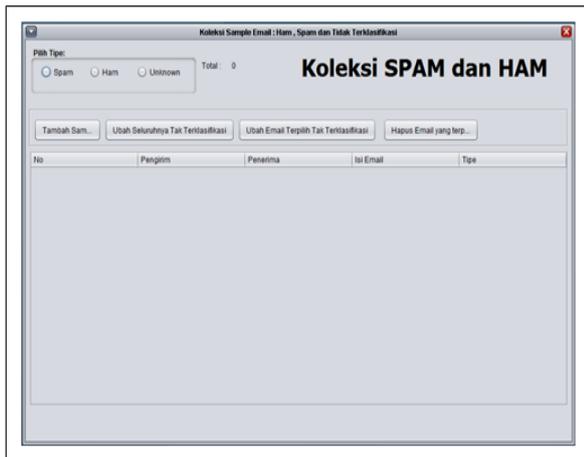
### 3.3 Tampilan Halaman Utama



Gambar 4.3 Tampilan Halaman Utama

Pada tampilan halaman utama aplikasi *spam filtering* berbasis *java desktop* untuk mengklasifikasikan surel ini dijalankan dengan *Netbeans IDE* untuk menampilkan halaman utama. Halaman utama atau *home* dijalankan dengan cara menekan pada aplikasi *Netbeans IDE* dan pilih kanan pada *emailclassfierbayes* lalu pilih *Run*. Tampilan halaman utama pada aplikasi ini seperti gambar diatas, di halaman utama terdapat beberapa menu diantaranya: sampel surel, proses, dan keluar. Menu-menu tersebut memiliki fungsi-fungsi tersendiri dan akan kita bahas satu persatu.

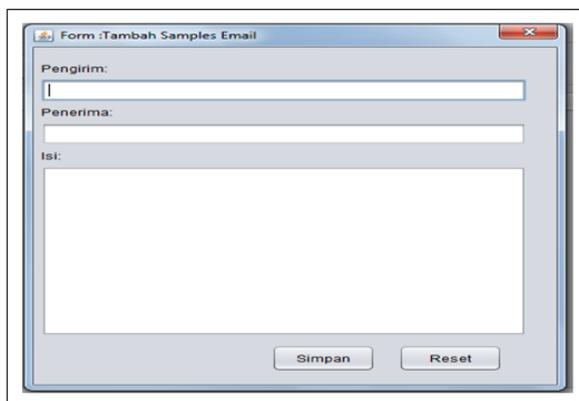
### 3.4 Tampilan Sampel Surel



**Gambar 4.4 Tampilan Sampel Surel**

Tampilan sampel surel terletak di halaman utama untuk menjalankannya buka halaman utama seperti yang telah dijelaskan tadi lalu pilih atau pilih sampel surel. Pada tampilan sampel surel terdapat menu pilih tipe, kita bisa memilih salah satunya dengan cara memilih salah satunya dengan pilihan seperti: *spam*, *ham* dan *unknow*, dan terdapat juga menu tambah sampel, menu ubah seluruhnya tak terklasifikasi, menu ubah surel terpilih tak terklasifikasi, dan menu hapus surel yang terpilih. Di bawah menu itu juga terdapat lima menu lainnya yaitu: menu nomor, pengirim, penerima, isi surel dan menu tipe. Semua dari menu yang telah diuraikan tadi mempunyai fungsi dan manfaat tersendiri.

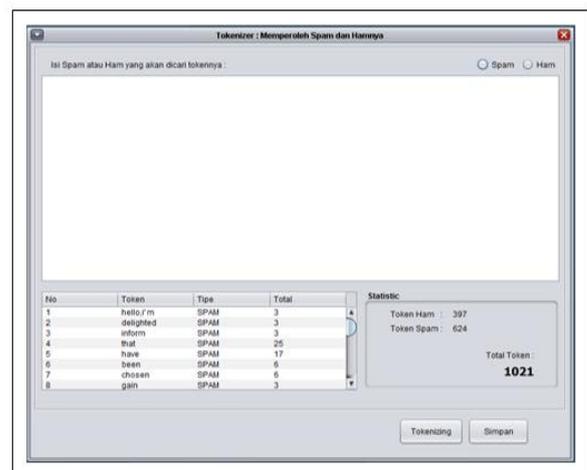
### 3.5 Tampilan Tambah Sampel Surel



**Gambar 4.5 Tampilan Tambah Sampel Surel**

Cara menjalankan tampilan tambah sampel surel seperti yang dijelaskan sebelumnya yaitu dengan cara buka *Netbeans IDE* buka halaman utama dan pilih sampel surel. Pada tampilan sampel surel itu terdapat tampilan tambah sampel surel, kita bisa memilih tambah sampel surel dan akan tampil seperti gambar di bawah ini. Pada tampilan *form* tambah sampel surel terdapat beberapa menu utama diantaranya: menu pengirim, menu penerima, dan menu isi. Disamping itu juga terdapat menu simpan dan menu *reset*, cara menjalankan aplikasi ini kita bisa mengisi *form* kosong pada menu pengirim, menu penerima dan menu isi. Kemudian kita juga bisa menyimpan dan meresetnya.

### 3.6 Tampilan Tokenizer



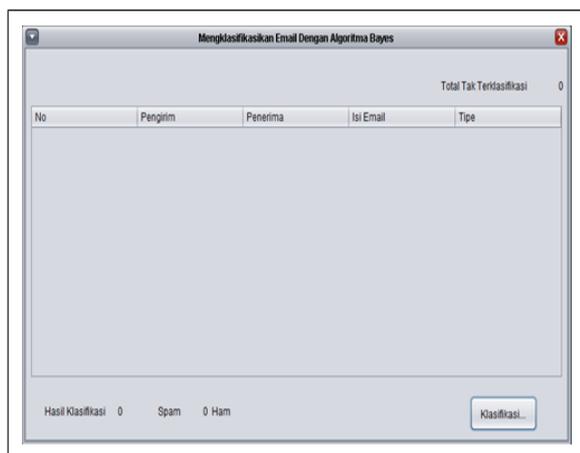
**Gambar 4.6 Tampilan Tokenizer**

Pada menu tampilan *tokenizer* kita bisa menjalankannya dengan cara membuka halaman utama dan memilih menu proses, mana menu proses terdapat dua menu pilihan untuk membuka tampilan *tokenizer* kita bisa memilih *tokenizer* atau dengan cara menekan tombol *Ctrl+T* pada *keyboard laptop* atau komputernya. Setelah selesai proses itu maka akan muncul tampilan seperti gambar di bawah ini, pada tampilan *tokenizer* terdapat *form* kosong untuk

mengisi *spam* atau *ham* yang akan dicari *token* nya, tapi sebelumnya pilih dulu menu yang akan dicari *token* nya. Apakah *spam* or *ham*, kita bisa memilihnya dengan cara mengklik *spam* or *ham*.

Cara untuk mencari *token* nya yaitu salin lalu tempel isi surel *spam* pada *form* kosong isi *spam* atau *ham*, lalu pilih menu *tokenizing* pada aplikasi dan akan muncul jumlah *statistic token ham* dan *token spam*. Kemudian kita juga bisa melihat hasil total *token* seperti gambar, untuk menyimpannya ke basis data *MySQL* kita bisa pilih *simpan* dan akan muncul tulisan data di perbaharui, maka data tersebut sudah tersimpan.

### 3.7 Tampilan Klasifikasi



**Gambar 4.7 Tampilan Klasifikasi**

Tampilan klasifikasi merupakan tampilan akhir atau hasil akhir dari proses mengklasifikasikan *email spam* dan *non spam*. Cara menjalankannya sama seperti cara menjalankan tampilan proses *tokenizer* cuma kalau tampilan klasifikasi cara membukanya bisa menggunakan *Ctrl+C*. Tampilan klasifikasi seperti gambar 4.7, disitu terdapat beberapa menu untuk mengklasifikasikan surel seperti: menu nomor, menu pengirim, menu penerima, menu isi surel, dan menu tipe. Untuk melihat hasil klasifikasi termasuk *spam* atau *ham* kita

bisa memilih menu klasifikasi maka akan muncul hasil klasifikasi berapa jumlah *spam* dan *ham*.

### 3.8 Tampilan Konfirmasi Keluar



**Gambar 4.8 Tampilan Konfirmasi Keluar**

Tampilan menu konfirmasi keluar merupakan perintah untuk menutup aplikasi *spam filtering* berbasis *java desktop* untuk mengklasifikasikan surel, menu ini terdapat di halaman utama, jika kita ingin menutup aplikasi ini kita bisa mengkliknya dan akan muncul menu seperti gambar diatas. Setelah selesai mengkliknya kita bisa memilih *yes* atau *no*. pilih *yes* untuk menutup aplikasi dan klik *no* untuk membatalkannya.

## 4. SIMPULAN

Dari hasil penelitian dan analisis yang dilakukan terhadap metode *Naive Bayes* pada aplikasi *spam filtering* berbasis *java desktop* maka dapat di ambil kesimpulan:

1. Klasifikasi surel *spam* dengan metode *Naive Bayes* yang telah dilakukan mempunyai tingkat kesalahan yang kecil.
2. *Bayesian spam filtering* adalah aplikasi dari *klasifikasi Naive Bayes* dalam hal *email filtering* yang berhubungan dengan peluang yang didapat dari beberapa percobaan.

3. *Spam filtering* merupakan salah satu aplikasi dari *teori probabilitas* dan *statistik* dalam bidang Sistem dan Teknologi Informasi yaitu *email filtering*.
4. Pembentukan basis data *spam* dan *non spam* menjadi suatu keharusan agar *filter Bayesan* selalu waspada dalam mengenali suatu *spam*. Di satu sisi kekuatan *anti-spam filter Bayesan* nampaknya cukup akurat dan signifikan dalam mendeteksi dan membunuh *spam*.
5. Pendeteksian suatu *email* masuk dalam kategori *spam* atau *non spam* membutuhkan suatu pelatihan bagi *filter Bayesan* agar terbiasa dengan kata-kata yang terkandung dalam *email* semakin baik proses pelatihan maka akan semakin pintar pula *filter Bayesan* dalam menyaring suatu *spam*. Yang menjadi kunci disini yaitu, memblok suatu *email* sebagai *spam* lebih aman dibanding meloloskannya sebagai *ham*, dalam rangka menjaga keamanan jaringan komputer pengguna atau institusi.
6. Tingkat keberhasilan dari program identifikasi *spam* ini dipengaruhi berdasarkan banyaknya jenis dan variasi kata pada kedua basis data, serta jumlah kata yang sama pada kedua basis data.

Christopher, L. 2011. Mengapa *Email Marketing* Anda dianggap *SPAM*, Modula:Bandung.

Graham, P., *BETTER BAYESIAN FILTERING*, January 2003 (terakses 25 November 2004).

[http://www.webopedia.com/TERM/D/What is data mining](http://www.webopedia.com/TERM/D/What_is_data_mining), tanggal 28 Agustus 2008.

<http://www.tnial.mil.id> (Cakrawala 2007 E-mail, Kendala dan Permasalahannya [17 Juli 2009]).

[http://www.emailaddressmanager.com/marketing\\_tips.html](http://www.emailaddressmanager.com/marketing_tips.html)

Kadir, A. 2005. *Dasar Pemograman Java 2*. Andi : Yogyakarta.

Kristanto, A. 2004. *Rekayasa Perangkat Lunak*. Gava Media : Yogyakarta.

Nugroho, A. 2010. *Rekayasa Perangkat Lunak Berorientasi Objek dengan USDP (Unified Software Development Process)*. Andi.Yogyakarta.

Santoso, B. 2007. *Data Mining: Terapan dengan MATLAB*. Graha Ilmu: Jakarta.

Supriyanto, A. 2005. *Pengantar Teknologi Informasi*, PT. Salemba Infotek, Jakarta

Sucahyo, YG. *Data mining – Menggali Informasi Yang terpendam*, tanggal 7 Juni 2008.

## DAFTAR RUJUKAN

- Rosa, AS, dan Shalahudin, M. 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (terstruktur dan Berorientasi Objek). Modula:Bandung.