

# PENERAPAN METODE NAÏVE BAYES UNTUK SISTEM KLASIFIKASI SMS PADA SMARTPHONE ANDROID

**Ebranda A.W** (EbrandaAW@gmail.com), Novi Triana(Novi\_triana13@yahoo.com)

**Mardiani, S.Si, M.T.I.** (Mardiani@stmik-mdp.net)

**Tinaliah, S.Kom.** (Tinaliah@stmik-mdp.net)

Jurusan Teknik Informatika

**STMIK MDP**

**Abstrak:** SMS merupakan media komunikasi yang paling sering digunakan walaupun sekarang Ponsel telah memiliki berbagai media komunikasi aplikasi *Messenger*. Maraknya aksi penipuan sekarang ini membuat masyarakat semakin tidak nyaman dengan menggunakan SMS. Maka aplikasi SMS *spam* ini yang *fitur* nya dapat menggolongkan atau mengklasifikasikan suatu SMS mana yang termasuk SMS *spam* dan bukan *spam* secara otomatis. Dengan pemilihan metode *Naïve Bayes* untuk melakukan proses klasifikasi diharapkan dapat mengklasifikasikan SMS dengan tingkat keakuratan yang tinggi sehingga *spam* SMS dapat teridentifikasi dengan baik. Metodologi pengembangan sistem menggunakan *Microsoft Solutions Framework*(MSF). Aplikasi ini dibuat dengan menggunakan sistem aplikasi dari *eclipse*. Dengan adanya aplikasi ini dapat memberikan kenyamanan pada pengguna dalam menggunakan SMS tanpa adanya gangguan dari SMS *spam*. Penerapan *Naïve Bayes* dapat mengklasifikasikan *spam* dengan baik dari data latih 80 SMS sehingga memperoleh tingkat akurasi 85,11%.

**Kata kunci:** SMS *Spam*, *Android*, *Naïve Bayes*, dan *eclipse*.

## 1 PENDAHULUAN

Short Message Service atau SMS merupakan media komunikasi yang paling sering digunakan. SMS telah menjadi bagian dari aktivitas keseharian masyarakat, walaupun sekarang berbagai Ponsel telah memiliki media komunikasi aplikasi *Messenger* seperti *Whatsapp Messenger*, *Blackberry Messenger*, *Line Messenger* dan banyak lagi lainnya.

Penggunaan SMS yang semakin luas ini dimanfaatkan oleh sebagian orang yang memiliki kepintaran untuk melakukan tindak kejahatan, seperti penipuan, ancaman, dan lain sebagainya melalui SMS. Hanya dengan bermodal ponsel dan pulsa, pelaku dapat melakukan aksinya dengan mudah dan merasa lebih aman karena akan sulit dilacak. Hal ini sulit dilacak, Karena setelah melakukan aksi penipu hanya tinggal membuang atau mematahkan SIM card yang dipakai. Maraknya aksi penipuan sekarang ini

membuat masyarakat semakin tidak nyaman dengan menggunakan SMS.

## 2 LANDASAN TEORI

### 2.1 SMS *Spam*

SMS *Spam* adalah Sebuah pesan teks (SMS) yang tidak diminta atau tidak diinginkan oleh pengguna yang dikirim ke perangkat seluler, yang biasanya mengandung materi pemasaran (promosi), penipuan, dan lain-lain, seperti email *spam*. Orang yang melakukan spam disebut spammer. Tindakan spam dikenal dengan nama *spamming*.

*Spam* dikirimkan melalui pesan teks dengan biaya operasional yang sangat rendah untuk mencapai para pelanggan-pelanggan yang diinginkan. Karena hambatan masuk yang rendah, maka banyak *spammers* yang muncul dan jumlah pesan yang tidak diminta menjadi sangat tinggi. Akibatnya, banyak

pihak yang dirugikan. (Karimul Makhtidi 2012).

## 2.2 Java

Java dikenal sebagai pemrograman tingkat yang berorientasi objek, atau lazim disebut dengan istilah *Object-Oriented Programming* (OOP).

Java disukai semua segmen karena mencakup seluruh aspek dari pemrograman desktop hingga pemrograman *mobile* (*Mobile Programming*) dan java juga terkenal *portable*, karena seluruh sistem operasi dapat menjalankan java, sistem operasi itu antara lain *window*, *linux*, *dos*, *unix* dan lain-lain (Yuniar 2005, h. 2).

## 2.3 Android

Android adalah sebuah open source software toolkit untuk perangkat bergerak yang dibuat oleh Google yang mencakup sistem operasi, *middleware* dan *key applications*. Android SDK menyediakan *tool* dan API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

Platform Android pertama kali dikembangkan oleh perusahaan Android Inc yang merupakan sebuah perusahaan baru yang bergerak di bidang perangkat lunak untuk ponsel. Pada tahun 2005 Android Inc dibeli oleh Google. Pada saat perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* mendukung pengembangan standar terbuka pada perangkat selular.

Hingga saat ini, terdapat beberapa versi dari sistem operasi Android, antara lain:

- a) Android Versi 1.1
- b) Android Versi 1.5 (*Cupcake*)
- c) Android Versi 1.6 (*Donut*)
- d) Android Versi 2.1 (*Eclair*)
- e) Android Versi 2.2 (*Froyo: Frozen Yoghurt*)
- f) Android Versi 2.3 (*GingerBread*)
- g) Android Versi 3.0 (*HoneyComb*)
- h) Android Versi 4.0 (*Ice Cream Sandwich*)
- i) Android Versi 4.1 (*Jelly Bean*)

## 2.4 Pemrograman Android

### 2.3.1 JDK (*Java Development Kit*)

JDK adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman Java. JDK berjalan diatas sebuah *virtual machine* yang dinamakan JVM (*Java Virtual Machine*). Dokumentasi JDK beris gembang, referensi halaman untuk i spesifikasi API, deskripsi fitur, panduan pen perkakas JDK dan utilitas, demo, dan link ke informasi terkait.

### 2.3.2 Eclipse

*Eclipse* adalah IDE (*Integrated Development Environment*) merupakan aplikasi pengembangan *Java/Android*, *Eclipse* memiliki plugins yang dapat membuat *project* yang berbasis *Android*. ADT (*Android Development Tools*) adalah *plugin* yang menghubungkan antara IDE *Eclipse* dan *Android SDK*, (Safaat H 2011, h. 16).

### 2.3.3 SQLite

SQLite adalah *database* yang sangat ringan dan bisa ditangani oleh android tanpa menggunakan *tool* lain. Database SQLite di android bisa langsung dipakai tanpa memerlukan *setup database* atau administrasi.

### 2.3.4 Naïve Bayes

*Naïve Bayes* adalah metode *Bayesian Learning* yang paling cepat dan sederhana. Hal ini berasal dari *teorema Bayes* dan hipotesis kebebasan, menghasilkan *klasifier* statistik berdasarkan peluang. Ini adalah teknik sederhana, dan harus digunakan sebelum mencoba metode yang lebih kompleks. *Naïve Bayes* dapat dirumuskan sebagai berikut :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \dots\dots\dots(1)$$

Pada proses pengklasifikasian SMS, maka pendekatan Bayes akan menyeleksi kategori teks yang memiliki probabilitas yang paling tinggi, nilai  $P(B)$  dapat di abaikan

karena nilainya adalah konstan untuk semua nilai  $P(A)$  dan karena atributnya tidak saling terkait maka didapatkan  $P(B/A)*P(A)$  maksimum yang didefinisikan sebagai berikut :

$$P(A|B) = \text{argmax } P(B|A)*P(A) \dots\dots(2)$$

Dengan perhitungan diatas kita dapat membandingkan nilai probabilitas mana yang paling tinggi.

### 2.3.5 Klasifikasi SMS

Cara untuk mengukur kinerja dari suatu *klasifier* teks secara efektif terhadap suatu term yaitu dengan mengukur *recall* (r) dan *precision* (p). jika jumlah keputusan yang *true positive*, *false positive*, dan *false negative* dianggap sebagai tp, fp, dan fn, dengan demikian *recall* (r) dan *precision* (p) dapat didefinisikan sebagai berikut :

$$p = \frac{tp}{tp+fp}, r = \frac{tp}{tp+fn} \dots\dots\dots(3)$$

*True positive* dapat diartikan sebagai pesan yang sah yang dianggap sebagai pesan yang sah, *false positive* adalah pesan yang sah dianggap sebagai pesan *spam*, dan *false negative* merupakan pesan spam.

Setelah kita mendapatkan definisi dari setiap precision dan recall maka kita dapat menghitung akurasi dengan definisi sebagai berikut :

$$\text{Akurasi} = \frac{tp + tp}{tp + tp + fp + fn} \dots\dots\dots(4)$$

### 2.3.6 Android SDK

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Android SDK menyediakan *tools* dan *api* untuk pengembangan *platform* aplikasi android dengan menggunakan bahasa pemrograman java (Mulyadi, 2010).

## 2.4 Metodologi MSF

*Microsoft Solutions Framework* (MSF) adalah “serangkaian prinsip, model, disiplin, konsep, dan tuntunan untuk membuat solusi IT” yang diciptakan oleh Microsoft. Bisa dibilang MSF adalah sebuah *Software Development Life-Cycle (SDLC)*. Namun MSF tidak hanya dapat diterapkan pada pembuatan aplikasi, selain itu dapat diterapkan pada proyek-proyek IT lain seperti pembuatan infrastruktur dan jaringan. Ini karena MSF tidak hanya mencakup perekayasaan perangkat lunak namun hampir mencakup pembuatan sebuah proyek IT secara keseluruhan. Karena itulah MSF tidak disebut sebagai sebuah *Metodologi* (seperti Metodologi Waterfall, Metodologi Spiral, dll) namun diberi nama *Framework*. Dalam *process model* ini, memiliki lima fase yaitu :

### 2.4.1 Envisioning Phase

Fase dimana visi, misi, tujuan dan batasan dari proyek ditetapkan secara formal dalam sebuah *Vision/Scope Document*. *Vision/Scope Document* ini bisa dibilang dokumen yang paling penting dalam keseluruhan dokumen proyek.

### 2.4.2 Planning Phase

Melakukan analisis sistem dan membuat rancangan awal tentang penerapan metode *Naïve Bayes* untuk sistem klasifikasi SMS pada *Smartphone* Android. Selain itu dalam tahap ini sistem akan dirancang pula bagian interaksi manusianya sehingga dapat mudah dalam penggunaannya (*user friendly*) serta, penulis merancang analisis kebutuhan dan membuat sistem dengan cara menggambarkan melalui model *use case*.

### 2.4.3 Developing Phase

Fase ini mulai merancang sistem secara utuh, merancang desain SMS pada *smartphone* Android dengan menggunakan *Eclipse* juno, melakukan pengkodean program dengan menggunakan SDK pada *platform* android dengan menggunakan bahasa pemrograman Java, dan melakukan pengkodean program desain aplikasi pada

smartphone Android dengan menggunakan Eclipse(IDE).

#### 2.4.4 Stabilizing Phase

Fase dimana produk diuji coba fungsinya dalam berbagai skenario, untuk menemukan dan mengatasi *error*, *bug*, ataupun faktor-faktor lain yang dapat mempengaruhi fungsi dan *performance* dari produk.

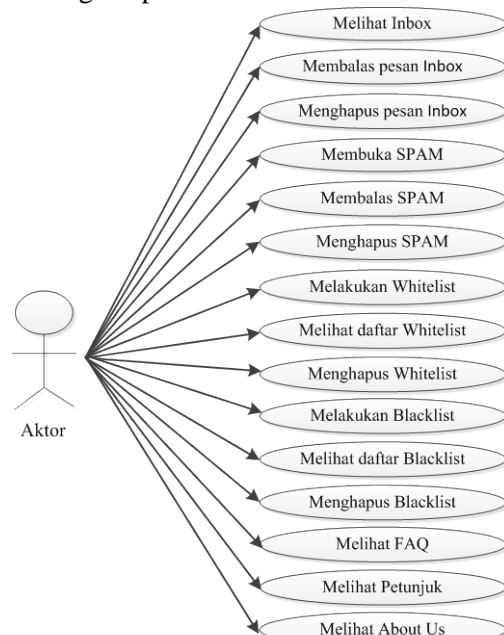
#### 2.4.5 Deploying Phase

Fase dimana produk dikirim, dipasang dan digunakan oleh *customer*, untuk kemudian dilakukan *review* atas segala yang telah dilakukan selama pengerjaannya, serta kemungkinan langkah pengembangan selanjutnya (Microsoft Corporation, 2003).

### 3 PERANCANGAN

#### 3.1 Diagram Use Case

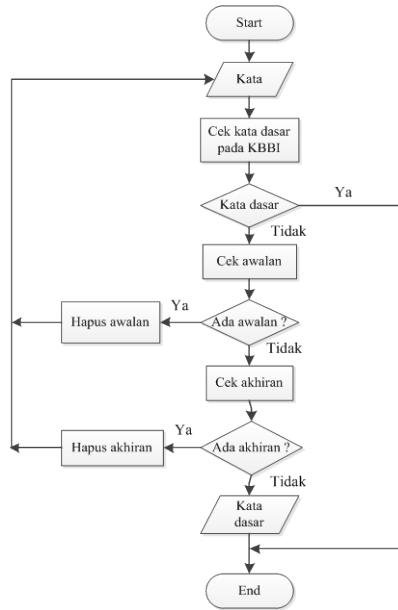
Diagram *use case* merupakan gambaran dari interaksi antara komponen-komponen suatu sistem yang akan dibangun. Berikut ini adalah desain *use case diagram* perancangan aplikasi :



Gambar 1: Diagram Use Case Aplikasi

Kelas yang dipakai dalam pengklasifikasian SMS *Spam* ini ada dua macam, yaitu kelas *spam* dan kelas *nonspam*. Tahapan dalam klasifikasi terdiri dari proses pengolahan data latih dan proses pengolahan data uji. Data latih yang digunakan sebanyak 80 dokumen SMS dengan SMS *spam* sebanyak 40 dokumen dan SMS *nonspam* sebanyak 40 dokumen. Seluruh dokumen SMS menggunakan Bahasa Indonesia. Berikut ini adalah tahapan klasifikasi SMS *Spam* :

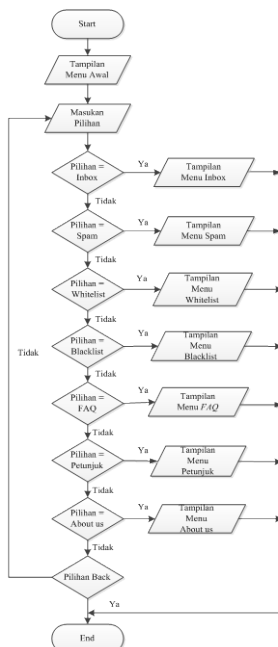
1. Tahap praproses data diawali dengan mengubah dokumen SMS (korpus) yang telah dikumpulkan ke dalam bentuk dokumen.
2. Tahap Tokenisasi Seluruh korpus (kata dasar yang telah ada akan dicek di kamus yang telah ada) masuk ke dalam proses tokenisasi untuk diambil token atau kata-kata yaitu kata yang dibatasi oleh *whitespace* dan semua karakter kecuali huruf dan angka, seperti: , . ? ! / ' " ; & ( ) + - \*.
3. Tahap Hapus *Stopwords* Penghapusan *stopwords* dilakukan pada kata yang berfungsi sebagai kata depan dan singkatan dari kata depan itu sendiri. *Stopwords* yang digunakan pada sistem ini dibatasi untuk kata-kata: di/d, ke/k, dari/dr, pada/pd, dan/n, atau/or, yang/yg, dengan/dgn, ini/ni/ne, itu/tu, juga/jg, untuk/utk/u, akan/kan/kn dan tetapi/tapi/tp.
4. Tahap *Stemming* dilakukan penghapusan prefiks dan sufiks pada kata yang mengandung imbuhan untuk mendapatkan kata dasar. Penentuan kata dasar juga mengacu kepada kamus kata dasar yang diperoleh dari KBBI (Kamus Besar Bahasa Indonesia) yang jumlah keseluruhan-nya 28.526 kata. Untuk mengetahui suatu kata terdapat pada KBBI, digunakan metode *string matching*, karena hasil tokenisasi dan kata dasar dalam KBBI sama-sama dalam bentuk *lowercase*. Berikut *Flowchart* dari algoritme *stemming*, seperti yang terlihat pada Gambar.



**Gambar 2: Flowchart Diagram Stemming**

### 3.1.1 Flowchart Diagram Aplikasi

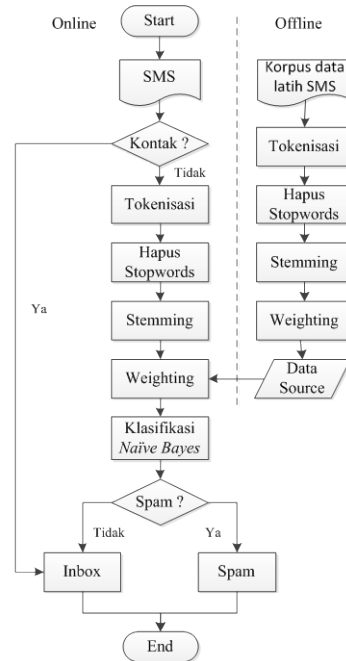
Membuat rancangan prosedur awal dalam proses menjalankan aplikasi dalam bentuk diagram alir (*flowchart*). Berikut ini merupakan *flowchart* yang menggambarkan alir prosedur untuk menjalankan aplikasi.



**Gambar 3: Flowchart Diagram Menu Awal**

### 3.1.2 Flowchart Diagram Spam

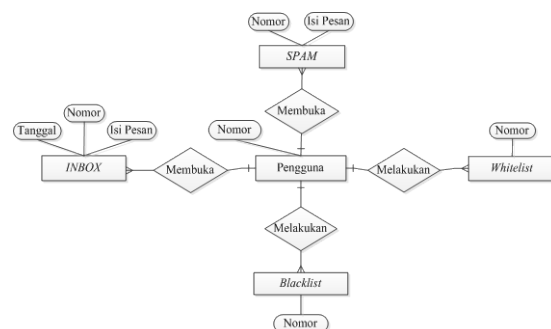
Sistem SMS Spam detector ini terbagi menjadi dua alur, yaitu alur *offline* dan alur *online*. Alur *offline* adalah tahapan proses klasifikasi yang dilakukan diluar perangkat *mobile* Android sedangkan alur *online* itu dilakukan di perangkat *mobile* atau emulator Android.



**Gambar 4: Flowchart Diagram Spam**

### 3.2 ER-Diagram

ER-Diagram pada sistem yang diusulkan untuk bertujuan memperjelas hubungan di antara tempat penyimpanan, seperti yang ditunjukkan pada gambar ER-Diagram dibawah ini :

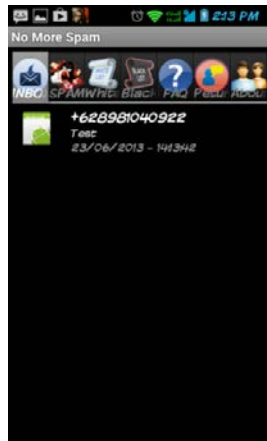


**Gambar 5: ER-Diagram Aplikasi**

## 4 RANCANGAN ANTARMUKA

### 4.1 Tampilan Menu Utama

Pada Menu Utama ini, pengguna dapat mengakses beberapa menu pilihan yang terdapat pada menu utama seperti yang terlihat pada gambar,

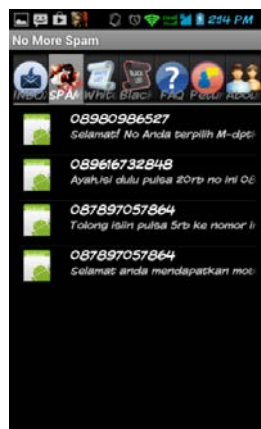


**Gambar 6: Menu Utama**

Dari menu utama ini ada beberapa tampilan menu pilihan yang dapat pilih oleh pengguna. Diantaranya Inbox, Spam, Whitelist, Blacklist, FAQ, Petunjuk, dan About us.

### 4.2 Tampilan Menu Spam

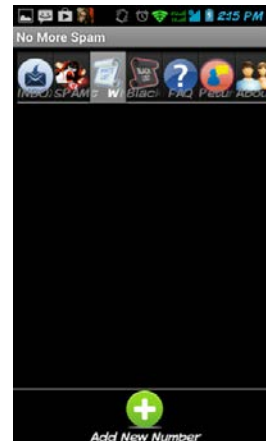
Pada tampilan Spam dibawah ini menampilkan SMS Spam yang sudah diklasifikasi dengan baik seperti yang terlihat pada gambar.



**Gambar 7: Menu Spam**

### 4.3 Tampilan Menu Whitelist

Pada Menu Whitelist pengguna dapat menambahkan nomor yang diinginkan seperti yang terlihat pada gambar,

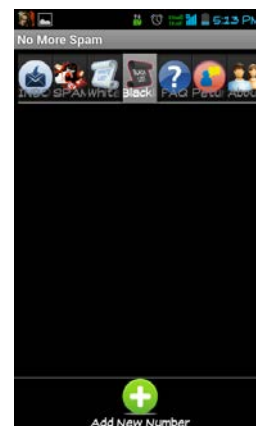


**Gambar 8: Menu Whitelist**

Pada button pada gambar menu whitelist digunakan untuk memasukkan nomor yang diizinkan oleh pengguna.

### 4.4 Tampilan Menu Blacklist

Pada Menu Blacklist pengguna dapat menambahkan nomor yang tidak diinginkan seperti yang terlihat pada gambar,



**Gambar 9: Menu Blacklist**

Pada button gambar menu blacklist digunakan untuk memasukkan/menambahkan nomor yang tidak diizinkan oleh pengguna jadi setiap SMS yang masuk dari nomor

tersebut akan masuk ke aplikasi ini meskipun SMS nya bukan SMS *Spam*.

memasukkan nomor dari sms atau spam ke blacklist atau whitelist.

## 5 PENUTUP

### 5.1 Kesimpulan

- a) Telah dihasilkan sistem deteksi SMS *spam* pada *smartphone* Android untuk Bahasa Indonesia dengan baik dan terbukti dari data latih yang digunakan dan cara menentukan SMS *Spam* dan bukan *Spam* ada di bab 3.
- b) Setelah melakukan pengujian dengan 80 SMS, 40 SMS Spam dan 40 SMS bukan Spam telah dihasilkan precision 86,96% dan recall 83,33% yang berarti nilai akurasi juga 85,11%.
- c) Kinerja klasifikasi pada sistem cukup bagus karena aplikasi dapat mengklasifikasikan SMS masuk dengan waktu yang relatif singkat.
- d) Dari Hasil kuisioner yang telah didapatkan bahwa pengguna dapat menggunakan aplikasi *Spam* ini secara mudah karena fitur petunjuk penggunaan sudah sangat jelas.

### 5.2 Saran

- a) Aplikasi ini dapat di terapkan pada *platform mobile* yang lain seperti *blackberry*, *windows phone*, dan IOS agar lebih banyak pengguna yang dapat menggunakan model aplikasi ini sehingga banyak yang dapat mengembangkan aplikasi ini.
- b) Menambahkan menu penyaringan SMS masuk berdasarkan kata kunci yang telah ditentukan
- c) Memperbarui *user interface* agar pengguna lebih tertarik menggunakan aplikasi ini serta menambahkan fitur-fitur yang dapat memperbaiki kinerja aplikasi ini. Seperti : memindahkan nomor yang di blacklist ke whitelist atau sebaliknya, serta

## DAFTAR PUSTAKA

- [1] Ahonen, T., Moore., A., 2010, “*Sms Text Message*”, Diambil 7 April 2013, dari <http://edition.cnn.com>.
- [2] A.S., Rosa., Shalahudin M. 2011, *Modul Pembelajaran Rekayasa Perangkat Lunak.*, Modula, Bandung.
- [3] Berry MW., Kogan J. 2010, *Text Mining: Applications and Theory*. Chichester: John Wiley & Sons.
- [4] Burnette, E. 2008, *Hello Android, Introducing Google's Mobile Development Platform*. Dallas: Pragmatic Bookshelf. Hal 30-34.
- [5] Darujati C., Gumelar B.A. 2012, *Pemanfaatan Teknik Supervised untuk Klasifikasi Teks Berbahasa Indonesia*, Fasilkom Universitas Narotama Surabaya.
- [6] Dyah Diwasasri, Ratnaningtyas. 2011, *Aplikasi Teorema Bayes dalam Penyaringan Email*, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.
- [7] Johanes, Andria. 2007, *Penerapan teknik Email Filtering Berbasis Ciri dan Jaringan Syaraf Tiruan Propagasi Balik*, Fasilkom UI.
- [8] Kevin, C., 2005, *Nearst-neighbor Searching and Met-ric Space Dimensions*. Nearest-Neighbor Methods for Learning and vision : Theory and Paractice.

- [9] Kusrini, 2009, *Algoritma Data Mining*, Andi Offset, Yogyakarta. Diambil 7 April 2013, dari <http://www.docstoc.com>.
- [10] Microsoft Corporation 2003, "*Microsoft Solution Framework White Paper*", Microsoft, United States. [21] Zhang, H, "*The Optimality of Naïve Bayes*", Diambil 26 Februari 2013, dari <http://www.cs.unb.ca>
- [11] Narbuko, C., 2003, *Metodologi Penelitian*, Bumi Aksara, Jakarta.
- [12] Nugroho, P., 2012, *Klasifikasi Email Spam Metode Naïve Bayes Menggunakan Java Programming*.
- [13] Palupiningsih, P., 2011. *Technical Paper On Data Mining*. Program S2 Ilmu Komputer Institut Pertanian Bogor.
- [14] Powers, Marlys Keeton, 2004, *MSF a pocket Guide*, Diambil 7 Maret 2013, dari <http://www.itiitrainers.com>.
- [15] Rahmawati, Risna, 2012, *Analisis Peringkat Teks Berdasarkan Query Menggunakan Metode Enumerasi*, Diambil 3 Mei 2013, dari <http://digilib.ittelkom.ac.id>
- [16] Rish, I., 2001, "*An Empirical Study of The Naive Bayes Classifier*", IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.
- [17] Suryabrata, S., 2008, *Metodologi Penelitian*, Rajawali Pers, Jakarta.
- [18] Sutedjo, B., 2004, *Algoritma & Teknik Pemrograman*, Andi Offset, Yogyakarta.
- [19] Wibowo, Ari, 2011, *Pengujian Kerelevanan Sistem Temu Kembali Informasi*, Diambil 3 Mei 2013, dari <http://p2m.polibatam.ac.id>
- [20] Wijaya H., Bobby, Ajie, H., Hafid M., 2011, *Microsoft Solution Framework*,