PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK SPAM FILTERING MENGGUNAKAN METODE BAYESIAN

TUGAS AKHIR

Disusun Oleh :

MOCHAMAD FACHRUDIN
5199 100 006

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2004

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK SPAM FILTERING MENGGUNAKAN METODE BAYESIAN

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan

Untuk Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui,

Dosen Pembimbing I	Dosen Pembimbing II
Febriliyan Samopa, S.Kom, M.Kom	Royyana M. Ijtihadie, S.Kom
NIP. 132 206 858	

SURABAYA

JULI 2004

ABSTRAK

Email telah menjadi kebutuhan bagi kebanyakan orang, sehingga dapat menjadi suatu media yang baik untuk penyebaran suatu informasi. Namun, hal ini kemudian banyak disalahgunakan untuk penyebaran virus dan spam. Oleh karena itu diperlukan suatu alternatif solusi untuk mengurangi efek negatif dari penyalahgunaan tersebut yakni dengan pemanfaatan email filter.

Dalam tugas akhir ini didesain dan diimplementasikan aplikasi email filter yang dapat berkerja pada server email dengan menggunakan metode bayesian. Metode ini memanfaatkan data-data email yang pernah didapat sebelumnya yang diolah secara statistik untuk digunakan sebagai acuan pengklasifikasian email-email yang akan diterima. Dengan proses ini berjalan pada suatu server email, sehingga server dapat mengatur pengiriman hanya untuk email-email yang diklasifikasikan bukan spam dan tidak mengandung virus.

Berdasarkan uji coba yang telah dilakukan, terbukti sistem yang dibuat dapat bekerja dengan baik. Tingkat akurasi dari hasil klasikasinya berkisar antara 95% sampai dengan 99% dengan membutuhkan waktu untuk klasifikasi kurang dari 1 detik pada setiap data email yang diperiksa.

KATA PENGANTAR

Alhamdulillah, tiada kata yang pantas terucap selain puji syukur bagi Allah SWT atas segala rahmat, nikmat, dan karunia-Nya, karena dengan pertolongan dan kekuatan-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul :

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK SPAM FILTERING MENGGUNAKAN METODE BAYESIAN

Laporan Tugas Akhir ini disusun sebagai salah satu persyaratan untuk memperoleh gelar sarjana strata 1 (S1) pada Jurusan Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa masih terdapat banyak kekurangan pada penyusunan Tugas Akhir ini. Oleh karena itu penulis membuka diri dan akan sangat menghargai bila ada kritik maupun saran yang membangun.

Pada kesempatan ini Penulis mengucapkan terima kasih yang sebesar-besarnya atas bantuan dan dukungan yang tak ternilai kepada:

1. Bapak dan Ibu atas segala pengorbanan, motivasi, nasehat, dan doa yang selalu dipanjatkan untuk anaknya, semoga Allah memberikan kebahagiaan dunia dan akhirat.
2. Bapak Febriliyan Samopa, S.Kom, M.Kom selaku dosen pembimbing I yang telah memberikan bimbingan dan arahan kepada penulis dalam pengerjaan Tugas Akhir.

3. Bapak Royyana Muslim, S.Kom selaku dosen pembimbing II yang dengan penuh kesabaran memberikan arahan dan bimbingan kepada penulis dalam pengerjaan Tugas Akhir.
4. Bapak Wahyu Suadi, S.Kom selaku dosen wali
5. Seluruh dosen Teknik Informatika yang telah menyampaikan ilmu dan pengalamannya pada saat proses perkuliahan.
6. Seluruh karyawan Teknik Informatika, terimakasih atas bantuan dan kerjasama yang baik semasa penulis mengikuti perkuliahan.
7. Rully, Liga, Firman, Rifqi, dan Indie, atas bantuannya dalam memberikan ide-ide yang cemerlang dalam proses pengerjaan Tugas Akhir.
8. Kamal, Ervan, Medi, Awe, Bayu, Shiddiq, dan semua teman-teman C0F yang tidak bisa saya sebutkan satu-persatu. Terima kasih atas persahabatan, bantuan dan kebersamaan kita selama ini dan tentu juga untuk selanjutnya.
9. Semua pihak yang telah mendukung penulis hingga Tugas Akhir ini selesai.

Untuk semuanya, penulis mengucapkan terima kasih yang sebesar-besarnya. Semoga Allah SWT menerima dan membalasnya dengan sebaik-baiknya balasan. Penulis mohon maaf yang sebesar-besarnya apabila penulis tidak sengaja pernah melakukan atau mengatakan hal-hal yang tidak berkenan.

Akhir kata penulis berharap agar Tugas akhir ini dapat bermanfaat bagi kampus Teknik Informatika ITS.

Surabaya, Juli 2004

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Seiring perkembangan internet yang demikian cepatnya, semakin banyak orang yang menggunakan layanan internet dalam kehidupan mereka. Mulai dari mencari informasi, saling menukar data, sampai dengan melakukan transaksi secara online. Setiap komputer yang ada di muka bumi ini dapat saling terhubung satu sama lain dan melakukan pertukaran data-data penting sehingga dapat menghemat jarak dan waktu, yang ditawarkan sebagai kemudahan dalam koneksi antar *user*.

Email merupakan suatu sarana yang sangat dibutuhkan oleh kebanyakan orang. Karena dengan email seseorang dapat berkomunikasi dengan orang lain dengan cepat dan murah. Akan tetapi banyak pihak-pihak yang menyalahgunakan email untuk hal-hal yang merugikan, yaitu diantaranya adalah spam dan virus.

Spamming adalah usaha menyampaikan suatu pesan, melalui internet (e-mail), pada seseorang yang tidak meminta pesan tersebut dan bahkan tidak menginginkannya. Hampir semua email spam berisi tentang iklan. Hal ini beralasan karena dengan memperkenalkan suatu produk melalui email, perusahaan tidak perlu mengeluarkan banyak biaya.

Software *e-mail spamming* ditawarkan dengan harga yang murah dan dalam software tersebut sudah tersedia ribuan alamat e-mail yang siap digunakan,

serta pengirim tidak memerlukan biaya tambahan untuk mengirimkan email spamnya, hal tersebut menjadikan jumlah dari spam semakin meningkat.

Spam mail sangat mengganggu bagi user. Misalnya jika dalam satu hari kita menerima 10-20 email spam. Ketika kita tidak membuka email kita dalam waktu 3 hari saja, maka kita akan sangat kesulitan untuk memisahkan antara email yang penting bagi kita dan yang merupakan spam. Dan akhirnya alamat email tersebut tidak bisa dimanfaatkan lagi.

Bagi mail server, spam akan banyak memakan resource dari server (bandwith, disk space). Proses baca tulis ke hardisk yang tidak berhenti akan membuat performa dari mail server menjadi turun.

Saat ini e-mail juga digunakan untuk menyebarkan virus-virus yang dapat merusak sistem komputer. Virus ini bisa berada pada *attachment* ataupun pada isi email. Dan melalui e-mail inilah suatu virus dapat tersebar dengan cepat.

1.2 TUJUAN

Tujuan pembuatan Tugas Akhir ini adalah untuk membuat sebuah aplikasi yang dapat melakukan spam filtering dan pendeteksian virus pada setiap e-mail yang diterima oleh mail gateway server dari remote host.

1.3 PERMASALAHAN

Permasalahan yang diangkat dalam tugas akhir ini adalah :

1. Bagaimana menentukan token-token yang sesuai dari keseluruhan text e-mail yang masuk.

2. Bagaimana penerapan algoritma Bayesian dalam membedakan spam ataupun bukan spam.
3. Bagaimana melakukan otomasi dalam proses filtering e-mail yang masuk.

1.4 BATASAN PERMASALAHAN

Dari permasalahan di atas, maka batasan dalam tugas akhir ini adalah:

1. Filter spam menggunakan algoritma bayesian.
2. Aplikasi hanya dapat berjalan secara optimal dalam melakukan prosesnya pada email yang bentuknya mengikuti kaidah MIME dan yang menggunakan charset US-ASCII.
3. Untuk pendeteksian virus, aplikasi ini memanfaatkan library yang disediakan oleh Clam Anti Virus, sehingga untuk database virus, aplikasi ini hanya memanfaatkan definisi virus yang disediakan oleh Clam Anti Virus.
4. Aplikasi ini tidak mendukung auto training, sehingga dalam penggunaan aplikasi ini, proses training harus dilakukan secara manual oleh sistem administrator.
5. *Tool* yang digunakan dalam digunakan dalam merancang, membuat dan menjalankan aplikasi adalah :
    a. *qmail* sebagai *mail gateway.*
    b. *gcc* sebagai pembangun aplikasi.
    c. *Clam Anti Virus* sebagai engine untuk mendeteksi virus.
6. *Library* tambahan yang digunakan dalam pembuatan aplikasi adalah:

a. *Libmysql* digunakan untuk pengaksesan database mysql.

b. *Libpcre* digunakan untuk memproses regular expression.

c. *Libclamav* digunakan untuk melakukan virus scanning.

1.5 METODOLOGI

Pembuatan tugas akhir ini terbagi menjadi beberapa tahapan pengerjaan, yang akan disebutkan sebagai berikut:

1. **Studi literatur dan perangkat lunak yang sudah ada**

   Studi literatur yang dilakukan adalah mempelajari konsep dan teknologi yang akan digunakan untuk membangun aplikasi tersebut. Selain itu juga mempelajari *software* yang sudah ada yang akan dijadikan acuan dalam pembuatan Tugas Akhir. Informasi tersebut diperoleh dengan membaca literatur ataupun jurnal-jurnal yang berhubungan.

2. **Perancangan perangkat lunak dan desain sistem**

   Pada tahap ini akan dilakukan analisa dan desain terhadap sistem, berdasarkan hasil yang telah diperoleh dari tahap 1.

3. **Pengembangan aplikasi**

   Pada tahap ini dilakukan perancangan dan pembuatan perangkat lunak dengan menggunakan konsep dan teknologi yang telah didapat pada Metodologi Pembuatan Tugas Akhir pada nomor 1.

4. **Uji coba dan evaluasi**

   Pada tahap ini aplikasi sudah selesai dan akan dievaluasi dengan menggunakan berbagai data yang bervariasi dan diperbaiki demi kelayakan software dan keberhasilan software sesuai dengan tujuannya dibangun.

5. Penyusunan Buku Tugas Akhir

Pada tahap terakhir ini akan disusun buku sebagai dokumentasi dari pelaksanaan Tugas Akhir. Dokumen ini juga diharapkan dapat berguna bagi pihak lain yang berkeinginan untuk mengembangkan sistem tersebut.

1.6 SISTEMATIKA PEMBAHASAN

Sistematika penulisan Tugas Akhir ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Dalam bab ini selain diuraikan tentang latar belakang, tujuan, permasalahan, batasan masalah, metodologi, dan sistematika pembahasan dari tugas akhir.

BAB 2 DASAR TEORI

Dalam bab ini memberikan penjelasan tentang teknologi-teknologi yang digunakan dalam mengembangkan perangkat lunak tugas akhir.

BAB 3 PERANCANGAN PERANGKAT LUNAK

Dalam bab ini memberikan penjelasan tentang deskripsi sistem yang akan dibuat, spesifikasi sistem, kebutuhan dan desain sistem dari perangkat lunak yang akan dibangun.

BAB 4 IMPLEMENTASI PERANGKAT LUNAK

Dalam bab ini akan meng-implementasikan hasil rancangan yang telah dibuat dalam bab III, dimana implementasi tersebut berupa prosedur dan fungsi yang digunakan dalam

melakukan beberapa proses dalam aplikasi, integrasi perangkat lunak yang telah dibuat dengan mail server (qmail), dan implementasi qmail sebagai mail gateway (*relay server*).

BAB 5 UJI COBA DAN EVALUASI

Dalam bab ini akan dilakukan uji coba dan evaluasi terhadap aplikasi yang telah dibuat pada tugas akhir ini. Proses ujicoba ini mengevaluasi tingkat keberhasilan aplikasi dalam melakukan proses filtering, dan berapa banyak resource yang dibutuhkan untuk melakukan proses filtering tersebut.

BAB 6 PENUTUP

Bab ini berisi tentang kesimpulan yang dapat diambil dari pengerjaan tugas akhir ini, dan saran-saran untuk mengembangkan sistem lebih lanjut.

BAB II

DASAR TEORI

BAB II

DASAR TEORI

2.1 E-MAIL

E-mail (*Electronic Mail*) adalah pesan yang dikirimkan melalui jaringan komputer kepada suatu grup maupun perorangan. E-mail berupa pesan teks, yang didalamnya user dapat menambahkan suatu file gambar, suara maupun video. Untuk mengirim dan menerima e-mail diperlukan koneksi jaringan yang telah terhubung ke komputer, dan sebuah alamat e-mail. Suatu alamat e-mail mengandung karakter @, misalnya: andi@yahoo.com.

Untuk mendapatkan alamat e-mail harus dilakukan proses registrasi pada suatu *mail server*. *Mail Server* inilah yang berfungsi untuk mengatur proses pengiriman dan penerimaan e-mail. Teknologi-teknologi yang ada pada mail server meliputi:

- SMTP (*Simple Mail Transfer Protocol*)

  Fungsi standar untuk mengatur komunikasi antar mail server melalui internet.

- POP3

  Suatu protocol e-mail yang digunakan untuk mengambil pesan yang tersimpan pada internet/intranet mail server. POP3 cocok digunakan oleh user yang menggunakan koneksi internet via dial-up (modem) karena user dapat mengambil semua e-mailnya yang tersimpan di mail server dan kemudian menyimpan e-mail tersebut di komputernya.

Untuk membaca semua e-mail tersebut user tidak perlu melakukan koneksi lagi dengan mail server. Sehingga dapat menghemat biaya.

- IMAP (*Internet Message Access Protocol*)

  Dapat menggantikan POP3 sebagai protocol utama yang digunakan oleh *email client* dalam berkomunikasi dengan mail server. Dengan menggunakan IMAP suatu *mail client* tidak hanya dapat mengambil email dari *mail server*, tetapi dapat juga memanipulasi email yang tersimpan di server tanpa harus mengambil/menerima email tersebut. Jadi email dapat dihapus, diubah statusnya dan protocol ini mendukung penggunaan beberapa *mailbox* oleh seorang user.

2.2 MIME (MULTIPURPOSE INTERNET MAIL EXTENSION)

Pada standar yang telah didefinisikan sebelumnya untuk bentuk suatu *message body* (RFC 822), yang mendefinisikan header suatu email dengan berdasarkan US-ASCII karakter dan mendefinisikan isi dari pesannya (*message body*) berupa teks dengan menggunakan US-ASCII karakter pula. MIME memperbarui format dari suatu email sehingga :

- isi dari email dapat mengandung teks yang tersusun dari karakter set selain US-ASCII,
- pendifinisian format yang berbeda untuk message body yang berikan pesan non-tekstual ,
- *message body* yang dapat terdiri dari banyak bagian (*multipart*), dan
- informasi header dalam karakter selain US-ASCII.

Metode yang digunakan untuk membaca suatu email disimpan pada bagian header email. Berikut ini akan dijelaskan beberapa informasi yang dapat diletakkan pada bagian header email.

2.2.1 Content-Type

Tujuan dari adanya field content-type adalah untuk mendeskripsikan data yang tersimpan pada body email sehingga user agent yang menerima email dapat menentukan mekanisme yang sesuai untuk dapat memberikan data tersebut kepada user. Nilai yang tercantum dalam content-type ini disebut *media type*.

Syntax dari content-type ditunjukkan pada gambar berikut:

```
content:= "Content-Type" ":" type "/" subtype  *(";" parameter)
type := discrete-type / composite-type
discrete-type := "text" / "image" / "audio" / "video" /
                 "application" / extension-token
composite-type := "message" / "multipart" / extension-token
extension-token := ietf-token / x-token
ietf-token :=<Sebuah extension token yang telah didefinisikan
              dengan standar RFC>
x-token := <String "X-" atau "x-" diikuti dengan tokennya tanpa
            didahului dengan spasi>
subtype := extension-token / iana-token
iana-token := <Sebuah publicly-defined extension token.>
parameter := attribute "=" value
attribute := token       ; attribute tidak case sensitive
value := token / quoted-string
token := 1*<Sembarang (US-ASCII) karakter kecuali SPACE, CTLs,
          atau tspecials>
tspecials :=  "(" / ")" / "<" / ">" / "@" / "," / ";" / ":" /
              "\" / <"> "/" / "[" / "]" / "?" / "="
```

Gambar 2.1 Sintaks dari Content-Type

Media Type terbagi dalam 2 bentuk utama yaitu *discrete* dan *composite*. Bentuk *discrete* terdiri dari *text*, *image*, *audio*, *video* dan *application*. Bentuk composite terdiri dari multipart dan message.

2.2.1.1 Multipart

Dalam multipart, terdapat satu atau lebih satuan data yang berbeda dikombinasikan dalam sebuah badan tunggal, media type "multipart" harus tercantum dalam header utama. *Body* email harus berisi satu atau lebih komponen *body*, masing-masing didahului oleh suatu garis membatasi (*boundary*), dan yang terakhir yang diikuti oleh suatu penutupan garis membatasi (*boundary*). Setelah *boundary*, body part berisikan area untuk header, suatu baris kosong, dan area untuk bagian isi.

Field dari Content-Type untuk entitas multipart memerlukan satu parameter, "boundary". Baris pemisah (boundary) menggambarkan sebagai satu baris terdiri atas dua tanda penghubung karakter ("-", nilai desimal 45) yang diikuti oleh nilai parameter boundary dari field Content-Type pada header, optional *whitespace* linier, dan diakhiri dengan CRLF.

Bentuk dari *content-type*-nya menjadi :

Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p

Tetapi contoh dibawah ini adalah salah

Content-Type: multipart/mixed; boundary=gc0pJq0M:08jU534c0p

Contoh tersebut salah karena adanya karakter ":" (titik dua). Supaya benar maka seharusnya ditulis dalam bentuk:

Content-Type: multipart/mixed; boundary="gc0pJq0M:08jU534c0p"

Sebagai contoh sangat sederhana, pesan berbentuk *multipart* berikut mempunyai dua komponen, keduanya berupa *plain text*, salah satu dengan ditulis secara eksplisit dan yang lain ditulis secara implisit:

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Date: Sun, 21 Mar 1993 23:56:48 -0800 (PST)
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"

This is the preamble.  It is to be ignored, though it
is a handy place for composition agents to include an
explanatory note to non-MIME conformant readers.

--simple boundary

This is implicitly typed plain US-ASCII text.
It does NOT end with a linebreak.

--simple boundary
Content-type: text/plain; charset=us-ascii

This is explicitly typed plain US-ASCII text.
It DOES end with a linebreak.

--simple boundary--

This is the epilogue.  It is also to be ignored.
```

Gambar 2.2 Contoh multipart message

2.2.2 Content-Transfer-Encoding

Banyak *media type* yang dikirimkan melalui email direpresentasikan dalam bentuk asalnya, sebagai 8bit karakter ataupun sebagai data biner. Data seperti itu tidak dapat dikirimkan melalui protokol transfer tertentu. Sebagai contoh RFC 821 (SMTP) membatasi pesan email sebagai teks 7bit ASCII dengan maksimal 1000 karakter termasuk CRLF.

Sangat diperlukan untuk melakukan *encoding* data-data tersebut menjadi bentuk 7bit ASCII. Metode untuk *encoding*-nya diletakkan pada content-transfer-encoding ini. Sintaks dari content-transfer-encoding adalah sebagai berikut:

```
encoding := "Content-Transfer-Encoding"":" mechanism
mechanism := "7bit" / "8bit" / "binary" /
             "quoted-printable" / "base64" /
              ietf-token / x-token

contoh penggunaan content-transfer-encoding:
 Content-Type: text/plain; charset=ISO-8859-1
 Content-transfer-encoding: base64
```

Gambar 2.3 Sintaks dan contoh dari content-transfer-encoding

Nilainya tidak *case sensitive,* jadi antara base64, Base64, ataupun BaSe64 adalah dianggap sama. Nilai default dari content-transfer-encoding ini adalah 7bit. Sehingga bila content-transfer-encoding tidak tercantum dalam header suatu email, email tersebut menggunakan 7bit encoding.

Jika nilai dari content-transfer-encoding berupa 7bit, 8bit, ataupun binary menunjukkan bahwa tidak dilakukan proses encoding pada data yang dikirimkan.

Jika content-transfer-encoding ini tercantum pada header utama suatu email, menunjukkan bahwa metode encoding tersebut yang digunakan pada seluruh isi dari email tersebut. Jika tercantum pada bagian *body part,* maka metode encoding tersebut hanya dipakai pada *body part* yang bersangkutan.

2.2.2.2 Quoted-Printable Encoding

Dalam bentuk encoding ini octet direpresentasikan dalam suatu bentuk yang mengacu pada aturan-aturan berikut ini:

1. Representasi karakter 8bit secara umum

   Semua oktet, kecuali CR atau LF yang merupakan bagian dari line break (CRLF) dari bentuk standar dari yang sedang di-*encode*, bisa direpresentasikan dengan "=" diikuti oleh dua digit heksadesimal yang merupakan representasi dari nilai ASCII oktetnya. Karakter heksadesimal yang digunakan adalah "0123456789ABCDEF". Harus menggunakan huruf besar. Sebagai contoh, nilai desimal 12 (US-ASCII form feed) dapat direpresentasikan sebagai "=0C", dan nilai desimal 61 (US-ASCII EQUAL SIGN) dapat direpresentasikan sebagai "=3D".

2. Representasi literal

   Oktet dengan nilai desimal dari 33 s.d 60 dan 62 s.d 126, dapat direpresentasikan sebagai karakter ASCII yang sesuai.

3. White Space

   Oktet dengan nilai 9 (US-ASCII TAB) dan 32 (US-ASCII SPACE) dapat direpresentasikan sebagai karakter ASCII-nya. Tetapi karakter ini tidak boleh diletakkan pada akhir baris. Karakter ini harus diikuti oleh karakter yang tercetak (*printable*). Misalnya, karakter "=" dapat mengikuti karakter *whitespace,* pada data yang di-*encode* untuk menunjukkan adanya *soft line break.* Jika karakter terakhir pada suatu baris merupakan *whitespace,* maka mengikuti aturan 1.

4. Line Break

   Suatu baris baru pada body pesan teks, direpresentasikan dengan CRLF.

5. Soft Line Break

   Quoted-Printable encoding mengharuskan 1 baris yang di-*encode* panjangnya tidak lebih dari 76 karakter. Jika dalam satu baris terdapat lebih dari 76 karakter, maka harus digunakan soft line break. Karakter "=" digunakan pada akhir baris untuk menandai adanya soft line break.

   Misalnya, pada baris yang berisi "Now's the time for all folk to come to the aid of their country.". Direpresentasikan menjadi:

   Now's the time =

   for all folk to come=

   to the aid of their country.

2.2.2.3 BASE64 Encoding

Base64 encoding mengambil tiga bytes, masing-masing terdiri dari delapan bit, dan menghadirkannya menjadi empat *printable* karakter di standard ASCII. Proses ini dikerjakan dalam dua langkah utama.

Langkah yang pertama akan mengkonversi tiga bytes menjadi empat angka-angka enam bit. Masing-Masing karakter dalam standard ASCII terdiri dari tujuh bit. Base64 hanya menggunakan 6 bit (sesuai dengan 2^6= 64 karakter) untuk memastikan data yang disandikan adalah data yang printable dan dapat dibaca. Tidak satupun dari karakter khusus yang tersedia dalam ASCII digunakan. 64 karakter (karenanya disebut Base64) adalah 10 digit, 26 karakter hurufkecil, 26 huruf besar, dan karakter '+' dan '/'.

Sebagai contoh, ke tiga bytes adalah 155, 162 dan 233, nilai bit yang bersesuaian adalah 100110111010001011101001, yang pada gilirannya sesuai dengan 6-bit menilai 38, 58, 11 dan 41.

Angka-Angka ini dikonversi ke karakter ASCII di langkah yang kedua menggunakan tabel Base64 encoding. Sehingga dari contoh tersebut diatas dapat diterjemahkan dalam urutan ASCII menjadi " m6Lp".

155 -> 10011011

162 -> 10100010

233 -> 11101001

100110 -> 38

111010 -> 58

001011 -> 11

101001 -> 41

38 -> m

58 -> 6

11 -> L

41 -> p

Dua langkah proses ini diberlakukan bagi keseluruhan urutan bytes yang di-*encode*. Untuk memastikan data yang disandikan dapat dengan baik dicetak dan tidak melebihi batas panjangnya baris, newline karakter dimasukkan/disisipkan untuk menjaga satu baris hanya mempunyai panjang 76 karakter. Newline karakter yang ada pada message asli disandikan seperti semua data lain.

Pada akhir proses *encode* mungkin ditemui suatu masalah. Jika ukuran dari data asli di bytes adalah kelipatan tiga, segalanya bekerja bagus. Jika tidak, data akan berakhir dengan satu atau dua 8-bit bytes. Padahal dalam proses encoding kita memerlukan persisnya tiga bytes.

Solusinya adalah dengan menambahkan bytes dengan suatu nilai ' 0' untuk menciptakan suatu kelompok 3-byte. Dua nilai '0' ditambahkan jika kita mempunyai satu byte data ekstra, satu nilai '0' ditambahkan jika ada dua ekstra bytes.

Nilai ' 0' ini tidak bisa diencode menggunakan tabel base64 encoding tersebut. Sehingga harus diwakili oleh suatu karakter ke-65. Untuk ini karakter yang digunakan adalah '='. Karakter '=' ini hanya akan muncul pada akhir data hasil proses encoding.

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

Gambar 2.4 Karakter Base64 Encoding

2.3 SPAM

Biasa disebut sebagai email sampah (*junk mail*). Merupakan email tidak diinginkan yang diterima secara terus-menerus oleh pengguna.

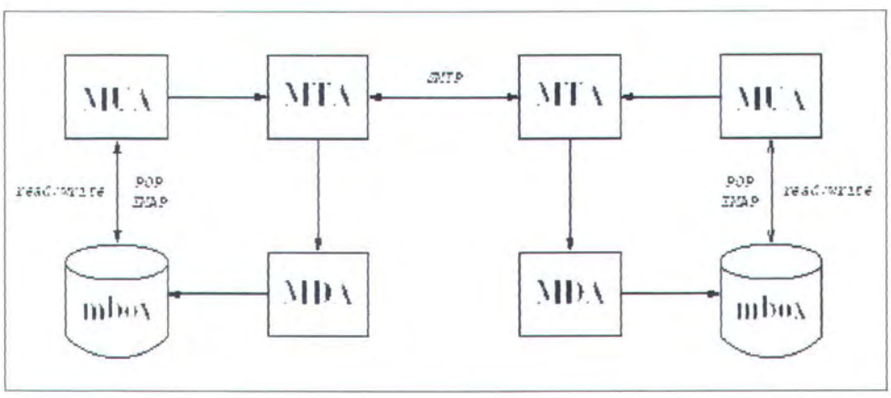
Masalah yang timbul dengan adanya spam ini diantaranya adalah:

- ***Cost-Shifting***. Pengiriman email spam memerlukan biaya yang murah. Dengan koneksi dial-up 28.8Kbps dan sebuah pc, spammer dapat mengirimkan ribuan email tiap jamnya. Tetapi dengan adanya spam, seorang pengguna email perlu mengeluarkan biaya tambahan untuk mengakses emailnya. Bagi ISP adanya spam menyebabkan peningkatan penggunaan bandwitdh.
- ***Fraud***. Spammer mengetahui bahwa sebagian besar penerima spam tidak mengharapkan email-email tersebut. Sehingga spammer menggunakan trik-trik tertentu supaya penerima membuka email tersebut. Misalnya dengan menggunakan subject email yang tidak berhubungan sama sekali dengan isi email yang dikirimkannya.
- ***Displacement of Normal Email.*** Banyaknya jumlah spam pada mailbox, membuat pengguna kesulitan dalam membedakan antara email spam dengan email yang penting. Sehingga efektifitas dan kegunaan dari email akan hilang.
- ***Annoyance Factor***. Koneksi internet yang terbatas membuat akses email bagi sebagian pengguna bukanlah hal yang dapat dilakukan dengan cepat. Akan sangat mengganggu bila telah dihabiskan

waktu lama untuk membuka mailbox, ternyata dari email-email yang diterima sebagian besar adalah spam.

2.4 QMAIL

User menyusun suatu pesan dengan menggunakan suatu MUA. MUA memberikannya kepada MTA untuk menyusun pengiriman. Jika pesan tersebut dikirimkan ke server lokal, MTA memberikannya kepada MDA yang lokal yang mengirimkannya kepada mailbox yang lokal yang tersedia untuk dilakukan pembacaan oleh MUA. Proses transaksi email ditunjukkan pada Gambar 2.5.



Gambar 2.5 Transaksi email sederhana

Jika pesan adalah untuk suatu penerima remote, MTA mengirimkannya ke suatu remote MTA yang kemudian memberikannya untuk MDA yang kemudian mengirimkannya untuk mailbox yang remote itu.

MUA , *Mail User Agent*, mengijinkan suatu end-user untuk membaca email yang datang, meresponnya dan menyusun suatu email baru.

MTA, *Mail Transport Agent*, bertanggung jawab untuk mengangkut email dari satu lokasi menuju ke lokasi yang lain.

MDA, *Mail Delivery Agent*, bertanggung jawab untuk mengirimkan suatu pesan email ke tujuan akhir nya.

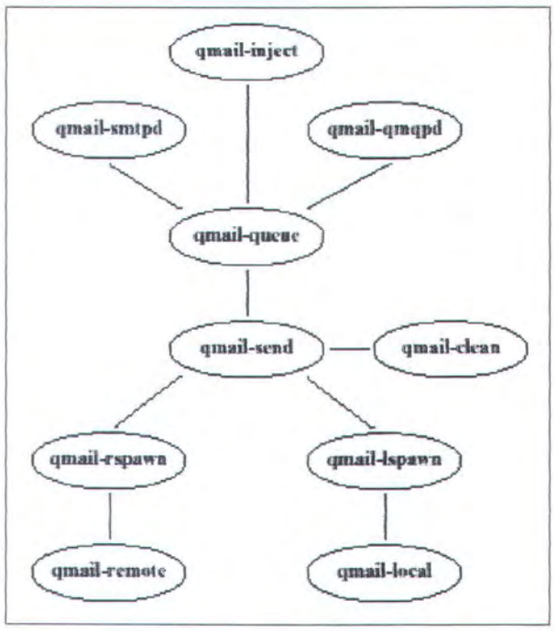
qmail adalah sebuah *internet mailer*. qmail hanya akan menggunakan SMTP untuk berkomunikasi dengan MTA-MTA yang lain. Tetapi qmail dapat dikonfigurasi untuk menggunakan protokol yang lebih efisien ketika berkomunikasi dengan qmail system yang lain.

qmail dikenal sebagai MTA yang sangat *secure*. Setiap tingkatan pemrosesan pesan dikerjakan oleh program yang berbeda. Setiap program berjalan pada user id yang berbeda dan spesifik.

Email datang melalui suatu koneksi SMTP ke qmail-smtpd, atau suatu koneksi QMQP ke qmail-qmqpd, atau via antrian tidak langsung yang berasal dari qmail-inject.

Qmail-queue dipanggil untuk menempatkan pesan dalam antrian itu. qmail-queue memisahkan pesan ke dalam tiga file, pesan itu sendiri dan dua salinan menyangkut detil amplop. kemudian diberikan isyarat pada qmail-send untuk mulai pengiriman.

Jika pesan adalah untuk suatu penerima lokal, qmail-send memerintahkan qmail-lspawn untuk mengirimkan pesan itu. qmail-lspawn memanggil qmail-local untuk mengirimkan pesan itu. qmail-local menangani perluasan alias dan, jika perlu, memerintahkan qmail-queue untuk menempatkan suatu pesan baru dalam antrian. qmail-local akhirnya mengirimkan pesan kepada direktori user yang dituju. Proses kerja pada qmail ditunjukkan Gambar 2.6.



Gambar 2.6 alur proses kerja pada qmail

Jika pesan adalah untuk suatu penerima remote, qmail-send memerintahkan qmail-rspawn untuk mengirimkan pesan itu. qmail-rspawn menghubungi qmail-remote untuk mengirimkan pesan kepada suatu MTA remote.

2.5 TEORI BAYESIAN

Opini tentang teorema bayes sebagai dasar untuk inferensi statistik telah menjadi perdebatan antara pihak yang menerima dan menolaknya sejak teorema ini dipublikasikan pada tahun 1763. Pada waktu itu diperkirakan bahwa argumen alternatif menyediakan sebuah pondasi yang memuaskan untuk hasil inferensi bayesian dikemukakan sebagai sesuatu yang menarik tetapi merupakan cara yang kurang tepat untuk memecahkan suatu masalah. Dan selanjutnya ditemukan adanya kesulitan-kesulitan yang tidak terduga yang muncul mengikuti alternatif-altenatif tersebut, sehingga menimbulkan ketertarikan terhadap teorema bayesian.

Teorema bayes untuk *reasoning*, yang telah dilupakan penggunaannya, akhirnya diperlukan kembali.

Teori bayes mengemukakan suatu hubungan antar kemungkinan marginal dan bersyarat. dapat dipandang bermakna dalam menginterpretasikan suatu informasi, dari suatu pengamatan [2], sebagai contoh, untuk menghasilkan suatu modifikasi atau membaharui distribusi kemungkinan. Teori bayes dipertimbangkan sah dalam semua penafsiran kemungkinan, tetapi penerapannya mempunyai lingkup lebih sedikit atau lebih besar. Probabilitas Bayesian didasarkan pada dugaan yang teori Bayes dapat diberlakukan pada proporsi apapun, apakah statemen tentang variabel yang bersifat percobaan, parameter, hipotesis, variabel tersembunyi (tidak diamati), atau berbagai macam statemen yang lain. Tetapi, menurut *frequentist school*, teori bayes hanya dapat diberlakukan bagi permasalahan di mana semua statemen adalah sekitar variabel bersifat percobaan sendiri, dan oleh karena itu beberapa mekanisme kesimpulan non-probabilistic harus digunakan untuk memberi alasan sekitar statemen yang lain [2].

Keteraturan statistik telah mendorong pengembangan dari probabilitas menggunakan konsep frekwensi relatif. kebanyakan prosedur ini biasanya digunakan untuk membuat test atau perkiraan statistik yang dikembangkan oleh orang-orang statistik menggunakan konsep ini secara eksklusif. Mereka pada umumnya disebut *frequentists*, dan posisi mereka disebut *frequentism. frequentist school* sering dihubungkan dengan nama Jerzy Neyman dan Egon Pearson yang menguraikan logika pengujian hipotesis statistik. figur lain yang Berpengaruh

menyangkut frequentist school meliputi John Venn, R.A. Fisher, dan Richard von Mises.

Misalkan y'=(y1,y2,...,yn) adalah suatu vector dari n kali pengamatan yang mampunyai distribusi kemungkinan p(y|θ) bergantung pada nilai parameter θ`=(θ1,..., θk). Misal parameter θ juga mempunyai distribusi kemungkinan p(θ). Sehingga :

$$p(y|\theta)p(\theta) = p(y,\theta) = p(\theta|y)p(y) \qquad (1)$$

Dengan data hasil pengamatan y, distribusi *conditional* dari θ adalah

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \qquad (2)$$

Yang dapat ditulis sebagai:

$$p(y) = \mathrm{E}p(y|\theta) = c^{-1} = \begin{cases} \int p(y|\theta)p(\theta)d\theta \Rightarrow \theta = continous \\ \sum p(y|\theta)p(\theta) \Rightarrow \theta = discrete \end{cases} \qquad (3)$$

Ketika jumlah maupun integralnya diambil dari range θ tertentu, dan ketika E[f(θ)] adalah suatu ekspektasi matematis dari f(θ) dengan melihat distribusi p(θ). Sehingga dapat diambil persamaan alternatif untuk 2.5.2 menjadi:

$$p(\theta|y) = cp(y|\theta)p(\theta) \qquad (4)$$

Persamaan 2.5.2 ataupun 2.5.4 inilah yang disebut sebagai teorema bayes. Dalam persamaan tersebut, p(θ),menunjukkan sesuatu yang diketahui tentang θ tanpa diketahui knowledge tentang datanya, disebut distribusi *prior*. Berikutnya, p(θ|y), yang menunjukkan nilai setelah kita mengetahui knowledge data θ, disebut dengan distribusi posterior dari θ dengan data y.

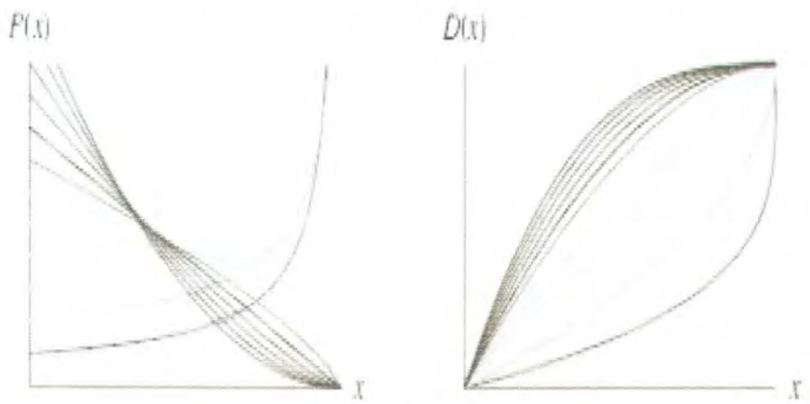
Sekarang dengan data y, p(y|θ) in 2.5.4 bisa dibuat suatu fungsi bukan dari y tetapi dari θ. Berdasarkan Fisher(1922), hal ini disebut sebagai fungsi *likelihood* dari θ untuk data y yang diberikan dapat dituliskan dengan l(θ|y). Sehingga persamaan dari teori bayes dapat dituliskan:

$$p(\theta \mid y) = l(\theta \mid y)p(\theta) \qquad (5)$$

Dengan kata lain, teorema bayes menunjukkan bahwa distribusi probabilitas posterior untuk θ pada data y adalah sebanding dengan perkalian antara distribusi prior data θ dan *likelihood* dari θ terhadap y.

2.6 DISTRIBUSI BETA

Adalah suatu tipe distribusi dalam ilmu statistik. Distribusi beta mempunyai dua parameter bebas yang diberi label menurut salah satu dari dua konvensi notational. Definisi variabelnya biasanya menggunakan α dan β ataupun menggunakan $\alpha' = \alpha - 1$ dan $\beta' = \beta - 1$. Distribusi beta digunakan sebagai distribusi prior untuk proporsi binomial dalam analisa Bayesian [1].



Gambar 2.7 Distribusi beta

Gambar 2.7 menunjukkan suatu distribusi beta dengan $\alpha = 1$ dan β yang bernilai diantara 0.25 s.d 3.00. Domainnya adalah [0,1] dan fungsi probabilitas P(x) fungsi distribusi D(x) didapatkan dari persamaan:

$$P(x) = \frac{(1-x)^{\beta-1} x^{\alpha-1}}{B(\alpha,\beta)} \tag{6}$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} (1-x)^{\beta-1} x^{\alpha-1} \tag{7}$$

$$D(x) = I(x;a,b) \tag{8}$$

Dimana $B(\alpha,\beta)$ merupakan fungsi beta, $I(x;a,b)$ merupakan fungsi beta yang *regularized* dan $\alpha, \beta > 0$.

Fungsi beta yang regularized didefinisikan seperti pada persamaan 9:

$$(p-2)(q-2) = 4 \tag{9}$$

$B(z;a,b)$ fungsi beta yang tidak sempurna dan $B(a,b)$ adalah fungsi beta (yang sempurna).

Distribusi ini dinormailisir:

$$\int_0^1 P(x)dx = 1 \tag{10}$$

Fungsi karakteristiknya adalah:

$$\Phi(t) = \int_0^1 \frac{x^{a-1}(1-x)^{b-1}}{\beta(a,b)} e^{-2\pi i x t} dx \tag{11}$$

$$= {}_1F_1(a;a+b;it)$$

${}_1F_1(a;b;z)$ adalah sebuah *confluent hypergeometric function.*

Raw moment diberikan oleh fungsi:

$$\mu'_{\tau} = \int_0^1 P(x)(x-\mu)^{\tau} dx = \frac{\Gamma(\alpha+\beta)\Gamma(\alpha+\tau)}{\Gamma(\alpha+\beta+\Gamma)\Gamma(\alpha)} \qquad (12)$$

dan *central moment oleh:*

$$\mu_{\tau} = \left(-\frac{\alpha}{\alpha+\beta}\right)^{\tau} {}_2F_1\left(-\tau,\alpha;\alpha+\beta;\frac{\alpha+\beta}{\alpha}\right) \qquad (13)$$

${}_2F_1(a;b;c;x)$ adalah sebuah *hypergeometric function.*

Fungsi rata-rata, varian, kemiringan, dan kurtosisnya ditunjukkan oleh fungsi 14 s.d 17

$$\mu = \frac{\alpha}{\alpha+\beta} \qquad (14)$$

$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \qquad (15)$$

$$\gamma_1 = \frac{2(\beta-\alpha)\sqrt{1+\alpha+\beta}}{\sqrt{\alpha\beta}(2+\alpha+\beta)} \qquad (16)$$

$$\gamma_2 = \frac{6\left[\alpha^3+\alpha^2(1-2\beta)+\beta^2(1+\beta)-2\alpha\beta(2+\beta)\right]}{\alpha\beta(\alpha+\beta+2)(\alpha+\beta+3)} \qquad (17)$$

Sehingga variasi hasil suatu eksperimen dengan distribusi $B(\alpha,\beta)$ adalah

$$\hat{x} = \frac{\alpha-1}{\alpha+\beta-2} \qquad (18)$$

2.7 REGULAR EXPRESSION

Suatu Regular Expression (regex) adalah suatu cara menguraikan suatu pola untuk dibandingkan dengan suatu teks. Definisi yang direktif akan menetapkan apa yang dibandingkan dengan regex. Menguraikan suatu pola dalam

teks misalnya, semua kata-kata yang mulai dengan huruf "A" atau "tiap-tiap 10-digit nomor telepon" atau bahkan "tiap-tiap kalimat dengan dua tanda koma di dalamnya, dan tidak mengandung huruf besar Q".

Regex ini bermanfaat dalam suatu pemrograman sebab dapat membuat beberapa atribut tertentu untuk dibandingkan terhadap koleksi sumber daya atau file dengan cara yang sangat fleksibel. Sebagai contoh, semua file .gif dan .jpg di direktori tertentu bisa ditulis dengan "/ images/.*(jpg|gif)$".

Pada bahasa pemrograman c di sistem operasi linux (gcc), tersedia library untuk melakukan operasi regex ini. Library yang dipakai adalah Perl Compatible Regular Expressions (PCRE).

Library PCRE adalah satu set fungsi yang menerapkan regex untuk melakukan perbandingan pattern yang menggunakan ilmu semantik dan sintaksis yang sama dengan Perl 5, dengan hanya beberapa perbedaan. Implementasi yang sekarang sesuai dengan Perl 5.005, dengan beberapa corak tambahan dari versi kemudiannya. Ini meliputi beberapa pendukungan bersifat percobaan dan belum sempurna untuk string UTF-8 encoded.

Fungsi pcre_compile(), pcre_study(), dan pcre_exec() digunakan untuk menyusun dan mempertemukan regex. Fungsi pcre_copy_substring(), pcre_get_substring(), dan pcre_get_substring_list() berfungsi untuk mengambil substring yang sesuai dari suatu string pokok yang dibandingkan; pcre_free_substring() dan pcre_free_substring_list() diberikan untuk membebaskan memori yang digunakan untuk penyimpan substringnya.

Fungsi pcre_compile() dipanggil untuk menyusun suatu pola dalam suatu format internal. Pola adalah suatu C string yang diakhiri oleh karakter NULL. *Return value* fungsi ini berupa sebuah pointer menunjuk ke blok memori tunggal yang diperoleh melalui pcre_malloc. berisi kode yang di-compile dan berhubungan data.

Potongan kode pada Gambar 2.8 menunjukkan contoh penggunaan pcre_compile() secara default.

Fungsi pcre_exec() digunakan untuk membandingkan suatu string pokok terhadap suatu pola yang telah di-*compile*. contoh sederhana dari penggunaan pcre_exec() ini dapat dilihat pada Gambar 2.9.

```
pcre *re;
const char *error;
int erroffset;
re = pcre_compile(
  "^A.*Z",          /* Polanya*/
  0,                /* option default*/
  &error,           /* untuk menyimpan pesan error */
  &erroffset,       /* untuk error offset*/
  NULL);            /* tabel karakter yang digunakan*/
```

Gambar 2.8 Contoh Penggunaan fungsi pcre_compile

Ada beberapa pembatasan ukuran dalam PCRE tetapi diharapkan agar mereka tidak akan pernah ada dalam praktek sebenarnya. Panjangnya pola yang akan di-*compile* maksimal adalah 65539 bytes. Semua nilai-nilai di pengulangan quantifiers harus kurang dari 65536. Jumlah maksimum *subpatterns* yang di-*capture* adalah 65535. Tidak ada batas untuk *non-capturing* subpatterns, hanya kedalaman maksimum bersarang dalam segala bentuk *subpattern parenthesized*,

termasuk subpatterns yang di-*capture*, pernyataan, dan jenis lain subpattern, adalah 200.

```
  int rc;
  int ovector[30];
  rc = pcre_exec(
    re,             /* hasil dari pcre_compile() */
    NULL,           /* jika tidak melakukan pcre_study diisi
dengan NULL*/
    "some string",  /* String yang akan dibandingkan*/
    11,             /* panjang dari stringnya*/
    0,              /* menunjukkan perbandingan dimulai dari
offset ke- 0*/
    0,              /* default options */
    ovector,        /* vector untuk informasi substring */
    30);            /* jumlah element dalam vector */
```

Gambar 2.9 Contoh Penggunaan Fungsi pcre_exec

Panjang maksimal dari suatu string pokok adalah jumlah yang paling besar yang dapat di-*handle* oleh integer. PCRE menggunakan rekursi untuk menangani subpatterns dan pengulangan tak tentu. Ini mengakibatkan *stack space* yang tersedia membatasi ukuran dari suatu string pokok yang dapat diproses oleh pola tertentu.

2.8 N-GRAM

Pada suatu *object* yang dapat direpresentasikan oleh sebuah urutan, maka salah satu cara untuk melakukan ekstraksi fiturnya adalah dengan mendeskripsikan *object* tersebut dalam pola dari bagian urutannya. N-gram adalah bagian dari suatu urutan dengan panjang n [7].

Untuk nilai n < 5 umumnya digunakaan penamaan latin, misal: bigram, trigram. Sedangkan untuk nilai n ≥ 5 digunakan awalan numerik, misal: 5-gram, 6-gram.

Metode n-gram dapat diterapkan pada suatu permasalahan yang berhubungan dengan *sequences*, misalnya:

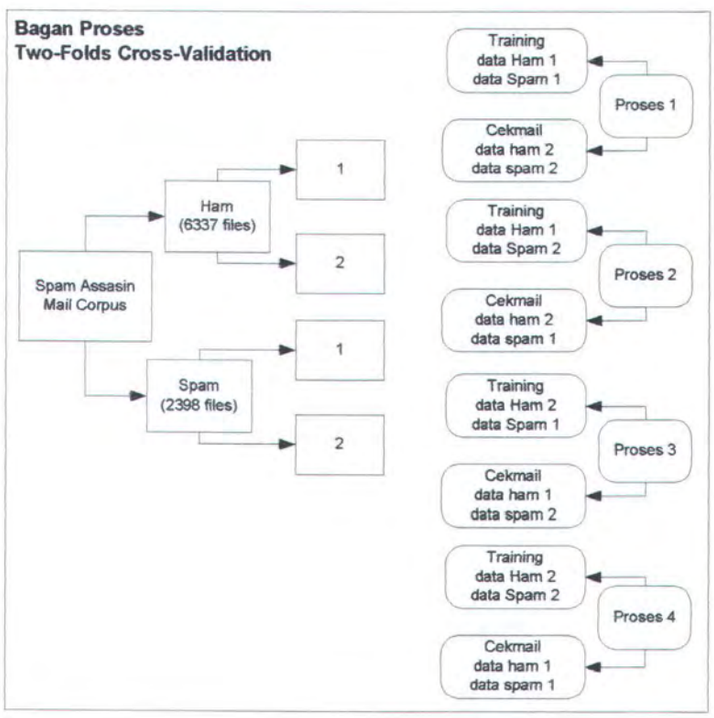
- Kompresi teks
- *Spelling error detection and correction*
- Pengenalan *optical character*
- *Information retrieval*
- Representasi tekstual
- Identifikasi bahasa
- Information filtering
- Automatic text categorization
- Medical record matching

2.9 METODE UJI COBA DAN EVALUASI

2.9.1 N-Folds Cross-Validation

N-folds cross-validation merupakan metode yang digunakan dalam uji coba aplikasi yang dibuat dalam Tugas Akhir ini. Dengan menggunakan metode ini, diharapkan variasi tingkat kesalahan yang ada dapat diminimalkan, sehingga hasil uji coba dapat merepresentasikan performa yang sebenarnya dari aplikasi yang telah dibuat.

Metode ini dilakukan dengan cara membagi data email corpus yang ada menjadi n bagian. Setiap satu bagian data dari email spam dan satu bagian dari email bukan spam digunakan untuk training dan bagian lainnya digunakan untuk uji coba. Selanjutnya akan dilakukan uji coba silang terhadap data-data tersebut. Sehingga akan dilakukan sebanyak n^2 uji coba dengan data training dan data yang diuji yang berbeda-beda. Pada Gambar 2.10 ditunjukkan detail proses uji coba secara two-fold cross-validation.



Gambar 2.10 Bagan proses two-folds cross-validation

2.9.2 Evaluasi Hasil Uji Coba

Dari proses uji coba yang dilakukan, dapat menghasilkan salah satu dari 4 macam kondisi yang berbeda, yaitu:

- True Accepted ($n_{H\to H}$), kondisi ketika suatu email bukan spam diklasifikasikan sebagai bukan spam.
- False Rejected ($n_{S\to S}$), kondisi ketika suatu email spam diklasifikasikan sebagai spam.
- True Rejected ($n_{H\to S}$), kondisi ketika suatu email bukan spam diklasifikasikan sebagai spam.
- False Accepted ($n_{S\to H}$), kondisi ketika suatu email spam diklasifikasikan sebagai bukan spam.

Pada proses klasifikasi data, terdapat dua macam nilai yang dapat dijadikan tolok ukur hasil evaluasi, yaitu keakuratan (Acc) dan tingkat kesalahan (Err). Dalam hal klasifikasi yang dilakukan perangkat lunak ini, keakuratan dan tingkat kesalahannya dapat dihitung dengan cara sebagai berikut.

$$Acc = \frac{n_{H\to H} + n_{S\to S}}{N_H + N_S} \qquad Err = \frac{n_{H\to S} + n_{S\to H}}{N_H + N_S} \tag{26}$$

N_S dan N_H adalah jumlah dari email spam dan bukan spam yang akan diklasifikasikan. Akan tetapi bila email bukan spam terklasifikasikan sebagai spam ($n_{H\to S}$) dianggap menimbulkan kerugian λ kali lebih banyak daripada bila email spam terklasifikasikan sebagai bukan spam ($n_{S\to H}$), sehingga diperlukan perhitungan yang menghasilkan nilai yang sensitive terhadap λ (WAcc). Nilai WAcc ini dapat dihitung dengan menggunakan persamaan sebagai berikut.

$$WAcc = \frac{(\lambda \cdot n_{H\to H}) + n_{S\to S}}{(\lambda \cdot N_H) + N_S} \qquad WErr = \frac{(\lambda \cdot n_{H\to S}) + n_{S\to H}}{(\lambda \cdot N_H) + N_S} \tag{27}$$

Dalam penggunaan nilai keakuratan dan tingkat kesalahan, diperlukan suatu nilai pembanding. Dalam hal ini yang digunakan sebagai pembanding adalah kondisi dimana semua email dapat masuk tanpa harus di-*filter*. Nilainya dihitung dengan persamaan sebagai berikut.

$$WAcc_B = \frac{\lambda \cdot N_H}{(\lambda \cdot N_H) + N_S} \qquad WErr_B = \frac{N_S}{(\lambda \cdot N_H) + N_S} \tag{28}$$

Sehingga nilai yang merepresentasikan tingkat keefektifan adanya perangkat lunak filtering ini dapat dihitung dengan persamaan

$$TCR = \frac{WErr_B}{WErr} = \frac{N_S}{(\lambda \cdot n_{H \to S}) + n_{S \to H}} \tag{29}$$

Semakin besar nilai TCR mengindikasikan performa yang semakin baik. Nilai TCR < 1 menunjukkan bahwa kondisi yang lebih baik terjadi bila perangkat lunak filtering tidak digunakan.

BAB III

PERANCANGAN PERANGKAT LUNAK

BAB III

PERANCANGAN PERANGKAT LUNAK

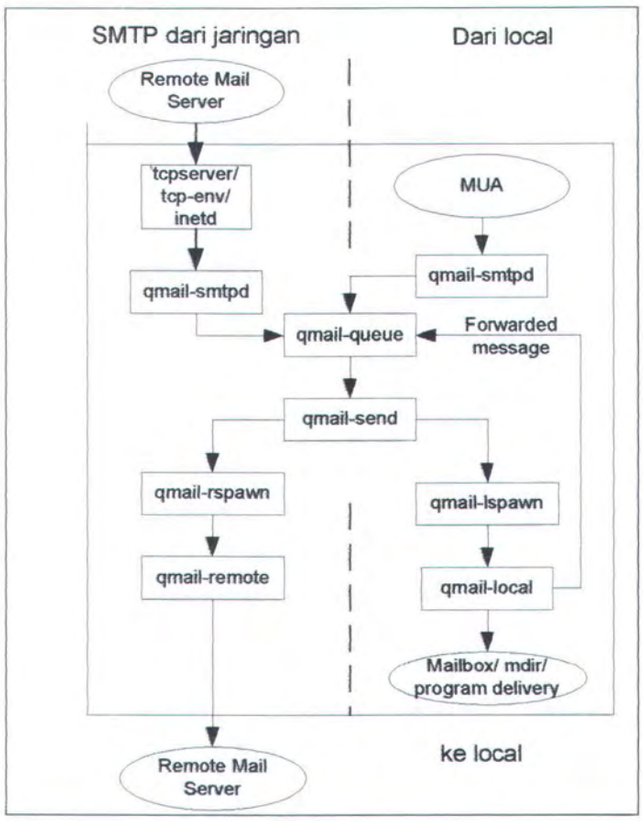
Bab ini menjelaskan tentang tahapan proses perancangan perangkat lunak yang dimulai dengan penggambaran aplikasi secara umum, maupun perancangan proses secara mendetail dari masing-masing perangkat lunak.

3.1 DESKRIPSI SISTEM

Perangkat lunak yang dibuat dalam tugas akhir ini berfungsi untuk melakukan filtering terhadap email yang akan dikirim ke local server. Proses filtering berjalan pada gateway, dalam hal ini akan digunakan qmail server sebagai *mail relay server*. *Relay server* ini berfungsi untuk mendistribusikan semua e-mail dari *remote server* ke server email lokal dan mengirimkan pesan email dari server lokal ke *remote* server. Perangkat lunak ini akan dijalankan pada file dot-qmail yang merupakan file konfigurasi pada qmail untuk mengatur instruksi pengiriman suatu email.

Dot-qmail memberikan instruksi kepada qmail-local tentang hal yang akan dilakukan terhadap suatu pesan ketika pesan tersebut diterima. Dengan dot-qmail, seorang user dapat mem-*forward* emailnya, menggunakan ezmlm mailing list, membuat alias di bawah *username*-nya sendiri, dan memanggil program eksternal seperti *autoresponders*.

Secara lengkap sistem kerja dari qmail server dapat dilihat pada Gambar 3.1.



Gambar 3.1 Sistem kerja qmail

3.2 SPESIFIKASI SISTEM

Spesifikasi perangkat lunak yang dikembangkan dalam sistem ini di jelaskan sebagai berikut :

- Pembuatan database pada mysql yang digunakan menyimpan probabilitas dari masing-masing token yang dihasilkan dari proses training.
- Pembuatan rutin program untuk melakukan email parser.
- Pembuatan aplikasi untuk proses training data.

- Pembuatan aplikasi untuk membedakan antara email spam dan bukan spam dengan menggunakan algoritma bayesian dengan menggunakan data hasil training pada mysql sebagai distribusi prior-nya.
- Konfigurasi qmail sehingga dapat berfungsi sebagai mail relay server yang hanya dapat me-relay email-email dari server email lokal.
- Pembuatan script .qmail-default pada server qmail yang didalamnya berisi instruksi untuk melakukan filtering terhadap email sebelum email tersebut didistribusikan ke server email lokal.

3.3 KEBUTUHAN SISTEM

Perangkat lunak yang dikembangkan dapat digunakan pada lingkungan sistem operasi linux, dengan berbagai macam distro. Perangkat lunak ini dikembangkan menggunakan kompiler gcc. Dibutuhkan juga mysql server, clam antivirus, libpcre, dan qmail.

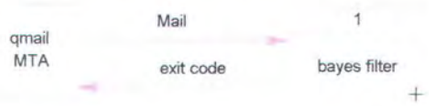
Mysql digunakan sebagai penyimpanan dari data hasil training. Libpcre dibutuhkan sebagai library tambahan pada gcc sehingga dapat menjalankan regular expression. Clam antivirus digunakan untuk men-scan email-email yang masuk dari kemungkinan mengandung virus. Dan qmail digunakan sebagai mail server yang akan menjalankan perangkat lunak yang dibangun ini.

Pada pengembangan perangkat lunak ini digunakan gcc 3.3 sebagai kompiler. Untuk melakukan kompilasi perangkat lunak ini diperlukan library tambahan, yaitu libmysql, libpcre dan libclamav.

3.4 PERANCANGAN PERANGKAT LUNAK

Pada subbab ini akan dijelaskan mengenai perancangan proses dalam aplikasi spam filtering. Untuk peranacangan proses dari perangkat lunak ini digunakan Diagram Aliran Data (DAD). Pembuatan DAD tersebut menggunakan Tool Power Designer versi 6.

DAD akan menunjukkan hubungan antara proses satu dengan yang lain serta data masukan dan data keluaran yang terlibat dalam proses–proses tersebut.

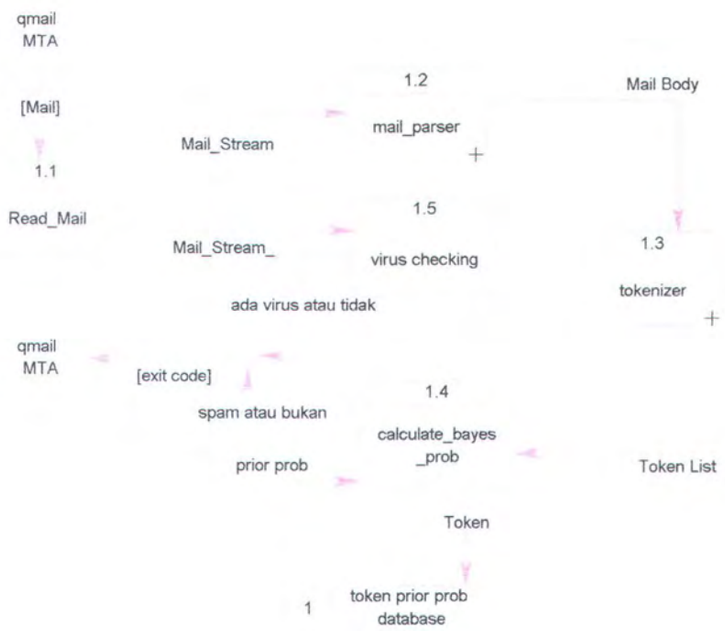


Gambar 3.2 DAD level 0 Aplikasi bayes filtering

Aplikasi bayes filter ini melakukan proses terhadap email dari MTA dan mengembalikan hasil dari prosesnya pada MTA. *Exit codes* ini untuk menunjukkan bahwa email tersebut termasuk spam ataupun bukan, dan jika email tersebut bukan spam apakah email tersebut mengandung virus atau tidak. *Exit codes* yang dikenali oleh qmail server diantaranya yaitu:

- 0, menunjukkan bahwa proses berhasil dan akan dijalankan instruksi selanjutnya pada *script*.
- 99, menunjukkan bahwa proses berhasil tetapi tidak akan menjalankan instruksi berikutnya proses berikutnya.
- 100, menunjukkan bahwa proses gagal dan akan mengirimkan pesan error kepada pengirim (*bounce message*).

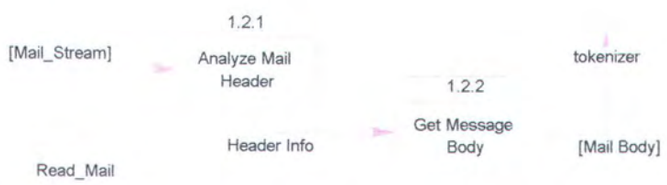
Detail dari proses filteringnya dapat dilihat pada Gambar 3.2.



Gambar 3.3 DAD level 1 proses bayes filtering

3.4.2 Email Parser

Detail proses yang dilakukan pada mail parser ini ditunjukkan pada Gambar 3.3.



Gambar 3.4 DAD level 2 proses Mail Parser

Dalam proses ini dilakukan parsing email dengan menggunakan standar MIME. Pertama yang dilakukan dalam proses ini adalah menganalisa header dari email. Dari header tersebut diperoleh informasi tentang bentuk dari emailnya. Pada proses ini dilakukan decoding terhadap isi dari email dengan menggunakan

base64, quoted printable, 7bit ataupun 8bit decoding. Jenis decoding yang digunakan bergantung pada informasi yang terdapat pada header emailnya.

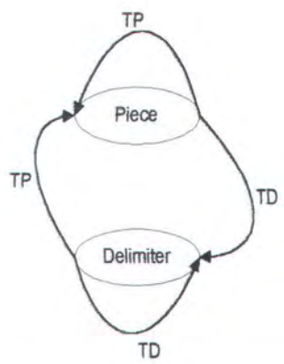
Proses ini menerima input berupa teks email. Output dari fungsi ini adalah yaitu berupa data-data tentang bagian from, subject, body dan attachment email dalam bentuk yang sudah terpisah dan sudah melalui proses decoding bilamana hal tersebut diperlukan.

3.4.3 Tokenizer

Proses ini memecah body email, subject dan bagian from dari header dalam token-token yang disimpan dalam suatu *list*. Setiap token yang sama hanya akan dicatat satu kali didalam *list*-nya. Detail proses yang dilakukan pada tokenizer ini ditunjukkan pada Gambar 3.5.

Gambar 3.5 DAD level 2 proses Tokenizer

Proses *tokenizer* ini digunakan metode *baseline*. Metode ini mempunyai 2 tipe karakter yaitu karakter sebagai *token-piece* dan *token delimiter*. *Tokenizer* membuat token dari *token-piece* hingga ditemukan *token delimiter*. Detail prosesnya ditunjukkan oleh Gambar 3.6.



Gambar 3.6 Automata dari baseline tokenizer

Token akan dimasukkan dalam list jika panjang tokennya lebih dari 3 karakter. Hal ini bertujuan untuk menghilangkan token-token yang maknanya tidak signifikan dalam suatu kata. Selanjutnya batas ukuran token yang masuk dalam list dapat diatur dalam file konfigurasinya. Karakter yang digunakan untuk memisahkan token-token (*token delimiter*) tersebut dapat diubah-ubah dalam konfigurasi program.

3.4.4 Metode Bayesian

Setelah didapatkan token-token yang sesuai dari teks email yang diperiksa, selanjutnya dilakukan proses perhitungan dengan metode bayesian untuk mendapatkan nilai probabilitas yang digunakan untuk menentukan apakah email yang diperiksa tersebut merupakan spam ataupun bukan.

3.4.4.1 Penghitungan Probabilitas Token

Diasumsikan adanya suatu body email (corpus) untuk proses training, yang telah dipisahkan secara manual antara email yang merupakan spam dan email yang bukan spam. Kemudian dengan menggunakan data email ini system

yang telah dibuat akan menggenerate probabilitas dari masing-masing kata/token yang merepresentasikan tingkat ke-spam-an dari token-token tersebut. Persamaan 19 menunjukkan cara penghitungan probabilitas dari masing-masing token tersebut.

- $b(w)$ = (jumlah dari email spam yang mengandung token w) / (jumlah dari seluruh email spam).
- $g(w)$ = (jumlah dari email ham yang mengandung token w) / (jumlah dari seluruh email ham).
- $p(w) = b(w) / (b(w) + g(w))$ (19)

p(w) dapat ditafsirkan sebagai kemungkinan secara acak e-mail yang terpilih yang berisi kata w akan merupakan suatu spam [3]. Spam filter dapat menghitung p(w) untuk tiap-tiap kata di suatu e-mail dan menggunakan informasi tersebut sebagai basis untuk kalkulasi lebih lanjut untuk menentukan apakah e-mail adalah ham atau spam.

3.4.4.2 Mengatasi Token yang Jarang Muncul

Ada suatu masalah yang dapat timbul dengan cara perhitungan kemungkinan seperti di atas manakala terdapat token yang jarang muncul. Sebagai contoh, jika suatu kata muncul hanya dalam satu e-mail, dan email tersebut merupakan suatu spam maka nilai p(w) akan menjadi 1.0. Tetapi sebenarnya masih belum tentu email berikutnya yang mengandung kata tersebut pasti adalah spam. Sebenarnya tidak dipunyai cukup data untuk mengetahui kemungkinan yang riil tersebut.

Bayesian memberikan suatu teknik untuk mengatasi masalah diatas. Cabang statistik ini didasarkan pada gagasan di mana kita tertarik akan derajat tingkat kepercayaan seseorang dalam peristiwa tertentu (kemungkinan bayesian dari suatu peristiwa) [2].

Ketika persisnya satu e-mail berisi kata tertentu dan e-mail itu adalah spam, derajat tingkat kepercayaan bahwa berikutnya kata tersebut akan muncul pada suatu email spam adalah tidak 100%. Karena adanya latar belakang informasi yang memandu. Telah diketahui dari data yang ada bahwa hampir manapun kata dapat muncul di dalam suatu spam atau konteks non-spam, dan dengan menggunakan satu data tersebut tidak cukup untuk mengetahui dengan sepenuhnya kemungkinan yang riil tersebut. Pendekatan bayesian membuat dapat dikombinasikannya latar belakang informasi yang diketahui dengan data yang telah dikumpulkan untuk suatu kata dengan sedemikian rupa, sehingga kedua aspek tersebut mempunyai makna yang penting dalam perhitungan. Dengan cara ini, ditentukan suatu derajat tingkat kepercayaan yang sesuai manakala ditemui kata tersebut akan terjadi di dalam suatu spam atau bukan. Persamaan 20 menunjukkan langkah perhitungan suatu tingkat kepercayaan (degree of belief).

$$f(w) = \frac{(s \times x) + (n \times p(w))}{s + n} \qquad (20)$$

dimana:

- s adalah *strength* yang diberikan terhadap latar belakang informasi.
- x adalah *assumed probability*, nilainya didasarkan pada latar belakang informasi secara umum mengenai suatu kata yang baru muncul pertama kali dalam suatu email.

- *n* adalah jumlah email yang telah diterima yang mengandung kata *w*.

Penggunaan s dan x pada akhirnya bertujuan untuk mengoptimasi *performance*. Nilai s dan x selanjutnya dapat diubah dari file konfigurasi perangkat lunak ini.

Dalam perhitungan selanjutnya nilai probabilitas yang digunakan adalah f(w). Bukan menggunakan p(w) yang dapat memunculkan nilai yang tidak realistis jika tidak ada data yang cukup. Persamaan f(w) ini juga dapat mengatasi bilamana tidak ada data untuk suatu kata tertentu.

Persamaan f(w) diatas disusun berdasarkan asumsi bahwa klasifikasi spam/ham untuk suatu email yang mengandung kata w merupakan suatu variabel acak *binomial* dengan *beta distribution prior* [6]. Test yang dilakukan mirip dengan eksperimen melemparkan koin beberapa kali. Ada n kali percobaan. Pada waktu pelemparan koin, masing-masing pelemparan adalah suatu percobaan, dan akan dihitung banyaknya kemunculan kepala. Tetapi dalam hal ini, percobaan adalah untuk mencari email berikutnya pada *training corpus* yang mengandung kata "free" dan melihat apakah email tersebut (yang mengandung kata "free") merupakan spam. Jika benar, maka eksperimen tersebut dianggap berhasil. Secara jelas tampak bahwa eksperimen tersebut merupakan eksperimen binomial (mempunyai dua nilai, ya dan tidak). Dan nilainya independent, jika suatu email mengandung kata free maka email berikutnya belum tentu akan mengandung kata free juga. Untuk eksperimen binomial dan diasumsikan distribusi prior-nya

merupakan distribusi beta maka untuk menghitung probabilitas tingkat keberhasilan dari n+1 kali percobaan digunakan persamaan 21.

$$E = \frac{u+q}{u+v+n} \tag{21}$$

- q adalah banyaknya percobaan yang sukses.
- n adalah banyaknya percobaan yang dilakukan.
- u dan v adalah parameter dari distribusi beta.

Dan untuk menunjukkan bahwa persamaan E adalah sama dengan f(w) maka dilakukan subtitusi, seperti ditunjukkan pada persamaan 22.

$$s = u+v\,;\quad s \times x = u\,;\quad q = n \times p(w)\,; \tag{22}$$

3.4.4.3 Mengkombinasikan Kemungkinan

Pada posisi ini, dapat dihitung suatu kemungkinan, f(w), untuk masing-masing kata yang muncul dalam suatu e-mail baru. Kemungkinan ini mencerminkan derajat tingkat kepercayaan, berdasar pada latar belakang pengetahuan dan berdasarkan data pada training corpus.

Jadi suatu email akan terdiri dari beberapa probabilitas dari masing-masing token didalamnya. Sehingga diperlukan penggabungan beberapa probabitas ini menjadi satu probabilitas saja yang merupakan indikator bahwa email tersebut merupakan spam atau bukan.

Dalam bidang statistik hal ini dikenal sebagai meta-analysis, kemungkinan kombinasi yang umum digunakan adalah sesuai dengan algoritma dari R. A. Fisher. Jika ada seperangkat kemungkinan p1, p2,..., pn. Pertama, dihitung - 2ln

p1* p2*...* pn. Kemudian, anggap hasil yang diperoleh memiliki suatu distribusi chi-square dengan 2n derajat kebebasan, dan menggunakan suatu tabel chi-square untuk menghitung kemungkinan sehingga menjadi suatu hasil yang ekstrim, dibanding hasil penghitungan sebelumnya [6].

R.A. Fisher menunjukkan bahwa nilai p yang merepresentasikan tingkat kepercayaan untuk menolak hipotesa sebelumnya dapat dihitung dengan persamaan 23.

$$H = C^{-1}(-2\ln\prod_{w} f(w), 2n) \tag{23}$$

C^{-1} adalah fungsi inverse chi-square. Parameter pertama dari fungsi ini berisi nilai chi-squarenya, dan parameter kedua berhubungan dengan nilai derajat kebebasan. Hasil dari fungsi ini sangat sensitif terhadap nilai f(w) yang mendekati 0. Yang dalam hal ini nilai 0 adalah merepresentasikan bahwa suatu email merupakan bukan spam.

Untuk meningkatkan kemampuan dalam mendeteksi email yang merupakan spam, maka dilakukan penghitungan juga terhadap nilai yang sensitif terhadap spam.

$$S = C^{-1}(-2\ln\prod_{w} 1 - f(w), 2n) \tag{24}$$

Kemudian kedua nilai probabilitas ini dikombinasikan sehingga dapat diketahui satu nilai yang merepresentasikan tingkat keyakinan dalam mentukan apakah email tersebut merupakan spam atau bukan.

$$I = (S - H + 1)/2 \tag{25}$$

Akhirnya didapatkan nilai I yang berkisar antara 0 sampai 1. Jika I mendekati 1 berarti email tersebut merupakan spam dan bila I mendekati 0 menunjukkan bahwa email tersebut bukan spam.

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Pada bab sebelumnya telah dijelaskan mengenai proses perancangan perangkat lunak dalam tugas akhir ini, bab ini akan menjelaskan proses implementasi dari perancangan yang telah dijelaskan pada bab sebelumnya.

4.1 LINGKUNGAN IMPLEMENTASI

Spesifikasi yang digunakan dalam proses pembuatan sistem filtering pada mail gateway ini adalah sebagai berikut:

1. Perangkat keras yang digunakan:
    - Pentium III 866MHz
    - SDRAM 256 MB PC 133
    - Hard Disk 20 GB 5400RPM
    - VGA AGP 8MB
2. Perangkat Lunak yang digunakan
    - Sistem Operasi Linux Mandrake 10.0
    - qmail 1.05
    - mysql 4.0.18
    - gcc 3.3.2
    - clamav 0.74

4.2 IMPLEMENTASI PERANGKAT LUNAK

Perangkat lunak yang dibuat ada 2 macam, yaitu perangkat lunak untuk melakukan training data dan perangkat lunak yang digunakan untuk melakukan deteksi terhadap spam dan virus yang akan dijalankan oleh server qmail. Kedua perangkat lunak ini mempunyai cara kerja yang sama. Hanya pada perangkat lunak untuk *training* data, penghitungan bayesian *probability* tidak dilakukan.

Pada perangkat lunak untuk training data proses akhir yang dilakukan adalah memasukkan token-token didalam list kedalam database mysql. Sedangkan pada perangkat lunak untuk deteksi, proses akhir yang dilakukan adalah mengambil data dari token-token yang sesuai dari database dan kemudian dilakukan proses penghitungan bayesian-nya.

Kedua perangkat lunak ini dibuat dengan menggunakan kompiler gcc, yang merupakan kompiler utama yang banyak digunakan didalam sistem operasi linux.

Berikut ini dijelaskan mengenai penggunaan variabel-variabel global dan fungsi-fungsi utama dalam perangkat lunak ini.

4.2.1 Variabel Global Perangkat Lunak

Pada perangkat lunak ini digunakan beberapa variabel global yang masing-masing kegunaannya akan dijelaskan sebagai berikut :

- Variabel-variabel yang digunakan untuk menyimpan rule dari beberapa regular expression yang sering digunakan dalam iterasi, diantaranya yaitu: re_dollar, re_url, re_html, re_body, re_high, re_quoted, re_titik.

- Variabel-variabel yang digunakan untuk menyimpan nilai dari file konfigurasi perangkat lunak, diantaranya yaitu: Virus_Check, Bounce_Spam, Remove_HTML_Tag, NGram_Process, Quote_Dec, URL_Dec, Assume_Prob, Strength, Ham_Limit, Spam_Limit.
- Variabel-variabel yang digunakan sebagai *pointer* didalam token *list*, diantaranya adalah: tok_first, tok_mid, tok_last, bay_first, bay_mid, bay_last.

4.2.2 Prosedur Perangkat Lunak

Prosedur yang digunakan dalam perangkat lunak ini sangat banyak, namun terbagi menjadi 5 bagian penting, yaitu:

- Prosedur yang menangani proses parsing email.
- Prosedur yang menangani proses pembentukan token.
- Prosedur yang menangani proses penghitungan probabilitas bayesian.
- Prosedur yang menangani proses penyimpanan dan pengambilan data dari database mysql.
- Prosedur yang menangani proses pendeteksian virus.

Masing-masing dari keempat bagian tersebut akan dijelaskan pada subbab berikut.

4.2.2.1 Prosedur yang Menangani Proses Parsing Email

Pada bagian ini digunakan untuk melakukan pembacaan *message body* dari email yang diperiksa. Sebelum dilakukan proses pembacaan terlebih dahulu dilakukan pemisahan antara bagian *header* dan *body* emailnya. Lalu dilakukan

analisa terhadap *header*-nya, yang kemudian informasi yang didapat digunakan untuk melakukan pembacaan bagian *body message*-nya.

Pada suatu file email, bagian header selalu terletak pada awal baris. Antara bagian *header* dan bagian isi dipisahkan oleh suatu baris kosong. Selengkapnya proses pembacaan *header* email adalah seperti ditunjukkan pada *pseudo code* Gambar 4.1.

```
msg_parse(msg)
   found ← cari(mail_subject,msg)
   If found then
      Msg_subject ← mail_subject

   Found ← cari(from_part,msg)
   If found then
      Msg_from ← from_part

   Found ← cari(content_type,msg)
   If found then
      Content ← get(content_type_value)
      If content = 'text' then
            Msg_content_type←0
            Found ← cari(content_transfer_encoding, msg)
            If found then
              Msg_encoding ← get(transfer_encoding_value)
      Else if content='multipart' then
            Msg_content_type←1
            Found ← cari(boundary,msg)
            If found then
              Msg_boundary ← get(boundary_value)
      EndIf
   EndIf

   Look_up empty_line
   Batas_header ← pos(empty_line);
   Remove_header
   Get_message_body (msg)
```

Gambar 4.1 Pseudo code fungsi message_parse

Proses pembacaan *body message* ditentukan oleh nilai yang didapat dari *content-type*–nya. Jika *content-type* berupa text berarti *message body* hanya terdiri

dari satu macam *message* saja. Jika *content-type* berisi *multipart* berarti *message body* bisa berisi lebih dari satu macam *message* yang masing-masing dari *message* tersebut dapat mempunyai bagian *header* masing-masing.

Nilai dari *content transfer encoding* juga berpengaruh dalam proses pembacaan *body message*-nya. Nilainya dapat berupa base64, quoted-printable, 7bit dan 8bit. Proses decoding yang akan dilakukan pada *body message* akan ditentukan oleh nilai dari variabel ini.

Setelah didapatkan informasi dari *header* email, maka proses pembacaan body message dapat dilakukan. Proses tersebut dapat dilihat pada *pseudo code* Gambar 4.2.

Pada email yang *content-type*-nya berupa text (content_type = 0), selanjutnya akan dilihat *content-transfer-encoding*-nya. Khusus untuk nilai encoding adalah base64, maka terlebih dahulu akan dilakukan proses decoding. setelah itu data akan dimasukkan dalam variabel msg->body.

Sedangkan untuk email yang berupa multipart proses yang terjadi jauh lebih rumit., Pesan terlebih dahulu harus dipecah menjadi bagian-bagian lebih kecil berdasarkan batas *boundary* yang ada. Lalu pada bagian-bagian tersebut akan diperiksa informasi headernya, karena pada bentuk MIME, bagian dari multipart dapat mempunyai informasi header masing-masing. Dari bagian-bagian ini yang akan dimasukkan dalam variabel msg->body adalah bagian yang merupakan text saja. Sehingga informasi lainnya seperti image, attachment, dan bentuk-bentuk yang lain akan diabaikan.

```
tokenize(isi)
  piece←0
  for x←1 to length(isi) do
   if isi[x] in token_separator then
      if length(piece) >= min_length
         AND length(piece) <= max_length then
            insert(isi, token_list)

      ▶NGramProcess bersifat optional
      if there's highbit in piece then
            NGramProcess(piece)
      if length(piece)>max_length then
            NGramProcess(piece)
      piece ← 0

      Next x
   Endif
   Piece ← isi[x];

   Next x
  Endfor
```

Gambar 4.3 Pseudo Code Fungsi Tokenize

Pada proses tokenize ini akan dicoba untuk melakukan N-Grams pada token-token tertentu. N-Grams adalah suatu proses pemecahan token dengan cara mengambil n karakter dari awal kalimat tersebut dan kemudian menggeser penunjuk posisi awal kalimat sebanyak satu karakter. Proses tersebut akan diulang sampai kalimat yang diproses panjangnya kurang dari n karakter [colloqium n-gram]. Pada aplikasi ini akan diterapkan metode 5-Grams.

Token-token yang akan diproses menggunakan metode n-grams ini adalah hanya token yang mengandung karakter highbit dan token yang panjangnya lebih dari panjang token yang telah dispesifikasikan dalam file konfigurasinya. Karena metode ngram ini memerlukan resource yang besar, maka panjang token maksimal yang dapat diproses adalah 50 karakter saja.

Token-token yang dihasilkan kemudian akan dimasukkan dalam suatu list. Dalam list tersebut jika terdapat token yang sama hanya akan dicatat satu kali saja.

4.2.2.3 Prosedur untuk Menghitung Probabilitas Bayesian

Proses ini adalah untuk menghitung kemungkinan suatu email yang diperiksa merupakan spam atau bukan. Pertama kali yang dihitung adalah nilai kemungkinan dari masing-masing token pembentuk email yang bersangkutan. Pseudo code dari fungsi ini ditunjukkan pada Gambar 4.4.

```
bayes_prob(tot_ham, tot_spam)
  bw ← jum_spam/tot_spam
  gw ← jum_ham/tot_ham
  pw ← bw/(bw+gw)
  if jum_ham=0 AND jum_spam=0 then pw ← 0
  gw ← jum_ham +jum_spam
  prob ← ((Strength * Assume_Prob)+(gw * pw)) /
         (Strength+gw)
```

Gambar 4.4 Pseudo Code Fungsi Bayes_Prob

Dari nilai probabilitas per token ini kemudian digabung menjadi satu nilai probabilitas. Proses penghitungannya seperti ditunjukkan pada Gambar 4.5.

Karena probabilitas per token yang telah didapat sebelumnya adalah untuk merepresentasikan kemungkinan suatu token terdapat pada email yang bukan spam. Maka pada pseudo code diatas, nilai S, probabilitas untuk suatu token terdapat pada spam, didapatkan dengan mengurangkan 1 dengan probabilitas tokennya.

```
bayes_combine(token data)
  While(data) do
    S ← (1.0 - prob) * S
    H ← prob * H
    Next_data
  EndWhile
  S ← 1.0 - chi2Q(-2.0 * ln(S), 2*count(data));
  H ← 1.0 - chi2Q(-2.0 * ln(H), 2*count(data));
  Prob ← (S-H+1.0)/2.0
```

Gambar 4.5 Pseudo Code Fungsi Bayes_Combine

4.2.2.4 Prosedur untuk menyimpan dan mengambil data dari database

Perangkat lunak ini memerlukan database untuk menyimpan data-data token hasil training. Database yang digunakan adalah mysql. Untuk keperluan penyimpanan ini, diperlukan dua buah tabel, yaitu:

1. Tabel jumlah

   Tabel ini digunakan untuk menyimpan jumlah keseluruhan email yang digunakan untuk training. Dalam tabel ini terdapat dua field yaitu ham dan spam, yang masing-masing digunakan untuk menyimpan informasi jumlah non-spam email dan spam email yang pernah dilakukan proses training. Tabel ini hanya berisi satu baris record saja.

2. Tabel Probability

   Tabel ini digunakan untuk meyimpan informasi mengenai token-token yang didapatkan dari hasil training. Dalam tabel ini terdapat tiga field yaitu field token sebagai primary key, field ham dan field spam. Field ham berisikan jumlah email non-spam yang digunakan dalam training yang mengandung token tersebut. Dan field spam berisi jumlah email spam yang mengandung token tersebut.

Dengan digunakannya mysql sebagai media penyimpanan, maka aplikasi ini membutuhkan fungsi dan prosedur untuk melakukan akses pada database mysql.

Untuk penyimpanan data, token digunakan sebagai *primary key*. Sehingga tidak boleh terdapat token yang sama lebih dari satu. Pseudo code prosedur untuk proses *update* dan *insert* data seperti ditunjukkan pada Gambar 4.6.

```
db_insert(the_token, jum_ham, jum_spam)
  If token exist in database then
     if token=the_token
           ▶update tabel
           ham←jum_ham
           spam←jum_spam
     Endif
  else
     ▶insert new_record
           token←the_token
           ham←jum_ham
           spam←jum_spam
  endif
```

Gambar 4.6 Pseudo Code Fungsi db_Insert

Pada proses insert data, terlihat bahwa terdapat dua proses, yaitu proses update dan proses insert. Hal ini dilakukan untuk menjaga bahwa sebuah token yang sama hanya akan dicatat satu kali saja dalam database.

Untuk proses pembacaan data dari database adalah seperti ditunjukkan pada gambar berikut.

```
db_look_prob(the_token)
  row←query data probability sesuai the_token
  if success then
     ham_prob←row[ham]
     spam_prob←row[spam]
     token←row[token]
  Endif
```

Gambar 4.7 Pseudo Code Fungsi db_Look_Prob

Fungsi db_look_prob menerima inputan berupa tokennya. Dan akan menghasilkan output berupa suatu struktur data yang berisi token, *ham probability* dan spam *probability.*

4.2.2.5 Prosedur untuk Pendeteksian Virus

Pada perangkat lunak ini, untuk melakukan pendeteksian terhadap virus digunakan library tambahan dari Clam Anti Virus. Library tersebut menjadikan perangkat lunak yang dibangun dapat memanfaatkan *virus definition* dari Clam AV. Berikut adalah pseudo code dari proses deteksi virus.

```
scan_virus(nmfile)
  File_desc←openfile(nmfile);
  Root←Read_virus_definition;
  ret ← cl_scandesc(fd, &virname, &size, root)
  if ret = CL_VIRUS then
       ►virus_detected
  else
       ►no_virus_detected
```

Gambar 4.8 Pseudo Code Fungsi Scan_Virus

Pada library versi terbaru, clam AV sudah mendukung proses scanning pada file email. Sehingga untuk scanning pada bagian attachment suatu email dapat langsung dilakukan tanpa memerlukan proses ekstraksi dan decoding terlebih dahulu.

4.2.3 Prosedur Tambahan untuk Optimasi Hasil Filtering

Dari fungsi-fungsi yang telah disebutkan diatas, perangkat lunak ini sebenarnya sudah dapat berjalan. Fungsi dan prosedur tambahan ini dikembangkan setelah melihat token-token yang dihasilkan dari perangkat lunak

ini terlihat mempunyai karakteristik-karakteristik tertentu. Beberapa fungsi dan prosedur tambahan yang dikembangkan diantaranya adalah:

- Prosedur untuk Menghapus *Tag* HTML.
- Prosedur untuk mengganti karakter $ atau ! yang muncul secara berurutan.
- Prosedur untuk mengganti karakter angka yang merepresentasikan nilai uang dengan suatu token yang spesifik.
- Prosedur untuk melakukan proses 5-grams pada token yang mengandung karakter high bit dan pada token yang panjangnya lebih dari batas maksimal panjang token yang diterima.

Prosedur dan fungsi ini dijalankan setelah didapatkan data mengenai body, pengirim, dan subject sebuah email. Dan sebelum data yang diperoleh tersebut masuk dalam proses tokenizer. Sehingga proses-proses ini berjalan diantara fungsi message_parse() dan fungsi tokenizer().

Prosedur untuk menghilangkan *Tag* HTML dikembangkan setelah melihat banyaknya token-token yang tidak masuk kedalam database ataupun token dapat masuk tetapi dengan bentuk yang berbeda karena adanya Tag HTML ini.

Misalnya didalam email terdapat kata "free", tetapi oleh pengirim email huruf 'f' dan 'e' –nya dicetak tebal. Sehingga pada text email dengan HTML *content* sebenarnya kata free tertulis dengan "free" yang jika kata ini ditampilkan pada mail client akan terbaca sebagai "**free**". Dengan adanya prosedur ini diharapkan kata-kata yang ditulis dengan cara seperti diatas dapat tetap dikenali oleh proses tokenizer sebagaimana kata yang sebenarnya.

Disamping itu karena hampir semua email spam mengandung suatu tag HTML, sehingga nilai kemungkinan suatu email menjadi spam akan sangat tinggi pada email yang mengandung tag HTML. Padahal banyak juga email bukan spam yang mengandung tag HTML. Sehingga kemungkinan akan terjadi banyak kesalahan mendeteksi bukan spam sebagai spam. Tetapi dalam proses ini, dilakukan juga proses untuk mengambil informasi mengenai warna dan alamat url yang terdapat didalam tag html yang akan dihilangkan. Pseudo code dari fungsi ini dapat dilihat pada Gambar 4.9

```
remove_tag(isi)
  While (length(isi)>0 AND
         hasil←cari pola_HTML_dalam_isi)
  do
     warna←cari color_pola dalam hasil
     URL←cari url_pola dalam hasil
     ▶hapus hasil from isi
     isi[length(isi)]←warna
     isi[length(isi)]←URL
  EndWhile
```

Gambar 4.9 Pseudo code fungsi memindahkan tag HTML

Prosedur untuk menjadikan $ ataupun ! yang muncul secara didasarkan pada asumsi bahwa pada email-email spam banyak didapati penulisan karakter $ ataupun ! dalam jumlah banyak dan muncul secara berurutan. Pada masing-masing email jumlah karakter yang muncul adalah tidak sama dan karakter $! digunakan sebagai token separator. Sehingga untuk mengoptimalkan hasil proses filtering akan dicoba mengganti karakter $ atau ! yang panjangnya lebih dari 5 karakter dengan sebuah token yang spesifik misalkan diganti dengan token 'XXSPAMXX'.

Karakter angka (0-9) dapat digunakan sebagai token *separator*. Tetapi karakter ini dapat mempunyai suatu karakteristik tertentu yang dapat digunakan untuk membedakan email spam ataupun bukan, misal angka-angka yang merepresentasikan jumlah uang banyak didapati pada email-email spam. Sehingga ketika angka digunakan dalam token separator, diperlukan suatu fungsi untuk menggantikan karakter angka yang merepresentasikan suatu nilai mata uang dengan token yang spesifik. Dalam hal ini akan digunakan token 'XXUANGXX'.

Perbandingan hasil filtering dari digunakannya proses-proses tambahan dengan tidak menggunakan tambahan ini dapat dilihat pada bab 5, uji coba dan evaluasi.

4.3 IMPLEMENTASI MAIL GATEWAY

Gateway adalah suatu mesin yang bertugas mengirimkan email yang datang ke *mailhost* internal yang relevan. Mesin ini menerima email dari *mailhost* dan dari sumber eksternal. Dan mesin ini mengirimkan keluar email dari mailhost *internal* ke MX host yang relevan. Untuk menjalankan fungsi ini diperlukan suatu *name server* yang menyediakan daftar mailhost internal dan MX host yang ada.

Dalam subbab ini akan dijelaskan mengenai setting dari qmail server sehingga dapat berfungsi sebagai mail gateway serta setting dari qmail sehingga dapat menjalankan perangkat lunak yang telah dibuat dalam Tugas Akhir ini.

Setting yang akan dijabarkan pada subbab merupakan contoh setting jika instalasi server qmail-nya mengikuti prosedur instalasi standar sebagaimana yang dijelaskan dalam "Life with qmail" [5].

4.3.1 Qmail sebagai Mail Gateway

Untuk menjadikan qmail menjadi suatu *mail gateway*, beberapa langkah yang harus dilakukan antara lain adalah sebagai berikut:

- Mengubah rcpthost.
- Membuat konfigurasi untuk *routing* SMTP.
- Mengubah file startup qmail.

Rcpthost berisi daftar nama domain yang diterima oleh qmail server tersebut. Misalkan SMTP menerima suatu pesan dari host lain:

```
MAIL FROM: <bill@irs.gov>
RCPT TO: joe@heaven.af.mil
```

Maka sebelum pesan tersebut diterima oleh qmail, terlebih dahulu akan dilakukan pengecekan terhadap rcpt-nya. Sehingga jika heaven.af.mil terdapat pada rcpthost maka pesan tersebut baru akan diterima.

Untuk menjadikan qmail sebagai mail gateway maka setting rcphost dilakukan, yaitu dengan mengubah file /var/qmail/control/rcpthost dan /var/qmail/control/locals menjadi berisi :

```
gateway.example.com
example.com
```

Juga dilakukan perubahan pada file /var/qmail/control/smtproutes dengan:

```
devel.example.com:mailhost.devel.example.com
sales.example.com:mailhost.sales.example.com
```

setelah itu perlu dibuat startup file untuk qmail-nya yaitu pada /service/qmail/run, berisi:

```
exec tcpserver -u 29991 -g 29998 -c 100 -v -R \
-x /var/qmail/etc/smtpd/rules.cdb \
0 smtp /var/qmail/bin/qmail-smtpd 2>&1
```

Setelah dilakukan konfigurasi untuk mail gateway, perlu juga dilakukan konfigurasi pada *mailhost*-nya. *Mailhost* merupakan suatu mesin yang akan melakukan proses pen-*delivery*-an local suatu pesan pada site tertentu. Mailhost menerima pesan dari null client dan sumber eksternal. Dan mengirimkan semua email yang keluar menuju ke suatu mail gateway.

Contoh setting untuk mailhost ini adalah seperti ditunjukkan pada Gambar 4.10.

```
/var/qmail/control/rcpthosts dan /var/qmail/control/locals
            mailhost.devel.example.com
            devel.example.com

/var/qmail/control/smtproutes
            :gateway.example.com

/service/qmail/run
            exec tcpserver -u 29991 -g 29998 -c 100 -v -R \
            -x /var/qmail/etc/smtpd/rules.cdb \
            0 smtp /var/qmail/bin/qmail-smtpd 2>&1
```

Gambar 4.10 Setting untuk mailhost

4.3.2 Integrasi Perangkat Lunak dengan qmail Gateway

Perangkat lunak yang dikembangkan dalam Tugas Akhir ini adalah suatu aplikasi yang dapat dijalankan dari command line. Sebelum aplikasi digunakan terlebih dahulu harus dilakukan training pada data email yang telah dipilih secara manual. Data yang dipakai untuk proses training akan sangat berpengaruh pada kualitas filter nantinya.

Proses training dapat dijalankan dengan perintah "bayes_training". Perintah ini dapat menerima tiga macam argument. Yaitu –r, digunakan untuk mengosongkan database. Argument –s digunakan untuk menunjukkan bahwa

direktori tersebut berisi koleksi email spam yang akan ditraining. Dan argumen –h digunakan untuk menunjukkan bahwa direktori yang mengikuti argumen ini merupakan direktori yang berisi email-email bukan spam yang akan ditraining. Gambar 4.11 menunjukkan tampilan dari proses training.

```
[root@dns /]bayes_training -s/mnt/win_d/corpus/spam/1
.
.
.
processing file 1196:
/mnt/win_d/corpus/spam/1/0089.51c746428bb5e2793a1c04ce1e0c72
c1 (2.86 KB)
processing file 1197:
/mnt/win_d/corpus/spam/1/0099.c4ff6dba0a5177d3c7d8ef54c89204
96 (6.56 KB)
                - insert to database
Training Completed
Adding ham: 0
Adding spam: 1197
```

Gambar 4.11 Tampilan Proses Training Data

Aplikasi ini memerlukan sebuah argument yang berisi nama file yang akan diperiksa.

```
[root@dns /]# bayes_cekmail /home/user/Maildir/cur/001
```

Gambar 4.12 Contoh cara menjalankan aplikasi dari shell

Pada mail gateway, diharapkan aplikasi ini dapat secara otomatis langsung dijalankan ketika ada suatu pesan yang masuk. Untuk itu perlu dilakukan konfigurasi pada .qmail-default –nya. Dalam prosedur instalasi standar file ini terletak pada ~alias/.qmail-default. Untuk ini dilakukan perubahan, sehingga sebelum pesan dikirimkan ke direktori user, pesan akan diperiksa terlebih dahulu oleh aplikasi bayes_cekmail ini.

```
| cat - > /temp/msg
| Bayes_cekmail /temp/msg
| cat /temp/msg | /var/qmail/bin/qmail-remote
mail.informatika.com "$USER" $DEFAULT@mhs.informatika.com
| rm -f /temp/msg
```

Gambar 4.13 Contoh script .qmail pada mail gateway

Pesan diterima dalam bentuk stream, sehingga perlu dibuat file temporary untuk menyimpan stream tersebut. Hal tersebut dapat dilakukan dengan memanggil perintah cat - > /temp/msg. Stream akan disimpan dalam file yang bernama /temp/msg. Pada saat proses pengiriman selesai maka file temporary tersebut harus dihapus. Hal ini bertujuan untuk menjaga privasi dari pesan-pesan user yang masuk. Gambar 4.14 menunjukkan status pada log server qmail ketika aplikasi ini dijalankan.

```
@4000000040ffdb0724db43bc status: local 0/20 remote 0/20
@4000000040ffdbce15a3351c new msg 125845
@4000000040ffdbce15a35074 info msg 125845: bytes 58510 from
<bayu@daytona.informatika.com> qp 7205 uid 401
@4000000040ffdbce1686f5cc starting delivery 1: msg 125845 to
local maman@mhs.informatika.com
@4000000040ffdbce16882294 status: local 1/20 remote 0/20
@4000000040ffdc7204746374 delivery 1: success:
Running_bayes_cekmail/Spam_Message_detected/Virus_detected:_
Worm.SomeFool.P/sending_to_qmail-remote/r
@4000000040ffdc7204748a84 status: local 0/20 remote 0/20
@4000000040ffdc7204749a24 end msg 125845
```

Gambar 4.14 File log pada qmail server setelah aplikasi diintegrasikan

BAB V

UJI COBA DAN EVALUASI PERANGKAT LUNAK

BAB V

UJI COBA DAN EVALUASI PERANGKAT LUNAK

Dalam bab ini akan dibahas mengenai lingkungan uji coba yang akan digunakan untuk menjalankan perangkat lunak. Kemudian akan dijelaskan mengenai data yang digunakan untuk keperluan uji coba. Dan terakhir akan dibahas mengenai cara-cara pengujian dan hasil uji coba serta evaluasi dari perangkat lunak yang telah dirancang dan dibuat.

5.1 LINGKUNGAN UJI COBA

Perangkat keras yang digunakan dalam proses uji coba ini adalah satu buah PC yang akan dijadikan sebagai mail gateway dengan spesifikasi sebagai berikut:

- Processor Intel Pentium III 866MHZ
- SDRAM 256MB PC 133
- HDD 20GB ATA 100 5400RPM
- NIC 10/100Mbps

Pada PC ini digunakan Linux Mandrake 10.0 sebagai sistem operasinya, dan menggunakan qmail 1.03 sebagai mail gateway.

5.2 UJI COBA DAN EVALUASI

Perangkat lunak yang dibangun dalam Tugas Akhir ini mempunyai suatu file konfigurasi yang dapat diubah sehingga didapatkan hasil yang lebih optimal.

File ini tersimpan pada mesin dimana perangkat lunak ini berjalan dengan nama file /etc/bayespam.conf.

Yang diamati dalam proses uji coba ini adalah jumlah kesalahan dalam pendeteksian spam yang dilakukan dan juga waktu yang diperlukan dalam proses deteksi itu sendiri.

Dalam uji coba ini akan digunakan SpamAssasin public corpus [4]. Corpus ini terbagi dalam beberapa file dengan karakteristik yang berbeda (seperti hard ham, easy ham, hard spam, easy spam). Karena perbedaan karakteristik yang ada ini tidak diperlukan, maka file hanya akan dibagi menjadi dua folder yaitu spam dan ham. SpamAssasin public corpus yang digunakan dalam uji coba ini berisi 8735 email yang terdiri dari 2398 email spam dan 6337 email bukan spam.

Yang diamati dalam proses uji coba adalah:

- Hasil yang diperoleh ketika setting untuk GroupMoneyVal, GroupDolar, GroupExclam, RemoveHTMLTag dan 5grams Processing diaktifkan.
- Perubahan nilai parameter strength

Pada setiap perubahan yang dilakukan pada nilai parameter tersebut, akan dilakukan *two-fold cross validation*. Selanjutnya hasil yang digunakan adalah nilai rata-rata dari masing-masing proses validasi silang tersebut. Hal ini dilakukan untuk mendapatkan hasil uji coba yang akurat.

Dalam perhitungan, nilai probabilitas yang terjadi selalu berkisar antara 0 sampai dengan 1 ($P \mid 0 \leq P \leq 1, P \in \mathrm{Re}al$). Jika nilai mendekati 0 berarti email bukan merupakan spam dan sebaliknya bila mendekati 1 berarti email tersebut

adalah spam. Spam limit merupakan batas nilai minimal untuk probabilitas akhir sehingga suatu email dianggap spam.

```
//Main Program Setting
AllowSpam=yes
VirusChecking=yes
AllowInfected=yes
BounceSpam=yes

//Tokenizer setting
QuotePrintableDecode=yes
URLDecode=yes
TokenizeFrom=yes
TokenizeSubject=yes
TokenizeBody=yes
MinLength=3
MaxLength=30
ExtTokenSeparator=[]{}()<>?"`#^&*;+|\:/=@

//modifikasi token ==> 4 testing purposes
GroupMoneyVal=no
GroupDolar=no
GroupExclam=no
5gramsProcessing=no

RemoveHTMLTag=yes
GetURL=yes
GetFontColor=yes

//mysql setting
mysql_hostname=localhost
mysql_username=root
mysql_password=
mysql_dbname=bayestemp

//Bayes setting
AssumeProb=0.5
Strength=1
ham_limit=0.45
spam_limit=0.55
```

Gambar 5.1 File Konfigurasi Perangkat Lunak (/etc/bayespam.conf)

5.2.1 Uji Coba dengan Perubahan Setting untuk Modifikasi Token

Gambar 5.1 menunjukkan file konfigurasi pada aplikasi spam filtering yang telah dibuat. Pada uji coba ini dilakukan perubahan-perubahan parameter konfigurasi yang berkaitan dengan tokenizer, yaitu :

- GroupMoneyVal
- GroupDolar
- GroupExclam
- 5gramsProcessing
- RemoveHTMLTag
- GetURL
- GetFontColor

Kondisi standar sebagai pembanding adalah dengan tidak melakukan proses-proses, diatas. Sehingga dengan uji coba ini dapat diketahui parameter-parameter yang dapat menghasilkan kondisi yang lebih optimal. Sebagian hasil dari uji coba ini dapat dilihat pada Tabel 5.1. Hasil uji coba secara lengkap untuk masing-masing parameternya dapat dilihat pada lampiran A.

Dari uji coba terlihat bahwa dengan melakukan n-gram processing, error rate menjadi semakin kecil. Tetapi waktu proses yang dibutuhkan oleh metode n-grams ini adalah yang paling lama dibandingkan dengan uji coba dengan parameter yang lain. Hal ini beralasan karena dengan melakukan n-gram proses, token yang dihasilkan akan semakin banyak, sehingga banyak memakai resource dari system.

Tabel 5.1 Hasil Uji Coba dengan Perubahan Parameter Token

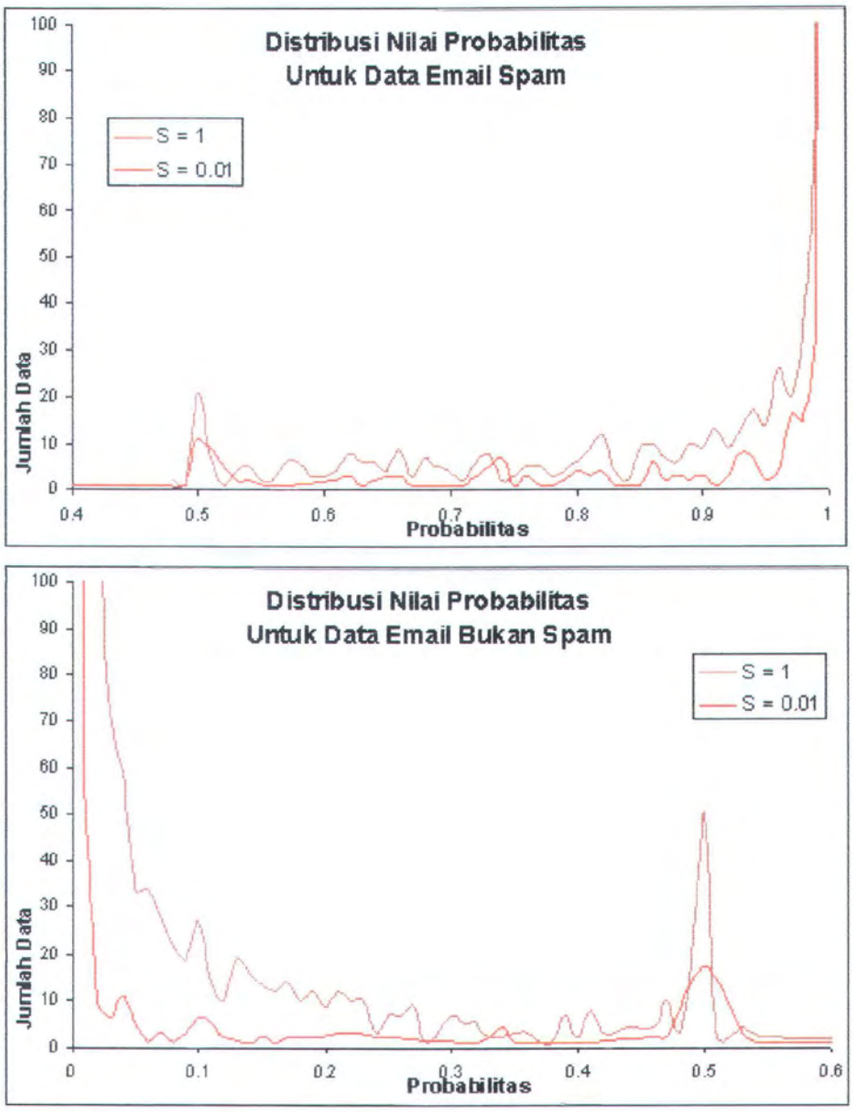
Konfigurasi	Spam Limit	Error Rate	TCR $\lambda = 100$	Waktu Proses (h:mm:ss)
Standar	0.55 0.80	1.2709% 3.5207%	1.3367 5.8991	47:32.7
n-gram process	0.55 0.80	1.2537% 3.7497%	2.8146 4.5633	54:47.4

Tag HTML dihapus	0.55 0.70	1.9235% 4.0875%	2.2644 6.7171	47:27.1
Ambil URL dan Warna pada Tag HTML	0.55 0.70	1.76% 3.78%	2.4101 7.2667	0:48:37
Pengelompokan $ dan !	0.55 0.80	1.2652% 3.5264%	1.3371 5.8919	45:06.9
Perubahan token yang berupa nilai uang	0.55 0.80	1.2709% 3.6352%	1.3008 5.7575	47:30.8

Dari keseluruhan uji coba yang menghasilkan nilai TCR paling tinggi adalah ketika tag HTML dihilangkan. Meskipun jumlah error rate meningkat, tetapi pada waktu tag HTML ini dihilangkan, banyak data-data bukan spam yang pada uji coba yang lain dianggap spam, pada uji coba dengan menghilangkan tag HTML ini jumlahnya berkurang. Tetapi jumlah email spam yang dianggap bukan spam, pada waktu tag HTML ini dihilangkan, jumlahnya meningkat. Karena nilai false positive yang rendah perubahan parameter ini menghasilkan nilai TCR yang tinggi.

5.2.2 Uji Coba dengan Perubahan Nilai Strength

Perubahan nilai strength dapat dilakukan pada kofigurasi aplikasi ini. Didalam perhitungan nilai *strength* dan asumsi nilai probabilitas awal (*assume prob*ability) merupakan parameter untuk distribusi beta. Nilai *strength* menggambarkan besarnya tingkat kepercayaan terhadap asumsi probabilitas awal yang ada. Sehingga semakin tinggi nilai *strength*-nya, maka hasil perhitungan yang didapat distribusinya akan semakin mendekati nilai asumsi awalnya dan sebaliknya. Semakin kecil nilai strength maka nilai probabilitas yang dihasilkan akan semakin mendekati 1 untuk data email spam, ataupun semakin mendekati 0 untuk data email bukan spam (Gambar 5.2).



Gambar 5.2 Distribusi nilai probabilitas untuk S = 1 dan S = 0.01

Uji coba ini dilakukan untuk mengetahui besar dari nilai *strength*, sehingga dapat menghasilkan klasifikasi yang paling optimal (dengan *error rate* yang rendah ataupun *cost ratio* yang tinggi).

Tabel 5.2 Hasil uji coba dengan perubahan nilai strength

No.	Nilai Strength	Spam Limit	Error Rate	TCR
1	1	0.55	1.22%	2.4040
2	1	0.8	3.72%	4.5851
3	0.5	0.55	1.09%	2.4040
4	0.5	0.85	3.50%	4.3363
5	0.1	0.55	0.98%	1.5766
6	0.1	0.9	2.74%	5.4811
7	0.05	0.55	1.00%	1.4784
8	0.05	0.99	4.30%	4.1813
9	0.01	0.55	1.04%	1.6804
10	0.01	0.99	3.66%	4.6293

Pada Tabel 5.2 terlihat bahwa hasil klasifikasi yang paling optimal terjadi pada nilai strength 0.1. Pada nilai *strength* ini tingkat keakuratan klasifikasi bisa mencapai lebih dari 99%.

5.2.3 Uji Coba untuk Mengetahui Kecepatan Proses Klasifikasi

Kecepatan proses merupakan suatu hal yang sangat penting dalam aplikasi ini, karena aplikasi ini akan berjalan pada suatu server email. Supaya dengean adanya aplikasi email ini tidak mengganggu jalannya proses pengiriman email, proses klasifikasi harus dapat berjalan dalam waktu yang sangat singkat.

Pengamatan terhadap waktu dilakukan pada saat proses training data dan pada saat klasifikasi data. Dan akan diamati pula perbandingan *queue* time (waktu antrian) pada server email antara menggunakan tambahan aplikasi yang telah dibuat dan tanpa menggunakan aplikasi filtering (kondisi qmail standar).

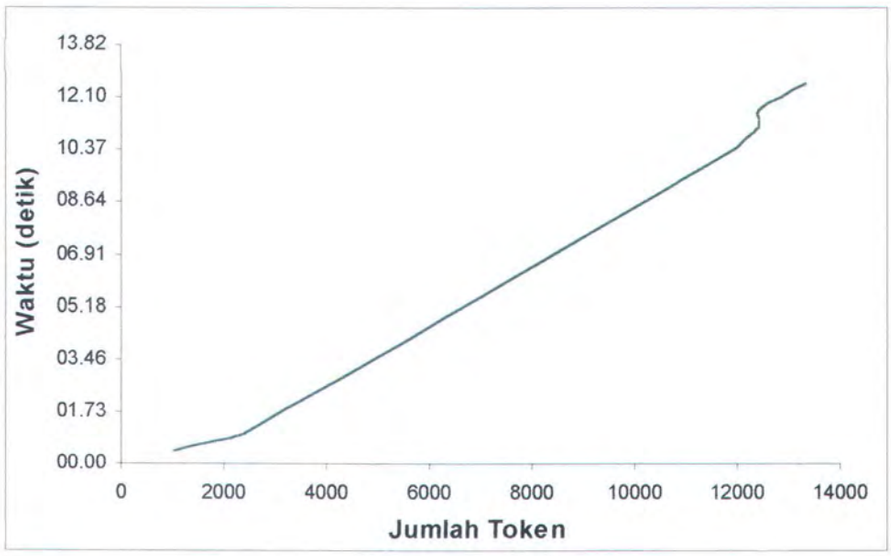
Tabel 5.3 Waktu yang diperlukan pada proses training

No	Jumlah Data	Ukuran Data (KB)	Jumlah Token yang dihasilkan	Waktu (detik)
Tanpa Proses N-Grams				
1	60	49.5	1030	00.42
2	60	135	2250	00.91
3	1300	2679.25	12471	11.74
4	1300	2240.21	11749	10.22
Dengan Proses N-Grams				
1	60	49.5	1058	00.43
2	60	135	2380	01.00
3	1300	2679.25	13327	12.51
4	1300	2240.21	12396	11.03

Pada Tabel 5.3 terlihat bahwa waktu yang diperlukan dalam proses training ini dipengaruhi oleh banyak token yang dihasilkan pada proses trainingnya. Karena semakin banyak jumlah token yang dihasilkan berarti proses untuk akses ke database mysql akan semakin sering. Proses akses database ini berkaitan dengan proses baca tulis ke disk yang membutuhkan waktu. Sehingga semakin sering terjadi proses akses database (semakin banyak token yang dihasilkan) waktu yang diperlukan dalam proses trainingnya akan semakin lama (Gambar 5.3).

Pengamatan terhadap waktu juga dilakukan pada saat uji coba klasifikasi data email. Dari hasil uji coba diketahui bahwa waktu yang diperlukan untuk melakukan klasifikasi untuk setiap email rata-rata kurang dari 1 detik. Bahkan bila proses untuk deteksi virus tidak dilakukan, waktu yang diperlukan untuk

mengklasifikasikan suatu email rata-rata hanya 0.05 detik saja (Tabel 5.4). Proses klasifikasi yang dapat berjalan dalam waktu yang sangat singkat ini membuat aplikasi layak untuk dijalankan pada suatu server email.



Gambar 5.3 Grafik perbandingan jumlah token dengan waktu proses

Tabel 5.4 Waktu yang diperlukan dalam proses klasifikasi email

No	Jumlah Data	Ukuran Data (KB)	Waktu (menit)
Tanpa Proses N-Grams			
1	60	49.5	0:01.29
2	60	135	0:02.07
3	1300	2240.21	0:35.67
4	1300	2679.25	0:38.04
5	1197	7997.63	1:45.80
Dengan Proses N-Grams			
1	60	49.5	0:01.32
2	60	135	0:02.11
3	1300	2240.21	0:36.26
4	1300	2679.25	0:38.55
5	1197	7997.63	1:56.19
Dengan Virus Check			
1	60	49.5	0:52.80
2	60	135	0:53.60
3	1300	2240.21	15:32.20

4	1300	2679.25	15:28.34
5	1197	7997.63	16:49.48
rata-rata per email:			
Ukuran Email		3.34	KB
Tanpa N-Gram		0.047	detik
Dengan N-Gram		0.050	detik
Dengan Virus Check		0.760	detik

Berikutnya uji coba dilakukan dengan mengintegrasikan aplikasi ini pada qmail. Sehingga secara otomatis aplikasi ini akan dijalankan sebelum qmail melakukan proses pengiriman menuju ke user ataupun mailhost tertentu. Untuk melihat waktu antrian (*queue time*) digunakan *tool qmailanalog*. *qmailanalog* ini dikembangkan untuk mengolah data pada log qmail sehingga menghasilkan informasi-informasi tertentu, misalnya jumlah email yang diterima, queue time yang dibutuhkan, delay pengiriman yang terjadi dan sebagainya. Untuk melakukan pengolahan log qmail digunakan *tool zsuccesses* dan *zoverall* dari *qmailanalog*.

```
[root@ns bin]# cat qmail.log | ./zsuccesses
Reasons for success
One line per reason for successful delivery.
              Information on each line:
* del is the number of deliveries that ended for
              this reason.
* xdelay is the total xdelay on those deliveries.
del  xdelay  reason
  2    0.00  start Process/Bayespam not sure bout
              this message/No virus detected./r
 14    0.00  start Process/Good Message detected/No
              virus detected./r
 11    0.00  start Process/Spam Message detected/No
              virus detected./r
```

Gambar 5.4 Pengolahan log qmail menggunakan zsuccesses

```
[root@ns bin]# cat qmail.log | ./zoverall
Basic statistics

qtime is the time spent by a message in the queue.

ddelay is the latency for a successful delivery to one
recipient---the
end of successful delivery, minus the time when the message
was queued.

xdelay is the latency for a delivery attempt---the time when
the attempt
finished, minus the time when it started. The average
concurrency is the
total xdelay for all deliveries divided by the time span;
this is a good
measure of how busy the mailer is.

Completed messages: 45
Recipients for completed messages: 45
Total delivery attempts for completed messages: 45
Average delivery attempts per completed message: 1
Bytes in completed messages: 17333496
Bytes weighted by success: 17333496
Average message qtime (s): 0

Total delivery attempts: 45
  success: 45
  failure: 0
  deferral: 0
Total ddelay (s): 0.000000
Average ddelay per success (s): 0.000000
Total xdelay (s): 0.000000
Average xdelay per delivery attempt (s): 0.000000
Time span (days): 0
```

Gambar 5.5 Pengolahan log qmail menggunakan zoverall

BAB VI

PENUTUP

BAB VI

PENUTUP

6.1 SIMPULAN

Dalam Tugas Akhir ini, telah dilakukan evaluasi terhadap proses perancangan dan pengembangan sistem. Berdasarkan evaluasi tersebut maka dapat diambil beberapa simpulan, yaitu :

1. Perangkat lunak dapat mengimplementasikan perhitungan probabilitas bayesian yang digunakan untuk mengetahui apakah suatu email tersebut tergolong spam atau bukan.
2. Perangkat lunak mampu melakukan proses filtering dengan prosentase kesalahan yang kecil yaitu berkisar 1% - 2% dalam waktu yang singkat.
3. Token yang terbentuk dari suatu tag HTML banyak menimbulkan kesalahan klasifikasi terhadap email-email bukan spam. Penghilangan tag HTML pada body email membuat hasil klasifikasi menjadi lebih optimal.
4. Aplikasi dapat berjalan secara otomatis karena qmail mempunyai fasilitas untuk hal ini.

6.2 SARAN

Berikut ini beberapa saran pengembangan yang mungkin dilakukan :

1. Tokenizer yang ada pada perangkat lunak ini dapat dikembangkan menjadi berbasis frase, bukan berbasis kata. Sehingga mampu meningkatkan ketepatan pendeteksian spam.

2. Dapat dilakukan penambahan metode whitelist dan blacklist pada email-email yang akan masuk. Sehingga tidak semua email yang masuk harus di-*scan* terlebih dahulu oleh perangkat lunak ini. Hal ini dapat mengurangi beban kerja dari mail server, pada traffic email yang tinggi.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

1 Beta Distribution. http://wolfram.mathworld.com

2 Box, G. E. P. dan Tiao, G. C. 1972, Bayesian Inference in Statistical Analysis. Department of Statistics University of Wisconsin.

3 Graham, Paul. 2003, A Plan for Spam. http://www.paulgraham.com/spam.html

4 http://spamassassin.org/publiccorpus/

5 Life With qmail. http://www.lifewithqmail.org/lwq.html

6 Robinson, Gary. 2004, May. Motivations for the Use of Fisher's Inverse Chi-Square Procedure in Spam Classification.

7 Tauritz, Daniel. Applications of n-grams. Department of Computer Science University of Missouri-Rolla

LAMPIRAN

LAMPIRAN A

UJI COBA PERANGKAT LUNAK

Berikut ini adalah hasil uji coba secara lengkap yang telah dilakukan. Uji coba dilakukan dengan konfigurasi nilai Strength = 1. TCR dihitung dengan nilai $\lambda = 100$.

No.	Spam Limit	False Positive	False Negative	Error Rate	Weighted Error Rate	Total Cost Ratio
Standar						
1	0.55	8.500	47.000	**1.2709%**	0.28%	1.3367
2	0.6	5.500	65.500	1.6258%	0.19%	1.9480
3	0.65	2.000	84.000	1.9693%	0.09%	4.2218
4	0.7	1.500	104.000	2.4158%	0.08%	4.7205
5	0.75	1.000	130.500	3.0112%	0.07%	5.2017
6	0.8	0.500	153.250	3.5207%	0.06%	**5.8991**
7	0.85	0.500	186.750	4.2878%	0.07%	5.0644
8	0.9	0.500	230.750	5.2954%	0.09%	4.2707
9	0.95	0.000	307.750	7.0472%	0.10%	3.8960
10	0.99	0.000	454.000	10.3962%	0.14%	2.6410
N-Grams Process						
1	0.55	3.750	51.000	**1.2537%**	0.13%	2.8146
2	0.6	3.000	71.000	1.6945%	0.12%	3.2318
3	0.65	2.500	94.000	2.2098%	0.11%	3.4855
4	0.7	2.000	116.500	2.7135%	0.10%	3.7883
5	0.75	1.500	142.500	3.2975%	0.09%	4.0991
6	0.8	1.000	162.750	3.7497%	0.08%	**4.5633**
7	0.85	1.000	191.000	4.3966%	0.09%	4.1203
8	0.9	1.000	233.000	5.3584%	0.10%	3.6006
9	0.95	0.500	301.500	6.9155%	0.11%	3.4111
10	0.99	0.500	427.750	9.8065%	0.15%	2.5097
Tag HTML dihapus						
1	0.55	4.500	79.500	**1.9235%**	0.17%	2.2644
2	0.6	2.500	110.000	2.5761%	0.11%	3.3306
3	0.65	1.000	140.000	3.2288%	0.08%	4.9958
4	0.7	0.000	178.500	4.0875%	0.06%	**6.7171**
5	0.75	0.000	210.750	4.8260%	0.07%	5.6892
6	0.8	0.000	252.500	5.7820%	0.08%	4.7485
7	0.85	0.000	309.250	7.0815%	0.10%	3.8771
8	0.9	0.000	395.500	9.0566%	0.12%	3.0316
9	0.95	0.000	513.250	11.7529%	0.16%	2.3361
10	0.99	0.000	723.500	16.5674%	0.23%	1.6572

No.	Spam Limit	False Positive	False Negative	Error Rate	Weighted Error Rate	Total Cost Ratio
Ambil URL dan informasi warna dari Tag HTML yang dihapus						
1	0.55	4.250	72.500	**1.7575%**	0.16%	2.4101
2	0.6	2.500	100.500	2.3586%	0.11%	3.4208
3	0.65	1.000	131.500	3.0341%	0.07%	5.1793
4	0.7	0.000	165.000	3.7783%	0.05%	**7.2667**
5	0.75	0.000	196.000	4.4882%	0.06%	6.1173
6	0.8	0.000	235.750	5.3984%	0.07%	5.0859
7	0.85	0.000	287.750	6.5892%	0.09%	4.1668
8	0.9	0.000	363.750	8.3295%	0.11%	3.2962
9	0.95	0.000	474.500	10.8656%	0.15%	2.5269
10	0.99	0.000	662.500	15.1706%	0.21%	1.8098
Pengelompokan $ dan !						
1	0.55	8.500	46.750	**1.2652%**	0.28%	1.3371
2	0.6	5.500	64.500	1.6029%	0.19%	1.9512
3	0.65	2.000	84.000	1.9693%	0.09%	4.2218
4	0.7	1.500	104.000	2.4158%	0.08%	4.7205
5	0.75	1.000	130.500	3.0112%	0.07%	5.2017
6	0.8	0.500	153.500	3.5264%	0.06%	**5.8919**
7	0.85	0.500	186.500	4.2821%	0.07%	5.0698
8	0.9	0.500	230.000	5.2782%	0.09%	4.2821
9	0.95	0.000	307.750	7.0472%	0.10%	3.8960
10	0.99	0.000	454.000	10.3962%	0.14%	2.6410
Perubahan token yang berupa nilai uang						
1	0.55	8.750	46.750	**1.2709%**	0.29%	1.3008
2	0.6	5.500	66.500	1.6487%	0.19%	1.9448
3	0.65	2.000	85.000	1.9922%	0.09%	4.2070
4	0.7	1.500	105.250	2.4445%	0.08%	4.6974
5	0.75	1.000	134.250	3.0971%	0.07%	5.1185
6	0.8	0.500	158.250	3.6352%	0.07%	**5.7575**
7	0.85	0.500	191.000	4.3852%	0.08%	4.9751
8	0.9	0.500	237.250	5.4442%	0.09%	4.1741
9	0.95	0.000	319.500	7.3162%	0.10%	3.7527
10	0.99	0.000	463.000	10.6022%	0.15%	2.5896

Berikut ini adalah keseluruhan hasil uji coba dengan melakukan perubahan pada nilai Strength. Konfigurasi yang digunakan adalah :

```
GroupMoneyVal=no
GroupDolar=yes
GroupExclam=yes
RemoveHTMLTag=no
NgramProcess=yes
```

No.	Spam *Limit*	False *Positive*	False *Negative*	Error Rate	Weighted Error Rate	Total Cost Ratio
Strength = 1						
1	0.55	4.500	48.750	**1.2194%**	0.16%	2.4040
2	0.6	3.000	70.750	1.6888%	0.12%	3.2340
3	0.65	2.500	92.500	2.1754%	0.11%	3.5007
4	0.7	2.000	116.000	2.7021%	0.10%	3.7943
5	0.75	1.500	139.750	3.2345%	0.09%	4.1381
6	0.8	1.000	161.500	3.7211%	0.08%	**4.5851**
7	0.85	1.000	187.500	4.3165%	0.09%	4.1704
8	0.9	1.000	229.500	5.2782%	0.10%	3.6388
9	0.95	0.500	299.000	6.8583%	0.11%	3.4355
10	0.99	0.500	425.250	9.7493%	0.15%	2.5229
Strength = 0.5						
1	0.55	5.250	42.500	**1.0934%**	0.18%	2.1128
2	0.6	4.000	52.250	1.2881%	0.14%	2.6512
3	0.65	2.500	69.750	1.6545%	0.10%	3.7498
4	0.7	2.500	90.750	2.1353%	0.11%	3.5187
5	0.75	2.000	111.250	2.5933%	0.10%	3.8522
6	0.8	1.500	127.000	2.9425%	0.09%	4.3285
7	0.85	1.250	151.500	3.4978%	0.09%	**4.3363**
8	0.9	1.000	178.500	4.1104%	0.09%	4.3052
9	0.95	1.000	230.750	5.3068%	0.10%	3.6251
10	0.99	1.000	336.750	7.7341%	0.14%	2.7453
Strength = 0.1						
1	0.55	7.250	35.500	**0.9789%**	0.24%	1.5766
2	0.6	5.500	40.250	1.0476%	0.19%	2.0313
3	0.65	5.000	49.000	1.2365%	0.17%	2.1840
4	0.7	4.250	60.250	1.4770%	0.15%	2.4709
5	0.75	3.250	75.750	1.8090%	0.13%	2.9919
6	0.8	2.500	86.500	2.0380%	0.11%	3.5632
7	0.85	2.500	100.500	2.3586%	0.11%	3.4208
8	0.9	1.000	118.750	2.7422%	0.07%	**5.4811**
9	0.95	1.000	149.250	3.4406%	0.08%	4.8104
10	0.99	1.000	219.250	5.0435%	0.10%	3.7557

No.	Spam Limit	False Positive	False Negative	Error Rate	Weighted Error Rate	Total Cost Ratio
Strength = 0.05						
1	0.55	7.750	36.000	**1.0018%**	0.26%	1.4784
2	0.6	6.750	39.500	1.0591%	0.22%	1.6781
3	0.65	6.000	44.250	1.1507%	0.20%	1.8611
4	0.7	4.500	54.250	1.3453%	0.16%	2.3778
5	0.75	4.000	65.250	1.5858%	0.15%	2.5771
6	0.8	3.750	77.250	1.8548%	0.14%	2.6512
7	0.85	2.500	92.250	2.1697%	0.11%	3.5033
8	0.9	2.000	107.000	2.4960%	0.10%	3.9055
9	0.95	1.750	130.750	3.0341%	0.10%	3.9215
10	0.99	1.000	186.750	4.2993%	0.09%	**4.1813**
Strength = 0.01						
1	0.55	6.750	38.500	**1.0362%**	0.22%	1.6804
2	0.6	6.000	41.250	1.0820%	0.20%	1.8698
3	0.65	6.000	46.500	1.2022%	0.20%	1.8546
4	0.7	5.250	53.750	1.3510%	0.18%	2.0717
5	0.75	4.750	62.500	1.5400%	0.17%	2.2307
6	0.8	4.250	73.000	1.7689%	0.16%	2.4076
7	0.85	3.250	82.000	1.9521%	0.13%	2.9459
8	0.9	2.500	94.250	2.2155%	0.11%	3.4829
9	0.95	1.750	113.750	2.6448%	0.09%	4.1524
10	0.99	1.000	159.000	3.6638%	0.08%	**4.6293**

LAMPIRAN B

PERHITUNGAN PROBABILITAS BAYESIAN

Berikut ini adalah beberapa contoh proses perhitungan probabilitas bayesian. Contoh email yang digunakan diambil dari SpamAssasin public Corpus. Dalam perhitungan digunakan konfigurasi *assume_prob* = 0.5, *Strength* = 0.1, *ham_limit* = 0.45 dan *spam_limit* = 0.55.

Total Training Data	
Ham	3168
Spam	1197

Untuk melakukan perhitungan pada contoh kasus, digunakan Microsoft Excel. Formula Excel yang digunakan antara lain adalah:

- Chidist, digunakan untuk menghitung nilai chi-square
- Product, digunakan untuk menghitung $\prod Probability$

Ham diklasifikasikan sebagai spam (false positive)

Subject: This Weeks Movie Trivia Question from All Things New England
From: "All Things New England - Movie Trivia" trivia@allthingsnewengland.com
Body:

Hello Friends! We hope you had a pleasant week Last weeks trivia questions was: What do these 3 films have in common: One Crazy Summer, Whispers in the Dark, Moby Dick? Answer: Nantucket Island Congratulations to our Winners: Caitlin O of New Bedford, Massachusetts Brigid M of Marblehead, Massachusetts

Special "Back to School" Offer! For a limited time order our "Back to School" Snack Basket and receive 20% Off & FREE SHIPPING! Just enter coupon code COLLEGE2002 in the space provided during checkout to receive your exclusive discount Click here to order today:
http://www.allthingsnewengland.com/cgi-local/store3/agora.cgi?product=SPECIAL

This weeks movie trivia questions is posted on our homepage www.allthingsnewengland.com Have a great weekend everyone! Sincerely, Joe, Liz, Jenna and Maggie All Things New England "We bring New England to you!" www.allthingsnewengland.com

Dilakukan tokenizer pada data email yang masuk dan kemudian dihitung probabilitas untuk masing-masing token berdasarkan dari data training yang ada dalam database.

Token	ham	spam	Prob
Whisper	4	0	0.012195
trivia	4	0	0.012195
pleasant	2	0	0.02381
Maggie	1	0	0.045455
posted	140	8	0.131617
weekend	48	5	0.216645
Thing	542	97	0.321444
What	993	236	0.386137
Summer	68	18	0.412066
Last	375	100	0.413771
space	109	30	0.421496
School	95	30	0.455306
Off	345	110	0.457665
question	302	108	0.486253
Massachusett	11	4	0.490485
the	2809	1022	0.490555
store3	0	0	0.5
Nantucket	0	0	0.5
Moby	0	0	0.5
Marblehead	0	0	0.5
One	999	413	0.522477
Winner	36	16	0.540421
week	270	122	0.54459
New	1028	467	0.545927
Thi	1864	871	0.55291
these	438	216	0.566184
time	846	432	0.574727
you	1642	970	0.6099
Snack	3	2	0.635548
provided	83	59	0.65283
today	282	208	0.661227
movie	59	44	0.663565
your	997	928	0.71126
Liz	1	1	0.715022
product	222	212	0.716455
order	181	206	0.750694
Special	129	208	0.810061
limited	55	97	0.823348
our	391	767	0.838464
receive	116	369	0.89375
you!	19	85	0.921714
Offer!	4	22	0.934048
Sincerely	12	70	0.938633

Token	ham	spam	Prob
Basket	3	0	0.016129
film	39	4	0.21416
Joe	54	7	0.255843
Dark	33	5	0.286786
had	511	87	0.310661
common	116	22	0.334317
checkout	4	1	0.400186
Crazy	30	8	0.413979
Answer	188	56	0.440851
Just	1053	345	0.46442
bring	108	36	0.46873
Island	24	8	0.468806
hope	140	49	0.480884
Back	409	147	0.487505
Dick	8	3	0.49813
Jenna	0	0	0.5
COLLEGE2002	0	0	0.5
cgi-local	0	0	0.5
Caitlin	0	0	0.5
Brigid	0	0	0.5
agora.cgi	0	0	0.5
http	2655	1011	0.501944
during	134	52	0.506668
code	248	99	0.513736
and	2452	987	0.515817
have	1537	644	0.525824
coupon	7	3	0.531143
For	2133	943	0.539184
All	1233	601	0.563323
from	1587	778	0.564734
Bedford	2	1	0.567335
great	274	149	0.590011
England	28	16	0.601735
Hello	62	50	0.680795
exclusive	32	30	0.7124
enter	80	76	0.715309
here	681	667	0.721603
Click	529	630	0.759126
homepage	8	10	0.766408
FREE	390	525	0.780803
discount	22	33	0.798247
20%	12	30	0.867831
Congratulation	8	21	0.872886

SHIPPING!	3	28	0.95961
Friends!	0	1	0.954545
everyone!	0	6	0.991803

Ket Tabel:

- Token ➔ hasil dari proses tokenizer terhadap data email yang masuk.
- Ham ➔ jumlah email **bukan spam** yang digunakan dalam training yang mengandung token yang bersangkutan.
- Spam ➔ jumlah email **spam** yang digunakan dalam training yang mengandung token yang bersangkutan.
- Probability ➔ hasil perhitungan bayesian terhadap token yang bersangkutan berdasarkan dari data hasil training.

Dihitung:

$$H = 1 - Chidist\left(-2 * Ln\left(\prod \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(1.68069E-31), 2 * 89\right) = 1 - Chidist\left(-2 * -70.860,\ 178\right)$$

$$= 1 - Chidist\left(141.7219,\ 178\right) = 1 - 0.9791$$

$$= 0.0209$$

$$S = 1 - Chidist\left(-2 * Ln\left(\prod 1 - \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(1.2017E-38), 2 * 89\right) = 1 - Chidist\left(-2 * -87.3145,\ 178\right)$$

$$= 1 - Chidist\left(174.629,\ 178\right) = 1 - 0.55741$$

$$= 0.44259$$

$$Prob = (S - H + 1)/2$$

$$= (0.44259 - 0.0209 + 1)/2$$

$$= 0.710845$$

➔ lebih besar dari spam_limit (> 0.55), diklasifikasikan sebagai spam

Ham diklasifikasikan sebagai Not Sure

Subject: Automated 30 day renewal reminder 2002-05-27
From: "Starflung NIC" <nic@starflung.com>
Body:

The following domains that are registered as belonging to you are due to expire within the next 60 days If you would like to renew them, please contact nic@starflung.com; otherwise they will be deactivated and may be registered by another Domain Name, Expiry Date nutmegclothing.com, 2002-06-26

Token	ham	spam	Prob
nic	2	0	0.02381
otherwise	73	9	0.246329
renew	7	1	0.277143
reminder	19	3	0.295654
they	883	196	0.370079
would	810	236	0.435389
that	2079	606	0.435494
them	579	185	0.458187
next	304	106	0.479939
The	2809	1022	0.490555
starflung.com	0	0	0.5
Starflung	0	0	0.5
nutmegclothing.com	0	0	0.5
registered	85	47	0.59399
you	1642	970	0.6099
will	842	604	0.654986
renewal	5	5	0.723538
within	115	194	0.816906
please	325	624	0.835532

Token	ham	spam	Prob
deactivated	4	0	0.012195
Date	899	59	0.148025
another	341	90	0.411273
like	893	285	0.4579
2002-06-26	0	0	0.5
2002-05-27	0	0	0.5
and	2452	987	0.515817
domain	73	34	0.552057
Automated	33	16	0.561893
Expiry	2	1	0.567335
are	1346	716	0.584689
may	467	255	0.591015
due	108	64	0.610583
belonging	10	7	0.648573
day	372	268	0.655943
following	177	149	0.690147
Name	319	429	0.780628
contact	133	230	0.820599
expire	6	14	0.858841

$$H = 1 - Chidist\left(-2 * Ln\left(\prod \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(1.40108E\text{-}14), 2 * 38\right) = 1 - Chidist\left(-2 * \text{-}31.8989,\ 76\right)$$

$$= 1 - Chidist\left(63.7979,\ 76\right) = 1 - 0.8397$$

$$= 0.1603$$

$$S = 1 - Chidist\left(-2 * Ln\left(\prod 1 - \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(2.3331E\text{-}14), 2 * 38\right) = 1 - Chidist\left(-2 * \text{-}31.38898,\ 76\right)$$

$$= 1 - Chidist\left(62.77797,\ 76\right) = 1 - 0.86146$$

$$= 0.13854$$

$$Prob = (S - H + 1)/2$$

$$= (0.13854 - 0.1603 + 1)/2$$

$$= 0.489143$$ ➔ lebih dari ham_limit (>0.45) dan kurang dari spam_limit (<0.55), diklasifikasikan sebagai Not Sure

Ham diklasifikasikan sebagai ham

Subject: [Spambayes] speed
From: skip@pobox.com (Skip Montanaro)

Body:

If the frequency of my laptop's disk chirps are any indication, I'd say hammie is about 3-5x faster than SpamAssassin.

Skip

Token	ham	spam	Prob
Spambaye	137	0	0.000365
pobox.com	62	0	0.000805
Montanaro	34	0	0.001466
SpamAssassin	175	1	0.015174
laptop	65	4	0.140579
Skip	60	7	0.236318
say	477	84	0.317937
speed	94	29	0.449535
indication	9	3	0.468967
the	2809	1022	0.490555
than	674	250	0.495379

Token	ham	spam	Prob
hammie	11	0	0.004505
chirp	1	0	0.045455
3-5x	1	0	0.045455
I'd	280	12	0.102008
about	951	292	0.44832
any	776	391	0.57146
faster	59	30	0.573612
disk	61	32	0.58122
are	1346	716	0.584689
frequency	7	7	0.724172

$$H = 1 - Chidist\left(-2 * Ln\left(\prod \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(1.2145E-22), 2 * 21\right) = 1 - Chidist\left(-2 * -50.4625,\ 42\right)$$

$$= 1 - Chidist\left(100.9250,\ 42\right) = 1 - (9.299E-07)$$

$$= 0.99999907$$

$$S = 1 - Chidist\left(-2 * Ln\left(\prod 1 - \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(0.000130154), 2 * 21\right) = 1 - Chidist\left(-2 * -8.9468,\ 42\right)$$

$$= 1 - Chidist\left(17.8936,\ 42\right) = 1 - 0.9996$$

$$= 0.0004$$

$$Prob = (S - H + 1)/2$$

$$= (0.0004 - 0.99999907 + 1)/2$$

$$= 0.000407488$$ ➔ kurang dari ham_limit (<0.45), diklasifikasikan sebagai Ham

Spam diklasifikasikan sebagai ham (false negative)

Subject: The Ultimate in PC Security and Surveillance.
From: "Nok-Nok" <remove@nok-nok.net>
Body:

PLEASE VISIT WWW.NOK-NOK.NET

Nok-Nok :::: The Ultimate in PC Security and Surveillance Software.

Nok-Nok is an award-winning security and surveillance software developed to monitor, check, record, chronicle, log, supervise, scrutinize, examine and remotely observe every input and screen view on a stand-alone or networked PC. Using Nok-Nok allows you to be fully aware of everything that is done or seen on a monitored computer - safely, securely and secretly. Nok-Nok is the most powerful software in its class, the easiest to use, and the most discreet. The same technology has been employed by businesses all over the world, "Fortune 500" companies, intelligence agencies, public sector institutions (universities, libraries and schools) and the military alike. Nok-Nok is also ideal for many specialized tasks such as usability testing, scientific studies of PC usage, software training tools, remote network security and much more.

Has your confidential data been stolen, manipulated or accidentally deleted? Are your employees sending out valuable company secrets? Are your children safe online? Is your spouse having an online affair? Is your computer being used to spy on you online or offline? Find out the answers with Nok-Nok ::: The Ultimate in PC Security and Surveillance Software. Nok-Nok will alert you to exactly who is doing what on your PC and when. Nok-Nok comes in two editions: Nok-Nok Home and Small Business Edition, and Nok-Nok Enterprise Server Edition. The Nok-Nok Software has been developed to be Windows platform independent: It runs on Windows 95/98/ME/NT/2000/XP. A Macintosh compatible version will be available soon for the consumer market.

Token	ham	spam	Prob
usability	18	0	0.002762
remotely	16	0	0.003106
Surveillance	9	0	0.005495
stand-alone	4	0	0.012195
supervise	3	0	0.016129
scrutinize	3	0	0.016129
stolen	19	2	0.219226
run	359	44	0.244991
version	362	47	0.255802
task	92	12	0.256856
platform	97	13	0.262046
testing	98	14	0.274558
screen	88	13	0.281298
spy	20	3	0.285111
Server	299	48	0.298242
remote	60	11	0.32694
Window	259	52	0.347038
when	840	179	0.36062
Using	521	114	0.366751
same	480	114	0.385984
what	993	236	0.386137
sector	21	5	0.386993
two	388	94	0.390709
used	365	89	0.392246

Token	ham	spam	Prob
librarie	40	0	0.001247
alike	14	0	0.003546
accidentally	8	0	0.006173
networked	6	0	0.008197
manipulated	2	0	0.02381
log	190	12	0.143392
award-winning	16	1	0.144029
monitor	53	5	0.200312
Macintosh	12	2	0.307462
network	344	58	0.308596
Enterprise	69	12	0.315428
institution	16	3	0.33254
done	207	50	0.390019
data	261	70	0.415169
having	220	62	0.427244
alert	35	10	0.430735
doing	230	67	0.43536
affair	17	5	0.437984
answer	188	56	0.440851
exactly	129	41	0.456893
being	432	139	0.459926
examine	15	5	0.468864
also	626	210	0.470298
ideal	29	10	0.477217

such	382	102	0.414088
world	527	143	0.417992
view	154	42	0.419255
use	878	252	0.431702
that	2079	606	0.435494
school	95	30	0.455306
technology	183	60	0.464609
scientific	15	5	0.468864
securely	6	2	0.469095
The	2809	1022	0.490555
seen	189	71	0.498554
www.nok-nok.net	0	0	0.5
nok-nok.net	0	0	0.5
Nok-Nok	0	0	0.5
record	127	51	0.515217
with	1651	706	0.530899
soon	137	60	0.536826
Security	187	84	0.543124
Small	177	80	0.544653
out	922	428	0.55128
training	75	35	0.552543
tool	170	80	0.554636
Ultimate	50	24	0.559464
who	617	298	0.561064
Software	325	159	0.56422
offline	9	5	0.594523
universitie	7	4	0.601048
you	1642	970	0.6099
over	442	268	0.616068
studie	24	15	0.622914
public	173	118	0.643471
online	258	185	0.654872
will	842	604	0.654986
powerful	73	54	0.66178
observe	4	3	0.662664
safe	52	39	0.664806
usage	26	21	0.680905
spouse	7	6	0.692571
your	997	928	0.71126
sending	87	87	0.725644
VISIT	128	200	0.805178
PLEASE	325	624	0.835532
secret	44	89	0.842346
safely	12	32	0.875042
remove	129	359	0.880382
specialized	3	9	0.884934
valuable	19	72	0.908883
secretly	1	6	0.934549

many	412	145	0.482259
developed	44	16	0.490437
employee	37	14	0.500355
aware	42	16	0.502047
edition	84	32	0.502049
much	488	186	0.502178
Find	447	172	0.504554
monitored	5	2	0.514043
computer	302	121	0.514654
and	2452	987	0.515817
been	681	275	0.516615
most	507	207	0.51936
for	2133	943	0.539184
more	969	447	0.549726
2000	103	48	0.552213
check	311	145	0.552352
everything	150	70	0.552566
come	323	154	0.557873
all	1233	601	0.563323
military	38	20	0.581966
Are	1346	716	0.584689
every	291	159	0.591164
input	105	61	0.605857
consumer	67	39	0.606287
intelligence	29	17	0.607834
children	52	31	0.611936
fully	66	44	0.638134
class	112	77	0.645256
Home	341	250	0.659876
market	146	108	0.661844
chronicle	8	6	0.663817
businesse	41	31	0.666558
easiest	27	21	0.67268
allow	176	143	0.68252
available	229	190	0.687053
companie	134	112	0.6886
compatible	23	22	0.716357
agencie	23	25	0.741549
deleted	22	26	0.757206
Business	257	306	0.759061
500	60	72	0.760336
company	182	243	0.779362
employed	16	23	0.791116
independent	46	79	0.819411
Fortune	16	60	0.907928
confidential	15	58	0.910419
discreet	0	6	0.991803

$$H = 1 - Chidist\left(-2 * Ln\left(\prod \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$
$$= 1 - Chidist\left(-2 * Ln(3.93691E-63), 2 * 143\right)$$
$$= 1 - Chidist\left(-2 * -143.6924,\ 286\right)$$
$$= 1 - Chidist\left(287.38492,\ 286\right) = 1 - (0.46586)$$
$$= 0.53414$$
$$S = 1 - Chidist\left(-2 * Ln\left(\prod 1 - \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$
$$= 1 - Chidist\left(-2 * Ln(2.03517E-52), 2 * 143\right)$$
$$= 1 - Chidist\left(-2 * -119.02384,\ 286\right)$$
$$= 1 - Chidist\left(238.0476871,\ 286\right) = 1 - 0.9822$$
$$= 0.0178$$

$$Prob = (S - H + 1)/2$$
$$= (0.0178 - 0.53414 + 1)/2$$
$$= 0.24184$$ ➔ kurang dari ham_limit (<0.45), diklasifikasikan sebagai Ham

Spam diklasifikasikan sebagai Not Sure

Subject: Proposal for cooperation
From: Oksana Svetlova <oksana@antee.com>
Body:

Hello I meant to contact http://taint.org webmaster,

Am I correct, I hope? Actually, I am not quite sure Did I or Natasha Fedorova write you before to your other address? Anyway, now I am here instead of her too So, my name is Oksana Svetlova, I live in St Petersburg, Russia I have an idea, how we could cooperate to mutual benefit, see, possibly you will find it interesting People are seeking to cut costs with this economy slow-down, right? The thing is that I could get you in touch with developers from Russia, who will be happy to do a web designer's work for you for considerably less, than you possibly take from your clients (or pay for web-design?), and with professional quality Also all programming, Internet-programming, graphic, flash design, content writing, translations, etc

If you do web-design, then you may accept more orders at once, getting a good share from each, and devote yourself only to that work, you find most interesting Alternatively, you can realize more your ideas to develop, more features to add to your existing sites, and with low costs: from $12, and an average of $18 an hour I have put links to their works at www.antee.com/samples_and_faq.html As for reliability, I just saw, how carefully they hand-picked their specialists, and under what tight management they do their projects So, from past experience, it was almost always possible to meet deadlines Also, communication is properly thought-out: a consultant is online in ICQ and

other instant messengers almost each day, also their company has local consultants in many states throughout the USA, in Europe and Australia

So, what do you think? A possibility for cooperation? Thank you very much, Oksana Svetlova oksana@antee.com PS Just in case you need any type of IT works now, please let me know a bit of details, they will return you a cost/time estimation

$$H = 1 - Chidist\left(-2 * Ln\left(\prod \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(2.489E-64), 2 * 177\right) = 1 - Chidist\left(-2 * -146.454 \;\; 354\right)$$

$$= 1 - Chidist\left(292.9070,\; 354\right) = 1 - 0.9921925$$

$$= 0.0078075$$

$$S = 1 - Chidist\left(-2 * Ln\left(\prod 1 - \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(2.579E-64), 2 * 177\right) = 1 - Chidist\left(-2 * -146.418,\; 354\right)$$

$$= 1 - Chidist\left(292.83634,\; 354\right) = 1 - 0.992257$$

$$= 0.007743$$

$$Prob = (S - H + 1)/2$$

$$= (0.007743 - 0.0078075 + 1)/2$$

$$= 0.49996773$$ ➔ lebih dari ham_limit (>0.45) dan kurang dari spam_limit (<0.55), diklasifikasikan sebagai Not Sure

Spam diklasifikasikan sebagai spam

Subject: GET IN AT THE TOP!
From: "good4u" <alewss6@hotmail.com>
Body:

DON'T MISS OUT ON AN AMAZING BUSINESS OPPORTUNITY AND WEIGHT LOSS PRODUCT! PLEASE VISIT www.good4u.autodreamteam.com THERE IS NO OBLIGATION AND IT'S WORTH A LOOK!

Token	ham	spam	Probability
THERE	1046	224	0.361755
WORTH	146	45	0.449286
DON'T	796	289	0.49003
THE	2809	1022	0.490555
TOP!	0	0	0.5
good4u	0	0	0.5
alewss6	0	0	0.5

Token	ham	spam	Probability
AMAZING	31	30	0.71884
BUSINESS	257	306	0.759061
hotmail.com	92	139	0.799818
VISIT	128	200	0.805178
LOSS	35	63	0.826174
PLEASE	325	624	0.835532
OPPORTUNITY	44	137	0.891566

AND	2452	987	0.515817	WEIGHT	16	51	0.893437
GET	1070	470	0.537577	LOOK!	1	4	0.905581
OUT	922	428	0.55128	OBLIGATION	21	84	0.913299
MISS	47	34	0.656703	PRODUCT!	0	7	0.992958

$$H = 1 - Chidist\left(-2 * Ln\left(\prod \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(7.779E-05), 2 * 22\right) = 1 - Chidist\left(-2 * -9.4614359,\ 44\right)$$

$$= 1 - Chidist\left(18.922872,\ 44\right) = 1 - 0.9996583$$

$$= 0.0003417$$

$$S = 1 - Chidist\left(-2 * Ln\left(\prod 1 - \Pr obability\right), 2 * count\left(\Pr obability\right)\right)$$

$$= 1 - Chidist\left(-2 * Ln(1.981E-14), 2 * 22\right) = 1 - Chidist\left(-2 * -31.55271,\ 44\right)$$

$$= 1 - Chidist\left(63.105425,\ 44\right) = 1 - 0.0308537$$

$$= 0.9691463$$

$$Prob = (S - H + 1)/2$$

$$= (0.9691463 - 0.0003417 + 1)/2$$

$= 0.984402$ ➔ lebih besar dari spam_limit (> 0.55), diklasifikasikan sebagai spam