

# Analisis Perancangan dan Implementasi Mail System Pada PT. PLN (Persero) SBAM

**Guntoro Barovih**  
STMIK PALCOMTECH  
E-mail: [guntorobarovih@gmail.com](mailto:guntorobarovih@gmail.com)

## **Abstrak**

*Email atau disebut juga sebagai surel (surat elektronik) merupakan suatu cara baru dalam berkomunikasi menggunakan surat. Dahulu berkirim surat menggunakan kertas, hal tersebut memunculkan permasalahan karena aktifitas logging bertambah karena permintaan kertas bertambah. Untuk menekan permasalahan tersebut dibuatlah sistem pesan elektronik yang dapat efektif dalam menunjang kegiatan berkirim surat. Baik kegiatan surat-menyurat perorangan maupun berkelompok seperti dalam perusahaan. PT. PLN (Persero) yang merupakan salah satu perusahaan yang banyak menggunakan kegiatan berkirim surat. Untuk menekan penggunaan kertas dalam kegiatan berkirim surat dalam perusahaan, maka dibangunlah sistem email agar kegiatan berkirim surat tetap berlangsung. Sistem email yang dibangun menggunakan metode Software Development Life Cycle (SDLC) dan pengujian blackbox. Hasil yang didapatkan adalah bahwa sistem yang dirancang bisa dimanfaatkan dengan baik dalam proses kegiatan berkirim surat dari satu node ke node lain. Tiap komponen sistem berfungsi dengan baik, security system mampu memfilter pesan spam dan sistem penghitung transaksi pesan memiliki kesamaan data dengan log sistem email dan disajikan dalam bentuk grafik.*

**Kata Kunci** — *E-mail, Software Development Life Cycle, Surat-menyurat, sistem email, blackbox.*

## **Abstract**

*Email or also known as e-mail (electronic mail) is a new way of communicating using the mail. First send a letter using paper, it gave rise to problems because of increased logging activities due to increased paper demand. To suppress these problems made the electronic messaging system can effectively support the activities send a letter. Both individual and group correspondence activities as in the company. PT. PLN (Persero) which is one of the many companies using the send mail activity. To suppress the use of paper in the activity data used in the company's letter, then woke up email system to send mail tertap activities take place. Email systems are built using the method of Software Development Life Cycle (SDLC) and blackbox testing. The result obtained is that the system designed can put to good use in the process of the activity send a letter from one node to another node. Each component of the system is functioning properly, the security system is able to filter out spam messages and system messages have the same transaction counter data with the system log email and presented in the form of graphs.*

**Keywords** — *E-mail, Software Development Life Cycle, Correspondence, mail system, blackbox.*

---

## 1. PENDAHULUAN

Kegiatan surat-menyurat dalam suatu kegiatan yang selalu dilakukan dalam kehidupan sehari-sehari yang berhubungan dengan kegiatan penjualan, penawaran maupun kegiatan lainnya. Sebelum adanya ARPANET sebagai wadah organisasi jaringan komputer, kegiatan bertukar surat-menyurat dilakukan dengan konvensional, dimana surat dituliskan dalam media kertas dan dikirimkan ke penyedia jasa pengiriman surat untuk menyampaikan surat ke alamat yang dituju. Semakin hari kebutuhan kertas untuk kegiatan ini bertambah besar dan mengakibatkan kegiatan logging atau penebangan pohon semakin bertambah seiring dengan meningkatnya permintaan kebutuhan kertas.

Seiring perkembangan jaringan komunikasi internet, maka munculah fasilitas pengiriman surat dalam bentuk digital atau sering disebut sebagai elektronik mail (*E-mail*). *E-mail* merupakan hasil evolusi dari perkembangan internet yang digagas oleh ARPANET dan awal mula munculnya *E-mail* di awal tahun 1970-an [1]. *E-mail* atau *Electronic Mail* atau surat elektronik adalah pesan, atau surat secara elektronik, baik berupa teks maupun gabungan dengan gambar yang dikirimkan dari satu alamat ke alamat lain di jaringan internet [2].

Dengan munculnya fasilitas *email* membuat proses pengiriman surat tidak perlu membutuhkan waktu yang lama seperti yang terjadi pada kegiatan pengiriman surat secara konvensional dan dapat mengurangi penggunaan kertas. *E-mail* dibedakan dalam 2 kategori, yaitu email public yang disediakan oleh penyedia *email* seperti google dan yahoo dan juga email *private* yaitu sistem *email* yang di bangun sendiri pada suatu instansi atau suatu perusahaan.

*E-mail* kategori ini sangat perlu digunakan sebagai penanda legalitas bahwa pengirim *email* merupakan salah satu pekerja yang ada pada perusahaan tersebut. PT. PLN (persero) SBAM merupakan salah satu perusahaan yang membutuhkan fasilitas sistem *E-mail* ini. Berdasarkan kebutuhan tersebut dirancanglah suatu sistem *email* yang bisa digunakan secara *private* pada perusahaan yang dikelola secara internal.

### 1.1. Penelitian Terdahulu

Di sumber lainnya [2] dihasilkan mobilitas pengguna yang tinggi memicu munculnya penyediaan fasilitas pengiriman surat elektronik pada 3 media, yaitu melalui pesan singkat, melalui telepon pesan suara dan akses internet bergerak. Namun ketiga media alternatif yang ditawarkan tidak dapat sepenuhnya menggantikan cara akses konvensional dan harus memperhatikan faktor kenyamanan pengguna.

Lubis dan Batubara [3] dihasilkan *software* yang berbasis *opensource* yang bisa digunakan untuk membangun fasilitas *email* server salah satunya adalah CentOS dan *Zimbra Collaboration Suites*.

Dari sumber lainnya [4] dihasilkan dengan adanya *mail exchange server cluster* kegagalan yang terdapat pada proses berjalannya server dapat dihilangkan dan komunikasi *e-mail* antara perusahaan dengan anak perusahaan tidak terganggu.

Kemudian di sumber lainnya [5] dihasilkan dengan adanya *mail server* yang terintegrasi dengan antar muka *mail compose* yang berbasis desktop yang dibangun menggunakan VB.Net dapat mempermudah dalam proses pertukaran data dan penyimpanan secara *online* berdasarkan masing-masing akun untuk melakukan transfer file antara anggota dimanapun dan kapanpun anggota berada.

Di sumber lainnya [6] dihasilkan mail server bisa diterapkan di virtual server di level sistem operasi menggunakan OpenVZ, agar dapat mengoptimalkan virtualisasi ini maka disarankan menggunakan sistem *cluster* dalam arsitektur virtual.

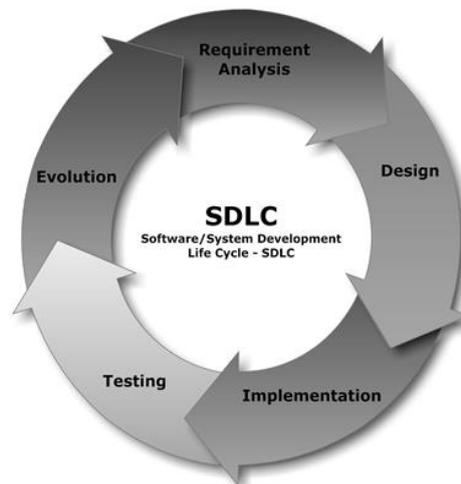
Di sumber lain [7] dihasilkan mail server yang digunakan adalah *mail zimbra* dengan menggunakan pengaman berganda dimana *port* yang digunakan hanya *port* untuk komunikasi *email*, *port* yang lain tidak diaktifkan untuk mencegah terjadinya serangan *hecker* dan *bruteforce method*.

Di sumber lainnya [8] dihasilkan pengujian *email* yang dilakukan di dapatkan bahwa sistem yang di bangun mampu untuk memblokir pesan-pesan yang mengandung *spam* dan dapat mengampaiakan pesan yang dikirim, tetapi kelemahan penelitian ini tidak adanya metode pengembangan dan metode pengujian yang dilakukan.

## 2. METODE PENELITIAN

[9] Jenis penelitian yang digunakan adalah penelitian terapan (*applied research*). Sedangkan metode yang digunakan yaitu *software development life cycle* (SDLC) tampak pada Gambar 1, dimana tahapan-tahapan yang dilakukan diantaranya :

1. Tahapan pertama yang dilakukan adalah mendengar kebutuhan yang diperlukan oleh pengguna dari kegiatan berkirim surat yang dilakukan sekarang ini. Dimana diperlukan suatu sistem yang dapat menggantikan cara berkirim surat konvensional ke sistem digital yang membutuhkan waktu relatif lebih singkat.
2. Dari kebutuhan penggunaan kemudian di kembangkan ke arak kebutuhan sistem perangkat keras dan sistem perangkat lunak yang bisa digunakan untuk membangun sistem *email* tersebut serta penempatan sistem ke dalam jaringan yang ada.
3. Dari tahapan analisis yang dilakukan kemudian dikembangkan ke perancangan sistem dan perancangan arsitektur jaringan mail sistem.
4. Pada tahap implementasi dari siklus SDLC diterapkan dalam bentuk *prototype* sebelum diterapkan langsung ke sistem yang sesungguhnya, hal sama juga untuk tahapan *testing* dimana *email* akan uji dengan cara saling bertukar pesan dan pengujian enkripsi dan filter *spam* demi menjaga ke aslian pesan sampai ke tujuan dan evolusi setiap kekurangan yang dilakukan pada saat pengujian dilakukan pembaharuan sampai ke aslian pesan yang dikirim tetap terjaga. Untuk mendapatkan kelayakan atau tidaknya sistem ini diterapkan pada PT. PLN (Persero) SBAM.

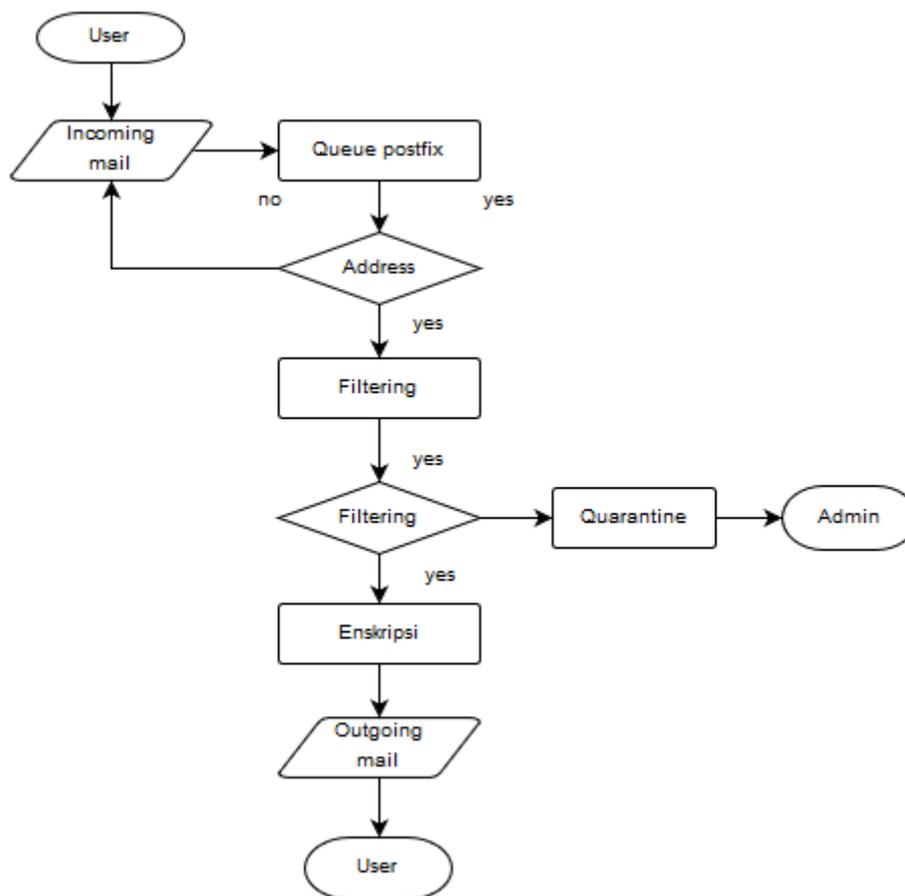


Gambar 1. Metode Software Development Life Cycle [9]

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Model proses

Model proses data yang berjalan pada sistem email jika di gambarkan secara singkat ke dalam bentuk alur proses tampak pada Gambar 2.



Gambar 2 Model proses alur komunikasi email

Pada Gambar 2 tersebut dapat dijelaskan pada saat *user* mengirim *email* melalui antar muka *roundcube*, maka *email* akan diteruskan ke *port* SMTP yang dikelola oleh postfix. Data yang masuk akan dikoreksi alamatnya di dalam data antrian. Jika alamat tidak diketemukan dan dinyatakan salah maka pesan akan dikembalikan kembali ke pengirim. Jika alamat yang dituju adalah benar maka pesan akan diteruskan ke sistem *filtering* yang melibatkan *clamav*, *amavisd-new*, *spamassassin* untuk mengecek muatan data yang dikirim melalui *email*, hal ini diciptakan untuk melindungi data pengguna.

Jika muatan data yang dikirim mengandung *spam*, *malware* dan sejenisnya. Maka, pesan akan dikirimkan ke ruang penyimpanan *email* terinfeksi dan diisolasi. Pesan pemberitahuan akan adanya pesan yang terinfeksi dikirimkan ke administrator. Jika pesan dinyatakan aman dan tidak terinfeksi. Maka pesan diteruskan ke DKIM sebagai pengenkripsi pesan dan pemberi kunci validasi agar *email* yang dikirim ke *public* di kenali oleh sistem jaringan ISP sebagai *email* yang bersih yang dikategorikan sebagai *white list mail* dan dapat diteruskan ke pengguna yang dituju oleh *email* tersebut.

### 3.2. Analisis Kebutuhan Sistem

Adapun analisis kebutuhan *user* yang digunakan diantaranya meliputi, kebutuhan perangkat keras, kebutuhan perangkat lunak, arsitektur jaringan.

### 3.3. Kebutuhan Perangkat Keras

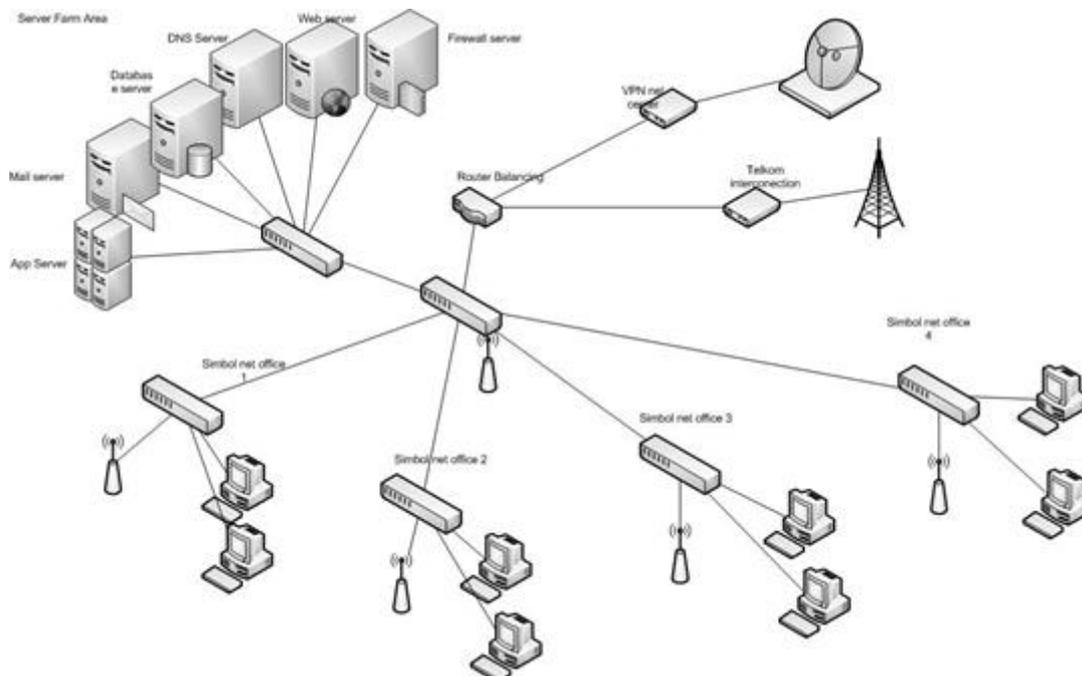
Adapun perangkat keras yang dibutuhkan dalam sistem *email* ini meliputi media server sebagai wadah pengelola *service* sistem *email* yang nantinya akan dijalankan. Kebutuhan ini dipertimbangkan mengingat waktu kerja yang dilakukan oleh sistem adalah terus menerus, maka diperlukan spesifikasi yang sangat baik dalam memproses sistem *email*. Spesifikasi tersebut antara lain : *processor xeon quad core 3.1Ghz*, 4Gb memori, 1TB harddisk. Dengan spesifikasi ini diharapkan dapat menopang kinerja distribusi *email* dengan sangat baik dan stabil.

### 3.4. Kebutuhan perangkat lunak

Adapun perangkat lunak yang dibutuhkan dalam sistem *email* ini meliputi kebutuhan sistem operasi, *Mail Transfer Agen*, *Mail User Agent*, *security system* yang seluruhnya di dukung oleh kebebasan dalam melakukan management sistem, yaitu menggunakan sistem dan perangkat lunak dukungan *opensource*, diantaranya, *slackware* sebagai sistem operasi, *postfix* sebagai pengelola *email*, *dovecot* sebagai pengirim *email*, *roundcube* sebagai antar muka, kolaborasi *clamav* dan *spamassasin* sebagai *security system* dan *dkim* dan *spf* sebagai enkripsi *email*.

### 3.5. Arsitektur jaringan

Sebelum dilakukan pengujian dari sistem email yang dibangun, langkah awal yang dilakukan adalah memetakan arsitektur jaringan yang akan digunakan oleh sistem mail nanti, hal ini dilakukan agar mendapatkan pengujian yang mendekati situasi nyata mengingat pengujian dan pembangunan sistem yang dilakukan dalam bentuk *prototype*. Arsitektur jaringan yang digunakan tampak pada gambar 3.

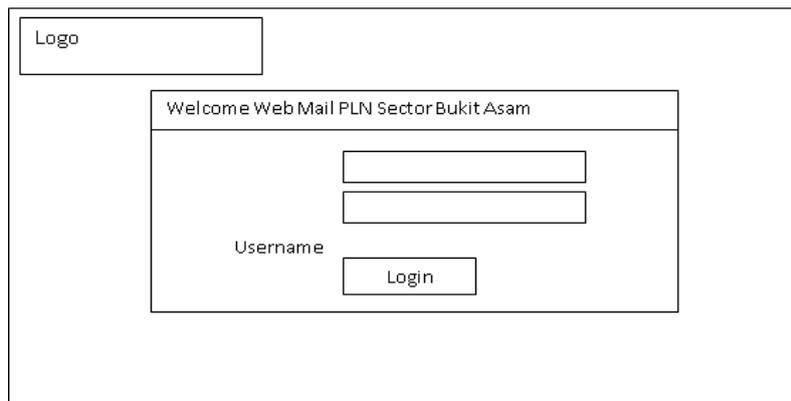


Gambar 3 Arsitektur jaringan PT. PLN (Persero) SBAM

*E-mail* server ditempatkan di jajaran server *farm* yang nantinya akan terintegrasi dengan DNS server. Jalur komunikasi *public* digunakan 2 jalur, dimana jalur pertama menggunakan jalur VPN perusahaan dan jalur *public*. Dari 2 jalur koneksi *public* tersebut akan terkonsentrasi di 1 *router balancing*. Dan di distribusikan ke *switch catalys* dan didistribusikan ke pengguna di *network* PLN SBAM.

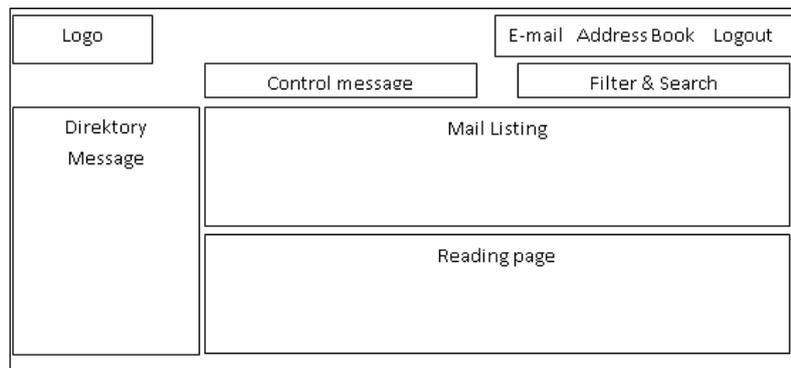
### 3.6. Analisis Desain Interface

Untuk memudahkan pengguna dalam berinteraksi dengan sistem email yang nanti digunakan maka dirancanglah suatu rancangan *interface* mulai dari halaman *login*, halaman ini digunakan sebagai pembatas dimana hanya *user* yang terdaftar di dalam sistem saja yang boleh menggunakan fasilitas *email*. Tampak pada Gambar 4



Gambar 4. Desain login page

Pada Gambar 5 merupakan tampilan halaman *inbox email* yang digunakan untuk menampilkan isi dari suatu pesan elektronik yang diterima maupun fungsi lain seperti pembuatan pesan baru hapus dan lain sebagainya.



Gambar 5. Desain mailbox user

Pada Gambar 6 merupakan tampilan halaman pembuat pesan (*Compose message*) halaman dibuat dengan sangat simpel dengan tujuan agar pengirim atau pembuat pesan lebih terfokus pada pesan yang akan dibuatnya.

Gambar 6. Halaman antar muka *compose message*

### 3.7. Interkoneksi server

Agar sistem *email* bisa berkomunikasi dengan *public*. Maka diperlukan sebuah DNS yang dapat mentranslasikan nama ke alamat yang mudah di ingat, Tabel 1 menyatakan pentranslasian nama sistem mail dari alamat yang mudah di ingat ke alamat *IP address*. Dan tabel 2 menjelaskan pentranslasian alamat *IP address* ke alamat yang mudah di ingat. Pemetaan ini dilakukan agar pada saat *user* melakukan pengaksesan ke sistem mail PLN SBAM tidak perlu menggunakan *IP address* cukup menggunakan alamat *Full Qualified Domain Name (FQDN)*.

Tabel 1. Forward zone mail system PLNSBAM

```

$TTL      86400
$ORIGIN   plnsbam.com.
@         IN  SOA      ns.plnsbam.com root.plnsbam.com (
                                42 ; serial (d. adams)
                                3H ; refresh
                                15M ; retry
                                1W ; expiry
                                1D ); minimum

mail.plnsbam.com.      IN  MX  0      202.45.67.10

@                     IN  NS      ns.plnsbam.com.
@                     IN  A       202.45.67.10
ns.plnsbam.com.       IN  A       202.45.67.10
mail.plnsbam.com.     IN  A       202.45.67.10
www.plnsbam.com.      IN  CNAME   plnsbam.com.
mailstats.plnsbam.com. IN  CNAME   plnsbam.com.

```

Tabel 2. Revers zone mail system PLNSBAM

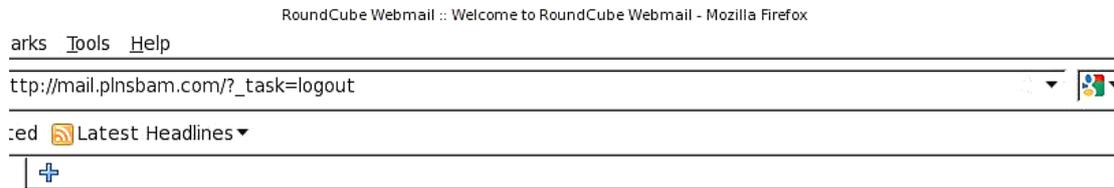
```

$TTL      86400
@         IN  SOA      ns.plnsbam.com.      root.plnsbam.com. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum

@         IN  NS      plnsbam.com.
10        IN  PTR     mail.plnsbam.com

```

Untuk mendapatkan hasil dari pengujian kegiatan pertukaran *email*, maka diterapkan sistem *email* kedalam bentuk *prototype* untuk dilakukan pengujian pertukaran *email*. Dari hasil tahap desain maka didapatkan bentuk *interface* yang sebenarnya tampak pada gambar 7, gambar 8 dan gambar 9.

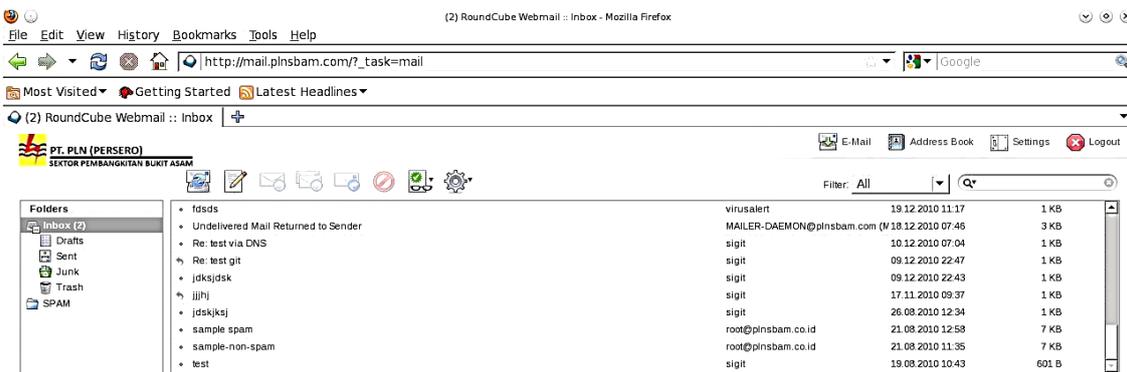


[Welcome Web Mail PLN Sector Bukit Asam]

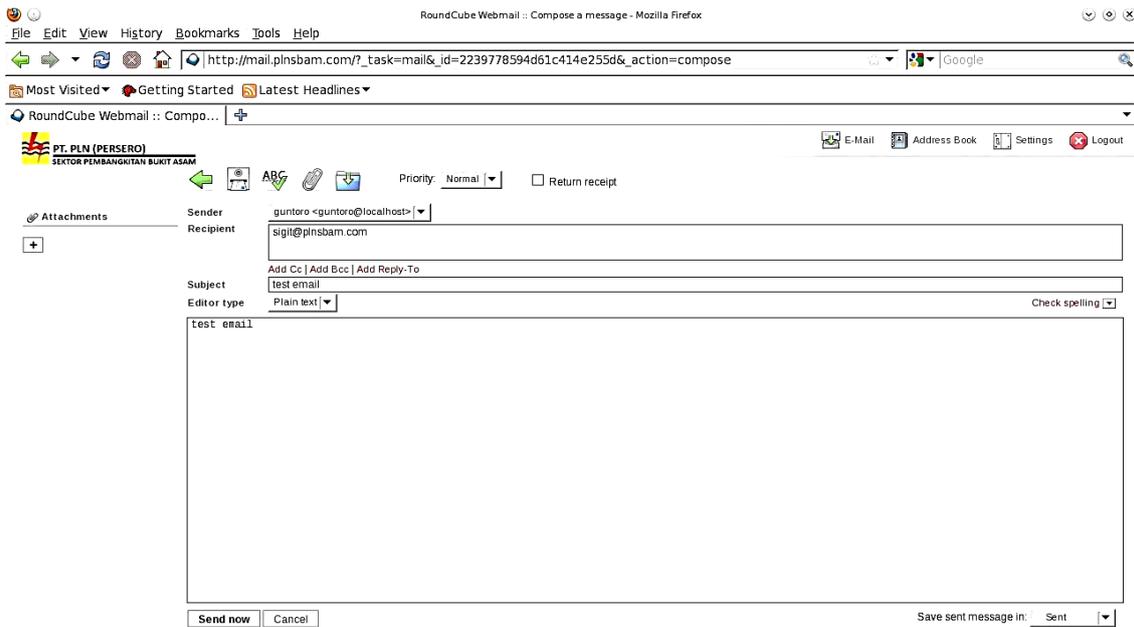
Username

Password

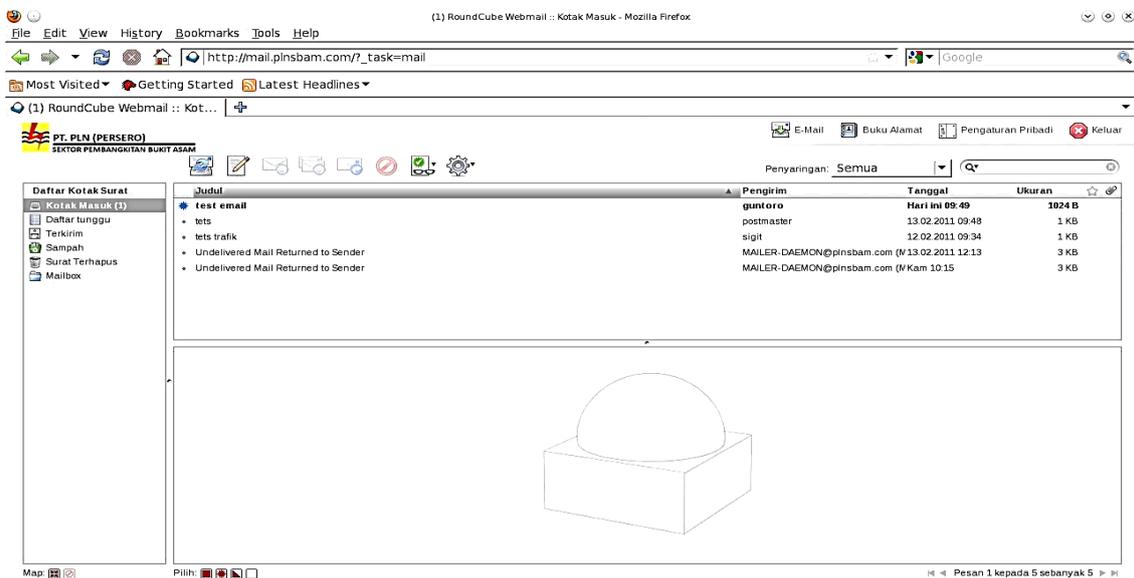
Gambar 7. Halaman *Login User*



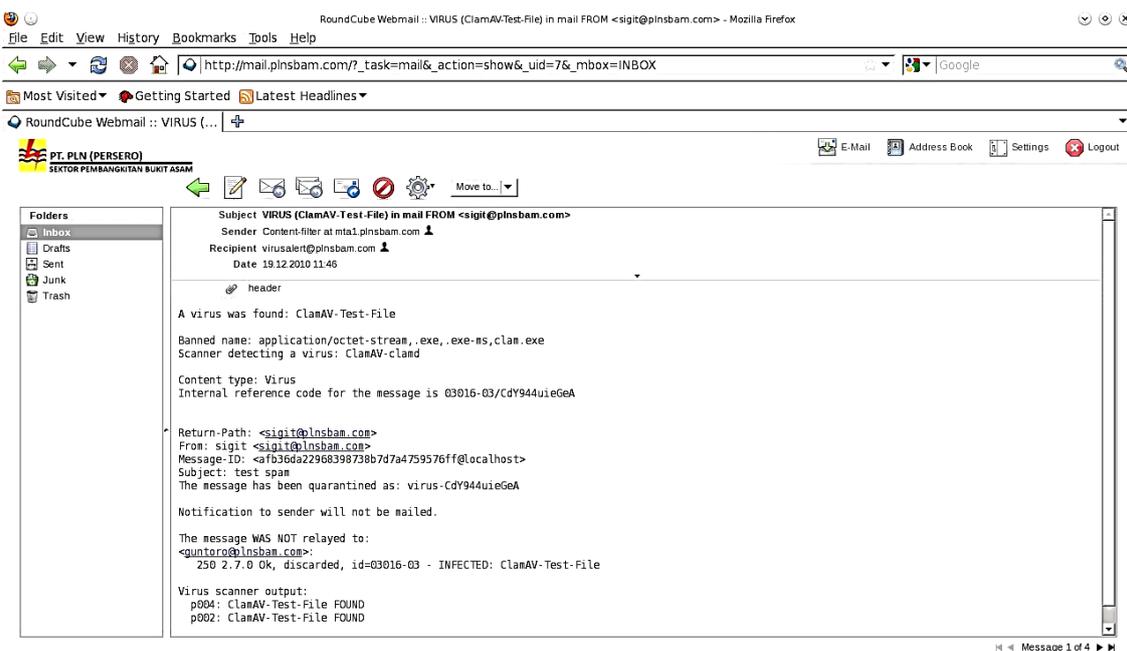
Gambar 8. Halaman mailbox user

Gambar 9. Halaman *compose message*

Dari penerapan antar muka sistem *email* kemudian dilakukan pengujian pengiriman *email* menggunakan pendekatan *blackbox testing*. Pengujian ini dilakukan untuk mendapatkan apakah *email* yang dikirim sampai dengan baik ke tujuan dan pemblokiran *email* yang mengandung *spam*. Dari hasil pengujian yang dilakukan didapatkan bahwa *email* yang dikirim sampai ke tujuan dan *email* yang mengandung *spam* berhasil di blokir dengan melibatkan serangkaian komponen *filtering system*. Tampak pada gambar 10 dan gambar 11.



Gambar 10. Pengujian pesan masuk



Gambar 11. Pengujian *spam*

Dari data *log system* pada saat dilakukan pengujian, maka didapatkan bahwa *email* yang mengandung *spam* berhasil di blokir oleh serangkaian sistem *security email* yang digunakan dan pesan yang dikirim oleh pengirim tidak akan tersampaikan ke tujuan karena sistem akan mengalihkan pesan yang mengandung *spam* ke *direktory quarantine* dan dihapus dari daftar antrian dan daftar pengiriman *email* yang ada di sistem.

Pemberitahuan adanya pesan yang mengandung *spam* akan dilaporkan ke administrator. tampak pada tabel 3.

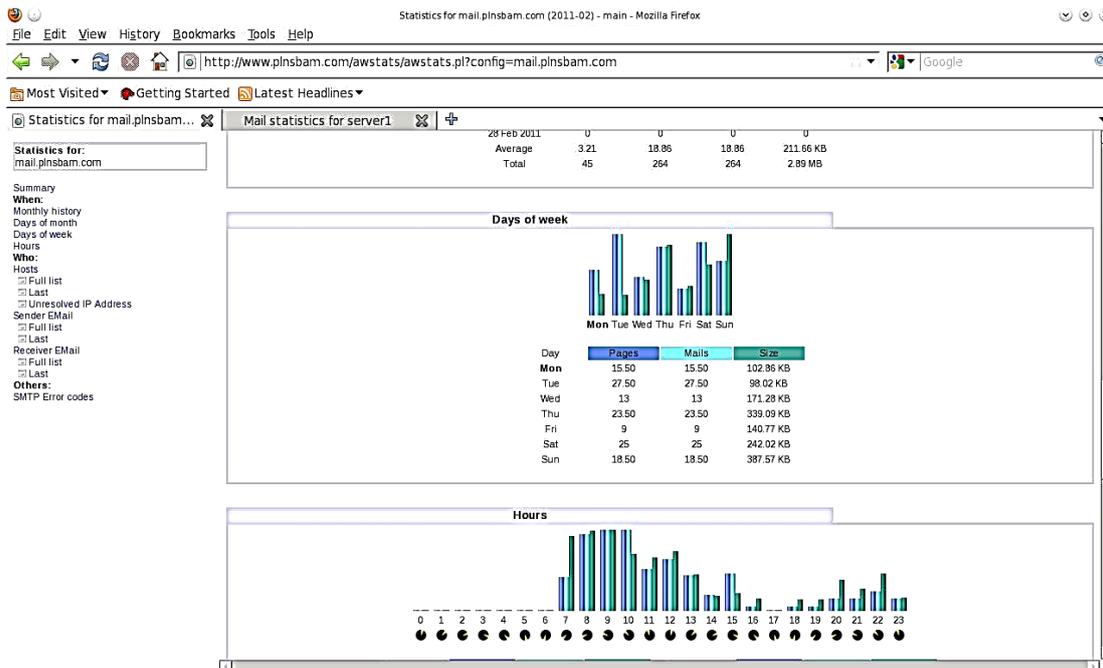
Tabel 3. Log pemblokiran pesan *spam*

```
Dec 19 12:21:39 plnsbam amavis[3016]: (03016-06) Blocked SPAM,
<guntoro@localhost.plnsbam.com> -> <sigit@plnsbam.com>, quarantine: spam-
OnwVF+PqvNIz.gz, Message-ID: <8d7d8c505e9fe31aeb387cf3df31b1d2@localhost>, mail_id:
OnwVF+PqvNIz, Hits: 999.999, size: 1943, 9282 ms

Dec 19 12:21:39 plnsbam postfix/lmtp[4127]: 7EFEB2449F9: to=<sigit@plnsbam.com>,
relay=127.0.0.1[127.0.0.1]:10024, delay=9.4, delays=0.07/0.01/0.01/9.3, dsn=2.5.0,
status=sent (250 2.5.0 Ok <sigit@plnsbam.com>, DSN suppressed (554 5.7.0 Reject,
id=03016-06 - SPAM))

Dec 19 12:21:39 plnsbam postfix/qmgr[3142]: 7EFEB2449F9: removed
```

Dari hasil pengujian *prototype* sistem *email* yang di bangun kemudian dilakukan pengujian dalam ruang lingkup pengguna jaringan untuk mendapatkan hasil yang lebih baik. Dari hasil pengujian yang dilakukan oleh pengguna, didapatkan bahwa sistem *email* yang di bangun bisa berjalan dengan baik tanpa ada kendala, hal ini diperlihatkan dengan banyaknya pengguna yang menggunakan fasilitas sistem *email* per waktunya yang digambarkan dalam bentuk grafik tampak pada Gambar 12.



Gambar 12. Statistik penggunaan email

#### 4. KESIMPULAN

Berdasarkan hasil pembahasan yang telah dijelaskan, maka Kesimpulan yang bisa diambil dalam sistem *email* ini, diantaranya.

1. Dari hasil pengujian yang dilakukan menggunakan pengujian *blackbox testing*, setiap komponen sistem berjalan dengan baik.
2. Dari hasil pengujian *spam* yang dilakukan komponen *security system* yang terpasang pada sistem *email* mampu memblokir atau memfilter setiap pesan masuk yang mengandung spam.
3. Sistem pencatatan pesan yang digunakan menyajikan kesamaan jumlah data yang didapatkan dari log email sistem. Sistem pencatat ini hanya menampilkan jumlah pesan saja tetapi tidak menyajikan pesan spam yang diterima oleh sistem email.
4. Sistem email yang dirancang dan diterapkan dapat berfungsi dengan baik seperti apa yang diharapkan dan aktifitas pertukaran pesan dapat berjalan dengan sempurna.

#### 5. SARAN

Berdasarkan kelemahan dari sistem yang dibangun. Adapun Saran yang bisa diberikan untuk pengembang berikutnya yaitu:

1. Sebaiknya sistem *email* yang akan dibangun melibatkan *multi server mail system*. Sehingga pada saat server utama berhenti melayani *service mail* dikarenakan *overload* maka dapat dialihkan ke *server backup*.
2. Perlunya perencanaan yang matang untuk menentukan sistem *email* yang digunakan terutama pada sistem keamanan sehingga kerahasiaan suatu pesan dapat terjaga.
3. Teknologi *email service* yang dibangun ditempatkan pada server fisik. Diharapkan pada pengembangan di masa mendatang bisa ditempatkan pada server virtual atau metode virtualisasi server sehingga tidak menghabiskan banyak *resource* dan *cost* yang akan dipergunakan.

---

DAFTAR PUSTAKA

- [1] Crocker, D., 2000, E-mail History, *http://www.livinginternet.com/e/ei.htm*, diakses 13 Januari 2015.
  - [2] Rachmadi, M., 2007, Integrasi Layanan Surat Elektronik, Pesan Singkat dan Telepon Sebagai Pendukung Mobilitas, *Algoritma*, No. 1, Volume 3, Hal 27-34.
  - [3] Lubis, H., Batubara, A. S., 2013, Implementasi Open Source Software Dalam Membangun Email Server, *Seminar Nasional Teknologi Informasi dan Komunikasi (SNASTIKOM 2013)*, Medan, 13-14 Maret 2013.
  - [4] Halim, F. A., 2007, Analisis dan Perancangan Exchange Server Cluster Untuk Mengatasi Kepadatan Antrian E-mail Studi Kasus PT. TC, *Seminar Nasional Sistem dan Informatika 2007*, Bali, 16 November 2007.
  - [5] Mujilahwati, S., Bahar, I., Muhtadin, S., 2013, Perancangan Mail Server Untuk Layanan Webmail dan Aplikasi Mail Compose Berbasis VB.Net 2010, *Jurnal Teknika*, No. 2, Vol. 5, Hal. 495-500.
  - [6] Bayu, T. I., Widiyari, I. R., Chandra, D. W., 2010, Penerapan Teknologi Virtualisasi Tingkat Sistem Operasi pada Server Linux Ubuntu 8.04 Menggunakan OpenVZ, *Jurnal Teknologi Informasi-Aiti*, No. 1, Vol. 7, Hal. 68-85.
  - [7] Mangunkusumo, I. E. S. W., Lumenta, A., Wowor, H., Sinsuw. H., 2013, Analisis dan Perancangan Keamanan Mail Server Zimbra pada Sistem Operasi Ubuntu 8.04, *e-journal Teknik Elektro dan Komputer*, No. 2, Vol. 2, Hal. 1-9.
  - [8] Barovich, G., 2013, Desain dan Implementasi Mail Server Berbasis Web Beserta Pengamannya pada PT. PLN (Persero) Sektor Pembangunan Bukit Asam Tanjung Enim, *Jurnal Mahasiswa Teknik Informatika STMIK Palcomtech*.
  - [9] Sukamto, R. A., Shalahudin, M., 2011, *Modul pembelajaran rekayasa perangkat lunak terstruktur dan berorientasi objek*. Modula, Bandung
-