LONTAR KOMPUTER VOL. 8, NO.1, APRIL 2017 **p-ISSN 2088-1541**
DOI : 10.24843/LKJITI.2017.v08.i01.p04 **e-ISSN 2541-5832**

Implementasi Pengamanan PGP pada Platform Zimbra *Mail* Server

Dandy Pramana Hostiadi[1], **Ida Bagus Suradarma**[2]

STMIK STIKOM Bali
Jl. Raya Puputan No. 86 Renon, Denpasar-Bali
[1]dandypramanahostiadi97@g*mail*.com
[2]suradarma@stikom-bali.ac.id

Abstrak

Elektronik mail merupakan model komunikasi yang sifatnya fundamental di era globalisasi, terbukti pada setiap bentuk registrasi data atau informasi, membutuhkan adanya data email (surat elektronik). Perkembangan teknologi komunikasi khususnya penggunaan email, membawa pengaruh terhadap tindak penyalahgunaan email seperti adanya aktivitas pencurian akun dan pemalsuan email. Keamanan komunikasi pada mail server seperti mail server ZIMBRA sudah terimplementasi dengan baik, seperti penggunaan ssl certificate, namun pengamanan tersebut masih standar. Isi email dapat terbaca dengan mudah (dalam teknik kriptografi dikatakan pembacaan plainteks) ketika user dan password telah diketahui oleh pihak ketiga maka. Metode pretty good privacy (PGP) diterapkan pada penelitian ini sebagai pengamanan komunikasi email, difokuskan pada isi email dengan mengenkripsi teks mail beserta attachment file. Mail engine yang digunakan yaitu *Zimbra mail server. Hasil dari penelitian menunjukkan bahwa pengamanan PGP mampu mengamankan isi email baik teks maupun attachment, dengan perbedaan size file attachment lebih besar pada penggunaan PGP dan mengubah header mail dari mail standar.*

Kata kunci: *Email, Zimbra Mail Server, PGP, Enkripsi.*

Abstract

Electronic mail is a communication model that is fundamental in the era of globalization. Proven on any form of registration data or information requires the presence of email address(electronic mail). The use of email itself cannot be separated from the abuse (such as stelling password and mail spoofing) from some parties so it needs security form in email communication. Communication security on mail server such as ZIMBRA mail server has been well-implemented, such as the use of ssl certificate. But the security is still standard. So, when user and password have been found out by third party, email content will be read easily (in cryptography technique it is called plaintext reading). On research that was conducted with pretty good privacy (PGP) method email communication security was focused on the email content by encrypting mail text along with the attachment file. In a study conducted, using the mail engine *Zimbra mail server. Result of research shows that PGP security is able to secure email content whether the text or the attachment, showing difference of attachment file size is bigger on PGP using and change mail header from the standard mail.*

Keywords: *Email, Zimbra Mail Server, PGP, Encryption.*

1. Pendahuluan

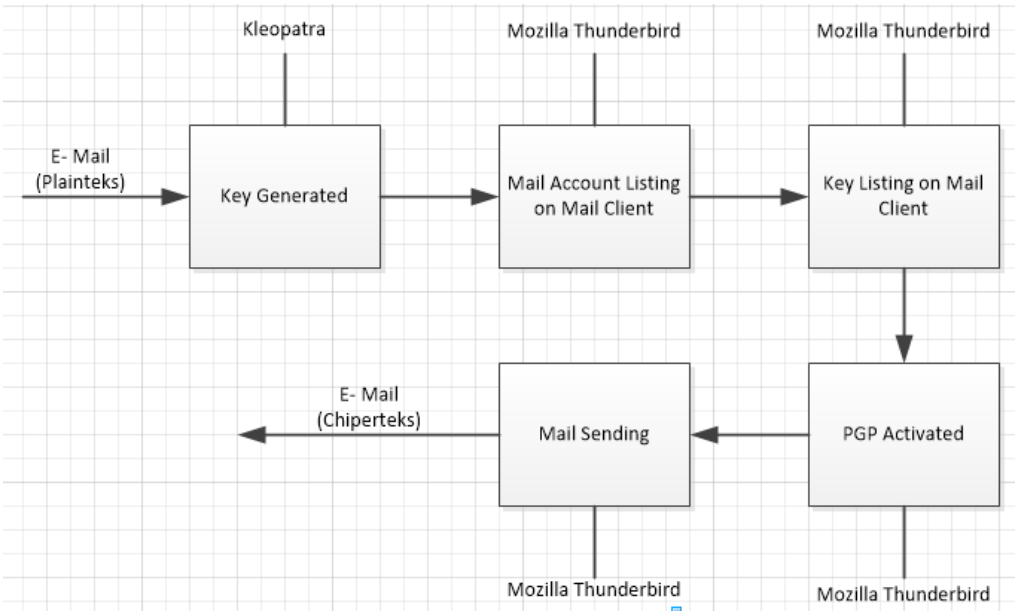
Email merupakan bentuk komunikasi yang dikatakan bersifat fundamental di-era globalisasi. Hal ini terlihat bahwa saat ini hampir pemanfaatan komunikasi selalu mensyarakatkan pencantuman alamat *email* (*electronic mail*). Contohnya adalah pada registrasi sosial media dimana pada form registrasi mewajibkan mencantumkan alamat *email*. Komunikasi di beberapa perusahaan maupun pemerintahan dalam komunikasi jarak jauh lintas pulau maupun negara juga memerlukan alamat *email*. Jumlah organisasi yang tak terhitung jumlahnya di seluruh dunia terus mengubah metode mereka dalam hal komunikasi yang dulu menggunakan kertas (*hardcopy*)

menjadi salah satu factor utama perubahan menuju basis komputerisasi (*email*) berdasarkan sistem untuk menyimpan data penting dan informasi. Hal ini dapat disimpulkan bahwa *email* (*email*) merupakan data / informasi primer bagi pelaku komunikasi di era global.

Seiring perkembangan penggunaan *email* sebagai bentuk komunikasi, tidak terlepas dari adanya pihak yang menyalahgunakan penggunaan *email* dan mengarah pada pelanggaran hukum. Seperti adanya penipuan dengan penggunaan account *mail* palsu atau adanya pencurian password dan pembajakan account *email* [1]. Secara teori pengamanan terhadap komunikasi perlu dilakukan. Mengamankan *email* adalah sesuatu yang harus dilakukan oleh pengguna sendiri, karena mereka adalah salah satu yang akan benar-benar menjadi tanggung jawab pengirim dan penerima pesan. Terdapat beberapa teknik pengamanan komunikasi dalam jaringan termasuk didalamnya adalah komunikasi *email* seperti teknik kriptografi, dengan algoritma yang berbeda [2]. Sebagai contoh adalah *mail* server Zimbra yang sudah menerapkan pengaman certificates ssl. Zimbra *mail* server sendiri merupakan *mail engine* yang sudah banyak digunakan di beberapa perusahaan dengan fitur yang fleksibel dan simple untuk digunakan serta bersifat *open source* [3]. Namun keamanan dengan certificate ssl belum menjamin sepenuhnya keamanan pada *mail server*. Seperti halnya pencurian account *mail*, ketika pihak yang tidak berkepentingan berhasil mendapatkan user *mail* dan password, maka dengan mudahnya membaca isi *email* dan mengetahui informasi yang sifatnya rahasia dalam *email*. Untuk mencegah hal tersebut maka keamanan komunikasi *email* perlu ditingkatkan dan salah satu caranya adalah dengan pengamanan *PGP* (*Pretty Good Privacy*). Dengan PGP pengamanan komunikasi *email* memfokuskan pada pengamanan isi *email* (mengantisipasi pembacaan isi *mail* secara mudah oleh pihak yang tidak berkepentingan) termasuk di dalamnya *adalah file attachment*. Pada Penelitian yang dilakukan, dimana mengimplementasikan pengamanan PGP pada Zimbra *mail* server akan melihat sejauh mana pengamanan yang dilakukan dengan melihat dari hasil PGP, analisa terhadap *file attachment* dan header *email*.

2. Metodologi Penelitian

Pada penelitian yang dilakukan alur metodologi penelitian digambarkan pada skema berikut :



Gambar 1. Alur Pengiriman PGP

Dari gambar 1, secara garis besar, pengiriman PGP melalui beberapa tahap, yaitu :

a. *Key Generated*

Key Generated adalah tahapan dimana *key* PGP di buat dan nantinya digunakan dalam pengamanan pengiriman e*mail*. *Key* yang dihasilkan dalam pembuatan kunci, memiliki

fitur untuk pembatasan waktu penggunaan, yang bertujuan untuk membatasi masa penggunaan *key* yang dibuat terhadap sesi pengiriman *email*. Setelah membuat kunci private diperlukan passphrase yang nantinya digunakan untuk mendekrip pesan saat mengirimkan dan di sisi penerima saat membaca pesan yang diterima. Kunci private yang telah dibuat harus dimiliki juga oleh penerima, dapat dilakukan dengan manual atau pengiriman konvensional ke penerima pesan. Perlu diingat bahwa dari sisi penerima apabila menggunakan *key* list yang berbeda (bukan *key* yang sama antara pengirim dan penerima) maka *key* yang digunakan di sisi penerima pesan tidak akan dapat digunakan sebelum *key* yang sama dikirimkan oleh pengirim

b. *Mail* Account Listing Pada *Mail* Client
Mail account yang digunakan adalah *mail* account dengan engine *mail* Zimbra. *Mail* account dengan engine Zimbra menggunakan konfigurasi SMTP IMAP. Adapun dalam penelitian yang dilakukan, untuk pembacaan *email* dan pengiriman yang mengenkripsi dengan *key* private yang dibuat menggunakan aplikasi Mozilla thunderbird. *Mail* account yang ada di list ke dalam *mail* client Mozilla Thunderbird

c. *Key* Listing pada *Mail* Account Zimbra
Hasil *Key* yang dibuat dalam hasil file extention .asc, di import ke dalam *mail* client. *Key* yang diimportkan harus sesuai dengan identitas yang ada saat men-generate *key* awal
Pada tahap ini harus dipastikan bahwa *key* yang digunkaan oleh pengirim dan penerima adalah sama. Karena penggunaan *key* yang berbeda maka akan berdampak pada pembacaan *email* yang diterima dimana e*mail* yang terbaca dalam bentuk chiperteks (tersandikan dan tidak terbaca)

d. PGP Activated
Pada tahapan ini, PGP diaktifkan dengan memberikan autorisasi pada teks *mail*. Bentuk pengaktifan dengan cara mencetak pilihan *button enigmail* dalam pengiriman. Aktifasi yang dilakukan juga berlaku pada pengiriman file attachment. Dan sign digital modul teraktifkan. Apabila telah diaktifkan maka proses akhir adlaah melakukan pengiriman e*mail*

e. *Mail* Sending
Pengiriman e*mail* yang dilakukan adalah dengan mengirimkan teks *mail* dan file attachment. Pengujian dan analisa dilakukan dengan membandingkan hasil pengirman e*mail* oleh sisi penerima *email*. Dimana untuk pengiriman teks *mail* dibandingan dengan pembacaan *mail* teks dengan aplikasi browser standar dan dibandingkan dengan pembacaan teks *mail* menggunakan aplikasi Mozilla Thunderbird. Untuk file attachment dilakukan dengan menganalisa besar size yang diciptakan dari hasil pengamanan PGP serta mengukur sebera jauh perbedaan yang muncul. Selain itu juga dilakukan penganalisaan terhadap *mail* header yang ada di kedua penerimaan baik menggunakan standar pembacaan berupa browser default dan aplikasi *mail* client.

3. Kajian Pustaka

a. *Mail* Server Zimbra
Mail server (juga dikenal sebagai sebuah *mail* transfer agent atau MTA, *mail* router atau *mail*er Internet) adalah sebuah aplikasi yang akan menerima *email* masuk dari pengguna lokal (orang-orang dalam satu domain) dan jarak jauh pengirim dan meneruskan *email* keluar untuk pengiriman. Sebuah komputer yang didedikasikan untuk menjalankan aplikasi tersebut juga disebut sebagai *mail* server [4]. Microsoft Exchange, q*mail*, Exim dan send*mail* adalah lebih umum di antara program-program server *mail*.
Zimbra adalah sebuah produk groupware yang dibuat oleh Zimbra, Inc yang berlokasi di Palo Alto, California, Amerika Serikat. Pada masa awal- awalnya perusahaan ini di beli oleh Yahoo! tepatnya pada bulan september 2007. Zimbra pada dasarnya sekelas dengan aplikasi Microsoft Exchange Server. Bedanya, Zimbra tersedia dalam 2 edisi, yaitu Open source Edition dan Network Edition. Dewasa ini zimbra merupakan software open source *mail* server yang mulai banyak digunakan dengan kemudahan instalasi dan

management. Di masa yang akan datang zimbra dapat menjadi suatu aplikasi *mail* server yang paling banyak digunakan seperti postfix, send*mail* dan q*mail*. Berikut aplikasi open source yang digunakan Zimbra Collaboration Suite yang sudah merupakan aplikasi standar yang dipakai di dunia industry [5]:

- Jetty, aplikasi server web yang menjalankan aplikasi Zimbra. • Postfix, aplikasi open source MTA (*Mail* Transfer Agent) yang menjalankan *email* server Zimbra.
- OpenLDAP, aplikasi open source sebagai Lightweight Directory Accses Protocol (LDAP) yang berguna untuk autentikasi user.
- MySQL, aplikasi database
- Lucane, aplikasi open-source power full text index dan search engine.
- Anti-Virus and anti-spam, aplikasi open source yang terdiri dari : Clamav anti virus scanner yang melindungi file dari serangan virus, SpamAssassin *mail* filter yang mengidentifikasi adanya Spam dan Amavisd-new sebagai interface antara MTA dengan yang lain.
- James/Sieve filtering, membuat filter untuk *email*

b. Kriptography

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima melalui mekanisme transmisi komunikasi tanpa adanya gangguan dari pihak ketiga. Menurut Bruce Scheiner dalam bukunya "Applied Cryptography", kriptografi adalah ilmu pengetahuan dan seni menjaga pesan tetap aman (secure).

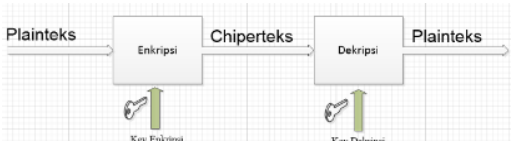
Prinsip - prinsip yang mendasari kriptografi yakni :

- *Confidelity* (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak - pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk di baca dan dipahami.
- *Data integrity* (keutuhan data) yaitu layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).
- *Authentication* (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak - pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
- *Non - repudiation* (anti - penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya)

Istilah - istilah yang digunakan dalam bidang kriptografi :

- Plaintext (M) adalah pesan yang hendak dikirimkan (berisi data asli).
- Ciphertext (C) adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi.
- Enkripsi (fungsi E) adalah proses pengubahan plaintextmenjadi ciphertext.
- Dekripsi (fungsi D) adalah kebalikan dari enkripsi yakni mengubah ciphertext menjadi plaintext , sehingga berupa data awal/asli

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Proses enkripsi mengubah plaintext menjadi ciphertext (dengan menggunakan kunci tertentu) sehingga isi informasi pada pesan tersebut sukar dimengerti



Gambar 2. Alur Enkripsi dan Dekripsi

Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu yang berisi elemen teks terang /plaintext dan yang berisi elemen teks sandi/ciphertext yang ditunjukkan pada matematis berikut :

Enkripsi :

$$E\,(M) \;=\; C \qquad (1)$$

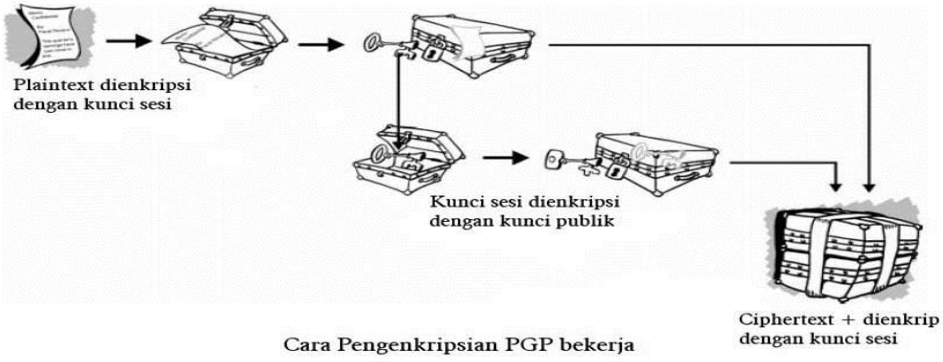
Dekripsi :

$$D\,(C) \;=\; M \;\text{atau}\; D\,(\,E\,(\,M\,)) \;=\; M \qquad (2)$$

Enkripsi dan dekripsi merupakan fungsi transformasi antara himpunan-himpunan tersebut. Apabila elemen-elemen teks terang dinotasikan dengan M, elemen-elemen teks sandi dinotasikan dengan C, sedang untuk proses enkripsi dinotasikan dengan E, dekripsi dengan notasi D. Dalam skenario sistem keamanan lainnya seperti steganografi, sebelum proses pengengkripsian dilakukan, pengirim harus memilih pesan yang sesuai dengan carrier message (contohnya gambar, video, audio, teks) dan pemilihan pesan crutial yang efektif disamping penggunaan password yang kuat (diketahui oleh penerima).

c. *Pretty Good Privacy* (PGP)

PGP (*Pretty Good Privacy*) adalah suatu metode program enkripsi informasi yang memiliki tingkat keamanan cukup tinggi bersifat rahasia dengan menggunakan "*Private-Public Key*" sebagai dasar autentifikasinya sehingga jangan sampai dengan mudah diketahui oleh orang lain yang tidak berhak. PGP membuat sebuah session *key*, dimana sebuah kunci rahasia pada saat itu. Kunci adalah sebuah bilangan acak yang dihasilkan dari gerakan acak dari mouse dan tombol yang anda tekan. Session *Key* ini berkerja dengan sangat aman, algoritma enkripsi konvesional yang cepat untuk meng-enkrip plaintext. Hasilnya adalah berupah chiper text. Sekali data dienkripsi, lalu session *key* ini dienkripsi lagi menggunakan kunci publik penerima. session *key* yang terenkripsi kunci publik *key* penerima dikirim dengn chipertext ke penerima. Proses deskripsi bekerja sebaliknya, Penerima menerima pesan lalu membuka pesan tersebut dengan kunci privatnya, namun pesan tersebut masih terenkripsi dengan session *key*. Dengan Menggunakan PGP, penerima mendekrip chipertext yang terenkripsi secara konvensional. Kombinasi dari 2 metode enkripsi menggabungkan kehandalan dari enkripsi kunci publik dengan kecepatan pada enkripsi konvensional. Enkripsi Konvensional kuarang lebih 1000x lebih cepat dari enkripsi kunci publik. Jadi enkripsi kunci publik memberikan sebuah solusi pada distribusi kunci dan masalah transmisi data. Dengan menggunakan keduanya, perfoma dan distribusi kunci dapat ditingkatkan tanpa mengorbankan sesuatu dalam keamanan.



Gambar 3. Alur Kerja PGP

Prinsip kerja dari PGP itu sendiri adalah :

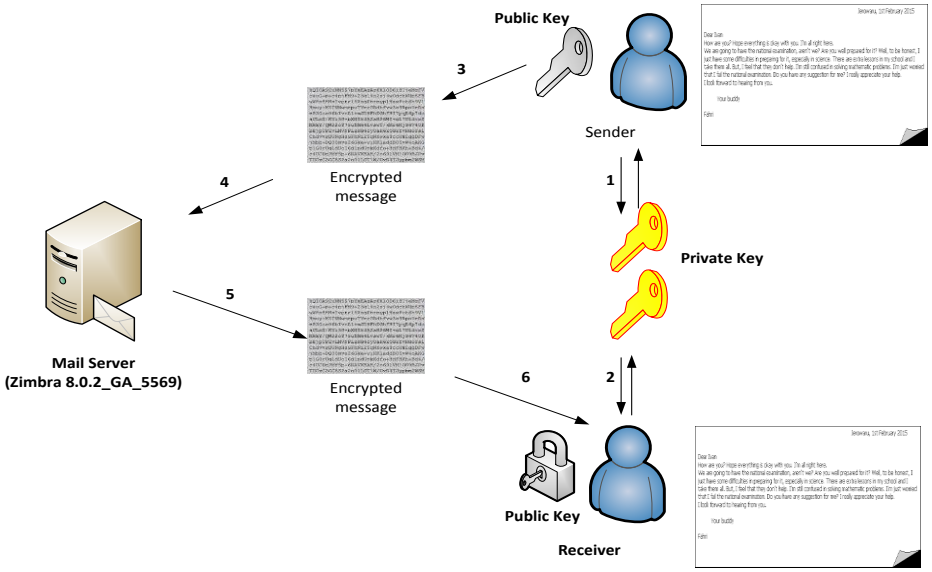
- PGP menggunakan teknik yang disebut Public-*key encryption* dengan dua kode yang saling berhubungan secara intrinsik, namun tidak mungkin untuk memecahkan satu dan yang lainnya.
- Jika membuat suatu kunci, secara otomatis akan dihasilkan sepasang kunci yaitu public *key* dan secret *key*. Pengirim dapat memberikan public *key* ke manapun tujuan yang diinginkan, melalui telephone, internet, *key*server, dsb. Secret *key* yang disimpan pada mesin pengirim dan menggunakan messager decipher akan

dikirimkan ke penerima oleh pengirim di sisi yang lain. Jadi yang akan menggunakan public *key* (yang hanya dapat didekripsi oleh oleh secret *key*), mengirimkan messages kepada penerima , dan penerima akan menggunakan secret *key* untuk membaca pesan dari pengirim.

- PGP menggunakan dua kunci yaitu kunci public (proses enkripsi) dan private (proses deskripsi). Menggunakan dua kuci tersebut dikarenakan adanya conventional crypto, disaat terjadi transfer informasi kunci, suatu secure channel diperlukan

4. Hasil dan Pembahasan

Pada penelitian yang dilakukan, sebelum melakukan pengujian berdasarkan metodelogi penelitian yang telah dibahas sebelumnya, maka dilakukan perencanaan dalam bentuk pembuatan rancangan arsitektur penelitian. Rancangan yang dimaksud adalah sebagai berikut.

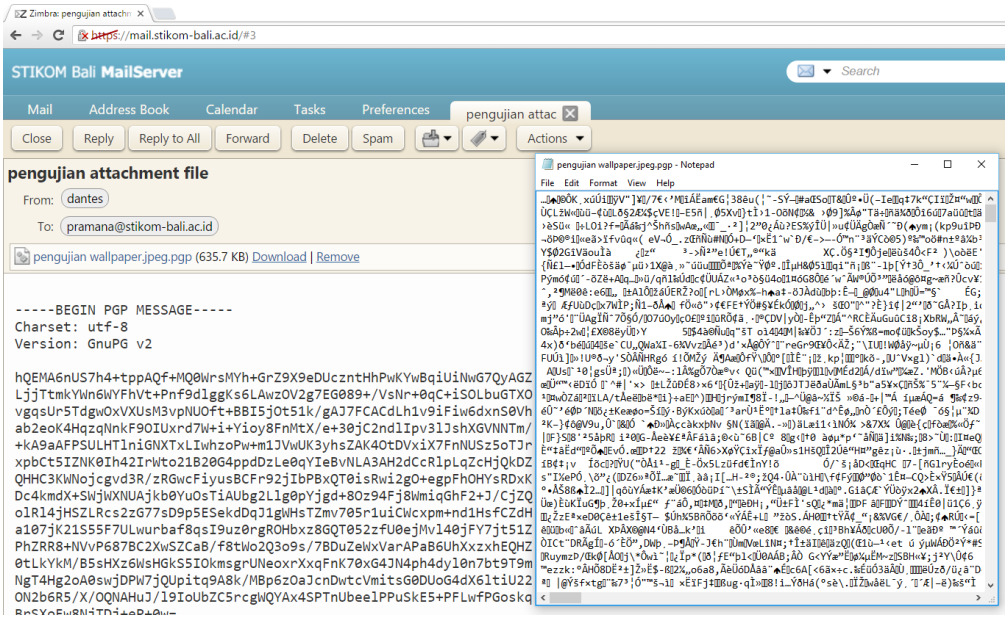


Gambar 4. Rancang Bangun Arsitektur PGP

Seperti yang sudah dibahas sebelumnya bahwa pengamanan komunikasi e*mail* menggunakan PGP teknik diawali dengan adanya kepemilikan kunci private. Kunci private ini dimiliki oleh kedua aktor yaitu pengirim e*mail* dan penerima e*mail*. Kunci ini harus ada sebelum pengiriman e*mail* dilakukan (proses 1 dan 2 pada gambar 5). Setelah kunci private dikirimkan oleh kedua aktor, dilakukannya pengiriman pesan oleh pengirim. Pesan yang dikirimkan dienkripsi sebelum dikirimkan ke penerima. Sehingga pesan dikirimkan adalah dalam bentuk chiperteks (pesan tersandikan). Adapun proses penyandian yang dilakukan adalah menggunakan kunci *public*. Pengiriman dilakukan di atas platform mesin zimbra. Dari sisi penerima, penerima akan menerima dalam bentuk chiperteks. Chiperteks yang diterima akan di dekrip kembali menggunakan kunci private yang telah dimiliki di awal komunikasi. Dengan kepemilikan kunci private, maka pesan dapat ditampilkan kembali. Apabila kunci private yang digunakan untuk mendekrip tidak sama dengan kunci private pengirim, maka pesan tersebut tidak dapat terbaca. Pengujian yang dilakukan dalam penelitian adalah dengan membandingkan mekanisme pengiriman dan penerimaan e*mail* antara pengiriman *email* standard dan pengiriman yang menggunakan teknik PGP. Hasil pengujian pertama adalah membandingkan bentuk PGP yang digambarkan sebagai berikut :

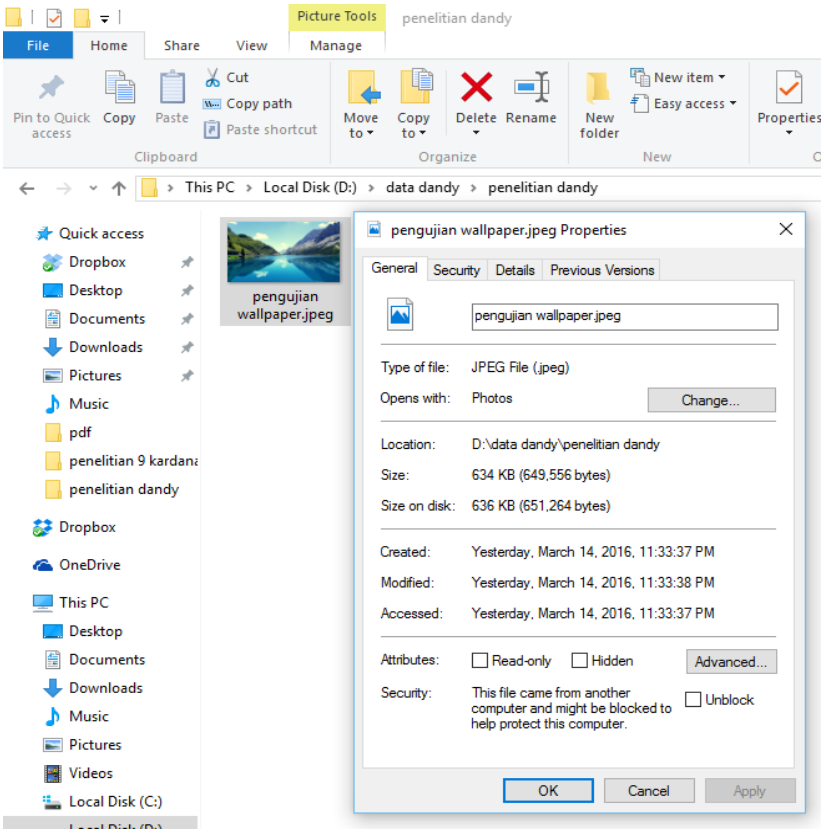
Gambar 5. Hasil PGP

Dari Gambar 5 dapat dilihat bahwa hasil pengamanan dengan PGP merubah teks dalam *email* menjadi teks yang tidak dapat terbaca atau dalam bentuk sandi yang disebut chiperteks. Pengujian kedua dilakukan dengan membandingkan file attachment pada pengiriman *email*, Setelah dilakukan pengiriman file attachment dianalisa dari bentuk penerimaan dan perubahan size ukuran file attachment. Hasil penerimaan file PGP dengan browser dan pembandingan ukuran size ditunjukkan pada gambar berikut :



Gambar 6a. Pembacaan *mail* dengan browser

p-ISSN 2088-1541
e-ISSN 2541-5832



Gambar 7b. Perbandingan file attachment

Hasil pembandingan kompresi data dari pengiriman PGP diambil dari pengujian pengukuran file pengiriman yang dicek berdasarkan nilai hash dan ukuran data (file size). Kemudian dilihat pembanding kompresi dengan PGP sehingga mendapatkan nilai kompresi dari data yang dikirimkan menggunakan PGP. Contoh data ditunjukkan pada tabel berikut :

Tabel 1. Tabel Uji Pembanding Data

No	Tipe File	File Pengirim (file asli)			File Penerima tanpa PGP			File Penerima dengan PGP		
		MD5 Hash		Size	MD5 Hash		Size	MD5 Hash		Size
		String	File	KB	String	File	KB	String	File	KB
1	ppt	3128e8da7038f8b714f4db398ea6b8fb	3f8d1981b51c4c7c7a8c27d55a58e835	1702	88f7c635113a679ef9584c96616854	9f7f921397568c7b75e5bbcf36957897	982	3f8d1981b51c4c7c7a8c27d55a58e835	3f8d1981b51c4c7c7a8c27d55a58e835	1702
2	docx	a3927eaa04e2a74d3ffe87445a95cb52	8f36188c2dd1cbb73cb7f754d69bcc57	86	9d1993f57dce776cd7e8e203a4f2bd23	1bd177effd7269d699cf108e796f1c2f	86.4	8f36188c2dd1cbb73cb7f754d69bcc57	c86f286c1a8ecb66397e39430024a64e	86
3	xlsx	63690d572f37c985533307d8b3a20bf3	090a9cc95004fcf29784329e89582014	9.01	ad7c37ac72e7a661102f60887a3ad0eb	54e98a32f081a5a1cd8039e8078849d6	9.34	32bc31b83cca4565c2c850ee5da68e9a	090a9cc95004fcf29784329e89582014	9.01
4	doc	0c15271c044c9db01c749d60b29de087	8a10678dd0fbec0cc68ab600368daafa	6880	68ca60b635ed54bc5c304ff46285ce91	df05281745a077a173a61b867393f212	3914	0eb91f33c8f7bdadf0314c5761250d2c	8a10678dd0fbec0cc68ab600368daafa	6880
5	pdf	671266150a083ef84c22e80ce2017d6c	cd3c6f08d6a70829950ee54fe8717095	108	0bb087ae605113c914989630 1defef58	bc153e8514c68febe74660a8d84ee919	104	6ae10672d3ef724d073dea63d37b3a41	cd3c6f08d6a70829950ee54fe8717095	108
6	ipa	7968ccffa88cddaa983ebc244453ff7e	29056a9c6c37fcb196a14f7697dc83e8	14626	771eb3da89f17e25a7d0a3e85370dc7e	448bcca5d0890d965b938b20ce5fcf0f	14627	31e6c2018c7133bc6f849e9afdc688e5	29056a9c6c37fcb196a14f7697dc83e8	14626
7	exe	13a37135ab04f4a7ef6a741b5531845e	a15923362cc6a42ebb3376c0c8d7c4cd	329	b002eff4afbcb7dc7158aa2585e5e108	0aa81822f5bbdc3e44b9378ebcac3b6f	148	51cbd3327e940cc9f26732 05eb4ac425	a15923362cc6a42ebb3376c0c8d7c4cd	329
8	xpi	0203eba59b160b8b5d5ee8ad2e10edf1	3216e114290a4b79b3643c5b24b7612f	30	b6af539c8f6eecf7f33b02534defc1bb	c2c4322553018c45ebdd3de7b8d97e5b	31	ddecb52fa9acdecdd5851e55e11c12af	3216e114290a4b79b3643c5b24b7612f	30
9	apk	86159b78d7b881d050420d0cd2346938	445e7b45daffd56c607d625600ec0e15	8567	41cea4d5961bf8c27bfbd41c4f101a61	22870062d61895761f66ad31215d1cd4	8568	e89bd0341c16e3e5e56239 32b547c112	445e7b45daffd56c607d625600ec0e15	8567
10	sisx	670f670b55c574a3fa09c3fe8fb3c16d	2cfe5076bba69b8b83b542d9bb3f58a6	1856	eb60f4ee39944ba826b38e2c66ed8460	42fb0d3b414385717e80af40c123cf88	1857	d9493e3c83a213dd3b8af8a27f443f74	2cfe5076bba69b8b83b542d9bb3f58a6	1856

11	svg	bbb9d21b7b9501735ae04f09c2fdf2dc	931eb597b54aa4846ce852d9b6752cec	6	b820bb614b0037f75e99e2257aff3fc9	07e4178c45e551766da36242d1a9e80b	3	d97cd625377199c170a8c681abec716f	931eb597b54aa4846ce852d9b6752cec	6
12	bmp	6baad7b80d3134c82d73a775fa7c98d1	2bc6646e18220759031056cdeb8783de	149	3c0d3f1955caf305f213ebf1240f2bb4	5b13aea9921298bf6ac6e7a7682c1d9e	100	69777863ecf8436281bf229a14e0b468	2bc6646e18220759031056cdeb8783de	149
13	ico	99cb115c43dcd3289c5832e062ca6446	6892a234406a1b5066ffd470451039d5	126	42199a5c2027d1fd9475970e83a0d57c	b5a0f7728d5dfbd7203221836e91015	75	b635691a98e06494de7cb1f3c3d99712	6892a234406a1b5066ffd470451039d5	126
14	jpg	fa8ecc21879ee4128773d01f525e411d	3b5672336f64ffe295187c4f394056dc	8	da00e4a807fb7931643a010893991dbb	4168fdfb90bd42718a7872530e286077	9	0fac958e0ec5c47a68e8c08b7f9143a8	3b5672336f64ffe295187c4f394056dc	8
15	png	092142e9ddcf8189ac7b8f8a811f9f35	4d8cc8b6dc0662dbbfac41bf323b053b	113	7f1e0bfe9c7baaea612cb614468fe0806	378f4d1c9c5917465521f9734e91141f	108	e820e262449bf783ab32b3a2fca9e619	4d8cc8b6dc0662dbbfac41bf323b053b	113
16	aiff	24ad56758ccd4c5b68c13df2ce34aa2a	52ce4540e93d31056c9f91c6de65fa9d	10450	463998a622637371b4bb31d917e55f92	bdd719f6d0e5ea7475f38cfad9879d39	7552	4577e96c57f04c9500236290fe2808c1	52ce4540e93d31056c9f91c6de65fa9d	10450
17	ogg	62dcafb3eedd4f5f957ce1bf47875694	43b2003306a9108450aaac471dc54c3d	3166	d2c008bf894cc0577bb45866e69e3078	87bae720c1eb6a4fbde9171cd68e5619	3161	318fabbb59f36e26ce4501f7801346c1	43b2003306a9108450aaac471dc54c3d	3166
18	wav	84f7082a09c9590acfd	84f7082a09c959b1374d73ca9080acfd	10451	2caa549dddd2f5df415b67d32c43d6ef	5b48452e958220abde7a5239be141340	7541	0fd578ca1147bde7aa7713391a0c674e	84f7082a09c959b1374d73ca9080acfd	10451
19	flac	b11a2bf636bec8e464a4b827763ae327	b17c38c8f95d9b3fc920452721be1352	335	1edc408e50c01d9a5e3e04135206c0dc	9a10821bd960eb9391763550cf36e261	330	2e0cc04cfeb6cf397b7089f6e3031494	b17c38c8f95d9b3fc920452721be1352	335
20	mp3	35f5bfa3aea70ecdc0877d5565056b1b	1f4759754c62c5e0ac6700364a77ebc1	78	1744efc106c9b9d1ca24ccbf74e17c9c	26374c2aea9500862d443e8bc934e4dd	46	7d56bbd1d532ed7bb52bae0967b397f1	1f4759754c62c5e0ac6700364a77ebc1	78
21	avi	d986c1f6314171d8445f1f73f5a30368	58231153bec6ec04f10d27c921cf57e1	2309	e190e0858704ae3dcef7198623c9d385	78836a590132baf80dafc3cbc5e90ee7	2080	330a48dd5c9e62f1a36a8267bdcd4f1c	58231153bec6ec04f10d27c921cf57e1	2309
22	flv	f5314b5e949539d47bcba22cad33a0e2	34534260d41cd44f899163e1e08540ad	1180	b03b12dd08858fc9a44a14e4f5551783	da7c0d94c0072637cd338d6826a86c21	1148	96f3cd3becaa5b72491fd57c5d5d661b	34534260d41cd44f899163e1e08540ad	1180
23	mpg	ef6cffbc711e8f4b4bf075d14653f3d3	05c41f69073bab98617daf04b760e99f	1648	2f9240725da20009f29ec2b39114582d	f6ce3aefbc61bf3b4cc0aa28284a7034	1377	6acf7baa6260b0e36fe3486ee859c9b3	05c41f69073bab98617daf04b760e99f	1648
24	3gp	5333bc735954c9b7162ff66f1c3b6c69	e44a090d0a30c7e6fd97ec36721b68bc	6933	d0b31bc041393c92d376527d2f386fd7	324bf3ec06a9ea81f62efdb62006a768	6825	f0ddc5fa0a0a760bab0823799609081f	e44a090d0a30c7e6fd97ec36721b68bc	6933
25	mp4	644190478f6de9b3d4e947e1b19fb657	adc2d5f563151624d33e82684b2471db	643	97c63fa1957dd6c4f221e33abd828e16	883ac3cb4a5422cc5eb06a7f97ae8439	614	7989c199656542d8a8301f196e5dba37	adc2d5f563151624d33e82684b2471db	643

Dari Tabel 1, dapat dilihat bahwa dari sisi penerima memiliki perbedaan ukuran data dari hasil pengiriman PGP. Prosentase yang dihitung adalah dengan membandingkan selisih antara penerimaan dengan *key* PGP dan tanpa *key PGP* berbanding file asli penerimaan. Dengan penghitungan sebagai berikut :

$$\text{Prosentase perubahan} = \frac{Size\ \ file\ \ Penerima\ (dengan\ key\ PGP) - Size\ file\ Penerima\ (Non\ PGP)}{Ukuran\ File\ Pengirim} \quad (3)$$

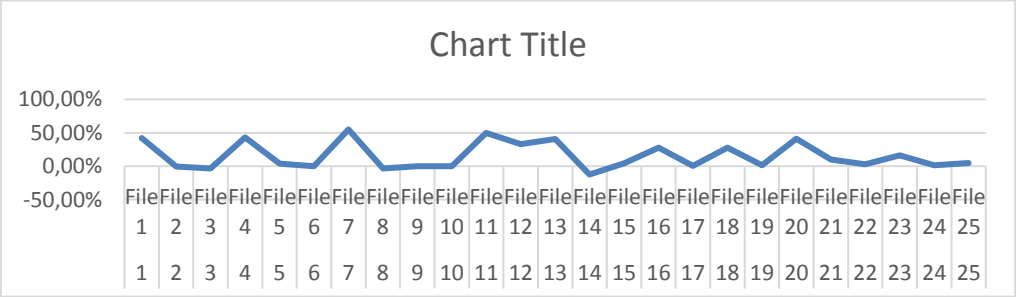
Hasil prosentase dapat ditunjukkan pada Tabel 2 berikut :

Tabel 2. Hasil Pengukuran Perbandingan

No	Tipe File	File Asli	Tanpa PGP	Dengan PGP	Perubahan	
		Size KB	**Size KB**	**Size KB**	**%**	**Keterangan**
1	File 1	1702	982	1702	42.30%	lebih kecil dari file asli
2	File 2	86	86.4	86	-0.47%	lebih besar dari file asli
3	File 3	9.01	9.34	9.01	-3.66%	lebih besar dari file asli
4	File 4	6880	3914	6880	43.11%	lebih kecil dari file asli
5	File 5	108	104	108	3.70%	lebih kecil dari file asli
6	File 6	14626	14627	14626	-0.01%	lebih besar dari file asli
7	File 7	329	148	329	55.02%	lebih kecil dari file asli
8	File 8	30	31	30	-3.33%	lebih besar dari file asli
9	File 9	8567	8568	8567	-0.01%	lebih besar dari file asli
10	File 10	1856	1857	1856	-0.05%	lebih besar dari file asli

11	File 11	6	3	6	50.00%	lebih kecil dari file asli
12	File 12	149	100	149	32.89%	lebih kecil dari file asli
13	File 13	126	75	126	40.48%	lebih kecil dari file asli
14	File 14	8	9	8	-12.50%	lebih besar dari file asli
15	File 15	113	108	113	4.42%	lebih kecil dari file asli
16	File 16	10450	7552	10450	27.73%	lebih kecil dari file asli
17	File 17	3166	3161	3166	0.16%	lebih kecil dari file asli
18	File 18	10451	7541	10451	27.84%	lebih kecil dari file asli
19	File 19	335	330	335	1.49%	lebih kecil dari file asli
20	File 20	78	46	78	41.03%	lebih kecil dari file asli
21	File 21	2309	2080	2309	9.92%	lebih kecil dari file asli
22	File 22	1180	1148	1180	2.71%	lebih kecil dari file asli
23	File 23	1648	1377	1648	16.44%	lebih kecil dari file asli
24	File 24	6933	6825	6933	1.56%	lebih kecil dari file asli
25	File 25	643	614	643	4.51%	lebih kecil dari file asli

Dari Tabel 2 dapat dilihat bahwa terdapat perubahan ukuran data. Perubahan terlihat saat penerima e*mail* menerima file attachment. Untuk hasil prosentase positif menunjukkan bahwa file yang diterima untuk penerimaan *key* PGP adalah lebih besar dibandingkan dengan tanpa *key* PGP yang artinya bahwa penerimaan e*mail* dengan PGP sesuai dengan file asli pengiriman dan memiliki resiko rendah terhadap kesalahan pengiriman data (dibuktikan bahwa penerimaan dengan PGP memiliki nilai hash yang sama dengan file asli). Sedangkan untuk nilai prosentasi negative (-) memiliki arti bahwa penerimaan e*mail* dengan PGP tidak sesuai dengan file asli pengiriman dan memiliki resiko lebih tinggi terhadap kesalahan pengiriman data. Dari Tabel 2 dapat digambarkan graph penerimaan e*mail* prosentasi pembanding ukuran data PGP sebagai berikut :



Gambar 8. Graph Prosentasi Pembanding *key* PGP

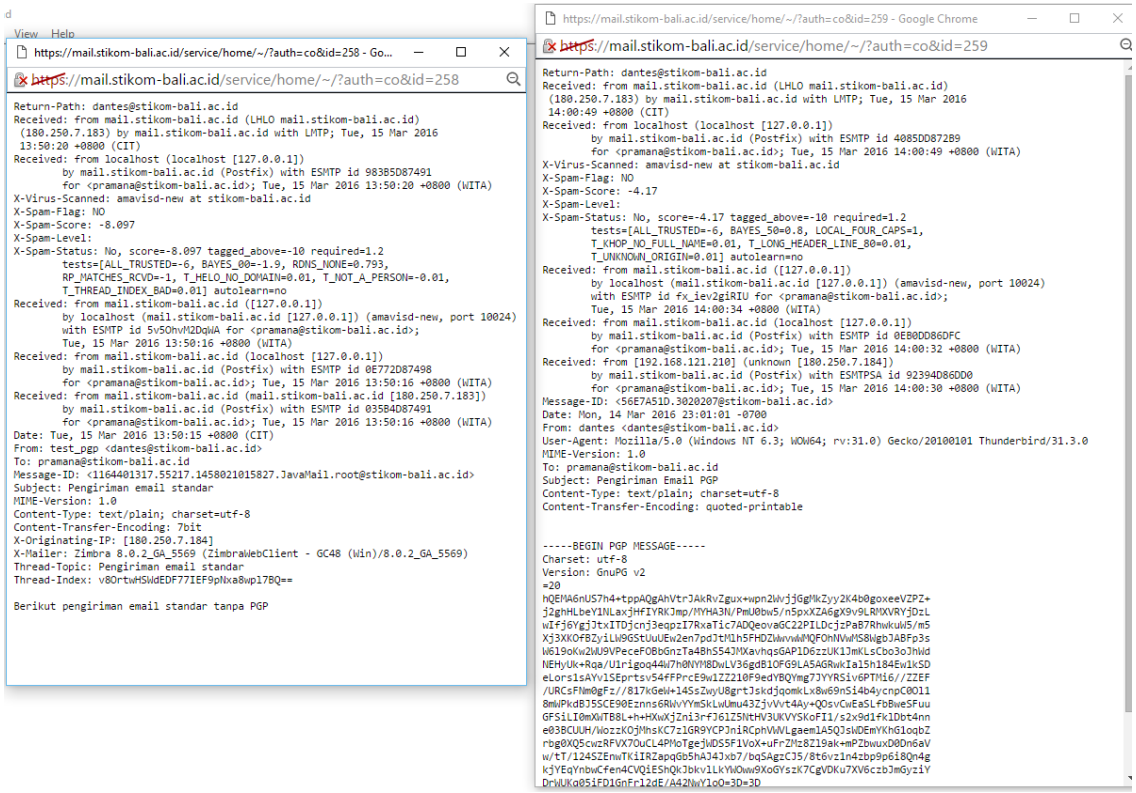
Untuk mengetahui rata – rata perubahan file dilakukan penghitungan rerata sebagai berikut :

$$x = \frac{X1 + X2 + X3 + \ldots\ldots + Xn}{n} \quad (4)$$

$$x = \frac{1}{n}\sum_{i=1}^{n} x1 \quad (5)$$

Dengan pengujian 100 data maka di dapat data stabilitas kompresi terhadap penggunaan *key* PGP adalah sebesar 15.41% yang artinya adalah penerimaan *email* tanpa penggunaan *key* PGP menunjukkan resiko keruasakan data lebih besar dibandingkan menggunakan *key* PGP yang seharusnya sama apabila dibandingkan dengan file asli pengirim. Pengujian berikutnya adalah

dilakukan dengan menganalisa header e*mail* antara *mail* yang dikirimkan dengan menggunakan teknik PGP dan tanpa PGP.



Gambar 9. Perbandingan *Mail* Header

Dari Gambar 7 dapat dijelaskan yang terlihat bahwa e*mail* yang menggunakan PGP memiliki informasi penggunaan User_agent (garis line hitam) yang menjelaskan penggunaan *mail* client Mozilla Thunderbird. Informasi Message ID pengiriman juga memiliki perbedaan dimana apabila e*mail* yang tanpa menggunakan PGP memberikan informasi detail posfix message detail, sedangkan *mail* yang menggunakan PGP tidak detail. Hal ini dikarenakan menggunakan aplikasi *mail* client Mozilla Thunderbird. Informasi teknik encoding yang ditampilkan (blok berwarna kuning) terhadap *mail* yang menggunakan PGP mengartikan bahwa hasil enkripsi tercetak, sedangkan tanpa PGP hanya menggunakan encoding standar pengiriman 7 bit (default *mail* server STIKOM)

5. Kesimpulan

Berdasarkan Penelitian yang dilakukan, dapat disimpulkan bahwa pengamanan menggunakan teknik PGP mampu mengamankan komunikasi *email*. Pihak yang tidak berkepentingan dapat saja mencuri dan mengetahui user *mail* account dan password, namun tidak dapat membaca isi dari *email* karena telah terenkripsi. Hasil analisa juga menunjukkan bahwa terdapat perbedaan size ukuran dari file attachment yang menggunakan pengamanan PGP, dimana size file menjadi lebih besar yang disebabkan adanya proses enkripsi dengan kunci private. Disisi lain, terlihat perbedaan *mail* header dimana pengamanan PGP memberikan identitas enkripsi dibandingkan *mail* yang tanpa PGP. Namun analisa *mail* header juga menunjukkan kurangnya detail informasi *mail* posfix pada pengamanan dengan teknik PGP.

Sebagai bahan pengembangan pada penelitian berikutnya, dapat dikembangkan penganalisaan terhadap uji coba *key* generate dari PGP untuk melihat ketahanan terhadap keamanan *key* PGP dan pengaruh header file dari hasil enkripsi dengan pengujian pada *mail* engine lainnya

Daftar Pustaka

[1] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts Essentials*. 2011.
[2] A. Dumka, R. Tomar, J. C. Patni, and A. Anand, "Taxonomy of *Email* Security Protocol," *Int. J. Innov. Res. Comput. Commun. Eng.*, 2014.
[3] A. Bacard, *The Computer Privacy Handbook*. Peachpit Press, 1995.
[4] M. Os *et al.*, "ZIMBRA *Mail* Server With Ubuntu 8 . 04."
[5] E. Zaida and Rusmanto, *Panduan Praktis Membangun Server Email Enterprise dengan Zimbra*. Jakarta: Dian Rakyat, 2010.