UNIVERSITAS INDONESIA

RANCANG BANGUN DAN IMPLEMENTASI MOBILE SECURED EMAIL BERBASIS GNU PRIVACY GUARD MENGGUNAKAN INFRASTRUKTUR PORTABLE EMAIL SERVER

SKRIPSI

FITRIANTA EKA PRASAJA

0806366541

FAKULTAS TEKNIK ELEKTRO

DEPARTEMEN TEKNIK ELEKTRO

DEPOK

JUNI 2011

UNIVERSITAS INDONESIA

RANCANG BANGUN DAN IMPLEMENTASI MOBILE SECURED EMAIL BERBASIS GNU PRIVACY GUARD MENGGUNAKAN INFRASTRUKTUR PORTABLE EMAIL SERVER

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

FITRIANTA EKA PRASAJA

0806366514

FAKULTAS TEKNIK

DEPARTEMEN TEKNIK ELEKTRO

UNIVERSITAS INDONESIA

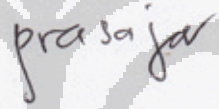
DEPOK

JUNI 2011

PERNYATAAN ORISINALITAS

Skripsi adalah hasil karya saya sendiri,
Dan semua sumber yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama	:	Fitrianta Eka Prasaja
NPM	:	0806366541
Tanda Tangan	:	



Tanggal	:	14 Juni 2011

HALAMAN PEGESAHAN

Skripsi ini diajukan oleh :

Nama : Fitrianta Eka Prasaja

NPM : 0806366541

Program Studi : Teknik Elektro

Judul Skripsi : RANCANG BANGUN DAN IMPLEMENTASI MOBILE SECURED EMAIL BERBASIS GNU PRIVACY GUARD MENGGUNAKAN INFRASTRUKTUR PORTABLE EMAIL SERVER

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Muhammad Salman ST., MIT ( )

Penguji : Dr. Ir. Anak Agung Putri Ratna M.Eng ( )

Penguji : Prima Dewi Purnamasari ST., MT., MSc ( )

Ditetapkan di : Depok

Tanggal : 11 Juli 2011

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah Subhanahu wa Ta'ala, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini sebagai syarat kelengkapan menjadi Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Tenik Elektro, Univeristas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari mulai perkulihan sampai dengan penyusunan skripsi ini, sangatlah sulit untuk menyelesaikannya. Oleh karena itu perkenankan saya untuk menyampaikan terima kasih kepada:

(1) Bapak Muhammad Salman S.T., MIT, selaku dosen pembimbing,
(2) Kedua orang tua saya yang tak henti-hentinya memberikan motivasi dan berdoa agar Allah senantiasa memperlancar segala urusan bagi putranya,
(3) Istri dan kedua anakku, Muhammad Prastika dan Athifa Azkadina Atha,
(4) Bapak Kepala Lembaga Sandi Negara dan Ketua Sekolah Tinggi Sandi Negara (STSN) yang telah mengijinkan saya untuk menempuh tugas belajar di jenjang S1 Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia,
(5) Rekan- rekan di lingkungan STSN dan Pusdiklat Bumi Sanapati, Bogor,
(6) Sahabat yang telah banyak membantu dalam perkuliahan dan penyusunan skripsi.

Akhir kata, saya berharap Allah Subhanahu wa Ta'ala berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini dapat membawa manfaat bagi ilmu pengetahuan.

Depok, 14 Juni 2011

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertandatangan di bawah ini:

Nama	:	Fitrianta Eka Prasaja
NPM	:	0806366541
Program Studi	:	Teknik Elektro
Departemen	:	Teknik Elektro
Fakultas	:	Teknik
Jenis Karya	:	Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (***Non-exclusive Royalty-Free Right***) atas karya ilmiah saya yang berjudul:

RANCANG BANGUN DAN IMPLEMENTASI MOBILE SECURED EMAIL BERBASIS GNU PRIVACY GUARD MENGGUNAKAN INFRASTRUKTUR PORTABLE EMAIL SERVER

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusif ini Universitas Indonesia berhak menyimpan, mengalih media /formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/ pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 11 Juli 2011

Yang menyatakan

prasaja

(Fitrianta Eka Prasaja)

DAFTAR ISI

ABSTRAK

Nama : Fitrianta Eka Prasaja
Program Studi : Teknik Elektro
Judul : Rancang Bangun dan Implementasi Mobile Secured Email Berbasis *GNU Privacy Guard* Menggunakan Infrasruktur Portable Email Server

GNU Privacy Guard (GnuPG, atau GPG) merupakan *hybrid crypto system* yang digunakan untuk mengamankan komunikasi dan penyimpanan data. Salah satunya dapat dimanfaatkan untuk pengamanan komunikasi lewat email.

GPG Webmail Versi 1.0 merupakan model aplikasi berbasis web untuk mengakses email yang berfungsi sebagai perantara dalam melakukan penyandian email dari email server dengan protokol POP3 dan SMTP. GPG Webmail Versi 1.0 ini diintegrasikan ke dalam distro linux server sehingga menjadi distro linux baru dilengkapi dengan paket aplikasi penyandian email berbasis web yang siap digunakan.

Aplikasi ini mempunyai keunggulan mudah dalam penggunaannya karena user tidak perlu melakukan instalasi serta mempermudah suatu organisasi yang belum mempunyai email Server sendiri namun ingin menyediakan fasilitas aman bagi anggotanya yang masih menggunakan email server yang bukan miliknya.

Keunggulan lainnya adalah tak perlu mengamankan PC/Laptop yang digunakan untuk mengakses aplikasi, karena data rahasia dan *secret key* tidak tersimpan dalam PC/Laptop tersebut, sehingga user bisa mengakses aplikasi dari PC/Laptop milik fasilitas umum.

Kata kunci:
Mobile, Portable, Server, Email, GnuPG

ABSTRACT

Name : Fitrianta Eka Prasaja
Program : Electro
Title : Design and Implementation of Mobile Secured Email Based on *GNU Privacy Guard* Using The Portable Email Server Infrastructure

GNU Privacy Guard (GnuPG or GPG) is a hybrid crypto system used to preserve communication and data storage, one of which can be utilized for communication pacification through e-Mail.

GPG Webmail Version 1.0 is a web based model application to access a web based e-Mail that serves as an intermediary in conducting e-mail encryption of e-Mail server with POP3 and SMTP protocols. GPG Webmail Version 1.0 is integrated into Linux distros servers so that the new linux distro is equipped by web based e-Mail encoding application package which is ready for use.

The application has the advantage of easy to use as the user does not need to install anything. It also simplifies an organization that has not got an e-Mail server itself but want to provide safe facilities for members who still use e-mail server that is not its property.

Another advantage is no need to secure the PC / laptop that is used to access the application as confidential data and secret key are not stored in the PC / laptop, so that users can access the applications from a public PC / Laptop.

keyword:
Mobile, Portable, Server, Email, GnuPG

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1. LATAR BELAKANG

Tahun 1971 adalah masa yang bersejarah dalam dunia internet, karena pada tahun tersebut email atau surat elektronik untuk pertama kalinya dikirimkan. Mulai tahun 1980-an email sudah bisa dinikmati oleh khalayak umum, baik itu email gratis maupun yang berbayar. Saat ini banyak email berbasis web yang diberikan secara cuma-cuma antara lain oleh Telkom, Yahoo!, Gmail, dan MSN.

Hingga kini pemanfaatan email semakin tinggi. Pada tahun 2008 diperkirakan ada sekitar 1,3 miliar lebih pengguna email di seluruh dunia [1].

Isu yang berkembang saat ini terkait dengan email adalah klaim Gmail yang kembali diserang sekelompok hacker dan berhasil menerobos beberapa ratus akun email, termasuk akun pejabat senior Pemerintah Amerika Serikat (AS), personel militer, dan aktivis politik. Raksasa internet asal Mountain View itu menduga serangan ini berasal dari China [2].

Dari peristiwa tersebut terlihat bahwa email menjadi obyek serangan untuk mendapatkan informasi/data. Keamanan data di email tidaklah terjamin dan selalu ada resiko terbuka untuk umum, dalam artian semua isinya dapat dibaca oleh orang lain. Hal ini disebabkan oleh karena email itu akan melewati banyak server sebelum sampai di tujuan. Tidak tertutup kemungkinan ada orang yang menyadap email yang dikirimkan tersebut.

Email telah menjadi media komunikasi yang penting sekaligus ruang privat bagi pemiliknya. Dibutuhkan teknik pengamanan terhadap data dalam email kita. Teknik yang biasa digunakan adalah dengan menggunakan Kriptografi. Program yang terkenal saat ini untuk pengamanan email adalah dengan *GnuPG Hybrid Crypto System*.

Dalam skripsi ini, penulis akan membuat aplikasi untuk mengakses account e-Mail POP3-SMTP seperti Gmail, Ymail, dll. Aplikasi yang dirancang berbasiskan web. Aplikasi ini juga akan diintegrasikan dengan GnuPG, sehingga dapat digunakan untuk mengirim dan menerima email dengan lebih aman karena

data yang ada di email menjadi tersandi. Selanjutnya Aplikasi tersebut akan di jadikan sebuah paket dalam distro linux server Ubuntu versi 10.10.

Penulis berharap hasil skripsi ini mampu dijadikan cara alternatif yang aman dalam mengakses sebuah akun email serta bisa dijadikan salah satu *tool* untuk melakukan kampanye tentang kesadaran keamanan informasi sekaligus sebagai sumbangsih terhadap perkembangan dunia linux (*open source*) di Indonesia.

.

1.2. PERUMUSAN MASALAH

Dari latar belakang penulisan sripsi ini maka dapat dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana merancang suatu aplikasi untuk mengakses email berbasis web yang aman dan mudah diakses user secara mobile atau dengan kata lain user aman dan mudah mengakses aplikasi dari PC/Laptop milik fasilitas umum (misal : warnet).
2. Bagaimana merancang suatu aplikasi untuk mengakses email berbasis web yang mampu berfungsi untuk mengirim dan menerima Secure email dari mail server dengan protokol POP3 dan SMTP.
3. Bagaimana merancang suatu aplikasi untuk mengakses email berbasis web yang terintegrasi dengan GnuPG sebagai fitur pengamanannya.
4. Bagaimana menyediakan protokol yang aman untuk komunikasi antara klien dan server.
5. Bagaimana mengkustomisasi distro Linux Ubuntu 10.10 sehingga menjadi distro server yang mampu menjalankan Aplikasi. Distro di tempatkan dalam media penyimpanan writeable (Flash Disk), sehingga mampu menjadi portable email server.

1.3. TUJUAN

Skripsi ini memiliki beberapa tujuan sebagai berikut :

1. Membuat aplikasi untuk mengakses email berbasis web/mobile yang berfungsi untuk mengirim dan menerima Secure email dari mail server dengan protokol POP3 dan SMTP.

2. Menjalankan fungsi GnuPG dari aplikasi email klien berbasis web.
3. Menjalankan Protokol SSL untuk jalur komunikasi dalam mengakses Aplikasi.
4. Membuat distro linux server yang berisi aplikasi Secure email berbasis GNUPG yang mampu diakses lewat jaringan dan dikemas dalam bentuk distro portable. Aplikasi ini dinamakan GPG WEBMAIL.

1.4. METODOLOGI

Perancangan Aplikasi ini menggunakan metodologi :

1. Studi Literature
    a. Linux dan pembuatan distro Linux.
    b. Pemrograman web untuk frontend.
    c. Protokol POP3 dan SMTP
    d. Pembuatan Web Server dangan fitur SSL
    e. Operasi GnuPG
2. Perancangan serta pembuatan Aplikasi.
    a. Sequence Diagram
    b. Use Case Diagram
3. Implementasi dan pengujian aplikasi
    a. Pengujian *Black Box*
    b. Pengujian Keamanan
    c. Pengujian *Web Performance*

1.5. SISTEMATIKA PENULISAN

Penulisan skripsi ini tersusun dari lima bab yang selanjutnya akan dijelaskan dengan sub bab. Secara keseluruhan skrispsi ini disusun sebagai berikut :

1. Pendahuluan

Bab pendahuluan merupakan bab yang berisi latar belakang, tujuan, perumusan masalah, metodologi dan sistematika penulisan.

2. **Landasan Teori**

   Bab ini menerangkan teori-teori dasar yang digunakan dalam penulisan skripsi seperti : linux dan *open source*, bahasa pemrograman yang digunakan dalam pembuatan aplikasi, GnuPG *Hybrid Crypto System*, web server, POP3, SMTP dan SSL

3. **Perancangan Aplikasi Monitoring dan Hardware Monitoring**

   Bab ini menjelaskan perancangan aplikasi. Meliputi *Sequence Diagram Use Case Diagram*.

4. **Implementasi dan Pengujian**

   Bab ini menggambarkan Implementasi *Logic Application*, *Template Aplication* dan implementasi distro Linux Ubuntu 10.10 sebagai server aplikasi

5. **Penutup**

   Bab ini berisi kesimpulan yang didapat dari skripsi serta saran yang membangun untuk penelitian selanjutnya

BAB II
KONSEP SECURED EMAIL BERBASIS GNU PRIVACY GUARD

2.1. GNUPG

Berdasarkan penggunaan kuncinya, algoritma kriptografi bisa dibagi menjadi dua, yakni kriptografi kunci-simetri dan kriptografi kunci-asimetri. Kriptografi kunci-simetri menggunakan kunci yang sama (secret key) untuk melakukan enkripsi dan dekripsi. Sedangkan kriptografi kunci-asimetri menggunakan kunci yang berbeda untuk melakukan enkripsi dan dekripsi. Pada kriptografi kunci-asimetri, enkripsi dilakukan dengan kunci publik, sedangkan dekripsi dilakukan dengan kunci privat. Karena itu, kriptografi kunci-asimetrik disebut juga kriptografi kunci-publik [3].

Penggunaan perangkat lunak yang memanfaatkan kriptografi, baik kunci-simetri maupun kunci-asimetri, sangatlah memudahkan pengguna untuk menjaga keamanan komunikasi ataupun data. Dengan memanfaatkan perangkat lunak tersebut, pengguna bisa melakukan enkripsi, tanda tangan digital, ataupun otentifikasi data. Salah satu perangkat lunak yang memanfaatkan kriptografi adalah *GNU Privacy Guard* (biasa disingkat menjadi GnuPG atau GPG).

2.1.1. Sekilas Tentang Gnu Privacy Guard

GnuPG merupakan perangkat lunak open source yang mengimplementasikan standar OpenPGP secara lengkap sebagaimana yang didefinisikan pada RFC4880. Sebagai perangkat lunak open source, GnuPG merupakan alternatif dari perangkat lunak PGP (Pretty Good Privacy). GnuPG biasanya sudah tercakup di dalam kebanyakan distro Linux, seperti Debian, MandrakeSoft, RedHat, dan SuSE. Dengan GnuPG, pengguna bisa mengenkripsi dan menandatangani data dan komunikasi. GnuPG merupakan perangkat lunak command line yang mudah diintegrasikan dengan aplikasi lainnya. Meskipun demikian, tersedia juga aplikasi tampilan (frontend) dan pustaka yang mendukungnya [4].

Meskipun GnuPG kebanyakan digunakan di lingkungan sistem operasi yang open souce seperti Linux dan FreeBSD, kode program GnuPG juga bisa digunakan di lingkungan Windows. Adapun GnuPG versi Windows, dinamakan Gpg4win. Gpg4win ini merupakan bundel program yang dikemas untuk Windows yang mencakup GnuPG, pengelola kunci (WinPT dan GPA), plugin untuk enkripsi email (GPGol), plugin untuk enkripsi file (GPGee), dan program email client (Claws Email). Pada skripsi ini, penulis melakukan eksperimen dengan menggunakan GnuPG versi Linux.

2.1.2. Cara Kerja Gnu Privacy Guard

Sebagaimana yang telah ditetapkan dalam standar OpenPGP, GnuPG menyediakan layanan integritas pesan dan file data dengan teknologi tanda tangan digital, enkripsi, kompresi, dan konversi Radix-64. GnuPG juga menyediakan layanan manajemen dan sertifikat kunci. Untuk menjamin kerahasiaan pesan atau file data, GnuPG menggunakan kombinasi kriptografi kunci-simetrik dan kriptografi kunci-publik. Adapun langkah-langkah menjaga kerahasian data pada pengiriman suatu pesan dengan melakukan enkripsi pada GnuPG adalah sebagai berikut:

a. Pengirim membuat pesan.
b. Pengirim membangkitkan sebuah bilangan acak atau memberikan sandi lewat (passphrase) sebagai session key untuk pesan saat ini.
c. Pengirim mengenkripsi session key tersebut dengan kunci publik masing-masing penerima. Hasil enkripsi *session key* ini menjadi awal dari pesan yang dikirim.
d. Pengirim mengenkripsi pesan (yang biasanya sudah dikompresi) yang akan dikirim dengan menggunakan *session key*.
e. Penerima mendekripsi pesan dengan kunci privatnya.
f. Penerima mendekripsi pesan dengan *session key*. Jika pesan yang diterima merupakan hasil kompresan, maka pesan harus didekompresi.

Baik layanan tanda tangan digital maupun kerahasiaan, bisa diaplikasikan pada pesan yang sama. Caranya, tanda tangan digital dibangkitkan dan dibubuhkan pada pesan. Lalu, pesan dan tanda tangan tersebut dienkripsi dengan menggunakan session key yang simetrik. Lalu, *session key* dienkripsi menggunakan enkripsi kunci-publik dan ditaruh di awal blok yang terenkripsi.

Sedangkan langkah-langkah otentifikasi yang dilakukan GnuPG melalui tanda tangan digital adalah sebagai berikut:

a. Pengirim membuat pesan.
b. Pengirim membangkitkan kode hash dari pesan.
c. Pengirim membangkitkan tanda tangan digital dari kode hash pesan dengan menggunakan kunci privat pengirim.
d. Tanda tangan digital tersebut dilekatkan pada pesan.
e. Penerima menerima pesan yang bertanda tangan.
f. Penerima membangkitkan kode hash dari pesan yang diterima dan memverifikasinya dengan menggunakan tanda tangan digital pada pesan. Jika verifikasi berhasil, berarti pesan yang diterima tersebut otentik.

2.1.3. Algoritma Pada GNU Privacy Guard

GnuPG yang digunakan penulis untuk eksperimen adalah versi 1.4.10. Versi ini mendukung beberapa macam algoritma.

a. Algoritma kunci-publik: RSA, RSA-E, RSA-S, ELG-E, DSA
b. Algoritma simetrik: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH
c. Algoritma hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224

Dengan bermacam algoritma di atas, pengguna bisa melakukan enkripsi atau memberikan tanda tangan digital dengan algoritma yang berbeda-beda. Untuk pembangkitan kunci, diberikan pilihan algoritma: DSA, El Gamal, dan RSA.

Pemilihan algoritma yang bagus tentunya sangat mempengaruhi keamanan data. Selain algoritma kriptografi di atas, GnuPG versi 1.4.10 ini juga mendukung beberapa algoritma kompresi, antara lain: ZIP, ZLIB, BZIP2.

2.2. SISTEM OPERASI LINUX

Linux adalah sebuah sistem operasi yang sangat mirip dengan sistem UNIX. Seluruh kode sumber linux termasuk *kernel*, *dev*ice driver, *libraries* (perpustakaan sistem), program dan *tool* pengembangan disebarkan secara bebas dengan lisensi GPL (General Public License). Sistem operasi Linux terdiri dari tiga bagian kode penting [5] :

a. Kernel, bertanggung jawab memelihara semua abstraksi penting dari sistem operasi, termasuk hal-hal seperti memory *virtual* dan proses-proses.
b. Perpustakaan sistem, menentukan kumpulan fungsi standar dimana aplikasi dapat berinteraksi dengan kernel, dan mengimplementasikan hampir semua fungsi sistem operasi yang tidak memerlukan hak akses penuh atas kernel.
c. Utilitas sistem merupakan program yang melakukan pekerjaan manajemen secara individual

2.2.1. Distribusi Linux

Distro adalah kernel linux ditambah dengan kumpulan paket-paket perangkat lunak dari GNU dan lainnya, yang dibundel menjadi satu, dengan tujuan untuk mempermudah proses distribusi perangkat lunak.

Sebelum membuat distro perlu dipahami dua karakter distro yang dibuat, berdasarkan sifat redistribusinya yaitu [6] :

a. Distro dipakai untuk untuk diri sendiri.
   Distro ini dibuat dengan basis LFS (*Linux From Scratch*) dan semua aplikasi dikompilasi dalam *pristine code*(kode program murni).
b. Distro dari turunan distro besar yang sudah maju.
   Biasanya yang dipakai rujukan adalah Redhat (misal Fedora), debian (misal ubuntu).

2.2.2. Distro Ubuntu

Ubuntu adalah salah satu distribusi Linux berbasiskan pada Debian dan memiliki antarmuka desktop. Proyek Ubuntu disponsori oleh Canonical Ltd (perusahaan milik Mark Shuttleworth). Nama Ubuntu diambil dari sebuah konsep ideologi di Afrika selatan [7].

Ubuntu dikonsentrasikan untuk penggunaan, bebas digunakan, secara reguler dirilis setiap 6 bulan dan mudah diinstalasi. Versi terbaru adalah versi 10.4 yang dirilis April 2010. Distro Linux Ubuntu mendukung berbagai arsitektur komputer seperti :

a. PC (Intel x86).
b. PC 64-bita (AMD64).
c. PowerPC (Apple iBook dan Powerbook, G4 dan G5).
d. Sun UltraSPARC.
e. T1 (Sun Fire T1000 dan T2000)

2.3. OPEN SOURCE SOFTWARE

2.3.1. Apache Web Server

Web server merupakan suatu server internet yang menggunakan protokol HTTP (Hypertext Transfer Protocol) untuk melayani semua proses pentransferan data. Hingga saat ini webserver merupakan server yang dapat dikatakan sebagai tulang punggung bagi semua internet. Hal ini dikarenakan web server tidak hanya melayani data berbetuk teks saja akan tetapi juga dapat menampilkan gambar dalam bentuk 2D maupun 3D, suara dan lain sebagainya [8].

Apache web server yang dibuat oleh Apache Software Foundation(ASF). ASF sendiri merupakan suatu komunitas para pengembang software yang kemudian setiap software yang nereka produksi berada dibawah lisensi Apache Licence yang berdifat bebas dan kode sumber terbuka.

Apache web server merupakan salah satu dari sekian banyak web server yang ada di internet, akan tetapi sebagian besar server dengan lingkungan Unix/Linux menggunakannya, hal ini disebabkan oleh beberapa alasan yaitu:

- Kecepatan yang lebih baik dibandingkan dengan webserver lainnya.

- Performance yang sangat baik.
- Dapat diperoleh dengan gratis (open source)

Apache sendiri merupakan program yang modular dan memiliki program layanan pendukung yang cukup banyak yang dapat digunakan oleh pengguna. Beberapa layanan tersebut adalah :

- Kontrol Akses
- *Common Gateway Interface* (CGI)
- *Personal Home Page*
- *Server Side Include* (SSI)

Apache *binary* atau biasa disebut dengan httpd di lingkungan sistem operasi Unix/Linux dan Apache.exe di lingkungan sistem operasi Microsoft Windows pada umumnya berjalan secara *background*. Tanpa melihat sistem operasi dimana apache diinstall, httpd/apache secara umum terdiri dari empat subdirektori, yaitu:

- conf

  Conf merupakan subdirektori tempat file-file konfigurasi dari apache. Salah satu yang terpenting adalah httpd.conf. dimana pada file tersebut merupakan konfigurasi dari web server apache dan menyatakan URLs yang dilayani.
- htdocs

  Subdirektori ini merupakan subdirektori dimana nantinya file-file HTML yang di minta oleh browser client. Direktori-direktori yang berada dibawahnya merupakan data-data umum yang nantinya akan disediakan sesuai dengan permintaan dari client browser.
- logs

  Subdirectory logs berisi file-file log yang mencatat akses terhdap web server, selain itu juga mencatat error yang terjadi saat pengaksesan.
- cgi-bin

  Subdirektori ini merupakan subdirectory dimana skrip-skrip CGI disimpan. Skrip-skrip tersebut dijalankan mewakili permintaan dari client

karena client tidak dimungkinkan secara langsung berinteraksi dengan sistem operasi.

2.3.2. MySQL Database Server

MySQL merupakan Database Managemen System (DBMS) Open Source. Sebagai database server, MySQL melayani banyak user yang ingin mengakses ke database-database. Proyek Pengembangan MySQL berada dibawah lisensi GNU General Publc License atau GPL. Selain berlisensi GPL MySQL juga mempunyai lisensi propeteri yang dimiliki dan disponsori oleh perusahaan Swedia MySQL AB, namun sekarang kepemilikan MySQL AB sudah berada dibawah Sun Microsystem yang merupakan Subsidiary Oracle Coproration [9].

Database terdiri dari koleksi data yang terorganisasi untuk satu atau beberapa tujuan. Salah satu cara pengklasifikasian isi data dengan database adalah melalui tipe data, sebagai contoh bibliograpgy, gambar, angka dan lain sebagainya. Pada dasarnya software atau program mengorgasisai data merujuk pada model database yang umum seperti model relasional (relational model), model hirarki (hiearki model) dan model jaringan (network model).

Database-database merupakan penampung yang berbasis software yang mana secara terstruktur mengoleksi dan menyimpan informasi sehingga user dapat mengambil (retrieve), menambah (add), mengupdate (update) dan menghapus informasi secara otomatis.

Dengan kata lain program database merupakan program yang didesain untuk user sehingga mereka dapat menambah maupun mengurangi informasi yang diperlukan. Struktur database sendiri merupakan sebuah table yang berisi informasi dan terdiri atas baris dan kolom.

Beberapa contoh database yang cukup dikenal luas seperi Oracle, Microsoft SQL Server, Microsoft Access, MySQL, PostgreSQL, Dbase, SQLite dan lain sebagainya. Sebagian database tersebut merupakan database server yang bias melayani banyak user dan banyak database.

Database Management Systems (DBMS) merupakan database yang mengorganisasi penyimpanan data. DBMS dapat mengontrol pembuatan, maintenance dan penggunaan struktur database dan para penggunanya. DBMS

sendiri dapat digolongkan menurut database model yang disupport, sebagaimana telah disebutkan sebelumnya.

Setiap database model cenderung membuat bahasa query untuk mengakses database. Salah satu bahasa query untuk database relasional adalah Structured Query Language (SQL). Bahasa SQL merupakan bahasa digunakan untuk memanajemen data pada Relational Database Management System (RDBMS).

SQL didasarkan pada aljabar relasi yang mempunyai ruang lingkup pada data query dan update, pembuatan skema database dan akses control. Secara umum SQL dibagi menjadi bentuk query, yaitu :

- Data Definition Languange (DDL), adalah sebuah metode query SQL yang berguna mendefinisikan data pada sebuah database.
- Data manipulation language (DML), merupakan metode query SQL yang berguna untuk memanipulasi tabel maupun database.

2.3.3. PHP

Hypertext Pre Processor atau kemudian disebut sebagai PHP adalah bahasa pemrograman dalam pengembangan web dinamis, berupa skrip-skrip yang diletakkan dan di eksekusi di server (server-side) [10].

PHP versi 5 sudah mendukung Object Oriented Programming (OOP). Beberapa kelebihan PHP dari bahasa pemrograman lain:

- Bahasa pemrograman PHP adalah bahasa skrip yang tidak melakukan kompilasi dalam penggunaanya.
- Web server yang mendukung PHP cukup banyak seperti Apache, IIS, lighthttpd, Ngix hingga Xitami.
- Mempunyai dukungan komunitas yang cukup banyak milis, forum dan lain sebagainya.
- Merupakan open source software yang dapat dijalankan semua platform sistem operasi.

PHP merupakan *server-side embedded script languange* yang artinya, sintaks-sintaksnya dan perintah yang diberikan akan sepenuhnya dijalankan server, akan tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang

dibangun pada umumnya akan memberikan hasil pada halaman web, akan tetapi prosesnya secara keseluruhan dijalankan di server.

Kode-kode PHP diembed pada halaman HTML dan disimpan pada server. Client memberikan REQUEST ke server halaman mana yang ingin dilihat. Apabila pada halaman tersebut terdapat skrip PHP, maka skrip tersebut akan diproses kemudian hasilnya akan ditampilkan bersama kode HTML yang lain. Secara singkat server melakukan hal-hal berikut:

- Membaca permintaan dari client
- Mencari halaman pada page/server
- Melakukan proses skrip php
- Mengirim halaman hasil tersebut ke client

2.4. MODEL PROTOKOL JARINGAN

Komunikasi dalam jaringan merupakan tugas yang sangat komplek, oleh karena itu diperlukan suatu struktur protokol model. Struktur protokol model jaringan yang cukup tepat adalah dengan cara menyusun menjadi sejumlah lapisan (*layer*). Model struktur jaringan membawa tugas yang sangat kompleks tersebut menjadi unit-unit yang lebih kecil.

Struktur protokol dirancang untuk dengan cara memecah permasalahan komunikasi data kedalam unit-unit yang lebih kecil. Hingga saat ini terdapat dua model protokol jaringan yang populer, yaitu model OSI dan model TCP/IP.

2.4.1. OSI Model

Model referensi jaringan terbuka OSI atau *OSI Reference Model for open networking* adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan International Organization for Standardization (ISO) di Eropa pada tahun 1977. OSI sendiri merupakan singkatan dari Open System Interconnection. Model ini disebut juga dengan model "Model tujuh lapis OSI" (OSI *seven layer model*).

Sebelum munculnya model referensi OSI, sistem jaringan komputer sangat tergantung kepada pemasok (vendor). OSI berupaya membentuk standar

umum jaringan komputer untuk menunjang interoperatibilitas antar pemasok yang berbeda.

Dalam suatu jaringan yang besar biasanya terdapat banyak protokol jaringan yang berbeda. Tidak adanya suatu protokol yang sama, membuat banyak perangkat tidak bisa saling berkomunikasi.

Model referensi ini pada awalnya ditujukan sebagai basis untuk mengembangkan protokol-protokol jaringan, meski pada kenyataannya inisatif ini mengalami kegagalan. Kegagalan itu disebabkan oleh beberapa faktor berikut:

a. Standar model referensi ini, Jika dibandingkan dengan model referensi DARPA (Model Internet) yang dikembangkan oleh Internet Engineering Task Force (IETF), sangat berdekatan. Model DARPA adalah model basis protokol TCP/IP yang populer digunakan.
b. Model referensi ini dianggap sangat kompleks. Beberapa fungsi (seperti halnya metode komunikasi connectionless) dianggap kurang bagus, sementara fungsi lainnya (seperti flow control dan koreksi kesalahan) diulang-ulang pada beberapa lapisan.
c. Pertumbuhan Internet dan protokol TCP/IP (sebuah protokol jaringan dunia nyata) membuat OSI Reference Model menjadi kurang diminati.

OSI Reference Model pun akhirnya dilihat sebagai sebuah model ideal dari koneksi logis yang harus terjadi agar komunikasi data dalam jaringan dapat berlangsung. Beberapa protokol yang digunakan dalam dunia nyata, semacam TCP/IP, DECnet dan IBM *Systems Network Architecture* (SNA) memetakan tumpukan protokol (protocol stack) mereka ke *OSI Reference Model*.

OSI Reference Model pun digunakan sebagai titik awal untuk mempelajari bagaimana beberapa protokol jaringan di dalam sebuah kumpulan protokol dapat berfungsi dan berinteraksi. Struktur tujuh lapis model OSI, bersamaan dengan protocol data unit pada setiap lapisan.

Gambar 2.1. OSI Model [11]

Dibawah ini merupakan tabel layer-layer pada OSI Model beserta penjelasan fungsi-fungsi secara umum.

Tabel 2.1. OSI 7 Layer [12]

Layer	Nama	Fungsi
Layer 7	*Application Layer*	Berfungsi sebagai antarmuka dengan aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan. Protokol yang berada dalam lapisan ini adalah HTTP, FTP, POP3,SMTP, dan NFS.
Layer 6	*Presentation Layer*	Berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan. Protokol yang berada dalam level ini adalah perangkat lunak redirektor (*redirector software*), seperti layanan Workstation (dalam Windows NT) dan juga *Network shell* (semacam *Virtual Network Computing* (VNC) atau *Remote Desktop Protocol* (RDP)).

Layer 5	*Session Layer*	Berfungsi untuk mendefinisikan bagaimana koneksi dapat dibuat, dipelihara, atau dihancurkan. Selain itu, di level ini juga dilakukan resolusi nama.
Layer 4	*Transport Layer*	Berfungsi untuk memecah data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima.
		Selain itu, pada level ini juga membuat sebuah tanda bahwa paket diterima dengan sukses *(acknowledgement),* dan mentransmisikan ulang terhadp paket-paket yang hilang di tengah jalan.
Layer 3	*Network layer*	Berfungsi untuk mendefinisikan alamat-alamat IP, membuat header untuk paket-paket, dan kemudian melakukan routing melalui i*nternetworking* dengan menggunakan router dan switch layer-3.
Layer 2	*Data-link Layer*	Befungsi untuk menentukan bagaimana bit-bit data dikelompokkan menjadi format yang disebut sebagai frame. Selain itu, pada level ini terjadi koreksi kesalahan, flow control, pengalamatan perangkat keras.
		(seperti halnya *Media Access Control Address* (MAC Address)), dan menetunkan bagaimana perangkat-perangkat jaringan seperti hub, bridge, repeater, dan switch layer 2 beroperasi. Spesifikasi IEEE 802, membagi level ini menjadi dua level anak, yaitu lapisan *Logical Link Control* (LLC) dan lapisan

		Media Access Control (MAC).
Layer 1	*Physical Layer*	Berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan (seperti halnya Ethernet atau Token Ring), topologi jaringan dan pengabelan. Selain itu, level ini juga mendefinisikan bagaimana Network Interface Card (NIC) dapat berinteraksi dengan media kabel atau radio.

2.4.2. TCP/IP

Transmission Control Protocol/Internet Protocol(TCP/IP) adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (protocol suite).

TCP/IP dikembangkan oleh DARPA (*US Defense Advanced Research Project Agent*) untuk paket-paket data yang dikirimkan dalam jaringan ARPANET. Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini.

Arsitektur TCP/IP tidaklah berbasis model referensi tujuh lapis OSI, tetapi menggunakan model referensi DARPA. Seperti diperlihatkan dalam diagram, TCP/IP merngimplemenasikan arsitektur berlapis yang terdiri atas empat lapis.

Empat lapis ini, dapat dipetakan (meski tidak secara langsung) terhadap model referensi OSI. Empat lapis ini, kadang-kadang disebut sebagai DARPA Model, Internet Model, atau DoD Model, mengingat TCP/IP merupakan protokol yang awalnya dikembangkan dari proyek ARPANET yang dimulai oleh Departemen Pertahanan Amerika Serikat.

Setiap lapisan pada TCP/IP yang dimiliki oleh kumpulan protokol (protocol suite). TCP/IP diasosiasikan dengan protokolnya masing-masing. Protokol utama dalam protokol TCP/IP adalah sebagai berikut:

a. Protokol lapisan aplikasi

   Bertanggung jawab untuk menyediakan akses kepada aplikasi terhadap layanan jaringan TCP/IP. Protokol ini mencakup protokol *Dynamic Host Configuration Protocol* (DHCP), *Domain Name System* (DNS), *Hypertext Transfer Protocol* (HTTP), *File Transfer Protocol* (FTP), *Telnet, Simple Email Transfer Protocol* (SMTP), *Simple Network Management Protocol* (SNMP), dan masih banyak protokol lainnya. Dalam beberapa implementasi stack protokol, seperti halnya Microsoft TCP/IP, protokol-protokol lapisan aplikasi berinteraksi dengan menggunakan antarmuka Windows Sockets (Winsock) atau NetBIOS over TCP/IP (NetBT).

b. Protokol lapisan antar-host

   Berguna untuk membuat komunikasi menggunakan sesi koneksi yang bersifat connection-oriented atau broadcast yang bersifat connectionless. Protokol dalam lapisan ini adalah *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP).

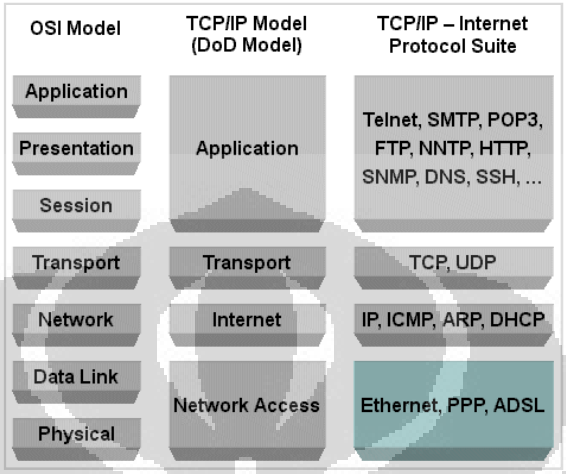
c. Protokol lapisan internetwork

   Bertanggung jawab untuk melakukan pemetaan (routing) dan enkapsulasi paket-paket data jaringan menjadi paket-paket IP. Protokol yang bekerja dalam lapisan ini adalah *Internet Protocol* (IP), *Address Resolution Protocol* (ARP), *Internet Control Message Protocol* (ICMP), dan *Internet Group Management Protocol* (IGMP).

d. Protokol lapisan antarmuka jaringan

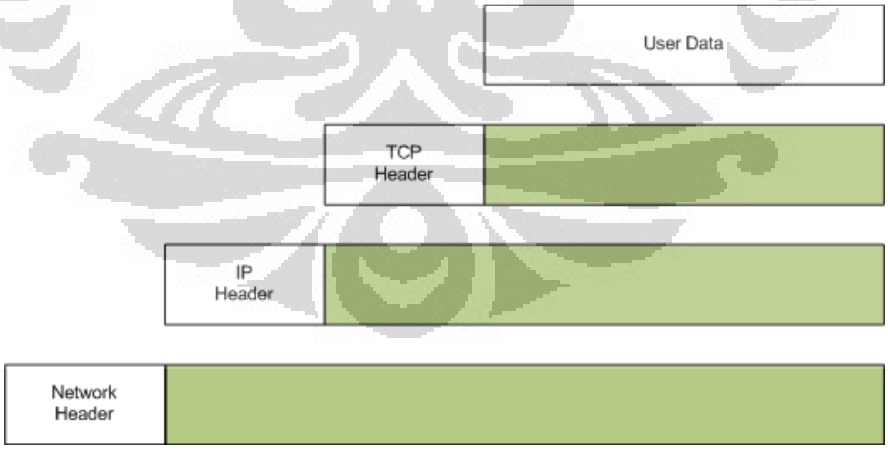
e. Bertanggung jawab untuk meletakkan frame-frame jaringan di atas media jaringan yang digunakan. TCP/IP dapat bekerja dengan banyak teknologi transport, mulai dari teknologi transport dalam LAN (seperti halnya Ethernet dan Token Ring), MAN dan WAN (seperti halnya dial-up modem yang berjalan di atas *Public Switched Telephone*

Network (PSTN), *Integrated Services Digital Network* (ISDN), serta *Asynchronous Transfer Mode* (ATM)).



Gambar 2.2. OSI, TCP/IP dan TCP/IP Suite [13]

Operasi dalam protokol TCP/IP adalah memindahkan *Protocol Data Unit* (PDU) sebagai data yang dialirkan dari satu sistem ke sistem lainnya dalam jaringan sebagai paket-paket data. Bentuk paket-paket PDU untuk data yang ditransmisikan melalui jaringan dapat dilihat pada gambar berikut.



Gambar 2.3. PDU TCP/IP [14]

Protokol TCP/IP menggunakan dua buah skema pengalamatan yang dapat digunakan untuk mengidentifikasikan sebuah komputer dalam sebuah jaringan atau jaringan dalam sebuah internetwork, yakni sebagai berikut:

a. Pengalamatan IP

   IP berupa alamat logis yang terdiri atas 32-bit (empat oktet berukuran 8-bit) yang umumnya ditulis dalam format www.xxx.yyy.zzz. Dengan menggunakan subnet mask yang diasosiasikan dengannya, sebuah alamat IP pun dapat dibagi menjadi dua bagian, yakni *Network Identifier* (NetID) yang dapat mengidentifikasikan jaringan lokal dalam sebuah *internetwork* dan *Host identifier* (HostID) yang dapat mengidentifikasikan host dalam jaringan tersebut. Sebagai contoh, alamat 205.116.008.044 dapat dibagi dengan menggunakan subnet mask 255.255.255.000 ke dalam Network ID 205.116.008.000 dan Host ID 44. Alamat IP merupakan kewajiban yang harus ditetapkan untuk sebuah host, yang dapat dilakukan secara manual (statis) atau menggunakan *Dynamic Host Configuration Protocol* (DHCP) (dinamis).

b. *Fully qualified domain name* (FQDN)

   Alamat ini merupakan alamat yang direpresentasikan dalam nama alfanumerik yang diekspresikan dalam bentuk <nama_host>. <nama_domain>, di mana <nama_domain> mengindentifikasikan jaringan di mana sebuah komputer berada, dan <nama_host> mengidentifikasikan sebuah komputer dalam jaringan.

Pengalamatan FQDN digunakan oleh skema penamaan domain *Domain Name System* (DNS). Sebagai contoh, alamat FQDN id.wikipedia.org merepresentasikan sebuah host dengan nama "id" yang terdapat di dalam domain jaringan "wikipedia.org". Nama domain wikipedia.org merupakan second-level domain yang terdaftar di dalam top-level domain .org, yang terdaftar dalam root DNS, yang memiliki nama "." (titik).

Penggunaan FQDN lebih bersahabat dan lebih mudah diingat ketimbang dengan menggunakan alamat IP. Akan tetapi, dalam TCP/IP, agar komunikasi

dapat berjalan, FQDN harus diterjemahkan terlebih dahulu (proses penerjemahan ini disebut sebagai resolusi nama) ke dalam alamat IP dengan menggunakan server yang menjalankan DNS, yang disebut dengan *Name Server* atau dengan menggunakan berkas hosts (/etc/hosts atau %systemroot%\system32 \drivers\etc\hosts) yang disimpan di dalam mesin yang bersangkutan.

Berikut ini adalah layanan tradisional yang dapat berjalan di atas protokol TCP/IP:

a. Pengiriman berkas (file transfer).

   File Transfer Protocol (FTP) memungkinkan pengguna komputer yang satu untuk dapat mengirim ataupun menerima berkas ke sebuah host di dalam jaringan. Metode otentikasi yang digunakannya adalah penggunaan nama pengguna (user name) dan [[password]], meskipun banyak juga FTP yang dapat diakses secara anonim (anonymous), alias tidak berpassword.

b. *Remote login*.

   Network terminal Protocol (telnet) memungkinkan pengguna komputer dapat melakukan log in ke dalam suatu komputer di dalam suatu jaringan secara jarak jauh. Jadi hal ini berarti bahwa pengguna menggunakan komputernya sebagai perpanjangan tangan dari komputer jaringan tersebut.

c. *Computer email*.

   Digunakan untuk menerapkan sistem surat elektronik.

d. *Network File System* (NFS).

   Pelayanan akses berkas-berkas yang dapat diakses dari jarak jauh yang memungkinkan klien-klien untuk mengakses berkas pada komputer jaringan, seolah-olah berkas tersebut disimpan secara lokal.

e. *Remote execution*.

   Memungkinkan pengguna komputer untuk menjalankan suatu program tertentu di dalam komputer yang berbeda. Biasanya berguna jika pengguna menggunakan komputer yang terbatas, sedangkan ia memerlukan sumber yg banyak dalam suatu sistem komputer.

Ada beberapa jenis *remote execution*, ada yang berupa perintah-perintah dasar saja, yaitu yang dapat dijalankan dalam system komputer yang sama dan ada pula yang menggunakan sistem *Remote Procedure Call* (RPC), yang memungkinkan program untuk memanggil subrutin yang akan dijalankan di sistem komputer yg berbeda. (sebagai contoh dalam Berkeley UNIX ada perintah rsh dan rexec.)

f. Name server

Name server berguna sebagai penyimpanan basis data nama host yang digunakan pada Internet

RFC (*Request For Comments*) merupakan standar yang digunakan dalam Internet. Diterbitkan oleh IAB yang merupakan komite independen yang terdiri atas para peneliti dan profesional yang mengerti teknis, kondisi dan evolusi Internet. Sebuah surat yang mengikuti nomor RFC menunjukan status RFC :

a. S : Standard, standar resmi bagi internet
b. DS : Draft standard, protokol tahap akhir sebelum disetujui sebagai standar
c. PS : Proposed Standard, protokol pertimbangan untuk standar masa depan
d. I : Informational, berisikan bahan-bahan diskusi yang sifatnya informasi
e. E : Experimental, protokol dalam tahap percobaan tetapi bukan pada jalur standar.
f. H : Historic, protokol-protokol yg telah digantikan atau tidak lagi dipertimbankan utk standarisasi.

2.5. APPLICATION PROTOCOL

2.5.1. EMAIL PROTOCOL POP3

POP3 (*Post Office Protocol version 3*) adalah protokol yang ada di layer aplikasi. Dimaksudkan untuk mengizinkan client dapat mengakses secara dinamis email yang masih ada di POP3 server. POP3 menawarkan pada

user untuk meninggalkan email-nya di POP3 server, dan mengambil 4email-nya tersebut dari sejumlah sistem sembarang. Untuk mengambil 4email dengan menggunakan POP3 dari suatu client, banyak pilihan yang dapat digunakan seperti Sun Microsystem Inc.'s Emailtool, QualComm Inc.'s Eudora, Netscape Comm. Corp.'s Netscape Email dan Microsoft Corp.'s Outlook Express.

POP3 tidak dimaksudkan untuk menyediakan operasi manipulasi 4email yang ada di server secara luas. Pada POP3, email diambil dari server dan kemudian dihapus (bisa juga tidak dihapus). Segala sesuatu tentang protokol POP3 ini dibahas dalam RFC (Request For Comment) 1725. Protokol yang lebih tinggi dan lebih kompleks, yaitu IMAP4, dibahas dalam RFC 1730.

Ada dua jenis mode pada POP3 yaitu mode offline dan mode inline. Pada mode offline, POP3 mengambil dan kemudian menghapus email yang tersimpan dari server. POP3 bekerja dengan baik pada mode ini, karena terutama memang didisain untuk berlaku sebagai sebuah sistem email yang memiliki sifat "*store-and-forward*". Server, pada mode offline, berlaku seperti sebuah tempat penampungan yang menyimpan email sampai user memintanya.

Pada mode inline, POP3 akan mengambil email dari server tanpa menghapus email yang sudah diambil tersebut. Mode ini lebih disukai oleh user yang sering berpindah tempat (nomadic user) karena memungkinkan mereka untuk melihat email yang sama dari tempat atau komputer yang berbeda. Akan tetapi untuk nomadic user yang selalu bekerja dan bepergian dengan selalu membawa notebook, dan tetap menginginkan agar email miliknya yang ada di server tidak dihapus, tentu saja menginginkan agar setiap kali mengambil email tidak semua email yang akan terambil, tapi hanya email yang belum pernah dia lihat saja yang akan diambil. Keinginan user seperti ini dapat dipenuhi dengan menggunakan informasi pada client yang memungkinkan untuk memberi tanda email yang sudah pernah dilihat. Setiap client layanan POP3 yang mendukung mode inline akan menyimpan informasi ini dalam sebuah file.

Pada user yang menggunakan Netscape Email, file yang menyimpan informasi ini adalah file popstate.dat, yang biasanya terdapat di /Program

Files/Netscape/Users/Email. File tersebut memberi tahu email yang mana saja yang sudah diambil sehingga tidak perlu diambil lagi. Jika file ini dihapus maka tentu saja pada pengambilan email berikutnya semua email akan terambil. Pada file ini kode dibelakang huruf k merupakan *unique-id*. *Unique-id* ini secara unik mengidentifikasi sebuah email dalam *emaildrop* sehingga masing-masing email memiliki *unique-id* yang berbeda. Jika misalnya email kita yang berada di komputer lokal sudah terhapus sedangkan kita ingin membacanya lagi, maka sebelum kita mengambil emaildrop dari server, file popstate.dat ini harus dihapus terlebih dahulu. Apabila kita belum menghapus file tersebut maka akan ada pesan : " no new messages on server ", yang diberikan oleh Netscape Email. Untuk pemakai Eudora, file yang menyimpan informasi ini adalah file lmos.dat, sedangkan untuk pengguna Outlook Express biasanya menggunakan file pop3uidl.dat.

Pada awalnya, server memulai layanan POP3 dengan mendengarkan permintaan pada TCP port 110. Ketika sebuah client meminta layanan tersebut, maka terjadilah hubungan TCP dengan server. Pada saat hubungan dimulai, POP3 server mengirim greeting (kata pembuka). Setelah itu client akan memberikan command (perintah) ke server dan POP3 server akan memberikan response (jawaban) sampai hubungan ditutup atau digagalkan. Perlu diingat bahwa user tidak memasukkan perintah ini, tapi software dari klien-lah yang mengirim perintah ini ke server.

Perintah-perintah di POP3 terdiri dari sebuah keyword yang tidak *case sensitive* (tidak mempersoalkan huruf kapital ataupun tidak), yang dapat diikuti oleh satu atau lebih argument. Keyword dan argument masing-masing dipisahkan oleh karakter SPACE (spasi). Keyword terdiri dari tiga atau empat karakter, sedangkan tiap argument dapat mencapai 40 karakter. Jawaban di POP3 terdiri dari sebuah indikator status dan sebuah keyword yang dapat diikuti oleh informasi tambahan. Ada dua indikator status : positif ("+OK") dan negatif ("-ERR"). Server harus memberikan jawaban +OK dan -ERR dalam huruf kapital. Pada perintah tertentu, server akan memberikan jawaban yang terdiri dari beberapa baris.

Sebuah sesi hubungan POP3 dibangun melalui tiga tahap, yaitu tahap *authorization*, *transaction* dan *update*. Sekali hubungan TCP dimulai dan POP3 server telah mengirimkan greeting , maka sesi hubungan telah memasuki tahap *authorization*. Pada tahap ini client mengirim nama dan password user ke server untuk membuktian keaslian user tersebut agar dapat mengambil email-nya. Ketika client telah berhasil membuktikan identitas dirinya, server akan memperoleh informasi yang berhubungan dengan email yang dimiliki client tersebut, dan sesi kini memasuki tahap transaction. Pada tahap inilah terjadi proses penerimaan email, penandaan email untuk penghapusan, pembatalan penandaan untuk penghapusan, penampilan statistik email atau perincian identitas email. Pada saat client telah memberikan perintah quit untuk mengakhiri hubungan, maka sesi memasuki tahap update. Pada tahap inilah server akan menjalankan semua perintah yang diperoleh selama tahap transaction dan menutup sesi dan selanjutnya hubungan TCP ditutup.

Sebuah server harus menjawab perintah yang tidak dikenal, tidak diimplementasi, atau tidak sesuai dengan sintaksis dengan indikator status negatif. Server juga harus memberikan indikator status negatif, jika ada client yang memberikan perintah tidak pada tahap yang seharusnya. Tidak ada metoda umum yang dapat digunakan oleh client untuk membedakan antara server yang tidak mengimplementasikan perintah tambahan dengan server yang tidak dapat atau tidak bersedia memproses perintah tambahan tersebut.

Sebuah POP3 server mungkin memiliki *autologout timer* untuk client yang sedang tidak aktif dalam rentang waktu tertentu. Timer seperti ini harus paling sedikit memiliki rentang waktu 10 menit. Jika sebuah server menerima sebarang perintah dari client didalam rentang waktu tersebut, maka hal ini sudah cukup untuk me-reset autologout timer tersebut. Ketika waktu rentang timer sudah habis, tanpa ada aktivitas dari client maka sesi hubungan tidak memasuki tahap UPDATE. Server akan menutup hubungan TCP tanpa menghapus email atau mengirim jawaban ke client.

Semua pesan yang disampaikan selama sesi hubungan POP3 harus disesuaikan dengan standar format dari Internet text messages. Internet text messages ini, secara terperinci dibahas dalam RFC 822. Tabel 2.2. dibawah ini

memperlihatkan perintah-perintah pada POP3 berikut tahap tempat perintah tersebut digunakan.

Tabel 2.2 Perintah pada POP3

Perintah	Tahap
USER <user name>	AUTHORIZATION
PASS <password>	
QUIT	
STAT	TRANSACTION
LIST [msg]	
RETR [msg]	
DELE [msg]	
NOOP	
RSET	
QUIT	UPDATE

2.5.2. EMAIL PROTOCOL SMTP

SMTP (*Simple Email Transfer Protocol*) merupakan salah satu protokol yang umum digunakan untuk pengiriman surat elektronik di Internet. Protokol ini dipergunakan untuk mengirimkan data dari komputer pengirim surat elektronik ke server surat elektronik penerima.

SMTP pertamakali didefinisikan oleh RFC 821 (1982, sering dikenal sebagai STD10) dan terakhirkali diperbaharui oleh RFC 5321 (2008) dimana terdapat *Extended SMTP* (ESMTP) tambahan dan SMTP merupakan protocol yang banyak dipakai sekarang ini. SMTP lebih spesifik untuk transportasi email keluar dan menggunakan port TCP. Koneksi SMTP diamankan oleh SSL yang lebih dikenal sebagai SMTPS.

Protokol ini timbul karena desain sistem surat elektronik yang mengharuskan adanya server surat elektronik yang menampung sementara sampai surat elektronik diambil oleh penerima yang berhak. Ketika email dan

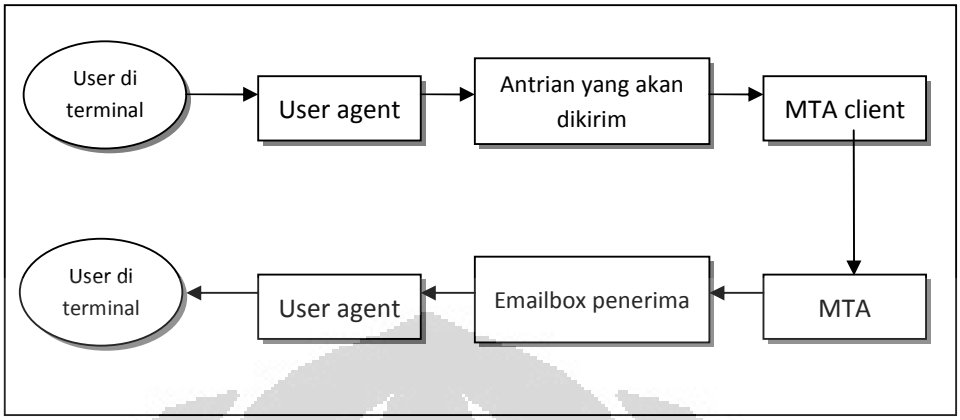
email transfer agent yang lain menggunakan SMTP untuk mengirim dan menerima pesan surat, user hanya menggunakan SMTP untuk mengirimkan pesan ke email server untuk membalasnya.

SMTP bisa dianalogikan sebagai kantor pos. Ketika kita mengirim sebuah e-email, komputer kita akan mengarahkan e-email tersebut ke sebuah SMTP server, untuk diteruskan ke email-server tujuan. Email-server tujuan ini bisa dianalogikan sebagai kotak pos di pagar depan rumah, atau kotak PO BOX di kantor pos. Email-email yang terkirim akan menempati di tempat tersebut hingga si pemiliknya mengambilnya. Urusan pengambilan e-email tersebut tergantung kapan di penerima memeriksa account e-emailnya.

SMTP merupakan text-based protocol, dimana setiap pengirim email berkomunikasi dengan penerima email dengan menggunakan *command string* dan mensuplay data-data yang penting diatas data-data yang dapat diandalkan untuk memerintah data stream chanel, spesifiknya Transmission Control Protocol (TCP). Sesi SMTP terdiri dari perintah yang berasal dari klien SMTP (agen yang memulai, pengirim dan transmitter) dan tanggapan yang sesuai dari server SMTP sehingga sesi dibuka dan parameter sesi dipertukarkan.

SMTP hanya protokol yang melakukan "push", artinya dia hanya bisa mengambil email dari client tetapi tidak bisa melakukan "pull", yaitu melayani pengambilan email di server oleh client. Pengambilan pesan atau email tersebut dilakukan dengan menggunakan protokol tersendiri yaitu protokop POP3 (*Post Office Protocol*) atauIMAP (*Internet Message Access Protocol*).

Cara kerja SMTP mirip yang dilakukan oleh FTP. SMTP menggunakan beberapa *spool* dan *queue*. Pesan yang dikirim oleh SMTP akan dikirimkan dalam *queue*. SMTP akan menghindari membalas pesan dari queue jika dihubungkan ke *remote machine*. Jika pesan tidak dapat dibalas dengan waktu yang telah ditentukan maka pesan akan dikembalikan ke pengirim atau dipindahkan. Interaksi antara message ke *User Agent* dan ke *Message Transfer Agent* hingga diterima oleh Penerima.



Gambar 2.4. Komponen Konseptual Sistem Email [15]

Email server hanya sebuah aplikasi yang berurusan dengan lalu lintas email, tidak secara langsung berhubungan dengan user yang akan berkirim email. Dalam pengiriman email, terdapat dua aplikasi yang diperlukan yaitu MTA (*Email Transfer Agent*), dan MUA (*Email User Agent*). Kerja sama antara MUA dan MTA dapat dianalogikan seperti agen perjalanan dan perusahaan perjalanan, dimana email merupakan orang yang akan melakukan perjalanan.

Secara garis besar MTA adalah sebuah aplikasi untuk mengantarkan email dan berfungsi sebagai berikut :

a. Pertukaran email menggunakan protokol TCP
b. Menerima email masuk (*incoming*)
c. Meneruskan email yang akan keluar (*outgoing*)
d. Mengatur antrian bila ada email masuk, keluar dan yang tertunda pengirimannya
e. MTA yang umum dipakai adalah sendemail dan qemail untuk Unix serta untuk di Ms Windows menggunakan Mdaemon.
f. Sedangkan MUA adalah aplikasi yang berfungsi sebagai interface antara email, dalam hal ini berhubungan dengan user yang memiliki email tersebut, dengan MTA yang mendukungnya. Berfungsi sebagai berikut :

- Menulis email dan membaca email yang masuk.
- Mengatur konfigurasi email sehingga sesuai dengan MTA yang mendukungnya.
- Memberikan kenyamanan kepada user dalam menerima dan mengirim email.

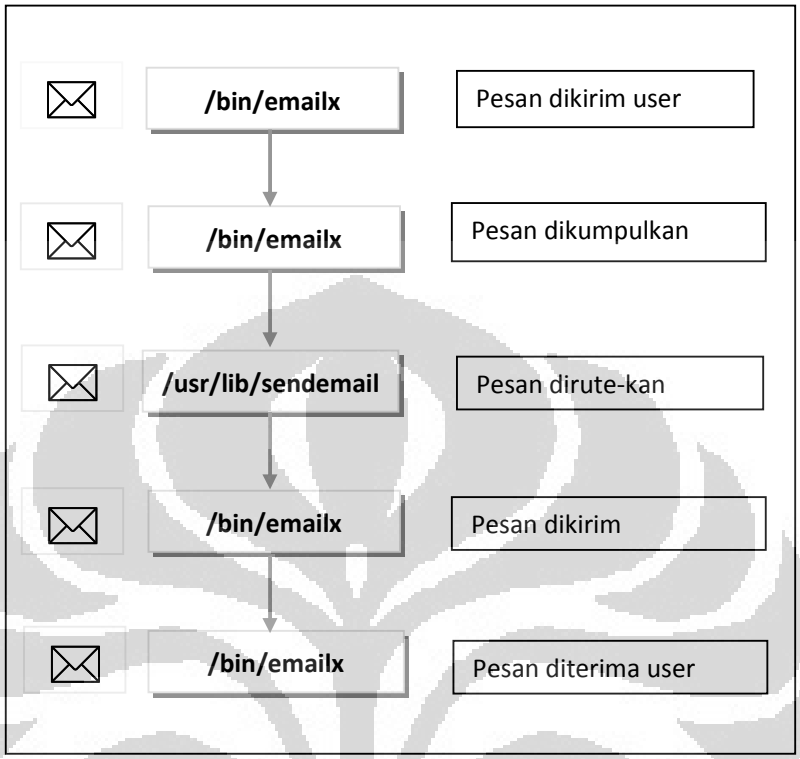
Beberapa agen email yang populer saat ini adalah Pine, Eudora, Netscape, Outlook dan Pegasus.

MTA akan menerima pesan yang berasal dari user di luar mesin melalui UUCP (via remail), user di luar mesin melalui TCP/IP dengan SMTP, dan user di mesin lokal melalui program MUA. Oleh MTA pesan tersebut akan dipilah-pilah berdasarkan 'rule' yang telah ditentukan, juga dengan memanfaatkan 'alias' yang telah didefinsikan. MTA akan merutekan proses pengiriman pesan hingga pesan tersebut dalam posisi : diluar sistem pengiriman dan penerimaan email.

Apakah dikirimkan lagi melalui TCP/IP atau UUCP (misal pesan dari user lokal yang ditujukan kepada user di luar mesin tersebut), atau

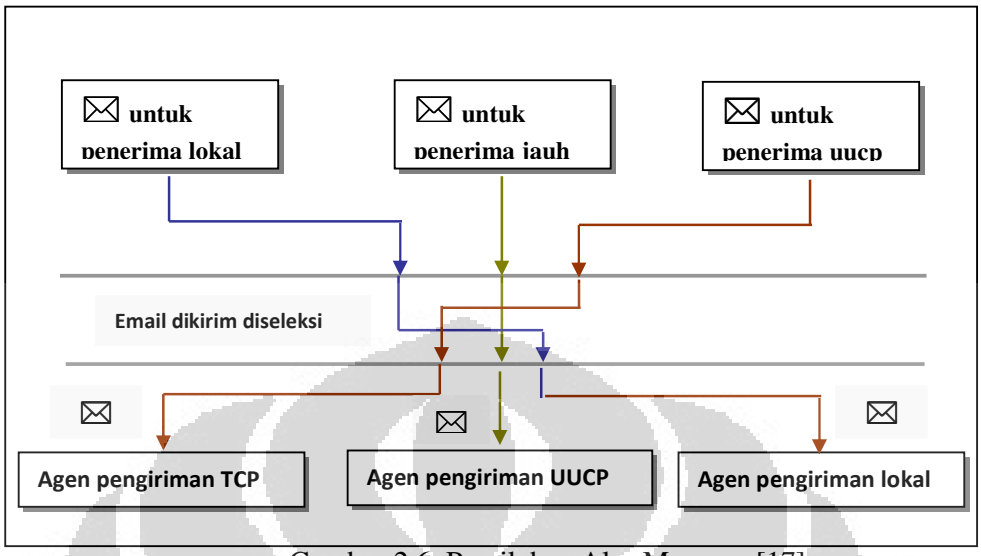
Langsung dikirimkan ke emailbox user lokal (misal pesan dari user lokal untuk user lokal lainnya).

Pada sistem Linux pengguna memiliki kebebasan untuk mengganti komponen tersebut sesuai kebutuhan. Pemisahan komponen sistem email ini mengakibatkan sistem email di Linux (Unix) menjadi luwes dan tidak terikat pada suatu solusi yang bersifat proprietary. Sehingga penambahan atau perubahan komponen mudah dilakukan untuk menyesuaikan dengan kebutuhan pengguna, misal untuk pemasangan anti virus atau pencegahan attachment, ataupun spam.



Gambar 2.5. Alur Data Pengiriman Email [16]

Pada sendemail (yang berfungsi sebagai pesan tranfer agent - MTA) terjadi proses pemilahan alur pesan, email yang ditujukan untuk user di luar mesin tersebut akan dikirimkan melalui TCP atau UUCP. Ini bergantung jarak pada sendemail. Sedang email yang ditujukan kepada user local akan diberikan pada email delivery agent untuk diproses dan dimasukkan ke emailbox dari user lokal tersebut. Proses pengolahan tambahan dapat dilakukan sebelum email tersebut dimasukkan ke emailbox user lokal jika ingin mencegahan virus attachment. Hal ini sangat mungkin untuk diterapkan dalam lingkungan Linux karena hubungan antara MTA dan MDA bersifat Open dan tidak menggunakan koneksi yang bersifat proprietary dan tidak diketahui oleh umum.



Gambar 2.6. Pemilahan Alur Message [17]

Email terdiri dari tiga buah komponen, yaitu

a. Envelope, atau amplop. Ini digunakan oleh MTA untuk pengiriman.
b. Header, digunakan oleh user agent. Ada kurang lebih sembilan field header, yaitu: Received, Message-Id, From, Date, Reply-To, X-Phone, X-emailer, To dan Subject. Setiap field header berisi sebuah nama yang diikuti oleh sebuah titik dua (:), dan nilai dari field header tersebut.
c. Body merupakan isi pesan dari pengirim ke penerima.

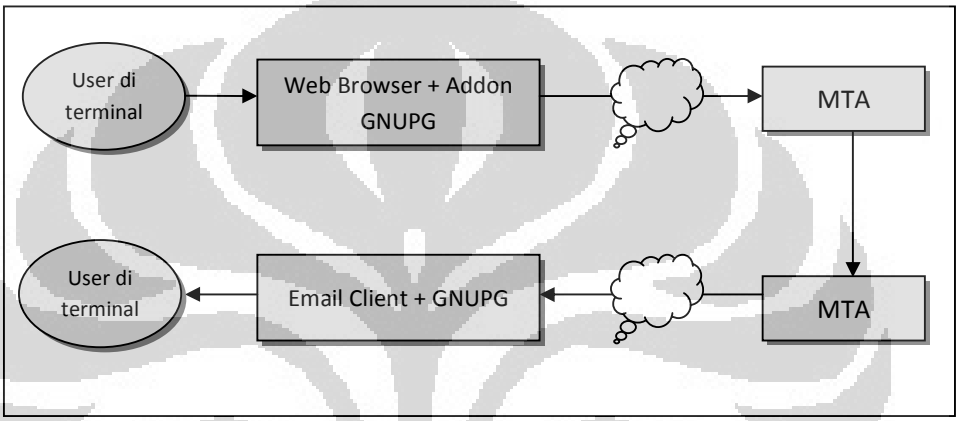
2.5.3. SECURITY PROTOCOL SSL

Protokol SSL (*secure socket layer*) berguna untuk mengenkripsi komunikasi antara pengguna dan sebuah situs. Contohnya pada situs E-commerce. Data yang dikirim melalui sambungan SSL dilindungi oleh enkripsi, sebuah mekanisme yang mencegah *eavesdropping* dan sabotase terhadap data yang dikirimkan [18]. SSL memungkinkan bahwa data pribadi yang dikirim kepada sebuah situs E-commerce, seperti nomor kartu kredit, akan dirahasiakan dan aman. Pengunjung situs dapat dengan mudah mendeteksi ketika sebuah situs mempunyai fasilitas SSL, yakni dengan adanya icon gembok emas

kecil atau address bar hijau dan alamat website dimulai dengan "https" bukan "http". Protokol SSL dapat digunakan pada sebuah Email Server yang menggunakan POP3 dan SMTP untuk keamanan pengiriman email.

BAB III
PERANCANGAN APLIKASI GPG WEBMAIL

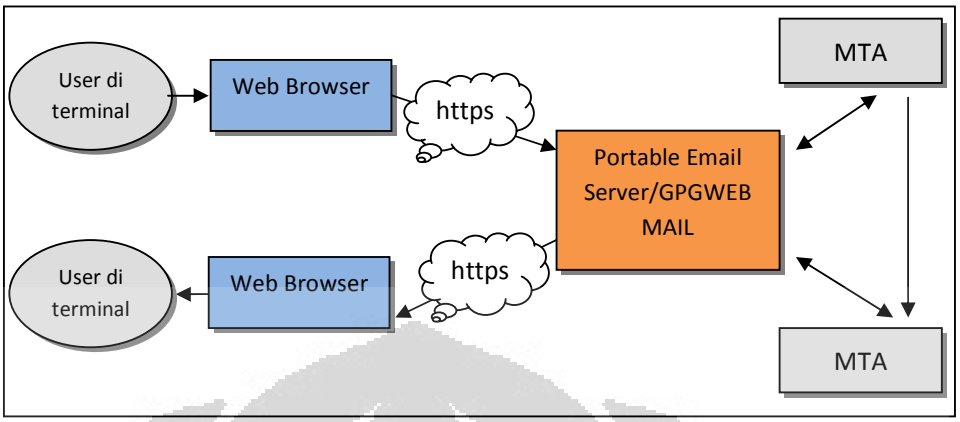
Sebelum memulai perancangan aplikasi, penulis akan membandingkan terlebih dahulu perbedaan infrastruktur yang telah ada dengan infrastuktur yang akan penulis rancang.



Gambar 3.1. Infrastruktur GNUPG on User

Gambar 3.1 menunjukkan infrastruktur yang saat ini sudah ada. Konsekuensi atas infrastruktur ini mempengaruhi perilaku user apabila ingin menggunakan GNUPG, yakni :

a. Aplikasi GNUPG harus diinstal di PC/Laptop di user. Ini mengakibatkan PC/Laptop user menjadi ruang privat yang harus diamankan karena secret key otomatis tersimpan di PC/Laptop.
b. Instalasi GNUPG juga merepotkan user jika harus menggunakan PC/Laptop fasilitas umum (misal : warnet).
c. Dan ada lagi, tidak semua Web Browser dan Email Client mempunyai addon GNUPG, tentunya ini membatasi user dalam memilih web browser.



Gambar 3.2. Infrastruktur GNUPG on Server

Pada gambar 3.2 terlihat bahwa GNUPG terinstal di server. Hal tersebut mengakibatkan perubahan perilaku user dalam menggunkan GNUPG, yakni :

a. User tidak perlu direpotkan dengan menginstal GNUPG.
b. User dapat menggunakan web browser apa saja untuk memanggil aplikasi.
c. User dapat dengan aman mengakses email dengan menggunakan PC/Laptop fasilitas umum (misal: warnet).

Dalam perancangan aplikasi yang akan diterapkan dalam infrastruktur yang baru ini, digunakan *Unified Modeling Language* (UML). UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya.

UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut [19].

UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.

Pada perancangan ini penulis merancang 2 (dua) macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

a. Use Case Diagram

b. Sequence Diagram

Pada perancangan *Use Case Diagram* dan *Sequence Diagram* ini penulis menggunakan *software Enterprise Architect v7.1* buatan *Sparx System.*

3.1. USE CASE DIAGRAM

Use Case Diagram yang berfungsi untuk memodelkan aplikasi berorientasi obyek. Use Case merupakan gambaran level tinggi dari apa yang akan sistem kerjakan, tanpa peduli tentang bagaimana sistem akan melakukannya secara detil.

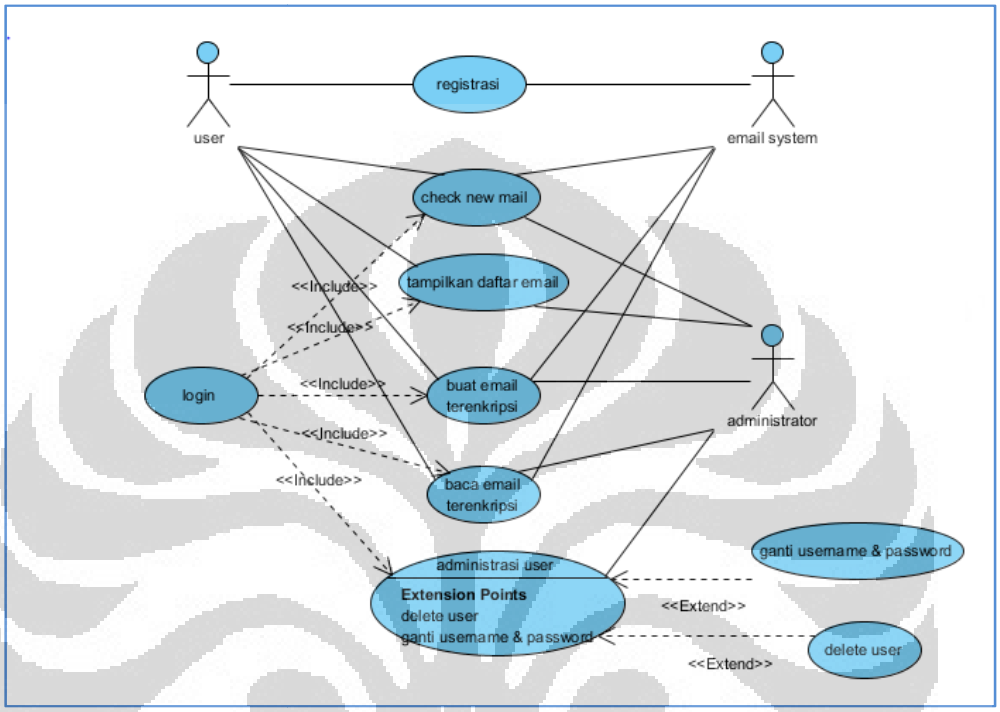
Use case juga menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.

Use case diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

Berikut gambar *Use case diagram* dari aplikasi yang akan dirancang.

1. *Use Case Diagram* Interaksi Klien dengan Aplikasi



Gambar 3.3. Use Case Diagram Interaksi Klien dengan Aplikasi

Keterangan :

a. Dari *Use Case diagra*m diatas tergambarkan bahwa terdapat user hanya sekali saja berhungan secara langsung dengan email system yang dalam hal ini adalah email server POP3-SMTP. Yakni pada saat user melakukan registrasi. Notifikasi link aktifasi setelah proses registrasi akan dikirim ke email server POP3-SMTP user. Oleh karena itu user harus login ke *email system* untuk melakukan aktifasi.

b. Setelah proses aktifasi maka user cukup mengirim dan membaca email terutama yang terenkripsi dari Aplikasi yang akan dirancang ini.

c. Aplikasi ini dirancang untuk berhubungan secara langsung dengan *Email System* terutama pada saat user melakukan *command* check new mail, buat email terenkripsi dan baca email terenkripsi (dekripsi). Ketiga proses

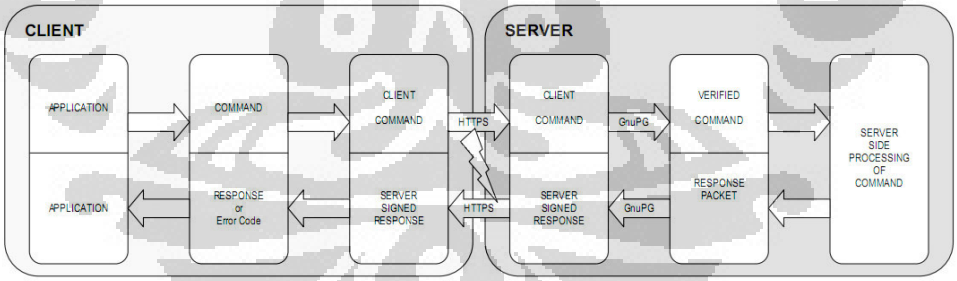
tersebut menggunakan protocol POP3 dan SMTP. User secara garis besar bisa melakukan 4 (empat) proses utama yang ada di dalam aplikasi ini, yakni :

- Check New email
- Tampilkan Daftar email
- Buat email Terenkripsi
- Baca email Terenkripsi

d. Sedangkan Administror dapat melakukan 6 (enam) proses utama yang ada dalam aplikasi ini, yakni :

- Check New email
- Tampilkan Daftar email
- Buat email Terenkripsi
- Baca email Terenkripsi
- Delete user
- Ganti username dan password

2. *Use case Diagram Client – Server Connection*



Gambar 3.4. *Use Case Diagram Client – Server Connection*

Keterangan :

a. *Use Case diagram* diatas menggambarkan *client – server connection.* Di dalam server terdapat software GnuPG yang akan melakukan proses enkripsi, dekripsi serta generate key.

b. Koneksi antara user dan server menggunakan protokol HTTPS. HTTPS merupakan protokol HTTP yang menggunakan *Secure Socket Layer*

(SSL). Sehingga diharapkan proses *Client – Server Connection* akan lebih aman.

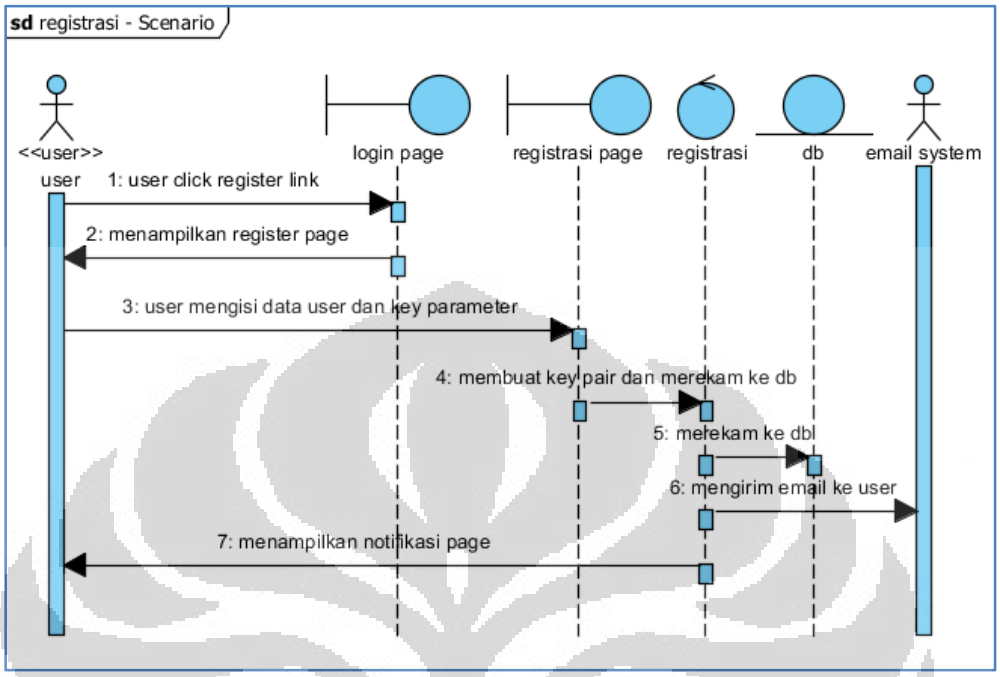
c. Server yang dimaksud di gambar di atas adalah distro linux yang didalamnya terdapat paket – paket sebagai berikut :
   - Software GnuPG
   - Web Server
   - Database
   - Aplikasi PHP yang menghubungkan client dengan GnuPG
   - Aplikasi PHP yang menghubungkan client dengan email Server POP3-SMTP
   - Aplikasi PHP yang menghubungkan client dengan database
d. Client yang dimaksud di gambar di atas adalah orang yang sudah terdaftar dalam database server. Client melakukan koneksi dengan user melalui web browser.

3.2. SEQUENCE DIAGRAM

Selanjutnya pembuatan *Sequence Diagram* untuk 8 (delapan) *Scenario* utama yang ada dalam Aplikasi ini. *Sequence diagram* merupakan sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah system. Interaksi tersebut berupa message yang digambarkan terhadap waktu.

Sequence diagram terdiri dari dimensi horizontal (objek-objek) dan dimensi vertical (waktu). Diagram ini juga menggambarkan urutan even yang terjadi. Dan lebih detail dalam menggambarkan aliran data, termasuk data atau behavior yang dikirimkan atau diterima.

1. Skenario Registrasi



Gambar 3.5 *Sequence Diagram* Proses Registrasi

Keterangan :

a. User yang akan menggunakan aplikasi ini harus mendaftar di halaman registrasi.

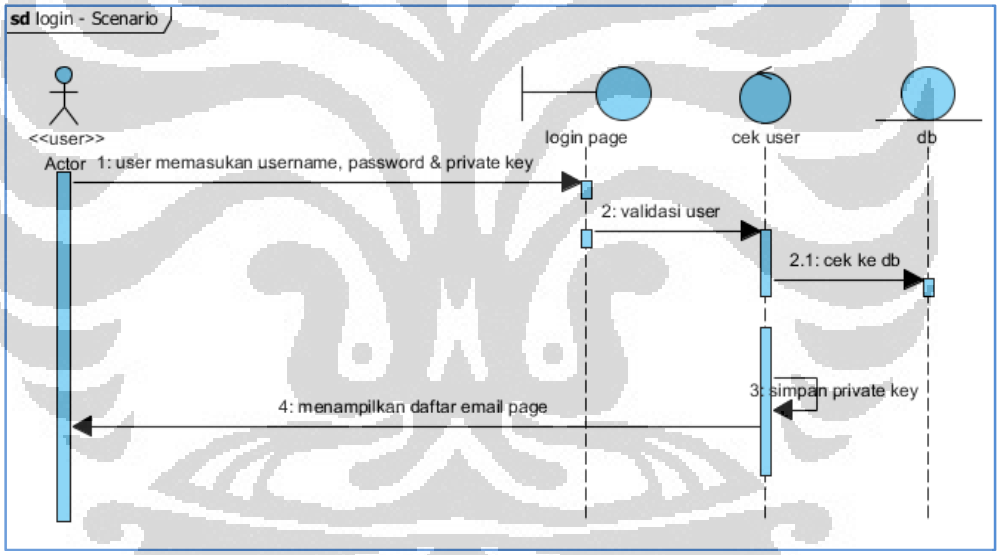
b. Terdapat 2 (dua) hal utama yang harus didaftarkan dalam proses registrasi:

- **Data User : Account Setting** (terdiri dari Username, Real Name, dan Password) dan **Email Setting** (terdiri dari Email account, POP3 Server, POP3 Port, POP3 Authentication, SMTP Server, SMTP Port, SMTP Authentication, Mail Username dan Mail Password)
- **Key Parameter** yakni mendaftarkan *GPG Key Setting* (terdiri dari Comment, Passphrase, Expire Dat, Key Type, Key Length, Subkey Type, Subkey Length)

c. Dari gambar 3.5 langkah ke 4 terjadi proses *keypair,* yakni proses GNUPG melakukan generate Key.

d. Setelah melakukan registrasi maka link aktifasi akan di kirim ke email user yang di daftarkan.
e. Saat mengaktifkan *account* maka user akan mendapatkan secret/privat key .
f. Secret/privat key harus disimpan sendiri oleh user, karena server aplikasi tidak menyimpannya.
g. Setelah secret/privat key di download oleh user maka server aplikasi akan menghapus secret/privat key tersebut dari software GnuPG.
h. Link Aktifasi akan *disable* setelah di gunakan.

2. Skenario Login

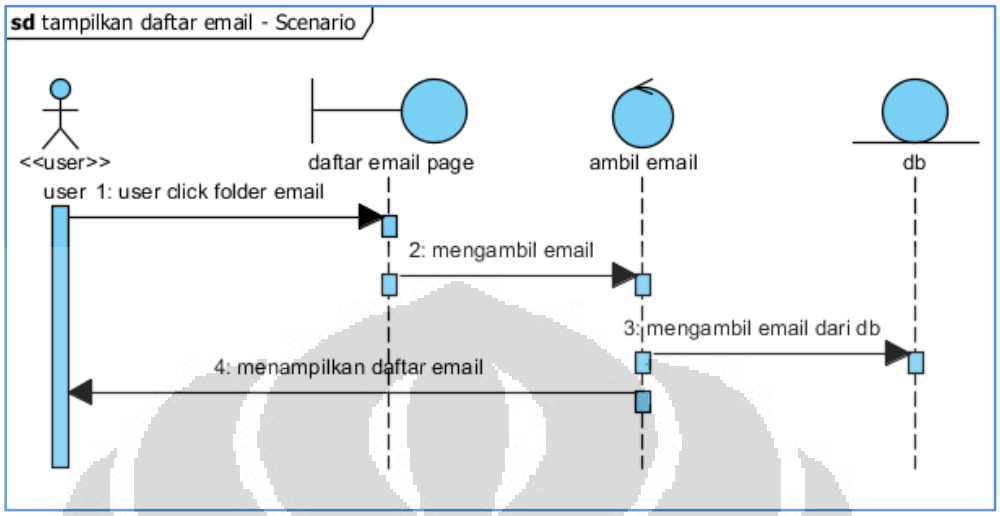


Gambar 3.6. *Sequence Diagram* Proses Login

Keterangan :

a. Login hanya bisa dilakukan setelah user melakukan aktifasi.
b. Login dilakukan dengan memasukkan username, password dan secret/privat key

3. Skenario Tampilkan Daftar Email

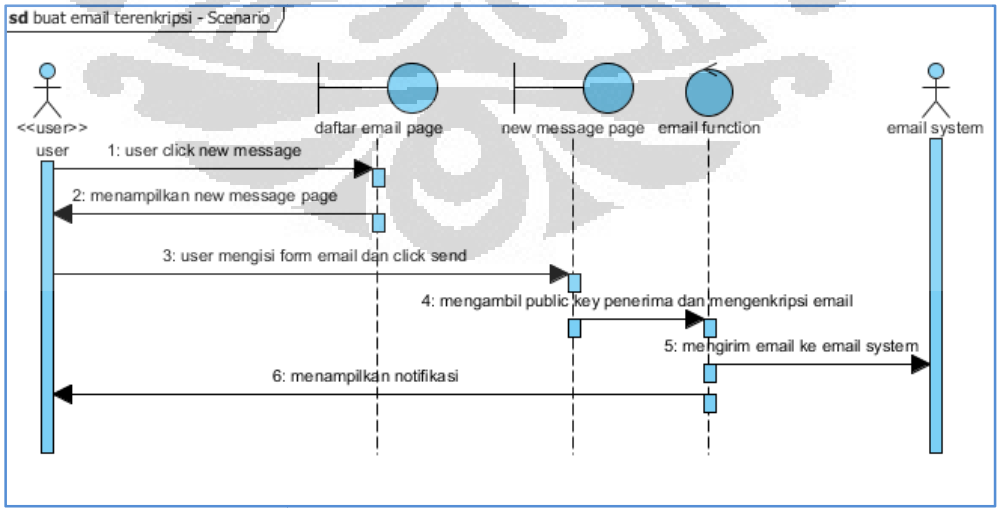


Gambar 3.7. Sequence Diagram Proses Daftar Email

Keterangan :

a. Setelah berhasil login maka user akan dibawa ke halaman "Tampilkan Daftar email"
b. Halaman ini berisi list email yang di ambil dari server email user yang sudah ditarik ke database aplikasi.
c. List email berisi pengirim dan subyek email

4. Skenario Buat Email Enkripsi

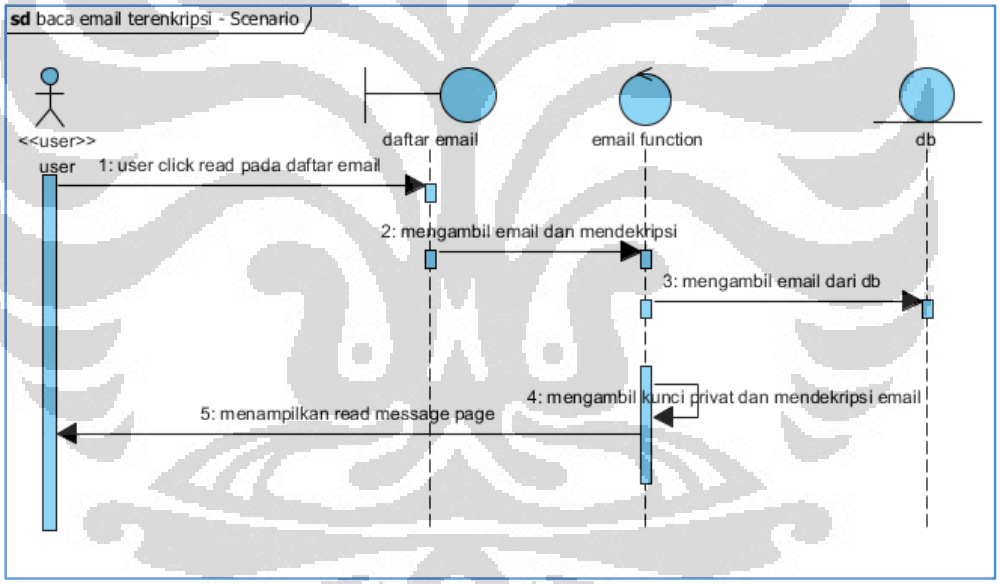


Gambar 3.8. *Sequence Diagram* Proses Buat Email Enkripsi

Keterangan :

a. User menekan tombol new message.
b. User akan masuk ke halaman “New Message Page”
c. User mengisikan alamat surat.
d. Alamat surat hanya berlaku bagi email user lain yang sudah terdaftar dalam Aplikasi ini.
e. Enkripsi dilakukan dengan Public Key milik alamat surat.
f. User mengisikan subyek surat.
g. User menulis pesan yang akan dienkripsi.

5. Skenario Baca Email Terenkripsi

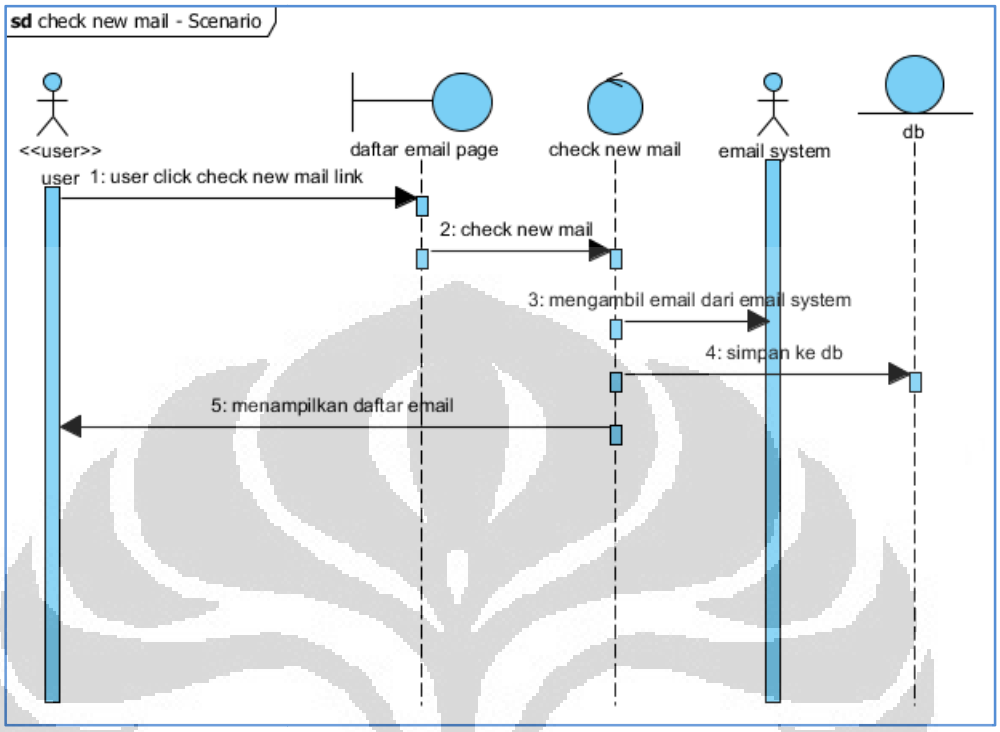


Gambar 3.9. *Sequence Diagram* Proses Baca Email Terenkripsi

Keterangan

a. Baca email terenkripsi bisa juga disebut dengan proses dekripsi.
b. User melakukan “klik read” pada daftar email yang ingin dibaca.
c. Bila Email yang dibaca adalah email yang terenkripsi maka secret/privat key user digunakan untuk mendekripsi.

6. Skenario Check New Email

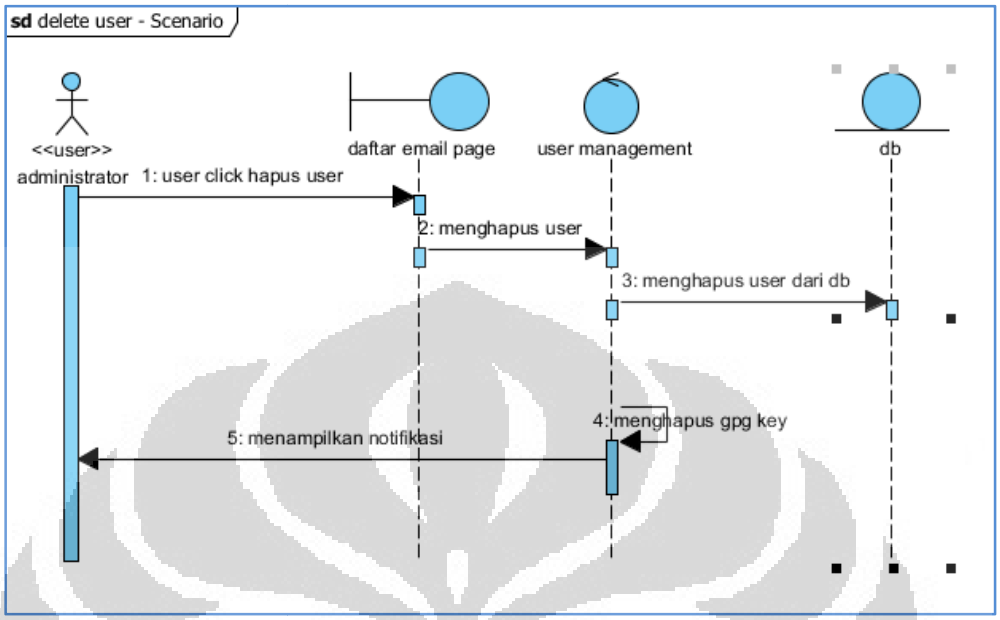


Gambar 3.10. *Sequence Diagram* Proses Check New Email

Keterangan :

a. User melakukan "click check new mail" link
b. Aplikasi akan mengambil/mendownload email dari email Server User.
c. Setelah terdownload maka akan disimpan di database aplikasi
d. Setelah masuk ke database aplikasi selanjutnya list email akan muncul di "daftar email Page"

7. Skenario Delete User

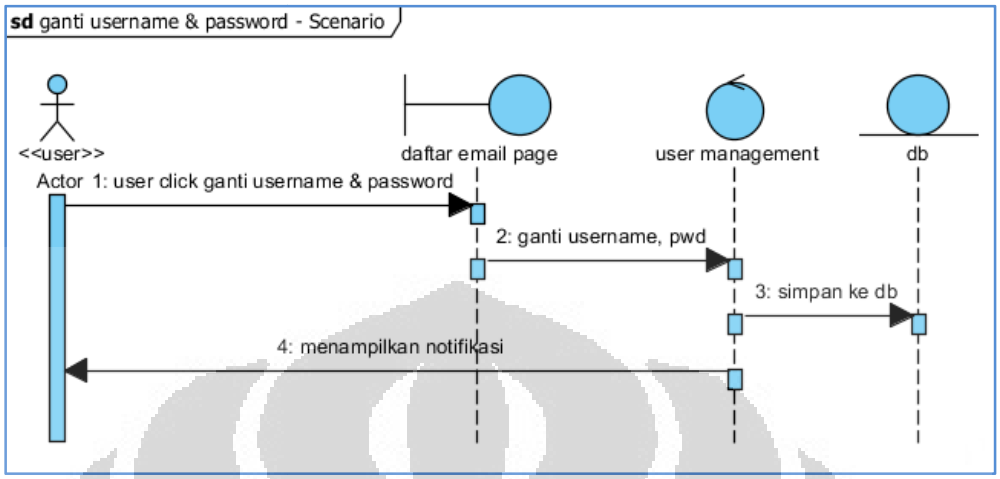


Gambar 3.11. *Sequence Diagram* Proses Delete User

Keterangan :

a. Skenario ini hanya bisa dilakukan oleh user yang bertindak sebagai Administrator
b. Administrator melakukan “click hapus user”
c. Administrator akan masuk ke halaman “daftar email page”.
d. Administrator memilih user mana yang akan dihapus.
e. Semua data user yang dihapus oleh administrator akan hilangdari database aplikasi.

8. Skenario Ganti Username dan password



Gambar 3.12. *Sequence Diagram* Proses Ganti Username dan Password

Keterangan :

a. Skenario ini hanya bisa dilakukan oleh user yang bertindak sebagai Administrator
b. Administrator melakukan “click ganti username dan password”
c. Administrator akan masuk “daftar email Page”
d. Administrator memilih account yang akan diganti username dan passwordnya.
e. Perubahan akan memperbarui data yang ada dalam database aplikasi.

BAB IV
IMPLEMENTASI DAN PENGUJIAN

4.1. IMPLEMENTASI

Tahap implementasi ini mencakup proses pembuatan aplikasi dan pembuatan distro server sebagai tempat aplikasi ini nantinya dijalankan. Aplikasi ini diberi nama GPG WEBMAIL, sedangkan distro portable email server diberi nama Distro GPG WEBMAIL SERVER.

4.1.1. Batasan Implementasi

Dalam mengimplementasikan perangkat lunak ini ada beberapa hal yang menjadi batasan implementasi, yaitu :

1. Basis data yang digunakan dalam pengimplementasian ini adalah MySQL.
2. Implementasi koneksi anatara klien dan server dilakukan dalam sebuah jaringan LAN.
3. Implementasi koneksi antara database dan *email system* menggunakan protokol POP3 dan SMTP
4. Fitur email yang dibangun pada versi awal ini baru menggunakan *field* To. Belum menggunakan field CC, BCC dan attachment.
5. Digunakan Apache web server untuk mengolah kode PHP atau HTML yang akan dikirim ke client.
6. Browser yang digunakan adalah mozilla firefox, sebagai media untuk menampilkan program hasil implementasi.
7. Perangkat keras yang digunakan sebagai server mempunyai spesifikasi : processor Core 2 Duo, Memori 1 GB DDR2, Hard Drive untuk media penyimpanan minimal 10 MB.
8. Digunakan LAMPP (linux, Apache, MySQL, PHP, Pearl) sebagai web server. Protokol SSL juga memanfaatkan fitur di LAMPP.

4.1.2. Implementasi *Logic Application*

Dari perancangan yang telah disusun di bab sebelumnya maka secara garis besar di tahap implementasi ini akan disusun 3 (tiga) hal pokok, yakni :

1. Aplikasi PHP yang menghubungkan client dengan GnuPG
2. Aplikasi PHP yang menghubungkan client dengan email Server POP3-SMTP
3. Aplikasi PHP yang menghubungkan client dengan database

4.1.2.1. Aplikasi PHP yang menghubungkan client dengan GnuPG

Implementasi untuk aplikasi PHP yang menghubungkan client dengan GnuPG tertulis sebagai *method* yang akan menjalankan *console* GnuPG dalam bentuk perintah di PHP. Berikut adalah parameter yang akan digunakan dalam proses implementasi, yakni :

1. *Constructor* GnuPG

Mendefinisikan path direktori dari program GnuPG yang diinstal dalam server. Termasuk mendefinisikan path dari keyring

- `string $program_path: Full program path for the GNUPG`
- `string $home_directory: Home directory of the keyring`

2. Decrypt

Parameter untuk mendefinisikan perintah untuk melakukan dekripsi (skenario baca email Terenkripsi)

- `string $KeyID: the key id to decrypt`
- `string $Passphrase: the passphrase to open the key used to decrypt`
- `string $Text: data to decrypt`

3. DeleteKey

Mendefinisikan perintah untuk melakukan deletekey. Perintah ini hanya bisa dilakukan oleh user yang mempunyai otoritas sebagai Admnistrator (skenario Delete User). Berikut adaalah parameternya :

- string $KeyID: the key id to be removed, if this is the secret key you must specify the fingerprint
- string $KeyKind: the kind of the keys, can be secret or public

4. Encrypt

Mendefinisikan perintah untuk melakukan enkripsi (scenario buat email terenkripsi). Berikut adaalah parameternya :

- string $KeyID: the key id used to encrypt
- string $Passphrase: the passphrase to open the key used to encrypt
- string $RecipientKeyID: the recipient key id
- string $Text: data to encrypt

5. Export

Mendefinisikan perintah untuk mendownload public key. Berikut adalah parameternya :

- string $KeyID: The Key ID to export

6. GenKey

Mendefinisikan perintah untuk membuat kunci. berjalan pada saat proses registrasi. Berikut adaalah parameternya :

- string $RealName: The real name of the user or key.
- string $Comment: Any explanatory commentary.
- string $Email: The email for the user.
- string $Passphrase: Passphrase for the secret key, default is not to use any passphrase.
- string $ExpireDate: Set the expiration date for the key (and the subkey).
- string $KeyType: Set the type of the key, the allowed values are DSA and RSA, default is DSA.
- int $KeyLength: Length of the key in bits, default is 1024.
- string $SubkeyType: This generates a secondary key, currently only one subkey can be handled ELG-E.
- int $SubkeyLength: Length of the subkey in bits, default is 1024.

7. Import

Mendefinisikan inputan key ke dalam keyring GnuPG. Berikut adaalah parameternya :

- `string $KeyBlock: The PGP block with the key.`

8. ListKeys

Mendefinisikan outputan public key yang ada di dalam keyring GnuPG. User dapat melihat output ini. Berikut adalah parameternya :

- `string $KeyKind: the kind of the keys, can be secret or public`

4.1.2.2. Aplikasi PHP yang menghubungkan client dengan email Server POP3-SMTP

Implementasi untuk Aplikasi PHP yang menghubungkan client dengan email Server POP3-SMTP tertulis sebagai *function* berikut ini :

1. Fungsi mendapatkan email dengan menyertakan koneksi ke incoming POP3. Koneksi POP3 nya meliputi username, password dan port yang akan digunkan untuk menarik isi email.

```
function getmail2db(&$dbobj, $userid,
$pop,$port,$username,$password,$auth="none"){
if($auth=="none")
$c = POP3::connect($pop, $username, $password, $port);
else
$c = POP3::connect($pop, $username, $password, $port,
$auth);
```

2. Fungsi mengirim email dengan menyertakan koneksi ke *outgoing SMTP*. Koneksi SMTP nya meliputi username, password dan port yang akan digunkan untuk megirim isi email.

```
function sendmail($from, $to, $subject,
$message,$smtp,$port,$username,$password,$auth="none")
{
$m = new MAIL;
$t = $m->AddTo($to);
$f = $m->From($from);
$s = $m->Subject($subject);
```

```
$m->Text($message);
// send mail using smtp function if($auth=="none")
$c = SMTP::connect($smtp, $port, $username,
$password);
```

4.1.2.3. Aplikasi PHP yang menghubungkan client dengan database

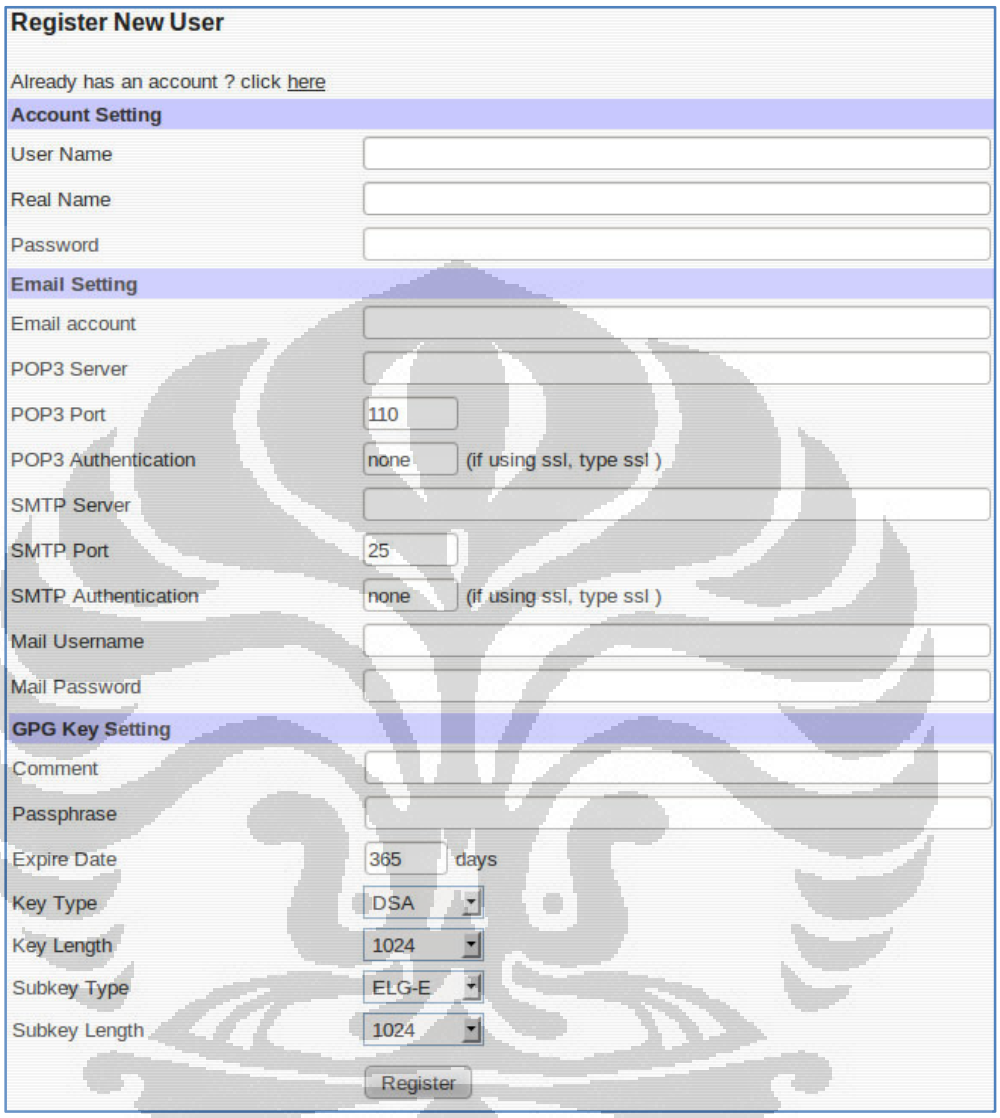
Implementasi untuk aplikasi PHP yang menghubungkan client dengan database. Script PHP yang dituliskan dalam tahap ini berfungsi sebagai konektor script aplikasi dengan database yang ada. Database dibangun dengan menggunakan MySQL.

```
<?php
//database server
define('DB_SERVER', "localhost");
//database login name
define('DB_USER', "username");
//database login password
define('DB_PASS', "password");
//database name
define('DB_DATABASE', "username_mydata");
//smart to define your table names also
define('TABLE_USERS', "users");
define('TABLE_NEWS', "news");?>
```

4.1.3. Implementasi *Template Application*

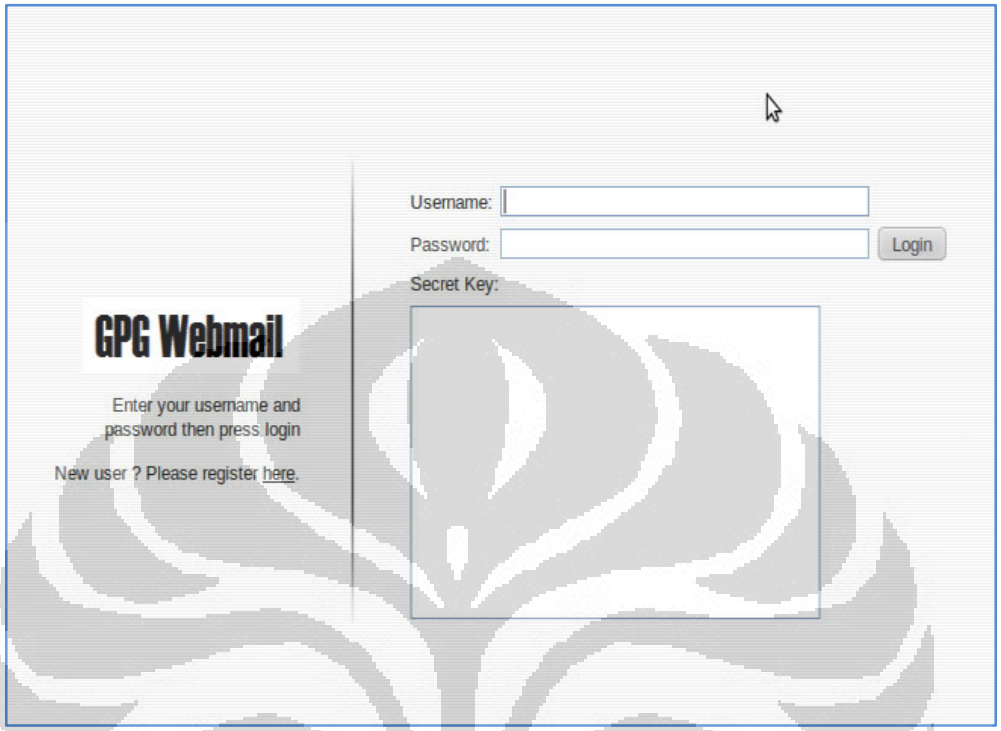
Implementasi yang akan dibuat pada tahap ini merupakan tampilan yang akan muncul di web browser klien. Disusun berdasarkan *Sequence diagram* yang telah dijelaskan di Bab III.

1. Scenario Registrasi



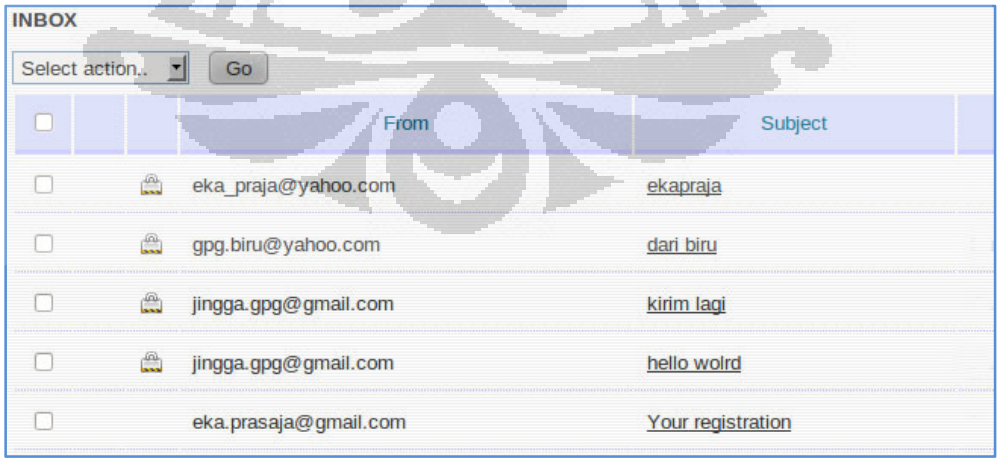
Gambar 4.1. Template Registrasi

2. Skenario Login



Gambar 4.2. *Template Login*

3. Skenario Tampilkan Daftar Email



Gambar 4.3. *Template* Tampilkan Daftar Email

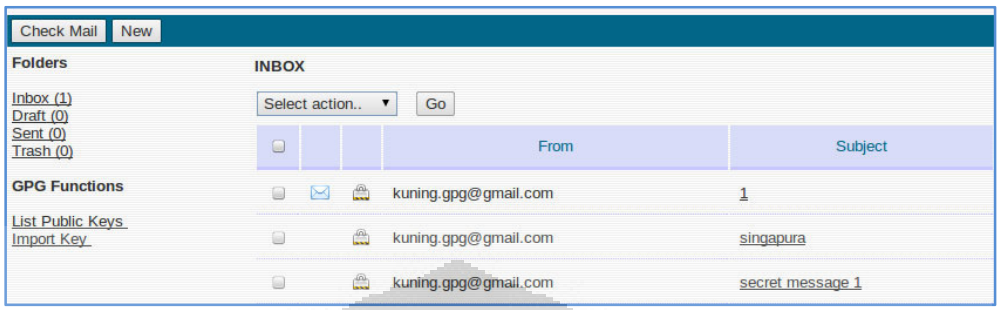
4. Skenario Buat email Enkripsi

Gambar 4.4. *Template* Buat Email Enkripsi

5. Skenario Baca email Terenkripsi

Gambar 4.5. *Template* Baca Email Terenkripsi

6. Skenario Check New Email



Gambar 4.6. *Template* Check New Email

7. Skenario Delete User

Gambar 4.7. *Template* Delete User

8. Skenario Ganti Username dan password

Gambar 4.8. *Template* Ganti Username dan Password

4.1.4. Implementasi Distro Linux Server

Berikut ini akan dijelaskan secara umum proses pembuatan distro server, pengintegrasian distro server dan termasuk kustomisasi yang dilakukan.

Proses pembuatan distro server secara umum terdiri dari dua tahap yaitu :

1. Persiapan spesifikasi kebutuhan system

   Pada tahap ini adalah mempersiapkan kebutuhan sistem untuk pebuatan distro server. Software-software yang diperlukan dalam proses pembuatan distro Linux Server Moonitoring sebagai berikut :

   a. Ubuntu 10.10 LTS
   b. Apache2 Web Server
   c. PHP5
   d. MySQL Database Server
   e. Aplikasi GPG Webmail versi 1.0.

2. Proses Pembuatan distro GPG Webmail Server

   Secara umum, proses kerja untuk pembuatan distro sebagai berikut :

   a. Membuat Direktory kerja.
   b. Mengunduh CD Image Ubuntu 10.10

c. Mendownload Paket-paket software yang diperlukan dan dependenciesnya.
d. Mount image Ubuntu CD.
e. Membuat copy dari image.
f. Menulis ulang isolinux/isolinux.cfg.
g. Membuat preseed/ gpgwebmailserver.seed.
h. Memasukkan paket-paket software yang diperlukan
i. Mengepak CD Image GPG Webmail Server

4.2. PENGUJIAN

Pengujian merupakan bagian yang penting dalam siklus pembangunan aplikasi. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari aplikasi. Tujuan dari pengujian ini adalah untuk menjamin bahwa aplikasi yang dibangun memiliki kualitas yang baik, yaitu mampu merepresentasikan kajian pokok dari spesifikasi, analisis, perancangan dari aplikasi itu sendiri.

Dalam pengujian aplikasi ini penulis menggunakan suatu metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak yang dibangun. Metode yang diambil adalah metode pengujian *Black Box*. Pengujian Black Box adalah pengujian yang sistemnya tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar.

Pengujian yang kedua adalah melihat sisi keamanan dari aplikasi yang dibangun ini. Uji keamanan yang dilakukan adalah melihat apakah data benar - benar terenkripsi dengan benar dan apakah koneksi yang dibangun antara client dan server juga aman

4.2.1. Pengujian *Black Box*

Pada metode ini data uji dibangkitkan, dieksekusi pada perangkat lunak, kemudian keluaran perangkat lunak dicek apakah sesuai dengan yang diharapkan.

Ada dua komponen yang harus diperhatikan dalam strategi pengujian, yaitu :

1. Faktor Pengujian yang merupakan hal-hal yang harus diperhatikan selama melakukan pengujian. Faktor pengujian ini dipilih sesuai dengan system yang akan diuji.
2. Tahapan pengujian yang merupakan langkah-langkah dalam melakukan pengujian.

Pengujian aplikasi ini, menggunakan data uji berupa sebuah data masukan dari menu aplikasi yang telah dibuat.

Tabel 4.1 Deskripsi Skenario Pengujian

Item Pengujian	**Deskripsi**	**Jenis Pengujian**
Proses Registrasi	Memeriksa proses registrasi	Black Box
Proses Login	Memeriksa proses login	Black Box
Proses Tampilkan Daftar Email	Memeriksa proses tampilkan daftar email	Black Box
Proses Buat email Enkripsi	Memeriksa proses buat email Enkripsi	Black Box
Proses Baca email Terenkripsi	Memeriksa proses Baca email Terenkripsi	Black Box
Proses Check New Email	Memeriksa Proses Check New Email	Black Box
Proses Delete User	Memeriksa Proses Delete User	Black Box
Proses Ganti Username dan password	Memeriksa Ganti Username dan password	Black Box

Tabel 4.2. Pengujian Proses Registrasi

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Mengisi Account Setting, Mengisi Email Setting, Mengisi GPG Key Setting	Generate Key Berhasil, account terdaftar	Sesuai dengan Gambar 4.1	[x] Diterima [ ] Ditolak

Tabel 4.3. Pengujian Proses Login

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Mengisi Username, password, secret/private key	Hak akses sesuai dengan bagiannya	Sesuai dengan Gambar 4.2	[x] Diterima [ ] Ditolak

Tabel 4.4. Pengujian Proses Tampilkan Daftar Email

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Login ke halaman utama user	List email sesuai dengan isi email yang terdaftar	Sesuai dengan Gambar 4.3	[x] Diterima [ ] Ditolak

Tabel 4.5. Pengujian Proses Buat email Enkripsi

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Mengisi email tujuan, subyek, message	email terenkripsi sukses terkirim	Sesuai dengan Gambar 4.4	[x] Diterima [ ] Ditolak

Tabel 4.6. Pengujian Proses Baca email Terenkripsi

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Mengisi Username, password, secret/private key	email terenkripsi sukses dibuka	Sesuai dengan Gambar 4.5	[x] Diterima [ ] Ditolak

Tabel 4.7. Pengujian Proses Check New Email

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Klik Check New Email	email baru terdownload ke database dan muncul di aplikasi	Sesuai dengan Gambar 4.6	[x] Diterima [ ] Ditolak

Tabel 4.8. Pengujian Proses Delete User

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Username admin, Password admin	Administrator dapat menghapus user account	Sesuai dengan Gambar 4.7	[x] Diterima [ ] Ditolak

Tabel 4.9. Pengujian Proses Ganti Username dan password

Data Masukan	Yang diharapkan	Hasil	Kesimpulan
Username admin, Password admin	Administrator dapat mengganti Username dan password user	Sesuai dengan Gambar 4.8	[x] Diterima [ ] Ditolak

4.2.2. Pengujian Keamanan

Pengujian ini menganalisa 2 (dua) hal utama yakni, **data enkripsi yang dihasilkan** dan **keamanan koneksi antara klien dan server**. Uji keamanan koneksi ini menggunakan teknik *Packet Sniffing* yang sering dilakukan *hacker* dalam upaya *Data Collection* [19].

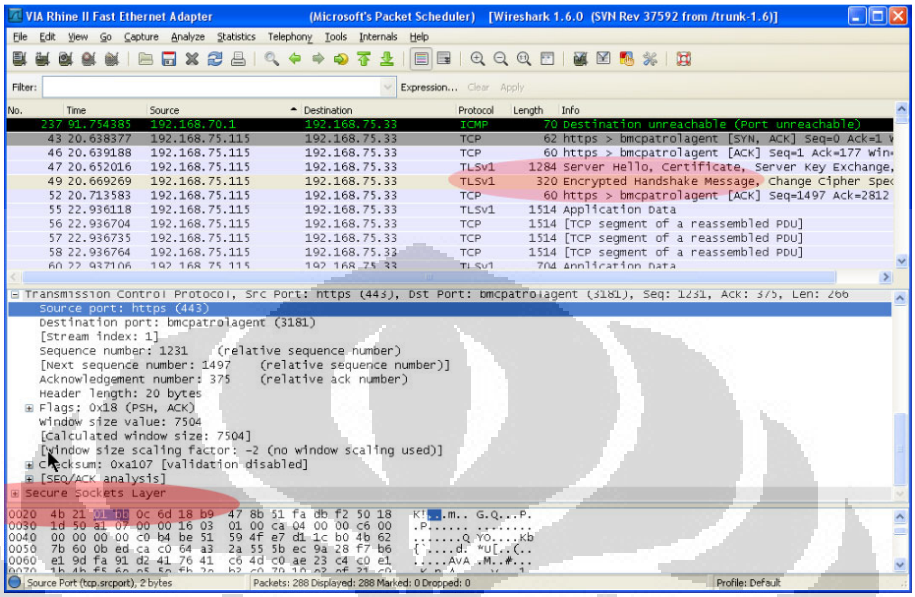
1. Data enkripsi yang dihasilkan dapat dilihat di inbox akun email POP3/SMTP user. Dari inbox tersebut terlihat bahwa data terenkripsi dengan benar sesuai output enkripsi email dari GnuPG. Ini adalah contoh hasilnya :

Tabel 4.10. Data Enkripsi Email dengan GNUPG

***Data Plain* :** *(data asli yang belum dienkripsi)*	***Data Enkripsi* :**
Sekelompok anggota parlemen meminta raja Bahrain Hamad bin Isa Al Khalifa menetapkan kondisi perang setelah aksi unjukrasa selama sepekan.	-----BEGIN PGP MESSAGE----- Version: GnuPG v1.4.10 (GNU/Linux) hQEOA/Io17YNgnxSEAP+LeOZza8OWYos exRMlFmcH+DoL/e64UWLGWxJXIuihdi/ VJ2t4worF/wt4gqKMBnzvSmIInn4MZu+ D9bKvU5puOBGrUHNCcXBpw8K7t1PHgmW uxGFJOwBPE9qZ0sK/Enu3fVM+smEIDNS KrdUi+YW1t8fHIRFy6WHNEn41VAjmOYD /i/MdcKkGbUYCvWmkbW9GqesR6GC4VZZ 2P0WkPUasxBQSCTY6DiDOTaniTvFm1by By2Hu5V+7h7jPeyuxQcjVJeWC9/ObUOL VdkAPginrA/3tT19D7I4/wPRrN7CDOFz rmiDGbjryBVbJKV1YrVmnkA1L+6G39PZ OA/WkaRxVjHO0oUB6mlT8HA4+esjx0cY nNeMieqfNxW7nGtSmuq88jEzj+7gyzDn e7HcVRVD2jfNccUYC44GHGCYebuyW6fU m8Z2Afjon5ABpW3Hem/BuBlx2xd9dQuK R1x5gTYpiOwK9eWjyvoevnJo9OdtrQlh VSWFaF9oqVpGaxwAFf8zn4as3RcOV2pZ =a50h -----END PGP MESSAGE-----

2. Uji keamanan koneksi dilakukan saat klien menjalin komunikasi dengan server. Caranya dengan memantau trafik/paket yang lewat di jaringan. Tool yang digunakan untukmelakukan koneksi ini adalah wireshak. Wireshark merupakan software untuk melakukan analisa lalu-lintas jaringan computer. Wireshark dapat membaca data secara langsung dari Ethernet, Token-Ring, FDDI, serial (PPP and SLIP),802.11 wireless LAN , dan koneksi ATM.

Ini adalah paket yang berhasil ditangkap oleh wireshark :



Gambar 4.9. *Screenshoot Capture* Wireshark

Gambar 4.9 memperlihatkan bahwa *Handshake* yang terjadi berjalan dalam mode SSL. Hal tersebut menunjukkan 2 (dua) hal :

a. Mengindentifikasikan bahwa enkripsi yang menjamin confidentiality dan pengamanan integritas data telah digunakan [20].

b. Penggunaan Algoritma enkripsi dan kunci antara klien dan server telah sama.

Dari hasil pemantauan *packet sniffing* yang lewat di jaringan terlihat bahwa protocol secure socket layer bekerja dengan benar. Sehingga koneksi dalam *mode secure* telah berhasil dilakukan.

4.2.3. Pengujian *Web Performance*

Dalam pengujian ini akan didapatkan *TOP LEVEL METRICS* yang menggambarkan kinerja keseluruhan sistem. Metrik yang digunakan untuk pengujian meliputi :

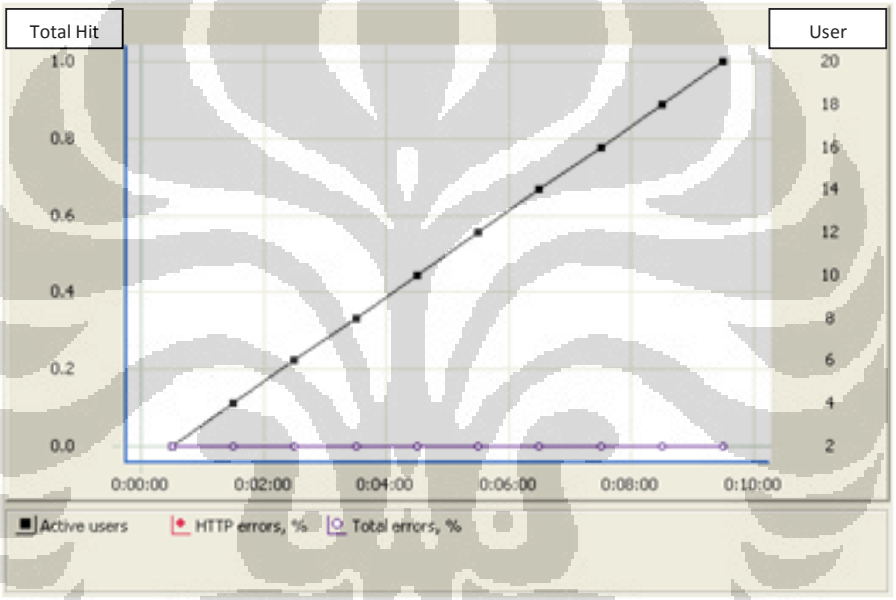
1. *Error*
2. *EverageBandwidth*
3. *Respon Time,* dilakukan pengujian per halaman.

Pengujian dilakukan dengan tool *Web Application Performance Tester (WAPT 7.1)*. Dengan skenario :

a. Diujicoba dalam sebuah jaringan LAN
b. Test Duration : 10 menit
c. Virtual User : 20 orang

Hasil pengujian menghasilkan output sebagai berikut :

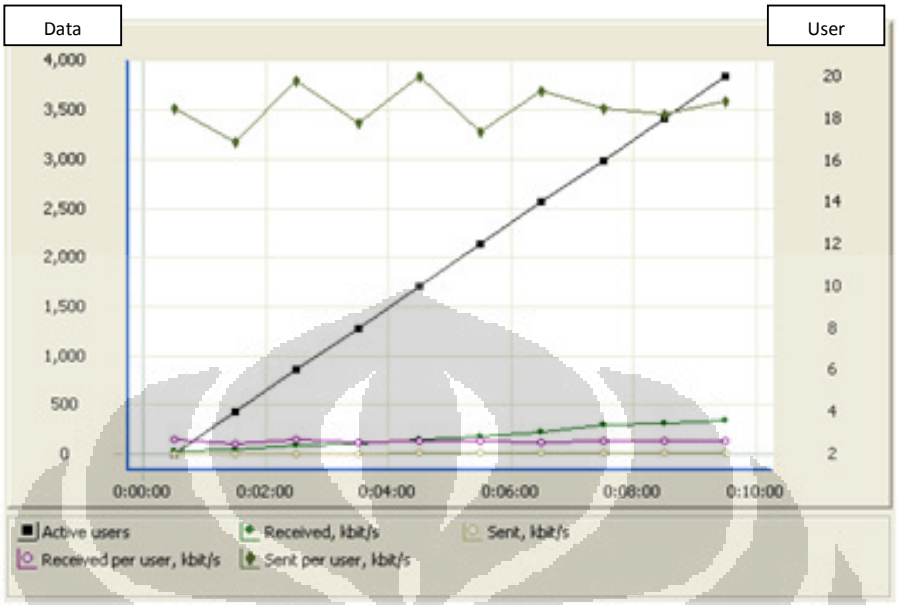
1. *Error*



Gambar 4.10. *HTTP Error*

Gambar 4.10 memperlihatkan tidak adanya error dari total hit yang ada. Artinya tidak ada masalah pada email server yang diuji, jalur koneksi lancar dan tidak ada timeout.

2. Everage Bandwidth



Gambar 4.11. *Average Bandwidth*

Gambar 4.11 memperlihatkan parameter :

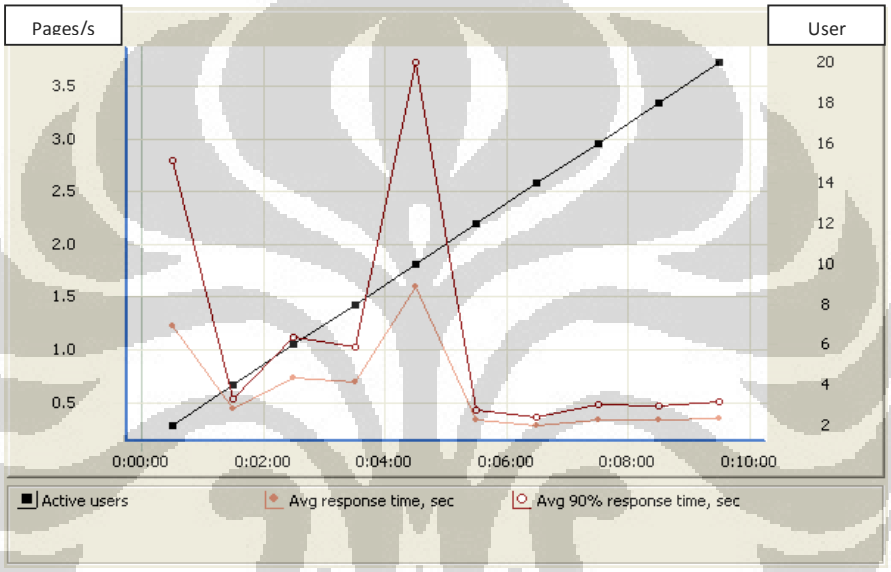
- *Sent:* besarnya kbits per second data yang dikirim ke email server.
- *Received:* besarnya kbits per second data yang diterima dari email server.
- *Sent per user:* menunjukkan kecepatan pengiriman data yang dilakukan tiap virtual user (dalam kbits per second).
- *Received per user:* menunjukkan kecepatan data yang diterima tiap virtual user (dalam kbits per second).

3. Response Time

Response time adalah waktu yang dihitung sejak page request byte pertama terkirim hingga byte terakhir diterima oleh email server.

Bisa juga diartikan waktu yang dibutuhkan sejak dilakukan klik pada sebuah tombol hingga halaman yang di klik terbuka sempurna. Pengujian respon time dilakukan per halaman dari aplikasi.

a. Response Time : https://192.168.75.115/gpgwebmail



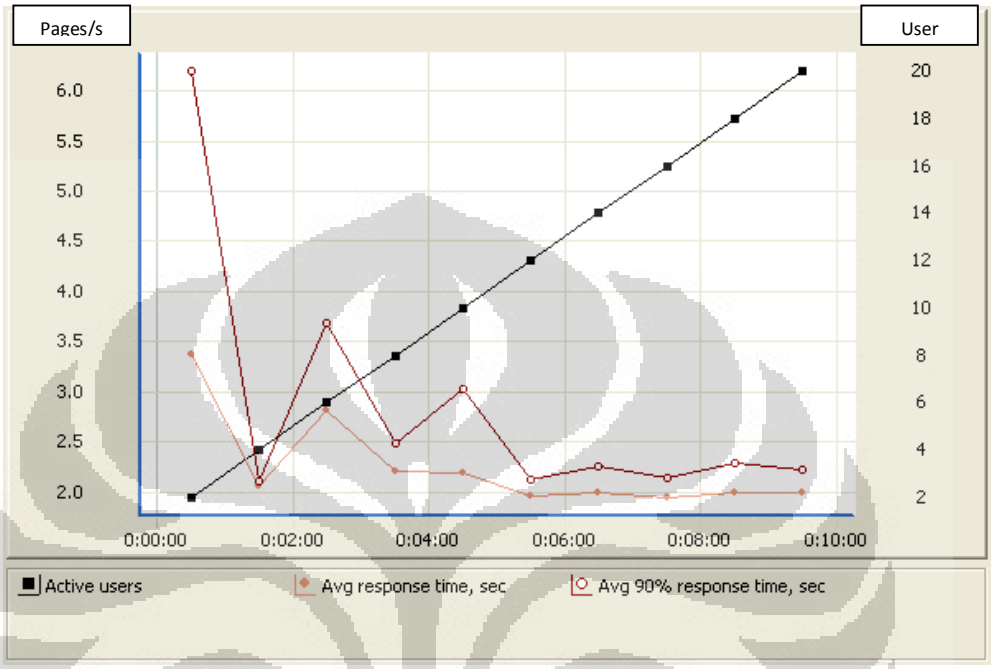
Gambar 4.12. *Response Time Halaman Home*

Gambar 4.12 memperlihatkan parameter :

- *Avg* response *time*: nilai rata - rata response time dari keseluruhan user.
- Avg 90% response *time :* 90% dari total *response time*

Saat menit ke 5 sebanyak 20 user diskenariokan melakukan akses page index.php dalam waktu bersamaan. Nilai *Avg 90% response time* masih berada di posisi kurang dari atau sama dengan total akses pages yang dilakukan oleh seluruh user. Ini berarti aplikasi berjalan baik

b. Response *Time* : https://192.168.75.115/gpgwebmail/login.php



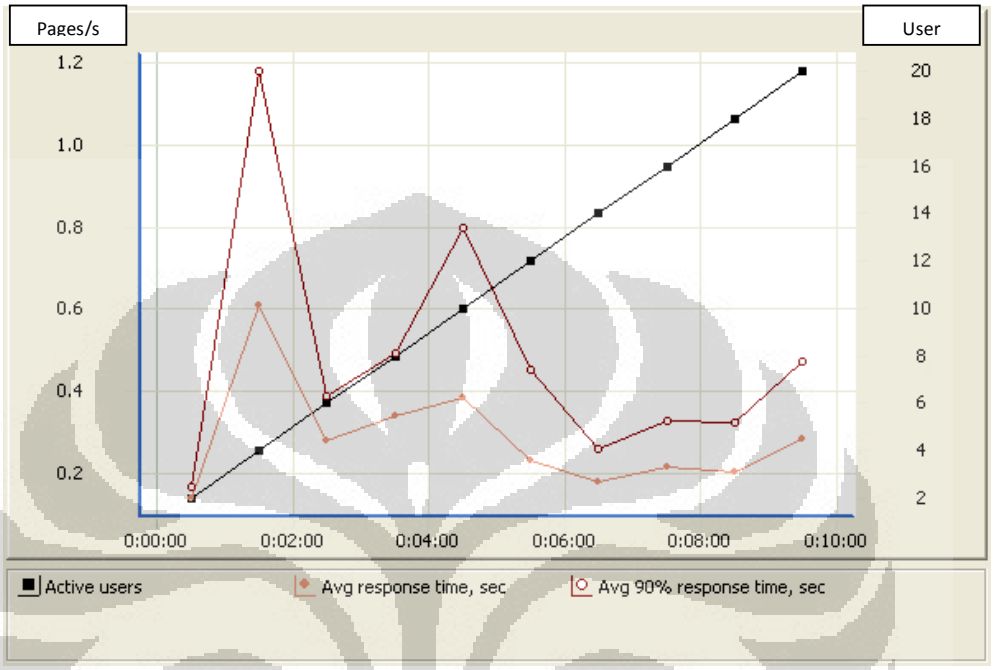
Gambar 4.13. *Response Time Halaman Login*

Gambar 4.13 memperlihatkan parameter :

- *Avg* response *time*: nilai rata - rata response time dari keseluruhan user.
- Avg 90% response *time :* 90% dari total *response time*

Saat menit ke 10 sebanyak 20 user diskenariokan melakukan akses page login.php dalam waktu bersamaan. Nilai *Avg 90% response time* masih berada di posisi kurang dari atau sama dengan total akses pages yang dilakukan oleh seluruh user. Ini berarti aplikasi berjalan baik

c. Response *Time* : https://192.168.75.115/gpgwebmail/showfolder.php



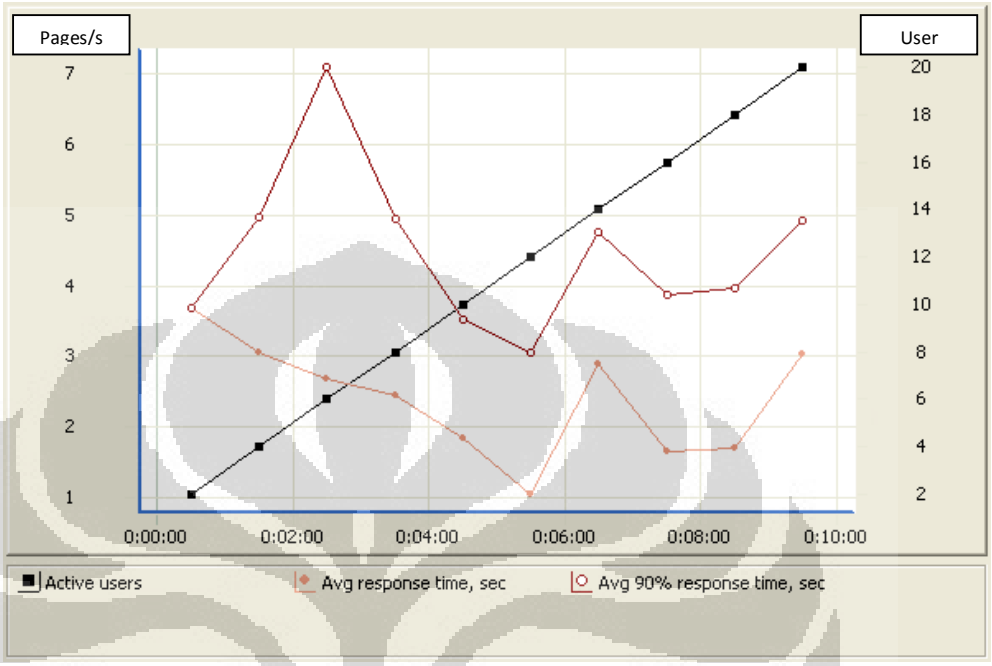
Gambar 4.14. *Response Time Halaman Tampilkan Daftar Email*

Gambar 4.14 memperlihatkan parameter :

- *Avg* response *time*: nilai rata - rata response time dari keseluruhan user.
- Avg 90% response *time :* 90% dari total *response time*

Memasuki menit ke 2 sebanyak 20 user diskenariokan melakukan akses page showfolder.php dalam waktu bersamaan. Nilai *Avg 90% response time* masih berada di posisi kurang dari atau sama dengan total akses pages yang dilakukan oleh seluruh user. Ini berarti aplikasi berjalan baik

d. *Response Time* : https://192.168.75.115/gpgwebmail/sendmail.php



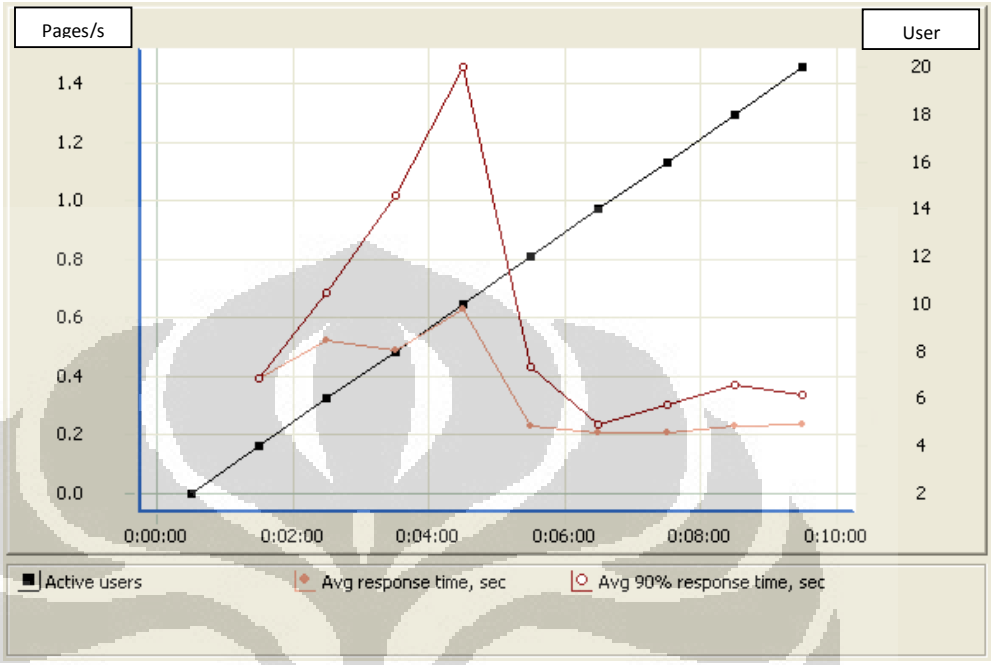
Gambar 4.15. *Response Time Halaman Buat Email Enkripsi*

Gambar 4.15 memperlihatkan parameter :

- *Avg* response *time*: nilai rata - rata response time dari keseluruhan user.
- Avg 90% response *time :* 90% dari total *response time*

Saat menit ke 3 sebanyak 20 user diskenariokan melakukan akses page sendpage.php dalam waktu bersamaan. Nilai *Avg 90% response time* masih berada di posisi kurang dari atau sama dengan total akses pages yang dilakukan oleh seluruh user. Ini berarti aplikasi berjalan baik

e. Response *Time* : https://192.168.75.115/gpgwebmail/showmsg.php



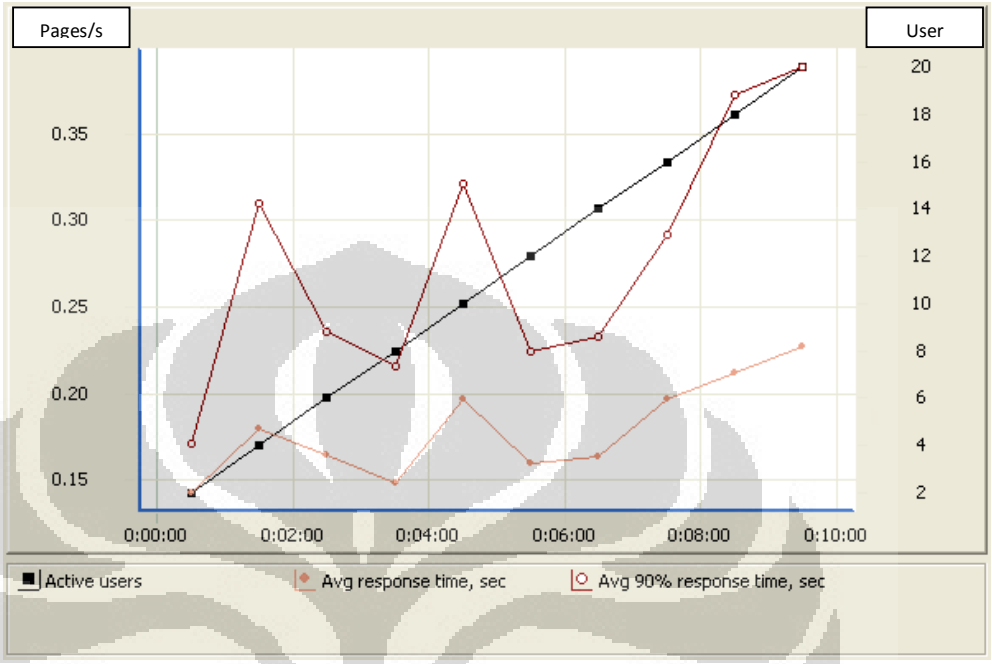
Gambar 4.16. *Response Time Halaman Baca Email Terenkripsi*

Gambar 4.16 memperlihatkan parameter :

- *Avg* response *time*: nilai rata - rata response time dari keseluruhan user.
- Avg 90% response *time :* 90% dari total *response time*

Saat menit ke 5 sebanyak 20 user diskenariokan melakukan akses page showmsg.php dalam waktu bersamaan. Nilai *Avg 90% response time* masih berada di posisi kurang dari atau sama dengan total akses pages yang dilakukan oleh seluruh user. Ini berarti aplikasi berjalan baik

f. Response *Time* : https://192.168.75.115/gpgwebmail/register.php



Gambar 4.17. *Response Time Halaman Baca Email Terenkripsi*

Gambar 4.16 memperlihatkan parameter :

- *Avg* response *time*: nilai rata - rata response time dari keseluruhan user.
- Avg 90% response *time :* 90% dari total *response time*

Saat menit ke 10 sebanyak 20 user diskenariokan melakukan akses page register.php dalam waktu bersamaan. Nilai *Avg 90% response time* masih berada di posisi kurang dari atau sama dengan total akses pages yang dilakukan oleh seluruh user. Ini berarti aplikasi berjalan baik

Hasil pengujian secara keseluruhan menghasilkan output sebagai berikut :

Tabel 4.11. Ringkasan *Web Performance*

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Total KBytes sent	Total KBytes received	Avg Response time, sec (with page elements)
uji	64	0	1257	0	1906	0	1009	13338	0.93(1.04)

4.3. ANALISA

Berdasarkan hasil pengujian diatas maka didapatkan :

a. Dengan pengujian *blackbox* menunjukkan aplikasi dapat berjalan dengan baik dan secara fungsional sistem dapat menghasilkan output yang diharapkan.
b. Berdasarkan hasil pengujian keamanan dapat menunjukkan bahwa aplikasi telah aman dari sisi data enkripsi yang dihasilkan dan koneksi yang dilakukan.
c. Dalam pengujian *web performance* didapatkan hasil yang baik. Ditunjukkan dengan nilai **avg** ***error sebesar 0%*** *dan* ***avg respon time dibawah 1 detik*** *serta* ***avg bandwidth sebesar 10,6 %***

BAB V
PENUTUP

5.1. KESIMPULAN

a. Perancangan dengan *sequence diagram* dan *use case diagram* telah berhasil diimplemntasikan dalam bentuk aplikasi siap pakai dan telah lulus pengujian.
b. Aplikasi yang dirancang telah lulus pengujian secara fungsional. Sistem dapat menghasilkan output yang diharapkan sesuai perancangan. Dibuktikan dengan pengujian *Black Box*.
c. Aplikasi berhasil dijalankan dalam protokol Secure Socket Layer.
d. Aplikasi telah berhasil menjalankan program GnuPG dan dapat diimplementasikan dalam layanan email dengan protokol POP3-SMTP.
e. Dalam pengujian *web performance* didapatkan hasil yang baik. Ditunjukkan dengan nilai **avg *error sebesar 0%*** *dan* ***avg respon time dibawah 1 detik*** *serta* ***avg bandwidth sebesar 10,6 %***

5.2. SARAN

Beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut sebagai berikut:

a. Distro dikembangkan dengan cara diintegrasikan dengan mail server sendiri. Bisa menggunakan aplikasi mail server open source.
b. Distro dikembangkan dengan cara diintegrasikan dengan distro *Security Onion* yang mempunyai sotfware SNORT yakni software *open source Network Intrusion Prevention System* (NIPS).
c. Distro dikembangkan dengan mengaktifkan field CC,BCC dan Attachment.
d. Akses ke server sebaiknya menggunakan jalur Virtual Private Network untuk menjamin tingkat keamanannya.

DAFTAR REFERENSI

[1] Royal Pingdom: Ramblings and tech news from the Pingdom team. "*Internet 2010 in Numbers,"* <http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers> (12 Januari 2011) diakses tanggal 10 Mei 2011 jam 21:45

[2] Oke Zone. "*Tuding China Serang Gmail, Usaha Google Bisa Terancam,"* http://techno.okezone.com/read/2011/06/06/55/464916/tuding-china-serang-gmail-usaha-google-bisa-terancam> (6 Juni 2011) diakses tanggal 23 Mei 2011 jam 20:21

[3] *Jelajah Kriptografi (2008).* Lembaga Sandi Negara. Hal. 87, 89

[4] GNUPG "The Gnu Privacy Guard," <http://gnupg.org/related_software/frontends.en.html> diakses tanggal 25 Juni 2011 jam 22:22

[5] Sofyan, Ahmad (2006). *Membuat Distro Linux Sendiri*. Jakarta: Dian Rakyat Hal.6

[7] Ubuntu Official Site "Ubuntu Documentation," <https://help.ubuntu.com/6.06/ubuntu/ desktopguide/ C/about-ubuntu.html> diakses tanggal 23 Mei 2011 jam 10:24

[8] " Apache Project," <http://projects.apache.org/> diakses tanggal 25 Juni 2011 jam 15:30

[9] Agus Saputra (2011). Trik dan Solusi Jitu Pemrograman PHP. Penerbit Elex Media Komputindo. Hal. 72

[10] Edy Winarno, S.T., M.Eng (2010). Mudah Membuat Website dan E-Commerce dengan PHP Framework. Penerbit Elex Media Komputindo. Hal. 4

[11] Joseph Migga Kizza (2009), PhD. *A Guide to Computer Network Security,* Springer Publishing. Hal. 19

[12] Joseph Migga Kizza (2009), PhD. *A Guide to Computer Network Security,* Springer Publishing. Hal. 20

[13] "TCP/IP," <http://www.protocols.com/pbook/tcpip1.htm> diakses tanggal 29 Juni 2011 jam 19:45

[14] Wikipedia. "Protocol Data Unit," <http://en.wikipedia.org/wiki/Protocol_data_unit> diakses tanggal 21 Juni 2011 jam 01:22

[15] Techdom. “Email Fundamentals,” <http://techdom.nl/microsoft/email-system-components-explained/> diakses tanggal 25 Mei 2011 jam 20:05

[16] CERT. “Ken Ratri (2004). Alur Data Pengiriman Email. Report : Keamanan E-Mail Dengan GPG,” <http://www.cert.or.id/~budi/courses/ec7010/dikmenjur/> diakses tanggal 03 Mei 2011 jam 11:15

[17] CERT. “Ken Ratri (2004). Pemilahan Alur Message. Report : Keamanan E-Mail Dengan GPG,” < http://www.cert.or.id/~budi/courses/ec7010/dikmenjur/> diakses tanggal 03 Mei 2011 jam 11:15

[18] William Stallings (2007). *Cryptography and Network Security, Third Edition*, Prentice Hall

[19] Greg Hoglund; Gary McGraw (2004) *Exploiting Software How to Break Code*, Addison-Wesley Professional. Hal. 7

[20] William Stallings (2007). *Cryptography and Network Security, 4th Edition*, Prentice Hall. Hal. 538

Universitas Indonesia (2004). *Pengantar penulisan imiah.*

Index.php

```
<?php
header('Location: login.php');
?>
```

Register.php

```
<?
include_once "config.inc.php";
include_once APP_INCLUDE_DIR.'/Savant3/Savant3.php';
include_once APP_INCLUDE_DIR.'/DBclass/Database.class.php';
include_once APP_INCLUDE_DIR."/tools.php";
require APP_INCLUDE_DIR.'/gnuPG_class.inc';

error_reporting(E_ALL);
// set options template path untuk savant
$options = array('template_path' => APP_TEMPLATE_DIR);
// initialize template engine
$savant = new Savant3($options);
$error_msg="";
$savant->title="GPG Webmail";
if(isset($_POST['register'])){
      $username=$_POST['username'];
      $Email=$_POST['email'];
      $db = new Database(DB_SERVER, DB_USER, DB_PASS,
DB_DATABASE);
      $db->connect();
      $sql = "SELECT * FROM user WHERE
username='$username'";
      $row = $db->query_first($sql);
      $jmlrow=$db->affected_rows;
      $sql = "SELECT * FROM user WHERE email='$Email'";
      $row = $db->query_first($sql);
      $jmlrow2=$db->affected_rows;
      if($jmlrow > 0 && $jmlrow2 > 0){
            $error_msg="Username or email address already
exist, please select another one..";
      } else {
            $RealName=$_POST['realname'];
            $Comment=$_POST['comment'];
            $Email=$_POST['email'];
            $Passphrase = $_POST['passphrase'];
            $ExpireDate = $_POST['expiredate'];
            $KeyType = $_POST['keytype'];
            $KeyLength = $_POST['keylength'];
            $SubkeyType = $_POST['subkeytype'];
            $SubkeyLength=$_POST['subkeylength'];
            // create the instance, giving the program path
and home directory
            $gpg = new gnuPG($program_gpg, $home_directory);
```

```
            $GeneratedKey = $gpg->GenKey($RealName,
$Comment, $Email, $Passphrase, $ExpireDate, $KeyType,
$KeyLength, $SubkeyType, $SubkeyLength);
            $fp="";
//          $error_msg=$GeneratedKey;
            if($GeneratedKey){
                  $fp=$GeneratedKey;
//                echo "The new key was created";
            }else{
                  $error_msg="Unable to create the new GPG
key";
//                echo "Unable to create the new key";
            }
            //save ke db
            $data['username'] = $_POST['username'];
            $data['realname'] = $_POST['realname'];
            $data['password'] = md5($_POST['password']);
            $data['email'] = $_POST['email'];
            $data['popserver'] = $_POST['popserver'];
            $data['popport'] = $_POST['popport'];
            $data['popauth'] = $_POST['popauth'];
            $data['smtpserver'] = $_POST['smtpserver'];
            $data['smtpport'] = $_POST['smtpport'];
            $data['smtpauth'] = $_POST['smtpauth'];
            $data['mailuname'] = $_POST['emailusername'];
            $data['mailpwd'] = $_POST['emailpassword'];
            $data['passphrase'] =
base64_encode($_POST['passphrase']);
            $data['fp'] = $fp;
            $data['isadmin'] = 0;
            $uid=$db->query_insert("user", $data);
            $key=$gpg->ExportSecretKey($fp);
            if($gpg->DeleteSecretKey($fp))
                  $delkey= "Secret key has been deleted.";
            else
                  $delkey= "Secret key unable to delete.";
            // insert to tabel reg
            $reg['userid'] = $uid;
            $reg['username'] = $_POST['username'];
            $reg['password'] = $_POST['password'];

      $tmp=$reg['userid'].$reg['username'].$reg['password'];
            $reg['token'] = md5($tmp);
            $reg['key'] = $key;
            $db->query_insert("reg", $reg);
            //send mail
            $message="Your account has been successfuly
created.<br>You must activated your account. Please
activated your account <a
href='".APP_ROOT_URL."reg.php?token=".$reg['token']."'>here
</a><p>Regards,<br>GPG Webmail</p>";
            $hasil=tools::sendmailhtml(ADMIN_EMAIL,
$data['email'], "Your registration",
```

```
$message,ADMIN_SMTP,ADMIN_SMTP_PORT,ADMIN_EMAIL,ADMIN_EMAIL
_PWD,ADMIN_SMTP_AUTH);
            if($hasil){
                  $result="Your detail registration has been
to your email. Please read it and activated your account.";
            }else {
                  $result="There was an error occur during
sending to your email.";
            }

      }
      if (strlen($error_msg)>0){
            $savant->error_msg=$error_msg;
            $savant->header = $savant->fetch('header.html');
            $savant->footer = $savant->fetch('footer.html');
            $savant->display('register.html');
      }else{
            $pesan="<p><h3>Registration
completed.</h3><br>Your account has been successfully
created.</p>";
//          $pesan.="<p><b>Please save this secret key,
because we don't save any secret
key..</b><br><pre>$key</pre></p>";
            $pesan.="<p>$result</p>";
            $savant->result=$pesan;
            $savant->header = $savant->fetch('header.html');
            $savant->footer = $savant->fetch('footer.html');
            $savant->display('register-result.html');

      }

} else {
      $savant->error_msg=$error_msg;
      $savant->header = $savant->fetch('header.html');
      $savant->footer = $savant->fetch('footer.html');
      $savant->display('register.html');
}
?>
```

Login.php

```
<?php
session_start();

include_once "config.inc.php";
include_once APP_INCLUDE_DIR.'/Savant3/Savant3.php';
include_once APP_INCLUDE_DIR.'/DBclass/Database.class.php';
include_once APP_INCLUDE_DIR."/tools.php";
require APP_INCLUDE_DIR.'/gnuPG_class.inc';

error_reporting(E_ALL);
// set options template path untuk savant
$options = array('template_path' => APP_TEMPLATE_DIR);
```

```
// initialize template engine
$savant = new Savant3($options);
$error_msg="";
$savant->title="GPG Webmail";
if(isset($_POST['submit'])){
     $username = isset($_POST['username']) ?
$_POST['username'] : '';
     $password = isset($_POST['password']) ?
$_POST['password'] : '';
     $keyblock = $_POST['keyblock'];
     $db = new Database(DB_SERVER, DB_USER, DB_PASS,
DB_DATABASE);
     $db->connect();
     $sql = "SELECT * FROM reg WHERE username='$username'";
     $row1 = $db->query_first($sql);
     $jmlrow1=$db->affected_rows;
     if($jmlrow1 > 0){
          $error_msg="Your account is not active. Please
check your email and activated fisrt.";
     } else {
          $sql = "SELECT * FROM user WHERE
username='$username'";
          $row = $db->query_first($sql);
          $jmlrow=$db->affected_rows;
          if($jmlrow > 0){
               //echo "Success!";
               if (md5($password) === $row['password']){
     //               echo "sukses";
     //import secret key
                    $gpg = new gnuPG($program_gpg,
$home_directory);
                    $hsl = $gpg->Import($keyblock);
                    $_SESSION['login']=true;
                    $_SESSION['userid']      =
$row['userid'];
                    $_SESSION['username']      =
$row['username'];
                    $_SESSION['realname']      =
$row['realname'];
                    $_SESSION['email']      =
$row['email'];
                    $_SESSION['pop_server']      =
$row['popserver'];
                    $_SESSION['pop_server_port'] =
intval($row['popport']);
                    $_SESSION['pop_server_auth'] =
$row['popauth'];
                    $_SESSION['smtp_server']      =
$row['smtpserver'];
                    $_SESSION['smtp_server_port'] =
intval($row['smtpport']);
                    $_SESSION['smtp_server_auth'] =
$row['smtpauth'];
```

```
                        $_SESSION['mail_username'] =
$row['mailuname'];
                        $_SESSION['mail_password'] =
$row['mailpwd'];
                        $_SESSION['fp'] = $row['fp'];
                        $_SESSION['passphrase'] =
$row['passphrase'];
                        if($row['isadmin']==1){
                        $_SESSION['isadmin'] = true;
                        }else{
                        $_SESSION['isadmin'] = false;
                        }

      //                echo $_SESSION['mail_server_port'];

      tools::getmail2db($db,$_SESSION['userid'],$_SESSION['p
op_server'], $_SESSION['pop_server_port'],
$_SESSION['mail_username'],$_SESSION['mail_password'],
$_SESSION['pop_server_auth']);
                        $sql="select * from email where
userid={$_SESSION['userid']} and folder='inbox' order by
date desc";
      //                $sql="select * from email order by
date desc";
                        $rows = $db->fetch_all_array($sql);
                        $savant->header = $savant-
>fetch('header.html');
                        $savant->footer = $savant-
>fetch('footer.html');

                        $savant->error_msg="";
                        $savant-
>folders=tools::getcountmail($db,$_SESSION['userid']).tools
::showlinkgpg();
                        if($_SESSION['isadmin']) $savant-
>folders .=tools::showlinkadmin();
                        $savant->foldername="<b>INBOX</b>";
                        $savant->listmail=$rows;
                        $savant->content = $savant-
>fetch('email-list.html');
                        $savant->display('main-email.html');

                }else{
                        $error_msg="Error: Invalid username
and/or password.";
                }
           }

           else{
                $error_msg="Error: Invalid username and/or
password.";
           }
      }
```

```
      $db->close();
      //    $savant->tpl_logged_in=false;
      if($error_msg!=""){
            $savant->error_msg=$error_msg;
            $savant->header = $savant->fetch('login-
header.html');
            $savant->footer = $savant->fetch('footer.html');
            $savant->display('login.html');
      }
} else {
//    $savant->tpl_logged_in=false;
      $savant->error_msg=$error_msg;
      $savant->header = $savant->fetch('login-header.html');
      $savant->footer = $savant->fetch('footer.html');
      $savant->display('login.html');
}

?>
```

Config.php

```
<?php
/**
 * Deskripsi: Berisi inisialisasi contant
 */
$program_gpg='';
$home_directory='';

//database server
define('DB_SERVER', "localhost");
//database login name
define('DB_USER', "root");
//database login password
define('DB_PASS', "");
//database name
define('DB_DATABASE', "gpgwebmail");

//inisialisasi struktur direktory dirname(__FILE__)
define("APP_ROOT_DIR", dirname(__FILE__));
define("APP_INCLUDE_DIR", APP_ROOT_DIR."/include");
define("APP_IMAGES_DIR", APP_ROOT_DIR."/images");
define("APP_CSS_DIR", APP_ROOT_DIR."/css");
//define("APP_UPLOAD_DIR", APP_ROOT_DIR."/upload");
//define("APP_TMP_DIR", APP_ROOT_DIR."/temp");
define("APP_TEMPLATE_DIR", APP_ROOT_DIR."/templates");
//define("APP_PDF_DIR", APP_ROOT_DIR."/pdf");
define("APP_ROOT_URL","http://localhost/gpgwebmail/");

define("ADMIN_EMAIL","");
define("ADMIN_EMAIL_PWD","");
define("ADMIN_SMTP","smtp.gmail.com");
define("ADMIN_POP","pop.gmail.com");
define("ADMIN_SMTP_PORT",465);
```

```
define("ADMIN_POP_PORT",995);
//jika authentikasi tidak menggunakan ssl diisi none
define("ADMIN_SMTP_AUTH","ssl");
define("ADMIN_POP_AUTH","ssl");
?>

showmsg.php
<?
session_start();

include_once "config.inc.php";
include_once APP_INCLUDE_DIR.'/Savant3/Savant3.php';
include_once APP_INCLUDE_DIR.'/DBclass/Database.class.php';
include_once APP_INCLUDE_DIR.'/gnuPG_class.inc';
include_once APP_INCLUDE_DIR."/tools.php";

error_reporting(E_ALL);
// set options template path untuk savant
$options = array('template_path' => APP_TEMPLATE_DIR);
// initialize template engine
$savant = new Savant3($options);
$error_msg="";
$savant->title="GPG Webmail";
if(isset($_REQUEST['id'])){
     $db = new Database(DB_SERVER, DB_USER, DB_PASS,
DB_DATABASE);
     $db->connect();

     $sql="select * from email where id={$_REQUEST['id']}";
     $row = $db->query_first($sql);

          $savant->header = $savant->fetch('header.html');
          $savant->footer = $savant->fetch('footer.html');
          $savant-
>folders=tools::getcountmail($db,$_SESSION['userid']).tools
::showlinkgpg();
     if($_SESSION['isadmin']) $savant->folders
.=tools::showlinkadmin();
          $savant->error_msg=$error_msg;
          $savant->mail=$row;
     if($row['fencrypt']==1){
          $gpg = new gnuPG($program_gpg, $home_directory);
          $keys=$gpg-
>ListKeys2($_SESSION['mail_username'],'secret');
//    foreach($Keys as $Key){
//print_r($keys);
//echo
"<br>".$keys[0]['KeyID']."<br>".$_SESSION['fp']."<br>".base
64_decode($_SESSION['passphrase'])."<br><pre>".$row['body']
."</pre>";
          $Text = $gpg->Decrypt($_SESSION['fp'],
base64_decode($_SESSION['passphrase']),
trim($row['body']));
```

```
//          if($Text==false) echo "<br> dekript error";
//          echo "<br>body:".$Text;
            $savant->body=$Text;

      }else{
            $savant->body=$row['body'];
      }
            $savant->content = $savant->fetch('mail-
read.html');
            $savant->display('main-email.html');
      $data['fread'] = 1;
      $db->query_update("email", $data,
"id={$_REQUEST['id']}");
      $db->close();
}
?>
```

Sendmail.php

```
<?php
session_start();

include_once "config.inc.php";
include_once APP_INCLUDE_DIR.'/Savant3/Savant3.php';
include_once APP_INCLUDE_DIR.'/DBclass/Database.class.php';
include_once APP_INCLUDE_DIR."/tools.php";
include_once APP_INCLUDE_DIR.'/gnuPG_class.inc';

error_reporting(E_ALL);
// set options template path untuk savant
$options = array('template_path' => APP_TEMPLATE_DIR);
// initialize template engine
$savant = new Savant3($options);
$error_msg="";
$savant->title="GPG Webmail";
//print_r($_POST);
$db = new Database(DB_SERVER, DB_USER, DB_PASS,
DB_DATABASE);
$db->connect();
if(isset($_POST['send'])){
      $mailto=trim($_POST['mailto']);
      $subject=trim($_POST['subject']);
      $message=trim($_POST['message']);
//echo "<pre>$message</pre>";

      $gpg = new gnuPG($program_gpg, $home_directory);
      $keys=$gpg->ListKeys2($mailto);
      if($keys)
            $message = $gpg->Encrypt($_SESSION['fp'],
base64_decode($_SESSION['passphrase']),$keys[0]['Fingerprin
t'], $message);
//echo "<pre>$message</pre>";
      $hasil=tools::sendmail($_SESSION['email'], $mailto,
$subject,
```

```
$message,$_SESSION['smtp_server'],$_SESSION['smtp_server_po
rt'],$_SESSION['mail_username'],$_SESSION['mail_password'],
$_SESSION['smtp_server_auth']);
      if($hasil){
            $result="Your message has been successfully
send.";
      }else {
            $result="Your message fail to send. There was an
error occur during sending your message.";
      }
      $savant->header = $savant->fetch('header.html');
      $savant->footer = $savant->fetch('footer.html');
//    $savant->error_msg=$error_msg;
      $savant->result=$result;
      $savant-
>folders=tools::getcountmail($db,$_SESSION['userid']).tools
::showlinkgpg();
      if($_SESSION['isadmin']) $savant->folders
.=tools::showlinkadmin();
      $savant->content = $savant->fetch('mail-send-
result.html');
      $savant->display('main-email.html');

}
if(isset($_POST['draf'])){

}
$db->close();
?>
```

Asklistmail.php

```
<?
session_start();

include_once "config.inc.php";
include_once APP_INCLUDE_DIR.'/Savant3/Savant3.php';
include_once APP_INCLUDE_DIR.'/DBclass/Database.class.php';
include_once APP_INCLUDE_DIR."/tools.php";

error_reporting(E_ALL);
// set options template path untuk savant
$options = array('template_path' => APP_TEMPLATE_DIR);
// initialize template engine
$savant = new Savant3($options);
$error_msg="";
$savant->title="GPG Webmail";
      $db = new Database(DB_SERVER, DB_USER, DB_PASS,
DB_DATABASE);
      $db->connect();
//print_r($_POST);
if(isset($_POST['submit'])){
      if(isset($_POST['pm_ids'])){
```

```
            if($_POST['actions']=="delete" ||
$_POST['actions1']=="delete"){
                  foreach($_POST['pm_ids'] as $id){
                        $sql = "DELETE FROM email WHERE
id=$id";
                        $db->query($sql);
                  }
            }
            if($_POST['actions']=="markread" ||
$_POST['actions1']=="markread"){
                  foreach($_POST['pm_ids'] as $id){
                        $data['fread'] = 1;
                        $db->query_update("email", $data,
"id=$id");
                  }
            }
            if($_POST['actions']=="markunread" ||
$_POST['actions1']=="markunread"){
                  foreach($_POST['pm_ids'] as $id){
                        $data['fread'] = 0;
                        $db->query_update("email", $data,
"id=$id");
                  }
            }
      }
      $sql="select * from email where
userid={$_SESSION['userid']} and
folder='{$_POST['folder']}' order by date desc";
      $rows = $db->fetch_all_array($sql);
      $savant->header = $savant->fetch('header.html');
      $savant->footer = $savant->fetch('footer.html');

      $savant->error_msg=$error_msg;
      $savant-
>folders=tools::getcountmail($db,$_SESSION['userid']).tools
::showlinkgpg();
      if($_SESSION['isadmin']) $savant->folders
.=tools::showlinkadmin();
      $savant->foldername=$_POST['folder'];
      $savant->listmail=$rows;
      $savant->content = $savant->fetch('email-list.html');
      $savant->display('main-email.html');
}
$db->close();
?>
```

Importkey.php

```
<?
session_start();

include_once "config.inc.php";
include_once APP_INCLUDE_DIR.'/Savant3/Savant3.php';
include_once APP_INCLUDE_DIR.'/DBclass/Database.class.php';
```

```
include_once APP_INCLUDE_DIR.'/gnuPG_class.inc';
include_once APP_INCLUDE_DIR."/tools.php";

error_reporting(E_ALL);
// set options template path untuk savant
$options = array('template_path' => APP_TEMPLATE_DIR);
// initialize template engine
$savant = new Savant3($options);
$error_msg="";
$savant->title="GPG Webmail";
     $db = new Database(DB_SERVER, DB_USER, DB_PASS,
DB_DATABASE);
     $db->connect();
if(isset($_POST['import'])){
//echo "masuk";
     $KeyBlock=$_POST['keyblock'];
//    echo $KeyBlock;
     // create the instance, giving the program path and
home directory
     $gpg = new gnuPG($program_gpg, $home_directory);
     $hasil=$gpg->Import($KeyBlock);
//    print_r($hasil);
     if($hasil===false){
          $error_msg=$gpg->error . "\n";
     }else{
          // show each of the imported keys
          foreach($hasil as $KeyID) {
               $error_msg= "The key {$KeyID['KeyID']},
{$KeyID['UserID']} was sucessfuly imported!!!\n";
//             echo "loop";
          }
          if(strlen($error_msg)==0) $error_msg="Import
successfuly";
     }
     $savant->header = $savant->fetch('header.html');
     $savant->footer = $savant->fetch('footer.html');

     $savant->error_msg=$error_msg;
     $savant-
>folders=tools::getcountmail($db,$_SESSION['userid']).tools
::showlinkgpg();
     if($_SESSION['isadmin']) $savant->folders
.=tools::showlinkadmin();
     $savant->msg="<b>Import Key</b>";
     $savant->content = $savant->fetch('importkey.html');
     $savant->display('main-email.html');

}else{

     $savant->header = $savant->fetch('header.html');
     $savant->footer = $savant->fetch('footer.html');

     $savant->error_msg=$error_msg;
```

```
      $savant-
>folders=tools::getcountmail($db,$_SESSION['userid']).tools
::showlinkgpg();
      if($_SESSION['isadmin']) $savant->folders
.=tools::showlinkadmin();
      $savant->msg="<b>Import Key</b>";
      $savant->content = $savant->fetch('importkey.html');
      $savant->display('main-email.html');
}
$db->close();
?>
```