# PENERAPAN KEAMANAN E-MAIL DENGAN MENGGUNAKAN GNU *PRIVACY GUARD* (GNUPG)

Hari Purwanto

Manajemen Informatika,
Fakultas Teknologi Industri, Universitas Suryadarma

***Abstrak :*** *Internet merupakan salah satu sarana komunikasi yang sedang berkembang sekarang. Fasilitas yang disediakan sangat beragam dan menjanjikan kecepatan pengiriman data ataupun informasi. Salah satu layanan aplikasi di Internet yang paling banyak digunakan adalah Electronic Mail (e-mail).*

*Keamanan sistem informasi berbasis Internet menjadi salah satu keharusan untuk diperhatikan. Hal tersebut dikarenakan jaringan Internet yang sifatnya publik dan global pada dasarnya tidak aman. Pada saat data terkirim dari suatu komputer ke komputer lain di dalam Internet, data itu akan melewati sejumlah komputer yang lain yang berarti akan memberikan kesempatan pada user lain untuk menyadap atau mengubah data tersebut.*

*Usaha - usaha penyadapan proses penyampaian e-mail melalui Internet semakin hari semakin meluas. Terlebih setelah masuknya transaksi dunia bisnis ke dunia Internet yang semakin memerlukan tingkat kerahasiaan tertentu Jika kita merasa data yang ada di sistem adalah hal penting dan memerlukan pengamanan, ada baiknya GnuPG (Gnu Privacy Guard) digunakan untuk memperkuat sistem pengamanan data yang ada. Program ini dapat digunakan untuk menyandikan pesan e-mail, data atau dokumen rahasia. Program ini dapat juga digunakan untuk mengirimkan data tersandi via jaringan secara aman.*

***Kata kunci :*** *e-mail, administrator, GNU Privacy Guard, enkripsi, kriptografi,*

## II. PENDAHULUAN

E-mail sudah digunakan orang sejak awal terbentuknya internet pada sekitar tahun 1969 dan merupakan salah satu fasilitas yang ada pada saat itu. Sesuai dengan perkembangan internet, penggunaan email ini juga semakin membesar. Salah satu lterna kenapa email dipakai orang karena memberikan cara yang mudah dan cepat dalam mengirimkan sebuah informasi. Selain itu dengan email dapat juga informasi yang ukurannya kecil sampai ke file yang ukurannya besar.

Namun sifat e-mail yang memanfaatkan penghantar elektronik tak sepenuhnya dimaksudkan sebagai medium pribadi karena menyimpan potensi bahaya penyalah gunaan yang bukan saja menjengkelkan tetapi juga dapat bersifat fatal.

Ketika kita mengirimkan suatu e-mail, maka e-mail tersebut disampaikan ke suatu sistem komputer yang mungkin kita tidak mengetahui administratornya. Dari komputer tersebut disampaikan ke sistem komputer lain, dan yang lainnya, dan lainnya, sampai kepada penerima yang dituju. Pada beberapa link di rantai ini, e-mail kita dapat dibaca oleh siapa saja yang diinginkan *system administrator* , atau oleh suatu biro penyelidikan yang sedang mencurigai suatu aktivitas kejahatan, atau berbagai kemungkinan lainnya. Tetapi secara ringkasnya adalah ketika kita mengirimkan suatu e-mail, kita tidak mengetahui siapa yang membaca pesan itu, penerima yang diharapkan ataupun barangkali orang lain.

Kerahasiaan email terancam bukan oleh para hacker, melainkan para *system administrator* sendiri. Para *system administrator* terkadang bosan tidak tahu apa yang harus dikerjakan selain membaca-baca email orang. Mereka dapat melakukannya tanpa sedikit pun meninggalkan jejak. Cara mengatasi hal ini adalah dengan mengenkripsi email anda. GNU Privacy Guard (GnuPG, atau GPG) adalah sistem enkripsi key public. Program ini di lterna pada semua mesin DICE Linux. GnuPG adalah suatu reimplementasi GNU dari program PGP (Pretty Good Privacy) Phil Zimmerman's,

memenuhi spesifikasi OpenPGP, Zimmerman yang pertama menulis dan mendistribusikan PGP.

Tulisan ini disusun dengan maksud:

1. Memberikan informasi kepada para pembaca mengenai teknologi enkripsi E-mail dalam usaha untuk menghindari pembacaan E-mail oleh yang tidak berhak.
2. Memberikan wacana tentang sekuriti e-mail *open source* dengan GNU Privacy Guard (GnuPG, atau GPG); suatu standard bagian dari Red Hat, Mandrake, Debian, Slackware, Suse, Freebsd dan lainnya.

Kerahasiaan email terancam bukan oleh para hacker, melainkan para *system administrator*-nya sendiri yang iseng dengan membaca email orang lain. Mereka dapat melakukannya tanpa sedikit pun meninggalkan jejak. Cara mengatasi hal ini maka untuk mendapatkan jaminan kerahasiaan di dalam pengiriman email maka perlu menggunakan program aplikasi yang menjanjikan keamanan yang lebih baik. Software open source dapat dijadikan lternative karena paling tidak memungkinkan para ekspert di luar perusahaan penyedia sistem tersebut untuk memeriksa secara lebih seksama dan menyeluruh.

Untuk memberikan penekanan khusus sesuai dengan judul penulisan ini, maka dilakukan pembatasan pembahasan, yaitu :

1. Pembahasan tentang tata cara pengiriman e-mail
2. Pembahasan tentang sistem Public Key Criptografi
3. Sistem sekuriti email dengan menggunakan Gnu Privacy Guard, cara instalasi, contoh penggunaan *command line*.

## II. SURAT ELEKTRONIK DAN SEKURITI

### 2.1 Definisi

Layanan paling populer di Internet adalah *Electronic Mail* atau orang sering menyingkatnya menjadi e-mail. Jika kita mempunyai program *client e-mail* misalnya *Microsoft Outlook* dan memiliki akses kelayanan *e-mail*, maka dapat mengirim e-mail ke setiap orang yang alamat e-mailnya kita ketahui.

Alamat e-mail merupakan gabungan dari nama ***user*** dan ***domain name*** **; user@ domainname**. Misalnya:
susan@kampoeng.com

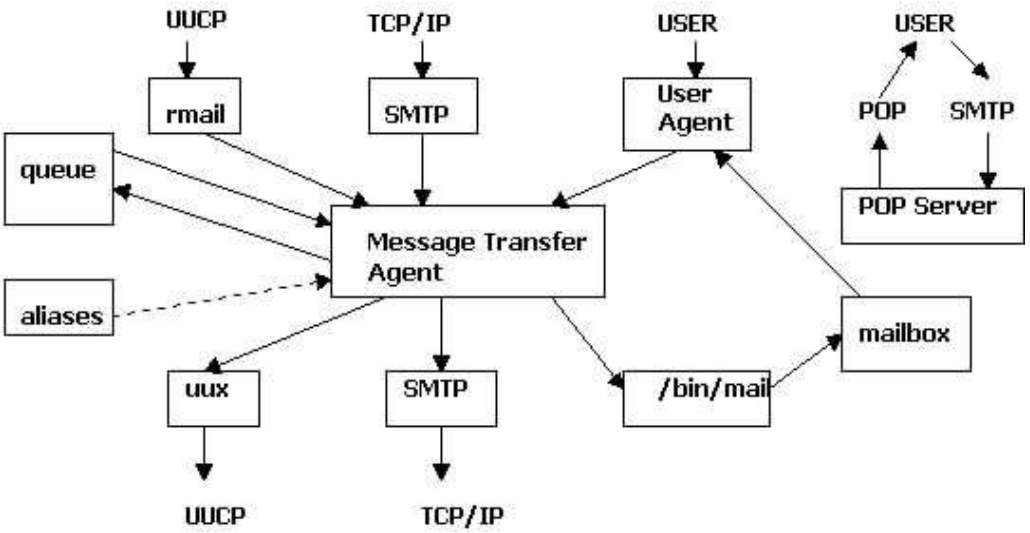
Untuk mengirim e-mail kepada seseorang maka harus membuat pesan terlebih dahulu. Pesan e-mail biasanya terdiri atas teks, namun dapat juga berisi file biner-seperti gambar grafis dan program. Pesan tersebut meliputi : nama dan alamat yang dituju, teks isi pesan. Pesan itu akan disampaikan melalui satu host ke host yang lain hingga mencapai tujuan, untuk lebih jelasnya diuraikan di bagian selanjutnya.

Tidak mudah untuk menyadap sebuah pesan e-mail, tetapi itu hal yang mungkin, untuk alasan inilah beberapa orang mengenkripsi pesan mereka agar tidak seorangpun kecuali penerima yang bisa membacanya.

### 2.2 Sistem Pengiriman Email Pada Linux

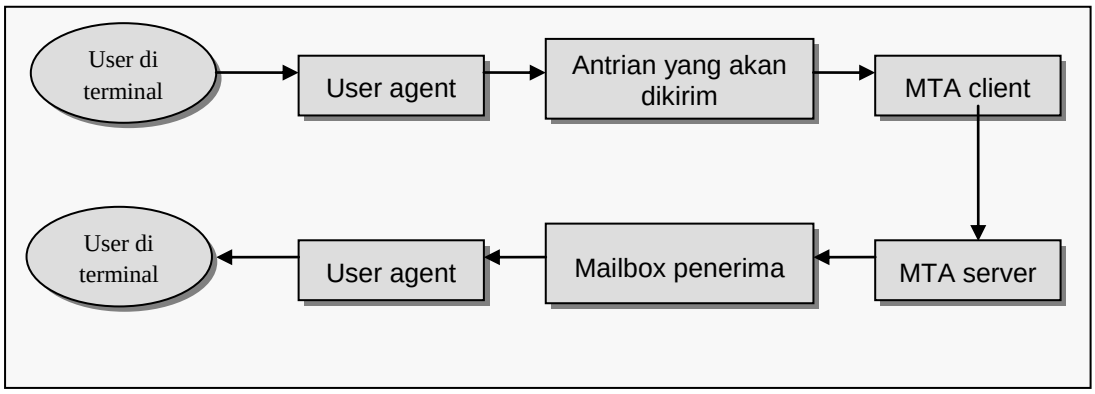
Di bawah ini dijelaskan secara singkat sistem e-mail yang umum pada sistem Unix/Linux. Pada sistem Linux proses pengiriman dan penerimaan mail melibatkan beberapa proses. Pengiriman sendiri akan memanfaatkan protokol seperti SMTP atau UUCP; protokol dalam berkomunikasi email, misalnya SMTP (*Simple Mail Transport Protocol*) yang bekerja di port 25. Protokol ini hanya bekerja untuk berkomunikasi dengan server mail remote, tidak untuk server lokal.

Sedangkan pengguna dapat membaca e-mailnya menggunakan protokol POP. Setiap pengguna memiliki 'mailbox' pada mail server. Di tempat inilah mail yang ditujukan kepada pengguna tersebut disimpan. Di sini komponen mail server bisa dipasang sesuai kebutuhan.




*Arsitektur sistem mail*

Pada gambar di bawah ini, bagaimana cara pertukaran email yang menggunakan TCP/IP




**Gambar komponen konseptual sistem email**

Mail server hanya sebuah aplikasi yang berhubungan dengan lalu lintas email, tidak secara langsung berhubungan dengan user yang akan berkirim email. Dalam pengiriman email, terdapat dua aplikasi yang diperlukan yaitu MTA (*Mail Transfer Agent*), dan MUA (*Mail 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 :

1. Pertukaran email menggunakan protokol TCP
2. Menerima email masuk (incoming)
3. Meneruskan email yang akan keluar (outgoing)
4. Mengatur antrian bila ada email masuk, keluar dan yang tertunda pengirimannya

MTA yang umum dipakai adalah sendmail dan qmail untuk Unix serta untuk di Ms Windows menggunakan Mdaemon.

Sedangkan MUA adalah aplikasi yang berfungsi sebagai interface antara email, dalam hal ini berhubungan dengan user

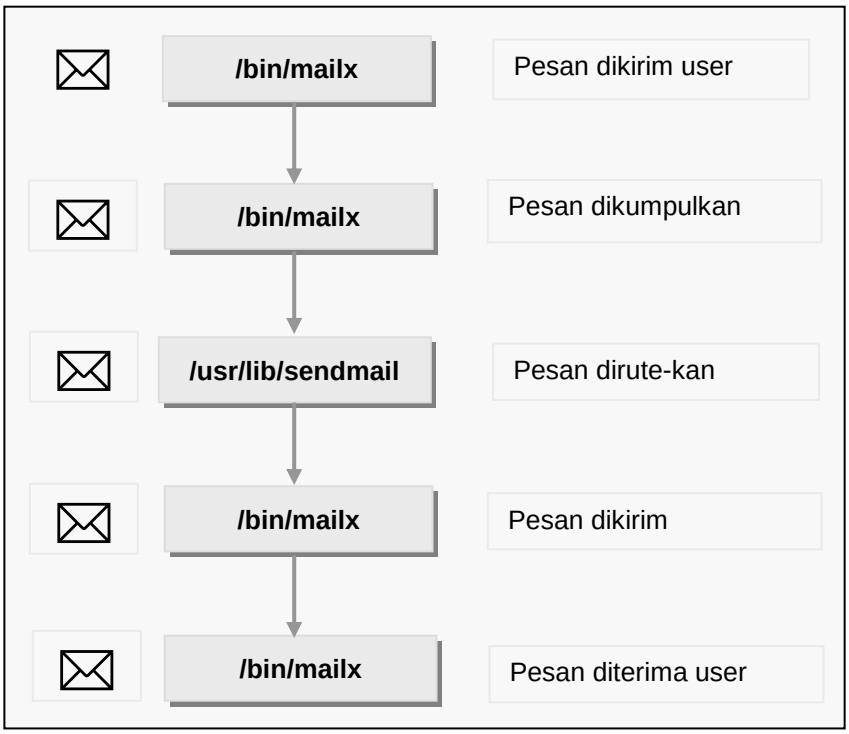
yang memiliki email tersebut, dengan MTA yang mendukungnya. Berfungsi sebagai berikut :

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

MTA akan menerima pesan yang berasal dari user di luar mesin melalui UUCP (via rmail), 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.

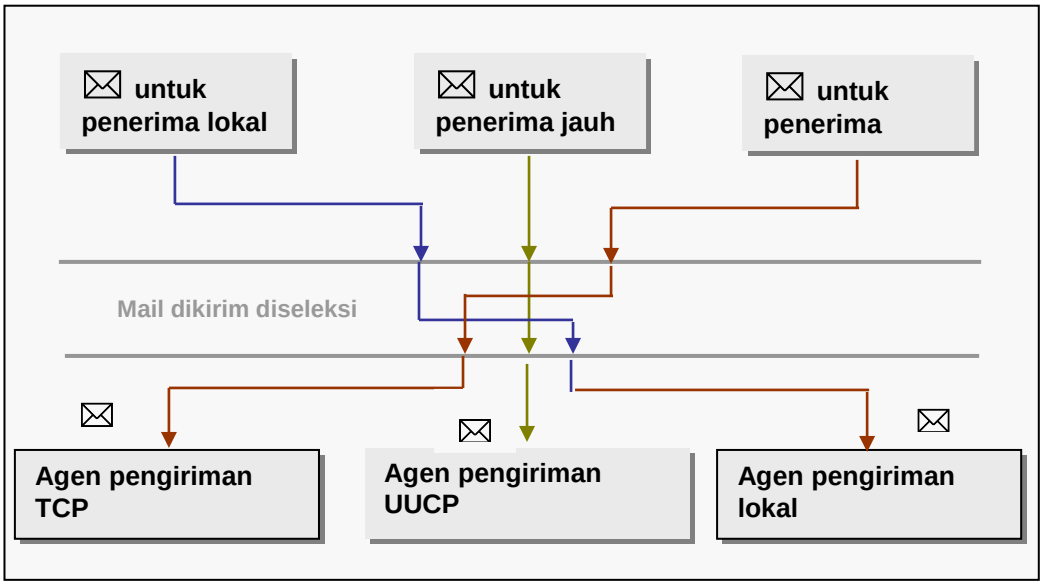
- Apakah dikirimkan lagi melalui TCP/IP atau UUCP (misal pesan dari user lokal yang ditujukan kepada user di luar mesin tersebut), atau
- Langsung dikirimkan ke mailbox user lokal (misal pesan dari user lokal untuk user lokal lainnya).

Pada sistem Linux pengguna memiliki kebebasan untuk mengganti komponen tersebut sesuai kebutuhan. Pemisahan komponen sistem email ini mengakibatkan sistem mail 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.




Pada sendmail (yang berfungsi sebagai pesan tranfer agent - MTA) terjadi proses pemilahan alur pesan, mail yang ditujukan untuk user di luar mesin tersebut akan dikirimkan melalui TCP atau UUCP. Ini bergantung jarak pada sendmail. Sedang mail yang ditujukan kepada user local akan diberikan pada mail delivery agent untuk diproses dan dimasukkan ke mailbox dari user lokal

tersebut. Proses pengolahan tambahan dapat dilakukan sebelum mail tersebut dimasukkan ke mailbox 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.




***Pemilahan alur message***

**2.3 Bagaimana E-mail Terkirim?**

Disini akan dicontohkan bagaimana proses pengiriman email. Kita akan mengirimkan sebuah alamat dan nanti akan kita lihat hubungan SMTPnya. Pada contoh dibawah ini kita akan menganalisa apa yang dikirimkan dan diterima sebuah MTA melalui SMTP.

Untuk mengirimkan sebuah email, hanya ada lima perintah yang digunakan, yaitu: HELO, MAIL,RCPT,DATA, dan QUIT. SMTP ini sangat sederhana prinsip kerjanya. Komunikasi antara server dan client terdiri dari teks-teks yang mudah dibaca. Mula-mula client menggunakan hubungan TCP ke port 25, dan menunggu kode jawaban 220.

dari server yang merupakan ucapan selamat datang ke server tersebut. Jawaban dari server ini harus dimulai dengan FQDN (*fully quanlified domain name* ) dari server, misal ai3.itb.ac.id. Selanjutnya clien memperkenalkan diri dengan perintah EHLO atau jika server masih versi lama maka cara memperkenalkan diri dengan perintah HELO. Perintah HELO adalah perintah primitif yang ada pada SMTP versi awal. Argumen dibelakang perintah tersebut adalah FQDN dari client, misalkan students.ee.itb.ac.id.

Server merespon dengan memberikan identitas dirinya kepada client. Jika komunikasi sudah terbentuk, client dapat mengirimkan lebih dari satu pesan, mengakhiri hubungan, atau meminta server untuk mengirimkan aturan bagi pengirim dan penerima, sehingga pesan dapat mengalir dengan arah yang sebaliknya.

Transaksi email dimulai dengan perintah MAIL, yang menjelaskan siapa pengirim pesan ini. Server selanjutnya mempersiapkan struktur datanya agar dapat menerima pesan baru, dan membalas perintah MAIL dengan kode 250, atau lengkapnya 250 ok. Perintah selanjutnya adalah RCPT dimana perintah ini menjelaskan siapa pemerimanya. Jika penerimanya ada banyak, maka akan ada beberapa perintah RCPT dapat dikeluarkan. Jika sudah server juga

harus membalas ke client bagi setiap perintah RCPT dengan mengirimkan respon 250 OK, atau jika ada kesalahan akan dibalas dengan respon 550 *No such user here*.

Isi pesan dikirim oleh client dengan perintah DATA yang diakhiri dengan mengirimkan satu baris data yang hanya berisi satu titik. Server merespon dengan mengirimkan pesan 354 start mail input dan menentukan urutan karakter tertentu yang dijadikan sebagai tanda akhir pesan email.

QUIT dikirim terakhir untuk mengakhiri transaksi pengiriman pesan mail. Server merespon dengan mengirimkan pesan 221, yang berarti setuju untuk menghentikan transaksi. Kedua pihak akhirnya menutup hubungan TCP.

**2.4 Komponen E-mail**

Email terdiri dari tiga buah komponen, yaitu:

*Envelope,* atau amplop. Ini digunakan oleh MTA untuk pengiriman. Dalam contoh sebelumnya, envelope ditandai dengan dua buah perintah SMTP :

MAIL from:
<susan@students.ee.itb.ac.id>
RCPT to:
<susan@lskk.itb.ac.id>

*Header,* digunakan oleh *user agent*. Ada kurang lebih sembilan field header, yaitu: Received, Message-Id, From, Date, Reply-To, X-Phone, X-mailer, To dan Subject. Setiap field header berisi sebuah nama yang diikuti oleh sebuah titik dua (:), dan nilai dari field header tersebut.

*Body* merupakan isi pesan dari pengirim ke penerima.

**2.5 Sekuriti**

Untuk melihat keamanan sistem Internet perlu diketahui cara kerja system Internet. Antara lain, yang perlu diperhatikan adalah hubungan antara komputer di Internet, dan protokol yang digunakan. Internet merupakan jalan raya yang dapat digunakan oleh semua orang (*public*). Untuk mencapai server tujuan, paket informasi harus melalui beberapa system (router, gateway, hosts, atau perangkat-perangkat komunikasi lainnya) yang kemungkinan besar berada di luar kontrol dari kita. Setiap titik yang dilalui

memiliki potensi untuk dibobol, disadap, dipalsukan.

**2.5.1 Kriptografi (*cryptography*)**

Merupakan ilmu dan seni untuk menjaga pesan agar aman.. Para pelaku atau praktisi kriptografi disebut ***cryptographers***. Sebuah algoritma kriptografik (*cryptographic algorithm*), disebut **cipher**, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat.

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah **enkripsi** (*encryption*). Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi data anda disandikan (*encrypted*) dengan menggunakan sebuah kunci (*key*). Untuk membuka (*decrypt*) data tersebut digunakan juga sebuah kunci yang dapat sama dengan kunci untuk mengenkripsi (untuk kasus *private key cryptography*) atau dengan kunci yang berbeda (untuk kasus *public key cryptography*).

*Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah "*encipher*". Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut **Dekripsi** (*decryption*). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah "*decipher*". *Cryptanalysis* adalah seni dan ilmu untuk memecahkan *ciphertext* tanpa bantuan kunci. *Cryptanalyst* adalah pelaku atau praktisi yang menjalankan *cryptanalysis*. *Cryptology* merupakan gabungan dari

*cryptography* dan *cryptanalysis*.

### 2.5.2 Penggunaan Enkripsi Untuk Meningkatkan Keamanan

Salah satau mekanisme untuk meningkatkan keamanan adalah dengan menggunakan teknologi enkripsi. Data-data yang anda kirimkan diubah sedemikian rupa sehingga tidak mudah disadap. Banyak servis di Internet yang masih menggunakan "*plain text*" untuk *authentication*, seperti penggunaan pasangan userid dan password. Informasi ini dapat dilihat dengan mudah oleh program penyadap atau pengendus (*sniffer*). Contoh servis yang menggunakan plain text antara lain:

- Akses jarak jauh dengan menggunakan telnet dan rlogin
- Transfer file dengan menggunakan FTP
- Akses email melalui POP3 dan IMAP4
- Pengiriman email melalui SMTP
- Akses web melalui HTTP

## III. ANALISIS MASALAH

Internet awalnya dikembangkan untuk menghubungkan antar pihak yang saling dipercaya dengan tujuan saling bertukar informasi. Meningkatnya ketergantungan kita terhadap Internet telah mengurangi peluang kita untuk mempertahankan privasi. Perubahan fungsi serta komunitas pengguna internet tampaknya belum diikuti dengan perubahan drastis teknologi jaringan yang mendasarinya. Teknologi yang digunakan relatif masih memanfaatkan TCP/IP yang serba terbuka. Terbuka di sini bukan berarti source code atau standarnya diketahui banyak orang, tetapi dalam mekanismenya yang masih membuka alamat tujuan dan pengirimnya.

Ketertutupan informasi yang berkaitan dengan suatu protokol bukan merupakan suatu jaminan bahwa protokol itu akan lebih aman. Seperti diketahui, algoritma atau mekanisme kriptografi yang menjadi sandaran usaha penyusunan jalur komunikasi aman pun menggunakan algoritma yang mekanismenya diketahui oleh orang banyak.

Email merupakan layanan paling populer di Internet. Banyak orang memanfaatkan layanan ini baik untuk urusan pribadi maupun bisnis karena relatif murah dan cepat. Tata cara pengiriman e-mail yang telah diuraikan di atas menunjukkan rentannya email yang kita kirim dibaca orang ataupun diubah isinya kemudian diteruskan lagi ke alamat yang dituju, oleh system administratornya yang iseng. Hal itu tentunya membuat kita merasa tidak aman untuk menggunakan layanan tersebut. Maka untuk mengatasinya kita harus dapat memilih software sekuriti yang tepat.

## VI. Keamanan E-Mail Dengan GnuPG

GnuPG adalah software enkripsi email pengganti PGP yang lengkap dan bebas (lisensi GPL). Dibuat oleh tim GnuPG yang terdiri dari Matthew Skala, Michael Roth, Niklas Hernaeus, R Guyomarch and Werner Koch. Gael Queri, Gregory Steuck, Janusz A. Urbanowicz, Marco d'Itri, Thiago Jung Bauermann, Urko Lusa and Walter Koch yang membuat translasi resmi dan Mike Ashley yang mengerjakan *GNU Privacy Handbook.*

GnuPG adalah suatu program yang digunakan untuk mengamankan komunikasi dan penyimpanan data. Program ini dapat menyandikan data serta membuat tanda tangan digital. Karena tidak menggunakan algoritma yang dipatenkan, GnuPG dapat digunakan secara bebas. GnuPG menggunakan kriptografi Public key (public key cryptography) sehingga para penggunanya dapat saling berkomunikasi secara aman. Dalam sistem Public key, setiap pengguna mempunyai sepasang kunci yang terdiri dari Private key dan Public key. Private key dirahasiakan; hanya diketahui oleh pemiliknya, sementara Public key dapat diberikan pada siapa saja yang dikehendaki pemilik, sehingga pemilik dapat berkomunikasi dengan pengguna lain yang diberi Public key tersebut.

Bebas karena tidak menggunakan algoritma enkripsi yang telah dipatenkan sehingga bisa dipakai oleh siapa saja tanpa batasan. GnuPG memenuhi spesifikasi OpenPGP RFC2440.

Beberapa fitur yang ditawarkan GnuPG adalah: penggantian penuh terhadap pemakaian PGP.

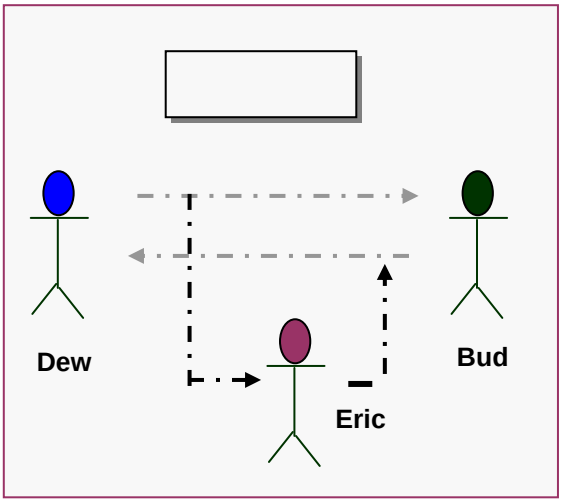
1. Dapat digunakan sebagai pengganti PGP (yang dipatenkan algoritmanya).
2. Tidak menggunakan algoritma yang dipatenkan.
3. Berlisensi GPL.
4. Ditulis dari nol, sehingga tidak menggunakan kode sumber atau algoritma dari program lainnya.
5. Implementasi penuh OpenPGP (RFC 2440)
6. Kemampuan yang lebih baik dibandingkan PGP.
7. Mampu menerjemahkan / memverifikasi pesan tersandi dari PGP 5.x
8. Mendukung algoritma ElGamal (tanda tangan dan penyandian), DSA, 3DES, BlowFish, TwoFish, CAST5, MD5, SHA-1, RIPE-MD-160 dan TIGER
9. Kemudahan implementasi algoritma penyandian baru dengan menggunakan modul ekstensi (extension module)
10. Identitas pengguna (UserID) diseragamkan dalam suatu bentuk standar.
11. Mendukung kunci dan tanda tangan yang dapat kadaluwarsa (hanya dapat digunakan dalam jangka waktu tertentu)
12. Mendukung bahasa Inggris, Denmark, Spanyol, Belanda, Perancis, Jerman, Jepang, Italia, Portugis (brasil dan portugal), Polandia, Rusia, dan swedia
13. Online Help system
14. Dapat mengirimkan Pesan kepada penerima anonim (optional)
15. Dukungan integral untuk HKP Keyserver (wwwkeys.pgp.net)
16. Mempunyai banyak program antarmuka grafis.

GnuPG bekerja sempurna di atas sistem operasi Linux dengan platform *x86, mips, alpha, sparc64* ataupun *powerpc*. Sistem operasi lain dengan platform x86 yang juga bekerja adalah *FreeBSD, OpenBSD, NetBSD* dan bahkan *Windows*. Platform lain dengan sistem operasi selain Linux masih dalam pengembangan.

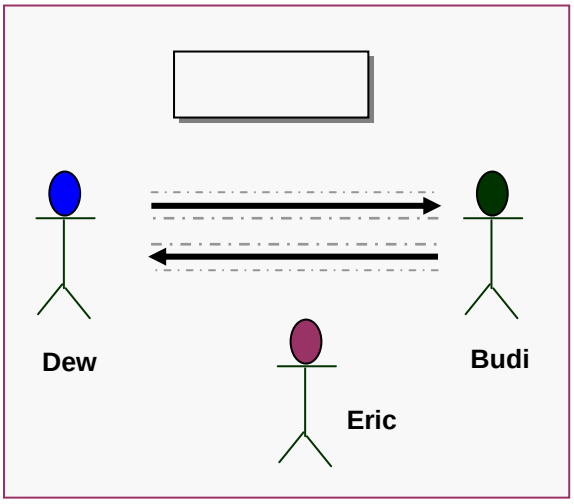
### 4.1 Public Key

Sejarah tentang kriptografi Public Key adalah berkenaan dengan penerjemahan, bermula dari cerita Whitfield Diffie Dan Martin Hellman yang pada tahun 1976 mengumumkan kepada dunia cara baru mengenkrip data,yaitu system kunci publik.

Cara kerjanya, contoh yang paling mudah adalah lalu lintas pesan yang disampaikan. Di bawah ini ada suatu ilustrasi Dewi Dan Budi dalam suatu percakapan, Dewi mengirimkan Budi suatu pesan sederhana, dan Budi menjawabnya dengan pesan sederhana pula. Tetapi Dewi Dan Budi tidak mengetahui adalah Eric tengah menginterupsi pesan ini, membaca, dan menyampaikan kembali. Ia mengubah sebagian dari pesan balasan yang dikirim ke Dewi dari Budi.




Sekarang Dewi dan Budi yang memanfaatkan kriptografi Public Key (dalam hal ini menggunakan Gnupg). semua mis-komunikasi dapat diketahui. Dewi dan Budi dapat mengenkrip pesan mereka sebelum dikirimkan, bahkan pesan itu tidak bisa dibaca oleh Erick. Juga tidak perlu menyembunyikan pesan mereka, tetapi harus memperhatikan dengan menguji identitas penulis pesan tersebut bahwa mereka bisa menandatangani pesan sebelum dikirim.




Secara keseluruhan proses kriptografi Public Key adalah Dewi mempuyai dua kunci yaitu Public Key dan Private Key, begitu juga Budi. Dewi dan Budi kemudian saling bertukar Public Key. Budi menulis suatu pesan ke Dewi, dan menggunakan Publik Key Dewi untuk mengenkrip pesan tersebut, setelah menerima pesan itu Dewi kemudian dapat mendekripsi pesan itu menggunakan Private Key Budi.

Dewi kemudian membalas pesan Budi agar dia tahu pesannya telah diterima. Setelah menulis balasannya Dewi kemudian menanandatangani pesan tersebut dengan Private Key Budi, kemudian Budi bisa memeriksa Tandatangan Dewi dengan menggunakan Publik Key Dewi. Jika segala sesuatunya sesuai Budi yakin bahwa pesan ini memang dari Dewi.

Berbagai hal untuk ingat Public Key yang kita berikan kepada setiap orang digunakan untuk mengenkrip pesan kepada kamu. Private Key yang kita miliki sesuatu yang harus dirahasiakan, karena Private Key kita satu-satunya kunci yang dapat mendekripsi pesan yang telah dienkrip dengan Public Key kita tersebut. Juga karena kita satu-satunya yang mempunyai akses ke Private Key kita, kemudian digunakan untuk memverifikasi pesan yang kita kirim.

Public Key crypto berbeda dari Private Key atau Symmetric Key. Sebelum adanya Public Key crypto orang-orang yang ingin suatu percakapan yang aman harus bertemu terlebih dahulu, dan saling bertukar key. Tetapi sekarang dengan Public Key crypto orang dapat terjamin keamanan percakapan, tetapi tidak sebelumnya sudah jumpa. Sistem ini mempunyai implikasi mengagumkan untuk suatu jaringan untuk orang yangsecara geografis tinggal berjauhan.

### 4.2 Menggunakan GnuPG

Software yang berhubungan dengan GnuPG dapat diperoleh di https://www.gnupg.org/download

#### 4.2.2 Perintah-Perintah Umum

**Membuat pasangan kunci**

Pertama-tama kita harus membuat sepasang kunci (kunci publik dan kunci pribadi) agar dapat menggunakan GnuPG dalam penyandian.

**Langkah 1**

Untuk membuat pasangan kunci, gunakan perintah :

```
[root@tasproject /]# gpg --gen-key

gpg (GnuPG) 1.0.2; Copyright (C) 2000
Free Software Foundation, Inc.
This program comes with ABSOLUTELY
NO WARRANTY.
This is free software, and you are
welcome to redistribute it
under certain conditions. See the file
COPYING for details.

gpg: /root/.gnupg: directory created
gpg: /root/.gnupg/options: new options
file created
gpg: you have to start GnuPG again, so
it can read the new options file
```

**Langkah 2**

Kita kembali menjalankan GnuPG dengan perintah :

[root@tasproject /]# gpg --gen-key
gpg (GnuPG) 1.0.4; Copyright (C) 2000 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg:/root/.gnupg/secring.gpg: keyring created
gpg: /root/.gnupg/pubring.gpg: keyring created
Please select what kind of key you want:
(1) DSA and ElGamal (default)
(2) DSA (sign only)
(4) ElGamal (sign and encrypt)
Your selection? {1} (*atau pilih sesuai kebutuhan anda*)

DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
minimum keysize is 768 bits
default keysize is 1024 bits
highest suggested keysize is 2048 bits
What keysize do you want? (1024) {2048} (*atau pilih sesuai kebutuhan anda*)
Do you really need such a large keysize? {y}
Requested keysize is 2048 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n> w = key expires in n weeks
<n> m = key expires in n months
<n> y = key expires in n years
Key is valid for? (0) {0} (*atau sesuai kebutuhan anda*)
Key does not expire at all
Is this correct (y/n)? {y}

You need a User-ID to identify your key; the software constructs the user id from Real Name, Comment and Email Address in this form:
Real name: {Dewi Yudo}
Email address: {Dewi@student.ee.itb.ac.id}
Comment: {tasproject} (*jika tidak diisi, tekan ENTER*)
You selected this USER-ID:
" Dewi Yudo (tasproject) < dewi@student.ee.itb.ac.id >"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? <o>
You need a Passphrase to protect your secret key.

Enter passphrase: {*masukkan kata sandi anda*}
We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy. ++++++++++.+++++^^^

public and secret key created and signed.

Sepasang kunci sandi telah dibuat, serta akan diletakkan pada direktori home dari root (~/root)

Kini akan kita mengulas mengenai berbagai masukan yang diminta saat pembuatan pasangan kunci

Please select what kind of key you want:

(1) DSA and ElGamal (default)
(2) DSA (sign only)
(4) ElGamal (sign and encrypt)
Your selection?

GnuPG mampu membuat tiga pasangan kunci sesuai algoritma penyandian yang diinginkan. Terdapat tiga pilihan :

1. Pilihan (1) akan menciptakan pasangan kunci DSA dan ElGamal. Pasangan kunci DSA adalah pasangan kunci primer yang digunakan untuk membuat tanda tangan digital, pasangan kunci ElGamal dibuat sebagai kunci pelengkap untuk melakukan penyandian.

2. Pilihan (2) hanya menciptakan pasangan kunci DSA saja.

3. Pilihan (4) akan menciptakan pasangan kunci ElGamal yang dapat digunakan untuk penyandian dan tanda tangan digital.

Pilihan [1] telah memadai bagi kebanyakan pengguna.

DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
minimum keysize is 768 bits
default keysize is 1024 bits
highest suggested keysize is 2048 bits
What keysize do you want? (1024)

Anda akan diminta untuk memilih ukuran kunci. Kunci DSA akan mempunyai ukuran 1024 bits, sedang ElGamal dapat bervariasi antara tiga pilihan diatas. Ada keuntungan dan kerugian akibat pemilihan kunci berukuran besar. Keuntungannya adalah: Semakin panjang kunci, semakin aman kunci tersebut terhadap bruteforce attack

Kerugiannya :

1. Proses penyandian data dan penerjemahan data tersandi akan lebih lama, sebanding dengan besarnya ukuran kunci yang digunakan.
2. Ukuran kunci yang lebih besar akan mempengaruhi panjang tanda tangan

Ukuran kunci default adalah 1024 bits. Ini telah memadai untuk hampir semua keperluan. Ukuran kunci tidak dapat diubah lagi setelah dipilih.

Kemudian anda harus menentukan berapa lama kunci ini berlaku :

Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n> w = key expires in n weeks
<n> m = key expires in n months
<n> y = key expires in n years
Key is valid for? (0)

Untuk kebanyakan pengguna, kunci yang tidak pernah kadaluwarsa telah memadai. Jika anda memilih untuk menggunakan kunci yang kadaluwarsa pada jangka waktu tertentu, maka jangka waktunya haruslah ditentukan dengan seksama, karena meski jangka waktu kadaluwarsa dapat diubah setelah kunci dibuat, kemungkinan akan sulit memberitahukan perubahannya pada pengguna yang memiliki kunci publik anda.

Anda juga harus memasukkan identitas anda dalam kunci yang akan dibuat :

You need a User-ID to identify your key; the software constructs the user id from Real Name, Comment and Email Address in this form:
"Dewi Yudo (tasproject) <Dewi@ee.itb.ac.id>"

Real name: *isikan nama*
Email address: *isikan alamat e-mail*
Comment : *isikan keterangan tambahan lain yang ingin anda sertakan*

GnuPG membutuhkan kata sandi (atau kalimat sandi) untuk melindungi kunci pribadi dan kunci publik yang anda punya. Anda harus mengisikan kata sandi untuk melindungi kunci pribadi anda.

You need a Passphrase to protect your secret key.

Enter passphrase: {*masukkan kata sandi anda*}

Panjang kata sandi adalah tidak terbatas. Kata sandi haruslah dipilih secara seksama, karena dari sudut pandang keamanan, bagian paling lemah dari GnuPG (dan sistem penyandian lainnya) adalah kata sandi (yang digunakan untuk membuka kunci pribadi ), karena perlindungan terakhir yang anda punya jika kunci pribadi anda diketahui orang lain adalah katasandi ini. Idealnya, kata sandi tidak boleh menggunakan kata-kata yang terdapat dalam kamus, serta mengandung campuran huruf kapital, huruf kecil, angka dan karakter non-alfabet lainnya. Kata sandi yang baik sangat krusial dalam keamanan penggunaan GnuPG.

### 4.2.3 Membuat Revocation Certificate

Segera setelah pasangan kunci dibuat, sebaiknya dibuat pula sertifikat penarikan-kembali (revocation certificate) untuk kunci publik primer menggunakan option --gen-revoke. Jika anda lupa kata sandi yang digunakan atau kunci pribadi anda hilang atau jatuh ke tangan orang lain, sertifikat penarikan-kembali ini dapat dipublikasikan untuk memberitahukan kepada pihak-pihak lain bahwa kunci publik anda yang mereka punyai sebaiknya tidak lagi digunakan.

```
[root@tasproject /]# gpg --output sertifikat-darurat.asc --gen-revoke kunci
```

Disini, kunci adalah pengenal kunci anda, dapat berupa IDkey dari pasangan kunci primer atau bagian lain dari UserID yang mengidentifikasikan pasangan kunci anda. Sertifikat akan dibuat pada file "sertifikat-darurat.asc". Sebaiknya sertifikat tidak disimpan pada direktori dimana orang lain dapat mengakses, karena jika demikian dapat terjadi seseorang mempublikasikan sertifikat penarikan-kembali dan mengakibatkan kunci publik menjadi tidak berguna.

### 4.2.4 Melihat Daftar Kunci Publik

Untuk melihat daftar kunci publik yang anda miliki, gunakan option --list-keys

```
[root@tasproject /]# gpg --list-keys

/root/.gnupg/pubring.gpg
--------------------------------
pub  1024D/5920142A 2001-02-22
Dewi Yudo (tasproject)
< Dewi@student.ee.itb.ac.id >
sub  1024g/3A9EEFCE 2001-02-22
[expires: 2001-05-23]
```

### 4.2.5 Mengekspor kunci publik

Anda dapat mempublikasikan kunci publik yang anda punya pada personal website, melalui berbagai keyserver di Internet, atau beragam cara lain. Untuk mengirimkan kunci publik anda pada orang lain, anda harus mengekspor kunci publi ktersebut dengan menggunakan option --export. argumen tambahan diperlukan untuk menentukan kunci publik yang akan diekspor.

Untuk mengekspor kunci publik anda dalam format binary, gunakan perintah berikut :

```
[root@tasproject /]# gpg --output k-publik_saya.gpg --export dewi
```

Option --output digunakan untuk menunjukan file keluaran kunci publik (binary), sedang --export diikuti identitas pemilik (dapat berupa nama, komentar atau e-mail) mengidentifikasikan pemilik kunci-publik yang diekspor.

Untuk mengekspor kunci publik dalam bentuk ASCII armored, gunakan :

```
[root@tasproject /]# gpg --export-armor > kunci_saya.asc
```

Disini option --export digunakan untuk mengekstrak kunci publik anda sedang option --armor akan membuat kunci publik tersebut berupa karakter ASCII, yang dapat ditampilkan pada website.

### 4.2.6 Mengimpor Kunci Publik

Setelah pasangan kunci selesai dibuat, anda harus memasukkan pasangan kunci tersebut kedalam database pasangan kunci (yang berisi koleksi pasangan kunci dari pihak lain yang dapat digunakan untuk penyandian / penterjemahan pesan tersandi dalam komunikasi antar personal). Untuk memasukkan pasangan kunci anda (atau pasangan kunci pihak lain) gunakan option --import

```
[root@tasproject /]# gpg --import <filename>
```

dengan <filename> adalah kunci publik yang ingin dimasukkan kedalam database.
Sebagai contoh :

```
[root@tasproject /]# gpg --import redhat.asc
gpg: key :9B4A4024: public key imported
```

gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: Total number processed: 1
gpg: imported: 1

Pada contoh diatas, kita memasukkan kunci publik RedHat Linux, yang terdapat dalam file redhat.asc (dapat didownload dari situs RedHat) kedalam database koleksi kunci publik kita.

### 4.2.7 Memvalidasi Kunci

Setelah kunci publik diimpor ke dalam databse kita, kunci tersebut harus divalidasi dengan memverifikasi "sidik jari kunci" (key fingerprint), dan kemudian memandatangani kunci tersebut untuk mengesahkannya menjadi kunci resmi. Sidik jari dari kunci dapat dilihat dengan option --fingerprint

[root@tasproject /]# gpg --fingerprint <UserID>

Contoh :

[root@tasproject /]# gpg --fingerprint Mandrake
pub 1024D/9B4A4024 2000-01-06 MandrakeSoft (MandrakeSoft official keys)
<mandrake@mandrakesoft.com>
Key fingerprint = 63A2 8CBD A7A8 387E 1A53 2C1E 59E7 0DEE 9B4A 4024
sub 1024g/686FF394 2000-01-06

Pada contoh diatas, kita memverifikasi kunci publik milik Mandrakesoft. Sidik jari kunci diverifikasi dengan mencocokan pada sidik jari kunci yang dimiliki oleh pemilik (dalam hal ini MandrakeSoft). Hal ini dapat dilakukan secara langsung, melalui telepon, e-mail atau sarana lain, sejauh dapat menjamin bahwa kita benar-benar berhubungan dengan pemilik kunci sebenarnya. Jika setelah dicocokkan hasilnya sama, maka kita dapat memastikan bahwa salinan kunci publik Mandrake yang kita punya adalah benar-benar kunci publik dari Mandrakesoft.

### 4.2.8 Menandatangani Kunci

Setelah mengimpor dan memverifikasi kunci publik yang kita masukkan pada database, kini kita dapat menandatangani kunci tersebut. Dengan menandatangani kunci kita menyatakan bahwa kita mengetahui pemilik kunci tersebut. Sebaiknya kita menandatangani kunci publik hanya jika kita 100% yakin akan keaslian kunci tersebut.

Untuk menandatangani kunci publik RedHat yang telah kita impor kedalam database, gunakan perintah :

[root@tasproject /]# gpg --sign-key <UserID>

Contoh :

[root@tasproject /]# gpg --sign-key RedHat

pub 1024D/DB42A60E created:1999-09-23 expires : never trust:-/q
sub 2048g/961630A2 created:1999-09-23 expires : never
(1)Red Hat, Inc
<security@redhat.com>

pub 1024D/DB42A60E created:1999-09-23 expires : never trust:-/q
Fingerprint : CA20 8686 2BD6 9DFC 65F6 ECC4 2191 80CD DB42 A60E

Red Hat, Inc <security@redhat.com>

Are you really sure that you want to sign this key
with your key :" Dewi Yudo (tasproject)" < dewi@student.ee.itb.ac.id >

Really Sign ? {y}

You need passphrase to unlock the secret key for
user :"Dewi Yudo (tasproject) < dewi@student.ee.itb.ac.id >"
1024 Bit DSA Key, ID 5920142A Created 2000-02-22

Enter passphrase : {*masukkan kata kunci*}

### 4.2.9 Memeriksa tanda tangan

Setelah kunci publik kita tandatangani, kita dapat memeriksa kunci tersebut untuk melihat daftar tandatangan yang ada pada kunci tersebut, serta tanda tangan yang kita tambahkan. Setiap UserID di kunci akan mempunyai satu atau lebih tanda tangan. Kita dapat memeriksa tanda tangan dari suatu kunci dengan menggunakan option --check-sigs.

Contoh :

[root@tasproject /]# gpg --check-sigs mandrake
pub 1024D/9B4A4024 2000-01-06 MandrakeSoft (MandrakeSoft official keys)
<mandrake@mandrakesoft.com>
sig! 9B4A4024 2000-01-06 MandrakeSoft (MandrakeSoft official keys)
<mandrake@mandrakesoft.com>
sig! 5920142A 2000-02-22 Tunggul Arif Siswoyo (tasproject)
<orion@student.undip.ac.id>
sub 1024g/686FF394 2000-01-06
sig! 9B4A4024 2000-01-06 MandrakeSoft (MandrakeSoft official keys)
mandrake@mandrakesoft.com

### 4.2.10 Penyandian dan Penterjemahan Data Tersandi

Cara menyandikan dan menterjemahkan data tersandi adalah sangat mudah. Jika kita ingin mengirimkan data untuk RedHat, maka sandikan data tersebut dengan menggunakan kunci publik dari RedHat, hingga hanya RedHat Inc, saja yang dapat menterjemahkan data tersandi tersebut dengan menggunakan kunci pribadinya. Jika Mandrake ingin mengirim pesan pada kita, Mandrake akan menyandikan pesan dengan menggunakan kunci publik milik kita, dan kita akan menterjemahkan data tersandi tersebut menggunakan kunci pribadi milik kita.

Untuk menyandikan data yang akan kita tujukan pada penerima, (kita harus mempunyai kunci publik penerima data) gunakan perintah berikut :

[root@tasproject /]# gpg -sear RedHat file-untuk-redhat.txt

You need passphrase to unlock the secret key for
user :"Dewi Yudo (tasproject) < dewi@student.ee.itb.ac.id >"
1024 Bit DSA Key, ID 5920142A Created 2000-02-22

Enter passphrase : {masukkan kata kunci}

Option "s" digunakan untuk menan-datangani pesan (signed),"e" untuk menyandikan pesan (encrypt), "a" untuk membuat file menjadi tersandi ASCII-armor (akan muncul file "file-untuk-redhat.asc" yang dapat dikirimkan via e-mail), "r" untuk menyandikan UserID penerima (dalam hal ini RedHat), dan file-untuk-redhat.txt adalah data yang ingin kita sandikan.

Untuk menerjemahkan pesan tersandi gunakan :

[root@tasproject /]gpg -d pesan_tersandi.asc

You need passphrase to unlock the secret key for
user :"Dewi Yudo (tasproject) < dewi@student.ee.itb.ac.id >"
1024 Bit DSA Key, ID 5920142A Created 2000-02-22

Enter passphrase : {masukkan kata kunci}

Disini, option -d (decrypt) akan menterjemahkan data tersandi berupa file "pesan_tersandi.asc"

Anda harus mempunyai kunci publik pengirim pesan pada database anda agar dapat menerjemahkan pesan tersandi dari pengirim tersebut.

### 4.2.11 Memeriksa Tanda tangan

Setelah kita selesai membuat pasangan kunci dan mempublikasikannya, dengan menggunakan option --verify dari GnuPG pihak lain dapat memeriksa

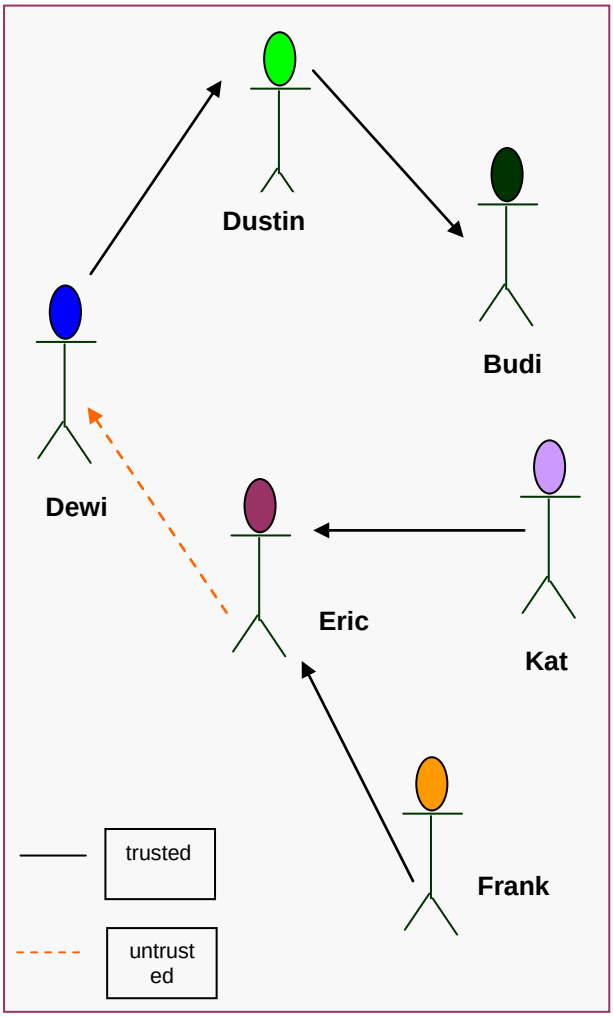
apakah data tersandi yang kita kirimkan, kita tandatangani.

Untuk memeriksa tanda tangan dari pesan tersandi, gunakan :

```
[root@tasproject /]# gpg --verify pesan_tersandi.asc
```

option --verify akan memeriksa tanda tangan dari file "pesan_tersandi.asc"

Gambaran di bawah menunjukkan bahwa jika Dewi percaya Dustin, dan Dustin percaya Budi, kemudian oleh transitivas ( yaitu. wakil) Dewi dapat percaya Budi. Jika Budi mengirimkan Dewi suatu pesan ditandatangani yang dapat dipasti kan tandatangan adalah sah dan pesan yang benar-benar datang dari Budi. Hal lain meskipun jika Kat percayaan pada Eric, tetapi Eric Tidak percaya Dewi, kemudian Kat dengan berterus terang tidak percaya Dewi.




**KESIMPULAN**

Jika kita merasa email atapun data yang ada di sistem adalah hal penting dan memerlukan pengamanan, ada baiknya GnuPG digunakan untuk memperkuat sistem pengamanan data yang ada. GnuPG adalah salah satu program open-source penyandian data terbaik yang ada, yang dapat digunakan secara bebas, karena program ini berlisensi GPL. Pada saat ini Open Source merupakan salah satu kandidat untuk penyediaan infrastruktur sistem yang aman. Tidak saja aman dari sisi teknologi tapi juga dari sudut pandang ketergantungan suatu negara.

Program ini dapat digunakan untuk menyandikan pesan e-mail, data atau dokumen rahasia juga dapat juga digunakan untuk mengirimkan data tersandi via jaringan secara aman.

Dari uraian di atas beberapa kegunaan GnuPG :

1. Mengirimkan e-mail tersandi.
2. Menyandikan data dan dokumen.
3. Mengirimkan data / dokumen tersandi melalui jaringan.

**REFERENSI**


1. GnuPG: Using Public Key Enkripsi School of Informatics, University of Edinburgh
2. http://www.inf.ed.ac.uk/teaching/modules/cs
3. the GPG handbook https://www.gnupg.org/docs;
4. GnuPG home page https://www.gnupg.org.
5. (http://www.linux-mandrake.com/en/cookerdevel.php3 ). Tools That Use GPG
6. Enigmail (http://enigmail.mozdev.org/)

7. Evolution (http://ximian.com/products/evolution/)
8. GPA (http://www.gnupg.org/gpa.html)