Bahan Kuliah ke-26

IF5054 Kriptografi

Kriptografi dalam Kehidupan Sehari-hari (Bagian 2)

Disusun oleh:

Ir. Rinaldi Munir, M.T.

Departemen Teknik Informatika
Institut Teknologi Bandung
2004

25.5 *E-commerce* di Internet

- *E-commerce* adalah transaksi barang dan jasa dengan menggunakan internet sebagai medianya. Di dalam dunia *e-commerce* ada pedagang (*merchant*) dan pembeli (*customer*). Padagang menawarkan barang/jasa melalui situs *web* yang dapat diakses oleh pembeli dari lokasi manapun di muka bumi. Dalam hal ini, situs *web* pedagang disebut *server* sedangkan pembeli disebut *client*. Pembeli mengakses *web* dengan program *browser* seperrti *Internet Explorer*.

  Situs *web Amazon.com* merupakan situs *e-commerce* yang terkenal (Gambar 25.4).

Gambar 25.4 Situs *amazon.com*

- Pembayaran barang umunya dilakukan dengan menggunakan kartu kredit, yang berarti bahwa pembeli harus mengirimkan nomor kartu kredit dan informasi lainnya melalui internet.

- Karena alasan keamanan yang menyangkut informasi kartu kredit maka transaksi barang lewat internet tidak terlalu populer. Banyak orang yang masih beranggapan *e-commerce* tidak aman; kekhawatiran yang wajar, namun masalah ini sebnarnya sudah dipikirkan solusinya. *Browsing web* secara aman adalah fitur paling penting pada *e-commerce*.

Secure Socket Layer (SSL)

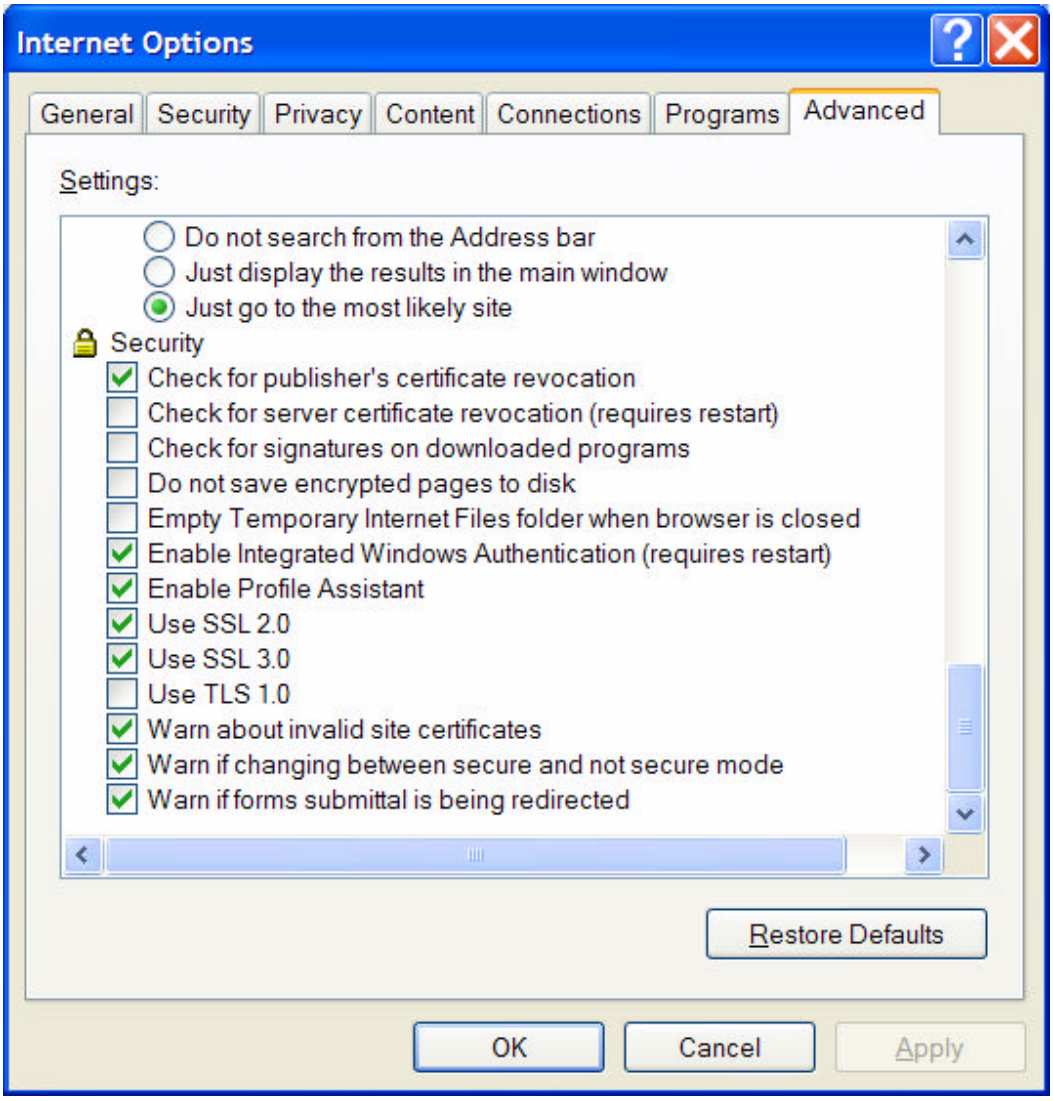
- *Secure Socket Layer* (*SSL*) adalah protokol yang digunakan untuk *browsing web* secara aman. Dalam hal ini, *SSL* bertindak sebagai protokol yang mengamankan komunikasi antara *client* dan *server.* Protokol ini memfasilitasi penggunaan enkripsi untuk data yang rahasia dan membantu menjamin integritas informasi yang dipertukarkan antara *website* dan *web browser.*

  SSL dikembangkan oleh *Netscape Communations* pada tahun 1994, dan menjadi protokol yang umum digunakan untuk komunikasi aman antara dua komputer pada Internet. *SSL* dibangun ke dalam banyak *web browser* (termasuk *Netscape Communicator* dan *Internet Explorer*). Ada beberapa versi *SSL*, versi 2 dan versi 3, tetapi versi 3 paling banyak digunakan saat ini.

- Untuk memastikan apakah *Internet Explorer* sudah siap menjalankan protokol *SSL*, klik dari *IE*:

  Tools → *Internet Options* → *Advanced*

  lalu cari pilihan *Security*, kemudian periksa apakah *SSL* versi 2.0 atau *SSL* versi 3.0 telah diberi tanda √ (Gambar 25.5).



Gambar 25.5 Opsi penggunaan *SSL* pada fitur *security* di dalam *Internet Explorer*.

- *SSL* beroperasi antara protokol komunikasi *TCP/IP* (*Transmission Control Protocol*/*Internet Protocol*) dan aplikasi (lihat gambar 25.5). *SSL* seolah-olah berlaku sebagai lapisan(*layer*) baru antara lapisasan transpor (*TCP*) dan lapisan aplikasi.

  TCP/IP adalah standard protokol yang digunakan untuk menghubungkan komputer dan jaringan dengan jaringan dari jaringan yang lebih besar, yaitu Internet.

Application (*HTTP, FTP, Telnet*)
Security (*SSL*)
Transport (*TCP*)
Network (*IP*)
Data link (*PPP*)
Phyiscal (*modem, ADSL, cable* TV)

Gambar 25.5 Lapisan (dan protokol) untuk *browsing* dengan *SSL*

- Di dalam standar komunikasi di Internet, pesan dari pengirim dilewatkan melalui *socket* (*port* khusus yang menerima dan mengirim informasi dari jaringan dengan *mode byte stream*). *Socket* kemudian menerjemahkan pesan tersebut melalui protokol *TCP/IP* (*Transmission Control Protocol*/*Internet Protocol*).

- Kebanyakan transmisi pesan di Internet dikirim sebagai kumpulan potongan pesan yang disebut **paket**. Pada sisi pengiriman, paket-paket dari sebuah pesan diberi nomor secara sekuensial. *IP* bertanggung jawab untuk merutekan paket (lintasan yang dilalui oleh paket), dan setiap paket mungkin menempuh rute yang berbeda di dalam Internet.

Tujuan sebuah paket ditentukan oleh *IP address*, yaitu nomor yang digunakan untuk mengidentifikasi sebuah komputer pada sebuah jaringan.

- Pada sisi penerima, *TCP* memastikan bahwa suatu paket sudah sampai, menyusunnya sesuai nomor urut, dan menentukan apakah paket tiba tanpa mengalami perubahan (misalnya berubah karena *physical error* selama transmisi). Jika paket mengalami perubahan atau ada data yang hilang, *TCP* meminta pengiriman ulang.

- Bila semua paket dari pesan berhasil mencapai *TCP/IP*, pesan tersebut kemudian dilewatkan ke *socket* penerima. *Socket* tersebut menerjemahkan pesan kembali menjadi bentuk yang dibaca oleh aplikasi penerima (contoh aplikasi adalah *HTTP*, *FTP*, *Telnet*).

- Dari penjelasan di atas dapat dilihat bahwa pada dasarnya *TCP/IP* tidak memiliki pengamanan komunikasi yang bagus. Bahkan, *TCP* tidak cukup canggih menentukan bilamana suatu paket berubah karena diubah oleh pihak ketiga (musuh), karena paket yang diubah tersebut dapat dianggap oleh *TCP* sebagai paket yang benar.

  Pada transaksi yang menggunakan *SSL*, *SSL* membangun hubungan (*connection*) yang aman antara dua *socket*, sehingga pengiriman pesan antara dua entitas dapat dijamin keamanannya.

- *SSL* disusun oleh dua sub-protokol:
  1. *SSL handshaking*, yaitu sub-protokol untuk membangun koneksi (kanal) yang aman untuk berkomunikasi,

2. *SSL record*, yaitu sub-protokol yang menggunakan kanal yang sudah aman. *SSL Record* membungkus seluruh data yang dikirim selama koneksi.

- *SSL* mengimplementasikan kriptografi kunci-publik dengan menggunakan algoritma *RSA* dan sertifikat digital untuk mengotentikasi *server* di dalam trasnsaksi dan untuk melindungi informasi rahasia yang dikirim antara dua buah *socket. Server* selalu diotentikasi, sedangkan *client* tidak harus diotentikasi oleh *server*.

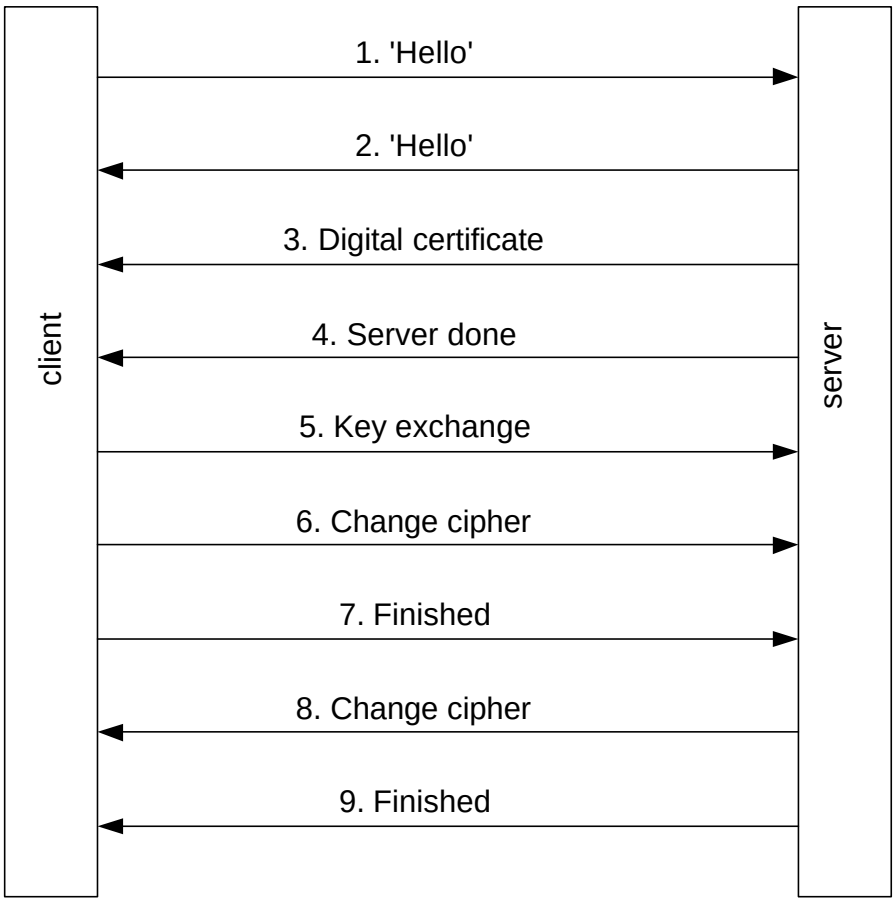
  Server diotentikasi agar *client* yakin bahwa ia mengakses situs *web* yang sah (dan bukan situs *web* palsu yang menyamar seolah-olah benar ia adalah *server* yang asli).

  Client tidak harus diotentikasi oleh *server* karena kebanyakan *server* menganggap nomor kartu kredit sudah cukup untuk mengotentikasi *client.*

- Perlu dicatat bahwa *SSL* adalah protokol *client-server*, yang dalam hal ini *web browser* adalah *client* dan *website* adalah *server. Client* yang memulai komunikasi, sedangkan *server* memberi respon terhadap permintaan *client*. Protokol *SSL* tidak bekerja kalau tidak diaktifkan terlebih dahulu (biasanya dengan meng-klik tombol yang disediakan di dalam *web server*)

Sub-protokol handshaking

- Sub-protokol *handshaking* diperlihatkan pada Gambar 25.6. Dari gambar tersebut terlihat bahwa *SSL* dimulai dengan pengiriman pesan *Hello* dari *client* ke *server* (1). *Server* merespon dengan mengirim pesan *Hello* (2) dan sertifikat digital ke *client* untuk otentikasi (3).



Gambar 25.6. Sub-protokol *handshaking* untuk membangun koneksi yang aman

Sertifikat digital berisi kunci publik *server*. Di dalam *browser client* terdapat daftar *CA* yang dipercaya. Jika sertifikat digital ditandatangi oleh salah satu *CA* di dalam daftar tersebut, maka *client* dapat memverifikasi kunci publik *server*.

Setelah proses otentikasi selesai, *server* mengirimkan pesan *server done* (4) kepada *client*.

Selanjutnya, *client* dan *server* menyepakati *session key* untuk melanjutkan transaksi melalui proses yang disebut *key exchange* (5). *Session key* adalah kunci rahasia yang digunakan selama transaksi. Nantinya, komunikasi antara *client* dan *server* dilakukan dengan menggunakan *session key* ini. Data yang akan ditansmisikan dienkripsi terlebih dahulu dengan *session key* melalui protokol *TCP/IP.*

Proses *exchange key* diawali dengan *client* mengirim nilai acak 384-bit yang disebut *premater key* kepada *server*. Nilai acak ini dikirim dalam bentuk terenkripsi (dienkripsi dengan kunci publik server). Melalui perhitungan yang cukup kompleks, *client* dan *server* menghitung *session key* yang diturunkan dari *premaster key*.

Setelah pertukaran kunci, *client* dan *server* menyepakati algoritma enkripsi (6). *SSL* mendukung banyak algoritma enkripsi, antara lain *DES*, *IDEA*, *RC2*, dan *RC4*. Sedangkan untuk fungsi *hash*, *SSL* mendukung algoritma *SHA* dan *MD5.*

Client mengirim pesan bahwa bahwa ia sudah selesai membangun sub-protokol (pesan 7). *Server* merespon *client* dengan mengirim pesan 8 dan 9.

Sampai di sini, proses pembentukan kanal yang aman sudah selesai. Bila sub-protokol ini sudah terbentuk, maka *http://* pada *URL* berubah menjadi *https://* (*http secure*)

Proses *SSL* yang cukup panjang ini mengakibatkan sistem menjadi lambat. Oleh karena itu, *SSL* diaktifkan hanya jika *client* memerlukan transmisi pesan yang benar-benar aman.

Sub-protokol SSL record

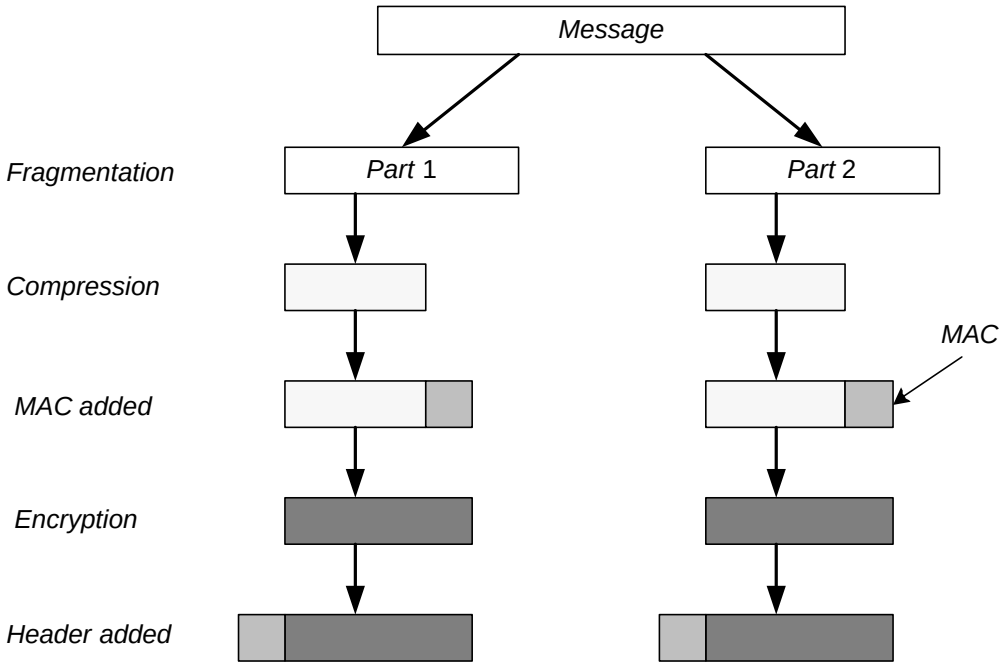
- Setelah kanal yang aman terbentuk, *client* dan *server* menggunakannya untuk menjalankan sub-protokol kedua (*SSL record*) untuk saling berkirim pesan. Misalnya *client* mengirim *HTTP request* ke *server*, dan *server* menjawab dengan mengirim *HTTP response*.

- Pesan dari *client* ke *server* (dan sebaliknya) dikirim dalam bentuk terenkripsi (pesan dienkripsi dengan menggunakan *session key*). Tetapi, sebelum pesan dikirim dengan *TCP/IP*, protokol *SSL* melakukan proses pembungkusan data sebagai berikut:
  1. pesan dipecah menjadi sejumlah blok yang masing-masing panjangnya 16 KB; setiap blok diberi nomor urut sekuensial.
  2. setiap blok kemudian dikompresi, lalu hasil kompresi disambung (concate) dengan session key;
  3. kemudian, hasil dari langkah 2 di atas di-*hash* dengan algoritma *MD5* (atau algoritma *hash* lain yang disepakati). Nilai *hash* ini ditambahkan ke setiap blok sebagai *MAC* (*Message Authentication Code*). Jadi, *MAC* dihitung sebagai berikut:

     MAC = Hash(*session key, compressed data block*)

  4. hasil dari langkah 3 kemudian dienkripsi dengan algoritma kriptografi simetri (misalnya RC4).
  5. terakhir, hasil dari langkah 4 diberi header (2 atau 3 byte), baru kemudian dikirim melalui koneksi TCP/IP aman yang terbentuk sebelumnya.

  Proses pembungkusan pesan oleh sub-protokol *SSL record* diperlihatkan pada Gambar 25.7.



Gambar 25.7. Pembungkusan pesan oleh *SSL record*

- Setelah data sampai di tempat penerima, sub-protokol *SSL* ini melakukan proses berkebalikan: mendekripsi data yang diterima, mengotentikasinya (dengan *MAC*), men-dekompresinya, lalu merakitnya.

- Meskipun *SSL* melindungi informasi yang dikirim melalui internet, tetapi ia tidak melindungi informasi yang sudah disimpan di dalam *server* pedagang (*merchant*). Bila pedagang *online* menerima informasi kartu kredit atas suatu pesanan barang, informasi tersebut mungkin di-dekripsi dan disimpan di dalam *server* pedagang sampai pesanan barang diantar. Jika *server* tidak aman dan data di dalamnya tidak dienkripsi, pihak yang tidak berhak dapat aja mengakses informasi rahasia tesrebut.

- Piranti keras, seperti kartu *peripheral component interconnect* (*PCI*) yang dirancang untuk digunakan di dalam transaksi *SSL*, dapat dipasang ke daalm *web server* untuk memproses transaksi *SSL*, sehingga mengurangi waktu pemrosesan dan memungkinkan server bebas mengerjakan tugas-tugas lain.

- Informasi lebih lanjut mengenai *SSL* dapat diperoleh dari tutorial *SSL* di `www.netscape.com/security/index.html`.

- Pada Tahun 1996, *Netscape Communications Corp.* mengajukan *SSL* ke *IETF* (*Internet Engineering Task Force*) untuk standardisasi. Hasilnya adalah *TLS* (*Transport Layer Security*). *TLS* dijelaskan di dalam *RFC* 2246 (untuk informasi lebih lanjut perihal *TLS*, kunjungi situs *IETF* di `www.ietf.org/rfc/rfc2246.txt`).

- *TLS* dapat dianggap sebagai *SSL* versi 3.1, dan implementasi pertamanya adalah pada Tahun 1999, tetapi belum jelas apakah *TSL* akan menggantikan *SSL*.

- *Wireless Transport Layer Security* (*WTLS*) adalah protokol keamanan data untuk *Wireless Application Protocol* (*WAP*). *WAP* adalah standard untuk komunikasi nirkabel (*wireless*) pada telepon *mobile* dan peralatan nirkabel lainnya. *WTLS* mengamankan kanal untuk komunikasi antara peralatan nirkabel dan *server* aplikasi.

25.6 Pengamanan *E-mail* dengan *PGP* (*Pretty Good Privacy*)

- *Pretty Good Privacy* atau *PGP* dikembangkan oleh Phil Zimmermann pada akhir tahu 1980. Pada mulanya, *PGP* digunakan untuk melindungi surat elektronik (*e-mail*) dengan memberi perlindungan kerahasiaan (enkripsi) dan otentikasi (tanda-tangan digital). Saat ini *PGP* tidak hanya ditujukan untuk keamanan *e-mail*, tetapi juga untuk keamanan berbagai file dan program pada komputer personal (*PC*).

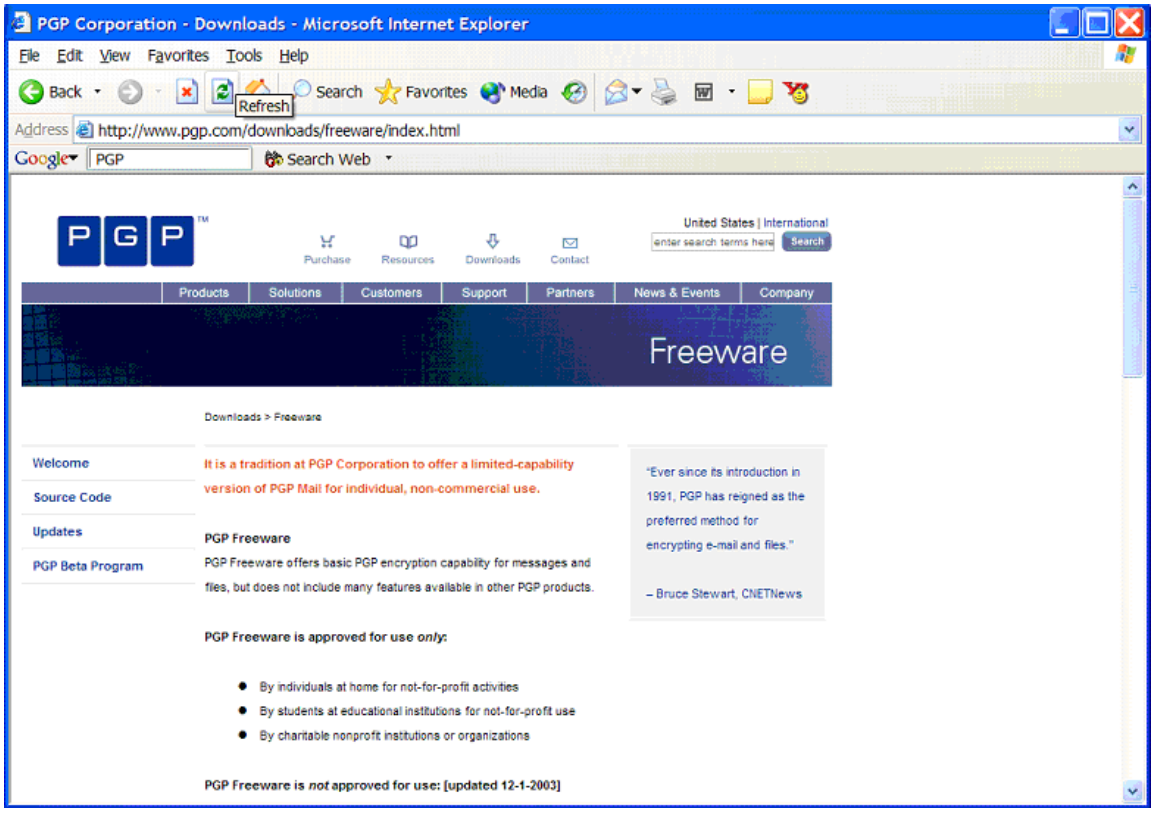
- *PGP* menggunakan kriptografi simetri dan kriptografi kunci-publik. Oleh karena itu, *PGP* mempunyai dua tingkatan kunci, yaitu kunci rahasia (simetri) – yang disebut juga *session key* – untuk enkripsi data, dan pasangan kunci privat-kunci publik untuk pemberian tanda tangan dan melindungi kunci simetri.

- *PGP* tersedia sebagai *freeware* maupun sebagai paket komersil dalam berbagai versi yang dapat dioperasikan dalam berbagai sistem operasi (*DOS*, *Windows*, *UNIX*, *Mac*). *Download* program *PGP* gratis dari situs `www.pgp.org` atau `www.pgpi.org` (lihat Gambar 25.8).

- Kunci simetri hanya dipakai sekali (*one-time*) dan dibuat secara otomatis dari gerakan tetikus (*mouse*) atau ketikan tombol kunci.

- Kode sumbernya juga dapat diakses dari Internet. *PGP* terbaru adalah *PGP* versi 8. *PGP* versi-versi awal menggunakan *IDEA* sebagai algoritma simetri dan *RSA* sebagai algoritma kunci-publik (asimetri), sedangkan versi-versi terakhir menggunakan algoritma *CAST* sebagai

algoritma simetri dan algoritam *DH* (Diffie-Hellman) sebagai algoritma kunci-publik.



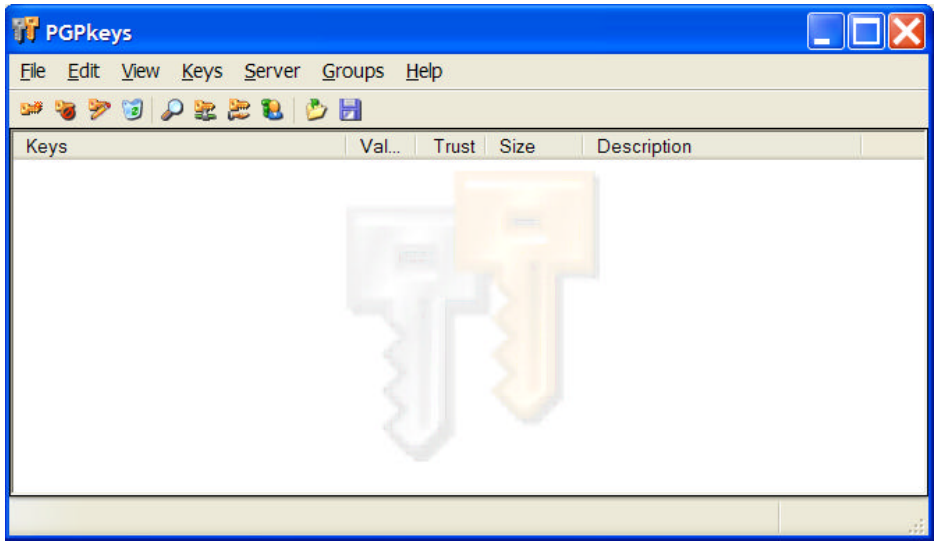
Gambar 25.8 Situs `www.pgp.com`

- *Download* dari situs *PGP* program *PGP* versi 8.0 *for Windows*, lalu instalasi PGP 8.0 ke dalam komputer anda.

- Pada versi *freeware* ini, ada tiga program *PGP* yang tersedia: *PGPdisk*, *PGPkeys* (pembangkitan dan manajemen kunci), dan *PGPmail* (enkripsi dan tanda-tangan digital untuk *file* maupun *e-mail*).

Membuat Pasangan Kunci Privat-Kunci Publik Baru

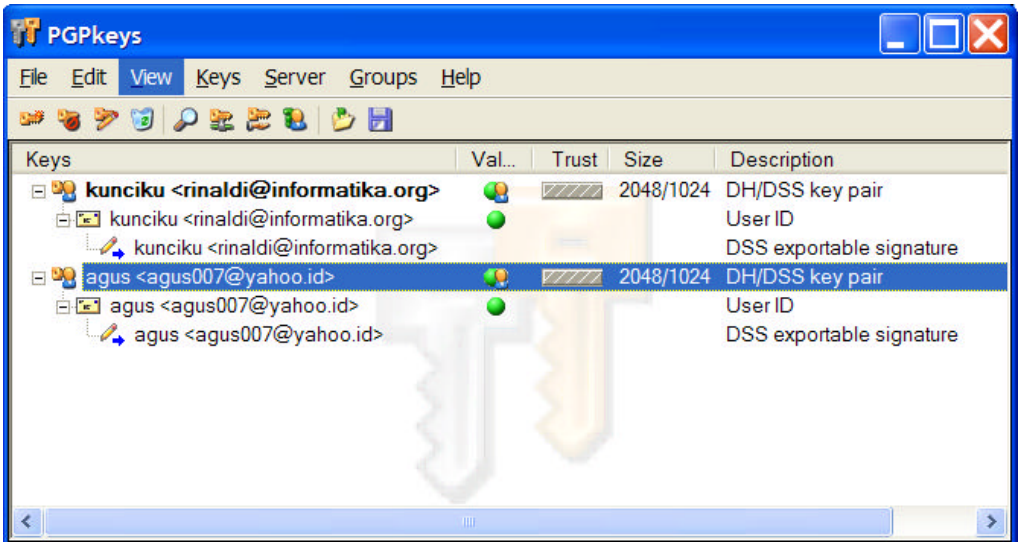
- Aktifkan *PGPkeys* sehingga muncul tampilan berikut:



- Pilih

  Keys → *New Key*

  selanjutnya akan ditampilkan *wizard* untuk membangkitkan pasangan kunci. Isilah beberapa isian yang disediakan. Contoh hasil pembangkitan beberapa pasangan kunci :



- Untuk melihat kunci publik, atau memberi kunci publik ke orang lain, ekspor kunci tersebut ke arsip (ekstensi arsip adalah `.asc`).

  Contoh kunci publik:

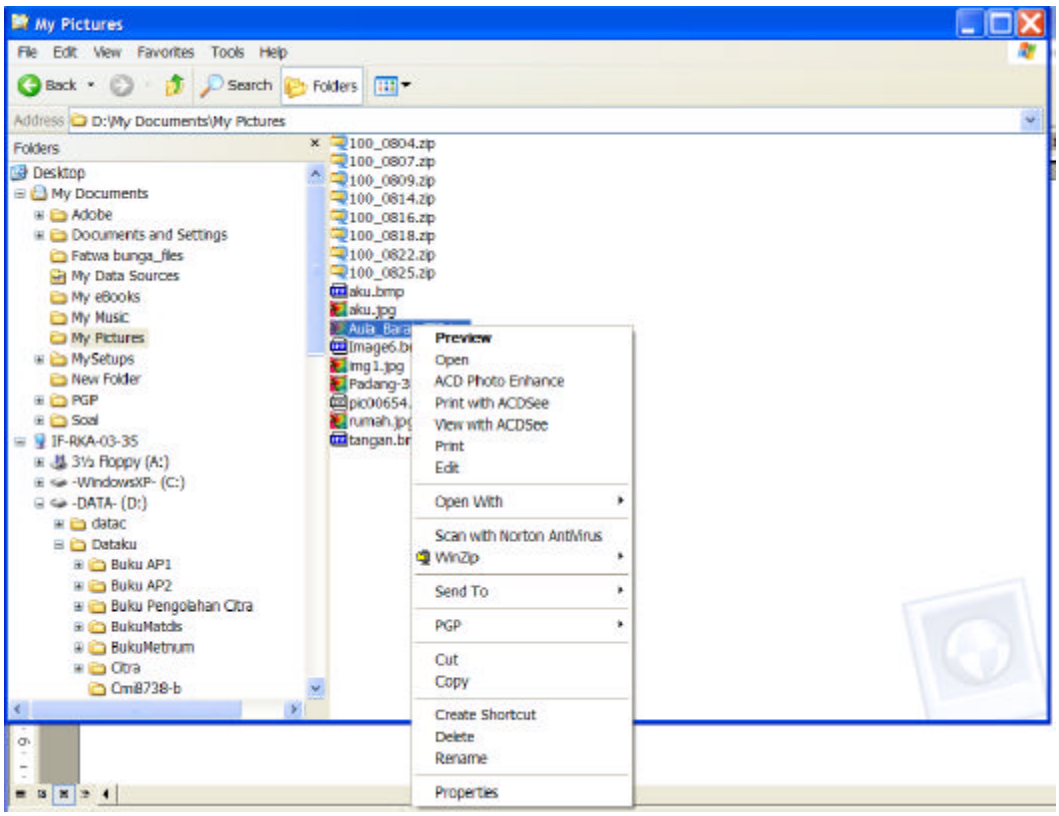
```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP 8.1 - not licensed for commercial use: www.pgp.com

mQGiBEHDmkIRBAD3p8b3phfk0FftdA2mRqEHLcg/iwF6VzcSde5ng9v86PeEB9xK
BMR9EiUjRdo1Us9YVi8awZ3iZG5EhX5sI/tbuBWJILCARhQzrn7Ww+sAuKrEPg4s
ggZtxYO1FsAbWhB/nKNqgDnYxxY3RbvOYlzH65Bk2xosR3H/YkHqc0L/SQCg/w0S
sh3fkWhymqao7rTJb4B/w2kD+QGWQlz81EkEbQaj3XeE4MdNMDjefKzxp/gP6I7Z
koJyiQxiIm1z4Q2R4iLniUX3hO7Vb9xre+J3s8D+rB0teJ70P7L2RNqK8QLVuqGh
lT1Yy4Okv5uuu0D4yTIOxB+vc3AlAoQwTVVSrKw5I8W7vaXvYBZd2m3w7ItDUjpP
uKhQA/9RjeQBq2QtBA2/7hLPP/NhSSfZ2C9A7rbN0ur3rG7mP0HB+hVFELR7tpW8
Mq+wPHP59qFlGWZpjR0E7svN96pLmQPW5x13Lc8Ip0D1z99o66vZ+UllRFNQR0kk
QO+V3kEIggWFpwOHi/Rz+vCVzrXRpR2CRSPinjxRswoC3vfnL7Qha3VuY2lrdSA8
cmluYWxkaUBpbmZvcm1hdGlrYS5vcmc+iQBdBBARAgAdBQJBw5pCBwsJCAcDAgoC
GQEFGwMAAAAFHgEAAAAACgkQFOUfEytY5dS/SwCg+wXNaoaVjnnMMsqUbF888cJF
WO0AoJWFiXP5yWfWqYSRXfqTAqYv0HGsuQINBEHDmkMQCAD2Qle3CH8IF3Kiutap
QvMF6PlTETlPtvFuuUs4INoBp1ajFOmPQFXz0AfGy0OplK33TGSGSfgMg71l6RfU
odNQ+PVZX9x2Uk89PY3bzpnhV5JZzf24rnRPxfx2vIPFRzBhznzJZv8V+bv9kV7H
AarTW56NoKVyOtQa8L9GAFgr5fSI/VhOSdvNILSd5JEHNmszbDgNRR0PfIizHHxb
LY7288kjwEPwpVsYjY67VYy4XTjTNP18F1dDox0YbN4zISy1Kv884bEpQBgRjXyE
pwpy1obEAxnIByl6ypUM2Zafq9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6q6Jew1Xp
Mgs7AAICCADgUJgMdoFAmVvW3rwTmXtx78O6st/vPoUqMHlGcQeAJ6jFZNj9YzE6
Q5Z3rB6Prv41oTyGBTm/iHFKhluhuA5Zce66KpODlXEWWKesBETkdqMClrmXdBQY
Pff1+NDSpTFfEiJ8YtTz9h3qETCUKEe5u/9oh1e4xCPhjvDTbZKCLV9k7mFyw4Ma
hdRY3moH/3UkDQJD1pD0xdr60d52vMoW71tY2TQ/2tAEbVrRncp9dVXAoqSsOr+J
qRvc0KhP5/5P2u50BobzRJ1nGrlGfRhbI0gR18bZtNLfLDXpHGuMwreYeDxcnUUG
z1gmHb0Xbe/ymsBQoRPqPCdiYM0HDF//iQBMBBgRAgAMBQJBw5pDBRsMAAAAAAoJ
EBTlHxMrWOXUOlcAn3ehXWUDWkHSTW7q6gHpK44VMmpBAKClarHaLAUahiGhHNt2
AMaQYklN/Q==
=GLD1
-----END PGP PUBLIC KEY BLOCK---
```

- Kunci publik orang lain dapat dimasukkan ke dalam daftar kunci dengan cara memilih menu *Keys* → *Import.*

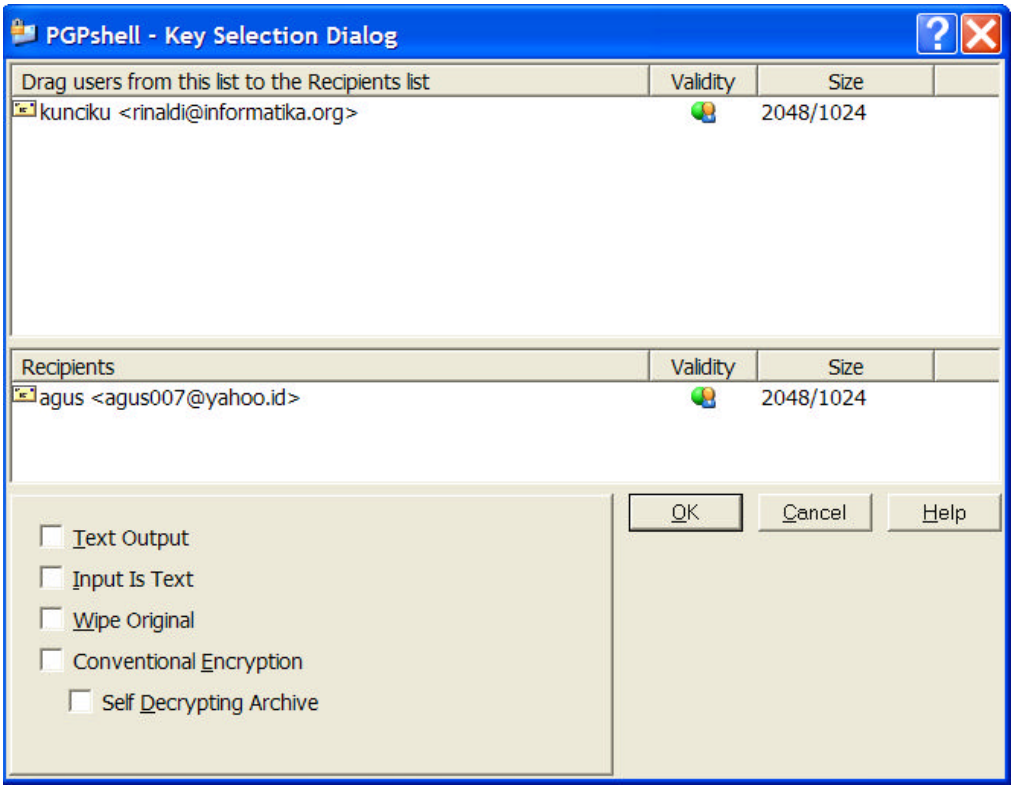
Mengenkripsi Arsip

(a) Mengenkripsi arsip yang akan dikirim

- Ada dua cara mengenkripsi arsip (*file*) yang akan dikirim
  1. Melalui *Windows Explorer*
     Pilih arsip yang akan dienkripsi, lalu klik kanan tetikus, dan pilih menu *PGP*.



Dari menu PGP, pilih *Encrypt*, sehingga muncul tampilan berikut:



Pilih *Recipents*(orang yang kita kirimi arsip), selanjutnya tekan *OK*. Arsip akan dienkripsi dengan kunci publik penerima.

Hasil enkripsi dapat dipilih untuk disimpan sebagai arsip teks (*Text Output* √) – dengan ektesni nama arsip `.asc` – atau sebagai arsip biner – dengan ekstensi nama arsip `.pgp`.

Contoh arsip `aula-barat.jpg` sebelum dienkripsi:

Arsip `aula-barat.jpg.asc` (*Text output*):

```
-----BEGIN PGP MESSAGE-----
Version: PGP 8.1 - not licensed for commercial use: www.pgp.com

qANQR1DBwU4Dj04oYGwvm0sQCADmCsgqnbpme3mDmoLCAp0lbHmeCtTR9dVQ0CZT
0P43mMkj7iR3F8pEaGOzeAsgj4YXy1iaYkZeUkujWuHeEbtZcjID9cRdOFy/jgNk
zvlSccANEhUcNRBqKCiZM/mesfUTOecXdMJ17T2ApsaMqjVxbHtDDnHGWlsX1Q5U
jysmIyZ9XcaCq9kJYGSK5u5yKYHAe9SQK8/dezYgfLKql//eNYK8ycwSYCzaNNBa
/ZdhAoydKuVCRaQmOROJJXYcI9A2MncjSrYXnELnJVykFM9sjn5xPptrLpybpJ4i
OyxzMzpaDCKAJ5De0lJDKWqGTY+FKwHnk9xmghn/2d5gYu+zB/4+69+UqrOW3jXU
Fna42wWPyWsZ8T5eJ3KvS6OZdIoP5NtT27iSwpJQWWm9X0BLEtv9pyddUWld7QE8
fy3wv+IgGMKLkh7ZgfgwckpBGao06LOHsyLk2YB2Jh25HSiszqqr0N112dhJbmd/
63R8WDtcfgNEh3IirAR6atZsOUPUJKzv6RC7ulQqxZHc3vL/dl+ElpujFOs50qs+
DSnofXvYPZYzrcCi0hS9IjRmPAQo1MwBgmNNpI1Tnp8A7gg09auSQEH3F4DEyEUB
t5s4SaEcJ0cZlt3Ps/HX1z1PTvkWJQbuJDsTVaZbL0KQosEt68EZWagMDGd25kBj
2fALpzoC0uwBkUOpPXtnTYvw/jafWXjtfogXeqHlN0cOu2mNb64S85RgLbv3q6V4
a0OSaLE9qNpSOnyqAibTxz1sKlChPZWwfu/ORkFdsgu4kFzLwXDCSEktWz2a9xJC
Uy5ybGALmRvPxQNMhX8b2JPb3fxtnBDrqRsMXnlgjXMR+8nmk57f6MuSEebyYscg
8PNKkTqKOXOMBc6ZPNOh4ZxnPzHZrsgHvLgbdudie3p9uIFVTEoi5V2qj5O5/oy+
kYgJg+ix+R28zA33iFIFhN7PTfWwuFIlgOpk+7cLr+KDl8TKnzIgfrKdzXYNjErl
jrtZ8ws5JMPDwAgQ6677dUq1i1g3P2zXJwmgsdF9A4uC2JvpJeCbKu3Sy6ZXP4CX

... (deleted, because too long)

-----END PGP MESSAGE-----
```

Untuk mendekripsi arsip, klik arsip yang terenkripsi, selanjutnya klik kanan tetikus, dan pilih *Decrypt*. Program *PGP* meminta anda memasukkan *passphrase* untuk kunci privat (harus sama dengan passphrase yang diisikan pada waktu pembangkitan pasangan kunci).

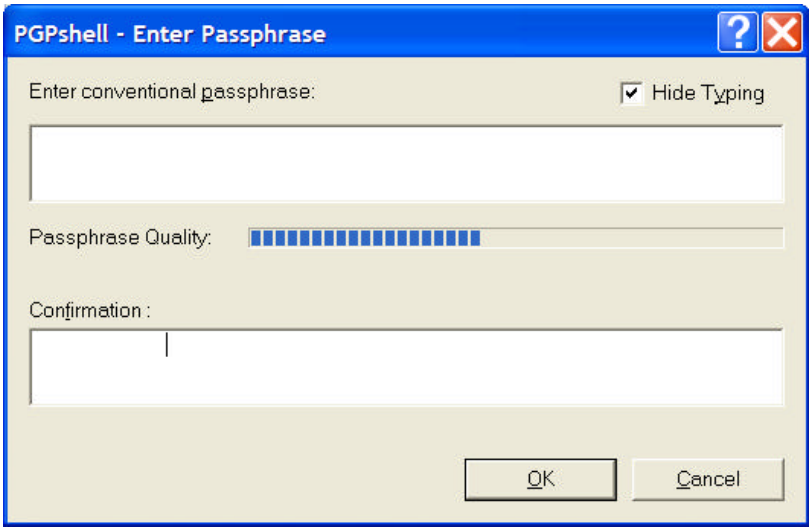
Jika *passphrase* benar, maka arsip akan didekripsi dengan menggunakan kunci privat yang berkoresponden dengan kunci publiknya.

2. Melalui program *PGPmail*
   Aktifkan program *PGPmail*, sehingga muncul tampilan berikut:

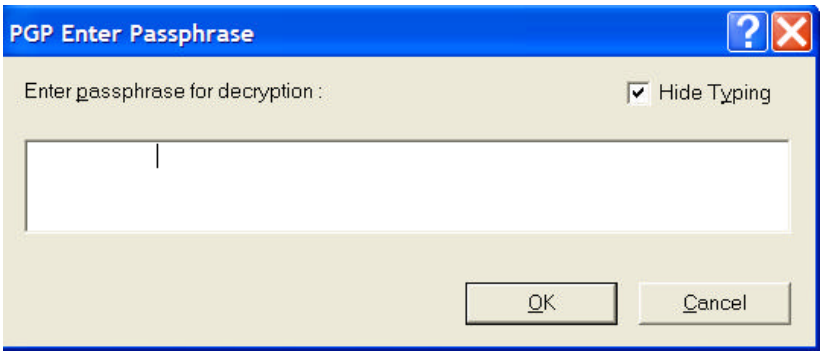
Pilih ikon surat+gembok, dan selanjutnya tahapan enkripsi sama seperti cara pertama.

(b) Mengenkripsi arsip dengan algoritma simetri

Jika opsi *Conventional Encryption* dipilih, maka arsip akan dienkripsi dengan algoritma simetri. Di sini kunci simetri dibangkitkan dari *passphrase* yang diketikkan oleh pengguna:



Untuk mendekripsi arsip, klik arsip tersebut, lalu klik kanan tetikus, pilih menu PGP, lalu pilih *Decrypt*:



Ketikkan *passphrase* yang sama seperti waktu enkripsi. Hasil dekripsi dapat disimpan dengan nama lain.

Contoh enkripsi arsip `bandung.txt`.

(i) Arsip `bandung.txt` sebelum dienkripsi

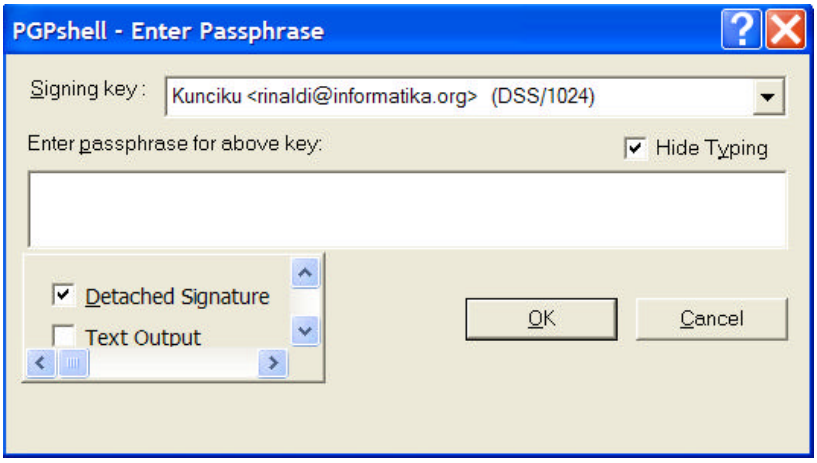
```
Pada bulan Oktober 2004 ini, suhu udara kota Bandung terasa lebih
panas dari hari-hari biasanya. Menurut laporan Dinas Meteorologi
Kota Bandung, suhu tertinggi kota Bandung adalah 32  derajat
Celcius pada Hari Rabu, 17 Oktober yang lalu. Suhu terseut sudah
menyamai suhu kota Jakarta pada hari-hari biasa. Menurut Kepala
Dinas Meteorologi, peningkatan suhu tersebut terjadi karena
posisi bumi sekarang ini lebih dekat ke matahari daripada hari-
hari biasa.

Sebutan Bandung sebagai kota sejuk dan dingin mungkin tidak lama
lagi akan tinggal kenangan. Disamping karena faktor alam, jumlah
penduduk yang padat, polusi dari pabrik di sekita Bandung, asap
knalpot kendaraan, ikut menambah kenaikan suhu udara kota.
```

(ii) Arsip `bandung.txt.asc` setelah dienkripsi

```
-----BEGIN PGP MESSAGE-----
Version: PGP 8.1 - not licensed for commercial use: www.pgp.com

qANQR1DDDQQJAwKRt3ROh/zvkWDSwQ8BSzulggHt+bRY/Ma3X/0iEnhSh4xs/q14
m7KjXHi0c7EoQnGvfhZiEA5lzASdqVpUkdR0bRI4F/Vn8D4RWqmmca1qm7KskqRo
+wenFvfQYGBeagM1WOWTrWBKJAPdVG88oCcOE97Bf5YC+Z5f57PAjp5CgrHXj09N
4ElNR2EHohBzhOEAGYIzzzxNBS4kUD8XdThUBqlKSqRO8ZxZora20qYc1oHe79TC
+4T5BG+B+AUCQsTGx8zL2GwoCF/r1ed2SldTJou952gLMpMa6BPvn37VFs1s7EUG
zXa56peaq+bPMYZYW8J69OeoIdDjX6avrbsVOpk07mbOBQl2XbpteKFBz+fjldYE
8MrWblaGL26Q0feoHhckwVsa5uiUrFkjG6mQQubddibenWFMkp64jhDtgyXLJUUi
onFyQsQCFAxQUrcDRw4/0ggmq+qgBtSD4mg2AhhsQCleNbAqWO72yVJcX73eHZ1M
b8NTCMwrsKfMxEs35tY1OU1/SKvi4DlqOgNb5ye0oTzKcpzgJBuk82yJOJFnXkaW
NUysrLONu0kgC/UI3Ma4mtBOAxk4TjNfmEZWmHcI0cDTQ2/FFto8gNxP54qveFB0
IRj8qcpf
=6oEr
-----END PGP MESSAGE-----
```

Memberi Tanda-tangan

- Pilih arsip yang akan ditandatangani, lalu klik kanan tetikus, dan pilih menu *PGP*, kemudian pilih *Sign.*



Arsip akan ditandatangani dengan kunci privat. Masukkan *passphrase* untuk kunci privat. Tandatangan dapat disimpan terpisah menjadi arsip khusus (ekstensi `.sig`) atau digabung menjadi satu dengan arsip (*Detached Signature* √).

Contoh penandatanganan arsip `bandung.txt`.

(i) Arsip `bandung.txt` sebelum ditandatangani

```
Pada bulan Oktober 2004 ini, suhu udara kota Bandung terasa lebih
panas dari hari-hari biasanya. Menurut laporan Dinas Meteorologi
Kota Bandung, suhu tertinggi kota Bandung adalah 32  derajat
Celcius pada Hari Rabu, 17 Oktober yang lalu. Suhu terseut sudah
menyamai suhu kota Jakarta pada hari-hari biasa. Menurut Kepala
Dinas Meteorologi, peningkatan suhu tersebut terjadi karena
posisi bumi sekarang ini lebih dekat ke matahari daripada hari-
hari biasa.

Sebutan Bandung sebagai kota sejuk dan dingin mungkin tidak lama
lagi akan tinggal kenangan. Disamping karena faktor alam, jumlah
penduduk yang padat, polusi dari pabrik di sekita Bandung, asap
knalpot kendaraan, ikut menambah kenaikan suhu udara kota.
```

(ii) Arsip `bandung.txt` setelah ditandatangani

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Pada bulan Oktober 2004 ini, suhu udara kota Bandung terasa lebih
panas dari hari-hari biasanya. Menurut laporan Dinas Meteorologi
Kota Bandung, suhu tertinggi kota Bandung adalah 32  derajat
Celcius pada Hari Rabu, 17 Oktober yang lalu. Suhu terseut sudah
menyamai suhu kota Jakarta pada
hari-hari biasa. Menurut Kepala Dinas Meteorologi, peningkatan
suhu tersebut terjadi karena posisi bumi sekarang ini lebih dekat
ke matahari daripada hari-hari biasa.

Sebutan Bandung sebagai kota sejuk dan dingin mungkin tidak lama
lagi akan tinggal kenangan. Disamping karena faktor alam, jumlah
penduduk yang padat, polusi dari pabrik di sekita Bandung, asap
knalpot kendaraan, ikut menambah kenaikan suhu udara kota.

-----BEGIN PGP SIGNATURE-----
Version: PGP 8.1 - not licensed for commercial use: www.pgp.com

iQA/AwUBQcOWFJTpyRekJ1FcEQI8xgCaAmBME/O/lIOfdvZZUfnHcgdhHPAAoPxJ
WPpSilIHIl63h3/iHoB9fIc2
=eKHn
-----END PGP SIGNATURE-----
```