

Pair Key (Private and Public) Encryption
Sistem Operasi Genap 2016/2017
Fakultas Ilmu Komputer – Universitas Indonesia
Oleh : Ibad Rahadian Saladdin (1406623695)

- **Pendahuluan**

Pada dunia IT, transmisi file dari satu orang ke orang lainnya merupakan hal yang biasa. File ini nantinya dapat digunakan untuk banyak keperluan, seperti kolaborasi untuk mengembangkan suatu program. Tetapi jika file yang ingin di-transmisikan jatuh ke tangan orang yang salah dan tidak bertanggung jawab, maka dapat menimbulkan banyak masalah seperti plagiarisme atau kebocoran data karena, terkadang dari file-file tersebut juga memiliki informasi-informasi yang sensitif yang tidak dapat dilihat oleh semua orang.

Maka dari itu, perlu dibuat suatu pengaman agar informasi-informasi yang ada pada sebuah *file* tidak mudah hilang ataupun dicuri oleh orang lain. Cara yang mudah adalah dengan OpenPGP Pair Key, dengan menggunakan *public* dan *private key*. *Public key*, dimana kunci ini tersebar dan dapat diakses oleh semua orang. Sedangkan, *private key* hanya dapat diakses oleh pembuat *key* tersebut. Enkripsi jenis ini memungkinkan kita secara langsung menentukan tujuan siapa yang dapat membuka enkripsi ini sehingga pihak-pihak lain yang tidak diberikan akses tidak dapat melihat file tersebut.

Berikut adalah panduan untuk melakukan enkripsi OpenPGP dengan menggunakan Windows (Kleopatra) dan juga Linux.

Pada panduan ini, digunakan **GitHub** sebagai repository untuk memudahkan transfer file antar pengguna. Disarankan bagi pengguna Windows untuk meng-*install tools* Kleopatra dan GitHub Client for Windows. Sedangkan bagi pengguna Linux untuk meng-*install* git dengan melakukan ini pada terminal Anda :

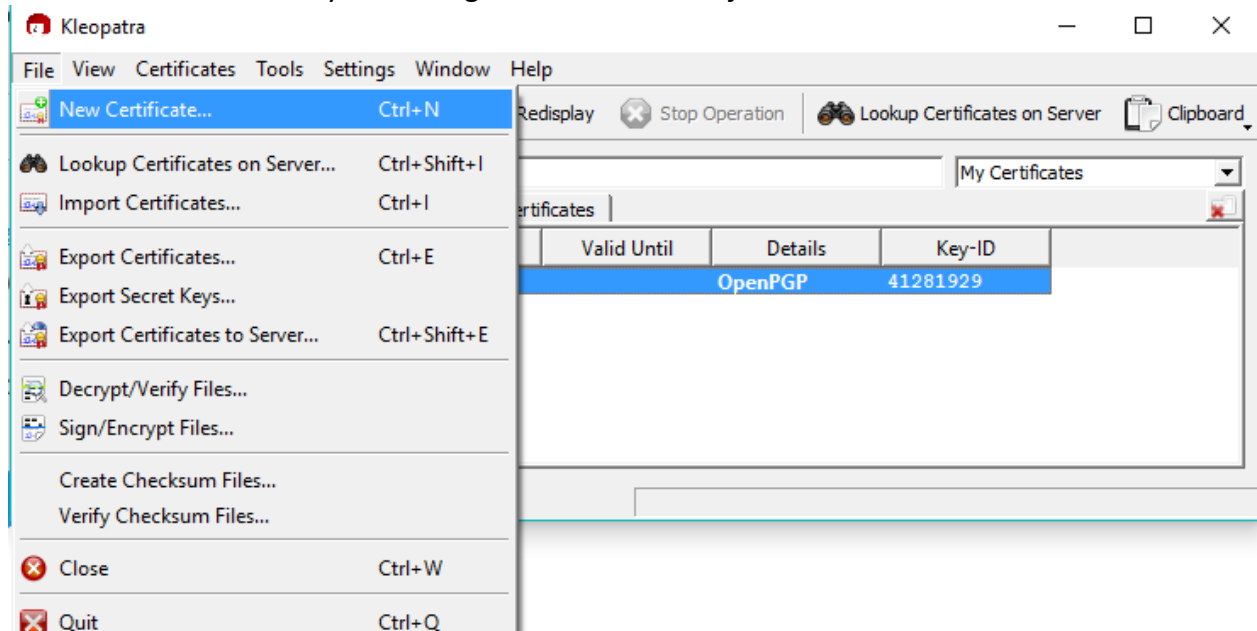
```
sudo apt-get update  
sudo apt-get install git
```

- **Membuat Key**

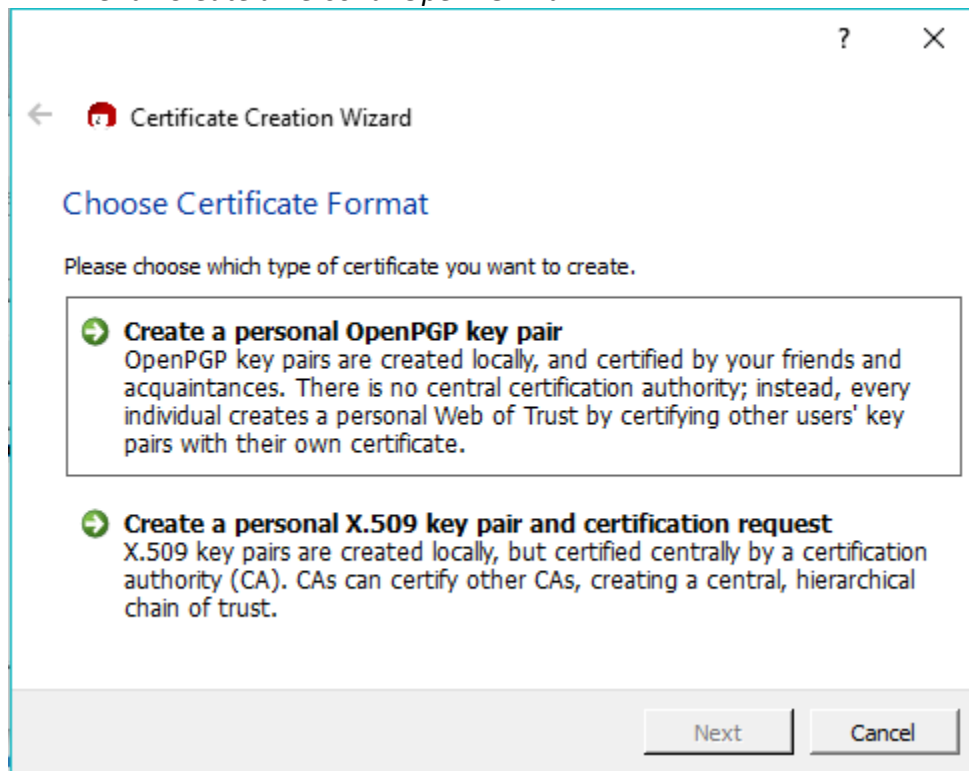
1. **Windows (Menggunakan Kleopatra)**

Tips : Silahkan unduh pada halaman berikut <https://www.gpg4win.org/features.html>

- a. Buka menu membuat key baru dengan *File -> New Certificate*.



- b. Pilih menu *"Create a Personal OpenPGP Pair"*



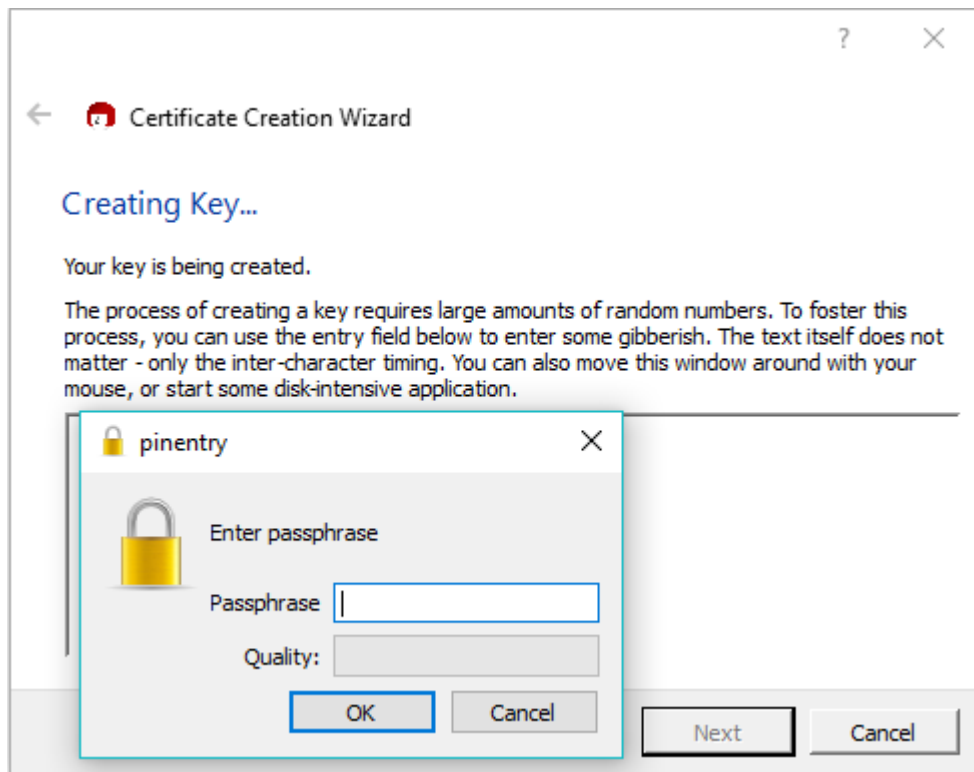
- c. Masukkan detail data diri Anda.

The screenshot shows the 'Certificate Creation Wizard' window with the title bar containing a question mark and a close button. The window has a back arrow and a red icon next to the title 'Certificate Creation Wizard'. The main heading is 'Enter Details'. Below it, a message says: 'Please enter your personal details below. If you want more control over the certificate parameters, click on the Advanced Settings button.' There are three input fields: 'Name:' with the value 'Ibad Rahadian Saladdin' (marked as required), 'EMail:' with the value 'ibad.rahadian@ui.ac.id' (marked as required), and 'Comment:' with the value 'New Key Test' (marked as optional). Below these fields, the summary text reads: 'Ibad Rahadian Saladdin (New Key Test) <ibad.rahadian@ui.ac.id>'. To the right of this text is a button labeled 'Advanced Settings...'. At the bottom right of the window are two buttons: 'Next' and 'Cancel'.

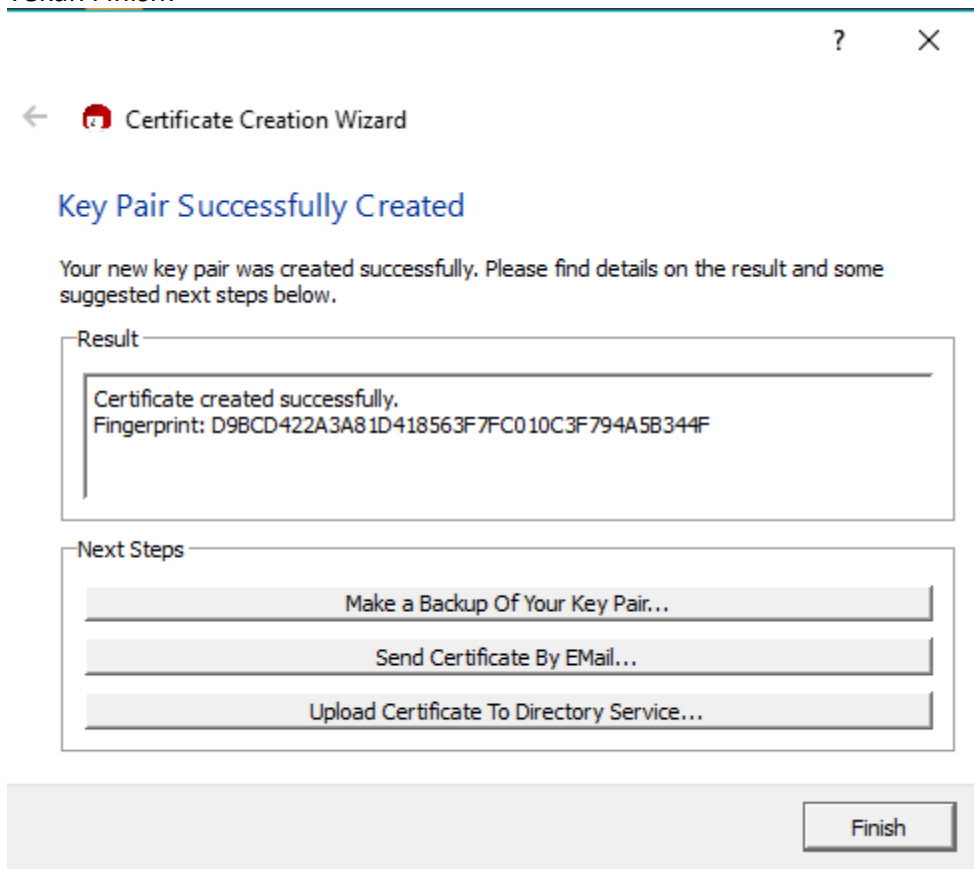
- d. Silahkan review data yang sudah Anda masukkan, tekan *Next*.

The screenshot shows the 'Certificate Creation Wizard' window with the title bar containing a question mark and a close button. The window has a back arrow and a red icon next to the title 'Certificate Creation Wizard'. The main heading is 'Review Certificate Parameters'. Below it, a message says: 'Please review the certificate parameters before proceeding to create the certificate.' There is a text box containing the following information: 'Name: Ibad Rahadian Saladdin', 'Email Address: ibad.rahadian@ui.ac.id', and 'Comment: New Key Test'. Below the text box is a checkbox labeled 'Show all details'. At the bottom right of the window are two buttons: 'Create Key' and 'Cancel'.

- e. Masukkan kata kunci Anda, jangan beritahukan kepada orang lain sebanyak dua kali.



f. Tekan *Finish*.



2. Linux

Tips : Semua operasi di Linux pada tutorial ini menggunakan *Command Line*. Jadi harus menggunakan Terminal.

- a. Inisiasi pembuatan key baru dengan `gpg --gen-key`
- b. Pilih menu “RSA and RSA (default)” dengan menekan ‘1’ lalu Enter.
- c. Masukkan jumlah bit yang diinginkan dengan ‘2048’ (default) atau sesuai dengan yang Anda inginkan.
- d. Masukkan berapa lama key ini berlaku, e.g ‘6m’ untuk waktu 6 bulan atau sesuai dengan yang Anda inginkan.
- e. Masukkan identitas anda (Nama, e-mail, dan *Comment*).
- f. Masukkan *passphrase* Anda.
- g. Cek apakah key sudah masuk dengan `gpg --list-keys`.
- h. Jika sudah ada, berarti key sudah selesai dibuat.

Dibawah ini terdapat ilustrasi yang telah dijalankan pada sistem Ubuntu.

Tips Menyimpan *Passphrase* agar tidak lupa

Passphrase pada enkripsi jenis ini penting untuk diingat, karena tidak ada mode lupa kata sandi agar tidak mudah untuk ditembus oleh orang lain. Langkah antistipatif yang dapat diambil adalah dengan membuat *hint* pada kata sandi atau cara-cara mudah agar kata sandi selalu diingat.

1. Tulislah *hint* kata sandi pada *notes* Anda di tempat yang aman, saya menyarankan menggunakan layanan Google Keep yang dapat diakses dari *mobile* dan web.
2. Buatlah *hint* yang unik, yang sangat internal sehingga orang-orang lain tidak mengetahui maksud yang Anda tulis.
3. Jangan tuliskan kata sandi Anda secara gamblang pada *notes* tersebut.

```

ibaadee@ibaadee-VirtualBox:~/Documents$ gpg --gen-key
gpg (GnuPG) 1.4.20; Copyright (C) 2015 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 2048
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) 6m
Key expires at Jun 07 Jul 2017 10:39:18 WIB
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Misaki
Email address: misaki@example.com
Comment: MISAKI
You selected this USER-ID:
    "Misaki (MISAKI) <misaki@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

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.
.....+++++
....+++++
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.
..+++++
..+++++
gpg: key 743AF538 marked as ultimately trusted
public and secret key created and signed.

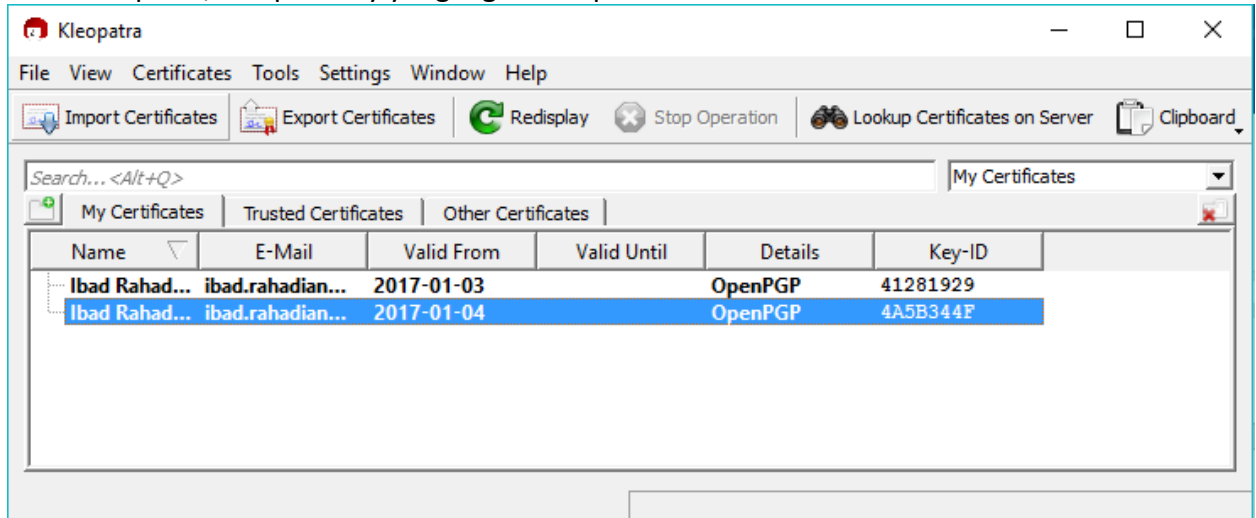
```

- **Export Key**

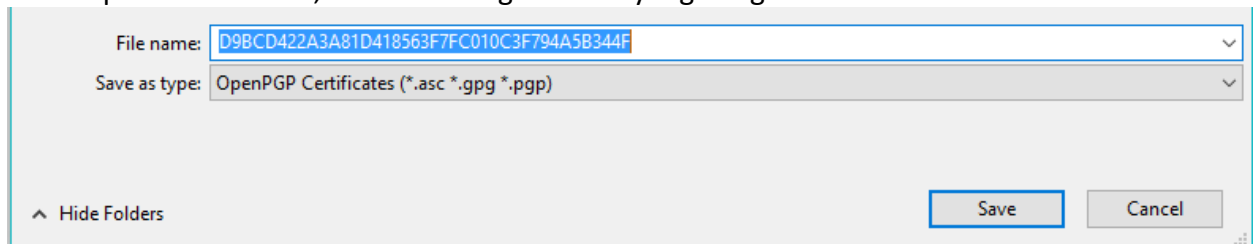
Tips : Export Key ini digunakan untuk pengguna lain dapat memberikan enkripsi filenya dan dapat dibuka dengan menggunakan key yang Anda berikan. Sertakan key Anda pada setiap pengiriman agar pengguna lain dapat memberikan akses kepada Anda.

- 1. Windows (Menggunakan Kleopatra)**

- a. Buka Kleopatra, lalu pilih key yang ingin di-export.



- b. Klik "Export Certificate", lalu save dengan nama yang diinginkan.



- 2. Linux**

- a. Arahkan ke folder tempat key ini nantinya akan disimpan.
- b. Ketik di terminal `gpg --armor --export youremail@example.com > mykey.asc`. "mykey.asc" merupakan nama file yang menyimpan key Anda dan dapat diberi nama sesuai dengan keinginan Anda.
- c. Key sudah siap untuk digunakan.

- **Import Key**

- 1. Windows**

- a. Buka Kleopatra, lalu pilih menu “Import Certificate”
- b. Arahkan ke *directory* penyimpanan key tersebut dengan ekstensi .txt, .asc, .gpg, .pgp
- c. Lihat di Kleopatra apakah key sudah masuk.
- d. Jika sudah ada, berarti key sudah berhasil di-import.

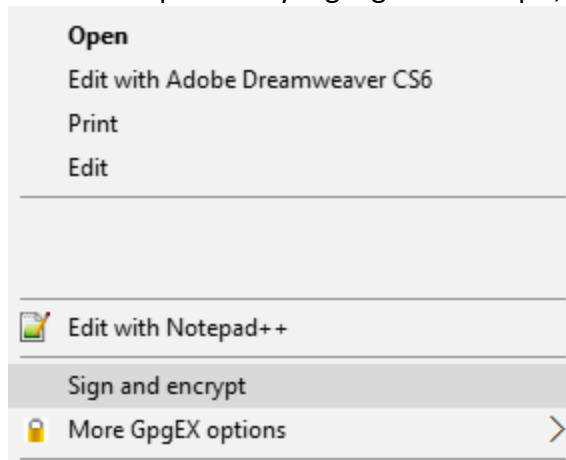
- 2. Linux**

- a. Arahkan kepada *directory* dimana key Anda disimpan.
- b. Jalankan `gpg --import mykey.asc`. “mykey.asc” dapat diisi dengan nama key Anda.
- c. Cek dengan `gpg -list-keys` untuk mengecek keberadaan key yang baru saja di-import.
- d. Jika sudah ada, berarti key sudah berhasil di-import.

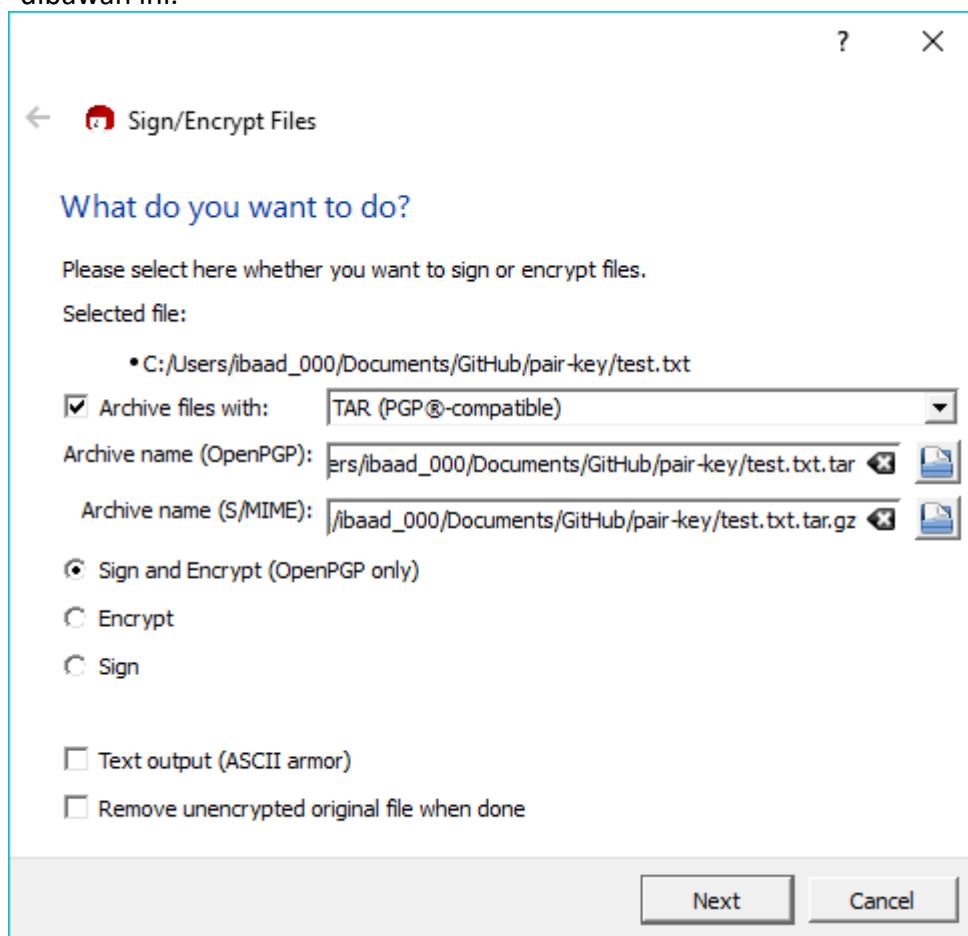
- Melakukan Enkripsi

1. Windows

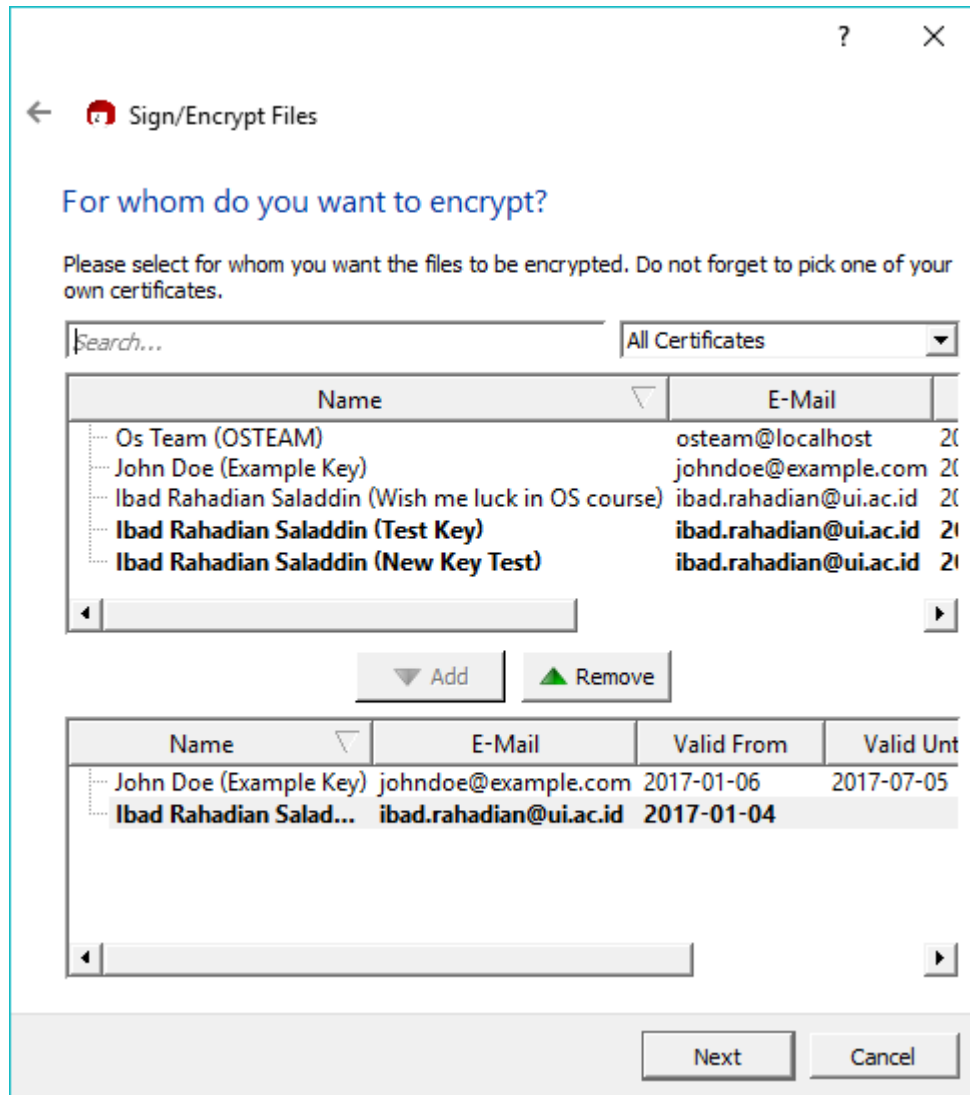
- a. Klik kanan pada file yang ingin di-enkripsi, lalu tekan *Sign and encrypt*



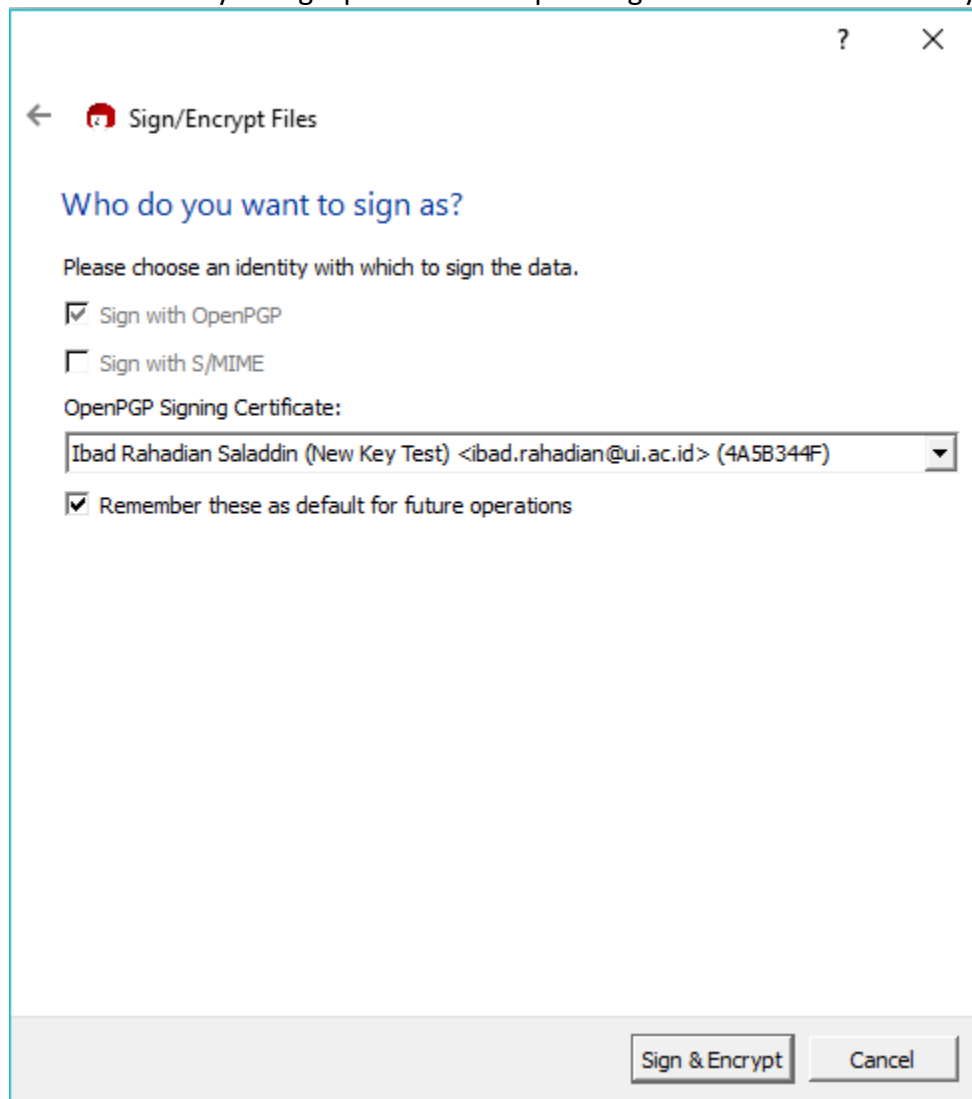
- b. Isi tujuan dan *Radio Button* “Sign and Encrypt (Open PGP only)” seperti pada gambar dibawah ini.



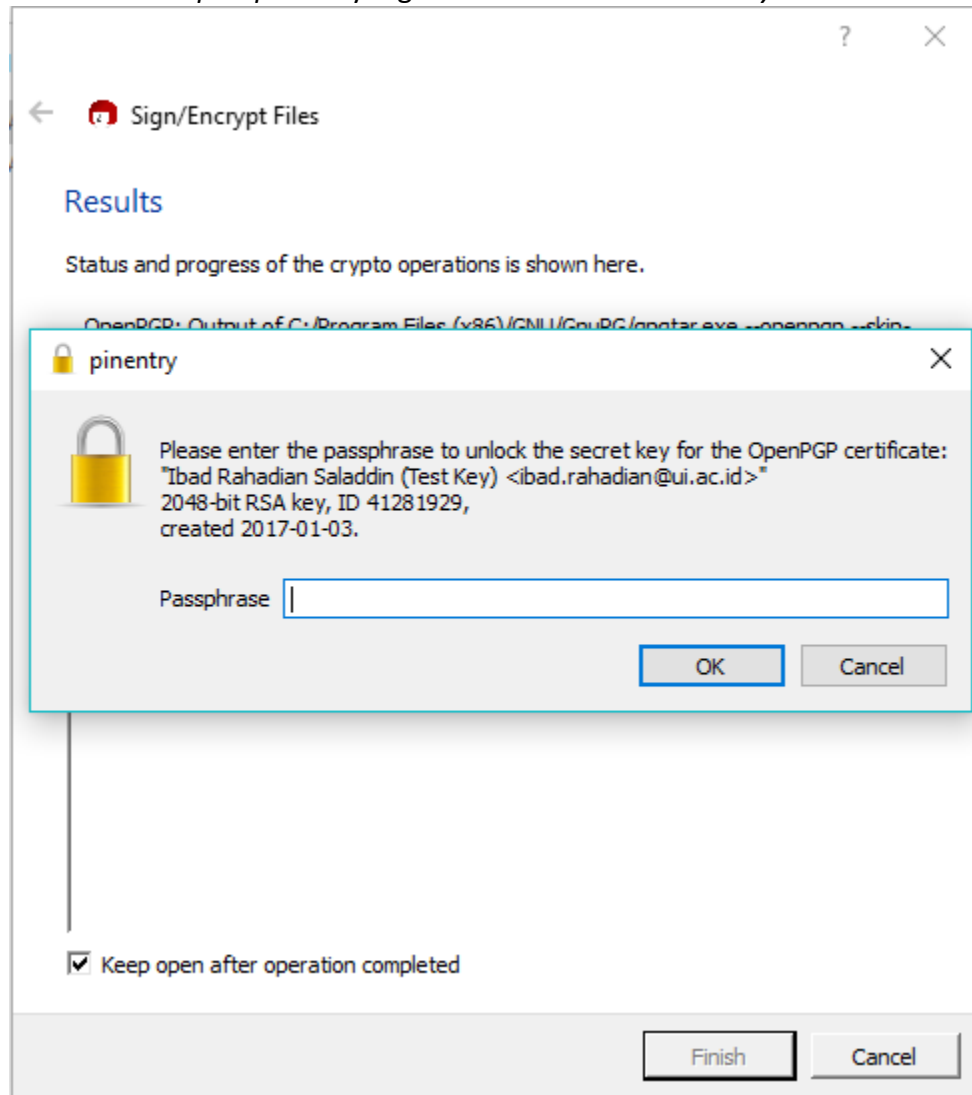
- c. Pilih *public key* user yang ingin dituju (termasuk diri sendiri) dengan me-klik tombol *Add*.



d. Masukkan key sebagai pembuat enkripsi dengan memilih salah satu key.



e. Masukkan *passphrase* yang dibuat ketika membuat *key*.



f. Silahkan tekan *Finish*. *File* telah ter-enkripsi.

2. Linux

Tips : Melakukan sign dan encrypt pada linux.

- Arahkan menuju folder dengan file yang ingin di-enkripsi.
- Buat kode hash dengan menggunakan sha1sum dengan menjalankan `sha1sum * > SHA1SUM`.
- Jalankan pengecekan sign dengan `sha1sum -c SHA1SUM`.

```
ibaadee@ibaadee-VirtualBox:~/Documents/nandemonaiya$ sha1sum * > SHA1SUM
ibaadee@ibaadee-VirtualBox:~/Documents/nandemonaiya$ sha1sum -c SHA1SUM
nandemonaiya.txt: OK
```

- Melakukan sign untuk file yang akan di-encrypt dengan `gpg --sign --armor --detach SHA1SUM`. Sign ini digunakan agar pengguna lain mengetahui kalo file ini hasil kerja Anda.

```
ibaadee@ibaadee-VirtualBox:~/Documents/nandemonaiya$ gpg --sign --armor --detach SHA1SUM

You need a passphrase to unlock the secret key for
user: "Misaki (MISAKI) <misaki@example.com>"
2048-bit RSA key, ID 743AF538, created 2017-01-08
```

- Lakukan verifikasi sign Anda dengan `gpg --verify SHA1SUM.asc`.

```
ibaadee@ibaadee-VirtualBox:~/Documents/nandemonaiya$ gpg --verify SHA1SUM.asc
gpg: assuming signed data in 'SHA1SUM'
gpg: Signature made Mon 08 Jan 2017 11:36:06 WIB using RSA key ID 743AF538
gpg: Good signature from "Misaki (MISAKI) <misaki@example.com>"
```

- Sekarang kita akan melakukan encrypt pada file yang kita inginkan dengan `gpg -o output nandemonaiya.txt.gpg --encrypt --recipient HYUGA --recipient MISAKI nandemonaiya.txt`. Setelah output, merupakan nama file target yang ingin Anda buat beserta dengan ekstensi .gpg yang merupakan file yang telah **ter-enkripsi**. Sedangkan recipient merupakan kepada siapa Anda akan memberikan akses file ini dan jangan lupa untuk **membuat recipient dengan key Anda sendiri**.

```
ibaadee@ibaadee-VirtualBox:~/Documents/nandemonaiya$ gpg --output nandemonaiya.txt.gpg --encrypt --recipient HYUGA --recipient MISAKI nandemonaiya.txt
gpg: 9904F44D: There is no assurance this key belongs to the named user

pub 2048R/9904F44D 2017-01-08 Hyuga (HYUGA) <hyuga@example.com>
Primary key fingerprint: E4FD A02C D9C0 234D D5E1 D030 B9A9 28FE 36D0 8430
Subkey fingerprint: DCC9 2932 12A1 2D49 265F 7475 E8A5 7ECA 9904 F44D

It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.

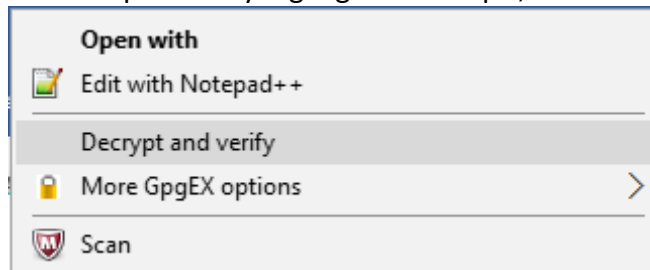
Use this key anyway? (y/N) y
```

- Jika file.gpg sudah ada, maka file sudah berhasil ter-enkripsi. Ketika Anda memberikan file ke tujuan Anda, pastikan Anda **menyertakan public key Anda**.

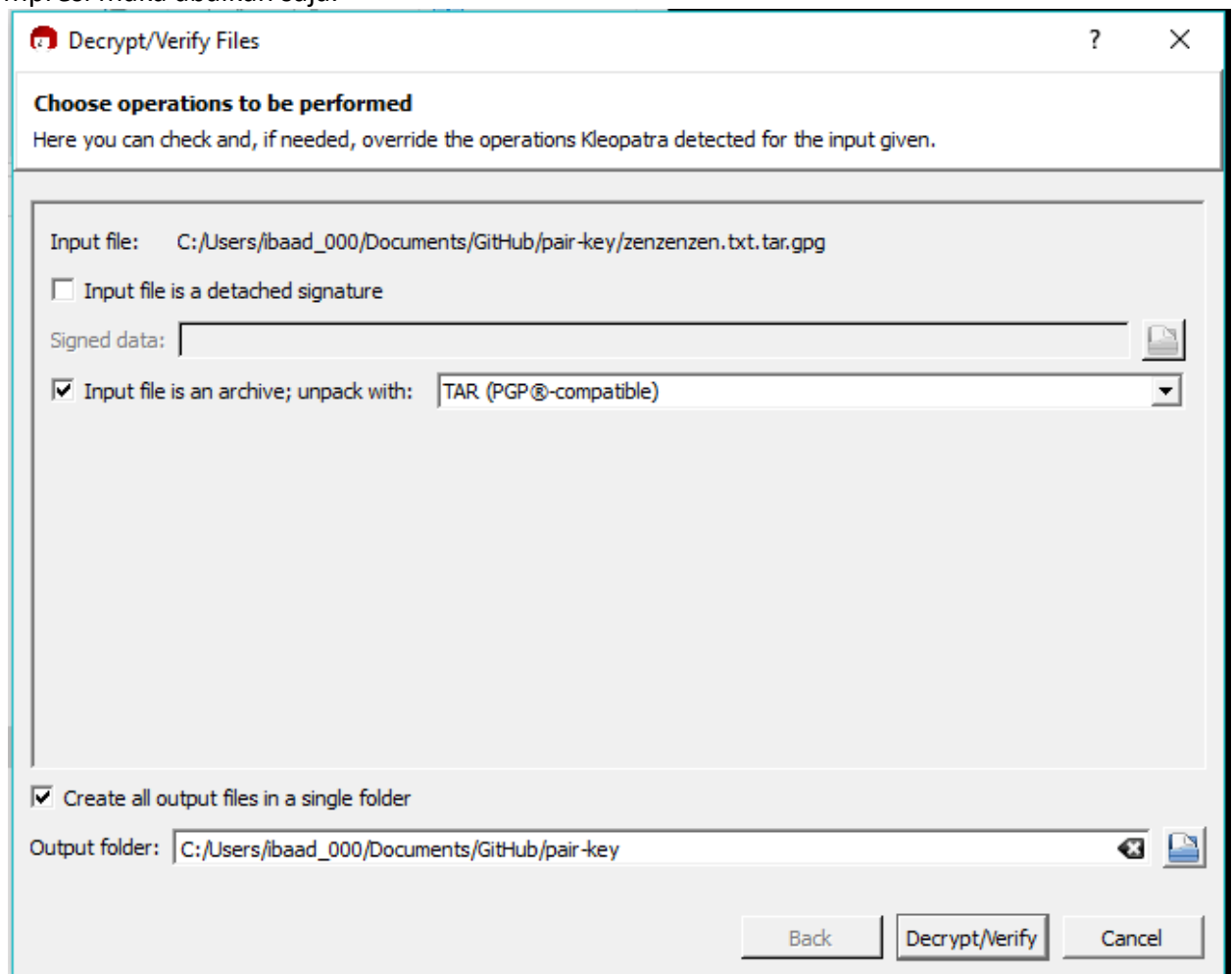
- **Melakukan Dekripsi**

- 1. Windows**

- a. Klik kanan pada file yang ingin di-dekripsi, lalu klik *Decrypt and verify*.



- b. Jika *file* yang ingin di-dekripsi menggunakan kompresi .tar, maka centang *check box* “*Input file is an archive; unpack with:*” dan pilih TAR(PGP®-compatible). Jika tidak menggunakan kompresi maka abaikan saja.



- c. Klik *Decrypt/Verify*.
- d. Masukkan *passphrase* yang dibuat pada awal pembuatan key.
- e. Klik *Finish*.

2. Linux

- a. Arahkan menuju folder yang mengandung file yang ingin anda dekripsi. Pastikan Anda sudah melakukan import **public key** dari **pembuat file tersebut**.
- b. Lakukan dekripsi pada file berekstensi .gpg dengan menjalankan `gpg --output zenzenzen.txt.tar --decrypt zenzenzen.txt.tar.gpg`. Setelah output merupakan nama target file setelah di-decrypt, disarankan menggunakan **nama yang sama** dengan hanya menghilangkan **ekstensi .gpg** saja.

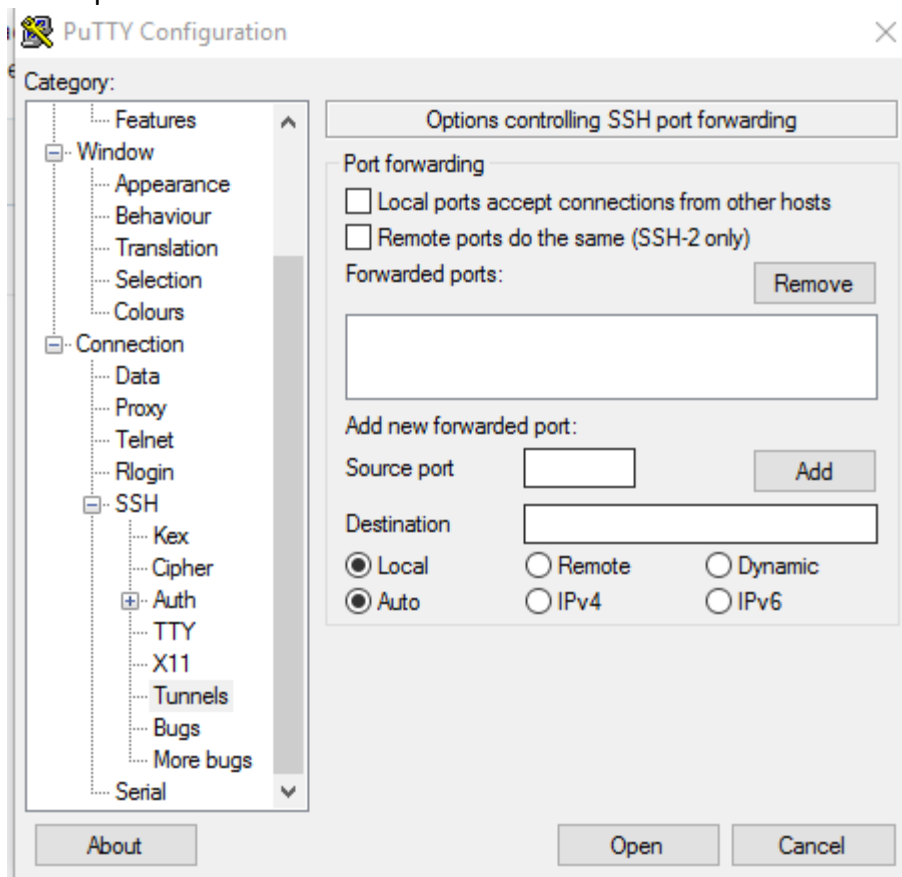
```
ibaadee@ibaadee-VirtualBox:~/Documents$ gpg --output zenzenzen.txt.tar --decrypt zenzenzen.txt.tar.gpg
You need a passphrase to unlock the secret key for
user: "Misaki (MISAKI) <misaki@example.com>"
2048-bit RSA key, ID DA1C18EA, created 2017-01-08 (main key ID 743AF538)

gpg: encrypted with 2048-bit RSA key, ID E0DE2AD0, created 2017-01-04
       "Ibad Rahadian Saladdin (New Key Test) <ibad.rahadian@ui.ac.id>"
gpg: encrypted with 2048-bit RSA key, ID DA1C18EA, created 2017-01-08
       "Misaki (MISAKI) <misaki@example.com>"
gpg: Signature made Mon 08 Jan 2017 11:09:13 WIB using RSA key ID 4A5B344F
gpg: Good signature from "Ibad Rahadian Saladdin (New Key Test) <ibad.rahadian@ui.ac.id>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:       There is no indication that the signature belongs to the owner.
Primary key fingerprint: D9BC D422 A3A8 1D41 8563 F7FC 010C 3F79 4A5B 344F
ibaadee@ibaadee-VirtualBox:~/Documents$ tar -xvf zenzenzen.txt.tar
zenzenzen.txt
ibaadee@ibaadee-VirtualBox:~/Documents$ vi zenzenzen.txt
```

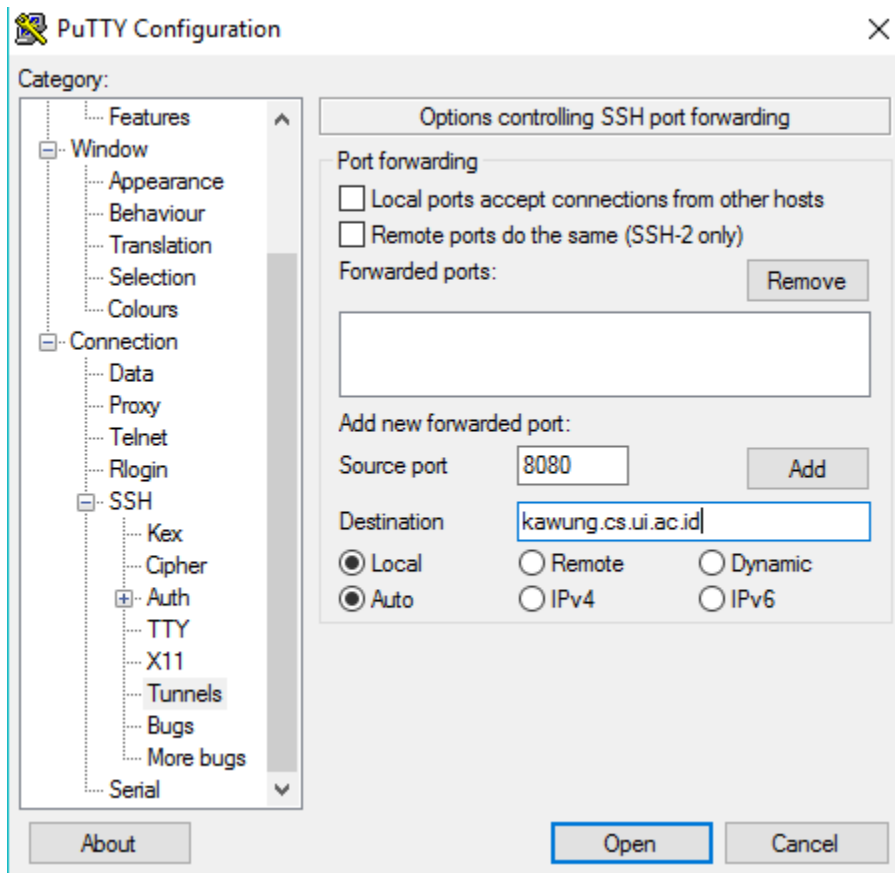
- c. Langkah ini hanya dijalankan jika file masih dikompresi, e.g ber-ekstensi .tar. Untuk melakukan dekompresi, maka dijalankan `tar -xvf zenzenzen.txt.tar`. Jika tidak memiliki kompresi, file dapat langsung dijalankan.
- d. Akan muncul hasil ekstrak dari kompresi. File ini sudah selesai di-*decrypt* dan dapat dijalankan.

- **Membuat Tunneling ke badak.cs.ui.ac.id**

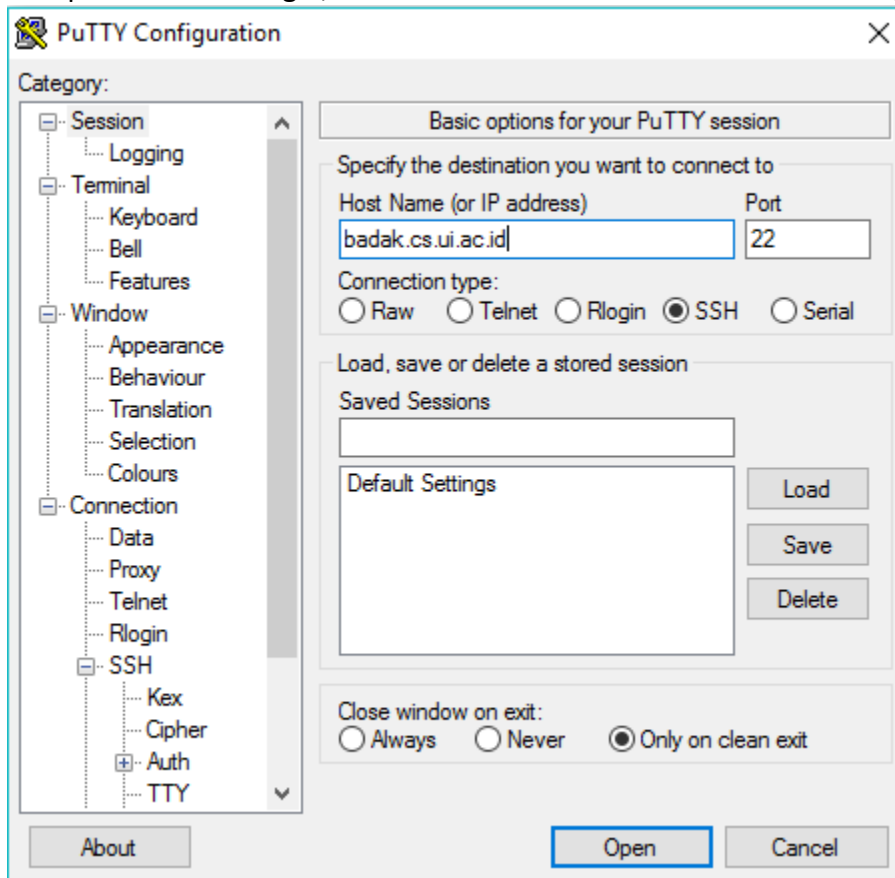
1. Buka aplikasi PuTTY



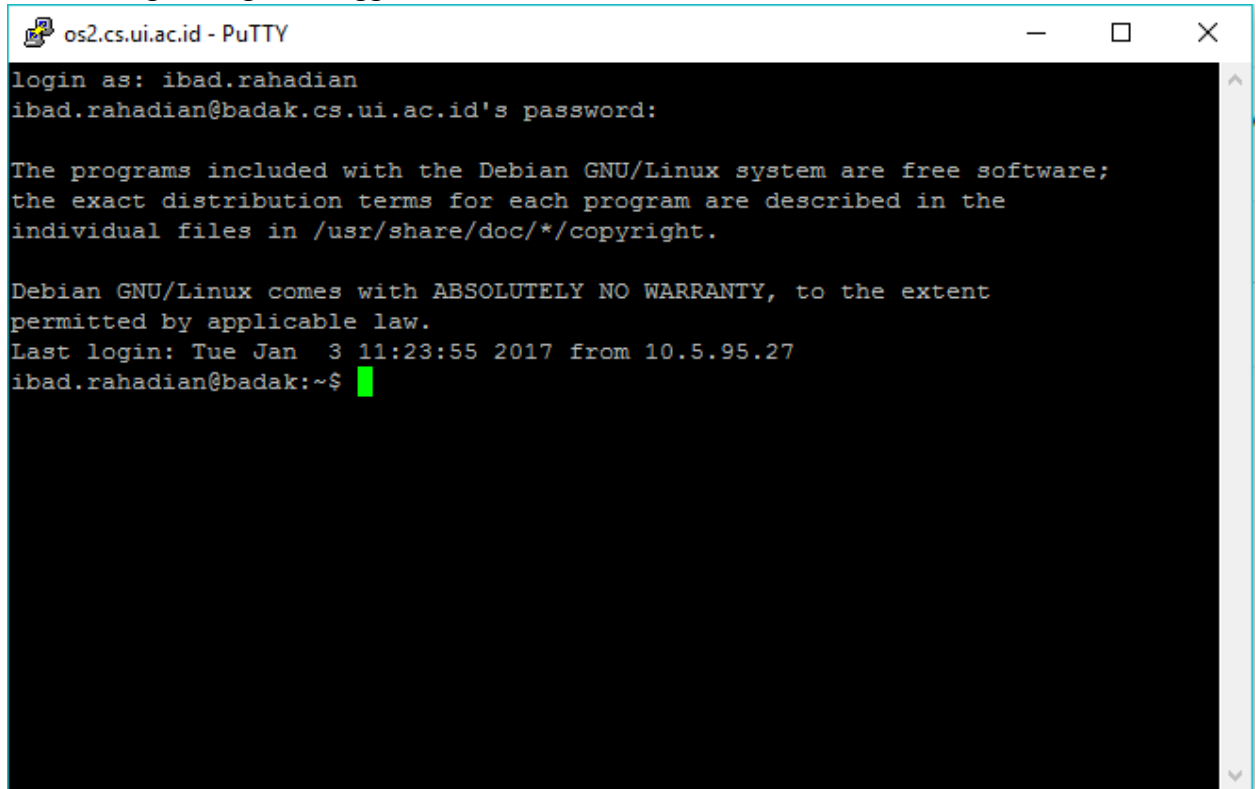
2. Pada kotak di sebelah kiri arahkan kepada halaman Connections -> SSH -> Tunnels, lalu masukkan port yang diinginkan dan nama server kawung.cs.ui.ac.id. Lalu, klik tombol *add*.



3. Lalu pada halaman login, lakukan akses ke halaman badak.cs.ui.ac.id, lalu klik *open*.



4. Lakukan login dengan menggunakan akun LDAP UI Anda.



The image shows a PuTTY terminal window titled "os2.cs.ui.ac.id - PuTTY". The terminal output displays the login process for the user "ibad.rahadian". It prompts for the password, shows the Debian GNU/Linux system information, and displays the last login time as "Tue Jan 3 11:23:55 2017 from 10.5.95.27". The prompt "ibad.rahadian@badak:~\$" is shown with a green cursor.

```
os2.cs.ui.ac.id - PuTTY
login as: ibad.rahadian
ibad.rahadian@badak.cs.ui.ac.id's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan  3 11:23:55 2017 from 10.5.95.27
ibad.rahadian@badak:~$
```