

# TUTORIAL LINUX NETWORKING MENGUNAKAN ZENITY



NAMA ANGGOTA KELOMPOK

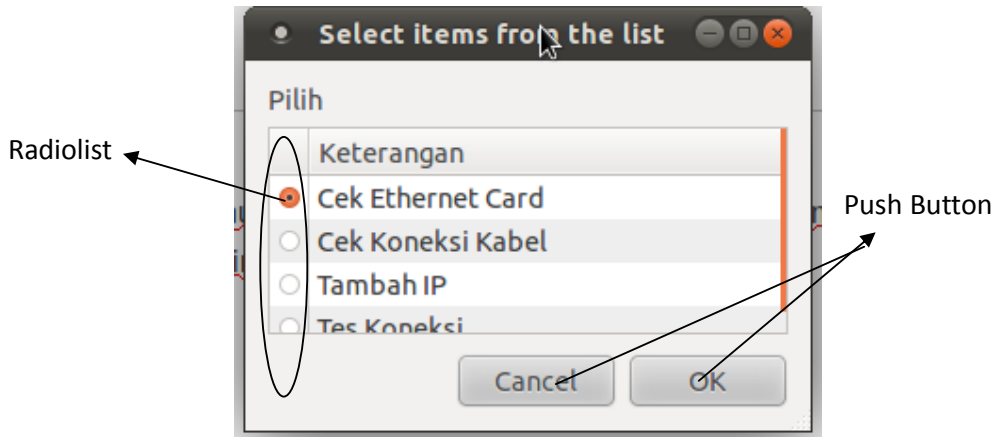
1. Agustinus Abraham (7209040034)
2. Tatag Adiktia (7209040045)
3. Mazda Yuda Adhitya (7209040048)

PRAKTIKUM JARINGAN KOMPUTER 1  
JURUSAN TEKNIK TELEKOMUNIKASI

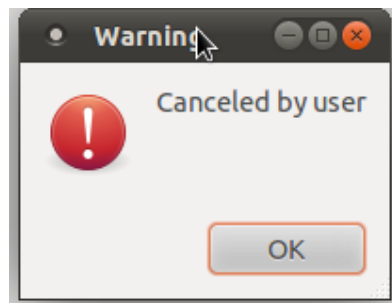
PENS-ITS

2011

Mula-mula kita harus mengetahui desain GUI (Grapical User Interface) yang akan dibuat. Untuk tampilan awal akan dibuat seperti tampilan di bawah ini

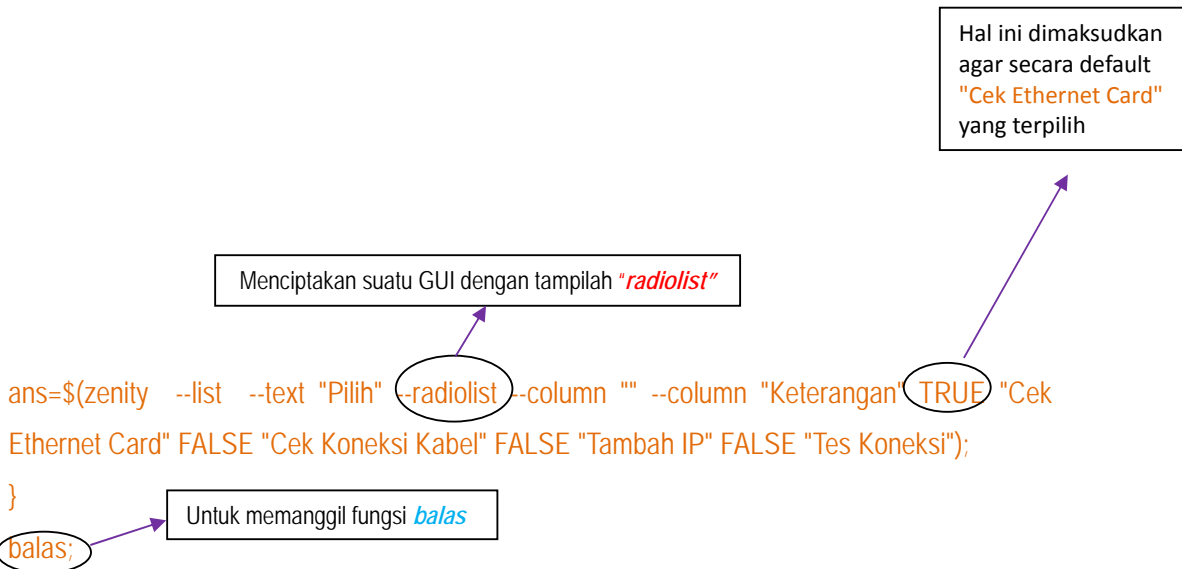


Tampilan di atas adalah tampilan awal dari GUI. Tampilan tersebut menggunakan perintah *zenity* yang akan dapat dipilih salah satu opsi dari keempat opsi tersebut diantaranya Cek Ethernet Card, Cek koneksi kabel, Tambah IP, serta tes koneksi. Karena kita hanya menginginkan hanya dapat memilih salah satu dari pilihan tersebut maka menggunakan radiolist. Selain hal itu juga terdapat push button "Cancel" dan "OK" sehingga dua pilihan tersebut dapat digunakan apakah setuju memilih misalnya Cek Ethernet Card jika ingin memilih itu maka tekan "OK" jika tidak maka dapat memilih opsi yang lainnya, jika ditekan cancel maka akan keluar dari tampilan tersebut dan akan muncul tampilan seperti dibawah ini.



Tampilan awal GUI tersebut menggunakan beberapa program zenity seperti dibawah ini

```
#!/bin/bash  
balas(){
```

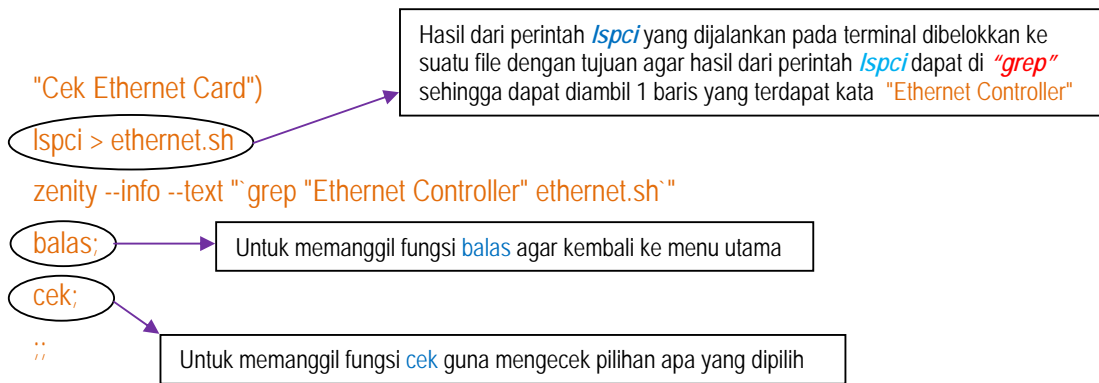


Sebelum masuk lebih dalam mengenai penjelasan tentang tampilan awal, dalam program yang tertulis di atas sudah menunjukkan bahwa tampilan awal adalah menggunakan radiolist, selain itu ada satu hal lagi yang harus digunakan agar nantinya dalam prosesnya akan seperti yang diharapkan. Perintah itu adalah case. Dimana perintah case adalah suatu perintah yang digunakan untuk sebuah pilihan yang lebih dari satu pilihan dan dapat memilih salah satu opsi, jika tidak memilih satu opsi tersebut maka akan dilanjutkan ke opsi berikutnya, singkatnya perintah case adalah suatu pengkondisian dimana ada suatu opsi yang harus dipilih salah satu. Selain itu program ini sebagian besar menggunakan fungsi. Hal ini dilakukan untuk tidak mengulang program yang sudah ditulis sehingga apabila nantinya terdapat program yang sama, maka kita hanya memanggilnya dengan nama fungsi tersebut yang sudah dibuat. Dalam program di atas nama fungsinya adalah fungsi *balas*, sehingga untuk memanggilnya cukup dengan menuliskan namanya *balas* diikuti dengan tanda titik koma ;.

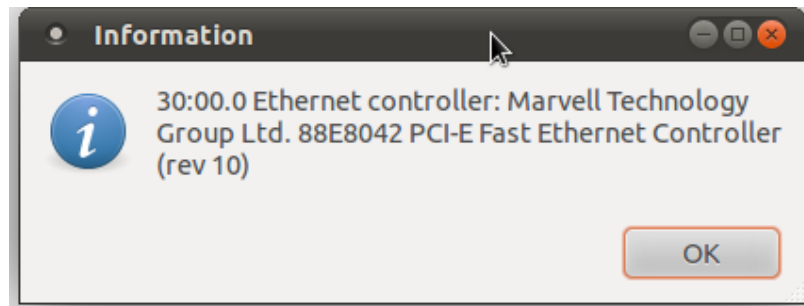
Sekarang kita menuju ke pilihan awal yaitu cek Ethernet Card. Dimana cek Ethernet Card digunakan untuk mengetahui apakah suatu ethernet card sudah terinstall atau belum pada komputer atau laptop yang kita pakai. Dalam penggunaan secara langsung pengecekan suatu Ethernet Card harus menggunakan terminal maka akan digunakan perintah *lspci*. Dalam perintah *lspci* terdapat suatu kalimat yang menunjukkan bahwa Ethernet Card sudah terinstall, sehingga untuk menampilkannya menggunakan zenity harus dilakukan suatu perintah untuk

mengambil kalimat yang menunjukkan bahwa Ethernet Card sudah terinstall dengan menggunakan perintah "grep" sekarang mari kita lihat program yang digunakan

```
cek(){  
case $ans in
```

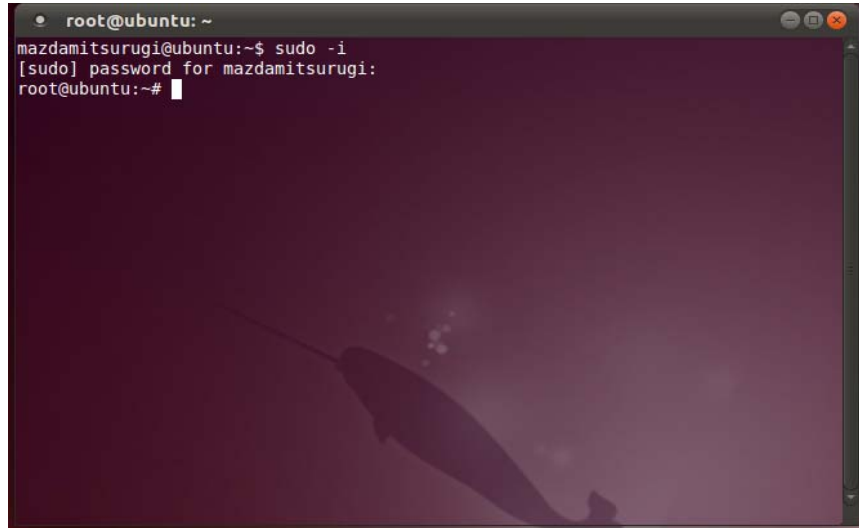


Pada program di atas diikut sertakan fungsi cek. Cek dalam hal ini menyangkup secara keseluruhan opsi-opsi pilihan didalamnya sehingga fungsi cek tersebut akan dapat menampilkan ke kondisi awal bersama dengan fungsi *balas*. Kemudian seperti yang dijelaskan sebelumnya bahwa program ini menggunakan perintah case yaitu pengkondisian, jika memilih Cek Ethernet Card, maka program akan menjalankan perintah *lspci*. Untuk mengambil kalimat yang ada perintah *lspci* dengan perintah "*grep*" maka isi dari perintah *lspci* tersebut akan dimasukan ke dalam sebuah editor bernama "ethernet.sh" dengan demikian isi dari perintah *lspci* tersebut akan masuk ke dalam ethernet.sh, sehingga dengan menggunakan perintah "*grep*" tersebut yang awal kalimatnya berbunyi "Ethernet Controller" pada editor ethernet.sh akan diambil dan ditampilkan pada window baru menggunakan perintah zenity yang bermode info, sehingga tampilannya apabila ethernet card sudah terinstall adalah seperti berikut



Berikutnya adalah adalah opsi Cek Koneksi Kabel. Maksudnya disini adalah mengesek apakah kabel yang digunakan untuk proses "*ping*" sudah terpasang atau belum pada koneksi eth-nya. Jika menggunakan laptop maka hanya ada satu eth yang digunakan untuk

pemasangan kabel dalam hal ini memakai kabel utp yaitu eth0. Untuk itulah pengecekan kabel ini diperlukan dan digunakan perintah *mii-tool*. Perlu diketahui oleh pengguna Linux untuk menjalankan perintah *mii-tool* ini user harus menggunakan super user (root terminal) pada terminal. Untuk mengubah agar menjadi super user pada terminal bisa menggunakan perintah "su" atau "sudo -i" kemudian ketikkan password yang digunakan pada laptop atau pc yang digunakan. Lihat gambar berikut untuk lebih jelasnya



Dengan menggunakan super user maka sudah dapat dijalankan perintah *mii-tool*. Program yang terdapat pada opsi Cek Koneksi kabel ini juga hampir sama dengan opsi Cek Ethernet Card dengan menggunakan perintah *grep*. Hal ini dikarenakan dengan perintah *grep* akan mengambil kalimat yang ada pada koneksi eth0. Lihat program berikut ini

```

"Cek Koneksi Kabel")
mii-tool > m.sh
b=$(grep eth m.sh);
case $b in
"eth0: no link")
zenity --error --text="grep eth m.sh"
;;
*)
zenity --info --text "grep eth m.sh"
;;
esac
  
```

Hasil dari perintah *mii-tool* yang dijalankan pada terminal diblokkan ke suatu file dengan tujuan agar hasil dari perintah *mii-tool* dapat di "grep" sehingga dapat diambil 1 baris yang terdapat kata "eth"

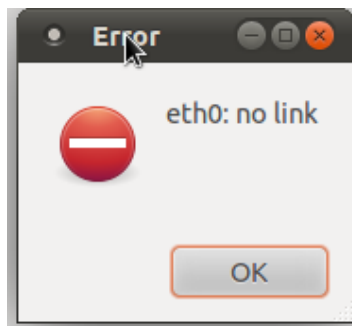
Hasil dari perintah "grep" dimasukkan ke dalam suatu variabel dengan tujuan agar dapat dicek hasilnya dengan perintah *case*

balas;

cek;

::

Sama seperti proses sebelumnya, sebelum melakukan proses "*grep*" maka isi dari perintah *mii-tool* tersebut di pindahkan ke text editor yang bernama m.sh. Dalam opsi Cek Koneksi Kabel ini terdapat perintah case, karena dalam koneksi kabel ini terdapat dua kondisi yang pertama yaitu kondisi saat kabel belum terpasang pada eth0 dan yang kedua yaitu saat kondisi kabel sudah terpasang pada eth0. Apabila kabel belum terpasang maka akan muncul tampilan



Sedangkan untuk tampilan apabila kabel sudah terpasang maka akan muncul tampilan



Berikutnya adalah adalah opsi Tambah IP. Maksudnya disini adalah mengganti IP dari suatu laptop atau pc yang digunakan dengan menggunakan perintah *ifconfig*. Perlu diketahui oleh pengguna Linux untuk menjalankan perintah *ifconfig* ini user harus menggunakan super user (root terminal) pada terminal. Untuk mengubah agar menjadi super user pada terminal bisa menggunakan perintah "*su*" atau "*sudo -i*" kemudian ketikkan passwaord yang diguinan pada laptop atau pc yang digunakan. Lihat gambar berikut untuk lebih jelasnya.

```
root@ubuntu: ~
mzdamitsurugi@ubuntu:~$ sudo -i
[sudo] password for mzdamitsurugi:
root@ubuntu:~#
```

Dengan menggunakan super user maka sudah dapat dijalankan perintah *ifconfig*. Dengan menggunakan perintah *ifconfig*, root dapat mengganti setting IP untuk jaringan. Contoh :

```
highway:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0C:F1:BA:38:43
          inet addr:10.252.102.143 Bcast:10.252.102.255
          Mask:255.255.255.0
          inet6 addr: fe80::20c:f1ff:feba:3843/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:7827318 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5486496 errors:0 dropped:0 overruns:0
          carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2529035045 (2.3 GiB) TX bytes:1421757215 (1.3
          GiB)

Lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:999 errors:0 dropped:0 overruns:0 frame:0
          TX packets:999 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:68831 (67.2 KiB) TX bytes:68831 (67.2 KiB)
```

Program yang terdapat pada opsi Tambah IP ini juga hampir sama dengan opsi Cek Koneksi kabel dengan menggunakan perintah *grep*. Hal ini dikarenakan dengan perintah *grep* akan mengambil kalimat yang ada kalimat **Bcast**:. Lihat program berikut ini :

"Tambah IP")

```
coba(){
```

Hasil dari perintah "GUI" dimasukkan ke dalam suatu variabel dengan tujuan agar dapat dicek hasilnya dengan perintah Pengkondisian

```
BEK=$(zenity --title "IP Address" --entry --text "ketikkan IP Address");
```

```
if [ "$?" = 1 ]; then
```

Untuk mengecek apakah "GUI" untuk memasukkan IP Address di cancel

```
zenity --warning \
```

```
--text="Canceled by user"
```

```
balas;
```

```
cek;
```

```
fi
```

```
case $BEK in
```

```
"")
```

Untuk mengecek apakah "GUI" untuk memasukkan IP Address tidak diisi

```
zenity --error --text="masukan IP"
```

```
coba;
```

```
::
```

```
*)
```

```
ifconfig eth0 $BEK
```

```
ifconfig > i.sh
```

```
highway:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0C:F1:BA:38:43
          inet addr:10.252.102.143 Bcast:10.252.102.255
          Mask:255.255.255.0
          inet6 addr: fe80::20c:f1ff:feba:3843/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:7827318 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5486496 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2529035045 (2.3 GiB) TX bytes:1421757215 (1.3 GiB)

Lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:999 errors:0 dropped:0 overruns:0 frame:0
          TX packets:999 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:68831 (67.2 KiB) TX bytes:68831 (67.2 KiB)
```

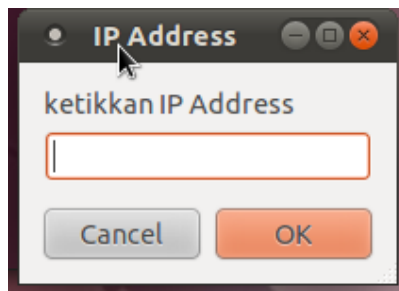
```
zenity --info --text "`grep "Bcast:" i.sh`"
```

Yang kita grep adalah kalimat yang ada kata **Bcast:** karena apabila yang kita grep adalah kalimat yang ada kata *inet* saja ataupun *inet addr* maka akan muncul kalimat lain yang tidak diinginkan seperti yang di koka merah diatas

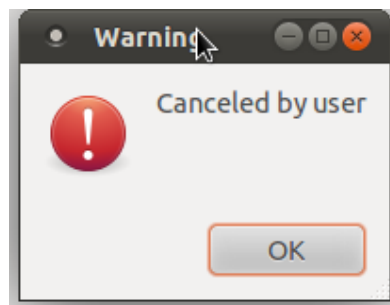


```
;;  
  
esac  
  
}  
  
coba;  
  
balas;  
  
cek;  
  
;;
```

Pada program di atas diikut sertakan fungsi *coba*. Coba dalam hal ini menyangkup secara keseluruhan program yang terdapat didalamnya. Pada fungsi *coba* tersebut, pertama-tama kita menampilkan suatu GUI (Graphical User Interface) yang meminta kita untuk mengetikkan suatu IP Address sesuai dengan keinginan kita dengan tampilan sebagai berikut :

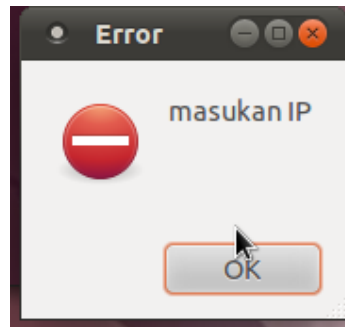


Apabila GUI tersebut kita "cancel" dengan cara mengklik push button "cancel" maka akan muncul suatu GUI peringatan (warning) yang menyatakan bahwa proses tersebut di batalkan oleh user (Canceled by user) dengan tampilan sebagai berikut :

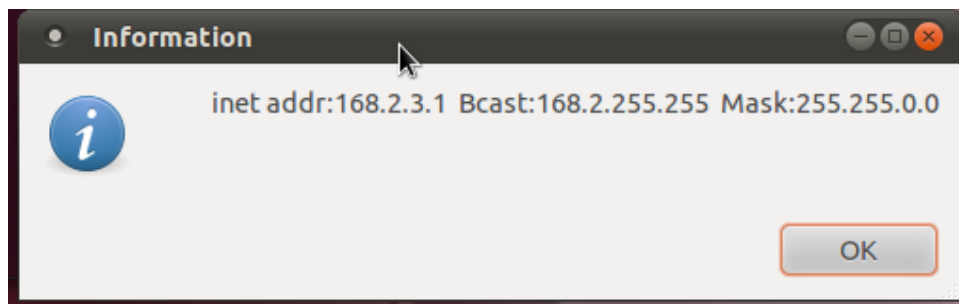


Dan setelah itu kita langsung menuju menu utama dengan cara memanggil fungsi

*balas* kemudian kita Untuk memanggil fungsi *cek* guna mengecek pilihan apa yang dipilih. Apabila kita tidak mengisi IP Address pada kolom yang telah disediakan maka akan muncul suatu GUI bertipe "error" dengan tulisan "masukan IP" dengan tampilan sebagai berikut:

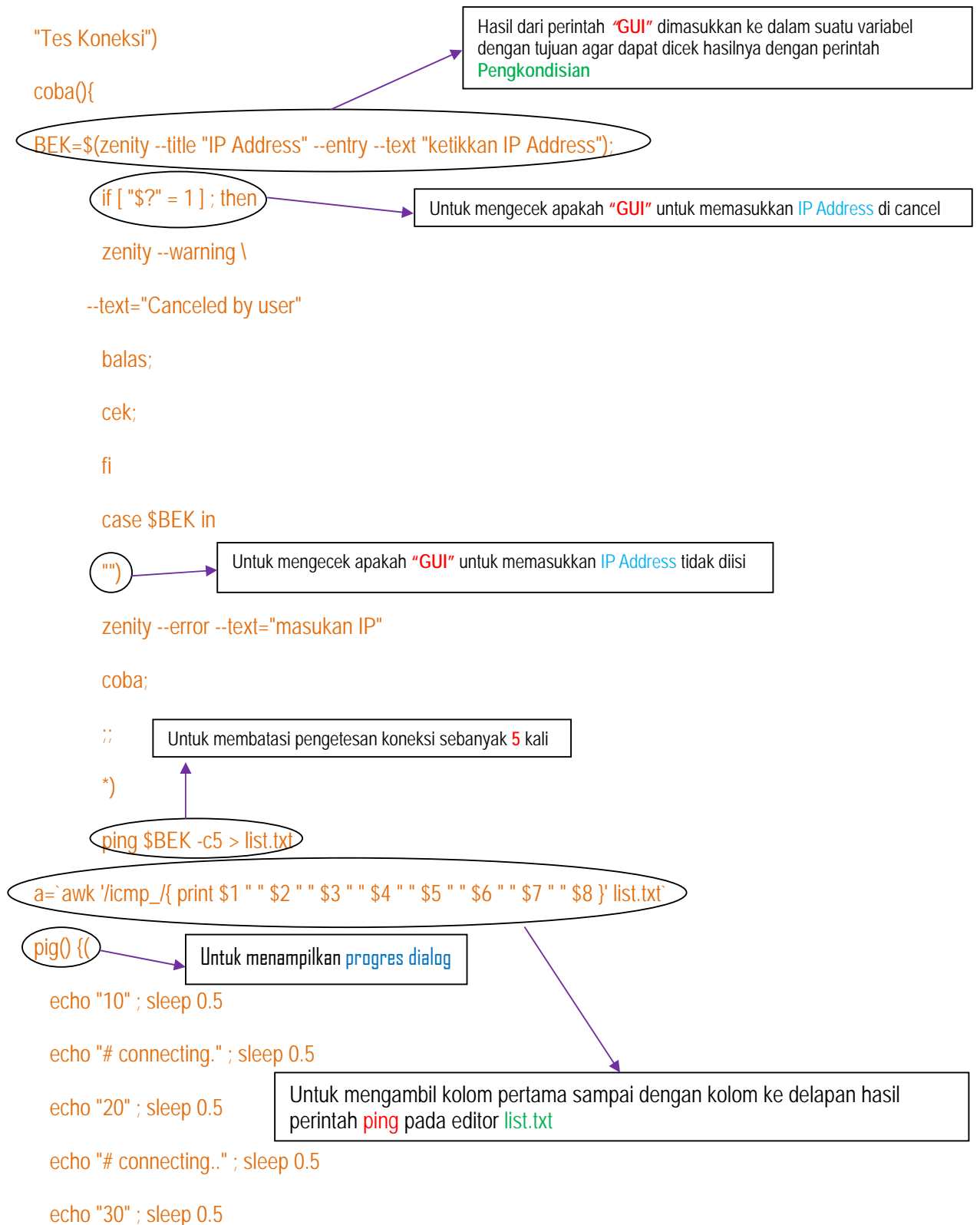


Dan setelah itu kita kembali lagi ke GUI yang meminta kita untuk mengetikkan suatu IP Address sesuai dengan keinginan kita. Sedangkan apabila kita mengetikkan suatu IP Address sesuai dengan keinginan kita.maka program secara otomatis akan mengganti IP Address dari laptop atau pc yang kita gunakan dengan memakai perintah *ifconfig* maka hasil dari perintah *ifconfig* tersebut akan dimasukkan ke dalam sebuah editor bernama "i.sh" dengan demikian hasil dari perintah *ifconfig* tersebut akan masuk ke dalam i.sh, sehingga dengan menggunakan perintah "*grep*" tersebut yang ada kalimatnya berbunyi "Bcast:" pada editor i.sh akan diambil dan ditampilkan pada window baru menggunakan perintah zenity yang bermode info, sehingga tampilannya apabila perintah *ifconfig* dapat dilakukan adalah seperti berikut



Berikutnya adalah adalah opsi Tes Koneksi. Dimana Tes Koneksi digunakan untuk mengetahui apakah komputer atau laptop yang kita pakai dapat menghubungi atau berkomunikasi dengan yang lainnya. Program yang terdapat pada opsi Tes Koneksi ini juga hampir sama dengan opsi Tambah IP.

Lihat program berikut ini :



```
echo "# connecting..." ; sleep 0.5
echo "40" ; sleep 0.5
echo "# connecting." ; sleep 0.5
echo "50" ; sleep 0.5
echo "# connecting.." ; sleep 0.5
echo "60" ; sleep 0.5
echo "# connecting..." ; sleep 0.5
echo "70" ; sleep 0.5
echo "# connecting." ; sleep 0.5
echo "80" ; sleep 0.5
echo "# connecting.." ; sleep 0.5
echo "90" ; sleep 0.5
echo "# connecting..." ; sleep 0.5
echo "100" ; sleep 0.5
echo "# complete" ;
) | zenity --progress \
    --title="Progress" \
    --text="connect to server" \
    --percentage=100
if [ "$?" = 1 ] ; then
    zenity --warning \
        --text="Canceled by user"
#     exit 0;
    coba;
fi
```

Untuk mengecek apakah "GUI" untuk progress dialog di cancel

```

}

pig;

case $a in
    "(")
        zenity --error --text "connect: Network is unreachable"
        ;;
    *)
        zenity --info --text "          KONEKSI BERHASIL"
        $a"
        ;;
esac

balas;

cek;

;;

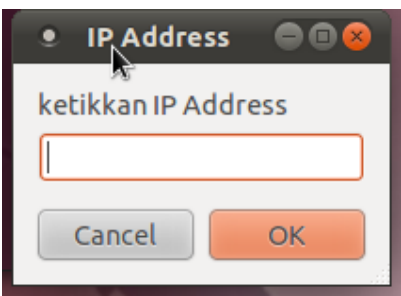
esac

}

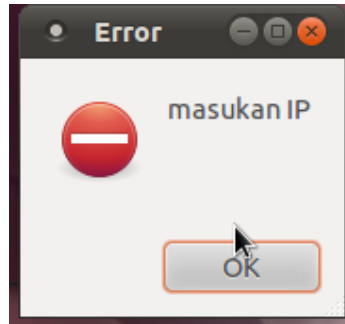
```

Untuk mengecek apakah "ping" gagal atau tidak

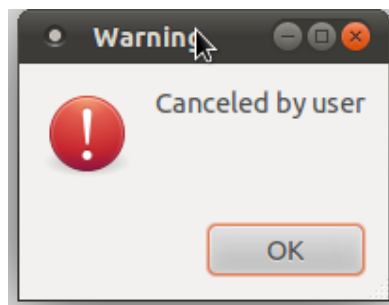
Dalam penggunaan secara langsung pengecekan suatu pc atau laptop yang kita gunakan dapat berkomunikasi dengan yang lainnya harus menggunakan terminal maka akan digunakan perintah *ping*. Sebelum kita dapat melakukan pengetesan suatu koneksi maka kita haus mengetahui suatu IP Address target. Untuk itulah pada opsi Tes Koneksi ini kita diminta untuk memasukkan suatu IP Address sehingga akan muncul tampilan GUI seperti berikut :



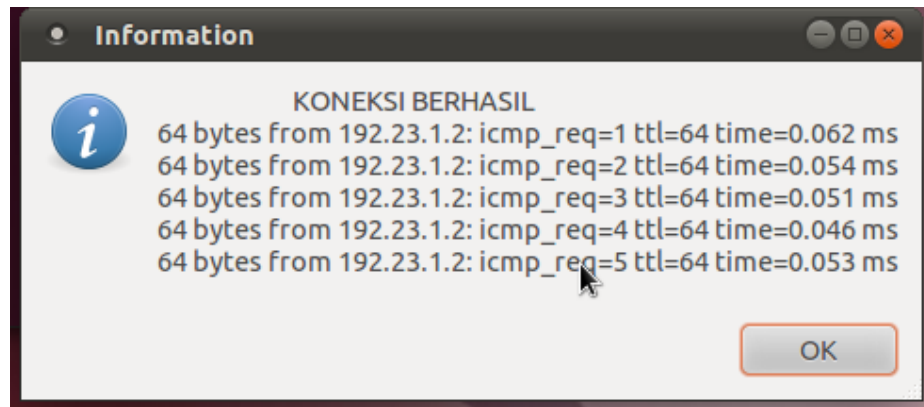
Apabila kita tidak mengisi suatu IP Address maka akan muncul suatu GUI bertipe "error" dengan tulisan "masukkan IP" seperti berikut ini :



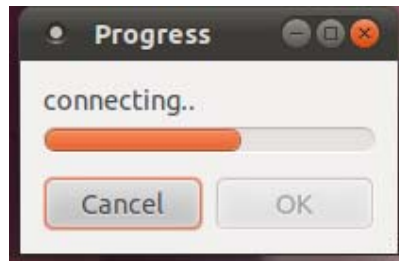
Setelah itu kita kembali lagi ke GUI yang meminta kita untuk mengetikkan suatu IP Address. Apabila proses tersebut kita "cancel" dengan cara mengklik push button "cancel" maka akan muncul suatu GUI peringatan (warning) yang menyatakan bahwa proses tersebut di batalkan oleh user (Canceled by user) dengan tampilan seperti berikut :



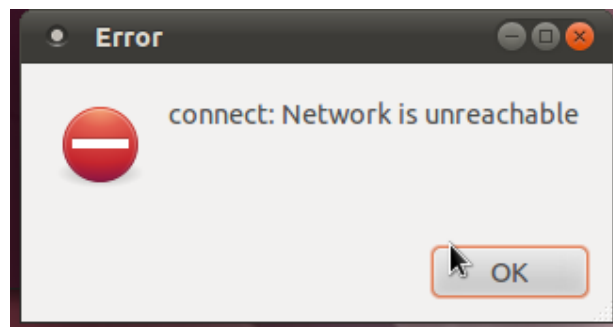
Kemudian kita langsung menuju menu utama dengan cara memanggil fungsi *balas* kemudian kita Untuk memanggil fungsi *cek* guna mengecek pilihan apa yang dipilih. Apabila kita telah mengetikkan suatu IP Address maka kita akan melakukan proses pengetesan koneksi menggunakan perintah *ping*. Dalam perintah *ping* tersebut apabila kita tidak membatasi berapa banyak pengetesan koneksi yang akan dilakukan maka pc atau laptop kita akan terus melakukan pengecekan koneksi, sehingga untuk itu maka kita harus membatasi banyaknya pengetesan koneksi yang akan dilakukan oleh pc atau laptop yang kita gunakan. Untuk mengambil kalimat yang merupakan hasil dari perintah *ping* maka digunakan suatu perintah "*awk*" maka hasil dari perintah *ping* tersebut akan dimasukkan ke dalam sebuah editor bernama "list.txt" dengan demikian hasil dari perintah *ping* tersebut akan masuk ke dalam list.txt, sehingga dengan menggunakan perintah "*awk*" tersebut kita dan ditampilkan pada window baru menggunakan perintah zenity yang bermode info, sehingga tampilannya apabila Tes Koneksi berhasil adalah seperti berikut:



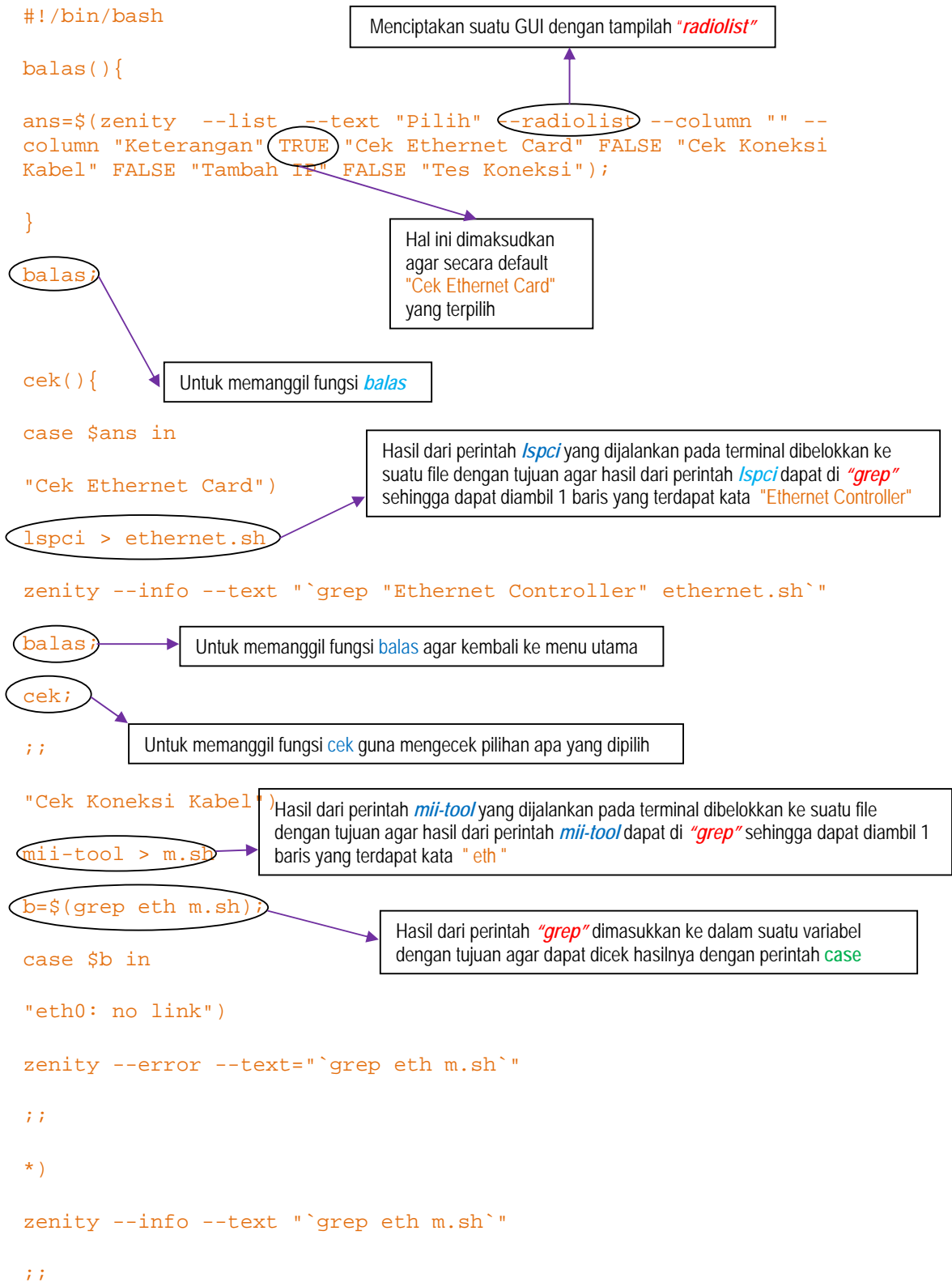
Sebelum tampil perintah zenity yang bermode info, kita menampilkan suatu progress dialog agar mempercantik penampilan dengan tampilan sebagai berikut :



Apabila Tes Koneksi tersebut gagal maka akan muncul tampilan seperti berikut :



## LAMPIRAN





```

esac

balas;

cek;

;;

"Tambah IP")

coba(){

```

Hasil dari perintah "GUI" dimasukkan ke dalam suatu variabel dengan tujuan agar dapat dicek hasilnya dengan perintah Pengkondisian

```

BEK=$(zenity --title "IP Address" --entry --text "ketikkan IP Address");

```

```

if [ "$?" = 1 ] ; then

```

Untuk mengecek apakah "GUI" untuk memasukkan IP Address di cancel

```

zenity --warning \

--text="Canceled by user"

```

```

balas;

cek;

fi

case $BEK in

```

```

" ")

```

Untuk mengecek apakah "GUI" untuk memasukkan IP Address tidak diisi

```

zenity --error --text="masukan IP"

```

```

coba;

;;

*)

ifconfig eth0 $BEK

ifconfig > i.sh

```

```

highway:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0C:F1:BA:38:43
          inet addr:10.252.102.143 Bcast:10.252.102.255
          Mask:255.255.255.0
          inet6 addr: fe80::20c:f1ff:feba:3843/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:7827318 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5486496 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2529035045 (2.3 GiB) TX bytes:1421757215 (1.3 GiB)

Lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:999 errors:0 dropped:0 overruns:0 frame:0
          TX packets:999 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:68831 (67.2 KiB) TX bytes:68831 (67.2 KiB)

```

```

zenity --info --text "`grep "Bcast:" i.sh`"

```

```

;;

esac

}

```

Yang kita grep adalah kalimat yang ada kata Bcast: karena apabila yang kita grep adalah kalimat yang ada kata inet saja ataupun inet addr maka akan muncul kalimat lain yang tidak diinginkan seperti yang di koka merah diatas

```

coba;

balas;

```

```
cek;
```

```
;;
```

```
"Tes Koneksi")
```

```
coba(){
```

```
BEK=$(zenity --title "IP Address" --entry --text "ketikkan IP Address");
```

Hasil dari perintah "GUI" dimasukkan ke dalam suatu variabel dengan tujuan agar dapat dicek hasilnya dengan perintah **Pengkondisian**

```
if [ "$?" = 1 ] ; then
```

Untuk mengecek apakah "GUI" untuk memasukkan IP Address di cancel

```
zenity --warning \
```

```
    --text="Canceled by user"
```

```
balas;
```

```
cek;
```

```
fi
```

```
case $BEK in
```

```
"")
```

Untuk mengecek apakah "GUI" untuk memasukkan IP Address tidak diisi

```
zenity --error --text="masukan IP"
```

```
coba;
```

```
;;
```

```
*)
```

```
ping $BEK -c5 > list.txt
```

```
a=`awk '/icmp_{ print $1 " " $2 " " $3 " " $4 " " $5 " " $6 " " $7 " " $8 }' list.txt`
```

```
pig() {(
```

Untuk menampilkan **progres dialog**

```
echo "10" ; sleep 0.5
```

```
echo "# connecting." ; sleep 0.5
```

Untuk mengambil kolom pertama sampai dengan kolom ke delapan hasil perintah **ping** pada editor **list.txt**

```
echo "20" ; sleep 0.5
```

```
echo "# connecting.." ; sleep 0.5
```

```
echo "30" ; sleep 0.5
```

```
echo "# connecting..." ; sleep 0.5
echo "40" ; sleep 0.5
echo "# connecting." ; sleep 0.5
echo "50" ; sleep 0.5
echo "# connecting.." ; sleep 0.5
echo "60" ; sleep 0.5
echo "# connecting..." ; sleep 0.5
echo "70" ; sleep 0.5
echo "# connecting." ; sleep 0.5
echo "80" ; sleep 0.5
echo "# connecting.." ; sleep 0.5
echo "90" ; sleep 0.5
echo "# connecting..." ; sleep 0.5
echo "100" ; sleep 0.5
echo "# complete" ;
) | zenity --progress \
    --title="Progress" \
    --text="connect to server" \
    --percentage=100
if [ "$?" = 1 ] ; then
    zenity --warning \
        --text="Canceled by user"
# exit 0;
coba;
fi
}
pig;
```



```
case $a in
```

```
"")
```

Untuk mengecek apakah "ping" gagal atau tidak

```
zenity --error --text "connect: Network is unreachable"
```

```
;;
```

```
*)
```

```
zenity --info --text "                                KONEKSI BERHASIL
```

```
$a"
```

```
;;
```

```
esac
```

```
balas;
```

```
cek;
```

```
    ;;
```

```
    esac
```

```
    }
```

```
coba;
```

```
;;
```

```
*)
```

```
    zenity --warning \  
        --text="Canceled by user"
```

```
        exit 0;
```

```
;;
```

```
esac
```

```
}
```

```
cek;
```