ROUTER

PENGGUNAAN BANYAK ROUTER MENUJU IP ADDRESS TUJUAN

PROYEK 3 Praktikum Jaringan Komputer 1

[Ismed Jauhar, Nur Annisa, Ima Ristiani]
Jurusan Teknik Telekomunikasi
PENS-ITS, 2011

PROYEK JARINGAN KOMPUTER BAB 3

DASAR TEORI

- **netstat**

Netstat (NETwork STATistics) adalah command-line tool yg menyediakan informasi tentang konfigurasi jaringan dan aktifitasnya.

Untuk menampilkan routing table :
#netstat -rn
-> -r : Kernel routing table
-> -n : Menampilkan alamat numerik.

Untuk menampilkan statistik interface :
#netstat -i
-> -i : Interface

Untuk menampilkan informasi tambahan interface :
#netstat -ie
-> -i : Interface
-> -e : Extended information
command ini sama dengan perintah "*ifconfig -a*"

Untuk menamplikan soket network :
#netstat -uta
-> -u : UDP
-> -t : TCP
-> -a : ALL
Kemungkinan yg muncul dari status soket adalah sebagai berikut :
ESTABLISHED : Koneksi terjalin
SYN_SENT : Soket berusaha untuk menjalin koneksi
SYN_RECV : Request koneksi sudah diterima dari network
FIN_WAIT1 : Soket close, dan koneksi shutdown
FIN_WAIT2 : Soket close, dan soket menunggu sisi remote shutdown
TIME_WAIT : Soket menunggu setelah close utk menangani paket yg masih di network
CLOSED : Soket tidak digunakan
CLOSE_WAIT : Sisi remote sudah shutdown, menunggu soket close.
LAST_ACK : Sisi remote sudah shutdown, dan soket sudah close, menunggu ack.
LISTEN : Soket sedang menerima koneksi
CLOSING : 2 sisi soket shutwodn
UNKNOWN : Meneketehe

Untuk menampilkan semua soket yg open (info tambahan) :
#netstat -aute
-> -a : ALL
-> -u : UDP
-> -t : TCP
-> -e : Extended

Untuk menampilkan semua soket yg listen
 # netstat -lt
-> -t : TCP
-> -l : Status soket

- **traceroute**

Kadang-kadang alamat web yang sering kita kunjungi tidak dapat diakses secepat biasanya, di internet hal ini dapat terjadi karena beberapa sebab, yang paling sering adalah karena jalur internet yang kita lalui memang sedang melamba tatau penuh atau server dari alamat web tersebut sedang diakses oleh banyak orang sehingga membutuhkan waktu bagi server tersebut untuk memproses permintaan kita.

Memang sulit untuk mendeteksi permasalahan yang ada pada server remote (server yang terletak di tempat lain), tetapi ada beberapa software yang dapat membantu kita untuk mendeteksi kondisi jaringan yang kita lalui.

Dua software yang paling sering penulis pakai untuk mendeteksi jaringan adalah ping dan traceroute. Yang akan kita pelajari di sini adalah penggunaan traceroute yang akan menunjukkan pada kita jalur router yang dilewati oleh paket yang kita kirimkan ke host tertentu. Untuk lebih memperjelas, berikut ini adalah contoh hasil traceroute ke
www.berkeley.edu:

$ traceroute www.berkeley.edu
traceroute to amber.Berkeley.EDU (128.32.25.12), 30 hops max, 40 byte packets
1 203.130.216.2 (203.130.216.2) 137 ms 151 ms 151 ms
2 203.130.216.1 (203.130.216.1) 151 ms 137 ms 138 ms
3 192.168.8.49 (192.168.8.49) 137 ms 151 ms 151 ms
4 S12-0-11.kbl.surabaya.telkom.net.id (202.134.3.45) 192 ms 151 ms 151 ms
5 FE0-0-gw3.cibinong.telkom.net.id (202.134.3.134) 165 ms 151 ms 151 ms
6 hssi-gw3.hk.telkom.net.id (202.134.3.1) 659 ms 659 ms 645 ms
7 202.130.129.61 (202.130.129.61) 645 ms 687 ms 659 ms
8 321.ATM5-0-0.XR1.HKG2.ALTER.NET (210.80.3.1) 645 ms 659 ms 645 ms
9 POS1-0-0.TR1.HKG2.Alter.Net (210.80.48.21) 672 ms 646 ms 645 ms
10 384.ATM4-0.IR1.LAX12.Alter.Net (210.80.50.189) 838 ms 796 ms 796 ms
11 137.39.31.222 (137.39.31.222) 810 ms 852 ms 810 ms
12 122.at-5-1-0.TR1.LAX9.ALTER.NET (152.63.10.237) 824 ms 810 ms 810 ms
13 297.at-1-0-0.XR1.LAX9.ALTER.NET (152.63.112.237) 824 ms 838 ms 824 ms
14 191.ATM6-0.BR1.LAX9.ALTER.NET (152.63.113.9) 837 ms 797 ms 810 ms
15 acr1-loopback.Anaheim.cw.net (208.172.34.61) 810 ms 1071 ms 782 ms
16 acr1-loopback.SanFranciscosfd.cw.net (206.24.210.61) 783 ms 810 ms 769 ms
17 BERK-7507--BERK.POS.calren2.net (198.32.249.69) 810 ms 1126 ms 796 ms
18 pos1-0.inr-000-eva.Berkeley.EDU (128.32.0.89) 796 ms 824 ms 796 ms

19 pos5-0-0.inr-001-eva.Berkeley.EDU (128.32.0.66) 796 ms 783 ms 783 ms
20 fast1-0-0.inr-007-eva.Berkeley.EDU (128.32.0.7) 810 ms 810 ms 797 ms
21 f8-0.inr-100-eva.Berkeley.EDU (128.32.235.100) 797 ms 782 ms 769 ms
22 amber.Berkeley.EDU (128.32.25.12) 796 ms 769 ms 810 ms

Traceroute akan menampilkan titik-titik perantara yang menjembatani anda dan titik tujuan anda, 'jembatan' inilah yang biasa disebut dengan router, data yang anda kirimkan akan meloncat melewati jembatan-jembatan ini. Ada tiga buah waktu yang menunjukkan berapa waktu yang dibutuhkan oleh paket tersebut untuk berjalan dari komputer anda ke router. Untuk dapat memahami seluruh data yang dihasilkan oleh traceroute tersebut, kita harus memahami bagaimana cara traceroute bekerja.

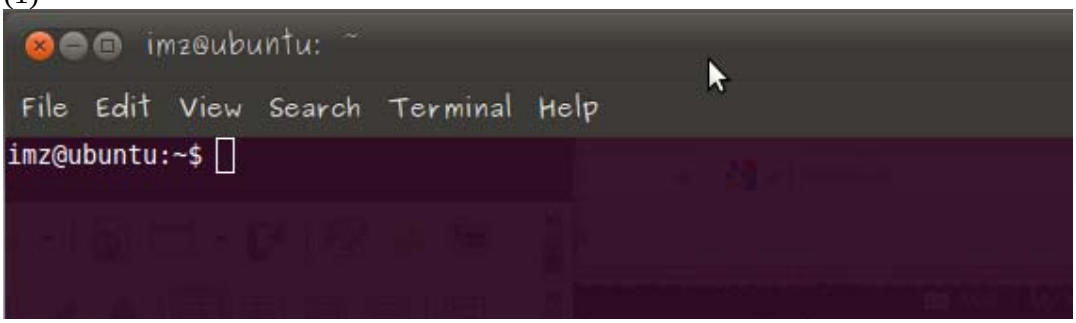
Traceroute menggunakan prinsip *TTL* dan paket *ICMP* pengiriman sebuah paket data yang disebut dengan *Internet Control Message Protocol* (ICMP) *Echo Request*. Paket ICMP ini biasanya digunakan untuk mengirimkan informasi tentang kondisi jaringan antara dua host (komputer). Jika sebuah host menerima Echo Request ini, dia harus merespon dengan mengirimkan Echo Reply, dengan menempatkan Echo Request ke bagian data pada Echo Reply.

Informasi berikutnya adalah *Time To Live*, setiap paket data yang dikirimkan melalui jaringan memiliki informasi yang disebut TTL, biasanya TTL ini diisi dengan angka yang relatif tinggi, (paket ping memiliki TTL 255). Setiap kali paket tersebut melewati sebuah router maka angka TTL ini akan dikurangi dengan satu, jika TTL suatu paket akhirnya bernilai 0, paket tersebut akan di drop atau dibuang oleh router yang menerimanya. Menurut aturan RFC untuk IP, TTL harus bernilai 60 (dan untuk ping 255). Kegunaan utama dari TTL ini supaya paket-paket data yang dikirim tidak '*live*' selamanya di dalam jaringan. Kegunaan yang lain, dengan informasi ini kita dapat mengetahui kira-kira berapa router yang dilewati oleh paket tersebut, dalam hal ini 255 dikurangi dengan N, dimana N adalah TTL yang kita lihat pada Echo Reply.

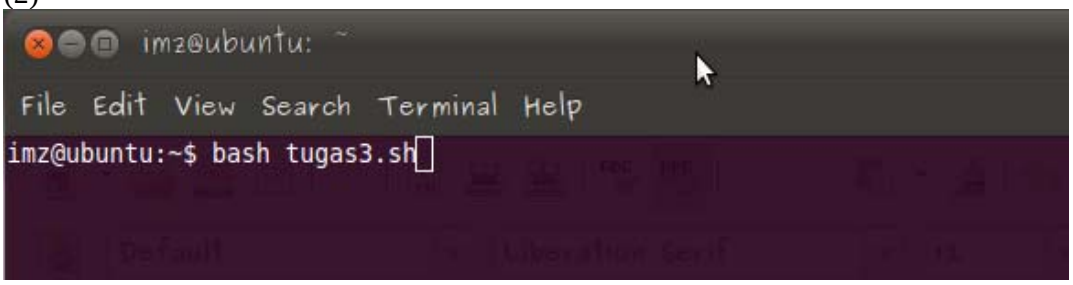
Traceroute mengirimkan sebuah paket ke port UDP yang tidak dipakai oleh servis lain pada komputer tujuan (defaultnya adalah port 33434). Untuk tiga paket pertama, traceroute mengirimkan paket yang memiliki TTL satu, maka sesampainya paket tersebut pada router pertama (menghasilkan loncatan yang pertama) TTL akan dikurangi dengan satu sehingga menjadi 0 kemudian paket tersebut akan di drop. Berikutnya router tersebut akan mengirimkan paket ICMP ke komputer kita yang berisi pemberitahuan bahwa TTL dari paket yang kita kirimkan sudah habis dan paket yang kita kirimkan di drop. Dari pesan ini, traceroute dapat menentukan nama router tempat data kita meloncat dan berapa waktu yang dibutuhkannya. Berikutnya traceroute akan mengirimkan

paket dengan nilai TTL yang ditambah satu demi satu sampai host tujuan dicapai. Karena itu traceroute menggunakan port yang tidak dipakai oleh servis lain sehingga paket yang dikirim mendapat respon dan tidak 'dimakan' oleh servis lain yang mungkin ada.

>>>**Mengecheck router yang digunakan untuk koneksi ke IP address**
Pertama, buat program dengan nama file : **tugas3.sh** *(terlampir)*
Buka terminal,
(1)



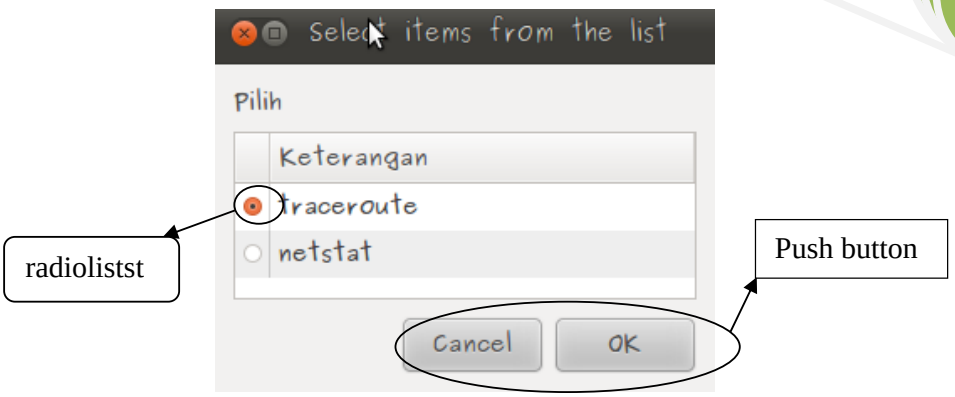
Nah,nantinya program **tugas3.sh** dijalankan dengan menggunakan terminal
(2)



Kemudian,ketika ditekan ***enter*** ,tampilan selanjutnya terbentuk dari sebuah fungsi,yaitu

```
#!/bin/bash
balas(){
ans=$(zenity  --list  --text "Pilih" --radiolist --column "" --column "Keterangan" TRUE "traceroute"
FALSE "netstat" );
}
balas;
```

menghasilkan tampilan menu utama seperti 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 traceroute dan netstat. Ketika kita 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 traceroute 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 di atas ini.

>> memilih traceroute

untuk kasus memilih "traceroute", fungsi programnya adalah

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

K=$(zenity --title "Tujuan Anda" --entry --text "ketikkan Tujuan Anda");
```

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

Maksud dari "$? =1" adalah ketika tampilan GUI meminta kita untuk menginputkan IP tujuan ,karena jika tidak kita masukkan inputkan IP tujuan ,maka program akan memanggil program dibawahnya , yaitu akan muncul warning" Canceled by user"

```
zenity --warning --text="Canceled by user"
balas;
cek;
fi
```

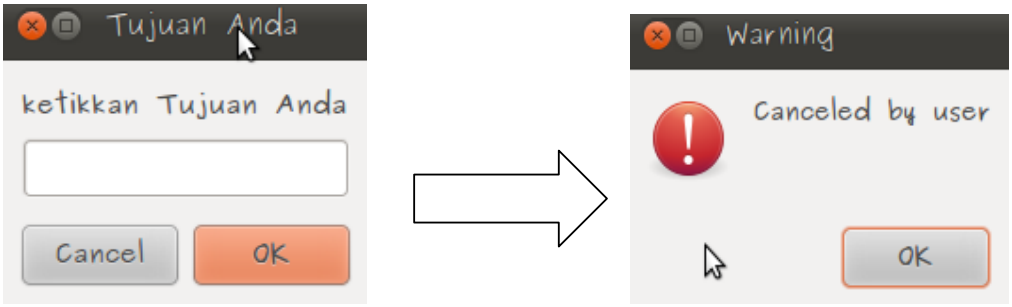
balas; → Memanggil fungsi *balas*

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(“ ; “) .

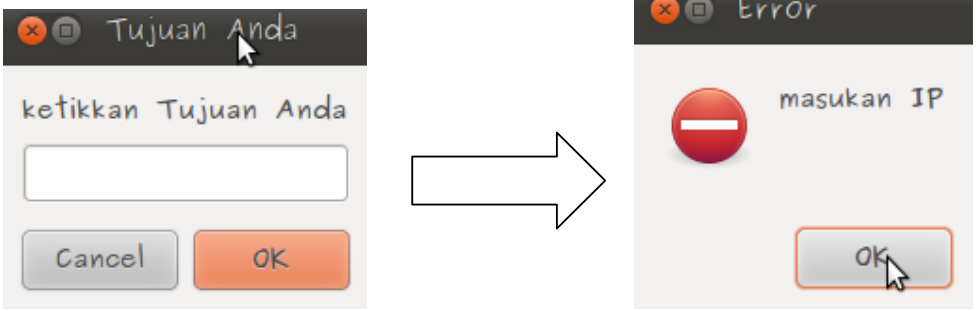
Pada tampilan utama GUI, jika kita memilih option ***traceroute*** ,maka kita diminta untuk memasukkan IP tujuan. Dan ketika kita memasukkan IP tersebut,maka IP akan disimpan dan dimasukkan ke variable ***K.*** Sehingga isi dari variable ***K*** sekarang adalah berisi IP tujuan.

Ketika muncul tampilan utama GUI dan kita memilih traceroute maka muncul GUI untuk meminta kita memasukkan IP tujuan dan kita batal memasukkan IP (kita click cancel) maka akan muncul warning Canceled by user.




```
case $K in
"")
zenity --error --text="masukan IP"
cek;
;;
*)
```

jika ketika memilih perintah traceroute,tapi tidak mengisi nomer IP, maka muncul tampilan



dikarenakan pada program **tugas3.sh** tampilan zenity error yang diatur muncul adalah “masukkan IP”

```
traceroute $K > trace1.sh
grep 'ms' trace1.sh > trace.sh
grep $K trace.sh > tes1.sh
a=`awk '{ print $1 }' tes1.sh`

let "hasil = $a-1"

zenity --info --text "Melewati $hasil router untuk sampai ke $K"
balas;
cek;
```

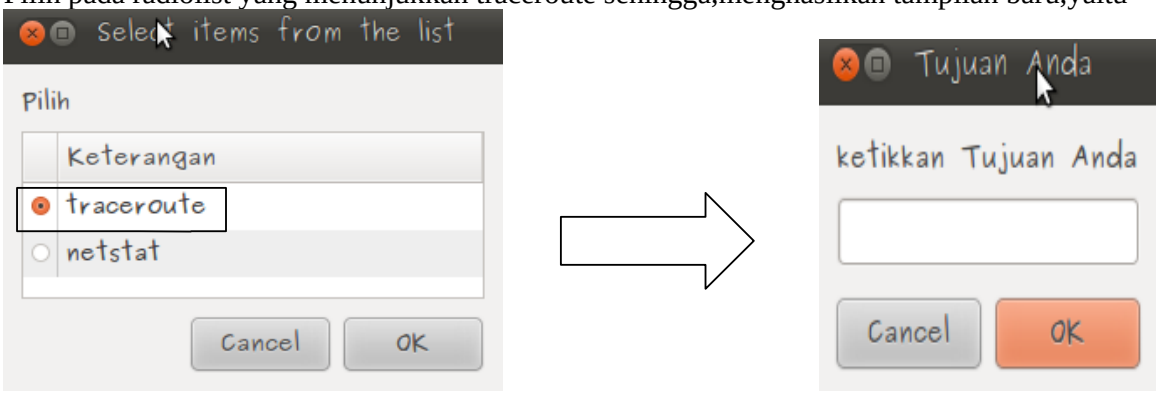
```
        ;;
        esac
;;
```

Lantas, cuplikan program diatas, penjelasanya adalah, traceroute dari K(IP tujuan), nilainya di-*copy*-kan ke file **trace1.sh,** kemudian, data yang mengandung unsur "ms" di-grep (diambil secara baris) dan di-*copy*-kan pada file bernama **trace.sh**. Selanjutnya, IP pada file trace.sh digrep(ambil) dan kemudian dimasukkan dalam file bernama **tes1.sh**.

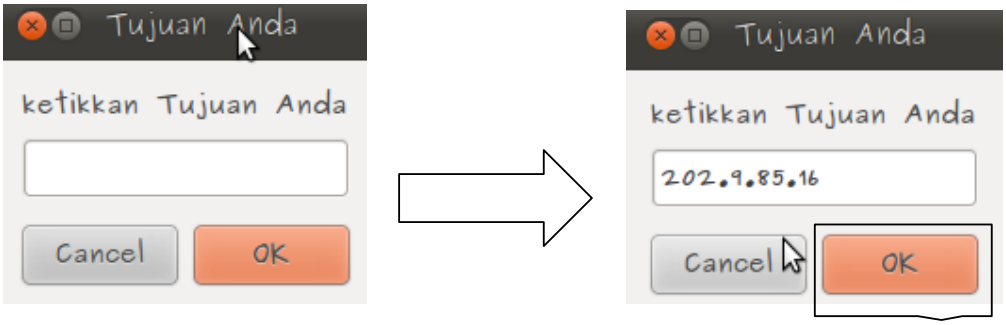
Lalu, data tersebut diambil secara kolom menggunakan perintah awk dari file **tes1.sh.** Dimana jalur yang dilewati untuk terkoneksi ke IP yang dituju ,jumlah jalurnya dikurangi 1 . Maksudnya,jika langsung menuju IP tujuan pada 202.9.85.16(www.eepis-its.edu), dari PC user, maka router yang dilewati hanya sebanyak 1 router.

Berikut tampilannya:

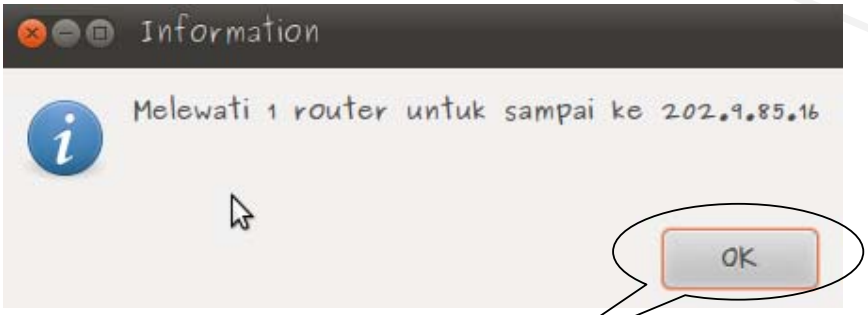
Pilih pada radiolist yang menunjukkan traceroute sehingga,menghasilkan tampilan baru,yaitu



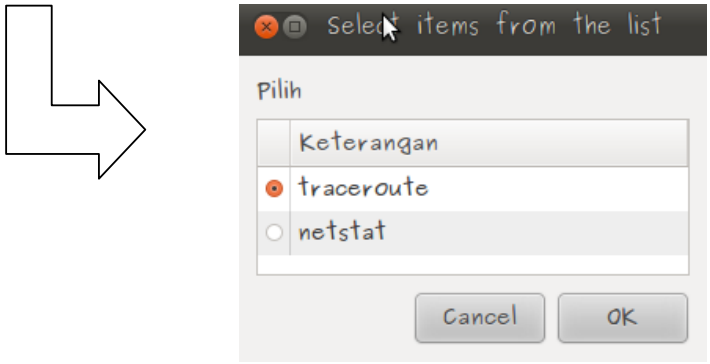
kemudian,masukkan nomer IP yang dituju,misal *202.9.85.16* (koneksi ke www.eepis-its.edu) lalu klik OK



tampilan selanjutnya

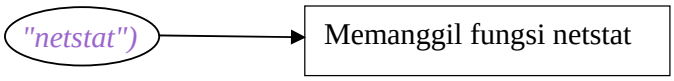


selanjutnya tekan button OK untuk kembali ke menu awal

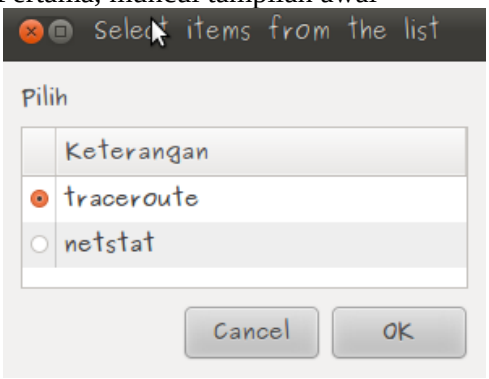


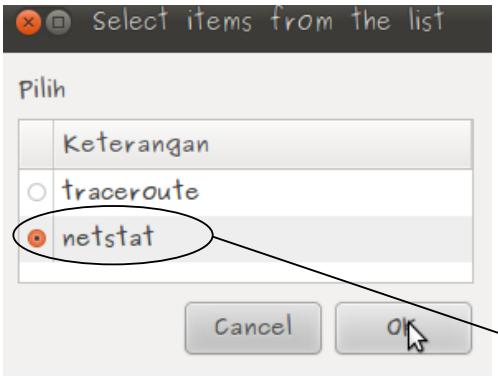
>>>***netstat***

option netstat digunakan untuk mengetahui berapa banyak dan mengetahui PC terkoneksi dengan ditandai state ESTABLISHED dengan jaringan yang kita pilih.



Pertama, muncul tampilan awal





kemudian pilih netstat

netstat -natu > n.sh

Hasil dari perintah ***netstat –natu*** dibelokkan ke file ***n.sh*** dengan tujuan agar data dari netstat tersebut dapat di ***grep***

Netstat -natu adalah option netstat yang digunakan untuk mengetahui apakan antar IP sudah saling terkoneksi, dengan ditandai dengan state ESTABLISHED. Kemudian dari seluruh data hasil perintah netstat –natu yang tersebut dibelokkan ke file baru yaitu n.sh dengan tujuan agar data dari netstat tersebut dapat di grep .

b=(`grep "ESTABLISHED" n.sh`);

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

data dari n.sh yang mengandung state ESTABLISHED tersebut di-grep kemudian disimpan di variable *b*. Sehingga *b* disini berisi data yang mengandung state ESTABLISHED. Kemudian hasilnya di check dengan *case.*
Hasil dari netstat untuk

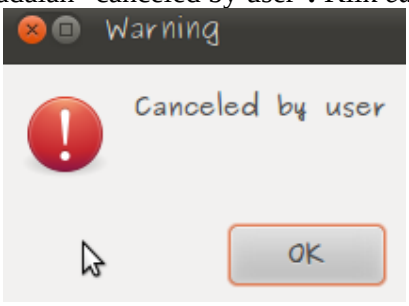
```
case $b in
"")
zenity --warning --text="tidak ada koneksi"

;;
*)

zenity --info --text "telah terhubung dengan
```

sehingga memunculkan tampilan


```
`awk '/ESTABLISHED{ print $5 }' n.sh`
$a"
        balas;
        cek;
        ;;
        esac
;;
*)
  zenity --warning \
       --text="Canceled by user"
        exit 0;
;;
esac
}
```

Mengambil dan mencetak data di kolom ke-5 yang berisi IP dan pada keadaan ESTABLISHED yang ada di file n.sh

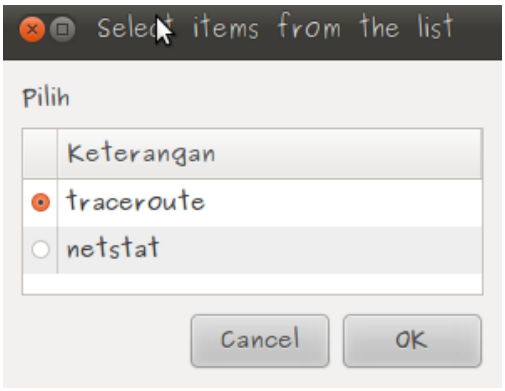
Untuk menutup tampilan menu dari program bash dengan judul ***tugas3.sh,***zenity yang muncul adalah "canceled by user"**.** Klik *cancel* sehingga tampil



program selanjutnya :

cek;

fungsi ini dipanggil kembali, agar pengecheckan ini kembali ke menu utama. Dengan cara klik **OK** . Sehingga memunculkan tampilan seperti di bawah ini



KESIMPULAN

Program ini dibuat untuk mengecheck seberapa banyak router yang digunakan pada PC untuk dapat berkoneksi pada IP yang dituju secara *wireless*