

**Administrasi dan Desain Jaringan**  
Mininet, MiniNAM, OpenFlow, dan PoxController



Oleh :

Ilham Kurnia Wahyudi ( 161402007 )

Ramzi Khauri ( 161402032 )

Muhammad Alisiraj Fachreza Siregar ( 161402062 )

Muhammad Raihan ( 161402066 )

Riki Ramadhan Syahputra ( 161402084 )

Link Github :

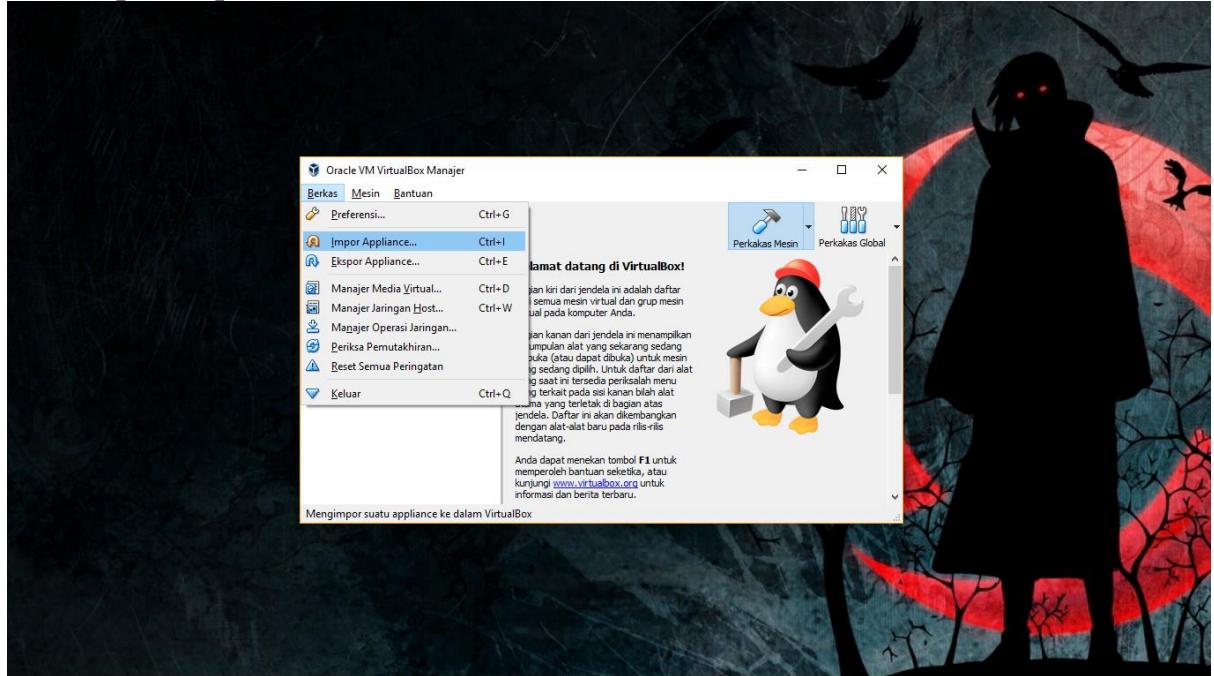
<https://github.com/Onmyoji16/TugasAdministrasiDataJaringan>

Fakultas Ilmu Komputer dan Teknologi Informasi  
Universitas Sumatera Utara  
2018

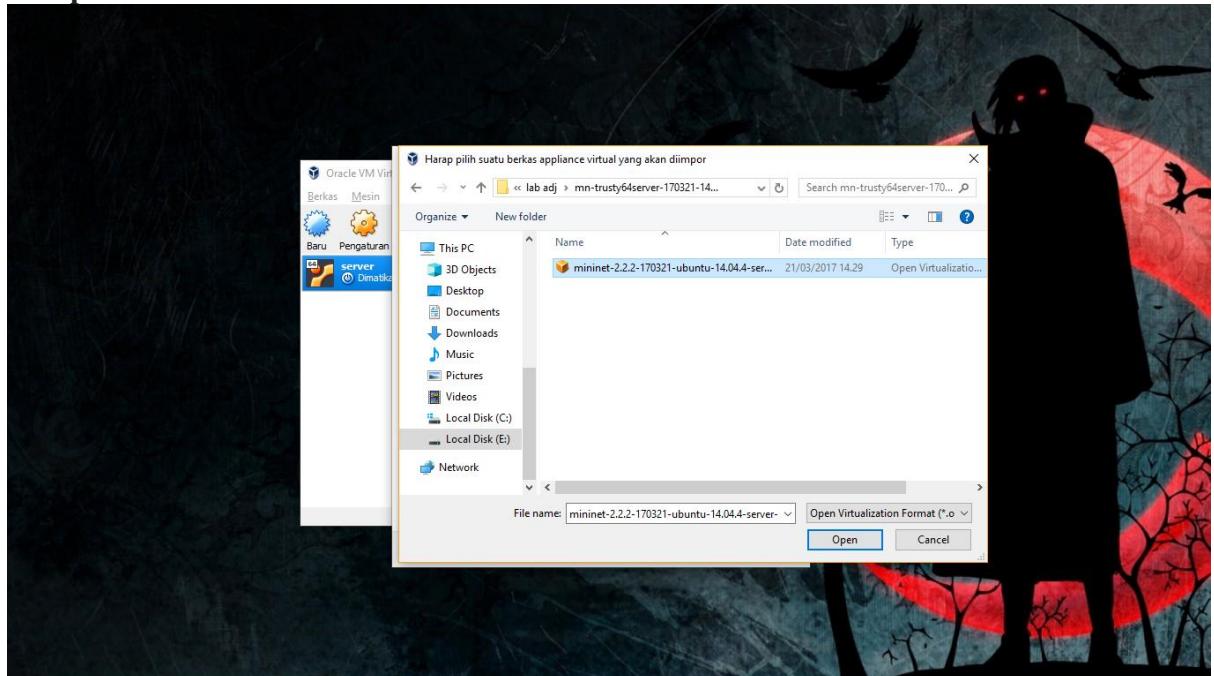
# Mininet

Mininet adalah sebuah simulator jaringan (jaringan virtual) yang dapat membuat sebuah jaringan yang terdiri dari virtual hosts, switches, controllers dan links. Virtual jaringan Mininet menyerupai jaringan fisik yang aslinya. Berikut adalah tahapan-tahapannya :

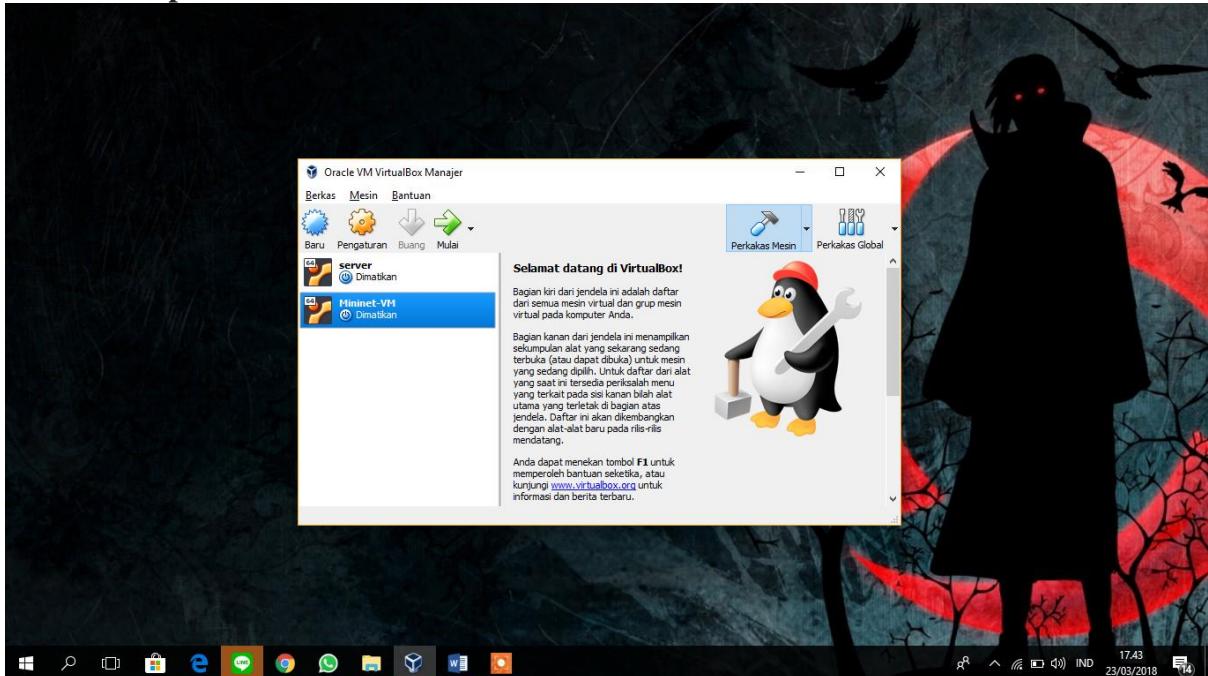
## 1.Persiapan Import



## 2.Import Mininet

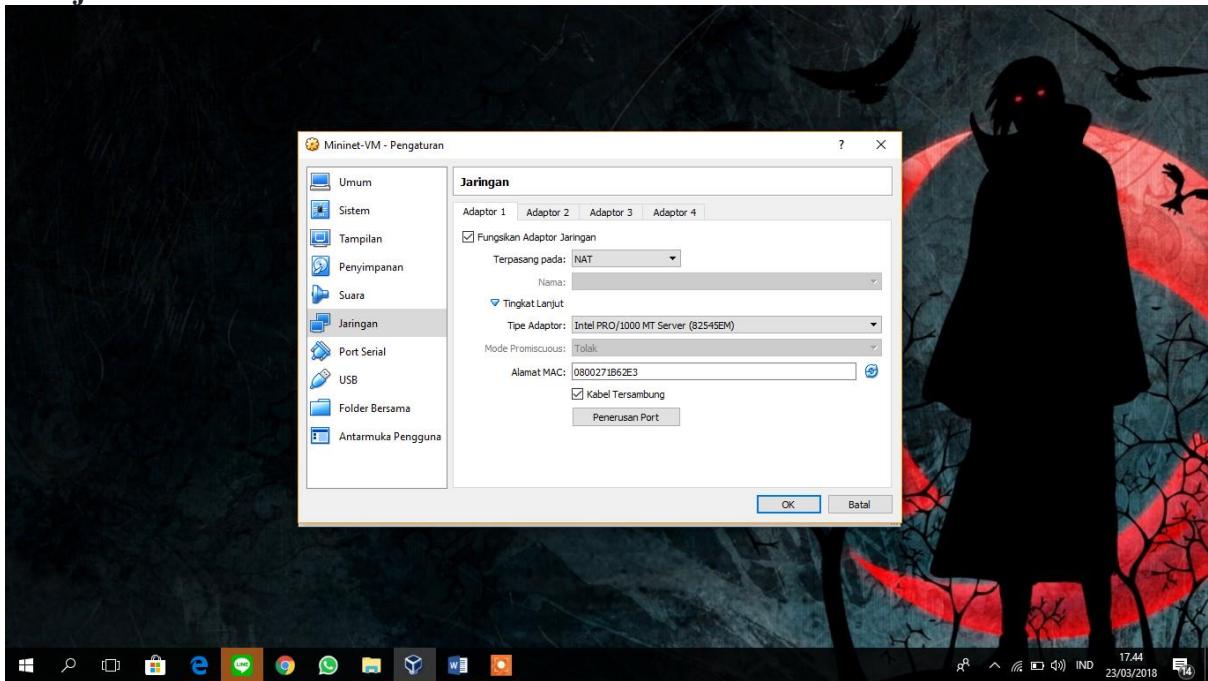


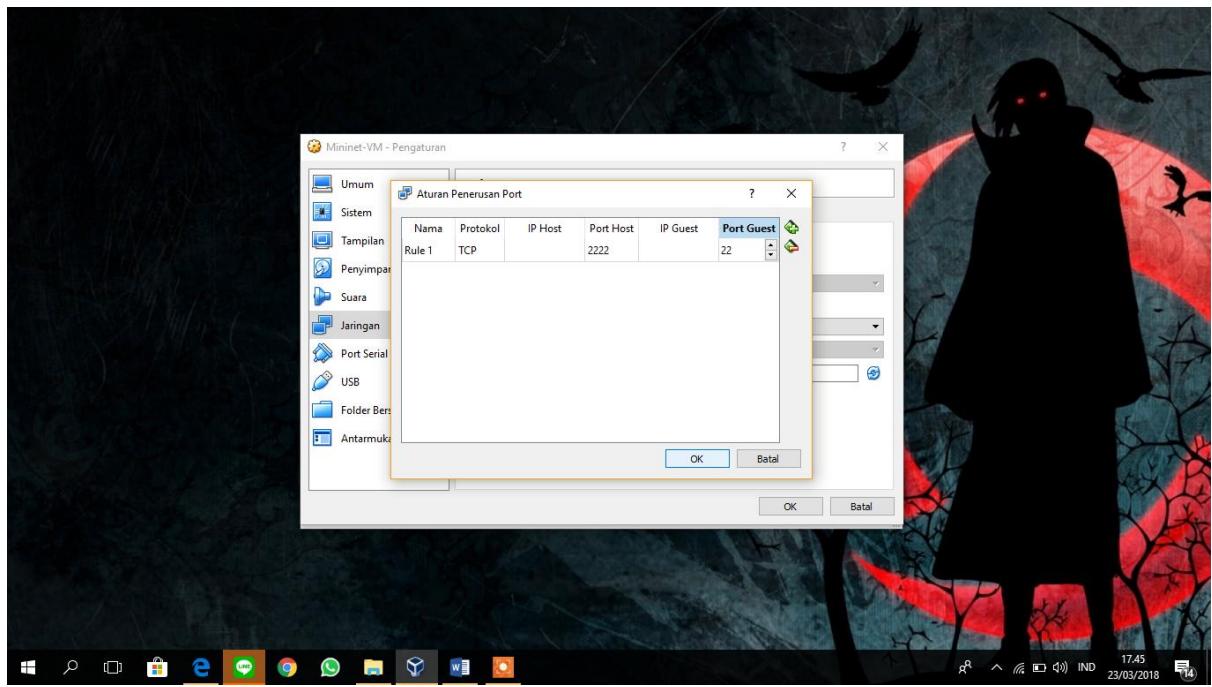
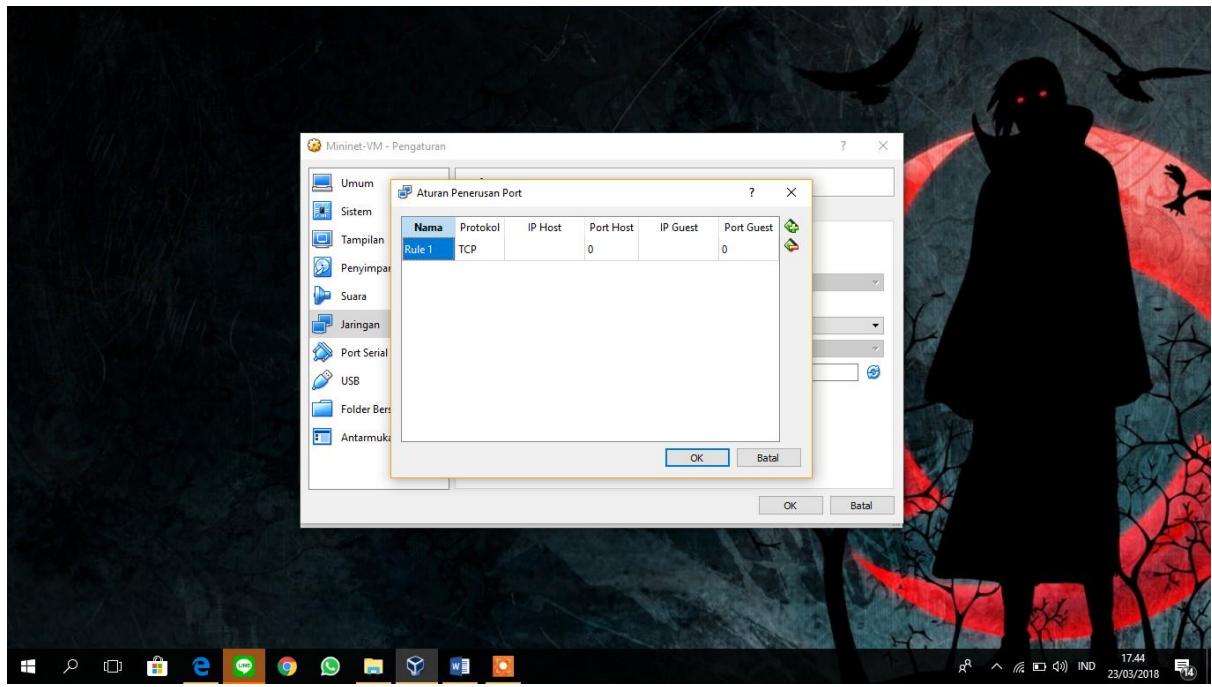
### 3.Sukses Import Mininet



### 4.Melakukan Configure Access

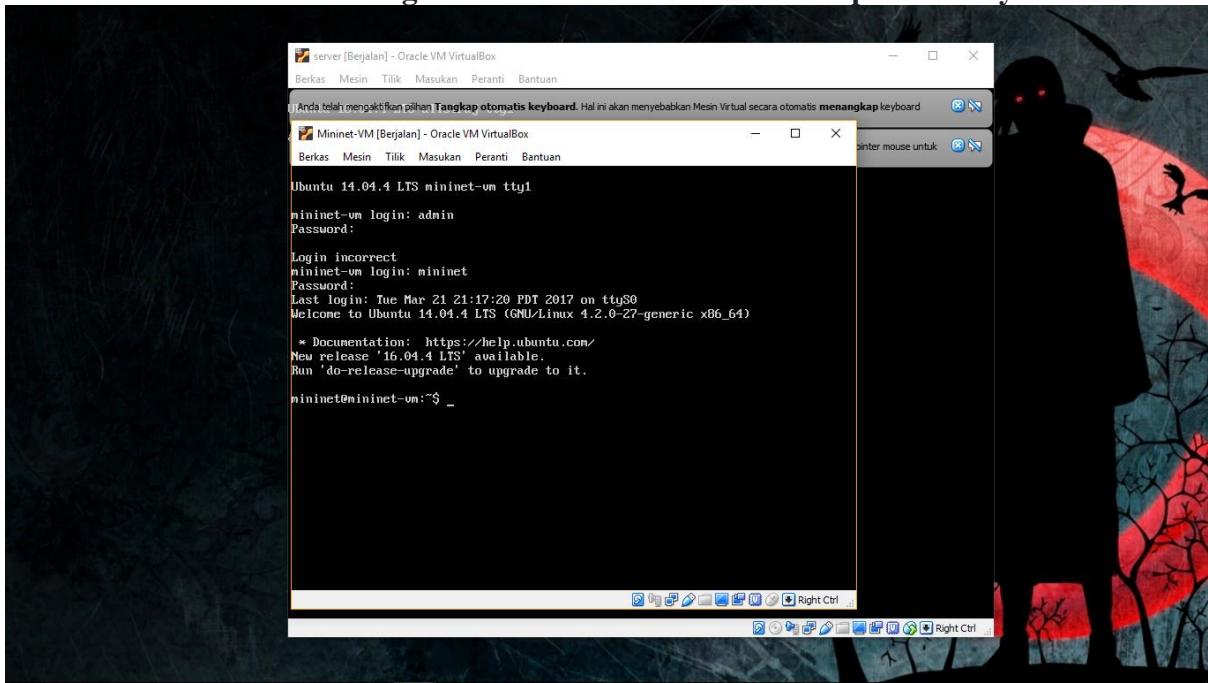
Melakukan setting agar network dapat berkomunikasi sehingga proses bisa dilanjutkan





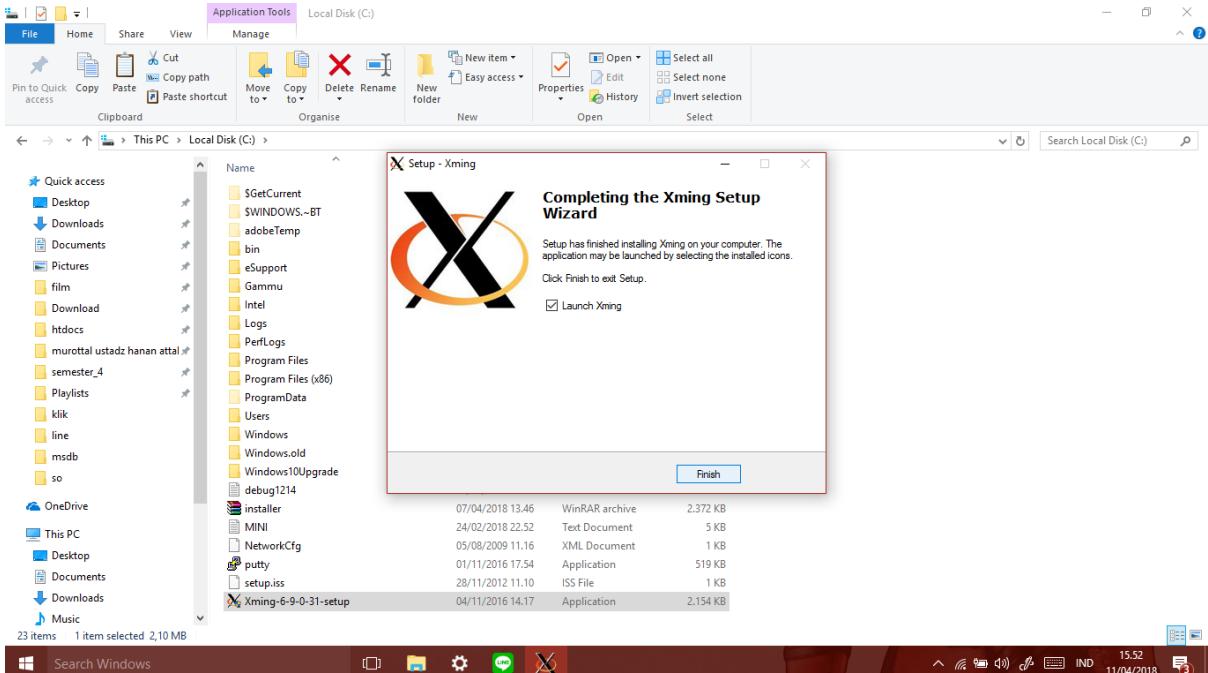
## 5.Menjalankan Mininet

Jalankan Mininet dengan memasukkan username dan passwordnya



## 6.Mengaktifkan Xming

Xming adalah software yang mengimplementasikan X server pada sistem operasi windows. Dengan xming kita dapat mengakses X server dari komputer lain dengan jalur SSH.



```
cmd Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ilhamkwr>cd ..
C:\Users>cd .\mininet@mininet-vm: ~
Using username "mininet".
mininet@localhost's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation: https://help.ubuntu.com/
 New release '16.04.4 LTS' available.
 Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Apr 11 02:02:15 2018
/usr/bin/xauth:  file /home/mininet/.Xauthority does not exist
mininet@mininet-vm:~$ xterm
xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-i
so10646-1'
mininet@mininet-vm:~$
```

## 7.Putty

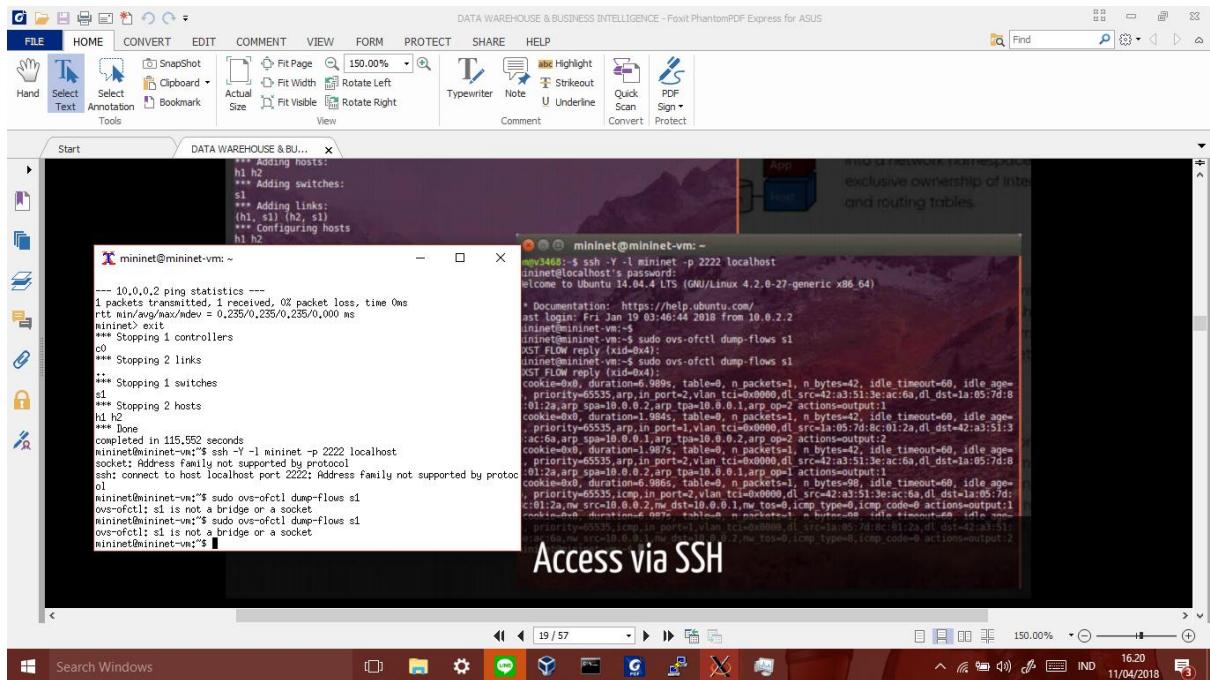
Putty adalah aplikasi open source terminal emulator yang dapat bertindak sebagai klien untuk login SSH,Telnet, dan protokol TCP.

```
cmd Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

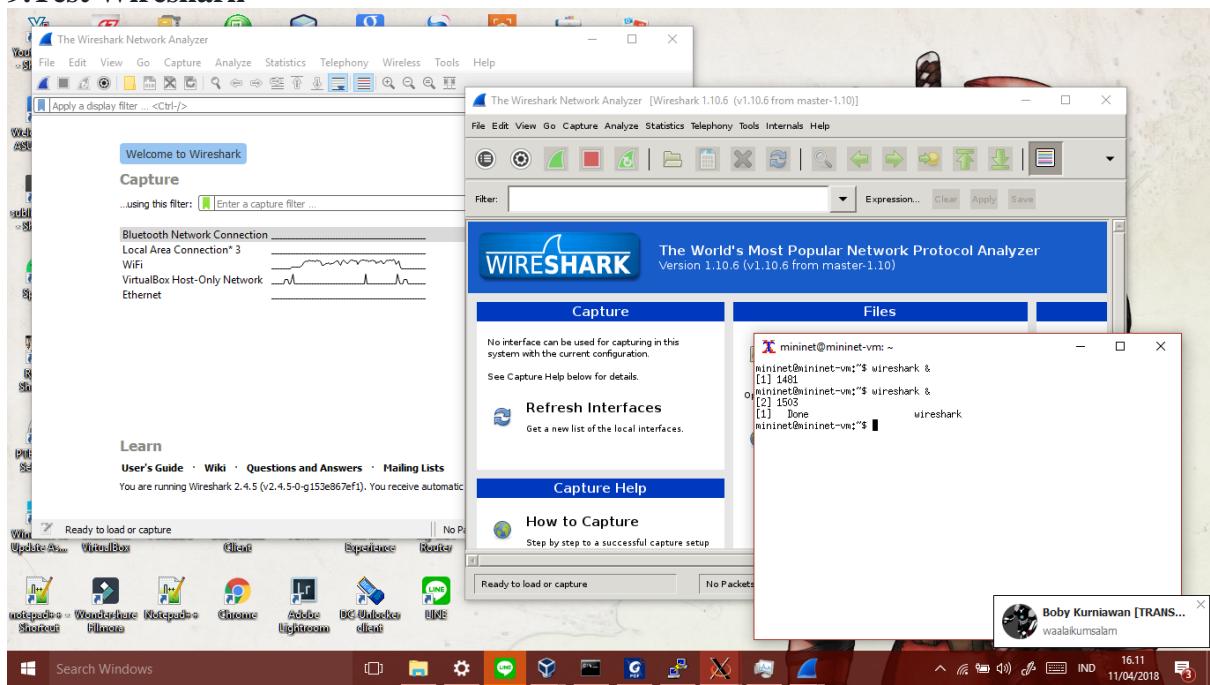
C:\Users\ilhamkwr>cd C:
C:\Users\ilhamkwr>C:
C:\Users\ilhamkwr>cd..
C:\Users>cd..
C:\>putty.exe -X -P 2222 -l mininet localhost
mininet@mininet-vm:~$
```

## 8. Access Via SSH

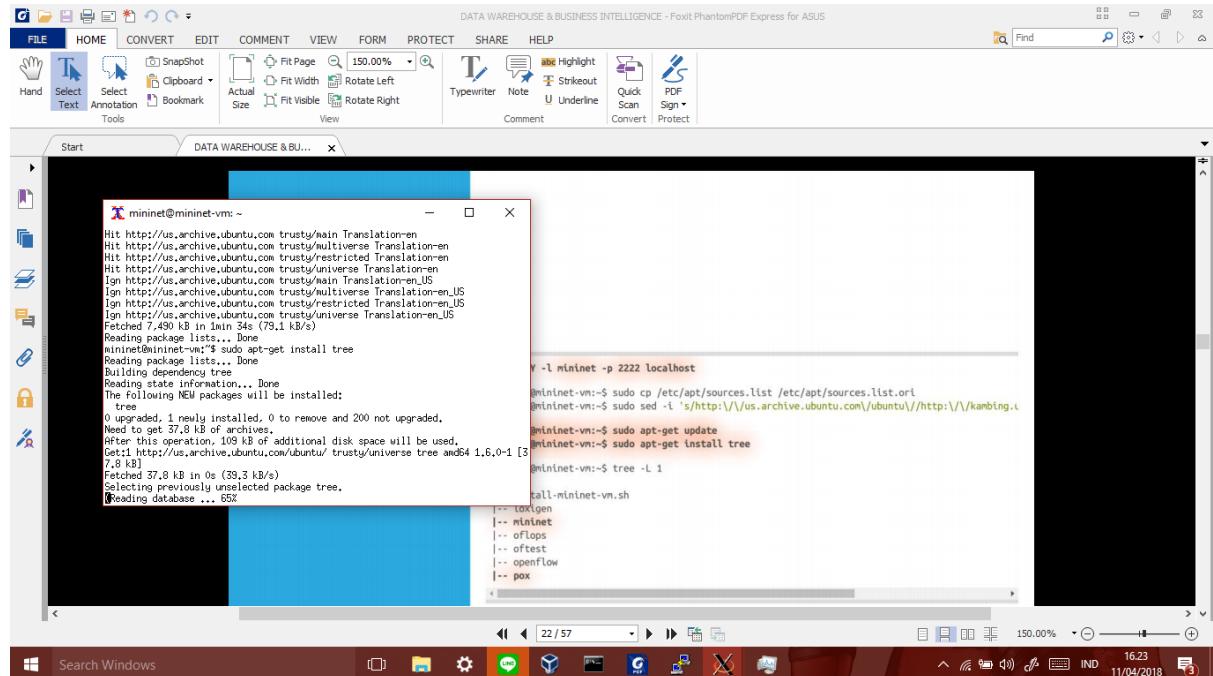
A screenshot of a Windows desktop environment. At the top is a taskbar with icons for File Explorer, Task View, Start, and several pinned applications. The main window is a PDF viewer titled "DATA WAREHOUSE & BUSINESS INTELLIGENCE - Foxit PhantomPDF Express for ASUS". Inside the PDF viewer, there are two terminal windows. The left terminal window is titled "mininet@mininet-vm:~" and shows the command-line interface for setting up a mininet network on an Ubuntu virtual machine. It includes commands like "ssh -Y -L mininet: -p 2222 localhost", "mininet@localhost's password: Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86\_64)", and various mininet configuration steps. The right terminal window is titled "mininet@mininet-vm:~" and shows host configuration commands such as "sudo mn", "mininet@mininet-vm:~\$ sudo mn \*\*\* Creating network \*\*\* Adding controller \*\*\* Adding hosts: h1 h2 \*\*\* Adding switches: s1 \*\*\* Adding links: (h1, s1) (h2, s1) \*\*\* Configuring hosts h1 h2 \*\*\* Starting controller c0 \*\*\* Starting 1 switches s1 \*\*\* Starting CLI: mininet> h1 ping -c1 h2 PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data. 64 bytes from 10.0.0.2: icmp\_seq=1 ttl=64 time=1.49 ms --- 10.0.0.2 ping statistics --- cookie=0x0, duration=6.989ms, table=0, n\_packets=1, n\_bytes=42, idle\_timeout=69, idle\_appeared=0, last\_rx=11:59:59.121919, last\_tx=11:59:59.121919, dl\_dif=1, dl\_dif\_ns=725". To the left of the terminal windows is a vertical toolbar containing icons for Snapshot, Clipboard, Typewriter, and other tools. The status bar at the bottom shows "19 / 57", "150.00%", and "3 new notifications".



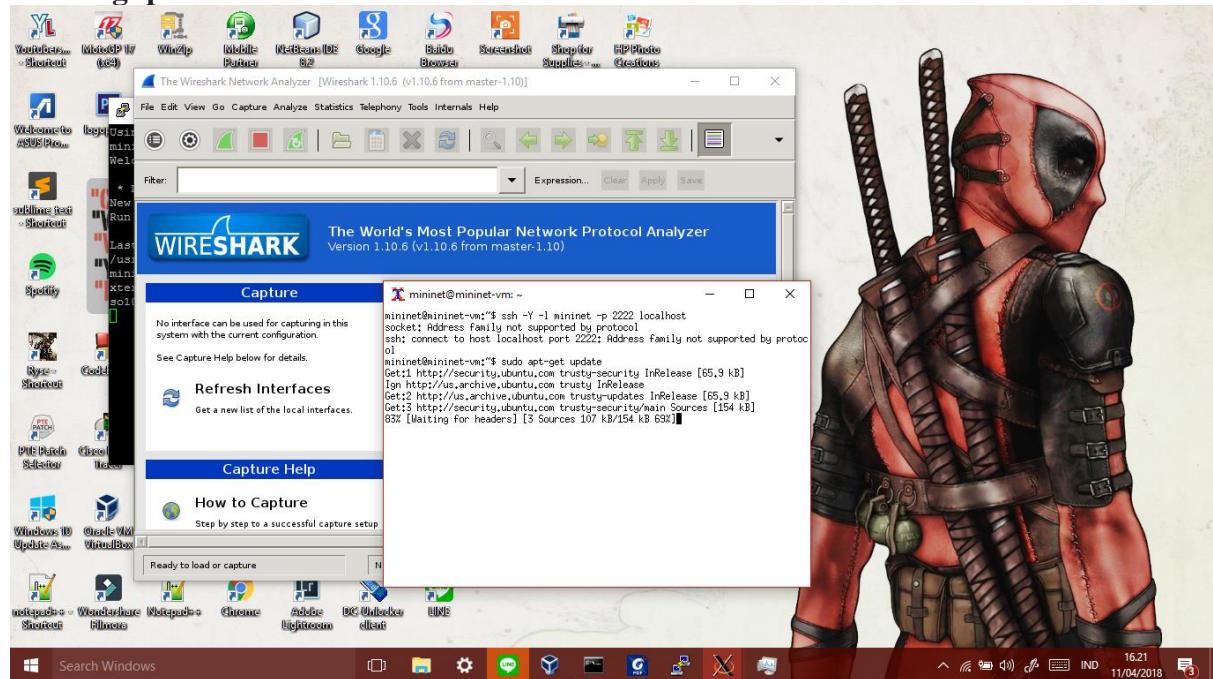
## 9. Test Wireshark



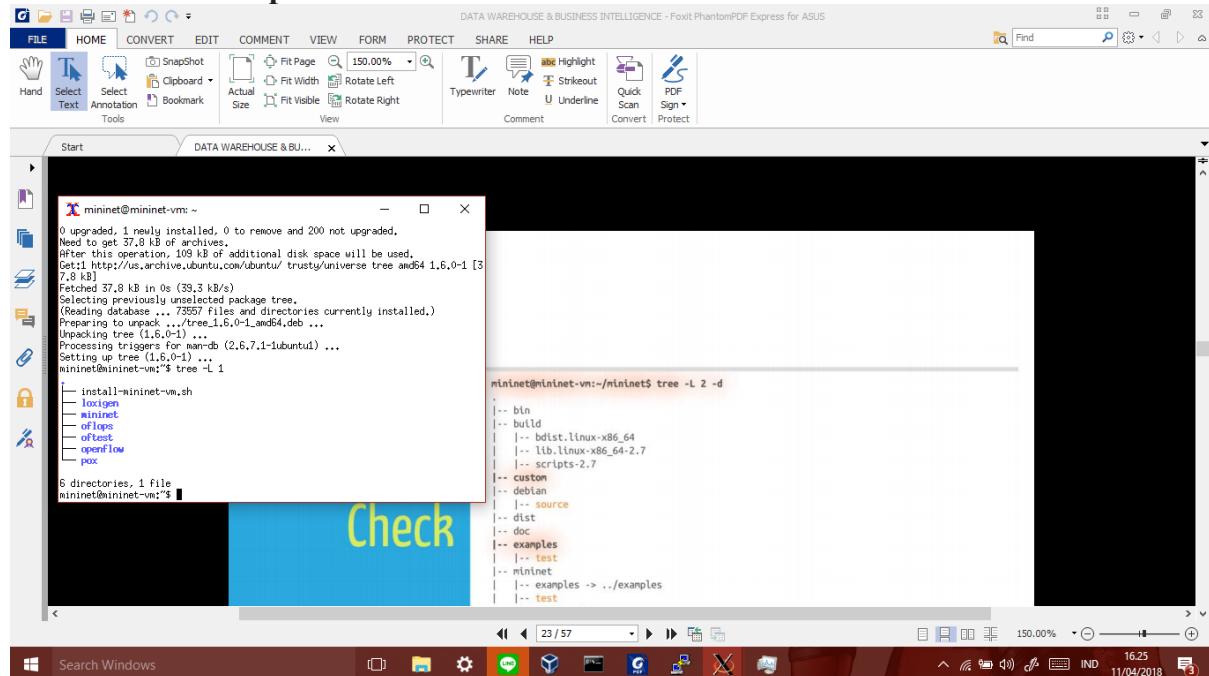
## 10.Melakukan Install Tree



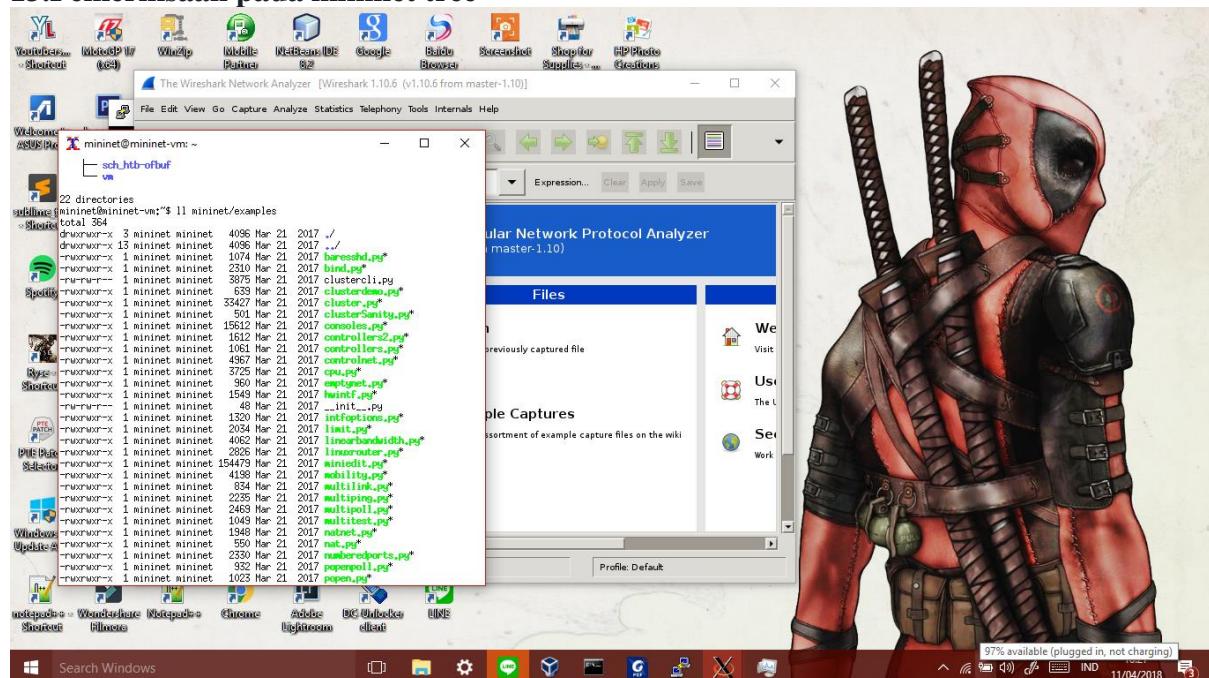
## 11.Mengupdate Tree



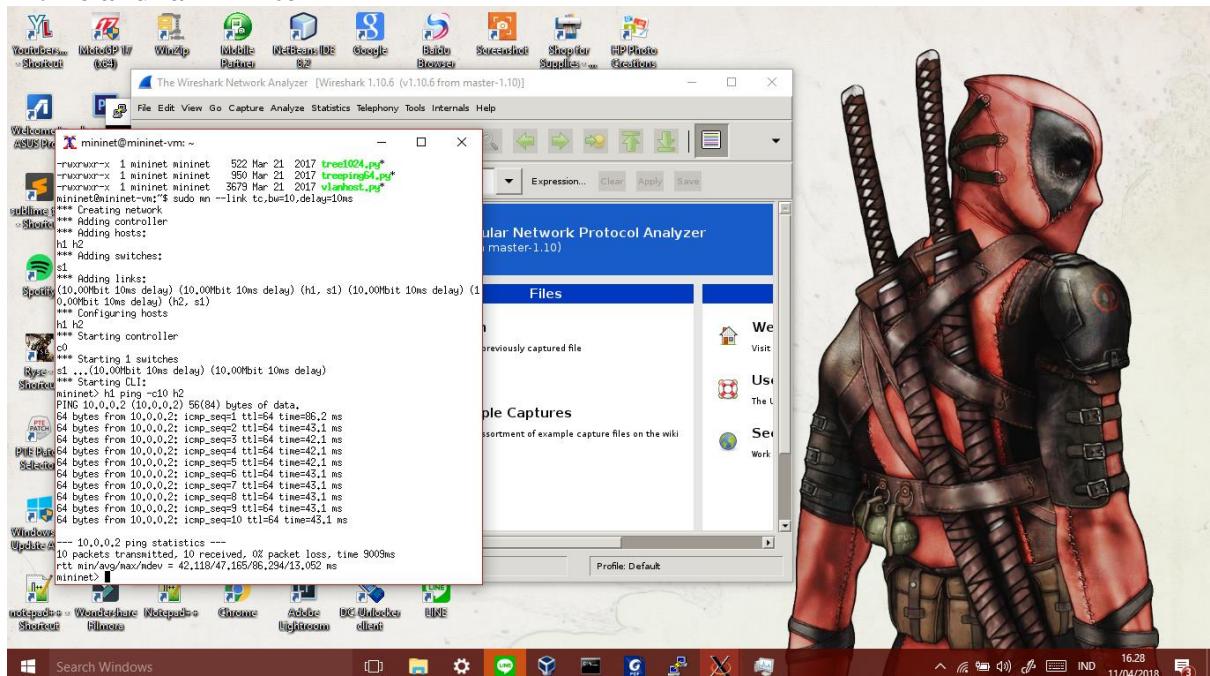
## 12. Melakukan cek pada tree-L1



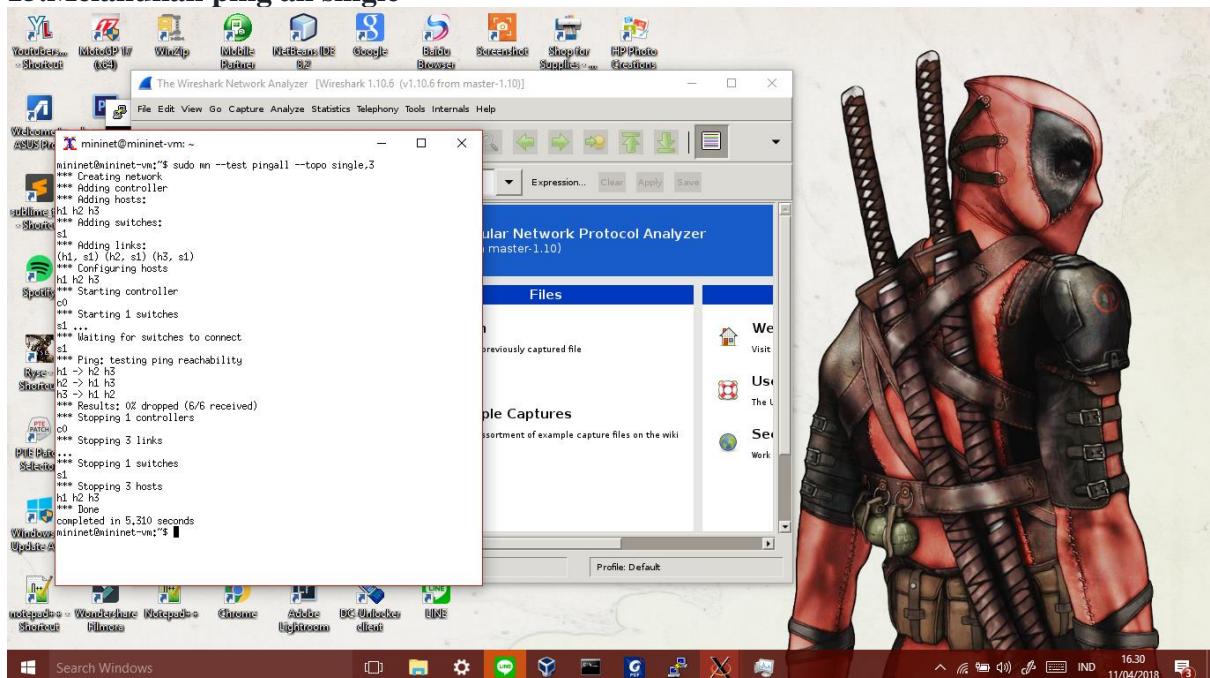
## 13. Pemeriksaan pada mininet tree



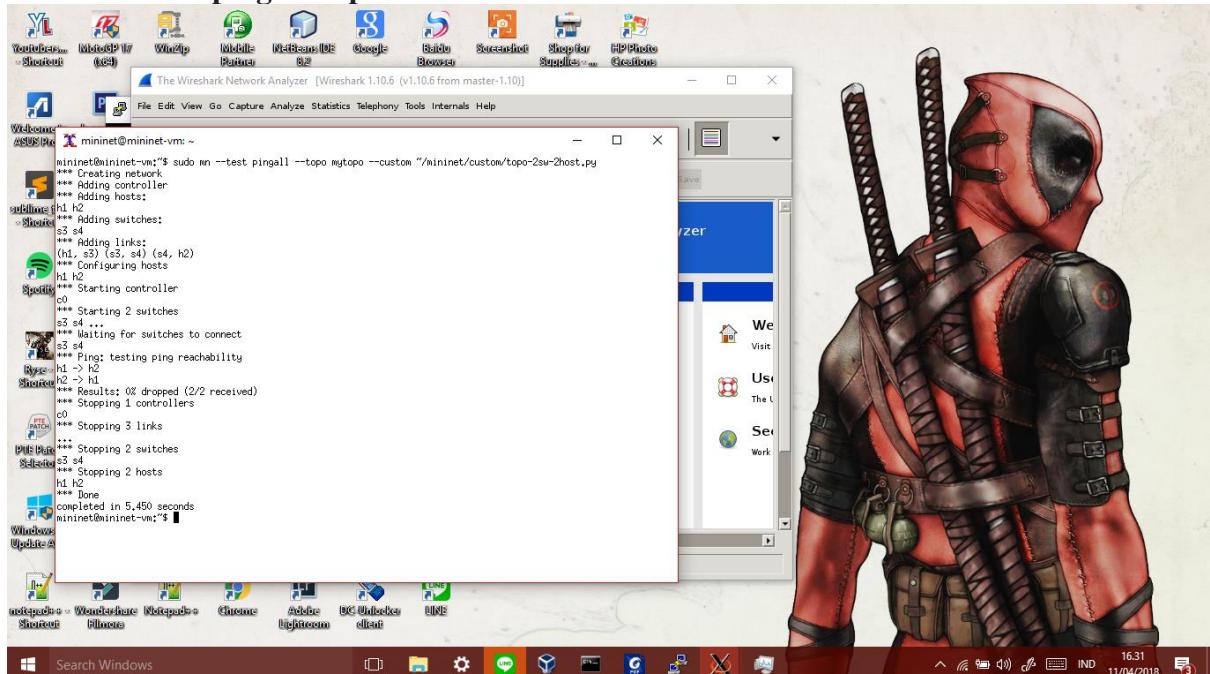
## 14.Melakukan link tc



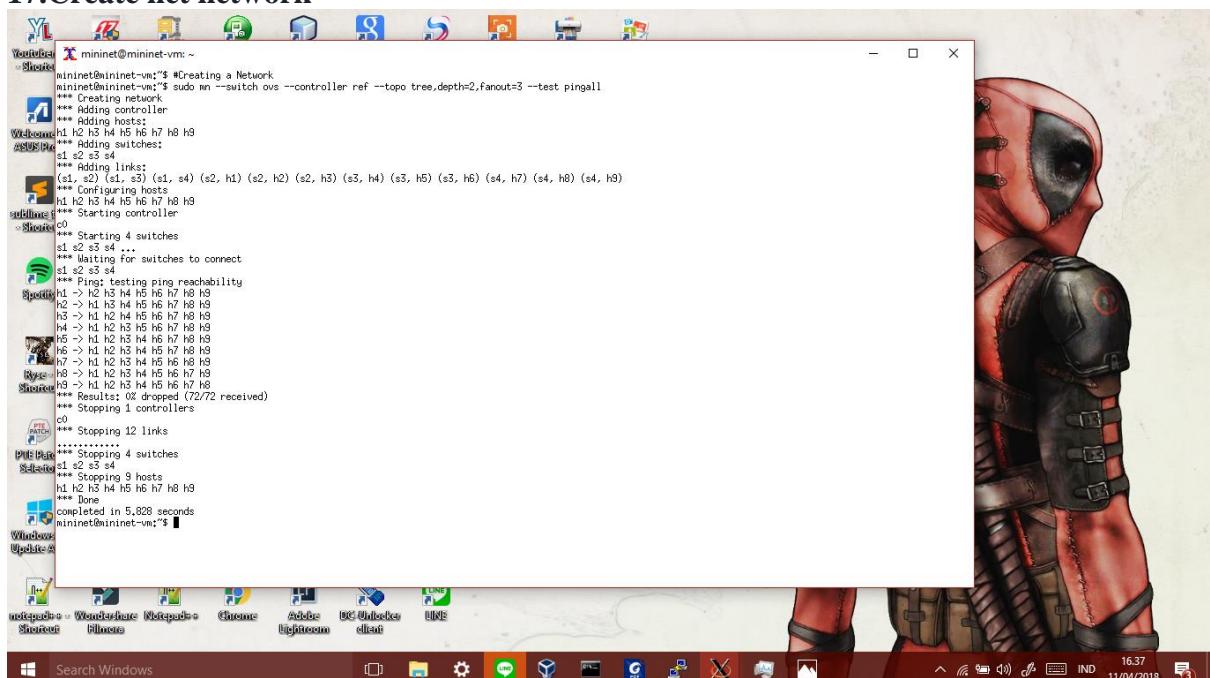
## 15.Melakukan ping all single



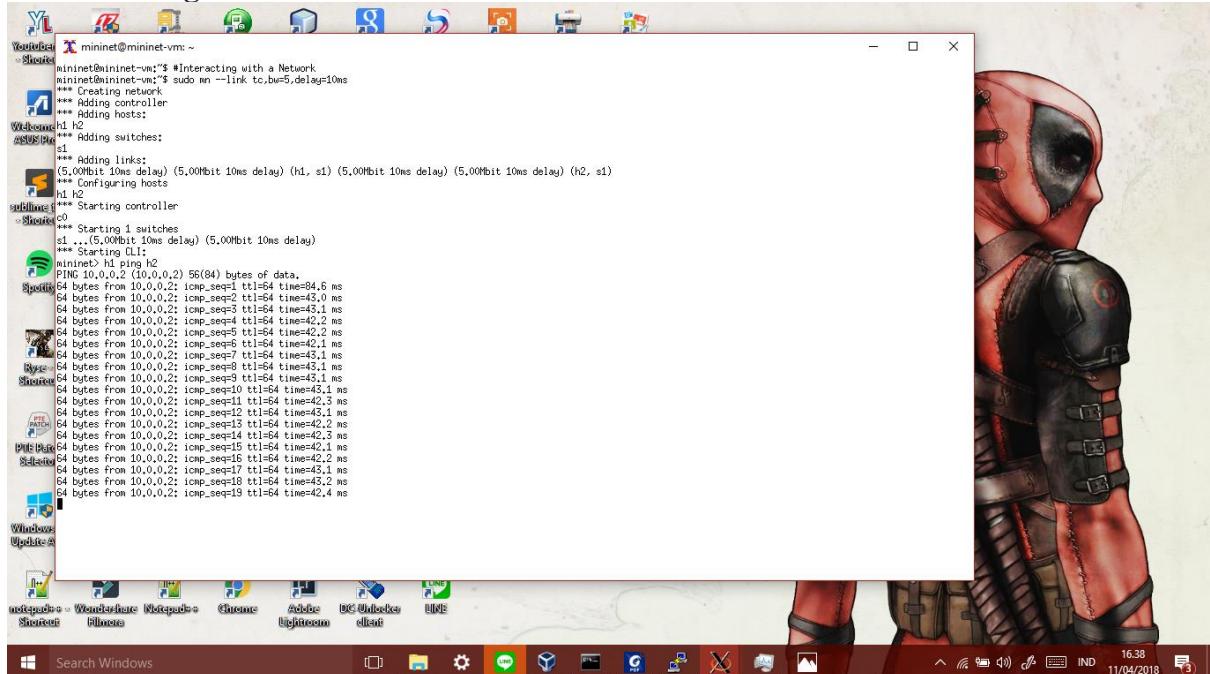
## 16.Melakukan ping all topo



## 17.Create net network



## 18. Interacting with a network



A screenshot of a Windows desktop environment. In the center is a terminal window titled 'mininet@mininet-vm: ~'. The terminal displays the following text:

```
mininet@mininet-vm:~$ # interacting with a Network
mininet@mininet-vm:~$ sudo mn --link tc,bw=5,delay=10ms
*** Creating network
*** Adding controller
  id: c0
*** Adding hosts:
  id: h1
  id: h2
  id: s1
*** Adding switches:
  id: s1
*** Adding links:
  (5.0Mbit 10ms delay) (5.0Mbit 10ms delay) (h1, s1) (5.0Mbit 10ms delay) (5.0Mbit 10ms delay) (h2, s1)
  Configuring hosts
  id: h2
  id: h1
  id: s1
s1:*** Starting controller
  id: c0
  *** Starting 1 switches
  id: s1
  *** Starting 1 hosts
  id: h1,h2
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=84.6 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=43.1 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=43.1 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=42.1 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=42.2 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=42.1 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=43.1 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=43.1 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=43.1 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=43.3 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=42.3 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=43.1 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=42.2 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=42.1 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=42.1 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=42.2 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=43.1 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=43.2 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=42.4 ms
```

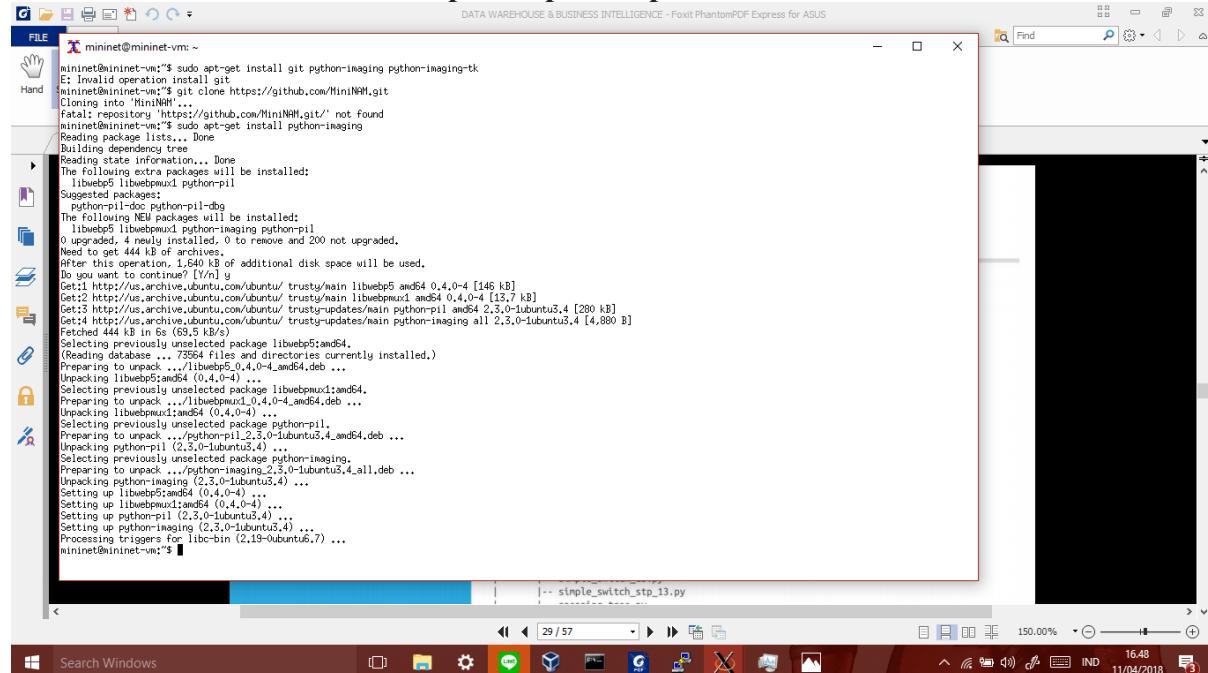
The desktop background features a Deadpool illustration. The taskbar at the bottom shows various pinned icons including OneDrive, Photos, Mail, News, Photos, Adobe Lightroom, AVG Utilities, LINE, and a folder icon. The system tray shows the date (11/04/2018), time (16:38), battery level, signal strength, and other standard icons.

# Mininam

Merupakan alat yang sangat bagus untuk debug protokol jaringan dimana ia berfungsi sebagai penyedia antarmuka pengguna grafis yang memungkinkan modifikasi dinamis preferensi dan filter paket yang sangat membantu pada poros memahami konsep jaringan. Langkah-langkah pada mininam adalah :

## 1. Mininam Setup

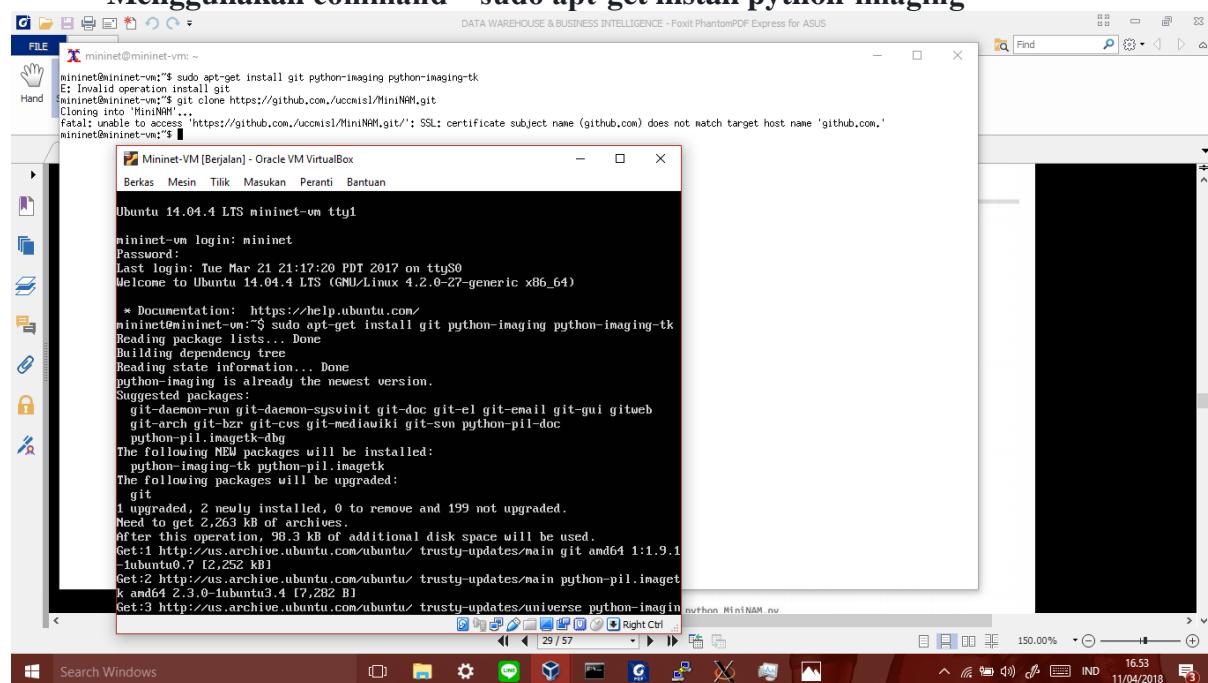
Install menggunakan command “ git clone <https://github.com/MiniNam.git> ” , dan bila berhasil terinstall akan tampil tampilan seperti berikut



```
mininet@mininet-vm:~$ sudo apt-get install git python-imaging python-imaging-tk
E: Invalid operation install git
mininet@mininet-vm:~$ git clone https://github.com/MiniNam.git
Cloning into 'Mininam'...
fatal: repository 'https://github.com/MiniNam.git/' not found
mininet@mininet-vm:~$ sudo apt-get install python-imaging
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
libwebp5 libwebpmux1 python-pil
Suggested packages:
python-pil-doc python-pil-dbg
The following NEW packages will be installed:
libwebp5 libwebpmux1 python-imaging python-pil
0 upgraded, 4 newly installed, 0 to remove and 200 not upgraded,
Need to get 444 kB of archives.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/main libwebp5 amd64 0.4.0-4 [146 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty/main libwebpmux1 amd64 0.4.0-4 [13.7 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main python-pil amd64 2.3.0-1ubuntu3.4 [280 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main python-imaging all 2.3.0-1ubuntu3.4 [4,880 B]
Fetched 444 kB in 0s (935 kB/s)
Selecting previously unselected package libwebp5:amd64.
(Reading database ... 7396 files and directories currently installed.)
Preparing to unpack .../libwebp5_0.4.0-4_amd64.deb ...
Unpacking libwebp5:amd64 (0.4.0-4) ...
Selecting previously unselected package libwebpmux1:amd64.
Preparing to unpack .../libwebpmux1_0.4.0-4_amd64.deb ...
Unpacking libwebpmux1:amd64 (0.4.0-4) ...
Selecting previously unselected package python-pil.
Preparing to unpack .../python-pil_2.3.0-1ubuntu3.4_amd64.deb ...
Unpacking python-pil (2.3.0-1ubuntu3.4) ...
Selecting previously unselected package python-imaging.
Preparing to unpack .../python-imaging_2.3.0-1ubuntu3.4_all.deb ...
Unpacking python-imaging (2.3.0-1ubuntu3.4) ...
Setting up libwebp5:amd64 (0.4.0-4) ...
Setting up libwebpmux1:amd64 (0.4.0-4) ...
Setting up python-pil (2.3.0-1ubuntu3.4) ...
Setting up python-imaging (2.3.0-1ubuntu3.4) ...
Processing triggers for libc-bin (2.19-0ubuntu6.7) ...
mininet@mininet-vm:~$
```

## 2. Mininam install tkinter

Menggunakan command “ sudo apt-get install python-imaging ”



```
mininet@mininet-vm:~$ sudo apt-get install git python-imaging python-imaging-tk
E: Invalid operation install git
mininet@mininet-vm:~$ git clone https://github.com/ucmisi/MiniNam.git
Cloning into 'Mininam'...
fatal: unable to access 'https://github.com/ucmisi/MiniNam.git/': SSL: certificate subject name (github.com) does not match target host name 'github.com.'
mininet@mininet-vm:~$
```

```
Mininet-VM [Berjalan] - Oracle VM VirtualBox
Berkas Mesin Tisk Masukan Peranti Bantuan

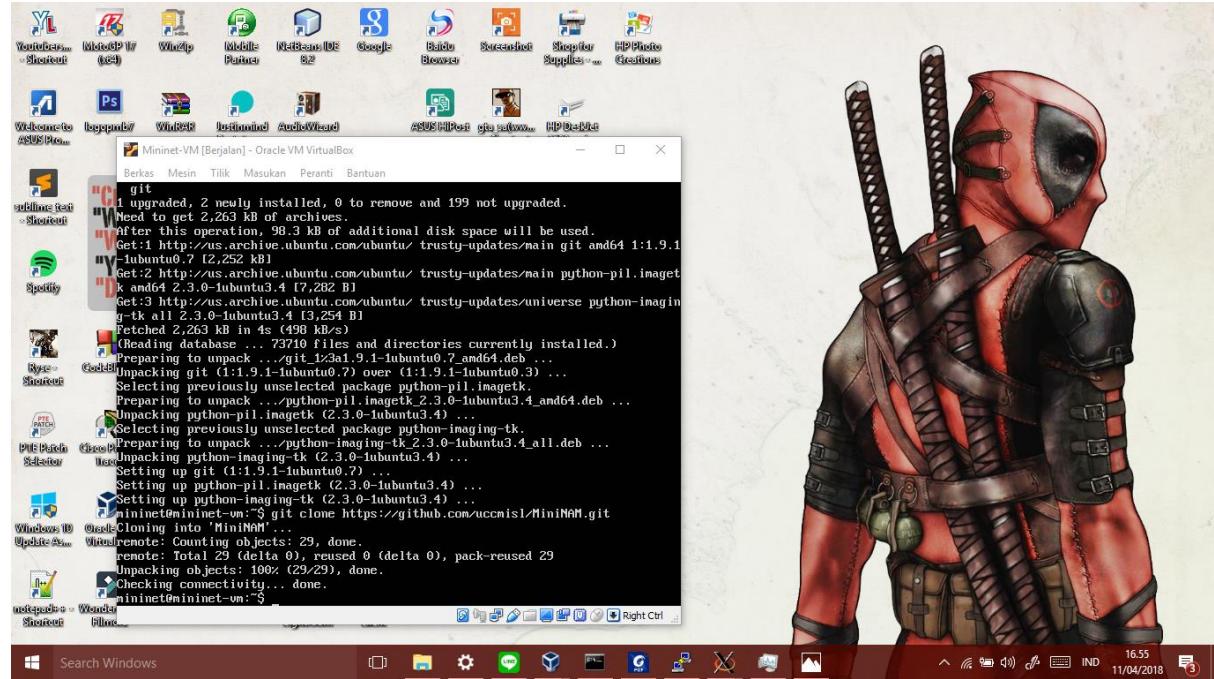
Ubuntu 14.04.4 LTS mininet-vm ttys1

mininet-vm login: mininet
Password:
Last login: Tue Mar 21 21:17:20 PDT 2017 on ttys0
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation: https://help.ubuntu.com/
mininet@mininet-vm:~$ sudo apt-get install git python-imaging python-imaging-tk
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-imaging is already the newest version.
Suggested packages:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitweb
  git-arch git-bzr git-cvs git-mediawiki git-svn python-pil-doc
  python-pil.imagetk-dbg
The following NEW packages will be installed:
  python-imaging-tk python-pil.imagetk
The following packages will be upgraded:
  git
1 upgraded, 2 newly installed, 0 to remove and 199 not upgraded.
Need to get 2,263 kB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main git amd64 1:1.9.1-1ubuntu0.7 [2,252 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main python-pil.imagetk amd64 2.3.0-1ubuntu3.4 17,282 B
Get:3 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/universe python-imaging-all amd64 2.3.0-1ubuntu3.4 17,282 B
mininet@mininet-vm:~$
```

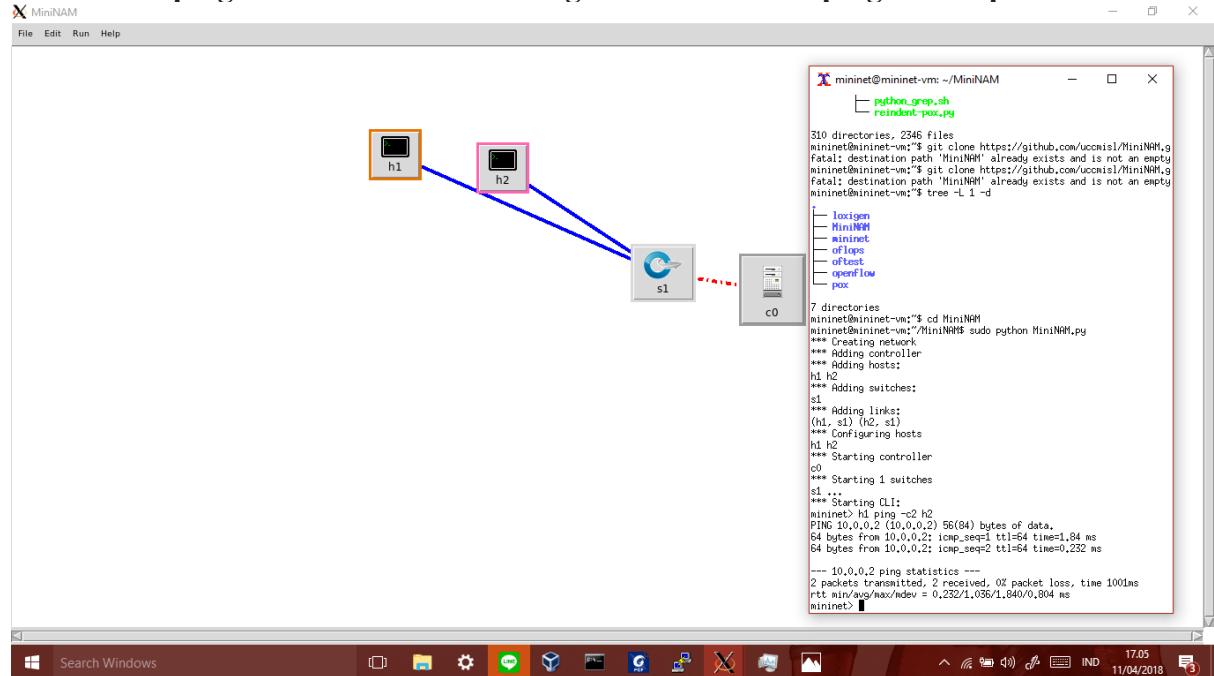
### 3. MiniNam install python imaging

Penggunaan command “ sudo apt-get-install git python-imaging-python-imaging-tk”



### 4. MiniNam Ping

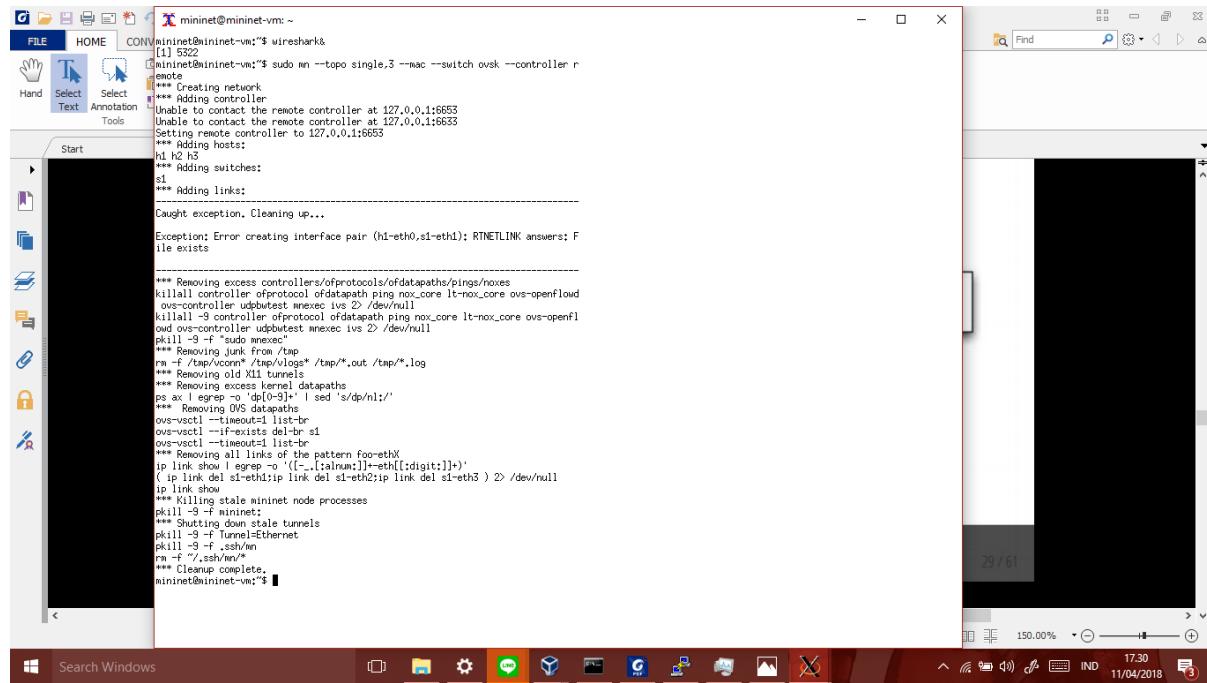
Dengan cara akses ke dalam mininam lalu berikan command ‘ sudo python MiniNam.py’ maka gui nya mininam muncul dengan sebuah setting bawaan , lalu lakukanlah ping dari host 1 ke host 2 dengan command ‘ h1 ping -c2 h2’ pada cli mininet.



# OpenFlow

Adalah sebuah protokol yang memungkinkan pengaturan penjaluran dan pengiriman packet ketika melalui sebuah switch dimana setiap switch hanya berfungsi meneruskan packet yang lewat ke port yang sesuai tanpa dapat membedakan type protokol data yang dikirimkan. Tahap-tahapnya sebagai berikut.

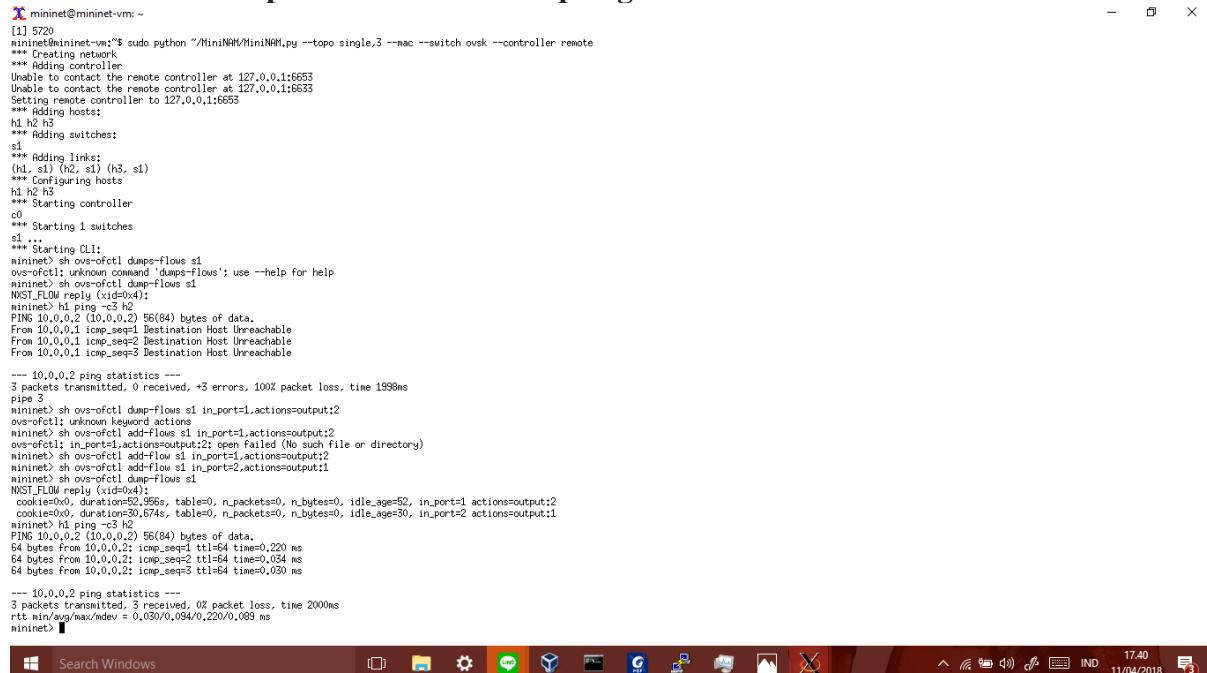
## 1.Membuat 3 host dan 1 switch



```
mininet@mininet-vm: ~
[1] 5222
mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
Caught exception. Cleaning up...
Exception: Error creating interface pair (h1-eth0,s1-eth1); RTNETLINK answers: F
ile exists

*** Removing excess controllers of protocol ofdtdatapath/pings/noxes
killall controller ofprotoctl ofdtdatapath ping nox_core lt-nox_core ovs-openflow
ovs-controller ubdbtest nxecc ixs 2>/dev/null
killall -9 controller ofprotoctl ofdtdatapath ping nox_core lt-nox_core ovs-openfl
out ovs-controller ubdbtest nxecc ixs 2>/dev/null
rm -rf /tmp/vconn* /tmp/vlogs* /tmp/*,out /tmp/*,log
*** Removing old XLL tunnels
*** Removing excess kernel datapaths
ps ax | egrep '(dp|dpctl) | sed 's/dp/nl;/
' | awk '{print $1}' | xargs rmmod
*** Removing OVS datapaths
ovs-vctl --timeout=1 list-br
ovs-vctl --if-exists del-br s1
ovs-vctl --timeout=2 list-br
*** Removing all links of the pattern foo-ethX
ip link del $(grep -o "[\.:][:alnum:]-eth[0-9]:[\.:][:alnum:]"`ip link del s1-eth2`ip link del s1-eth3` ) 2>/dev/null
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet
pkill -9 -f kernel-tunnels
pkill -9 -f kernel-Ethernet
pkill -9 -f sshd
rm -rf ~/.ssh/known_hosts
*** Cleanup complete.
mininet@mininet-vm:~$
```

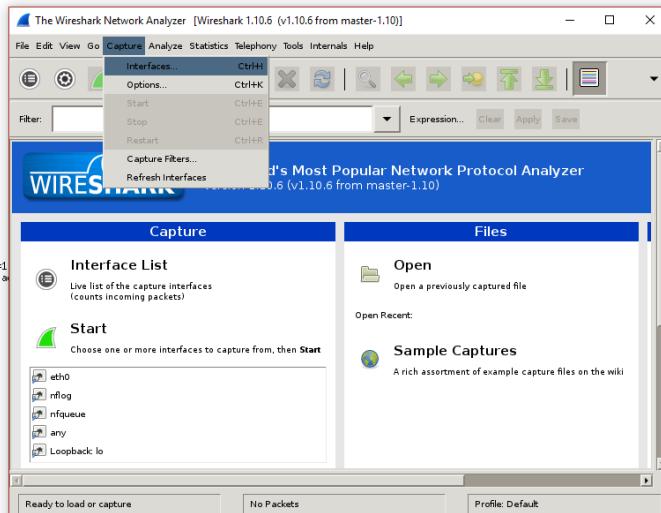
## 2.Menambahkan openswitch flow ke tropologi



```
mininet@mininet-vm: ~
[1] 5720
mininet@mininet-vm:~$ sudo python ~/Mininet/MininAM.py --topo single,3 --mac --switch ovsk --controller remote
*** Creating network
*** Remote controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI...
mininet> sh ovs-ofctl dump-flows s1
ovs-ofctl: unknown command 'dump-flows'; use --help for help
mininet> sh ovs-ofctl dump-flows s1
NST_FLOW reply (xid=4):
PING 10.0.0.2 icmp_seq=1 ttl=64 56(84) bytes of data.
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 1998ms
ping: S
mininet> sh ovs-ofctl dump-flows s1 in_port=1,actions=output:2
ovs-ofctl: unknown keyword actions
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:2
ovs-ofctl: add-flow:2: error: '2' open failed (No such file or directory)
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:1
mininet> sh ovs-ofctl add-flow s1 in_port=2,actions=output:1
mininet> sh ovs-ofctl dump-flows s1
NST_FLOW reply (xid=4):
cookie=0xb0, duration=39.956s, table=0, n_packets=0, n_bytes=0, idle_age=52, in_port=1 actions=output:2
cookie=0xb1, duration=39.674s, table=0, n_packets=0, n_bytes=0, idle_age=30, in_port=2 actions=output:1
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.220 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.034 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.030 ms
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.030/0.094/0.220/0.089 ms
mininet>
```

### 3.Membuka wireshark dengan perintah “sudo wireshark &”

```
mininet@mininet-vm: ~
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1
*** Starting CLI:
mininet> sh ovs-ofctl dump-flows s1
NOMT_FLOW reply (xid=0x4):
mininet> sh ovs-ofctl dump-flows s1 in_port=1,actions=output:2
ovs-ofctl: unknown keyword actions
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:2
ovs-ofctl: in_port=1,actions=output:2: open failed (No such file or directory)
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:2
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:1
mininet> sh ovs-ofctl dump-flows s1
NOMT_FLOW reply (xid=0x4):
cookie=0x0, duration=15.143s, table=0, n_packets=0, n_bytes=0, idle_age=15, in_port=1
cookie=0x0, duration=4.103s, table=0, n_packets=0, n_bytes=0, idle_age=4, in_port=2 a
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.206 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.033 ms
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.033/0.093/0.206/0.079 ms
mininet> sudo wireshark
*** [Unknown command: sudo wireshark]
mininet> e*** Stopping 1 controllers
c0
*** Stopping 3 links
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
[1]+ Done wireshark
mininet@mininet-vm:~$ sudo wireshark
[1] 6793
mininet@mininet-vm:~$
```

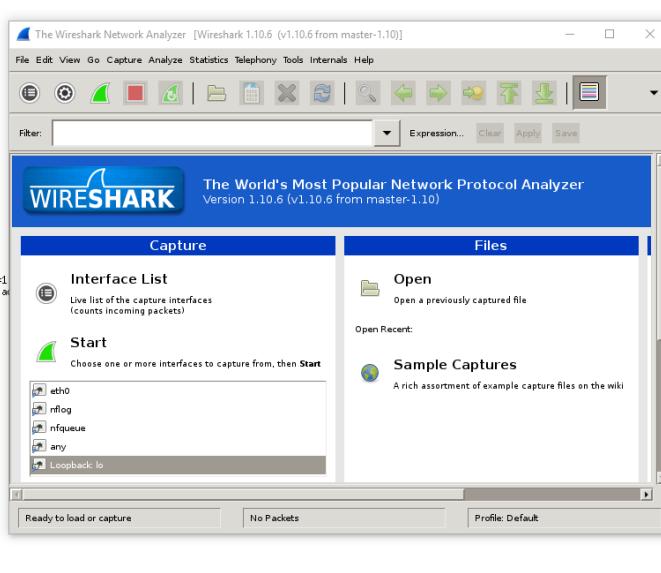


### 4.Membuka capture interface dan memilih io lalu start

```
Wireshark: Capture Interfaces
any none 2944 10
lo 127.0.0.1 1380 6

Help Start Stop Options Close

*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1
*** Starting CLI:
mininet> sh ovs-ofctl dump-flows s1
NOMT_FLOW reply (xid=0x4):
mininet> sh ovs-ofctl dump-flows s1 in_port=1,actions=output:2
ovs-ofctl: unknown keyword actions
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:2
ovs-ofctl: in_port=1,actions=output:2: open failed (No such file or directory)
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:2
mininet> sh ovs-ofctl add-flow s1 in_port=2,actions=output:1
mininet> sh ovs-ofctl dump-flows s1
NOMT_FLOW reply (xid=0x4):
cookie=0x0, duration=15.143s, table=0, n_packets=0, n_bytes=0, idle_age=15, in_port=1
cookie=0x0, duration=4.103s, table=0, n_packets=0, n_bytes=0, idle_age=4, in_port=2 a
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.206 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.033 ms
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.033/0.093/0.206/0.079 ms
mininet> sudo wireshark
*** [Unknown command: sudo wireshark]
mininet> e*** Stopping 1 controllers
c0
*** Stopping 3 links
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
[1]+ Done wireshark
mininet@mininet-vm:~$ sudo wireshark
[1] 6793
mininet@mininet-vm:~$
```



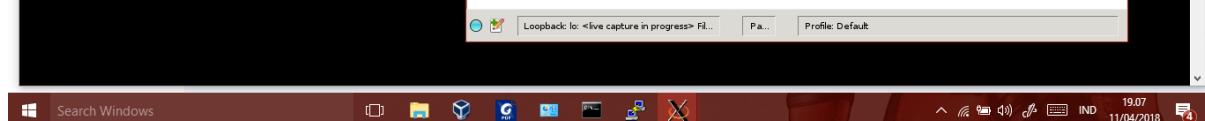
## **5.Membuat referensi OF controller**

```
[3] mininet@mininet-vm: ~
[3] mininet@mininet-vm: ~$ Apr 11 04:01:25|000001|vcom_tcp|ERR|ptcp; bind: Address already in use
Apr 11 04:01:25|000002|controller|ERR|ptcp; connect: Address already in use
controller: no active or passive switch connections
etomem --topo single,3 --mac --switch ovsk --controller r
etomem: command not found
[3] Exit 1 controller ptcp:
[3] Exit 1 controller ptcp:
mininet@mininet-vm: ~$ controller ptcp: 6
[3] 6981
mininet@mininet-vm: ~$ Apr 11 04:01:56|000001|vcom_tcp|ERR|ptcp; bind: Address already in use
Apr 11 04:01:56|000002|controller|ERR|ptcp; connect: Address already in use
controller: no active or passive switch connections
etomem --topo single,3 --mac --switch ovsk --controller r
etomem: command not found
[3] Exit 1 controller ptcp:
[3] Exit 1 controller ptcp:
mininet@mininet-vm: ~$ controller ptcp: 8
[3] 7089
mininet@mininet-vm: ~$ Apr 11 04:05:15|000001|vcom_tcp|ERR|ptcp; bind: Address already in use
Apr 11 04:05:15|000002|controller|ERR|ptcp; connect: Address already in use
controller: no active or passive switch connections
sudo mn --topo single,3 --mac --switch ovsk --controller r
-----
Caught exception. Cleaning up...
Exception: error: r is unknown - please specify one of ['ovsk', 'none', 'remote', 'default', 'nox', 'rsig', 'ref']

*** Removing excess controllers/oftprotocols/ofdatapaths/pings/noxes
killall controlleroftprotocolofdatapath ping_nox_core.ltnox_core.ltnox_core.ows-openflowd ovs-controller udplibtest minnexec
pskill -9 -f "sudo ./mininet"
*** Removing junk files from /tmp
rm -rf /tmp/vconn/* /tmp/vlogs/* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
grep -e 'egrep "(dp[0-9]+)" | sed "s/dp/\$nl/;"'
*** Removing OVS datapaths
ovs-vstcl --timeout=1 list-br
ovs-vstcl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o "([-:][a-zA-Z0-9]+)-eth[0-9]+"
ip link show | egrep -o "([-:][a-zA-Z0-9]+)-eth[0-9]+"
*** Killing stale mininet node processes
pskill -9 -f mininet:
*** Shutting down stale tunnels
pskill -9 -f TunnelEthernet
pskill -9 -f .ssh/wn
rm -rf ./sshd
*** Cleanup complete.
[2]- Terminated controller ptcp:
[3]- Exit 1 controller ptcp:
mininet@mininet-vm: ~$
```



## **6.Filter OF pada wireshark dan apply**



## 7.Inspect packet

The screenshot shows two windows side-by-side, both displaying network traffic captured from a loopback interface (Loopback: lo) using Wireshark 1.10.6. The traffic consists of 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface 0.

**Top Window (Wireshark):**

- Frame 5741:** 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface 0.
- Protocol Tree:** Shows the structure of the packet, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and OpenFlow (LOXI).
- Hex View:** Displays the raw hex data of the packet.
- Source Code:** Shows the C code for the OFPT\_FEATURES\_REPLY message.
- Bottom Panel:** Includes a "Loopback: lo: <live capture in progress>" status message and a "Profile: Default" button.

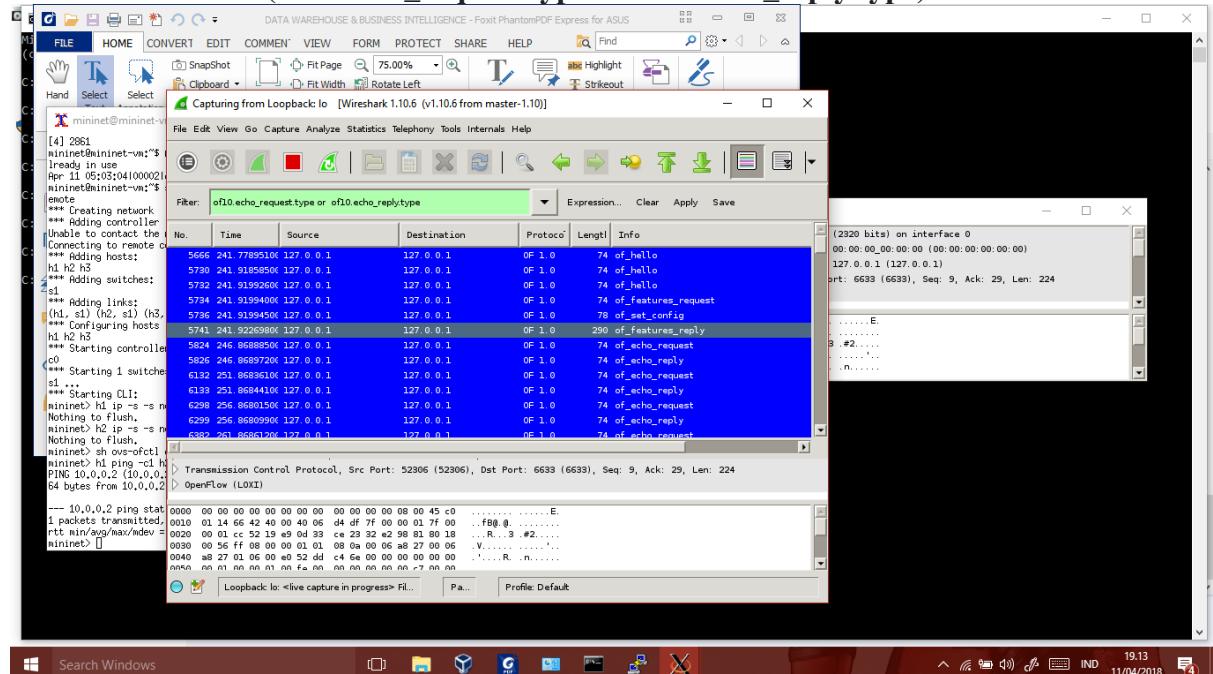
**Bottom Window (Windows File Explorer):**

- File Path:** C:\mininet\mininet-v1\\$
- Log Output:**

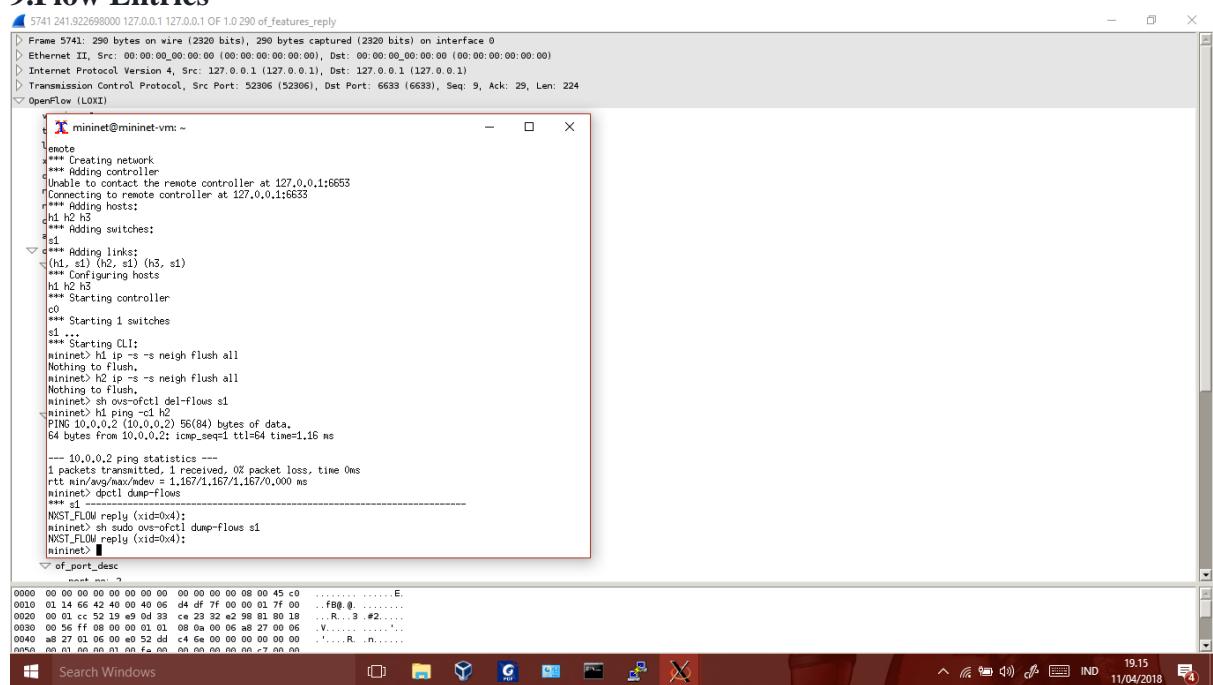
```
Apr 11 05:03:04 1000021
mininet@mininet-v1:~$ ls
lready in use
Apr 11 05:03:04 1000021
mininet@mininet-v1:~$ 
*** Creating network
*** Adding controller
Unable to contact the remote controller
Connecting to remote controller
*** Adding hosts:
*** Adding switches:
s1
*** Adding links:
(s1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controllers
c0
*** Starting switches
s1
*** Starting CLI:
mininet> h1 ip -s -s n
Nothing to flush,
mininet> h2 ip -s -s n
Nothing to flush,
mininet> h3 ip -s -s n
Nothing to flush,
mininet> ping -c 3 h1
PING 10.0.0.2 (10.0.0.2) 64 bytes (12 words) from 10.0.0.2:
-- 10.0.0.2 ping statistics --
1 packets transmitted, 1 packets received, 0% packet loss
rtt min/avg/max/mdev = 0.000 ms
mininet> []

```
- System Status Bar:** Shows the date (11/04/2018), time (19:09), and battery level (IND).

## 8.Filter of dan not (of10.echo\_request.type or of10.echo\_reply.type)



## 9.Flow Entries



## 10. Benchmark kernel-v vs -user space

The screenshot shows a Windows terminal window titled "mininet@mininet-vm: ~". The window displays a command-line session with several commands entered:

- ping statistics (showing 1 packet transmitted, 1 received, 0% packet loss, time 0ms)
- iperf (showing results: 32.2 Gbits/sec, 32.3 Gbits/sec)
- dpctl dump-flows (listing flows in the OpenFlow table)
- NXST\_FLOW reply (xid=0x4) for various entries, including:
  - cookie=0x0, duration=30.118s, table=0, n\_packets=42, idle\_timeout=60, idle\_age=30, priority=65335, arp\_in\_port=3, vlan\_tci=0x0000, dl\_src=00:00:00:00:03:01, dl\_dst=00:00:00:00:00:00, nw\_src=10.0.0.1, nw\_dst=10.0.0.3, nw\_tos=0, tp\_src=59342, tp\_dst=5001 actions=output;3
  - cookie=0x0, duration=30.118s, table=0, n\_packets=1, idle\_timeout=60, idle\_age=30, priority=65335, arp\_in\_port=3, vlan\_tci=0x0000, dl\_src=00:00:00:00:00:00, dl\_dst=00:00:00:00:00:00, nw\_src=10.0.0.1, nw\_dst=10.0.0.1, nw\_tos=0, tp\_src=59338, tp\_dst=59338 actions=output;1
  - cookie=0x0, duration=29.611s, table=0, n\_packets=5, idle\_timeout=60, idle\_age=28, priority=65335, arp\_in\_port=3, vlan\_tci=0x0000, dl\_src=00:00:00:00:01:03, dl\_dst=00:00:00:00:00:01, nw\_src=10.0.0.3, nw\_dst=10.0.0.1, nw\_tos=0, tp\_src=5001, tp\_dst=59340 actions=output;1
  - cookie=0x0, duration=30.118s, table=0, n\_packets=74, idle\_timeout=60, idle\_age=30, priority=65335, arp\_in\_port=3, vlan\_tci=0x0000, dl\_src=00:00:00:00:00:00, dl\_dst=00:00:00:00:00:01, nw\_src=10.0.0.1, nw\_dst=10.0.0.3, nw\_tos=0, tp\_src=59338, tp\_dst=5001 actions=output;3
  - cookie=0x0, duration=29.604s, table=0, n\_packets=208659, n\_bytes=1905362, idle\_timeout=60, idle\_age=24, priority=65335, arp\_in\_port=3, vlan\_tci=0x0000, dl\_src=00:00:00:00:00:03, dl\_dst=00:00:00:00:00:01, nw\_src=10.0.0.3, nw\_dst=10.0.0.1, nw\_tos=0, tp\_src=5001, tp\_dst=59342 actions=output;1
  - cookie=0x0, duration=29.593s, table=0, n\_packets=1, n\_bytes=86, idle\_timeout=60, idle\_age=29, priority=65335, arp\_in\_port=3, vlan\_tci=0x0000, dl\_src=00:00:00:00:00:01, dl\_dst=00:00:00:00:00:03, nw\_src=10.0.0.1, nw\_dst=10.0.0.3, nw\_tos=0, tp\_src=59340, tp\_dst=5001 actions=output;1
- of\_port\_desc (displaying port descriptions)

The terminal window is part of a larger desktop environment with a taskbar at the bottom showing various icons and system status.

# POX Controller

Adalah salah satu controller SDN yang mendukung protokol OpenFlow. POX adalah controller yang berbasis bahasa Python. Langkah-langkahnya adalah sebagai berikut.

## 1.POX Controller files pada Mininet

```
X mininet@mininet-vm: ~/pox/pox/forwarding
mininet@mininet-vm:~$ tree -L 1 -d
.
├── loxigen
├── MiniNAM
├── mininet
├── oflops
├── oftest
└── openflow
└── pox

7 directories
mininet@mininet-vm:~$ cd pox
mininet@mininet-vm:~/pox$ cd pox/misc
mininet@mininet-vm:~/pox/pox/misc$ ll
total 76
drwxrwxr-x 3 mininet mininet 4096 Mar 21 2017 ../
drwxrwxr-x 15 mininet mininet 4096 Mar 21 2017 ../..
-rw-rw-r-- 1 mininet mininet 1240 Mar 21 2017 cbench.py
-rw-rw-r-- 1 mininet mininet 1079 Mar 21 2017 full_payload.py
-rw-rw-r-- 1 mininet mininet 5214 Mar 21 2017 gephi_topo.py
-rw-rw-r-- 1 mininet mininet 689 Mar 21 2017 __init__.py
-rw-rw-r-- 1 mininet mininet 10251 Mar 21 2017 ip_loadbalancer.py
-rw-rw-r-- 1 mininet mininet 3794 Mar 21 2017 mac_blocker.py
-rw-rw-r-- 1 mininet mininet 14375 Mar 21 2017 nat.py
-rw-rw-r-- 1 mininet mininet 4582 Mar 21 2017 of_tutorial.py
-rw-rw-r-- 1 mininet mininet 2096 Mar 21 2017 pidfile.py
drwxrwxr-x 2 mininet mininet 4096 Mar 21 2017 telnetd/
mininet@mininet-vm:~/pox/pox/misc$ cd ..
mininet@mininet-vm:~/pox/pox$ forwarding
forwarding: command not found
mininet@mininet-vm:~/pox/pox$ cd forwarding
mininet@mininet-vm:~/pox/pox/forwarding$ ll
total 96
drwxrwxr-x 2 mininet mininet 4096 Mar 21 2017 ../
drwxrwxr-x 15 mininet mininet 4096 Mar 21 2017 ../..
-rw-rw-r-- 1 mininet mininet 1092 Mar 21 2017 hub.py
-rw-rw-r-- 1 mininet mininet 651 Mar 21 2017 __init__.py
-rw-rw-r-- 1 mininet mininet 4426 Mar 21 2017 l2_flowvisor.py
-rw-rw-r-- 1 mininet mininet 6692 Mar 21 2017 l2_learning.py
-rw-rw-r-- 1 mininet mininet 15558 Mar 21 2017 l2_multi.py
-rw-rw-r-- 1 mininet mininet 4324 Mar 21 2017 l2_nx.py
-rw-rw-r-- 1 mininet mininet 2105 Mar 21 2017 l2_nx_self_learning.py
-rw-rw-r-- 1 mininet mininet 2882 Mar 21 2017 l2_pairs.py
-rw-rw-r-- 1 mininet mininet 12330 Mar 21 2017 l3_learning.py
-rw-rw-r-- 1 mininet mininet 14102 Mar 21 2017 topo_proactive.py
mininet@mininet-vm:~/pox/pox/forwarding$ █
```

## 2. Memilih of\_tutorial.py pada pox/pox/misc

```
X mininet@mininet-vm: ~/pox/pox/misc
def resend_packet (self, packet_in, out_port):
    """Instructs the switch to resend a packet that it had sent to us.
    "packet_in" is the ofp.packet_in object the switch had sent to the
    controller due to a table-miss.
    """
    msg = of.ofp_packet_out()
    msg.data = packet_in

    # Add an action to send to the specified port
    action = of.ofp_action_output(port = out_port)
    msg.actions.append(action)

    # Send message to switch
    self.connection.send(msg)

def act_like_hub (self, packet, packet_in):
    """Implement hub-like behavior -- send all packets to all ports besides
    the input port.
    """
    # We want to output to all ports -- we do that using the special
    # OFPP_ALL port as the output port. (We could have also used
    # OFPP_FLOOD.)
    self.resend_packet(packet_in, of.OFPP_ALL)

    # Note that if we didn't get a valid buffer_id, a slightly better
    # implementation would check that we got the full data before
    # sending it (len(packet_in.data) should be == packet_in.total_len).

def act_like_switch (self, packet, packet_in):
    """Implement switch-like behavior.

    """ * * * DELETE THIS LINE TO START WORKING ON THIS (AND THE ONE BELOW!) * *
    # Here's some pseudocode to start you off implementing a learning
    # switch. You'll need to rewrite it as real Python code.

    # Learn the port for the source MAC
    self.mac_to_port ... <add or update entry>

    # The port associated with the destination MAC of the packet is known:
    # Send packet out the associated port
    self.resend_packet(packet_in, ...)

    # Once you have the above working, try pushing a flow entry
    # instead of resending the packet (comment out the above and
    # uncomment and complete the below.)
```



```
X mininet@mininet-vm: ~/pox/pox/misc
drwxr-x 2 mininet mininet 4096 Mar 21 2017 telnetd/
mininet@mininet-vm:~/pox/pox/misc$ cat of_tutorial.py
# Copyright 2012 James McCauley

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

"""
This component is for use with the OpenFlow tutorial.

It acts as a simple hub, but can be modified to act like an L2
learning switch.

It's roughly similar to the one Brandon Heller did for NOX.
"""

from pox.core import core
import pox.openflow.libopenflow_01 as of

log = core.getLogger()

class Tutorial (object):
    """
    A Tutorial object is created for each switch that connects.
    A Connection object for that switch is passed to the __init__ function.
    """

    def __init__ (self, connection):
        # Keep track of the connection to the switch so that we can
        # send it messages!
        self.connection = connection

        # This binds our PacketIn event listener
        connection.addListeners(self)

        # Use this table to keep track of which ethernet address is on
        # which switch port (keys are MACs, values are ports).
        self.mac_to_port = {}

    def resend_packet (self, packet_in, out_port):
        """Instructs the switch to resend a packet that it had sent to us.
```



```

X mininet@mininet-vm: ~/pox/pox/misc

# Once you have the above working, try pushing a flow entry
# instead of resending the packet (comment out the above and
# uncomment and complete the below.)

log.debug("Installing flow...")
# Maybe the log statement should have source/destination/port?

#msg = of.ofp_flow_mod()
#
## Set Fields to match received packet
#msg.match = of.ofp_match.from_packet(packet)
#
#< Set other Fields of flow_mod (timeouts? buffer_id?) >
#
#< Add an output action, and send -- similar to resend_packet() >

else:
    # Flood the packet out everything but the input port
    # This part looks familiar, right?
    self.resend_packet(packet_in, OF.OFPP_ALL)

""" * DELETE THIS LINE TO START WORKING ON THIS *

def _handle_PacketIn (self, event):
    """
    Handles packet in messages from the switch.

    packet = event.parsed # This is the parsed packet data.
    if not packet.parsed:
        log.warning("Ignoring incomplete packet")
        return

    packet_in = event.ofp # The actual ofp_packet_in message.

    # Comment out the following line and uncomment the one after
    # when starting the exercise.
    self.act_like_hub(packet, packet_in)
    #self.act_like_switch(packet, packet_in)

def launch():
    """
    Starts the component
    """
    def start_switch (event):
        log.debug("Controlling %s" % (event.connection,))
        Tutorial(event.connection)
        core.openflow.addListenersByName("ConnectionUp", start_switch)
mininet@mininet-vm:~/pox/pox/misc$ 

```

3. memulai pox controller dengan '`./pox.py log.level --DEBUG misc.ofTutorial`' pada directory 'pox'. lalu mulai membuat network pada terminal lain dengan perintah '`sudo mn --topo single,3 --mac --witch ovs --controller remote`'. jika pox controller terhubung, akan muncul '`INFO:openflow.of_01:[00-00-00-00-00-01 2] connected`' pada terminal poxcontroller. lalu mulailah membuka terminal untuk node 1,2, dan 3 dengan perintah '`xterm h1 h2 h3`' pada console jaringan yang akan memuncul 3 terminal untuk masing-masing node.

```

X mininet@mininet-vm: ~/pox

mininet@mininet-vm:~/pox$ ./pox.py log.level --DEBUG misc.ofTutorial
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
DEBUG:core:POX 0.2.0 (carp) going up...
DEBUG:core:Running on CPython (2.7.6/Oct 26 2016 20:30:19)
DEBUG:core:Platform is Linux-4.2.0-27-generic-x86_64-with-Ubuntu-14.04-trusty
INFO:core:POX 0.2.0 (carp) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[None 1] closed
INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
DEBUG:misc.ofTutorial:Controlling [00-00-00-00-00-01 2]
xterm h1 h2 h3
INFO:openflow.of_01:[00-00-00-00-00-01 2] closed
INFO:openflow.of_01:[None 3] closed
INFO:openflow.of_01:[00-00-00-00-00-01 4] connected
DEBUG:misc.ofTutorial:Controlling [00-00-00-00-00-01 4]
INFO:openflow.of_01:[00-00-00-00-00-01 4] closed
INFO:openflow.of_01:[None 5] closed
INFO:openflow.of_01:[00-00-00-00-00-01 6] connected
DEBUG:misc.ofTutorial:Controlling [00-00-00-00-00-01 6]
INFO:openflow.of_01:[00-00-00-00-00-01 6] closed
INFO:openflow.of_01:[None 7] closed
INFO:openflow.of_01:[00-00-00-00-00-01 8] connected
DEBUG:misc.ofTutorial:Controlling [00-00-00-00-00-01 8]

```

```

mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1 h2 h3
mininet> 

```

```

root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=23.6 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time
rtt min/avg/max/mdev = 23.666/23.666/23.666/0.000 ms
root@mininet-vm:~# 

root@mininet-vm:~# tcpdump -XX -n -i h2-eth0 > h2.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h2-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
[...]

```

```

root@mininet-vm:~# tcpdump -XX -n -i h3-eth0 > h3.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h3-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
[...]

```

4. berikan perintah 'tcpdump -XX -n -i h2-eth0 > h2.txt' pada terminal node 2, dan 'tcpdump -XX -n -i h3-eth0 > h3.txt' pada terminal node 3, perintah tersebut memerintahkan terminal node untuk merekam aktivitas node dan menyimpannya pada file h2.txt dan h3.txt. Lalu lakukan ping pada terminal node 1 dengan perintah 'ping c1 10.0.0.2'. lalu hentikan perintah merekam aktivitas node 2 dan 3 dengan menekan tombol ctrl+c. lalu buka file h2.txt dan h3.txt untuk melihat hasil dari perintah 'ping c1 10.0.0.2' tadi.

```

X mininet@mininet-vm: ~
  -  □  ×

0x0060: 3637 67
23:44:47.690364 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 22245, seq 1, length
64
0x0000: 0000 0000 0001 0000 0000 0002 0800 4500 .....E.
0x0010: 0054 129b 0000 4001 540c 0a00 0002 0a00 .T....@.T.....
0x0020: 0001 0000 da03 56e5 0001 5fba c95a 0000 .....V.....Z..
0x0030: 0000 dd2d 0a00 0000 0000 1011 1213 1415 ...-.....
0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 .....!#$%
0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 & ()*+,-.:/012345
0x0060: 3637 67

23:44:52.705900 ARP, Request who-has 10.0.0.1 tell 10.0.0.2, length 28
0x0000: 0000 0000 0001 0000 0000 0002 0806 0001 .....
0x0010: 0800 0604 0001 0000 0000 0002 0a00 0002 .....
0x0020: 0000 0000 0000 0a00 0001 .....*.

23:44:52.736178 ARP, Reply 10.0.0.1 is-at 00:00:00:00:00:01, length 28
0x0000: 0000 0000 0002 0000 0000 0001 0806 0001 .....
0x0010: 0800 0604 0002 0000 0000 0001 0a00 0001 .....
0x0020: 0000 0000 0002 0a00 0002 .....*.

mininet@mininet-vm:~$ cat h3.txt
23:44:47.689440 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28
0x0000: ffff ffff ffff 0000 0000 0001 0806 0001 .....
0x0010: 0800 0604 0001 0000 0000 0001 0a00 0001 .....
0x0020: 0000 0000 0000 0a00 0002 .....*.

23:44:47.689939 ARP, Reply 10.0.0.2 is-at 00:00:00:00:00:02, length 28
0x0000: 0000 0000 0001 0000 0000 0002 0806 0001 .....
0x0010: 0800 0604 0002 0000 0000 0002 0a00 0002 .....
0x0020: 0000 0000 0001 0a00 0001 .....*.

23:44:47.690353 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 22245, seq 1, leng
th 64
0x0000: 0000 0000 0002 0000 0000 0001 0800 4500 .....E.
0x0010: 0054 4cdd 4000 4001 d9c9 0a00 0001 0a00 .TL.@.0.....
0x0020: 0002 0800 d203 56e5 0001 5fba c95a 0000 .....V.....Z..
0x0030: 0000 dd2d 0a00 0000 0000 1011 1213 1415 ...-.....
0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 .....!#$%
0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 & ()*+,-.:/012345
0x0060: 3637 67

23:44:47.690766 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 22245, seq 1, length
64
0x0000: 0000 0000 0001 0000 0000 0002 0800 4500 .....E.
0x0010: 0054 129b 0000 4001 540c 0a00 0002 0a00 .T....@.T.....
0x0020: 0001 0000 da03 56e5 0001 5fba c95a 0000 .....V.....Z..
0x0030: 0000 dd2d 0a00 0000 0000 1011 1213 1415 ...-.....
0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 .....!#$%
0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 & ()*+,-.:/012345
0x0060: 3637 67

23:44:52.735717 ARP, Request who-has 10.0.0.1 tell 10.0.0.2, length 28
0x0000: 0000 0000 0001 0000 0000 0002 0806 0001 .....
0x0010: 0800 0604 0001 0000 0000 0002 0a00 0002 .....
0x0020: 0000 0000 0000 0a00 0001 .....*.

23:44:52.736176 ARP, Reply 10.0.0.1 is-at 00:00:00:00:00:01, length 28
0x0000: 0000 0000 0002 0000 0000 0001 0806 0001 .....
0x0010: 0800 0604 0002 0000 0000 0001 0a00 0001 .....
0x0020: 0000 0000 0002 0a00 0002 .....*.

mininet@mininet-vm:~$ █

```

X mininet@mininet-vm: ~/pox

```
mininet@mininet-vm:~/pox$ ./pox.py log.level --DEBUG misc.of_tutorial
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
DEBUG:core:POX 0.2.0 (carp) going up...
DEBUG:core:Running on CPython (2.7.6/Oct 26 2016 20:30:19)
DEBUG:core:Platform is Linux-4.2.0-27-generic-x86_64-with-Ubuntu-14.04-trusty
[INFO:core:POX 0.2.0 (carp) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
[INFO:openflow.of_01:[None 1] closed
[INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 2]
<term h1 h2 h3
[INFO:openflow.of_01:[00-00-00-00-00-01 2] closed
[INFO:openflow.of_01:[None 3] closed
[INFO:openflow.of_01:[00-00-00-00-00-01 4] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 4]
[INFO:openflow.of_01:[00-00-00-00-00-01 4] closed
[INFO:openflow.of_01:[None 5] closed
[INFO:openflow.of_01:[00-00-00-00-00-01 6] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 6]
[INFO:openflow.of_01:[00-00-00-00-00-01 6] closed
[INFO:openflow.of_01:[None 7] closed
[INFO:openflow.of_01:[00-00-00-00-00-01 8] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 8]
[INFO:openflow.of_01:[00-00-00-00-00-01 8] closed
]
```

X mininet@mininet-vm: ~

```
mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller r
emote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> sh ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
mininet> pingpair
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['17.4 Mbits/sec', '18.7 Mbits/sec']
mininet> █
```

5. berikan perintah merekam aktivitas node lagi pada node 2 dan 3. lalu kali ini kita mencoba melakukan ping ke ip yg tidak ada, disini contohnya 10.0.0.5. lalu buka h2.txt dan h3.txt maka hasilnya akan seperti gambar, ip 10.0.0.1 merequest "who has 10.0.05 ? " sampai 3 kali dan tidak menemukannya.

```
T "Node: h1"

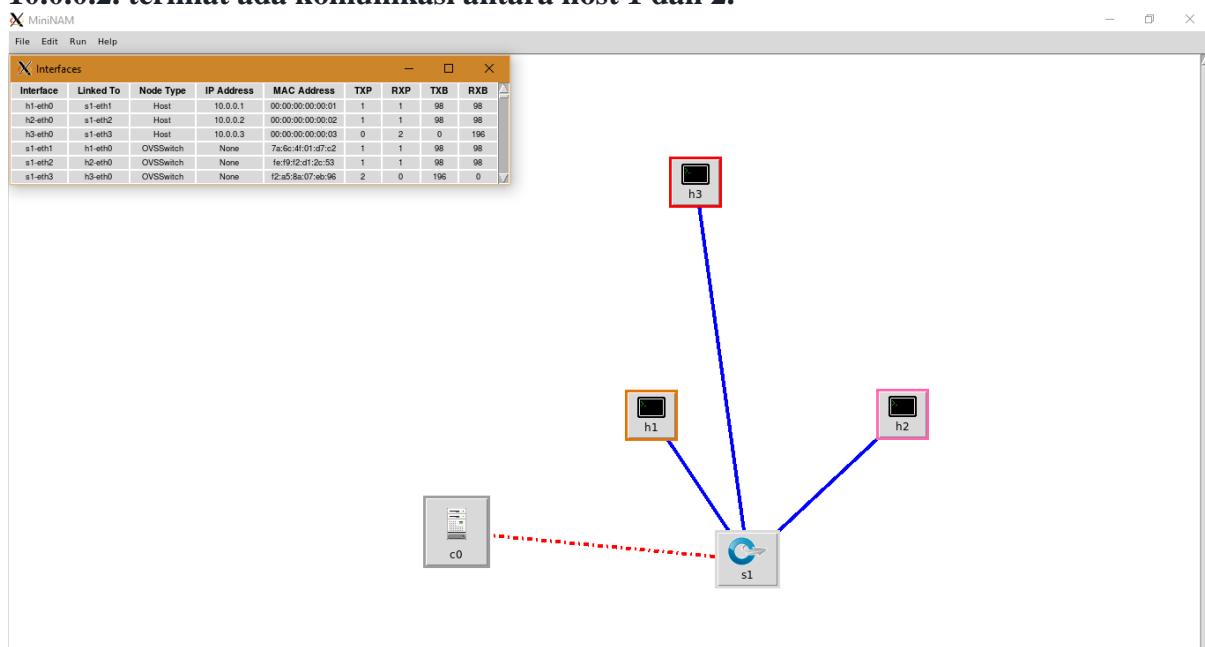
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.098/6.098/6.098/0.000 ms
root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=51.7 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 51.747/51.747/51.747/0.000 ms
root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=9.66 ms

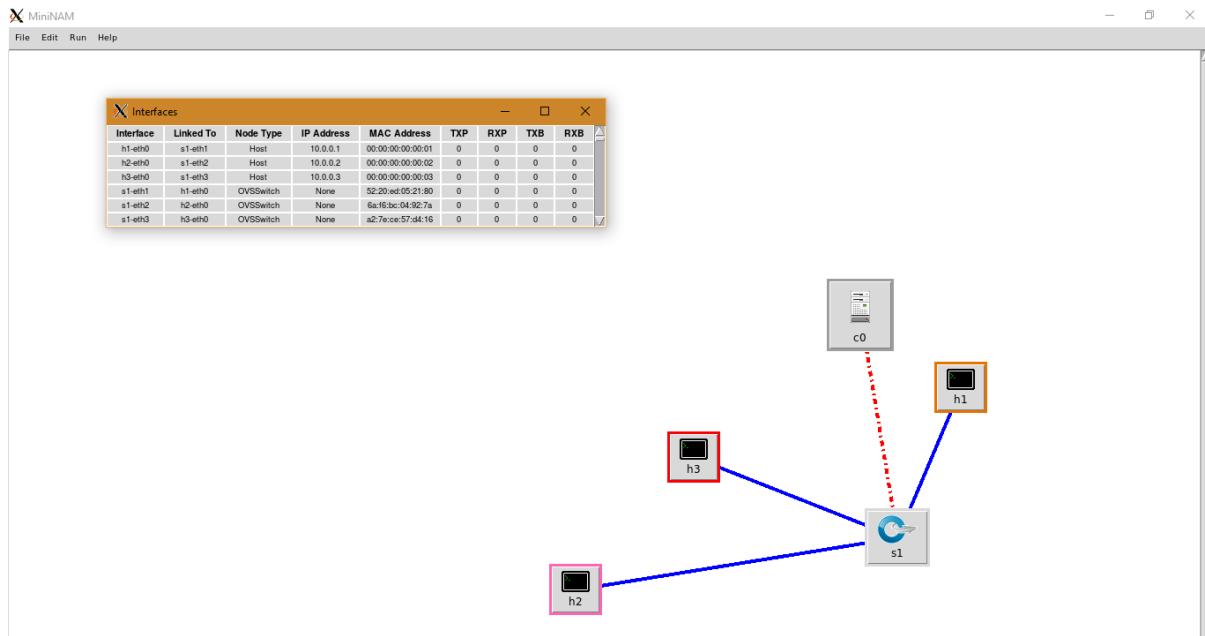
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 9.660/9.660/9.660/0.000 ms
root@mininet-vm:~# ping -c1 10.0.0.5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.5 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
root@mininet-vm:~#
```

6. Tampilan jaringan yg kita buat tadi pada MiniNAM ketika ping dari 10.0.0.1 ke 10.0.0.2. terlihat ada komunikasi antara host 1 dan 2.



7. Tampilan jaringan yg kita buat tadi pada MiniNAM ketika ping dari 10.0.0.1 ke 10.0.0.5. terlihat tidak ada komunikasi antar host.



8. disini kita akan melakukan perbandingan antara kecepatan menggunakan hub dan switch. berikan perintah iperf pada console.

```
mininet> sh ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
mininet> pingpair
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['6.61 Mbits/sec', '7.32 Mbits/sec']
mininet> █
```

## 9. switch lebih cepat daripada hub

The image shows two terminal windows side-by-side. The left window, titled '9.OpenFlowBenchmark.PNG - Photos', displays the output of an iperf test between hosts h1 and h3. It shows a bandwidth of approximately 383 Mbit/sec. The right window, titled '4.POXControllerHubBehavior.PNG - Photos', shows the output of a ping test between hosts h1 and h3, with a result of 0% dropped packets.

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['383 Mbit/sec', '389 Mbit/sec']
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 1 switches
^[[As1
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 27.655 seconds
mininet@mininet-vm:~$ sudo mn --topo single,3 --controller remote --switch pox
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6653
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1
*** Starting CLI:
mininet> sh ovs-ofctl dump-flows s1
ovs-ofctl: s1 is not a bridge or a socket
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:3
ovs-ofctl: s1 is not a bridge or a socket
mininet> sh ovs-ofctl add-flow s1 in_port=3,act=drop
ovs-ofctl: s1 is not a bridge or a socket
mininet> 
```

```
mininet> sh ovs-ofctl dump-flows s1
ovs_OFLOW reply (xid=0x4):
mininet> pingpair
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['6.61 Mbit/sec', '7.32 Mbit/sec']
mininet> 
```

10. Menambah fungsi `act_like_switch` ke `of_tutorial.py`, dengan menggunakan vim editor, vim sudah ada pada mininet jadi kita tinggal memasukkan command “`vi f_tutorial.py`” pada directory `pox/pox/misc`. Cara menggunakan vim editor bisa dilihat di “ <https://vim.rtorr.com/>”

T mininet@mininet-vm: ~/pox/pox/misc

```
def act_like_switch (self, packet, packet_in):
    """
    Implement switch-like behavior.
    """

    # Here's some psuedocode to start you off implementing a learning
    # switch. You'll need to rewrite it as real Python code.

    # Learn the port for the source MAC
    self.mac_to_port[packet.src] = packet_in.in_port
    if packet.dst in self.mac_to_port:
        print("Packet sent to Control Plane")
        # Send packet out the associated port
        self.resend_packet(packet_in, self.mac_to_port[packet.dst])

    # Once you have the above working, try pushing a flow entry
    # instead of resending the packet (comment out the above and
    # uncomment and complete the below.)

    #log.debug("Installing flow...")
    # Maybe the log statement should have source/destination/port?
    msg = of.ofp_flow_mod()
    msg.match.dl_dst = packet.dst
    ## Set fields to match received packet
    #msg.match = of.ofp_match.from_packet(packet)
    msg.actions.append(of.ofp_action_output(port=self.mac_to_port[packet.dst]))
    #< Set other fields of flow_mod (timeouts? buffer_id?) >
    self.connection.send(msg)
    #< Add an output action, and send -- similar to resend_packet() >

else:
    # Flood the packet out everything but the input port
    # This part looks familiar, right?
    self.resend_packet(packet_in, of.OFPP_ALL)
```