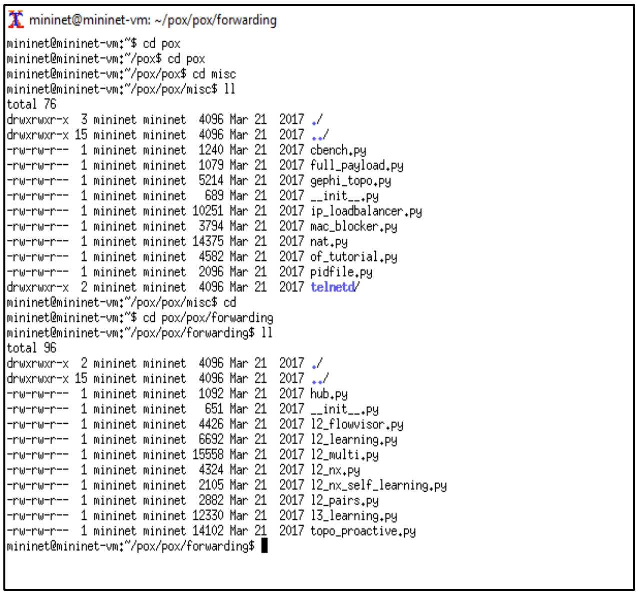
DESI B TAMBUNAN (161402005)
SONIA E TELAUMBANUA (161402045)
NOVALINA SIMBOLON (161402077)
CHYNTIA CLAUDIA (161402101)
TIRZA PRISKILLA (161402110)

POX CONTROLLER

1. Aktifkan controller POX

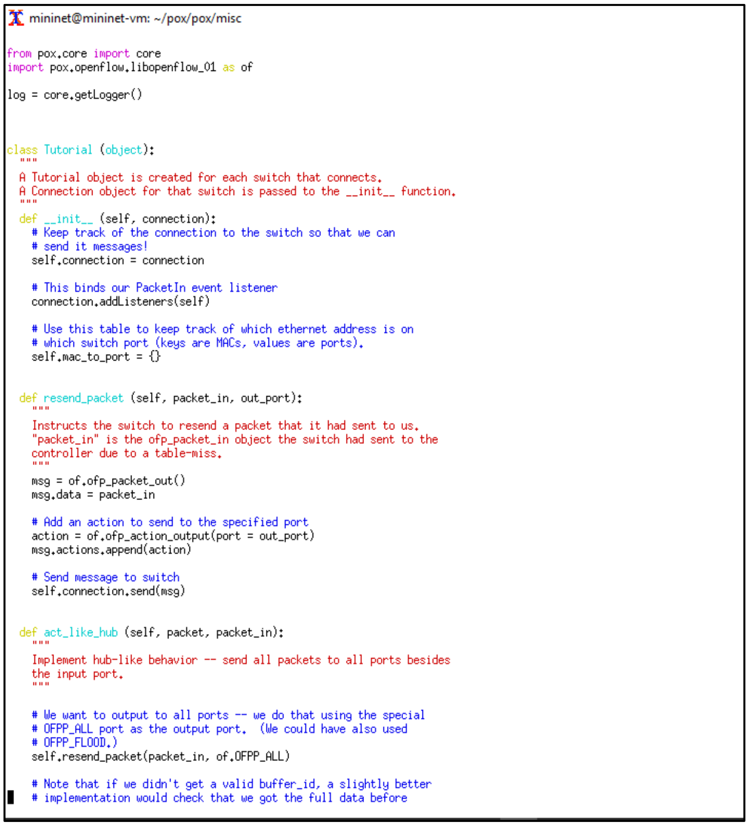
```
root@ubuntu:~# cd /pox root@ubuntu:~/pox# ./pox.py log.level --DEBUG
misc.of_tutorial
```



mininet@mininet-vm: ~/pox/pox/forwarding

```
mininet@mininet-vm:~$ cd pox
mininet@mininet-vm:~/pox$ cd pox
mininet@mininet-vm:~/pox/pox$ cd 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:~$ cd pox/pox/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. Of_tutorial.py

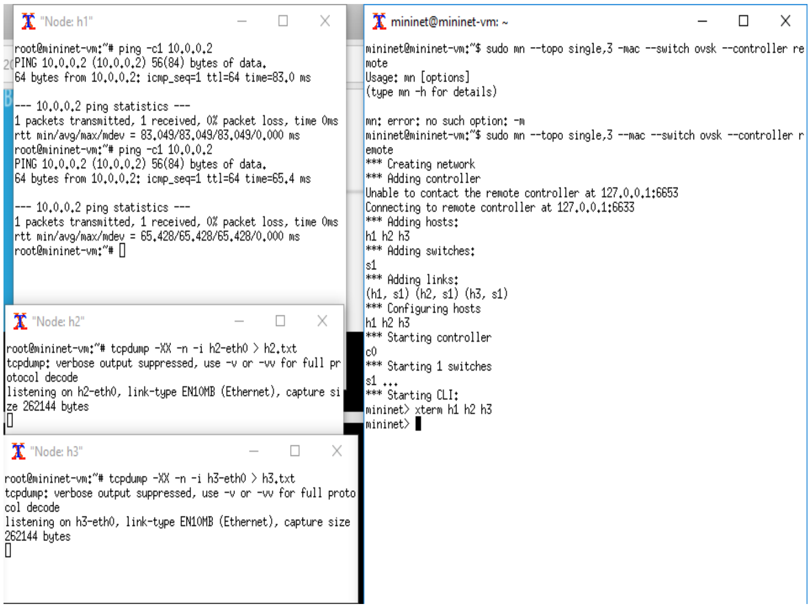


mininet@mininet-vm: ~/pox/pox/misc

```
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.
    "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
```

3. Membuat 3 host dengan 1 switch

Sudo mn –topo single,3 –mac –switch ovsk –controller remote



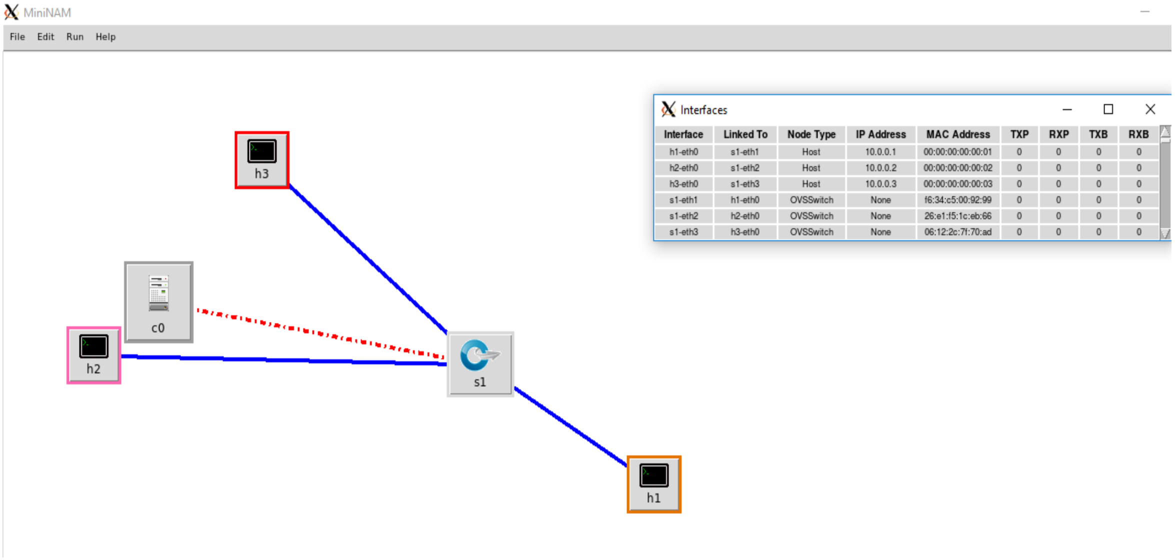
```
root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=83.0 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 83.049/83.049/83.049/0.000 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=65.4 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 65.428/65.428/65.428/0.000 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 pr
otocol decode
listening on h2-eth0, link-type EN10MB (Ethernet), capture si
ze 262144 bytes
```

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

```
mininet@mininet-vm:~$ sudo mn --topo single,3 -mac --switch ovsk --controller re
mote
Usage: mn [options]
(type mn -h for details)

mn: error: no such option: -m
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> xterm h1 h2 h3
mininet> 
```

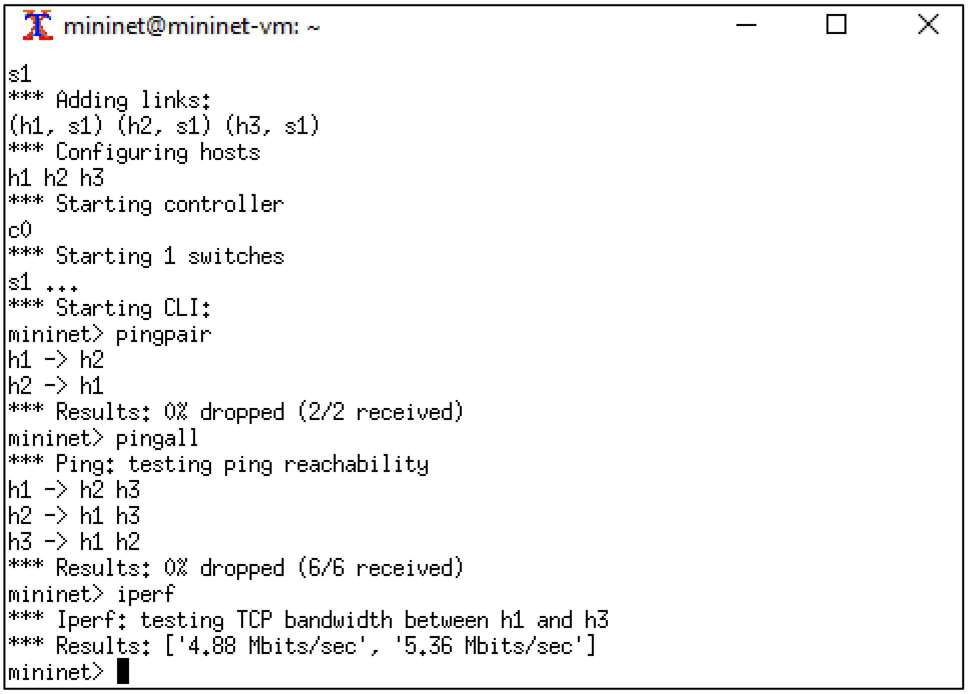
4. h1 ping –c1 10.0.0.2



Interface	Linked To	Node Type	IP Address	MAC Address	TXP	RXP	TXB	RXB
h1-eth0	s1-eth1	Host	10.0.0.1	00:00:00:00:00:01	0	0	0	0
h2-eth0	s1-eth2	Host	10.0.0.2	00:00:00:00:00:02	0	0	0	0
h3-eth0	s1-eth3	Host	10.0.0.3	00:00:00:00:00:03	0	0	0	0
s1-eth1	h1-eth0	OVSSwitch	None	f6:34:c5:00:92:99	0	0	0	0
s1-eth2	h2-eth0	OVSSwitch	None	26:e1:f5:1c:eb:66	0	0	0	0
s1-eth3	h3-eth0	OVSSwitch	None	06:12:2c:7f:70:ad	0	0	0	0

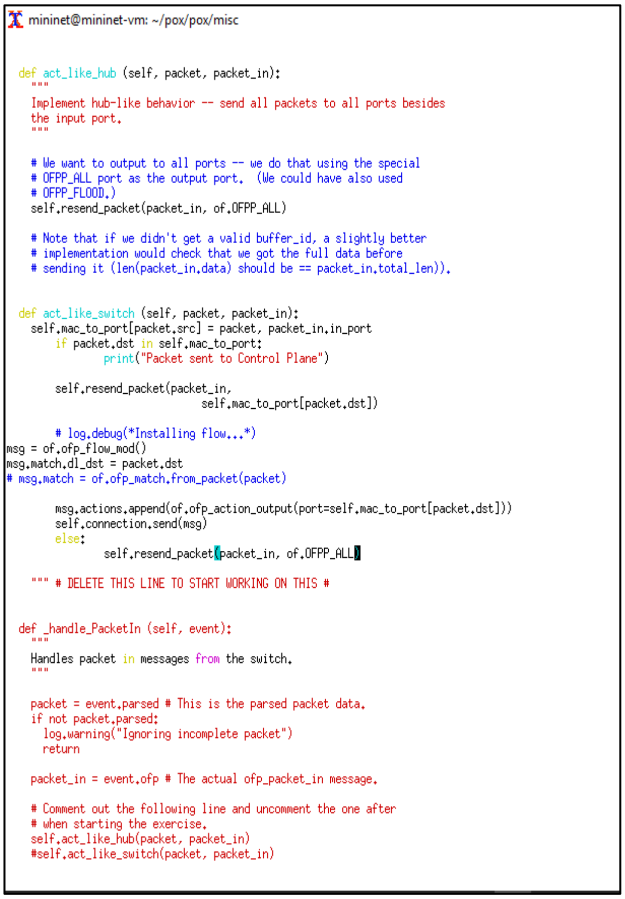
5. Manual Entry

```
mininet>pingpair
minine>pingall
mininet>iperf
```



```
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
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: ['4.88 Mbits/sec', '5.36 Mbits/sec']
mininet>
```

6. Of_tutorial.py



```
  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):
    self.mac_to_port[packet.src] = packet, packet_in.in_port
        if packet.dst in self.mac_to_port:
                print("Packet sent to Control Plane")

        self.resend_packet(packet_in,
                                self.mac_to_port[packet.dst])

        # log.debug(*Installing flow...*)
msg = of.ofp_flow_mod()
msg.match.dl_dst = packet.dst
# msg.match = of.ofp_match.from_packet(packet)

        msg.actions.append(of.ofp_action_output(port=self.mac_to_port[packet.dst]))
        self.connection.send(msg)
        else:
                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)
```