

DAFTAR PUSTAKA

- Abidin, N. Z. (2021). Analisis Performansi Controller POX Dan RYU Pada Jaringan Software Defined Network Dengan Protokol Spanning Tree. *Repository.Uinjkt.Ac.Id*, 118. <https://repository.uinjkt.ac.id/dspace/handle/123456789/56384>
- Adrian, R. (2017). Optimasi Cost pada Open Shortest Path First di Jaringan Software Defined-Network. *Techno.Com*, 16(4), 421–434. <https://doi.org/10.33633/tc.v16i4.1532>
- Ainy, M. (2017). Routing Interior dan Eksterior. *Fakultas Komputer*.
- Attamimi, I., Yahya, W., & Hanafi, M. H. (2017). Analisis Perbandingan Algoritma Floyd-Warshall dan Dijkstra untuk Menentukan Jalur Terpendek Pada Jaringan Openflow. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK)*, 1(12), 1842–1849.
- Cintasari, E. P. (2018). Analisis Kinerja Jaringan Software Defined Network (SDN) Dengan Protokol OpenFlow pada Mininet. *Repository.Uinjkt.Ac.Id*. <http://repository.uinjkt.ac.id/dspace/handle/123456789/53507>
- Dwi Rahmawan, A., & Risqiwati, D. (2020). Analisa Performansi Controller Pada Arsitektur Jaringan Software Defined Network (SDN). *REPOSITOR*, 2(12), 1727–1738.
- Edgar, R., Hanuranto, A. T., & Mentari, O. (2019). *Perancangan Dan Analisis Sistem Pada Kontroler Pox, Ryu, Dan Opendaylight Pada Software Defined Network*. 6(2), 4433–4441.
- Indriani Lestaringati, S. (2018). Analisis Kinerja Arsitektur Software-Defined Network Berbasis OpenDaylight Controller. *Jurnal Teknik Komputer Unikom-Komputika*, 7(1), 194–200.
- Irmawati, A., Irawati, I. D., & Hariyani, Y. S. (2017). Implementasi Protokol Routing Ospf Pada Software Defined Network Berbasis Routeflow. *E-Proceeding of Apllied Science*, 3(2), 1067–1074.
- Iryani, N., Ramadhani, A. D., & Sari, M. K. (2021). Analisis Performansi Routing OSPF menggunakan RYU Controller dan POX Controller pada Software Defined Networking. *Jurnal Telekomunikasi Dan Komputer*, 11(1), 73. <https://doi.org/10.22441/incomtech.v11i1.10187>
- Kementrian Kesehatan Republik Indonesia. (2018). UNIVERSITAS SUMATERA UTARA Poliklinik UNIVERSITAS SUMATERA UTARA. *Jurnal Pembangunan Wilayah & Kota*, 1(3), 82–91.
- Khattak, Z. K., Awais, M., & Departemen, I. (2014). *Evaluasi Kinerja OpenDaylight SDN Pengendali*.

- Khoerul, A., & Ronald, A. (2017). Analisis Performa Jaringan Software Defined Network Berdasarkan Penggunaan Cost Pada Protokol Routing Open Shortest Path First. *Citee*, 1–8.
- Stmik-amik-riau, L. K., Muzawi, R., & Hardianto, R. (2016). *Perancangan Server Dan Analisis Quality of Service (QoS) Jaringan Diskless PXE Linux Pada. 1(1).*
- Sudiyatmoko, A. R., Hertiana, S. N., & Negara, R. M. (2016). Analisis Performansi Perutingan Link State Menggunakan Algoritma Dijkstra Pada Platform Software Defined Network (SDN). *JURNAL INFOTEL - Informatika Telekomunikasi Elektronika*, 8(1), 40. <https://doi.org/10.20895/infotel.v8i1.50>
- Sulfiana, A. (2019). Implementasi dan Analisis Kinerja Software Defined Network Controller Floodlight dan Onos. *Estuarine, Coastal and Shelf Science*, 2020(1), 473–484.
- Thomas, E. E., Palit, H., & Noertjahyana, A. (2018). Aplikasi Manajemen Jaringan Berbasis Software Defined Networking. *Jurnal Infra Petra*, 031. <http://publication.petra.ac.id/index.php/teknik-informatika/article/view/6384>
- Time, R., & Time, R. (2020). *1. Perkenalan. 0123456789.*
- Ummah, I. (2016). Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking. *Indonesian Journal on Computing (Indo-JC)*, 1(1), 95–106. <https://doi.org/10.21108/indojc.2016.1.1.20>
- Utami, P. R. (2020). Analisis Perbandingan Quality of Service Jaringan Internet Berbasis Wireless Pada Layanan Internet Service Provider (Isp) Indihome Dan First Media. *Jurnal Ilmiah Teknologi Dan Rekayasa*, 25(2), 125–137. <https://doi.org/10.35760/tr.2020.v25i2.2723>
- Zhu, L., Karim, M. M., Sharif, K., Li, F., Du, X., & Guizani, M. (2019). *SDN Controllers: Benchmarking & Performance Evaluation.* 1–14. <http://arxiv.org/abs/1902.04491>
- 홍종욱. (2019). 3월 1일의 밤은 대한민국의 봄이었다 — 권보드래, 3월 1일의 밤: 폭력의 세기에 꾸는 평화의 꿈? (돌베개, 2019) —. *Concept and Communication*, null(23), 301–316. <https://doi.org/10.15797/concom.2019..23.009>

LAMPIRAN

Lampiran 1 Script OSPF dan RYU pada Mininet

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import Node, RemoteController, OVSSwitch
from mininet.log import setLogLevel, info
from mininet.cli import CLI
import time
import os

class LinuxRouter(Node):
    "Node dengan IP forwarding diaktifkan."

    def config(self, **params):
        super(LinuxRouter, self).config(**params)
        # Aktifkan forwarding pada router
        self.cmd('sysctl net.ipv4.ip_forward=1')

    def terminate(self):
        self.cmd('sysctl net.ipv4.ip_forward=0')
        super(LinuxRouter, self).terminate()

class NetworkTopo(Topo):
    "Sebuah LinuxRouter yang menghubungkan tiga subnet IP"

    def build(self, **_opts):
        # Alamat IP untuk antarmuka router
        r1_eth1_ip = '10.0.1.1/24' # r1 ke h1
        r2_eth1_ip = '10.0.2.1/24' # r2 ke h2
        r3_eth1_ip = '10.0.3.1/24' # r3 ke h3

        r1_eth2_ip = '10.0.10.1/24' # r1 ke r2
        r2_eth2_ip = '10.0.10.2/24' # r2 ke r1

        r1_eth3_ip = '10.0.11.1/24' # r1 ke r3
        r3_eth2_ip = '10.0.11.2/24' # r3 ke r1

        r2_eth3_ip = '10.0.12.1/24' # r2 ke r3
        r3_eth3_ip = '10.0.12.2/24' # r3 ke r2

        # Tambahan alamat IP untuk antarmuka switch ke router
        s1_eth1_ip = '10.0.1.2/24' # s1 ke r1
        s2_eth1_ip = '10.0.2.2/24' # s2 ke r2
        s3_eth1_ip = '10.0.3.2/24' # s3 ke r3

        # Menambahkan router
        r1 = self.addHost('r1', cls=LinuxRouter, ip=r1_eth1_ip)
```

```

r2 = self.addHost('r2', cls=LinuxRouter, ip=r2_eth1_ip)
r3 = self.addHost('r3', cls=LinuxRouter, ip=r3_eth1_ip)

# Menambahkan host
h1 = self.addHost('h1', ip='10.0.1.100/24', defaultRoute='via
10.0.1.1')
h2 = self.addHost('h2', ip='10.0.2.100/24', defaultRoute='via
10.0.2.1')
h3 = self.addHost('h3', ip='10.0.3.100/24', defaultRoute='via
10.0.3.1')

# Menambahkan switch
s1 = self.addSwitch('s1', cls=OVSSwitch,
protocols='OpenFlow13')
s2 = self.addSwitch('s2', cls=OVSSwitch,
protocols='OpenFlow13')
s3 = self.addSwitch('s3', cls=OVSSwitch,
protocols='OpenFlow13')

# Membuat link dengan alamat IP untuk antarmuka switch ke
router
self.addLink(s1, r1, intfName1='s1-eth1', params1={'ip':
s1_eth1_ip}, intfName2='r1-eth1', params2={'ip': r1_eth1_ip})
# Switch s1 ke Router r1
self.addLink(s2, r2, intfName1='s2-eth1', params1={'ip':
s2_eth1_ip}, intfName2='r2-eth1', params2={'ip': r2_eth1_ip})
# Switch s2 ke Router r2
self.addLink(s3, r3, intfName1='s3-eth1', params1={'ip':
s3_eth1_ip}, intfName2='r3-eth1', params2={'ip': r3_eth1_ip})
# Switch s3 ke Router r3

self.addLink(r1, h1, intfName1='r1-eth4', params1={'ip':
'10.0.1.1/24'}) # Router r1 ke Host h1
self.addLink(r2, h2, intfName1='r2-eth4', params1={'ip':
'10.0.2.1/24'}) # Router r2 ke Host h2
self.addLink(r3, h3, intfName1='r3-eth4', params1={'ip':
'10.0.3.1/24'}) # Router r3 ke Host h3

self.addLink(r1, r2, intfName1='r1-eth2', intfName2='r2-
eth2', params1={'ip': r1_eth2_ip}, params2={'ip': r2_eth2_ip})
self.addLink(r1, r3, intfName1='r1-eth3', intfName2='r3-
eth2', params1={'ip': r1_eth3_ip}, params2={'ip': r3_eth2_ip})
self.addLink(r2, r3, intfName1='r2-eth3', intfName2='r3-
eth3', params1={'ip': r2_eth3_ip}, params2={'ip': r3_eth3_ip})

# Link tambahan untuk menghubungkan switch
self.addLink(s1, s2)
self.addLink(s2, s3)

```

```

def configure_control_plane(self, net):
    "Konfigurasi zebra dan ospf untuk setiap router"
    r1 = net.getNodeByName('r1')
    r2 = net.getNodeByName('r2')
    r3 = net.getNodeByName('r3')

    r1.cmd('zebra -f /usr/local/etc/r1zebra.conf -d -z
~/r1zebra.api -i ~/r1zebra.interface')
    time.sleep(1)
    r2.cmd('zebra -f /usr/local/etc/r2zebra.conf -d -z
~/r2zebra.api -i ~/r2zebra.interface')
    r3.cmd('zebra -f /usr/local/etc/r3zebra.conf -d -z
~/r3zebra.api -i ~/r3zebra.interface')

    r1.cmd('ospfd -f /usr/local/etc/r1ospfd.conf -d -z
~/r1zebra.api -i ~/r1ospfd.interface')
    r2.cmd('ospfd -f /usr/local/etc/r2ospfd.conf -d -z
~/r2zebra.api -i ~/r2ospfd.interface')
    r3.cmd('ospfd -f /usr/local/etc/r3ospfd.conf -d -z
~/r3zebra.api -i ~/r3ospfd.interface')

def run():
    "Uji router Linux"
    topo = NetworkTopo()
    net = Mininet(controller=RemoteController('ryu',
ip='127.0.0.1', port=6633), topo=topo, switch=OVSSwitch)
    net.start()
    info('*** Tabel Routing pada Router:\n')

    topo.configure_control_plane(net)

    # Verifikasi koneksi controller
    for switch in net.switches:
        switch.cmd('ovs-vsctl set-controller %s tcp:127.0.0.1:6633' %
switch.name)

    # Tunggu beberapa detik agar topologi muncul di Ryu
    time.sleep(10)

    # Jalankan iperf untuk menghasilkan trafik UDP
    info('*** Menghasilkan trafik UDP:\n')
    h1, h2, h3 = net.get('h1', 'h2', 'h3')
    h1.cmd('iperf -s -u -i 1 -p 5001 -w 128K -f k &')
    time.sleep(1) # Tunggu server untuk memulai
    h2.cmd('iperf -c 10.0.1.100 -u -i 1 -p 5001 -w 128K -f k -b
128M &')

    # Tambahkan rute default untuk h3
    h3.cmd('ip route add default via 10.0.3.1')

```

```

CLI(net)

net.stop()

os.system("killall -9 ospfd zebra")
os.system("rm -f *api*")
os.system("rm -f *interface*")

if __name__ == '__main__':
    setLogLevel('info')
    run()

```

Lampiran 2 Script OSPF dan OpenDaylight pada Mininet

```

from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import Node, RemoteController, OVSSwitch
from mininet.log import setLogLevel, info
from mininet.cli import CLI
import time
import os

class LinuxRouter(Node):
    "Node dengan IP forwarding diaktifkan."

    def config(self, **params):
        super(LinuxRouter, self).config(**params)
        # Aktifkan forwarding pada router
        self.cmd('sysctl net.ipv4.ip_forward=1')

    def terminate(self):
        self.cmd('sysctl net.ipv4.ip_forward=0')
        super(LinuxRouter, self).terminate()

class NetworkTopo(Topo):
    "Sebuah LinuxRouter yang menghubungkan tiga subnet IP"

    def build(self, **_opts):
        # Alamat IP untuk antarmuka router
        r1_eth1_ip = '10.0.1.1/24' # r1 ke h1
        r2_eth1_ip = '10.0.2.1/24' # r2 ke h2
        r3_eth1_ip = '10.0.3.1/24' # r3 ke h3

        r1_eth2_ip = '10.0.10.1/24' # r1 ke r2

```

```

r2_eth2_ip = '10.0.10.2/24' # r2 ke r1

r1_eth3_ip = '10.0.11.1/24' # r1 ke r3
r3_eth2_ip = '10.0.11.2/24' # r3 ke r1

r2_eth3_ip = '10.0.12.1/24' # r2 ke r3
r3_eth3_ip = '10.0.12.2/24' # r3 ke r2

# Tambahkan alamat IP untuk antarmuka switch ke router
s1_eth1_ip = '10.0.1.2/24' # s1 ke r1
s2_eth1_ip = '10.0.2.2/24' # s2 ke r2
s3_eth1_ip = '10.0.3.2/24' # s3 ke r3

# Menambahkan router
r1 = self.addHost('r1', cls=LinuxRouter, ip=r1_eth1_ip)
r2 = self.addHost('r2', cls=LinuxRouter, ip=r2_eth1_ip)
r3 = self.addHost('r3', cls=LinuxRouter, ip=r3_eth1_ip)

# Menambahkan host
h1 = self.addHost('h1', ip='10.0.1.100/24', defaultRoute='via
10.0.1.1')
h2 = self.addHost('h2', ip='10.0.2.100/24', defaultRoute='via
10.0.2.1')
h3 = self.addHost('h3', ip='10.0.3.100/24', defaultRoute='via
10.0.3.1')

# Menambahkan switch
s1 = self.addSwitch('s1', cls=OVSSwitch,
protocols='OpenFlow13')
s2 = self.addSwitch('s2', cls=OVSSwitch,
protocols='OpenFlow13')
s3 = self.addSwitch('s3', cls=OVSSwitch,
protocols='OpenFlow13')

# Membuat link dengan alamat IP untuk antarmuka switch ke
router
self.addLink(s1, r1, intfName1='s1-eth1', params1={'ip':
s1_eth1_ip}, intfName2='r1-eth1', params2={'ip': r1_eth1_ip})
# Switch s1 ke Router r1
self.addLink(s2, r2, intfName1='s2-eth1', params1={'ip':
s2_eth1_ip}, intfName2='r2-eth1', params2={'ip': r2_eth1_ip})
# Switch s2 ke Router r2
self.addLink(s3, r3, intfName1='s3-eth1', params1={'ip':
s3_eth1_ip}, intfName2='r3-eth1', params2={'ip': r3_eth1_ip})
# Switch s3 ke Router r3

self.addLink(r1, h1, intfName1='r1-eth4', params1={'ip':
'10.0.1.1/24'}) # Router r1 ke Host h1

```

```

self.addLink(r2, h2, intfName1='r2-eth4', params1={'ip':
'10.0.2.1/24'}) # Router r2 ke Host h2
self.addLink(r3, h3, intfName1='r3-eth4', params1={'ip':
'10.0.3.1/24'}) # Router r3 ke Host h3

self.addLink(r1, r2, intfName1='r1-eth2', intfName2='r2-
eth2', params1={'ip': r1_eth2_ip}, params2={'ip': r2_eth2_ip})
self.addLink(r1, r3, intfName1='r1-eth3', intfName2='r3-
eth2', params1={'ip': r1_eth3_ip}, params2={'ip': r3_eth2_ip})
self.addLink(r2, r3, intfName1='r2-eth3', intfName2='r3-
eth3', params1={'ip': r2_eth3_ip}, params2={'ip': r3_eth3_ip})

# Link tambahan untuk menghubungkan switch
self.addLink(s1, s2)
self.addLink(s2, s3)

def configure_control_plane(self, net):
    "Konfigurasi zebra dan ospf untuk setiap router"
    r1 = net.getNodeByName('r1')
    r2 = net.getNodeByName('r2')
    r3 = net.getNodeByName('r3')

    r1.cmd('zebra -f /usr/local/etc/r1zebra.conf -d -z
~/r1zebra.api -i ~/r1zebra.interface')
    time.sleep(1)
    r2.cmd('zebra -f /usr/local/etc/r2zebra.conf -d -z
~/r2zebra.api -i ~/r2zebra.interface')
    r3.cmd('zebra -f /usr/local/etc/r3zebra.conf -d -z
~/r3zebra.api -i ~/r3zebra.interface')

    r1.cmd('ospfd -f /usr/local/etc/r1ospfd.conf -d -z
~/r1zebra.api -i ~/r1ospfd.interface')
    r2.cmd('ospfd -f /usr/local/etc/r2ospfd.conf -d -z
~/r2zebra.api -i ~/r2ospfd.interface')
    r3.cmd('ospfd -f /usr/local/etc/r3ospfd.conf -d -z
~/r3zebra.api -i ~/r3ospfd.interface')

def run():
    "Uji router Linux"
    topo = NetworkTopo()
    net = Mininet(controller=RemoteController('odl',
ip='127.0.0.1', port=6653), topo=topo, switch=OVSSwitch)
    net.start()
    info('*** Tabel Routing pada Router:\n')

    topo.configure_control_plane(net)

# Verifikasi koneksi controller
for switch in net.switches:

```



```

switch.cmd('ovs-vsctl set-controller %s tcp:127.0.0.1:6653' %
switch.name)

# Tunggu beberapa detik agar topologi muncul di OpenDaylight
time.sleep(10)

# Jalankan iperf untuk menghasilkan trafik UDP
info('*** Menghasilkan trafik UDP:\n')
h1, h2, h3 = net.get('h1', 'h2', 'h3')
h1.cmd('iperf -s -u -i 1 -p 5001 -w 128K -f k &')
time.sleep(1) # Tunggu server untuk memulai
h2.cmd('iperf -c 10.0.1.100 -u -i 1 -p 5001 -w 128K -f k -b
128M &')

# Tambahkan rute default untuk h3
h3.cmd('ip route add default via 10.0.3.1')

CLI(net)

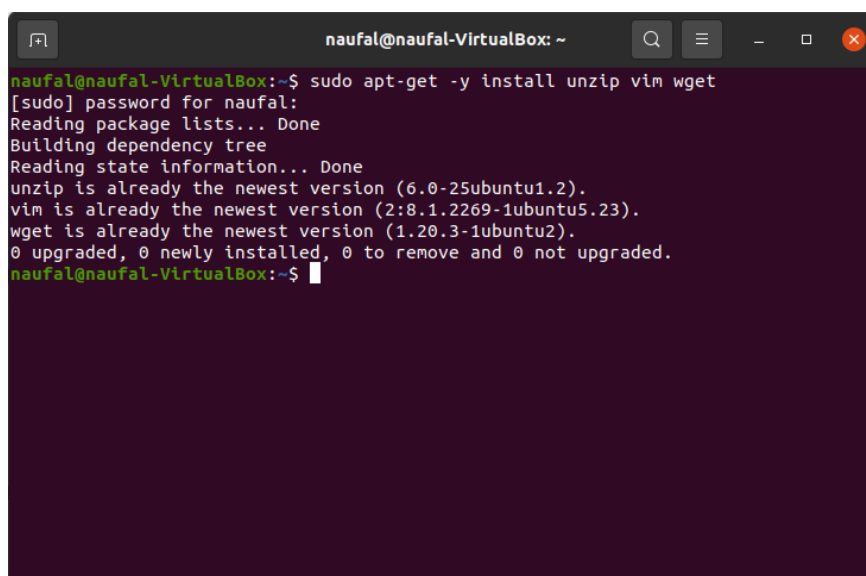
net.stop()

os.system("killall -9 ospfd zebra")
os.system("rm -f *api*")
os.system("rm -f *interface*")

if __name__ == '__main__':
    setLogLevel('info')
    run()

```

Lampiran 3 Tampilan Sistem



```

naufal@naufal-VirtualBox: ~
naufal@naufal-VirtualBox:~$ sudo apt-get -y install unzip vim wget
[sudo] password for naufal:
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-25ubuntu1.2).
vim is already the newest version (2:8.1.2269-1ubuntu5.23).
wget is already the newest version (1.20.3-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
naufal@naufal-VirtualBox:~$

```

Lampiran 3.1 Instal Dependency OpenDaylight

```

naufal@naufal-VirtualBox: ~
unzip is already the newest version (6.0-25ubuntu1.2).
vim is already the newest version (2:8.1.2269-1ubuntu5.23).
wget is already the newest version (1.20.3-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
naufal@naufal-VirtualBox:~$ sudo apt-get -y install openjdk-8-jre
[sudo] password for naufal:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jre is already the newest version (8u412-ga-1~20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
naufal@naufal-VirtualBox:~$ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

   Selection    Path                                            Priority  Status
-----
* 0             /usr/lib/jvm/java-11-openjdk-amd64/bin/java  1111    auto m
ode
  1             /usr/lib/jvm/java-11-openjdk-amd64/bin/java  1111    manual
mode
  2             /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java 1081    manual
mode

Press <enter> to keep the current choice[*], or type selection number: █

```

Lampiran 3.2 Instalasi Jre

```

naufal@naufal-VirtualBox: /usr/local/karaf
naufal@naufal-VirtualBox:~/Opendaylight$ cd
naufal@naufal-VirtualBox:~$ cd /usr/local/karaf/
naufal@naufal-VirtualBox:/usr/local/karaf$ sudo update-alternatives --install /u
sr/bin/karaf karaf /usr/local/karaf/karaf-0.8.4/bin/karaf 1
[sudo] password for naufal:
naufal@naufal-VirtualBox:/usr/local/karaf$ sudo update-alternatives --config kar
af
There is only one alternative in link group karaf (providing /usr/bin/karaf): /u
sr/local/karaf/karaf-0.8.4/bin/karaf
Nothing to configure.
naufal@naufal-VirtualBox:/usr/local/karaf$ which karaf
/usr/bin/karaf
naufal@naufal-VirtualBox:/usr/local/karaf$ █

```

Lampiran 3.3 Instalasi Karaf atau OpenDaylight

```

naufal@naufal-VirtualBox:~/mininet$ sudo mn --test pingall
*** 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 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.491 seconds
naufal@naufal-VirtualBox:~/mininet$

```

Lampiran 3.4 Instalasi Mininet

```

naufal@naufal-VirtualBox: ~/ryu
naufal@naufal-VirtualBox:~$ cd ryu
naufal@naufal-VirtualBox:~/ryu$ ryu-manager
loading app ryu.controller.ofp_handler
instantiating app ryu.controller.ofp_handler of OFPHandler
hub: uncaught exception: Traceback (most recent call last):
  File "/home/naufal/.local/lib/python3.8/site-packages/ryu/lib/hub.py", line 60
    , in _launch
      return func(*args, **kwargs)
  File "/home/naufal/.local/lib/python3.8/site-packages/ryu/controller/controlle
r.py", line 204, in server_loop
    server = StreamServer((CONF.ofp_listen_host,
  File "/home/naufal/.local/lib/python3.8/site-packages/ryu/lib/hub.py", line 12
7, in __init__
    self.server = eventlet.listen(listen_info)
  File "/home/naufal/.local/lib/python3.8/site-packages/eventlet/convenience.py"
, line 78, in listen
    sock.bind(addr)
OSError: [Errno 98] Address already in use
hub: uncaught exception: Traceback (most recent call last):
  File "/home/naufal/.local/lib/python3.8/site-packages/ryu/lib/hub.py", line 60
    , in _launch
      return func(*args, **kwargs)
  File "/home/naufal/.local/lib/python3.8/site-packages/ryu/controller/controlle

```

Lampiran 3.5 Instalasi RYU

```

naufal@naufal-VirtualBox: ~/mininet
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Done
completed in 67.520 seconds
naufal@naufal-VirtualBox:~/mininet$ sudo mn --custom topo.py --topo project --
controller=remote,ip=127.0.0.1,port=6653
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (h5, s3) (h6, s3) (h7, s4) (h8, s4) (h9, s
s) (h10, s5) (h11, s6) (h12, s6) (s1, s2) (s3, s1) (s4, s1) (s5, s2) (s6, s2)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Starting CLI:
mininet>

```

Lampiran 3.6 Intergrasi Mininet dan OpenDaylight

```

naufal@naufal-VirtualBox
(h10, s5) (h11, s6) (h12, s6) (s1, s2) (s3, s
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Results: 0% dropped (132/132 received)
mininet>
naufal@naufal-VirtualBox: ~/ryu
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
EVENT ofp_event->switches EventOFPPacketIn
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn

```

Lampiran 3.7 Intergrasi RYU dan Mininet

```

● opendaylight.service - OpenDaylight Controller
   Loaded: loaded (/etc/systemd/system/opendaylight.service; enabled; v>
   Active: active (running) since Thu 2024-06-13 12:53:55 WITA; 1min 19>
   Main PID: 5303 (karaf)
     Tasks: 136 (limit: 4598)
    Memory: 1.1G
   CGroup: /system.slice/opendaylight.service
           └─5303 /bin/sh /usr/bin/karaf server
           └─5373 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java -Djava>

Jun 13 12:53:55 naufal-VirtualBox systemd[1]: Started OpenDaylight Contro>
Jun 13 12:53:55 naufal-VirtualBox karaf[5304]: link: /etc/alternatives/ka>
Jun 13 12:53:55 naufal-VirtualBox karaf[5304]: link: /usr/local/karaf/kar>
Jun 13 12:53:56 naufal-VirtualBox karaf[5373]: Apache Karaf starting up.>
Jun 13 12:54:34 naufal-VirtualBox karaf[5373]: [7.8K blob data]
Jun 13 12:54:34 naufal-VirtualBox karaf[5373]: Karaf started in 35s. Bund>

~
~
~
lines 1-16/16 (END)

```

Lampiran 3.8 Mengaktifkan OpenDaylight Service

```

naufal@naufal-VirtualBox: /usr/local/karaf$ sudo -E karaf
link: /etc/alternatives/karaf
link: /usr/local/karaf/karaf-0.8.4/bin/karaf
Apache Karaf starting up. Press Enter to open the shell now...
100% [=====]
Karaf started in 43s. Bundle stats: 419 active, 420 total

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.

opendaylight-user@root>

```

Lampiran 3.9 Run OpenDaylight

```

"Node: h1"
25] 42,0-43,0 sec 16385 KBytes 134229 Kbits/sec 0,003 ms 22/11436 (0,19%)
25] 43,0-44,0 sec 15769 KBytes 129184 Kbits/sec 0,004 ms 428/11413 (3,8%)
25] 44,0-45,0 sec 16143 KBytes 132241 Kbits/sec 0,012 ms 168/11413 (1,5%)
25] 45,0-46,0 sec 15866 KBytes 129972 Kbits/sec 0,007 ms 362/11414 (3,2%)
25] 46,0-47,0 sec 16092 KBytes 131830 Kbits/sec 0,007 ms 198/11408 (1,7%)
25] 47,0-48,0 sec 16127 KBytes 132112 Kbits/sec 0,005 ms 178/11412 (1,6%)
25] 48,0-49,0 sec 16180 KBytes 132547 Kbits/sec 0,048 ms 141/11412 (1,2%)
25] 49,0-50,0 sec 16130 KBytes 132135 Kbits/sec 0,011 ms 58/11294 (0,51%)
25] 50,0-51,0 sec 16391 KBytes 134276 Kbits/sec 0,008 ms 111/11529 (0,96%)
25] 51,0-52,0 sec 15870 KBytes 130007 Kbits/sec 0,011 ms 260/11315 (2,3%)
25] 52,0-53,0 sec 16181 KBytes 132559 Kbits/sec 0,008 ms 248/11520 (2,2%)
25] 53,0-54,0 sec 16154 KBytes 132335 Kbits/sec 0,009 ms 152/11405 (1,3%)
25] 54,0-55,0 sec 16144 KBytes 132253 Kbits/sec 0,009 ms 130/11376 (1,1%)
25] 55,0-56,0 sec 16207 KBytes 132770 Kbits/sec 0,014 ms 169/11459 (1,5%)
25] 56,0-57,0 sec 16028 KBytes 131300 Kbits/sec 0,007 ms 214/11379 (1,9%)
25] 57,0-58,0 sec 16176 KBytes 132512 Kbits/sec 0,009 ms 138/11406 (1,2%)
25] 58,0-59,0 sec 16108 KBytes 131959 Kbits/sec 0,008 ms 230/11451 (2%)
25] 0,0-60,0 sec 964031 KBytes 131623 Kbits/sec 0,008 ms 13244/684787 (1,9%)
25] 0,0000-59,9999 sec 483 datagrams received out-of-order

"Node: h2"
25] 43,0-44,0 sec 16382 KBytes 134205 Kbits/sec
25] 44,0-45,0 sec 16377 KBytes 134158 Kbits/sec
25] 45,0-46,0 sec 16394 KBytes 134299 Kbits/sec
25] 46,0-47,0 sec 16381 KBytes 134193 Kbits/sec
25] 47,0-48,0 sec 16382 KBytes 134205 Kbits/sec
25] 48,0-49,0 sec 16382 KBytes 134205 Kbits/sec
25] 49,0-50,0 sec 16213 KBytes 132817 Kbits/sec
25] 50,0-51,0 sec 16550 KBytes 135581 Kbits/sec
25] 51,0-52,0 sec 16243 KBytes 133064 Kbits/sec
25] 52,0-53,0 sec 16535 KBytes 135452 Kbits/sec
25] 53,0-54,0 sec 16375 KBytes 134146 Kbits/sec
25] 54,0-55,0 sec 16331 KBytes 133782 Kbits/sec
25] 55,0-56,0 sec 16448 KBytes 134746 Kbits/sec
25] 56,0-57,0 sec 16337 KBytes 133829 Kbits/sec
25] 57,0-58,0 sec 16374 KBytes 134135 Kbits/sec
25] 58,0-59,0 sec 16438 KBytes 134664 Kbits/sec
25] 59,0-60,0 sec 16385 KBytes 134229 Kbits/sec
25] 0,0-60,0 sec 983044 KBytes 134218 Kbits/sec
25] Sent 684787 datagrams
25] Server Report:
25] 0,0-60,0 sec 964031 KBytes 131623 Kbits/sec 0,007 ms 13244/684787 (1,9%)
25] 0,0000-59,9999 sec 483 datagrams received out-of-order
root@naufal-VirtualBox: /home/naufal/mininet#

```

Lampiran 3.10 Pengujian Throughput

```
root@naufal-VirtualBox:/home/naufal/mininet# ping 10.0.1.100 -s 1024 -c 10
PING 10.0.1.100 (10.0.1.100) 1024(1052) bytes of data.
1032 bytes from 10.0.1.100: icmp_seq=1 ttl=62 time=0,123 ms
1032 bytes from 10.0.1.100: icmp_seq=2 ttl=62 time=0,096 ms
1032 bytes from 10.0.1.100: icmp_seq=3 ttl=62 time=0,108 ms
1032 bytes from 10.0.1.100: icmp_seq=4 ttl=62 time=0,105 ms
1032 bytes from 10.0.1.100: icmp_seq=5 ttl=62 time=0,097 ms
1032 bytes from 10.0.1.100: icmp_seq=6 ttl=62 time=0,113 ms
1032 bytes from 10.0.1.100: icmp_seq=7 ttl=62 time=0,068 ms
1032 bytes from 10.0.1.100: icmp_seq=8 ttl=62 time=0,083 ms
1032 bytes from 10.0.1.100: icmp_seq=9 ttl=62 time=0,083 ms
1032 bytes from 10.0.1.100: icmp_seq=10 ttl=62 time=0,080 ms

--- 10.0.1.100 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9217ms
rtt min/avg/max/mdev = 0,068/0,095/0,123/0,016 ms
root@naufal-VirtualBox:/home/naufal/mininet# █
```

Lampiran 3.11 Pengujian *Latency*



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 29045/UN4.7.1/TD.06/2023

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : 1. Dr.Eng. Ir. Muhammad Niswar, ST., M.IT Pemb. I
2. Dr.Eng. Zulkifli Tahir, ST., M.Sc. Pemb. II

Isi : 1. Berdasarkan Surat Ketua Departemen Teknik Informatika Fakultas Teknik Nomor. 1427/UN4.7.7./TD.06/2023 tanggal 12 Desember 2023 tentang usul DOSEN PEMBIMBING MAHASISWA, maka dengan ini kami menugaskan Saudara untuk membimbing penulisan Skripsi/Tugas Akhir mahasiswa Teknik Informatika Fakultas Teknik Universitas Hasanuddin di bawah ini :

N a m a : Muhammad Naufal Faliq No. Stambuk : D121 17 1503

Judul Skripsi/Tugas Akhir :

“ Analisis Kinerja Routing Protokol OSPF Menggunakan Controller Ryu dan Opendaylight pada Jaringan Software Defined Network (SDN) ”

2. Surat penugasan pembimbing ini mulai berlaku sejak tanggal ditetapkannya dan berakhir sampai selesainya penulisan Skripsi/Tugas Akhir mahasiswa tersebut.
3. Agar penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

Ditetapkan di Gowa
Pada tanggal 12 Desember 2023
a.n. Dekan,
Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP. 197310101998021001

Tembusan :

1. Dekan FT-UH,
2. Ketua Departemen Teknik Informatika FT-UH,
3. Mahasiswa yang bersangkutan



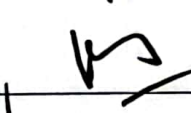

• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1

"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"










KARTU BIMBINGAN SKRIPSI

Prodi SI Teknik Informatika Universitas Hasanuddin

Stb.	Nama Mahasiswa
D121171503	Muh. Naufal Faliq

Pembimbing	Nama Pembimbing	Paraf & Tgl. Persetujuan Ujian Akhir
I	Dr. Eng. Muhammad Niswar, S.T., M.IT	
II	Dr. Eng. Zulkifli Tahir, S.T., M.Sc	
No. SK Pemb		

Judul Skripsi	ANALISIS KINERJA PROTOKOL ROUTING OSPF MENGGUNAKAN CONTROLLER RYU DAN OPENDAYLIGHT PADA JARINGAN SOFTWARE DEFINED NETWORK (SDN)
---------------	---

No	Tanggal Bimbingan	Uraian Kegiatan Bimbingan	Paraf Pemb.
1	19 Maret 2024	Bimbingan mengenai sistem	
2	27 April 2024	Konsultasi mengenai topologi	
3	13 Mei 2024	Presentasi sistem	
4	3 Juni 2024	Presentasi mengenai OSPF pada sistem	
5			
6			
7			
8	23 Februari 2024	Konsultasi mengenai sistem	
9	15 Maret 2024	Konsultasi mengenai pengujian parameter	
10	14 Mei 2024	Konsultasi sistem	
11	4 Juni 2024	Konsul penulisan	
12	14 Juni 2024	Konsultasi penulisan	
13			
14			
15			

LEMBAR PERBAIKAN SKRIPSI


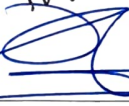
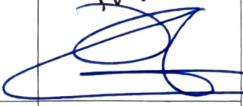

“ANALISIS KINERJA PROTOKOL ROUTING OSPF MENGUNAKAN CONTROLLER RYU DAN OPENDAYLIGHT PADA JARINGAN SOFTWARE DEFINED NETWORK (SDN)”

OLEH:


MUHAMMAD NAUFAL FALIQ
D121171520

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana pada tanggal 31 Juli 2024.
Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr-Eng.Ir. Muhammad Niswar, ST, M.InfoTech	
Sekretaris	Dr.Eng. Zulkifli Tahir, S.T., M.Sc.	
Anggota	Adnan, S.T, M.T, Ph.D	
	Iqra Aswad, S.T., M.T.	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr-Eng.Ir. Muhammad Niswar, ST, M.InfoTech	
II	Dr.Eng. Zulkifli Tahir, S.T., M.Sc.	