Laporan Tugas Besar
Mata Kuliah Jaringan Komputer
“Topology Mininet”

Laporan ini dibuat dengan tujuan untuk memenuhi tugas mata kuliah Jaringan Komputer

Dosen Pengampu :

Disusun oleh :
Kelvyn Lukito - 1301200104

Laporan Tugas Besar
Program Studi Informatika
Fakultas Informatika
Universitas Telkom

2022

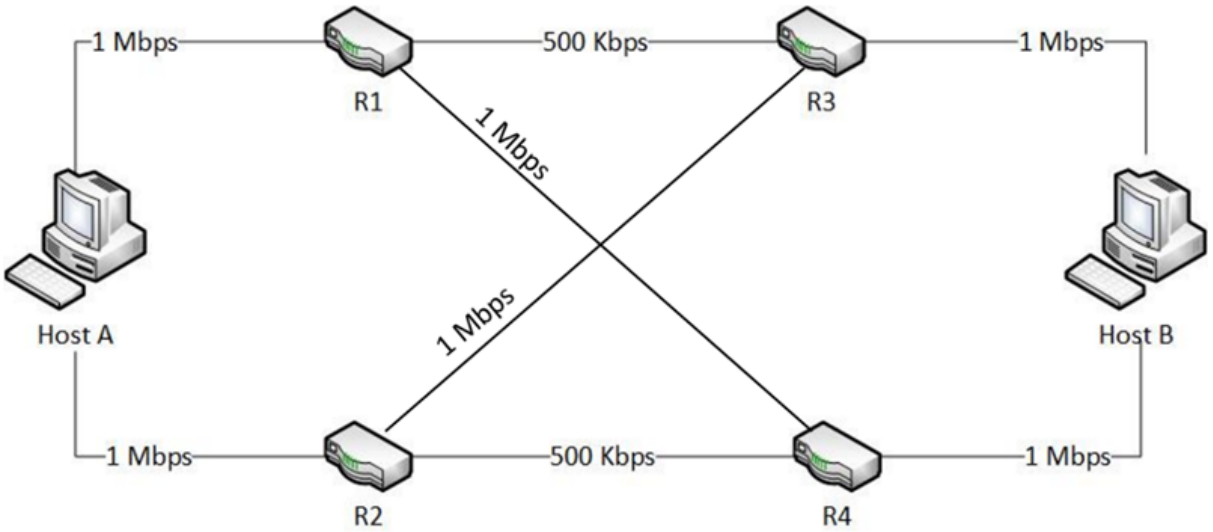
Daftar Isi

Spesifikasi Tugas Besar

Jaringan Komputer Semester Genap TA 2021/2022

1.1 Simulasi Pada Mininet

Topologi yang digunakan pada skenario ini dapat dilihat pada gambar 12.1.

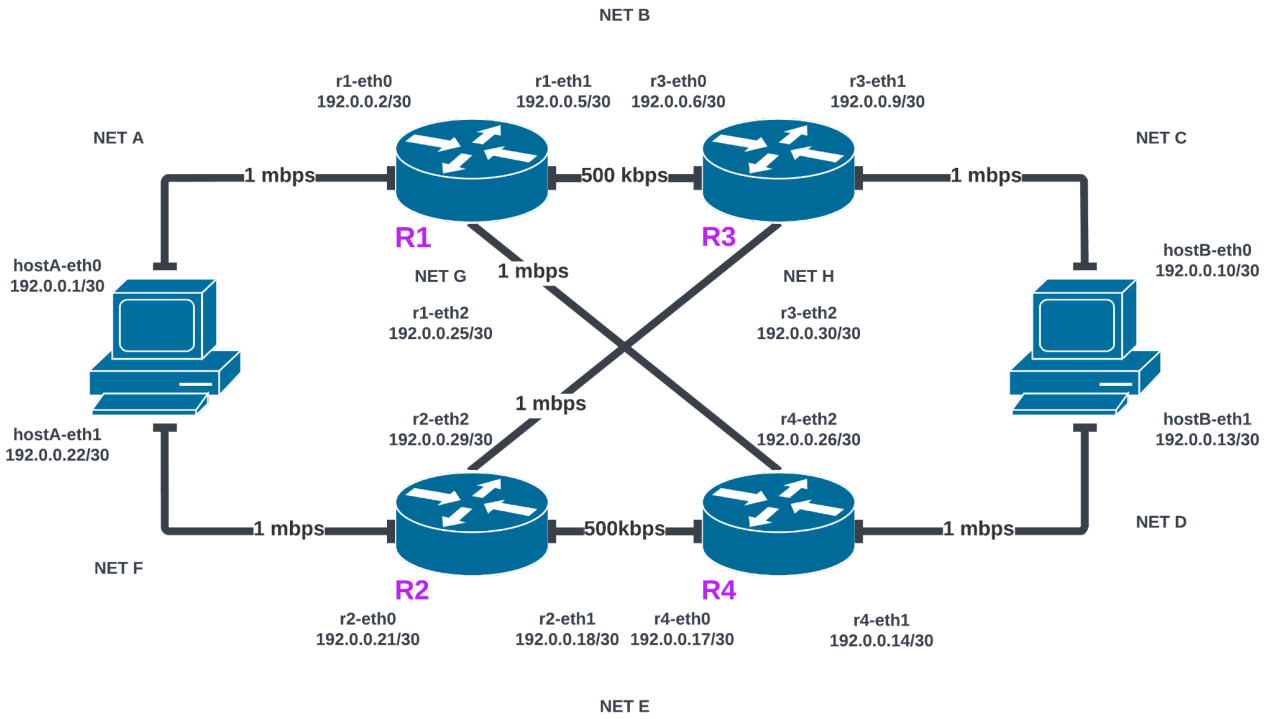


Gambar 12.1 Topologi untuk tugas besar

1) CLO 1

Pada CLO ini terdapat spesifikasi pengerjaan dan kriteria penilaian yang akan dilakukan.

- **Goal :** Build topology sesuai dengan soal.
- https://lucid.app/lucidchart/6c33c683-9dd9-497b-a5bd-9f3adeca2226/edit?viewport_loc=10%2C150%2C2220%2C1030%2C0_0&invitationId=inv_749f72f2-b2f7-4a63-9d25-6cbc80a6364e#



- Desain subnet masing-masing network

Subnet Name	Needed Size	Allocated Size	Address	Mask	Dec Mask	Assignable Range	Broadcast
A	2	2	192.0.0.0	/30	255.255.255.252	192.0.0.1 - 192.0.0.2	192.0.0.3
B	2	2	192.0.0.4	/30	255.255.255.252	192.0.0.5 - 192.0.0.6	192.0.0.7
C	2	2	192.0.0.8	/30	255.255.255.252	192.0.0.9 - 192.0.0.10	192.0.0.11
D	2	2	192.0.0.12	/30	255.255.255.252	192.0.0.13 - 192.0.0.14	192.0.0.15
E	2	2	192.0.0.16	/30	255.255.255.252	192.0.0.17 - 192.0.0.18	192.0.0.19
F	2	2	192.0.0.20	/30	255.255.255.252	192.0.0.21 - 192.0.0.22	192.0.0.23
G	2	2	192.0.0.24	/30	255.255.255.252	192.0.0.25 - 192.0.0.26	192.0.0.27
H	2	2	192.0.0.28	/30	255.255.255.252	192.0.0.29 - 192.0.0.30	192.0.0.31

192.0.0.0/24

NET	Need	Allocated	Address	Mask	Deck Mask	Range	Broadcast
A	2	2	192.0.0.0	/30	255.255.255.252	192.0.0.1 - 192.0.0.2	192.0.0.3
B	2	2	192.0.0.4	/30	255.255.255.252	192.0.0.5 - 192.0.0.6	192.0.0.7
C	2	2	192.0.0.8	/30	255.255.255.252	192.0.0.9 - 192.0.0.10	192.0.0.11
D	2	2	192.0.0.12	/30	255.255.255.252	192.0.0.13 - 192.0.0.14	192.0.0.15
E	2	2	192.0.0.16	/30	255.255.255.252	192.0.0.17 - 192.0.0.18	192.0.0.19
F	2	2	192.0.0.20	/30	255.255.255.252	192.0.0.21 -	192.0.0.23

						192.0.0.22	
G	2	2	192.0.0.24	/30	255.255.255.252	192.0.0.25 - 192.0.0.26	192.0.0.27
H	2	2	192.0.0.28	/30	255.255.255.252	192.0.0.29 - 192.0.0.30	192.0.0.31

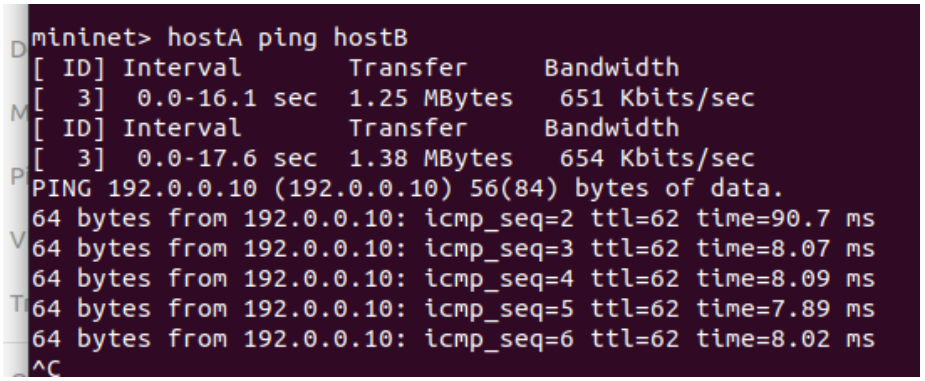
- Assign IP sesuai subnet.

```
def Topologi():
    # Konfigurasi mininet
    Link = TCLink
    host = CPULimitedHost
    net = Mininet(link=Link, host = host)
    key = "net.mptcp.mptcp_enabled"

    # Konfigurasi host
    hostA = net.addHost('hostA') #,ip='192.0.0.1/30


    hostB = net.addHost('hostB') #,ip='192.0.0.10/30

    # Konfigurasi Router
    r1 = net.addHost('r1', cls=LinuxRouter) #,ip='192.0.0.2/30'
    r2 = net.addHost('r2', cls=LinuxRouter) #,ip='192.0.0.21/30'
    r3 = net.addHost('r3', cls=LinuxRouter) #,ip='192.0.0.9/30'
    r4 = net.addHost('r4', cls=LinuxRouter) #,ip='192.0.0.14/30'


    # Connection specification
    linkopts0 = dict(bw=0.5, delay='1ms', loss=0, max_queue_size=20, use_tbf=True)
    linkopts1 = dict(bw=1, delay='1ms', loss=0, max_queue_size=20, use_tbf=True)

    # Configuration network
    # Router <--> Router

    net.addLink(r1, r3, cls=TCLink, bw=0.5, intfName1='r1-eth1', intfName2='r3-eth0') #,params1={'ip':
'192.0.0.5/30'},params2={'ip': '192.0.0.6/30'}

    net.addLink(r2, r4, cls=TCLink, bw=0.5, intfName1='r2-eth1', intfName2='r4-eth0') #,params1={'ip':
'192.0.0.18/30'},params2={'ip': '192.0.0.17/30'}

    net.addLink(r1, r4, cls=TCLink, bw=1, intfName1='r1-eth2', intfName2='r4-eth2') # ,params1={'ip':
'192.0.0.25/30'}, params2={'ip': '192.0.0.26/30'}

    net.addLink(r2, r3, cls=TCLink, bw=1, intfName1='r2-eth2', intfName2='r3-eth2') #,params1={'ip':
'192.0.0.29/30'}, params2={'ip': '192.0.0.30/30'}

    # Router <--> Host

    net.addLink(hostA, r1, cls=TCLink, bw=1, intfName1='hostA-eth0', intfName2='r1-eth0')
#,params1={'ip': '192.0.0.1/30'}, params2={'ip': '192.0.0.2/30'}
```

```
    net.addLink(hostA, r2, cls=TCLink, bw=1, intfName1='hostA-eth1', intfName2='r2-eth0')
#,params1={'ip': '192.0.0.22/30'}, params2={'ip': '192.0.0.21/30'}

    net.addLink(hostB, r3, cls=TCLink, bw=1, intfName1='hostB-eth0', intfName2='r3-eth1')
#,params1={'ip': '192.0.0.10/30'}, params2={'ip': '192.0.0.9/30'}

    net.addLink(hostB, r4, cls=TCLink, bw=1, intfName1='hostB-eth1', intfName2='r4-eth1')
#,params1={'ip': '192.0.0.13/30'}, params2={'ip': '192.0.0.14/30'}

    net.build()

    r1.cmd("sysctl net.ipv4.ip_forward=1")
    r2.cmd("sysctl net.ipv4.ip_forward=1")
    r3.cmd("sysctl net.ipv4.ip_forward=1")
    r4.cmd("sysctl net.ipv4.ip_forward=1")

    hostA.cmd('ifconfig hostA-eth0 0')
    hostA.cmd('ifconfig hostA-eth0 192.0.0.1 netmask 255.255.255.252')

    hostA.cmd('ifconfig hostA-eth1 1')
    hostA.cmd('ifconfig hostA-eth1 192.0.0.22 netmask 255.255.255.252')

    hostB.cmd('ifconfig hostB-eth0 0')
    hostB.cmd('ifconfig hostB-eth0 192.0.0.10 netmask 255.255.255.252')

    hostB.cmd('ifconfig hostB-eth1 1')
    hostB.cmd('ifconfig hostB-eth1 192.0.0.13 netmask 255.255.255.252')

    r1.cmd('ifconfig r1-eth0 0')
    r1.cmd('ifconfig r1-eth0 192.0.0.2 netmask 255.255.255.252')
    r1.cmd('ifconfig r1-eth1 0')
    r1.cmd('ifconfig r1-eth1 192.0.0.5 netmask 255.255.255.252')
    r1.cmd('ifconfig r1-eth2 0')
    r1.cmd('ifconfig r1-eth2 192.0.0.25 netmask 255.255.255.252')

    r2.cmd('ifconfig r2-eth0 0')
    r2.cmd('ifconfig r2-eth0 192.0.0.21 netmask 255.255.255.252')
    r2.cmd('ifconfig r2-eth1 0')
    r2.cmd('ifconfig r2-eth1 192.0.0.18 netmask 255.255.255.252')
    r2.cmd('ifconfig r2-eth2 0')
    r2.cmd('ifconfig r2-eth2 192.0.0.29 netmask 255.255.255.252')

    r3.cmd('ifconfig r3-eth0 0')
    r3.cmd('ifconfig r3-eth0 192.0.0.6 netmask 255.255.255.252')
    r3.cmd('ifconfig r3-eth1 0')
    r3.cmd('ifconfig r3-eth1 192.0.0.9 netmask 255.255.255.252')
    r3.cmd('ifconfig r3-eth2 0')
    r3.cmd('ifconfig r3-eth2 192.0.0.30 netmask 255.255.255.252')

    r4.cmd('ifconfig r4-eth0 0')
    r4.cmd('ifconfig r4-eth0 192.0.0.17 netmask 255.255.255.252')
    r4.cmd('ifconfig r4-eth1 0')
    r4.cmd('ifconfig r4-eth1 192.0.0.14 netmask 255.255.255.252')
```

```
r4.cmd('ifconfig r4-eth2 0')
r4.cmd('ifconfig r4-eth2 192.0.0.26 netmask 255.255.255.252')


#Routing pada Host 1
hostA.cmd("ip rule add from 192.0.0.1 table 1")
hostA.cmd("ip rule add from 192.0.0.22 table 2")
hostA.cmd("ip route add 192.0.0.0/30 dev hostA-eth0 scope link table 1")
#hostA.cmd("ip route add default via 192.168.1.1 dev hostA-eth0 table 1")
hostA.cmd("ip route add 192.0.0.20/30 dev hostA-eth1 scope link table 2")
#hostA.cmd("ip route add default via 192.168.92.1 dev hostA-eth0 table 2")
hostA.cmd("ip route add default scope global nexthop via 192.0.0.2 dev hostA-eth0")

#Routing pada Host 2
hostB.cmd("ip rule add from 192.0.0.10 table 3")
hostB.cmd("ip rule add from 192.0.0.13 table 4")
hostB.cmd("ip route add 192.0.0.8/30 dev hostB-eth0 scope link table 3")
#hostB.cmd("ip route add default via 192.168.1.1 dev hostB-eth0 table 1")
hostB.cmd("ip route add 192.0.0.12/30 dev hostB-eth1 scope link table 4")
#hostB.cmd("ip route add default via 192.168.92.1 dev hostA-eth0 table 2")
hostB.cmd("ip route add default scope global nexthop via 192.0.0.14 dev hostB-eth1")
```

- Uji konektivitas dengan ping antara 2 host yang berada dalam 1 network.



- **Penilaian yang akan dilakukan adalah :**
  - Kesesuaian topologi yang dibangun dengan soal yang diberikan (30).
  - Ketepatan penjelasan topologi yang dibangun (50).
  - Konektivitas antar host yang berada pada subnet yang sama (20).
  - NILAI TOTAL = 100.

2) CLO 2

Pada CLO ini terdapat spesifikasi pengerjaan dan kriteria penilaian yang akan dilakukan.

- **Goal :** Mengimplementasikan mekanisme Routing pada topologi yang ada.

```
#Routing pada Host 1
   hostA.cmd("ip rule add from 192.0.0.1 table 1")
   hostA.cmd("ip rule add from 192.0.0.22 table 2")
   hostA.cmd("ip route add 192.0.0.0/30 dev hostA-eth0 scope link table 1")
   hostA.cmd("ip route add default via 192.0.0.2 dev hostA-eth0 table 1")
   hostA.cmd("ip route add 192.0.0.20/30 dev hostA-eth1 scope link table 2")
   hostA.cmd("ip route add default via 192.0.0.21 dev hostA-eth1 table 2")
   hostA.cmd("ip route add default scope global nexthop via 192.0.0.21 dev hostA-eth1")
   hostA.cmd("ip route add default scope global nexthop via 192.0.0.2 dev hostA-eth0")

   #Routing pada Host 2
   hostB.cmd("ip rule add from 192.0.0.10 table 1")
   hostB.cmd("ip rule add from 192.0.0.13 table 2")
   hostB.cmd("ip route add 192.0.0.8/30 dev hostB-eth0 scope link table 1")
   hostB.cmd("ip route add default via 192.0.0.9 dev hostB-eth0 table 1")
   hostB.cmd("ip route add 192.0.0.12/30 dev hostB-eth1 scope link table 2")
   hostB.cmd("ip route add default via 192.0.0.14 dev hostB-eth1 table 2")
   hostB.cmd("ip route add default scope global nexthop via 192.0.0.14 dev hostB-eth1")
   hostB.cmd("ip route add default scope global nexthop via 192.0.0.9 dev hostB-eth0")




   #Routing Router1
   r1.cmd("ip rule add from 192.0.0.2 table 1")
   r1.cmd("ip rule add from 192.0.0.5 table 2")
   r1.cmd("ip rule add from 192.0.0.25 table 3")
   r1.cmd("ip route add 192.0.0.0/30 dev r1-eth0 scope link table 1")
   r1.cmd("ip route add default via 192.0.0.1 dev r1-eth0 table 1")
   r1.cmd("ip route add 192.0.0.4/30 dev r1-eth1 scope link table 2")
   r1.cmd("ip route add default via 192.0.0.6 dev r1-eth1 table 2")
   r1.cmd("ip route add 192.0.0.24/30 dev r1-eth2 scope link table 3")
   r1.cmd("ip route add default via 192.0.0.26 dev r1-eth2 table 3")
   r1.cmd("ip route add default scope global nexthop via 192.0.0.1 dev r1-eth0")
   r1.cmd("ip route add default scope global nexthop via 192.0.0.6 dev r1-eth1")
   r1.cmd("ip route add default scope global nexthop via 192.0.0.26 dev r1-eth2")

   #Routing Router2
   r2.cmd("ip rule add from 192.0.0.21 table 1")
   r2.cmd("ip rule add from 192.0.0.18 table 2")
   r2.cmd("ip rule add from 192.0.0.29 table 3")
   r2.cmd("ip route add 192.0.0.20/30 dev r2-eth0 scope link table 1")
   r2.cmd("ip route add default via 192.0.0.22 dev r2-eth0 table 1")
   r2.cmd("ip route add 192.0.0.16/30 dev r2-eth1 scope link table 2")
   r2.cmd("ip route add default via 192.0.0.17 dev r2-eth1 table 2")
   r2.cmd("ip route add 192.0.0.28/30 dev r2-eth2 scope link table 3")
   r2.cmd("ip route add default via 192.0.0.30 dev r2-eth2 table 3")
   r2.cmd("ip route add default scope global nexthop via 192.0.0.22 dev r2-eth0")
   r2.cmd("ip route add default scope global nexthop via 192.0.0.17 dev r2-eth1")
   r2.cmd("ip route add default scope global nexthop via 192.0.0.30 dev r2-eth2")

   #Routing Router3
```

```
r3.cmd("ip rule add from 192.0.0.6 table 1")
r3.cmd("ip rule add from 192.0.0.9 table 2")
r3.cmd("ip rule add from 192.0.0.30 table 3")
r3.cmd("ip route add 192.0.0.4/30 dev r3-eth0 scope link table 1")
r3.cmd("ip route add default via 192.0.0.5 dev r3-eth0 table 1")
r3.cmd("ip route add 192.0.0.8/30 dev r3-eth1 scope link table 2")
r3.cmd("ip route add default via 192.0.0.10 dev r3-eth1 table 2")
r3.cmd("ip route add 192.0.0.28/30 dev r3-eth2 scope link table 3")
r3.cmd("ip route add default via 192.0.0.21 dev r3-eth2 table 3")
r3.cmd("ip route add default scope global nexthop via 192.0.0.10 dev r3-eth1")
r3.cmd("ip route add default scope global nexthop via 192.0.0.5 dev r3-eth0")
r3.cmd("ip route add default scope global nexthop via 192.0.0.29 dev r3-eth2")

#Routing Router4
r4.cmd("ip rule add from 192.0.0.17 table 1")
r4.cmd("ip rule add from 192.0.0.14 table 2")
r4.cmd("ip rule add from 192.0.0.26 table 3")
r4.cmd("ip route add 192.0.0.16/30 dev r4-eth0 scope link table 1")
r4.cmd("ip route add default via 192.0.0.18 dev r4-eth0 table 1")
r4.cmd("ip route add 192.0.0.12/30 dev r4-eth1 scope link table 2")
r4.cmd("ip route add default via 192.0.0.13 dev r4-eth1 table 2")
r4.cmd("ip route add 192.0.0.24/30 dev r4-eth2 scope link table 3")
r4.cmd("ip route add default via 192.0.0.25 dev r4-eth2 table 3")
r4.cmd("ip route add default scope global nexthop via 192.0.0.13 dev r4-eth1")
r4.cmd("ip route add default scope global nexthop via 192.0.0.18 dev r4-eth0")
r4.cmd("ip route add default scope global nexthop via 192.0.0.25 dev r4-eth2")

r1.cmd("route add -net 192.0.0.8/30 gw 192.0.0.6")
r1.cmd("route add -net 192.0.0.12/30 gw 192.0.0.26")
r1.cmd("route add -net 192.0.0.28/30 gw 192.0.0.6")
r1.cmd("route add -net 192.0.0.16/30 gw 192.0.0.26")

r2.cmd("route add -net 192.0.0.8/30 gw 192.0.0.30")
r2.cmd("route add -net 192.0.0.12/30 gw 192.0.0.17")
r2.cmd("route add -net 192.0.0.4/30 gw 192.0.0.30")
r2.cmd("route add -net 192.0.0.24/30 gw 192.0.0.17")

r3.cmd("route add -net 192.0.0.0/30 gw 192.0.0.5")
r3.cmd("route add -net 192.0.0.16/30 gw 192.0.0.29")
r3.cmd("route add -net 192.0.0.20/30 gw 192.0.0.29")
r3.cmd("route add -net 192.0.0.24/30 gw 192.0.0.5")

r4.cmd("route add -net 192.0.0.4/30 gw 192.0.0.25")
r4.cmd("route add -net 192.0.0.28/30 gw 192.0.0.18")
r4.cmd("route add -net 192.0.0.20/30 gw 192.0.0.18")
r4.cmd("route add -net 192.0.0.0/30 gw 192.0.0.25")
```

- Uji konektivitas menggunakan ping.
- Membuat tabel routing di semua host, dibuktikan dengan ping antar host.

```
mininet> hostA ping hostB
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-16.1 sec  1.25 MBytes   651 Kbits/sec
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-17.6 sec  1.38 MBytes   654 Kbits/sec
PING 192.0.0.10 (192.0.0.10) 56(84) bytes of data.
64 bytes from 192.0.0.10: icmp_seq=2 ttl=62 time=90.7 ms
64 bytes from 192.0.0.10: icmp_seq=3 ttl=62 time=8.07 ms
64 bytes from 192.0.0.10: icmp_seq=4 ttl=62 time=8.09 ms
64 bytes from 192.0.0.10: icmp_seq=5 ttl=62 time=7.89 ms
64 bytes from 192.0.0.10: icmp_seq=6 ttl=62 time=8.02 ms
^C
```

- Menganalisis routing yang digunakan menggunakan traceroute

```
mininet> hostA traceroute hostB
------------------------------------------------------------
Client connecting to 192.0.0.10, TCP port 5001
Binding to local address 192.0.0.1
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.0.0.1 port 51851 connected with 192.0.0.10 port 5001
------------------------------------------------------------
Client connecting to 192.0.0.10, TCP port 5001
Binding to local address 192.0.0.22
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.0.0.22 port 44427 connected with 192.0.0.10 port 5001
traceroute to 192.0.0.10 (192.0.0.10), 30 hops max, 60 byte packets
 1  192.0.0.21 (192.0.0.21)  224.140 ms  224.019 ms  223.982 ms
 2  192.0.0.30 (192.0.0.30)  241.009 ms  240.982 ms  240.962 ms
 3  192.0.0.10 (192.0.0.10)  532.194 ms  533.335 ms  533.320 ms
mininet>
```

```
mininet> hostB traceroute hostA
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 192.0.0.10 port 5001 connected with 192.0.0.1 port 51851
[  5] local 192.0.0.10 port 5001 connected with 192.0.0.22 port 44427
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-34.0 sec  2.25 MBytes   555 Kbits/sec
[  4]  0.0-38.0 sec  1.88 MBytes   413 Kbits/sec
traceroute to 192.0.0.1 (192.0.0.1), 30 hops max, 60 byte packets
 1  192.0.0.14 (192.0.0.14)  7.141 ms  7.087 ms  7.072 ms
 2  192.0.0.25 (192.0.0.25)  21.060 ms  21.048 ms  21.038 ms
 3  192.0.0.1 (192.0.0.1)  25.411 ms  25.400 ms  25.388 ms
mininet>
```

- **Penilaian yang akan dilakukan adalah :**
  - Ketepatan implementasi routing sesuai spesifikasi yang ada (30).
  - Ketepatan penjelasan proses routing yang diimplementasikan (50).

- Konektivitas antar host yang berada pada subnet berbeda (20).
- NILAI TOTAL = 100.

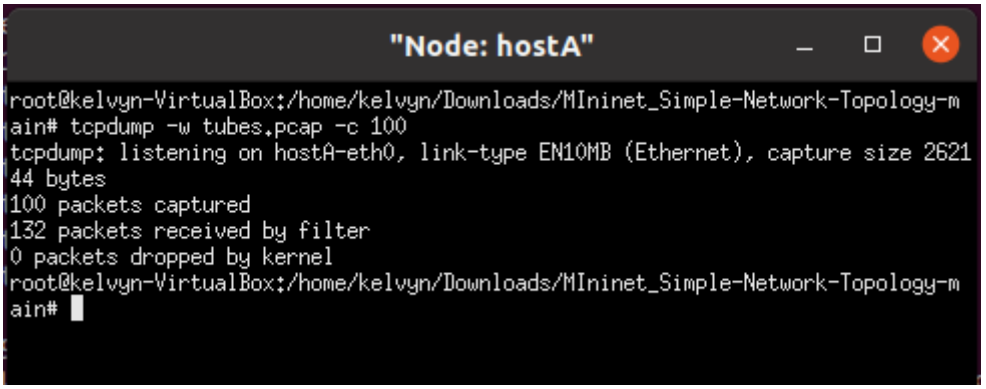
3) CLO 3

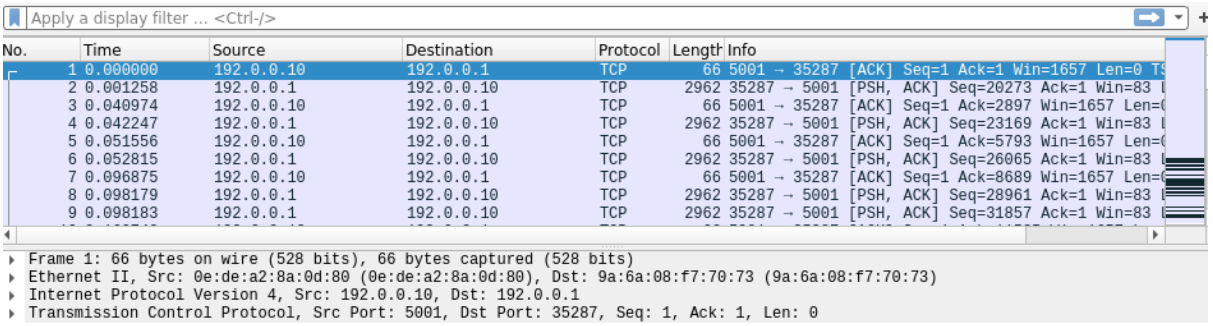
Pada CLO ini terdapat spesifikasi pengerjaan dan kriteria penilaian yang akan dilakukan.

- **Goal :** Membuktikan bahwa TCP telah diimplementasikan dengan benar pada topologi.
  - Generate *traffic* menggunakan iPerf.

```
mininet> iperf hostA r1
*** Iperf: testing TCP bandwidth between hostA and r1
*** Results: ['488 Kbits/sec', '847 Kbits/sec']
mininet>
```

  - Capture trafik menggunakan custom script atau Wireshark untuk diinspeksi, dibuktikan dengan trafik di Wireshark/tcpdump.





- **Penilaian yang akan dilakukan adalah :**
  - Ketepatan implementasi trafik TCP (40).
  - Ketepatan penjelasan apa itu trafik TCP dan perbedaannya dengan UDP (60).

- NILAI TOTAL = 100.

4) CLO 4

Pada CLO ini terdapat spesifikasi pengerjaan dan kriteria penilaian yang akan dilakukan.

- **Goal :** Menginspeksi penggunaan queue pada router jaringan.
  - Generate *traffic* menggunakan iPerf.

```
# Setting up traffic
hostB.cmd("iperf -s &")
hostA.cmd("iperf -t 30 -B 192.0.0.1 -c 192.0.0.10 &")
hostA.cmd("iperf -t 30 -B 192.0.0.22 -c 192.0.0.10 &")
```

  - Set ukuran buffer pada router : 20, 40, 60 dan 100.

```
# Buffer 20
# Configuration network
# Router <--> Router

    net.addLink(r1, r3, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=20, use_tbf=True,
intfName1='r1-eth1', intfName2='r3-eth0') #,params1={'ip': '192.0.0.5/30'},params2={'ip':
'192.0.0.6/30'}

    net.addLink(r2, r4, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=20, use_tbf=True,
intfName1='r2-eth1', intfName2='r4-eth0') #,params1={'ip': '192.0.0.18/30'},params2={'ip':
'192.0.0.17/30'}

    net.addLink(r1, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=20, use_tbf=True,
intfName1='r1-eth2', intfName2='r4-eth2') # ,params1={'ip': '192.0.0.25/30'}, params2={'ip':
'192.0.0.26/30'}

    net.addLink(r2, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=20, use_tbf=True,
intfName1='r2-eth2', intfName2='r3-eth2') #,params1={'ip': '192.0.0.29/30'}, params2={'ip':
'192.0.0.30/30'}

    # Router <--> Host

    net.addLink(hostA, r1, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=20,
use_tbf=True, intfName1='hostA-eth0', intfName2='r1-eth0') #,params1={'ip': '192.0.0.1/30'},
params2={'ip': '192.0.0.2/30'}

    net.addLink(hostA, r2, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=20,
use_tbf=True, intfName1='hostA-eth1', intfName2='r2-eth0') #,params1={'ip': '192.0.0.22/30'},
params2={'ip': '192.0.0.21/30'}

    net.addLink(hostB, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=20,
use_tbf=True, intfName1='hostB-eth0', intfName2='r3-eth1') #,params1={'ip': '192.0.0.10/30'},
params2={'ip': '192.0.0.9/30'}

    net.addLink(hostB, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=20,
```

```
use_tbf=True, intfName1='hostB-eth1', intfName2='r4-eth1') #,params1={'ip': '192.0.0.13/30'}, params2={'ip': '192.0.0.14/30'}
```

```
# Buffer 40
# Configuration network
# Router <--> Router

    net.addLink(r1, r3, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='r1-eth1', intfName2='r3-eth0') #,params1={'ip': '192.0.0.5/30'},params2={'ip': '192.0.0.6/30'}

    net.addLink(r2, r4, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='r2-eth1', intfName2='r4-eth0') #,params1={'ip': '192.0.0.18/30'},params2={'ip': '192.0.0.17/30'}

    net.addLink(r1, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='r1-eth2', intfName2='r4-eth2') # ,params1={'ip': '192.0.0.25/30'}, params2={'ip': '192.0.0.26/30'}

    net.addLink(r2, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='r2-eth2', intfName2='r3-eth2') #,params1={'ip': '192.0.0.29/30'}, params2={'ip': '192.0.0.30/30'}

    # Router <--> Host

    net.addLink(hostA, r1, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='hostA-eth0', intfName2='r1-eth0') #,params1={'ip': '192.0.0.1/30'}, params2={'ip': '192.0.0.2/30'}

    net.addLink(hostA, r2, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='hostA-eth1', intfName2='r2-eth0') #,params1={'ip': '192.0.0.22/30'}, params2={'ip': '192.0.0.21/30'}

    net.addLink(hostB, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='hostB-eth0', intfName2='r3-eth1') #,params1={'ip': '192.0.0.10/30'}, params2={'ip': '192.0.0.9/30'}

    net.addLink(hostB, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=40, use_tbf=True, intfName1='hostB-eth1', intfName2='r4-eth1') #,params1={'ip': '192.0.0.13/30'}, params2={'ip': '192.0.0.14/30'}
```

```
# Buffer 60
# Configuration network
# Router <--> Router

    net.addLink(r1, r3, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=60, use_tbf=True, intfName1='r1-eth1', intfName2='r3-eth0') #,params1={'ip': '192.0.0.5/30'},params2={'ip': '192.0.0.6/30'}

    net.addLink(r2, r4, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=60, use_tbf=True, intfName1='r2-eth1', intfName2='r4-eth0') #,params1={'ip': '192.0.0.18/30'},params2={'ip':
```

```
'192.0.0.17/30'}

    net.addLink(r1, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=60, use_tbf=True,
intfName1='r1-eth2', intfName2='r4-eth2') # ,params1={'ip': '192.0.0.25/30'}, params2={'ip':
'192.0.0.26/30'}

    net.addLink(r2, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=60, use_tbf=True,
intfName1='r2-eth2', intfName2='r3-eth2') #,params1={'ip': '192.0.0.29/30'}, params2={'ip':
'192.0.0.30/30'}

    # Router <--> Host

    net.addLink(hostA, r1, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=60,
use_tbf=True, intfName1='hostA-eth0', intfName2='r1-eth0') #,params1={'ip': '192.0.0.1/30'},
params2={'ip': '192.0.0.2/30'}

    net.addLink(hostA, r2, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=60,
use_tbf=True, intfName1='hostA-eth1', intfName2='r2-eth0') #,params1={'ip': '192.0.0.22/30'},
params2={'ip': '192.0.0.21/30'}

    net.addLink(hostB, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=60,
use_tbf=True, intfName1='hostB-eth0', intfName2='r3-eth1') #,params1={'ip': '192.0.0.10/30'},
params2={'ip': '192.0.0.9/30'}

    net.addLink(hostB, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=60,
use_tbf=True, intfName1='hostB-eth1', intfName2='r4-eth1') #,params1={'ip': '192.0.0.13/30'},
params2={'ip': '192.0.0.14/30'}
```

```
# Buffer 100
# Configuration network
# Router <--> Router

    net.addLink(r1, r3, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=100,
use_tbf=True, intfName1='r1-eth1', intfName2='r3-eth0') #,params1={'ip':
'192.0.0.5/30'},params2={'ip': '192.0.0.6/30'}

    net.addLink(r2, r4, cls=TCLink, bw=0.5, delay='1ms', loss=0, max_queue_size=100,
use_tbf=True, intfName1='r2-eth1', intfName2='r4-eth0') #,params1={'ip':
'192.0.0.18/30'},params2={'ip': '192.0.0.17/30'}

    net.addLink(r1, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=100, use_tbf=True,
intfName1='r1-eth2', intfName2='r4-eth2') # ,params1={'ip': '192.0.0.25/30'}, params2={'ip':
'192.0.0.26/30'}

    net.addLink(r2, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=100, use_tbf=True,
intfName1='r2-eth2', intfName2='r3-eth2') #,params1={'ip': '192.0.0.29/30'}, params2={'ip':
'192.0.0.30/30'}

    # Router <--> Host

    net.addLink(hostA, r1, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=100,
use_tbf=True, intfName1='hostA-eth0', intfName2='r1-eth0') #,params1={'ip': '192.0.0.1/30'},
```

```
params2={'ip': '192.0.0.2/30'}

    net.addLink(hostA, r2, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=100,
use_tbf=True, intfName1='hostA-eth1', intfName2='r2-eth0') #,params1={'ip': '192.0.0.22/30'},
params2={'ip': '192.0.0.21/30'}

    net.addLink(hostB, r3, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=100,
use_tbf=True, intfName1='hostB-eth0', intfName2='r3-eth1') #,params1={'ip': '192.0.0.10/30'},
params2={'ip': '192.0.0.9/30'}

    net.addLink(hostB, r4, cls=TCLink, bw=1, delay='1ms', loss=0, max_queue_size=100,
use_tbf=True, intfName1='hostB-eth1', intfName2='r4-eth1') #,params1={'ip': '192.0.0.13/30'},
params2={'ip': '192.0.0.14/30'}
```

- Capture pengaruh ukuran buffer terhadap *delay.*
- Analisis eksperimen hasil variasi ukuran buffer.
- Mahasiswa mengerti caranya mengubah buffer dan mengenai pengaruh besar buffer.

Buffer 20

```
PING 192.0.0.10 (192.0.0.10) 56(84) bytes of data.
64 bytes from 192.0.0.10: icmp_seq=1 ttl=62 time=290 ms
64 bytes from 192.0.0.10: icmp_seq=2 ttl=62 time=331 ms
64 bytes from 192.0.0.10: icmp_seq=3 ttl=62 time=372 ms
64 bytes from 192.0.0.10: icmp_seq=4 ttl=62 time=429 ms
^C
```

Buffer 40

```
[  3] local 192.0.0.22 port 37119 connected with 192.0.0.10
PING 192.0.0.10 (192.0.0.10) 56(84) bytes of data.
64 bytes from 192.0.0.10: icmp_seq=1 ttl=62 time=427 ms
64 bytes from 192.0.0.10: icmp_seq=2 ttl=62 time=456 ms
64 bytes from 192.0.0.10: icmp_seq=3 ttl=62 time=449 ms
64 bytes from 192.0.0.10: icmp_seq=4 ttl=62 time=467 ms
^C
```

Buffer 60

```
[  3] local 192.0.0.22 port 41131 connected with 192.0.0.10
PING 192.0.0.10 (192.0.0.10) 56(84) bytes of data.
64 bytes from 192.0.0.10: icmp_seq=1 ttl=62 time=1300 ms
64 bytes from 192.0.0.10: icmp_seq=2 ttl=62 time=965 ms
64 bytes from 192.0.0.10: icmp_seq=3 ttl=62 time=858 ms
64 bytes from 192.0.0.10: icmp_seq=4 ttl=62 time=913 ms
^C
```

Buffer 100

```
[  3] local 192.0.0.22 port 56485 connected with 192.0.0.10 port
PING 192.0.0.10 (192.0.0.10) 56(84) bytes of data.
64 bytes from 192.0.0.10: icmp_seq=1 ttl=62 time=1784 ms
64 bytes from 192.0.0.10: icmp_seq=2 ttl=62 time=2224 ms
64 bytes from 192.0.0.10: icmp_seq=4 ttl=62 time=2328 ms
64 bytes from 192.0.0.10: icmp_seq=5 ttl=62 time=2296 ms
64 bytes from 192.0.0.10: icmp_seq=7 ttl=62 time=1760 ms
^C
```

Kesimpulan : Semakin besar nilai buffer yang diterapkan, semakin lama waktu yang dibutuhkan pada saat pengiriman packet pada sebuah jaringan.