

Laporan Tugas Besar
Mata Kuliah Jaringan Komputer



Oleh:

Herman Gemilang (1301204014)

Program Studi S1 Informatika

Fakultas Informatika Telkom University

2022

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penggunaan komputer dewasa ini sudah sedemikian luasnya, meliputi segala aspek bidang ilmu pengetahuan. Dalam pendidikan misalnya, diupayakan bahwa komputer dapat digunakan sebagai alat bantu penunjang program pembelajaran dan sebagai alat yang dapat digunakan untuk mengajar. Pada Tugas laporan tugas besar kali ini penulis mencoba membuat topologi jaringan dari beberapa host dan router berdasarkan topologi dan subnetting yang telah penulis tentukan. Protokol yang digunakan adalah TCP dan menggunakan implementasi mininet untuk simulasi topologi jaringan.

1.2 Tujuan

Tujuan dari pembuatan laporan ini adalah untuk mengimplemetasikan topologi jaringan dan mengimplementasikan ilmu yang didapat selama praktikum jaringan komputer, serta laporan ini bertujuan untuk memenuhi tugas besar mata kuliah jaringan komputer.

1.3 Batasan Masalah

Laporan ini terbagi menjadi 4 bagian (akan disebut sebagai CLO) yaitu.

- CLO1 = Build topology sesuai dengan soal.
- CLO2 = Mengimplementasikan mekanisme Routing pada topologi yang ada.
- CLO3 = Membuktikan bahwa TCP telah di-implementasikan dengan benar pada topologi.
- CLO4 = Menginspeksi penggunaan queue pada router jaringan

BAB 2

LANDASAN TEORI

2.1 TOPOLOGI JARINGAN

Topologi jaringan mengacu pada bagaimana berbagai node, perangkat, dan koneksi di jaringan Anda secara fisik atau logis berkaitan satu sama lain. Diibaratkan jaringan Anda sebagai kota, dan topologi sebagai peta jalan. Sama seperti ada banyak cara untuk mengatur dan memelihara kota seperti memastikan jalan dan jalan raya dapat memfasilitasi perjalanan antar bagian kota untuk mendapatkan lalu lintas terbanyak ada beberapa cara untuk mengatur jaringan. Masing-masing memiliki kelebihan dan kekurangan dan tergantung pada kebutuhan perusahaan Anda, pengaturan tertentu dapat memberi Anda tingkat konektivitas dan keamanan yang lebih besar. Ada dua pendekatan untuk topologi jaringan: fisik dan logis. Topologi jaringan fisik, seperti namanya, mengacu pada koneksi fisik dan interkoneksi antara node dan jaringan — kabel, kabel, dan sebagainya. Topologi jaringan logis sedikit lebih abstrak dan strategis, mengacu pada pemahaman konseptual tentang bagaimana dan mengapa jaringan diatur seperti itu, dan bagaimana data bergerak melaluinya.

2.2 TCP

Transmission Control Protocol atau biasa disingkat dengan TCP adalah protokol lapisan transport yang paling umum, TCP merupakan protokol tingkat transportasi Internet yang menyediakan komunikasi ujung-ke-ujung yang andal antara dua proses. Proses request, sering dikenal sebagai client, meminta layanan dari proses server. Proses client dan server dapat diakses pada mesin masing-masing dengan nomor port TCP yang ditetapkan untuk mereka. Banyak layanan lapisan aplikasi standar memiliki nomor port TCP terkenal yang ditetapkan oleh otoritas pusat. Misalnya, server Simple Mail Transfer Protocol beroperasi pada port TCP 25 yang terkenal. TCP membawa byte data dari proses tingkat yang lebih tinggi dengan mengemasnya ke dalam segmen TCP. Data segmen TCP kemudian dikemas oleh IP ke dalam bagian data paket IP.

2.3 Mininet

Mininet adalah emulator jaringan yang membuat jaringan host virtual, sakelar, pengontrol, dan tautan. Host mininet menjalankan perangkat lunak jaringan Linux standar, dan sakelarnya mendukung open flow untuk perutean khusus yang sangat fleksibel dan Jaringan yang Ditentukan Perangkat Lunak. Mininet mendukung penelitian, pengembangan, pembelajaran, pembuatan prototipe, pengujian, debugging, dan tugas lainnya yang dapat memperoleh manfaat dari memiliki jaringan eksperimental yang lengkap pada sebuah laptop atau komputer lain.

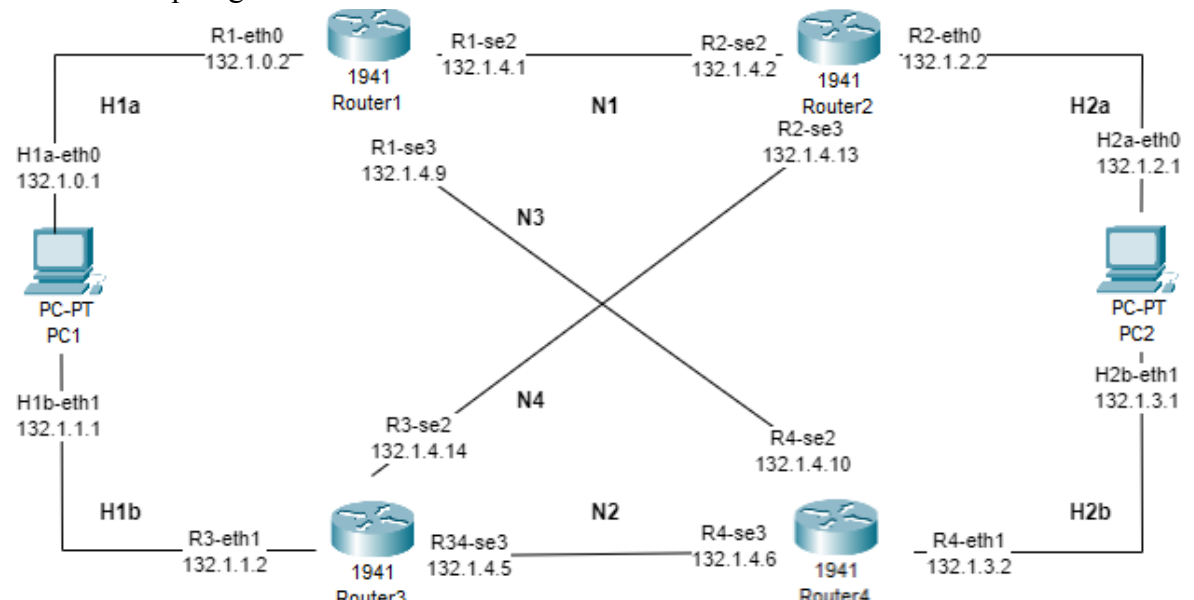
BAB 3

IMPLEMENTASI DAN ANALISIS

I. CLO 1

Membangun topologi jaringan

a. Desain Topologi



b. Subnetting Table

Name	Host Size	Net ID	Host Range	Broadcast	Prefi x	Subnetmask
H1a	254	132.1.0.0	132.1.0.1-132.1.0.254	132.1.0.255	24	255.255.255.0
H1b	254	132.1.1.0	132.1.1.1-132.1.1.254	132.1.1.255	24	255.255.255.0
H2a	254	132.1.2.0	132.1.2.1-132.1.2.254	132.1.2.255	24	255.255.255.0
H2b	254	132.1.3.0	132.1.3.1-132.1.3.254	132.1.3.255	24	255.255.255.0
N1	2	132.1.4.0	132.1.4.1-132.1.4.2	132.1.4.3	30	255.255.255.252
N2	2	132.1.4.4	132.1.4.5-132.1.4.6	132.1.4.7	30	255.255.255.252
N3	2	132.1.4.8	132.1.4.9-132.1.4.10	132.1.4.11	30	255.255.255.252
N4	2	132.1.4.12	132.1.4.13-132.1.4.14	132.1.4.15	30	255.255.255.252

c. Menguji konektifitas yang berada pada satu jaringan

1. H1a

```
mininet> hA ping 132.1.0.1
PING 132.1.0.1 (132.1.0.1) 56(84) bytes of data.
64 bytes from 132.1.0.1: icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from 132.1.0.1: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 132.1.0.1: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 132.1.0.1: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 132.1.0.1: icmp_seq=5 ttl=64 time=0.051 ms
64 bytes from 132.1.0.1: icmp_seq=6 ttl=64 time=0.041 ms
64 bytes from 132.1.0.1: icmp_seq=7 ttl=64 time=0.110 ms
64 bytes from 132.1.0.1: icmp_seq=8 ttl=64 time=0.046 ms
64 bytes from 132.1.0.1: icmp_seq=9 ttl=64 time=0.043 ms
64 bytes from 132.1.0.1: icmp_seq=10 ttl=64 time=0.091 ms
64 bytes from 132.1.0.1: icmp_seq=11 ttl=64 time=0.044 ms
```

2. H1b

```
mininet> hA ping 132.1.1.1
PING 132.1.1.1 (132.1.1.1) 56(84) bytes of data.
64 bytes from 132.1.1.1: icmp_seq=1 ttl=64 time=0.066 ms
64 bytes from 132.1.1.1: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 132.1.1.1: icmp_seq=3 ttl=64 time=0.082 ms
64 bytes from 132.1.1.1: icmp_seq=4 ttl=64 time=0.043 ms
64 bytes from 132.1.1.1: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 132.1.1.1: icmp_seq=6 ttl=64 time=0.044 ms
```

3. H2A

```
PING 132.1.2.1 (132.1.2.1) 56(84) bytes of data.
64 bytes from 132.1.2.1: icmp_seq=1 ttl=62 time=0.111 ms
64 bytes from 132.1.2.1: icmp_seq=2 ttl=62 time=0.078 ms
64 bytes from 132.1.2.1: icmp_seq=3 ttl=62 time=0.084 ms
64 bytes from 132.1.2.1: icmp_seq=4 ttl=62 time=0.080 ms
64 bytes from 132.1.2.1: icmp_seq=5 ttl=62 time=0.131 ms
64 bytes from 132.1.2.1: icmp_seq=6 ttl=62 time=0.079 ms
^C
--- 132.1.2.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5148ms
```

4. H2b

```
mininet> hA ping 132.1.3.1
PING 132.1.3.1 (132.1.3.1) 56(84) bytes of data.
64 bytes from 132.1.3.1: icmp_seq=1 ttl=62 time=0.117 ms
64 bytes from 132.1.3.1: icmp_seq=2 ttl=62 time=0.092 ms
64 bytes from 132.1.3.1: icmp_seq=3 ttl=62 time=0.096 ms
64 bytes from 132.1.3.1: icmp_seq=4 ttl=62 time=0.180 ms
64 bytes from 132.1.3.1: icmp_seq=5 ttl=62 time=0.166 ms
^C
--- 132.1.3.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4110ms
```

5. N1

```
mininet> hA ping 132.1.4.1
PING 132.1.4.1 (132.1.4.1) 56(84) bytes of data.
64 bytes from 132.1.4.1: icmp_seq=1 ttl=64 time=0.132 ms
64 bytes from 132.1.4.1: icmp_seq=2 ttl=64 time=0.100 ms
64 bytes from 132.1.4.1: icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from 132.1.4.1: icmp_seq=4 ttl=64 time=0.163 ms
64 bytes from 132.1.4.1: icmp_seq=5 ttl=64 time=0.065 ms
64 bytes from 132.1.4.1: icmp_seq=6 ttl=64 time=0.056 ms
^C
--- 132.1.4.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5141ms
```

6. N2

```
mininet> hA ping 132.1.4.5
PING 132.1.4.5 (132.1.4.5) 56(84) bytes of data.
64 bytes from 132.1.4.5: icmp_seq=1 ttl=62 time=0.111 ms
64 bytes from 132.1.4.5: icmp_seq=2 ttl=62 time=0.083 ms
64 bytes from 132.1.4.5: icmp_seq=3 ttl=62 time=0.091 ms
64 bytes from 132.1.4.5: icmp_seq=4 ttl=62 time=0.078 ms
64 bytes from 132.1.4.5: icmp_seq=5 ttl=62 time=0.098 ms
64 bytes from 132.1.4.5: icmp_seq=6 ttl=62 time=0.093 ms
^C
--- 132.1.4.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5219ms
```

7. N3

```
mininet> hA ping 132.1.4.9
PING 132.1.4.9 (132.1.4.9) 56(84) bytes of data.
64 bytes from 132.1.4.9: icmp_seq=1 ttl=64 time=0.119 ms
64 bytes from 132.1.4.9: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 132.1.4.9: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from 132.1.4.9: icmp_seq=4 ttl=64 time=0.057 ms
64 bytes from 132.1.4.9: icmp_seq=5 ttl=64 time=0.055 ms
64 bytes from 132.1.4.9: icmp_seq=6 ttl=64 time=0.068 ms
^C
--- 132.1.4.9 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5113ms
```

8. N4

```
mininet> hA ping 132.1.4.13
PING 132.1.4.13 (132.1.4.13) 56(84) bytes of data.
64 bytes from 132.1.4.13: icmp_seq=1 ttl=63 time=0.096 ms
64 bytes from 132.1.4.13: icmp_seq=2 ttl=63 time=0.088 ms
64 bytes from 132.1.4.13: icmp_seq=3 ttl=63 time=0.070 ms
64 bytes from 132.1.4.13: icmp_seq=4 ttl=63 time=0.074 ms
64 bytes from 132.1.4.13: icmp_seq=5 ttl=63 time=0.085 ms
64 bytes from 132.1.4.13: icmp_seq=6 ttl=63 time=0.084 ms
64 bytes from 132.1.4.13: icmp_seq=7 ttl=63 time=0.070 ms
^C
--- 132.1.4.13 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6268ms
```

II. CLO2

Mengimplementasikan Routing pada topologi yang sudah ada

- a. Membuat routing table pada semua host

```

mininet> hA netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags       MSS Window  irtt Iface
0.0.0.0          132.1.0.2      0.0.0.0         UG          0 0        0 hA-eth0
132.1.0.0        0.0.0.0        255.255.255.0   U           0 0        0 hA-eth0
132.1.1.0        0.0.0.0        255.255.255.0   U           0 0        0 hA-eth1
mininet> hB netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags       MSS Window  irtt Iface
0.0.0.0          132.1.2.2      0.0.0.0         UG          0 0        0 hB-eth0
132.1.2.0        0.0.0.0        255.255.255.0   U           0 0        0 hB-eth0
132.1.3.0        0.0.0.0        255.255.255.0   U           0 0        0 hB-eth1
mininet> r1 netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags       MSS Window  irtt Iface
132.1.0.0        0.0.0.0        255.255.255.0   U           0 0        0 r1-eth0
132.1.1.0        132.1.4.10     255.255.255.0   UG          0 0        0 r1-se3
132.1.2.0        132.1.4.2      255.255.255.0   UG          0 0        0 r1-se2
132.1.3.0        132.1.4.10     255.255.255.0   UG          0 0        0 r1-se3
132.1.4.0        0.0.0.0        255.255.255.252 U           0 0        0 r1-se2
132.1.4.4        132.1.4.10     255.255.255.252 UG          0 0        0 r1-se3
132.1.4.8        0.0.0.0        255.255.255.252 U           0 0        0 r1-se3
132.1.4.12       132.1.4.2      255.255.255.252 UG          0 0        0 r1-se2
mininet> r2 netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags       MSS Window  irtt Iface
132.1.0.0        132.1.4.1      255.255.255.0   UG          0 0        0 r2-se2
132.1.1.0        132.1.4.14     255.255.255.0   UG          0 0        0 r2-se3
132.1.2.0        0.0.0.0        255.255.255.0   U           0 0        0 r2-eth0
132.1.3.0        132.1.4.14     255.255.255.0   UG          0 0        0 r2-se3
132.1.4.0        0.0.0.0        255.255.255.252 U           0 0        0 r2-se2
132.1.4.4        132.1.4.14     255.255.255.252 UG          0 0        0 r2-se3
132.1.4.8        132.1.4.1      255.255.255.252 UG          0 0        0 r2-se2
132.1.4.12       0.0.0.0        255.255.255.252 U           0 0        0 r2-se3
mininet> r3 netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags       MSS Window  irtt Iface
132.1.0.0        132.1.4.6      255.255.255.0   UG          0 0        0 r3-se3
132.1.1.0        0.0.0.0        255.255.255.0   U           0 0        0 r3-eth1
132.1.2.0        132.1.4.13     255.255.255.0   UG          0 0        0 r3-se2
132.1.3.0        132.1.4.6      255.255.255.0   UG          0 0        0 r3-se3
132.1.4.0        132.1.4.13     255.255.255.252 UG          0 0        0 r3-se2
132.1.4.4        0.0.0.0        255.255.255.252 U           0 0        0 r3-se3
132.1.4.8        132.1.4.6      255.255.255.252 UG          0 0        0 r3-se3
132.1.4.12       0.0.0.0        255.255.255.252 U           0 0        0 r3-se2
mininet> r4 netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags       MSS Window  irtt Iface
132.1.0.0        132.1.4.9      255.255.255.0   UG          0 0        0 r4-se2
132.1.1.0        132.1.4.5      255.255.255.0   UG          0 0        0 r4-se3
132.1.2.0        132.1.4.5      255.255.255.0   UG          0 0        0 r4-se3
132.1.3.0        0.0.0.0        255.255.255.0   U           0 0        0 r4-eth1
132.1.4.0        132.1.4.9      255.255.255.252 UG          0 0        0 r4-se2
132.1.4.4        0.0.0.0        255.255.255.252 U           0 0        0 r4-se3
132.1.4.8        0.0.0.0        255.255.255.252 U           0 0        0 r4-se2
132.1.4.12       132.1.4.5      255.255.255.252 UG          0 0        0 r4-se3
mininet>

```

b. Uji konektivitas Dengan melakukan ping


```
mininet> pingall
*** Ping: testing ping reachability
hA -> hB r1 r2 r3 r4
hB -> hA r1 r2 r3 r4
r1 -> hA hB r2 r3 r4
r2 -> hA hB r1 r3 r4
r3 -> hA hB r1 r2 r4
r4 -> hA hB r1 r2 r3
*** Results: 0% dropped (30/30 received)
mininet>
```

R

- c. Uji konektivitas dengan menggunakan traceroute

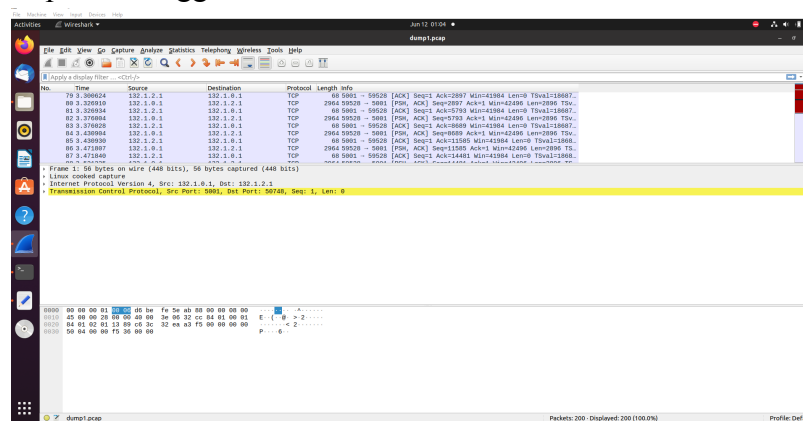
```
mininet> hA traceroute hB
traceroute to 132.1.2.1 (132.1.2.1), 30 hops max, 60 byte packets
 1 132.1.0.2 (132.1.0.2) 0.338 ms 0.288 ms 0.274 ms
 2 132.1.4.2 (132.1.4.2) 0.261 ms 0.236 ms 0.221 ms
 3 132.1.2.1 (132.1.2.1) 0.206 ms 0.182 ms 0.163 ms
mininet> hB traceroute hA
traceroute to 132.1.0.1 (132.1.0.1), 30 hops max, 60 byte packets
 1 132.1.2.2 (132.1.2.2) 0.385 ms 0.330 ms 0.314 ms
 2 132.1.4.1 (132.1.4.1) 0.299 ms 0.271 ms 0.253 ms
 3 132.1.0.1 (132.1.0.1) 0.234 ms 0.202 ms 0.180 ms
mininet>
```

III. CLO3

- a. Generate traffic h1 ke h2 menggunakan iperf

```
mininet> iperf hA hB
*** Iperf: testing TCP bandwidth between hA and hB
*** Results: ['472 Kbits/sec', '1.11 Mbits/sec']
```

- b. Inspeksi menggunakan wireshark



IV. CLO4

Pengaruh queue pada router jaringan

- a. Buffer = 20

```
mininet> iperf hA hB
*** Iperf: testing TCP bandwidth between hA and hB
*** Results: ['478 Kbits/sec', '1.12 Mbits/sec']
```

- b. Buffer = 40

```
mininet> iperf hA hB
*** Iperf: testing TCP bandwidth between hA and hB
*** Results: ['478 Kbits/sec', '1.13 Mbits/sec']
```

- c. Buffer = 60

```
mininet> iperf hA hB
*** Iperf: testing TCP bandwidth between hA and hB
*** Results: ['478 Kbits/sec', '1.15 Mbits/sec']
```

- d. Buffer = 100

```
mininet> iperf hA hB
*** Iperf: testing TCP bandwidth between hA and hB
*** Results: ['478 Kbits/sec', '1.16 Mbits/sec']
```

- e. Analisis

Buffer adalah sebuah daerah memori yang menyimpan data ketika data tersebut du transfer antara dua paket, jadi sesudah dilakukan percobaan untuk buffer pada jaringan sebelumnya, dan didapat bahwa semakin besar buffer maka semakin cepat juga kecepatan jaringan tersebut.

BAB 4

KESIMPULAN

Setelah dilakukan serangkaian pengujian diatas didapatkan kesimpulan bahwa IP dengan subnetting dan topologi diatas menjalani uji konektivitas dengan ping dengan baik dibuktikan dengan 0% packet dropped, IP juga menjalani uji traceroute dengan baik seperti yang dapat dilihat pada bab analisis diatas. Sayangnya penulis tidak bisa membuat packet dengan trafik TCP muncul di wireshark/TCPDUMP.