

TUGAS PRAKTIKUM UDP 2

PEMROGRAMAN JARINGAN



Nama : Rida Adila
NRP : 05111840000002
Kelas : PROGJAR D

Dosen : Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.

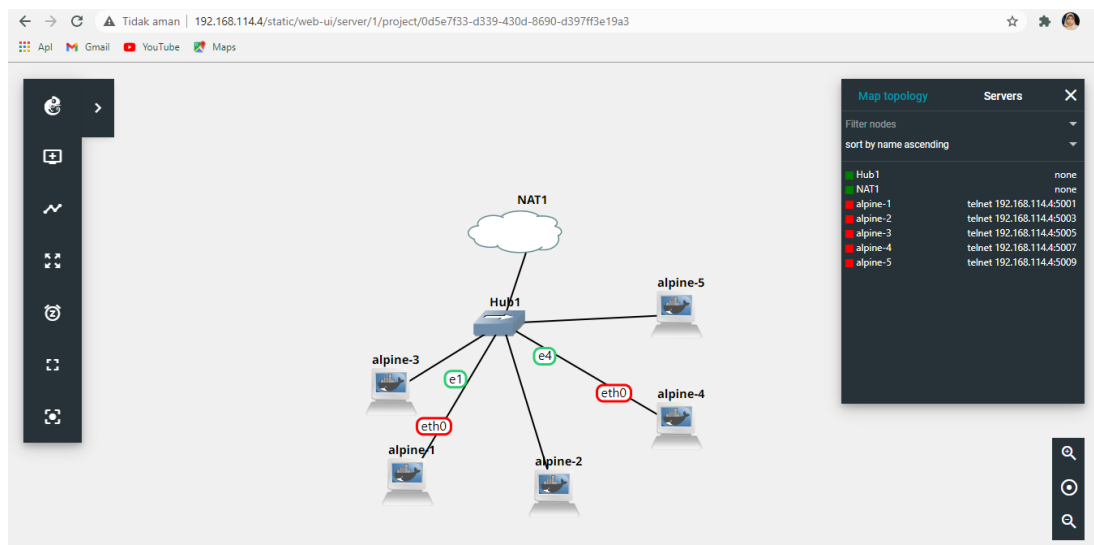
Departemen Infomatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember (ITS) Surabaya
2021

PRAKTIKUM UDP 2

- 1) Loadlah file tersebut di simulator.
- 2) Dalam praktikum ini lakukan broadcast dari sebuah client udp di alpine-1 agar dapat membroadcast pengiriman ke alpine-2 dan alpine-3 sekaligus, tambahkan host alpine-4 dan alpine-5 yang dapat dikirim broadcast juga
- 3) Gunakan file progjar2/udpcclient_broadcast.py dan progjar2/udpserver_broadcast.py.
- 4) Sesuaikan parameter dan variabel program agar sesuai dengan lingkungan jaringan ,
- 5) Jalankan dengan urutan, server dan kemudian client
- 6) Jalankan dengan urutan client, baru kemudian server
- 7) Apakah perbedaan yang terjadi ?

PRAKTIKUM

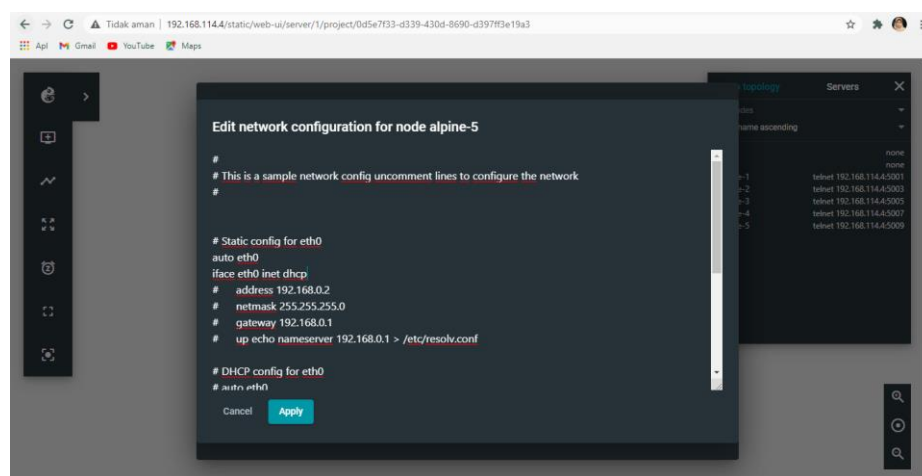
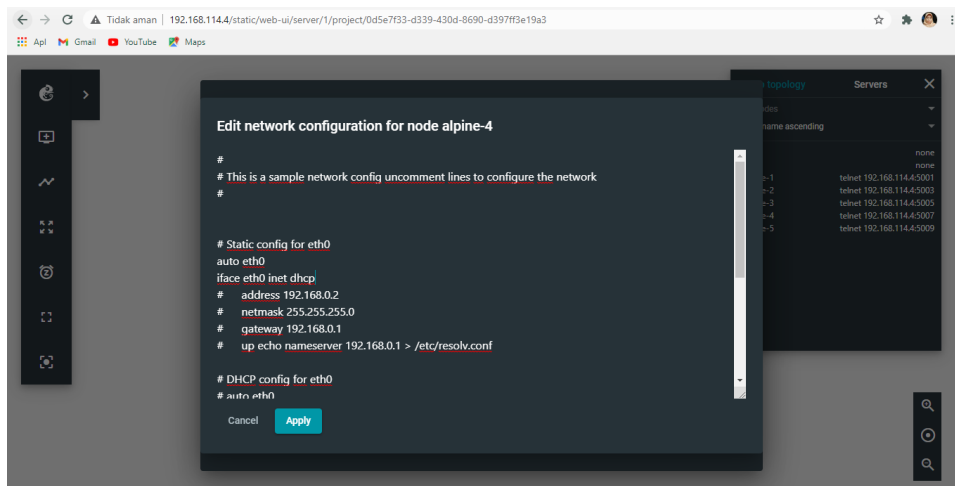
- 1) Menambahkan alpine-4 dan alpine-5



- 2) Melakukan konfigurasi IP address DHCP pada alpine 4 dan alpine 5

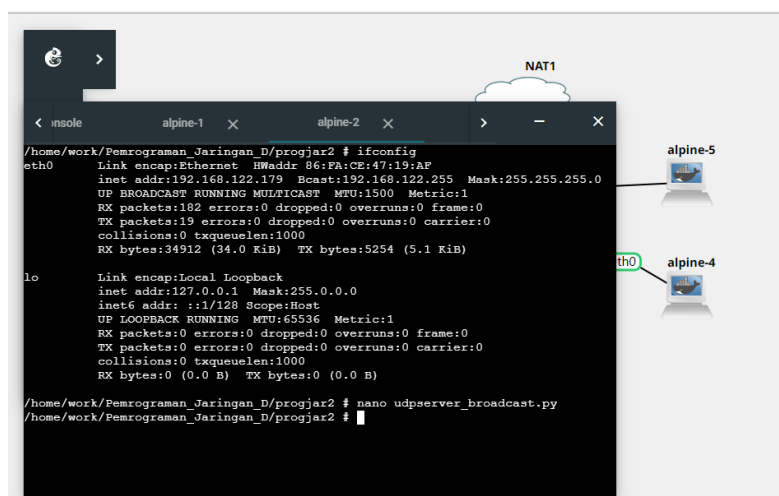
Pada masing masing alpine 4 dan alpine 5, klik kanan lalu pilih configure. Kemudian pilih Edit Network Configuration. Setelah itu :

- Uncomment pada bagian → auto eth0
 - Uncomment pada bagian → iface eth0 inet static. Ganti kata “static” dengan “dhcp”.
- Lalu apply configurationnya. Sehingga Nampak hasilnya seperti di bawah ini

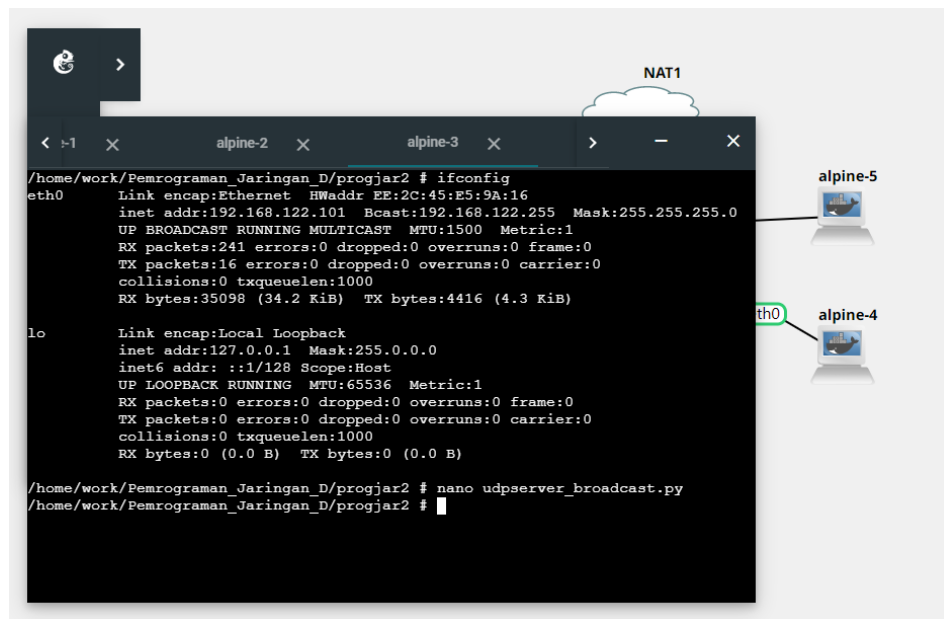


3) Lakukan ifconfig pada alpine2 – alpine5

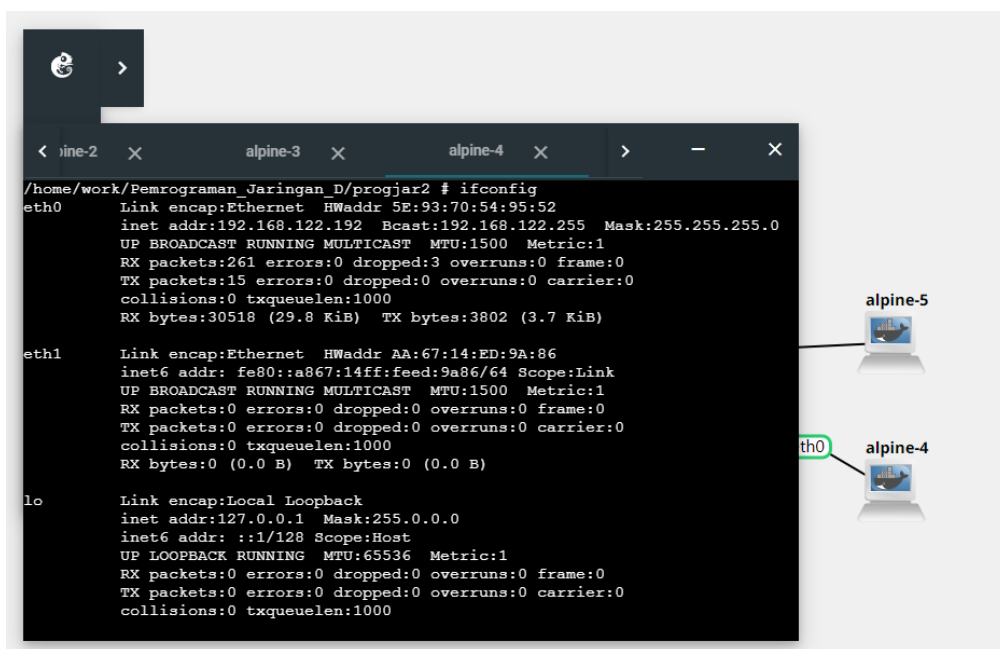
a) Ifconfig alpine2 :



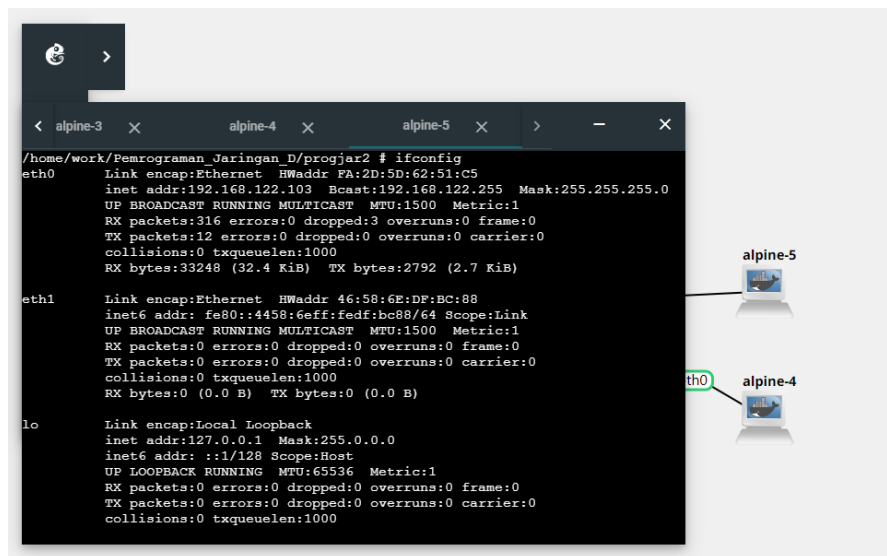
b) Ifconfig alpine3 :



c) Ifconfig alpine4:



d) Ifconfig alpine5 :

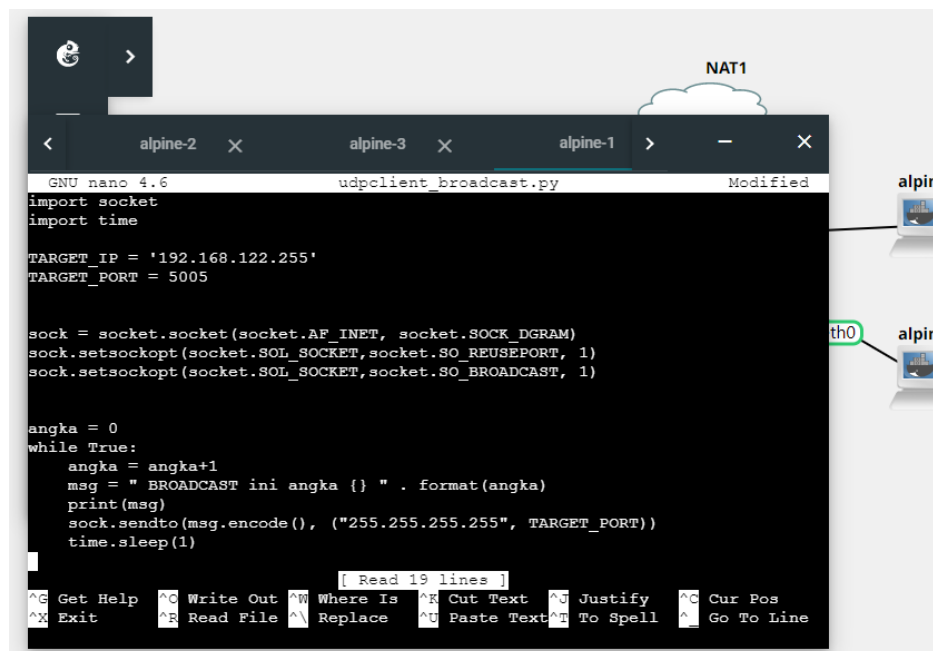


```
/home/work/Pemrograman_Jaringan_D/progjar2 # ifconfig
eth0      Link encap:Ethernet  HWaddr FA:2D:5D:62:51:C5
          inet addr:192.168.122.103  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:316 errors:0 dropped:3 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33248 (32.4 KiB)  TX bytes:2792 (2.7 KiB)

eth1      Link encap:Ethernet  HWaddr 46:58:6E:DF:BC:88
          inet6 addr: fe80::4458:6eff:fedf:bc88/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

4) Pada alpine1, ubah file udpclient_broadcast.py seperti dibawah ini. Dimana TARGET_IP valuenya diganti denga ip broadcast dari alpine2 – alpine5 . Semua nilai dari ip broadcast tsb sama yakni 192.168.122.255. Serta uncomment pada baris yang berisi “time.sleep(1)” , hal ini agar data yang dikirimkan tidak terlalu cepat sehingga ada jeda waktu



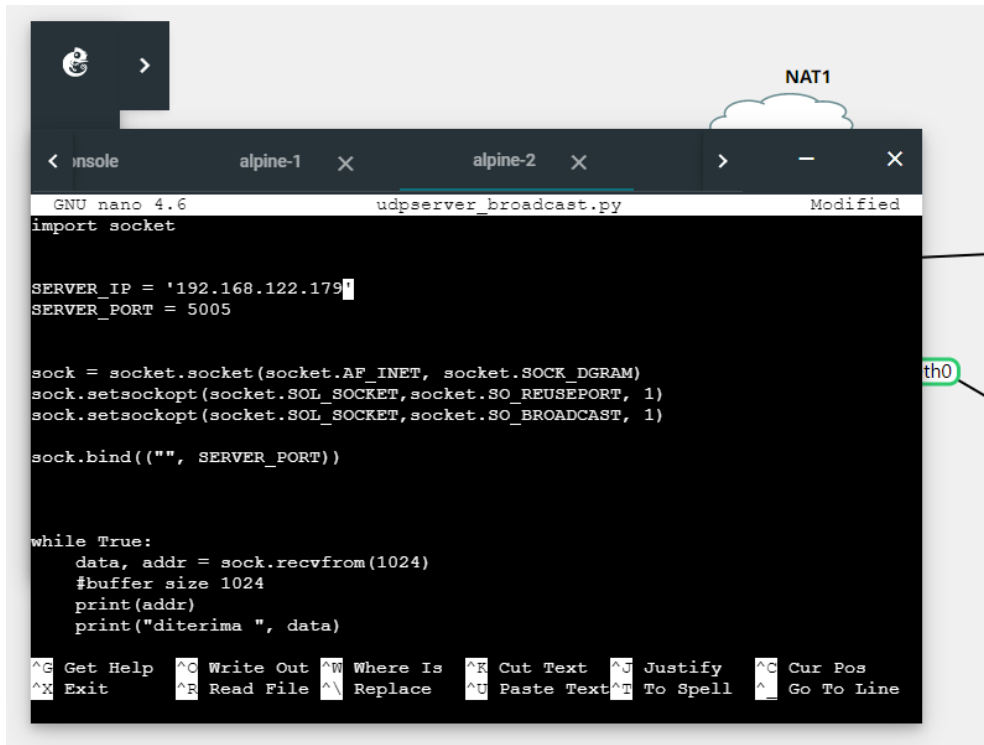
```
GNU nano 4.6  udpclient_broadcast.py  Modified
import socket
import time

TARGET_IP = '192.168.122.255'
TARGET_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEPORT, 1)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)

angka = 0
while True:
    angka = angka+1
    msg = " BROADCAST ini angka {} " . format(angka)
    print(msg)
    sock.sendto(msg.encode(), ("255.255.255.255", TARGET_PORT))
    time.sleep(1)
```

- 5) Kemudian pada alpine2, ubah SERVER_IP sesuai dengan hasil ifconfig pada alpine2, yakni 192.168.122.179



The screenshot shows a terminal window titled 'alpine-2' with a nano editor open to a file named 'udpserver_broadcast.py'. The code in the editor is as follows:

```
GNU nano 4.6      udpserver_broadcast.py      Modified
import socket

SERVER_IP = '192.168.122.179'
SERVER_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEPORT, 1)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)

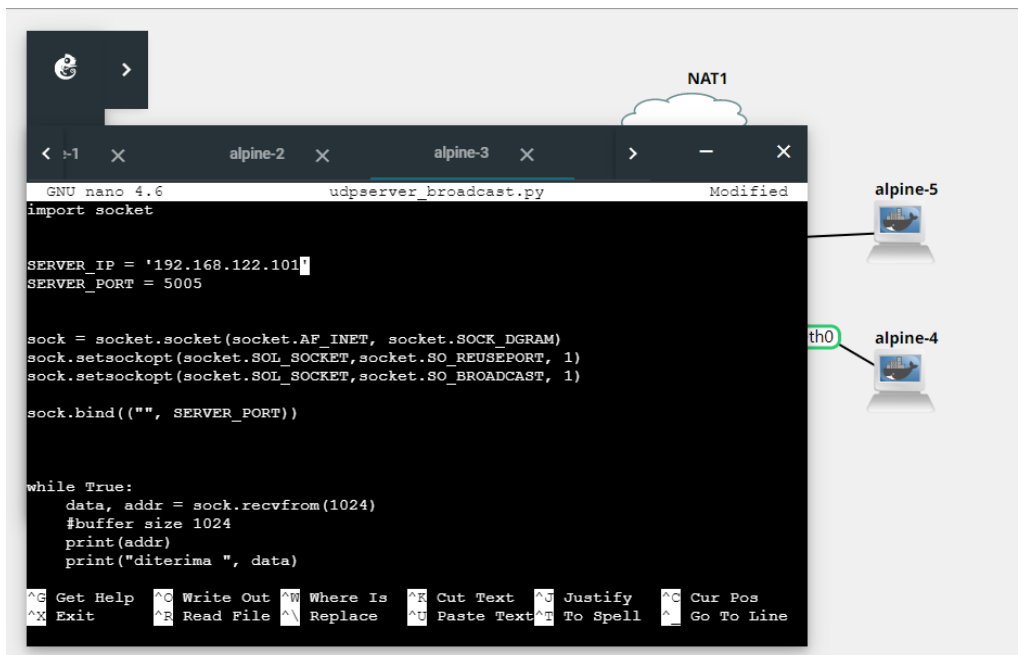
sock.bind(("", SERVER_PORT))

while True:
    data, addr = sock.recvfrom(1024)
    #buffer size 1024
    print(addr)
    print("diterima ", data)
```

At the bottom of the terminal, there is a help menu with the following shortcuts:

^G	Get Help	^O	Write Out	^W	Where Is	^R	Cut Text	^J	Justify	^C	Cur Pos
^X	Exit	^R	Read File	^_\	Replace	^U	Paste Text	^T	To Spell	^_	Go To Line

- 6) Kemudian pada alpine3, ubah SERVER_IP sesuai dengan hasil ifconfig pada alpine2, yakni 192.168.122.101



The screenshot shows a terminal window titled 'alpine-3' with a nano editor open to a file named 'udpserver_broadcast.py'. The code in the editor is as follows:

```
GNU nano 4.6      udpserver_broadcast.py      Modified
import socket

SERVER_IP = '192.168.122.101'
SERVER_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEPORT, 1)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)

sock.bind(("", SERVER_PORT))

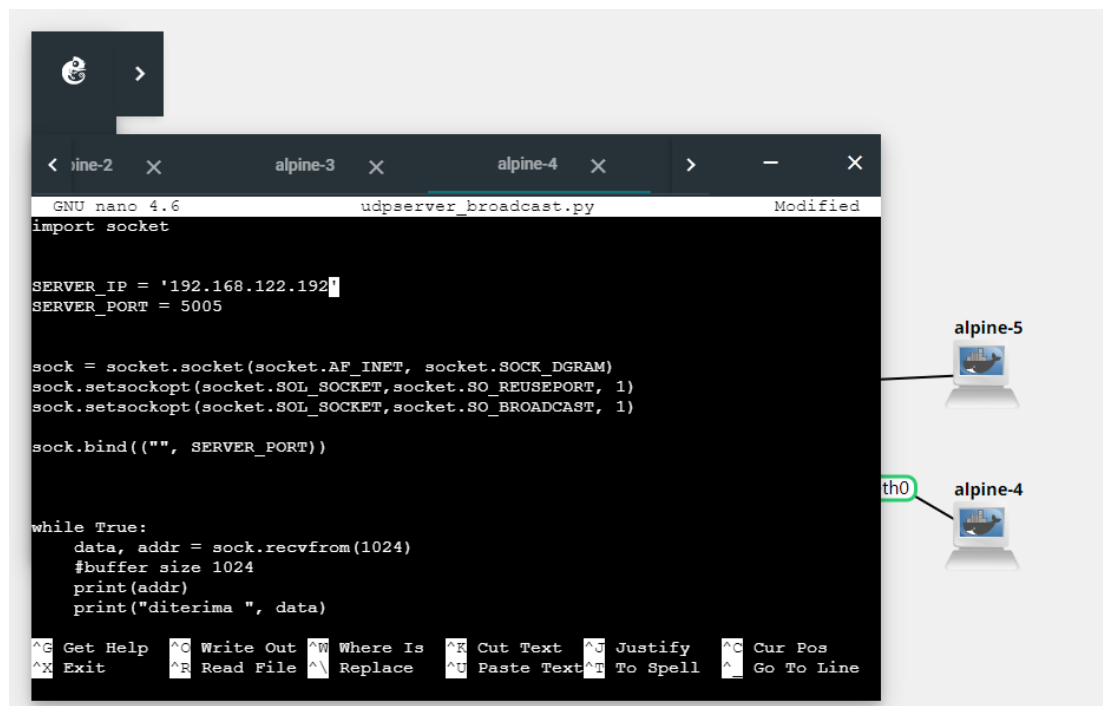
while True:
    data, addr = sock.recvfrom(1024)
    #buffer size 1024
    print(addr)
    print("diterima ", data)
```

At the bottom of the terminal, there is a help menu with the following shortcuts:

^G	Get Help	^O	Write Out	^W	Where Is	^R	Cut Text	^J	Justify	^C	Cur Pos
^X	Exit	^R	Read File	^_\	Replace	^U	Paste Text	^T	To Spell	^_	Go To Line

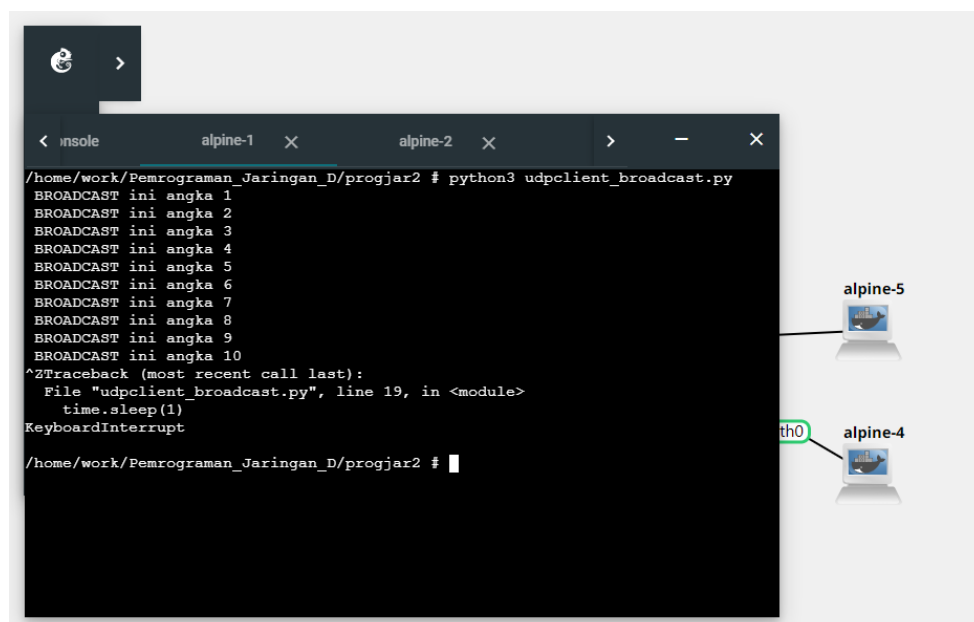
In the background, a network diagram is visible showing three laptops labeled 'alpine-4', 'alpine-5', and 'alpine-3'. A cloud labeled 'NAT1' is connected to the laptops. A green circle highlights the 'eth0' interface on alpine-4.

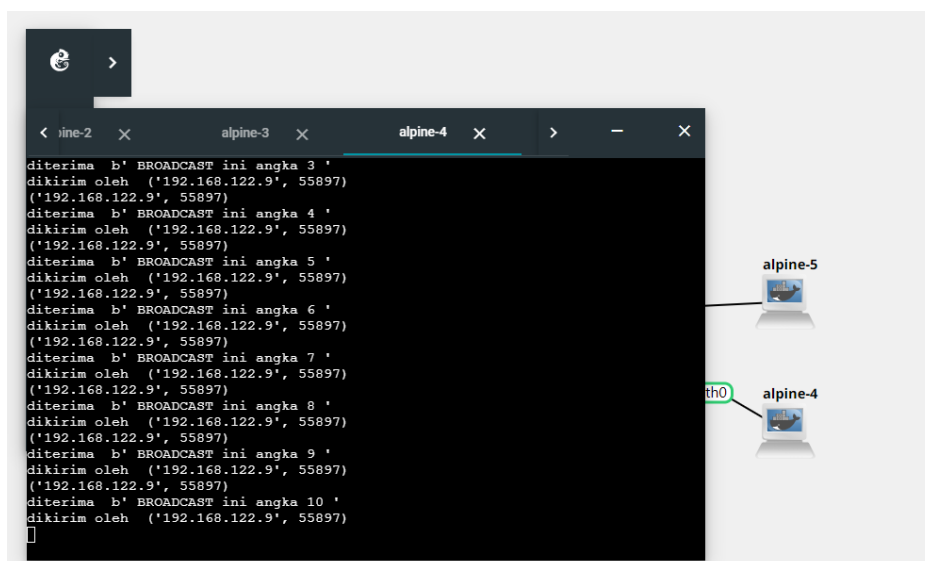
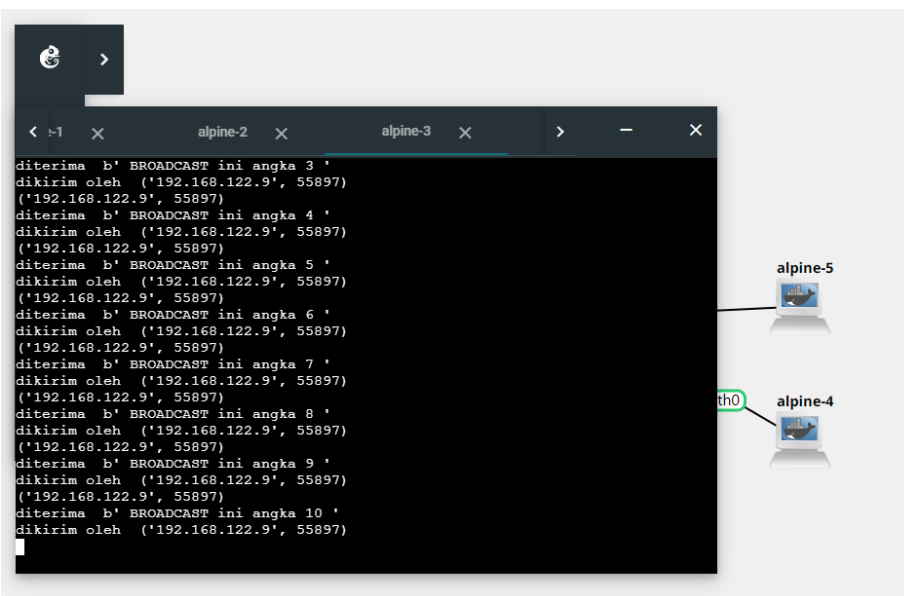
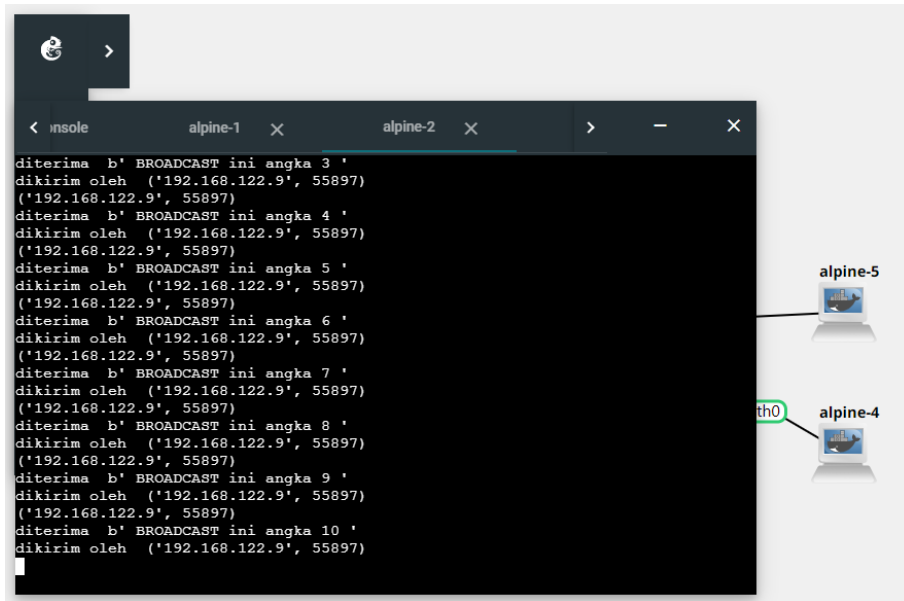
- 7) Kemudian pada alpine4, ubah SERVER_IP sesuai dengan hasil ifconfig pada alpine2, yakni 192.168.122.192

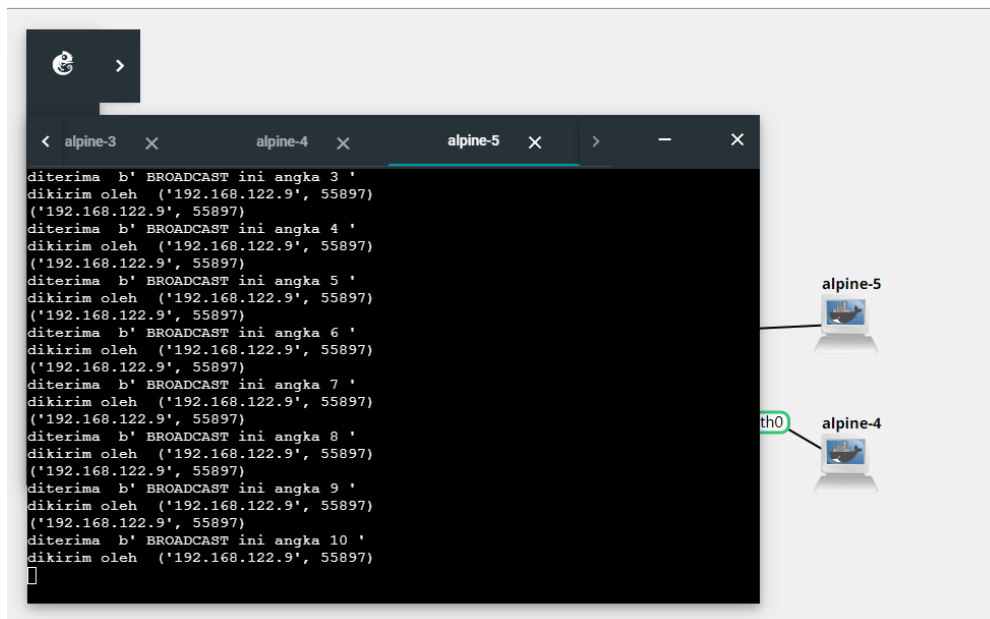


- 8) Kemudian pada alpine5, ubah SERVER_IP sesuai dengan hasil ifconfig pada alpine2, yakni 192.168.122.103

- Hasil dijalankan dengan urutan server – client :

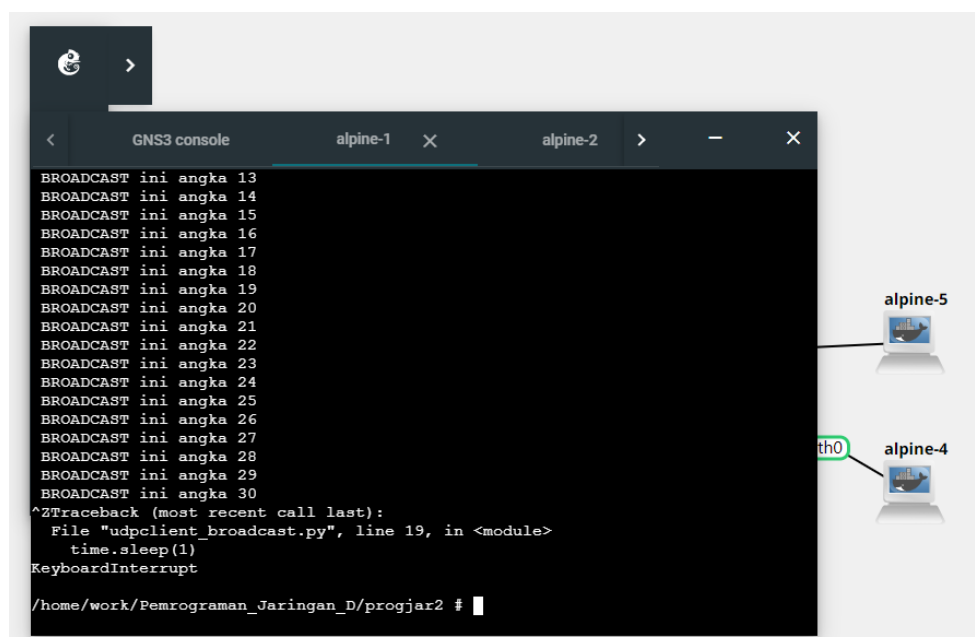


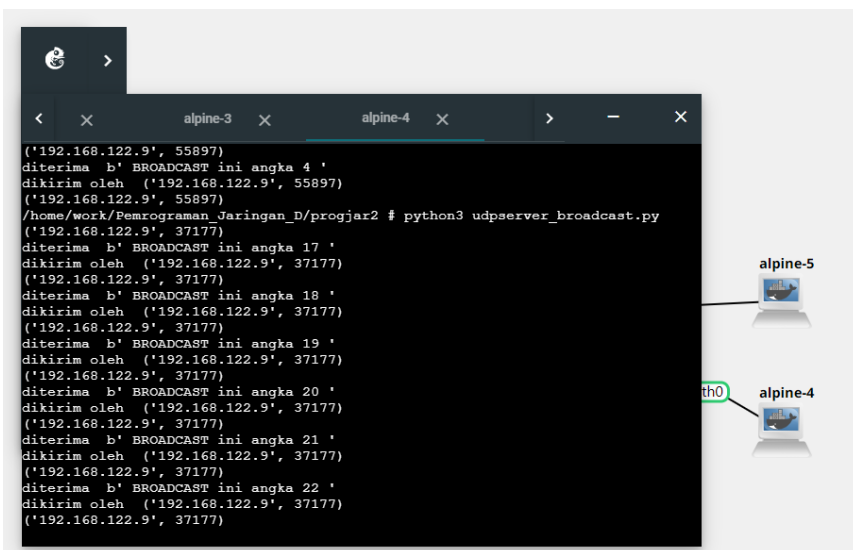
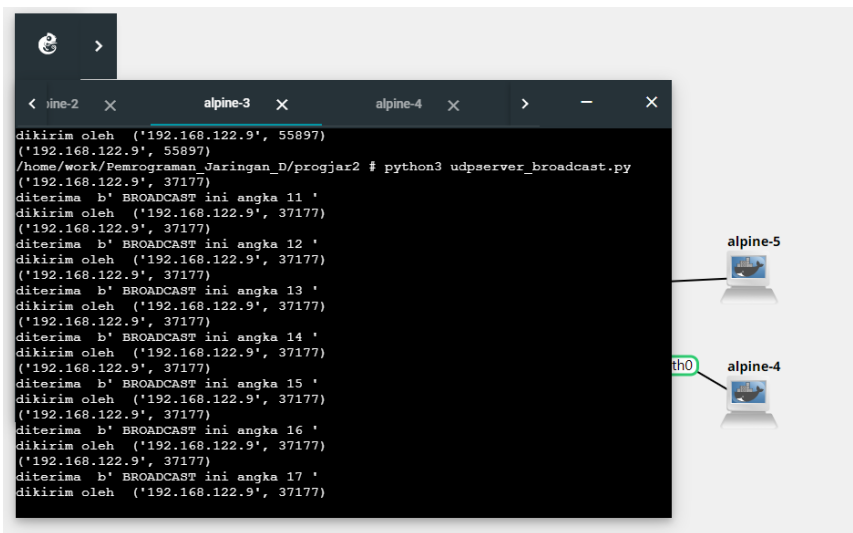
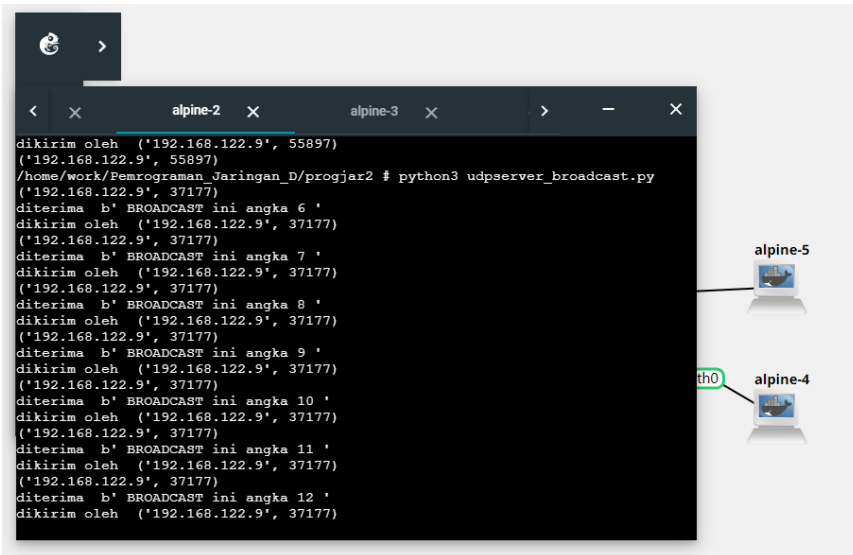


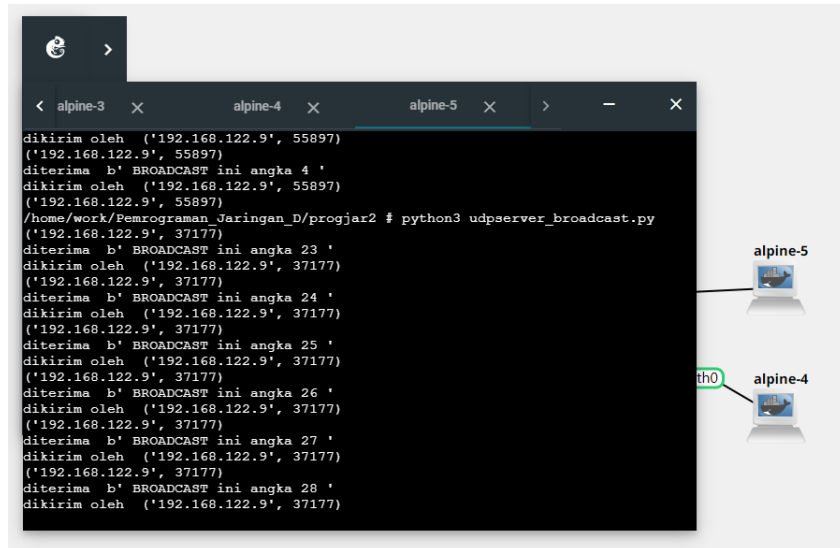


Pada hasil dijalankan dengan urutan server-client terlihat bahwa semua server menerima data secara lengkap, yang mana pada alpine2- alpine5 menerima angka 1-10

- Hasil dijalankan dengan urutan client-server :







```
alpine-3 x alpine-4 x alpine-5 x
dikirim oleh ('192.168.122.9', 55897)
('192.168.122.9', 55897)
diterima b' BROADCAST ini angka 4 '
dikirim oleh ('192.168.122.9', 55897)
('192.168.122.9', 55897)
/home/work/Pemrograman_Jaringan_D/progjar2 # python3 udpserver_broadcast.py
('192.168.122.9', 37177)
diterima b' BROADCAST ini angka 23 '
dikirim oleh ('192.168.122.9', 37177)
('192.168.122.9', 37177)
diterima b' BROADCAST ini angka 24 '
dikirim oleh ('192.168.122.9', 37177)
('192.168.122.9', 37177)
diterima b' BROADCAST ini angka 25 '
dikirim oleh ('192.168.122.9', 37177)
('192.168.122.9', 37177)
diterima b' BROADCAST ini angka 26 '
dikirim oleh ('192.168.122.9', 37177)
('192.168.122.9', 37177)
diterima b' BROADCAST ini angka 27 '
dikirim oleh ('192.168.122.9', 37177)
('192.168.122.9', 37177)
diterima b' BROADCAST ini angka 28 '
dikirim oleh ('192.168.122.9', 37177)
```

Pada hasil dijalankan dengan urutan client-server terlihat bahwa semua server menerima data tidak lengkap dan urutan data angka yang diterima berbeda-beda, yang mana pada :

- *Alpine 2 menerima data angka mulai angka 6-30*
- *Alpine 3 menerima data angka mulai angka 11-30*
- *Alpine 4 menerima data angka mulai angka 17-30*
- *Alpine 5 menerima data angka mulai angka 23-30*

Kesimpulan : Pada saat menjalankan proses broadcast dengan UDP dimulai dari server – client, maka data yang diterima oleh server semuanya sama dan lengkap karena server dijalankan terlebih dahulu baru client. Sedangkan saat dijalankan dengan urutan client-server, data yang diterima tidak lengkap dan berbeda-beda tiap servernya karena client dijalankan terlebih dahulu dan juga tergantung dari urutan server yang dijalankan