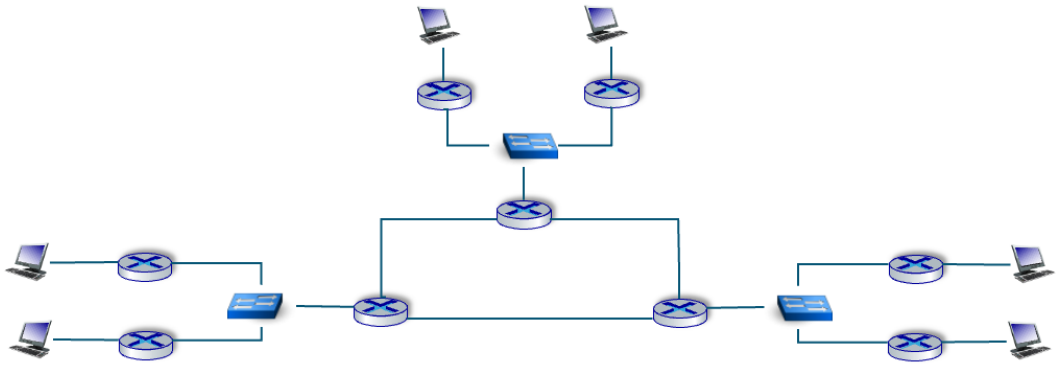
Tugas Inter-domain Routing BGP

Dani Adrian
225150201111009

1. Topologi Jaringan



Topologi ini terdiri dari:

- 9 Router (R1, R2, R3, R1_1, R1_2, R2_1, R2_2, R3_1, R3_2)
- 3 Switch (S1, S2, S3)
- 6 Host (C1_1, C1_2, C2_1, C2_2, C3_1, C3_2)

2. Konfigurasi

Skenario Percobaan 1 (Local Preference dan AS-Path Prepending)

Import Policy - Local Preference

Export Policy - AS-Path Prepending

- R1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1
service integrated-vtysh-config
!
interface R1-eth0
 ip address 10.11.1.1/24
exit
!
interface R1-eth1
 ip address 10.10.1.1/24
exit
!
interface R1-eth2
 ip address 10.10.2.1/24
exit
!
router bgp 100
 no bgp ebgp-requires-policy
 neighbor 10.10.1.2 remote-as 200
```

```
 neighbor 10.10.2.2 remote-as 300

 neighbor 10.10.1.2 soft-reconfiguration inbound
 neighbor 10.10.1.2 route-map SET_LOCAL_PREF in

 neighbor 10.10.2.2 soft-reconfiguration inbound
 neighbor 10.10.2.2 route-map PREPEND_AS_PATH out

 !
 address-family ipv4 unicast
  network 10.11.1.1/24
  network 172.16.1.1/24
  network 172.16.2.1/24
exit-address-family
!
route-map SET_LOCAL_PREF permit 10
 set local-preference 200
exit
!
route-map PREPEND_AS_PATH permit 10
 set as-path prepend 100 100 100
exit
!
router ospf
 ospf router-id 1.1.1.1
 default-information originate always
 network 10.11.1.0/24 area 1
 network 10.10.1.0/24 area 0
 network 10.10.2.0/24 area 0
exit
!
line vty
```

- R1_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_1
service integrated-vtysh-config
!
interface R1_1-eth0
 ip address 10.11.1.2/24
exit
!
interface R1_1-eth1
 ip address 172.16.1.1/24
```

```
exit
!
router ospf
 ospf router-id 1.1.1.2
 network 10.11.1.0/24 area 1
 network 172.16.1.0/24 area 1
exit
!
line vty
```

- R1_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_2
service integrated-vtysh-config
!
interface R1_2-eth0
 ip address 10.11.1.3/24
exit
!
interface R1_2-eth1
 ip address 172.16.2.1/24
exit
!
router ospf
 ospf router-id 1.1.1.3
 network 10.11.1.0/24 area 1
 network 172.16.2.0/24 area 1
exit
!
line vty
```

- R2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2
service integrated-vtysh-config
!
interface R2-eth0
 ip address 10.12.1.1/24
exit
!
interface R2-eth1
 ip address 10.10.1.2/24
exit
```

```
!
interface R2-eth2
 ip address 10.10.3.2/24
exit
!
router bgp 200
 no bgp ebgp-requires-policy
 neighbor 10.10.1.1 remote-as 100
 neighbor 10.10.3.1 remote-as 300

 neighbor 10.10.1.1 soft-reconfiguration inbound
 neighbor 10.10.3.1 soft-reconfiguration inbound

 !
 address-family ipv4 unicast
  network 10.12.1.1/24
  network 172.17.1.1/24
  network 172.17.2.1/24
exit-address-family
!
router ospf
 ospf router-id 2.2.2.1
 default-information originate always
 network 10.12.1.0/24 area 2
 network 10.10.1.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R2_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_1
service integrated-vtysh-config
!
interface R2_1-eth0
 ip address 10.12.1.2/24
exit
!
interface R2_1-eth1
 ip address 172.17.1.1/24
exit
!
router ospf
```

```
 ospf router-id 2.2.2.2
 network 10.12.1.0/24 area 2
 network 172.17.1.0/24 area 2
exit
!
line vty
```

- R2_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_2
service integrated-vtysh-config
!
interface R2_2-eth0
 ip address 10.12.1.3/24
exit
!
interface R2_2-eth1
 ip address 172.17.2.1/24
exit
!
router ospf
 ospf router-id 2.2.2.3
 network 10.12.1.0/24 area 2
 network 172.17.2.0/24 area 2
exit
!
line vty
```

- R3 :

```
frr version 8.5.4
frr defaults traditional
hostname R3
service integrated-vtysh-config
!
interface R3-eth0
 ip address 10.13.1.1/24
exit
!
interface R3-eth1
 ip address 10.10.2.2/24
exit
!
interface R3-eth2
 ip address 10.10.3.1/24
```

```
exit
!
router bgp 300
 no bgp ebgp-requires-policy
 neighbor 10.10.2.1 remote-as 100
 neighbor 10.10.3.2 remote-as 200

 neighbor 10.10.2.1 soft-reconfiguration inbound
 neighbor 10.10.3.2 soft-reconfiguration inbound

 !
 address-family ipv4 unicast
  network 10.13.1.1/24
  network 172.18.1.1/24
  network 172.18.2.1/24
exit-address-family
!
router ospf
 ospf router-id 3.3.3.1
 default-information originate always
 network 10.13.1.0/24 area 3
 network 10.10.2.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R3_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_1
service integrated-vtysh-config
!
interface R3_1-eth0
 ip address 10.13.1.2/24
exit
!
interface R3_1-eth1
 ip address 172.18.1.1/24
exit
!
router ospf
 ospf router-id 3.3.3.2
 network 10.13.1.0/24 area 3
 network 172.18.1.0/24 area 3
```

```
exit
!
line vty
```

- R3_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_2
service integrated-vtysh-config
!
interface R3_2-eth0
 ip address 10.13.1.3/24
exit
!
interface R3_2-eth1
 ip address 172.18.2.1/24
exit
!
router ospf
 ospf router-id 3.3.3.3
 network 10.13.1.0/24 area 3
 network 172.18.2.0/24 area 3
exit
!
line vty
```

Skenario Percobaan 2 (Prefix filtering)

Import Policy - Prefix filtering (summarizing - hanya menerima /23 saja)
Export Policy - Prefix filtering (hanya export prefix sendiri (/23 dan /24), bukan menjadi transit)

- R1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1
service integrated-vtysh-config
!
interface R1-eth0
 ip address 10.11.1.1/24
exit
!
interface R1-eth1
 ip address 10.10.1.1/24
exit
```

```
!
interface R1-eth2
 ip address 10.10.2.1/24
exit
!
router bgp 100
 neighbor 10.10.1.2 remote-as 200
 neighbor 10.10.2.2 remote-as 300

 neighbor 10.10.1.2 soft-reconfiguration inbound
 neighbor 10.10.2.2 soft-reconfiguration inbound

 !
 address-family ipv4 unicast
  network 10.11.1.1/24
  network 172.16.1.1/23
  network 172.16.2.1/23
  neighbor 10.10.1.2 route-map IMPORT_FILTER in
  neighbor 10.10.2.2 route-map IMPORT_FILTER in
  neighbor 10.10.1.2 route-map EXPORT_FILTER out
  neighbor 10.10.2.2 route-map EXPORT_FILTER out
 exit-address-family
!
ip prefix-list ACCEPT_ONLY_23 seq 5 permit
0.0.0.0/0 ge 23 le 23
ip prefix-list OWN_PREFIXES seq 5 permit 0.0.0.0/0
ge 23 le 24
!
route-map IMPORT_FILTER permit 10
 match ip address prefix-list ACCEPT_ONLY_23
exit
!
route-map EXPORT_FILTER permit 10
 match ip address prefix-list OWN_PREFIXES
exit
!
router ospf
 ospf router-id 1.1.1.1
 default-information originate always
 network 10.11.1.0/24 area 1
 network 10.10.1.0/24 area 0
 network 10.10.2.0/24 area 0
exit
!
line vty
```

- R1_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_1
service integrated-vtysh-config
!
interface R1_1-eth0
 ip address 10.11.1.2/24
exit
!
interface R1_1-eth1
 ip address 172.16.1.1/23
exit
!
router ospf
 ospf router-id 1.1.1.2
 network 10.11.1.0/24 area 1
 network 172.16.1.0/23 area 1
exit
!
line vty
```

- R1_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R1_2
service integrated-vtysh-config
!
interface R1_2-eth0
 ip address 10.11.1.3/24
exit
!
interface R1_2-eth1
 ip address 172.16.2.1/23
exit
!
router ospf
 ospf router-id 1.1.1.3
 network 10.11.1.0/24 area 1
 network 172.16.2.0/23 area 1
exit
!
line vty
```

- R2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2
service integrated-vtysh-config
!
interface R2-eth0
 ip address 10.12.1.1/24
exit
!
interface R2-eth1
 ip address 10.10.1.2/24
exit
!
interface R2-eth2
 ip address 10.10.3.2/24
exit
!
router bgp 200
 neighbor 10.10.1.1 remote-as 100
 neighbor 10.10.3.1 remote-as 300

 neighbor 10.10.1.1 soft-reconfiguration inbound
 neighbor 10.10.3.1 soft-reconfiguration inbound

 !
 address-family ipv4 unicast
  network 10.12.1.1/24
  network 172.17.1.1/23
  network 172.17.2.1/23
  neighbor 10.10.1.1 route-map IMPORT_FILTER in
  neighbor 10.10.3.1 route-map IMPORT_FILTER in
  neighbor 10.10.1.1 route-map EXPORT_FILTER out
  neighbor 10.10.3.1 route-map EXPORT_FILTER out
 exit-address-family
!
ip prefix-list ACCEPT_ONLY_23 permit 0.0.0.0/0 ge
23 le 23
ip prefix-list OWN_PREFIXES permit 0.0.0.0/0 ge 23
le 24
!
route-map IMPORT_FILTER permit 10
 match ip address prefix-list ACCEPT_ONLY_23
exit
!
route-map EXPORT_FILTER permit 10
 match ip address prefix-list OWN_PREFIXES
```

```
exit
!
router ospf
 ospf router-id 2.2.2.1
 default-information originate always
 network 10.12.1.0/24 area 2
 network 10.10.1.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R2_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_1
service integrated-vtysh-config
!
interface R2_1-eth0
 ip address 10.12.1.2/24
exit
!
interface R2_1-eth1
 ip address 172.17.1.1/23
exit
!
router ospf
 ospf router-id 2.2.2.2
 network 10.12.1.0/24 area 2
 network 172.17.1.0/23 area 2
exit
!
line vty
```

- R2_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R2_2
service integrated-vtysh-config
!
interface R2_2-eth0
 ip address 10.12.1.3/24
exit
!
interface R2_2-eth1
```

```
 ip address 172.17.2.1/23
exit
!
router ospf
 ospf router-id 2.2.2.3
 network 10.12.1.0/24 area 2
 network 172.17.2.0/23 area 2
exit
!
line vty
```

- R3 :

```
frr version 8.5.4
frr defaults traditional
hostname R3
service integrated-vtysh-config
!
interface R3-eth0
 ip address 10.13.1.1/24
exit
!
interface R3-eth1
 ip address 10.10.2.2/24
exit
!
interface R3-eth2
 ip address 10.10.3.1/24
exit
!
router bgp 300
 neighbor 10.10.2.1 remote-as 100
 neighbor 10.10.3.2 remote-as 200

 neighbor 10.10.2.1 soft-reconfiguration inbound
 neighbor 10.10.3.2 soft-reconfiguration inbound

 !
 address-family ipv4 unicast
  network 10.13.1.1/24
  network 172.18.1.1/23
  network 172.18.2.1/23
  neighbor 10.10.2.1 route-map IMPORT_FILTER in
  neighbor 10.10.3.2 route-map IMPORT_FILTER in
  neighbor 10.10.2.1 route-map EXPORT_FILTER out
  neighbor 10.10.3.2 route-map EXPORT_FILTER out
```

```
 exit-address-family
!
ip prefix-list ACCEPT_ONLY_23 permit 0.0.0.0/0 ge
23 le 23
ip prefix-list OWN_PREFIXES permit 0.0.0.0/0 ge 23
le 24
!
route-map IMPORT_FILTER permit 10
 match ip address prefix-list ACCEPT_ONLY_23
exit
!
route-map EXPORT_FILTER permit 10
 match ip address prefix-list OWN_PREFIXES
exit
!
router ospf
 ospf router-id 3.3.3.1
 default-information originate always
 network 10.13.1.0/24 area 3
 network 10.10.2.0/24 area 0
 network 10.10.3.0/24 area 0
exit
!
line vty
```

- R3_1 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_1
service integrated-vtysh-config
!
interface R3_1-eth0
 ip address 10.13.1.2/24
exit
!
interface R3_1-eth1
 ip address 172.18.1.1/23
exit
!
router ospf
 ospf router-id 3.3.3.2
 network 10.13.1.0/24 area 3
 network 172.18.1.0/23 area 3
exit
!
line vty
```

- R3_2 :

```
frr version 8.5.4
frr defaults traditional
hostname R3_2
service integrated-vtysh-config
!
interface R3_2-eth0
 ip address 10.13.1.3/24
exit
!
interface R3_2-eth1
 ip address 172.18.2.1/23
exit
!
router ospf
 ospf router-id 3.3.3.3
 network 10.13.1.0/24 area 3
 network 172.18.2.0/23 area 3
exit
!
line vty
```

3. Implementasi

Script Mininet :

ospf-lab.py

```
#!/usr/bin/env python3

from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
import shutil
import time
from pathlib import Path
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.nodelib import LinuxBridge
import argparse

class LinuxRouter( Node ):
     def config( self, **params ):
```

```
          super( LinuxRouter, self).config( **params )
          self.cmd( 'sysctl -w net.ipv4.ip_forward=1' )
          self.cmd('/usr/lib/frr/zebra -A 127.0.0.1 -s
90000000 -f /etc/frr/frr.conf -d')
          self.cmd('/usr/lib/frr/staticd -A 127.0.0.1 -f
/etc/frr/frr.conf -d')
          self.cmd('/usr/lib/frr/ospfd -A 127.0.0.1 -f
/etc/frr/frr.conf -d')
          self.cmd('/usr/lib/frr/bgpd -A 127.0.0.1 -f
/etc/frr/frr.conf -d')

          # region
          # self.cmd( 'sysctl -w
net.ipv6.conf.all.forwarding=1' )
          # self.cmd('/usr/lib/frr/pimd -A 127.0.0.1 -f
/etc/frr/frr.conf -d')
          # self.cmd('/usr/lib/frr/pim6d -A ::1 -f
/etc/frr/frr.conf -d')
          # self.cmd('/usr/lib/frr/isisd -A 127.0.0.1 -f
/etc/frr/frr.conf -d')
          # self.cmd('/usr/lib/frr/ospf6d -A ::1 -f
/etc/frr/frr.conf -d')
          # endregion

     def terminate( self ):
          self.cmd( 'killall zebra staticd ospfd ospf6d
bgpd pathd pimd pim6d ldpd isisd nhrpd vrrpd fabricd' )
          super( LinuxRouter, self ).terminate()

     def start (self):
          return

class OSPFLab(Topo):

     def generate_config(self, router_name, path):
          """ Generate an empty config for each
router.\n
               path: the path of router configs
directory
          """
          router = {"name":router_name}
          path = path % router
          #print(path)
          #config template directory path
          template_path = Path("Template/router")
          Path(path).mkdir(exist_ok=True, parents=True)
```

```
          #copy files from the config template folder
          for file in template_path.iterdir():
               shutil.copy(file, path)

          #modify hostname
          self.replace_hostname(path+"/frr.conf",
"dummy", router_name)
          self.replace_hostname(path+"/vtysh.conf",
"dummy", router_name)

          self.add_ospf_configuration(path+"/frr.conf",
router_name)

          return

     def replace_hostname(self, filepath, toReplace,
replacement):
          """ Replace hostname in a router config \n
               filepath: path to the config file\n
               toReplace: the hostname to replace\n
               replacement: the new hostname\n
          """
          with open(filepath, 'r') as f:
               content = f.readlines()
               for linenum in range (len(content)):
                    if (content[linenum] == "hostname
"+toReplace+"\n"):
                         content[linenum] = "hostname "+
replacement+"\n"
          with open(filepath, "w") as f:
               f.writelines(content)
          return

     def parse_argument(self ):
          parser = argparse.ArgumentParser()
          parser.add_argument( "-g","--generateConfig",

     help="Generate router config files.\n"

     +"This will overwrite existing files",

     action="store_true")
          parser.add_argument("-v", "--verbose",

     help="Prints detailed logs during network creation
and stop",
```

```
     action="store_true")
          flags = parser.parse_args()
          return flags

     def build(self, *args, **kwargs):
          flags = self.parse_argument()
          if(flags.verbose):
               setLogLevel( 'info' )

          # directory to keep the configurations
          config_path = "/home/riady/net101/frr-
config/%(name)s"

          # private directory that will useed by the
routers by bind mounting
          privateDirs = [ ( '/var/log' ),
                          ( '/etc/frr',
config_path),
                          ( '/var/run' ),
                          '/var/mn' ]

          #  R1 subnet
          C1_1 = self.addHost('C1_1',
ip="172.16.1.2/24", defaultRoute="via 172.16.1.1")
          C1_2 = self.addHost('C1_2',
ip="172.16.2.2/24", defaultRoute="via 172.16.2.1")
          R1 = self.addNode("R1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          S1 = self.addSwitch("S1", inNamespace=True)
          R1_1 = self.addNode("R1_1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
          R1_2 = self.addNode("R1_2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)

          # add links for subnet 1
          self.addLink(S1, R1,  intfName2="R1-eth0")
          self.addLink(S1, R1_1, intfName2="R1_1-eth0")
          self.addLink(S1, R1_2, intfName2="R1_2-eth0")

          #region
          # Do not manually set the interface name of a
switch's interface
          # mininet will not be able to automatically
add the interfaces to its bridge
          #endregion
```

```
        self.addLink(C1_1,R1_1, intfName2="R1_1-eth1")
        self.addLink(C1_2,R1_2, intfName2="R1_2-eth1")

        # R2 Subnet
        C2_1 = self.addHost('C2_1',
ip="172.17.1.2/24", defaultRoute="via 172.17.1.1")
        C2_2 = self.addHost('C2_2',
ip="172.17.2.2/24", defaultRoute="via 172.17.2.1")
        R2 = self.addNode("R2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
        S2 = self.addSwitch("S2", inNamespace=True)

        R2_1 = self.addNode("R2_1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
        R2_2 = self.addNode("R2_2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)

        # add links for subnet 2
        self.addLink(S2, R2, intfName2="R2-eth0")
        self.addLink(S2, R2_1, intfName2="R2_1-eth0")
        self.addLink(S2, R2_2, intfName2="R2_2-eth0")

        self.addLink(C2_1,R2_1, intfName2="R2_1-eth1")
        self.addLink(C2_2,R2_2, intfName2="R2_2-eth1")

        # R3 Subnet
        C3_1 = self.addHost('C3_1',
ip="172.18.1.2/24", defaultRoute="via 172.18.1.1")
        C3_2 = self.addHost('C3_2',
ip="172.18.2.2/24", defaultRoute="via 172.18.2.1")
        R3 = self.addNode("R3", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
        S3 = self.addSwitch("S3", inNamespace=True)
        R3_1 = self.addNode("R3_1", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)
        R3_2 = self.addNode("R3_2", cls=LinuxRouter,
ip=None, privateDirs=privateDirs, inNamespace=True)

        # add links for subnet 3
        self.addLink(S3,R3, intfName2="R3-eth0")
        self.addLink(S3,R3_1, intfName2="R3_1-eth0")
        self.addLink(S3,R3_2, intfName2="R3_2-eth0")

        self.addLink(C3_1,R3_1, intfName2="R3_1-eth1")
        self.addLink(C3_2,R3_2, intfName2="R3_2-eth1")

        # Add links between backbone routers
```

```
          self.addLink(R1,R2, intfName1="R1-eth1",
intfName2="R2-eth1")
          self.addLink(R1,R3, intfName1="R1-eth2",
intfName2="R3-eth1")
          self.addLink(R2,R3, intfName1="R2-eth2",
intfName2="R3-eth2")

          confdir = Path(config_path % {"name": ""})
          if (not flags.generateConfig):
               if (not Path.exists(confdir)):
                    # Automatically set to generate
config files if config Path doesn't exists, such as when
first time running the program
                    print("If this is your first time
running the program, ")
                    print("consider running the program
with \"-h\" to see the options")
                    print("="*40)
                    flags.generateConfig=True

          if (flags.generateConfig):
               # Configuration files will be created for
each routers
               for n in self.nodes():
                    print(n)
                    if "cls" in self.nodeInfo(n):
                         node_info = self.nodeInfo(n)
                         if node_info["cls"].__name__ ==
"LinuxRouter":
                              self.generate_config(n,
config_path)
                              pass

          super().build(*args, **kwargs)


start = time.time()
print("This the topology for the OSPF lab")
print("="*40)
net = Mininet(topo=OSPFLab(), switch=LinuxBridge,
controller=None)
finish = time.time()
print("Finished initializing network in:", finish-start,
"seconds")

try:
     pass
```

```
        net.start()
        CLI(net)

finally:
        start = time.time()
        net.stop()
        finish = time.time()
        print("Finished stopping network in:", finish-
start, "seconds")
```

Penjelasan script:

Script Python di atas membuat topologi jaringan menggunakan Mininet, yang merupakan simulator jaringan untuk menguji topologi dan protokol. Script ini menggunakan beberapa modul dari Mininet.

Pada dasarnya, script ini mendefinisikan topologi bernama OSPFLab, di mana setiap router diimplementasikan sebagai node dengan kelas khusus bernama LinuxRouter. Kelas ini diatur untuk menjalankan beberapa routing daemons dari FRR (Free Range Routing) seperti zebra, ospfd, dan bgpd, serta mengaktifkan IP forwarding.

Topologi terdiri dari tiga subnet yang dihubungkan oleh tiga backbone router (R1, R2, R3) dan router lainnya dengan beberapa host (C1_1, C1_2, C2_1, C2_2, C3_1, C3_2), serta switch (S1, S2, S3). Script juga menyediakan opsi untuk menghasilkan file konfigurasi router, dan jika direktori konfigurasi belum ada, file konfigurasi akan dibuat secara otomatis. Jika konfigurasi sudah ada, maka konfigurasi tersebut akan digunakan.

Setelah topologi dibangun, Mininet dijalankan dan menyediakan CLI (Command Line Interface) untuk berinteraksi. Pada akhirnya, script menghentikan jaringan dan mencetak waktu yang dibutuhkan untuk memulai dan menghentikannya.

4. Pengujian dan Hasil

Langkah-langkah pengujian:

Skenario Percobaan 1 (Local Preference dan AS-Path Prepending)

```
R1# show route-map
ZEBRA:
route-map: PREPEND_AS_PATH Invoked: 0 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 0
  Match clauses:
  Set clauses:
  Call clause:
  Action:
    Exit routemap
route-map: SET_LOCAL_PREF Invoked: 0 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 0
  Match clauses:
  Set clauses:
  Call clause:
  Action:
    Exit routemap
OSPF:
route-map: PREPEND_AS_PATH Invoked: 0 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 0
  Match clauses:
  Set clauses:
  Call clause:
  Action:
    Exit routemap
route-map: SET_LOCAL_PREF Invoked: 0 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 0
  Match clauses:
  Set clauses:
  Call clause:
  Action:
    Exit routemap
BGP:
route-map: PREPEND_AS_PATH Invoked: 13 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 13
  Match clauses:
  Set clauses:
    as-path prepend 100 100 100
  Call clause:
  Action:
    Exit routemap
route-map: SET_LOCAL_PREF Invoked: 6 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 6
  Match clauses:
  Set clauses:
    local-preference 200
  Call clause:
  Action:
    Exit routemap
```

```
R1# show ip bgp neighbors 10.10.2.2 advertised-routes
BGP table version is 13, local router ID is 10.11.1.1, vrf id 0
Default local pref 100, local AS 100
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *> 10.11.1.0/24     0.0.0.0                  0         32768 100 100 100 i
 *> 10.12.1.0/24     0.0.0.0                       200      0 100 100 100 200 i
 *> 10.13.1.0/24     0.0.0.0                       200      0 100 100 100 200 300 i
 *> 172.16.1.0/24    0.0.0.0                  0         32768 100 100 100 i
 *> 172.16.2.0/24    0.0.0.0                  0         32768 100 100 100 i
 *> 172.17.1.0/24    0.0.0.0                       200      0 100 100 100 200 i
 *> 172.17.2.0/24    0.0.0.0                       200      0 100 100 100 200 i
 *> 172.18.1.0/24    0.0.0.0                       200      0 100 100 100 200 300 i
 *> 172.18.2.0/24    0.0.0.0                       200      0 100 100 100 200 300 i

Total number of prefixes 9
R1# show ip bgp neighbors 10.10.2.2 advertised-routes
BGP table version is 13, local router ID is 10.11.1.1, vrf id 0
Default local pref 100, local AS 100
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *> 10.11.1.0/24     0.0.0.0                  0         32768 100 100 100 i
 *> 10.12.1.0/24     0.0.0.0                       200      0 100 100 100 200 i
 *> 10.13.1.0/24     0.0.0.0                       200      0 100 100 100 200 300 i
 *> 172.16.1.0/24    0.0.0.0                  0         32768 100 100 100 i
 *> 172.16.2.0/24    0.0.0.0                  0         32768 100 100 100 i
 *> 172.17.1.0/24    0.0.0.0                       200      0 100 100 100 200 i
 *> 172.17.2.0/24    0.0.0.0                       200      0 100 100 100 200 i
 *> 172.18.1.0/24    0.0.0.0                       200      0 100 100 100 200 300 i
 *> 172.18.2.0/24    0.0.0.0                       200      0 100 100 100 200 300 i

Total number of prefixes 9
R1#
```

```
mininet> R1 traceroute R3
traceroute to 10.13.1.1 (10.13.1.1), 30 hops max, 60 byte packets
 1  10.10.1.2 (10.10.1.2)  0.277 ms  0.206 ms  0.169 ms
 2  10.13.1.1 (10.13.1.1)  0.159 ms  0.134 ms  0.125 ms
mininet> R3 traceroute R1
traceroute to 10.11.1.1 (10.11.1.1), 30 hops max, 60 byte packets
 1  10.10.3.2 (10.10.3.2)  0.152 ms  0.119 ms  0.113 ms
 2  10.11.1.1 (10.11.1.1)  0.107 ms  0.090 ms  0.083 ms
```

Skenario Percobaan 2 (Prefix filtering)

```
mininet> R1 vtysh

Hello, this is FRRouting (version 8.5.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

R1# show ip bgp neighbors 10.10.2.2 received-routes
BGP table version is 7, local router ID is 10.11.1.1, vrf id 0
Default local pref 100, local AS 100
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *> 10.13.1.0/24     10.10.2.2                0             0 300 i
 *> 172.16.0.0/23    10.10.2.2                              0 300 100 i
 *> 172.16.2.0/23    10.10.2.2                              0 300 100 i
 *> 172.17.0.0/23    10.10.2.2                              0 300 200 i
 *> 172.17.2.0/23    10.10.2.2                              0 300 200 i
 *> 172.18.0.0/23    10.10.2.2                0             0 300 i
 *> 172.18.2.0/23    10.10.2.2                0             0 300 i

Total number of prefixes 7 (1 filtered)
R1# show ip bgp neighbors 10.10.2.2 routes
BGP table version is 7, local router ID is 10.11.1.1, vrf id 0
Default local pref 100, local AS 100
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *  172.17.0.0/23    10.10.2.2                              0 300 200 i
 *  172.17.2.0/23    10.10.2.2                              0 300 200 i
 *> 172.18.0.0/23    10.10.2.2                0             0 300 i
 *> 172.18.2.0/23    10.10.2.2                0             0 300 i

Displayed  4 routes and 11 total paths
R1# show ip prefix-list
ZEBRA: ip prefix-list ACCEPT_ONLY_23: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 23
ZEBRA: ip prefix-list OWN_PREFIXES: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 24
OSPF: ip prefix-list ACCEPT_ONLY_23: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 23
OSPF: ip prefix-list OWN_PREFIXES: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 24
BGP: ip prefix-list ACCEPT_ONLY_23: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 23
BGP: ip prefix-list OWN_PREFIXES: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 24
```


```
R1# show ip prefix-list
ZEBRA: ip prefix-list ACCEPT_ONLY_23: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 23
ZEBRA: ip prefix-list OWN_PREFIXES: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 24
OSPF: ip prefix-list ACCEPT_ONLY_23: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 23
OSPF: ip prefix-list OWN_PREFIXES: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 24
BGP: ip prefix-list ACCEPT_ONLY_23: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 23
BGP: ip prefix-list OWN_PREFIXES: 1 entries
   seq 5 permit 0.0.0.0/0 ge 23 le 24
```

```
R1# show ip bgp neighbors 10.10.1.2 advertised-routes
BGP table version is 7, local router ID is 10.11.1.1, vrf id 0
Default local pref 100, local AS 100
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *> 10.11.1.0/24     0.0.0.0                  0         32768 i
 *> 172.16.0.0/23    0.0.0.0                  0         32768 i
 *> 172.16.2.0/23    0.0.0.0                  0         32768 i
 *> 172.17.0.0/23    0.0.0.0                                0 200 i
 *> 172.17.2.0/23    0.0.0.0                                0 200 i
 *> 172.18.0.0/23    0.0.0.0                                0 300 i
 *> 172.18.2.0/23    0.0.0.0                                0 300 i

Total number of prefixes 7
R1#
```

Analisis:

Skenario Percobaan 1 (Local Preference dan AS-Path Prepending)

Dalam skenario 1, dilakukan konfigurasi prepend AS-Path dan local preference pada BGP. Terdapat dua mekanisme penting untuk mengelola kebijakan routing BGP: local preference dan AS-path prepending.

Local Preference (set local-preference): Pada route-map SET_LOCAL_PREF, nilai local preference diatur menjadi 200 untuk rute yang diterima dari tetangga BGP di 10.10.1.2. Local preference digunakan untuk menentukan jalur yang lebih diutamakan di dalam autonomous system (AS) dalam routing BGP. Semakin tinggi nilainya, semakin disukai jalur tersebut. Dalam konfigurasi ini, rute dari AS 200 mendapatkan prioritas lebih tinggi dibanding rute lain dengan local preference lebih rendah.

AS-path Prepending (set as-path prepend): Pada route-map PREPEND_AS_PATH, dilakukan penambahan AS path (AS-path prepend) sebanyak tiga kali (100 100 100) untuk rute yang diekspor ke tetangga di 10.10.2.2 (AS 300). AS-path prepending adalah teknik yang membuat jalur kurang diinginkan oleh tetangga BGP dengan menambahkan AS yang sama beberapa kali, sehingga jalur terlihat lebih panjang dan dianggap kurang optimal oleh router lain.

Secara keseluruhan, local preference digunakan untuk menentukan jalur terbaik di dalam AS, sedangkan AS-path prepending mempengaruhi bagaimana tetangga memilih jalur menuju AS kita.

Skenario Percobaan 2 (Prefix filtering)

Dalam konfigurasi ini, digunakan dua prefix-list bersama dengan route-map untuk mengatur kebijakan penerimaan dan pengiklanan rute di BGP.

Prefix-list ACCEPT_ONLY_23: Prefix-list ini digunakan untuk memfilter rute yang diterima dari tetangga BGP pada arah inbound. Hanya rute dengan ukuran subnet tepat /23 yang diizinkan. Prefix-list ini diterapkan melalui route-map IMPORT_FILTER, yang memfilter rute yang diterima dari tetangga 10.10.2.2 dan 10.10.1.2 sehingga hanya rute dengan ukuran subnet /23 yang dapat diterima.

Prefix-list OWN_PREFIXES: Prefix-list ini digunakan untuk memfilter rute yang diiklankan (outbound) dari R3 ke tetangga BGP. Hanya rute dengan subnet mask antara /23 dan /24 yang diizinkan untuk diiklankan. Prefix-list ini diterapkan melalui route-map EXPORT_FILTER, yang memastikan bahwa hanya rute dengan kriteria tersebut yang diiklankan ke tetangga 10.10.2.2 dan 10.10.1.2.

Secara keseluruhan, kebijakan ini memastikan bahwa R1 hanya menerima rute dengan subnet mask /23 dan mengiklankan rute dengan subnet mask antara /23 dan /24 kepada tetangga BGP-nya.

5. Kesimpulan

Kebijakan BGP dapat diterapkan secara efektif dengan menggunakan local preference, AS-path prepending, dan prefix filtering untuk mengatur prioritas jalur dan rute yang diiklankan ke tetangga BGP. Pada skenario 1, local preference digunakan untuk mengatur preferensi jalur dalam AS, di mana rute dengan nilai local preference lebih tinggi akan diprioritaskan. AS-path prepending berfungsi untuk mempengaruhi pilihan jalur oleh tetangga BGP dengan memperpanjang jalur, sehingga rute tersebut menjadi kurang diinginkan oleh tetangga. Pada skenario 2, prefix filtering digunakan untuk membatasi rute yang diterima dan diiklankan. Import filtering memastikan hanya rute dengan ukuran subnet tertentu (/23) yang diterima, sedangkan export filtering membatasi rute yang diiklankan hanya pada subnet dengan ukuran /23 dan /24. Secara keseluruhan, kebijakan ini memberikan kontrol penuh terhadap lalu lintas BGP dan memungkinkan manajemen rute yang lebih efisien sesuai kebutuhan jaringan.