

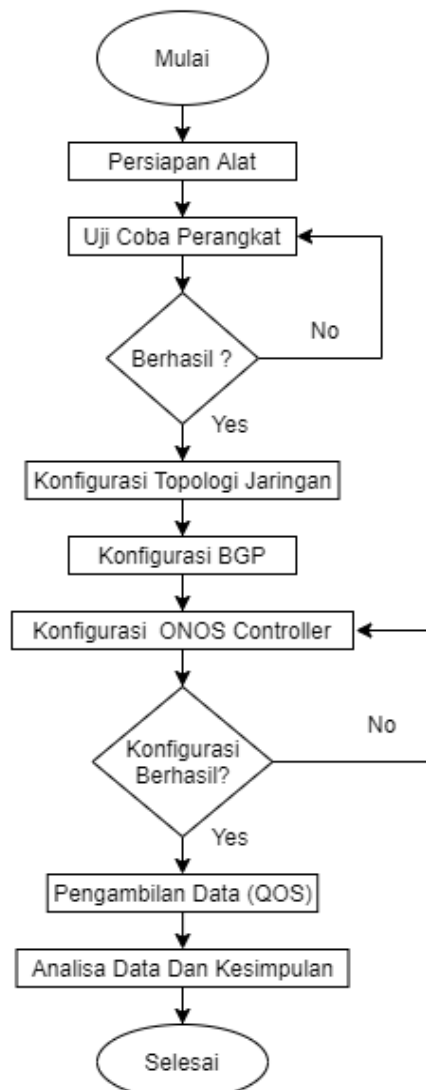
BAB 3

METODELOGI PENELITIAN

Penelitian ini dilakukan menggunakan konsep kuantitatif dimana upaya untuk menyelesaikan penelitian ini dengan mengambil data kemudian di ukur dengan angka untuk dapat di analisa sesuai hasil dari uji coba pada penelitian ini. Untuk penyelesaian penelitian ini penulis melakukan riset dan referensi melalui buku, website, jurnal beserta youtube.

3.1 Flowchart

Beberapa tahap yang di lakukan dalam penelitian ini mengacu pada flowchart di bawah ini :



Gambar 3.1 Flowchart Alur Simulasi

Pada penelitian dibutuhkan serangkaian tahap dalam proses pengerjaan, mulai dari mempersiapkan perangkat hingga pada tahap analisa dan kesimpulan. Serangkaian tahap penelitian ini diperlukan untuk mencapai kinerja dari proses penelitian. Tahap pertama yaitu dimulai dengan persiapan perangkat dimana di tahap ini adalah mempersiapkan untuk simulasi yang ada pada perangkat ubuntu linux.

Uji coba pada perangkat ubuntu bertujuan menguji coba dari perangkat ubuntu untuk mengetahui apakah sudah bekerja sesuai dengan seharusnya atau belum. Apa bila uji coba tidak berhasil maka alur dari tahapan akan kembali ke uji coba kembali, sedangkan apa bila uji coba berhasil maka akan lanjut ke tahap selanjutnya. Pada tahap selanjutnya adalah konfigurasi topologi jaringan dimana pada tahapan ini adalah membuat *BGP* dengan topologi *full mesh* pada *gns3*. Selanjutnya masuk ke tahap konfigurasi jaringan, dimana pada tahap ini adalah memberikan pengalamatan pada topologi agar saling terhubung.

Tahap konfigurasi kontroler *ONOS* di lakukan dengan konfigurasi *tool* dan aplikasi yang telah selesai diinstal, dan lakukan konfigurasi pada setiap perangkat dengan cara memberi *script* pada topologi, dimana menggunakan bahasa pemrograman *python*. Dalam topologi jaringan itu terdapat perangkat switch, *host*, *ovs*, *Frrouting* dan *SDN controller* yang jumlah dari setiap perangkat akan diuraikan pada pembahasan selanjutnya. Untuk *controller* yang digunakan adalah *ONOS controller*. Selanjutnya apa bila konfigurasi *ONOS* tidak berhasil maka akan kembali ke tahap konfigurasi sistem jaringan, dan apa bila konfigurasi berhasil maka akan lanjut ke tahapan selanjutnya. Untuk tahap selanjutnya adalah pengambilan data, tahap ini untuk mengambil atau merecord data yang akan digunakan sebagai bahan yang akan di analisa. Tahap selanjutnya adalah menganalisa data yang sudah didapatkan dan mengambil kesimpulannya.

3.2 Perangkat Simulasi

3.2.1 Perangkat Keras (*Hardware*)

Perangkat-perangkat yang di gunakan untuk penelitian ini yaitu satu buah PC dengan spesifikasi sebagai berikut :

Table 3.1 Spesifikasi PC

Sistem Operasi	Ubuntu 16.04
<i>Processor</i>	Intel(R) Core(TM) i7-9700KF CPU @3.60GHz 3.60 GHz
RAM	32.0 GB
Hardisk	500 GB

3.2.2. Perangkat Lunak (*Software*)

Perangkat lunak (*software*) digunakan sebagai penunjang topologi jaringan uji dalam penelitian ini. Perangkat tersebut diuraikan sebagai berikut :

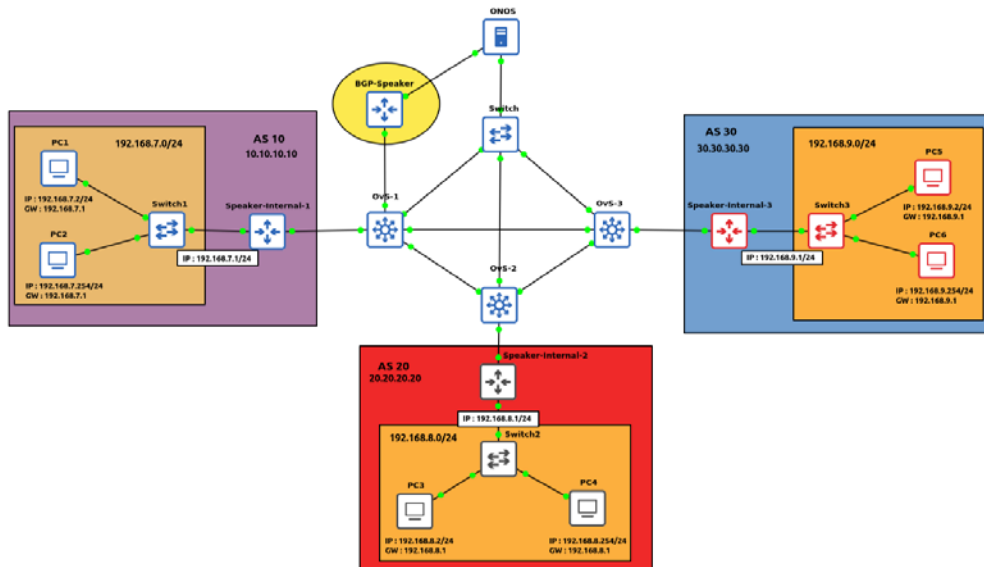
1. GNS3 digunakan untuk mensimulasikan data *plane* dari SDN. GNS3 di install pada PC dan akan di hubungkan pada *software* KVM yang digunakan dalam penelitian ini.
2. Open v Switch digunakan mengontrol/ mengatur *trafik flow* pada switch. Yang telah di install pada Linux Ubuntu 16.04 yang digunakan pada penelitian ini.
3. *ONOS Controller* digunakan untuk menhandel seluruh infrastruktur jaringan
4. DIT-G digunakan untuk penangkapan *traffic* yang berada pada topologi jaringan . Penangkapan *traffic* tersebut dimaksudkan untuk memvalidasi atau memastikan bahwa konfigurasi berjalan didalam jaringan atau tidak.
5. *Frouting* di gunakan untuk implementasi *routing BGP* pada SDN.
6. KVM digunakan untuk menghubungkan gns3 vm

3.2.3. Rancangan Penelitian

Pada rancang bangun terlebih dahulu membuat Topologi jaringan berkaitan tentang bagaimana perangkat jaringan terkoneksi yang membentuk topologi *full mesh*. Membuat topologi itu sendiri bertujuan untuk membentuk bagaimana prangkat jaringan terkoneksi apa bila di lihat secara fisik (tampak kelihatan bentuk fisik jaringan). Pada penelitian ini menggunakan jenis topologi jaringan *full mesh* menggunakan 4 *fr routing*, 3 *ovs*, 1 *switch*, 6 *host* dan 1 *controller*.

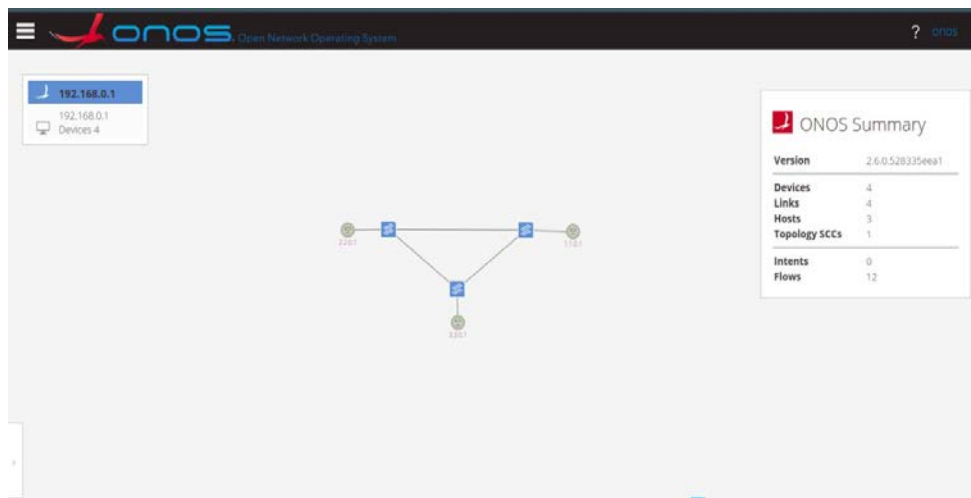
Tabel 3.2 Konfigurasi Topologi SDN

Device	Interface	IP Address	Prefix	Default Getaway
PC 1	eth 0	192.168.7.2	/24	192.168.7.1
PC 2	eth 0	192.168.7.254	/24	192.168.7.1
PC 3	eth 0	192.168.8.2	/24	192.168.8.1
PC 4	eth 0	192.168.8.254	/24	192.168.8.1
PC 5	eth 0	192.168.9.2	/24	192.168.9.1
PC 6	eth 0	192.168.9.254	/24	192.168.9.1
FR-R 1	eth 0	1.1.0.1	/30	
	eth 1	192.168.7.1	/24	
FR-R 2	eth 0	2.2.0.1	/30	
	eth 1	192.168.8.1	/24	
FR-R 3	eth 0	3.3.0.1	/30	
	eth 1	192.168.9.1	/24	
BGP SPEAKER	eth 0	192.168.0.2	/30	
	eth 1	1.1.0.2	/30	
	eth 1	2.2.0.2	/30	
	eth 1	3.3.0.2	/30	
OVS 1	eth 0	20.20.20.1	/24	
OVS 2	eth 0	20.20.20.2	/24	
OVS 3	eth 0	20.20.20.3	/24	
ONOS	eth 0	192.168.0.1	/30	
	eth 1	29.29.20.99	/24	
	eth 2	50.50.50.99	/30	



Gambar 3.2 Topologi jaringan *full mesh BGP SDN* pada GNS3

Pada topologi di atas terdapat 3 AS (*Autonomus system*) yaitu AS 10, AS 20 dan AS 30 yang terdapat 3 *Frouting*, 3 OvS, 1 switch dan 6 *host*. Pada AS10 terdapat 1 ovs, 1 *Frouting*, dan 2 *host*, pada AS20 terdapat 1 ovs, 1 *Frouting*, dan 2 *host*, dan pada AS30 terdapat 1 ovs, 1 *Frouting*, dan 2 *host*. dan akan di hubungkan menggunakan konfigurasi *routing* protokol *BGP*. Jadi *Frouting* sebagai router untuk *BGP* pada SDN, lalu untuk Open vSwitch sebagai yang menjalankan fungsi data *plan* jaringan SDN, Seperti pada gambar prangkat akan terhubung ke AS *number* yang ada. Pada router 1 yaitu *IBGP* Speaker yang pada dasarnya untuk membangun sebuah SDN-IP perlu *BGP* speaker, yang fungsinya untuk menerima piringan pada *eBGP* yang lain. hal ini yang memungkinkan konfigurasi jaringan dilakukan secara terpusat. dan *ONOS* sebagai *controller* pada SDN. Jadi Setiap prangkat akan terhubung dan membentuk topologi *full mash*. Sistem oprasi *ONOS* digunakan sebagai *controller* yang bekerja sebagai control *plan* pada topologi SDN yang mengatur jalannya prangkat data *plan*, berikut simulasi GUI *ONOS* yang di akses pada



Gambar 3.3 Topologi SDN pada ONOS

Dengan melakukan simulasi jaringan SDN dengan topologi *full mesh* pada gns3. Data yang akan diambil yaitu *delay*, *jitter*, dan *throughput*. Topologi yang dijalankan pada gns3 dilakukan pengontrolan dengan menggunakan *controller* *ONOS* dan menjalankan *BGP*. Konfigurasi *routing* *BGP* dilakukan dengan *Frouting*. Penelitian ini bertujuan untuk mengetahui bagaimana kinerja *routing* dengan protokol *BGP* pada jaringan SDN dengan menggunakan *Frouting* pada topologi *full mesh*. Data akan diambil dengan melakukan pengiriman data dari masing-masing *host* ke *host* lainnya. Data yang akan di ambil yaitu menangkap

trafik dengan DIT-G saat *host* mengirimkan data. Dari *data delay*, *jitter*, dan *throughput* akan dianalisis dan disimpulkan. Untuk menghubungkan semua perangkat satu sama lain ada beberapa konfigurasi yang harus di lakukan yaitu :

3.2.3.1 Konfigurasi pada Host

Konfigurasi yang lakukan pada *host* yaitu memberikan *IP Address*, *Prefix* dan *Getaway* berikut info setelah dilakukan konfigurasi dengan perintah “*show ip all*” Pada gambar 3.4 merupakan *IP Address*, *Prefix* dan *Getaway* pada PC 1 di berikan pengalamatan bertujuan agar dapat saling dengan host yang lain.

```
PC1> show ip all
```

NAME	IP/MASK	GATEWAY
PC1	192.168.7.254/24	192.168.7.1

Gambar 3.4. Melihat Info IP Pada Host

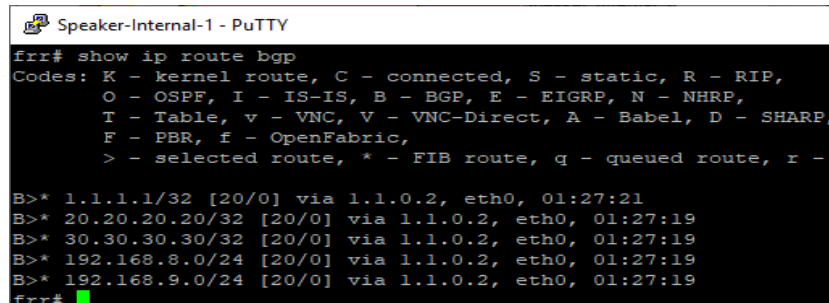
3.2.3.2 Konfigurasi pada Frrouter

Konfigurasi yang lakukan pada router yaitu memberikan *IP address* beserta *prefix* sesuai *interfacenya* dan mengkonfigurasi *routing BGP* pada jaringan SDN. Berikut info setelah dilakukan konfigurasi dengan perintah “*show running-config*” pada router untuk melihat apakah sudah diconfig atau belum ipnya, Dapat dilihat pada gambar 3.4 informasi yang tertera sudah di berikan ip pada setiap *interface* dan *BGP router*.

```
hostname frr
service integrated-vtysh-config
!
interface eth0
 ip address 1.1.0.1/30
!
interface eth1
 ip address 192.168.7.1/24
!
interface lo
 ip address 10.10.10.10/32
!
router bgp 10
 bgp router-id 10.10.10.10
 neighbor 1.1.0.2 remote-as 100
!
 address-family ipv4 unicast
  network 10.10.10.10/32
  network 192.168.7.0/24
 exit-address-family
!
```

Gambar 3.5 Melihat Info Konfigurasi Pada Router BGP

Hasil konfigurasi pada speaker-internal-1 dibuktikan dengan cara “*show ip route bgp*”, terdapat routing tabel yang berisikan 5 informasi BGP *peering*. 1 BGP *Speaker*, 2 *Speaker internal* dan 2 *Network clien*



```

Speaker-Internal-1 - PuTTY
frr# show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r -
B>* 1.1.1.1/32 [20/0] via 1.1.0.2, eth0, 01:27:21
B>* 20.20.20.20/32 [20/0] via 1.1.0.2, eth0, 01:27:19
B>* 30.30.30.30/32 [20/0] via 1.1.0.2, eth0, 01:27:19
B>* 192.168.8.0/24 [20/0] via 1.1.0.2, eth0, 01:27:19
B>* 192.168.9.0/24 [20/0] via 1.1.0.2, eth0, 01:27:19
frr#

```

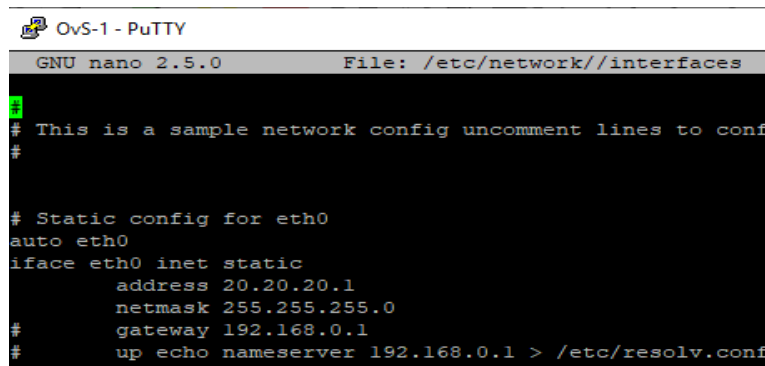
Gambar 3.6 Info Router BGP

Tabel 3.3 Konfigurasi BGP Pada SDN

Router	ASN	BGP Router ID
Router Speaker Internal-1	10	10.10.10.10
Router Speaker Internal-2	20	20.20.20.20
Router Speaker Internal-3	30	30.30.30.30
Router Speaker	100	1.1.1.1

3.2.3.3 Konfigurasi Pada OvS

Konfigurasi yang dilakukan memberikan ip *address* sesuai dengan *interface* nya agar bisa saling bisa terhubung. Seluruh OvS menggunakan 1 ip *network* yang sama sehingga antar OvS dapat saling berkomunikasi dengan melalui *management controller*.



```

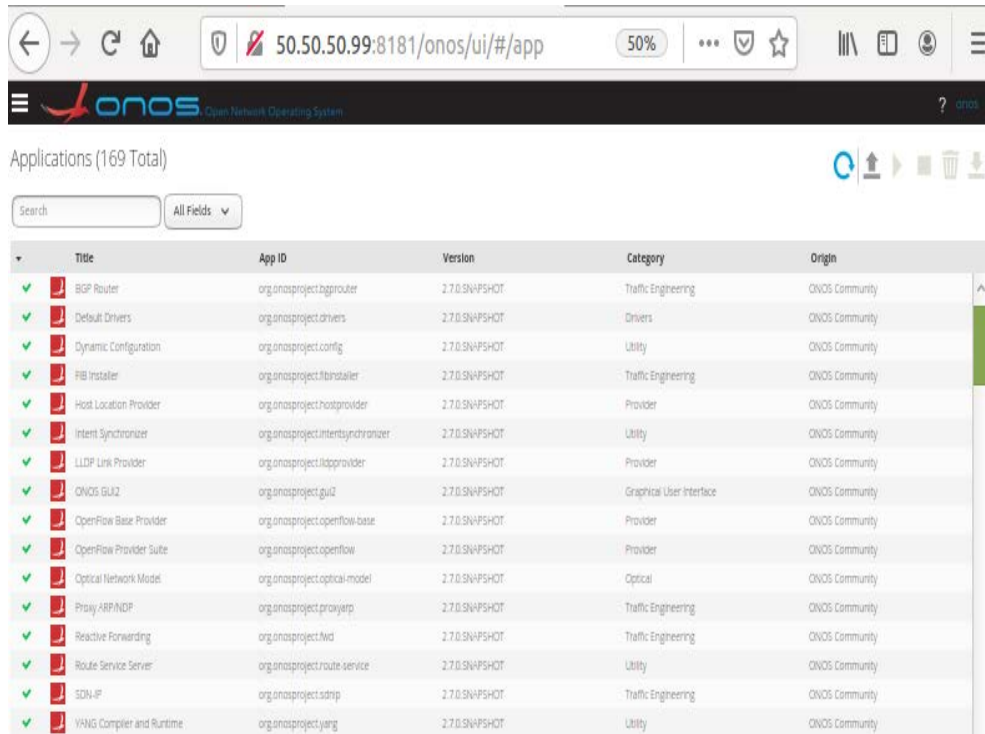
OvS-1 - PuTTY
GNU nano 2.5.0 File: /etc/network/interfaces
# This is a sample network config uncomment lines to confi
#
# Static config for eth0
auto eth0
iface eth0 inet static
    address 20.20.20.1
    netmask 255.255.255.0
#    gateway 192.168.0.1
#    up echo nameserver 192.168.0.1 > /etc/resolv.conf

```

Gambar 3.7 Konfigurasi Alamat Pada OvS

3.2.3.4 Konfigurasi Pada ONOS Controller

Konfigurasi pada *ONOS* dapat di akses dengan 2 cara yaitu GUI dan CLI. Cara mengakses *ONOS* gui dengan menggunakan browser lalu memasukan url



Gambar 3.8 Featur Pada GUI ONOS

berikut “50.50.50.99:8181/ONOS/ui/index.html, saat masuk dalam *ONOS* dapat mengatur *application* yang akan di aktifkan. Berikut aplikasi aplikasi yang di aktifkan pada *ONOS*.

Pada gambar 3.5 *featur* pada GUI *ONOS* terdapat 14 *applications* yang dibutuhkan untuk membangun infrastruktur SDN-IP. Setelah semua perangkat sudah dikonfigurasi maka selanjutnya mengkonfigurasi *network* SDN-IP pada *control plan*. Ada beberapa beberapa tahapan yang dilakukan untuk mengkonfigurasi SDN-IP agar dapat merespon ARP dan konektivitas untuk jalur pada *BGP*. Cara untuk melakukan konfigurasi yaitu dengan meghupload file *network-cfg.json* ke dalam *control plan*, dengan perintah “`curl --user onos:rocks -X POST -H "Content-Type: application/json" http://50.50.50.99:8181/onos/v1/network/configuration/ -d @/tmp/network-cfg.json`”


```

{
  "ports" : [
    "of:00004a4c7dfadb46/2" : [
      "interfaces" : [
        [
          "ips" : [ "1.1.0.2/30" ],
          "mac" : "0e:21:f8:20:27:00"
        ]
      ]
    ],
    "of:00008214adf16444/2" : [
      "interfaces" : [
        [
          "ips" : [ "2.2.0.2/30" ],
          "mac" : "0e:21:f8:e7:a8:00"
        ]
      ]
    ],
    "of:00008608fa00aa42/2" : [
      "interfaces" : [
        [
          "ips" : [ "3.3.0.2/30" ],
          "mac" : "0e:21:f8:7f:c8:00"
        ]
      ]
    ]
  ],
  "apps" : [
    "org.onosproject.router" : {
      "bgp" : {
        "bgpSpeakers" : [
          [
            "name" : "speaker-internal",
            "connectPoint" :
              "of:00004a4c7dfadb46/5",
            "peers" : [
              "1.1.0.1",
              "2.2.0.2",
              "3.3.0.3"
            ]
          ]
        ]
      }
    ]
  ]
}

```

Gambar 3.9 File Konfigurasi *Network* Pada SDN

Pada gambar 3.6 dapat di lihat terdapat informasi pada *network-cfg.json* yang berisikan IP Address dari seluruh speaker-internal yang terhubung ke OvS. Terdapat informasi *openflow* ID dari setiap OvS dan MAC Address setiap speaker-internal. Dengan dikonfigurasinya pengaturan pada *network-cfg.json* maka intens pada *ONOS controller* akan terisi. Tujuan mengupload JSON agar semua dapat terhubung dengan controller ONOS dan bisa saling memberikan informasi dengan baik.

Percobaan dilakukan pada topologi dengan melakukan pengiriman dan penerimaan data dengan besaran yang bervariasi yaitu 12,8 MB, 25,6 MB, dan 32

MB. Untuk *protocol* menggunakan 2 *protocol* data yang berbeda yaitu TCP (*Transmission Control Protocol*) dan UDP (*User Data Protocol*), semua trafik tersebut akan dilakukan pengujian dengan QOS (*Quality of Service*) di antaranya yaitu *delay*, *jitter*, dan *throughput*. seperti telah di tentukan pada tabel berikut :

Table 3.4 Skenario Pengujian

NO	Skenario	Protocol Data	Besar Data (MB)	Banyak Pengujian
1	AS 10-AS20	UDP	12,8	30
2			25,6	30
3			32	30
4		TCP	12,8	30
5			25,6	30
6			32	30
7	AS20-AS30	UDP	12,8	30
8			25,6	30
9			32	30
10		TCP	12,8	30
11			25,6	30
12			32	30