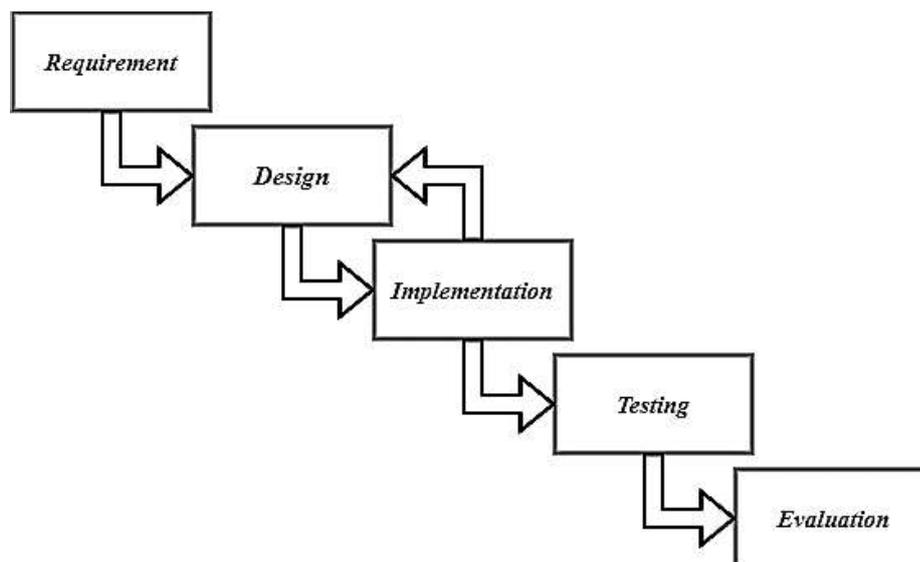


## BAB III METODE PENELITIAN

### 3.1 Desain Penelitian

Untuk mendukung keberhasilan suatu penelitian maka diperlukan adanya suatu metode yang digunakan, penelitian ini menggunakan metode penelitian *System Development Life Cycle (SDLC)* dengan model *waterfall*. Model pengembangan *waterfall* diperkenalkan oleh Watson Royce pada tahun 1970, model pengembangan *waterfall* adalah salah satu model yang sering digunakan pada metode SDLC untuk mengembangkan sistem informasi oleh para peneliti (Isaias & Issa, 2015). Tahapan pengembangan dengan model ini dilakukan secara sistematis dan berurutan dengan 5 tahapan yaitu *analyst, design, implementation testing, dan maintenance* (Abdul Wahid, 2020).

Untuk mendapatkan hasil yang baik dari penelitian ini, harus terlebih dahulu melalui tahapan-tahapan model *waterfall* yang digunakan pada penelitian ini, dengan beberapa tahapan sebagai berikut:



Gambar 3. 1 Desain Penelitian

#### 3.1.1 Requirement

Tahap Requirement dalam model *Waterfall* adalah langkah awal dalam mempersiapkan penelitian dimana persyaratan atau kebutuhan penelitian dikumpulkan, dan didokumentasikan secara rinci seperti pada Tabel 3.1. Tidak hanya dari software dan hardware setiap *tools* otomasi memiliki beberapa persiapan sebelum melakukan otomasi seperti pada Tabel 3.2.

Tabel 3. 1  
Instrumen Penelitian

<i>Software</i>	<i>Hardware</i>
Ubuntu 20.04 GNS3 2.2.42 Qemu 4.2.1 Debian Server 11.7 <i>Images Router C3745</i> Ansible 8 Puppet Bolt 3.0 <i>Library Python (Netmiko, Paramiko, Scrapli, Time, Timeit, Pandas)</i> Wireshark 3.4.8	<i>Processor : Intel G5400 4CPUs</i> 3.70Ghz RAM : 8GB SSD : 120GB HDD : 1TB

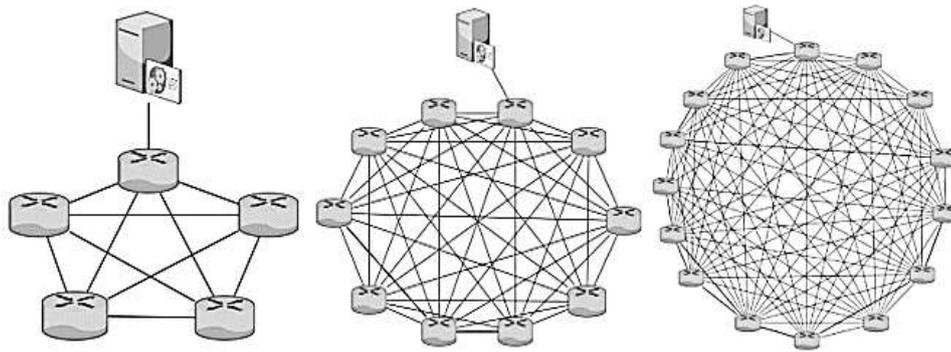
Tabel 3. 2  
Kebutuhan Prekonfigurasi

<b>Ansible</b>	<b>Netmiko</b>	<b>Paramiko</b>	<b>Puppet</b>	<b>Scrapli</b>
ansible.cfg	Scripts	Scripts	Akses SSH	Scripts
Akses SSH	Akses SSH	Akses SSH	<i>Inventory.yaml</i>	Akses SSH
<i>Inventory</i>	<i>Inventory</i>	<i>Inventory</i>	<i>Manifest</i>	<i>Inventory</i>
<i>Ansible-playbook</i>				

### 3.1.2 Design

Dengan perkembangan jaringan yang sangat besar dan heterogen penelitian ini perlu dilakukan untuk menciptakan sistem otomasi yang efektif, penelitian ini berencana untuk menggunakan topologi *full-mesh* untuk menguji performa *network automation tools* dalam mengimplementasikan *network automation tools*.

Pengujian *network automation tools* dilakukan dengan 3 skenario topologi jaringan full-mesh yang dibangun dengan 5 *router*, 10 *router* dan 15 *router* seperti pada Gambar 3.2. Pada setiap skenario pengujian akan dilakukan sebanyak 10 set dari Alamat IP A ke Alamat IP B.



Gambar 3. 2 Topologi jaringan penelitian

Untuk dapat membandingkan *tools* otomasi dengan mudah, hasil dari 10 set otomasi setiap parameter dihitung nilai rata-rata sehingga hasil yang dibandingkan merupakan hasil rata-rata parameter, formula yang digunakan untuk menghitung nilai rata-rata adalah sebagai berikut

$$\bar{x} = \frac{\sum x}{n} \dots\dots\dots(1)$$

$\bar{x}$  = Nilai Rata-rata

$\sum x$  = Jumlah data

$n$  = Banyak data

### 3.1.3 Implementation

Pada tahap ini sistem yang telah dibuat akan coba diterapkan dalam skala yang kecil untuk mengetahui apakah sistem yang telah dibuat sudah mampu berjalan dengan baik atau tidak. Melakukan otomasi dengan *tools* yang digunakan pada penelitian ini memiliki cara yang berbeda-beda dalam menjalankan skrip tersebut. Dalam penggunaan skrip python seperti netmiko, paramiko, dan scrapli. Skrip dijalankan dengan memasukkan perintah python3 “nama\_file.py” maka skrip yang telah dibuat akan jalan dan menampilkan proses otomasi berjalan. Pada *tools* otomasi ansible untuk menjalankan ansible-playbook dapat dengan memasukkan perintah ansible-playbook run “nama\_playbook.yml”. Untuk menjalankan *tools* otomasi puppet dapat dengan memasukkan perintah bolt apply “nama\_manifest.yml” -t target untuk menjalankan manifest dan menentukan target yang akan dituju untuk diotomasi.

```
PLAY RECAP *****
**
R1      : ok=6    changed=5    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
R2      : ok=6    changed=5    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
R3      : ok=6    changed=5    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
R4      : ok=6    changed=5    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
R5      : ok=6    changed=5    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Gambar 3. 3 Hasil otomasi dengan ansible

Pada Gambar 3.3 adalah hasil otomasi yang berhasil dilakukan dengan menggunakan *tools* ansible, dari Gambar 3.3 dapat dilihat indikator yang ditampilkan oleh ansible terdapat 6 dengan keterangan “ok” dan 0 *failed*.

```
configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#router ospf 1
R5(config-router)#network 10.10.9.0 0.0.0.255 area 0
R5(config-router)#int F1/3
R5(config-if)#no switchport
R5(config-if)#ip add 10.10.19.2 255.255.255.0
R5(config-if)#no sh
R5(config-if)#router ospf 1
R5(config-router)#network 10.10.19.0 0.0.0.255 area 1
R5(config-router)#end
R5#
root@NetworkAutomation-1:~/netmiko#
```

Gambar 3. 4 Hasil otomasi dengan netmiko

Pada Gambar 3.4 menunjukkan hasil dari otomasi jaringan menggunakan *tools* netmiko, dari *tools* tersebut juga tidak menampilkan adanya kesalahan dalam proses otomasi.

```
R5>enable
Password:
R5#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#router ospf 1
R5(config-router)#network 10.10.9.0 0.0.0.255 area 0
R5(config-router)#int F1/0
R5(config-if)#no sw
R5(config-if)#ip add 10.10.13.2 255.255.255.0
R5(config-if)#no sh
R5(config-if)#router ospf 1
R5(config-router)#network 10.10.13.0 0.0.0.255 area 1
R5(config-router)#int F1/1
R5(config-if)#no sw
R5(config-if)#ip add 10.10.16.2 255.255.255.0
R5(config-if)#no sh
R5(config-if)#router ospf 1
R5(config-router)#network 10.10.16.0 0.0.0.255 area 1
R5(config-router)#int F1/2
R5(config-if)#no sw
R5(config-if)#ip add 10.10.18.2 255.255.255.0
R5(config-if)#no sh
R5(config-if)#router ospf 1
R5(config-router)#network 10.10.18.0 0.0.0.255 area 1
R5(config-router)#int F1/3
R5(config-if)#no sw
R5(config-if)#ip add 10.10.19.2 255.255.255.0
R5(config-if)#no sh
R5(config-if)#router ospf 1
R5(config-router)#network 10.10.19.0 0.0.0.255 area 1
R5(config-router)#
root@NetworkAutomation-1:~/paramiko#
```

Gambar 3. 5 Hasil otomasi dengan paramiko

Pada Gambar 3.5 juga menunjukkan hasil dari otomasi jaringan menggunakan *tools* paramiko, dari *tools* tersebut juga tidak menampilkan adanya kesalahan dalam proses otomasi.

Pada Gambar 3.6 menunjukkan hasil dari otomasi jaringan menggunakan *tools* puppet, dari *tools* tersebut juga tidak menampilkan adanya kesalahan dalam proses otomasi dengan keterangan “*Finished: apply catalog with 0 failures in 22.24 sec*” dan “*Successful on target 10.10.18.2*”.

```
Notice: /Stage[main]/Main/Ios_config[konfigurasi alamat IP]/command: command changed to "int f1/0\nno sw\nno sh\nip add 10.10.18.2 255.255.255.0\nint f1/1\nno sw\nno sh\nip add 10.20.26.2 255.255.255.0\nint f1/2\nno sw\nno sh\nip add 10.20.33.2 255.255.255.0\nint f1/3\nno sw\nno sh\nip add 10.20.39.2 255.255.255.0\nint f1/4\nno sw\nno sh\nip add 10.20.44.2 255.255.255.0\nint f1/5\nno sw\nno sh\nip add 10.20.48.2 255.255.255.0\nint f1/6\nno sw\nno sh\nip add 10.20.51.1 255.255.255.0\nint f1/7\nno sw\nno sh\nip add 10.20.53.2 255.255.255.0\nint f1/8\nno sw\nno sh\nip add 10.20.54.2 255.255.255.0"
Notice: /Stage[main]/Main/Ios_config[konfigurasi router OSPF]/command: command changed to "router ospf 1\nnetw 10.10.18.0 0.0.0.255 area 1\nnetw 10.20.26.0 0.0.0.255 area 1\nnetw 10.20.33.0 0.0.0.255 area 1\nnetw 10.20.39.0 0.0.0.255 area 1\nnetw 10.20.44.0 0.0.0.255 area 1\nnetw 10.20.48.0 0.0.0.255 area 1\nnetw 10.20.51.0 0.0.0.255 area 1\nnetw 10.20.53.0 0.0.0.255 area 1\nnetw 10.20.54.0 0.0.0.255 area 1"
changed: 2, failed: 0, unchanged: 0 skipped: 0, noop: 0
Finished: apply catalog with 0 failures in 22.24 sec
Successful on 1 target: 10.10.18.2
Ran on 1 target in 28.4 sec
```

Gambar 3. 6 Hasil otomasi dengan puppet

```
KONFIGURASI ROUTER2
int F1/0
no switchport
ip add 10.10.10.2 255.255.255.0
no sh
router ospf 1
network 10.10.10.0 0.0.0.255 area 1

int F1/1
no switchport
ip add 10.10.24.1 255.255.255.0
no sh
router ospf 1
network 10.10.24.0 0.0.0.255 area 1

int F1/2
no switchport
ip add 10.10.25.1 255.255.255.0
no sh
router ospf 1
network 10.10.25.0 0.0.0.255 area 1
```

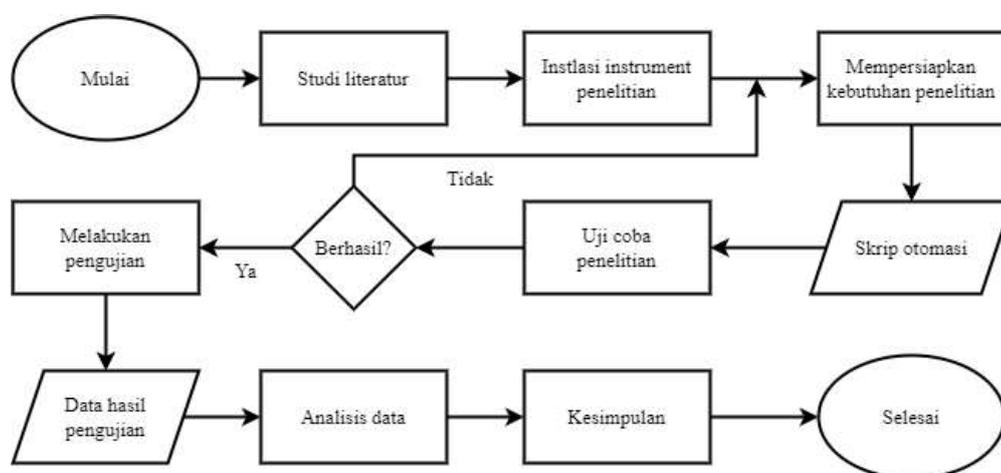
Gambar 3. 7 Hasil otomasi dengan scrapli

Pada Gambar 3.7 menunjukkan hasil dari otomasi jaringan menggunakan *tools* scrapli, dari *tools* tersebut juga tidak menampilkan adanya kesalahan dalam proses otomasi. Pada scrapli, netmiko dan paramiko dalam proses otomasi akan menampilkan perintah-perintah yang diinput kepada *router* seperti pada gambar 3.4, 3.5 dan 3.7. Setelah seluruh *tools* berhasil dilakukan uji coba maka penelitian dapat dilanjutkan untuk tahap *testing* atau pengujian.



### 3.2 Alur penelitian

Alur penelitian memiliki manfaat penting dalam mengarahkan penelitian dengan struktur yang sistematis, membantu merencanakan metode yang tepat, mengendalikan variabel dan menghindari bias, serta meningkatkan kredibilitas hasil penelitian. Ini juga memungkinkan replikabilitas, efisiensi sumber daya, dan kemudahan komunikasi hasil. Dengan alur penelitian yang jelas, peneliti dapat mencapai tujuan penelitian dengan lebih efektif dan berkontribusi pada perkembangan pengetahuan. Berikut alur penelitian yang dilakukan pada penelitian ini.



Gambar 3. 10 Alur penelitian

Berikut penjelasan tahapan alur penelitian yang ada pada Gambar 3.10 yang diterapkan pada penelitian ini:

#### 1. Studi Literatur

Tahap ini adalah proses menyelidiki literatur dari berbagai sumber yang relevan untuk membantu dalam proses pengerjaan penelitian melalui berbagai wawasan dan pemahaman.

#### 2. Instalasi instrument penelitian

Tahap ini adalah proses persiapan yang dilakukan untuk menciptakan lingkungan penelitian agar penelitian tetap berjalan sesuai dengan arah dan tujuan penelitian.

#### 3. Mempersiapkan kebutuhan penelitian

Pada tahap ini, proses yang dikerjakan adalah membangun atau memperbaiki skrip dan desain topologi untuk penelitian. Hasil dari tahap ini berupa skrip

otomasi yang selanjutnya dilakukan uji coba atau bisa digunakan untuk melakukan pengujian.

#### 4. Uji coba penelitian

Tahap ini dilakukan untuk mengetahui apakah skrip yang telah dibangun sudah mampu melakukan eksekusi terhadap seluruh perintah atau masih memiliki kendala. Jika terdapat kendala pada tahap ini maka penelitian akan kembali kepada tahap Mempersiapkan kebutuhan penelitian.

#### 5. Pengujian

Pada tahap ini dilakukan pengujian untuk didapatkan suatu data agar di analisis untuk mendapatkan sebuah hasil dan kesimpulan dari penelitian ini dengan data-data yang valid.

#### 6. Analisis data

Analisis data dilakukan untuk mendapatkan sebuah kesimpulan dari penelitian ini, data yang digunakan untuk di analisis adalah data yang didapatkan dari hasil pengujian.

#### 7. Kesimpulan

Dari hasil analisis data penelitian dapat diakhiri dengan membuat kesimpulan dari hasil pengujian dengan rangkuman singkat dari poin-poin penting yang telah dibahas dalam penelitian