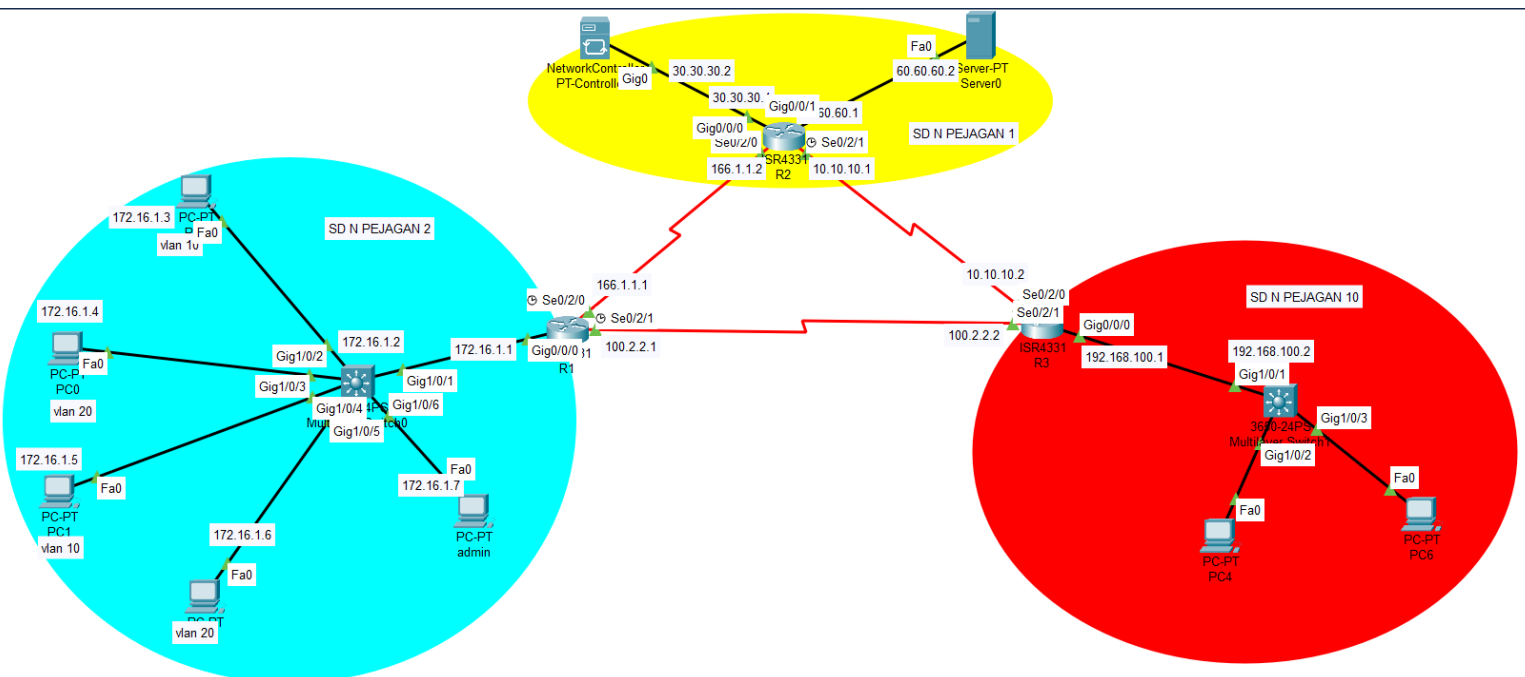


Nama : Afrizal Syahruluddin Yusuf
Kelas : B
Program : Network Programmability

Studi kasus SD Negeri Pejagan 1 sebagai Server. SD Negeri Pejagan 2 sebagai client dan 1 admin. SD Negeri Pejagan 10 sebagai client. Ke 3 SD Negeri tersebut berada di Kabupaten Bangkalan, Jawa Timur. Sebagai sekolah rujukan program Sekolah penggerak dan pusat pelaksanaan Assessment Nasional.

a. Cisco Packet Tracer 8.x.

- Desain memuat minimal 3 (tiga) unit Cisco 4331 routers, 2 (dua) unit Cisco 3650 Switches, 6 (enam) unit PC client, 1 (satu) Server dan wajib menggunakan 1 (satu) Network Controller.



- Desain jaringan menerapkan skema pengalamatan Variable Length Subnet Mask (VLSM) dan menerapkan Virtual Local Area Network (VLAN) serta menggunakan routing protocol.
(KONFIGURASI ROUTER 1)

```
conf t
ip domain-name afrizal.com
crypto key generate rsa
! 1024
```

```
ip ssh ver 2
enable password cisco
```

```
line console 0
password cisco
```

```
line vty 0 4
transport input ssh
login local
username afrizal password cisco
```

```
router ospf 1
log-adjacency-changes
network 10.0.0.0 0.255.255.255 area 1
network 11.0.0.0 0.255.255.255 area 1
network 192.168.1.0 0.0.0.255 area 1
```

```
router ospf 1
log-adjacency-changes
network 166.1.0.0 0.0.255.255 area 1
network 100.0.0.0 0.255.255.255 area 1
network 172.16.0.0 0.0.255.255 area 1
```

(KONFIGURASI ROUTER 2)

```
conf t
ip domain-name afrizal.com
crypto key generate rsa
! 1024
```

```
ip ssh ver 2
enable password cisco
```

```
line console 0
password cisco
```

```
line vty 0 4
transport input ssh
login local
username afrizal password cisco
```

```
router ospf 1
log-adjacency-changes
network 166.1.0.0 0.0.255.255 area 1
network 10.0.0.0 0.255.255.255 area 1
network 30.0.0.0 0.255.255.255 area 1
network 60.0.0.0 0.255.255.255 area 1
```

(KONFIGURASI ROUTER 3)

```
conf t
```

```
ip domain-name afrizal.com
crypto key generate rsa
! 1024
```

```
ip ssh ver 2
enable password cisco
```

```
line console 0
password cisco
```

```
line vty 0 4
transport input ssh
login local
username afrizal password cisco
```

```
router ospf 1
log-adjacency-changes
network 10.0.0.0 0.255.255.255 area 1
network 100.0.0.0 0.255.255.255 area 1
network 192.168.100.0 0.0.0.255 area 1
```

(SWITCH 2)

```
enable
conf t
```

```
ip domain-name afrizal.com
crypto key generate rsa
! 1024
```

```
ip ssh ver 2
enable password cisco
line console 0
password cisco
line vty 0 4
transport input ssh
login local
username afrizal password cisco
```

```
int vlan 1
interface Vlan1
ip address 172.16.1.2 255.255.0.0
ip default-gateway 172.16.1.1
no shutdown
```

```
vlan 10
name office
```

```
exit
int vlan 10

interface GigabitEthernet1/0/2
switchport mode access
switchport access vlan 10

interface GigabitEthernet1/0/4
switchport mode access
switchport access vlan 10

interface GigabitEthernet1/0/3
switchport mode access
switchport access vlan 20

interface GigabitEthernet1/0/5
switchport mode access
switchport access vlan 20
```

(SWITCH 1)

```
enable
conf t

ip domain-name afrizal.com
crypto key generate rsa
! 1024

ip ssh ver 2
enable password cisco

line console 0
password cisco

line vty 0 4
transport input ssh
login local
username afrizal password cisco

interface Vlan1
ip address 192.168.100.2 255.255.255.0
exit
ip default-gateway 192.168.100.1
```

- Konfigurasi dasar pengamanan perangkat jaringan meliputi menerapkan password privilege, console dan remote access hanya melalui Secure Shell (SSH).

enable

conf t

ip domain-name afrizal.com

crypto key generate rsa

! 1024

ip ssh ver 2

enable password cisco

line console 0

password cisco

line vty 0 4

transport input ssh

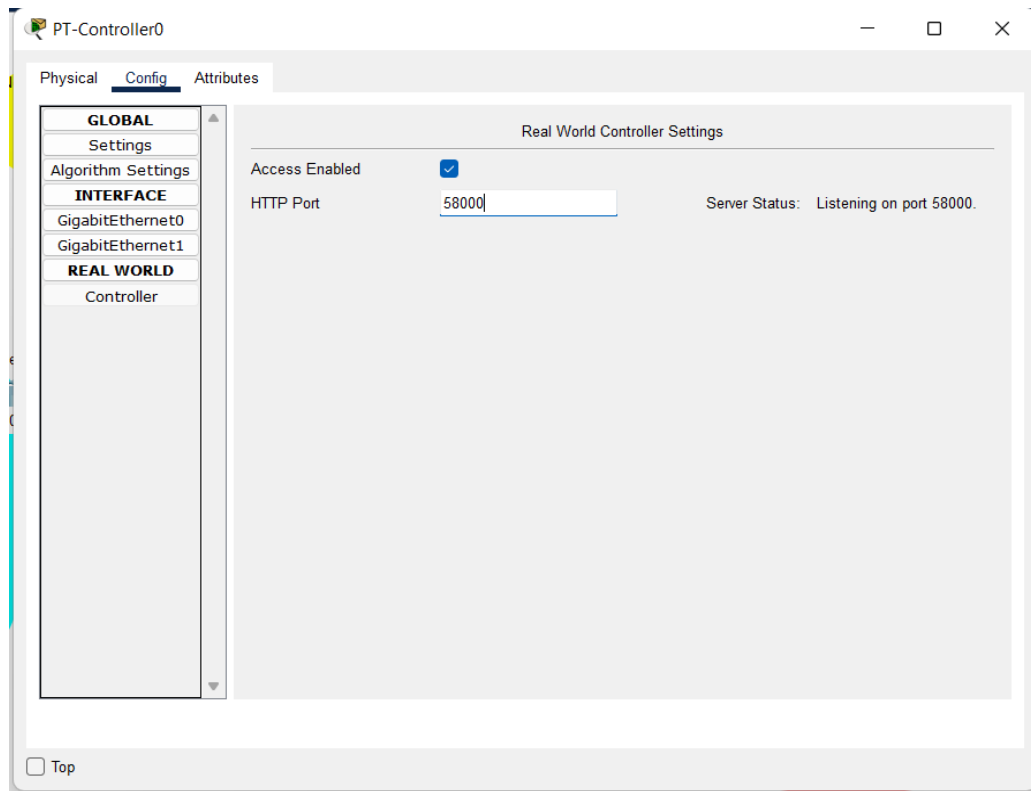
login local

username afrizal password cisco

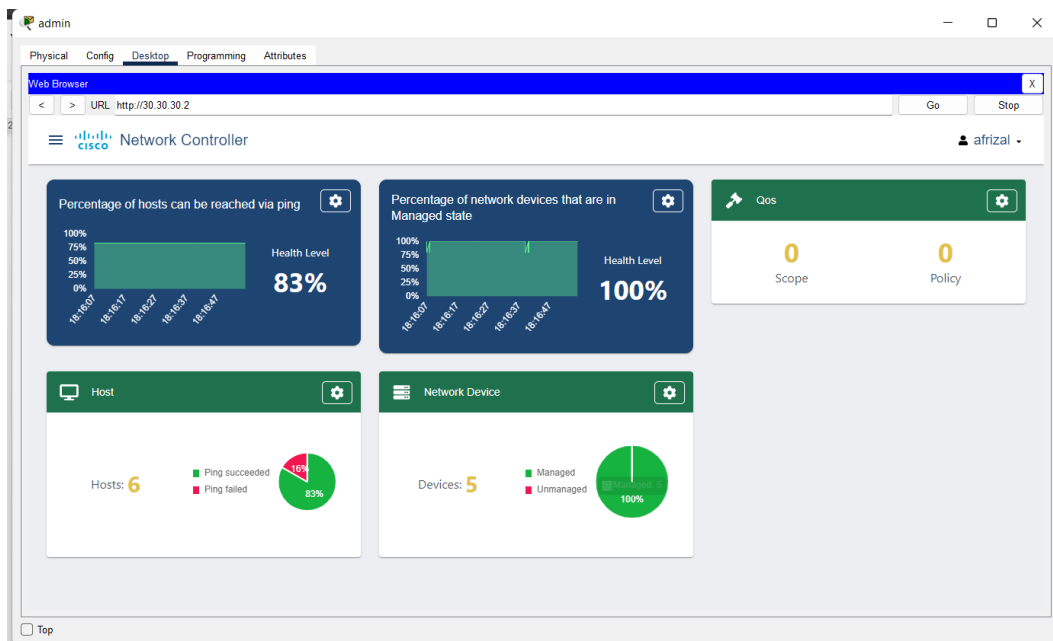
- Kode program yang dibuat mengimplementasikan fitur REST-API dari Network Controller.

API yang digunakan yakni REST-API dari network controller yang diakses melalui URL dari port 58000 dengan format "<http://{IP}:{PORT}/api/v1/ticket>" ada dua cara yakni menggunakan code python atau postman untuk mengakses API tersebut.

The screenshot shows the Cisco Network Controller REST API interface. On the left, there is a sidebar with navigation links: API SUMMARY, TICKETS (with sub-links addTicket and deleteTicket), and INVENTORY (with multiple sub-links). The main area displays a curl command in a dark box: `curl -X POST "http://{IP}:{PORT}/api/v1/ticket"`. Below this, the 'Parameters' section shows 'Body parameters' as a table with one entry: 'user'. The description for 'user' is a JSON object: `{ "Required: password,username", "password": "string", "password": "password", "username": "string", "username": "username" }`.



Dashboard Network Controller



Contoh menggunakan code Python untuk mengakses API dari network Controller

get-ticket.py

digunakan untuk mendapatkan token atau tiket untuk mengakses get-host dan get-network-device.

```
get-ticket.py X get-host.py get-network-device.py
get-ticket.py > ...
1  import json
2  import requests
3  api_url = "http://localhost:58000/api/v1/ticket"
4
5  headers = {
6      "content-type" : "application/json"
7  }
8
9  body_json = {
10     "username": "afrizal",
11     "password": "cisco"
12 }
13
14 resp = requests.post(api_url, json.dumps(body_json), headers=headers, verify=False)
15
16 print("Ticket request status: ", resp.status_code)
17 response_json = resp.json()
18
19 serviceTicket = response_json["response"]["serviceTicket"]
20 print("The service ticket number is: ", serviceTicket)
```

Contoh output program:

```
Requests.exceptions.InvalidURL: Invalid URL 'http://localhost:58000/api/v1/ticket':
PS E:\KOMINFO\api> python .\get-ticket.py
Ticket request status: 201
The service ticket number is: NC-4-1a3db6158fef445cbeeb-nbi
```

get-host.py

digunakan untuk mengakses data raw json yang berisi response dari request setiap host

```
get-ticket.py get-host.py X get-network-device.py
get-host.py > ...
1  import json
2  import requests
3  api_url = "http://localhost:58000/api/v1/host"
4
5  headers = {'content-type': 'application/json', 'X-Auth-Token': "NC-6-1d2046fb0b1b486cbbbb-nbi"}
6
7  resp = requests.get(api_url, headers=headers, verify=False, params="")
8
9  print("Request status: ", resp.json())
10
11 response_json = resp.json()
12 hosts = response_json["response"]
13
```

Contoh output program:

```
PS E:\KOMINFO\api> python .\get-host.py
Request status: {'response': [{'connectedAPMacAddress': '', 'connectedAPName': '', 'connectedInterfaceName': 'GigabitEthernet0/0/1', 'connectedNetworkDeviceIpAddress': '30.30.30.1', 'connectedNetworkDeviceName': 'R2', 'hostIp': '60.60.60.2', 'hostMac': '0001.9633.4452', 'hostName': 'Server0', 'hostType': 'Server', 'id': 'PTT0810W77-uuid', 'lastUpdated': '2022-07-04 12:10:42', 'pingStatus': 'SUCCESS'}, {'connectedAPMacAddress': '', 'connectedAPName': '', 'connectedInterfaceName': 'GigabitEthernet1/0/6', 'connectedNetworkDeviceIpAddress': '172.16.1.2', 'connectedNetworkDeviceName': 'SW1', 'hostIp': '172.16.1.7', 'hostMac': '0030.F2EE.374B', 'hostName': 'admin', 'hostType': 'Pc', 'id': 'PTT0810I12I-uuid', 'lastUpdated': '2022-07-04 12:10:42', 'pingStatus': 'SUCCESS'}, {'connectedInterfaceName': '', 'connectedNetworkDeviceIpAddress': '', 'connectedNetworkDeviceName': '', 'hostIp': '172.16.1.5', 'hostMac': '', 'hostType': '', 'id': 'PTT0810WA1U-uuid', 'lastUpdated': '2022-07-04 12:10:41', 'pingStatus': 'FAILURE'}, {'connectedAPMacAddress': '', 'connectedAPName': '', 'connectedInterfaceName': 'GigabitEthernet1/0/3', 'connectedNetworkDeviceIpAddress': '192.168.100.2', 'connectedNetworkDeviceName': 'SW2', 'hostIp': '192.168.100.4', 'hostMac': '0001.4303.B347', 'hostName': 'PC6', 'hostType': 'Pc', 'id': 'PTT081063K7-uuid', 'lastUpdated': '2022-07-04 12:10:43', 'pingStatus': 'SUCCESS'}, {'connectedAPMacAddress': '', 'connectedAPName': '', 'connectedInterfaceName': 'GigabitEthernet1/0/2', 'connectedNetworkDeviceIpAddress': '192.168.100.2', 'connectedNetworkDeviceName': 'SW2', 'hostIp': '192.168.100.3', 'hostMac': '00D0.FF32.1908', 'hostName': 'PC4', 'hostType': 'Pc', 'id': 'PTT0810A80U-uuid', 'lastUpdated': '2022-07-04 12:10:26', 'pingStatus': 'SUCCESS'}], 'version': '1.0'}
```

get-network-device.py

digunakan untuk mendapatkan informasi network device apa saja yang digunakan beserta alamat IP yang digunakan

```
get-ticket.py  get-host.py  get-network-device.py X
get-network-device.py > ...
1  import json
2  import requests
3  api_url = "http://localhost:58000/api/v1/network-device"
4
5  headers={"X-Auth-Token": "NC-6-1d2046fb0b1b486cbbbb-nbi"}
6
7  resp = requests.get(api_url, headers=headers, verify=False)
8
9  print("Request status: ", resp.status_code)
10
11  response_json = resp.json()
12  networkDevices = response_json["response"]
13
14  for networkDevice in networkDevices:
15      print(networkDevice["hostname"], "\t", networkDevice["platformId"], "\t", networkDevice["managementIpAddress"])
```

Contoh output program:

```
PS E:\KOMINFO\api> python .\get-network-device.py
Request status: 200
R1      ISR4300      172.16.1.1
R2      ISR4300      60.60.60.1
R3      ISR4300      10.10.10.2
SW1     3650      172.16.1.2
SW2     3650      192.168.100.2
```


Contoh menggunakan POSTMAN untuk mengakses API dari network Controller

POSTMAN (REST-API)

POST GET TIKET

kurang lebih sama dengan get-tiket.py, digunakan untuk mendapatkan token atau tiket untuk mengakses get-host dan get-network-device. Pilih POST pada form sebelah kiri URL, kemudian masukkan URL yang di copy yaitu <http://localhost:58000/api/v1/ticket> jangan lupa untuk memasukkan username dan password pada Body lalu pilih raw dengan format file JSON, agar bisa masuk autentikasi, kemudian tekan tombol SEND. Sehingga didapatkan service tiket dengan nomor sekian seperti pada contoh.

The screenshot displays the Postman application interface. At the top, there are tabs for different requests: 'POST GET TICKET' (selected), 'GET GET NETWORK', and 'GET GET HOST'. Below the tabs, the URL is set to 'http://localhost:58000/api/v1/ticket'. The 'Body' tab is selected, showing a JSON payload:

```
{  "username": "afrizal",  "password": "cisco"}
```

. The 'Headers' tab shows 9 headers. The 'Body' tab also shows a 'raw' format with a 'JSON' dropdown. The 'Send' button is visible. On the right side, there is a user profile card for 'Afrizal' with the email 'afrizalsyahruddinyusuf@gmail.com' and a 'View Profile' button. Below the profile card are links for 'Settings', 'Sign Out', 'Switch Accounts', and 'Add Account'. At the bottom, the 'Body' tab is selected, showing the response in 'Pretty' format:

```
{  "response": {    "idleTimeout": 900,    "serviceTicket": "NC-6-1d2046fb0b1b486cbbbb-nbi",    "sessionTimeout": 3600  },  "version": "1.0"}
```

. The status bar at the bottom right shows 'Status: 201 Created', 'Time: 4 ms', 'Size: 401 B', and a 'Sav' button.

GET GET HOST

kurang lebih sama dengan get-host.py, digunakan untuk mengakses data raw json yang berisi response dari request setiap host. Pilih POST pada form sebelah kiri URL, kemudian masukkan URL yang di copy yaitu <http://localhost:58000/api/v1/host> jangan lupa pada Headers lalu masukkan KEY X-Auth-Token dan isi VALUE dengan token/ticket yang ada dari GET TICKET. Sehingga didapatkan response dengan format JSON.

The screenshot displays a REST client interface with the following details:

- Request Method:** GET
- URL:** `http://localhost:58000/api/v1/host`
- Headers:** X-Auth-Token (checked), NC-6-1d2046fb0b1b486cbbbb-nbi
- Response Body (JSON):**

```
1  {
2    "response": [
3      {
4        "connectedAPMacAddress": "",
5        "connectedAPName": "",
6        "connectedInterfaceName": "GigabitEthernet0/0/1",
7        "connectedNetworkDeviceIpAddress": "30.30.30.1",
8        "connectedNetworkDeviceName": "R2",
9        "hostIp": "60.60.60.2",
10       "hostMac": "0001.9633.4452",
11       "hostName": "Seiver0",
12       "hostType": "Seiver",
13       "id": "PTT0810M977-uuid",
14       "lastUpdated": "2022-07-02 20:00:22",
15       "pingStatus": "SUCCESS"
16     },
17     {
```

On the right side, a user profile dropdown is visible for 'Afrizal' (afrizalsyahuluddinyusuf@gmail.com), with options for 'View Profile', 'Settings', 'Sign Out', 'Switch Accounts', and 'Add Account'.

GET GET NETWORK

kurang lebih sama dengan get-network-device.py, digunakan untuk mengakses data raw json yang berisi response dari request setiap host. Pilih POST pada form sebelah kiri URL, kemudian masukkan URL yang di copy yaitu <http://localhost:58000/api/v1/network-device> jangan lupa pada Headers lalu masukkan KEY X-Auth-Token dan isi VALUE dengan token/ticket yang ada dari GET TICKET. Sehingga didapatkan response dengan format JSON.

The screenshot displays a REST client interface with the following details:

- Request Method:** GET
- URL:** `http://localhost:58000/api/v1/network-device`
- Headers:** X-Auth-Token (checked), Value: `NC-6-1d2046fb0b1b486cbbbb-nbi`
- Response Body (JSON):**

```
1  {
2    "response": [
3      {
4        "collectionStatus": "Managed",
5        "connectedInterfaceName": [
6          "GigabitEthernet1/0/1",
7          "Serial0/2/0",
8          "Serial0/2/1"
9        ],
10       "connectedNetworkDeviceIpAddress": [
11         "172.16.1.2",
12         "166.1.1.2",
13         "100.2.2.2"
14       ],
15       "connectedNetworkDeviceName": [
16         "SW1",
17         "R2",
```

On the right side, a user profile dropdown menu is visible for 'Afrizal' (afrizalsyahruddinyusuf@gmail.com), with options like 'View Profile', 'Settings', 'Sign Out', 'Switch Accounts', and 'Add Account'.