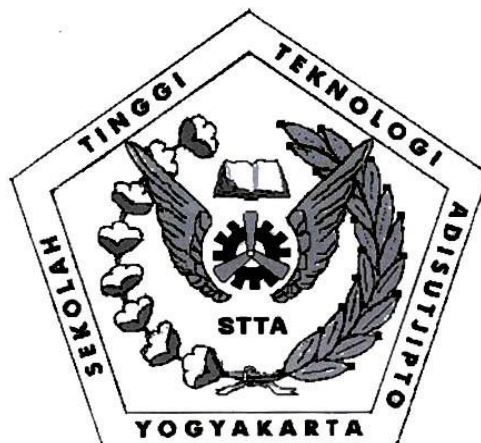


Kode Rumpun Ilmu: 458/ Teknik Informatika

Bidang Fokus : TIK

**LAPORAN  
PENELITIAN PEMULA**



***ROUTING PROTOCOL BERBASIS WEB PADA CISCO  
ROUTER 2800 SERIES MEMANFAATKAN APPLICATION  
PROGRAMMING INTERFACE (API) DARI CISCO***

Peneliti


**Sudaryanto, S.T., M.Eng. NIDN. 0511097901**  
**Ega Ari Wiyasa NIM. 16030017**

Penelitian Internal STTA  
Tahun Anggaran 2019/2020

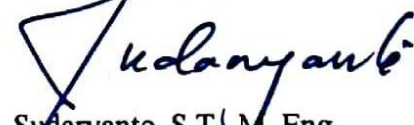
**SEKOLAH TINGGI TEKNOLOGI ADISUTJIPTO  
YOGYAKARTA  
2020**

## Halaman Pengesahan Penelitian Internal STTA

1. Judul Pengabdian : *Routing Protocol* Berbasis Web Pada Cisco Router 2800 Series Memanfaatkan *Application Programming Interface* (API) Dari Cisco
2. Bidang Penelitian : Informatika
3. Identitas Pengusul
  - a. Ketua Peneliti  
Nama : Sudaryanto, S.T., M. Eng.  
NIDN : 0511097901  
Jabatan Fungsional : Asisten Ahli  
Bidang Keahlian : Teknik Informatika
  - b. Asisten Peneliti  
Nama : Ega Ari Wiyasa  
NIM : 16030017
4. Lokasi Penelitian :
  - a. Tempat : Sekolah Tinggi Teknologi Adisutjipto
  - b. Kecamatan : Banguntapan
  - c. Kabupaten : Bantul
  - d. Propinsi : Daerah Istimewa Yogyakarta
5. Biaya dan Sumber Dana Penelitian : Rp. 1.500.000,-  
Sekolah Tinggi Teknologi Adisutjipto
6. Publikasi/Bukti Luaran : SENATIK (Seminar Nasional Teknologi Informasi dan Kedirgantaraan) STTA Yogyakarta

Mengetahui  
Waket I  
  
Dedet Hermawan, S., S.T., M.T.  
NIDN. 0521047001

Yogyakarta, November 2020  
Peneliti

  
Sudaryanto, S.T., M. Eng.  
NIDN. 0511097901

Mengetahui  
Ka.P3M  
  
Dr. Okto Dinaryanto, ST, M.M, M.Eng.  
NIDN. 0504107202

*ROUTING PROTOCOLS* BERBASIS WEB PADA CISCO *ROUTER 2800 SERIES* MEMANFAATKAN *APPLICATION PROGRAMMING INTERFACE (API)* DARI CISCO

Oleh

**Sudaryanto, S.T., M.Eng.**  
**0511097901**

**Ringkasan**

Pada saat ini *routing* merupakan teknologi yang banyak dibutuhkan. Hal ini dikarenakan *routing* pada sebuah *router* dapat menghubungkan jaringan yang berbeda. *Routing* dapat melakukan *management* jaringan yang akan terhubung dengan jaringan lainnya. Cisco *Router type 2800 series* merupakan *router* yang dapat dilakukan *static routing* ataupun *dynamic routing*. Akan tetapi permasalahan yang ditemukan pada Cisco *router type 2800 series* yaitu tidak ada aplikasi berbasis *Graphics User Interface (GUI)* untuk dapat melakukan konfigurasi pada *router*. Selama ini untuk dapat melakukan konfigurasi pada Cisco *router type 2800 series* yaitu dengan menggunakan *command line*. Untuk itu dibuat aplikasi berbasis web untuk mempermudah administrator dalam melakukan konfigurasi pada *router*. Aplikasi ini memanfaatkan API dari Cisco untuk menghubungkan antara aplikasi berbasis web dengan *router*. Dengan menggunakan aplikasi berbasis web, seorang administrator dapat melakukan *monitoring* dan manajemen jaringan secara cepat, tepat dan mampu meminimalisir kesalahan yang akan terjadi. Aplikasi berbasis web memiliki nilai efektivitas sebesar 85,4% lebih baik. Hal ini membuktikan bahwa aplikasi berbasis web dapat mempermudah seorang administrator dalam melakukan *routing*.

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	<b>Error! Bookmark not defined.</b>
RINGKASAN.....	<b>Error! Bookmark not defined.</b> iii
DAFTAR ISI.....	iiiv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	ix
DAFTAR LAMPIRAN.....	<b>Error! Bookmark not defined.</b>
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Metodologi Penelitian.....	3
BAB II. TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.1.1 Penelitian Yang Sudah Dilakukan Peneliti Lainnya.....	5
2.2 Landasan Teori.....	7
2.2.1 <i>Router</i> .....	7
2.2.2 <i>Application Programming Interface (API)</i> .....	8
2.2.3 <i>Routing Statis</i> .....	8
2.2.4 Nilai Efektivitas.....	9
BAB III. PERANCANGAN PERANGKAT LUNAK.....	10
3.1 Spesifikasi Kebutuhan <i>Hardware</i> dan <i>Software</i> .....	10
3.2 Perancangan.....	11
3.2.1 <i>Activity Diagram</i> .....	11
3.2.2 <i>Use Case Diagram</i> .....	13
3.2.3 <i>Interface Routing</i> Berbasis Web.....	14
BAB IV. IMPLEMENTASI DAN PEMBAHASAN.....	25
4.1 Hasil Perancangan.....	25

4.1.1	Implementasi <i>Form Login</i> .....	25
4.1.2	Implementasi <i>Form Home</i> .....	27
4.1.3	Implementasi <i>Form Interface</i> .....	28
4.1.4	Implementasi <i>Form RIP</i> .....	29
4.1.5	Implementasi <i>Form Static</i> .....	31
4.1.6	Implementasi <i>Form Report Menu</i> .....	32
4.1.7	Implementasi <i>Form Report Show IP</i> .....	33
4.1.8	Implementasi <i>Form Report Show Routing RIP</i> .....	34
4.1.9	Implementasi <i>Form Report Show Routing Static</i> .....	34
4.1.10	Implementasi <i>Form Delete Menu</i> .....	35
4.1.11	Implementasi <i>Form Delete IP</i> .....	36
4.1.12	Implementasi <i>Form Delete RIP</i> .....	37
4.1.13	Implementasi <i>Form Delete Static</i> .....	39
4.1.14	Implementasi Telnet .....	41
4.1.15	Implementasi API Cisco .....	42
4.1.16	Implementasi <i>Port Fowarding</i> .....	43
4.2	<i>Scheme</i> Pengujian .....	43
4.3	Pengujian .....	45
4.3.1	<i>Design Jaringan</i> .....	48
4.3.2	<i>Test Ping Tanpa Routing</i> .....	50
4.3.3	Pengujian <i>Dynamic Routing</i> .....	52
4.3.4	Pengujian <i>Static Routing</i> .....	56
4.3.5	Perbandingan Efesiensi GUI Dan <i>Command Line</i> .....	60
4.4	Pembahasan .....	66
BAB V. PENUTUP.....		71
5.1	Kesimpulan.....	71
5.2	Saran .....	71
DAFTAR PUSTAKA .....		72

## DAFTAR GAMBAR

Gambar 2. 1 <i>Router</i> .....	8
Gambar 3. 1 <i>Activity Diagram Konfigurasi</i> .....	12
Gambar 3. 2 <i>Activity Diagram Delete</i> .....	12
Gambar 3. 3 <i>Activity Diagram Show</i> .....	13
Gambar 3. 4 <i>Use Case</i> .....	14
Gambar 3. 5 <i>Form Login</i> .....	15
Gambar 3. 6 <i>Form Login Manual</i> .....	15
Gambar 3. 7 <i>Form Input Router</i> .....	16
Gambar 3. 8 <i>Form Home</i> .....	17
Gambar 3. 9 <i>Form Interface</i> .....	17
Gambar 3. 10 <i>Form RIP</i> .....	18
Gambar 3. 11 <i>Form Static</i> .....	19
Gambar 3. 12 <i>Form Report Menu</i> .....	19
Gambar 3. 13 <i>Form Report Show IP</i> .....	20
Gambar 3. 14 <i>Form Report Routing RIP</i> .....	20
Gambar 3. 15 <i>Form Report Static</i> .....	21
Gambar 3. 16 <i>Form Delete Menu</i> .....	22
Gambar 3. 17 <i>Form Delete IP</i> .....	22
Gambar 3. 18 <i>Form Delete RIP</i> .....	23
Gambar 3. 19 <i>Form Delete Static</i> .....	24
Gambar 4. 1 Implementasi <i>Form Login</i> .....	25
Gambar 4. 2 Implementasi <i>Alert Form Login</i> .....	26
Gambar 4. 3 Implementasi <i>Input Manual</i> .....	26
Gambar 4. 4 Implementasi <i>Konfigurasi Router</i> .....	27
Gambar 4. 5 Implementasi <i>Form Home</i> .....	27
Gambar 4. 6 Implementasi <i>Form Interface</i> .....	28
Gambar 4. 7 Implementasi <i>Alert Interface</i> .....	29
Gambar 4. 8 Implementasi <i>Confirm Static</i> .....	29
Gambar 4. 9 Implementasi <i>Form RIP</i> .....	30

Gambar 4. 10 Implementasi <i>Alert Form</i> RIP .....	30
Gambar 4. 11 Implementasi <i>Form Static</i> .....	31
Gambar 4. 12 Implementasi <i>Alert Form Static</i> .....	31
Gambar 4. 13 Implementasi <i>Form Alert Confirm Static</i> .....	32
Gambar 4. 14 Implementasi <i>Form Report Menu</i> .....	33
Gambar 4. 15 Implementasi <i>Form Report Show IP</i> .....	33
Gambar 4. 16 Implementasi <i>Form Report RIP</i> .....	34
Gambar 4. 17 Implementasi <i>Form Report Static</i> .....	35
Gambar 4. 18 Implementasi <i>Form Delete Menu</i> .....	35
Gambar 4. 19 Implementasi <i>Form Delete IP</i> .....	36
Gambar 4. 20 Implementasi <i>Alert Confirm Form Delete IP</i> .....	36
Gambar 4. 21 Implementasi <i>Alert Delete IP</i> .....	37
Gambar 4. 22 Implementasi <i>Form Delete RIP</i> .....	38
Gambar 4. 23 Implementasi <i>Alert Form Delete RIP</i> .....	38
Gambar 4. 24 Implementasi <i>Alert Confirm Form Delete RIP</i> .....	39
Gambar 4. 25 Implementasi <i>Form Delete Static</i> .....	40
Gambar 4. 26 Implementasi <i>Alert Form Delete Static</i> .....	40
Gambar 4. 27 Implementasi <i>Alert Confirm Delete Static</i> .....	41
Gambar 4. 28 Implementasi Telnet.....	42
Gambar 4. 29 Implementasi API Cisco.....	42
Gambar 4. 30 Implementasi Ngrok.....	43
Gambar 4. 31 <i>Scheme</i> Pengujian.....	44
Gambar 4. 32 Laboratorium Jaringan .....	45
Gambar 4. 33 <i>Router Cisco Type 2800 Series</i> .....	45
Gambar 4. 34 <i>Personal Computer</i> .....	46
Gambar 4. 35 <i>Switch Menuju Router-0</i> .....	47
Gambar 4. 36 <i>Switch Menuju Router-1</i> .....	47
Gambar 4. 37 <i>Switch Menuju Router-0 dan Router-1</i> .....	47
Gambar 4. 38 <i>Design Jaringan</i> .....	48
Gambar 4. 39 <i>Design Jaringan Dengan Internet</i> .....	49
Gambar 4. 40 Tes Ping Tanpa <i>Routing</i> .....	50

Gambar 4. 41 Tes Ping RIP PC-1 .....	52
Gambar 4. 42 Tes Ping RIP PC-2 .....	53
Gambar 4. 43 Tes Ping RIP PC-3 .....	53
Gambar 4. 44 Tes Ping RIP PC-4 .....	54
Gambar 4. 45 <i>Tracert</i> RIP.....	54
Gambar 4. 46 Tes Ping <i>Static</i> PC-1 .....	56
Gambar 4. 47 Tes Ping <i>Static</i> PC-2 .....	57
Gambar 4. 48 Tes Ping <i>Static</i> PC-3 .....	57
Gambar 4. 49 Tes Ping <i>Static</i> PC-4 .....	58
Gambar 4. 50 <i>Tracert Static Routing</i> .....	58
Gambar 4. 51 Hasil <i>Dynamic Routing</i> Router0 .....	66
Gambar 4. 52 Hasil <i>Dynamic Routing</i> Router1 .....	67
Gambar 4. 53 Hasil <i>Static Routing</i> Router0.....	67
Gambar 4. 54 Hasil <i>Static Routing</i> Router1 .....	68



## DAFTAR TABEL

Tabel 4. 1 IP Address .....	49
Tabel 4. 2 Tes Ping Tanpa <i>Routing</i> .....	51
Tabel 4. 3 Tes Ping RIP .....	55
Tabel 4. 4 Tes Ping <i>Static</i> .....	59
Tabel 4. 5 Pengujian Efektivitas <i>Input IP</i> .....	61
Tabel 4. 6 Pengujian Efektivitas <i>Dynamic Routing</i> .....	61
Tabel 4. 7 Pengujian Efektivitas <i>Static Routing</i> .....	62
Tabel 4. 8 Pengujian Efektivitas <i>Show IP</i> .....	62
Tabel 4. 9 Pengujian Efektivitas <i>Show Dynamic Routing</i> .....	63
Tabel 4. 10 Pengujian Efektivitas <i>Show Static Routing</i> .....	63
Tabel 4. 11 Pengujian Efektivitas <i>Delete IP</i> .....	64
Tabel 4. 12 Pengujian Efektivitas <i>Delete Dynamic Routing</i> .....	64
Tabel 4. 13 Pengujian Efektivitas <i>Delete Static Routing</i> .....	65
Tabel 4. 14 Pengujian Efektivitas <i>Save</i> .....	65
Tabel 4. 15 <i>Configuration</i> .....	68
Tabel 4. 16 Nilai Efektivitas .....	69

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini hampir seluruh kebutuhan manusia dapat didukung oleh perkembangan teknologi. Seiring dengan perkembangan jaman, teknologi berkembang dengan sangat pesat. Hal ini dapat dibuktikan dengan kehidupan masyarakat yang sudah mulai meninggalkan kegiatan secara konvensional. Salah satu yang berkembang pesat yaitu teknologi jaringan komputer. Berbagai perusahaan dan instansi lainnya sudah mulai menggunakan jaringan komputer guna mengikuti arus perkembangan teknologi yang tidak dapat dihindari. Salah satu penyebab pentingnya jaringan komputer adalah kebutuhan akan informasi yang begitu cepat.

Bagian jaringan komputer yang saat ini sangat diperlukan adalah *Routing Protocol*. Adanya teknik *Routing Protocol* ini digunakan untuk mem-*broadcast*, mempelajari jaringan yang saling terhubung dan rute (*network path*) yang tersedia. Serta *router* yang berbeda bisa saling bertukar informasi untuk mendapatkan rute paling efisien ke tujuan. Hal ini bertujuan diantaranya mengatasi *broadcast domain*, keamanan, peningkatan performa dan lain sebagainya. Pentingnya sebuah *Routing Protocol* dalam teknologi jaringan komputer tak terlepas dari manfaat yang telah dibahas di atas. Sehingga administrator jaringan diharuskan mampu melakukan konfigurasi *Routing Protocol* tersebut.

Miftah (2016) menjelaskan *router* dinamis yang dapat dikonfigurasi. *Router* dinamis mampu menentukan jalur yang terbaik dalam menentukan pengiriman terbaik untuk sampai tujuan. *Router* yang terhubung akan saling memberikan informasi dan secara bersama akan membuat tabel *routing* secara otomatis.

Cisco *Router* merupakan salah satu *router* yang diproduksi oleh Cisco. Konfigurasi *Routing Protocol* pada Cisco *Router* masih berbasis *Command Line*, sehingga menuntut administrator jaringan untuk memahami secara keseluruhan perintah-perintah dari konfigurasi *Routing Protocol*. Hal ini tentunya tidak mudah dan kurang begitu efisien. Untuk itu dibutuhkan suatu perangkat lunak berbasis

*Graphics User Interface* (GUI) yang di dalamnya terdapat perintah-perintah dari konfigurasi *Routing Protocol*. Perangkat lunak tersebut yang akan mempermudah administrator dalam melakukan konfigurasi *Routing Protocol*.

Pada penelitian Ayuningtyas et al (2020) yang berjudul “*Sistem Manajemen Virtual Local Area Network (VLAN) Pada Cisco Catalyst 3750 Berbasis Web*” telah berhasil dibuat perangkat lunak berbasis *Graphics User Interface* (GUI) yang mampu melakukan konfigurasi *Virtual Local Area Network* (VLAN), akan tetapi aplikasi berbasis *Graphics User Interface* (GUI) hanya dapat digunakan pada Cisco *Switch Catalyst 3750*. Keterbatasan penelitian tersebut dalam hal konfigurasi sebuah *device* menjadi dasar dilakukannya penelitian ini. Pada penelitian ini membahas tentang bagaimana membuat *routing protocol* berbasis web untuk melakukan konfigurasi *dynamic routing* dan *static routing* pada cisco *router 2800 series* dari berbagai tempat tanpa harus menyentuh *router*. Diharapkan dengan adanya penelitian ini dapat melengkapi kekurangan yang terdapat pada penelitian sebelumnya.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang masalah di atas dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara untuk memanfaatkan *Application Programming Interface* (API) Cisco dalam pembuatan *Routing Protocol* berbasis web.
2. Bagaimana cara untuk membuat *Routing Protocol* berbasis web yang dapat diakses dari jarak jauh (*Remote*) pada Cisco *Router*.
3. Apakah terdapat perbedaan konektivitas *router* yang dikonfigurasi menggunakan *command line* dan *Graphical User Interface* (GUI).
4. Berapa nilai efektifitas penggunaan *Graphical User Interface* (GUI) untuk melakukan konfigurasi Cisco *router type 2800 series* dibandingkan dengan menggunakan *command line*.

### 1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah disebutkan, maka didapatkan batasan masalah sebagai berikut:

1. *Routing Protocol* berbasis web dibuat dalam bentuk *Graphical User Interface (GUI)*.
2. *Routing Protocol* berbasis web yang dibuat hanya dapat mengkonfigurasi satu *router* yang sesuai dengan alamat IP yang ditentukan.
3. Aplikasi web dibuat untuk melakukan konfigurasi *dynamic routing*.
4. Aplikasi web dibuat untuk melakukan konfigurasi *static routing*.
5. Aplikasi berbasis web hanya dapat terhubung dengan *router* melalui telnet.

### 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Membuat *User Interface* untuk *Management* dan *Monitoring Routing Protocol* berbasis web pada *Cisco Router*.
2. Mampu melakukan *Management* dan *Monitoring* dari tempat yang berbeda, sehingga tidak perlu bersentuhan secara langsung dengan perangkatnya.

Berdasarkan masalah dan tujuan di atas, manfaat penelitian ini adalah:

1. Membantu administrator dalam melakukan *Management Routing Protocol* pada *Cisco Router* dari tempat yang berbeda tanpa harus bersentuhan langsung dengan perangkatnya.
2. Membantu administrator dalam melakukan *Monitoring Routing Protocol* pada *Cisco Router* dari tempat yang berbeda tanpa harus bersentuhan langsung dengan perangkatnya.

### 1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah:

1. Pengumpulan Data

#### A. Observasi

Pada metode ini dilakukan pengamatan secara langsung di laboratorium suatu instansi mengenai hal-hal yang berhubungan dengan manajemen *Routing Protocol*.

#### B. Wawancara

Data diperoleh dari beberapa proses, yaitu wawancara dengan pengurus (administrator) laboratorium serta wawancara dengan dosen yang berkompeten di bidangnya.

#### C. Studi Literatur

Bertujuan untuk mempelajari teori-teori dengan membaca beberapa buku dan jurnal yang berhubungan dengan permasalahan yang dibahas. Khususnya kajian mengenai *Routing Protocol*.

##### 2. Perancangan Sistem

Pada penelitian ini perancangan sistem dibuat menggunakan *Unified Modeling Language (UML)*.

##### 3. Implementasi dan Uji Coba

Pada tahapan metode ini dilakukan implementasi serta uji coba aplikasi yang dibuat. Implementasi dan uji coba dilakukan pada perangkat Cisco *Router*.

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

##### **2.1.1 Penelitian Yang Sudah Dilakukan Peneliti Lainnya**

Pada penelitian Sudaryanto (2018) telah mengetahui *switch manageable* mempunyai fungsi yang sama dengan *switch unmanageable*. Akan tetapi banyak fitur tambahan yang dapat membedakan *switch unmanageable* dalam meningkatkan kualitas dari jaringan tersebut. Sebagai contoh fitur yang sering digunakan yaitu konfigurasi *Virtual Local Area Network (VLAN)*, *traffic* jaringan yang bisa dikontrol, dapat melakukan proses *routing*, serta mampu digunakan untuk keamanan menggunakan *switch port security* yang berfungsi untuk menangani hak akses menuju jaringan tersebut. *Switch unmanageable* hanya bekerja pada *layer data link* atau *layer 2*, pada *switch* jenis ini tidak bisa melakukan proses konfigurasi. Peneliti telah berhasil mengimplementasikan *switch manageable* di Laboratorium Komputasi Sekolah Tinggi Teknologi Adisutjipto Yogyakarta untuk menangani sistem keamanan menggunakan *port security* dengan tujuan untuk mencari teknik yang tepat dalam sistem keamanan pada Laboratorium Komputasi Sekolah Tinggi Teknologi Adisutjipto Yogyakarta.

Penelitian Sudaryanto (2018) meneliti tentang perbedaan *switch* yang berjalan pada *layer 3* dan *layer 2*. Hal ini bertujuan untuk mengetahui pengaruh *switch multilayer* untuk kecepatan pengiriman pada suatu jaringan komputer.

Penelitian Sudaryanto & Nurhayati (2019) berhasil mengimplementasikan keamanan jaringan dengan menggunakan *port security* di Laboratorium Komputasi Sekolah Tinggi Teknologi Adisutjipto Yogyakarta. Dengan konfigurasi menggunakan *Graphics User Interface (GUI)* berbasis web yang mampu *monitoring* dengan notifikasi *email*.

Banyak penelitian yang membahas tentang *management* keamanan jaringan dengan menggunakan *port* yang tersedia pada *switch* seperti *default* atau *static port security*, *port security dynamic learning*, *sticky port security* (Sulaiman, 2016), *monitoring* jaringan dengan menggunakan *SNMP* (Rinaldo, 2016), dan

menggunakan web, di mana *monitoring* jaringan menggunakan piranti mikrotik dengan memanfaatkan *Application Programming Interface* (API) mikrotik. Belum ada penelitian yang *manage* dan *monitoring* jaringan dengan menggunakan piranti Cisco dengan memanfaatkan API Cisco. Maka dengan adanya penelitian ini untuk melakukan konfigurasi *static routing* dan *dynamic routing* pada Cisco *router type 2800 series* untuk mempelajari jaringan yang terhubung.

Pada penelitian Ayuningtyas et al (2020) konfigurasi *Virtual Local Area Network* (VLAN) pada Cisco *catalyst 3750* dengan memanfaatkan *Application Programming Interface* (API) dari Cisco. Penelitian ini *focus* terhadap Cisco *catalyst 3750* untuk *management* dan *monitoring Virtual Local Area Network* (VLAN).

Kendala yang dihadapi untuk *management* dan *monitoring* jaringan yaitu harus menghampiri *router* yang akan dikonfigurasi. Serta untuk *management* dan *monitoring router* harus memasukan perintah berbasis *command line*. Maka dari itu penelitian ini bertujuan untuk mengembangkan penelitian sebelumnya yang *focus* terhadap *switch*. Penelitian ini akan membuat *Graphics User Interface* (GUI) untuk dapat melakukan konfigurasi *static routing* dan *dynamic routing* pada Cisco *router type 2800 series*.

*Routing* merupakan proses memindahkan data dari satu *network* ke *network* lainnya dengan cara *forward* pada paket data melalui *gateway*. *Routing* yang akan menentukan kemana data akan dikirim agar dapat mencapai tujuan yang diinginkan. Ada beberapa informasi yang dibutuhkan *router* dalam melakukan *routing*:

1. *Destination*.
2. Sumber Informasi
3. Pemilihan Rute
4. Menjaga Informasi *Routing*.

Menurut (Jati et al, 2018) *Routing Information Protocol* (RIP) merupakan salah satu cara menghubungkan *network* yang berbeda. RIP menggunakan metode *Triggered Update*, yang memiliki perhitungan waktu untuk mengetahui kapan *router* harus kembali memberikan informasi *routing*. Cukup mudah untuk melakukan *routing* dengan menggunakan *Routing Information Protocol* (RIP).

Menurut (Sulaiman et al, 2018) *Router* yang terhubung menggunakan *Routing Information Protocol* (RIP) tidak akan mengalami *looping*, hal ini dikarenakan salah satu tautan akan digunakan tautan cadangan. Dengan menggunakan RIP jalur terbaik yang akan dilalui paket data dapat dioptimalkan dengan jaringan yang ada.

Menurut (Sirika & Mahajan, 2016) *Routing Information Protocol* (RIP) merupakan salah satu routing *intraarea* yang digunakan dalam otonom sistem. *Routing Information Protocol* memiliki beberapa sifat seperti:

1. Memiliki jarak vektor *routing protocol*.
2. Memiliki jumlah *hop maximum* sebanyak 15.

Melakukan pembaruan setiap 30 detik.

## **2.2 Landasan Teori**

### **2.2.1 Router**

*Router* memiliki bentuk seperti komputer mini yang digunakan untuk melakukan proses enkripsi. Perkembangan *router* terus berjalan hingga saat ini. *Router* sudah lebih modern dan memiliki fasilitas yang dapat digunakan dalam kepentingan sebuah jaringan. Selain itu kecepatan sebuah *router* saat ini sudah cukup memadai. Pada umumnya sebuah *router* menyediakan *port Fast Ethernet* yang memiliki kecepatan *maximum* 100 Mbps, namun ada beberapa *router* yang memang menyediakan *port Gigabit Ethernet* dengan kecepatan hingga 1 Gbps. *Router* merupakan sebuah *hardware* yang digunakan untuk mengirimkan paket data melalui sebuah jaringan. Hal ini biasa dikenal sebagai *routing*. *Router* dapat menghubungkan antar jaringan baik dalam skala kecil atau skala besar. Sementara itu *router* memiliki beberapa fasilitas yang menjadi unggulan, seperti *Dynamic Host Configuration Procotol* (DHCP). DHCP dapat membagikan *IP Address* secara otomatis ke setiap jaringan yang terhubung ke *router*. Selain DHCP masih banyak fasilitas lain yang dapat diatur oleh seorang administrator. Contoh *router* dapat dilihat pada Gambar 2.1.





Gambar 2. 1 Router

### 2.2.2 *Application Programming Interface (API)*

*Application Programming Interface* atau biasa disebut sebagai API merupakan perangkat lunak yang berfungsi untuk menghubungkan kedua aplikasi. Developer biasa menggunakan API dalam pembuatan sebuah aplikasi atau *website*.

Menurut (Ayuningtyas et al, 2020) terdapat tiga buah definisi mengenai *Application Programming Interface (API)*, antara lain:

1. API merupakan sekumpulan instruksi yang berada di dua entitas, yaitu entitas yang berjalan di sisi *Application Layer* dan yang berjalan di sisi sistem operasi.
2. API merupakan kumpulan dari beragam perintah, fungsi, dan protokol di dalam jaringan komputer, yang bekerja sama di dalam menjadikan aplikasi untuk dapat berkomunikasi dengan sistem operasi, perangkat keras komputer, dan komputer lainnya dalam jaringan komputer.
3. API merupakan kombinasi dari beragam perintah dan prosedur yang terurut, sehingga memudahkan di dalam pengembangan perangkat lunak.

### 2.2.3 *Routing Statis*

Menurut (Widodo, 2012) *Routing* statis merupakan suatu mekanisme *routing* yang dilakukan dengan cara manual dengan membuat tabel *routing* pada setiap

*router*. *Routing* statis biasa digunakan untuk jaringan skala kecil yang terdiri dari dua atau tiga *router*. *Router* statis harus dipelihara secara manual, hal ini disebabkan karena tabel statis tidak bisa melakukan pertukaran tabel *routing* secara dinamis. Perintah yang digunakan untuk melakukan konfigurasi statis yaitu *ip route [destination] [subnetmask] [nexthop]*. Sebagai contoh *ip route 10.10.10.0 255.255.255.0 11.11.11.2*.

#### 2.2.4 Nilai Efektivitas

Menurut (Irawan & Febriyanti, 2016) Efektivitas merupakan perhitungan seberapa besar tingkat kelekatan *output* yang dicapai dengan *output* yang diharapkan dari sejumlah *input*. Berdasarkan hal itu nilai efektivitas dapat dihitung dengan menggunakan rumus sebagai berikut:

$$Efektivitas = \frac{Rata - rata\_sebelum\_treatment}{Rata - rata\_setelah\_treatment} \times 100\%$$

Rumus nilai efektivitas digunakan untuk menghasilkan nilai akhir dalam bentuk persentase.

Rumus untuk mencari nilai efektivitas *command line* dengan *Graphics User Interface* (GUI) dengan menggunakan rumus sebagai berikut:

$$Efektivitas = \frac{Command\ line - Graphics\ User\ Interface}{Command\ line} \times 100\%$$

Keterangan:

1. *Command line* = merupakan nilai rata-rata dari lima percobaan pengujian.
2. *Graphics User Interface* = merupakan nilai rata-rata dari nilai percobaan pengujian.

Sedangkan untuk mencari nilai efektivitas total menggunakan rumus

$$Efektivitas_{total} = \frac{\sum Ne}{\sum P}. \text{ Keterangan:}$$

1.  $\sum Ne$  = Jumlah Nilai Efektivitas.

## **BAB III**

### **PERANCANGAN PERANGKAT LUNAK**

#### **3.1 Spesifikasi Kebutuhan *Hardware* dan *Software***

Dalam implementasi sistem *Routing Protocol* perlu sebuah *hardware* yang terdiri dari *router* dan perangkat komputer yang sudah dilengkapi dengan komponen jaringan didalamnya dan sebuah *software* yang dapat menunjang implementasi sistem keamanan di Laboratorium Komputasi STTA.

##### 1. Spesifikasi *Hardware*

Perangkat keras (*Hardware*) merupakan perangkat secara fisik yang dapat dilihat dan dipegang. Sistem perangkat keras secara fungsional terdiri dari *input*, *process*, *output* dan *memory*. Adapun spesifikasi *hardware* yang digunakan dalam pengaplikasian sistem ini, yaitu:

##### A. Cisco Router 2800

- a. *Port Console*
- b. *Port Aux*
- c. *Port USB*
- d. *Compact Flash*
- e. *2 Port Fast Ethernet.*

##### B. Komputer

- a. *Processor AMD A6-5200 2.00 GHz.*
- b. *Random Access Memory 6 GB.*
- c. *Hardisk 500 GB.*
- d. *Radeon™ HD Graphics 8400.*
- e. *Monitor 14”.*

## 2. Spesifikasi *Software*

Perangkat lunak (*Software*) merupakan perangkat yang sifatnya abstrak yang berisi instruksi, program, prosedur, pengendali, pendukung dan aktivitas- aktivitas pengolahan perintah pada sistem komputer. Adapun spesifikasi minimum *software* yang dibutuhkan dalam pengaplikasian sistem ini, yaitu:

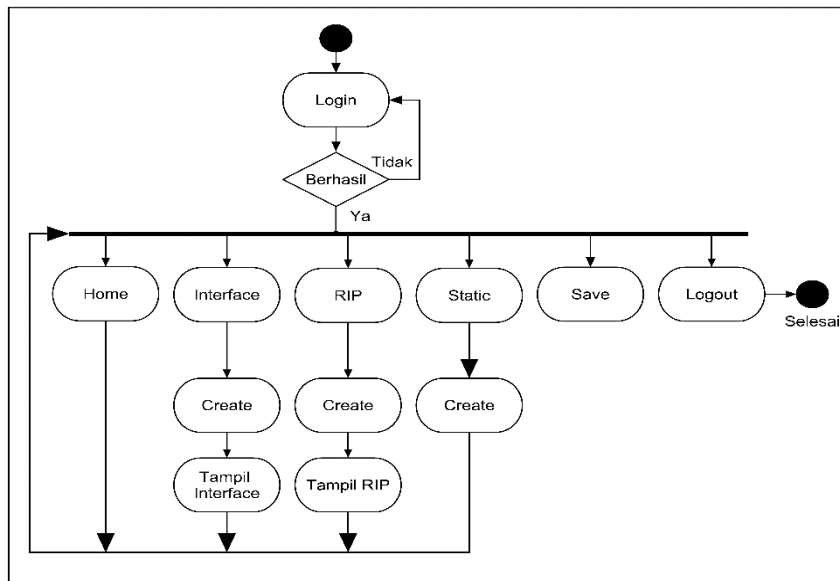
- A. Cisco *Router* 2800
  - a. Ios.
  - b. Cisco.
- B. Komputer
  - a. *Operating System Windows 7 Ultimate* 64-bit (6.1, build 7601).
  - b. *Sublime Text* 3
  - c. *XAMPP* v3.2.4
  - d. *Google Chrome* v79.0.3945.130 64-bit

## 3.2 Perancangan

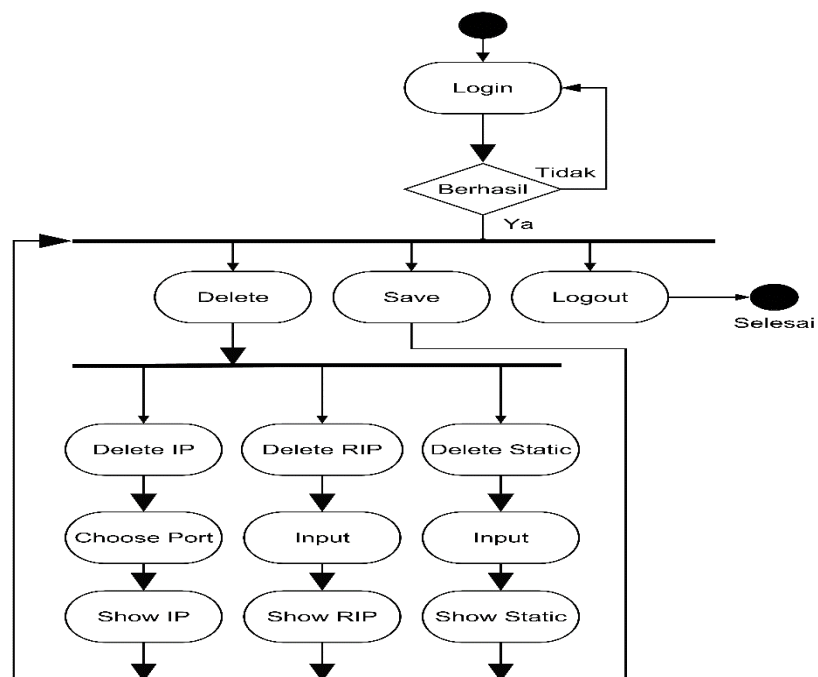
### 3.2.1 *Activity Diagram*

Pada Gambar 3.1 sampai Gambar 3.3 menjelaskan alur dari dalam sistem. Gambar 3.1 menjelaskan setelah *user* berhasil melakukan *login*, maka *user* dapat menggunakan menu konfigurasi seperti *interface*, *RIP*, *Static*, *Save* dan *Logout*. Gambar 3.2 menjelaskan setelah *user* berhasil *login* maka *user* dapat menggunakan menu *delete*, *save* dan *logout*. Gambar 3.3 menjelaskan setelah *user* berhasil *login* maka *user* dapat menggunakan menu *report*, *save* dan *logout* pada aplikasi. Pada menu *interface* berfungsi untuk memberikan IP pada *port router*. Pada Bagian *RIP* dapat digunakan untuk melakukan konfigurasi *RIP* pada sebuah *router*. Pada bagian *static* digunakan untuk memberikan *static routing* pada sebuah *router*. Pada bagian *Report* terdapat submenu yang berisi *Show IP*, *Show Routing RIP*, *Show Routing Static* yang berfungsi untuk menampilkan IP pada *port* dan juga hasil *routing*. Sedangkan pada menu *delete* terdapat submenu yang terdiri dari *Delete IP*, *Delete RIP* dan *Delete Static* yang berfungsi untuk menghapus IP dan konfigurasi. Pada menu *save* berfungsi untuk menyimpan konfigurasi yang sudah dilakukan sebelumnya, hal ini bertujuan agar konfigurasi tidak hilang pada saat *router*

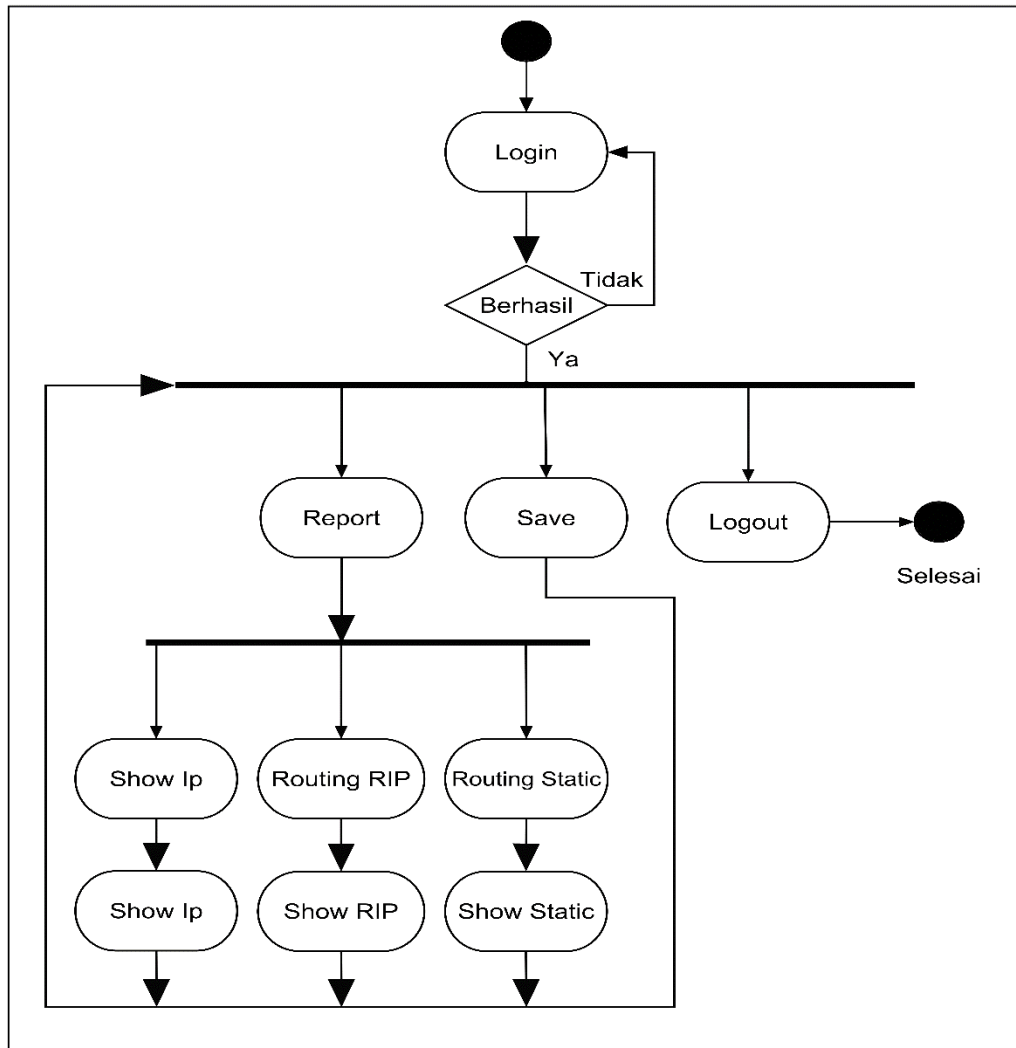
melakukan *restart*. Sedangkan pada menu *logout* berfungsi untuk keluar dari aplikasi dengan menghapus *username* dan *password* yang sudah pernah dilakukan sebelumnya. Hal ini bertujuan agar pengguna lain tidak dapat menggunakan aplikasi sebelum melakukan *login*.



Gambar 3. 1 Activity Diagram Konfigurasi



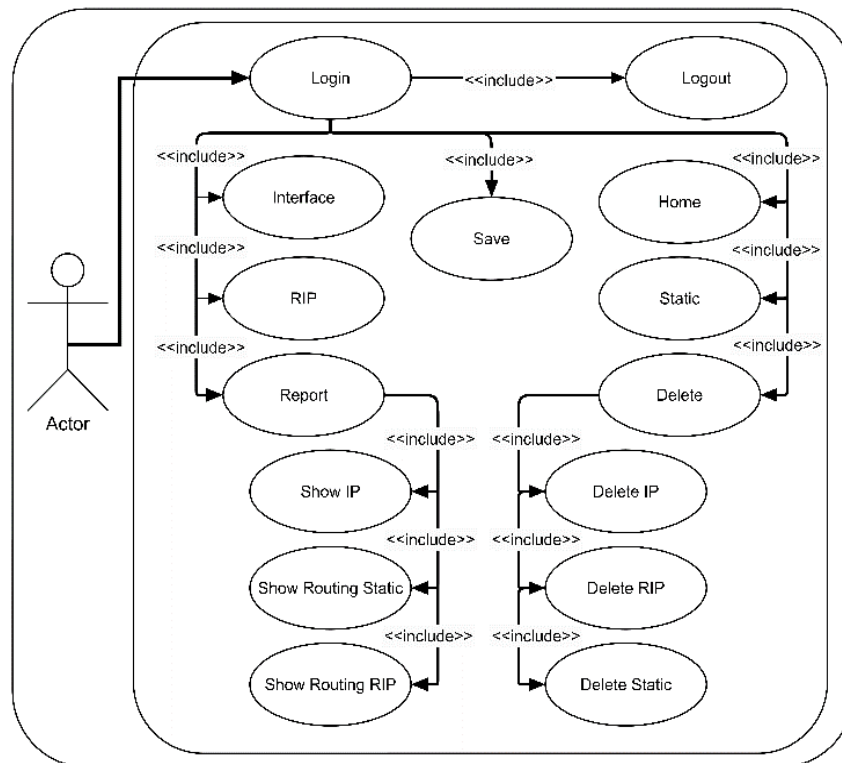
Gambar 3. 2 Activity Diagram Delete



Gambar 3. 3 Activity Diagram Show

### 3.2.2 Use Case Diagram

Pada *use case diagram*, Aktor dapat menggunakan aplikasi setelah melakukan *login*, aktor dapat menggunakan fasilitas seperti *input IP* pada *form interface*, melakukan *routing RIP*, melakukan *routing static*, menampilkan hasil *routing*, menghapus *routing*, *save* dan juga *logout* untuk keluar dari aplikasi. Gambar *Use case* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Use Case

### 3.2.3 Interface Routing Berbasis Web

*Interface* atau perancangan antarmuka berfungsi sebagai gambaran untuk membuat aplikasi yang bertujuan untuk mempermudah dalam *development* atau pembangunan aplikasi. Berikut tampilan rancangan *interface*:

#### A. Form Login

*Form login* merupakan tampilan awal ketika *user* akan menggunakan aplikasi. Pada *form login* terdapat beberapa kolom *input* seperti kolom *input username*, *password* dan juga *select button* yang berisi beberapa pilihan *router*. Setelah *user* mengisi *form login* maka *user* harus menekan tombol *login*, apabila berhasil akan diarahkan kepada *form home* dan apabila *username* atau *password* salah maka akan kembali ke halaman *login*. Terdapat *link input* manual apabila *router* yang dituju tidak ada pada *select button*, hal ini untuk mempermudah *user* dalam melakukan *routing*. Sehingga aplikasi dapat digunakan pada *router* dengan *host* yang berbeda-beda. *Form login* dapat dilihat pada Gambar 3.5.

The diagram shows a login form layout. At the top center is a box labeled 'Gambar'. Below it is a larger container with a 'LOGIN' button. Underneath the button are three input fields: 'Username', 'Password', and 'Choose Router' with a dropdown arrow. Below these is another 'LOGIN' button with the text 'Input Manual' underneath it. At the bottom of the page is a 'FOOTER' section.

Gambar 3. 5 *Form Login*

*Link Input* manual yang telah disediakan akan mengarahkan *user* pada *form login* yang berisi dua kolom *input* yaitu kolom *input username* dan juga *password*. *User* harus memasukan *username* serta *password* agar bisa menuju ke *form* berikutnya. Apabila *username* atau *password* salah, maka akan diarahkan kepada *form login* utama. *Form login input* manual dapat dilihat pada Gambar 3.6.

The diagram shows a manual login form layout. At the top center is a box labeled 'Gambar'. Below it is a larger container with a 'LOGIN' button. Underneath the button are two input fields: 'Username' and 'Password'. Below these is another 'LOGIN' button with the text 'Kembali' underneath it. At the bottom of the page is a 'FOOTER' section.

Gambar 3. 6 *Form Login Manual*



Setelah melakukan *login* pada *form login* manual, *user* akan diarahkan pada *form* berikutnya. Pada *form* ini terdapat beberapa kolom input seperti *hostname*, *username password*, *password router*, *password enable router*, dan *port router*. Kolom *input* ini yang akan menghubungkan aplikasi dengan *router* sesuai dengan alamat yang dimiliki *router*. Sehingga dapat membuat aplikasi menjadi dinamis dan mudah untuk digunakan. Apabila kolom *input* sudah di isi sesuai dengan keinginan maka *user* akan diarahkan pada *form home*. *Form input router* dapat dilihat pada Gambar 3.7.

The diagram illustrates the layout of the Router Input Form. It consists of a main container with a 'Gambar' label at the top. Inside this container is a vertical stack of input fields: 'Hostname', 'Username Router', 'Password Router', 'Password Enable Router', and 'Port Router'. Below these fields is a 'NEXT' button. At the bottom of the main container is a 'FOOTER' section.

Gambar 3. 7 *Form Input Router*

#### B. *Form Home*

*Form Home* merupakan tampilan pertama pada saat *user* berhasil melakukan *login*. Pada *form home* ini terdapat penjelasan singkat yang dapat digunakan *user* sebagai panduan dalam menggunakan aplikasi. Tampilan *Form Home* dapat dilihat pada Gambar 3.8.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
<b>CONFIGURATION ROUTING</b>									
Gambar									
Penjelasan Fungsi-fungsi Pada Aplikasi									
FOOTER									

Gambar 3. 8 Form Home

### C. Form Interface

*Form Interface* digunakan untuk memasukan IP address pada port router. Terdapat beberapa kolom input pada *form interface* yaitu *select button* untuk memilih *port* yang akan diberikan IP address, kolom *input IP* dan kolom *input subnet mask*. Setelah *user* mengisi kolom *input*, *user* dapat melihat hasil *execution* pada bagian *input execution*. Apabila *execution* berhasil maka *form* akan menampilkan hasil sesuai *port* dan inputan yang telah diisi sebelumnya. Gambar *Form interface* dapat dilihat pada Gambar 3.9.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
<b>INTERFACE</b>									
Choose Port ▼									
Input IP									
Input Subnet Mask									
Add Reset									
<b>INPUT EXECUTION</b>									
Show Interface									
FOOTER									

Gambar 3. 9 Form Interface

#### D. Form RIP

Pada *form* RIP terdapat satu kolom input yang digunakan untuk memasukan *network*. *User* hanya perlu memasukan *network* untuk melakukan *Routing Information Protocol* (RIP). Setelah itu *user* dapat melihat hasil *execution* pada bagian *input execution*. Apabila berhasil, *network* yang diinputkan akan tampil pada *form* tersebut. Gambar *form* RIP dapat dilihat pada Gambar 3.10.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-bottom: 10px;"> <b>INPUT NETWORK</b> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <input style="width: 90%; border: none;" type="text" value="Masukan Network"/> </div> <div style="text-align: center; margin-bottom: 10px;"> <input type="button" value="Add"/> <input type="button" value="Reset"/> </div> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-bottom: 10px;"> <b>INPUT EXECUTION</b> </div> <div style="text-align: center; margin-bottom: 10px;"> <input type="button" value="Show IP"/> </div> </div>									
FOOTER									

Gambar 3. 10 *Form* RIP

#### E. Form Static

Pada *Form static* terdapat beberapa kolom *input* seperti kolom *input destination*/kolom *input IP address*, *subnet mask* dan juga *next hop*. *User* hanya perlu mengisi kolom *input* dan menekan tombol *add* untuk melakukan *static routing*. Gambar *form static* dapat dilihat pada Gambar 3.11.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
------	----------------	------	-----------	-----	--------	----------	----------	------	--------

**STATIC ROUTES**

FOOTER

Gambar 3. 11 *Form Static*

#### F. *Form Report Menu*

*Form report* menu berfungsi sebagai kumpulan report yang ada pada aplikasi. *User* dapat menggunakan *form* ini untuk mempermudah dalam memilih *report* yang diperlukan. Gambar *form report* menu dapat dilihat pada Gambar 3.12.

LOGO	Routing	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
------	---------	------	-----------	-----	--------	----------	----------	------	--------

DELETE MENU

ROUTING STATIC

IP-ADDRESS

ROUTING DYNAMIC

~~PILIH REPORT~~  
 SHOW IP  
 SHOW ROUTING RIP  
 SHOW ROUTING STATIC

FOOTER

Gambar 3. 12 *Form Report Menu*

### G. Form Report Show IP

*Form report show IP* digunakan untuk menampilkan *IP address* pada *port router*. Sehingga *user* dapat dengan mudah melihat *IP address* yang sudah diberikan pada *port router*. Gambar *form report show IP* dapat dilihat pada Gambar 3.13.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT		
<table border="1" style="margin: auto;"> <tr> <td style="text-align: center;"><b>IP ROUTE</b></td> </tr> <tr> <td style="text-align: center; height: 100px;">Output</td> </tr> </table>										<b>IP ROUTE</b>	Output
<b>IP ROUTE</b>											
Output											
FOOTER											

Gambar 3. 13 *Form Report Show IP*

### H. Form Report Routing RIP

*Form report routing RIP* digunakan untuk menampilkan koneksi RIP yang sudah terhubung. Gambar *form report routing RIP* dapat dilihat pada Gambar 3.14.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT		
<table border="1" style="margin: auto;"> <tr> <td style="text-align: center;"><b>CONFIGURATION DYNAMIC</b></td> </tr> <tr> <td style="text-align: center; height: 100px;">Output</td> </tr> </table>										<b>CONFIGURATION DYNAMIC</b>	Output
<b>CONFIGURATION DYNAMIC</b>											
Output											
FOOTER											

Gambar 3. 14 *Form Report Routing RIP*

### I. Form Report Routing Static

*Form report routing static* berfungsi untuk menampilkan hasil *routing static*. *User* hanya perlu menjalankan *form* ini dengan menekan *button report static* yang sudah disediakan pada bagian *header*. Maka secara otomatis *form* akan *execution* dengan sendirinya. Gambar *form report routing static* dapat dilihat pada Gambar 3.15.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"><div style="border: 1px solid black; padding: 5px; text-align: center; margin-bottom: 10px;">CONFIGURATION STATIC</div><div style="border: 1px solid black; height: 200px; margin: 10px auto; width: 90%; text-align: center; vertical-align: middle;">Output</div></div>									
FOOTER									

Gambar 3. 15 *Form Report Static*

### J. Form Delete Menu

Pada *form delete* menu, terdapat beberapa *button* yang dapat digunakan untuk mengarahkan *user* pada *form delete* yang akan digunakan. Gambar *form delete* menu dapat dilihat pada Gambar 3.16.

LOGO	Routing	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: 80%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">DELETE MENU</div> <div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <span>DELETE IP</span> <span>DELETE RIP</span> <span>DELETE STATIC</span> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; text-align: center;">DELETE EXECUTION</div> </div> <div style="border: 1px solid black; padding: 5px; margin-left: 100px; width: 100px;">       ~~PILIH MENU~~        DELETE IP        DELETE RIP        DELETE STATIC     </div>									
FOOTER									

Gambar 3. 16 Form Delete Menu

#### K. Form Delete IP

Form delete IP digunakan untuk menghapus IP address pada port tertentu. Terdapat satu kolom *input* pada form ini, yang berfungsi untuk menentukan port yang akan dihapus. User hanya perlu menentukan port mana yang akan dipilih, setelah itu user harus menekan tombol *submit* untuk *execution*. Apabila *execution* berhasil, maka output akan ditampilkan pada bagian *delete execution*. Gambar form delete IP dapat dilihat pada Gambar 3.17.

Logo	Routing	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: 80%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">DELETE IP</div> <div style="margin-bottom: 10px;"> <input type="text" value="Choose Port"/> </div> <div style="display: flex; justify-content: center; margin-bottom: 10px;"> <span>Add</span> <span>Reset</span> <span>Back</span> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; text-align: center;">DELETE EXECUTION</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; text-align: center;">Output</div> </div>									
FOOTER									

Gambar 3. 17 Form Delete IP

### L. Form Delete RIP

*Form delete Routing Information Protocol (RIP)* digunakan untuk menghapus konfigurasi RIP yang ada pada *router*. *Form delete* digunakan apabila *user* memerlukan fungsi untuk menghapus konfigurasi RIP dan tidak menggunakannya lagi. *Replace* tidak akan terjadi pada konfigurasi RIP, oleh karena itu *form delete* sangat penting untuk *management* sebuah jaringan pada Cisco *type 2800 series*. Hanya ada satu kolom *input* yang dapat digunakan untuk menghapus konfigurasi. *User* hanya perlu memasukan *network*, setelah itu *user* harus menekan tombol *submit* untuk menjalankan perintah *delete* pada *router*. Apabila *execution* berhasil, maka *output* akan ditampilkan pada bagian *delete execution*. Gambar *form delete* RIP dapat dilihat pada Gambar 3.18.

Logo	<b>Routing</b>	HOME	INTERFACE	RIP	STATIC	REPORT ▼	DELETE ▼	SAVE	LOGOUT
------	----------------	------	-----------	-----	--------	----------	----------	------	--------

**DELETE NETWORK**

**DELETE EXECUTION**

Output

FOOTER

Gambar 3. 18 *Form Delete* RIP



### M. *Form Delete Static*

*Form delete static* digunakan untuk menghapus konfigurasi *static routing* pada sebuah *router*. Pada *form* ini terdapat beberapa kolom *input* yang dapat digunakan seperti kolom *input destination*, kolom *input subnet mask* dan kolom *input next hop*. *User* hanya perlu mengisi kolom *input* dan menekan tombol *submit* untuk menjalankan perintah *delete* pada *router*. Gambar *form delete static* dapat dilihat pada Gambar 3.19.

The image shows a web application interface for deleting static routes. At the top, there is a navigation menu with the following items: Logo, Routing (highlighted), HOME, INTERFACE, RIP, STATIC, REPORT (with a dropdown arrow), DELETE (with a dropdown arrow), SAVE, and LOGOUT. The main content area is titled "DELETE STATIC ROUTES" and contains three input fields: "Destination", "Subnet Mask", and "Next Hop". Below these fields are three buttons: "Add", "Reset", and "Back". At the bottom of the page, there is a "FOOTER" section.

Gambar 3. 19 *Form Delete Static*

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

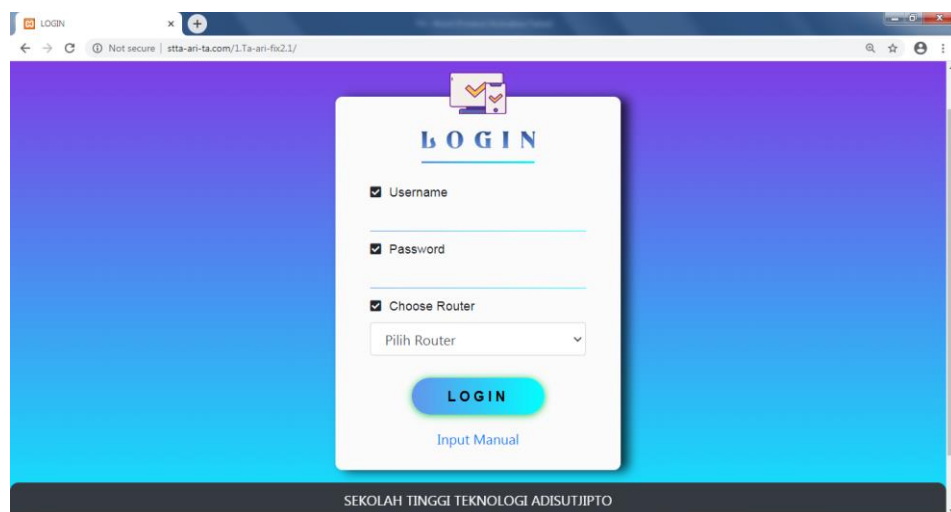
#### 4.1 Hasil Perancangan

Pada subbab ini akan dijelaskan mengenai perancangan ataupun implementasi yang sudah dijelaskan pada Bab IV. Implementasi ini berupa screenshot aplikasi yang akan digunakan untuk melakukan *routing static* ataupun *dynamic*.

##### 4.1.1 Implementasi *Form Login*

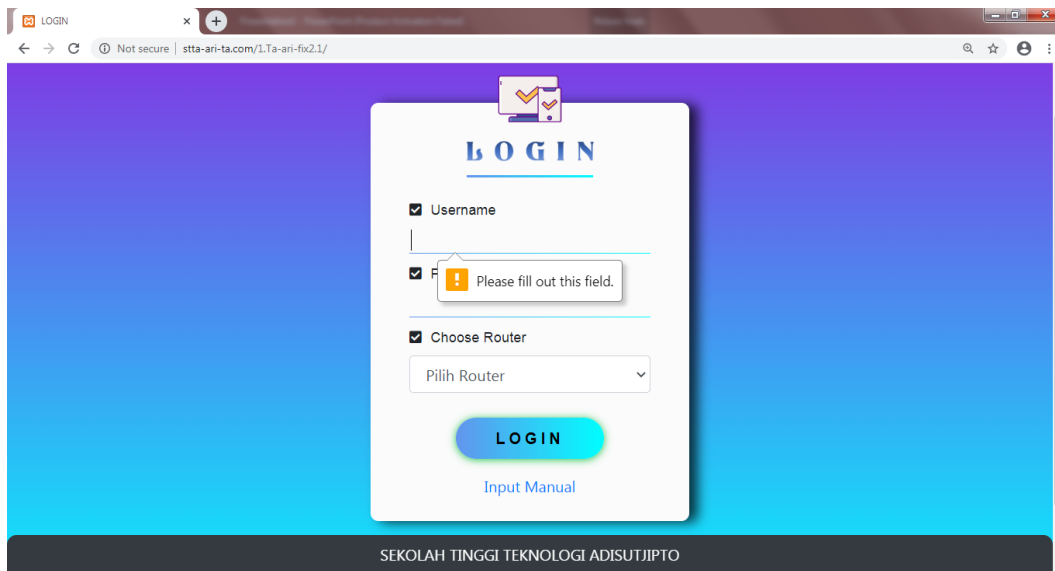
Implementasi pada *form login* ini dilakukan dengan mengisi kolom *input* yang sudah disediakan yaitu kolom *input username*, *password* dan juga *select button* untuk memilih *router* mana yang akan digunakan. Ada beberapa hal yang harus diperhatikan oleh *user* pada saat *login*:

1. Memasukan Kolom *input username* dengan “admin”, kolom *input Password* dengan “stta” dan memilih *select button*. Apabila kolom *input username*, *password* dan *select button* tidak diisi ataupun tidak sesuai dengan kriteria, maka akan kembali ke halaman *login*. Dan apabila berhasil akan menuju ke halaman *home*. Gambar implementasi form login dapat dilihat pada Gambar 4.1.



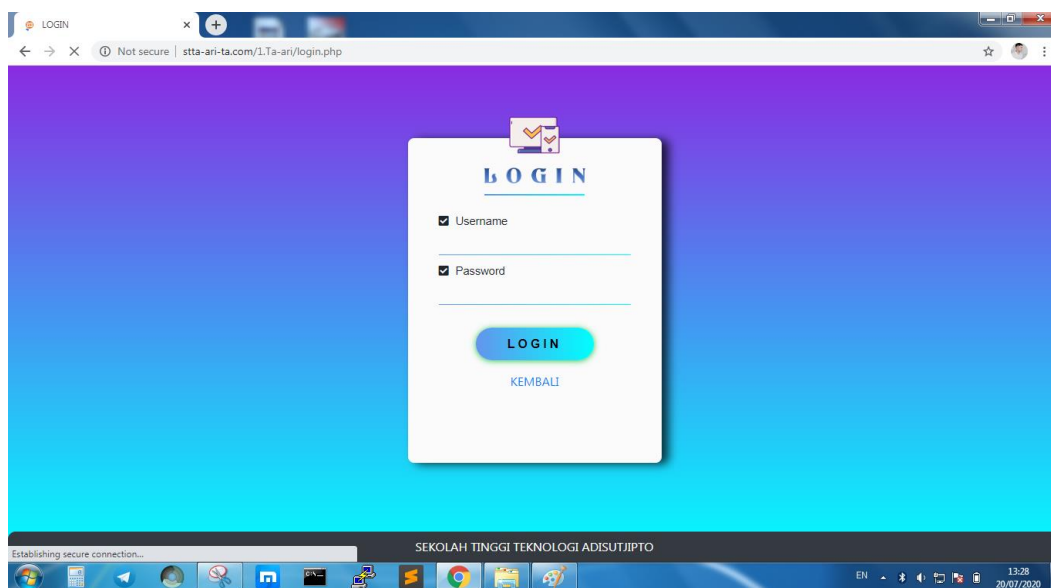
Gambar 4. 1 Implementasi *Form Login*

2. Apabila kolom *input Username, Password* dan *select button* tidak diisi maka akan tampil *alert* untuk memberitahu *user* bahwa kolom *input* harus diisi. Gambar *alert login* dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Implementasi *Alert Form Login*

3. Apabila tidak ada pilihan yang sesuai dengan konfigurasi *router* pada *select button*, *user* dapat menggunakan *input* manual untuk dapat menggunakan aplikasi. Gambar *input* manual dapat dilihat pada Gambar 4.3.



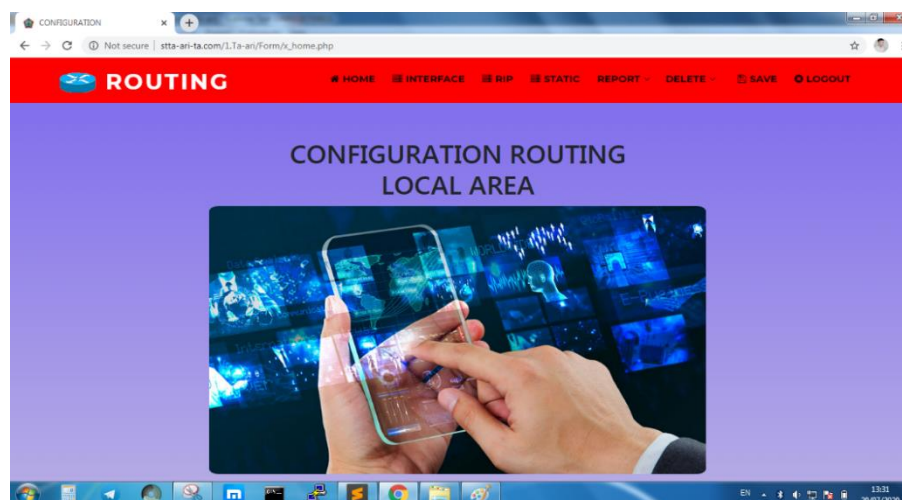
Gambar 4. 3 Implementasi *Input Manual*

4. Apabila *user* melakukan *login* melalui “input manual” maka *user* harus mengisi konfigurasi secara manual terlebih dahulu. Hal ini dapat memudahkan *user* dalam melakukan koneksi dengan *router*. Gambar konfigurasi *router* dapat dilihat pada Gambar 4.4.

Gambar 4. 4 Implementasi Konfigurasi *Router*

#### 4.1.2 Implementasi *Form Home*

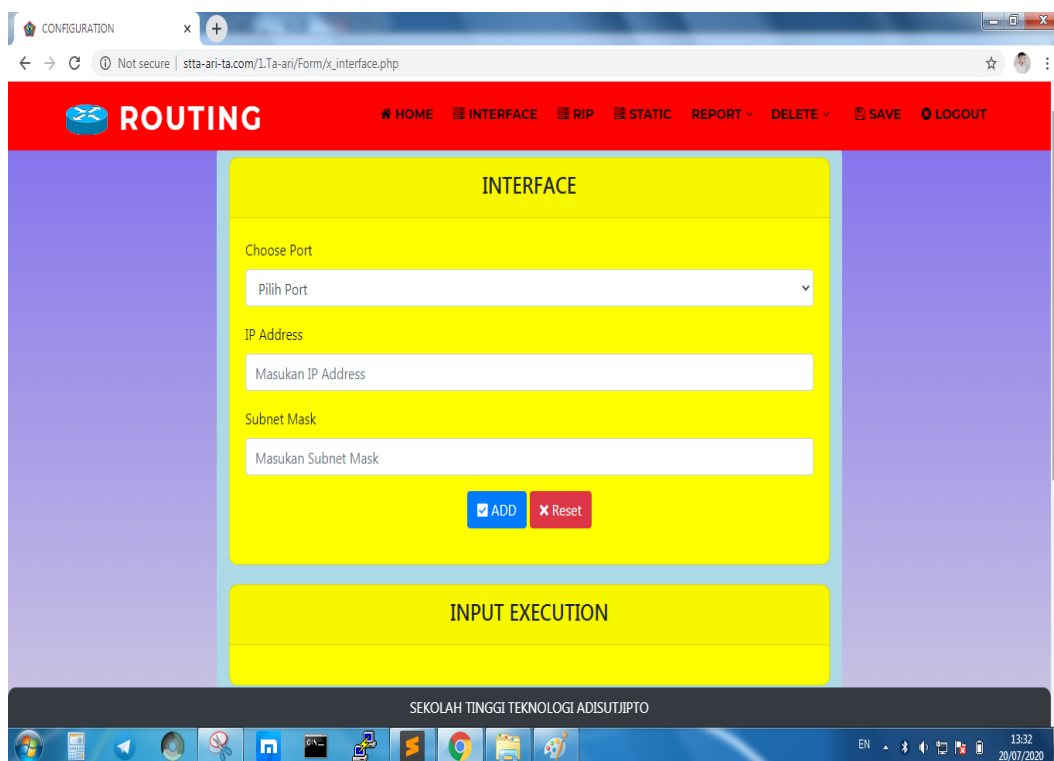
*Form home* merupakan tampilan awal ketika *user* berhasil melakukan *login*. Pada *Form* ini menampilkan beberapa panduan tentang aplikasi secara singkat. Hal ini dapat mempermudah *user* dalam menggunakan aplikasi. *Form Home* dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Implementasi *Form Home*

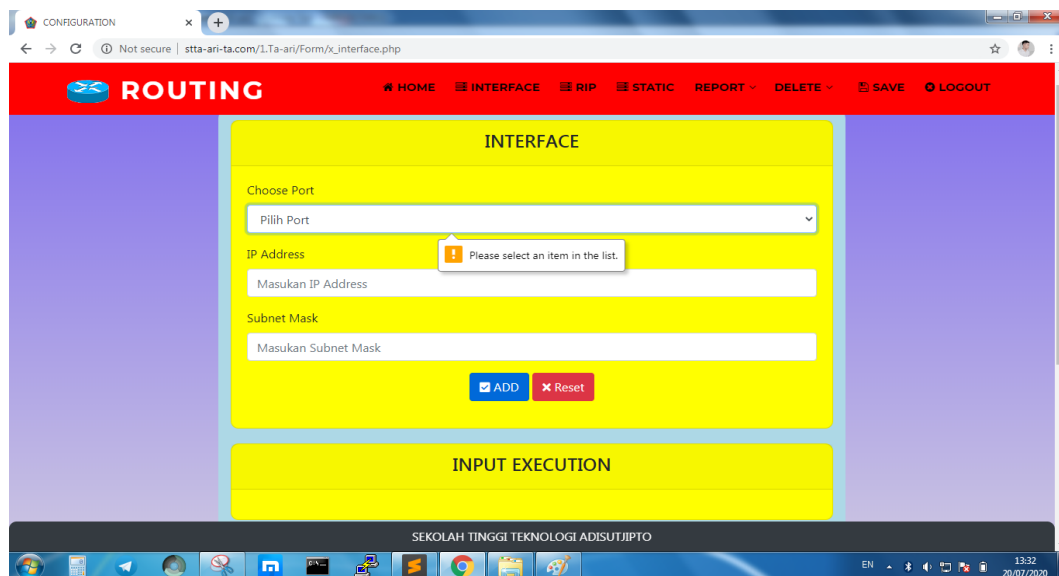
### 4.1.3 Implementasi *Form Interface*

Pada *form interface* terdapat tiga kolom *input* yang dapat diisi oleh *user*. *Form interface* digunakan untuk menambahkan *IP address* pada *port router*. Pemberian *IP address* pada sebuah *port router* sangat penting. Untuk melakukan *static routing* ataupun *dynamic routing*, *user* memerlukan *IP address* pada *port router* guna kelancaran dalam melakukan *routing*. *User* hanya perlu mengisi kolom *input* dan memilih *port router* yang akan diberi *IP address*, setelah itu *user* harus menekan tombol *add* untuk melanjutkan proses *input* *IP*. Implementasi *form interface* dapat dilihat pada Gambar 4.6.



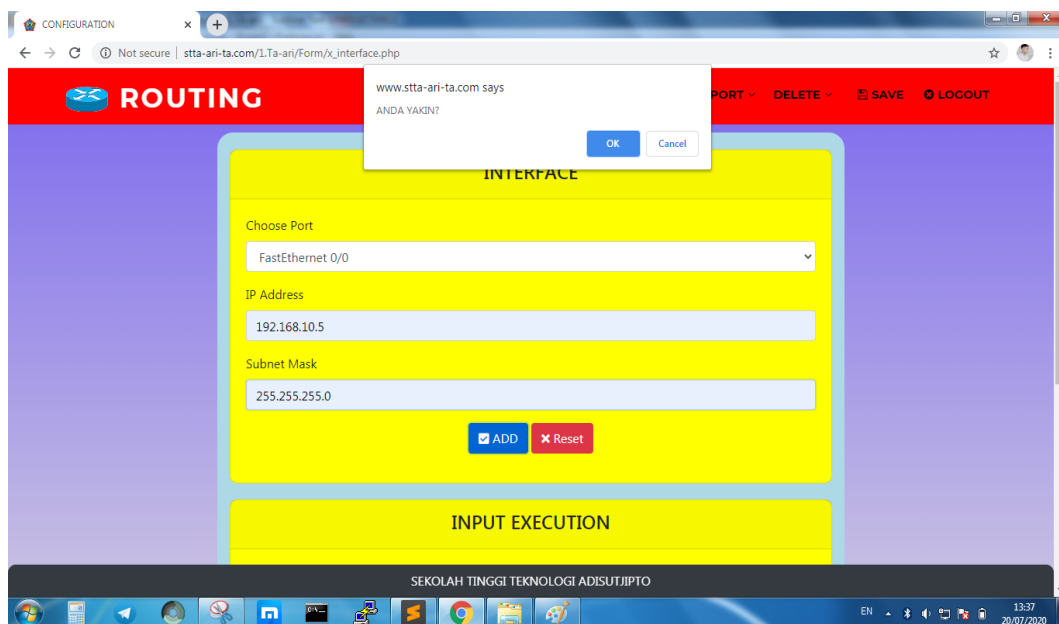
Gambar 4. 6 Implementasi *Form Interface*

Apabila *user* tidak mengisi salah satu kolom *input*, maka akan tampil *alert* yang menunjukkan bahwa kolom *input* tersebut wajib diisi agar *user* dapat melanjutkan proses *input* *IP*. *Alert* pada *form interface* dapat dilihat pada Gambar 4.7.



Gambar 4. 7 Implementasi *Alert Interface*

Pada tombol *add* terdapat *confirm* yang dapat memberikan pilihan untuk melanjutkan *execution* atau tidak. Hal tersebut untuk memastikan *user* sebelum melanjutkan *execution*. *Alert confirm* dapat dilihat pada Gambar 4.8.

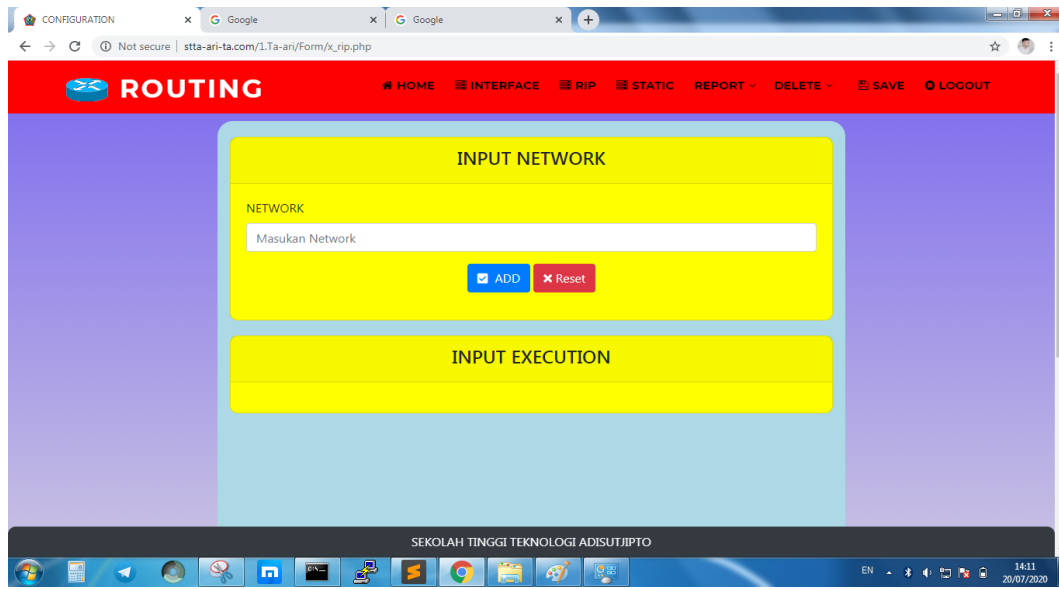


Gambar 4. 8 Implementasi *Confirm Static*

#### 4.1.4 Implementasi *Form RIP*

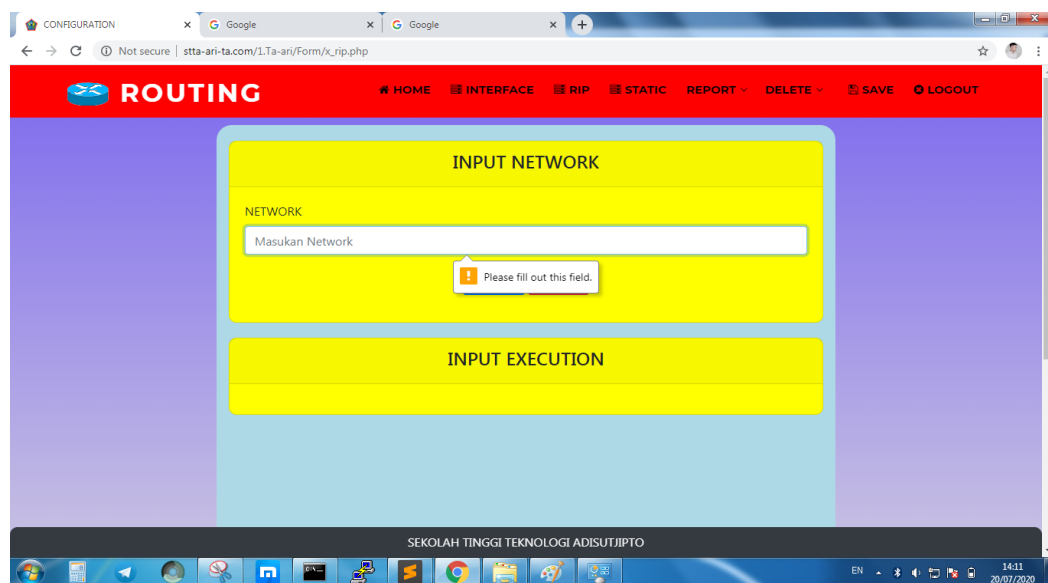
Pada *form RIP* terdapat satu kolom *input*. *User* dapat menggunakan *form RIP* untuk melakukan konfigurasi secara *dynamic*. *User* perlu mengisi *network* pada

kolom *input* yang sudah disediakan, setelah itu *user* perlu menekan tombol *add* untuk melanjutkan proses. Apabila proses berhasil maka hasil *execution* akan ditampilkan pada *form* yang sama. Implementasi *form* RIP dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Implementasi *Form* RIP

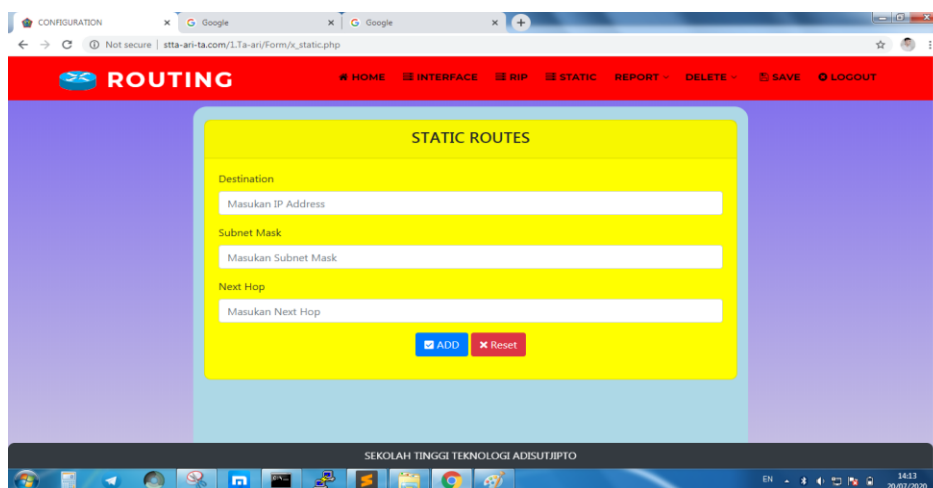
Apabila kolom *input network* tidak diisi maka tombol *add* tidak akan berfungsi dan akan tampil *alert* yang menunjukkan kolom *input* tersebut harus diisi. *Alert* pada *form* RIP dapat dilihat pada gambar 4.10.



Gambar 4. 10 Implementasi *Alert Form* RIP

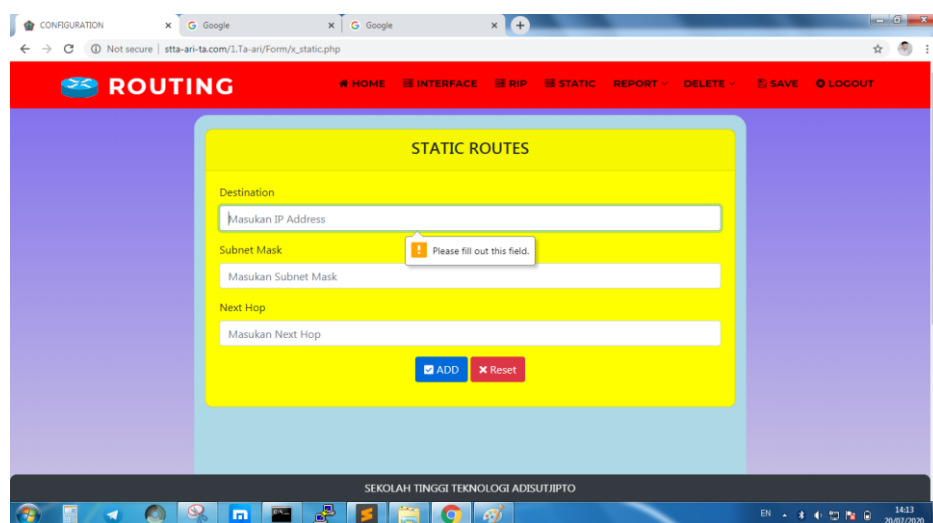
#### 4.1.5 Implementasi *Form Static*

Pada *form static* terdapat tiga kolom *input*. *Form static* berfungsi untuk melakukan konfigurasi *static routing* pada *router*. *User* perlu mengisi *destination*, *subnet mask* dan *next hop* pada kolom *input*, setelah itu menekan tombol *add* untuk proses konfigurasi *static routing*. Implementasi *form static* dapat dilihat pada Gambar 4.11.



Gambar 4. 11 Implementasi *Form Static*

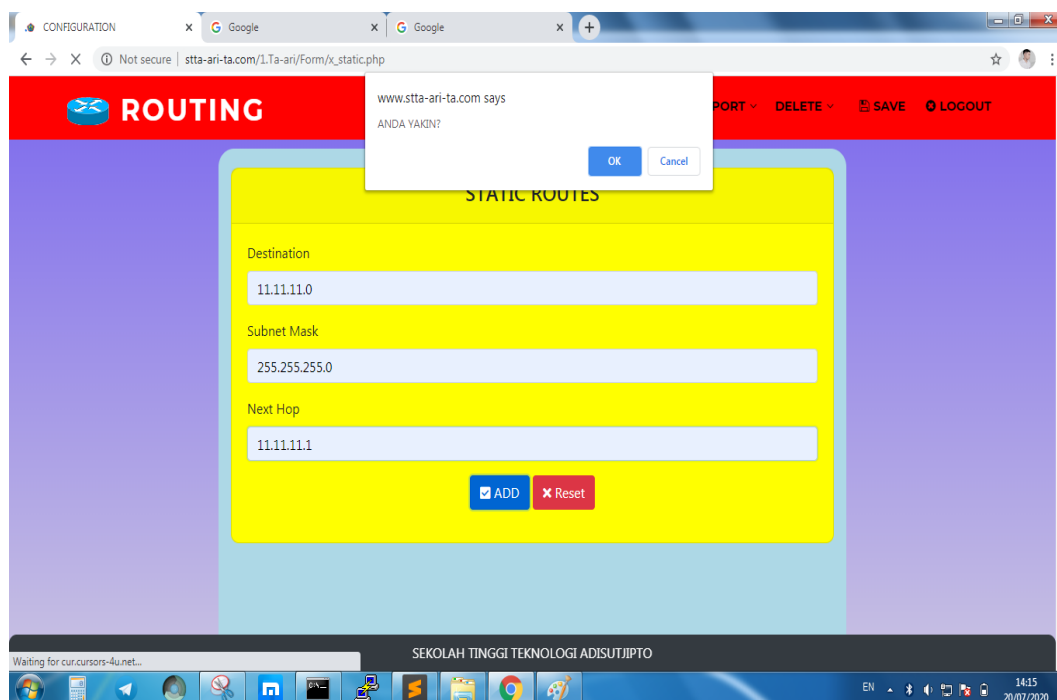
*User* diwajibkan mengisi kolom *input*. Apabila salah satu tidak diisi, maka akan tampil *alert* yang akan mengingatkan *user* untuk mengisi kolom *input* tersebut. Gambar *alert* dapat dilihat pada Gambar 4.12.



Gambar 4. 12 Implementasi *Alert Form Static*



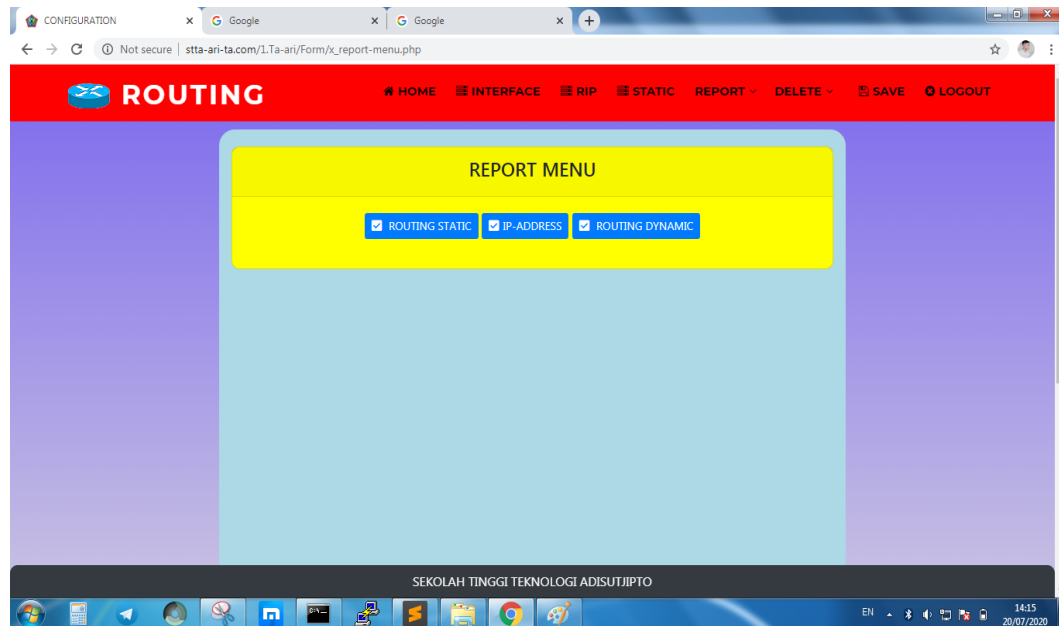
Setelah *user* mengisi kolom *input*, *user* perlu menekan tombol *add* untuk melanjutkan proses. Apabila tombol *add* ditekan maka akan tampil *alert confirm* yang akan memberikan pilihan untuk melanjutkan proses atau tidak. Hal ini digunakan untuk meyakinkan *user* dalam menggunakan fasilitas tersebut. *Alert confirm* dapat dilihat pada Gambar 4.13.



Gambar 4. 13 Implementasi *Form Alert Confirm Static*

#### 4.1.6 Implementasi *Form Report Menu*

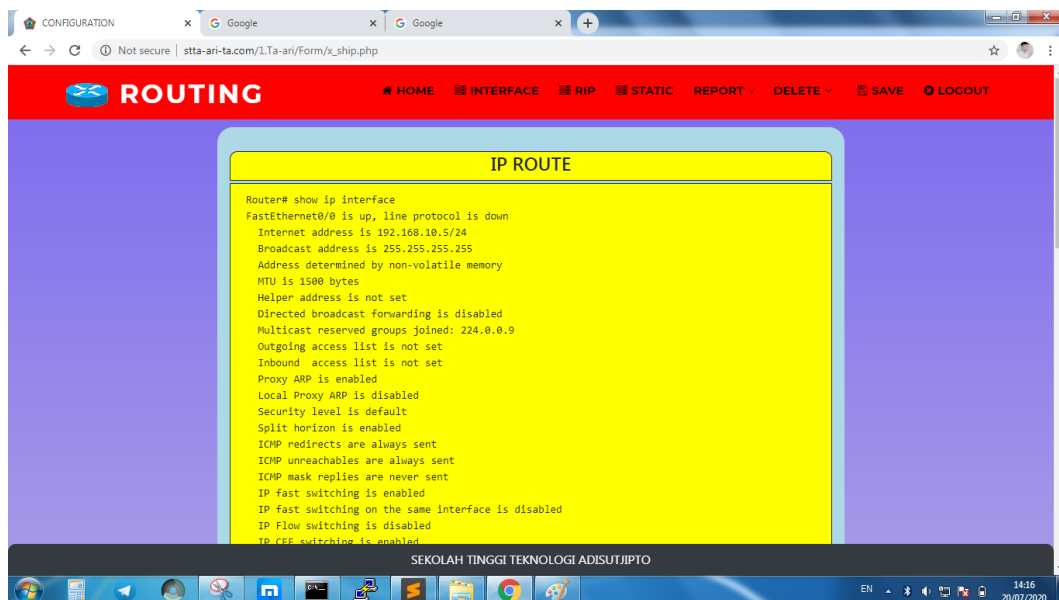
Pada *form* menu terdapat tiga pilihan berupa tombol seperti *routing static*, *routing dynamic* dan juga *show IP*. Dengan adanya *form report* menu dapat mempermudah *user* dalam menggunakan aplikasi. Hal ini dikarenakan seluruh fasilitas menu ditempatkan pada satu *form* yang sama. *Form report* menu dapat digunakan pada bagian *header* dan juga *form report* menu itu sendiri. Tombol tersebut berfungsi untuk mengarahkan pengguna pada *option* yang dipilih. Gambar *form report* menu dapat dilihat pada Gambar 4.14.



Gambar 4. 14 Implementasi *Form Report Menu*

#### 4.1.7 Implementasi *Form Report Show IP*

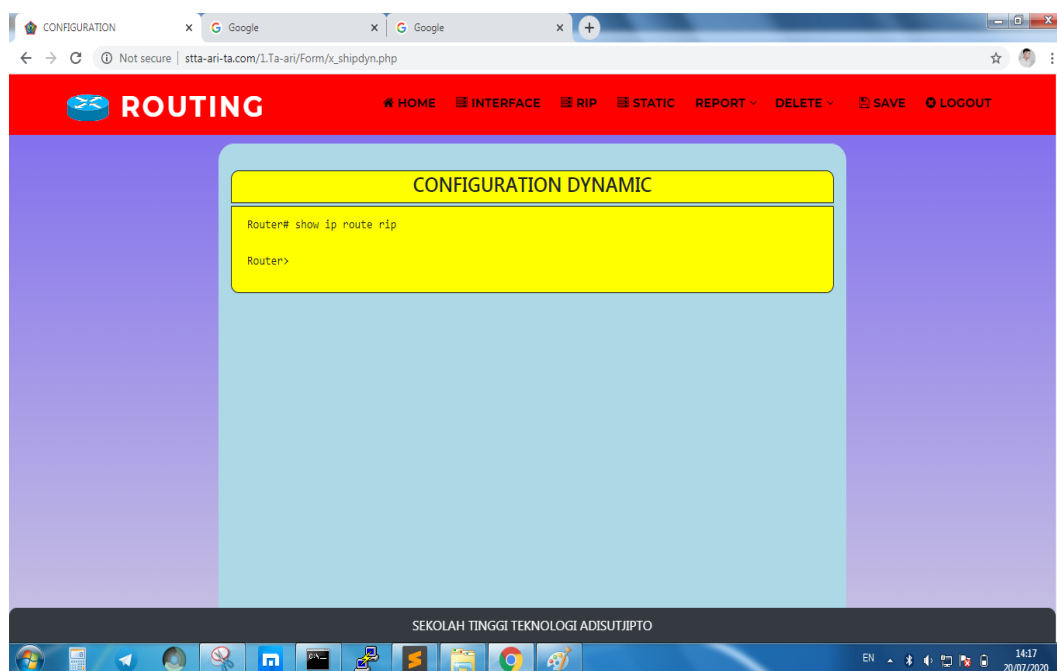
*Form show IP* digunakan untuk melihat IP pada *port router*. *User* hanya perlu menekan tombol *show IP* pada menu *report*. Apabila koneksi berhasil maka akan tampil hasil *show IP*. *Form report show IP* dapat dilihat pada Gambar 4.15.



Gambar 4. 15 Implementasi *Form Report Show IP*

#### 4.1.8 Implementasi *Form Report Show Routing RIP*

*Form report routing RIP* digunakan untuk melihat *routing* yang terhubung dengan menggunakan RIP. Untuk menggunakan *show routing RIP*, *user* hanya perlu menekan tombol *show routing RIP* pada bagian menu *report*. Apabila proses *show routing RIP* berhasil, maka akan tampil pada bagian *form*. Implementasi *form report RIP* dapat dilihat pada Gambar 4.16.



Gambar 4. 16 Implementasi *Form Report RIP*

#### 4.1.9 Implementasi *Form Report Show Routing Static*

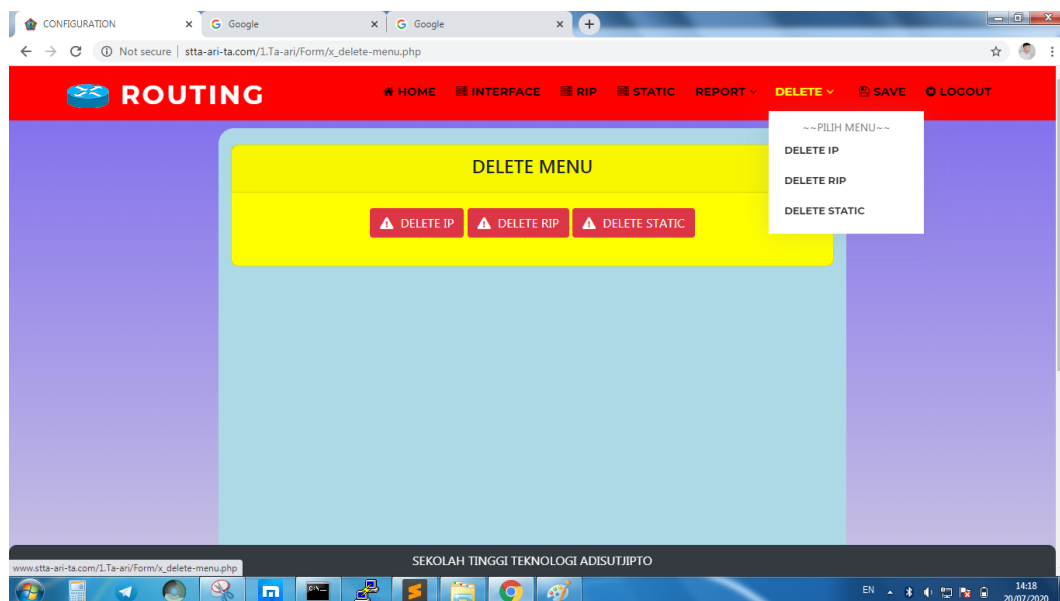
*Form report show routing static* berfungsi untuk menampilkan hasil *routing static* pada *router*. *Form report show routing static* ini cukup penting pada aplikasi ini. Karena dengan menggunakan *form report show routing static* dapat mempermudah *user* untuk melihat hasil *routing* ataupun *routing* yang sudah diproses sebelumnya. *User* perlu menekan tombol *show routing static* pada bagian menu *report* untuk menjalankan proses *show routing static*. Implementasi *form report show routing static* dapat dilihat pada Gambar 4.17.



Gambar 4. 17 Implementasi *Form Report Static*

#### 4.1.10 Implementasi *Form Delete Menu*

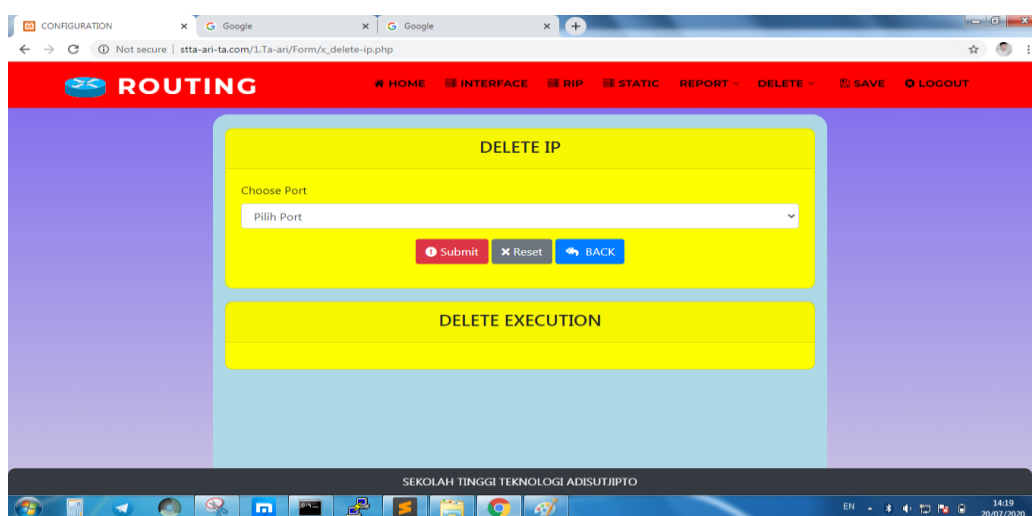
Pada *form delete* menu terdapat tiga tombol yang terdiri dari tombol *delete IP*, *delete RIP* dan *delete static*. Tombol tersebut akan mengarahkan *user* pada *form* yang dipilih. Implementasi *form delete* menu dapat dilihat pada Gambar 4.18.



Gambar 4. 18 Implementasi *Form Delete Menu*

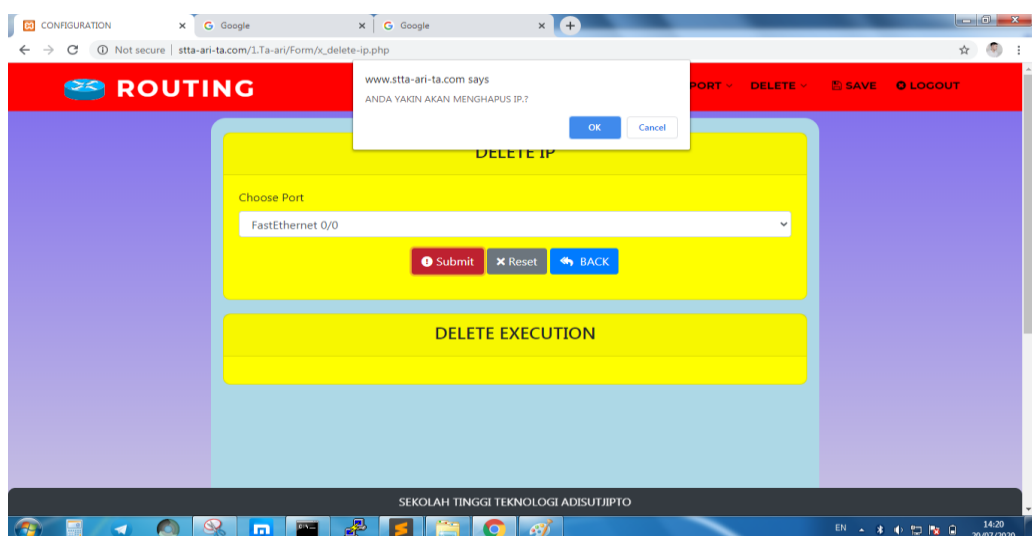
#### 4.1.11 Implementasi *Form Delete IP*

*Form delete IP* berfungsi untuk menghapus *IP address* pada *port*. *User* perlu memilih *port* yang akan digunakan, setelah itu *user* perlu menekan tombol *submit* untuk melanjutkan proses. Apabila proses berhasil maka akan tampil *execution output* pada bagian *form*. Implementasi *form delete IP* dapat dilihat pada Gambar 4.19.



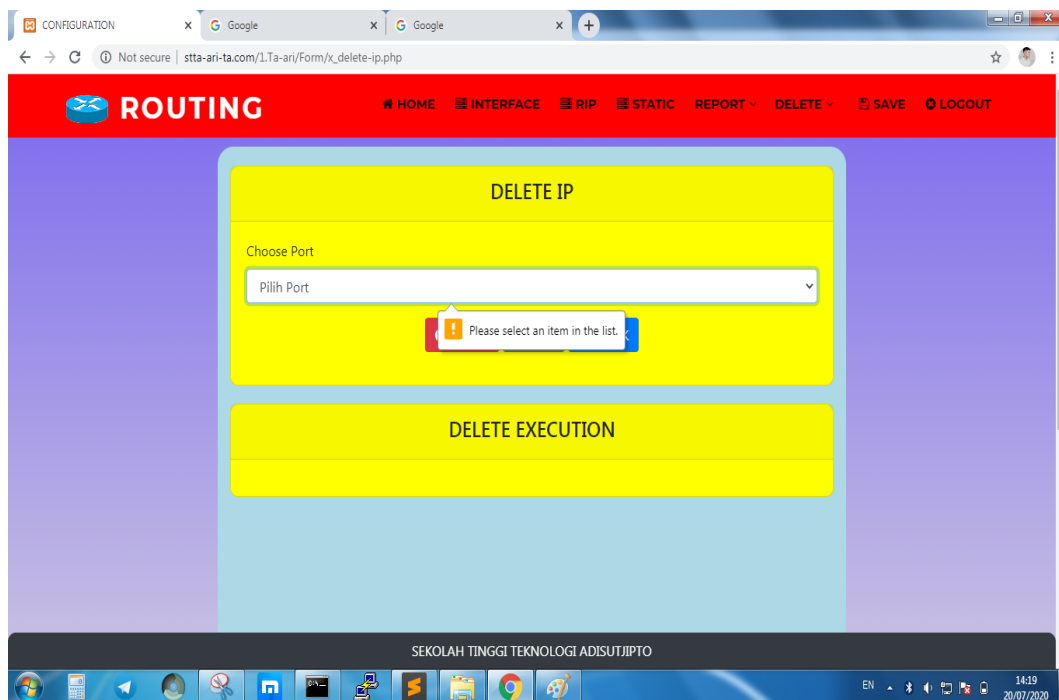
Gambar 4. 19 Implementasi *Form Delete IP*

Pada *form delete IP*, terdapat *alert* konfirmasi yang akan tampil untuk memilih proses “iya” ataupun “tidak”. Gambar *alert confirm* dapat dilihat pada Gambar 4.20.



Gambar 4. 20 Implementasi *Alert Confirm Form Delete IP*

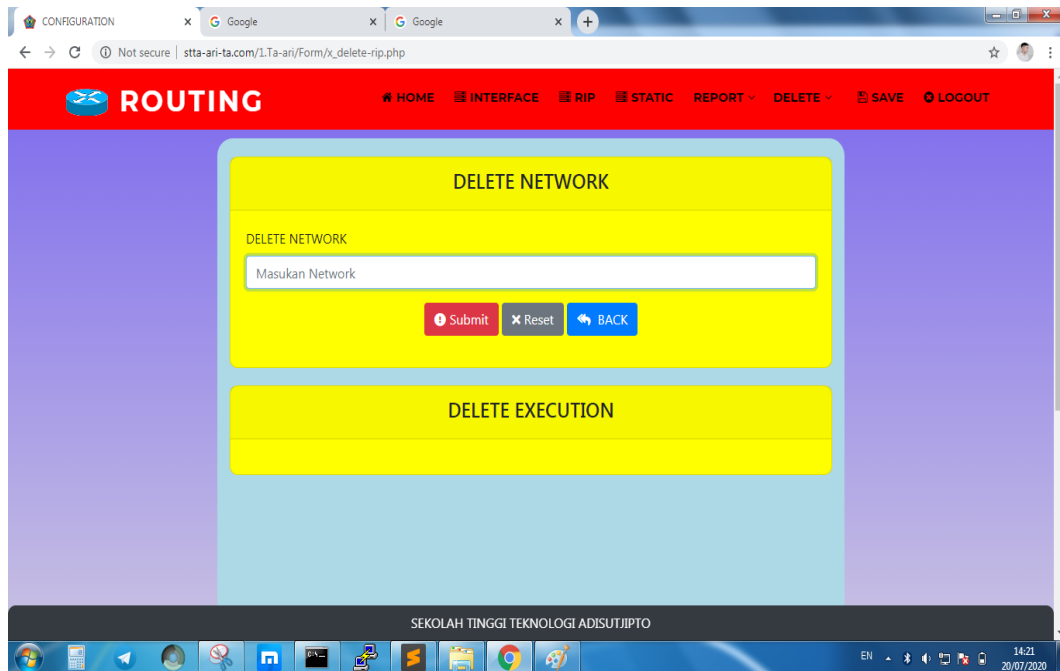
Serta apabila kolom *input* tidak diisi, maka *user* tidak dapat melanjutkan proses *delete* IP dan akan tampil *alert* yang akan mengingatkan bahwa kolom tersebut harus diisi. Hal ini berguna untuk meyakinkan pengguna sebelum melakukan *delete* IP pada sebuah *router*. Implementasi *alert* kolom input pada form delete IP dapat dilihat pada Gambar 4.21.



Gambar 4. 21 Implementasi *Alert Delete* IP

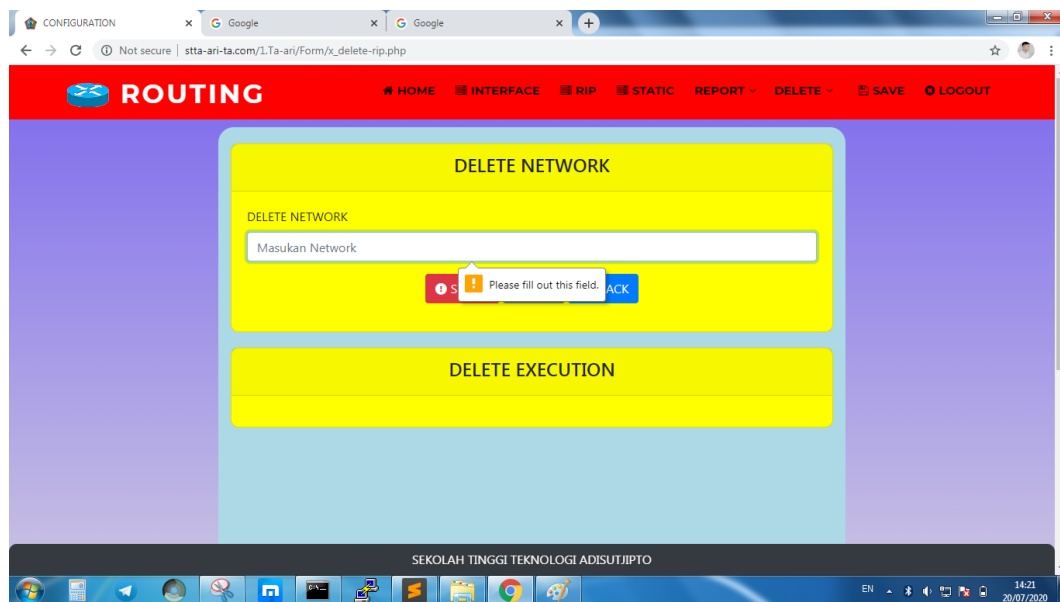
#### 4.1.12 Implementasi *Form Delete* RIP

*Form delete* RIP digunakan untuk menghapus konfigurasi RIP. Terdapat satu kolom *input* untuk menentukan *network* yang akan dihapus. Setelah mengisi kolom *input*, *user* hanya perlu menekan tombol *submit* untuk melanjutkan proses *delete*. Apabila berhasil maka akan tampil hasil akhir pada *delete execution*. *Form delete* RIP sangat penting untuk menghapus konfigurasi RIP pada sebuah *router*. Implementasi *form delete* RIP dapat dilihat pada Gambar 4.22.



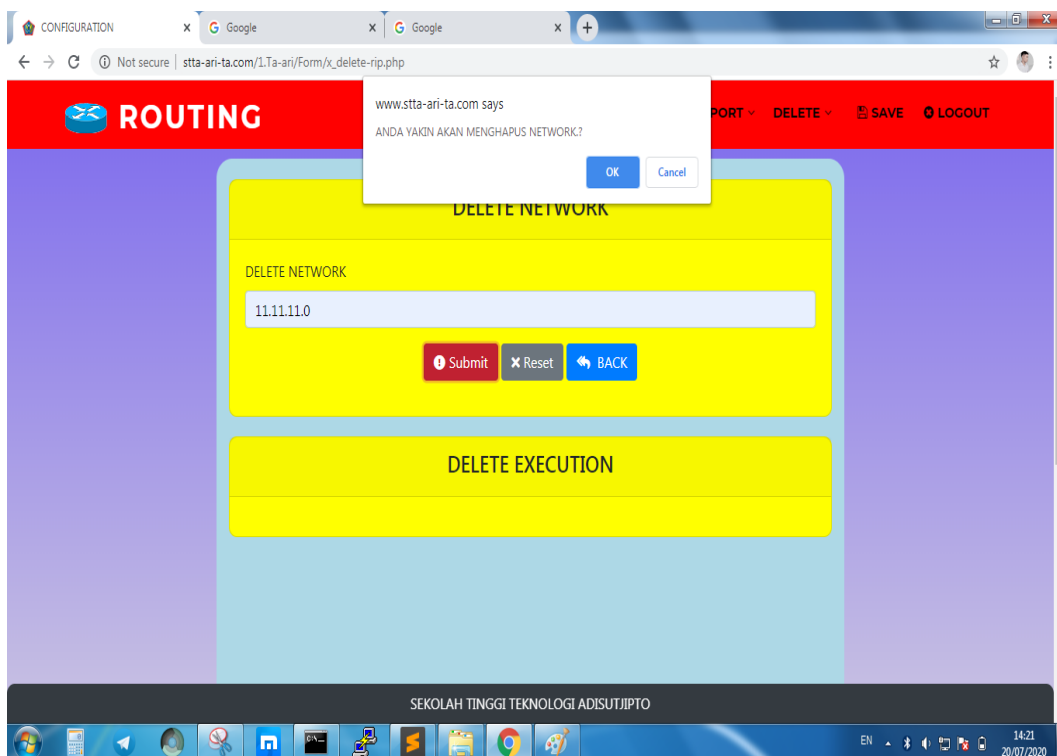
Gambar 4. 22 Implementasi *Form Delete RIP*

Apabila kolom *input* tidak diisi, maka akan tampil *alert* untuk memberi tahu bahwa kolom tersebut harus diisi. *Alert* dapat dilihat pada Gambar 4.23.



Gambar 4. 23 Implementasi *Alert Form Delete RIP*

Pada tombol *submit* terdapat *confirm* yang dapat memberikan pilihan untuk melanjutkan *execution* atau tidak. *Confirm* pada tombol *submit* dapat dilihat pada Gambar 4.24.



Gambar 4. 24 Implementasi *Alert Confirm Form Delete RIP*

#### 4.1.13 Implementasi *Form Delete Static*

*Form delete static* digunakan untuk menghapus konfigurasi *static* pada *router*. Terdapat beberapa kolom *input* yang dapat diisi oleh *user*. Kolom *input* terdiri dari *destination*, *subnet mask* dan juga *next hop*. Dengan menggunakan *form delete static*, *user* dapat menghapus konfigurasi *static routing* yang ada pada *router* secara individual. Untuk menggunakan *form delete static*, *user* perlu mengisi kolom *input*, setelah itu *user* dapat melanjutkan *execution* dengan menekan tombol *submit*. Implementasi *form delete static* dapat dilihat pada Gambar 4.25.

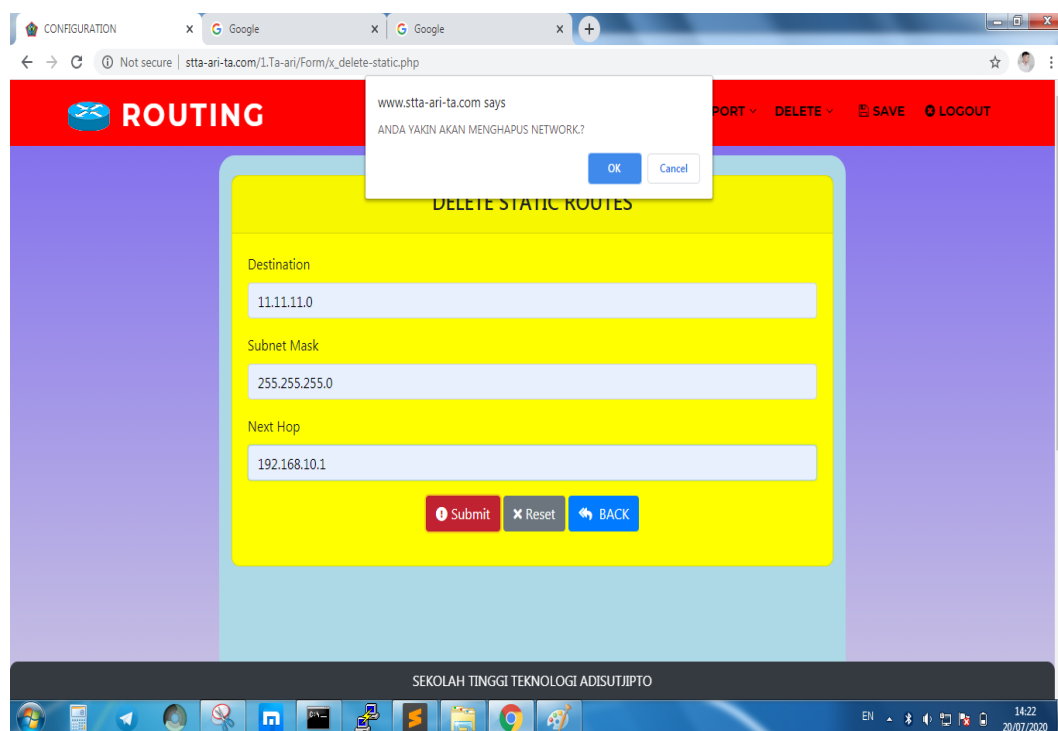


Gambar 4. 25 Implementasi *Form Delete Static*

Terdapat *alert* pada kolom input yang dapat mengingatkan *user* apabila kolom *input* belum diisi. *Alert* pada *form delete static* dapat dilihat pada Gambar 4.26.

Gambar 4. 26 Implementasi *Alert Form Delete Static*

Pada tombol *submit* terdapat fungsi *confirm* yang dapat menentukan apakah *user* akan melanjutkan proses atau tidak. Hal ini berguna untuk meyakinkan *user* sebelum melanjutkan proses pada *router*. *Alert confirm* dapat dilihat pada Gambar 4.27.



Gambar 4. 27 Implementasi *Alert Confirm Delete Static*

#### 4.1.14 Implementasi Telnet

Pembuatan Telnet pada *router* Cisco *type* 2800 berguna untuk mempermudah administrator dalam melakukan konfigurasi pada sebuah *router*. Dengan membuat Telnet, administrator tidak perlu menggunakan aplikasi pengolahan lainnya. Dalam menggunakan Telnet terbilang cukup mudah, administrator hanya memerlukan alamat IP yang sudah dibuat sebelumnya. Telnet yang sudah diaplikasikan atau dibuat pada *router* dapat digunakan untuk menghubungkan aplikasi berbasis web dengan *router*. Sehingga *user* perlu memasukan alamat IP pada aplikasi berbasis web sebelum menggunakan aplikasi. Proses pembuatan Telnet pada *router* Cisco *type* 2800 *series* dapat dilihat pada Gambar 4.28.

```

COM1 - Tera Term VT
File Edit Setup Control Window Help

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line vty 0 15
Router(config-line)#login local
Router(config-line)#username stta password stta123
Router(config)#interface fa0/0
Router(config-if)#ip address 11.11.11.4 255.255.255.0
Router(config-if)#no sh
Router(config-if)#
*Jan 1 00:12:57.479: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jan 1 00:12:58.479: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Jan 1 00:12:59.655: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
Router(config-if)#exit
Router(config)#enable password stta098
Router(config)#

```

Gambar 4. 28 Implementasi Telnet

#### 4.1.15 Implementasi API Cisco

API Cisco digunakan untuk menghubungkan aplikasi *Static dan Routing Information Protocol* berbasis web dengan perangkat lunak pada *router Cisco type 2800 series*. Dalam hal ini selain untuk menghubungkan kedua aplikasi, API Cisco dapat memproses perintah-perintah yang ada pada aplikasi berbasis web agar dapat di proses pada *Cisco router type 2800*. API Cisco dapat dilihat pada Gambar 4.29.

```

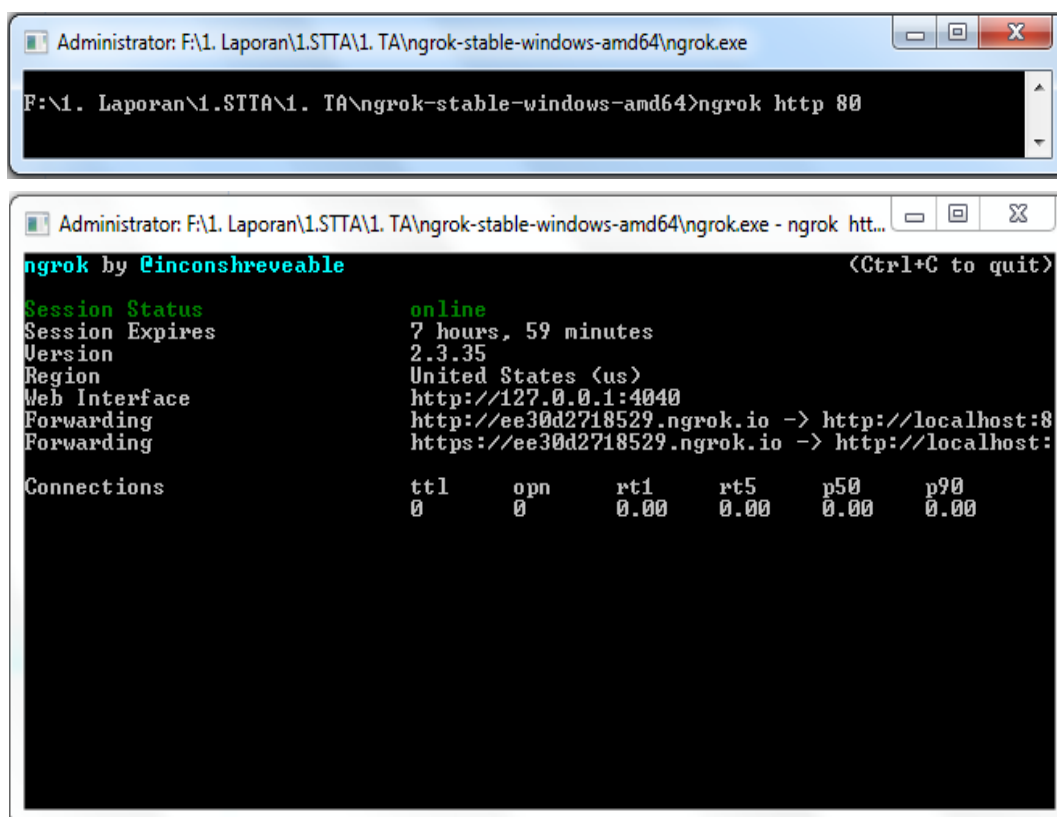
29
30 public function connect()
31 {
32
33     $this->_connection = fsockopen($this->_hostname, $this->_portrouter, $errno, $errstr, $this->_timeout);
34     if ($this->_connection === false) {
35         die("Error: Connection Failed for $this->_hostname\n");
36     } // if
37     stream_set_timeout($this->_connection, $this->_timeout);
38     $this->_readTo(':');
39     if (substr($this->_data, -9) == 'Username:') {
40         $this->_send($this->_username);
41         $this->_readTo(':');
42     } // if
43     $this->_send($this->_pw_router);
44     $this->_prompt = '>';
45     $this->_readTo($this->_prompt);
46     if (strpos($this->_data, $this->_prompt) === false) {
47         fclose($this->_connection);
48         die("Error: Authentication Failed for $this->_hostname\n");
49     } // if
50 } // connect
51

```

Gambar 4. 29 Implementasi API Cisco

#### 4.1.16 Implementasi *Port Forwarding*

Ngrok dimanfaatkan untuk melakukan *port forwarding*. Dengan menggunakan Ngrok, *server* lokal yang tidak memiliki IP publik dapat diakses dari luar (internet) dengan menggunakan *port* 80. Ngrok yang menyediakan fasilitas *port forwarding* dapat membantu dalam sistem *client server*. Sehingga aplikasi berbasis web dapat digunakan dari jarak jauh (*remote*). Implementasi Ngrok dapat dilihat pada Gambar 4.30.



The image shows two screenshots of a Windows command prompt window. The top screenshot shows the command prompt with the command `ngrok http 80` entered. The bottom screenshot shows the output of the command, which includes the ngrok logo, session status (online), session expiration (7 hours, 59 minutes), version (2.3.35), region (United States), web interface (http://127.0.0.1:4040), forwarding rules (http://ee30d2718529.ngrok.io -> http://localhost:80 and https://ee30d2718529.ngrok.io -> http://localhost:80), and a table of connections.

```
Administrator: F:\1. Laporan\1.STTA\1. TA\ngrok-stable-windows-amd64\ngrok.exe
F:\1. Laporan\1.STTA\1. TA\ngrok-stable-windows-amd64>ngrok http 80

Administrator: F:\1. Laporan\1.STTA\1. TA\ngrok-stable-windows-amd64\ngrok.exe - ngrok htt...
ngrok by @inconshreveable                                     (Ctrl+C to quit)

Session Status               online
Session Expires              7 hours, 59 minutes
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://ee30d2718529.ngrok.io -> http://localhost:80
                              https://ee30d2718529.ngrok.io -> http://localhost:80

Connections                   ttl    opn    rt1    rt5    p50    p90
                              0      0      0.00  0.00  0.00  0.00
```

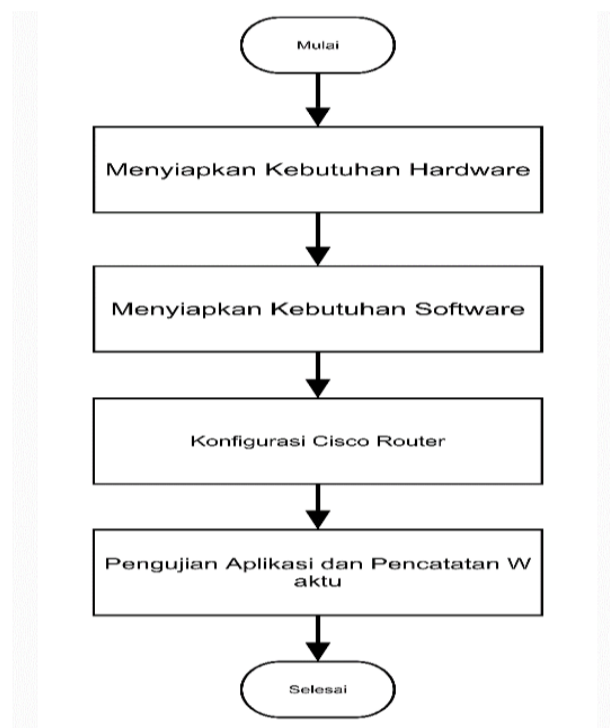
Gambar 4. 30 Implementasi Ngrok

#### 4.2 *Scheme Pengujian*

Pada Gambar 4.31 dapat dijelaskan *scheme* pengujian yang akan digunakan pada saat pengujian aplikasi. Pada *scheme* pengujian *user* harus mempersiapkan *hardware* yang akan digunakan selama pengujian berlangsung. *Hardware* yang digunakan untuk pengujian yaitu dua buah *router*, tiga buah *switch*, empat buah *personal computer*, satu buah laptop, dan beberapa kabel penghubung (RJ45).

Setelah *hardware* siap untuk digunakan, *user* perlu menyiapkan *software* pada laptop. *Software* yang akan digunakan yaitu *xampp*, *chrome*, *operating system* dan juga aplikasi “*Static Dan Routing Information Protocol (RIP) Berbasis Web Pada Cisco Router 2800 Series Memanfaatkan Application Programming Interface (API) Dari Cisco*”.

Tahap selanjutnya yaitu konfigurasi pada *Cisco router type 2800 series*. Dalam hal ini konfigurasi yang akan dilakukan meliputi pembuatan *Telnet*, *username router*, *password router*, *enable router* serta memberikan sebuah alamat *IP address* pada salah satu *port router*. Setelah semua siap untuk digunakan, tahap selanjutnya yaitu melaksanakan pengujian untuk membuat *static routing* dan *dynamic routing* dengan menggunakan aplikasi berbasis web. Selain pengujian terhadap “proses” pengujian juga dilakukan dalam hal “waktu”. Pengujian dilakukan sebanyak lima kali percobaan tanpa kesalahan. Setelah itu akan dicatat “waktu” pengujian dan akan diambil nilai akhir menggunakan perhitungan *average*. Langkah ini berguna untuk menentukan apakah aplikasi berbasis web dapat lebih efisien dan dinamis dibandingkan menggunakan *command line*. Gambar *scheme* pengujian dapat dilihat pada Gambar 4.31.



Gambar 4. 31 *Scheme* Pengujian

### 4.3 Pengujian

Pengujian dilakukan dengan cara tes ping pada setiap *personal computer* yang terhubung. Hal ini dilakukan untuk mengetahui apakah *routing* berhasil atau tidak. Pelaksanaan pengujian dilakukan di laboratorium jaringan Sekolah Tinggi Teknologi Adisutjipto. Gambar laboratorium jaringan dapat dilihat pada Gambar 4.32.



Gambar 4. 32 Laboratorium Jaringan

Pada Gambar 4.33 merupakan *router Cisco type 2800 series* yang digunakan untuk melaksanakan pengujian aplikasi. Gambar *router Cisco type 2800 series* dapat dilihat pada Gambar 4.33.



Gambar 4. 33 Router Cisco Type 2800 Series

Pada Gambar 4.34 merupakan *personal computer* yang akan melakukan tes ping. Gambar *personal computer* dapat dilihat pada Gambar 4.34



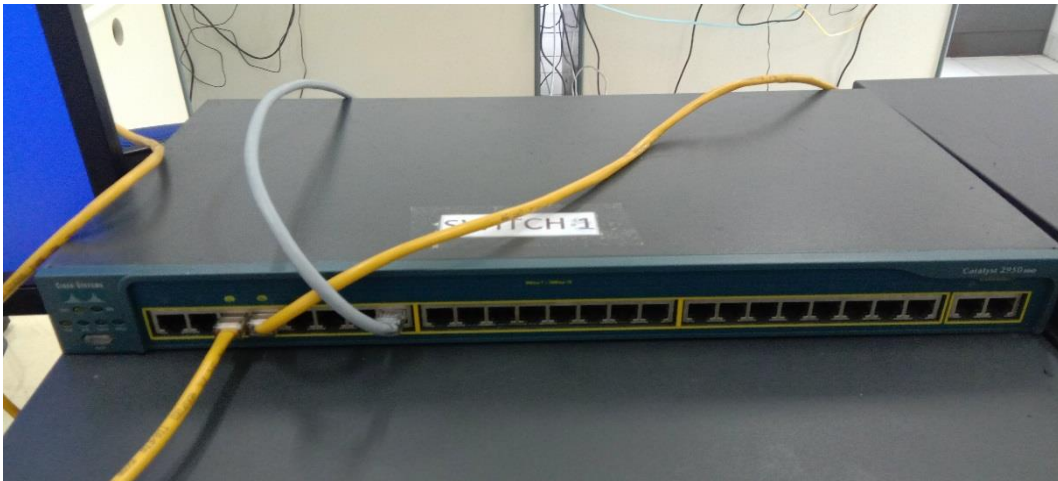
Gambar 4. 34 *Personal Computer*

Pada Gambar 4.35 sampai 4.37 merupakan *switch* yang digunakan untuk menghubungkan *personal computer* dengan *router*. *Switch* pada Gambar 4.35 berfungsi untuk menghubungkan *personal computer* satu dan *personal computer* dua dengan menggunakan IP Address 10.10.10.0/24 menuju *router* pada *interface fastethernet 0/1* dengan IP Address 10.10.10.66/24. Gambar 4.36 merupakan *switch* yang berfungsi untuk menghubungkan “PC-3” dan “PC-4” menggunakan jaringan 12.12.12.0/24 menuju *router-0* pada *interface fastethernet 0/1* dengan alamat IP Address 12.12.12.2/24. Sedangkan *switch* pada Gambar 4.37 merupakan *switch* yang menghubungkan antara laptop atau *web server* dengan *router-0* dan *router-1* menggunakan jaringan 11.11.11.0/24 menuju *router-0 interface fastethernet 0/0* menggunakan IP Address 11.11.11.2/24 dan menuju *router-1 fastethernet 0/0* menggunakan IP Address 11.11.11.4/24. Gambar *switch* dapat dilihat pada Gambar 4.35 sampai 4.37.





Gambar 4. 35 Switch Menuju Router-0



Gambar 4. 36 Switch Menuju Router-1

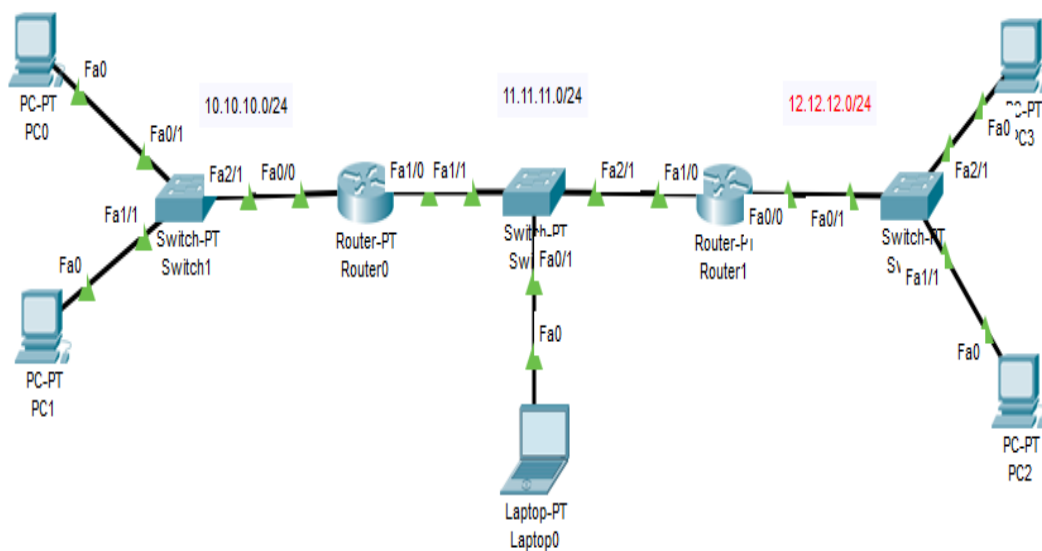


Gambar 4. 37 Switch Menuju Router-0 dan Router-1



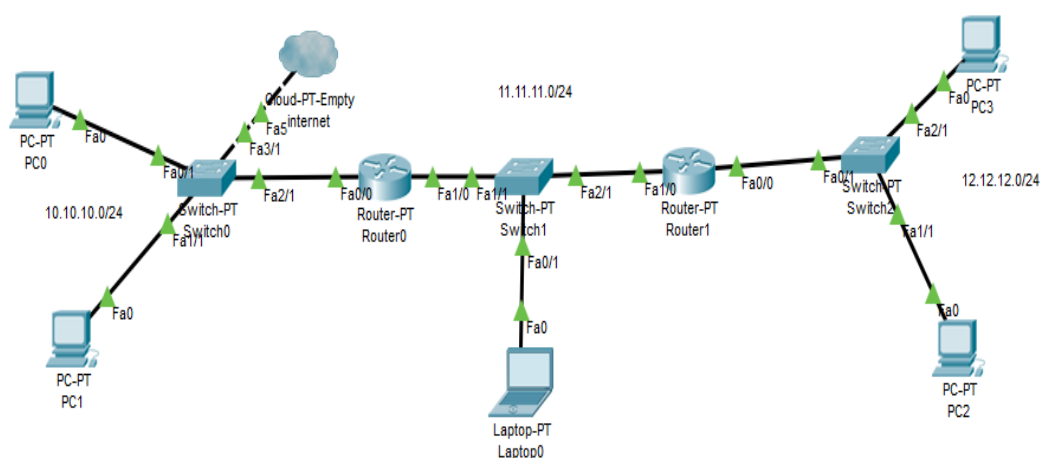
### 4.3.1 Design Jaringan

*Design* jaringan pada Gambar 4.38 merupakan topologi jaringan yang digunakan untuk melaksanakan pengujian pada sistem aplikasi. Pada topologi ini terdapat dua buah *router* Cisco type 2800 series dengan nama “*router-0*” dan “*router-1*”. *Router* tersebut yang akan digunakan untuk pengujian *static routing* dan juga *dynamic routing* menggunakan aplikasi berbasis web. Selain *router*, terdapat tiga buah *switch* yang digunakan untuk menghubungkan *personal computer* dengan *router*. *Switch* pada topologi ini berfungsi untuk memperbanyak *port* pada sebuah *router*. *Switch* pada Gambar 4.31 diberi nama “*Switch-1*”, “*Switch-2*” dan “*Switch-3*”. Selain *router* dan *switch*, terdapat empat buah *personal computer* serta sebuah laptop yang berfungsi sebagai *web server* yang sudah diberi aplikasi “*Static Dan Routing Information Protocol (RIP) Berbasis web Pada Cisco Router 2800 Series Memanfaatkan Application Programming Interface (API) Dari Cisco*”. Gambar *design* jaringan dapat dilihat pada Gambar 4.38.



Gambar 4. 38 *Design* Jaringan

Pada Gambar 4.39 merupakan *design* jaringan yang diberi akses internet. *Design* jaringan 4.39 merupakan jaringan yang dapat melakukan konfigurasi *router* dari luar (internet). *Web server* pada laptop akan diberi internet yang berfungsi untuk menjalankan ngrok, sehingga *web server* dapat diakses melalui alamat *fowarding* seperti pada Gambar 4.30. *Router0* dan *router1* dikonfigurasi *nat* agar setiap *PC* mendapatkan layanan internet. *Design* jaringan dengan internet dapat dilihat pada Gambar 4.39



Gambar 4. 39 *Design* Jaringan Dengan Internet

Pada Tabel 4.1 dapat dijelaskan mengenai *port* yang terhubung pada setiap *personal computer* dengan *router* serta penjelasan IP pada setiap unit yang ada. Hal ini dapat mempermudah dalam pengujian agar *admin/user* tidak lupa pada IP yang sudah diberikan untuk setiap *device*. Pada Tabel 4.1 dapat dilihat alur atau *scheme* yang akan dilakukan dalam pengujian aplikasi berbasis web. Tabel IP dapat dilihat pada Tabel 4.1.

Tabel 4. 1 IP Address

NO	NAMA	IP ADDRESS	SUBNET MASK	INTERFACE	INPUT
1	PC0	10.10.10.4	255.255.255.0	FastEthernet 0/1	Router-0
2	PC1	10.10.10.6	255.255.255.0	FastEthernet 0/1	Router-0
3	PC2	12.12.12.4	255.255.255.0	FastEthernet 0/1	Router-1

Tabel 4. 1 Lanjutan

NO	NAMA	IP ADDRESS	SUBNET MASK	INTERFACE	INPUT
4	PC3	12.12.12.6	255.255.255.0	FastEthernet 0/1	Router-1
5	LAPTOP	11.11.11.6	255.255.255.0	FastEthernet 0/0	Router-0, Router-1
6	Router-0	10.10.10.66	255.255.255.0	FastEthernet 0/1	-
7	Router-0	11.11.11.2	255.255.255.0	FastEthernet 0/0	-
8	Router-1	11.11.11.4	255.255.255.0	FastEthernet 0/0	-
9	Router-1	12.12.12.2	255.255.255.0	FastEthernet 0/1	-

### 4.3.2 Test Ping Tanpa Routing

Pada tahap ini, seluruh *personal computer* yang terhubung dengan *router* tidak menggunakan *static routing* ataupun *dynamic routing*. Tes ping dilakukan pada setiap *personal computer*. Tes ping tanpa *routing* dapat dilihat pada Gambar 4.40 dan Tabel 4.2.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\User>ping 10.10.10.6
Pinging 10.10.10.6 with 32 bytes of data:
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Ping statistics for 10.10.10.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\User>ping 12.12.12.4
Pinging 12.12.12.4 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 12.12.12.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\User>ping 12.12.12.6
Pinging 12.12.12.6 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 12.12.12.6:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\User>ping 11.11.11.6
Pinging 11.11.11.6 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 11.11.11.6:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\User>

```

Gambar 4. 40 Tes Ping Tanpa Routing

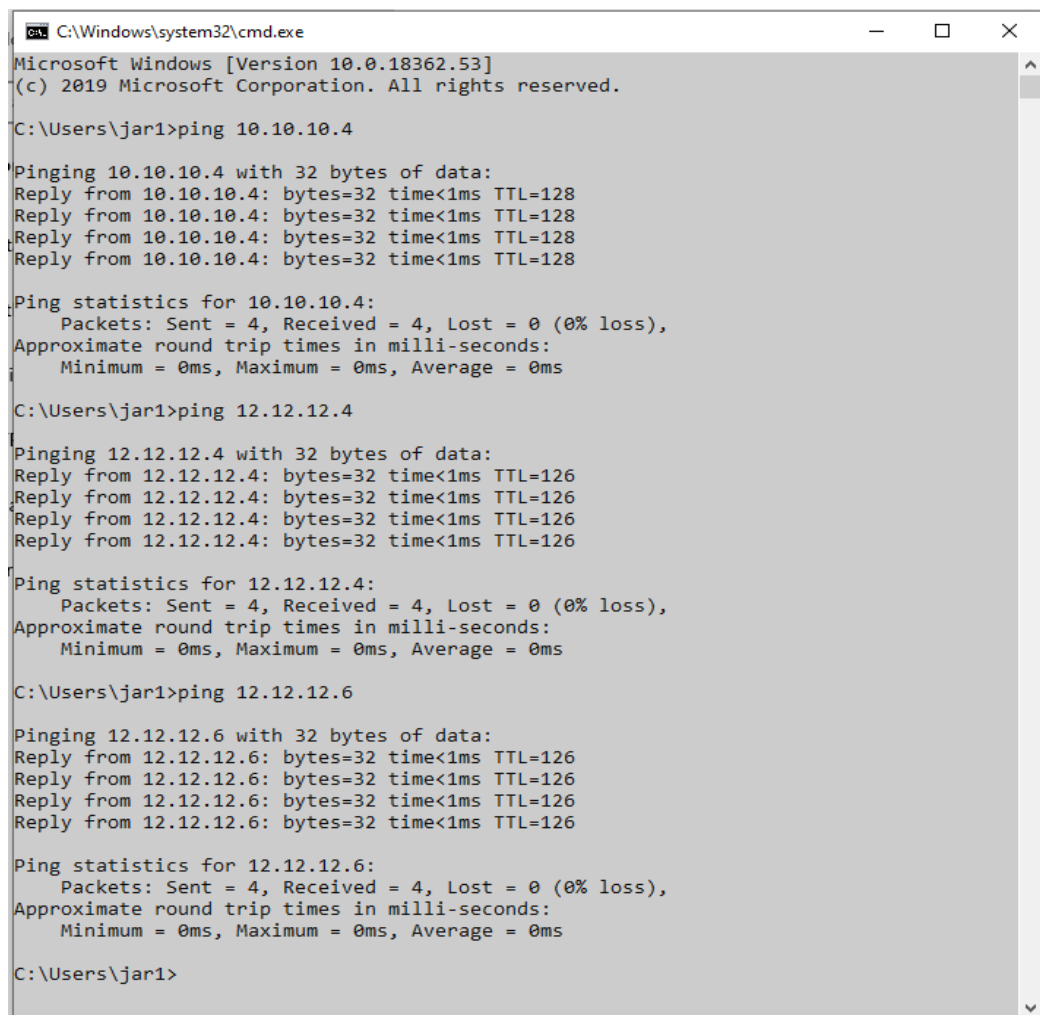
Tabel 4. 2 Tes Ping Tanpa *Routing*

<b>NO</b>	<b>NAMA</b>	<b>IP TUJUAN</b>	<b>NAMA TUJUAN</b>	<b>INTERFACE</b>	<b>HASIL</b>
1	PC1	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
2	PC1	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
3	PC1	11.11.11.4	Router-1	FastEthernet 0/0	Tidak Berhasil
4	PC1	12.12.12.2	Router-1	FastEthernet 0/1	Tidak Berhasil
5	PC1	10.10.10.4	PC2	FastEthernet 0/0	Berhasil
6	PC1	12.12.12.4	PC3	FastEthernet 0/0	Tidak Berhasil
7	PC1	12.12.12.6	PC4	FastEthernet 0/0	Tidak Berhasil
8	PC2	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
9	PC2	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
10	PC2	11.11.11.4	Router-1	FastEthernet 0/0	Tidak Berhasil
11	PC2	12.12.12.2	Router-1	FastEthernet 0/1	Tidak Berhasil
12	PC2	10.10.10.6	PC1	FastEthernet 0/0	Berhasil
13	PC2	12.12.12.4	PC3	FastEthernet 0/0	Tidak Berhasil
14	PC2	12.12.12.6	PC4	FastEthernet 0/0	Tidak Berhasil
15	PC3	10.10.10.66	Router-0	FastEthernet 0/1	Tidak Berhasil
16	PC3	11.11.11.2	Router-0	FastEthernet 0/0	Tidak Berhasil
17	PC3	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
18	PC3	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
19	PC3	10.10.10.6	PC1	FastEthernet 0/0	Tidak Berhasil
20	PC3	10.10.10.4	PC2	FastEthernet 0/0	Tidak Berhasil
21	PC3	12.12.12.6	PC4	FastEthernet 0/0	Berhasil
22	PC4	10.10.10.66	Router-0	FastEthernet 0/1	Tidak Berhasil
23	PC4	11.11.11.2	Router-0	FastEthernet 0/0	Tidak Berhasil
24	PC4	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
25	PC4	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
26	PC4	10.10.10.6	PC1	FastEthernet 0/0	Tidak Berhasil
27	PC4	10.10.10.4	PC2	FastEthernet 0/0	Tidak Berhasil
28	PC4	12.12.12.4	PC3	FastEthernet 0/0	Berhasil

### 4.3.3 Pengujian *Dynamic Routing*

Pada tahap ini, konfigurasi *router* dengan menggunakan *Routing Information Protocol*. Aplikasi berbasis web akan melakukan proses konfigurasi pada *router-0* dan *router-1*. Admin perlu memasukan *network* pada setiap *router* dengan menggunakan fasilitas yang ada pada *form* RIP.

Tes ping dilakukan pada setiap *personal computer*, hal ini berfungsi untuk memastikan bahwa *personal computer* sudah terhubung. setiap *personal computer* akan mencoba tes ping dengan jaringan yang berbeda. Apabila konfigurasi berhasil maka akan tampil seperti Gambar 4.41 sampai 4.44 Tes ping dapat dilihat pada Gambar 4.41 sampai 4.44.



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\jar1>ping 10.10.10.4

Pinging 10.10.10.4 with 32 bytes of data:
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar1>ping 12.12.12.4

Pinging 12.12.12.4 with 32 bytes of data:
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126

Ping statistics for 12.12.12.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar1>ping 12.12.12.6

Pinging 12.12.12.6 with 32 bytes of data:
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126

Ping statistics for 12.12.12.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar1>

```

Gambar 4. 41 Tes Ping RIP PC-1

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User>ping 10.10.10.6

Pinging 10.10.10.6 with 32 bytes of data:
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 12.12.12.4

Pinging 12.12.12.4 with 32 bytes of data:
Reply from 12.12.12.4: bytes=32 time=904ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126

Ping statistics for 12.12.12.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 904ms, Average = 226ms

C:\Users\User>ping 12.12.12.6

Pinging 12.12.12.6 with 32 bytes of data:
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126

Ping statistics for 12.12.12.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>

```

Gambar 4. 42 Tes Ping RIP PC-2

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\jar3>ping 10.10.10.6

Pinging 10.10.10.6 with 32 bytes of data:
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar3>ping 10.10.10.4

Pinging 10.10.10.4 with 32 bytes of data:
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar3>ping 12.12.12.6

Pinging 12.12.12.6 with 32 bytes of data:
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128

Ping statistics for 12.12.12.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar3>

```

Gambar 4. 43 Tes Ping RIP PC-3

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User>ping 10.10.10.6

Pinging 10.10.10.6 with 32 bytes of data:
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 10.10.10.4

Pinging 10.10.10.4 with 32 bytes of data:
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 12.12.12.4

Pinging 12.12.12.4 with 32 bytes of data:
Reply from 12.12.12.4: bytes=32 time<1ms TTL=128
Reply from 12.12.12.4: bytes=32 time<1ms TTL=128
Reply from 12.12.12.4: bytes=32 time<1ms TTL=128
Reply from 12.12.12.4: bytes=32 time<1ms TTL=128

Ping statistics for 12.12.12.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\User>

```

Gambar 4. 44 Tes Ping RIP PC-4

Pada Gambar 4.45 merupakan hasil *tracert* yang dilakukan pada PC yang memiliki jaringan 12.12.12.0/24 menuju PC yang memiliki jaringan 10.10.10.0/24 dengan konfigurasi *dynamic routing*. *Tracert* dilakukan untuk mengetahui jalur yang dilalui hingga mencapai suatu tujuan. *Tracert* dapat dilihat pada Gambar 4.45.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\jar3>tracert 10.10.10.4

Tracing route to 10.10.10.4 over a maximum of 30 hops

  1  <1 ms    <1 ms    <1 ms    12.12.12.2
  2  <1 ms    <1 ms    <1 ms    11.11.11.2
  3  <1 ms    <1 ms    <1 ms    10.10.10.4

Trace complete.

C:\Users\jar3>

```

Gambar 4. 45 Tracert RIP

Pada Tabel 4.3 dapat dijelaskan koneksi setiap *personal computer* dengan *router*. Pada tahap ini, dapat dibuktikan bahwa *dynamic routing* dengan menggunakan aplikasi berbasis web dapat berjalan sesuai dengan perancangan. Tabel tes ping RIP dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Tes Ping RIP

<b>NO</b>	<b>NAMA</b>	<b>IP TUJUAN</b>	<b>NAMA TUJUAN</b>	<b>INTERFACE</b>	<b>HASIL</b>
1	PC1	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
2	PC1	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
3	PC1	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
4	PC1	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
5	PC1	10.10.10.4	PC2	FastEthernet 0/0	Berhasil
6	PC1	12.12.12.4	PC3	FastEthernet 0/0	Berhasil
7	PC1	12.12.12.6	PC4	FastEthernet 0/0	Berhasil
8	PC2	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
9	PC2	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
10	PC2	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
11	PC2	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
12	PC2	10.10.10.6	PC1	FastEthernet 0/0	Berhasil
13	PC2	12.12.12.4	PC3	FastEthernet 0/0	Berhasil
14	PC2	12.12.12.6	PC4	FastEthernet 0/0	Berhasil
15	PC3	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
16	PC3	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
17	PC3	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
18	PC3	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
19	PC3	10.10.10.6	PC1	FastEthernet 0/0	Berhasil
21	PC3	12.12.12.6	PC4	FastEthernet 0/0	Berhasil
22	PC4	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
23	PC4	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil



Tabel 4. 3 Lanjutan

NO	NAMA	IP TUJUAN	NAMA TUJUAN	INTERFACE	HASIL
24	PC4	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
25	PC4	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
26	PC4	10.10.10.6	PC1	FastEthernet 0/0	Berhasil
27	PC4	10.10.10.4	PC2	FastEthernet 0/0	Berhasil
28	PC4	12.12.12.4	PC3	FastEthernet 0/0	Berhasil

Pada Tabel 4.3 membuktikan bahwa konfigurasi secara *dynamic routing* dengan menggunakan *Graphical User Interface* (GUI) dan *command line* memiliki hasil yang sama.

#### 4.3.4 Pengujian *Static Routing*

Pada tahap ini, aplikasi berbasis web digunakan untuk melakukan pengujian *static routing*. Konfigurasi *static routing* akan dilakukan pada setiap *router* dengan menggunakan aplikasi berbasis web. *User* dapat menggunakan *form static* untuk melakukan konfigurasi *static routing*. *User* perlu mengisi kolom *input* pada *form static* yang sudah disediakan. *User* hanya memerlukan *Destination*, *subnet mask* dan *next-hop*. Setelah konfigurasi selesai, dilakukan pengujian dengan cara tes ping ada setiap *personal computer*. Tes ping dapat dilihat pada Gambar 4.46 sampai 4.49.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\jar1>ping 10.10.10.4
Pinging 10.10.10.4 with 32 bytes of data:
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\jar1>ping 12.12.12.4
Pinging 12.12.12.4 with 32 bytes of data:
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Ping statistics for 12.12.12.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\jar1>ping 12.12.12.6
Pinging 12.12.12.6 with 32 bytes of data:
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Ping statistics for 12.12.12.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\jar1>

```

Gambar 4. 46 Tes Ping *Static* PC-1

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User>ping 10.10.10.6

Pinging 10.10.10.6 with 32 bytes of data:
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128
Reply from 10.10.10.6: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 12.12.12.4

Pinging 12.12.12.4 with 32 bytes of data:
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126
Reply from 12.12.12.4: bytes=32 time<1ms TTL=126

Ping statistics for 12.12.12.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 12.12.12.6

Pinging 12.12.12.6 with 32 bytes of data:
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126
Reply from 12.12.12.6: bytes=32 time<1ms TTL=126

Ping statistics for 12.12.12.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>

```

Gambar 4. 47 Tes Ping Static PC-2

```

C:\Windows\system32\cmd.exe
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\jar3>ping 10.10.10.4

Pinging 10.10.10.4 with 32 bytes of data:
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar3>ping 10.10.10.6

Pinging 10.10.10.6 with 32 bytes of data:
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar3>ping 12.12.12.6

Pinging 12.12.12.6 with 32 bytes of data:
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128
Reply from 12.12.12.6: bytes=32 time<1ms TTL=128

Ping statistics for 12.12.12.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\jar3>

```

Gambar 4. 48 Tes Ping Static PC-3

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User>ping 10.10.10.4

Pinging 10.10.10.4 with 32 bytes of data:
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126
Reply from 10.10.10.4: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 10.10.10.6

Pinging 10.10.10.6 with 32 bytes of data:
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126
Reply from 10.10.10.6: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 12.12.12.4

Pinging 12.12.12.4 with 32 bytes of data:
Reply from 12.12.12.4: bytes=32 time<1ms TTL=128
Reply from 12.12.12.4: bytes=32 time<1ms TTL=128
Reply from 12.12.12.4: bytes=32 time<1ms TTL=128
Reply from 12.12.12.4: bytes=32 time=1ms TTL=128

Ping statistics for 12.12.12.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\User>

```

Gambar 4. 49 Tes Ping *Static* PC-4

Pada Gambar 4.50 merupakan hasil *tracert* yang dilakukan dari PC dengan jaringan 10.10.10.0/24 menuju PC dengan jaringan 12.12.12.0/24. *Tracert* dilakukan guna mengetahui jalur yang dilalui untuk mencapai tujuan. *Tracert* dapat dilihat pada Gambar 4.50.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User>tracert 12.12.12.4

Tracing route to 12.12.12.4 over a maximum of 30 hops

  1  <1 ms    <1 ms    <1 ms    10.10.10.66
  2  <1 ms    <1 ms    <1 ms    11.11.11.4
  3  <1 ms    <1 ms    <1 ms    12.12.12.4

Trace complete.

C:\Users\User>

```

Gambar 4. 50 *Tracert Static Routing*

Pada Tabel 4.4 dapat dijelaskan koneksi setiap *personal computer* dengan *router*. Pada tahap ini, konfigurasi *router* dengan menggunakan *static routing*. Pengujian dilakukan dengan menggunakan aplikasi berbasis web. Dapat disimpulkan dengan “HASIL” bahwa konfigurasi *static routing* dapat berjalan sesuai dengan fungsinya. Tabel koneksi dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Tes Ping *Static*

<b>NO</b>	<b>NAMA</b>	<b>IP TUJUAN</b>	<b>NAMA TUJUAN</b>	<b>INTERFACE</b>	<b>HASIL</b>
1	PC1	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
2	PC1	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
3	PC1	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
4	PC1	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
5	PC1	10.10.10.4	PC2	FastEthernet 0/0	Berhasil
6	PC1	12.12.12.4	PC3	FastEthernet 0/0	Berhasil
7	PC1	12.12.12.6	PC4	FastEthernet 0/0	Berhasil
8	PC2	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
9	PC2	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
10	PC2	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
11	PC2	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
12	PC2	10.10.10.6	PC1	FastEthernet 0/0	Berhasil
13	PC2	12.12.12.4	PC3	FastEthernet 0/0	Berhasil
14	PC2	12.12.12.6	PC4	FastEthernet 0/0	Berhasil
15	PC3	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
16	PC3	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
17	PC3	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
18	PC3	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
19	PC3	10.10.10.6	PC1	FastEthernet 0/0	Berhasil
20	PC3	10.10.10.4	PC2	FastEthernet 0/0	Berhasil
21	PC3	12.12.12.6	PC4	FastEthernet 0/0	Berhasil

Tabel 4. 4 Lanjutan

NO	NAMA	IP TUJUAN	NAMA TUJUAN	INTERFACE	HASIL
22	PC4	10.10.10.66	Router-0	FastEthernet 0/1	Berhasil
23	PC4	11.11.11.2	Router-0	FastEthernet 0/0	Berhasil
24	PC4	11.11.11.4	Router-1	FastEthernet 0/0	Berhasil
25	PC4	12.12.12.2	Router-1	FastEthernet 0/1	Berhasil
26	PC4	10.10.10.6	PC1	FastEthernet 0/0	Berhasil
27	PC4	10.10.10.4	PC2	FastEthernet 0/0	Berhasil
28	PC4	12.12.12.4	PC3	FastEthernet 0/0	Berhasil

Pada Tabel 4.4 merupakan hasil pengujian tes ping dengan menggunakan *static routing*. Konfigurasi pada Tabel 4.4 dilakukan dengan menggunakan aplikasi “Static Dan Routing Information Protocol (RIP) Berbasis Web Pada Cisco Router Type 2800 Series Memanfaatkan Application Programming Interface (API) Dari Cisco”. Hal ini membuktikan bahwa konfigurasi *static routing* pada *router* dengan menggunakan *Graphical User Interface* (GUI) dan *command line* memiliki hasil yang sama.

#### 4.3.5 Perbandingan Efisiensi GUI Dan *Command Line*

Pada tahap ini, perbandingan efisiensi dilakukan dengan menggunakan dua metode yang berbeda. Metode yang pertama yaitu melalui *command line* dan metode yang kedua yaitu melalui *Graphical User Interface* (GUI). Pengujian pada setiap metode dilakukan dengan mencatat waktu menggunakan *stopwatch*. Masing-masing metode dilakukan lima kali percobaan. Setiap percobaan dilakukan tanpa kesalahan dan menggunakan “perintah” yang sesuai dengan aturan yang ada. Setelah itu hasil dari lima percobaan akan diambil nilai *average*. Hasil akhir dari setiap metode akan menentukan apakah konfigurasi dengan menggunakan *Graphical User Interface* (GUI) akan lebih cepat, tepat dan akurat dibandingkan dengan menggunakan *command line*. Pada tahap ini, administrator dapat menentukan metode yang akan digunakan untuk mempermudah dalam konfigurasi *router*. Tabel pengujian efisiensi dapat dilihat pada Tabel 4.5 sampai Tabel 4.14.

Tabel 4. 5 Pengujian Efektivitas *Input IP*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Input IP	Command Line	1	44 detik	41,4
			2	40 detik	
			3	43 detik	
			4	39 detik	
			5	41 detik	
2	Input IP	GUI	1	12 detik	10,6
			2	11 detik	
			3	10 detik	
			4	9 detik	
			5	11 detik	

Tabel 4. 6 Pengujian Efektivitas *Dynamic Routing*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Dynamic Routing	Command Line	1	79 detik	69,2
			2	71 detik	
			3	66 detik	
			4	65 detik	
			5	65 detik	
2	Dynamic Routing	GUI	1	13 detik	12,4
			2	13 detik	
			3	12 detik	
			4	12 detik	
			5	12 detik	

Tabel 4. 7 Pengujian Efektivitas *Static Routing*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Static Routing	Command Line	1	63 detik	66,8
			2	71 detik	
			3	67 detik	
			4	65 detik	
			5	68 detik	
2	Static Routing	GUI	1	11 detik	11,2
			2	11 detik	
			3	12 detik	
			4	11 detik	
			5	11 detik	

Tabel 4. 8 Pengujian Efektivitas *Show IP*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Show IP	Command Line	1	10 detik	9,6
			2	9 detik	
			3	11 detik	
			4	10 detik	
			5	8 detik	
2	Show IP	GUI	1	1 detik	0,86
			2	1 detik	
			3	0,71 detik	
			4	1 detik	
			5	0,63 detik	

Tabel 4. 9 Pengujian Efektivitas *Show Dynamic Routing*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Show Dynamic Routing	Command Line	1	12 detik	10,2
			2	11 detik	
			3	9 detik	
			4	10 detik	
			5	9 detik	
2	Show Dynamic Routing	GUI	1	0,76 detik	0,71
			2	0,67 detik	
			3	0,69 detik	
			4	0,76 detik	
			5	0,69 detik	

Tabel 4. 10 Pengujian Efektivitas *Show Static Routing*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Show Static Routing	Command Line	1	10 detik	9,8
			2	11 detik	
			3	11 detik	
			4	8 detik	
			5	9 detik	
2	Show Static Routing	GUI	1	0,80	0,7
			2	0,72	
			3	0,69	
			4	0,60	
			5	0,69	



Tabel 4. 11 Pengujian Efektivitas *Delete IP*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Delete IP	Command Line	1	44 detik	41,4
			2	40 detik	
			3	43 detik	
			4	39 detik	
			5	41 detik	
2	Delete IP	GUI	1	4	3,8
			2	4	
			3	4	
			4	3	
			5	4	

Tabel 4. 12 Pengujian Efektivitas *Delete Dynamic Routing*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Delete Dynamic Routing	Command Line	1	45 detik	44
			2	42 detik	
			3	46 detik	
			4	43 detik	
			5	44 detik	
2	Delete Dynamic Routing	GUI	1	3 detik	2,8
			2	3 detik	
			3	3 detik	
			4	2 detik	
			5	3 detik	

Tabel 4. 13 Pengujian Efektivitas *Delete Static Routing*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Delete Static Routing	Command Line	1	42 detik	39,8
			2	36 detik	
			3	40 detik	
			4	38 detik	
			5	43 detik	
2	Delete Static Routing	GUI	1	11	10,8
			2	10	
			3	11	
			4	10	
			5	12	

Tabel 4. 14 Pengujian Efektivitas *Save*

No	Konfigurasi	Jalur	Percobaan	Waktu	Average
1	Save	Command Line	1	17 detik	17,6
			2	17 detik	
			3	17 detik	
			4	19 detik	
			5	18 detik	
2	Save	GUI	1	4 detik	2,8
			2	3 detik	
			3	3 detik	
			4	2 detik	
			5	2 detik	

#### 4.4 Pembahasan

Pada Gambar 4.51 merupakan hasil dari *dynamic routing* dengan menggunakan aplikasi berbasis web. Gambar 4.51 merupakan router0 dengan menggunakan jaringan 10.10.10.0/24 pada *port fastethernet* 0/1 dan 11.11.11.0/24 pada *port fastethernet* 0/0. Router0 memiliki konfigurasi *dynamic routing* dengan *network* 10.10.10.0 dan *network* 11.11.11.0. hasil *dynamic routing* dapat dilihat pada Gambar 4.51.



```

Router# show ip route rip
 12.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R   12.0.0.0/8 [120/1] via 11.11.11.4, 00:00:20, FastEthernet0/0
router1>

Router# show ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 11 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface         Send Recv Triggered RIP Key-chain
  FastEthernet0/0     1     1 2
  FastEthernet0/1     1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    11.0.0.0
  Routing Information Sources:

```

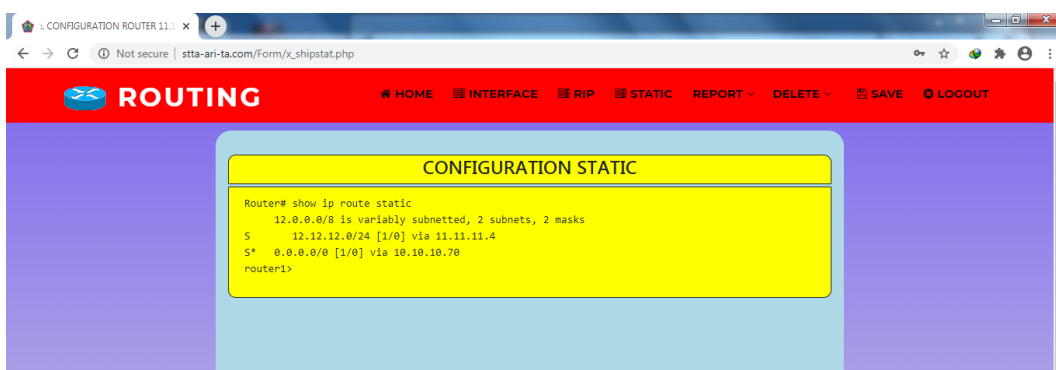
Gambar 4. 51 Hasil *Dynamic Routing* Router0

Pada Gambar 4.52 merupakan hasil dari *dynamic routing* dengan menggunakan aplikasi berbasis web. Gambar 4.52 merupakan router1 dengan jaringan 11.11.11.0/24 pada *port fastethernet* 0/0 dan 12.12.12.0/24 pada *port fastethernet* 0/1. Router1 memiliki konfigurasi *dynamic routing* dengan *network* 11.11.11.0 dan *network* 12.12.12.0. Hasil dari *dynamic routing* pada router1 dapat dilihat pada Gambar 4.52.



Gambar 4. 52 Hasil *Dynamic Routing* Router1

Gambar 4.53 merupakan hasil konfigurasi *static routing* pada router0. Pada router0 memiliki dua jaringan yang berbeda 10.10.10.0/24 pada *port fastethernet* 0/1 dan 11.11.11.0/24 pada *port fastethernet* 0/0. *Static routing* dilakukan dengan perintah `ip route [destination] [subnetmask] [nexthop]`. Hasil *static routing* dapat dilihat pada Gambar 4.53.



Gambar 4. 53 Hasil *Static Routing* Router0

Gambar 4.54 merupakan hasil dari konfigurasi *static routing* menggunakan aplikasi berbasis web. Perintah untuk melakukan *static routing* pada router1 yaitu dengan menggunakan *ip route* [destination] [subnetmask] [nextthop]. Router1 memiliki jaringan 11.11.11.0/24 pada *port fastethernet* 0/0 dan 12.12.12.0/24 pada *port fastethernet* 0/1. Hasil *static routing* dapat dilihat pada Gambar 4.54



Gambar 4. 54 Hasil *Static Routing* Router1

Pada Tabel 4.15 merupakan tabel hasil konfigurasi yang dilakukan pada *router* dengan menggunakan aplikasi berbasis web. Konfigurasi *dynamic routing* pada router0 “Berhasil” dan router1 “Berhasil”. Konfigurasi *static routing* pada router0 “Berhasil” dan pada router1 “Berhasil”. Hal ini membuktikan bahwa dengan menggunakan aplikasi berbasis web dapat melakukan konfigurasi *static* dan *dynamic* pada kedua *router*.

Tabel 4. 15 *Configuration*

NO	Configurasi	Router0	Router1
1	Dynamic Routing	Berhasil	Berhasil
2	Static Routing	Berhasil	Berhasil

Berdasarkan Tabel 4.5 sampai Tabel 4.14, dapat disimpulkan bahwa dengan menggunakan *Graphics User Interface* (GUI) dapat mempercepat dan mempermudah seorang administrator dalam melakukan konfigurasi *static routing* ataupun *dynamic routing* pada sebuah *router*. Sehingga dapat menghasilkan waktu yang efisien dalam melakukan konfigurasi *static routing* dan *dynamic routing*.

Tabel 4. 16 Nilai Efektivitas

No	Pengujian	CLI	GUI	Nilai Efektivitas (%)
1	Efektivitas Input IP	41,4	10,6	74
2	Efektivitas Dynamic Routing	69,2	12,4	82
3	Efektivitas Static Routing	66,8	11,2	83
4	Efektivitas Show IP	9,6	0,86	91
5	Efektivitas Show Dynamic Routing	10,2	0,71	93
6	Efektivitas Show Static Routing	9,8	0,7	92
7	Efektivitas Delete IP	41,4	3,8	90
8	Efektivitas Delete Dynamic Routing	44	2,8	93
9	Efektivitas Delete Static Routing	39,8	10,8	72
10	Efektivitas Save	17,6	2,8	84
<b>HASIL</b>				<b>85,4%</b>

$$Efektivitas = \frac{Command\ line - Graphics\ User\ Interface}{Command\ line} \times 100\%$$
 merupakan

rumus yang digunakan untuk mencari nilai efektivitas dari setiap bagian. Hasil dari perhitungan dapat dilihat pada bagian “Nilai Efektivitas (%)” Tabel 4.16.

Setelah mendapatkan nilai efektivitas pada setiap bagian, maka akan diambil nilai efektifitas secara keseluruhan dengan rumus  $Efektivitas_{total} = \frac{\sum Ne}{\sum P}$ . Pada Tabel 4.16 didapatkan nilai efektivitas total dengan nilai 85,4%.

Pada Tabel 4.16 dapat dijelaskan bahwa nilai efektivitas dengan menggunakan *command line* dan *Graphics User Interface* (GUI) menghasilkan nilai akhir 85,4%. Hal ini membuktikan bahwa konfigurasi menggunakan *Graphics User Interface* (GUI) lebih efektif dibandingkan dengan menggunakan *command line*.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Implementasi dan pembahasan pada “*Static Dan Routing Information Protocol (RIP) Berbasis Web Pada Cisco Router 2800 Series Memanfaatkan Application Programming Interface (API) Dari Cisco*” menghasilkan kesimpulan sebagai berikut:

1. Konfigurasi *static routing* dan *dynamic routing* dapat dilakukan dengan aplikasi berbasis web yang dihubungkan melalui API pada Cisco Router type 2800 series. Sehingga tidak diperlukan *command line* untuk melakukan konfigurasi *static routing* dan *dynamic routing*.
2. Konfigurasi *static routing* dan *dynamic routing* dapat dilakukan dari jarak jauh (*online*) dengan syarat ketersediaan jaringan internet dan web browser, ataupun jarak dekat (*offline*).
3. Aplikasi berbasis web ini terbukti dapat digunakan untuk membuat *static routing* ataupun *dynamic routing*. Sehingga dapat digunakan untuk mengatur hubungan antar komputer.
4. Berdasarkan nilai perhitungan efektivitas, *Graphics User Interface (GUI)* lebih unggul dibandingkan dengan menggunakan *command line*. *Graphics User Interface (GUI)* memiliki keunggulan dalam hal kecepatan, ketepatan dan dapat mempermudah administrator.

#### **5.2 Saran**

Saran yang dapat diberikan untuk pengembangan lebih lanjut pada aplikasi berbasis web ini adalah sebagai berikut:

1. Sistem pada aplikasi berbasis web ini tidak *focus* pada *static routing* ataupun *dynamic routing*. Akan tetapi dapat ditambahkan *routing* seperti EIGRP, OSPF dan lainnya.
2. API tidak hanya dapat digunakan pada Cisco Router Type 2800 series, melainkan dapat digunakan pada Cisco Router lainnya.



## DAFTAR PUSTAKA

- Abdullah, R. (2018). *7 In 1 Pemrograman Web Untuk Pemula*. Jakarta: PT Elex Media Komputindo.
- Ayuningtyas, A., Sudaryanto, & Cessara, D. D. (2020). *Sistem Manajemen Virtual Local Area Network (VLAN) Pada Cisco Catalyst 3750 Berbasis Web*. In *Simetris*, 11(1).
- Irawan, A., & Febriyanti, C. (2016). *Efektifitas Mathmagic Dalam Peningkatan Hasil*. In *Formatif*, 6(1).
- Jati, W. S., Nurwasito, H., & Data, M. (2018). *Perbandingan Kinerja Protocol Routing Open Shortest Path First (OSPF) Dan Routing Information Protocol (RIP) Menggunakan Simulator Cisco Packet Tracer*. 2(8).
- Miftah, Z. (2016). *Analisis Peningkatan Kinerja LAN Dengan Routing Dinamis Berbasis OSPF Single Area Dan InterVLAN Menggunakan Cisco Packet Tracer 7.1*, 9(2).
- Rinaldo, R. (2016). *Implementasi Sistem Monitoring Jaringan Menggunakan Mikrotik Router OS Di Universitas Islam Batik Surakarta*. 16(2).
- Sirika, S., & Mahajan, S. (2016). *Survey on Dynamic Routing Protocols*. 5(01).
- Sudaryanto. (2018). *Implementation Port Security For Security System Network At The Computing Laboratory Of Adisutjipto Technology College*. In Conference SENATIK STT Adisutjipto Yogyakarta, 4.
- Sudaryanto. (2018). *The Effect Of Multi Layer Switching For Data Transfer Speeds On Computer Network*. In *Compiler STT Adisutjipto*, 7(2).
- Sudaryanto, & Nurhayati, D. (2019). *Monitoring Interfaces FastEthernet On Cisco Catalyst 3750 To Ensure Use Of The Security Computer Network In STTA Computing Laboratories*. In Conference SENATIK STT Adisutjipto Yogyakarta, 5.
- Sulaiman, O. K. (2016). *Analisis Sistem Keamanan Jaringan Dengan Menggunakan Switch Port Security*. 1(1).
- Sulaiman, O. K., Siregar, A. M., Nasution, K., & Haramaini, T. (2018). *Bellman Ford algorithm-in RoutingInformation Protocol*. 1007.
- Sulistiono, H. (2018). *Coding Mudah Dengan CodeIgniter, JQuery, Bootstrap Dan Datatable*. Jakarta: PT Elex Media Komputindo.

Widodo, S. (2012). *Pemantauan Jaringan Komputer dengan DNS Server Berbasis Routing Statis Menggunakan Wireshark*. In Polines, 1(2).