**Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python**

**Rheza Adhyatmaka Wiryawan**

15/386772/SV/10158

**Departemen Teknik Elektro Dan Informatika**

**Sekolah Vokasi - Universitas Gadjah Mada, Yogyakarta**

rheza.adhyatmaka.w@mail.ugm.ac.id

***Intisari−*** Banyaknya perangkat jaringan yang terpasang merupakan sebuah tantangan bagi perusahaan. Semua perangkat yang terpasang perlu dilakukan pemeliharaan dan konfigurasi jaringan secara berkala agar jaringan dapat berjalan dengan baik dan tidak menghambat proses bisnis perusahaan. Pada cara tradisional administrator jaringan perlu masuk ke sistem perangkat secara satu persatu sehingga akan memakan waktu yang lama dan kurang efisien. Otomatisasi jaringan merupakan solusi untuk melakukan pekerjaan-pekerjaan yang rumit dan repetitif tersebut. Pekerjaan yang bersifat repetisi seperi *backup* konfigurasi, *restore* konfigurasi, dan lain-lain dapat dilakukan oleh skrip. Pada tugas akhir ini membuat sistem aplikasi otomatisasi administrasi jaringan berbasis web dengan memanfaatkan *library* utama paramiko sebagai penghubung dan otomatisasi jaringan dari *server* ke perangkat jaringan menggunakan protokol SSHv2 dan *framework* Django sebagai pengembangan web. Pengujian yang dilakukan terhadap aplikasi menggunakan metode *Black-box testing*. Aplikasi ini dapat dimanfaatkan sebagai otomatisasi jaringan dalam hal konfigurasi, *maintenance* seperti *backup*, *restore* dan lain-lain secara terpusat sehingga akan lebih termanajemen lebih baik.

**Kata Kunci : Otomatisasi*,* Python*,* Django*,* Paramiko*,* Manajemenjaringan*.***

**I. PENDAHULUAN**

Pada era transformasi digital yang berkembang pesat saat ini perusahaan yang dapat mengerjakan suatu tugas dengan lebih cepat dan efisien merupakan suatu nilai lebih bagi perusahaan itu sendiri. Pada perusahaan yang bergerak di bidang jaringan internet saat ini memiliki banyak perangkat jaringan seperti *router*, *switch*, *server* yang terpasang pada topologi jaringan yang digunakan. Banyaknya perangkat jaringan yang terpasang merupakan sebuah tantangan bagi perusahaan. Semua perangkat yang terpasang perlu dilakukan pemeliharaan, perawatan, dan konfigurasi jaringan secara berkala agar jaringan dapat berjalan dengan baik dan tidak menghambat proses bisnis perusahaan. Untuk melakukan semua hal tersebut admin jaringan perlu masuk ke sistem perangkat secara satu persatu sehingga akan memakan waktu yang lama dan kurang efisien.

Metode tradisional yang digunakan untuk konfigurasi perangkat jaringan sangat memakan waktu dan juga harus mempertimbangkan pengetahuan mengenai *syntax* atau konfigurasi dasar dari setiap *vendor* karena setiap *vendor* memiliki *syntax* yang berbeda. konsep *Software Defined Networks* (SDN) mencoba menghilangkan ketergantungan mengenai perbedaan *vendor* dengan menggunakan standar protokol seperti *OpenFlow*, tetapi jaringan tradisional non-SDN tetap perlu diperhatikan. (Macedo, dkk., 2015).

Otomatisasi jaringan merupakan solusi untuk melakukan pekerjaan-pekerjaan yang rumit tersebut dan dapat diimpelementasikan ke perangkat lama yang mendukung SSH sehingga pekerjaan bisa diselesaikan jauh lebih cepat dan juga efisien dalam pemeliharaan jaringan dengan prosedur yang lebih mudah diikuti dan diimplementasikan di dalam jaringan yang berskala besar. Dengan otomatisasi jaringan dapat meringankan beban kerja administrator maupun *network engineer*. Pekerjaan yang bersifat repetisi seperi *backup* konfigurasi, *restore* konfigurasi, menambahkan maupun mengubah konfigurasi yang bersifat repetitif lainnya dapat dilakukan menggunakan skrip yang tentunya akan lebih efisien. Selain itu dengan otomatisasi jaringan dapat mengurangi *resource* yang dibutuhkan dalam hal sumber daya manusia untuk melakukan pekerjaan administrasi jaringan, dan dengan otomatisasi jaringan dapat menghindari *error* yang disebabkan oleh kesalahan konfigurasi manusia.

Pada beberapa perangkat jaringan seperti Mikrotik sudah disediakan fitur *scripting* untuk melakukan konfigurasi, namun skrip tersebut tidak terpusat sehingga perlu menerapkannya secara satu persatu ditiap perangkat sehingga akan memakan waktu yang cukup lama. Selain itu terdapat kendala seperti perbedaan *syntax* jika seorang admin jaringan akan mengkonfigurasi perangkat jaringan yang berbeda *vendor*. Oleh karena itu diperlukan sebuah sistem secara terpusat yang mampu melakukan otomatisasi pada perangkat jaringan sehingga pekerjaan akan lebih cepat dan efisien, salah satu caranya dengan menggunakan bahasa pemrograman python.

**II. RUMUSAN MASALAH**

Rumusan permasalahan pada penelitian ini adalah bagaimana membuat sistem otomatisasi administrasi pada jaringan secara terpusat dan dapat digunakan untuk mengkonfigurasi perangkat jaringan dengan *vendor* yang berbeda melalui sistem yang berbasis website.

**III. TUJUAN PROYEK AKHIR**

Adapun tujuan dari penelitian ini adalah menghasilkan sebuah sistem aplikasi berbasis website untuk melakukan administrasi jaringan berupa konfigurasi *routing*, *virtual* LAN (VLAN), *backup* dan *restore* konfigurasi perangkat jaringan dengan vendor yang berbeda sehingga pekerjaan administrasi jaringan dapat dilakukan dengan lebih cepat dan efisien.

**IV. TINJAUAN PUSTAKA**

Otomatisasi jaringan menggunakan logika pemrograman untuk memanajemen sumber dan layanan jaringan, hal tersebut dapat memungkinkan teknisi jaringan secara cepat dalam mengkonfigurasi dan mengintegrasikan intrastruktur jaringan (layer 1 – 3) dan layanan aplikasi (layer 4 – 7) (Redhat., 2018). Pada penelitian oleh Paul MIHĂILĂ, Titus BĂLAN, Radu CURPEN dan Florin SANDU (2017) yang berjudul "*Network Automation and Abstraction using Python Programming Methods*" menunjukkan pentingnya otomatisasi dalam jaringan konvensional yang tidak mendukung protokol *OpenFlow* SDN. Pada jaringan konvensional yang masih menggunakan perangkat jaringan model lama yang diproduksi oleh beberapa *vendor* yang berbeda sangat sulit untuk dikendalikan atau dikontrol, karena perbedaan *syntax* dasar dan belum mendukung protokol *OpenFlow* untuk diterapkan jaringan SDN. Pada penelitian ini dilakukan otomatisasi pembuatan VLAN pada 3 *switch* Cisco yang dikontrol menggunakan skrip python. Terdapat dua *library* yang digunakan yaitu paramiko dan netmiko, dimana terdapat beberapa perbedaan antara penggunaan kedua *library* tersebut. Pada *library* netmiko penggunaan untuk koneksi ke *switch* menggunakan ConnectHandler yang juga menggunakan SSH dibagian *backend* nya, selain itu netmiko dapat menentukan tipe perangkat yang akan diatur. Dengan menggunakan Phyton, administrator jaringan tidak perlu mengkonfigurasi sendiri setiap perangkat, administrator hanya perlu membuat infrastruktur yang tetap dan dengan menerapkan scripting otomatisasi. Otomatisasi dapat menggunakan Phyton dan koneksi Secure Shell.

Ahmad Rosid Komarudin (2018) pada buku berjudul "Otomatisasi Administrasi Jaringan Dengan Script Python" menerapkan otomatisasi menggunakan *library* Paramiko, Netmiko, Pyntc, Napalm, dan Ansible dan diterapkan pada perangkat Cisco. Semua *tools* yang digunakan tersebut dapat diimplementasikan ke perangkat Cisco untuk otomatisasi jaringan. Dari percobaan yang telah dilakukan tools Napalm dan Ansible merupakan tools dengan fitur yang paling *powerfull* untuk diterapkan di perangkat Cisco. Ansible memiliki modul tersendiri untuk jaringan dan mendukung banyak vendor perangkat jaringan. Sedangkan dukungan vendor Napalm masih terbatas seperti Arista EOS, Cisco IOS, Cisco IOS-XR, Cisco NX-OS dan Juniper JunOS.

Otomatisasi jaringan yang di implementasikan ke Router OS Mikrotik menggunakan Ansible dilakukan I Made Bayu Swastika dan I Gede Oka Gartria Atitama (2017) pada jurnal yang berjudul “Otomatisasi Konfigurasi Mikrotik Router Menggunakan *Software* Ansible”. Otomatisasi konfigurasi yang dilakukan yaitu mengatur *bandwidth* menggunakan fitur *Queue Tree* pada Mikrotik. Pada Ansible untuk menjalankan otomatisasi dengan mengeksekusi *file* yml yang merupakan *file* dengan isi perintah konfigurasi Mikrotik yang akan dijalankan. Selain itu ditambahkan modul RouterOS API pada Ansible agar *software* Ansible dapat terhubung dan mengkonfigurasi router Mikrotik melalui API. Untuk pengujian menggunakan metode *black box testing*. Hasil dari penelitian tersebut adalah *software* Ansible dapat digunakan untuk mengkonfigurasi router Mikrotik. Ansible dapat mengkonfigurasi beberapa *queue tree* secara berurutan dalam sekali eksekusi *file* dan membutuhkan waktu proses kurang lebih 1 menit untuk 19 konfigurasi *queue tree*.

Penggunakan *library* Paramiko yang diimplementasikan ke perangkat Cisco dilakukan oleh Eric Chou (2017) pada buku yang berjudul "*Mastering Python Networking Second Edition*". Paramiko merupakan implementasi Phyton dari protokol SSHv2, paramiko menyederhanakan interaksi SSHv2 dengan perangkat yang diremote. Tidak seperti pxssh, paramiko hanya fokus pada protokol SSHv2 dan menyediakan operasi pada sisi klien dan server. Percobaan yang dilakukan yaitu dengan membuat skrip dengan hasil interaktif *shell* untuk beberapa fungsi. Fungsi yang dibuat yaitu menampilkan versi dari IOS Cisco, menampilkan konfigurasi yang sedang berjalan pada IOS Cisco. Selain itu juga mengkombinasikan dengan berkas *JavaScript Object Notation* (JSON) sebagai tempat informasi data perangkat yang akan diremote. Penggunaan tipe data JSON akan sangat mudah di translasikan ke tipe data dictionary pemrograman Phyton.

Penelitian berjudul “*Network Automation using Ansible for Cisco Routers Basic Configuration*” yang dilakukan oleh Jonathan Wijaya (2018) melakukan otomatisasi terhadap router Cisco. Terdapat 3 file utama di directory Ansible yaitu file hosts, ansible.cfg, dan *file* Ansible Playbook. *File* ansible.cfg merupakan *file* konfigurasi yang digunakan untuk mengatur parameter pada Ansible. Otomatisasi yang dilakukan yaitu konfigurasi *hostname*, alamat ip pada *interface* *router* dan *loopback*, dan konfigurasi *routing* OSPF. Otomatisasi dilakukan di dua *router* Cisco dan hasilnya kedua *router* tersebut dapat dikonfigurasi melalui Ansible.

**V. LANDASAN TEORI**

1. Konsep Jaringan Komputer

Jaringan komputer adalah hubungan dua simpul yang umumnya berupa komputer atau yang lebih ditujukan untuk melakukan pertukaran data atau untuk berbagi perangkat lunak, perangkat keras dan bahkan berbagi kekuatan pemrosesan. Jaringan komputer ini di dalamnya berisi kumpulan beberapa komputer dan perangkat lain seperti *router, switch,* dan sebagainya yang saling terhubung satu sama lain melalui media perantara. Media perantara ini dapat berupa media kabel ataupun tanpa kabel. Untuk saling dapat bertukar *data*, maka dari satu komputer ke komputer lain harus saling terhubung. Perangkat-perangkat utama yang membangun jarigan komputer tersebut antara lain, *router, switch, host, firewall* dan lainnya. Tujuan utama dibangunnya suatu sistem jaringan komputer adalah untuk membawa data dan informasi dari sisi pengirim menuju ke penerima secara cepat dan tepat tanpa adanya kesalahan melalui media transmisi atau media komunikasi tertentu (Oetomo, 2003).

1. Otomatisasi Jaringan

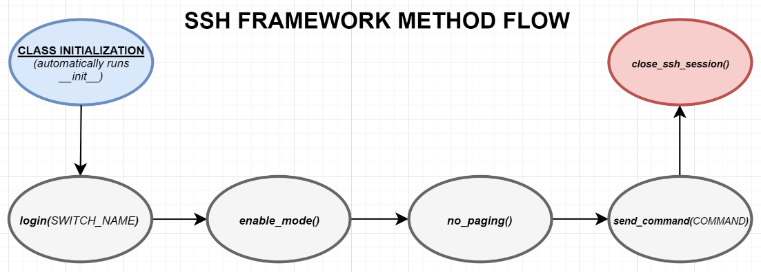
Otomatisasi jaringan atau *network automation* merupakan sebuah proses pengotomatisasian konfigurasi, manajemen, dan operasional pada suatu jaringan komputer (Ulinic dan House, 2018). Salah satu manfaat otomatisasi jaringan yaitu perubahan konfigurasi yang lebih cepat di perangkat jaringan dengan jumlah yang besar. Otomatisasi tidak hanya terbatas pada infrastruktur yang besar saja tetapi juga dapat diimplentasikan pada jaringan dengan infrastruktur yang berskala lebih kecil, dimana hal tersebut dapat digunakan untuk memastikan konsistensi konfigurasi antara perangkat satu dengan perangkat lainnya dan mengurangi faktor kesalahan manusia. Sebagaimana yang diketahui *human errors* atau kesalahan yang diakibatkan oleh manusia merupakan salah satu alasan terbesar di hampir setiap isu permasalah termasuk pada masalah keamanan, *downtime, unavailability* dan lain sebagainya pada perangkat jaringan.

Otomatisasi jaringan merupakan bentuk implementasi dari kombinasi perangkat keras dan perangkat lunak yang secara otomatis menjalankan tugas berulang atau repetisi dalam jaringan. Bahasa scripting secara luas digunakan oleh administrator jaringan dan administrator sistem untuk mengotomatisasikan tugas atau pekerjaan. Bahasa pemrograman yang biasa digunakan oleh administrator untuk proses otomatisasi adalah bahasa Phyton. Selain mudah diterapkan, bahasa Python memiliki beragam *library* atau pustaka yang mendukung otomatisasi pada jaringan.

1. Paramiko

Paramiko merupakan *library* atau pustaka Python yang mengimplementasi protokol SSHv2, menyediakan fungsionalitas klien dan *server*. Sementara itu memanfaatkan ekstensi Python C untuk kriptografi tingkat rendah (Kriptografi), Paramiko sendiri menggunkan antarmuka Python dibagian konsep jaringan SSH.

High-level *client* API dimulai dengan pembuatan objek SSH *Client*. Untuk kontrol langsung dengan melewati *socket* ke *transport*, dan menggunakan *start\_server* atau *start\_client* untuk bernegosiasi *dengan remote host* baik sebagai *server* atau klien. Pada sisi klien bertanggung jawab untuk mengautentikasi menggunakan kata sandi *atau private key*, dan memeriksa *key* atau kunci *host server*. Sebagai *server* bertanggung jawab untuk menentukan pengguna, kata sandi, dan kunci mana yang diizikan, dan jenis saluran atau *channel* apa yang diizinkan. Setelah selesai, kedua pihak ( klien dan *server* ) akan mendapatkan saluran atau *channel* yang dikontrol, pengiriman dan penerimaan data melalui sesi yang dienkripsi (Forcier, 2018). Pada gambar 4.1 berikut merupakan metode aliran framework SSH.



Gambar 5.1. Metode aliran SSH *Framework*

Dimulai dengan menginisialisasi beberapa kondisisi seperti informasi login ke perangkat, dilanjutkan dengan login ke perangkat yang dituju dan mengaktifkan *privilege mode*. Menjalankan perintah “term len 0” untuk menampilkan hasil dalam satu halaman, selanjutnya mengirimkan perintah atau *syntax* ke perangkat tujuan dan diakhiri dengan menutup sesi SSH.

1. Django

Berdasarkan dari halaman resmi website Django (https://www.djangoproject.com/,2019), Django merupakan sebuah web *framework* yang *open source* dan berbasis Python. Web *framework* merupakan sebuah kerangka kerja yang berfungsi mempermudah pengembang dalam mengembangkankan projek websitenya. Django menggunakan konsep *Model View* and *Template* (MVT). Pada gambar 4.2 berikut merupakan cara kerja dari konsep *Model View and Template*.



Gambar 5.1. Konsep *Model, View, Template*

*Model* merupakan bagian yang digunakan untuk berinteraksi dengan database. *Template* adalah bagian yang berhubungan mengenai tampilan bagian depan atau *front-end* seperti HTML, XML dan lain sebagainya. Sedangkan *view* merupakan bagian yang menghubungkan *model* dengan *template* yang berisikan logika pengolah data dari *model* dan ditampilkan di *template*.

1. Rapid Apllication Development

*Rapid Application Development* (RAD) merupakan metodologi pengembangan perangkat lunak yang berfokus terhadap pembuatan aplikasi dalam waktu yang sangat singkat.

Model RAD secara umum mengurangi penekanan pada tahap perencanaan dan meningkatkan penekanan pada tahap pengembangan perangkat lunak. Proses RAD memungkinkan pengembang dalam menciptakan sistem fungsional secara utuh dalam periode waktu yang sangat pendek (60 hari hingga 90 hari).

Menurut Kendall (2002), model RAD terdapat tiga fase yang melibatkan penganalisis dan pengguna dalam tahapan penilaian, perancangan, dan penerapan. Ketiga fase tersebut adalah requirement planning, RAD design workshop, dan implementation.

1. Black-Box Testing

*Black-Box Testing* merupakan metode pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak tanpa memperhatikan struktur internal dari pearngkat lunak yang diuji. Metode ini dapat digunakan dalam setiap tahapan pengujian seperti *unit testing, integration testing, system testing,* dan *acceptance testing* (Patton, 2001).

**VI. HIPOTESIS**

Mengacu pada uraian-uraian sub bab terdahulu, seperti perumusan masalah, tinjauan pustaka, dan landasan teori, maka dapat dikemukakan hipotesis sebagai berikut :

1. Library atau pustaka paramiko dapat digunakan sebagai otomatisai administrasi perangkat jaringan dalam hal konfigurasi ip *routing*, VLAN, *backup*, *restore* konfigurasi, dan lain-lain, sedangkan dalam pembuatan web menggunakan *framework* Django sehingga sistem aplikasi dapat diakses dengan mudah dan penggunaan server agar dapat diakses secara terpusat.
2. Aplikasi sistem otomatisasi administrasi dapat diterapkan ke perangkat jaringan dengan vendor yang berbeda.

**VII. METODOLOGI PENELITIAN**

A. Alat dan Bahan

Dalam melaksanakan penelitian ini, terdapat beberapa perangkat keras dan perangkat lunak yang dibutuhkan guna menunjang penelitian yang akan dilakukan. Adapun beberapa perangkat tersebut yaitu:

**I. Perangkat Keras**

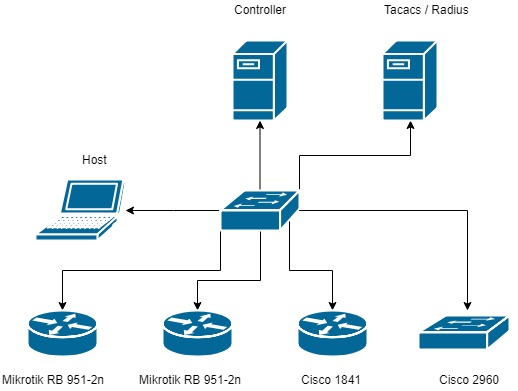
1. 1 PC / Laptop
2. 1 unit Server
3. 2 unit router Mikrotik RB951-2n
4. 1 unit router Cisco 1841
5. 1 unit router Cisco 2960
6. 1 unit switch unmanageable

**II. Perangkat Lunak**

1. Ubuntu
2. Python 3.6.7
3. Paramiko 2.4
4. Django 2.1.7
5. GNS3 2.1
6. Vmware 14.1.0

B. Rancangan Topologi

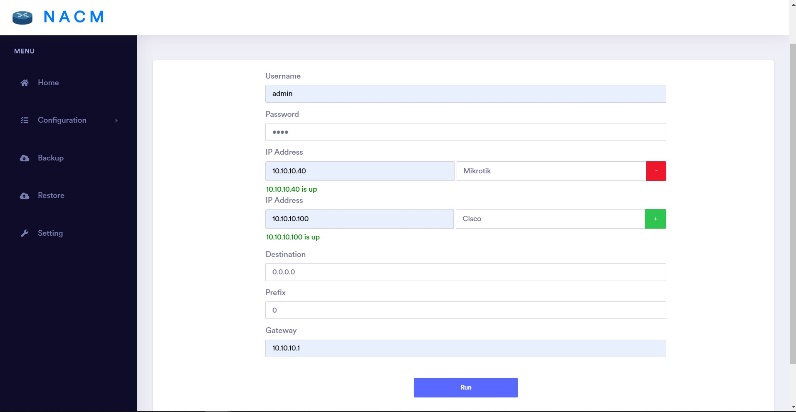
Topologi yang digunakan untuk pengujian dalam pengembangan aplikasi otomatisasi administrasi jaringan berbasis website menggunakan bahasa pemrograman Python adalah sebagai berikut :



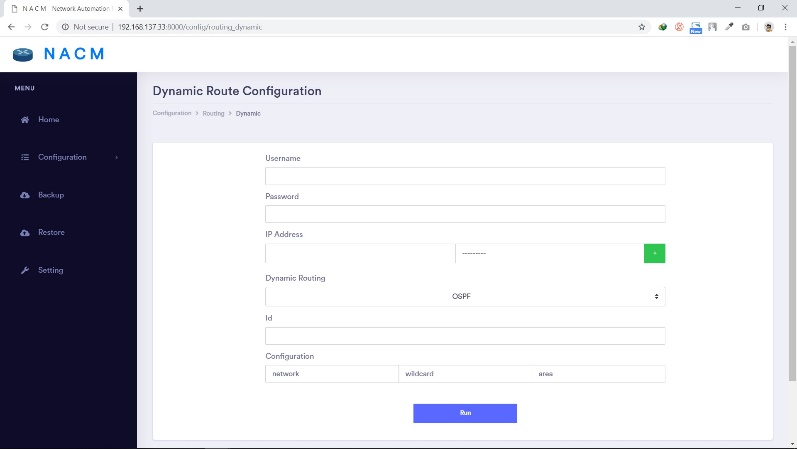
Gambar 6.1. Gambar rancangan topologi

C. Rancangan Antarmuka

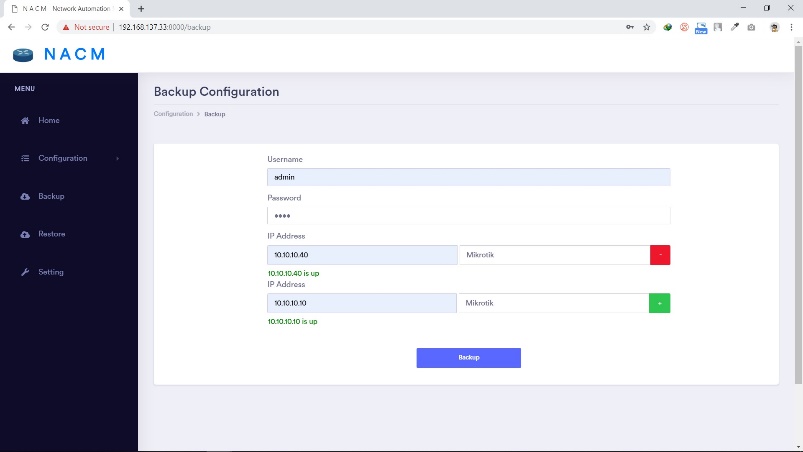
Antarmuka Aplikasi dirancang berdasarkan kebutuhan interaksi pengguna untuk melakukan otomatisasi jaringan dengan mengedepankan kemudahan dan kenyamanan pengguna (*User Interface & User Experience*). Berikut adalah rancangan beberapa antarmuka aplikasi:



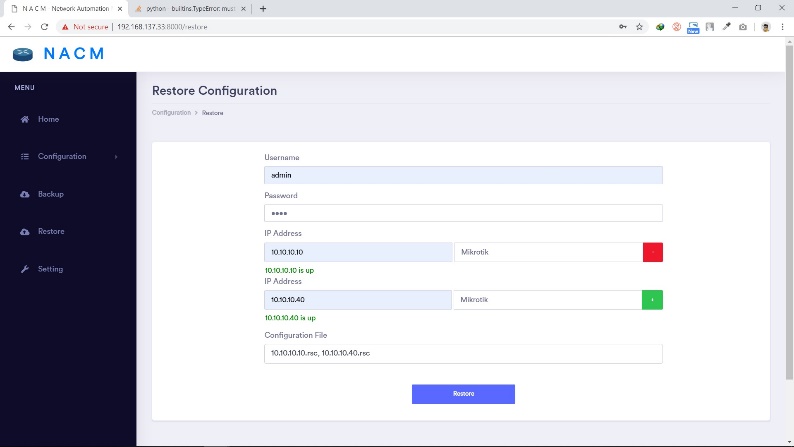
Gambar 7.1. *Mockup menu static routing*



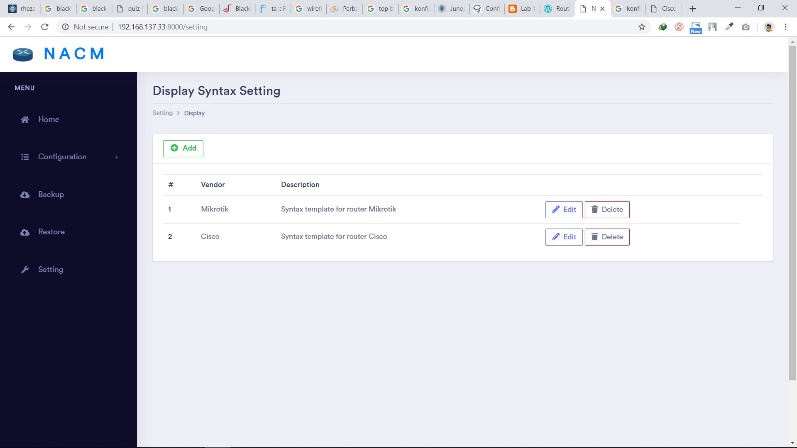
Gambar 7.2. *Mockup menu dynamic routing*



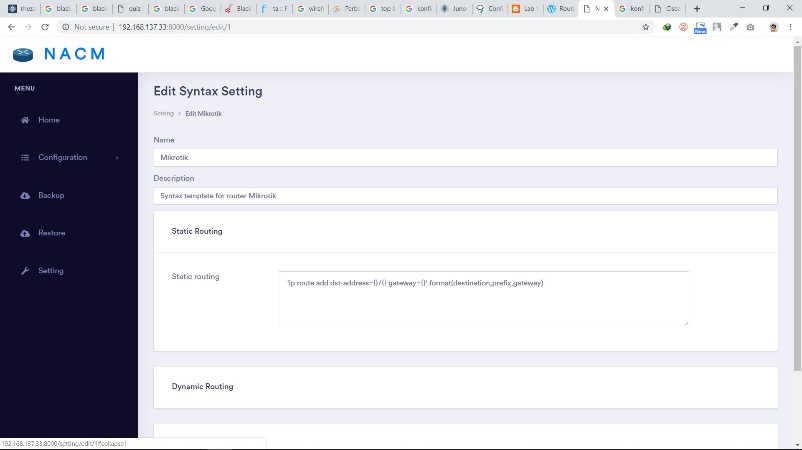
Gambar 7.3. *Mockup menu backup*



Gambar 7.4. *Mockup menu restore*



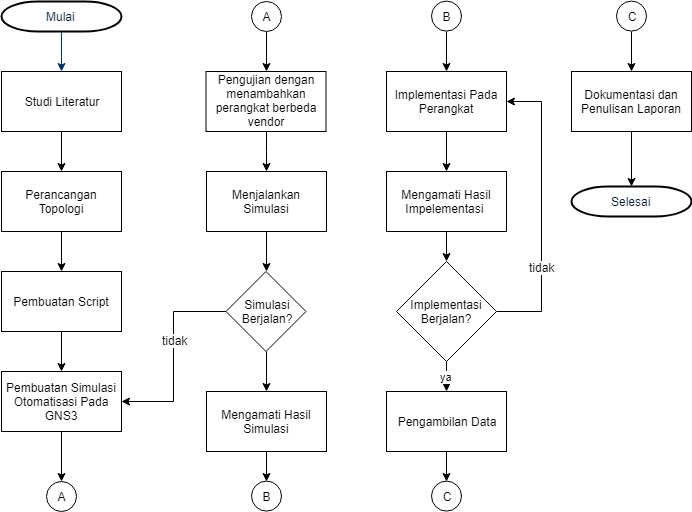
Gambar 7.5. *Mockup menu setting vendor*



Gambar 7.5. *Mockup menu edit setting vendor*

D. Metode Penelitian

Metode penelitian yang dilakukan penulis dalam penelitian ini dapat dilihat pada *flow chart* berikut:



Gambar 7.6. *Flowchart* Metode Penelitian

**VIII. ANALISIS HASIL**

Hasil dari penelitian yang dilakukan adalah berupa sistem aplikasi otomatisasi administrasi jaringan berbasis website. Fungsi otomatisasi yang dapat dilakukan yaitu berupa konfigurasi *routing static*, *dynamic* yang terdiri dari routing Ripv1, Ripv2, OSPF, dan BGP. Selain itu konfigurasi penambahan VLAN, melakukan *backup* konfigurasi dan *restore* konfigurasi pada setiap perangkat jaringan. Pengujian sistem aplikasi diterapkan ke perangkat jaringan dengan *vendor* yang berbeda dan menggunakan metode Black-box testing, dimana pengujian dibuat sebuah tabel *test case* untuk menguji setiap fungsi dari sistem aplikasi apakah berjalan dengan baik atau tidak,

**DAFTAR PUSTAKA**

Mihăilă, P., Bălan, T., Curpen, R. dan Sandu, F. 2017. *Network Automation and Abstraction using Python Programming Methods*. Electronics and Computer Department. Transilvania University. Jurnal.

Swastika, I., Atitama, G. 2017. Otomatisasi Konfigurasi Mikrotik *Router* Menggunakan *Software* Ansible. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Udayana. Jurnal.

Wijaya, J. 2018. *Network Automation using Ansible for Cisco Router Basic Configuration.* Teknik Elektro dan Informatika. Institut Teknologi Bandung. Jurnal.

Golinski, A. 2017. *Automating Network System Configurations for Vendor-Specific Network Elements.* KTH Royal Institute of Technology.

Puumala, J. 2017. *Automation of Router Configuration.* Haaga-Helia University.

Macedo, D., Guedes, D.,Vieira, M., Nogueira, M. 2015. "*Programmable Networks—From Software-Defined Radio to Software-Defined Networking*," in IEEE Communications Surveys & Tutorials, vol. 17, no. 2, pp. 1102-112.

Komarudin, A. 2018. Otomatisasi Administrasi Jaringan Dengan *Script* Python. Jasakom. Jakarta.

Chou, E. 2017. *Mastering Python networking*. 2nd ed. packt. Birmingham.

Edelman, J., Oswalt, M. dan Lowe, S. 2015. *Network programmability and automation*. O’Reilly Media.

Ulinic, M. (2016). *Network Automation with Salt and NAPALM*. [online] Nanog.org. <https://www.nanog.org/sites/default/files/NANOG68%20Network%20Automation%20with%20Salt%20and%20NAPALM%20Mircea%20Ulinic%20Cloudflare%20(1).pdf>. Diakses 31 November 2018.

Soleman, A. 2018. Implementasi *Access Control List* dan *Dynamic Control List* Berbasis Aplikasi *Desktop* Pada *Software-Defined Network* Menggunakan *Floodlight Controller*. Sekolah Vokasi. Universitas Gadjah Mada. Tugas Akhir

Edityomurti, S. 2018. Pengembangan Aplikasi Manajemen *Flow* Berbasis Android Pada Jaringan *Software Defined Network* Menggunakan *Opendaylight Controller*. Sekolah Vokasi. Universitas Gadjah Mada. Tugas Akhir

Agusriandi, Saputra, A., Raehan, S., Asmar, K. 2018. Analisis Rancangan Jaringan Komputer dalam Mendukung *E-Government*. Bogor: CV Kitami Software-Defined Networking: The New Norm for Networks. ONF White Paper. April 13, 20122.

Casey, J. 2018. *MikroTik Automated MPLS L3VPN Lab*. Linkendin. <https://www.linkedin.com/pulse/mikrotik-automated-mpls-l3vpn-lab-jedadiah-casey/>. Diakses 31 November. 2018.

Hermawan, E. 2017. Aplikasi Radius atau Tacacs+ Server Untuk Mengatur dan Mengontrol Penggunaan Jaringan Komputer di Politeknik Negeri Balikpapan. Jurusan Teknik Elektronika. Politeknik Negeri Balikpapan. Tugas akhir.

Putra, M. 2018. Analisis dan Perancangan Sistem Informasi Pelayanan Bagian Teknologi Informasi (Studi Kasus : PT PLN (Persero) Area Ciputat). Fakultas Sains dan Teknologi. UIN Syarif Hidayatullah Jakarta.

Kadir, A. 2003. Jaringan Komputer. Andi. Yogyakarta.

Oetomo, B. 2003. Konsep dan Perancangan Jaringan Komputer. Andi. Yogyakarta.

Supardi, Y. 2017. Semua Bisa Menjadi Programmer Python Basic. Elex Media Komputindo. Jakarta.

Boehm, B.W. 1987. A Spiral Model of Software Development and Enchancement. Jurnal. University of Zagreb, Zagreb, Croatia.

Forcier, J. 2018. Paramiko’s documentation. [Online]. http://docs.paramiko.org/en/2.4/. Diakses pada tanggal 1 Desember 2018.

Anwar, S. 2018. Penerapan Django Rest Framework dan Teknologi Otentikasi Oauth 2.0 Untuk Sistem Informasi Akademik Universitas Lampung Versi Android. Fakultas Matematika Dan Ilmu Pengetahuan Alam. Universitas Lampung. Skripsi.

Saraswati, D. 2014. LKP : Implementasi User Interface Zorin OS Untuk Bidang Pendidikan. Stikom Surabaya. Undergraduate thesis

Hermawan, E. 2017. Aplikasi Radius Atau Tacacs + Server Untuk Mengatur Dan Mengontrol Penggunaan Jaringan Komputer Di Politeknik Negeri Balikpapan. Teknik Elektronika. Politeknik Negeri Balikpapan. Tugas Akhir.

Wicaksono, P. 2011. METODE AUTENTIKASI PADA TACACS+ SERVER. Teknik Elektro. Universitas Diponegoro Semarang. Jurnal.

R, Naz dan M, Khan. 2015. *Rapid Application Development Techniques : A Critical Review*. International Journal of Software Engineering and Its Application, vol. 9 No. 11, pp. 163-176.

Kendall, J.E. dan Kendall, K.E. 2010. Analisis dan Perancangan Sistem. Jakarta: Indeks.

Patton, Ron. 2003. *Software Testing*. SAMS Publishing.