

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Jaringan

2.1.1 Pengertian Jaringan Telekomunikasi

Jaringan telekomunikasi adalah rangkaian perangkat telekomunikasi yang digunakan dalam melakukan aktivitas telekomunikasi. Jaringan telekomunikasi merupakan bagian dari kegiatan penyelenggaraan telekomunikasi yang telah diatur oleh Kementerian Komunikasi dan Informatika Republik Indonesia dalam Undang-Undang Telekomunikasi nomor 36 tahun 1999. [8]

Jaringan telekomunikasi terdiri atas dari tiga bagian utama. Bagian-bagian utama tersebut diantaranya adalah perangkat transmisi, perangkat penyambung, dan terminal. Perangkat transmisi bertugas menyampaikan informasi dari suatu tempat ke tempat yang lain, baik dekat maupun jauh. Media transmisinya dapat berupa kabel, serat optik, maupun udara, tergantung jarak dari tempat-tempat yang dihubungkan serta tergantung pada beberapa banyak tempat yang saling dihubungkan. Perangkat penyambungan sering dilakukan dengan menggunakan sistem manual bila diperlukan, dengan memanggil seorang operator yang bertugas menyambungkan pengguna dengan pengguna lain yang diinginkannya. Terminal adalah peralatan yang bertugas mengubah sinyal informasi asli (suara manusia atau lainnya) menjadi sinyal elektrik atau elektromagnetik atau cahaya. [9]

Perangkat telekomunikasi bertugas menghubungkan pengguna dengan pengguna lain. Kedua pengguna ini dapat berdekatan tetapi bisa berjauhan. Kalau menilik arti harfiah dari telekomunikasi (tele = jauh, komunikasi = hubungan dengan pertukaran informasi), berarti teknik telekomunikasi dikembangkan manusia untuk menebus perbedaan jarak yang jauhnya tidak terbatas menjadi perbedaan waktu yang sekecil mungkin. Perbedaan jarak yang jauh dapat ditempuh dengan waktu yang sekecil mungkin dengan cara mengubah semua bentuk informasi yang ingin disampaikan oleh manusia kepada yang lainnya menjadi bentuk gelombang elektromagnetik.

Gelombang elektromagnetik dapat bergerak dengan kecepatan yang sangat tinggi, yakni di ruang hampa adalah seratus ribu km per detik. Jaringan telekomunikasi adalah segenap perangkat telekomunikasi yang dapat menghubungkan penggunanya (umumnya manusia) dengan pengguna lain, sehingga kedua pengguna tersebut dapat saling bertukar informasi (dengan cara bicara, menulis, menggambar atau mengetik) pada saat itu juga.

2.1.2 Jenis-Jenis Jaringan

Dalam jaringan telekomunikasi, terdapat dua jenis jaringan yang berbeda yang dibedakan berdasarkan:

a. Berdasarkan Jenis Transmisi

Dalam mempelajari jenis jaringan komputer terdapat beberapa klasifikasi yang cukup penting yaitu teknologi transmisi dan jarak. Secara teori, jaringan komputer dibagi berdasarkan transmisi dan jarak. Terdapat dua jenis jaringan berdasarkan teknologi transmisi, yaitu: jaringan *broadcast* dan jaringan *point-to-point*.

Pada umumnya jaringan yang lebih kecil dan terlokalisasi secara geografis cenderung memakai broadcasting, sedangkan jaringan yang lebih besar menggunakan point-to-point.

b. Berdasarkan Geografis

Alternatif lain dalam melakukan klasifikasi sebuah jaringan adalah berdasarkan pada cakupan geografis sebuah jaringan. Berikut ini klasifikasi jaringan berdasarkan geografis:

- *Personal Area Network* (PAN)
- *Local Area Network* (LAN)
- *Metropolitan Area Network* (MAN)
- *Wide Area Network* (WAN)
- *Wireless LAN* (WLAN)
- Internet

LAN, MAN, WAN, dan internet bisa dikatakan sebagai *true network*, artinya komputer-komputer yang berkomunikasi dengan cara bertukar data/pesan melalui kabel yang lebih panjang. [10]

2.1.3 Topologi Jaringan

Jalur telepon, atau media lain, yang digunakan untuk pengiriman data merupakan media yang sangat mahal. Untuk mengatasinya, sejumlah teknik digunakan supaya sebanyak-banyaknya data dapat dikirimkan lewat sesedikit mungkin jalur telepon. Kompromi harus diperhitungkan antara biaya jalur, kecepatan pengiriman, dan kebutuhan setiap sirkit, sehingga muncul sejumlah konfigurasi jaringan dan teknik-teknik yang digunakan. Alternatif-alternatif utama yang perlu dipertimbangkan dalam merancang suatu jaringan adalah: sambungan *dial-up* melalui PSTN, *leased-line* pribadi titik-ke-titik, kilostream dan megastream, dan pensaklaran paket (*pocket switching*). Dua alternatif di atas menggunakan isyarat analog pada frekuensi suara, dan memerlukan modem pada kedua ujung jalur transmisi. Tiga alternatif lain menggunakan isyarat digital sehingga tidak perlu menggunakan modem. Dalam beberapa hal, kecuali untuk jaringan yang paling kecil, jaringan campuran berukuran sedang merupakan pecahan yang paling efisien. (Insap Santosa, 1995: 237) [11]

Cara yang digunakan untuk menyambung sejumlah terminal dan komputer ke dalam suatu jaringan disebut topologi. Topologi jaringan menyatakan susunan komputer secara fisik dalam suatu jaringan. Saat ini sejumlah topologi yang berbeda sering digunakan. (Insap Santosa, 1995: 155).

Terdapat macam-macam topologi jaringan, antara lain:

a. Topologi Bintang (*Star*)

Pada topologi ini terdapat komponen yang bertindak sebagai pusat pengontrol. Semua simpul yang hendak berkomunikasi selalu melalui pusat pengontrol tersebut. Dalam hal ini, pusat pengontrol berupa hub atau switch. (Abdul Kadir, 2013: 283) [12]

b. Topologi Cincin (Ring)

Topologi cincin mirip dengan topologi bus. Informasi dikirim oleh sebuah komputer akan dilewatkan ke media transmisi, melewati satu komputer berikutnya. (Abdul Kadir, 2013: 284)

c. Topologi Bus

Pada topologi ini semua sinyal (umumnya komputer) dihubungkan melalui kabel yang disebut bus. Kabel yang digunakan adalah kabel *coaxial*. Jika seseorang, pemakai mengirimkan pesan ke seseorang pemakai lain maka pesan tersebut akan melalui bus. Setiap komputer perlu membaca alamat dalam pesan. Sekiranya alamat pada pesan cocok dengan alamat computer pembaca, komputer tersebut dengan mengambil pesan tersebut. (Abdul Kadir, 2013: 285). Topologi bus biasa digunakan untuk LAN dengan jumlah komputer yang sedikit. Sebagai contoh, topologi ini dapat digunakan pada warnet. Pada topologi bus, kegagalan pada simpul (bukan pada bus) tidak mempengaruhi simpul yang lain.

d. Topologi Pohon (*Cluster*)

Topologi pohon sebenarnya merupakan pengembangan dari topologi bintang dengan salah satu simpul menjadi pengontrol bagi sejumlah simpul yang berada di bawahnya. (Abdul Kadir, 2013: 285)

e. Topologi *Hybrid*

Topologi *Hybrid* merupakan gabungan dari dua atau lebih topologi yang berbeda dan berpadu menjadi satu bentuk baru pada sistem jaringan komputer. Bila topologi yang berbeda terhubung ke satu sama lainnya dan tidak memperlihatkan satu karakteristik/ciri khas topologi tertentu maka topologi tersebut dapat dikatakan topologi jaringan *hybrid*.

f. Topologi *Peer to Peer*

Topologi *peer-to-peer* merupakan diagram jaringan komputer dimana penggunaan konsep *peer-to-peer*, yang berarti bahwa setiap komputer dalam jaringan berinteraksi satu sama lain tanpa server sehingga setiap perangkat PC bisa berfungsi sebagai *client* sekaligus sebagai server itu sendiri.

g. Topologi Linier

Jenis topologi linier sebenarnya merupakan perluasan dari jenis topologi bus. Topologi linier atau biasanya disebut topologi bus beruntut, yang mana kabel utama di dalam topologi harus dihubungkan dengan setiap titik-titik yang ada di komputer dengan *T-Connector*. Pada topologi ini biasanya menggunakan satu kabel utama guna menghubungkan tiap titik sambungan pada setiap komputer.

h. Topologi Tak Beraturan (*Mesh Network*)

Pada penelitian ini, penulis menggunakan topologi *mesh network* karena topologi *mesh network* mempunyai sejumlah simpul yang setiap simpulnya tersambung secara total dengan simpul-simpul yang lain, sehingga hubungan satu simpul ke simpul lain dapat melewati sejumlah jalur.

2.1.4 Perangkat Jaringan

Perangkat yang beroperasi pada layer 3 (*Network Layer*) adalah *router* dan *switch*. Pada penelitian ini, penulis membatasi masalah dengan membahas *router* saja. Seperti yang telah diketahui, *router* adalah sebuah alat yang meneruskan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*.

Proses *routing* terjadi pada lapisan 3 (lapisan jaringan seperti *internet protocol*) dari *stack* protokol tujuh lapis OSI. Proses ini merupakan suatu proses dimana *router* meneruskan paket ke jaringan yang dituju. Suatu *router* menggunakan *IP Address* untuk mengirim paket untuk host pada network lain.

Apabila paket-paket tersebut ditujukan untuk host pada satu network yang sama, maka *router* akan menghalangi pengiriman paket-paket tersebut.

Agar keputusan *routing* tersebut benar, *router* harus dapat mengetahui bagaimana untuk mencapai tujuan. Ketika *router* menggunakan *routing* dinamis, informasi ini dipelajari dari *router* yang lain.

Seorang administrator jaringan memilih suatu protokol *routing* dinamis berdasarkan keadaan topologi jaringannya. Hal ini dapat berdasarkan *bandwidth* yang tersedia, proses *power* dalam *router*, merek dan model dari *router*, dan protokol yang digunakan dalam jaringan. *Routing* dinamis dapat diterapkan di jaringan skala besar dan membutuhkan kemampuan yang besar dari seorang administrator jaringan. Ketika menggunakan *routing* statis, seorang administrator jaringan harus melakukan konfigurasi informasi tentang jaringan yang ingin dituju secara manual, yang berarti bahwa administrator jaringan harus memasukkan atau menghapus rute statis jika terjadi perubahan topologi. Oleh sebab itu, jika menggunakan *routing* statis pada jaringan skala besar, akan sangat membuang waktu bagi seorang administrator jaringan.

Terdapat beberapa jenis *router* yang dibedakan berdasarkan:

- a. Jenis-jenis *router* berdasarkan pengaplikasiannya, antara lain: *router* aplikasi, *router hardware*, dan *router* PC.
- b. Jenis-jenis *router* berdasarkan mekanismenya, antara lain: *router* statis, *router* dinamis, dan *router wireless*.

Selain itu, terdapat pula beberapa merek *router* yang sering digunakan oleh beberapa perusahaan, kelompok, maupun perseorangan, antara lain:

- a. Mikrotik

MikroTikls adalah nama perusahaan pemegang lisensi MikoTik yang berlokasi di Riga, Ibukota Latvia, sebuah negara pecahan Uni Soviet yang bersebelahan dengan Rusia. Mikrotikls merupakan produsen *software* dan *hardware* router MikroTik. Dengan mikrotik, teknologi internet menjadi lebih cepat, handal dan terjangkau untuk kalangan pengguna yang lebih luas.

Mikrotik menurut jenisnya dapat dibagi menjadi dua, yaitu:

1. Mikrotik RouterOS

Mikrotik RouterOS adalah sebuah *software* yang berfungsi mengubah komputer menjadi sebuah router. Mikrotik RouterOS layaknya IOS Cisco yang diinstal di dalam Router Cisco,

hanya saja IOS Cisco tidak bisa diinstal di dalam komputer kecuali menggunakan emulator seperti GNS3 dan *dynamips*. Pada dasarnya RouterOS merupakan sistem operasi Mikrotik RouterBoard yang berbasis Kernel Linux v2.6.

2. Mikrotik RouterBoard

Selain bisa diinstal di dalam komputer, Mikrotik RouterOS juga bisa diinstal pada sebuah *hardware* khusus yang bernama RouterBoard.

b. Cisco

Cisco telah membuat standarisasi jaringan komputer yang menjadi acuan pokok berbagai lembaga di dunia baik secara teoritis maupun praktis. Sama seperti komputer, perangkat jaringan komputer juga membutuhkan sistem operasi agar dapat melakukan tugasnya. *Router* dan *switch* Cisco memiliki beberapa jenis OS seperti Cisco IOS, Unified, FX-OS (firepower), NX-OS (nexus), dan yang lainnya. IOS juga terbagi menjadi IOS-XE dan IOS-XR. Di *track CCNA routing switching*, OS yang dipelajari hanya satu, yaitu IOS. IOS digunakan oleh *router* dan *switch*.

c. Huawei

Router Huawei menggunakan sistem operasi yang dinamakan VRP (*Versatile Routing Platform*). Ini seperti Cisco IOS, TiMOS Nokia (Alcatel-Lucent) dan JUNOS Juniper.

2.2 Network Automation

2.2.1 Pengertian Network Automation

Otomatisasi jaringan adalah proses mengotomatisasi konfigurasi, manajemen, pengujian, penyebaran, dan operasi perangkat fisik dan virtual dalam jaringan. Setiap hari tugas dan fungsi jaringan dilakukan secara otomatis. Menggunakan kombinasi solusi berbasis perangkat keras dan perangkat lunak, organisasi besar, penyedia layanan, dan perusahaan dapat menerapkan otomatisasi jaringan untuk mengontrol dan mengelola proses berulang dan meningkatkan ketersediaan layanan jaringan.

Saat ini, otomatisasi jaringan sepenuhnya mampu melakukan tugas-tugas berikut:

- a. Mengetahui topologi jaringan yang cocok
- b. Mengelola *bandwidth* dan menemukan rute ulang cepat untuk mengimplementasikan jalur komputasi terbaik
- c. Melakukan analisis akar penyebab apabila terjadi error
- d. Memperbarui dan menginstal rute
- e. Mengatur tolok ukur kinerja
- f. Memperbarui perangkat lunak
- g. Menerapkan keamanan dan kepatuhan

2.2.2 Prinsip Kerja Network Automation

Otomatisasi jaringan menggunakan logika yang dapat diprogram untuk mengelola sumber daya jaringan dan layanan. Hal ini dapat membantu administrator jaringan untuk dengan cepat melakukan konfigurasi dan mengintegrasikan infrastruktur jaringan (lapisan 1-3) dan layanan aplikasi (lapisan 4-7) secara aman.

Penyedia layanan telekomunikasi adalah yang pertama mengadopsi otomatisasi jaringan untuk merampingkan jaringan skala web perusahaan yang tumbuh dengan cepat, tetapi semua organisasi sekarang dapat mengambil manfaat dari teknologi otomasi jaringan.

Otomatisasi jaringan juga dapat diimplementasikan ke perangkat yang mendukung protokol SSH (*Secure Shell*) sehingga pekerjaan bisa diselesaikan jauh lebih cepat dan juga efisien dalam pemeliharaan jaringan dengan prosedur yang lebih mudah diikuti dan diimplementasikan di dalam jaringan berskala besar.
[12]

2.2.3 Manfaat Network Automation

Dengan mengotomatisasi fitur-fitur jaringan dan mengimplementasikan produk-produk perangkat lunak yang menawarkan otomatisasi, organisasi mendapat manfaat dari hal-hal berikut:

- a. Biaya lebih rendah
- b. Meningkatkan kelangsungan bisnis
- c. Meningkatkan tenaga kerja strategis
- d. Wawasan yang lebih luas dan kontrol jaringan
- e. Tingkatkan kelincahan bisnis

2.3 Piranti Pengembangan Aplikasi Otomatisasi Jaringan

Untuk dapat membuat aplikasi otomatisasi jaringan berbasis web membutuhkan beberapa perangkat lunak. Kebutuhan perangkat lunak merupakan kondisi, syarat, kriteria, atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pengguna. Adapun beberapa perangkat lunak yang dibutuhkan, antara lain:

2.3.1 *Web Application*

Web Application (aplikasi web) adalah program aplikasi yang disimpan di server jarak jauh dan dikirimkan melalui internet melalui antarmuka browser. Menurut editor Web.AppStorm Jarel Remick, setiap komponen situs web yang menjalankan beberapa fungsi untuk pengguna memenuhi syarat sebagai aplikasi web. Aplikasi web dapat dirancang untuk berbagai macam kegunaan dan dapat digunakan oleh siapa saja; dari organisasi ke individu karena berbagai alasan. Aplikasi web yang umum digunakan dapat mencakup *webmail*, kalkulator *online*, atau toko *e-commerce*. Beberapa aplikasi web hanya dapat diakses oleh browser tertentu, namun, sebagian besar tersedia apapun *browser*-nya. [13]

Aplikasi web tidak perlu diunduh karena diakses melalui jaringan. Pengguna dapat mengakses aplikasi web melalui browser web seperti Google Chrome, Mozilla Firefox atau Safari. Agar aplikasi web dapat beroperasi, diperlukan server web, server aplikasi, dan *database*. Server web mengelola permintaan yang datang dari klien, sedangkan server aplikasi menyelesaikan tugas yang diminta. *Database* dapat digunakan untuk menyimpan informasi yang dibutuhkan.

Aplikasi web biasanya memiliki siklus pengembangan yang singkat dan dapat dibuat dengan tim pengembang kecil. Sebagian besar aplikasi web ditulis dalam JavaScript, HTML5, atau *Cascading Style Sheets (CSS)*. Pemrograman sisi klien biasanya menggunakan bahasa-bahasa ini, yang membantu membangun *front-end* aplikasi. Pemrograman sisi server dilakukan untuk membuat skrip yang akan digunakan aplikasi web. Bahasa seperti python, java, dan ruby biasanya digunakan dalam pemrograman sisi server.

Komponen pada aplikasi web terdiri dari tiga bagian, yaitu:

- a. *Presentation Tier* merupakan bagian *front-end* dari aplikasi web yang berhubungan langsung dengan web *client*. *Presentation tier* ini dikenal juga sebagai *service web server*, seperti IIS atau Apache. Tugas dari *presentation tier* ini adalah menerima inputan dari *end-user* dan menampilkan *output* ke *end-user* melalui *webclient*.
- b. *Application Tier* adalah bagian *logic* atau proses jalannya dari aplikasi web. Pada *layer* ini semua instruksi pemrograman menghubungkan antara inputan yang dilakukan oleh user pada *presentation tier* dengan data yang ada pada *back-end* untuk melakukan operasi dari aplikasi web. Bahasa pemrograman yang biasanya digunakan adalah C, Java, PHP, dan Visual Basic.
- c. *Persistent Tier* disebut juga sebagai sumber data untuk aplikasi. Semua kebutuhan data yang diperlukan pada *aplication tier* maka akan mengambil dari *persistent tier* ini. Dan jika aplikasi web memerlukan penyimpanan data maka akan disimpan pada *persistent storage*. Pada umumnya web aplikasi yang menggunakan *persistent tier* ini adalah yang menggunakan *Database Management System (DBMS)*, seperti Ms SQL, Oracle, atau MySQL.

Jenis-jenis aplikasi web di bawah ini bisa digunakan sesuai dengan kebutuhan yang diinginkan. *Web Application* dapat diklasifikasikan menjadi tipe dan jenis aplikasi web beserta kegunaannya berikut ini: aplikasi web bisnis, aplikasi *online web service*, aplikasi web berita dan informasi, aplikasi web profil, aplikasi web *social network*, aplikasi web *banking*, dan aplikasi SEO (*Search Engine Optimize*).

2.3.2 Python

Python adalah bahasa pemrograman *open source* yang digunakan untuk aplikasi *stand alone* dan juga aplikasi *script* atau yang biasa disebut *scripting application* di dalam berbagai domain.

Python dirancang untuk memberikan kemudahan kepada programmer baik dari segi efisiensi waktu, maupun kemudahan dalam pengembangan program dan dalam hal kompatibilitas dengan sistem. Python merupakan salah satu bahasa pemrograman tingkat tinggi (*High Level Language*) yang bersifat interpreter, interaktif, dan berorientasi objek. Di dalam bahasa python sendiri pun terdapat bahasa tingkat rendah (*Low Level Language*) yang berhubungan dengan bahasa mesin atau *assembly*.

Python memiliki beberapa kelebihan dibandingkan dengan bahasa pemrograman lainnya, yaitu:

- a. Pemrograman tingkat tinggi (*High Level Language*).
- b. Mudah dipelajari.
- c. Mudah digunakan.
- d. Mudah dalam pengembangan.
- e. Manajemen memori dinamis.
- f. Pemrograman berorientasi objek (*Object Oriented Programming*).
- g. Platform independent.
- h. Bersifat open source dan tidak berbayar.

Penggunaan python dalam pembuatan aplikasi otomatisasi jaringan berbasis web ini adalah untuk mengirimkan perintah langsung dari admin menggunakan SSH untuk *login* ke setiap *router* di web, lalu, meneruskan perintah tersebut untuk dilakukan konfigurasi maupun monitoring.

2.3.3 Paramiko

Paramiko adalah implementasi Python (2,7, 3,4+) dari protokol SSHv2, menyediakan fungsionalitas klien dan server. Meskipun memanfaatkan ekstensi Python C untuk kriptografi tingkat rendah, paramiko sendiri adalah antarmuka Python murni di sekitar konsep jaringan SSH.

Cara kerja SSH adalah memakai model *client server*, dimana SSH *client* ini akan melakukan koneksi ke SSH *server*. Dalam proses koneksi ke SSH *server*, SSH *client* akan menggunakan kunci kriptografi, fungsinya adalah melakukan verifikasi dan identifikasi SSH *server* yang dituju. Jika kunci yang dipakai SSH *client* tidak sama dengan *server*, maka koneksi tidak akan tersambung. SSH *client* dan *server* bisa tersambung atau terkoneksi jika memakai kunci yang sama, tentunya setelah melalui proses verifikasi. Nantinya, koneksi yang tersambung terlebih dahulu akan dienkripsi dengan memakai *symmetric encryption* serta *hashing algorithm*. Tujuan dilakukannya proses enkripsi pada SSH adalah untuk memastikan keutuhan data dan kerahasiaan data guna keperluan antara *client* serta *server*.

SSH *client* merupakan sebuah aplikasi yang fungsinya untuk menghubungkan sistem operasi dengan SSH *server*. SSH *client* bisa diterapkan pada beberapa sistem operasi, bahkan yang sebelumnya telah terpasang dalam sebuah perangkat seperti Linux dan MacOS. Lalu, bagi yang menggunakan sistem operasi Windows bisa memakai aplikasi seperti PuTTY.

Konsep jaringan SSH yang digunakan pada paramiko adalah untuk melakukan *remote* pada setiap *router* yang dituju, yang kemudian router tersebut akan dikirimkan perintah yang sesuai dengan admin tulis.

Bahasa pemrograman python dapat login ke sebuah router dengan SSH dikarenakan adanya library paramiko untuk koneksi SSH ke setiap router yang ada. Dan cara kerja paramiko agar dapat login SSH secara otomatis ke *multi-virtual router* dikarenakan penulis memanfaatkan fungsi “for” looping untuk mengulang ke setiap router yang ada.

2.3.4 Django

Django adalah kerangka kerja python web tingkat tinggi yang mendorong pengembangan yang cepat dan desain pragmatis yang bersih. Dibangun oleh pengembang yang berpengalaman, hal ini berarti menangani banyak kerumitan pengembangan web, sehingga pengguna dapat fokus pada penulisan aplikasi pengguna tanpa perlu menemukan kembali awalan. Ini gratis dan sumber terbuka.

Peran django dalam pembuatan aplikasi otomatisasi jaringan adalah sebagai penghubung antara *library* paramiko ke tampilan web. Agar aplikasi otomatisasi jaringan dapat ditampilkan pada sebuah web, penulis menggunakan *framework* django.

2.3.5 Aplikasi Simulasi GNS3

Aplikasi GNS3 (*Graphical Network Simulator 3*) menyediakan simulasi-simulasi yang lengkap dan akurat, sehingga terkait dengan:

- a. Dynamips, sebuah emulator IOS Cisco.
- b. Dynagen, sebuah front-end berbasis teks untuk Dynamips.
- c. Qemu, sebuah mesin emulator dan virtualizer generik dan open source.
- d. VirtualBox, sebuah *software* virtualisasi yang gratis dan powerful. [5]

2.4 Metode Penelitian

Metode penelitian yang akan digunakan merupakan metode untuk perancangan aplikasi berbasis web. Untuk perancangan aplikasi berbasis web, penulis menggunakan metode RUP (*Rational Unified Process*). Sedangkan untuk pengujian hasil aplikasi, penulis menggunakan metode *Black Box Testing*.

2.4.1 RUP (*Rational Unified Process*)

RUP (*Rational Unified Process*) adalah pendekatan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang lebih baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh Rational Software yang diakuisisi oleh IBM di bulan Februari 2003.

Proses pengulangan/iteratif pada RUP secara global dapat dilihat sebagai berikut:



Gambar 2.1 Proses *iterative* RUP

RUP (*Rational Unified Process*) memiliki 4 tahap atau fase yang dapat dilakukan pula secara iteratif. Berikut adalah penjelasan untuk setiap fase RUP (*Rational Unified Process*):

a. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

Pada pembuatan aplikasi otomatisasi jaringan, tahap ini merupakan tahap bagi penulis untuk mengidentifikasi sistem yang telah ada dan yang akan dikembangkan, termasuk arsitektur, fitur, dan *use case* sistem dari hasil wawancara dan observasi yang penulis lakukan, meliputi dari hasil penelitian atau skripsi-skripsi terdahulu.

b. *Elaboration* (perluasaan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*). Kemudian, penulis melakukan desain secara lengkap berdasarkan hasil analisis di tahap *inception* untuk pembuatan aplikasi otomatisasi jaringan.

c. *Contruaction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Pada tahapan ini menjelaskan bagaimana mengimplementasi hasil desain dan melakukan uji coba terhadap aplikasi yang telah dibuat. Dalam tahapan implementasi dijelaskan perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk mengimplementasi aplikasi ini. Sedangkan pada tahapan uji coba dilakukan *testing*. *Testing* diperlukan untuk menjamin kualitas aplikasi yang telah dibuat apakah telah sesuai dengan yang diharapkan.

d. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh user.. Aktifitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user [10]. Pada pembuatan aplikasi otomatisasi jaringan, tahap ini merupakan tahap akhir bagi penulis untuk menyerahkan aplikasi ke pengguna.

2.4.2 *Black Box Testing*

Terdapat beberapa metode pengujian hasil aplikasi, tetapi, pada penelitian ini, penulis menggunakan metode *black box testing* untuk melakukan suatu pengujian yang ditujukan pada tampilan perangkat lunak. *Black box testing* memeriksa beberapa aspek penting dari sistem, seperti aspek yang dilihat hanya pada fungsionalitas dan output, lebih fokus pada hasil dan tanpa melihat proses yang terjadi di dalam sistem.

Tujuan utama dari *black box testing* adalah mencari kesalahan atau *error* dengan kategori sebagai berikut:

- a. Mencari fungsi yang salah atau kurang.
- b. Mencari kesalahan pada tampilan.
- c. Mencari kesalahan pada struktur data atau akses *database*.
- d. Mencari kesalahan pada *behavior* atau performa.
- e. Mencari kesalahan pada inisialisasi (awal) dan akhiran.

2.5 State of Art (Penelitian Terdahulu)

Tabel 2.1 State of Art (Penelitian Terdahulu)

Penulis	Tahun	Judul	Software	Metode	Hasil	Kekurangan
Rheza Adhyatmaka Wiryawan dan Nur Rohman Rosyid [1]	2019	Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python	Bahasa Python, Library Paramiko	Metode RAD (<i>Rapid Application Development</i>) dan diterapkan pada perangkat Mikrotik dan Cisco. Metode black box testing untuk pengujian hasil aplikasi.	Menghasilkan lima buah fitur konfigurasi <i>routing</i> , <i>vlan</i> , <i>backup</i> , <i>restore</i> , dan <i>setting</i> . Fungsi pada semua fitur aplikasi yang dikembangkan berfungsi dengan baik dan berhasil diterapkan pada perangkat Mikrotik dan Cisco.	Kurangnya fitur log pada aplikasi tersebut yang sangat penting bagi administrator jaringan untuk melakukan monitoring terhadap perangkat jaringan. Halaman pengguna dan admin tidak dibedakan sehingga sulit untuk menyimpan database.
Ahmad Rosid Komarudin [11]	2018	Otomatisasi Administrasi Jaringan Dengan Script Python	Bahasa Python, Library Paramiko, Netmiko, Pyntc, Napalm, dan Ansible	Dihasilkan masih dalam bentuk script dan hanya dapat diterapkan pada perangkat Cisco.	Semua tools yang digunakan dapat diimplementasikan ke perangkat Cisco untuk otomatisasi jaringan.	Otomatisasi jaringan yang dilakukan masih menggunakan bahasa script, belum ada pengembangan aplikasi. Tools Napalm dan

						<p>Ansible merupakan tools dengan fitur paling <i>powerfull</i> yang hanya bisa diterapkan di perangkat Cisco.</p> <p>Dukungan merek perangkat pada tools Napalm masih terbatas seperti Arista EOS, Cisco IOS, Cisco IOS-XR, Cisco NX-OS dan Juniper JunOS.</p>
<p>I Made Bayu Swatika dan I Gede Oka Gartria Atitama [2]</p>	2017	<p>Otomatisasi Konfigurasi Mikrotik Router Menggunakan Software Ansible</p>	<p>Software Ansible</p>	<p>Otomatisasi jaringan dilakukan dengan mengatur bandwidth menggunakan fitur Queue Tree pada Mikrotik.</p>	<p>Menambahkan modul RouterOS API pada Ansible agar software Ansible dapat terhubung dan dapat melakukan konfigurasi router Mikrotik melalui API.</p>	<p>Otomatisasi jaringan yang dilakukan masih menggunakan bahasa script, belum ada pengembangan aplikasi.</p> <p>Hanya dapat beroperasi pada satu jenis perangkat jaringan saja, yaitu Mikrotik.</p>

				Metode Black-Box Testing untuk pengujian hasil.	Pengujian pada penelitian tersebut menghasilkan bahwa <i>software</i> Ansible dapat melakukan konfigurasi beberapa <i>queue tree</i> secara berurutan dalam sekali eksekusi file dengan waktu proses kurang lebih 1 menit untuk 19 konfigurasi <i>queue tree</i> .	
Paul MIHĂILĂ, Titus BĂLAN, Radu CURPEN dan Florin SANDU [9]	2017	<i>Network Automation and Abstraction using Python Programming Methods</i>	Bahasa Python, Library Paramiko dan Netmiko	Dihasilkan masih dalam bentuk script dan hanya dapat diterapkan pada perangkat Cisco.	Dilakukan otomatisasi pembuatan VLAN pada 3 switch Cisco yang dikontrol menggunakan <i>script</i> Python.	Otomatisasi jaringan yang dilakukan masih menggunakan bahasa <i>script</i> , belum ada pengembangan aplikasi.

				<p>Pada library Netmiko penggunaan untuk koneksi ke switch menggunakan <i>ConnectHandler</i> yang juga menggunakan SSH dibagian backend, selain itu Netmiko dapat menentukan tipe perangkat yang akan diatur.</p> <p>Otomatisasi dapat menggunakan Phyton dan koneksi Secure Shell.</p>	<p>Hanya dapat beroperasi pada satu jenis perangkat jaringan saja, yaitu Cisco.</p>
--	--	--	--	---	---