BAB II
TINJAUAN PUSTAKA

2.1 Landasan Teori

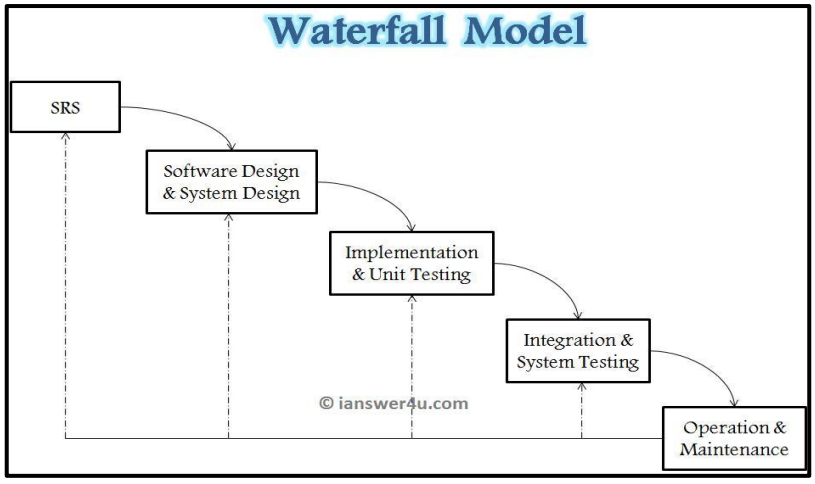
2.1.1 System Development Life Cycle (SDLC)

SDLC adalah tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan *programmer* dalam membangun sistem informasi dan metode dalam mengembangkan sistem tersebut. Sistem yang dibangun dengan dengan menggunakan SDLC akan memudahkan dalam mengidentifkasi masalah dan merancang sistem sesuai kebutuhan dalam menyelesaikan permasalahan tersebut. Salah satu SDLC yang paling sering digunakan dalam pengembangan sistem yaitu SDLC *Waterfall.*

SDLC *Waterfall* sesuai namanya SDLC ini berkembang secara sistematis dari satu tahap ke tahap lain layaknya air terjun. Metode *waterfall* merupakan suatu metode dalam pengembangan *software* dimana pengerjaannya harus dilakukan secara berurutan yang dimulai dari tahap perencanaan konsep, pemodelan(*design*), implementasi, pengujian dan pemeliharaan.

A.Tahapan Metode Waterfall

Berikut ini merupakan tahapan tahapan pengembangan dalam metode *waterfall*.



Gambar 2.1 Tahapan Metode Waterfall

1.*Requirement Analysis*

Langkah pertama dimulai dengan membangun keseluruhan elemen sistem dan memilah bagian-bagian mana yang akan dijadikan bahan pengembangan perangkat lunak, dengan memperhatikan hubungannya dengan Hardware, User, dan Database.

2.*System Design*

Pada proses desain, dilakukan penerjemahan syarat kebutuhan ke sebuah perancangan desain perangkat lunak yang dapat diperkirakan sebelum dibuatnya proses pengkodean (*coding*). Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail algoritma prosedural. Berikut contoh desain sistem yang biasanya dibuat dan digunakan.

- *Data Flow Diagram* (DFD).
- *Flowchart.*
- *Mind Map.*
- *Entity Relationship Diagram* (ERD).
- *Context Diagram, etc*.

3.*Implementation*

Pada tahap ini terjadi proses menerjemahkan perancangan desain ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan kode kode bahasa pemrograman. Kode program yang dihasilkan masih berupa modul-modul kecil yang nantinya akan digabungkan pada tahap berikutnya.

4.*Integration & Testing*

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan fungsi pada *software* terdapat kesalahan atau tidak.

5.*Operation & Maintenance*

Ini merupakan tahap terakhir dalam model *waterfall*. *Software* yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam

memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.1.2 Konsep Jaringan Komputer

Gambar 2.2 Jaringan komputer

Menurut Jafar Noor Yudianto (2007) , *Jaringan Komputer* adalah sebuah sistem yang terdiri atas komputer-komputer yang didesain untuk dapat berbagi sumber daya seperti printer dan CPU, berkomunikasi seperti surel dan pesan instan, dan dapat mengakses informasi seperti dari pramban web. Tujuan jaringan komputer adalah setiap jaringan dapat meminta dan memberikan layanan(service). Design ini di sebut dengan sistem client-server, dan digunakan pada hampir seluruh aplikasi jaringan computer.

Tujuan membangun jaringan komputer adalah untuk membawa informasi secara tepat dan tanpa adanya kesalahan dari sisi pengirim *(transmitter)* menuju ke sisi penerima *(receiver)* melalui media komunikasi. Kendala-kendala dalam membangun jaringan komputer adalah masih mahalnya fasilitas komunikasi yang tersedia dan bagaimana memanfaatkan jaringan komunikasiyang ada secara efektif dan efisien serta jalur transmisi yang digunakan tidak benar-benar bebas dari masalah ganguan (*noise*). Manfaat yang didapatkan dalam membangun jaringan komputer adalah sebagai *sharing resources,* media komunikasi, integrasi data, pengembangan dan pemeliharaan, keamanan data, sumberdaya lebih efisien dan informasi terkini.

Klasifikasi jaringan komputer terbagi menjadi 3 yaitu:

1. Berdasarkan geografisnya, berdasarkan geografis jaringan komputer terbagi menjadi 3 yaitu : LAN*(Local Area Network),* MAN*(Metropolitan Area Network),* dan WAN*(Wide Area Network).*
2. Berdasarkan fungsi, berdasarkan fungsinya jaringan komputer terbagi menjadi Jaringan klien-server*(Client-Server)* dan Jaringan ujung ke ujung*(peer-to-peer).*
3. Berdasarkan topologi jaringan, berdasarkan topologinya terbagi menjadi, topologi bus, topologi bintang, topologi cincin, topologi mesh, topologi pohon, dan topologi linier.

2.1.3 Otomatisasi Jaringan

Menurut Imron Rosyadi(2018), *Network Automation* secara bahasa berarti Otomatisasi Jaringan, sedangkan secara istilah yaitu proses mengotomatisasi konfigurasi, manajemen, pengujian, penyebaran, dan operasi perangkat fisik dan virtual dalam suatu jaringan. Setiap hari tugas dan fungsi jaringan dilakukan secara otomatis. Menggunakan kombinasi solusi perangkat keras dan berbasis perangkat lunak, organisasi besar, penyedia layanan, dan perusahaan dapat menerapkan otomatisasi jaringan untuk mengontrol dan mengelola proses berulang dan meningkatkan ketersediaan layanan jaringan.

Pada inti Manajemen Konfigurasi Jaringan dan Otomasi Jaringan, tampaknya Anda hanya mengurangi kesalahan manusia dari proses operasional. Namun, Otomasi Jaringan lebih dari itu; itu mempercepat proses digitalisasi, membantu meningkatkan produktivitas dan mengurangi biaya operasional.

Otomasi Jaringan berkontribusi pada dua area utama:

1. Mengotomatiskan TI: TI mengotomatiskan jika tidak memakan sumber daya, pemfokusan perangkat keras, dan penyediaan manual, konfigurasi, pembaruan, dan lainnya.
2. Visibilitas Jaringan yang Ditingkatkan: Ini memperluas visibilitas tim pemantau jaringan untuk mengungkap dependensi dan memberikan

pemecahan masalah yang lebih cepat untuk jaringan dan kebijakan keamanannya.

Masalah jaringan menjadi penting dalam organisasi di mana visibilitas dan kontrol pada alur kerja perubahan tidak dapat diakses dengan fungsi NMS. Konfigurasi yang salah, pelanggaran kepatuhan yang sering terjadi, dan konflik yang timbul dalam pengaturan konfigurasi umumnya merupakan masalah utama di balik masalah jaringan sistemik. Masalah tersebut dapat menimbulkan tantangan yang signifikan baik untuk kinerja jaringan maupun hasil bisnis.

2.1.4 Static Routing dan Dynamic Routing

Secara umum mekanisme koordinasi routing dapat dipelajari oleh router dalam dua metode, yaitu:

• Dimasukkan secara manual oleh administrator jaringan, disebut Static Routes.

• Dikumpulkan melalui proses-proses dinamis yang berjalan di jaringan, disebut sebagai Dynamic Routes.

- Static Routing

Routing statik (static route) adalah pengaturan routing paling sederhana yang dapat dilakukan pada jaringan komputer. Static route adalah rute-rute ke host atau jaringan tujuan yang dimasukkan secara manual oleh administrator jaringan ke route table suatu router. Static route mendefinisikan alamat IP hop router berikutnya dan interface lokal yang digunakan untuk mem-forward paket ke tujuan tertentu (hop router berikutnya).

Static route memiliki keunggulan untuk menghemat bandwidth jaringan karena static route tidak membangkitkan trafik route update untuk memberikan informasi perubahan rute yang berlaku (sah) saat ini ke router-router lain. Penggunaan routing statik dalam sebuah jaringan yang kecil tentu bukanlah suatu masalah, hanya beberapa entri yang perlu diisikan pada forwarding table di setiap router. Namun tentu dapat dibayangkan bagaimana jika harus melengkapi forwarding table di setiap router yang jumlahnya tidak sedikit dalam jaringan yang besar. Apalagi jika untuk mengisi entri-entri di seluruh router di Internet yang jumlahnya banyak sekali dan terus bertambah setiap hari.

Jadi penggunaan static route cenderung membutuhkan waktu ekstra ketika memanajemen jaringan. Hal ini disebabkan karena sistem administrator harus secara manual meng-update route table setiap terjadi perubahan konfigurasi jaringan.

- Dynamic Routing

Routing dinamik adalah cara yang digunakan untuk melepaskan kewajiban mengisi entri-entri forwarding table secara manual. Protokol routing mengatur router-router sehingga dapat berkomunikasi satu dengan yang lain dan saling memberikan informasi routing yang dapat mengubah isi forwarding table, tergantung keadaan jaringannya. Dengan cara ini, router-router mengetahui keadaan jaringan yang terakhir dan mampu meneruskan datagram ke arah yang benar.

Routing dinamik yang popular saat ini mengacu pada dua tipe algoritma yang dikenalkan oleh Bellman Ford dengan algoritma distance vectornya dan oleh Djikstra dengan algoritma link statenya. Cisco kemudian mengembangkan protocol untuk perangkat routernya yang merupakan gabungan dari kedua algoritma tersebut yang diberi nama protocol EIGRP.

2.1.5 Aplikasi Berbasis Web

Gambar 2.3 Aplikasi Berbasis Web

Menurut Andi Hadmoko (2019), *Aplikasi Berbasis Web* adalah sebuah aplikasi yang diakses melalui web browser dengan menggunakan jaringan sebagai media transmisi. Aplikasi web juga merupakan sebuah perangkat lunak atau software yang di kodekan dengan bahasa pemogramman seperti html, javascript, css, ruby, python, php, dan bahasa pemogramman lainnya.

Jenis-jenis Aplikasi berbasis web :

1. Web Sosial Media

Website juga dapat dimanfaatkan untuk sarana komunikasi dalam bentuk percakapan *online* yang dapat dilakukan oleh setiap orang secara cepat dan *real-time*. Atau, biasa disebut dengan media sosial. Contohnya adalah Facebook, Twitter, Instagram, dll.

2. Web Berbasis Sistem Informasi

Website juga digunakan untuk sarana membantu aktivitas usaha dan pekerjaan manusia. Sehingga proses pekerjaan yang dilakukan dapat tersistem, terpusat, dan termonitoring dengan baik menggunakan aplikasi. Saat ini dikenal dengan sistem informasi. Sistem informasi sendiri memiliki beberapa jenis, disesuaikan dengan kebutuhan dari bidang kerja masing – masing. Contohnya adalah sistem informasi koperasi, SIAKAD (Sistem Informasi Akademik), *Fleet Management System*, *Hospital Management*, dan masih banyak lagi SI yang lain.

3. Web Jual Beli dan Bisnis

Kemudian, website juga dapat digunakan untuk sarana transaksi jual beli secara online. Saat ini disebut dengan *e-commerce.* Dengan menggunakan *e-commerce* segala kebutuhan anda terkait produk barang atau jasa dapat diproses hanya dengan menggunakan aplikasi web. Contoh aplikasi yang banyak digunakan di Indonesia adalah Tokopedia, Shopee, Bukalapak, dan platform *e-commerce* lainnya. Anda dapat memilih berbagai produk mulai dari yang baru, bekas, harga murah hingga termahal dapat anda dapatkan melalui aplikasi.

4. Web Pencarian

Web pencarian biasa disebut dengan *Search Engine.* Tentunya, anda hampir setiap hari selalu mengakses yang namanya mesin pencari seperti Google, Yahoo, Youtube, dll. Mesin pencari dapat melakukan berbagai pencarian informasi secara cepat dan akurat.

5. Web Informasi dan Berita

Dari aplikasi berbasis website juga dapat menampilkan informasi dan berita teraktual dan terkini dari seluruh dunia. Contoh web berita di Indonesia adalah Detik.com, Kompas.com, Tribunnews, dll.

6. Aplikasi Web Server

Definisi dari aplikasi web server adalah sebuah perangkat aplikasi, dimana anda dapat menerima *request* (permintaan) dan juga bisa mengirim respon atau tanggapan dalam protokol HTTP (*Hypertext Transfer Protocol*). Di dalam proses implementasinya, tentu saja sudah terprogram dengan bantuan bahasa pemrograman *server-side* atau lebih dikenal dengan istilah back end. Untuk jenis aplikasi web server dikembangkan oleh *user* yang ingin membangun sebuah *client / server* pada sebuah website, khususnya di kalangan *IT development*. Contoh dari jenis ini adalah Apache HTTP Web Server, Nginx, XAMPP, Apache Tomcat, Lighttpd, LAMP, WAMP, MAMP, dan masih banyak contoh yang lain lagi.

7. Aplikasi Web Browser

Apa itu aplikasi web browser? Jika dilihat dari segi istilah, aplikasi web browser adalah sebuah perangkat lunak (*software*) yang dipergunakan untuk membuka dan menjalankan halaman atau situs website. Contoh dari web browser yang saat ini banyak digunakan adalah Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Baidu, Opera, Vivaldi, Torch, Maxthon, UC Browser, dan masih banyak lagi contoh browser yang lainnya.

2.1.6 Python

Gambar 2.4 Logo Python

Menurut Wibowo (2017), *Phyton* adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, *Phyton* lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat *Phyton* sangat mudah dipelajari baik untuk

pemula maupun untuk yang sudah menguasai bahasa pemrograman lain. Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah sibuk mencari *syntax error*.

Phyton diciptakan oleh Guido van Rossum pertama kali di Scitchting Mathematisch Centrum (CWI) di Belanda pada awal tahun 1990-an. Bahasa *Phyton* terinspirasi dari bahasa pemrograman ABC. Sampai sekarang, Guido masih menjadi penulis utama untuk *Phyton*, meskipun bersifat *open source* sehingga ribuan orang juga berkontribusi dalam mengembangkannya. Di tahun 1995, Guido melanjutkan pembuatan *Phyton* di *Corporation for National Research Initiative* (CNRI) di Virginia Amerika, dimana dia merilis beberapa versi dari *Phyton*.

Pada Mei 2000, Guido dan tim *Phyton* pindah ke *BeOpen.com* dan membentuk tim *BeOpen PhytonLabs*. Di bulan Oktober pada tahun yang sama, tim *Phyton* pindah ke *Digital Creation* (sekarang menjadi Perusahaan *Zoe*). Pada tahun 2001, dibentuklah Organisasi *Phyton* yaitu *Phyton Software Foundation* (PSF). PSF merupakan organisasi nirlaba yang dibuat khusus untuk semua hal yang berkaitan dengan hak intelektual *Phyton*. Perusahaan Zoe menjadi anggota sponsor dari PSF. Semua versi *Phyton* yang dirilis bersifat open source. Dalam sejarahnya, hampir semua rilis *Phyton* menggunakan lisensi *GFLcompatible*. Nama *Phyton* sendiri tidak berasal dari nama ular yang kita kenal. Guido adalah penggemar grup komedi Inggris bernama Monty *Phyton*. Ia kemudian menamai bahasa ciptaannya dengan nama *Phyton*. *Phyton* adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, *Phyton* lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat *Phyton* sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain. Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah sibuk mencari *syntax error* (Wibowo, 2017).

1.1.7 Django

Gambar 2.5 Logo Django

Dikutip dari laman Real Python, "*Django* adalah kerangka kerja Web Python tingkat tinggi yang mendorong pengembangan cepat dan desain pragmatis yang bersih. Kerangka kerja Web adalah seperangkat komponen yang menyediakan cara standar untuk mengembangkan situs web dengan cepat dan mudah. Tujuan utama Django adalah untuk memudahkan pembuatan situs web berbasis basis data yang kompleks. Beberapa situs terkenal yang menggunakan Django termasuk PBS, Instagram, Disqus, Washington Times, Bitbucket dan Mozilla."

Kelebihan menggunakan *Django* sebagai web *framework* adalah sebagai berikut.

1. Kerangka kerja *Django* memiliki *template, libraries,* dan *API* yang dirancang untuk bekerja sama untuk bertumbuh dan konektivitas.
2. *Django* cocok dengan segala ukuran proyek, dari proyek kecil sampai proyek besar.
3. *Django* menggunakan *Phyton* yang mana adalaha salah satu bahasa pemrograman yang paling populer pada tahun 2015. Sekarang menjadi bahasa pemrograman yang paling populer untuk dipelajari.
4. *Django* adalah perangkat yang paling banyak memiliki fitur dibandingkan dengan frameworks lainnya. *Django* memuat segala sesuatu yang dibutuhkan untuk membangun sebuah aplikasi.

1.1.8 Paramiko

Gambar 2.6 Logo Paramiko

Di kutip dari laman resmi Paramiko, *Paramiko* adalah implementasi *Phyton* (2.7, 3.4+) dari protokol SSHv2 [1], yang menyediakan fungsionalitas klien dan server. Sementara itu memanfaatkan ekstensi *Phyton* C untuk kriptografi tingkat rendah (Kriptografi), *Paramiko* sendiri adalah antarmuka *Phyton* murni di sekitar konsep jaringan SSH. SSH didefinisikan dalam RFC 4251, RFC 4252, RFC 4253 dan RFC 4254. Implementasi kerja utama dari protokol ini adalah proyek OpenSSH. *Paramiko* mengimplementasikan sebagian besar set fitur SSH, tetapi terkadang ada celah.

1.1.9 SSH

Gambar 2.7 Logo SSH

Secure Sheel / SSH adalah aplikasi pengganti remote login seperti telnet, rsh, dan rlogin, yang jauh lebih aman. Fungsi utama aplikasi ini adalah untuk mengakses mesin secara remote. Sama seperti telnet, SSH Client menyediakan User dengan Shell untuk remote ke mesin. Tidak seperti telnet, SSH menyediakan

koneksi enkripsi antara klien dengan server. Dalam prakteknya, penggunaan menggunakan telnet dan ssh seperti perbedaan dengan mengakses website biasa dengan website yang lebih aman (HTTPS).

SSH memungkinkan Anda untuk melakukan *remote* server atau perangkat lain yang terkoneksi dengan jaringan internet. Berbeda dengan Telnet, koneksi yang terjadi di *SSH* dienkripsi menggunakan beberapa tekno*Log*i, seperti *enkripsi simetris, enkripsi asimetris, dan hashing*. Ketiganya merupakan teknik kriptografi yang menjamin koneksi yang terenkripsi. Itulah mengapa dinamakan *SSH* yang berasal dari kata *Secure Shell Connection* atau koneksi Shell yang aman. Jika menggunakan SSH, Anda mempunyai opsi untuk melakukan autentikasi pengguna remote sebelum melakukan koneksi. Selain itu, *SSH* juga dapat mengirimkan input dari *SSH client ke host (server)* kemudian mengirimkan kembali hasilnya kembali ke *user client.*

SSH akan mengantar Anda pada sejumlah keuntungan. Berikut manfaat *SSH* :

1. Bebas Mengontrol *Hosting*

Menggunakan *SSH*, kontrol hosting jarak jauh bukan lagi hal yang mustahil. Ibaratnya, Anda memiliki *remote control* untuk mengendalikan pesawat Anda sesuai keinginan. Dengan demikian, *SSH* memungkinkan Anda memiliki hak penuh untuk mengontrol *hosting.* Tanpa harus mendatangi server fisiknya, Anda bebas memantau *Log* server, menginstal/menghapus instalasi aplikasi, transfer data, dan banyak lagi.

2. Menghindari *Cyber Crime*

SSH menyediakan teknik *kriptografi* kunci tertentu. Secara otomatis, *SSH* akan memutus koneksi ketika ada hacker yang berusaha membajak koneksi Anda.Alhasil, Anda pun selamat dari berbagai serangan *cyber crime seperti spoofing IP dan DNS*, manipulasi data, pelacakan ilegal, dan lain-lain.

3. Keamanan Ketat dengan Sistem *Autentikasi*

Pencurian data oleh *hacker* akan lebih sulit dilakukan jika Anda menggunakan *SSH*. Sebab, S*SH* memungkinkan enkripsi data sehingga *hacker* tidak bisa meretas password dan informasi pengguna begitu saja.

1.1.10 GNS3

Gambar 2.8 Logo GNS3

GNS3 adalah aplikasi simulator jaringan (Graphic Simulator Network) berbasis GUI yang di rilis pada tahun 2008. Dengan menggunakan GNS3 maka kita dapat mensimulasikan perangkat asli secara baik dengan bantuan emulator ataupun teknologi virtualisasi. Salah satu teknologi emulator yaitu dynamips, yang digunakan untuk mensimulasikan Cisco IOS. Dulu, untuk mensimulasikan (mengemulasikan) router Cisco, kita perlu menginstall dynamips terlebih dahulu, bisa di Windows, Linux, FreeBSD, atau MAC OS. Dengan GNS3, semuanya sudah include, sudah dikemas sedemikian rupa dengan interfacenya, sehingga jauh lebih mudah digunakan seperti sekarang ini.(Aksen Akbar(2020)).

GNS3 memungkinkan simulasi jaringan yang komplek, karena menggunakan operating system asli dari perangkat jaringan seperti cisco dan juniper. Sehingga kita berada kondisi lebih nyata dalam mengkonfigurasi router langsung daripada di Cisco Packet Tracer. GNS3 adalah alat pelengkap yang sangat baik untuk laboratorium nyata bagi network engineer, administrator dan orang-orang yang ingin belajar untuk sertifikasi seperti Cisco CCNA, CCNP, CCIP dan CCIE serta Juniper JNCIA, JNCIS dan JNCIE. GNS3 sering dipakai sebagai simulator router Cisco, Juniper, Mikrotik dan Virtual Machine.

1.1.11 Use Case Diagram

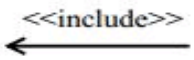
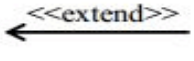
Use case diagram adalah satu dari berbagai jenis diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya.

Tentunya, use case diagram merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan, tentu perlunya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi sistem itu sendiri, seperti yang terdapat pada use case diagram.

Adapun, fungsi dari use case diagram sebagai berikut:

- Berguna memperlihatkan proses aktivitas secara urut dalam sistem.
- Mampu menggambarkan proses bisnis, bahkan menampilkan urutan aktivitas pada sebuah proses.
- Sebagai *bridge* atau jembatan antara pembuat dengan konsumen untuk mendeskripsikan sebuah sistem.

Tabel 2.1 Simbol-simbol Use Case Diagram

Simbol	**Keterangan**
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*
	*Use case* : Abstraksi dan interaksi antara sistem dan aktor
	*Association* : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

1.1.12 Activity Diagram

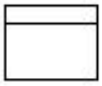
Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas.Alur atau aktivitas berupa bisa berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut. Dalam buku *Rekayasa Perangkat Lunak* karangan Rosa A.S mengatakan, "Diagram aktivitas tidak menjelaskan kelakuan aktor. Dapat diartikan bahwa dalam pembuatan *activity diagram* hanya dapat dipakai untuk menggambarkan alur kerja atau aktivitas sistem saja."

Kapan saat untuk menggunakan *activity diagram*? *Activity diagram* mesti digunakan sejajar (*horizontal*) dengan teknik pemodelan lainnya, seperti diagram *Use Case* dan diagram *State*. Kamu bisa menggunakan *activity diagram* agar dapat memodelkan alur kerja sistem dengan baik. *Activity diagram* berfungsi juga untuk menganalisis diagram *use case* dengan cara mendeskripsikan aktor, tindakan yang perlu dilakukan, dan kapan harus terjadi. Diagram ini menggambarkan sebuah algoritma dan pemodelan sekuensial yang kompleks dengan proses paralel. Selanjutnya mari kita bahas mengenai tujuan dari pada *activity diagram* itu sendiri.

Berikut beberapa tujuan dari *activity diagram:*

1. Menjelaskan urutan aktivitas dalam suatu proses.
2. Di dalam dunia bisnis biasanya digunakan untuk *modeling* (memperlihatkan urutan proses bisnis).
3. Mudah dalam memahami proses yang ada dalam sistem secara keseluruhan.
4. Merupakan metode perancangan yang terstruktur, mirip dengan *Flowchart* maupun *Data Flow Diagram* (DFD).
5. Mengetahui aktivitas aktor/pengguna berdasarkan *use case*/diagram yang dibuat sebelumnya.

Tabel 2.2 Simbol-simbol Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem,aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

1.2 Referensi Jurnal

2.2.1 Analisis Kinerja Otomasi Jaringan Ospf Menggunakan Paramiko dan Netmiko

Pada penelitian sebelumnya yang dilakukan (Anggi, Dzikri Abrariansyah (2020)) dalam jurnal yang berjudul "***Analisis Kinerja Otomasi Jaringan Ospf Menggunakan Paramiko dan Netmiko***". Dalam Abstraknya "Mengkonfigurasi

perangkat jaringan secara manual terutama pada jaringan berskala besar dapat memakan waktu yang relatif lama, baik itu melakukan proses *troubleshooting* maupun membangun jaringan dari awal. Selain membutuhkan waktu yang relatif lama, biaya untuk membangun jaringan tersebut juga besar. Otomasi jaringan merupakan salah satu teknik yang dapat menanganinya. Dengan otomatisasi jaringan, konfigurasi perangkat jaringan seperti konfigurasi *Protocol routing* dapat dilakukan secara otomatis. Konfigurasi dapat diselesaikan dengan cepat dan tentunya dapat meminimalkan biaya yang dikeluarkan. Otomasi jaringan perlu menggunakan program dengan bahasa pemrograman tertentu, salah satunya python. Bahasa pemrograman Python memiliki berbagai jenis library yang mendukung otomatisasi jaringan. Dalam penelitian ini menggunakan Paramiko dan Netmiko. Paramiko adalah *library* Python yang dapat digunakan sebagai klien SSH untuk mengontrol perangkat jaringan. Pada penelitian ini akan dibandingkan kinerja Paramiko dan Netmiko dalam otomatisasi konfigurasi routing OSPF, tujuannya untuk dapat mengetahui kinerja keduanya baik dari segi waktu yang dihasilkan oleh program dalam memberikan perintah konfigurasi OSPF ke masing-masing Router dan waktu konvergensi jaringan OSPF setelah perintah konfigurasi diberikan. Selain itu, penelitian ini akan membandingkan penggunaan protokol SSH dan Telnet. Protokol SSH memiliki tingkat keamanan yang lebih baik daripada Telnet, sehingga keduanya akan dibandingkan dengan parameter pengujian yang sama seperti Paramiko dan Netmiko. Penelitian ini berhasil mengimplementasikan otomatisasi jaringan OSPF menggunakan library Paramiko dan Netmiko. Waktu proses konfigurasi OSPF menggunakan Paramiko lebih cepat dengan selisih 50,155 detik dari Netmiko. Waktu konvergensi setelah pemberian dengan Paramiko lebih cepat dengan selisih 50,17 detik dari Netmiko. Nilai throughput yang dihasilkan dengan menggunakan Paramiko lebih besar 30,27% dari Netmiko, sedangkan delay di Netmiko 30,21% lebih besar dari Paramiko.”

2.2.2 Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansible Pada Laboratorium Komputer

Pada penelitian sebelumnya yang dilakukan (M Rifki Afandia, Puspanda Hattab, Agus Efendi (2020)) dalam jurnal yang berjudul "***Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansibel Pada Labolatorium Komputer***". Otomatisasi jaringan merupakan sebuah solusi dalam mengimplementasikan beberapa pekerjaan yang rumit atau berulang. Dalam metode tradisional konfigurasi pada perangkat jaringan dilakukan dengan masuk kedalam perangkat jaringan untuk mengkonfigurasi perangkat jaringan tersebut, pekerjaan tersebut terlihat mudah bila yang dikonfigurasi masih satu atau dua perangkat jaringan, hal ini akan terlihat rumit jika terdapat banyak perangkat jaringan seperti router dan server yang harus dikonfigurasi. Ansible bertugas sebagai alat otomatisasi jaringan pada skala besar dan kecil pada sebuah jaringan komputer, serta Ansible bertugas untuk mempermudah pekerjaan dalam mengkonfigurasi perangkat jaringan.

2.2.3 Perbandingan Kinerja Library Paramiko dan Netmiko Dalam Proses Otomasi Jaringan

Proses konfigurasi perangkat jaringan dalam skala besar memerlukan waktu konfigurasi yang relatif lama dan apabila hal tersebut dilakukan secara manual oleh seorang network administrator, maka kemungkinan terjadinya kesalahan dalam menuliskan perintah konfigurasi menjadi semakin besar. Konsep otomasi jaringan dapat menjadi solusi dari permasalahan tersebut. Proses konfigurasi akan dilakukan secara terpusat. Dalam membuat konsep otomasi jaringan diperlukan bahasa pemrograman untuk mengkomunikasikan antara komputer pusat dengan semua perangkat dalam jaringan. Pada penelitian ini digunakan dua library dalam bahasa pemrograman Python yaitu Paramiko dan Netmiko untuk mengaktifkan proses routing dinamis pada perangkat router dimana protokol routing yang digunakan adalah OSPF. Dari hasil uji perbandingan performansi proses otomasi jaringan memperlihatkan bahwa penggunaan library Paramiko memberikan waktu konfigurasi ke perangkat router 4,14 kali lebih cepat dari Netmiko.

2.2.4 Penerapan Keamanan Remote Server Melalui Ssh Dengan Kombinasi Kriptografi Asimetris Dan Autentikasi Dua Langkah

Keamanan sistem menjadi syarat mutlak yang harus diperhatikan oleh pengguna, khususnya administrator jaringan guna melindungi data. Setiap melakukan pemeliharaan dan *monitoring* sistem, seorang administrator membutuhkan akses masuk dengan cara *remote* seperti *secure shell* (SSH) ke dalam sistem dengan saluran yang aman. Secara *default*, SSH tidak aman karena berpeluang besar akun diambil alih dengan teknik *brute force*. Penerapan kriptografi asimetris pada akun SSH dinilai aman daripada *remote login* SSH secara *default*. Akan tetapi hal tersebut masih belum cukup karena bisa saja peretas justru mengendalikan komputer yang digunakan oleh administrator dalam melakukan *remote* server. Selain itu juga dapat diretas dengan teknik *key scanning*. Kombinasi kriptografi asimetris dan autentikasi dua langkah dapat menjadi solusi sehingga server akan sangat susah ditembus.

2.2.5 *Python Network Automation Labs: SSH paramiko and netmiko*

Dalam jurnal *Python Network Automation Labs: SSH paramiko and netmiko* ini, Anda akan menggunakan pustaka SSH Python, paramiko dan netmiko, untuk mengontrol perangkat jaringan Anda. paramiko adalah apa yang Ansible andalkan untuk manajemen koneksi SSH ke perangkat jaringan, dan netmiko adalah versi paramiko yang ramah insinyur karena netmiko juga bergantung pada paramiko. Dengan mempelajari cara kerja modul jaringan ini, Anda dapat mempelajari cara kerja bagian dalam aplikasi lain yang mengandalkan modul jaringan ini. Di paruh pertama bab ini, Anda akan belajar mengganti tugas-tugas manual insinyur jaringan dasar menggunakan skrip Python dan pustaka paramiko. Di paruh kedua bab ini, Anda akan belajar menulis skrip Python menggunakan pustaka netmiko. Setelah Anda menguasai cara menggunakan modul SSH ini, Anda dapat segera menerapkannya ke pekerjaan Anda. Lab SSH ini akan menjadi landasan dalam mengembangkan aplikasi peningkatan IOS XE di akhir buku ini.

2.2.6 Future Library Network Automation

Michael Cooke,Jurnal Masyarakat Amerika untuk Ilmu Informasi 28 (5), 254-258, 1977, Jurnal ini menyarankan kemungkinan arah baru untuk

memecahkan beberapa masalah yang mengurangi jaringan perpustakaan kooperatif terkomputerisasi saat ini. Penulis berurusan dengan tiga masalah utama: berbagai tingkat kebutuhan bibliografi; kurangnya integrasi semua pemrosesan teknis perpustakaan dalam sistem otomatis; kemacetan dan biaya arsitektur jaringan komputer yang ada. Ide-ide yang disajikan menggunakan data terdistribusi dan komputasi terdistribusi untuk mendukung jaringan dan memperkenalkan sistem kegiatan bibliografi yang berjenjang untuk mengoptimalkan manfaat kerjasama

2.2.7 Analysis Of College English Teaching Model Bassed on Network Automation Transformation

Xiaoli Bao,Web Konferensi E3S 257, 02054, 2021.Pendidikan bahasa Inggris di perguruan tinggi dan universitas selalu dibatasi pada mode pengajaran konvensional dengan berbagai batasan seperti sistem perkuliahan, tujuan pengajaran dan kualitas siswa selama bertahun-tahun. Berdasarkan topik hangat transformasi otomatisasi, makalah ini akan menganalisis masalah yang dihadapi pengajaran bahasa Inggris di perguruan tinggi dan universitas dari perspektif pendidikan Internet+ paling populer, mengumpulkan dan mengatur data yang relevan, dan kemudian mengajukan saran yang relevan untuk pengembangan bahasa Inggris. mengajar di perguruan tinggi dan universitas, dan mengedepankan prospek pengembangannya di masa depan, dengan tujuan mempromosikan transformasi mode pengajaran bahasa Inggris perguruan tinggi dalam rangka meningkatkan kualitas pengajaran kelas bahasa Inggris.

2.2.8 The Effectiveness of Automatic Network Administration(ANA) in Network Automation Universitas Pendidikan Ganesha

GS Santyadiputra, IME Listartha, GAJ Saskara,Jurnal Fisika: Seri Konferensi 1810 (1), 012028, 2021.Kegiatan administrasi jaringan menimbulkan tantangan skalabilitas. Tantangan muncul ketika perangkat yang ditangani sangat banyak. Alih-alih menggunakan metode tradisional, organisasi TI perlu menerapkan metode otomatisasi. Sangat tidak efisien untuk mengelola sejumlah besar perangkat jaringan dengan masuk ke dalamnya satu per satu. Kesalahan

manusia sangat rentan terjadi, yang akan menyebabkan inkonsistensi konfigurasi, hingga kesalahan konfigurasi, yang akan berdampak pada buruknya layanan jaringan yang disediakan. Metode otomatisasi pada jaringan dikenal sebagai Network Automation (NA). NA menggunakan pendekatan algoritmik dalam menyusun langkah-langkah dalam pemecahan masalah. NA dijalankan dalam bentuk program aplikasi yang memiliki fungsi utama mengotomatisasi aktivitas rutin, kompleks, berulang, dan komprehensif. Aktivitas administrasi jaringan yang dapat diotomatisasi meliputi otomatisasi perutean, administrasi virtual, dan pencadangan atau pemulihan. Simulasi NA dilaksanakan oleh ANA. ANA adalah pengujian yang dilakukan dengan menggunakan bahasa pemrograman Python dengan memanfaatkan Paramiko dan Netmiko Library dalam lingkungan virtual menggunakan GNS3 dan Virtual Box. Tujuan dari pengembangan ANA adalah agar kegiatan administrasi jaringan berjalan lebih efisien. Hasil penerapan ANA dapat dijadikan bahan pertimbangan dalam kegiatan administrasi jaringan. Hasil penelitian menunjukkan ANA efektif untuk membuktikan tujuan NA.

2.2.9 A Self-Organizing Defense Technique for SSH Dictionary Attack

V Akilandeswari, S Mercy ShalinieInstitut Informasi Internasional (Tokyo). Informasi 17 (2), 821, 2014,Tujuan utama artikel ini adalah untuk mengusulkan model mitigasi yang mencegah server menjadi tuan rumah untuk merumuskan jaringan yang disusupi melalui serangan kamus SSH. Untuk mendeteksi serangan SSH, studi entropi pada lalu lintas kueri DNS dibuat, dan algoritma deteksi berbasis pola lalu lintas aktif diimplementasikan. Ini menghitung jarak Euclidian dari alamat IP dan menyebutnya sebagai kata kunci permintaan DNS. Kemudian pola trafik query DNS diestimasi untuk menentukan perubahan trafik. Dengan pola yang diamati, model yang diusulkan penulis mendefinisikan kebijakan pemblokiran untuk menghentikan serangan SSH sebelum mereka mendapatkan kesempatan untuk mendapatkan akses ke server SSH. Kebijakan pemblokiran dibingkai menggunakan distribusi bersama lalu lintas permintaan permintaan DNS dan aktivitas pengguna. Selain itu, model pertahanan yang mengatur dirinya sendiri memberikan keamanan yang lebih besar

melalui kebijakan pemblokiran proaktif dengan tingkat deteksi tinggi dan tingkat positif palsu yang rendah.Lihat di search.proquest.com Artikel terkait yadda.icm.edu.pl,Perbandingan jaringan WLAN 2,4 dan 5 GHz untuk tujuan lokasi indoor dan outdoor,ukasz Chruszczyk, Adam Zając, Damian Grzechca Jurnal Internasional Elektronika dan Telekomunikasi 62, 2016.Makalah ini menyajikan perbandingan prototipe sistem lokasi yang dibangun dengan komponen standar infrastruktur jaringan WLAN 2,4 dan 5 GHz. Sistem ini dapat digunakan untuk pemosisian pribadi atau objek lain, baik untuk lingkungan indoor maupun outdoor. Sistem bersifat lokal, yaitu wilayah operasionalnya terbatas pada jangkauan operasi jaringan WLAN. Sistem ini didasarkan pada komponen WLAN standar dan tersedia secara luas (titik akses, adaptor jaringan). Tujuannya adalah untuk menghindari modifikasi perangkat keras dan perangkat lunak. Juga perhitungan posisi tidak boleh menjadi operasi yang haus daya. Metode lokasi didasarkan pada Received Signal Strength Indication (RSSI) yang dikembalikan oleh sebagian besar IC RF (termasuk WLAN). Fokus utama adalah penelitian tentang seberapa banyak akurasi (dan kegunaan) yang dapat diharapkan dari perangkat keras WLAN standar. Skenario statis dan dinamis telah diuji dan dibandingkan.

2.2.10 An Automatic Remote Monitoring System for Large Network

Daiana Tomescu, Adelaida Heiman, Alina Badescu,2019 IEEE International Conference on Computational Science and Engineering (CSE) dan IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 71-73, 2019.Pemantauan fungsionalitas jaringan besar merupakan masalah yang sulit dan semakin banyak ditemui akhir-akhir ini karena perluasan jaringan yang ada dan perkembangan aplikasi yang membutuhkan jaringan sensor, sistem deteksi, dan jaringan telekomunikasi yang besar. Tujuan utama dari makalah ini adalah untuk secara otomatis memantau jaringan yang terdiri dari sejumlah besar sensor (misalnya 124) yang ditempatkan di permukaan yang luas (lebih dari 20 km persegi). Data yang terkumpul digunakan untuk memodelkan fenomena fisik di atmosfer, sehingga kualitas data menjadi penting dan dipengaruhi oleh fungsi

peralatan yang salah. Solusi yang diusulkan berada di luar tujuan ilmiah asli jaringan dan dapat berhasil diterapkan di aplikasi lain atau lainnya