

BAB II

LANDASAN TEORI

2.1. Tinjauan Penelitian

Pada bab ini terdiri dari definisi-definisi dan kajian literatur yang digunakan sebagai acuan dan landasan teori untuk membantu penyusunan tugas akhir ini

2.1.1. Network Automation

Menurut (Islami et al., 2020) Istilah *Network Automation* berasal dari bagaimana cara melakukan manajemen, *testing*, dan *deployment* dengan menggunakan program yang berjalan pada *control node*. Dengan metode *Network Automation*, tugas teknisi atau administrator jaringan menjadi lebih ringan dalam menangani jumlah konfigurasi perangkat pada jumlah yang banyak, sehingga mengurangi waktu dan kesalahan manusia dalam melakukan konfigurasi karena semua konfigurasi sudah tertulis pada *file*, maka setiap konfigurasi akan dilakukan secara otomatis, *file* yang berisikan konfigurasi tersebut akan digunakan untuk menjalankan konfigurasi yang berulang. Ada banyak metode untuk melakukan *Network Automation* seperti *Network Programmability* dengan menggunakan bahasa pemrograman Python dan *library* seperti Paramiko dan Sqlite3. Ada juga *tools* untuk melakukan *Automation* seperti Ansible, Terraform, SaltStack, Chef, Puppet, dan Napalm. Pada penelitian ini penulis menggunakan metode *Network Programmability* dengan Python dan *library* Paramiko dan Sqlite3.

Menurut (Red Hat, 2021) Pada jaringan tradisional, konfigurasi dan *update* secara manual pada perangkat jaringan terlalu lambat dan tinggi tingkat *error*. *Network Automation* memberikan kontrol lebih ke kebutuhan jaringan, berikut beberapa hal yang dapat ditingkatkan dengan *Network Automation* :

- Ketersediaan jaringan untuk *testing* dan manajemen,
- Meningkatkan kemampuan administrator untuk respon terhadap perubahan
- Mengidentifikasi kesalahan dan melakukan *troubleshoot* pada seluruh jaringan

- Kemampuan melihat keseluruhan konfigurasi pada satu *platform* untuk menyesuaikan kebutuhan

1. Jenis *Network Automation*

(Edelman et al., 2018) Otomatisasi selalu dikaitkan dengan kecepatan, dan beberapa hal pada bagian *Network* tidak selalu membutuhkan kecepatan baik itu konfigurasi atau kebutuhan sehari-hari. Sehingga penerapan otomatisasi tidak terlalu efisien pada bagian yang tidak tepat. *Network Automation* memiliki beberapa jenis untuk setiap kebutuhan. Berikut jenis *Network Automation*:

- *Device Provisioning*

Device Provisioning merupakan jenis yang termudah dan cepat untuk memulai melakukan otomatisasi dengan cara membuat sebuah *file* konfigurasi yang akan digunakan untuk otomatisasi pengiriman konfigurasi ke perangkat jaringan.

Terdapat dua langkah untuk menjalankan *Device Provisioning*, langkah pertama dengan cara membuat *file* konfigurasi, dan langkah kedua mengirim *file* tersebut ke perangkat.

Otomatisasi pada *Device Provisioning* memerlukan informasi *syntax* konfigurasi dari setiap vendor perangkat, sehingga perlu dipisahkan antara *value* (nilai yang ingin diberikan) dan *syntax*

- *Data Collection*

tools untuk monitoring umumnya menggunakan *Simple Network Management Protocol (SNMP)* untuk mengambil data Management Information Base (MIB) dan menyimpan ke dalam *tool* monitoring. Berdasarkan data tersebut tidak semua data yang disimpan akan terpakai. Pada *Data Collection* memfokuskan kontrol terhadap apa yang dibutuhkan dan bagaimana data tersebut ditampilkan

- *Reporting*

Data Collection dan *Reporting* merupakan jenis *automation* yang berkaitan, dengan data yang didapat dari proses otomatisasi pengumpulan

data pada *Data Collection* maka diperlukan sebuah laporan untuk mendapatkan informasi dari data tersebut. *Reporting*

2.1.2. Dashboard

Dashboard merupakan tampilan dari data dalam bentuk visualisasi menarik yang dapat dengan mudah dipahami oleh pengguna atau para pengambil keputusan. Menurut (Sofyan et al., 2020) Sistem *Dashboard*, data dan informasi dapat ditampilkan dengan cepat dan mudah dipahami, sehingga memberikan kejelasan informasi penting untuk pengambilan keputusan.

Dashboard memiliki beberapa tipe dengan fungsi berbeda pada masing-masing dashboard, menurut (Walewowan et al., 2020), yaitu:

1. Strategic dashboard digunakan untuk mendukung manajemen level strategis memberikan informasi dalam membuat keputusan bisnis, memprediksi peluang, dan memberikan arahan pencapaian tujuan strategis
2. Analytical dashboard berfokus pada proses analisis untuk menentukan penyebab dari suatu kondisi atau kejadian tertentu.
3. Operational dashboard yang berfungsi sebagai pendukung monitoring dari aktifitas proses bisnis yang spesifik. Fokus pada monitoring aktifitas dan kejadian yang tidak berubah secara konstan

2.1.3. WiFi

(Muhammad & Hasan, 2016a) *Wireless* (nirkabel) merupakan teknologi yang memungkinkan bertukar data tanpa media kabel, *Wireless Fidelity (WiFi)* adalah perangkat standar yang digunakan untuk komunikasi jaringan lokal tanpa kabel yang didasari pada spesifikasi IEEE 802.11.

Berikut beberapa istilah yang digunakan dalam jaringan *wireless* :

- *Wireless Local Area Network (WLAN)* merupakan jaringan *wireless* yang menggunakan gelombang radio sebagai media komunikasi. Frekuensi yang digunakan pada jaringan *wireless* adalah 2.4 Ghz dan 5 Ghz.

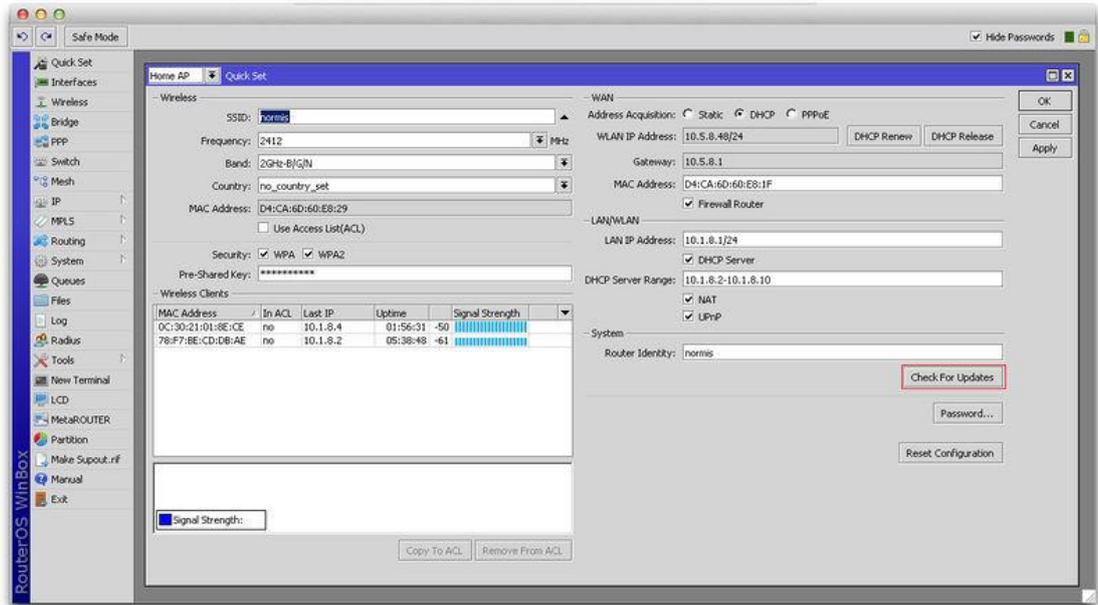
- *Wireless Fidelity (WiFi)* merupakan merek yang menggantikan istilah IEEE 802.11, pada awalnya *WiFi* diperuntukan hanya untuk jaringan lokal, namun sekarang dapat digunakan untuk mengakses *Internet*.
- *Hotspot* adalah pemanfaatan dari teknologi WLAN pada tempat-tempat umum.
- *Captive Portal* halaman *web* untuk memasukan *username* dan *password* pengguna *hotspot* yang ingin mengakses *internet*. Saat user mengakses *hotspot* maka akan diarahkan (*redirect*) ke *captive portal*.
- *Service Set Identifier (SSID)* identifikasi atau nama dari jaringan *wireless*.

2.1.4. MikroTIK

(Ahamed, 2018) MikroTik adalah sebuah perusahaan asal Latvia yang didirikan pada tahun 1996 untuk pengembangan struktur ISP Router dan *WiFi*, MikroTik menyediakan *Hardware* dan *Software* untuk koneksi internet diberbagai bidang. Pada tahun 1997 MikroTik membuat software RouterOS yang menyediakan stabilitas, kontrol dan fleksibilitas untuk data interface dan routing. Pada tahun 2002 MikroTik memiliki *hardware* sendiri yaitu *RouterBoard*. Berikut penjelasan dari RouterOS dan RouterBoard menurut (Fitria & Prihanto, 2018):

- MikroTIK *RouterOS*

MikroTik *RouterOS* merupakan perangkat lunak yang dapat diinstal pada *Personal Computer (PC)* melalui CD. File yang dibutuhkan dapat diunduh dalam bentuk *file image* MikroTik RouterOS dari website resmi MikroTik, www.mikrotik.com. Namun, *file image* ini merupakan versi *trial* MikroTik yang hanya dapat dalam waktu 24 jam saja. Untuk dapat menggunakannya secara *full time*, harus membeli lisensi *key* dengan catatan satu lisensi hanya untuk satu *harddisk*.

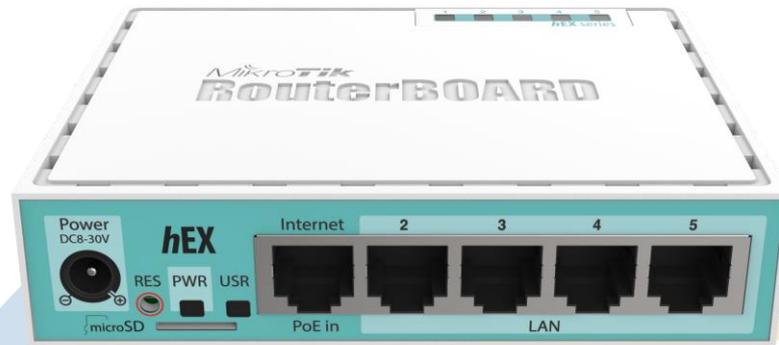


Gambar 2.1 MikroTIK RouterOS

- MikroTIK *RouterBoard*

Merupakan MikroTik dalam bentuk perangkat keras yang khusus dikemas dalam *board router*, atau sering disebut *routerboard*, yang di dalamnya sudah terinstal sistem operasi MikroTik RouterOS. Untuk versi ini, lisensi sudah termasuk dalam board MikroTIK. Pada *RouterBoard* ini pengguna langsung dapat memakainya, tanpa harus melakukan instalasi sistem operasi. *RouterBoard* ini dikemas dalam beberapa bentuk dan kelengkapannya sendiri sendiri. Ada yang difungsikan sebagai *Indoor Router*, *Outdoor Router* maupun ada yang dilengkapi dengan *wireless router*

STT - NF



Gambar 2.2 MikroTIK RouterBoard

(Muhammad & Hasan, 2016b) MikroTik RouterOS dapat dijadikan *router* berbasis komputer yang banyak digunakan untuk menjalankan beberapa aplikasi selain routing seperti dapat digunakan sebagai manajemen akses seperti *bandwidth*, *firewall*, *wireless access point*, hotspot, dan lainnya. Perangkat MikroTIK baik *hardware* ataupun *software* membutuhkan lisensi berbayar untuk dapat menggunakan fitur secara penuh. Lisensi MikroTik terdiri dari *level*, mulai dari *level 0* hingga *level 6*. Setiap *level* memiliki kemampuan yang berbeda. Berikut penjelasan setiap *level* lisensi MikroTik:

- *Level 0* (gratis). *Level* ini tidak membutuhkan lisensi untuk menggunakannya dan penggunaan fitur hanya dibatasi selama 24 jam setelah instalasi dilakukan.
- *Level 1* (demo). Pada *level* ini *user* dapat menggunakannya sebagai routing standar dengan 1 pengaturan, dan tidak memiliki limitasi waktu untuk menggunakannya.
- *Level 3*. *Level* ini mencakup *level 1* ditambah kemampuan untuk memajemen *hardware* berbasis kartu jaringan atau *Ethernet* dan pengelolaan perangkat *wireless tipe client*.

- *Level 4.* *Level* ini mencakup *level 1* dan *3* ditambah kemampuan untuk mengelola perangkat *wireless tipe access point*.
- *Level 5.* *Level* ini mencakup *level 1, 3, dan 4* ditambah kemampuan mengelola jumlah pengguna hotspot yang lebih banyak.
- *Level 6.* *Level* ini mencakup semua *Level* dan tidak memiliki limitasi atau batasan apapun.

A. MikroTIK Hotspot

Menurut (Fadlan Abdillah Hasibuan & Subhiyanto, 2021) Mikrotik Hotspot adalah sebuah fitur autentikasi *user* dan *password* untuk pengguna mengakses suatu jaringan internet melalui perangkat MikroTIK. Hotspot merupakan fitur gabungan dari berbagai *service* yang ada di Mikrotik, antara lain:

1. *DHCP (Dynamic Host Control Protocol) server*, digunakan untuk memberikan layanan *IP* otomatis ke *user*.
2. *Firewall NAT*, untuk mentranslasi *IP* user ke *IP* yang bisa dikenali ke *internet*.
3. *Firewall filter*, untuk memblock user yang belum melakukan *login*.
4. *Proxy*, untuk memberikan tampilan halaman *login*.

2.1.5. Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpreter, interaktif, *object oriented* dan dapat berjalan di hampir semua *platform*. Python sebagai bahasa tingkat tinggi yang mudah untuk dipelajari karena sintaksnya yang jelas dan juga elegan, karena sintaksnya lebih menggunakan bahasa manusia daripada bahasa komputer, dan memiliki modul - modul yang efisien dan siap langsung digunakan. *Source code* bahasa Python akan dikompilasi menjadi format *bytecode* yang akan dieksekusi. Kode Python lebih lambat dieksekusi dibandingkan dengan bahasa pemrograman lain yang bersifat *low-level* (Sivam, 2018).

Keunggulan Bahasa Pemrograman Python Menurut (Kadarina & Ibnu Fajar, 2019) :

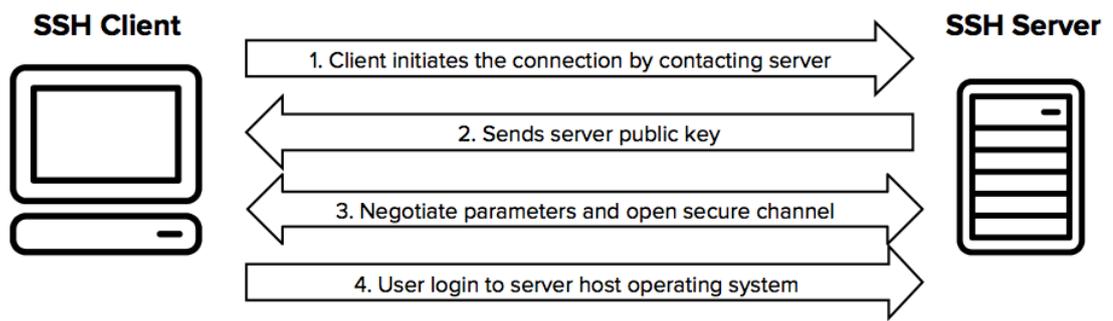
- Merupakan bahasa pemrograman tingkat tinggi yang mudah dipelajari karena sintaks yang jelas dan mudah dibaca karena lebih mendekati bahasa manusia.
- Tersedia banyak *library* yang dapat digunakan, yang kebanyakan ditulis oleh bahasa C.
- Bahasa Python dapat berjalan diberbagai *platform* tanpa harus menulis kode untuk *platform* tertentu.
- Dapat digunakan untuk mengembangkan berbagai hal seperti *Software, Hardware, Internet of Things, Web Development, Video Game, dan Mobile Apps*

A. Paramiko

(Mihailă et al., 2017) Paramiko merupakan *library* Python versi 2.7 dan Python versi 3.4+ yang mengimplementasikan protokol SSH (*Secure Shell*) versi 2, menyediakan fungsi pada sisi *client* dan *server*, paramiko merupakan *interface* Python asli pada jaringan *SSH* dan memanfaatkan ekstensi Python C pada tingkat kriptografi rendah.

- Secure Shell (SSH)

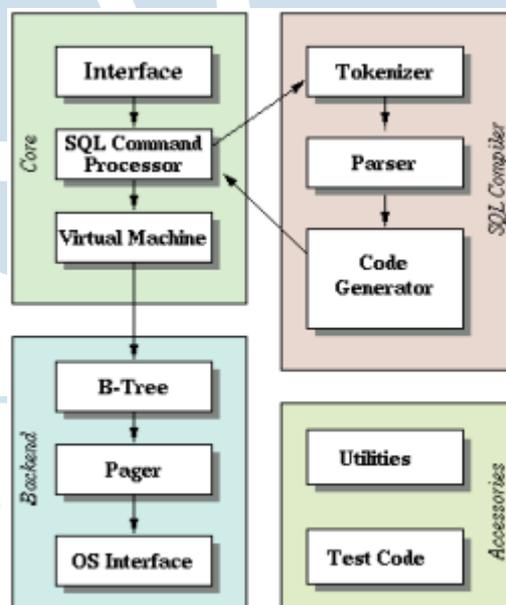
Secure Shell (SSH) merupakan protokol yang digunakan untuk melakukan *remote access* dan manajemen perangkat melalui jaringan dengan menggunakan enkripsi untuk pengamanan data dalam jaringan. *SSH* memerlukan proses *session* untuk terhubung dengan komputer *client*. *SSH* merupakan protokol *client-server* sehingga perangkat yang menjadi *server* dapat dikontrol oleh perangkat yang menjadi *client*. Pada prosesnya *client* mengirimkan *request* terhadap *server*, *server* akan menerima *request* tersebut melalui *port 22* yang mana merupakan *port* standar *SSH*, setelahnya *client* dapat menginisiasi koneksi terhadap *server* tetapi diperlukan *permission user-password* setelah itu *client* dapat mengontrol perangkat *server* seperti pada Gambar 2.3 (Karki, 2021).



Gambar 2.3 Komunikasi SSH

2.1.6. Database SQLite3

Pengembangan *Dashboard* ini membutuhkan *Database* yang digunakan untuk melakukan penyimpanan data, dalam penelitian ini akan menggunakan SQLite versi 3 (SQLite3). Menurut (S.T. Bhosale et al., 2015), SQLite3 merupakan *database* yang implementasinya dapat melakukan *self-contained*, *serverless*, dan sebagainya. Sehingga penggunaan SQLite3 lebih ringan, SQLite3 dapat digunakan pada bahasa pemrograman seperti C, C++, BASIC, C#, Python, Java and Delphi. Adapun Arsitektur dari SQLite3 dapat dilihat pada gambar 2.4 berikut:

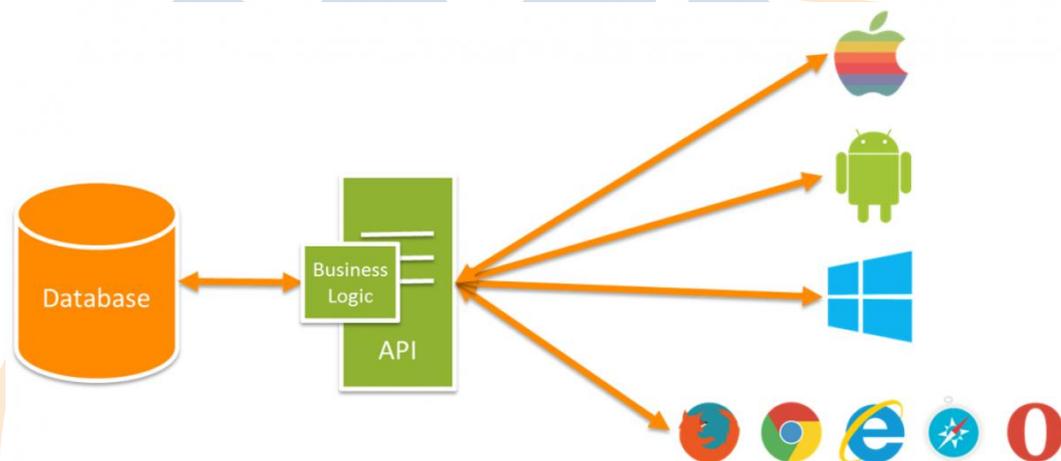


Gambar 2.4 Arsitektur SQLite3

Fitur pada SQLite3 sebagai berikut:

- SQLite merupakan *database* yang *atomic, consistent, isolated,* dan *durable*
- Tidak perlu administrasi
- Mendukung *database* dengan ukuran *terabyte*
- Simple dan mudah untuk *API*
- Dapat digunakan untuk berbagai bahasa pemrograman
- Dapat digunakan untuk *cross-platform*

2.1.7. *Application Programming Interface (API)*



Gambar 2.5 *Application Programming Interface*

Menurut (Meng et al., 2018), *Application Programming Interfaces (API)* digunakan untuk membuka layanan atau data yang disediakan oleh aplikasi *software* melalui serangkaian sumber daya yang ditentukan, seperti metode, objek atau *URI (Uniform Resource Identifier)*. Sumber daya yang dibuka ini maka berbagai aplikasi dapat mengakses data atau layanan tanpa harus mengimplementasikan prosedur dibawahnya. *API* menyediakan abstraksi tingkat tinggi yang dapat melakukan tugas pemrograman, mendukung desain dan pembagian dari pendistribusian aplikasi *software* dan penggunaan ulang kode.

2.1.9. Sistem Operasi Ubuntu

Ubuntu merupakan salah satu dari distribusi Linux Debian, nama Ubuntu sendiri berasal dari bahasa Afrika yang berarti “Kemanusiaan terhadap sesama manusia”. Ubuntu merupakan sistem operasi yang berbasis Linux yang bisa didapat secara bebas dan memiliki dukungan dari komunitas dan ahli. Proyek Ubuntu disponsori oleh perusahaan Canonical Ltd yang berasal dari Afrika Selatan. Tujuan dari distribusi Linux Ubuntu adalah membawa semangat yang terkandung di dalam filosofi Ubuntu ke dalam dunia perangkat lunak. Ubuntu merupakan sistem operasi berbasis linux, ubuntu tersedia secara gratis dengan dukungan komunitas dan profesional. Ubuntu dapat digunakan untuk *desktop* dan *server*. Ubuntu *desktop* memiliki standar aplikasi *desktop* seperti *word processing*, *spreadsheet application*, *internet access application*, *web server software*, *email software*, *programming language*, dan lainnya. Ubuntu *server* didesain khusus dengan kernel yang dikustom untuk bekerja sebagai sistem operasi untuk menjadi *server*, kernel linux ubuntu *server* dapat bekerja dengan *multiprocessor* dan menggunakan penjadwalan *input/output (I/O)*, ubuntu *server* memiliki lisensi *open source* dan sebagai turunan dari distro linux debian. Ubuntu *server* dapat digunakan dengan syarat minimum perangkat dengan 300 Mhz *processor*, 64MB *memory*, 500MB *hard drive*, dan VGA 640 × 480 (Idhom et al., 2018).

2.1.10. Flask Framework

Menurut (Irsyad, 2018) Flask merupakan *web framework* yang ditulis dengan bahasa Python. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu *web*. Dengan menggunakan Flask dan bahasa Python, pembuatan *web* menjadi terstruktur dan dapat mengatur *behaviour* suatu *web* dengan lebih mudah. Flask termasuk pada jenis *micro framework*, *micro pada framework* Flask bermaksud untuk membuat *core* aplikasi tetap sederhana tetapi dapat dengan mudah dikembangkan, Flask secara *default* tidak menyediakan pustaka (*library*) umum seperti *database*, *form validation*, dan sebagainya karena sudah didukung oleh pustaka yang disediakan oleh pihak ketiga.

Karena Flask merupakan *microframework* yang berarti Flask mempunyai *core* yang sangat sederhana dan kecil, fitur-fitur bawaan dari Flask sendiri termasuk sedikit jumlahnya, diantaranya adalah

- *Built-in development server.*
- *Debugger* cepat.
- *Integrated support* untuk pengetesan unit.
- Kompatibel dengan mesin aplikasi Google.
- *RESTful request dispatching.*
- *Jinja2 templating.*
- Mendukung *secure cookies.*
- Berbasis *unicode.*
- Mengikuti WSGI 1.0

Karena Flask membutuhkan aplikasi pihak ketiga untuk menjalankan fiturnya maka ini bisa menjadi keunggulan dan kekurangan dari *framework* Flask, berikut adalah keunggulan dan kekurangan dari Flask:

- Ringan untuk dijalankan karena mempunyai *core* yang sederhana dan desain modular.
- Dapat menangani fungsi HTTP (*Hypertext Transfer Protocol*) *request* dengan mudah.
- *API (Application Programming Interface)* yang baik dan koheren.
- Dokumentasi yang banyak dan terstruktur dengan baik, penuh dengan contoh yang dapat digunakan langsung.
- Mudah untuk dipasang dan di-*deploy* untuk produksi.
- Mudah untuk diperiksa secara menyeluruh. (*Unit testability.*)
- Fleksibilitas tinggi, dengan konfigurasi yang sangat mudah diubah.
- Tidak memiliki *ORM (Object-relational mapping)* dan *database layer* bawaan, sehingga harus menggunakan aplikasi dari pihak ketiga.

2.2. Penelitian Terkait

Penelitian ini tidak secara keseluruhan hal yang baru, melainkan sudah ada penelitian sebelumnya yang terkait, berikut adalah tabel yang membahas penelitian terkait yang peneliti ambil untuk dijadikan referensi :

Tabel 2.1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Hasil
1	(Fariz, 2020)	Perancangan dan Implementasi <i>Zero Touch Provisioning</i> untuk pembuatan MikroTIK Hotspot	<i>Zero Touch Provisioning</i>	Sistem <i>ZTP</i> yang dapat mengirimkan konfigurasi ke perangkat yang terhubung
2	(Wiryawan & Rosyid, 2019)	Pengembangan Aplikasi otomatisasi administrasi jaringan berbasis <i>website</i> menggunakan bahasa pemrograman Python	<i>Network Management</i>	Telah berhasil dikembangkan aplikasi otomatisasi jaringan berbasis <i>web</i> dengan <i>library</i> paramiko dan <i>web framework</i> django dengan lima fitur konfigurasi <i>Routing, Vlan, Backup, Restore, dan Setting</i> . Metode pengujian <i>Black-box</i> menunjukkan semua fungsi pada aplikasi yang dikembangkan berfungsi baik dan berhasil

				diterapkan pada <i>vendor</i> Cisco dan Mikrotik
3	(Ginting & Hadi, 2020)	<p>Pengujian Konfigurasi Otomatis Penambahan <i>Gateway</i> pada <i>Virtual Router</i> menggunakan Aplikasi Otomatisasi Jaringan berbasis <i>Web</i></p>	<i>Network Automation</i>	<p>Aplikasi otomatisasi jaringan berbasis <i>web</i> ini dapat membantu administrator jaringan dalam melakukan konfigurasi penambahan <i>gateway</i> otomatis pada sebuah topologi jaringan yang terhubung dengan aplikasi <i>GNS3</i>. Konfigurasi otomatis penambahan <i>gateway</i> pada <i>virtual router</i> menggunakan aplikasi otomatisasi jaringan berhasil dilakukan yang dibuktikan dengan pengujian koneksi. Aplikasi otomatisasi jaringan berbasis <i>web</i> ini dapat berfungsi dengan baik yang dibuktikan dengan melakukan pengujian metode <i>black-box testing</i>.</p>