**Sistem Manajemen User Hotspot Berbasis Web Menggunakan API**

**(Yunan Sawiji, 13111070,** [**yunansawijo@gmail.com**](mailto:yunansawijo@gmail.com)**)**

**(Imam Suharjo, ST., MT. 0503019201,** [**imam@mercubuana-yogya.ac.id**](mailto:imam@mercubuana-yogya.ac.id)**)**

**ABSTRAK**

Peranan teknologi yang semakin canggih menuntut layanan yang semakin mudah, cepat dan praktis terutama dalam pengoperasian perangkat *hotspot*. Banyakyang mampu dijangkau oleh masyarakat menyebabkan perkembangan pola pikir bahwa layanan *internet* dapat diakses di mana saja. Salah satu yang memanfaatkan teknologi ini adalah untuk menajemen *user* *hotspot.*

Penelitian ini merancang sistem manajemen berbasis *web* dengan memanfaatkan *Mikrotik API* sebagai penunjang manajemen *bandwidth*. Sistem dirancang menggunakan metode *API*. Penelitian membagi kedalam tiga bagian yaitu pengumpulan data, kebutuhan pengembangan sistem dan metodelogi pengembangan sistem. Proses pengumpulan data pengguna *WiFi*. Kebutuan selama pengembangan sistem disesuaikan dengan *device* yang ada, sedangkan *metodelogi* pengembangan sistem menggunakan pendekatan *Waterfall Approach.*

Berdasarkan dari hasil pengujian segi fungsional sistem dan *server* menunjukkan hasil bahwa fungsionalitas sistem dan *server* telah berjalan dengan baik. Hasil pengujian bahwa sistem ini mudah dipahami dan membantu dalam manajemen *user hotspot* berbasis *web* secara *online*, sehingga memudahkan pemilik *WIFI* menagatur pengguna *WIFI.*

Analisa dan perancangan program, pada sistem *Aplikasi Web* berbasis *API* dapat disimpulkan bahwa sistem manajemen berbasis *API* dapat menyelesaikan permasalahan dalam penggunaan *internet* secara adil. Sistem dapat memberikan kemudahan kepada operator untuk mengakses *mikrotik* dikarenakan bisa diakses dimana saja dan fitur tampilan dapat disesuaikan dengan kebutuhan dari operator.

**Kata Kunci**: *Mikrotik API*, *PHP*, *MySQL*, *uptime limit, wifi local, hotspot, device*

ABSTRACT

The role of technology which gets more and more sophisticated demands easier, faster and more practical service provision, especially in operating hotspot devices. A lot of things can be reached by people, which makes them think that internet service can be accessed from anywhere. One of the things that enables it is by using hotspot user management.

This research designs a web-based management system by using an API Mikrotik router as a bandwidth management support. The system was designed using an API method. This research is divided into three parts, namely data collection, system development needs, and system development methodology. The Wi-Fi users’ data collection process for the need during the system development is adjusted to the available device, whereas the system development methodology used a waterfall approach.

Based on the test results in terms of system and server functionality, it was shown that the system and server functionality was running well. The results showed that the system was easy to understand and can help in web-based hotspot user management, which facilitated Wi-Fi owners to manage Wi-Fi users.

In terms of program analysis and designing, with the use of API-based web application system it can be concluded that API-based management system can solve problems in internet use fairly. The system can provide facility to the operator to access the Mikrotik router because it can be accessed from anywhere, and the display features can be adjusted to the operator’s needs.

**Keywords**: *API Mikrotik*, *PHP*, *MySQL*, *uptime limit, wifi local, hotspot, device*

# PENDAHULUAN

Salah satu kebutuhan manusia pada masa sekarang adalah *internet*. Seiring dengan perkembangan masyarakat mempengaruhi akan kebutuhan teknologi di Indonesia, Sebagian penyedia layanan *WIFI* memanfaatkan aplikasi perangkat lunak untuk mensetting *hotspot* dan menajemen *hotspot*. Masih banyak menggunakan cara manual dalam memanajemen dan pemanfaatan *Internet*. Pemanfaatan teknologi web adalah terciptanya manajemen *user hotspot* melalui *website*, lewat *website* bisa diatur untuk pembatasan pengguna *hotspot*, dalam mangatur pengguna *hotspot* dimana pun bisa diakses yang terpenting ada jaringan *Internet*. Penggunaan layanan *Internet* pada umumnya diakses oleh pengguna tidak tetap, sehingga dibutuhkan teknik atau metode yang dapat mengalokasikan *bandwidth* secara dinamis dengan batasan pengguna yang masih dapat terkontrol (Asnawi, 2017).

Kondisi saat ini masih menggunakan cara secara manual untuk mengatur pengguna *hotspot* dan pembebasan penggunaan akses *Internet* oleh pengguna internet. Sebuah *website* lokal dan sistem manajemen penggunaan akses Internet yang mampu untuk memberikan solusi dalam manajement user *hotspot* berbasis web. Mekanisme sistem secara ringkassebagai penggunasupayatidak bisa menggunakan *kuota* *Internet* secara berlebihan, sebagai pemilik *hotspot* membuka laptop maka langsung bisa mengakses *website* dan terbuka tampilan *web browser* pertama dari *link mikrotik* yang sudah terbuka.

Berdasarkan latar belakang tesebut, penulis berusaha merancang sebuah sistem yang dapat melakukan manajement secara *online* dan mengangkatnya menjadi sebuah penelitian yang bejudul “Sistem Manajemen *User Hotspot* Berbasis *Web* Menggunakan *API*”. Adanya penelitian ini diharapkan adanya perubahan yang signifikan dalam mempermudah dalam manajemen *user hotspot* dapat terealisasi (Mardiana, 2015).

# TINJAUAN PUSTAKA PUSTAKA & LANDASAN TEORI

## **2.1 Tinjauan Pustaka**

Mengacu pada jurnal judul “**Sistem Manajemen *User Hotspot* Berbasis *Web* Menggunakan *API.*** Dengan objek penelitiannya adalah *router mikrotik*. Sistem manajemen *user hotspot* dibuat dengan *platfrom web* dengan menggunakan bahasa pemograman *php*. Hasil dari penelitian berupa pengaturan *bandwidth* dengan lebih efektif.

Berdasarkan pada jurnal yang berjudul “manajemen *user hotspot* menggunakan *API*” Peneliti membagi *bandwidth* per-*client* sebesar 1*Mbp*s. Pembagian *bandwidth* dilakukan dengan menggunakan *aplikasi web*. Untuk membuktikan penulis melakukan uji coba pertama dengan mendownload sebuah *file* berukuran 105MB dari sebuah *website* dan melihat hasil limitasi *bandwidth* yang telah dibuat apakah sesuai dengan yang diharapkan. Dari pengujian yang dilakukan didapatkan hasil kecepatan rata-rata data *transfer* menunjukan hasil pengukuran sesuai dengan limit *bandwidth* yang di tetapkan dan apakah sudah sesuai dengan yang di tetapkan (Anwar, 2017). Pada penelitian yang berjudul “Teknik *mark packet* dan *mark connection* untuk manajemen *bandwidth* dengan pendekatan *top-down*”; Berlaku *paket mark* dan teknik koneksi tanda untuk membagi kapasitas *bandwidth* yang tersedia dan rata-rata monitor lalu lintas data untuk mendukung kebutuhan layanan jaringan dengan itu setiap *client* dapat bekerja dengan kapasitas *bandwidth* dalam mengakses *internet*. Hasil penelitian ini menunjukan cara mengalokasikan angka *bandwidth* yang sedang tersedia dalam setiap pengguna untuk memiliki akses *internet* yang stabil dan lazim dan meringankan operator dalam memantau pengguna (Kosasi, 2013).

## **2.2** **Sistem Manajemen *Bandwidth***

Manajemen *Bandwidth* Menggunakan *Simple Queue* pada sebuah jaringan yang mempunyai banyak *client*, diperlukan sebuah mekanisme pengaturan *bandwidth* dengan tujuan mencegah terjadinya monopoli penggunaan *bandwidth* sehingga semua client bisa mendapatkan jatah *bandwidth* masing-masing. *QOS*(Quality *of services*) atau lebih dikenal dengan *Bandwidth* Manajemen, merupakan metode yang digunakan untuk memenuhi kebutuhan tersebut.   
Pada *RouterOS Mikrotik* penerapan *QoS* bisa dilakukan dengan fungsi *Queue*.  Cara paling mudah untuk melakukan *queue* pada *RouterOS* adalah dengan menggunakan *Simple Queue*, melakukan pengaturan *bandwidth* secara sederhana berdasarkan *IP Address client* dengan menentukan kecepatan *upload* dan *download* maksimum yang bisa dicapai oleh *client*.

**Metode Pembagian *Bandwidth Share***

Selain digunakan untuk melakukan manajemen *bandwidth fix,* memanfaatkan *Simple Queue* untuk melakukan pengaturan *bandwidth share* dengan menerapkan *Limitasi* Bertingkat (Adellia, 2017)

## **2.3** **Pengalamatan IP dan Topologi**

*IP (Internet Protocol) Address* merupakan alamat yang diberikan kepada komputer-komputer yang terhubung dalam suatu jaringan. *IP Address* terdiri dari dua bagian, yaitu: *Network ID* dan *Host ID*. *Network ID* menentukan alamat dalam jaringan (*network address*) sedangkan *Host ID* menentukan alamat dari peralatan jaringan yang sifatnya unik untuk membedakan antara satu mesin dengan mesin yang lain. Dimisalkan sebuah rumah, *Network ID* seperti alamat rumah dan *Host ID* seperti nomor rumah.

Berdasarkan perkembangannya *IP Address* dibagi menjadi dua jenis:

*IPv4 (Internet Protocol versi 4)* merupakan *IP Address* yang terdiri dari 32 *bit* yang dibagi menjadi 4 segmen berukuran 8 *bit.*

*IPv6 (Internet Protocol versi 6*) merupakan *IP Address* yang terdiri dari 128 *bit* yang digunakan untuk mengatasi permintaan *IP Address* yang semakin meningkat, *IP Address* terdiri dari 32 *bit* angka *biner* yang dituliskan dalam bentuk empat kelompok masing-masing kelompok terdiri dari delapan *(oktaf) bit* dipisahkan oleh tanda titik.

contoh : 11000000.101010000.00000000.01100100

*IP Address* dapat juga ditulis dalam bentuk angka desimal dalam empat kelompok, dari angka 0-255.

contoh : 192.168.0.100 *IP Address* dibedakan menjadi 3 kelas, yaitu kelas A, kelas B, dan kelas C. Tujuan dibedakannya kelas *IP* adalah untuk menentukan jumlah komputer yang bisa terhubung dalam sebuah jaringan.

**Kelas A** Terdiri dari 8 *bit* pertama digunakan untuk *Network ID*, dan 24 *bit* berikutnya merupakan *Host ID.* *IP* kelas A terdapat 126 *Network*, yakni dari nomor 1.xxx.xxx.xxx sampai dengan 126.xxx.xxx.xxx (xxx merupakan variable yang nilainya dari 0 sampai dengan 225).

– *Format* *IP* kelas A

0NNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

– *Bit* pertama nilainya 0

– *Network ID* adalah 8 *bit* dan *Host ID* adalah 24 *bit*

– *Bit* pertama diisi antara 0 sampai dengan 127

– Range *IP* antara 1.xxx.xxx.xxx – 126.xxx.xxx.xxx

– Jumlah *Network* adalah 126 (untuk 0 dan 127 dicadangkan)

– Jumlah *Host* adalah 16.777.214

– Contoh *IP Address* 10.11.22.33 maka *Network ID* adalah 10 dan *Host ID* adalah 11.22.33

Jadi *IP* di atas mempunyai *host* dengan nomor 11.22.33 pada jaringan 10.

**Kelas B**

Terdiri dari 16 *bit* pertama digunakan untuk *Network ID*, dan 16 *bit* berikutnya merupakan *Host ID*. *IP* kelas B terdapat 16.384 *Network*, yakni dari nomor 128.0.xxx.xxx sampai dengan 191.255.xxx.xxx (xxx merupakan *variable* yang nilainya dari 0 sampai dengan 255).

– *Format IP* kelas B

10NNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHHH

(N = *Network ID*, H = *Host ID*)

– *Bit* pertama nilainya 10

– *Network ID* adalah 16 *bit* dan *Host ID* adalah 16 *bit*

– *Bit* pertama diisi antara 128 sampai dengan 191

– *Range IP* antara 128.0.xxx.xxx – 191.255.xxx.xxx

– Jumlah *Network* adalah 16.384 (64 x 256)

– Jumlah *Host* adalah 65.532

– Contoh *IP Address* 130.1.2.3 maka *Network ID* adalah 130.1 dan *Host* *ID* adalah 2.3

Jadi *IP* di atas mempunyai *Host* dengan nomor 2.3 pada jaringan 130.1

**Kelas C**

Terdiri dari 24 *bit* pertama digunakan untuk *Network ID*, dan 8 *bit* berikutnya merupakan *Host ID. IP* Kelas C terdapat 2.097.152 *Network*, yakni dari nomor 192.0.0xxx sampai dengan 223.255.255.xxx (xxx merupakan *variable* yang nilainya dari 0 sampai dengan 255).

– *Format IP* kelas C

110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

– *Bit* pertama nilainya 110

– *Network ID* adalah 24 *bit* dan *Host ID* adalah 8 *bit*

– *Bit* pertama diisi antara 192 sampai dengan 223

– *Range IP* antara 192.0.0.xxx – 233.255.255.xxx

– Jumlah *Network* adalah 2.097.152 (32 x 256 x 256)

– Jumlah *Host* adalah 254

– Contoh *IP Address* 192.168.0.100 maka *Network ID* adalah 192.168.0 dan *Host ID* adalah 100

Jadi *IP* di atas mempunyai *host* dengan nomor 100 pada jaringan 192.168.0

***IP Address Private*** *IP Address Private* merupakan alamat-alamat *IP* yang disediakan untuk digunakan pada jaringan local (*LAN*). *IP Address Private* digunakan untuk komunikasi pada jaringan yang tidak langsung dengan *internet.*

1. *IP Address Private* Kelas A memiliki *Range IP Address* antara 10.0.0.1 – 10.255.255.254
2. *IP Address Private* Kelas B memiliki *Range IP Address* antara 172.16.0.1 – 172.31.255.254
3. *IP Address Private* Kelas C memiliki *Range IP Address* antara 192.168.0.1 – 192.168.255.254

***IP Address Public***

*IP Address Public* merupakan alamat-alamat *IP* yang disediakan untuk digunakan pada jaringan *internet.*

*IP* jenis ini :

1. Harus daftarkan ke *ICANN* (*Internet Corporation for Assigned Names and Numbers*) atau *InterNIC*.

Ada biaya registrasinya. *IP Address Public* banyak digunakan pada *webserver, e-mail server* dan *gateway/proxy server* (Putra, 2016).

## **2.4 Wifi dan Hotspot**

*WiFi* adalah singkatan dari *Wireless Fidelity*. *WiFi* merupakan *teknologi* yang memanfaatkan peralatan *elektronik* yang gunanya untuk bertukar data *nirkabel* seperti menggunakan gelombang *radio* dan *wireless* melalui jaringan *computer* seperti koneksi *internet* dengan kecepatan tinggi. Awalnya *WiFi* lebih diperuntukkan untuk penggunaan perangkat *nirkabel* dan Jaringan Area Lokal (*Local Area Network / LAN*). Tetapi sekarang *WiFi* lebih digunakan untuk mengakses *internet.*

*Hotspot* merupakan tempat yang memiliki layanan *internet* dengan menggunakan teknologi *Wireless* *LAN* misalnya digunakan di Universitas/perguruan tinggi, mall, restaurant, perpustakaan, kantor. Dan sebenarnya untuk lebih jelasnya *hotspot* itu ketika lokasi *user* dimana atau ketika kita dapat mengakses *WiFi* melalui *mobile computer* misalnya seperti *laptop* tanpa menggunakan koneksi kabel untuk *internet* (Sridianti, 2019).

## **2.5 Topologi Jaringan**

*Topologi* yang mendefenisikan bagaimana jaringan *komputer, printer,* perangkat jaringan, dan perangkat lainnya dapat terhubung. Dengan kata lain, *topologi* jaringan menggambarkan *layout* kabel dan perangkat serta jalur yang digunakan oleh pengiriman data. *Topologi* sangat memperngaruhi bagaimana jaringan bekerja. Model atau *topologi* jaringan adalah bentuk dari jaringan yang dapat dibentuk, dimana bentuk *topologi* berpengaruh terhadap pemilihan jenis kabel(*media transmisi*) yang digunakan.

*Topologi* jaringan dapat dibagi menjadi 5 kategori utama yaitu *topologi bus*, *topologi star*, *topologi* *ring*, *topologi* *mesh*, dan *topologi* *tree*

a. Topologi *Bus* Model *BUS*, dimana *komputer* dan *server* dihubungkan pada sebuah kabel saja secara secara berderet. Ujung-ujung kabel data diberi komponen *elektronik* yang disebut *terminator*, yaitu *resistor* terbungkus logam dengan nilai tahanan sebesar 50 *ohm*.

b. Topologi *Star* Dalam model ini dapat dipergunakan alat tambahan yang disebut *hub* sebagai penghubungnya. *Hub* memiliki lubang *konektor* sejumlah tertentu, ada yang memiliki 8 buah lubang *koneksi* (disebut *port*), 12 *port* dan 24 *port*, kabel data dari masing-masing *komputer* atau *server* dihubungkan pada alat ini

c. Topologi *Ring* Dalam hubungan *komputer* model ini, kabel penghubung antar *komputer* dibuat seperti lingkaran (*ring*). Komputer yang dihubugkan secara berderet pada sebuah kabel data kemudian ujung satu dan ujung satunya dari kabel tersebut dihubungkan.

d. Topologi *Mesh*

Topologi ini merupakan rangkaian jaringan yang saling terhubung secara mutlak dimana setiap perangkat *komputer* terhubung secara langsung ke setiap titik perangkat lainnya. Setiap titik *komputer* akan mempunyai titik untuk berkomunikasi secara langsung dengan titik perangkat *komputer* lain yang menjadi tujuannya.

e. Topologi *Tree*

Topologi *Tree* merupakan salah satu topologi yang juga paling banyak diterapkan di dalam jaringan *komputer*, dengan bentuk *geometris* menyerupai pohon (*tree*). Pada topologi *Tree* terdapat sebuah *komputer* (atau perangkat jaringan *komputer* berupa *hub* ataupun *switch*) pada *level* teratas (*root*) yang menjadi pusat utama (Efendi, 2018).

# METODOLOGI PENELITIAN

Dalam penyusunan penelitian ini, penulis membutuhkan informasi dan referensi yang relatif lengkap untuk mendukung materi urutan pembahasan. Adapun informasi yang diperlukan adalah studi pustaka dan *Bandwidth* dari *ISP* menggunakan *MiFI.* Dokumen yang termasuk di dalamnya yaitu penelitian terdahulu, jurnal, dan buku yang berkaitan dengan objek penilitian.

## **3.1. Metode Analisis**

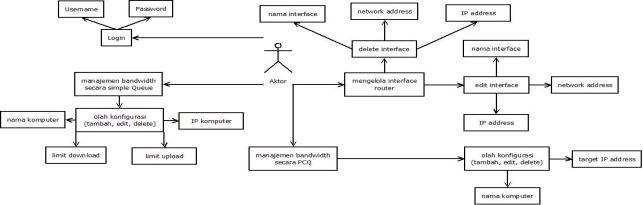
Metode analisis yang digunakan pada penelitian ini adalah metode *API*  (*Application Programming Interface*). dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari *aplikasi* atau dengan *aplikasi* yang berbeda secara bersamaan dan mengambil studi Kasus *Bandwidth* dari ISP MiFi M6 smartfren. *Quality of service* atau *QoS* digunakan untuk mengukur tingkat kualitas koneksi jaringan TCP/IP *internet*. Ada beberapa metode untuk mengukur kualitas koneksi seperti konsumsi *bandwidth* oleh *user*, ketersediaan koneksi, *latency*, dan *losses*.

Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangaat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem *operasi, library,* dan *web*.

## **3.2. Representasi Pengetahuan**

### 3.2.1. Desain Penelitian

Tahap yang berikutnya merupakan tahap desain. Dari hasil analisis kebutuhan fungsional seluruhnya kemudian dirancang dengan menggunakan *use case diagram.*

Gambar 2 merupakan *use case diagram* program bantu mikrotik. *Use case* *diagram* merupakan *diagram* yang dibuat berdasarkan sudut pandang *user.* Pada *use case diagram, user* seolah olah dilibatkan pada tahap *analisis* dan *desain* sistem. *Use case diagram* merupakan suatu konstruksi yang membantu *analisis* sistem dalam dalam menentukan bagaimana keadaan sistem. *Use case diagram* menggambarkan sistem dengan mengacu pada apa yang akan dilakukan *user* terhadap sistem

### 3.2.2 Internet

*Internet*(*interconnection networking*) adalah jaringan komunikasi global yang terbuka dan menghubungkan jutaan bahkan milyaran jaringan komputer dengan berbagai tipe dan jenis, dengan menggunakan tipe komunikasi seperti *telepon*, *satelit* dan lain sebagainya (Hindro, 2019).

### 3.2.3. Mikrotik Routerboard

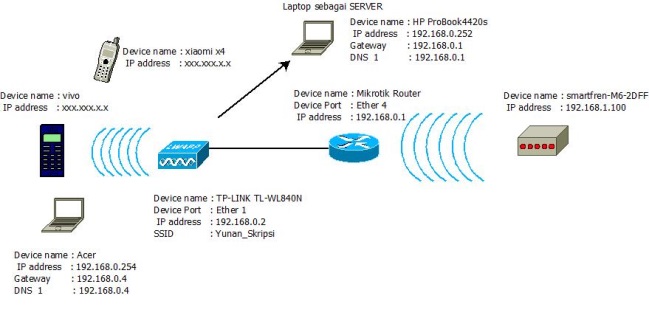
*Router* adalah sebuah alat yang mengirimkan paket data melalui sebuah jaringan atau *internet* menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing. Router* berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lain (Icki82, 2015)

### 3.2.4. *Download*

*Download* adalah proses transmisi sebuah *file* dari sebuah sistem komputer ke sistem komputer lainnya. Dari *internet*, *user* yang melakukan proses *download* adalah proses seorang *user* meminta/*request* sebuah *file* dari sebuah komputer lain(*website*, *server* atau yang lainnya) dan menerimanya (Prakoso, 2013)

# 4. PEMBAHASAN

*Topologi* jaringan dalam pembahasan ini dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Topologi Jaringan

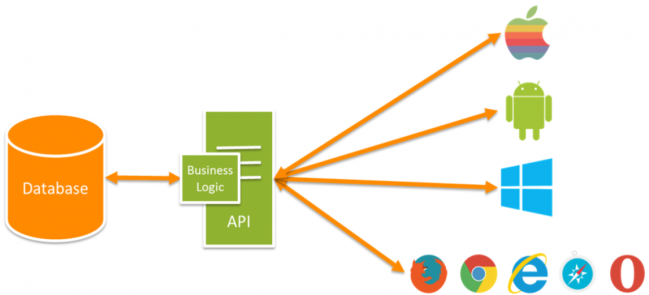
Pada pembahasan ini, untuk mendapatkan data yang akan dianalisis digunakan skenario percobaan sebagai berikut:

1. Untuk mengukur hasil pada penggunanan manajemen *bandwidth* menggunakan *simple tree* dilakukan dengan speedtest *bandwidth*.

2. Pengukuran dilakukan dengan membatasi *bandwidth Download* *limit max 1Mb*dan *bandiwdth* *Upload limit max 756Kb* .

## **4.1 API**

*API* adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari *aplikasi* atau dengan *aplikasi* yang berbeda secara bersamaan*. API* terdiri dari berbagai elemen seperti *function, protocols*, dan *tools* lainnya yang memungkinkan *developers* untuk membuat *aplikasi.* Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangaat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem *operasi, library, dan web.*



*API* yang bekerja pada tingkat sistem operasi membantu *aplikasi* berkomunikasi dengan *layer* dasar dan satu sama lain mengikuti serangkaian *protokol* dan *spesifikasi*. Contoh yang dapat menggambarkan *spesifikasi* tersebut adalah *POSIX* (*Portable Operating System Interface*). Dengan menggunakan standar *POSIX*, *aplikasi* yang di *compile* untuk bekerja pada sistem operasi tertentu juga dapat bekerja pada sistem lain yang memiliki kriteria yang sama. *Software library* juga memiliki peran penting dalam menciptakan *compatibility* antar sistem yang berbeda.

*Aplikasi* yang berinteraksi dengan *library* harus mengikuti serangkaian aturan yang ditentukan oleh *API*. Pendekatan ini memudahkan *software* *developer* untuk membuat *aplikasi* yang berkomunikasi dengan berbagai *library* tanpa harus memikirkan kembali strategi yang digunakan selama semua *library* mengikut *API* yang sama. Kelebihan dari metode ini menunjukkan betapa mudahnya menggunakan *library* yang sama dengan bahasa pemrograman yang berbeda.

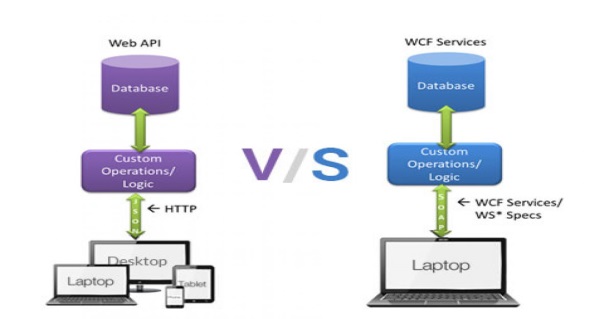
*Web API* dalam diakses melalui *protokol HTTP*, ini adalah konsep bukan *teknologi.* membuat *Web API* dengan menggunakan teknologi yang berbeda seperti *PHP, Java, .NET.*  Misalnya *Rest API* dari *Twitter* menyediakan akses *read* dan *write* data dengan mengintegrasikan *twitter* kedalam *aplikasi* kita sendiri.

*Web API* seperti sebuah alamat *web* (*end point*) yang dibuat untuk menangani beberapa *task* sesuai *request* yang diterima, juga terkadang memiliki parameter sebagai data yang dibutuhkan agar dapat menampilkan hasil yang diinginkan, juga pada beberapa kasus untuk mengakses *API* dibutuhkan kode *otentikasi* yang telah diizinkan untuk melihat data yang diinginkan. Semua rule ini ditentukan oleh *programmer* yang membuatnya.

**Perbedaan Web API dan Web Service**

Seringkali mendapat pertanyaan tentang apa perbedaan antara *web API* dan *web service*? Berikut adalah perbedaan diantara keduanya:

1. Semua *web service* menggunakan *API* tapi tidak semua *API* digunakan sebagai *web service.*
2. *Web service* memfasilitasi untuk melakukan interaksi antara dua perangkat atau *aplikasi* melalui jaringan. Sedangkan *API* bertindak sebagai penghubung antara dua *aplikasi* berbeda sehingga bisa berkomunikasi satu sama lain baik dengan ataupun tanpa jaringan.
3. *Web service* hanya menggunakan 3 jenis yaitu *SOAP, REST*, atau *XML-RPC* untuk berkomunikasi sedangkan *API* dapat menggunakan jenis apapun.
4. *Web service* selalu membutuhkan jaringan untuk pengoperasiannya sedangkan *API* tidak selalu memerlukan jaringan untuk operasinya.
5. *API* memfasilitasi interaksi langsung antara 2 *aplikasi* yang berbeda sedangkan *Web service* tidak, hanya menjembatani dalam sebuah jaringan.

Gambar 2. 2 Perbedaan API dan Web Sistem

Kesimpulan Metode pendekatan yang digunakan untuk berkomunikasi dengan *API* menggunakan *protokol* yang telah dijelaskan diatas masih tergolong *protokol* yang umum digunakan, masih ada banyak lagi yang belum tuliskan seiring perkembangan. Sedangkan bagaimana cara membuat *API* akan dibahas tergantung bahasa pemrograman apa yang digunakan (Sandi, 2017).

## **4.2 REST API**

*REST* (REpresentational State Transfer) merupakan standar arsitektur komunikasi berbasis *web* yang sering diterapkan dalam pengembangan layanan berbasis *web*. Umumnya menggunakan *HTTP (Hypertext Transfer Protocol)* sebagai *protocol* untuk komunikasi data. *REST* pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.

Pada arsitektur *REST*, *REST* server menyediakan *resources* (sumber daya/data) dan *REST client* mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global *ID*. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML (Feridi, 2019).

**a. Keuntungan REST**

1. Bahasa dan *platform agnostic.*

2. Lebih sederhana/simpel untuk dikembangkan ketimbang *SOAP.*

3. Mudah dipelajari, tidak bergantung pada *tools.*

4. Ringkas, tidak membutuhkan layer pertukaran pesan (*messaging)* tambahan

secara desain dan filosofi lebih dekat dengan *web.*

**b. Kelemahan REST**

1. Mengasumsi model point-to-point komunikasi - tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara.

2. Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, dll, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan ("dipecahkan sendiri").

3. Berkaitan dengan model transport HTTP.

Berikut metode HTTP yang umum digunakan dalam arsitektur berbasis *REST* :

1. *GET,* menyediakan hanya akses baca pada resource.

2. *PUT*, digunakan untuk menciptakan resource baru.

3.*DELETE*, digunakan untuk menghapus resource.

4.*POST*, digunakan untuk memperbarui resource yang ada atau membuat resource baru..

5.*OPTIONS*, digunakan untuk mendapatkan operasi yang disupport pada resource.

*Web service* adalah standar yang digunakan untuk melakukan pertukaran data antar *aplikasi* atau sistem, karena *aplikasi* yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada *platform* yang berbeda. Contoh *implementasi* dari *web service* antara lain adalah *SOAP* dan *REST.*

*Web service* yang berbasis arsitektur *REST* kemudian dikenal sebagai *RESTful* *web services*. Layanan *web* ini menggunakan metode *HTTP* untuk menerapkan konsep arsitektur *REST*.

*Client* mengirimkan sebuah data atau *request* melalui *HTTP Request* dan kemudian server merespon melalui *HTTP Response*.

**Komponen dari *http request* :**

1. *Verb*, *HTTP method* yang digunakan misalnya *GET, POST, DELETE, PUT* dll.

2. *Uniform Resource Identifier* (URI) untuk mengidentifikasikan lokasi *resource* pada *server*.

3. *HTTP Version*, menunjukkan versi dari *HTTP* yang digunakan, contoh *HTTP* v1.1.

4. *Request Header*, berisi metadata untuk *HTTP Request*. Contoh, tipe *client/browser*, format yang didukung oleh *client*, format dari *body* pesan, seting *cache.*

5. *Request Body*, konten dari data.

**Komponen dari *http response :***

1. Status/*Response Code*, mengindikasikan status *server* terhadap *resource* yang di *request*. misal : 404, artinya *resource* tidak ditemukan dan 200 *response* *OK*.

2. *HTTP Version*, menunjukkan versi dari *HTTP* yang digunakan.

3. *Response Header*, berisi metadata untuk *HTTP Response*. Contoh, tipe *server*, panjang *content*, tipe *content*, waktu *response*, dll

4. *Response Body*, konten dari data yang diberikan.

# Daftar Pustaka

Adellia, D. (2017). *SIMPLE QUEUE PADA MIKROTIK*.

Anwar, K. (2017). *Rancang Bangun Sistem Manajemen User Hotspot Menggunakan Mikrotik Php Api Berbasis Web Di Pondok Pesantren Al-Luqmaniyyah*, 5.

Asnawi, M. F. (2017). *Aplikasi Konfigurasi Mikrotik Sebagai Manajemen Bandwidth Dan Internet Gateway Berbasis Web*, 7.

Azizizzu. (2011). *Captive Portal*, 1.

CITRAWEB SOLUSI TEKNOLOGI, P. (2005). *Setting Dasar Hotspot Mikrotik*, 1.

CITRAWEB SOLUSI TEKNOLOGI, P. (2015). *Mendalami HTB pada QoS RouterOS Mikrotik*, 1.

Efendi, I. (2018). *Pengertian dan Macam macam Topologi Jaringan*, 1.

Googelio. (2018). *Macam Jenis Kecepatan Internet Yang Pelu Diketahui*, 1.

Hermawan. (2019). *Pengertian Server, Fungsi Server Beserta Cara Kerja dan Jenis-jenis Server*, 1.

Hindro. (2019). *Pengertian Internet*, 1.

Icki82. (2015). *Pengertian dan fungsi mikrotik routerboard*, 1.

K, Y. (2018). *Pengertian Bandwidth dan Fungsinya*, 1.

Khermawan203. (2018). *Pengertian, Fungsi dan cara kerja dan perbedaan DHCP Server dan DHCP Client*, 1.

Kosasi, S. (2013). *TEKNIK MARK PACKET DAN MARK CONNECTION*, 10.

Mardiana, G. (2015). *Sistem Pemesanan Menu Berbasis Web Memanfaatkan Mikrotik API*, 50.

N, S. (2014). *Pengertian Download Dan Upload Secara Lebih Lengkap*, 1.

Prakoso, P. S. (2013). *Pengertian Upload Dan Download*, 1.

Putra, A. M. (2016). *Jenis-jenis IP Address*, 1.

Sandi, A. (2017). *Mengenal Apa itu Web API*, 1.

Saputra, R. (2012). *Pengertian IP Address Dan Fungsinya*, 1.

Sridianti. (2019). *Perbedaan Wifi dan Hotspot*, 1.

Suharjo, I. (2017). jurnal agrisains. *LOG ANALYSIS IN THE USER ACCESS ON THE WEB SERVICES SERVER FOR SECURITY SISTEM EVALUATION*, 1.

Zulhijah, S. (2009). *pengertian DHCP*, 1.