

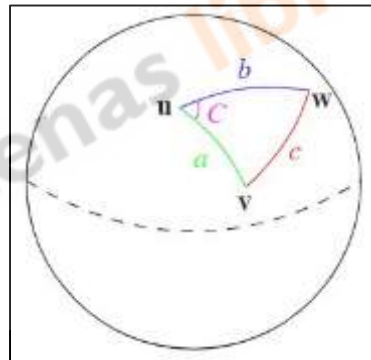
## BAB II

### LANDASAN TEORI

Pada BAB II ini dijelaskan mengenai landasan teori yang berkaitan dengan penelitian *Pencarian Agen Travel Menggunakan Metode Haversine di Kota Bandung*.

#### 2.1. Theorema Haversine

Metode Haversine Formula adalah sebuah persamaan yang penting dalam bidang navigasi, dan dapat digunakan untuk menghitung jarak antara dua titik, berdasarkan posisi garis lintang (*latitude*) dan posisi garis bujur (*longitude*) sebagai variabel masukan (Adiwilaga, 2014). *Haversine Formula* memberikan jarak lingkaran besar antara dua titik pada permukaan bumi berdasarkan garis bujur dan garis lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari - jari  $R$  6.371 km



Gambar 2.1 Ilustrasi Hukum Kosinus Bulat  
(Sumber : Institut Teknologi Bandung)

Pertama ditentukan terlebih dahulu titik awal dan titik tujuan, titik awal berupa *latitude1* ( $lat_1$ ) dan *longitude1* ( $long_1$ ), titik tujuan berupa *latitude2* ( $lat_2$ ) dan *longitude2* ( $long_2$ ). Titik awal dan titik tujuan tersebut berbentuk desimal derajat yang dilakukan perubahan menjadi nilai sudut radian, kemudian melakukan perhitungan dengan rumus *Haversine Formula* (Adiwilaga, 2014). Contoh perhitungan dengan menggunakan rumus *Theorema Haversine* sebagai berikut:

$$\begin{aligned}
&= (6371,1 * ((2 * \text{ASIN}(\text{SQRT}((\text{SIN}((\text{RADIANS}(-6.883675) - \text{RADIANS}(-6.894896)) / 2)^2) + \text{COS}(\text{RADIANS}(-6.883675)) * \text{COS}(\text{RADIANS}(-6.894896))) * (\text{SIN}((\text{RADIANS}(107.595555) - \text{RADIANS}(107.603720)) / 2)^2)))))) * 1000 \\
&= 1539,256128 \text{ m}
\end{aligned}$$

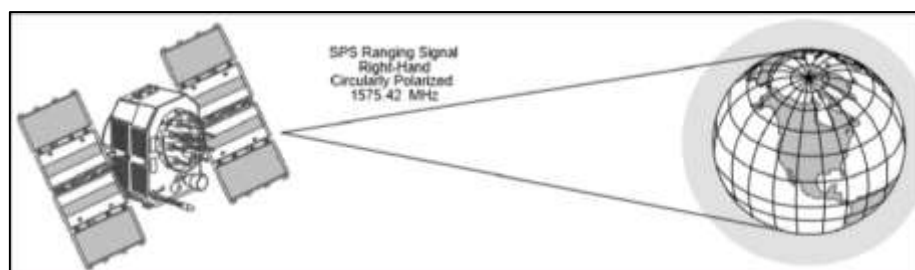
*Theorema Haversine formula* digunakan sebagai metode yang dapat membantu untuk mengetahui pengukuran jarak antara dua titik dengan mengasumsikan bahwa bumi bulat.

## 2.2. Global Positioning System (GPS)

*Global Positioning System (GPS)* merupakan sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit dan metode *Triangulasi*. *GPS* membutuhkan 24 satelit buatan untuk dapat memantau seluruh titik yang ada pada permukaan bumi, sedangkan untuk dapat menentukan koordinat objek yang ingin diamati *GPS* harus terhubung minimal dengan 3 buah satelit (Purnawan, 2018). Hal tersebut berkaitan dengan konsep perhitungan metode *triangulasi* pada *GPS*. Satelit yang terhubung mengirimkan sinyal berupa nilai koordinat (*latitude dan longitude*) yang diterima oleh *GPS receiver*.

Berikut adalah tahapan pembacaan oleh sensor GPS pada *smartphone*:

1. *GPS receiver* menerima transmisi sinyal satelit.
2. Membaca data besaran koordinat lintang dan bujur (*latitude dan longitude*) berdasarkan sinyal satelit yang diterima.



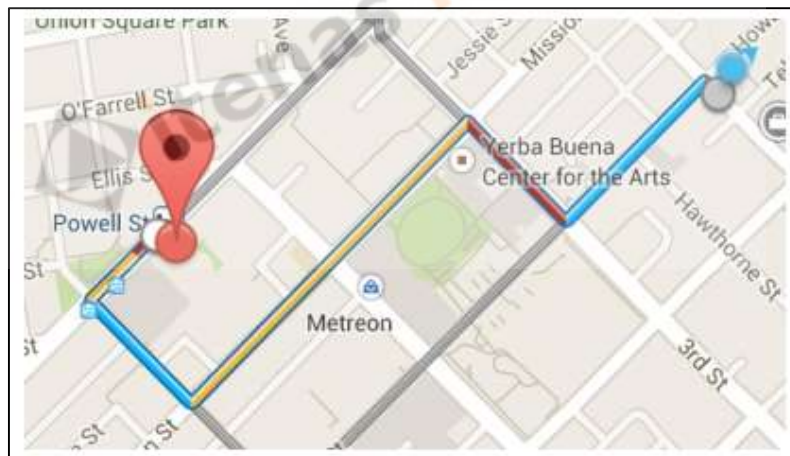
Gambar 2.2 Jarak Sinyal SPS dan Transmisi  
(Sumber : Navstar 1998. Triangulasi sinyal GPS)

*GPS* berfungsi untuk menentukan letak di permukaan bumi dengan bantuan sinkronisasi sinyal satelit yang mengirimkan sinyal gelombang mikro ke Bumi, sehingga dapat digunakan untuk menentukan kecepatan, arah, letak, dan waktu,

juga dapat menentukan lokasi posisi koordinat objek yang diamati untuk diketahui keakurasiannya dari perpindahan lokasi posisi koordinat setiap detiknya (Susilo,2014). Salah satu kelemahan dari GPS pada penelitian ini yaitu adanya faktor kesalahan dalam membaca koordinat posisi, yang tentunya berpengaruh terhadap keakurasiannya

### 2.3. Google Maps API

*Google Maps API* adalah layanan gratis yang diberikan oleh Google dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat digunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Dapat dilakukan penambahan fitur *Google Maps* dalam web yang telah dibuat atau pada blog yang berbayar maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu library yang berbentuk JavaScript (Winardi, 2014). Tampilan *google maps* dapat dilihat pada Gambar 4.



Gambar 2.3 Google Maps API

(Sumber: PT. Susanto Bersaudara Intermedia Pers)

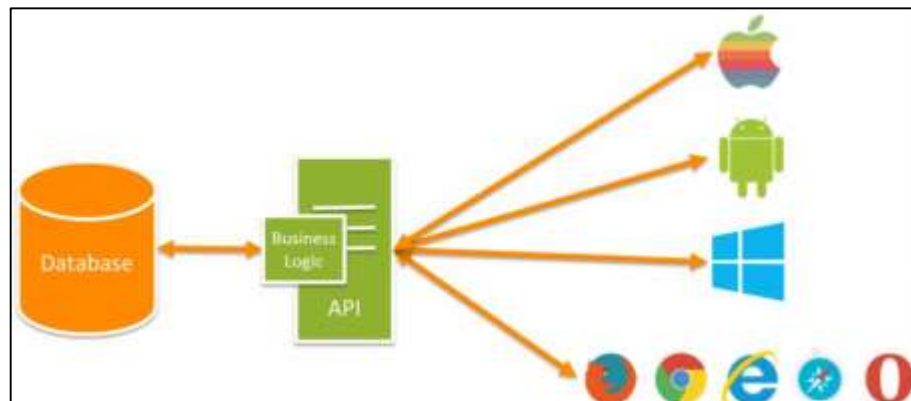
Cara membuat *Google Maps* untuk ditampilkan pada suatu web atau android sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi internet yang sangat stabil. Dengan menggunakan *Google Maps API*, dengan ini dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga dapat fokus hanya pada data yang

ditampilkan. Dengan kata lain, yang dilakukan hanya membuat suatu data sedangkan peta yang ditampilkan adalah milik *Google* (Winardi, 2014). Dalam pembuatan program *Google Maps API* menggunakan urutan sebagai berikut:

1. *API Key* dibuat pada *Google Console*
2. *Google Maps API JavaScript* yang sudah di tambahkan *API Key* dimasukkan kedalam *Android Studio*.
4. Fungsi *JavaScript* ditulis untuk membuat objek peta digital.
5. Inisiasi peta dalam *Activity* dengan *on create*.

## 2.4 Web Service

Layanan Web atau API Web adalah antarmuka yang dapat diakses mesin untuk data atau tindakan yang ditawarkan oleh server melalui HTTP. "Layanan Web" digunakan untuk antarmuka XML berbasis-RPC, sedangkan "*Web API*" digunakan untuk hal lebih ringan, seperti hanya untuk menampilkan data dengan format *JSON* atau *XML* (Verborgh, 2017). Tujuan penggunaan *API* adalah untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. Penerapan *API* sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Seperti yang ditunjukkan pada Gambar 5 bahwa web *API* mengakses *databases*, dan kemudian perangkat browser, android, dan windows mengakses database lewat *API* sehingga tidak dibutuhkan lagi pembuatan query yang sama pada setiap platform karena semua query untuk mengakses database telah disimpan pada web *API*.



Gambar 2.4 Web API

(Sumber : PT. Codepolitan Integrasi Indonesia)

Seperti namanya, Web *API* diakses melalui protokol HTTP, ini adalah konsep bukan teknologi. Kita bisa membuat Web *API* dengan menggunakan teknologi yang berbeda seperti *PHP*, *Java*, *.NET*, dll. Web *API* memiliki beberapa fitur, diantaranya adalah:

1. Mendukung fungsi CRUD yang bekerja melalui *HTTP* protocol dengan method GET, POST, PUT dan DELETE
2. Memiliki *response Accept Header* dan *HTTP status code*
3. *Response* dengan format *JSON*, *XML* atau format apapun yang kamu inginkan. Akan tetapi kebanyakan digunakan kedalam format *JSON*.
4. Mendukung fitur MVC seperti *routing*, *controllers*, *action results*, *filter*, *model*, *IOC container*, dll.
5. Web *API* dapat berjalan di *Apache* atau *web server* lainnya yang didukung sesuai bahasa pemrograman yang digunakan (Anugrah, 2019). 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