

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang telah ada sebelumnya dan digunakan sebagai rujukan dalam penelitian ini adalah sebagai berikut :

Rohman (2013) membuat “Sistem Informasi Geografis Obyek Wisata Pantai di Kabupaten Gunung Kidul Menggunakan Google Maps API”. Pada sistem ini memberikan informasi lengkap tentang obyek wisata pantai di Kabupaten Gunung Kidul beserta sarana penunjang dan dapat menampilkan rute alternatif, dengan menggunakan HTML dan CSS untuk membuat tampilan web. Namun pada sistem ini belum terdapat fungsi pencarian terhadap obyek wisata yang terkait.

Informasi berikutnya diberikan oleh Herawati (2014) membuat “ Sistem Informasi Geografis Terminal Bus Dalam Kota Di DKI Jakarta”. Sistem ini menggunakan *Macromedia Dreamweaver* sebagai *Design editor* juga bahasa pemrograman PHP dan *JavaScript* untuk mendesain tampilan web dengan mengambil peta dari *server* Google Maps, sistem ini memberikan informasi terminal bus yang ada di DKI Jakarta, dan rute angkutan umum ke terminal, tetapi sistem ini tidak melakukan perhitung jarak menuju terminal.

Selanjutnya Informasi sejenis juga diberikan oleh Nuzulianto (2015) membuat “Sistem Informasi Geografis (SIG) Pada Lokasi Wisata Kabupaten Bantul Berbasis Web Untuk Wisatawan”. Pada sistem ini menginformasikan rute dan

lokasi pariwisata yang ada di kabupaten Bantul, dengan menggunakan *HTML* dan *CSS* untuk mendesain tampilan Web. Sistem ini tanpa dilengkapi dengan perhitungan jarak dan rute untuk menuju lokasi.

Informasi serupa diberikan pula oleh Abraim (2016) membuat “Sistem Informasi Geografis Wisata Kuliner Di Yogyakarta Berbasis Web” pada sistem ini menginformasikan lokasi-lokasi wisata kuliner yang ada di kota Yogyakarta, dengan menggunakan Google maps dengan menggunakan bahasa pemrograman PHP, sistem ini dilengkapi dengan grafik perkembangan pengunjung yang melihat web kuliner, tapi tidak dilengkapi dengan perhitungan jarak dari pengguna ke lokasi wisata kuliner.

Dan juga Informasi yang diberikan oleh Pramono (2016), membuat “Sistem Informasi Geografis Pencarian Apotik Terdekat Di Kota Yogyakarta” pada sistem ini menginformasi lokasi apotik-apotik yang ada di kota Yogyakarta kepada pengguna dengan memberikan informasi lokasi terdekat dari titik pengguna berada, dengan menggunakan metode *Spherical Law of Cosines* dan Google Maps untuk pemetaan dan *HTML5 Geolocation* untuk mendesain web, namun pada sistem tidak membuat perbedaan tipe pelayanan apotik antar K24 dengan apotik swasta.

Sedangkan pada judul penelitian ini dilakukan pemetaan dengan menggunakan *Google Maps API*. Peta yang ditampilkan dari *server Google Maps*. Pada penelitian ini tidak hanya menampilkan peta dan informasi terminal yang ada di DIY, tapi dapat mengetahui jarak terdekat lokasi terminal dari posisi pengguna berada dengan menggunakan *HTML5 Geolocation* juga metode

Spherical Law of Cosines dan *Framework Codeigniter* untuk mendesain web. Selain itu terdapat fungsi pencarian terhadap terminal, juga sistem ini dapat menampilkan koordinat posisi pengguna.

Tabel 2.1 Perbandingan Penelitian Sebelumnya.

| Penulis | Judul | Domain | Teknologi | Hasil |
|-------------------|--|----------------|--|---|
| Rohman (2013) | Sistem Informasi Geografis Obyek Wisata Pantai di Kabupaten Gunung Kidul Menggunakan Google Maps API | Pariwisata | HTML, CSS, dan Google Maps | <ul style="list-style-type: none"> - Sitem Informasi wisata pantai - Informasi rute tanpa dilengkapi total jarak - Tidak ada form pencarian |
| Herawati (2014) | Sistem Informasi Geografis Terminal Bus Dalam Kota Di DKI Jakarta | Terminal | <i>Macromedia Dreamweaver</i> , PHP, <i>JavaScript</i> , dan Google Maps | <ul style="list-style-type: none"> - Sistem Informasi Geografis Terminal Bus di DKI Jakarta - Menampilkan informasi terminal dan rute angkuta umum ke terminal. |
| Nuzulianto (2015) | Sistem Informasi Geografis Pada Lokasi Wisata Kabupaten Bantul Berbasis Web Untuk Wisatawan | Pariwisata | HTML, CSS, dan Google Maps | <ul style="list-style-type: none"> - Sistem Informasi Pariwisata - Tidak ada form pencarian - Tidak menampilkan jarak dan rute |
| Arbain (2016) | Sistem Informasi Geografis Wisata Kuliner Di Yogyakarta Berbasis Web | Wisata Kuliner | HTML, CSS, dan Google Maps | <ul style="list-style-type: none"> - Sistem Informasi lokasi wisata kuliner di kota Yogyakarta -Tidak dilengkapi dengan perhitungan jarak dari pengguna ke lokasi wisata kuliner. |

Lanjutan Tabel 2.1 Perbandingan Penelitian Sebelumnya.

| Penulis | Judul | Domain | Teknologi | Hasil |
|----------------|---|----------|---|--|
| Pramono (2016) | Sistem Informasi Geografis Pencarian Apotik Terdekat Di Kota Yogyakarta | Apotik | <i>Spherical Law of Cosines, HTML5 Geolocation</i> dan Google Maps | <ul style="list-style-type: none"> - Sistem Informasi Pencarian Apotik Terdekat - Tidak membuat perbedaan tipe apotik antar K24 dengan apotik swasta. |
| Maubanu (2017) | Sistem Informasi Geografis Berbasis Web Menggunakan <i>HTML5 Geolocation</i> Untuk Pencarian Terminal Terdekat Di DIY | Terminal | <i>Framework Codeigniter, HTML5 Geolocation, Spherical Law of Cosines, PHP, Javascript</i> dan <i>Google Maps API</i> | <ul style="list-style-type: none"> - Sistem Informasi Pencarian terminal terdekat. - Sistem dapat mengidentifikasi dan menampilkan lokasi pengguna - Terdapat fungsi pencarian terminal berdasarkan radius dan nama terminal - Menampilkan informasi terminal dan rute dari lokasi pengguna. |

2.2 Dasar Teori

2.2.1 Terminal

UU RI No. 14 tahun 1992 pasal 9 dan 10 menjelaskan bahwa terminal merupakan penunjang untuk kelancaran mobilitas orang maupun arus barang dan untuk terlaksananya keterpaduan yang saling berhubung dengan lancar dan tertib, di tempat-tempat tertentu dapat di bangun dan di selenggarakan terminal. Terminal juga diartikan sebagai:

1. Titik simpul dari jaringan transportasi jalan yang berfungsi sebagai pelayanan umum.
2. Tempat pengendalian, pengawasan, pengaturan, dan pengoperasian lalu lintas.
3. Prasarana angkutan yang merupakan bagian dari sistem transportasi untuk melancarkan arus angkutan dan barang.
4. Unsur tata ruang yang mempunyai peranan penting bagi efisiensi kehidupan kota.

Jumlah terminal yang tercatat pada Dinas Perhubungan (Dishub) DIY sebanyak empat belas terminal dengan digolongkan menjadi dua tipe pelayanan, yaitu tipe A dan tipe B. Tipe A merupakan terminal yang melayani Bus umum Antar Kota Antar Provinsi (AKAP) yaitu Terminal Giwangan Yogyakarta, Terminal Dhaksinarga Wonosari, Terminal Jombor, Terminal Palbapang, dan Terminal Wates. Sedangkan terminal tipe B merupakan terminal yang melayani Bus Angkutan Pedesaan, yaitu Terminal Condongcatur, Terminal Prambanan, Terminal Parangtritis Baru, Terminal Gabusan, Terminal Imogiri, Terminal Sedayu, Terminal Panggang, Terminal Piyungan, dan Terminal Brosot.

2.2.2 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) adalah sebuah sistem atau teknologi berbasis komputer yang dibangun dengan tujuan untuk mengumpulkan, menyimpan, mengolah, dan menganalisa, serta menyajikan data dan informasi

dari suatu obyek fenomena yang berkaitan dengan letak atau keberadaannya di permukaan bumi (Ekadinata, 2008).

2.2.3 *Latitude dan Longitude*

Latitude disebut juga garis lintang. Garis lintang merupakan garis vertikal yang mengatur sudut antara suatu titik dengan garis katulistiwa. Titik di utara garis katulistiwa dinamakan lintang utara, sedangkan titik di selatan katulistiwa dinamakan lintang selatan. *Longitude* di sebut juga garis bujur. Garis bujur yaitu garis horizontal yang mengatur sudut antara suatu titik dengan titik nol bumi, yaitu *Greenwich* di London, Britania Raya yang merupakan titik 0^0 atau 360^0 yang diterima secara internasional. Titik di barat 0^0 dinamakan bujur barat, sedangkan titik di timur 0^0 dinamakan bujur timur (Sirenden dan Dachi, 2012).

2.2.4 *Google Maps Javascript API*

Google Maps merupakan sebuah layanan dari Google yang sangat bagus dalam hal pemetaan, dimana keunggulan dari Google Maps ini adalah teknologi pemetaan yang *user-friendl* dalam tampilan yang memberikan informasi bisnis, termasuk dalam lokasi kontak dan arah perjalanan.

Google Maps dapat dimanfaatkan untuk berbagai keperluan seperti digunakan dalam pengembangan sistem informasi geografis, penambahan lokasi untuk bisnis, untuk dunia pendidikan atau dalam berbagai bidang. Hal ini dikarenakan layanan bersifat gratis.

Google Maps API adalah *javascript* . Dengan menggunakan *Google Maps API* hanya berkonsentrasi tentang data dan biarkan urusan peta ditangani oleh Google Maps, sehingga dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal.

Google Maps API merupakan kumpulan layanan yang memungkinkan untuk memasukan peta, *geocoding* tempat. Dimana layanan ini memberikan akses untuk memanfaatkan peta Google dalam sebuah aplikasi (Fatoni, 2012).

2.2.5 HTML 5 Geolocation

Geolocation HTML5 membantu dalam mengidentifikasi lokasi pengguna, yang dapat digunakan untuk memberikan informasi spesifik lokasi atau rincian navigasi posisi pengguna. *Geolocation* melacak keberadaan perangkat yang digunakan dan disimpulkan dengan memanfaatkan RFID, WiFi dan Bluetooth MAC alamat, dan GPS/CDMA ID sel, serta input pengguna, Namun *HTML5 Geolocation* ini bisa diajak kompromi, artinya selama pengguna tidak menyetujui, posisi tidak tersedia hal ini untuk menjaga *privacy* dari pengunjung. (Widyatmoko, 2012)

Browser yang *support HTML5 Geolocation* dapat berjalan adalah Chrome: > 5.0 ,Internet Explorer: > 9.0, Firefox: > 3.5 , Safari: > 5.0, Opera: > 16.0. *Script* dibawah ini untuk membantu mendeteksi dukungan browser *API Geolocation*:

```
if (navigator.geolocation) {
    // Dapatkan posisi pengguna saat ini
} else {
    alert('Geolokasi tidak didukung di browser Anda');
}
```

Lokasi pengguna saat ini dapat diperoleh dengan menggunakan fungsi *getCurrentPosition* dari objek *navigator.geolocation*. Fungsi ini menerima tiga parameter yaitu *Success callback function*, *Error callback function* dan *position options*. Jika data lokasi berhasil diambil, fungsi *callback* sukses akan dipanggil dengan objek posisi yang diperoleh sebagai parameter masukannya. Jika tidak, fungsi *callback error* akan dipanggil dengan objek kesalahan sebagai parameter inputnya. Dibawah ini merupakan contoh scriptnya:

```
if (navigator.geolocation) {

// Dapatkan posisi pengguna saat ini
navigator.geolocation.getCurrentPosition(showPosition,
showError, optn);
} else {
    alert('Geolocation is not supported in your
browser');
}
```

Fungsi Callback Sukses apabila lokasi berhasil di *share*:

```
function showPosition(position) {
document.write('Latitude:
'+position.coords.latitude+'Longitude:
'+position.coords.longitude); }
```

Fungsi Callback Error apabila lokasi gagal di *share*:

```
function showError(error) {
    switch(error.code) {
        case error.PERMISSION_DENIED:
            alert("User denied the request for
Geolocation.");
            break;
        case error.POSITION_UNAVAILABLE:
```

```

        alert("Location information is unavailable.");
        break;
    case error.TIMEOUT:
        alert("The request to get user location timed
        out.");
        break;
    case error.UNKNOWN_ERROR:
        alert("An unknown error occurred.");
        break;
}

```

2.2.6 Spherical Law of Cosines

Perhitungan komputer dapat memberikan tingkat presisi yang sangat akurat sehingga dengan menggunakan rumus *Spherical Law Of Cosines* sederhana, dapat menentukan posisi dengan cukup akurat.

$$d = \text{acos}(\text{cos}(\text{lat1}) \cdot \text{cos}(\text{lat2}) \cdot \text{cos}(\text{long2} - \text{long1}) + \text{sin}(\text{lat2})).R$$

Gambar. 2.1 Rumus *Spherical Law of Cosines*

Keterangan:

R = Jari-jari bumi sebesar 6.371 (KM)

d = Jarak (KM)

lat1 = *Latitude* titik awal posisi pengguna berada

long1 = *Longitude* titik awal posisi pengguna berada

lat2 = *Latitude* titik akhir posisi terminal berada

long2 = *Longitude* titik akhir posisi terminal berada

Contoh Perhitungan *Spherical Law of Cosines*:

Menghitung jarak antara koordinat A (lat1 = -7.8038678 °, lon1 = 110.3806635°) dan koordinat B (lat2 = -7.804541°, lon2 = 110.36303°).

Penyelesaian :

1. Lat1 = -7.8038678 ° = -0,1362031875 rad
lon1 = 110.3806635° = 1.9265060086 rad
lat2 = -7.804541° = -0.13621493706 rad
lon2 = 110.36303° = 1.9261982456 rad
2. $d = \text{ACOS}(\text{SIN}(-0.1362031875) \times \text{SIN}(-0.13621493706) + \text{COS}(-0.1362031875) \times \text{COS}(-0.13621493706) \times \text{COS}(1.926198246 - 1.9265060086)) \times 6371 = 1,97348096$
3. Jadi jarak antara koordinat A dan koordinat B jika ditarik garis lurus adalah 1,9 Km.

Penerapan rumus *Spherical Law of Cosines* pada penelitian ini akan digunakan pada aplikasi *server* yaitu untuk menghitung jarak antara lokasi pengguna dengan lokasi terminal.

2.2.7 *HyperText PreProcessor (PHP)*

HyperText PreProcessor (PHP) adalah bahasa *script* yang dapat ditanamkan atau disisipkan kedalam HTML, yang banyak dipakai untuk membuat situs web dinamis. PHP dapat juga digunakan untuk membangun sebuah CSS. Sebagian besar *sintaks* mirip dengan bahasa C, Java, dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk

memungkinkan perancang dalam menulis halaman web menjadi dinamis dan cepat (Badiyanto, 2013).

2.2.8 Framework Codeigniter

Framework Codeigniter merupakan aplikasi open source yang berupa *Framework* dengan model MVC untuk membangun website dinamis menggunakan PHP. *Codeigniter* memudahkan developer website untuk membuat aplikasi website dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web. Berawal dari bahasa pemrograman *Small Talk*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun suatu aplikasi seperti manipulasi data, *user interface* dan bagian yang menjadi kontrol aplikasi. Terhadap 3 jenis komponen yang membangun suatu MVC *pattern* dalam aplikasi sebagai berikut.

- a. *View*, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML yang diatur oleh *controller*. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
- b. Model, biasanya berhubungan langsung dengan databases untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari

bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

- c. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dengan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi (Westriningsih, 2011)

2.2.9 My Structured Query Language (MySQL)

Database bisa dikatakan sebagai suatu kumpulan dari data yang tersimpan dalam tabel dan diatur atau diorganisasikan sehingga data tersebut bisa diambil atau dicari dengan mudah dan efisien. *Database* MySQL merupakan sebuah database server SQL *multiuser* dan *multithreaded*. MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris (kira-kira 100 *gigabyte* data). database ini dibuat untuk keperluan sistem *database* yang cepat, handal, dan mudah digunakan. Walaupun memiliki kemampuan yang cukup baik, MySQL untuk sistem operasi *Unix* Bersifat *Freeware*, dan terdapat versi *Shareware* untuk sistem operasi windows (Badiyanto, 2013).

2.2.10 Unified Modeling Language (UML)

Unified Modeling Language (UML) Merupakan sistem arsitektur yang bekerja dalam OOAD dengan satu bahasa yang konsten yang menentukan,

visualisasi, mengkonstruksi, dan mendokumentasikan artifact yang terdapat dalam sistem software.

Tujuan utama UML diantaranya untuk:

- a. Memberikan Model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- b. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- c. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan (Suhendar, 2002).