**BAB II**

**LANDASAN TEORI**

Untuk memudahkan dalam pemahaman dalam penyusunan laporan skripsi, peneliti mengadakan studi pustaka mengenai arti dan istilah yang digunakan dalam penelitian sehingga memudahkan dalam pemecahan suatu masalah yang akan ditemui dalam perancangan sistem aplikasi pencarian rute terpendek.

**2.1 *Optical Distribution Cabinet* (ODC)**

ODC merupakan suatu perangkat pasif yang diinstal di luar STO, bisa di luar ruangan (outdoor), dan bisa juga di dalam ruangan (*indoor*). ODC ini mempunyai fungsi yaitu sebagai berikut :

1. Sebagai titik transmisi ujung kabel feeder dan pangkal kabel distribusi.
2. Sebagai titik distribusi kabel kapasitas besar (feeder) menjadi kabel yang kapasitasnya lebih kecil (distribusi).
3. Tempat splitter.
4. Tempat penyambungan kabel optik.

Kapasitas ODC ada bermacam-macam sesuai dengan kebutuhan, yaitu ODC berkapasitas 86, 144, 288, 576 port (Arham, 2018).

**2.2 Sistem Informasi Geografis**

Pemetaan dan Geografi adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang berkaitan erat dengan sistem pemetaan dan analisis terhadap segala sesuatu serta peristiwa-peristiwa yang terjadi di muka bumi. Teknologi GIS mengintegrasikan operasi pengolahan data bebasis database yang biasa digunakan saat ini, seperti pengambilan data berdasarkan kebutuhan, serta analisis statistik dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya. Web-GIS merupakan Sistem Informasi Geografi berbasis web yang terdiri dari beberapa komponen yang saling terkait. WebGIS merupakan gabungan antara design grafis pemetaan, peta digital dengan analisa geografis, pemrograman komputer, dan sebuah database yang saling terhubung menjadi satu bagian web design dan web pemetaan (Mushallian, 2017).

**2.3 Peta**

Peta merupakan suatu representasi konvensional (miniatur) dari unsur-unsur fisik alamiah dan buatan manusia dari sebagian atau keseluruhan permukaan bumi di atas media bidang datar dengan skala tertentu. Peta adalah gambaran permukaan bumi dengan skala tertentu, digambar pada bidang datar melalui sistem proyeksi tertentu (Prihandito, 1989).

**2.4 Google Maps API**

Google Maps API menyediakan sejumlah utilitas untuk memanipulasi peta dan menambahkan konten didalam peta melalui berbagai layanan, mengijinkan pembuatan aplikasi map yang kuat pada situs web (Febiyan et al., 2010).

Cara membuat API\_KEY Google Maps :

* + 1. Masuk ke Google Cloud Console dan login menggunakan akun Gmail atau akun Google.
    2. Membuat Akun *Billing* untuk pembayaran dengan melampirkan Kartu Kredit atau PayPal.
    3. Buat project agar bisa mengaktifkan API Google Map dengan masuk ke halaman google console new project.
    4. Klik Select dan pilih project yang baru saja dibuat.
    5. Lihat dan arahkan kursor ke API Manager pada menu sebelah kiri, dan klik Dashboard.
    6. Jika baru pertama membuat membuka Google Cloud Console maka harus mengaktifkan API terlebih dahulu dengan mengklik tombol Enable API.
    7. Selanjutnya akan banyak sekali API yang bisa gunakan untuk pengembangan berbagai aplikasi, lalu pilih salah satu.
    8. Pada halaman selanjutnya klik tombol Enable API.
    9. Selanjutnya klik tombol Create Credentials.
    10. Sekarang pilih API yang akan gunakan, lalu klik tombol biru dibawahnya.

Sekarang klik Done setelah itu sudah mendapat API\_KEY pada kolom Key.

**2.5 Formula *Haversine***

Jalur terpendek antara dua titik adalah sepanjang busur lingkaran besar itu adalah lingkaran yang digambar di permukaan bola yang berpusat pada titik yang sama dengan bola dan memiliki jari-jari yang sama. Dua titik terletak pada lingkaran besar yang unik, yang dibagi menjadi dua busur. Jalur terpendek antara titik adalah sepanjang yang lebih pendek dari dua busur ini (Marianne, 2014).

Jadi rumus haversine ialah persamaan yang penting dalam navigasi, yang menghasilkan jarak lingkaran besar antara dua titik (latitude dan longitude) pada permukaan bola (bumi) berdasarkan bujur dan lintang. Penggunaan rumus ini cukup akurat untuk sebagian besar 17 perhitungan, juga mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi.

Rumus haversine sebagai berikut :

D = acos(sin(lat1) . sin(lat2) + cos(lat1) . cos(lat2) . cos(long2-long1)) \* R......(2.1)

“sudut diukur dalam radian”

Keterangan:

D = jarak (km)

R = jari-jari bumi sebesar 6371 (km)

Lat1 = titik koordinat lintang awal

Lat2 = titik koordinat lintang tujuan

Long1 = titik koordinat bujur awal

Long2 = titik koordinat bujur tujuan

**2.6 XAMPP**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Palit, Rindengan and Lumenta, 2015)

Berikut adalah kelebihan dan kekurangan XAMPP :

Kelebihan :

1. Database Storage Engine ini banyak digunakan oleh programmer apalagi oleh web developer karena sifatnya yang free. Untuk yang expert sudah ada yang bayar.
2. Kemampuannya sudah bisa diandalkan, mempunyai kapasitas yang cukup mumpuni sekitar 60.000 tabel dengan jumlah record mencapai 5.000.000.000 bahkan untuk yang terbaru sudah lebih.
3. Engine ini multiplatform sehingga mampu diaplikasikan di berbagai sistem operasi. MySQL cocok diaplikasikan diaplikasi kelas kecil dan menengah.
4. Kelebihan paling utama engine ini adalah kecepatannya.

Kekurangan :

1. Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
2. Memiliki keterbatasan kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung server karena tidak menerapkan konsep Technology Cluster

**2.7 *Unified Modelling Language* (UML)**

Unified Modelling Language (UML) adalah sebuah Teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada system. UML memiliki banyak diagram yang digunakan untuk melakukan pemodelan data maupun sistem, tetapi dalam buku ini dibahas 4 diagram yang biasa digunakan yaitu use case, activity diagram, class diagram, sequence diagram (Mulyani, 2016).

**2.7.1 *Use Case***

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di 20 dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Tabel *Use Case* dapat dilihat pada tabel 2.1 :

Tabel 2.1 *Use Case Diagram*

| No. | Simbol | Keterangan |
| --- | --- | --- |
| 1 | Use case | Abstraksi dan interaksi antara sistem dan aktor |
| 2 | Aktor | Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *usecase* |
| 3 | Assosciation | Abstraksi dari penghubung antara aktor dengan *usecase* |
| 4 | Ekstensi  <<extend>> | Menunjukan bahwa suatu *usecase* merupakan tambahan fungsional dari usecase lainnya jika kondisi terpenuhi |
| 5 | Generalisasi | Menunjukan spesialisasi aktor untuk dapat berpartisipasi dengan usecase |
| 6 | <<include>> | Menunjukan bahwa suatu usecase merupakan tambahan fungsional dari usecase lainnya jika suatu kondisi terpenuhi |

**2.7.2 *Activity Diagram***

Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Tabel *Acticity Diagram* dapat dilihat pada tabel 2.2 :

Tabel 2.2 *Activity Diagram*

| No. | Simbol | Keterangan |
| --- | --- | --- |
| 1 | Status Awal | Status awal aktivitas sistem. |
| 2 | Aktivitas | Aktivitas yang dilakukan sistem. |
| 3 | Percabangan | Asosiasi percabangan dimana jika ada ada pilihan aktivitas lebih dari satu. |
| 4 | Penggabungan | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5 | Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 | *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
| 7 | *State transition* | Menggambarkan aliran perpindahan kontrol antara state |
| 8 | *Fork* | Menggambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan |

**2.7.3 *Class Diagram***

*Class diagram* digunakan untuk menggambarkan struktur dalam objek sistem. Diagram ini menunjukkan class object yang menyusun sistem dan juga hubungan antara class object. Kelas memiliki atribut dan metode atau operasi. Atribut adalah variable - variabel yang mendeskripsikan properti dengan bentuk sebaris teks dalam kelas tersebut, sedangkan metode adalah fungsi yang dimiliki oleh kelas yang dalam class diagram dilambangkan menggunakan simbol-simbol. Tabel *Class Diagram* dapat dilihat pada tabel 2.3 :

Tabel 2.3 *Class Diagram*

| No | Simbol Class Diagram | Kegunaan |
| --- | --- | --- |
| 1 |  | Kelas pada struktur sistem |
| 2 |  | Sama   dengan   konsep interface dalam pemrograman berorientasi objek |
| 3 |  | Relasi  antarkelas  dengan  makna  umum,  asosiasi  biasanya  juga  disertai  dengan  multiplicity |
| 4 |  | Relasi  antarkelas  dengan  makna  kelas yang   atu  digunakan  oleh  kelas  yang lain,   asosiasi   biasanya   juga   disertai dengan multiplicity |
| 5 |  | Relasi antarkelas dengan maknageneralisasi-spesialisasi (umum khusus) |
| 6 |  | Relasi antarkelas dengan makna kebergantungan antarkelas |

**2.7.4 *Squence Diagram***

*Sequence diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Banyaknya diagram sekuen yang harus digambarkan adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya 21 pesan sudah dicakup pada diagram sekuen. Semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Tabel *Squence Diagram* dapat dilihat pada tabel 2.4 :

Tabel 2.4 *Squence Diagram*

| No. | Simbol | Keterangan |
| --- | --- | --- |
| 1 | Aktor | aktors merupakan komponen berbentuk stik figure. Komponen ini bertindak sebagai perwakilan seorang pengguna yang berinteraksi baik di dalam, maupun di luar sistem. |
| 2 | Lifeline | Menyatakan kehidupan suatu objek. |
| 3 | Objek | Object merupakan komponen berbentuk kotak. Komponen ini akan mendemonstrasikan bagaimana sebuah objek akan berperilaku dalam sebuah konteks sistem tertentu. |
| 4 | Activation box | Activation box merupakan komponen berbentuk persegi panjang yang merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugas tertentu. Semakin lama waktu yang diperlukan untuk menyelesaikan tugas pada suatu objek maka komponen ini akan berubah semakin panjang. |
| 5 | Pesan tipe create | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah  pada objek yang dibuat |
| 6 | Pesan Tipe Return | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke ojek tertentu, arah panah mengarah pada objek yang menerima  kembalian |

**2.8 PHP (*Hypertext Preprocessor*)**

PHP (*hypertext preprocessor*) adalah bahasa pemrograman web *server-side* yang bersifat *open source*. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (*server side HTML embreded scripting*). *PHP* adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan di tampilkan dibuat saat halaman itu diminta oleh *client* (Anhar, 2010).

Berikut beberapa jenis/tipe data yang sering digunakan pada bahasa pemrograman *PHP* , diantaranya :

**2.8.1 *Integer***

Merupakan tipe data *PHP* berupa bilangan bulat. Tipe data ini sering sekali digunakan pada pemrograman *PHP* khususnya yang berkaitan dengan bilangan bulat.

**2.8.2 *Float***

Dapat dikatakan juga bilangan berkoma (,) tetapi dalam pemrogramannya tidak menggunakan koma melainkan titik (.).

**2.8.3 *String***

Merupakan tipe data berupa karakter yang didalamnya bisa berupa teks atau kata.

**2.8.4 *Boolean***

Merupakan tipe data yang bisa bernilai *True* atau *False*. Tipe data boolean biasanya digunakan untuk logika, seperti pada kondisi *if* atau *looping* (pengulangan).

**2.8.5 *Array***

Merupakan kumpulan dari tipe data, didalamnya memiliki banyak nilai. Jadi dapat dikatakan *array* merupakan tipe data yang terdiri dari kumpulan tipe data.

**2.8.6 *Object***

Tipe data *object* dapat berupa bilangan, fungsi, maupun variabel. Tipe data ini digunakan supaya programer terbiasa dengan *Object Oriented Programing* (OPP) sehingga data yang dimasukan pada *script* program dapat diringkas.

**2.8.7 *Null***

*Null* merupakan tipe data khusus yang hanya memiliki satu nilai : *Null*. Sebuah variable yang berisi *Null* adalah sebuah variable yang tidak memiliki nilai, yang biasanya nanti dalam database akan ditentukan fungsi selanjutnya. *Null* tidak sama dengan nilai kosong(“) atau empty melainkan sebuah tanda nilai spesial yang sebenarnya tidak memiliki nilai.

**2.9 MySQL**

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (*database management system*) atau *DBMS* yang *multithread*, *multi user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* AB 23 membuat *MySQL General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersil untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *General Public License* (GPL) (Haqi, 2019).

**2.10 Dart**

Dart adalah bahasa pemrograman yang diproduksi oleh Google, dirancang oleh Lars Bak dan Kasper Lund. Dart pertama kali dikenalkan pada 10 Oktober 2011. Versi 1.0 dari bahasa pemrograman ini baru dirilis pada bulan November 2013.

Dart dapat digunakan untuk membuat aplikasi server (berbentuk *command-line interfcae*), web maupun *mobile* (Android dan iOS) (Raharjo, 2019).

**2.11 Flutter**

Flutter adalah *software development kit* (SDK) buatan Google yang berfungsi untuk membuat aplikasi *mobile* menggunakan bahasa pemrograman Dart, baik untuk Android maupun iOS. Dengan Flutter, aplikasi Android dan iOS dapat dibuat menggunakan basis kode dan bahasa pemrograman yang sama, yaitu Dart, bahasa pemrograman yang juga diproduksi oleh Google pada tahun 2011. Sebelumnya, aplikasi murni (*native*) untuk Android perlu dibuat menggunakan bahasa permograman Java atau Kotlin, sedangkan aplikasi iOS perlu dibuat menggunakan bahasa pemrograman Objective-C atau Swift. Flutter ditujukan untuk mempermudah dan mempercepat proses pengembangan aplikasi *mobile* yang dapat berjalan di atas Android dn iOS, tanpa harus mempelajari dua bahasa pemrograman secara terpisah (Raharjo, 2019).

**2.12 Kajian Pustaka**

Yulianto, Whelly (2015) melakukan penelitian pengukuran jarak hotel terdekat dengan metode *haversine* *formula*. Latar belakang masalah dari penelitian ini adalah pencarian lokasi kebutuhan tempat menginap di kota Malang bagi para wisatawan atau pebisnis yang dekat dengan posisinya saat itu. Tujuan dari penelitian ini adalah dapat membangun suatu aplikasi yang dapat merekomendasikan hotel terdekat dari posisi pengguna dan memberikan petunjuk arah sehingga dapat meminimalisir tenaga, waktu dan biaya. Metode yang digunakan untuk menyelesaikan masalah adalah dengan metode *Haversine Formula*. Hasil dari penelitian ini adalah sebuah sistem yang dapat menampilkan lokasi hotel terdekat dari posisi pengguna.

Arizal, Muhammad (2017) melakukan penelitian pengukuran jarak bengkel terdekat dengan metode *Haversine Formula*. Latar belakang masalah dari penelitian ini adalah pencarian lokasi bengkel terdekat disekitar area kota/kabupaten Kediri. Tujuan dari penelitian ini adalah dapat membangun suatu aplikasi yang dapat mengetahui lokasi bengkel terdekat dari posisi pengguna sehingga pengguna dapat menghemat waktu dalam menemukan bengket terdekat karena tidak perlu lagi mencari secara manual dengan cara bertanya pada masyarakat sekitar.

Andriani, Wita (2019) melakukan penelitian pencarian lokasi Stasiun Pengisian Bahan bakar Umum (SPBU) terdekat di wilayah Ciputat dengan menggunakan metode *Haversin Formula*. Latar belakang masalah dari penelitian ini adalah perlunya suatu sistem aplikasi agar dapat membantu pengendara dalam mencari SPBU terdekat dari lokasi pengendara berada menuju lokasi yang dituju dengan memanfaatkan teknologi GPS. Tujuan dari penelitian ini adalah untuk mengimplementasikan *Formula Haversine* dalam mencari SPBU di Ciputat dari sisi akurasi jarak.

Agil, Rido (2019) melakukan penelitian pengukuran jarak tempat pariwisata terdekat di wilayah kabupaten Kebumen dengan metode *haversine* *formula*. Latar belakang masalah dari penelitian ini adalah kurangnya sumber yang menyediakan informasi tentang pariwisata di kabupaten Kebumen. Tujuan dari penelitian ini adalah dapat membangun suatu aplikasi yang dapat merekomendasikan tempat pariwisata terdekat dari posisi pengguna dan memberikan petunjuk arah. Metode yang digunakan untuk menyelesaikan masalah adalah dengan metode *Haversine Formula*. Hasil dari penelitian ini adalah sebuah sistem yang dapat menampilkan lokasi tempat pariwisata di wilayah kabupaten Kebumen.

Yulsiliviana, Ekawati (2021) melakukan penelitian pencarian *driver* pelanggan *dry cleaning order* terdekat dengan menerapkan metode *haversine formula*. Latar belakang masalah dari penelitian ini adalah perlunya suatu sistem yang dapat mencarikan *driver* dari pelanggan *dry cleaning order* yang terdekat dari pelanggan. Tujuan dari penelitian ini adalah membangun sistem untuk membantu pencarian *driver* terdekat dengan pemesan agar cepat dalam mengambil pakaian kotor.

Tabel 2.5 Persamaan dan perbedaan artikel yang di *review* dengan penelitian yang dilakukan.

|  |  |  |
| --- | --- | --- |
| Peneliti (tahun) | Persamaan | Perbedaan |
| Yulianto, Whelly (2015) | Membuat sistem yang dapat membantu pencarian lokasi dari objek yang ditentukan | Objek penelitian hotel di wilayah Malang |
| Arizal, Muhammad (2017) | Membuat sistem pencarian lokasi terdekat dengan metode *haversine formula* | Objek penelitian bengkel di wilayah kabupaten Kediri |
| Andriani, Wita (2019) | Membuat sistem pencarian lokasi dengan memanfaatkan teknologi GPS | Objek penelitian SPBU di wilayah Ciputat |
| Agil, Rido (2019) | Membuat sistem pencarian lokasi terdekat dengan pengguna menggunakan metode *haversine formula* | Objek penelitian tempat pariwisata di kabupaten Kebumen |
| Yulsiliviana, Ekawati (2021) | Membuat sistem pencarian lokasi dengan memanfaatkan teknologi GPS | Objek penelitian lokasi *driver dry cleaning order* |

Tabel 2.5 Memperlihatkan persamaan dan perbedaan dengan penelitian. Menurut pandangan penulis persamaan dari jurnal tersebut adalah membuat sistem yang dapat mengotomatisasi pencarian jarak terdekat menggunakan metode *Haversine Formula*. Perbedaannya adalah pada objek penelitian.