

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penelitian terkait ini, penulis mengkaji penelitian terdahulu mengenai sistem aplikasi yang dilakukan oleh penelitian-penelitian sebelumnya yang dapat menjadi dasar dari penelitian yang dilakukan. Beberapa penelitian mengenai sistem pencarian lokasi *fitness* beserta layanan jasa *personal trainer* di antaranya adalah sebagai berikut:

Pada penelitian yang dilakukan oleh (Koloay et al., 2020) “Rancang Bangun Aplikasi *Fitness* Berbasis Android (Studi Kasus : Popeye Gym Suwaan)” Dalam peneliti akan membuat aplikasi *fitness* , jadi dengan begitu lebih mempermudah para member mendapatkan gerakan *fitness* yang baik dan benar. Berdasarkan permasalahan ini peneliti mengembangkan suatu penelitian dalam bentuk aplikasi android yang dapat membantu mempermudah para member gym mendapatkan informasi serta pengetahuan tentang gerakan *fitness*.

Pada penelitian yang dilakukan oleh (Nasution, 2021) “Rancang Bangun Aplikasi Pencarian Sarana Kebugaran di Kabupaten Purwakarta Menggunakan Location Based Service Berbasis Android” dalam aplikasi ini akan membangun aplikasi pencarian dan pemesanan *fitness center* berplatform android, kemudian juga membangun aplikasi yang menampilkan lokasi terdekat dari jarak pengguna. Hasil dari aplikasi yang dibangun dapat membantu bisnis *fitness* mempromosikan tempat *fitness* di kab Purwokerto kepada calon konsumen, dalam hal lain memudahkan pengguna dalam pencarian informasi yang ada serta mempermudah penggunaan dalam menghubungi pemilik bisnis *fitness* dengan cepat melalui aplikasi android.

Pada penelitian yang dilakukan oleh (Triyanto et al., 2021) ” Analisis Profesional *Personal trainer* di *Fitness center* Hotel Kota Semarang Tengah” dalam aplikasi ini dan hasil observasi yang dilakukan kepada member banyak member yang memilih jasa pelayanan *personal trainer* komplain terhadap hasil program latihan dikarenakan kurangnya pengetahuan dan keterampilan *personal trainer* mengenai program latihan sehingga apa yang ingin dicapai oleh member tidak sesuai dengan harapannya, padahal harga untuk menggunakan jasa pelayanan

seorang *personal trainer* sangat mahal, maka dari itu member mengharapkan seorang *personal trainer* harus profesional, akan tetapi banyak seorang *personal trainer* yang bekerja berdasarkan dari pengalaman pribadi sewaktu menjadi member di *fitness center* atau pernah menjuarai dalam berbagai kompetisi fisik, angkat berat atau binaraga. Oleh karena itu, peneliti tertarik ingin mengambil tema tentang Analisis Profesional *Personal trainer* di *Fitness center* Hotel Kota Semarang Tengah Tahun 2019.

Pada penelitian yang dilakukan oleh (Ramadhan, 2019) “Implementasi *Location Based Service (LBS)* Untuk Informasi Persewaan Alat Outdoor di Kota Madiun” dalam penelitian membuat sistem Informasi mengenai persewaan alat outdoor di kota madiun yang belum memadai menjadi dasar permasalahan sistem ini akan dibuat. Nantinya sistem ini menjadi media yang menjembatani antara pengguna jasa dengan penyedia jasa persewaan peralatan alat outdoor khususnya di kota madiun dalam konteks jarak terdekat. Dalam sistem ini nanti pengguna hanya perlu mencari layanan jasa persewaan alat outdoor melalui sistem berdasarkan jarak terdekat dari lokasi pengguna ke lokasi persewaan alat outdoor, serta dapat melihat estimasi jarak, waktu serta jalan yang ditempuh dari lokasi pengguna ke lokasi persewaan. Bagi penyedia jasa persewaan alat outdoor nantinya data-data nya dimasukan terlebih dahulu pada sistem supaya dapat muncul pada sistem untuk mempermudah pengguna untuk mencari informasi mengenai persewaan. Pada sistem ini nantinya dapat menjadi media bagi para jasa persewaan alat outdoor dengan melakukan promosi dan memperluas target pemasaran serta dapat meningkatkan pendapatan.

Pada penelitian yang dilakukan oleh (Oktofri, 2021) “Aplikasi Pencarian Lokasi Dokter Praktik Dan Apotek Berdasarkan Kriteria Pasien Di Kota Pontianak Menggunakan Metode *Location Based Service (LBS) Berbasis Progressive Web App*” Aplikasi ini menggunakan metode *Location Based Service* untuk menentukan dan mengatur jarak radius pencarian pada persebaran dokter praktik yang ada di Kota Pontianak.

Pada penelitian yang dilakukan oleh (Prayetno, 2021) “Aplikasi *Marketing Perumahan Multi User* Dengan Menggunakan Metode *Location Based Service* studi kasus: DPD REI Provinsi Kalimantan Barat” Dengan pengimplementasian aplikasi,

DPD REI Kalbar dapat memfasilitasi setiap anggota yang tergabung di bawah asosiasi REI untuk membantu dalam meningkatkan promosi penjualan rumah dengan informasi yang lebih detail seperti letak lokasi rumah, tipe rumah, harga rumah, dan fasilitas umum apa saja yang terdapat disekitar lokasi perumahan. Aplikasi dibangun dengan menerapkan *marketing multi user* dan metode *Location Based Service (LBS)*. Dengan tujuan setiap agensi dapat mengelola perumahan yang dipasarkan serta memiliki akun sendiri untuk setiap agensinya, sedangkan penerapan LBS itu sendiri digunakan untuk menampilkan informasi terkait fasilitas umum terdekat yang berada disekitar lokasi perumahan, sehingga dapat menjadikan pertimbangan untuk masyarakat dalam memilih perumahan yang diinginkan.

Berdasarkan penelitian terkait yang sudah dipaparkan di atas, maka pada penelitian ini penulis akan membangun sebuah Aplikasi Pencarian Lokasi *Fitness* Beserta Layanan Jasa *Personal Trainer* di Kota Pontianak Menggunakan Metode *Location Based Service*. Perbedaan penelitian ini dengan penelitian-penelitian sebelumnya adalah aplikasi yang dibuat penulis menggunakan metode *location based service* serta penentuan lokasi *fitness*, serta dapat melakukan booking jasa *personal trainer* yang dimuat dalam bentuk *website*.

Berikut adalah hal-hal yang membedakan setiap kajian terkait yang dijelaskan pada tabel 2.1 berikut ini.

Tabel 2.1 Kajian Terkait

No	Penulis	Judul	Keterangan
1.	Klaudio Koloay, Sherwin R. U. A. Sompie, Sary D. E. Paturusi (2020) Teknik Elektro Universitas Sam Ratulangi Manado	Rancang Bangun Aplikasi <i>Fitness</i> Berbasis Android (Studi Kasus : Popeye Gym Suwaan)	1. Aplikasi Berbasis Android 2. Aplikasi dapat menampilkan Gerakan- gerakan dalam latihan fitness. 3. Dapat mengukur berat badan

No	Penulis	Judul	Keterangan
			ideal member fitness.
2.	Winda Suci Lestari Nasution (2021) Universitas Esa Unggul Jakarta	Rancang Bangun Aplikasi Pencarian Sarana Kebugaran di Kabupaten Purwakarta Menggunakan Location Based Service Berbasis Android	1. Aplikasi Berbasis Android 2. Menggunakan metode <i>location based service</i> . 3. Dapat mendaftar secara online. 4. Aplikasi membantu rajawali Fitness.
3.	Triyanto, Lusiana, & Purnama, Y. (2021) Universitas PJKR FKIP Universitas Wahid Hasyim Semarang	Analisis Profesional <i>Personal trainer</i> di <i>Fitness center</i> Hotel Kota Semarang Tengah	1. Analisis Profesional Personal Trainer di Hotel Kota Semarang Tengah.
4.	Ramadhan, S. R. (2019) Universitas Teknologi Yogyakarta	Implementasi Location Based Service (LBS) Untuk Informasi Persewaan Alat Outdoor di Kota Madiun	1. Aplikasi Berbasis <i>web</i> 2. Menggunakan metode <i>location based service</i> . 3. pengguna dapat melihat jarak terpendek instansi persewaan

No	Penulis	Judul	Keterangan
5.	Fajrie Dwi Oktofri (2021), Universitas Tanjungpura Pontianak	Aplikasi Pencarian Lokasi Dokter Praktik Dan Apotek Berdasarkan Kriteria Pasien Di Kota Pontianak Menggunakan Metode <i>Location Based Service</i> (LBS) Berbasis <i>Progressive Web App</i> .	<ol style="list-style-type: none"> 1. Aplikasi berbasis <i>Progressive Web App</i>. 2. Menampilkan persebaran lokasi dokter praktik dan apotek. 3. Persebaran dokter praktik di lokasi Apotek, Klinik, Optik, dan Praktik Mandiri. 4. Menampilkan Jadwal Operasional. 5. Memanfaatkan <i>GPS</i> dan <i>Google Maps</i> API sebagai acuan lokasi pengguna untuk menentukan lokasi pencarian terdekat.
6.	Wiwin Galuh Prayetno (2021) Universitas	Aplikasi <i>Marketing</i> Perumahan <i>Multi User</i> Dengan	<ol style="list-style-type: none"> 1. Aplikasi Berbasis <i>web</i>

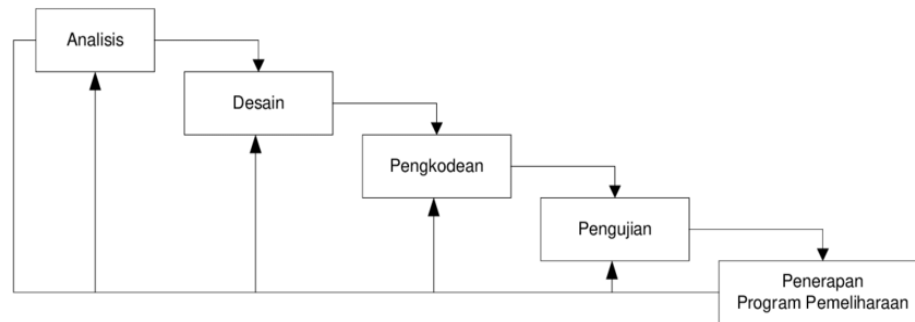
No	Penulis	Judul	Keterangan
	Tanjungpura Pontianak	Menggunakan Metode <i>Location Based Service</i> (Studi Kasus: DPD REI Provinsi Kalimantan Barat)	<ol style="list-style-type: none"> 2. Menggunakan metode <i>location based service</i>. 3. Menggunakan Google maps API untuk menentukan titik rumah dan fasilitas umum yang dicari 4. Aplikasi menampilkan titik-titik fasilitas umum yang sudah dipilih pada saat melakukan pencarian dengan radius yang dapat ditentukan oleh sistem. 5. Aplikasi dapat melakukan booking rumah.

Tabel 2.2 Penelitian yang dilakukan.

No	Penulis	Judul	Keterangan
1.	Andri Kurniawan (2022), Universitas Tanjungpura Pontianak	Aplikasi pencarian lokasi <i>fitness</i> beserta layanan jasa <i>personal</i> <i>trainer</i> di Kota Pontianak menggunakan <i>location based</i> <i>service</i>	<ol style="list-style-type: none"> 1. Aplikasi Berbasis <i>web</i> 2. Menggunakan metode <i>location based service</i>. 3. Menampilkan persebaran lokasi pusat kebugaran. 4. Memanfaatkan <i>GPS dan Google Maps API</i> sebagai acuan lokasi pengguna untuk menentukan lokasi pencarian pusat kebugaran. 5. Aplikasi dapat melakukan <i>booking personal trainer</i>.

2.2 Model Waterfall

Terdapat beberapa metodologi *Systems Development Life Cycle (SDLC)* yang biasa digunakan dalam membangun sebuah sistem, salah satunya adalah model *waterfall*. *Waterfall* merupakan model yang bersifat sistematis dan termasuk dalam model klasik, nama lainnya adalah *Linear Sequential Model* (Pressman, 2001). Tahapan-tahapan model *waterfall* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model Waterfall (Pressman, 2001)

Penjelasan tahapan-tahapan *waterfall* tersebut yaitu:

1. *Analysis* (Analisis)

Fase ini merupakan proses analisa terhadap sistem yang sedang berjalan dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem, sehingga kebutuhan yang diperlukan untuk sistem baru akan didapatkan.

2. *Design* (Perancangan)

Perancangan merupakan proses penentuan cara kerja sistem dalam hal perancangan antarmuka, database, dan perancangan alur program. Perancangan diperlukan untuk menggambarkan sistem baru dengan tujuan memenuhi kebutuhan pengguna.

3. Pengkodean

Tahapan implementasi yaitu tahap rancangan sistem yang dibentuk menjadi suatu kode program untuk pembuatan sistem.

4. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur atau tidak dan memastikan sistem terhindar dari *error* yang terjadi. Testing juga dilakukan untuk memastikan kevalidan dalam proses input sehingga dapat menghasilkan output yang sesuai.

5. *Maintenance* (Pemeliharaan)

Fase ini yaitu pemeliharaan dan pengembangan sistem yang berguna untuk melihat kemampuannya, mengecek jika masih ada ditemukan error atau ada penambahan fitur-fitur yang belum ada pada sistem tersebut. Pengembangan diperlukan ketika adanya perubahan dari pengguna seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

2.3 Gym

Gymnasium adalah lapangan atau ruangan luas yang digunakan sebagai tempat untuk berlatih dan berolahraga. Bangunan ini dikelilingi oleh banyak kolom dan juga terdapat lobby, toilet, ruang sauna, dan sebagainya. Ruangan yang di dalamnya terdapat peralatan untuk latihan fisik atau untuk membentuk tubuh (Longman, Dictionary of Contemporary English, British National Corpus). *Gymnasium* ada sejak dahulu dan terdapat di negara romawi, karena bangsa romawi sudah mengenal adanya adu kekuatan dan disediakan tempat untuk adu kekuatan yaitu Colosseum. *Gymnasium* biasanya terdapat fasilitas yang menunjang olahraga lainnya. Biasanya seperti loker dan semacam fasilitas service umum, seperti toilet, cleaning service sehingga hampir sama dengan bangunan publik lainnya.

2.4 Aplikasi

Menurut (Karnovi & Habibi, 2020), aplikasi adalah sebuah program yang siap pakai yang bisa dipakai untuk menjalankan sebuah perintah dari pengguna aplikasi itu sendiri. Karnovi & Habibi (2020) juga menyatakan bahwa aplikasi adalah pemecah masalah yang memakai salah satu teknik pemrosesan data aplikasi yang mengacu pada sebuah komputerasi atau *smartphone* yang diinginkan atau diharapkan.

Berdasarkan pengertian aplikasi di atas, maka dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang dibuat dengan tujuan untuk mempermudah penggunaanya dalam mengerjakan suatu pekerjaan berdasarkan kemudahan dengan memanfaatkan teknik pemrosesan data yang telah di program sebelumnya.

2.5 Fitness center

Dalam Bahasa Inggris *fitness center* berasal dari kata *fitness* dan *center*, yang dalam Bahasa Indonesia *fitness* artinya kebugaran dan *center* artinya pusat, jadi *fitness center* adalah pusat kebugaran. Menurut (Giriwijoyo, 2004) “pusat kebugaran adalah suatu kegiatan dalam ruangan dengan menawarkan kegiatan olahraga dari yang tanpa menggunakan alat, sampai yang menggunakan alat-alat yang mahal dan canggih, yang diantaranya bertujuan prestasi.” Kemudian Hanafi (1997:9) menjelaskan bahwa: “pusat kebugaran adalah tempat olahraga dalam

ruangan yang menawarkan berbagai program latihan kebugaran dengan fasilitas dan peralatan yang mutakhir”. Dikarenakan usaha ini bergerak di bidang jasa, maka para pelaku usaha ini pasti bersaing dalam menarik pelanggan sebanyak-banyaknya dan berupaya untuk meningkatkan kepuasan pelanggannya, hal ini dilakukan dengan cara menawarkan pelayanan yang bervariasi diantaranya penyediaan sarana dan prasarana yang lengkap dan nyaman hingga penyediaan jasa *personal trainer* atau instruktur. Upaya-upaya yang telah disebutkan sebelumnya dilakukan oleh pengelola merupakan salah satu cara untuk menciptakan kepuasan pelanggan sehingga pelanggan dapat tertarik menggunakan fasilitas yang disediakan oleh pengelola.

2.6 Personal Trainer

Industri kebugaran di Indonesia yang berkembang pesat telah memungkinkan berkembangnya profesi sebagai pelatih pribadi (*personal trainer*) atau lebih sering dikenal dengan sebutan PT. *Personal trainer* adalah suatu profesi dimana penampilan dan pencitraan diri menjadi jati diri. Menurut (Choky Wijaya, 2011) yang dikutip dari Sheknows, American Council on Exercise mengatakan bahwa “dengan menggunakan *personal trainer* akan membuat members termotivasi lebih giat untuk melakukan latihan kebugaran dan juga membantu monitoring asupan makanan yang sehari-hari dikonsumsi oleh members”. Bagi para members pemula, seorang *personal trainer* dianggap perlu untuk membantu melakukan latihan kebugaran secara lebih optimal sejak awal.

2.7 Sistem Informasi Geografis

Sistem informasi geografis merupakan sebuah data yang mengacu pada posisi, objek dan hubungan diantaranya dalam ruang bumi. SIG secara khusus dibuat untuk mengelola data yang bereferensi keruangan dan merupakan sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis seperti data yang diidentifikasi menurut lokasinya dalam sebuah *database*. SIG terdiri dari suatu komponen yang terdiri dari perangkat keras, perangkat lunak, sumber daya manusia dan data yang bekerja bersama secara efektif untuk memasukkan, menyimpan, memperbaiki, memperbaharui, mengelola, memanipulasi, mengintegrasikan, menganalisa dan

menampilkan data dalam suatu informasi berbasis geografis. Adanya aplikasi SIG dapat menjawab beberapa pertanyaan mengenai lokasi, kondisi, trend, pola dan pemodelan dan penayangan data lokasi secara digital bahkan dapat diintegrasikan secara beragam mulai dari citra satelit, foto udara, dan peta (Hutagulung, 2020:79)

2.8 Global Positioning System (GPS)

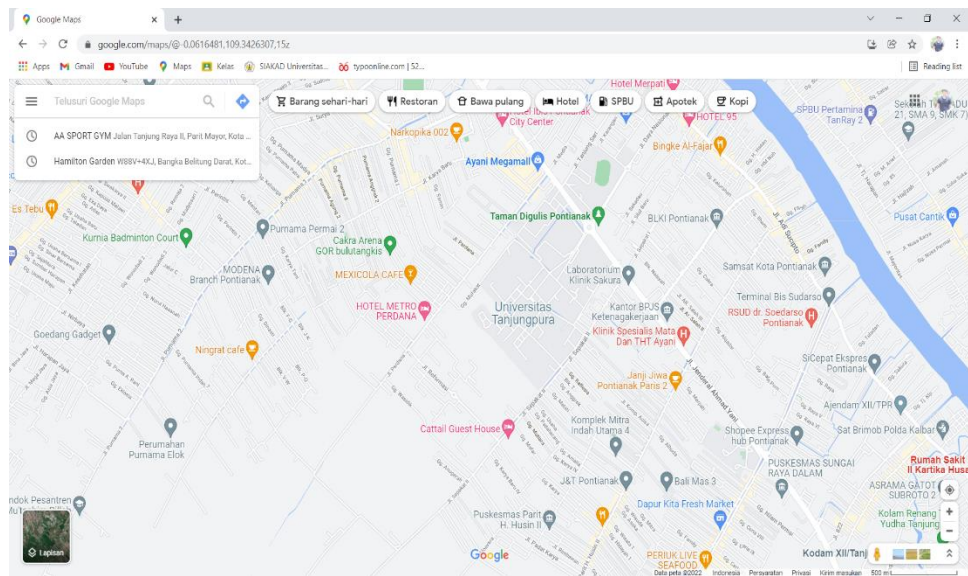
Global Positioning System atau sering disebut *GPS* adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. *GPS* pertama kali dikembangkan oleh Amerika Serikat dengan tujuan untuk membantu pasukan Amerika Serikat dalam menghadapi perang Vietnam. Fungsi utama dari *GPS* adalah untuk memberikan informasi tentang posisi, kecepatan dan waktu secara akurat. Akurasi atau ketepatan perlu mendapat perhatian bagi penentuan koordinat sebuah titik atau lokasi. Koordinat posisi ini akan selalu mempunyai ‘faktor kesalahan’, yang lebih dikenal dengan ‘tingkat akurasi’. Misalnya alat tersebut menunjukkan sebuah titik koordinat dengan akurasi 3 meter, artinya posisi sebenarnya bisa berada dimana saja dalam radius 3 meter dari titik lokasi. Tingkat akurasi *GPS* sangat dipengaruhi oleh faktor sekeliling yang mengurangi kekuatan sinyal satelit. Ada banyak hal yang dapat mengurangi kekuatan sinyal satelit, diantaranya:

1. Kondisi geografis, selama kita masih dapat melihat langit yang cukup luas, kekuatan sinyal semakin baik pula
2. Hutan, Makin lebat hutannya maka makin berkurang sinyal yang didapat.
3. Saat menyelam ke dalam air, *GPS* tidak akan berfungsi.
4. Berada diantara gedung tinggi atau di dalam gedung juga dapat mengganggu kekuatan sinyal.

GPS memungkinkan developer mengakses lokasi pengguna kapan pun. Sistem bisa melacak (*tracking*) lokasi ketika pengguna mengubah lokasi, bisa menentukan lokasi ponsel kemudian mengakses web untuk menentukan lokasi publik yang dekat dengan pengguna (Kasman, 2013)

2.9 Google Maps

Google Maps adalah sebuah jasa peta global virtual gratis dan online yang disediakan oleh perusahaan *Google*. *Google Maps* yang dapat ditemukan di alamat <http://maps.google.com>. *Google Maps* menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia. *Google Maps* juga menawarkan pencarian suatu tempat dan rute perjalanan. Dapat dilihat pada gambar 2.2.

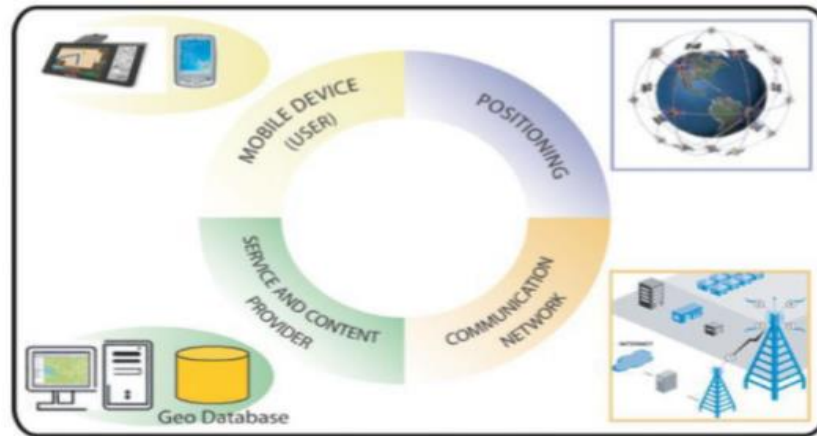


Gambar 2.2 Tampilan *Google Maps*

Google Maps API adalah sebuah layanan (*service*) yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi enterprise di dalam websitenya, *Google Maps API* adalah suatu *library* yang berbentuk *Javascript* (Kindarto, 2008). Pada penelitian ini *Google Map API* akan menjadi *service* seperti *interface* bagi pengguna berupa *Map* yang akan dipakai.

2.10 Location Based Service (LBS)

Location Based Service (LBS) atau Layanan Berbasis Lokasi merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak (Steiniger et al., 2006). Dalam layanan Berbasis Lokasi terdapat lima komponen penting seperti terlihat pada Gambar 2.3.



Gambar 2.3 Komponen Dasar *Location Based Service* (Steiniger et al., 2006)

Setiap komponen mempunyai fungsi (Steiniger et al., 2006).

1. *Mobile Devices*, merupakan suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan *text*.
2. *Communication Network*, komponen ini mengirim data pengguna dan informasi yang diminta dari *Mobile terminal* ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupa jaringan *seluler* (GSM, CDMA), *Wireless Local Area Network* (WLAN), atau *Wireless Wide Area Network* (WWAN).
3. *Positioning Component*, digunakan untuk memproses suatu layanan maka posisi pengguna harus diketahui.
4. *Service dan Application Provider*, penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.
5. *Data dan Content Provider*, penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, *data* dapat diminta dari *data dan content provider*.

Pada penelitian ini memanfaatkan Layanan Berbasis Lokasi atau yang kita kenal dengan *LBS* merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak seperti *smartphone* yang akan dipakai pada penelitian ini.

2.11 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. (Nugroho, 2005).

Sedangkan menurut (A. S., Rosa dan Shalahuddin, 2013) UML (*Unified Modeling Language*) adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat

Di Dalam UML ada 13 buah diagram yang dikelompokkan kedalam tiga kategori yaitu:

1. *Structure Diagrams*, yaitu kumpulan diagram-diagram yang menggambarkan struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams*, yaitu kumpulan diagram-diagram yang menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi di dalam sistem.
3. *Interaction Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan interaksi sistem dengan sistem lain ataupun interaksi antar subsistem dalam sebuah system.




Secara garis besar, beberapa diagram utama sudah dapat menggambarkan keseluruhan sistem. Diagram tersebut antara lain *use case diagram*, *class diagram*, *sequence diagram*, dan *activity diagram*.

2.11.1 Use Case Diagram

Use Case Diagram bersifat statis, yang memperlihatkan himpunan *Use Case* dan aktor-aktor (suatu jenis khusus dari kelas) dan menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sistem (Widodo, Prabowo.P, 2011).

Deskripsi simbol-simbol yang digunakan pada *Use Case Diagram* dapat dilihat pada Tabel 2.3 (Nugroho, 2010).

Tabel 2.3 Deskripsi Notasi pada Use Case Diagram



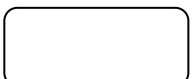
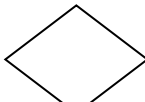

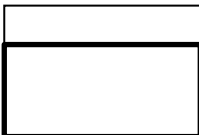
No	Notasi	Nama	Deskripsi
1		Use Case	Menggambarkan fungsionalitas yang dimiliki sistem
2		Actor	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang dikembangkan
3		Association	Lintasan komunikasi antara actor dengan use case
4	<<extend>> ----->	Extended (Ekstensi)	Penambahan perilaku ke suatu

No	Notasi	Nama	Deskripsi
			<i>use case</i> dasar
5	<pre><<include>> <- - - - -</pre>	<p><i>Include</i> (Menggunakan)</p>	<p>Penambahan perilaku ke suatu <i>use case</i> dasar yang secara <i>explicit</i> mendeskripsikan penambahan tersebut</p>
6	<pre>—————></pre>	<p>Generalisasi / <i>generalization</i></p>	<p>Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya</p>

2.11.2 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity diagram* sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan *parallel behavior* atau menggambarkan interaksi antara beberapa *use case*. Berikut merupakan simbol notasi *Activity Diagram* pada Tabel 2.4 (Rosa A. S dan M. Salahuddin, 2013).

Tabel 2.4 Deskripsi Notasi pada Activity Diagram.

No	Notasi	Nama	Deskripsi
1.		Status Awal (<i>initial node</i>)	Status awal aktivitas, sebuah diagram aktivitas memiliki status awal.
2.		Status Akhir (<i>final node</i>)	Status akhir yang dilakukan sistem.
3.		Aktivitas (<i>activity</i>)	Aktivitas yang dilakukan oleh sistem, biasanya diawali oleh kata kerja.
4.		Percabangan (<i>decision</i>)	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
5.		Penggabungan (<i>join</i>)	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.11.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. (A. S., Rosa dan Shalahudin, M, 2003). *Class Diagram* mendeskripsikan jenis – jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka.(Fowler, 2005). *Class Diagram* digunakan untuk menggambarkan hubungan kelas – kelas antara satu dengan yang lain serta memiliki atribut dan

operasi yang terdapat dalam sistem yang akan dibuat. Atribut merupakan variabel – variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu kelas. Atribut dan metode dapat memiliki salah satu sifat sebagai berikut:

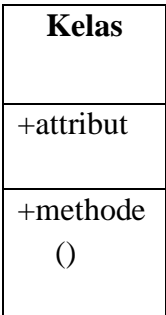


1. *Private* (-), hanya dapat digunakan oleh *class* yang memilikinya
2. *Public* (+), dapat digunakan oleh *class* lain.
3. *Protected* (#), hanya dapat dipanggil oleh *class* yang bersangkutan jumlah suatu anak yang mewarisinya.

Nilai kardinalitas dan *multiplicity* sebuah *class* menunjukkan jumlah suatu objek yang dapat berhubungan dengan objek lain. Berikut nilai kardinalitas atau *multiplicity* pada Tabel 2.5 (Tohari, Hamim 2014) dan notasi *class diagram* pada Tabel 2.5 (Rosa A. S dan M. Salahuddin, 2013).

Tabel 2.5 Jenis-jenis Multiplicity

No	Indikator	Keterangan
1	0 .. 1	Nol atau satu
2	1	Hanya satu
3	0 .. *	Nol atau lebih
4	1 .. *	Satu atau lebih

Tabel 2.6 Deskripsi Notasi pada Class Diagram

No	Notasi	Nama	Deskripsi
1		<i>Class</i>	Menggambarkan konsep dasar pemodelan sistem.
2		Asosiasi (<i>Association</i>)	Sebuah garis solid antara dua <i>class</i> , ditarik dari <i>class</i> sumber ke <i>class</i> target lebih spesifik, digunakan dalam struktur pewarisan.
3		Ketergantungan (<i>Dependency</i>)	Relasi antara dua elemen jika perubahan definisi sebuah elemen

No	Notasi	Nama	Deskripsi
			(<i>supplier</i> atau sumber) dapat menyebabkan perubahan pada elemen lainnya (<i>Client</i> atau target).

2.12 Teknologi Pendukung

2.12.1 Web

World Wide Web (WWW) atau biasa disebut dengan web merupakan salah satu sumber daya Internet yang berkembang pesat. Pertama kali aplikasi web dibangun hanya dengan menggunakan bahasa yang disebut *HTML* (*HyperText Markup Language*) dan protokol yang digunakan dinamakan *HTTP* (*HyperText Transfer Protocol*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan *HTML* yang sekarang ini terdapat banyak skrip seperti: PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah *applet* (*java*). *a web* atau aplikasi berbasis *web* (*Web-based application*) adalah aplikasi untuk menyampaikan informasi kepada pengguna yang menggunakan layanan *Internet* berbasis *web*.

Dalam aplikasi tersebut, terjadi pertukaran antara *klien* (komputer yang meminta informasi) dengan *server* (komputer yang memasok atau menanggapi informasi). *Web* memberikan informasi secara *online* melalui internet langsung. *Klien* melakukan permintaan informasi dengan menggunakan *browser* (contoh *browser*: *Internet Explorer*, *Opera*, *Mozilla*, dan sebagainya). *Server* menerima informasi dan melayani permintaan dari *client*. Hal ini biasa disebut dengan *web server* (contoh *web server*: *Apache*, *IIS*, *Xitami*, dan sebagainya). Setelah itu, *web server* akan berkomunikasi dengan *middleware* (contoh *middleware*: *ASP*, *JSP*, *PHP*, dan sebagainya) untuk bisa berhubungan dengan basis data atau *database* (contoh *database*: *access*, *oracle*, *sql*, dan sebagainya). Setelah berinteraksi dengan *database*, *server* yang telah mendapatkan informasi akan memberikan tanggapan terhadap *klien* yang meminta informasi tadi (Abdul Kadir, 2005).

2.12.2 XAMPP

XAMPP merupakan paket *PHP* dan *MySQL* berbasis *open source*, yang digunakan sebagai alat pembantu pengembangan aplikasi berbasis *PHP*. *XAMPP*

mengkombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket (Riyanto, 2010).

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Dengan menginstall *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, *PHP* dan *MySQL* secara manual, *XAMPP* akan menginstalasi dan mengkonfigurasinya secara otomatis (Fathoroni et al., 2020).

2.12.3 MYSQL

MySQL adalah *software* atau program aplikasi database, yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi teks dan juga angka. Semua informasi data yang dipakai di dalam satu program aplikasi disimpan dalam satu *software* database, yaitu MySQL.

Dengan menggunakan database MySQL, maka data yang tersimpan di dalam database dapat diakses secara bersama menggunakan beberapa komputer/laptop yang berbeda, konsep ini sering disebut dengan sistem *multi user*. Database MySQL juga dapat diinstal pada sebuah komputer pusat (*server*) yang tersimpan di dalam ruang *server*, kemudian datanya diakses melalui komputer/laptop yang terinstal dengan program *client* seperti program kasir toko atau program akademik siswa yang letaknya berbeda tempat, dan konsep tersebut disebut dengan sistem *client/server* (Nugroho, 2014).

2.12.4 Javascript

Javascript adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer (IE)*, *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan *tag SCRIPT* (Sunyoto, 2007). Beberapa hal tentang *Javascript*:

1. *Javascript* didesain untuk menambah interaktif suatu *web*.
2. *Javascript* merupakan sebuah bahasa *scripting*.
3. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
4. *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*).
5. *Javascript* biasanya disisipkan (*embedded*) dalam halaman *HTML*.

6. *Javascript* adalah bahasa *interpreter* (yang berarti *script* dieksekusi tanpa proses kompilasi).

2.12.5 HTML

HTML merupakan halaman yang berada pada suatu situs internet atau *web*. *HTML* merupakan metode yang menautkan (*link*) satu dokumen ke dokumen lain melalui teks. Menurut Deris Setiawan, *HTML* merupakan *framework* internet, hampir semua situs web yang ada menggunakan *HTML* untuk menampilkan teks, grafik, suara, dan animasinya (Kuswayatno, 2006).

HTML adalah suatu bahasa yang dikenali oleh web browser untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*) (Oktavian & Puji, 2010).

Menurut (Achmad Solihin, 2016), *HTML* merupakan singkatan dari *Hyper Text Markup Language*. *HTML* dikembangkan pertama kali oleh Tim Berners-Lee bersamaan dengan protokol *HTTP* (*Hypertext Transfer Protocol*) pada tahun 1989. Tujuan utama pengembangan *HTML* adalah untuk menghubungkan suatu halaman web dengan halaman web lainnya. Tentunya pada awal pengembangannya.

2.12.6 Bootstrap

Bootstrap merupakan sebuah *framework CSS* yang memudahkan pengembang untuk membangun website yang menarik dan responsif. *Bootstrap* adalah *CSS* tetapi dibentuk dengan *LESS*, sebuah *pre-processor* yang memberi fleksibilitas dari penggunaan *CSS* biasa. *Bootstrap* dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan *web* yang mengutamakan desain (Otto, 2011).

2.12.7 PHP

Menurut (MADCOM, 2016), “PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs *web* dinamis”. PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP license*. Untuk membuat program PHP kita diharuskan untuk menginstal *web server* terlebih dahulu. PHP mendukung komentar seperti pada bahasa ‘C’, ‘C++’, dan Unix shell-style. (*Perl style*).

2.13 Pengujian perangkat lunak

Pengujian perangkat lunak merupakan suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah sesuai dengan yang diharapkan atau belum. Menurut Pressman (2002), pengujian adalah proses eksekusi suatu program untuk menemukan kesalahan sebelum digunakan oleh pengguna akhir (*end-user*).

2.13.1 Pengujian *Black Box*

Black Box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman, 2010).

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung untuk menemukan hal-hal berikut (Mustaqbal dan Firdaus, 2015).

1. Fungsi yang tidak benar atau tidak ada
2. Kesalahan antarmuka (*interface errors*)
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan performansi (*performance errors*)
5. Kesalahan inisialisasi dan terminasi

Adapun teknik pengujian secara *black box* menurut (Sukamto et al., 2015),

adalah sebagai berikut:

1. *Boundary value Analysis (BVA) / Limit Testing* adalah banyak kesalahan terjadi pada kesalahan masukan. BVA mengizinkan untuk menguji seleksi kasus uji yang menguji batasan nilai input. BVA merupakan komplemen dari *equivalence partitioning*. Lebih pada memilih elemen-elemen didalam kelas ekivalen pada bagian sisi batas dari kelas.
2. *Comparison Testing* adalah pengujian pada beberapa aplikasi *reliability* dari sebuah perangkat lunak. Redundansi perangkat keras dan perangkat lunak mungkin digunakan untuk meminimalisir kesalahan (*error*). Untuk redundansi perangkat lunak, gunakan tim yang terpisah untuk mengembangkan setiap versi perangkat lunak yang independen. Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama. Jalankan semua versi dengan paralel dan perbandingan keluaran secara *real-time*.
3. *Sample Testing* adalah pengujian yang melibatkan beberapa nilai yang terpilih dari sebuah kelas ekuivalen. Mengintegrasikan nilai pada kasus uji dan nilai-nilai yang terpilih mungkin dipilih dengan urutan tertentu atau interval tertentu
4. *Robustness Testing* adalah pengujian dengan data input dipilih diluar spesifikasi yang telah didefinisikan. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid
5. *Behavior Testing* adalah pengujian dengan hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur *data stack*.
6. *Requirement Testing* adalah spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (*input, output, fungsi, performansi*) diidentifikasi pada tahap spesifikasi kebutuhan dan desain. *requirement testing* melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program.
7. *Performance Testing* adalah mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi dan lain-lain. Untuk mencari tahu beban kerja atau kondisi konfigurasi program dan dapat digunakan untuk menguji batasan lingkup program.
8. Uji ketahanan (*Endurance Testing*) adalah melibatkan kasus uji yang diulang-

ulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai spesifikasi kebutuhan.

9. *Equivalence partitioning* adalah membagi input menjadi kelas-kelas data yang dapat digunakan untuk meregenerasi kasus uji dengan tujuan untuk menemukan kelas-kelas kesalahan. Selain itu, *equivalence partitioning* berdasarkan pada kesamaan kelas-kelas kondisi *input*. Sebuah kelas yang ekuivalen merepresentasikan kumpulan status/kondisi yang valid atau tidak valid. Sebuah kondisi input dapat berupa nilai numerik yang spesifik, rentan nilai, kumpulan nilai yang berkaitan, atau kondisi *boolean*.
10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*) adalah teknik yang merupakan suplemen dari *equivalence testing* dengan menyediakan cara untuk memilih kombinasi data input dan melibatkan kondisi *input* (*Cause*) dan kondisi *output* (*Effect*) untuk mencegah pendefinisian kasus uji yang terlalu banyak.

2.13.2 Pengujian Skala Likert

Perhitungan hasil dari kuesioner dilakukan dengan menggunakan cara *Skala Likert*. *Skala Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Sugiyono, 2014). Untuk setiap pilihan jawaban diberi skor, maka responden harus menggambarkan, mendukung pertanyaan dengan jawaban yang dipilih. Dengan *skala likert*, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak ukur menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan.

Tabel 2.7 Skala Penilaian Untuk Pertanyaan Positif dan Negatif

Nilai	Kriteria
1	Sangat Buruk
2	Buruk
3	Cukup
4	Baik
5	Sangat Baik