

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Studi

Dalam penelitian ini digunakan tinjauan studi yang dapat mendukung penelitian, berikut ini merupakan tinjauan studi yang diambil:

Tabel 2.1 Daftar Tabel Literatur

No Literatur	Penulis	Tahun	Judul
Literatur 01	Nova Agustina, Slamet Risnanto	2016	Aplikasi <i>Location Based Service</i> untuk informasi dan pencarian lokasi pariwisata di kota Cimahi berbasis <i>android</i>
Literatur 02	Salman Alfarisi Hasbullah	2017	Aplikasi Pencari Tempat Pencucian Mobil Dan Motor Di Kota Malang Menggunakan Metode <i>Location Based Service</i> (LBS) Berbasis <i>Android</i>
Literatur 03	D. S. Ramdan, Agi Priawan	2017	Penerapan <i>Location Based Service</i> dan <i>QR-Code</i> Dalam Pemetaan Lokasi Berbasis <i>Android</i>
Literatur 04	Erviana Devie, Edy Winarno	2018	Aplikasi <i>Location Based Service</i> Untuk Informasi Kuliner Di Yogyakarta
Literatur 05	Ariefka Rahman	2018	Aplikasi Mad Ento “Madura <i>Explore And Tour</i> ” Dengan Penerapan Metode <i>Location Based Service</i> (LBS) Berbasis <i>Android</i>
Literatur 06	Hanafi Eko Kurniawan, Erma Susanti, Suwanto Raharjo	2018	Pengembangan Aplikasi <i>Location Based Service</i> Cafe Menggunakan Metode Algoritma Dijkstra DAN <i>Haversine</i> Di Wilayah Yogyakarta Berbasis <i>Android</i>
Literatur 07	I Made Prasatya Mertha, Vincent Simadiputra, Eko Setyawan, Suharjito	2019	Implementasi WebGIS untuk Pemetaan Objek Wisata Kota Jakarta Barat dengan Metode <i>Location Based Service</i> menggunakan <i>Google Maps API</i>
Literatur 08	Algi Muhkamaat, Rianto, Neng Ika Kurniati	2019	Penerapan Metode <i>Haversine</i> pada Aplikasi Layanan Perbaikan Kendaraan Berbasis <i>Location Based Service</i>
Literatur 09	Nur Aeni Widiastuti, Teguh Tamrin	2020	Penerapan Aplikasi <i>Mobile Location Based Service</i> Untuk Persebaran Usaha Mikro Kecil

			Menengah Dikabupaten Jepara
Literatur 10	Risnu Arya Kusuma, Yus Sholva, Rudy Dwi Nyoto	2020	Aplikasi Peringatan Rambu Lalu Lintas dengan Metode <i>Location Based Service</i> Berbasis <i>Mobile</i>

2.1.1 Tinjauan Literatur 01

Pariwisata adalah salah satu sektor yang dapat diandalkan sebagai sumber devisa negara. Dalam perjalanan untuk mencari lokasi terdekat ke lokasi pariwisata yang diinginkan, untuk mencari lokasi pariwisata yang ada di Kota Cimahi melalui perangkat *mobile* menjadi sangat penting mengingat wisatawan yang sulit untuk menemukan lokasi pariwisata. *Location Based Service* (LBS) merupakan salah satu layanan informasi yang memanfaatkan kemampuan penunjuk lokasi pada perangkat *mobile* dan dapat diakses melalui jaringan *mobile*. Tujuan dari penulisan tugas akhir ini adalah untuk menghasilkan suatu aplikasi *Location Based Service* pada perangkat *mobile Android* yang mampu membantu untuk mencari informasi dan lokasi pariwisata dari lokasi *user* berikut rute yang dapat ditempuh menuju lokasi pariwisata tersebut. Berdasarkan hasil dari implementasi dan ujicoba yang dilakukan maka peneliti menarik kesimpulan bahwa aplikasi *Location Based Service* yang dibuat dapat membantu *user* untuk mencari informasi, lokasi pariwisata dan juga menentukan rute yang dapat ditempuh menuju lokasi pariwisata tersebut.

2.1.2 Tinjauan Literatur 02

Aplikasi Pencari Tempat Pencucian Mobil dan Motor ini adalah aplikasi yang dapat membantu menghadirkan informasi mengenai Tempat Pencucian Mobil dan Motor yang ada di Kota Malang. Dapat digunakan dalam Sistem Operasi *Android* dan *Global Positioning System* (GPS). Untuk dapat

menggunakan aplikasi ini pengguna harus terhubung ke internet karena membutuhkan data yang di ambil dari *Google Maps*. Fasilitas utama yang terdapat dalam aplikasi ini adalah adanya fitur *Map* dan *Rute* yang mempermudah pengguna dalam menemukan lokasi yang dituju. Dengan kemajuan teknologi dan hadirnya *smartphone*, kebutuhan orang akan navigasi dan penunjuk arah sudah tergantikan oleh teknologi LBS. *Location Based Service* (LBS) adalah layanan berbasis lokasi atau dengan kata lain untuk menemukan lokasi suatu tempat yang dicari seperti rumah sakit, pom bensin dan fasilitas publik lainnya. Layanan ini menggunakan teknologi GPS yang merupakan sistem navigasi berbasis satelit. Umumnya teknologi seperti ini sudah disematkan pada tiap *smartphone* khususnya *smartphone* yang berbasis *android*.

2.1.3 Tinjauan Literatur 03

Salah satu hal yang paling penting dalam pelaksanaan sebuah acara atau kegiatan adalah lokasi. Oleh karena itu, pencantuman alamat atau lokasi dalam sebuah kegiatan atau acara merupakan hal yang wajib dan harus dilakukan dengan tepat dan akurat, sehingga para tamu undangan atau para peserta dapat mencapai lokasi dengan efektif dan efisien. Ada beberapa cara yang dilakukan untuk melakukan pencantuman sebuah lokasi, beberapa diantaranya adalah dengan menggunakan *text* dan gambar denah lokasi yang ditulis dan digambar pada kartu undangan, poster dan media promosi lainnya. Penggunaan *text* dan gambar denah lokasi memiliki beberapa keterbatasan, diantaranya adalah penggambaran denah lokasi yang tidak akurat karena keterbatasan lebar denah lokasi yang bisa digambar dan tidak bisa memberikan navigasi atau rute menuju lokasi tersebut, sehingga dapat menyebabkan kesasar. Beberapa masalah tersebut dapat diselesaikan dengan

pendekatan aplikasi digital berbasis *Android*. Saat ini *Android* menjadi OS *mobile* yang populer dan banyak digunakan oleh kebanyakan masyarakat. *Android* juga sudah mendukung *Location Based Service* (LBS) dan *Global Positioning System* (GPS), yang bisa digunakan untuk melakukan pemetaan lokasi, membuat navigasi dan rute untuk mencapai sebuah lokasi. Untuk menyederhanakan pencantuman alamat lokasi pada media promosi, dapat menggunakan *QR-Code* yang dapat discan atau dibaca oleh sebuah aplikasi digital. *User* hanya perlu men-scan *QR-Code* yang tercantum dalam media promosi, kemudian aplikasi akan menerjemahkan *QR-Code* tersebut menjadi sebuah peta lokasi dan membuka navigasi untuk memandu user mencapai lokasi tersebut.

2.1.4 Tinjauan Literatur 04

Aplikasi *Location Based Service* (LBS) banyak dikembangkan karena mempunyai kemanfaatan yang sangat penting untuk berbagai hal, tak terkecuali untuk informasi kuliner. Dalam penelitian ini dibuat sebuah aplikasi kuliner dengan menggunakan metode *Location Based Service* untuk mendapatkan suatu titik lokasi pada aplikasi, aplikasi yang dirancang untuk mempermudah pengguna mengunjungi berbagai macam kuliner saat berkunjung ke Yogyakarta dengan menampilkan estimasi jarak dari posisi awal pengguna ke tempat tujuan. Penelitian ini memanfaatkan *Global Positioning System* pada perangkat android. Pada perancangan ini digunakan rumus *Haversine* untuk menghitung jarak tempuh pengguna dari posisi awal menuju lokasi tujuan secara manual. Hasil dari pengujian yang dilakukan pada salah satu data menunjukkan selisih dari jarak tempuh sebanyak ± 40 km.

2.1.5 Tinjauan Literatur 05

Pulau Madura merupakan sebuah pulau yang terdiri dari 4 Kabupaten yaitu Kabupaten Bangkalan, Sampang, Pamekasan, dan Sumenep. Jika kita berbicara mengenai sektor pariwisata di Pulau Madura, tidak bisa dipungkiri bahwasanya banyak sekali objek wisata yang sangat bagus dan menarik di Pulau Madura namun kurang terekspose dengan baik. Selama ini sistem promosi pariwisata masih menggunakan cara yang bisa dibilang konvensional yang mana masih menggunakan brosur dan media cetak. Metode promosi tersebut memanglah kurang efektif mengingat kurangnya informasi yang dapat diimplementasikan dalam media tersebut. Oleh karena itu lahirlah inovasi baru berupa Aplikasi Mad Ento “Madura *Explore And Tour*” dengan penerapan metode *Location Based Service* (LBS) berbasis Android dengan harapan mampu membantu sistem promosi pariwisata di Pulau Madura. Aplikasi Mad Ento adalah salah satu aplikasi yang menggunakan sistem operasi *android* yang bertujuan untuk memudahkan pengguna yang dalam hal ini sasarannya ialah wisatawan, untuk dapat mempermudah perjalanan wisatanya di Pulau Madura. Aplikasi Mad Ento diharapkan dapat memfasilitasi wisatawan akan informasi tempat wisata, pusat oleh-oleh, penginapan, kuliner dan *event*. Aplikasi ini menggunakan metode *Location Based Service* (LBS) untuk membantu wisatawan memperoleh lokasi wisata, pusat oleh-oleh, penginapan dan kuliner yang ingin dituju.

2.1.6 Tinjauan Literatur 06

Keberadaan cafe di kota metropolitan mempunyai peran sebagai tempat untuk bersantai dan berkumpul, cafe juga di butuhkan oleh masyarakat kota itu sendiri maupun para pendatang. Pada penelitian ini membahas penggunaan

aplikasi sebagai media untuk mencari informasi cafe yang dipadu dengan menggunakan aplikasi mobile berbasis android. Metode yang digunakan adalah Location Based Service (LBS) menggunakan algoritma Dijkstra dan Haversine sesuai dengan lokasi pengguna berada untuk mendapatkan lokasi maupun rute menuju lokasi cafe yang dituju secara cepat dan mudah. Aplikasi pencarian cafe ini memiliki fitur utama yakni mampu mencari lokasi cafe yang terdekat sesuai dengan lokasi pengguna saat itu dengan memanfaatkan teknologi GPS (Global Positioning System). Dengan adanya aplikasi ini sehingga dapat mempermudah pengguna atau masyarakat umum yang ingin menuju ke lokasi cafe tanpa menemui kendala.

2.1.7 Tinjauan Literatur 07

Jakarta Barat adalah salah satu kota di wilayah provinsi DKI Jakarta yang mempunyai potensi wisata yang besar terutama pada wisata budaya dan sejarah. Dimana di kota tersebut terdapat banyak bangunan peninggalan sejarah pada masa kolonial Belanda. Tetapi pada kenyataannya informasi mengenai hal tersebut masih terbatas terutama bagi wisatawan mancanegara yang lebih mengenal Bali sebagai Indonesia. Salah satu cara mengenalkan wisata tersebut adalah dengan memanfaatkan teknologi webGIS sebagaimana tujuan dari penelitian ini adalah menghasilkan sebuah webGIS wisata kota Jakarta Barat yang bisa diakses melalui media internet. Dengan webGIS dapat ditampilkan wisata kota Jakarta Barat yang dilengkapi dengan titik lokasi peta serta rute perjalanannya dan bisa diakses melalui *browser* internet tanpa bantuan *software* pengolah GIS. Wisata yang ditampilkan adalah wisata budaya dan sejarah, wisata kuliner, wisata belanja, wisata keluarga dan wisata religi. Pembuatan webGIS menggunakan metode

Location Based Service (LBS) dengan memanfaatkan teknologi *Google Maps API* dan *Framework CodeIgniter*. Dengan LBS, koordinat lokasi wisata berupa *latitude* dan *longitude* yang berasal dari aplikasi *Google Maps* bisa dengan mudah dimasukkan dalam struktur kode webGIS.

2.1.8 Tinjauan Literatur 08

Kendaraan merupakan alat transportasi yang membantu masyarakat dalam melakukan aktivitas di kehidupan sehari-hari, tetapi selain manfaat menggunakan kendaraan bermotor, pada umumnya masyarakat juga dihadapi dengan resiko saat menggunakan kendaraan tersebut yaitu jika terjadi kerusakan pada mesin secara tiba-tiba atau biasa disebut dengan istilah mogok. Media informasi yang dapat diterapkan pada kasus kendaraan rusak secara mendadak tersebut adalah dengan membangun sebuah aplikasi *android* untuk memberikan informasi mengenai bengkel kendaraan yang berada di sekitar lokasi kendaraan pengguna yang mengalami kerusakan menggunakan *Location Based Service*. Tujuan dari penelitian ini adalah membangun sebuah aplikasi dengan menerapkan metode *Haversine* untuk menemukan bengkel terdekat pada aplikasi layanan perbaikan kendaraan. *Formula Haversine* membutuhkan masukan *latitude* dan *longitude* lokasi pengguna dan lokasi bengkel sebagai masukan untuk menemukan bengkel terdekat. Penelitian ini didukung dengan metodologi pengembangan perangkat lunak yaitu metode *Extreme Programming* dalam melakukan analisis dan desain perancangan terhadap aplikasi yang dibangun. Hasil penelitian ini yaitu untuk membantu pengguna aplikasi yang ingin mencari informasi bengkel kendaraan terdekat dengan pencarian jarak terdekat berdasarkan lokasi pengguna saat ini ke lokasi bengkel.

2.1.9 Tinjauan Literatur 09

Kabupaten Jepara memiliki sentra industri yang tersebar di wilayahnya. Beberapa sentra industri diantaranya sentra kerajinan gerabah dan genteng, sentra industri tenun troso, sentra industri kerajinan monel, sentra industri mebel, dan lainnya. Persebaran sentra yang ada menyebabkan sentra industri yang lokasinya strategis lebih mudah dikunjungi daripada pengrajin yang lokasinya kurang strategis. Oleh sebab itu, untuk mengatasi permasalahan tersebut dan demi keberlangsungan sentra yang ada maka peneliti mengusulkan untuk perancangan aplikasi *mobile* persebaran Usaha Mikro Kecil Menengah dengan pengembangan metode *location based service*. Aplikasi *mobile* ini terintegrasi dengan *google Maps* API sehingga pembeli dapat mengakses jarak lokasi sentra yang dicari dengan lokasi dia berada saat itu. Metode ini memiliki kelebihan diantaranya: informasi lokasi yang diberikan *real time*, sebagai penunjuk arah untuk lokasi yang dituju. Hasil penelitian ini dapat dilihat dari pengujian *blackbox testing* dan uji kelayakan aplikasi kepada pengguna menghasilkan *persentase* sebesar 77% yang dinyatakan sangat layak untuk digunakan.

2.1.10 Tinjauan Literatur 10

Rambu lalu lintas adalah bagian dari perlengkapan jalan yang memuat lambang, huruf, angka, kalimat dan/atau perpaduan di antaranya, dan berfungsi untuk menginformasikan peraturan dan perintah yang ada dalam jalan tersebut. Dalam penerapannya, banyak sekali pengendara yang melanggar rambu yang dikarenakan ketidaktahuan terhadap rambu tersebut. Faktor yang menyebabkan ketidaktahuan pengendara terhadap rambu diantaranya adalah rambu tidak terlihat oleh pengendara yang dikarenakan posisi rambu yang tidak terjangkau pandangan

dan kondisi plang rambu yang kurang baik serta ketidaktahuan pengemudi terhadap arti dari rambu tersebut. Dengan permasalahan tersebut, diperlukan aplikasi yang dapat memberi tahu pengemudi mengenai informasi rambu yang berada disekitarnya ketika sedang berkendara. Aplikasi ini memberikan peringatan rambu lalu lintas dengan metode location based service untuk menyesuaikan data rambu dengan lokasi pengemudi, yang disampaikan dalam bentuk tampilan dan suara agar tidak mengganggu pengemudi dalam mengemudikan kendaraan serta penilaian perilaku pengemudi terhadap rambu yang dilanggar. Penilaian perilaku terhadap rambu yang berlaku dapat mendorong pengemudi agar lebih mematuhi rambu. Hasil dari penelitian ini menghasilkan aplikasi yang dapat digunakan sebagai pemandu rambu ketika sedang berkendara. Pengujian sistem mendapatkan hasil cukup baik pada pengujian menggunakan metode pengujian BlackBox dengan memiliki beberapa batasan tertentu. Dengan adanya aplikasi ini, diharapkan pengemudi dapat mematuhi rambu lalu lintas dan meminimalisir adanya pelanggaran serta kecelakaan lalu lintas.

Berdasarkan tinjauan studi literatur yang dipaparkan diatas maka penulis menjelaskan bahwa yang menjadi perbedaan pada penelitian sebelumnya dan penelitian yang akan diusulkan ialah perbedaan objek penelitian dan manfaat tambahan selain mencari lokasi juga memberikan informasi mengenai bagi pengguna atau masyarakat yang membutuhkan jasa *service* dan renovasi bangunan dengan mudah, cepat dan terpercaya di kota Bandalampung.

2.2 *Location Based Service (LBS)*

Menurut Erviana & Winarno (2018) metode *Location Based Service* merupakan salah satu metode layanan berbasis lokasi untuk mengetahui rute perjalanan, yang memberikan hasil dengan tingkat keakuratan yang cukup tinggi serta dapat memberikan informasi lokasi. Layanan ini menggunakan teknologi *Global Positioning Service (GPS)* dan *cell-based location* dari Google. Selain itu, LBS tersebut terdiri dari beberapa komponen di antaranya *mobile devices* berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk meminta informasi, *communication network* berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari piranti *mobile*-nya untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna, *position component*, dan *service and content provider* penyedia layanan yang menyediakan layanan berbeda ke pengguna seperti pencarian rute, kalkulasi posisi, dan lainnya. Sederhananya, dengan layanan LBS kita dapat mengetahui posisi dimana kita berada. Langkah awal yang dilakukan oleh metode LBS dalam pencarian lokasi yaitu mengaktifkan fungsi pencarian dimana posisi pengguna yang ada diponsel diperoleh dari *positioning service* pada GPS. Setelah itu perangkat ponsel akan mengirim informasi yang berisi tujuan untuk mencari dan mengirimkan posisi melalui jaringan komunikasi ke *gateway*. Kemudian, *service* menganalisis pesan dan memutuskan informasi tambahan pencarian dan posisi pengguna dan selanjutnya *service* akan menemukan bahwa informasi mengenai jalan, jarak dan memeriksa apakah tujuan tersebut dapat dicapai dan akan ditampilkan kepada pengguna dalam bentuk peta.

2.3 Unified Modeling Language (UML)

Menurut Rosa dan Shalahuddin (2014) berpendapat bahwa UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requerement*, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. Sedangkan Mulyani (2016) mengatakan UML (*Unified Modeling Language*) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek.

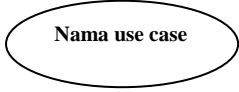
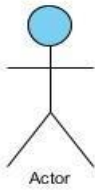

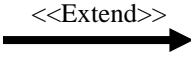
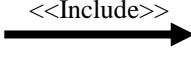

UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya (Rosa dan Shalahuddin, 2014):

2.3.1 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa dan Shalahuddin, 2014).

Berikut adalah simbol-simbol yang ada pada diagram *use case* (Rosa dan Shalahuddin, 2014):

Tabel 2.2 Simbol-simbol *Use Case Diagram*

No.	Notasi	Keterangan	Simbol
1.	<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor	
2.	Aktor / Actor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang	
3.	Asosiasi / Association	Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor	
4	Ekstensi / Extend	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.	
5	Include	<i>Use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.	
6	Generalisasi / Generalization	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.	

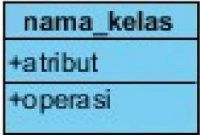






Sumber: (Rosa dan Shalahuddin, 2014)

2.3.2 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa dan Shalahuddin, 2014).

Berikut adalah simbol-simbol yang ada pada diagram *Class Diagram* (Rosa dan Shalahuddin, 2014):

Tabel 2.3 Simbol *Class Diagram*

No	Notasi	Keterangan	Simbol
1	Class / Kelas	Kelas pada struktur sistem	
2	Interface / Antarmuka	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek	
3	Association / Asosiasi	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>	
4	Directed association / Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>	
5	Generalisasi	Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus)	
6	Dependency / Kebergantungan	Relasi antar kelas dengan makna ketergantungan antar kelas	
7	Aggregation / Agresi	Relasi antar kelas dengan makna semua – bagian (<i>whole-part</i>)	

Sumber: (Rosa dan Shalahuddin, 2014)






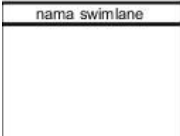
2.3.3 Activity Diagram

Activity Diagram ini menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan aktor, jadi aktifitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2014).

Berikut adalah simbol-simbol yang ada pada diagram *Activity Diagram* (Rosa dan Shalahuddin, 2014):

Tabel 2.4 Simbol *Activity Diagram*

No.	Notasi	Keterangan	Simbol
1.	Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal	
2.	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja	
3.	<i>Decision</i> / Percabangan	Asosiasi percabangan Dimana jika ada pilihan aktifitas lebih dari satu	
4.	<i>Join</i> / Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu	
5.	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir	
6.	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi	

Sumber: (Rosa dan Shalahuddin, 2014)







2.3.4 *Sequence Diagram*


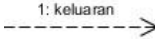
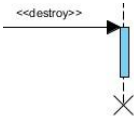
Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima

antar objek. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa dan Shalahuddin, 2014).

Berikut adalah simbol-simbol yang ada pada diagram *Sequence Diagram* (Rosa dan Shalahuddin, 2014):

Tabel 2.5 Simbol *Sequence Diagram*

No.	Notasi	Keterangan	Simbol
1.	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan system informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama actor	
2.	Lifeline / Garis hidup	Menyatakan kehidupan suatu objek	
3.	Objek	Menyatakan objek yang berinteraksi pesan	
4.	Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.	
5.	Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat	
6.	Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek	

No.	Notasi	Keterangan	Simbol
		yang memiliki operasi / metode, karena ini memanggil operasi / metode, maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi	
7.	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim	
8.	Pesan tipe <i>return</i>	Menyatakan suatu objek yang menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian	
9.	Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>	

Sumber: (Rosa dan Shalahuddin, 2014)

2.4 Global Positioning Sistem (GPS)

Global Positioning System (GPS) menurut Greg Millete dan Adam Stroud (2012) adalah sistem satelit navigasi dan penentuan posisi yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga-dimensi serta informasi mengenai waktu, secara terus menerus di seluruh dunia tanpa bergantung waktu dan cuaca. Bagian yang paling penting dalam sistem navigasi GPS adalah satelit yang berada di orbit bumi yang semuanya dapat memancarkan sinyal ke bumi yang kemudian dapat ditangkap oleh alat penerima sinyal tersebut atau *GPS Receiver* (Sumitro et al., 2017). Selain satelit terdapat 2 sistem lain yang saling berhubungan, sehingga jadilah 3 bagian

penting dalam sistem GPS. Ketiga bagian tersebut terdiri dari: *GPS Control Segment* (Bagian Kontrol), *GPS Space Segment* (bagian angkasa), dan *GPS User Segment* (bagian pengguna). Seperti terlihat dalam Gambar 2.1 dibawah ini adalah cara kerja dari GPS terhadap ketiga satelit tersebut untuk menentukan posisi.



Gambar 2.1 Cara kerja *Global Positioning System* (GPS)
<http://www.navigadget.com/wp-content/postimages/2007/01/a-gps-944.jpg>

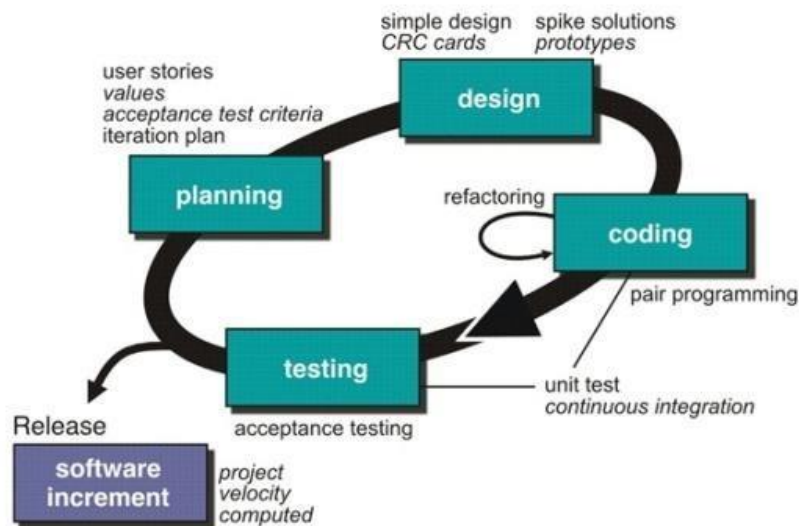
2.5 Metode Pengembangan Sistem

Menurut Fatoni & Dwi (2016) dalam jurnalnya menjelaskan bahwa, Metode *Extreme Programming* sering juga dikenal dengan metode XP. Metode ini dicetuskan oleh Kent Beck, seorang pakar *software engineering*. *Extreme programming* adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, adaptif dan fleksibel. Nilai dasar metode *extreme programming* :

- 1) *Communication*: Memfokuskan komunikasi yang baik antara *programmer* dengan *user* maupun antar *programmer*.

- 2) *Courage*: Pengembang perangkat lunak harus selalu memiliki keyakinan, keberanian dan integritas dalam melakukan tugasnya.
- 3) *Simplicity*: Lakukan semua dengan sederhana.
- 4) *Feedback*: Mengandalkan *feedback* sehingga dibutuhkan anggota tim yang berkualitas.
- 5) *Quality Work* : Proses berkualitas berimplikasi pada perangkat lunak yang berkualitas sebagai hasil akhirnya.

Seperti terlihat dalam Gambar 2.2 dibawah ini adalah kerangka kerja *Extreme Programming* yang mendukung percepatan pembangunan suatu sistem.



Gambar 2.2 Kerangka Kerja *Extreme Programming*
Sumber : (Fatoni & Dwi, 2016)

2.6 Geographic Information Sistem (GIS)

Menurut Manongga, Papilaya & Pandie (2010) dalam jurnalnya menjelaskan bahwa Sistem Informasi Geografis (SIG) merupakan sistem informasi berbasis komputer yang merupakan penggabungan antara unsur peta (geografis) dan

informasi tentang peta tersebut (data atribut), yang dirancang untuk mendapatkan, mengolah, memanipulasi, analisis, memperagakan dan menampilkan data spasial untuk menyelesaikan perencanaan, mengolah dan meneliti permasalahan. SIG pada dasarnya merupakan gabungan dari tiga unsur pokok yaitu: sistem, informasi dan geografis.

2.7 Google Maps API

Google Maps API adalah sebuah antarmuka pemrograman aplikasi yang menyediakan berbagai fungsi yang dapat digunakan untuk menampilkan peta dari *google maps* ke dalam *website* atau perangkat *android*. Tidak hanya menampilkan saja dengan menggunakan antarmuka pemrograman ini pengembangan dapat mengelola serta menambah berbagai grafis pada peta yang telah disediakan sehingga fungsional peta semakin meluas dan dapat dikembangkan sesuai dengan keinginan pengembang, layanan ini diberikan secara gratis oleh *google*. dan sebelum kita mencari informasi lokasi, alamat terlebih dahulu pengguna harus memberikan kata kunci. *google maps* dibuat dengan secara ekstensif dan interaktif sehingga pengguna bisa menggeser peta dan men *zoom* sesuai penggunaan (Ramadhani, Anis & Masruro, 2013)

2.8 Firebase

Menurut Sanad, Achmand & Dewiani (2019) *Firebase* merupakan *platform* untuk aplikasi *realtime*. Ketika data berubah, maka aplikasi yang terhubung dengan *firebase* akan meng-*update* secara langsung melalui setiap *device* (perangkat) baik *website* ataupun *mobile*. *Firebase* mempunyai *library* (pustaka) yang lengkap untuk sebagian besar *platform web* dan *mobile* dan dapat digabungkan dengan berbagai *framework* lain seperti *node*, *java*, *javascript*, dan lain-lain. *Application*

Programming Interface (API) untuk menyimpan dan sinkronisasi data akan disimpan sebagai bit dalam bentuk *JSON (JavaScript Object Notation)* pada *cloud* dan akan disinkronisasi secara *realtime*. *Firebase Realtime Database* merupakan basis data online yang dapat digunakan sebagai media penyimpanan data dari aplikasi. *Database* milik *firebase* merupakan *database* yang bersifat *non-relational* atau *NoSQL*, dimana *database* ini merupakan jenis *database* yang tidak menggunakan sistem tabel dalam implementasinya serta tidak menyimpan data secara lokal pada perangkat melainkan pada awan.

2.9 Android

Menurut Ali (2017) *Android* merupakan sistem operasi *platform mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Sehingga banyak digemari masyarakat karena bersifat *open source*. *Android* juga menyediakan *framework* aplikasi yang memungkinkan *developer* untuk membangun aplikasi inovatif dan *game* untuk perangkat *mobile* dengan menggunakan bahasa pemrograman *Java*. Perkembangan *Android* saat ini memungkinkan *developer* untuk mengembangkan aplikasi *Android* selain menggunakan *Java* (Supriyono, Saputra, Sudarmilah, & Darsono, 2014). Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia* (Berbudhi Rachman H, 2013).

2.10 Android Studio

Menurut Ash-Shidiq, Rumani & Saputra (2017) *Android Studio* merupakan sebuah *Integrated Development Environment (IDE)* untuk *platform Android*.

Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi *Google I/O* oleh Produk Manajer *Google*, Ellie Powers. *Android Studio* bersifat gratis dibawah *Apache License 2.0*. *Android Studio* awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan *JetBrainns' IntelliJ IDEA*, *Android Studio* di desain khusus untuk *Android Development*. Ini sudah bisa diunduh untuk *Windows*, *Mac OS X*, dan *Linux*.

2.11 Pengujian *Black Box*

Proses untuk mengecek apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai standar atau belum. Pengujian sistem dapat menggunakan metode *black box testing* yaitu merupakan pendekatan komplementer dari teknik *white box testing*, karena pengujian *black box testing* mampu mengungkap kesalahan yang lebih luas. *Black box testing* berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program (Yakub, 2012).

Jadi pengujian sistem dapat dilakukan dengan pengecekan *input*, pengecekan *output* dan pengecekan proses.

1. Pengecekan *input*, meliputi kelengkapan item-item input, kemudahan pengoperasian, kemudahan manipulasi data, dan pengendalian kesalahan.
2. Pengecekan proses, dilakukan dengan pengecekan output program.
3. Pengecekan *output*, meliputi pengecekan terhadap format dan bentuk-bentuk laporan.

Kelebihan *Black Box Testing* adalah :

1. Tidak perlu melihat *source code* secara detail.
2. Mendeteksi kesalahan pengetikan / *Typo*.
3. Mendeteksi kesalahan *Design / UserInterface* dari sebuah *software / website*.
4. Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
5. Seorang *Tester* tidak harus *Programmer*.

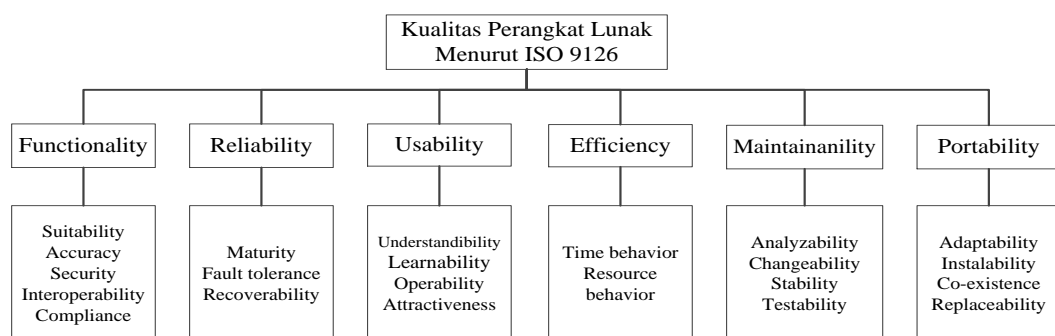
Kekurangan *Black Box Testing* adalah :

1. Ketergantungan dengan dokumen dan design *software* tersebut.
2. Tidak sampai level *code*, sehingga tester tidak mengetahui *levelsecurity* dari *software* tersebut.

2.12 Pengujian Sistem (ISO 9126)

Kualitas perangkat lunak dapat dinilai melalui ukuran-ukuran dan metode-metode tertentu, serta melalui pengujian-pengujian software. Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut (Al-Qutaish, 2010)

1. *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
2. *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
3. *Usability* (Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
4. *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
5. *Maintainability* (Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.
6. *Portability* (Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.



Gambar 2.3 Model Kualitas Perangkat Lunak Model ISO 9126

Sumber: (Al-Qutaish, 2010)

Masing-masing karakteristik kualitas perangkat lunak model ISO 9126 dibagi menjadi beberapa sub-karakteristik kualitas, yaitu:

Tabel 2.6 Karakteristik Kualitas Perangkat Lunak Model ISO 9126

Karakteristik	Sub- karakteristik	Deskripsi
<i>Functionality</i>	<i>Suitability</i>	Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
	<i>Accuracy</i>	Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.
	<i>Security</i>	Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (<i>hacker</i>) maupun otorisasi dalam modifikasi data.
	<i>Interoperability</i>	Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
<i>Reliability</i>	<i>Compliance</i>	Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.
	<i>Maturity</i>	Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.
	<i>Fault tolerance</i>	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak
	<i>Recoverability</i>	Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.
<i>Usability</i>	<i>Understandibility</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipahami.
	<i>Learnability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipelajari.
	<i>Operability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.
	<i>Attractiveness</i>	Kemampuan perangkat lunak dalam menarik pengguna.
<i>Efficiency</i>	<i>Time behavior</i>	Kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya.
	<i>Resource Behavior</i>	Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.

Karakteristik	Sub- karakteristik	Deskripsi
<i>Maintainability</i>	<i>Analyzability</i>	Kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan.
	<i>Changeability</i>	Kemampuan perangkat lunak untuk dimodifikasi tertentu.
	<i>Stability</i>	Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
	<i>Testability</i>	Kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain.
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda.
	<i>Instalability</i>	Kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
	<i>Coexistence</i>	Kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya.
	<i>Replaceability</i>	Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya.

Sumber: (Al-Qutaish, 2010)