

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini digunakan tinjauan Pustaka yang dapat mendukung peneliti.

Berikut ini adalah daftar literatur yang digunakan antara lain :

Tabel 2.1 Tinjauan Literatur

No	Penulis	Judul	Metode	Hasil
1	Dwi Ardana Ragil Saputra 2016	Penerapan Algoritma <i>Dijkstra</i> pada Aplikasi Pencarian Rute Bus Trans Semarang	Algoritma <i>Dijkstra</i>	Aplikasi sebagai sarana yang dapat membantu calon pengguna Bus Trans Semarang untuk dapat mengetahui koridor-koridor yang ada pada Kota Semarang dan dapat menentukan pilihan koridor tujuan yang dituju dengan pasti.
2	Vina Meitasari, Ali Nurdin, Aryanti 2017	Implementasi Metode <i>Dijkstra</i> Dalam Mobile Aplikasi Pencarian SPBU Terdekat Di Kota Palembang	Algoritma <i>Dijkstra</i>	Aplikasi untuk menemukan keberadaan SPBU yang berada disekitar pengemudi dengan algoritma <i>dijkstra</i> .
3	Mahatir Mohamad, Imam Ahmad, Yusra Fernando 2017	Pemetaan Potensi Pariwisata Di Kabupaten Waykanan Menggunakan Algoritma <i>Dijkstra</i>	Algoritma <i>Dijkstra</i>	Hasil yang di peroleh penulisan berhasil memetakan 40 pariwisata dan juga berhasil <i>Dijkstra</i> di tempuh untuk rute terpendek.

Tabel 2.1 Tinjauan Literatur (Lanjutan)

No	Penulis	Judul	Metode	Hasil
4	Eka Ismantohadi, Iryanto 2018	Penerapan Algoritma <i>Dijkstra</i> Untuk Penentuan Jalur Terbaik Evakuasi Tsunami – Studi Kasus: Kelurahan Sanur Bali	Algoritma Dijkstra	Dalam artikel ini, hasil jalur evakuasi terbaik dikelompokkan berdasarkan letak tempat evakuasi dan area aman. Selain itu, perbandingan hasil penelitian ini dengan metode pencarian rute lain dari referensi menunjukkan kesesuaian yang baik.
5	Elda Candra Galih, Rio Andriyat Krisdiawan 2018	Implementasi algoritma <i>Dijkstra</i> pada aplikasi wisata Kuningan berbasis android	Algoritma Dijkstra	penulis menggunakan algoritma <i>Dijkstra</i> untuk menentukan rute terpendek dalam mencari rute dari lokasi awal ke lokasi yang diinginkan dari objek pariwisata
6	Hariska Paunsyah, Husni Mubarak, Rahmi Nur Shofa 2019	Penentuan Jalur Terpendek Menggunakan Google Maps Api Pada Sistem Informasi Geografis (Gis) Panti Sosial Di Kota Tasikmalaya	Algoritma Dijkstra	Pembangunan SIG Panti Sosial menggunakan pendekatan Object <i>Oriented Programming</i> dengan metode <i>Extreme Programing</i> .
7	Fenomena Zulfiqar Andromeda, Theresia Dwiati Wismarini 2020	Penerapan Algoritma <i>Dijkstra</i> Pada Aplikasi Sistem Informasi Geografis Di PT Tunas Artha Gardatama Semarang	Algoritma Dijkstra	Aplikasi yang bertujuan untuk menerapkan sebuah algoritma untuk mencari jalur terpendek dari lokasi awal menuju lokasi ATM-ATM tujuan.

2.1.1 Tinjauan Literatur 01

Kemajuan transportasi kini menjadi pendukung aktifitas utama dalam kegiatan manusia. Berbagai alat transportasi kini banyak disediakan, terlebih transportasi darat yang menggunakan jalan raya sebagai media transportasi untuk mencapai tujuan. Seiring berkembangnya alat transportasi maka jalan yang dibutuhkan akan semakin banyak. Kota Semarang sebagai salah satu kota metropolitan yang memiliki pengguna transportasi dalam jumlah besar sehingga menjadi salah satu kota padat di Indonesia. Pemerintah Kota Semarang memberikan fasilitas Bus Trans Semarang untuk para penggunanya sebagai upaya penanggulangan kemacetan yang terjadi pada Kota Semarang. Berdasarkan permasalahan tersebut, maka pada penelitian ini akan dibangun aplikasi pencarian rute Bus Trans Semarang, dan menerapkan metode *Dijkstra* pada perpindahan jalur pada transferpoint. Tujuan yang ingin dicapai dalam penelitian ini adalah aplikasi yang mampu memberikan informasi rute dan penjadwalan Bus Trans Semarang. Sedangkan manfaatnya antara lain sebagai sarana yang dapat membantu calon pengguna Bus Trans Semarang untuk dapat mengetahui koridor-koridor yang ada pada Kota Semarang dan dapat menentukan pilihan koridor tujuan yang dituju dengan pasti. Serta mampu memberi informasi titik-titik koridor Bus Trans Semarang yang sesuai koridor yang tersedia.

2.1.2 Tinjauan Literatur 02

Kota Palembang merupakan Ibu Kota Provinsi Sumatera Selatan. Merupakan Kota terbesar kedua yang ada di Pulau Sumatera setelah Medan. Peningkatan jumlah kepadatan penduduk di Kota Palembang semakin

mempengaruhi pula jumlah pengguna transportasi umum maupun pribadi. Dengan diiringi kemajuan teknologi yang dikembangkan untuk mempermudah pekerjaan maupun dalam perolehan informasi. Teknologi sudah begitu canggihnya. Komunikasi, kini telah menjadi suatu kemudahan yang memungkinkan setiap orang dapat berhubungan. Keberadaan SPBU sangat penting bagi para pengguna kendaraan bermotor baik untuk kendaraan umum maupun untuk transportasi umum. Stasiun pengisian bahan bakar untuk umum (SPBU) merupakan prasarana umum yang disediakan distributor bahan bakar minyak (BBM). Sistem yang bekerja pada aplikasi ini diintegrasikan dengan solusi perhitungan Algoritma *Dijkstra* pada pemecahan pencarian lintasan terdekatnya. Prinsip algoritma *Dijkstra* adalah dengan pencarian dua lintasan yang paling kecil. Algoritma *Dijkstra* memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Perancangan ini dibuat dapat beroperasi pada perangkat selular berbasis android. Dimana perangkat android bersifat *open source*. Sebagai konsekuensinya, siapa pun boleh memanfaatkannya dengan gratis, termasuk dalam hal kode sumber yang digunakan untuk menyusun sistem operasi tersebut. Perangkat selular berplatform *android* saat ini banyak digunakan pada masyarakat pada umumnya. Penggunaan aplikasi android juga dapat dengan mudah diakses oleh masyarakat luas. Dengan aplikasi ini diharapkan dapat memberikan keefektifan waktu dalam perjalanan saat akan mencari SPBU terdekat.

2.1.3 Tinjauan Literatur 03

Kabupaten Waykanan mempunyai potensi-potensi wisata yang unik dan menarik. Mulai dari wisata budaya sampai wisata alam yang tidak kalah indahnnya dengan Kabupaten yang lain di Provinsi Lampung. Dari sekian potensi wisata yang

ada, wisata alam menjadi favorit dari para wisatawan luar daerah maupun dalam daerah. Sebagian besar lokasi wisata yang ada di Kabupaten Waykanan berada di daerah pelosok dan jauh dari pusat kota. Tujuan penelitian ini adalah untuk merancang dan membangun aplikasi pemetaan potensi pariwisata di kabupaten Waykanan dengan menggunakan algoritma *Dijkstra* untuk mempermudah wisatawan yang ingin berkunjung dengan menghemat waktu dan biaya. Teknik pengumpulan data yang dilakukan pada penelitian ini yaitu menggunakan wawancara, observasi, kuisisioner, dan tinjauan pustaka, sumber data/subjek yaitu dinas pariwisata di kabupaten Waykanan dan masyarakat yang berada dekat dengan lokasi wisata tersebut. Hasil yang di peroleh penulisan berhasil memetakan 40 pariwisata dan juga berhasil *Dijkstra* di tempuh untuk rute terpendek.

2.1.4 Tinjauan Literatur 04

Tsunami cukup sering terjadi di Indonesia. Hal ini tidak terlepas dari fakta letak geografis Indonesia yang terletak pada tiga lempeng tektonik utama. Bali termasuk salah satu wilayah yang memiliki potensi tsunami. Mengingat hal tersebut, adanya jalur evakuasi terbaik sebagai sarana mempercepat tindakan untuk menjauhkan warga setempat dari bahaya yang ditimbulkan (evakuasi) menjadi sangat penting. Dalam penelitian ini, dipilih Kelurahan Sanur Denpasar Bali sebagai objek penelitian tersebut. Penelitian ini bertujuan untuk mencari rute/jalur evakuasi terbaik di kelurahan tersebut. Pencarian jalur evakuasi terbaik tersebut dicari dengan algoritma *Dijkstra*. Dalam artikel ini, hasil jalur evakuasi terbaik dikelompokkan berdasarkan letak tempat evakuasi dan area aman. Selain itu,

perbandingan hasil penelitian ini dengan metode pencarian rute lain dari referensi menunjukkan kesesuaian yang baik.

2.1.5 Tinjauan Literatur 05

Kabupaten Kuningan adalah salah satu tujuan wisata yang cukup populer di kalangan wisatawan. Kuningan memiliki banyak objek wisata yang potensial untuk wisata alam, sejarah, jiarah, maupun penelitian. Untuk menemukan destinasi pariwisata, wisatawan biasanya mencari aplikasi mesin pencari atau peta wisata sehingga memakan waktu yang lama. Untuk memudahkan wisatawan dalam mencari destinasi wisata, perlu sebuah aplikasi. Aplikasi yang akan dikembangkan berdasarkan penelitian yang diadakan di Dinas Pemuda Olahraga dan Pariwisata Kabupaten Kuningan. Aplikasi berbasis *mobile* diperlukan dalam menemukan lokasi objek wisata dan menyediakan deskripsi wisata, dan fasilitas yang ada di objek wisata. Untuk mencari rute terdekat, akan ditampilkan dalam bentuk peta. Wisatawan dapat memilih lokasi tempat wisata dan lokasi awal wisatawan, maka sistem akan menampilkan hasil rute yang ada di peta. Dan dalam aplikasi ini juga tersedia layanan untuk mencari lokasi wisatawan menggunakan teknologi GPS. Dan penulis menggunakan algoritma *Dijkstra* untuk menentukan rute terpendek dalam mencari rute dari lokasi awal ke lokasi yang diinginkan dari objek pariwisata. Penyimpanan data dalam aplikasi ini, penulis menggunakan database *MySQL* dan *Apache* sebagai *web server* dalam menyimpan data objek dan data node tujuan dan rute yang bersangkutan dengan mencari lokasi objek wisata. Dan pada pengembangan aplikasi ini berbasis *client server*.

2.1.6 Tinjauan Literatur 06

Kurangnya informasi mengenai lokasi Panti sosial yang ada, menyebabkan donator mengalami kesulitan untuk menyalurkan dana bantuannya. Pembangunan Sistem Informasi Geografis (SIG) pada penelitian ini dimaksudkan untuk memudahkan donator dalam mengetahui serta mendapatkan informasi yang akurat mengenai keberadaan Panti Sosial yang ada di Kota Tasikmalaya. Penerapan Algoritma *Dijkstra* dimaksudkan untuk menghitung jalur terpendek dari tempat donatur ke Panti Sosial yang akan dituju, sehingga memudahkan donatur dalam menyalurkan dana bantuannya. Pembangunan SIG Panti Sosial menggunakan pendekatan *Object Oriented Programming* dengan metode *Extreme Programing*, hal tersebut dikarenakan SIG yang dibangun berdasarkan kebutuhan dari Donatur. SIG Panti Sosial dapat memberikan kemudahan bagi Donatur dalam mencari lokasi Panti Sosial yang ada di Kota Tasikmalaya, serta memberikan informasi jalur terpendek dari lokasi donator ke lokasi Panti Sosial yang akan dituju.

2.1.7 Tinjauan Literatur 07

Pencarian rute terpendek merupakan permasalahan yang sering ditemukan dalam kehidupan sehari-hari. Untuk itu penentuan rute terpendek diperlukan ketepatan dalam pemilihan jalur terpendek ke lokasi tujuan. Kendala yang dihadapi para kurir dari berbagai perusahaan jasa pengiriman barang masih banyak yang belum mengetahui lokasi tujuan para konsumennya. Untuk itu diperlukan sebuah sistem yang dapat membantu kurir dalam hal menentukan lokasi tujuan dan rute mana yang akan dilalui. Sehingga perjalanan yang dilalui menjadi efisien dalam pengantaran barang ketempat tujuan. Berdasarkan beberapa hal tersebut, penulis melakukan penelitian yang bertujuan untuk menerapkan sebuah algoritma untuk

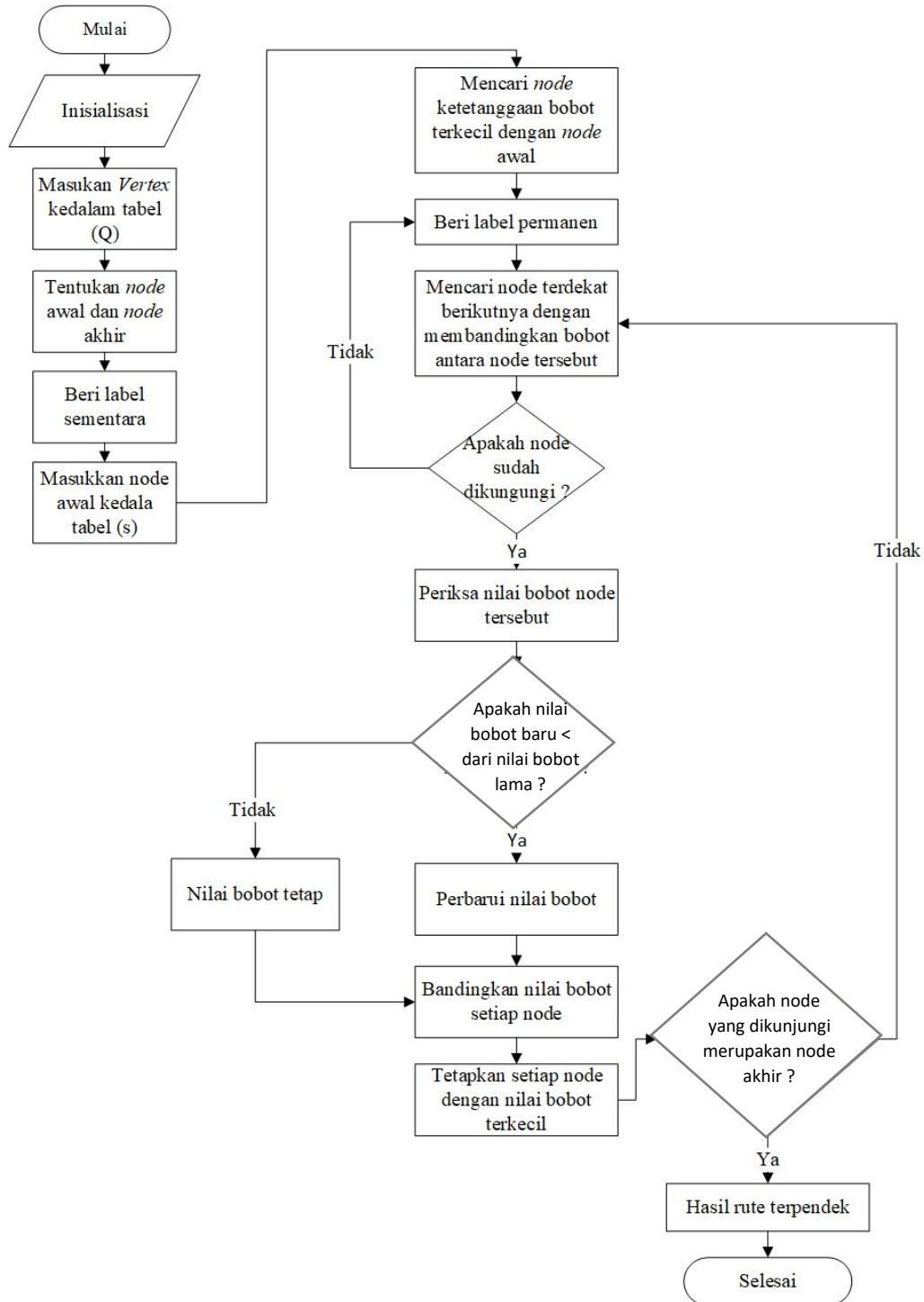
mencari jalur terpendek dari lokasi awal menuju lokasi ATM-ATM tujuan, yang dalam hal ini menggunakan algoritma Dijkstra. Algoritma ini diterapkan pada sebuah aplikasi sistem informasi geografis untuk PT Tunas Artha Gardatama Semarang yang berbasis android.

Berdasarkan tinjauan literatur yang dipaparkan diatas maka penulis menjelaskan perbedaan pada sebelumnya dan penelitian yang akan diusukan ialah perbedaan objek penelitian dimana objek yang akan di gunakan adalah fasilitas Kesehatan di Kabupaten dan dapat memberikan maanfaat kepada masyarakat untuk mengetahui letak fasilitas Kesehatan yang diinginkan.

2.2 Algoritma Dijkstra

Pencarian rute terpendek termasuk kedalam terori graf. Algoritma yang sangat terkenal untuk menyelesaikan persoalan ini adalah algoritma *Dijkstra*. Algoritma ini ditemukan oleh seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra. "*Dijkstra*" diartikan sebagai "algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah" Siswanto (2013). Cara kerja algoritma ini memakai strategi *greedy*, dimana pada setiap langkah yang dipilih sisi yang berbobot terkecil yang menghubungkan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter tempat asal, dan tempat tujuan. Algoritma ini menyelesaikan masalah mencari sebuah lintasan pendek (sebuah lintasan yang mempunyai panjang minimum) dari vertex a ke vertex z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif, namun jika terjadi demikian, maka penyelesaian yang diberikan tidak terbatas Geovani (2016).

Deskripsi matematis untuk graph dapat diwakili $G = \{V, E\}$, yang berarti sebuah grafik (G) didefinisikan oleh satu set simpul (Vertex = V) dan koleksi Edge (E), untuk flowchart algoritma dijkstra dapat dilihat pada Gambar 2.1 berikut :



Gambar 2.1 Flowchart Algoritma Dijkstra

2.3 Pemetaan

Pengertian pemetaan menurut Soekidjo (1994) adalah "Pengelompokkan suatu kumpulan wilayah yang berkaitan dengan beberapa letak geografis wilayah yang meliputi dataran tinggi, pegunungan, sumber daya dan potensi penduduk yang berpengaruh terhadap sosial kultural yang memiliki ciri khas khusus dalam penggunaan skala yang tepat".

2.4 Google Maps Api

Google Maps pertama kali didesain dan dibuat oleh dua programmer Denmark yaitu Lars Rasmussen dan Jeins eilstrup di perusahaan yang bernama *Where 2 Technologies*, kemudian di akuisisi oleh *google* pada tahun 2004. *Google maps* merupakan solusi pemetaan canggih dari *google*, untuk melihat lokasi dan mencari alamat. *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

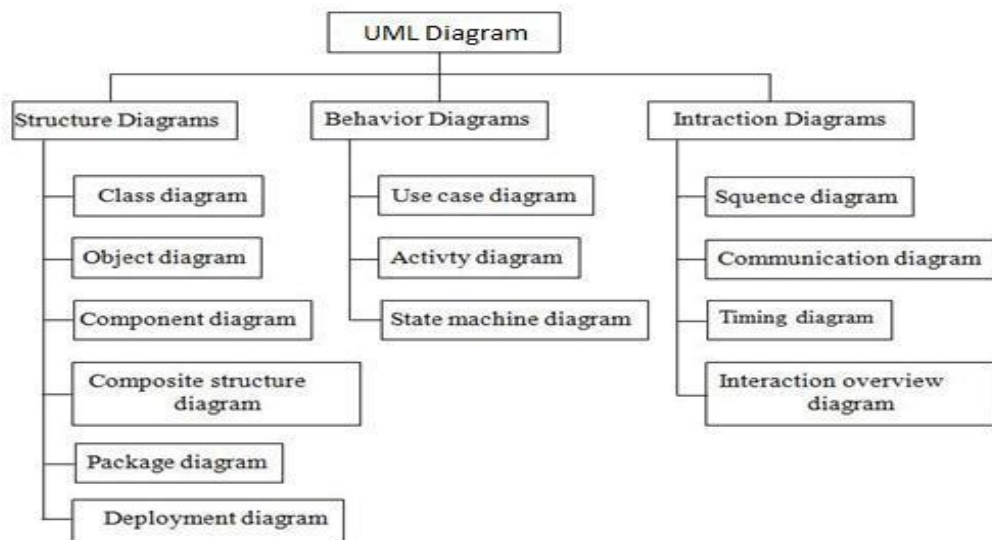
Pada awal sebelum adanya Application Programming Interface yang dikenal sebagai API banyak pengembang menemukan cara untuk dapat menggabungkan *google maps* kedalam situs *web* dan perangkat *android* mereka, oleh karena itu *google* melihat adanya sebuah kebutuhan API umum, kemudian tepatnya di bulan Juni tahun 2005 *google* merilis dan dipublikasikannya *Google Maps Api* untuk mempermudah pengembang dalam menggabungkan kedalam perangkat *android* atau situs *web* Mereka.

2.5 Unified Modeling Language (UML)

Menurut Rosa dan Shalahuddin (2015) UML adalah “salah standar Bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat Analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. Dan menurut Menurut Empiris and Manufaktur (2016) “Unified Modeling Language (UML) adalah bahasa yang telah telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak”. Dari teori-teori tersebut dapat diambil kesimpulan bahwa UML adalah Bahasa yang digunakan untuk membangun sebuah system perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi. UML terdiri dari 13 macam diagram yang dikelompokan dalam tiga kategori. Berikut adalah Pembagian kategori dan macam-macam diagram tersebut.

2.6 Diagram UML

Diagram UML berbentuk grafik yang menunjukkan symbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan again dari suatu *view* tertentu dan Ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain:



Gambar 2.2 Diagram UML

Sumber : Rosa dan Shalahuddin (2014 :140)

Berikut adalah penjelasan singkat dari pembagian katagori tersebut :

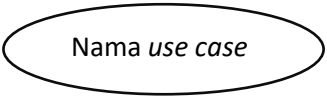
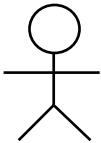
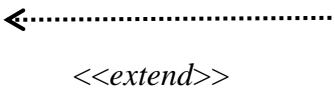
1. *Structure* diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior* diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction* diagrams yaitu diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem ini maupun interaksi antar subsistem pada suatu sistem.

2.6.1 Use Case Diagram


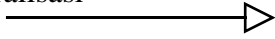

Menurut Rosa dan Shalahuddin (2015) *Use Case* diagram merupakan pemodelan untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara besar, use case digunakan untuk

mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *Use Case* (Rosa dan Shalahuddin, 2015):

Tabel 2.2 Simbol-simbol *Use Case Diagram*

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>Use Case</i></p>
<p>Aktor / <i>actor</i></p> 	<p>Orang, aktor, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>Use Case</i> tambahan kesebuah <i>Use Case</i> Dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan itu;</p>

Tabel 2.2 Simbol-simbol *Use Case Diagram* (Lanjutan)

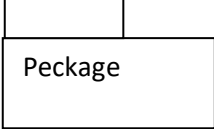






Simbol	Keterangan
<p><i>Include</i></p> 	<p>Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> di mana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini.</p>
<p>Generalisasi</p> 	<p>Relasi antara kelas dengan makna generalisasi-generalisasi (umum khusus)</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan aktor.</p>

Sumber : (Rosa dan Shalahuddin, 2015)


2.6.2 Class Diagram

Menurut Rosa dan Shalahuddin (2015) 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. Berikut adalah simbol-simbol yang ada pada diagram *Class* (Rosa dan Shalahuddin, 2015):

Tabel 2.3 Simbol-simbol class diagram

Simbol	Keterangan
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas.
Operasi 	Kelas pada struktur system
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>Association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (whole-part)
Asosiasi berarah / <i>Directed Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Tabel 2.3 Simbol-simbol class diagram (Lanjutan)



Simbol	Keterangan
Generalisasi 	Relasi antara kelas dengan makna generalisasi-generalisasi (umum khusus)

Sumber : (Rosa dan Shalahuddin, 2015)

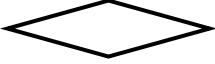


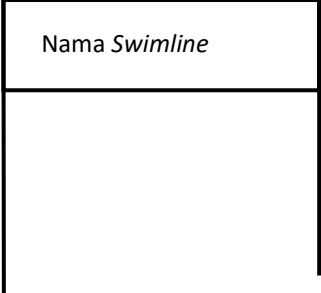
2.6.3 Activity Diagram

Menurut Rosa dan Shalahuddin (2015) *Activity Diagram* adalah diagram aktivitas yang menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram *Activifity* (Rosa dan Shalahuddin, 2015):

Tabel 2.4 *Activity diagram*

Simbol	Keterangan
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja

Tabel 2.4 Activity diagram (Lanjutan)

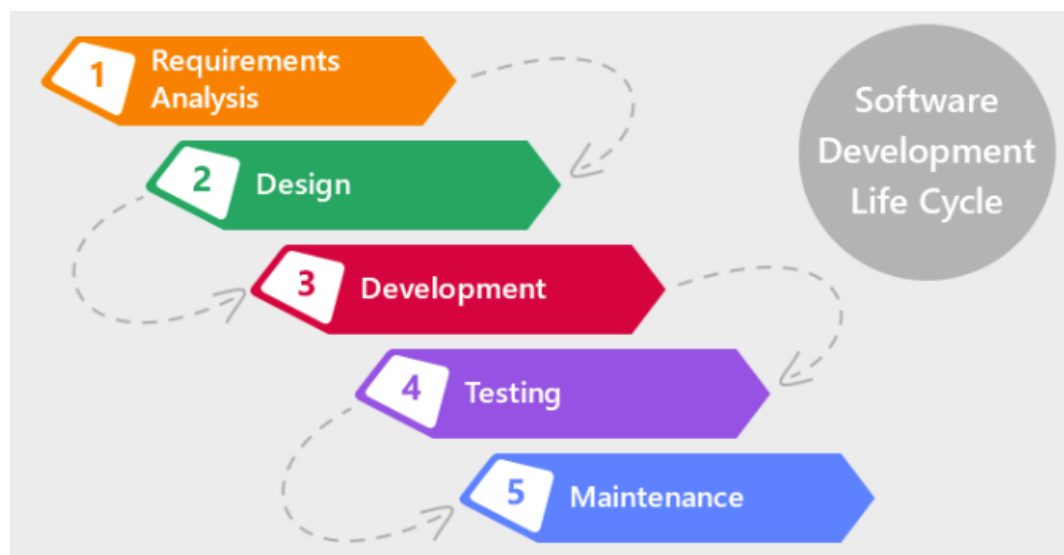
Simbol	Keterangan
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan Diana lebih dari satu aktivitas digambarkan menjadi satu sehingga menjadi saling terhubung.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : (Rosa dan Shalahuddin, 2015)

2.7 Metode Pengembangan Sistem

Menurut Sommerville (2011) model *waterfall* merupakan model yang bersifat sistematis dan sekuensial dalam pengembangan *software*. Nama model ini sebenarnya adalah *Linear sequential model* yang sering dikenal dengan model *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi model ini paling banyak dipakai oleh *software engineering* (SE). Metode ini melakukan pendekatan secara berurutan dan sistematis. Disebut dengan model *waterfall* dikarenakan Dalam metode air terjun setiap tahapan harus diselesaikan terlebih dahulu secara penuh sebelum diteruskan ke tahap berikutnya untuk menghindari terjadinya pengulangan tahapan.

Berikut adalah model air terjun (*waterfall*):



Gambar 2.3 Model Air terjun (*waterfall*)

Tahapan-tahapan utama dari model airterjun ini, secara langsung menggambarkan pokok perkembangan aktivitas-aktivitas berikut ini:

1. *Requirements Analysis*

Pada tahap ini mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh software yang akan dibangun. Hal ini sangat penting, mengingat software harus dapat berinteraksi dengan elemen-elemen yang lain seperti hardware dan database.

2. *Design*

Pada tahap ini, merancang gambaran umum tentang kebutuhan informasi untuk user, selain itu juga tahap ini dirancang bagaimana tampilan system supaya mudah digunakan oleh user.

3. *Implementation*

Setelah design dirasa cukup atau unit program, kemudian dilakukan pengkodean program pada tahap implemetasi ini. Setelah pengkodean selesai dilakukan testing/running tanpa di integritaskan.

4. *Verification*

pada tahap ini, program di uji coba menjadi sebuah sistem yang lengkap, untuk menyakinkan bahwa persyaratan perangkat lunak telah dipenuhi kemudian sistem disampaikan kepada konsumen.

5. *Maintenance*

Sistem di pasang dan digunakan pada tahap ini, kemudian dilakukan pemeliharaan dan perbaikan.

2.8 ISO 25010

Standar ISO/IEC 25010 pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. Dokumen standard ISO/IEC 25010 sangat panjang. Hal ini dikarenakan orang memiliki motivasi berbeda yang memungkinkan untuk tertarik pada kualitas perangkat lunak (Maliki and Wiharja 2014).

Model ini merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, dimana model ini berkaitan dengan model kualitas perangkat lunak yang merupakan pengembangan dari model sebelumnya. Pada model ini terdapat beberapa sub-karakteristik tambahan dan beberapa sub-karakteristik yang dipindahkan ke karakteristik lain. Karakteristik atau faktor kualitas internal dan eksternal yang terdapat pada model ISO-25010 meliputi delapan karakteristik sebagai berikut:

a. *Functional Suitability*

Pengujian functional suitability merupakan tingkat dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi yang spesifik.

b. *Performance efficiency*

Pengujian performance ini dilakukan untuk mengukur karakteristik performa dari komponen aplikasi. Aspek ini merupakan aspek untuk mengukur keandalan sistem informasi yang digunakan pengembangan aplikasi sistem informasi.

c. Compatibility

Pengujian *compatibility* ini merupakan testing yang dilakukan dengan menjalankan produk, sistem atau komponen dengan perangkat keras dan/atau perangkat lunak dalam suatu sumber daya yang sama.

d. Usability

Pengujian *usability* ini merupakan kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.

e. Reliability

Pengujian *reliability* ini merupakan kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.

f. Security

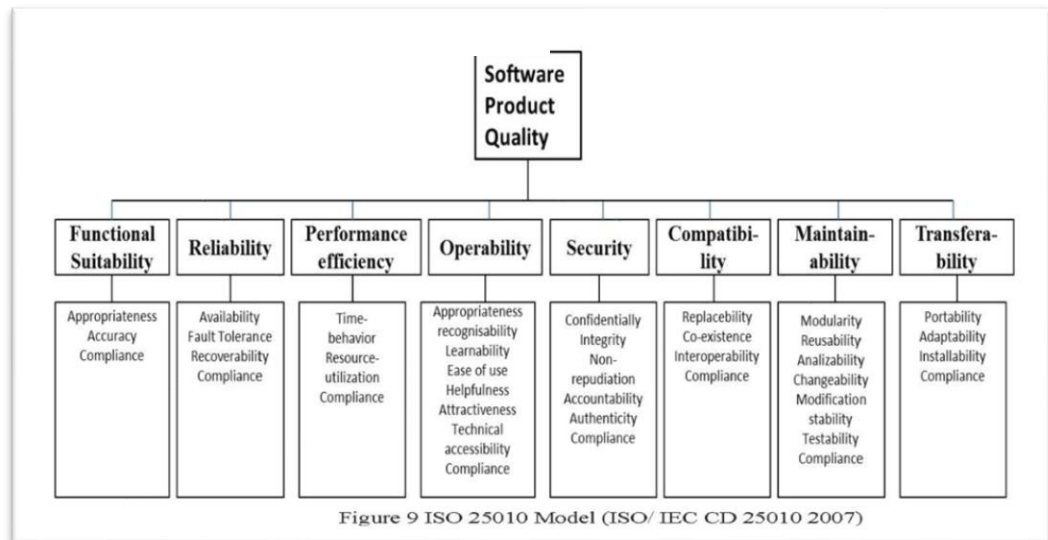
Pengujian *security* dilakukan untuk menguji keamanan aplikasi web terhadap serangan atau attack kepada sistem.

g. Maintainability

Pengujian *maintainability* dilakukan untuk menguji efektifitas dan efisiensi perangkat lunak untuk dimodifikasi atau dikembangkan.

h. Portability

Pengujian portability ini merupakan kemampuan perangkat lunak untuk di transfer dari satu lingkungan ke lingkungan lain.



Gambar 2.4 Model Kualitas Perangkat Lunak Model ISO 25010

Masing-masing karakteristik kualitas perangkat lunak model ISO 25010 dibagi menjadi beberapa sub-karakteristik kualitas. Tabel karakteristik Kualitas Perangkat Lunak Model ISO 25010 dapat dilihat pada tabel 2.6 berikut:

Tabel 2.5 Karakteristik ISO 25010

Karakteristik	Sub Karakteristik	Deskripsi
<i>Functionality</i>	<i>Functional Completeness</i>	Kemampuan perangkat lunak dalam menyediakan fungsi dapat mencakup semua tugas dan tujuan pengguna secara spesifik.
	<i>Functional Correctness</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem mampu menyediakan hasil yang tepat sesuai dengan kebutuhan
	<i>Functional Appropriateness</i>	Kemampuan perangkat lunak dalam fungsi yang disediakan mampu menyelesaikan tugas dan tujuan secara spesifik.
<i>Performance Efficiency</i>	<i>Time Behaviour</i>	Kemampuan perangkat lunak sejauh mana respon dan lama proses sebuah produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria.
	<i>Resource Utilization</i>	Kemampuan perangkat lunak sejauh mana jumlah dan jenis sumber daya yang digunakan produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria.
	<i>Capacity</i>	Kemampuan perangkat lunak sejauh mana batas maksimal sebuah produk atau sistem mampu memenuhi kriteria.

Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)

Karakteristik	Sub Karakteristik	Deskripsi
<i>Compatibility</i>	<i>Co-existence</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien saat berbagi sumber daya dengan produk atau sistem lain tanpa memberi dampak terhadap produk atau sistem tersebut.
	<i>Interoperability</i>	Kemampuan perangkat lunak sejauh mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.
<i>Usability</i>	<i>Appropriateness</i> <i>Recognizability</i>	Kemampuan perangkat lunak sejauh mana pengguna mengetahui sebuah produk atau sistem sesuai dengan kebutuhan mereka.
	<i>Learnability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan pengguna tertentu untuk mencapai tujuan mempelajari sebuah produk atau sistem secara efektif, efisien dan bebas dari resiko serta memenuhi kepuasan dalam konteks penggunaan.

Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)

Karakteristik	Sub Karakteristik	Deskripsi
<i>Usability</i>	<i>Operability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
	<i>User Error Protection</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem melindungi pengguna dalam melakukan kesalahan.
	<i>User Interface Aesthetics</i>	Kemampuan perangkat lunak sejauh mana tampilan antarmuka memenuhi kesenangan dan kepuasan pengguna.
	<i>Accessibility</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan oleh pengguna secara luas untuk mencapai tujuan tertentu dalam konteks
	<i>Maturity</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen memenuhi kriteria reliabilitas dibawah kondisi normal
	<i>Availability</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen.
	<i>Fault Tolerance</i>	Kemampuan perangkat lunak sejauh mana sistem, produk atau komponen.
	<i>Recoverability</i>	Kemampuan perangkat lunak ketika produk atau sistem mengalami kejadian atau kegagalan dapat mengembalikan data dan menjalankan kondisi sistem

Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)

Karakteristik	Sub Karakteristik	Deskripsi
<i>Security</i>	<i>Cofidentially</i>	Kemampuan perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna, apakah sebagai ancaman atau kejengajaan.
	<i>Integrity</i>	Merupakan tingkat dimana kelengkapan dan ketepatan dari sejumlah asset telah dijaga.
	<i>Non-repudiation</i>	Kemampuan perangkat lunak dimana aksi atau tindakan yang dilakukan telah terbukti sehingga hal tersebut tidak dapat ditolak.
	<i>Accountability</i>	Kemampuan perangkat lunak dimana aksi dari sebuah entitas dapat ditelusuri keunikannya terhadap entitas.
	<i>Authenticity</i>	Kemampuan perangkat lunak dimana identitas dari sebuah subjek atau sumber dapat diterima.
<i>Maintainbility</i>	<i>Modularity</i>	Kemampuan perangkat lunak dimana sistem atau program terdiri dari komponen lain.
	<i>Reusability</i>	Kemampuan perangkat lunak dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan asset lainnya.

Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)

Karakteristik	Sub Karakteristik	Deskripsi
<i>Maintainability</i>	<i>Analyzability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak.
	<i>Modifiability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat menghindari efek yang tidak diharapkan dari modifikasi.
	<i>Testability</i>	Kemampuan perangkat lunak dimana perangkat lunak memungkinkan perangkat lunak untuk dilakukan validasi.
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak dapat beradaptasi dengan perubahan lingkungan atau sistem yang berbeda.
	<i>Installability</i>	Kemampuan perangkat lunak dapat digunakan dalam lingkungan atau sistem tertentu.
	<i>Replaceability</i>	Kemampuan perangkat lunak dapat menggantikan perangkat lunak lain apakah ada kebergantungan kepada perangkat tersebut digunakan.

Adapun alasan penggunaan ISO 25010 karena ISO sudah berstandar *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*.

2.9 Android Studio

Android Studio adalah sebuah *Integrated Development Environment (IDE)* untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android.

2.10 Global Positioning Service (GPS)

Global Positioning System (GPS) merupakan suatu kumpulan satelit dan sistem kontrol yang memungkinkan sebuah penerima GPS untuk mendapatkan lokasinya dipermukaan bumi 24 jam sehari. Sistem ini menggunakan minimal 4 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan posisi, kecepatan, arah, dan waktu (Prasetyo, 2010) Ada tiga bagian penting dari sistem ini, yaitu

1. Bagian Kontrol

Berfungsi sebagai pengontrol, Setiap satelit dapat berada sedikit diluar orbit, sehingga bagian ini melacak orbit satelit, lokasi, ketinggian, dan kecepatan. Sinyal-sinyal satelit diterima oleh bagian kontrol, dikoreksi, dan dikirimkan kembali ke satelit. Koreksi data lokasi yang tepat dari satelit ini disebut dengan data ephemeris, yang nantinya akan di kirimkan kepada alat navigasi kita.

2. Bagian Angkasa

Bagian ini terdiri dari kumpulan satelit-satelit yang berada di orbit bumi, sekitar 12.000 mil di atas permukaan bumi. Kumpulan satelit-satelit ini diatur sedemikian rupa sehingga alat navigasi setiap saat dapat menerima paling sedikit sinyal dari empat buah satelit. Sinyal satelit ini dapat melewati awan, kaca, atau plastik, tetapi tidak dapat melewati gedung atau gunung.

3. Bagian Pengguna

Bagian ini terdiri dari alat navigasi yang digunakan. Satelit akan memancarkan data almanak dan ephemeris yang akan diterima oleh alat navigasi secara teratur.

2.11 Latitude Longitude

Latitude disebut juga garis lintang. Garis lintang merupakan garis vertical yang mengukur sudut antara suatu titik dengan garis katulistiwa. Titik di utara garis katulistiwa dinamakan lintang utara, sedangkan titik diselatan katulistiwa dinamakan lintang selatan. Longitude disebut juga garis bujur. Garis bujur yaitu garis horizontal yang mengukur sudut antara satu titik dengan titik nol bumi, yaitu Greenwich di London, Britania Raya yang merupakan titik 0 0 atau 3600 yang diterima secara internasional. Titik dibarat bujur 0 0 dinamakan bujur barat, sedangkan titik di timur 3600 dinamakan bujur timur. Sirenden dan Dachi (2012).