



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Menurut Januarti (2012) dalam Ikhsan dan Kurniawan (2015:13), komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer tersebut dan menghasilkan keluaran informasi yang dihasilkan setelah diolah.

Menurut Iskandar (2015), mendefinisikan komputer adalah serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasar urutan instruksi ataupun program yang diberikan kepadanya.

Menurut Hamacher, Vranesic, dan Zaky (2001), komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi input digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya, dan menghasilkan output berupa informasi.

Jadi, komputer adalah serangkaian alat elektronik yang digunakan untuk mengolah data menjadi suatu informasi.

2.2. Teori Judul

2.2.1. Pengertian LRT(*Light Rail Transit*)

Menurut Aji (2014), *Light Rail Transit* (LRT) adalah salah satu sistem kereta api penumpang yang beroperasi di kawasan perkotaan yang konstruksinya ringan dan bisa berjalan bersama lalu lintas lain datau dalam lintasan khusus (tram). Angkutan Kereta Api ringan (LRT) adalah bentuk rel dialiri listrik yang telah di kembangkan secara bertahap dari tram untuk sistem angkutan cepat yang sebagian di operasikan pada jalurnya sendiri. Trem merupakan kereta yang



memiliki rel khusus didalam kota, dengan Trem yang berselang waktu 5-10 menit berangkat, merupakan solusi untuk kemacetan.

Jadi, LRT (*Light Rail Transit*) adalah suatu alat transportasi yaitu kereta api penumpang yang dapat berjalan dengan lalu lintas lainnya.

2.2.2. Pengertian Algoritma A*(Star)

Menurut Syukriah, Falahah, dan Solihin (2016:220), mengemukakan bahwa algoritma A* (*Star*) dikenal sebagai salah satu algoritma yang paling sering digunakan untuk pencarian jalur (*path finding*) dan penerusan grafis (*graph traversal*), yaitu proses plotting jalur yang paling efisien antar titik, yang disebut dengan *nodes*.

Metode A* adalah metode yang merupakan hasil pengembangan dari metode dasar *Best First Search*. Metode ini mengevaluasi setiap titik dengan mengkombinasikan dengan $g(n)$, nilai untuk mencapai titik n dari titik awal, dan $h(n)$, nilai perkiraan untuk mencapai tujuan dari titik n tersebut.

Algoritma ini menggunakan fungsi *distance – plus – cost* (biasanya di notasikan dengan $f(x)$) untuk menentukan urutan kunjungan pencarian *node* di dalam tree. Gabungan jarak – plus – biaya merupakan penjumlahan dari dua fungsi, yaitu fungsi *path – cost* (selalu dinotasikan dengan $g(x)$, dimungkinkan bernilai heuristik ataupun tidak), dan sebuah kemungkinan penerimaan atas “perkiraan heuristik” jarak ke titik tujuan (dinotasikan dengan $h(x)$). Fungsi *path-cost* $g(x)$ adalah jumlah biaya yang harus dikeluarkan dari node awal menuju node tujuan.

Dengan terlebih dahulu mencari rute yang tampaknya mempunyai kemungkinan besar untuk menuju ke arah tujuan, algoritma ini mengambil jarak perjalanan ke arah tujuan (dimana $g(x)$ bagian dari heuristik adalah biaya dari awal). Beberapa terminologi dasar yang terdapat pada algoritma ini adalah *starting point*, simpul (*nodes*), *A*, *open list*, *closed list*, harga (*cost*), halangan (*unwalkable*).



Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (*starting point*) menuju simpul tujuan dengan memperhatikan harga (F) terkecil. Diawali dengan menempatkan A pada *starting point*, memasukkan seluruh simpul yang bertetangga dan tidak memiliki atribut rintangan dengan A ke dalam *open list*, mencari nilai H terkecil dari simpul-simpul dalam *open list* tersebut, memindahkan A ke simpul yang memiliki nilai H terkecil. Simpul sebelum A disimpan sebagai parent dari A dan dimasukkan ke dalam *closed list*. Jika terdapat simpul lain yang bertetangga dengan A (yang sudah berpindah) namun belum termasuk kedalam anggota *open list*, maka masukkan simpul-simpul tersebut ke dalam *open list*. Setelah itu, bandingkan nilai G yang ada dengan nilai G sebelumnya. Jika nilai G sebelumnya lebih kecil maka A kembali ke posisi awal. Simpul yang pernah dicoba dimasukkan ke dalam *closed list*. Hal tersebut dilakukan berulang-ulang hingga terdapat solusi atau tidak ada lagi simpul lain yang berada pada *open list*.

Jadi, Algoritma A*(star) adalah algoritma yang berfungsi untuk menentukan pencarian rute terpendek suatu perjalanan.

2.2.3. Pengertian Android

Menurut Safaat (2012:1), mendefinisikan bahwa android merupakan suatu software (perangkat lunak) yang digunakan pada mobile device (perangkat berjalan) yang meliputi sistem operasi, middleware, dan aplikasi inti. Android Standard Development Kit (SDK) menyediakan alat dan Application Programming Interface (API) yang diperlukan untuk memulai pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java, yaitu kode Java yang terkompilasi dengan data dan file resources yang dibutuhkan aplikasi dan digabungkan oleh app tools menjadi paket Android. File tersebut ditandai dengan ekstensi .apk. File inilah yang didistribusikan sebagai aplikasi dan diinstal pada perangkat mobile.

Menurut Muhammad dan Endang (2015), mendefinisikan android adalah sebuah software stack open source diciptakan untuk beragam perangkat dengan faktor bentuk yang berbeda. Tujuan utama dari Android adalah untuk



menciptakan sebuah platform perangkat lunak terbuka yang tersedia untuk operator, OEM, dan pengembang dalam membuat ide-ide inovatif mereka menjadi kenyataan.

Menurut Nazruddin (2012), android adalah aplikasi sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

Jadi, android adalah sebuah sistem operasi berbasis linux yang digunakan untuk perangkat bergerak seperti layar sentuh dan lain – lain.

2.3. Teori Khusus

2.3.1. Usecase Diagram

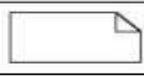
Menurut Fowler (2005:141), usecase adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Use Case mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

Menurut Gellysa dan Siregar (2015:94), mengemukakan bahwa usecase diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat.

Menurut Havaluddin (2011:6), usecase adalah diagram yang menggambarkan actor, usecase dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor.

Jadi, usecase adalah suatu diagram yang menggambarkan interaksi anatara pengguna sistem dengan sistem. Simbol-simbol yang digunakan dalam Use Case Diagram yaitu:



GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2.1 Simbol – Simbol Usecase

2.3.2. Class Diagram

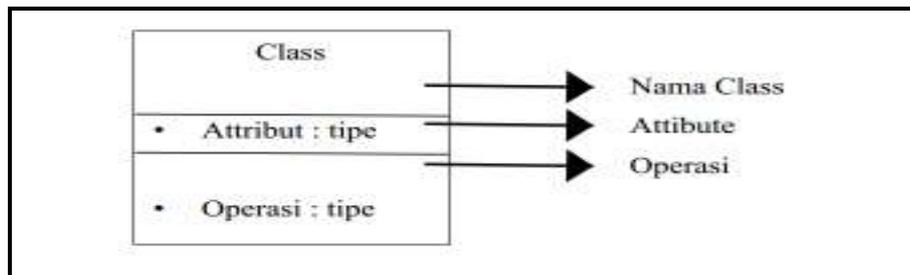
Menurut Sukamto dan Shalahuddin dalam Aprianti (2016:23), Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas - kelas yang akan dibuat untuk membangun sistem.

Menurut Gellysa dan Siregar (2015:95), mengemukakan bahwa *class diagram* merupakan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Menurut Haviluddin (2011:3), *class diagram* menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan

antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai

Kelas memiliki yang disebut atribut dan metode atau operasi. Berikut ini adalah struktur dari diagram kelas:



Gambar 2.2 Struktur Diagram Kelas (*Class Diagram*)

2.4. Teori Pemrograman

2.4.1. Pengertian Android Studio

Menurut Purba, android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android. Android Studio adalah Lingkungan Pengembangan Terpadu-Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA.

Menurut Anisa, Arief, dan Muhammad (2017:49) Android Studio adalah sebuah Integrated Development Environment (IDE) utama Google untuk mengembangkan pada platform Android. Karena Android Studio merupakan IDE dari Google, maka software ini dapat secara langsung terintegrasi dengan Google Maps menggunakan API Key yang dibuat di laman yang disediakan dari Google Maps API untuk mengintegrasikan peta dengan software sehingga peta akan secara otomatis ditampilkan di aplikasi yang dibuat.

Jadi, android studio adalah suatu integrated development environment (IDE) resmi untuk pengembangan aplikasi android.



2.4.2. Pengertian PHP (*Hypertext Preprocessor*)

Menurut Peranginangin (2006:2), PHP merupakan singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa script side server dalam pengembangan internet yang disisipkan pada dokumen HTML ().

Menurut Wahyono (2005:5), PHP atau *Hypertext Preprocessor* merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. Artinya, dapat membentuk suatu tampilan berdasarkan permintaan terkini.

Menurut Winarno dan Zaki (2014 : 49) dalam Kurniawan PHP adalah sebuah bahasa pemrograman web berbasis *server (server-side)* yang mampu memarsing kode PHP dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan website yang dinamis di sisi *client*(browser). PHP adalah bahasa *script* yang sangat cocok untuk pengembangan web dan dapat dimasukkan.

Jadi, PHP (*Hypertext Preprocessor*) adalah suatu bahasa pemrograman dengan bentuk script code yang akan disisipkan ke dalam HTML (*HyperText Markup Language*).

2.4.3. Pengertian Google Maps API

Penelitian tentang desain aplikasi ini berbasis WebGIS yang terintegrasi dengan Google Maps API. Google Maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google dapat ditemukan di <http://maps.google.com>. Ia menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia dan baru-baru ini, Bulan, dan juga menawarkan perencanaan rute dan pencari letak bisnis di U.S., Kanada, Jepang, Hong Kong, Cina, UK, Irlandia (hanya pusat kota) dan beberapa bagian Eropa. Google Maps masih berada dalam tahap beta. Google Maps API merupakan aplikasi interface yang dapat diakses lewat javascript agar Google Map dapat ditampilkan pada halaman web yang sedang kita bangun. Untuk dapat mengakses Google Maps, Kita harus melakukan pendaftaran Api Key terlebih dahulu dengan data pendaftaran berupa nama domain web yang kita bangun.



Menurut Faya dan Fitrin (2013:164), 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.

Menurut Imadnuddin (2016:A49), Google Maps API adalah salah satu Application Programming Interface (API) yang dimiliki Google. API ini mempunyai fitur untuk melakukan aktivitas-aktivitas yang berkaitan dengan Google Maps, antara lain menampilkan peta, mencari rute terdekat antara dua tempat, dan lain sebagainya. Google Maps API tersedia untuk platform android, iOS, web, dan juga web service.

Jadi, Google Maps API adalah suatu aplikasi yang disediakan oleh google, yang menyediakan berbagai fitur untuk melakukan aktifitas yang berkaitan dengan google maps.