

BAB III

LANDASAN TEORI

Bab ini akan membahas hal-hal yang mendasari dibuatnya aplikasi, arsitektur, bahasa pemrograman dan *tools* yang digunakan dalam pembuatan aplikasi *mobile* untuk peta jalur pendakian gunung berbasis GIS dan GPS.

3.1. GUNUNG

3.1.1. Pengertian Gunung

Definisi gunung adalah suatu bentuk permukaan tanah yang letaknya jauh lebih tinggi dari pada tanah-tanah sekitarnya. Gunung pada umumnya lebih besar dibandingkan dengan bukit, tetapi bukit disuatu tempat bisa jadi lebih tinggi dibandingkan apa yang disebut gunung ditempat lain. Gunung pada umumnya memiliki lereng yang curam dan tajam atau bisa juga dikelilingi oleh puncak-puncak atau pegunungan. Pada beberapa ketinggian gunung di Indonesia bisa memiliki dua atau lebih iklim, jenis-jenis tumbuh-tumbuhan dan kehidupan yang berbeda (Sondakh & Tumbel, 2016).

3.1.2. Pendaki Gunung

Panjat gunung adalah olahraga, hobi, ataupun profesi berjalan dan mendaki pegunungan. Bermula sebagai usaha total seluruh kemampuan hidup untuk mencapai titik tertinggi pegunungan yang didaki, olahraga ini berkembang ke arah kekhususan yang menyebutkan sejumlah aspek pegunungan yang berbeda dan paling tidak sekarang ada 3 aspek: keahlian

cadas, salju, dan ski, bergantung pada rute yang dipilihnya. Semuanya memerlukan keterampilan teknik dan atletik selain itu pengalaman juga penting.

3.2. SISTEM INFORMASI

Definisi informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang (Davis, 1995). McLeod (1995) mengatakan bahwa informasi adalah data yang telah diproses, atau data yang memiliki arti (Fatta, 2007).

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal dari data *item*. Data adalah kenyataan yang menggambarkan sesuatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*events*) adalah sesuatu yang terjadi pada saat tertentu. Kesatuan nyata (*fact & entity*) adalah berupa suatu objek nyata dari tempat, orang, benda yang betul-betul ada dan terjadi. Kualitas dari suatu informasi tergantung tiga hal (Kusrini & Koniyo, 2007), yaitu :

1. Informasi harus akurat (*Accurate*)

Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa menyesatkan. Akurat juga berarti informasi jelas mencerminkan maksudnya. Informasi harus akurat dari sumber informasi sampai penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah dan merusak informasi tersebut.

2. Tepat pada waktunya (*Time Lines*)

Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat.

3. Relevan

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya. Disamping karakteristik, nilai informasi juga ikut menentukan kualitasnya. Nilai informasi (*value of information*) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya.

3.3. GIS (*Geographic Information System*)

Sistem Informasi Geografis (SIG) merupakan suatu teknologi informasi yang didalamnya menggabungkan teknologi pengumpulan data, teknologi sistem basis data dan teknologi sistem komputer yang berbasis keruangan/spasial untuk menganalisis data dan menampilkannya menjadi informasi yang bermanfaat (Chang, 2004).

Data SIG dapat dibagi menjadi dua macam, yaitu data grafis dan data atribut atau tabular. Data grafis adalah data yang menggambarkan bentuk atau kenampakan objek di permukaan bumi. Sedangkan data tabular adalah data deskriptif yang menyatakan nilai dari data grafis tersebut.

Secara garis besar, data grafis dibedakan menjadi tiga macam yaitu data titik (*point*), garis (*line*) dan area (*region* atau *polygon*). Data grafis Titik biasanya digunakan untuk mewakili objek kota, bangunan, kondisi lahan. Data Garis dapat dipakai untuk menggambarkan jalan, sungai, jaringan atau saluran dan lain-lain. Sementara data Area digunakan untuk mewakili batas administrasi, penggunaan lahan, kemiringan lereng dan lain-lain. Sedangkan struktur data SIG ada dua macam yaitu *vector* dan *raster*. Pada struktur data *vector* posisi objek dicatat pada sistem koordinat. Sedangkan objek pada struktur data *raster* disimpan pada *grid* dua dimensi yaitu baris dan kolom.

Data atribut atau tabular merupakan data yang menyimpan informasi mengenai nilai atau besaran dari data grafis. Untuk struktur data *vector*, data atribut tersimpan secara terpisah dalam bentuk tabel. Sementara pada struktur data *raster* nilai data grafisnya tersimpan langsung pada nilai *grid* atau *pixel* tersebut.

3.4. LBS (*Location Based Service*)

Location Based Service (LBS) atau layanan berbasis lokasi adalah sebuah layanan informasi yang dapat diakses dengan perangkat bergerak melalui jaringan dan mampu menampilkan posisi secara geografis keberadaan perangkat bergerak tersebut. Location Based Service dapat berfungsi sebagai layanan untuk mengidentifikasi lokasi dari seseorang atau suatu objek tertentu.

Salah satu unsur utama LBS menyediakan teknologi pencarian lokasi (*Location Providers*) yang digunakan oleh perangkat *mobile*. *Location Providers* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. *Location Providers* berada pada paket Android yaitu dalam paket "android.location". Lokasi, perpindahan, serta kedekatan dengan lokasi tertentu dapat ditentukan melalui *Location Manager*. Pada piranti *mobile* fungsi LBS menjadi salah satu komponen penting. Piranti ini berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Selain itu, piranti *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS (*Global Positioning System*).

3.5. GPS (*Global Positioning System*)

GPS (*Global Positioning System*) adalah sebuah sistem navigasi berbasis radio yang menyediakan informasi koordinat posisi, kecepatan, dan waktu kepada pengguna di seluruh dunia. Jasa penggunaan satelit GPS tidak dikenakan biaya. Pengguna hanya membutuhkan GPS *receiver* untuk dapat mengetahui koordinat lokasi. Keakuratan koordinat lokasi tergantung pada tipe GPS *receiver*.

GPS terdiri dari tiga bagian yaitu satelit yang mengorbit bumi (Satelit GPS mengelilingi bumi 2x sehari), stasiun pengendali dan pemantau di bumi, dan GPS *receiver* (alat penerima GPS). Satelit GPS dikelola

oleh Amerika Serikat. Alat penerima GPS inilah yang dipakai oleh pengguna untuk melihat koordinat posisi. Selain itu GPS juga berfungsi untuk menentukan waktu.

Satelit GPS memancarkan dua sinyal yaitu frekuensi L1 (1575.42 MHz) dan L2 (1227.60 MHz). Sinyal L1 dimodulasikan dengan dua sinyal *pseudo-random* yaitu kode P (*Protected*) dan kode C/A (*coarse/aquisition*). Sinyal L2 hanya membawa kode P. Setiap satelit mentransmisikan kode yang unik sehingga penerima (*GPS Receiver*) dapat mengidentifikasi sinyal dari setiap satelit (Hartanto, 2012).

3.6. OPEN STREET MAP

Open Street Map (OSM) adalah sebuah proyek berbasis web untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survei menggunakan GPS, mendigitasi citra satelit, dan mengumpulkan serta membebaskan data geografis yang tersedia di publik.

Melalui Open Data Commons Open Database License 1.0, kontributor OSM dapat memiliki, memodifikasi, dan membagikan data peta secara luas. Terdapat beragam jenis peta digital yang tersedia di internet, namun sebagian besar memiliki keterbatasan secara legal maupun teknis. Hal ini membuat masyarakat, pemerintah, peneliti dan akademisi, inovator, dan banyak pihak lainnya tidak dapat menggunakan data yang tersedia di dalam peta tersebut secara bebas. Di sisi lain, baik peta dasar OSM maupun data yang tersedia di dalamnya dapat diunduh secara gratis dan terbuka, untuk kemudian

digunakan dan didistribusikan kembali (OpenStreetMap Indonesia, 2016).

3.7. WEB SERVER

Web Server adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP (*Hypertext Transfer Protocol*) atau HTTPS (*Hypertext Transfer Protokol Secure*) dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML (*Hyper Text Markup Language*).

3.8. MVC (Model View Control)

MVC adalah arsitektur aplikasi yang memisahkan kode-kode aplikasi dalam tiga lapisan yaitu *Model*, *View* dan *Control*. MVC termasuk dalam arsitektural *design pattern* yang menghendaki organisasi kode yang terstruktur dan tidak bercampur aduk. Ketika aplikasi sudah sangat besar dan menangani struktur data yang kompleks, harus ada pemisahan yang jelas antara *domain model*, komponen *view* dan *control* yang mengatur penampilan *model* dalam *view* (Microsoft, 2015).

3.9. LARAVEL

Framework adalah komponen pemrograman yang siap *re-use* (dapat digunakan ulang) kapan saja, sehingga *programmer* tidak harus membuat skrip yang sama untuk tugas yang sama. Laravel merupakan sebuah *framework* PHP (*Hypertext Preprocessor*) yang sangat populer dikalangan *programmer* saat ini. Pembuat *laravel* adalah Taylor Otwell. Alasan pembuatan *framework* ini adalah karena Taylor Otwell kurang puas dengan *framework* CodeIgniter,

karena ada beberapa fitur fungsional menurut Taylor penting yang tidak *support*, seperti kotak autentikasi dan *routing*. Taylor sendiri memiliki latar belakang .Net. Seperti yang sudah umum diketahui bahwa teknologi .Net biasa digunakan untuk membuat aplikasi *enterprise*. Dan Taylor Otwell mencoba mengaplikasikan pengalamannya tersebut ke dalam Laravel (Kasman, 2015). Berikut adalah beberapa kelebihan Laravel :

1. *Expressif*

Laravel adalah sebuah *framework* PHP (*Hypertext Preprocessor*) yang *expressif*, artinya ketika melihat suatu *syntax* Laravel, seorang *programmer* diharapkan akan langsung mengetahui kegunaan dari *syntax* tersebut meskipun belum pernah mempelajarinya apalagi menggunakannya.

2. *Simple*

Laravel begitu *simple* dikarenakan adanya *Eloquent ORM*. *Eloquent ORM* memungkinkan pengembang untuk mengambil semua data yang ada pada tabel dengan cara membuat *class model*.

3. Dikembangkan secara khusus untuk PHP 5.3

PHP 5.3 memiliki banyak fitur baru dalam segi bahasa, yang membuat PHP terasa lebih *modern* dan *powerfull*. Sehingga dengan menggunakan *framework* Laravel ini tidak ada *backward compatibility* dengan PHP versi sebelumnya.

4. Dokumentasi yang baik

Laravel dibuat dengan dokumentasi yang sangat lengkap. *Core Developer* dari laravel sendiri

berkomitmen untuk selalu menyertakan dokumentasi yang lengkap setiap kali melakukan rilis versi terbarunya.

3.10. APLIKASI

Menurut (Ibisa, 2011) aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi para penggunanya. Sedangkan Menurut (Shelly & Vermaat, 2008) aplikasi adalah seperangkat intruksi dalam komputer yang dirancang agar kita menyelesaikan tugas-tugas tertentu. Aplikasi biasanya berupa *software* yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan suatu fungsi bagi pengguna. Beberapa aktivitas yang dilakukan oleh manusia seperti sistem untuk *software* jual beli, permainan atau *game online*, pelayanan masyarakat dan hampir semua proses yang dilakukan oleh manusia dapat dibantu dengan menggunakan suatu aplikasi.

3.11. ANDROID

Android adalah sebuah Sistem Operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel atau *smartphone* kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat selular. Di dunia terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung dari Google atau dikenal sebagai *Open Handset Distribution*(OHD).

September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon selular ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010, Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericson, Thosiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance (OHA), OHA mengumumkan produk perdana mereka, Android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6.

Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. Pada saat ini sebagian besar vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis android, vendor-vendor itu antara lain Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, Nexian, IMO, Asus dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi android.

Hal ini karena android adalah sistem operasi yang open source sehingga bebas didistribusikan dan dipakai oleh vendor manapun (Safaat, 2011).

3.12. MYSQL

MySQL adalah database yang menghubungkan *script* PHP (*Hypertext Preprocessor*) menggunakan perintah *query* dan *escaps character* yang sama dengan PHP. MySQL mempunyai tampilan *client* yang mempermudah dalam mengakses database (Sadeli, 2013).

3.13. SQLITE

Android memiliki fasilitas untuk membuat database yang dikenal dengan SQLite. SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi SQL *interface* dan penggunaan *memory* yang sangat sedikit dengan kecepatan yang sangat cepat SQLite di android termasuk dalam Android *runtime*, sehingga setiap versi dari android dapat membuat database dengan SQLite.

3.14. KML (Keyhole Markup Language)

Keyhole Markup Language (KML) adalah sebuah XML (*Extensible Markup Language*) berbasis bahasa geografis skema untuk mengungkapkan penjelasan dan visualisasi yang ada atau masa depan *Web-based*, peta dua dimensi dan tiga dimensi Bumi *browser*. KML ini dikembangkan untuk digunakan dengan Google Earth, yang awalnya bernama *Keyhole Earth Viewer*. Itu diciptakan oleh Keyhole, Inc, yang diakuisisi oleh Google pada tahun 2004.

KML adalah standar internasional yang *Open Geospatial Consortium*. *Google Earth* adalah program pertama bisa melihat dan mengedit grafis file KML, dan proyek lainnya seperti *Marmer* juga mulai mengembangkan dukungan KML. Dapat menggunakan alamat KML yang disediakan untuk memperoleh informasi yang anda butuhkan dengan menggunakan perangkat lunak yang mendukung. Misalnya *Google Earth*, *Mobile Atlas Creator* (MOBAC), dll (Kementerian Kehutanan Republik Indonesia).

3.15. JAVA

Java merupakan bahasa pemrograman yang dikembangkan dari bahasa pemrograman C++. Berawal dari proyek penelitian perusahaan Sun Microsystems dengan nama *Sandi Green* pada tahun 1991. Terdapat prediksi bahwa mikroprosesor akan digunakan luas pada peralatan-peralatan elektronik. Karena adanya bermacam tipe mikroprosesor, maka dibutuhkan sebuah bahasa pemrograman yang dapat berjalan di semua mikroprosesor. Terciptalah sebuah bahasa pemrograman baru oleh James Gosling yaitu salah satu orang yang berperan besar dalam proyek tersebut, program ini diberi nama *Oak*. Sesuai dengan pohon *Oak* yang tumbuh dan bisa dilihat melalui jendela kerjanya di Sun Microsystems.

Selang beberapa waktu kemudian, ditemukan bahwa sudah ada bahasa pemrograman dengan nama *Oak*. Akhirnya setelah beberapa pegawai Sun mengunjungi sebuah kedai kopi nama bahasa pemrograman ini diganti dengan *Java*. *Java* merupakan salah satu jenis biji kopi yang ada di kedai tersebut yaitu biji kopi jawa. Sun Microsystems mengumumkan kehadiran bahasa *Java* secara formal di

tahun 1995. Bahasa ini mulai disambut hangat masyarakat luas seiring dengan meledaknya era internet (Hakim & Sutarto, 2012).

3.16. GENNYMOTION

Genymotion adalah emulator android yang terdiri satu set lengkap sensor dan fitur untuk berinteraksi dengan *android virtual*. Kegunaan Genymotion untuk menguji aplikasi android pada berbagai perangkat *virtual* untuk keperluan pengembangan, pengujian dan demonstrasi. Genymotion sangat cepat, mudah untuk menginstal dan teruji kuat untuk *widget sensor* dan fitur interaksi. Aplikasi emulator ini tersedia untuk Windows, Mac OS X dan Sistem Operasi Linux.

3.17. UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML model dapat dibuat untuk semua jenis aplikasi piranti lunak, dimana aplikasi dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam berbagai bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB (*Visual Basic*) atau C.

Seperti bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantic*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu dan UML *syntax* mendefinisikan bagaimana bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya seperti Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*) dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). UML mendefinisikan diagram sebagai berikut:

1. *Use Case Diagram*

Menggambarkan fungsionalitas dari sebuah sistem (apa fungsinya), yang mempresentasikan sebuah interaksi antara aktor dengan sistem (sebuah pekerjaan). Misalnya menambah data/membuat laporan. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan.

2. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain.

3. *Sequence Diagram*

Diagram ini menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah

event untuk menghasilkan *output* tertentu. Diawali dari sebuah aktivitas tertentu, kemudian berproses mengikuti urutan tertentu, yang bisa terlihat melalui *message* antar objeknya.

