

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini mengacu pada beberapa penelitian sebelumnya yang berkaitan dengan sistem yang akan dibuat. Adapun beberapa penelitian terdahulu yang pernah dilakukan yang akan diuraikan dibawah.

Karima & Aisyatul, (2013) dalam jurnal yang berjudul “Aplikasi Informasi Jalur Menuju Gunung Di Jawa Tengah berbasis Android” membuat aplikasi tentang informasi jalur pendakian gunung-gunung di Jawa Tengah. Dalam aplikasi ini terdapat profil gunung-gunung di jawa tengah, galeri foto, dan *maps* jalur pendakian gunung di jawa tengah saja. Aplikasi ini dibangun menggunakan eclips dengan Bahasa pemrograman Java.

Tresnawati, (2015) dalam jurnal yang berjudul “Pengembangan Aplikasi Panduan Mendaki Gunung Menggunakan Metodologi Sistem Multimedia Luther-Sutopo” membuat sebuah aplikasi panduan mendaki gunung. Dalam aplikasi ini terdapat beberapa fitur seperti pengetahuan dasar, info gunung, data pendakian, persiapan, pelaksanaan, dan bantuan.

Anshori, (2014) dalam skripsi yang berjudul “ Aplikasi Panduan Mendaki Gunung Berbasis Android” membuat aplikasi berbasis Android sebagai sarana informasi untuk melakukan pendakian gunung. Dalam aplikasi ini terdapat Beberapa fitur seperti penyusunan rencana, persiapan fisik, alat gunung, mendaki, survival, P3K, dan galeri.

Dari ketiga penelitian yang telah dilakukan diatas, masih terdapat beberapa kekurangan seperti kurangnya informasi gunung yang ada di indonesia, tidak adanya kompas, tidak ada menejemen peralatan yang dibutuhkan. Maka peneliti akan menambahkan beberapa fitur solusi aplikasi yang di buat yaitu :

1. Sistem yang dapat menghitung dan menampilkan hasil untuk manajemen peralatan selama pendakian.
2. Menampilkan informasi gunung se Indonesia, seperti nama gunung, ketinggian, estimasi waktu pendakian, lokasi gunung, suhu, jenis gunung, dan tempat istirahat atau tempat pendirian tenda.
3. Sistem yang akan di lengkapi dengan fitur kompas.

2.2 Landasan Teori

Untuk mendukung penelitian ini, maka perlu dikemukakan hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam penelitian ini.

2.2.1 Pendaki gunung

Pendakian gunung adalah suatu bentuk kegiatan yang dilakukan di alam terbuka dengan melakukan perjalanan menaiki pegunungan. Gunung dengan segala aspeknya merupakan lingkungan yang asing bagi organ tubuh manusia, apalagi bagi pendaki yang hidup didataran rendah, itulah sebabnya pendaki gunung memerlukan kesiapan untuk menyesuaikan diri dengan keadaan digunung. Perlengkapan yang baik dan tingkat kebugaran jasmani yang prima adalah salah satu usaha untuk mengurangi bahaya dalam pendakian gunung.

2.2.2 Pariwisata

Suwantoro (2004) menyatakan pariwisata pada hakikatnya merupakan proses kepergian seseorang dari satu daerah ke daerah lain dengan berbagai tujuan seperti ekonomi, sosial, budaya, politik ataupun hanya sekedar ingin tahu atau menambah pengalaman. Sedangkan menurut Spillane (1991)” pariwisata merupakan kegiatan perjalanan seseorang yang dilakukan dengan tujuan-tujuan istirahat, dan lain sebagainya yang harus memenuhi tiga persyaratan yaitu, bersifat sementara, sukarela, dan tidak untuk mendapatkan keuntungan materi”.

Definisi dari pariwisata sendiri memang tidak selalu sama antar ilmuwan. Namun pada intinya kegiatan pariwisata merupakan suatu kegiatan baik perseorangan ataupun kelompok ke suatu wilayah tertentu dengan berbagai motivasi perjalanan, mulai dari mencari kesenangan, beristirahat untuk menghilangkan stress, rasa ingin tahu, spiritual, bisnis, dan lain sebagainya.

2.2.3 Aplikasi

Hartono, Pengenalan Komputer (1992) menyatakan bahwa aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*. Sedangkan menurut Kamus Besar Bahasa Indonesia (1998), “Aplikasi adalah penerapan dari rancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

Aplikasi merupakan sebuah program computer yang dibuat dengan tujuan untuk membantu pengguna mengerjakan tugas khusus, serta berisi rangkaian perintah yang dieksekusi oleh computer. Program aplikasi merupakan program siap

pakai. Program yang dibuat untuk melaksanakan suatu fungsi bagi pengguna. Contoh aplikasi ialah program pemutar music dan pemroses angka. Aplikasi menggunakan sistem operasi (OS) computer dan aplikasi pendukung lainnya.

2.2.4 Android

Menurut Safaat H, (2012) *Android* adalah sebuah sistem operasi yang sengaja diciptakan untuk perangkat bergerak berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. *Android* secara umum digunakan pada *smartphone* dan tablet PC. Fungsi sistem operasi *android* sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan Blackberry OS pada Blackberry. *Android* tidak terikat pada satu merek telepon seluler saja, beberapa *vendor* terkenal yang sudah menggunakan *Android* antara lain Samsung, Asus, Nexus, dsb.

Android awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan yang membuat *software* untuk telepon seluler yang berada di Palo Alto, California Amerika Serikat. Pada tahun 2005 Google membeli atau mengakuisisi perusahaan yang waktu itu bernama *Android*, sebuah perusahaan yang berkecimpung di dunia *mobile*. Setelah diambil alih oleh Google, pada November 2007 Google mengumumkan bahwa mereka sedang mengembangkan telepon seluler Google dengan sistem operasi terbaru yang mereka beri nama *Android*. Sistem operasi ini sebenarnya didasarkan pada kernel Linux dan dirancang serta digunakan oleh *Open Handset Alliance* yang terdiri dari sekelompok pembuat *hardware*, *carrier*, dan perusahaan terkait perangkat *mobile* lainnya, diantaranya Google, HTC, Intel,

Motorola, Qualcomm, T-Mobile, Nvidia, dan lain-lain. *Open Handset Alliance* dibentuk untuk mengembangkan android yang bersifat *open source* yang *source* nya di berikan secara gratis bagi para pengembang (ilmuti, 2017)

2.2.5 System Development Life Cycle

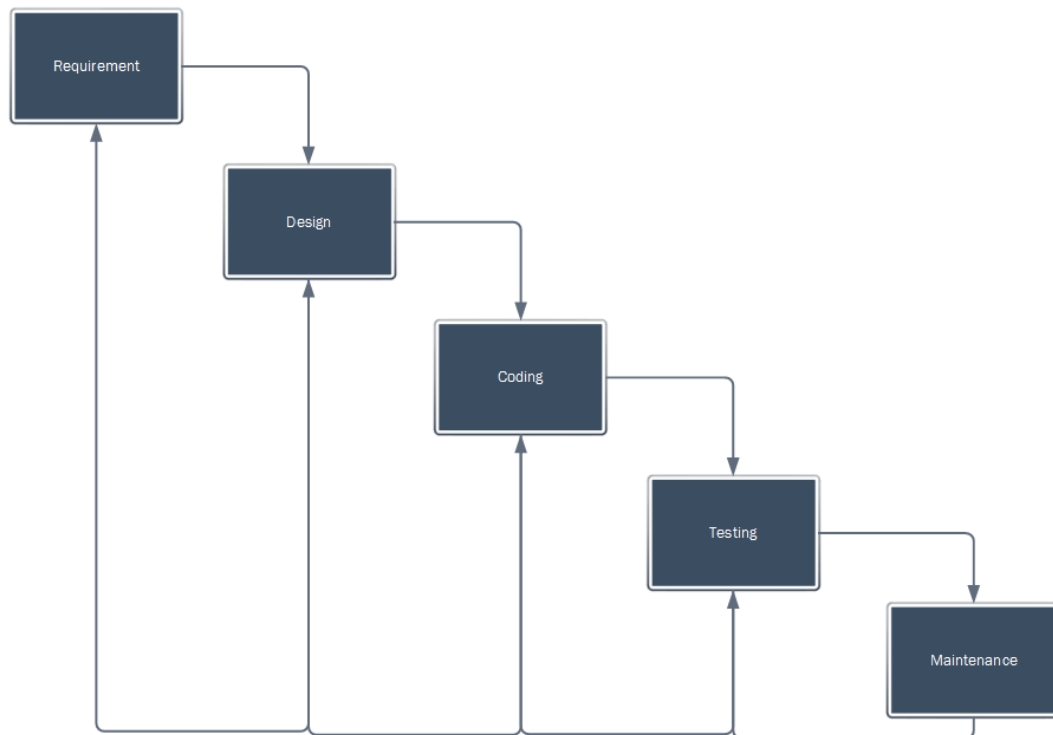
Fatwa, (2007) SDLC adalah kerangka yang mendeskripsikan sejumlah kegiatan yang dilakukan menggunakan beberapa tahap pada saat pembuatan sebuah *software*. Selain itu, SDLC juga penting pada saat melakukan *Maintenance software*.

Model SDLC yang dipakai dalam penelitian ini adalah model *Waterfall*. Menurut (Sommerville, 2011), “model *Waterfall* adalah sebuah contoh dari proses perancangan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan di jadwalkan sebelum dikerjakan”. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)*. Menurut (Bassil, 2012), disebut *waterfall* karena tahap demi tahap yang harus dilalui menunggu selesainya tahap sebelumnya dan berjalan berurutan

Alasan menggunakan model *Waterfall* yaitu memiliki beberapa kelebihan, diantara kelebihan tersebut adalah sebagai berikut:

1. Merupakan model pengembangan terstruktur.
2. Setiap fase dapat diimplementasikan dengan dokumentasi yang detail dari fase sebelumnya.
3. Aktifitas pengujian dapat dimulai di awal proyek, sehingga mengurangi waktu proyek.

Berikut ini merupakan contoh metode *waterfall* seperti pada gambar 2.1 :



Gambar 2.0.1 Metode *waterfall*

2.2.6 Teknologi Pengembangan Aplikasi

Untuk teknologi pengembangan aplikasi penulis menggunakan pengembangan teknologi *UML* (*Unified Modeling Language*). Karena menggunakan *UML* mempermudah dalam pengembangan aplikasi yang berkelanjutan. Berikut adalah penjelasan mengenai *UML* :

2.2.6.1 *Unified Modeling Language (UML)*

Siau & Cao (2001) *Unified Modeling Language (UML)* adalah sebuah bahasa pemodelan visual yang dirancang khusus untuk pengembangan dan analisis sistem berorientasi objek dan desain. *UML* pertama kali dikembangkan oleh Grady Booch, Jim Rumbaugh, dan Ivars Jacobson pada pertengahan tahun 1990.

Dibawah ini adalah Model UML yang dipakai dalam pengembangan aplikasi ini antara lain *Use Case Diagram*, *Activity Diagram*, dan *Flowchart Diagram*.

1. *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dapat dikatakan *Use Case* karena dapat digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

2. *Activity Diagram*

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis

3. *Flowchart Diagram*

Flowchart diagram merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

4. *Class Diagram*

Class diagram adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (attribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

Class diagram menggambarkan struktur dan deskripsi *class*, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain (Dharwiyanti & Wahono, 2003).

2.2.6.2 Bahasa Pemrograman Java

Menurut definisi *Sun Microsystem*, didalam buku (Shalahuddin, 2010)Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan.

Program Java merupakan bahasa yang bersifat umum (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin, serta mampu berjalan di beberapa *platform* sistem operasi yang berbeda. Java juga dikenal dengan slogannya “Tulis sekali, jalankan dimana pun”. Java merupakan bahasa pemrograman yang paling populer digunakan dan secara luas dimanfaatkan dalam berbagai jenis perangkat lunak aplikasi *mobile* ataupun aplikasi berbasis web.

Kelebihan yang dimiliki oleh Java adalah sebagai berikut:

1. *Multiplatform*

Java dapat dijalankan dalam beberapa *platform* komputer dan sistem operasi yang berbeda.

2. OOP atau *Object Oriented Programming*

Semua aspek yang terdapat dalam Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut objek. Hal ini sangat memudahkan pemrogram untuk

mendesain, membuat, mengembangkan dan mengalokasikan kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir.

3. *Multithread*

Kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. *Thread* dalam Java memiliki kemampuan untuk memanfaatkan kelebihan multi prosessor apabila sistem operasi yang digunakan mendukung multi prosessor.

4. Dapat di distribusi dengan mudah

Java memiliki *library* rutin yang lengkap untuk dirangkai pada protokol TCP/IP (seperti HTTP dan FTP) dengan mudah. Kemampuan *networking* Java lebih kuat dan lebih mudah digunakan. Java memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah soket koneksi. Java juga memudahkan pembuatan CGI (*Common Gateway Interface*).

5. Bersifat dinamis

Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang, Java bersifat dinamis dalam tahap *linking*. *Class* yang ada dapat di *link* sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di *link* dari beberapa sumber, bahkan dari sumber dalam jaringan Internet.

Kekurangan yang dimiliki oleh Java adalah sebagai berikut:

1. Pada satu slogan nya

“Tulis sekali dan jalankan dimana saja” tidak sepenuhnya benar. Beberapa hal harus disesuaikan jika dijalankan pada *platform* yang berbeda. Misalnya untuk J2SE dengan *platform* SWT-AWT *bridge* tidak dapat berfungsi di Mac OS X.

2. Aplikasi Java di dekompilasi

Dekompilasi adalah suatu proses membalikkan sebuah aplikasi menjadi kode sumbernya. Hal ini memungkinkan terjadi pula pada *platform* .NET dari *Microsoft* sehingga program yang dihasilkan mudah dibajak kodenya karena sulit untuk disembunyikan.

3. Penggunaan memori yang banyak

Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan *Pascal*. Pada umumnya, hal ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena *trend* memori terpasang semakin murah), tetapi menjadi masalah bagi mereka yang masih harus berlutut dengan mesin komputer berumur lebih dari 4 tahun.

2.2.7 Perangkat Pendukung

A. *Android Studio*

Android Studio adalah sebuah *Integrated Development Environment* (IDE) yang diperkenalkan 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.

Sebagai pengembangan dari *Eclipse*, *Android studio* mempunyai banyak fitur-fitur baru dibandingkan dengan *Eclipse* IDE. Berbeda dengan *Eclipse* yang menggunakan *Ant*, *Android studio* menggunakan *Gradle* sebagai *build environment* (Jadibaru,2015).

Fitur-fitur lainnya adalah sebagai berikut:

1. Menggunakan *Gradle-based build system* yang fleksibel.
2. Dapat mem-*build multiple* APK.
3. *Template support* untuk *Google Services* dan berbagai macam tipe perangkat.
4. *Layout editor* yang lebih bagus.
5. *Built-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. *Import library* langsung dari *Maven repository*.

B. *Adobe Photoshop CS5*

Adobe Photoshop CS5 sebuah program untuk pemrosesan grafik raster yang mendukung banyak format grafis. Adobe Photoshop memungkinkan untuk membuat gambar baru dan mengedit gambar. Adobe Photoshop digunakan untuk membuat gambar fotorealistik untuk bekerja dengan gambar berwarna scan, retouching, koreksi warna, transformasi grafik, pemisahan warna, dan lain-lain. Aplikasi Adobe Photoshop merupakan program aplikasi pengolah image atau gambar bitmap. Image atau gambar Bitmap yang sering disebut Raster, merupakan gambar yang dibentuk dari grid-grid warna. Grid ini adalah elemen dasar dari sebuah image atau gambar yang disebut pixel atau picture elements (Boediman, 1995).

C. *Microsoft Visio 2013*

Menurut Helmers dalam bukunya yang berjudul “Microsoft Visio 2013 Step by Step (2013)”, Microsoft Visio adalah aplikasi utama untuk membuat semua diagram bisnis, mulai dari flowchart, network diagram, dan organization charts,

untuk membuat denah dan brainstorming diagram. Microsoft 2013 melanjutkan kegunaan dari kebiasaan user interface, atau dikenal sebagai keterkaitan, hal itu telah diperkenalkan pada Visio 2010. Terlepas dari apa yang mungkin terpikirkan dari hubungannya dengan aplikasi Microsoft Office lainnya, dengan Visio rasanya seperti di rumah, terutama karena tujuan dari keterkaitan user interface gaya presentasi visual dari kelompok yang terkait fungsi, dan Visio termasuk didalamnya, pertama dan terutama, sebuah produk visual. ()

2.2.8 Black Box Testing

Mustaqbal Firdaus & Rahmadi (2015) *Black Box Testing* merupakan pengujian yang mudah digunakan. Metode berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program.

Fakhri M. K., (2015) metode pengujian *Black Box* digunakan untuk menguji sistem dari segi *user* yang dititik beratkan pada pengujian kinerja, spesifikasi dan antarmuka sistem tersebut tanpa menguji kode program yang ada. “*Black Box Testing* tidak membutuhkan pengetahuan mengenai alur internal (*internal path*)”, struktur atau implementasi dari *Software Under Test* (SUT). Karena itu uji coba *Black Box* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program.

