BAB II

LANDASAN TEORI

2.1 Tinjauan Jurnal

Pada penyusunan skripsi ini penulis merujuk pada bebrapa jurnal tentang pembuatan sistem aplikasi android dan fungsi dari sistem aplikasi android itu sendiri.

Dalam jurnal yang dibuat oleh Taluban dan Arliyanto Zai (2013 : 49) Pariwisata merupakan sumber devisa yang besar dari satu kota. Untuk meningkatkan sektor pariwisata, diperlukan diperlukan sebuah dukungan baik dari segi keamanan, infrasrtuktur dan juga fasilitas informasi pariwisata. Kota Medan merupakan salah satu ibu kota propinsi Sumatera Utara yang memiliki banyak objek wisata, namun dikelola dengan baik sehingga tempat – tempat wisata tersebut sangat jarang dikunjungi. Untuk itu penulis mencoba merancang sebuah aplikasi *mobile toure system* berbasis *smartphone android* untuk pemandu wisata kota medan. Metode pengembangan sistem dalam metode pengembangan ini menggunakan *waterfall*. Dalam penelitian ini, metode pengembangan sistem yang di terapkan hanya sampai pada taahap desain dan tidak melibatkan tahap yang selanjutnya yaitu implementasi, pengujian, dan *maintenance*. Hasil penelitian adalah sebuah model sistem dan desain aplikasi. Simpulan dari penelitian ini adalah model dan desain aplikasi ini nantinya diharapkan dapat membantu wisatawan yanghendak mengunjungi kota medan.

Dalam jurnal yang dibuat oleh Latifah dan Prasetyo (2015 : 82) Algoritma ini diberi nama sesuai nama penemunya, *Edsger Wybe* Djikstra. Algoritma *Dijkstra* mencari lintasan terpendek dalam sejumlah langkah. Algoritma ini menggunakan prinsip *Greedy* yang menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukkannya ke dalam himpunan solusi. *Input algoritma* ini adalah sebuah *graph* berarah yang berbobot (*weighted directed graph*) G dan sebuah sumber *vertex* s dalam G dan V adalah himpunan semua *vertices* dalam *graph* G (Rosen,1999):
Properti algoritma *Dijkstra*:

1. *Matriks* ketetanggaan M[mij]
   mij = bobot sisi (*i, j*)
   mii = 0 *mij* = ∞, jika tidak ada sisi dari simpul *i* ke simpul *j*
2. Larik S = [*si*] yang dalam hal ini,
   si = 1, jika simpul *i* termasuk ke dalam lintasan terpendek
   si = 0, jika simpul *i* tidak termasuk ke dalam lintasan terpendek

2.2 Konsep Dasar Program

1. Java

Menurut Andi (2011:1) mengatakan bahwa:

> Java adalah bahasa pemrograman yang belum lama ada, namun telah dikmbangkan dan dipergunakan secara luas, baik oleh pengembang Sun Microsystems maupun pihak lain seperti IBM. Karena java merupakan bahasa merupakan bahasa *multiplatform* , sekali anda membuat kode pada *platform* tertentu dan kompilasi, hasil kompilasi *bytecode* dapat dijalankan pada semua sistem operasi, baik Windows, Linux, Max OS, dn sebagainya.

2. Android

Menurut Safaat H (2015:10) menagatakan bahwa:

> Android adalah sebuah sistem operasi untuk prankgat *mobile* berbasis linux yang mencangkup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeliv Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*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.

3. *Software Development Kit* (SDK)

*Menu*rut Safaat H (2015:5) mengatakan bahwa:

> SDK (*Software Development Kit*) adalah *tool* API (*Aplication Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahsa pemrograman *java.* Android merupakan subset perangkat lunak pada ponsel yang meliputi sistem operasi, *middelware* dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu API untuk mulai mengembangkan aplikasi pada *platform* menggunakan bahasa pemrograman *java.*

4. *Jave Development Kit* (JDK)

Menurut Pangestu dan Widayati (2011:69) mengatakan bahwa: "JDK merupakan perangkat lunak yang digunakan untuk menajemen dan membangun berbagai aplikasi java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan *compiler* dan debugger yang diperlukan untuk mengembangkan applet dan aplikasi".

5. *Android Developer Tools* (ADT)

Menurut Safaat H (2015:6) mengatakan bahwa :

> *Android Development Tools* (ADT) adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi project android, membuat GUI aplikasi, dan menambakan komponen-kompenen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat melakukan pembuatan *package* android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Pengembangan aplikasi android dengan enggunakan ADT di eclipse sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi. Berikut *versi* ADT untuk eclipse yang sudah dirilis:

1. ADT 12.0.0 (*July* 2011)
2. ADT 11.0.0 (*Juni* 2011)
3. ADT 10.0.1 (*March* 2011)
4. ADT 10.0.0 (*February* 2011)
5. ADT 9.0.0 (*January* 2011)
6. ADT 8.0.1 (*December* 2010)
7. ADT 8.0.0 (*December* 2010)
8. ADT 0.9.9 (*September* 2010)

9. ADT 0.9.8 (*September* 2010)
10. ADT 0.9.7 (*May* 2010)
11. ADT 0.9.6 (*March* 2010)
12. ADT 0.9.5 (*December* 2009)
13. ADT 0.9.4 (*October* 2009)

Semakin tinggi *platform* android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya *platform* baru diikutioleh munculnya versi ADT ysng terbaru. Untuk melakukan instalisai ADT di eclipse dapat dilakukan secara *on-line* maupun *ofline.* Unduk *download* ADT ini dapat dilakukan di *http://developer.android.com/sdk/eclipse-adt.html*.

6. Eclipse

Menurut Pratama dan Hermawan (2012:3) mengatakan bahwa: "Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*)".

sifat dari Eclipse adalah :

a. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
b. *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
c. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat

lunak, seperti dokumentasi, *test* perangkat lunak, pengembangan web, dan lain sebagainya.

7. XML

Menurut Pangestu dan Widayati (2011:69) mengatakan bahwa:

> XML adalah singkatan dari *eXtensible Markup Language*. Bahasa *markup* adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan, dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan *tag*. Bahasa markup lain yang *populer* seperti HTML, menggambarkan kepada browser web tentang bagaimana menampilkan format teks, data, dan grafik ke layar komputer ketika sedang mengunjungi sebuah situs web. XML adalah sebuah bahasa *markup* yang digunakan untuk mengolah meta data (informasi tentang data) yang menggambarkan struktur dan maksud/tujuan data yang terdapat dalam dokumen XML, namun bukan menggambarkan format tampilan data tersebut. XML adalah sebuah standar sederhana yang digunakan untuk medeskrippsikan data teks dengan cara *self-describing* (deskripsi diri). XML juga dapat digunakan untuk mendefinisikan domain tertentu lainnya, seperti musik, matematika, keuangan dan lain-lain yang menggunakan bahasa *markup* terstruktur.

8. Basis Data (*Database*)

Menurut Fathansyah (2015:14) mengatakan bahwa: " *Database* adalah sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi sejumlah objek basis data (seperti tabel, index, dan lain lain). Di samping data, setiap basis data juga menyimpan definisi struktur (baik untuk basis data maupun objek-objeknya secara rinci)".

9. MySQL

Menurut Andi (2012:13) mengatakan bahwa: " MySQL adalah aplikasi yang digunakan membuat *query* dalam bahasa *database*, table maupun manipulasi data".

10. XAMPP

Menurut Andi (2012:5) mengatakan bahwa:

> XAMPP merupakan sebuah aplikasi *web server*. *Web server* sendiri adalah aplikasi tempat menyimppan file-file maupun data-data untuk membuat *website*. Juga sering diartikan sebagai layanan data pada web browser. Fugsi dari *web server* sebagai penerima permintaan berupa halaman *client* dan mengirimkan kembali hasil yang diminta dalam bentuk halaman *web*. *Xampp* dapat anda *download* melalui *http://www.apachefrieng.org/en/xampp-windows.html*.

11. Adobe Dreamwever

Menurut Andi (2012:4) mengatakan bahwa:

> *Adobe Dreamwever* merupakan sebuah aplikasi untuk membuat berbagai *script web* yang sering disebut denga *web editor*. Kelebihan *Dreamwever* dibanding *notepad++* adalah pada *Dreamwever* dilengkapi dengan tampilan desain secara langsung tampa harus menyimpan file terlebih dahulu. Pada *Dreamwever*, anda tidak harus mengetik program tapi biasa hanya dengan mencari *property* yang anda inginkan untuk dipakai dalam mebuat *website*.

2.3 Metode Algoritma

Menurut Saniman dan Fathoni (2008:120) mengatakan bahwa: "Algoritma adalah susunan langkah-langkah sistematis dan logis dalam memecahkan suatu masalah".

Ada 3 cara dalam menyusun algoritma yaitu :

a. Dengan merumuskan langkah-langkah pemecahan masalah melalui kalimat yang terstuktur (tersusun secara logis)

b. Menggabungkan kalimat dengan penggalan statements yang ada di suatu bahasa pemrograman (misal : pascal).

c. Menggunakan diagram alir (*flowchart*) .

2.4 Pengujian Sistem

*Menu*rut Ayuliana (2009) mengatakan bahwa :

> Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari software. Karna itu uji coba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox*.

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

a. Fungsi-fungsi yang salah atau hilang

b. Kesalahan *interface*

c. Kesalahan dalam struktur data atau akses *database eksternal*

d. Kesalahan performa

e. kesalahan inisialisasi dan terminasi

Tidak seperti metode *whitebox* yang dilaksanakan diawal proses, ujicoba *blackbox* diaplikasikan dibeberapa tahapan berikutnya. Karena ujicoba *blackbox* dengan sengaja meng abaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi domain. Ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut :

a. Bagaimana validitas fungsionalnya diuji?

b. Jenis *input* seperti apa yang akan menghasilkan kasus uji yang baik ?

c. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?

d. Bagaimana batasan-batasan kelas data diisolasi?

e. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?

f. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Dengan mengaplikasikan ujicoba *blackbox*, diharapkan dapat menghasilkan se kumpulan kasus uji yang me*menu*hi kriteria berikut:

a. kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari ujikasus tambahan harus didesain untuk mencapai ujicoba yang cukup beralasan

b. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, dari pada kesalahan yang terhubung hanya dengan suatu ujicoba yang spesifik.

2.5 Peralatan Pendukung (*Tool System*)

Sistem pakar diagnose kerusakan komputer ini dirancang menggunakan UML dan dibuat ke dalam diagram-diagram yaitu *Activity*, *Use case*, *Sequence*,*Class* dan *Deployment*.

1. UML (*Unified Modelling Language*)

Menurut Sulistyorini (2009:23) mengatakan bahwa: "*Unified Modelling Lang-uage* (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 dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET.

1. Activity Diagram

Menurut Sulistyorini (2009:24) mengatakan bahwa "*Activity Diagram* bersifat dinamis. *Activity Diagram* adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek".

2. Use Case Diagram

Menurut Sulistyorini (2009:24) mengatakan bahwa: "*Use Case Diagram* bersifat statis.*Use Case Diagram* memperlihatkan himpunan *use case* dan aktor-*aktor* (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna".

3. Sequence Diagram

Menurut Haviludin (2011:6) mengatakan bahwa: "*squence diagram* menjelaskan intraksi objek yang di susun berdasarkan urutan waktu. Secara mudahnya *squence* diagram adalah gambaran tahap demi tahap, temasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*".

4. Class Diagram

Menurut Wibowo dan Herlwati (2011:37) mengatakan bahwa: " Diagram kelas adalah inti dari proses permodelan objek. Baik *forward engineering* maupun *reverse engineering* sebaliknya perubahan kode program menjadi model ".

5. Deployment Diagram

Menurut Sulistyorini (2009:24) mengatakan bahwa:

> *Deployment Diagram* bersifat statis. Deployment Diagram memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run time*). Dengan ini memuat simpul – simpul (*node*) beserta komponen – komponen yang ada di dalamnya. *Deployment diagra*m berhubungan erat dengan diagram kompoen dimana *deployment* diagram memuat satu atau lebih komponen– komponen. Diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed comput).*