BAB II

TINJAUAN PUSTAKA

A. Penelusuran Referensi

Penelitian tentang sebuah aplikasi informasi suatu tempat yang menggunakan navigasi sebagai penunjuk arah sudah banyak dilakukan diantaranya :

Penelitian yang dilakukan oleh Richard R F S , Kodrat Imam Satoto, Kurniawan Teguh Martonopada tahun 2012 dengan judul "Implementasi sistem informasi geografis daerah pariwisata kota semarang berbasis android dengan *Global Positioning system* (GPS)". Dalam tulisan ini menceritakan bagaimana wisata – wisata yang berada di daerah semarang serta bagaimana mempromosikan pariwisata kota semarang (Richarcd, Imam, & Teguh M, 2012, Implementasi sistem informasi geografis daerah pariwisata kota semarang berbasis android dengan *Global Positioning system* (GPS))

Penelitian yang dilakukan oleh Idris Rosadi pada tahun 2015 dengan judul "Aplikasi Panduan Wisata Di Bogor Berbasis Android". Dalam tulisan ini menceritakan tentang lokasi wisata yang terdapat di daerah bogor. (Rosadi, 2015, Aplikasi Panduan Wisata Di Bogor Berbasis Android)

Penelitian yang dilakukan oleh Eriza Siti Mulyani pada tahun 2010 dengan judul "Aplikasi *Location Based Service* (LBS) Taman Mini Indonesia Indah (TMII) Berbasis Android". Dalam tulisan ini bagaimana luasnya taman mini Indonesia indah sehingga membutuhkan aplikasi *Location based service*

agar memberi kemudahan bagi pengunjung. (Mulyani, 2013, Aplikasi *Location Based Service* (LBS) Taman Mini Indonesia Indah (TMII) Berbasis Android).

Penelitian yang dilakukan oleh Sendiana Hadi Sujatma pada tahun 2016 dengan judul Perancang Aplikasi Informasi Tempat Bersejarah Di Pulau Jawa Berbasis Android. Dalam tulisanya ini mendeskripsikan tentang beberapa informasi tempat yang bersejarah di pulau jawa. (Sujatma, 2016, Perancang Aplikasi Informasi Tempat Bersejarah Di Pulau Jawa Berbasis Android)

B. Android

Android adalah sebuah *platform* untuk sebuah perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middle ware* dan aplikasi. Android menyediakan *platform* tebuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *Google Inc*. membeli *Android Inc*. yang merupakan pendatang baru yang menciptakan piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk *Google, HTC, intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.*

Pada perilisan perdana android, 5 November 2007, android bersama *Open Handset Allliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat

seluler. (Safaat N, 2015, P*emrograman Aplikasi Mobile Smartphone Dari Tablet PC Berbasis Android : 17*)

C. Pengertian Aplikasi Android

Android merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile, i-Phone OS, Symbian*, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk *platform* mereka. (Hermawan, 2013, Mudah Membuat Aplikasi Android : 26).

D. Konsep Dasar Aplikasi Android

Android adalah *platform* pada *gadget* dan *handphone* yang kemampuannya hampir sama dengan pc, dapat mengolah data dan dapat menggunakan *internet* serta berkomunikasi menggunakan jaringan cellular seperti *handphone* pada umumnya.

E. Arsitektur Aplikasi Android

Arsitektur yang tersedia pada Android adalah:

1. *Applications*dan *widgets*

   Applications dan *widgets* ini adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien *email*, program SMS, kalender, peta, *browser*, kontak dan lain-lain. Semua aplikasi ditulis menggunakan pemrograman Java.

2. *Application Framework*

   Android adalah "*Open Development Platform*" yaitu android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

   Sehingga bisa kita simpulkan *applications framework* ini adalah*layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem

operasi android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termsuk di dalam *Applications frameworks* adalah sebagai berikut:

a. Views

b. Content Provider

c. Resource Maganer

d. Notification Manager

e. Activity Manager

3. *Libraries*

Libraries ini adalah *layer* di mana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas *kernel*, *layer* ini meliputi berbagai *library* C/C++ inti seperti *Libe* dan *SSL*, serta:

a. *Libraries* media untuk pemutaran media *audio* dan *video*

b. *Libraries* untuk manajemen tampilan

c. *Libraries Graphics* mencakup *SGL* dan *OpenGL* untuk grafis 2D dan 3D

d. *Libraries SQlite* untuk mendukung *database*

e. *Libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*

f. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embeded web view*

g. *Libraries* 3D yang mencakup implementasi *OpenGL ES 1,0 API's*

4. *Android Runtime*

Layer yang membuat aplikasi android dapat dijalankan di mana dalam prosesnya menggunakan implementasi *linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. Di dalam *Android Runtime* dibagi menjadi dua bagian yaitu:

a. Core Libraries: Aplikasi android dibangun dalam bahasa java, sementara *Dalvik* sebagai *virtual*mesinnya bukan *Virtual Machine* Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh Core *Libraries.*

b. *Dalvik Virtual Machine*: Virtual mesin berbasis *resister* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien , di mana merupakan pengembangan yang mampu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

5. *Linux Kernel*

Linux kernel adalah *layer* di mana inti dari *operating sistem* dari android itu berada. Berisi file-file sistem yang mengatur *system*

processing, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya. *Linux Kernel* yang digunakan android adalah *linux kernel release 2.6.*

F. Karakteristik Aplikasi Android

Menurut Safaat (2011), Android adalah sebuah kumpulan perangkat lunak untuk perangkat *mobile* yang mencakup *sistem operasi*, *middleware* dan aplikasi utama *mobile*. Android memiliki 4 (empat) karakteristik sebagai berikut:

1. Terbuka

   Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain. Android menggunakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan *open source*, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. *Platform* ini akan terus berkembang untuk membangun aplikasi *mobile* yang inovatif.

2. Semua Aplikasi dibuat sama

   Android tidak memberikan perbedaan terhadap aplikasi utama dari telpon dan aplikasi pihak ketiga (*third-party application*). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap

kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

   Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari *web* dengan data pada ponsel seseorang seperti kontak pengguna, kalender, atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

   Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan *library* yang di pergunakan *tools* yang dapat digunakan untuk membangun aplikasi yang semakin baik. Android memiliki sekumpulan *tools* yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Google Inc. Sepenuhnya membangun Android dan menjadikan bersifat terbuka (*open source*) sehingga para pengembang dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari *Google* dan dapat membangun Android tanpa adanya batasan-batasan. *Android Software Development Kit* (SDK) menyediakan alat dan *Application Programming Interface* (API) yang diperlukan untuk mulai mengembangkan apliaksi pada *platform Android* menggunakan bahasa pemrograman *java*.

G. Versi Android

Adapun Versi–versi android yang pernah dirilis adalah sebagaiberikut:

1. Android Versi 1.1

   Pada 9 Maret 2009, *Google* merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan *estetis* pada aplikasi, jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan gmail dan pemberitahuan email.

2. Android Versi 1.5 (*Cupcake*)

   Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan android dan sdk (*software development kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, meng-upload video ke youtube dan gambar ke *picasa* langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan tehubung secara otomatis ke *headset Bluetooth, animal layar*, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android Versi 1.6 (*Donut*)

   Donut (versi 1.6) dirilis pasa September dengan menampilkan proses pencarian yang lebih baik di banding sebelumnya, penggunaan baterai indikator dan kontrol *applet VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus,

kamera, *camcorder* dan galeri yang diitegrasikan, *CDMA/EVDO, VPN*, *Gestures* dan *Text to speech engine.*

4. Android Versi 2.0 / 2.1 (*Eclair*)

   Pada 3 Desember 2009 kembali diluncurkan ponsel android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan *Google Maps 3.1.2*, perubahan *UI* dengan browser baru dengan dukungan *HTML5.*

5. Android Versi 2.2 (*Froyo: Frozen Yoghurt*)

   Pada bulan Mei 2010 Android versi 2.2 diluncurkan. Android inilah yang sekarang sangat banyak beredar di pasaran.

6. Android Versi 2.3 (*Gingerbread*)

   Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya.

7. Android Versi 3.0 (*Honeycomb*)

   Dirilis Februari 2011 sebagai android 3.0 revisi 1 serta android versi 3.0 revisi 2 telah dirilis pada juli 2011.

8. Android Versi 3.1 (*Ice Cream Sandwich*)

   Diumumkan secara resmi pada 10 Mei 2011 diajang *Google I/O Developer Conference (San Francisco)*, pihak *Google* mengklaim "*Android Ice Cream Sandwich*" akan dapat digunakan baik di *smartphone* maupun *tablet*

9. Android Versi 4.0 (*Jelly Bean*)

Android versi 4.1 (*Jelly Bean*), Android *Jelly Bean* yang diluncurkan pada acara *Google I/O* lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, *UI* yang baru dan pencarian melalui *Voice Search* yang lebih cepat.

10. Android Versi 4.4 (*Kitkat*)

    Sampai dengan v4.4.4 Dirilis pertama pada tanggal 31 bulan Oktober tahun 2013 di namakan dengan *Android kitkat*. OS android *kitkat* memiliki tampilan 100% lebih *dinamis* dan berbeda total dengan android *jelly bean*,android *kitkat* di optimasi pada sisi konsumsi baterai dan kinerja OS lebih cepat ketika dijalankan pada perangkat memiliki spesifikasi lebih rendah,seperti kita tahu jika andorid *jelly bean* memiliki kelebihan pada sisi konsumsi baterai yang lebih tinggi dan ketika di jalankan di perangkat yang memiliki versi rendah os ini tidak maksimal.

11. AndroidVersi 5.0(*Lollipop*)

    Google akhirnya secara resmi merilis sistem operasi Android versi 5.0 *Lollipop*. Secara teknis, Android *Lollipop* diklaim sebagai update sistem operasi terbesar yang pernah dilakukan *Google*. Konon, perangkat yang menggunakan OS Android *Lollipop* ini akan mampu berintegrasi antar perangkat seperti *smartphone, tablet* dan *smartwatch* berbasis Android. Perangkat *smartphone* yang beruntung mencicipi sistem operasi ini untuk pertama kali adalah dari keluarga baru *Nexus*

yang akan segera di luncurkan dalam waktu dekat ini, antara lain *Nexus 6* dan *Nexus 9*. (Rosadi, 2015, Aplikasi Panduan Wisata Di Bogor Berbasis Android : 12)

H. Aplikasi

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan tertentu (khusus).

Aplikasi secara umum adalah suatu proses dari cara yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Aplikasi (*application*) juga bisa disebut sebagai perangkat lunak (*software*) yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, dan *Microsoft Excel* . (Rosadi, 2015, Aplikasi Panduan Wisata Di Bogor Berbasis Android : 19)

I. Hotel

Hotel adalah suatu bangunan yang dikelola secara komersil guna memberikan fasilitas penginapan kepada masyarakat umum dengan fasilitas antara lain jasa penginapan, pelayanan barang bawaan, pelayanan makanan dan minuman, penggunaan fasilitas perabot dan hiasan-hiasan yang ada di dalamnya serta jasa pencucian pakaian. (Endar Sri, 2014, Pengertian Hotel Dan Jenis Hotel).

J. Ponorogo

Ponorogo adalah sebuah kabupaten di provinsi jawa timur. Kabupaten ini terletak di koordinat 111° 17’ - 111° 52’ BT dan 7° 49’ - 8° 20’ LS dengan ketinggian antara 92 sampai 2.563 meter diatas permukaan laut dan memiliki luas wilayah 1.371,78 km². Kabupaten ini terletak disebelah barat dari provinsi Jawa Timur dan berbatasan langsung dengan provinsi Jawa Tengah atau lebih tepatnya 220 km arah barat daya dari ibu kota provinsi Jawa Timur, Surabaya. Pada tahun 2015 berdasarkan hasil Sensus Penduduk, jumlah penduduk Kabupaten Ponorogo adalah 986.224 jiwa.

K. Eclipse

Eclipse yang diluncurkan oleh *IBM* pada tanggal 5 November 2001 merupakan sebuah *IDE* yang gratis dan *open source* atau yang dapat dikembangkan dan digunakan untuk membangun sebuah program komputer dan dapat dijalankan di semua *platform*. (Satyaputra & Aritonang, 2013, *Java For Beginners with Eclipse 4.2 juno : 15*)

Eclipse menggunakan bahasa pemrograman *XML (Extensible Markup Language)* yang mana *XML* adalah bahasa *markup* yang digunakan untuk menyimpan data (tidak ada program) dan tidak tergantung dengan *tools* tertentu (seperti *editor*, *dbms*, *compiler*, dsb). *XML* merupakan suatu bahasa *Markup*. *Markup* yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu untuk memformat dokumen teks dengan tag

sendiri agar dapat dimengerti. Pada android *XML* digunakan untuk merancang *interface* pada sebuah program yang akan dibuat.

Sejak tahun 2006, *Eclipse Foundation* mengkoordinasikan peluncuran *Eclipse* secara rutin dan simultan yang dikenal dengan nama *Simultaneous Release*. Setiap versi peluncuran terdiri dari *Eclipse Platform* dan juga sejumlah proyek yang terlibat dalam proyek *Eclipse*. (Noviani, 2015, Pengenalan tentang sejarah singkat mengenai *eclipse* : 4)

Tujuan sistem ini adalah untuk menyediakan distribusi *Eclipse* dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah *deployment* dan *maintenance* untuk sistem *enterprise*, serta untuk kenyamanan. Peluncuran *simultan* dijadwalkan pada bulan Juni setiap tahunnya. (Noviani, 2015, Pengenalan tentang sejarah singkat mengenai *eclipse* : 4)

Pada saat ini, *Eclipse* merupakan salah satu *IDE* favorit karena gratis dan *open source. Open source* berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari *Eclipse* yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut *plug-in*. (Noviani, 2015, Pengenalan tentang sejarah singkat mengenai *eclipse* : 2)

Eclipse awalnya dikembangkan oleh *IBM* untuk menggantikan perangkat lunak pengembangan *IBM Visual Age for Java 4.0. Produk Eclipse* ini diluncurkan oleh *IBM* pada tanggal 5 November 2001. *IBM* menginvestasikan *US$ 40* juta untuk pengembangannya. Sejak 5 November

2001, *konsorsium Eclipse Foundation* mengambil alih pengembangan *Eclipse* lebih lanjut. (Noviani, 2015, Pengenalan tentang sejarah singkat mengenai *eclipse* : 2)

Sejak versi 3.0, *Eclipse* pada dasarnya merupakan sebuah *kernel*. Apa yang dapat digunakan di dalam *Eclipse* sebenarnya adalah fungsi dari *plug-in* yang sudah dipasang (diinstal). Ini merupakan basis dari *Eclipse* yang dinamakan *Rich Client Platform (RCP)*. Berikut ini adalah komponen yang membentuk *RCP* :

1. *Core platform*
2. *OSGi*
3. *SWT (Standard Widget Toolkit)*
4. *Jface*
5. *Eclipse Workbench*

Secara standar *Eclipse* selalu dilengkapi dengan *JDT (Java Development Tools), plug-in* yang membuat *Eclipse* kompatibel untuk mengembangkan program *Java*, dan *PDE (Plug-in Development Environment)* untuk mengembangkan *plug-in* baru. *Eclipse* beserta *plug-in* nya diimplementasikan dalam bahasa pemrograman *Java*.

Konsep *Eclipse* adalah *IDE* adalah terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja dan tidak untuk sesuatu yang spesifik.

Eclipse tidak saja untuk mengembangkan program *Java*, tetapi juga untuk berbagai macam keperluan. Perluasan apapun cukup dengan menginstal *plug-in* yang dibutuhkan.

Apabila ingin mengembangkan program *C/C++* maka telah terdapat *plug-in CDT (C/C++ Development Tools)* yang dapat dipasang di *Eclipse* untuk *Eclipse* menjadi perangkat untuk pengembangan *C/C++*.

Pengembangan secara visual bukan hal yang tidak mungkin oleh *Eclipse, plug-in UML2* tersedia untuk membuat diagram *UML*. (Noviani, 2015, Pengenalan tentang sejarah singkat mengenai *eclipse* : 4)

L. JAVA

Java merupakan bahasa pemrograman yang dikembangkan dengan menggunakan bahasa C, sehingga pengembang (*programmer*) C tidak megalami kesulitan beralih ke *Java*. (Supardi, 2014, *Semua Bisa Menjadi Programmer Android : 23*)

Java diciptakan oleh James Gosling dan Patrick Naughton dalam suatu projek dari *Sun Microsystems* sekitar tahun 1991. Pada mulanya ingin diberi nama *OAK* (berasal dari nama pohon yang terdapat pada kantor James Gosling), tetapi karena kata *OAK* telah ada pada *Sun Microsystems*, maka diberi nama *Java* (dari inspirasi minum kopi). (Supardi, 2014, *Semua Bisa Menjadi Programmer Android : 25*).

Pertama kali *Java* dirilis (dikeluarkan) disebut JDK (*Java Development Kit*), hingga versi *Java 1.1*. mulai Java 1.2 *Sun Microsystems* menyebutnya JSDK (*Java Software Defelopment Kit*) atau *Java2*. Mulai *Java2* ini juga lingkungan eksekusi dipisahkan dengan nama JRE (*Java Runtime Environment*). JRE termasuk juga dalam JVM (*Java Virtual Machine*). JVM

merupakan inti dari teknologi *Java*, sehingga bahasa *Java* dapat dibaca pada mesin komputer tententu (Supardi, 2014, *Semua Bisa Menjadi Programmer Android : 29*)

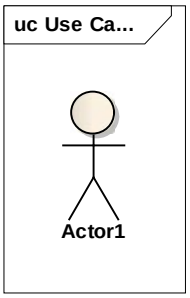
M. *Use Case Diagram*

Use case diagram merupakan sekelompok simbol dan gambar yang menunjukkan keadaan bagaimana sistem digunakan, apa yang dapat pengguna lakukan terhadap sistem. *Use case* diagram tidak membahas bagaimana fungsi dalam pemrograman berkerja, pada intinya *use case* diagram menggambarkan interaksi antara pengguna dengan sistem, *use case* diagram mendeskripsikan suatu skenario penggunaan yang bersifat spesifik dalam bahasa yang jelas dan dapat dimengerti dari sudut pandang pengguna (Pressman, 2013).

Dalam beberapa kasus *use case diagram* tidak selalu diperlukan namun penjelasan melalui bentuknya dapat menghasilkan pemahaman yang lebih baik terhadap sistem. Beberapa simbol yang digunakan pada *use case diagram* yaitu:

1. *Actor* (Pengguna)

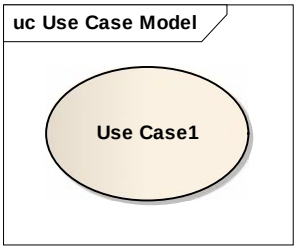
   Actor merupakan simbol dari pengguna yang disertakan ke dalam *use case diagram* yang disesuaikan dengan kebutuhan sistem. Pada Gambar 1 menunjukkan simbol *actor* dalam *use case diagram.*



Gambar 1 Simbol *actor* dalam *use case*

2. *Use Case*

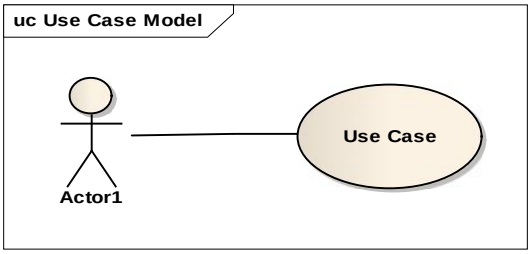
Use case adalah inti simbol dalam *use case* diagram, yang menggambarkan interaksi pengguna dengan sistem yang digambarkan dalam *use case* diagram. Pada Gambar 2 menunjukkan simbol *use case* dalam *use case* diagram.



Gambar 2 Simbol dari *use case*

3. *Association*

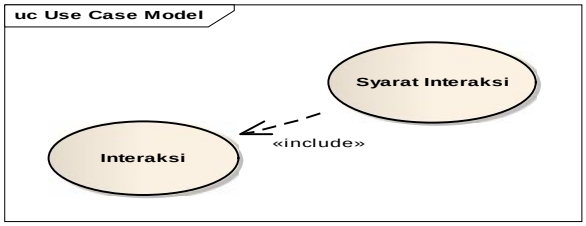
Association adalah garis yang menghubungkan antara *actor* dengan *use case*, garis *association* menjelaskan siapa (*actor*) yang berinteraksi terhadap fungsi pada sistem. Pada Gambar 3 menunjukkan simbol *association* yang menghubungkan antara *actor* dengan *use case*.



Gambar 3 *Actor* dan *use case* dihubungkan dengan *association*

4. *Include*

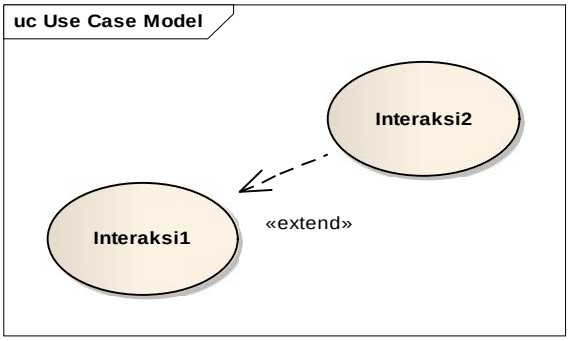
Include adalah garis putus-putus yang menghubungkan antara satu *use case* dengan *use case* lainnya yang menunjukkan bahwa suatu tindakan pengguna kepada aplikasi perlu disertai tindakan lainnya atau *include* juga dapat diartikan sebagai pemisahan interaksi yang terlalu luas untuk digambarkan hanya dengan 1 *use case* sehingga 1 *use case* yang terlalu luas dapat dibagi menjadi beberapa *use case* yang dihubungkan dengan garis *include* kepada *use case* yang utama. Pada Gambar 4 menunjukkan elemen *include* pada *use case* yang menghubungkan *use case* syarat interaksi dengan *use case* interaksi, pada Gambar 4 dicontohkan sebuah skema *use case* diagram yang menunjukkan agar perangkat lunak dapat menjalankan keinginan dari *actor* maka *actor* juga perlu melakukan interaksi terhadap *use case* syarat interaksi.



Gambar 4 *Include* penghubung antara syarat interaksi kepada interaksi

5. *Extends*

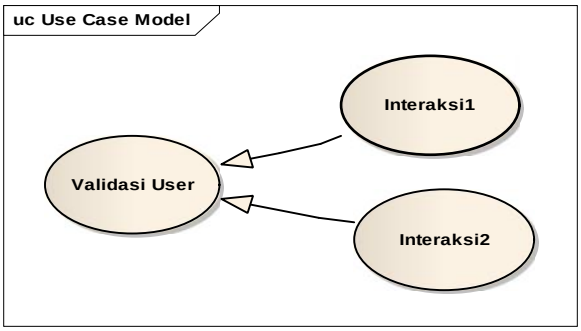
Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu. Gambar 5 menggambarkan bahwa *use case* interaksi 2 dapat berdiri sendiri walau tanpa *use case* interaksi 1.



Gambar 5 *Extends* pada *use case diagram*

6. *Generalization*

Hubungan generalisasi dan spesialisasi antara dua buah *use case*. Gambar 6 menunjukkan generalisasi dari *use case* interaksi 1 dan interaksi 2 dengan *use case* validasi *user*.

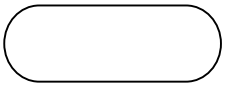
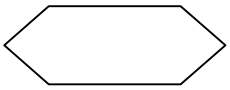
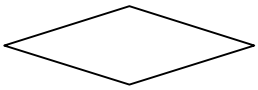


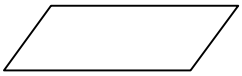
Gambar 6 *Generalization* pada *use case diagram*

N. *Flowchart*

Sistem *flowchart* dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urut-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Simbol-simbol yang digunakan antara lain :

Tabel 1 Simbol-simbol *flowchart.*

Simbol *Flowchart*	Fungsi
	**Terminal** Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses.
	***Preparation*** Digunakan untuk mempersiapkan nilai awal dari duatu variabel yang akan diproses dan digunakan untuk proses *loop.*
	***Decision*** Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.
	**Proses** Untuk menggambarkan suatu proses yang sedang dieksekusi.

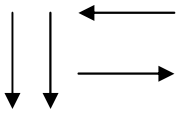


Input / Output

Digunakan untuk menggambarkan proses *input (read)* maupun proses *output (print).*

Subroutine

Menggambarkan proses pemanggilan subprogram dari mainprogram.



Flow line

Merupakan gambaran arus proses dari suatu kegiatan ke kegiatan lain.

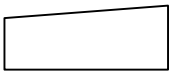
Connector

Sebagai penghubung antara suatu proses dengan proses lainnya yang ada pada suatu lembar halaman.

Page connector

Sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berbeda lembar halaman.

Manual operation

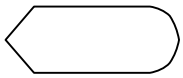


Simbol yang digunakan untuk menggambarkan suatu kegiatan atau proses yang bersifat manual.

Printer

Menggambarkan suatu dokumen atau suatu kegiatan mencetak suatu informasi.

Console

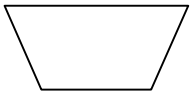


Simbol yang digunakan untuk menampilkan data atau informasi melalui monitor atau CRT (*Cathode Ray Tube)*

Disk

Untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk.*

Manual input



Menggambarkan proses pemasukan data melalui manual.

Tape

Untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk.*

Sumber : (Tata Sutabri, S.Kom, 2004)

O. Google Maps API

Sebelum membahas *Google API*, harus memahami dulu tentang API. API adalah kependekan dari *Aplication Programing Interface*. Dengan bahasa yang lebih sederhana API adalah fungsi pemrograman yang disediakan oleh aplikasi atau layanan agar layanan tersebut bisa di integerasikan dengan aplikasi yang kita buat. (Putra A. Candra, (2014), Pengantar *Google Maps API*)

Jadi *Google Maps API* adalah fungsi-fungsi pemrograman yang disediakan oleh *Google Maps* agar *Google Maps* bisa di integrasikan ke dalam web atau aplikasi yang sedang dibuat. (Putra A. Candra, (2014), Pengantar *Google Maps API*)

Google Maps API ini bisa dipakai secara gratis atau *Open Source* tanpa perlu mengeluarkan biaya untuk lisensi. Hanya saja ada pembatasan request peta pada satu hari yaitu 2500, jika ingin lebih dari itu bisa membeli lisensi *Google Maps API for bisnis*. (Putra A. Candra, (2014), Pengantar *Google Maps API*)

Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh *Google*, diantaranya

1. *Roadmap* adalah untuk menampilkan peta biasa dengan 2 dimensi.
2. *Satellite* adalah untuk menampilkan *satellite*

3. *Terrain* adalah untuk menunjukan *relief* fisik permukaan bumi dan menunjukan seberapa tingginya suatu lokasi contohnya gunung dan sungai.
4. *Hybrid* adalah untuk menunjukan foto *satelite* yang diatasnya tergambar pula apa yang tampil pada *roadmap*.