

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian mengenai sistem informasi geografis berbasis *android* telah banyak dilakukan oleh peneliti-peneliti lain sebelumnya. Penelitian yang terlebih dahulu dilakukan antar lain adalah penelitian yang dilakukan oleh Bambang Yuwono, Agus Ariwibowo dan Febry Setiawan pada tahun 2015. Penelitian ini mengambil judul Sistem Informasi Geografis Berbasis Android Untuk Pariwisata di Daerah Magelang. Penelitian ini mengambil studi kasus di daerah Magelang dan menggunakan metode *Grapple*. Penelitian ini menggunakan *Minimum Requirement Android version 2.2 Froyo*, dan hasil yang ditampilkan adalah wisata alam dan kuliner.

Penelitian berikutnya adalah penelitian yang dilakukan oleh Rudi Dwi Atmojo, Dharma Palekahelu, dan Ari Setiawan pada tahun 2013 dengan judul Sistem Informasi Geografis Objek Wisata di Kabupaten Boyolali Berbasis Android. Penelitian ini menggunakan metode *waterfall*. Penelitian ini menghasilkan sebuah aplikasi dengan *output* berupa lokasi objek wisata alam dan wisata kuliner yang ada pada kabupaten Boyolali, dan media penyimpanan data yang digunakan adalah *MySQL Database*.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Kartika Santoso dan Muhammad Rais pada tahun 2015, yang berjudul Implementasi Sistem Informasi Geografis Daerah Pariwisata Kabupaten Temanggung Berbasis *Android* dengan *Global Positioning System (GPS)*. Pada penelitian ini metode penelitian yang digunakan adalah metode *waterfall* dengan mengambil studi kasus di kabupaten Temanggung. Aplikasi yang dihasilkan dari penelitian ini adalah aplikasi berbasis *android* yang menggunakan *Global Positioning System (GPS)* untuk pencarian rute. Aplikasi juga akan menampilkan semua objek wisata yang ada di kabupaten Temanggung yang telah diinput oleh *admin*, tanpa membagi kategori objek wisata.

Penelitian serupa berikutnya adalah penelitian yang dilakukan oleh Rachmat Ramdani pada tahun 2015. Penelitian ini mengambil judul tentang Sistem Informasi Geografis Untuk Objek Wisata Di Jawa Barat Dengan Memanfaatkan Sistem Pendukung Keputusan Berbasis *Android*. Pada penelitian ini menambahkan sistem pendukung keputusan untuk menentukan objek wisata yang direkomendasikan kepada pengguna, berdasarkan 8 kriteria yang diinputkan oleh pengguna sendiri. Metode yang digunakan untuk mengambil keputusan menggunakan metode *Multi Criteria Decision Making (MCDM)*. Kriteria yang digunakan mencakup kondisi keuangan, jarak tempuh, tujuan wisata dan jenis daerah yang dikunjungi. Sedangkan metodologi penelitian yang digunakan adalah metode *waterfall*. Hasil dari penelitian ini adalah sebuah aplikasi berbasis *android* yang menampilkan objek wisata serta merekomendasikan tujuan wisata kepada pengguna berdasarkan 8 kriteria yang dimasukkan oleh pengguna aplikasi. Pembuatan aplikasi sendiri menggunakan *software Eclipse*, peta digital yang digunakan adalah *Google Map*, dan media penyimpanan data yang digunakan adalah *SQLite*.

Penelitian serupa yang lainnya adalah penelitian yang dilakukan oleh Oki Ria Hermawan dan Harjono. Penelitian yang dilakukan pada tahun 2016 ini mengambil judul Sistem Informasi Geografis Tempat Wisata di Kabupaten Banyumas Berbasis *Android*. Pada penelitian ini peneliti menggunakan metode *waterfall* sebagai metode penelitian yang digunakan. Hasil dari penelitian ini adalah sebuah aplikasi berbasis *android* yang menampilkan koordinat lokasi objek wisata di kabupaten banyumas. Selain aplikasi berbasis *android*, penelitian ini juga menghasilkan sebuah *website* yang menjadi *web server* dari aplikasi di perangkat *android*. Peta yang digunakan adalah *Google Map* dan media penyimpanan data yang digunakan adalah *MySQL*.

Tabel 2.1 Perbandingan Penelitian

No.	Nama Peneliti, Tahun	Judul Penelitian	Metode	Hasil
1.	Bambang Yuwono, Agus Ariwibowo dan Febry Setiawan, (2015).	Sistem Informasi Geografis Berbasis Android Untuk Pariwisata di Daerah Magelang	<i>Grapple</i>	Aplikasi <i>android</i> yang menampilkan objek wisata di daerah Magelang.
2.	Rudi Dwi Atmojo, Dharma Palekahelu, dan Ari Setiawan, (2013).	Sistem Informasi Geografis Objek Wisata di Kabupaten Boyolali Berbasis Android	<i>Waterfall</i>	Aplikasi <i>android</i> yang menampilkan objek wisata di kabupaten Boyolali.
3.	Kartika Santoso dan Muhammad Rais, (2015).	Sistem Informasi Geografis Daerah Pariwisata Kabupaten Temanggung Berbasis <i>Android</i> dengan <i>Global Positioning System (GPS)</i> .	<i>Waterfall</i>	Aplikasi <i>android</i> yang menampilkan objek wisata di kabupaten Temanggung, dan menggunakan <i>GPS</i> untuk menentukan rute dan jaraknya.

4.	Rachmat Ramdani, (2015).	Sistem Informasi Geografis Untuk Objek Wisata Di Jawa Barat Dengan Memanfaatkan Sistem Pendukung Keputusan Berbasis <i>Android</i>	<i>Waterfall</i> , dan <i>Multi Criteria Decision Making</i>	Aplikasi yang merekomendasikan objek wisata di Jawa Barat, berdasarkan 10 kriteria yang diinput oleh pengguna aplikasi.
5.	Oki Ria Hermawan dan Harjono, (2016).	Sistem Informasi Geografis Tempat Wisata di Kabupaten Banyumas Berbasis Android	<i>Waterfall</i>	Aplikasi berbasis <i>android</i> yang menampilkan lokasi objek wisata di kabupaten Banyumas, dan <i>website</i> sebagai <i>web server</i> yang nantinya akan dikelola oleh admin.
6.	Emanuel Fernandez, (2019).	Sistem Informasi Geografis Objek Wisata Di Kabupaten Flores Timur Berbasis Android.	<i>SDLC</i>	Aplikasi berbasis <i>android</i> yang menampilkan informasi termasuk letak geografis dari pada objek wisata serta akomodasi di Kabupaten Flores Timur

Merujuk pada penelitian yang dilakukan oleh Oki Ria Hermawan dan Harjono pada tahun 2016, dengan judul Sistem Informasi Geografis Tempat Wisata di Kabupaten Banyumas Berbasis Android, maka penelitian kali ini menggunakan metode *Systems Development Life Cycle (SDLC)* dengan menambahkan pembagian kategori wisata berdasarkan jenis wisata yang ingin dituju. Penelitian kali ini pula dilakukan di Kabupaten Flores Timur. Peta yang digunakan adalah *Google Map* untuk menampilkan lokasi objek wisata di Kabupaten Flores Timur. Maka penelitian kali ini dilakukan dengan mengambil judul Sistem Informasi Geografis Objek Wisata Di Kabupaten Flores Timur Berbasis Android.

2.2 Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) adalah ilmu pengetahuan yang berbasis pada perangkat lunak komputer yang digunakan untuk memberikan bentuk digital dan analisa terhadap permukaan geografi bumi sehingga membentuk suatu informasi keruangan yang tepat dan akurat (Suryantoro, 2013).

SIG merupakan sebuah sistem informasi yang didesain untuk bekerja dengan sumber data spasial. SIG merupakan suatu media yang sangat handal untuk merepresentasikan data *Remote Sensing (RS)* menjadi informasi yang berguna bagi banyak pihak untuk berbagai keperluan (Indarto, 2013).

2.2.1 Definisi SIG

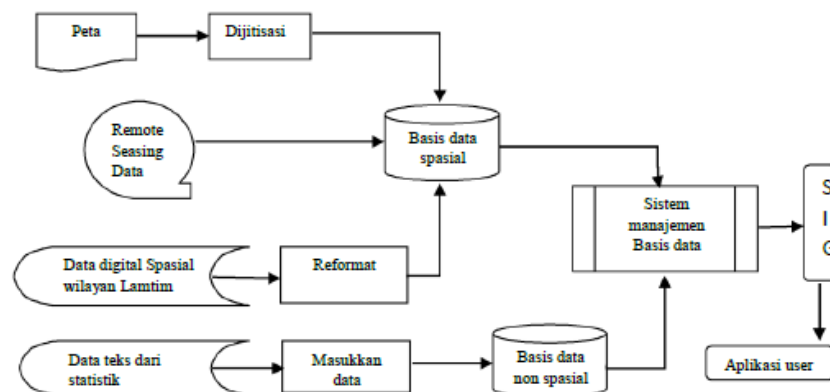
Istilah *geography* digunakan karena SIG dibangun berdasarkan pada geografi atau spasial. Geografi atau spasial ini mengarah pada spesifikasi lokasi dalam suatu *space*. SIG merupakan sistem komputer yang berbasis pada sistem informasi yang digunakan untuk memberikan bentuk digital dan analisis terhadap permukaan geografi bumi. Geografi adalah informasi mengenal permukaan bumi dan semua obyek yang berada di atasnya, sedangkan sistem informasi geografis (SIG) adalah sistem informasi khusus yang mengelola data

yang memiliki informasi spasial (bereferensi keruangan). Sistem informasi geografis merupakan bentuk sistem informasi yang menyajikan informasi dalam bentuk grafis dengan menggunakan peta sebagai antarmuka. SIG tersusun atas konsep beberapa lapisan (*layer*) dan relasi (Prahasta, 2002).

Maka SIG juga dapat dikatakan sebagai suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objek-objek yang terdapat di permukaan bumi. Jadi, SIG merupakan sejenis perangkat lunak, perangkat keras, manusia, prosedur, basis data, dan fasilitas jaringan komunikasi yang digunakan untuk memfasilitasi proses-proses pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran data/informasi geografis berikut atribut-atributnya (Prahasta, 2014).

2.2.2 Pengintegrasian SIG

Integrasi aplikasi SIG adalah suatu inisiatif yang diperlukan untuk mengupayakan secara efektif tentang pengumpulan, manajemen, akses, penyampaian, dan utilisasi data spasial maupun nonspasial, maka integrasi aplikasi SIG merupakan segala sesuatu yang terkait dengan membangun fasilitas dan mengembangkan koordinasi dalam melakukan pertukaran dan berbagai pakai data geospasial maupun nonspasial (Suryantoro, 2013).



Gambar 2.1 Pengintegrasian Aplikasi SIG (Suryantoro, 2013)

2.2.3 Sub Sistem SIG

Dijelaskan oleh Eddy Prahasta tahun 2014, SIG dapat diuraikan menjadi beberapa subsistem sebagai berikut (Prahasta, 2014).

- a. *Data Input*: mengumpulkan, mempersiapkan dan menyimpan data spasial & atributnya. Subsistem ini bertanggung jawab dalam mengkonversi format data aslinya ke dalam format SIG-nya.
- b. *Data Output*: menampilkan & menghasilkan keluaran basisdata spasial *softcopy & hardcopy* seperti halnya tabel, grafik, *report*, peta dan lain sebagainya.
- c. *Data Management*: mengorganisasikan data spasial & tabel atribut ke dalam sistem basisdata hingga mudah untuk dipanggil kembali, di-*Update*, dan di-*edit*.
- d. *Data Manipulation & Analysis*: menentukan informasi yang dihasilkan oleh SIG. selain itu, subsistem ini memanipulasi dan memodelkan data untuk menghasilkan informasi yang diharapkan.

2.2.4 Komponen SIG

SIG merupakan suatu sistem komputer yang terintegrasi di tingkat fungsional dan jaringan. Secara rinci SIG tersebut dapat beroperasi membutuhkan komponen- komponen sebagai berikut (Suryantoro, 2013):

- Orang yang menjalankan sistem meliputi mengoperasikan, mengembangkan bahkan memperoleh manfaat dari sistem. Kategori orang yang menjadi bagian dari SIG ini beragam, misalnya operator, analis, *programmer*, basis data administrator, bahkan *stakeholder*.
- Aplikasi merupakan kumpulan dari prosedur-prosedur yang digunakan untuk mengolah data informasi.
- Data yang digunakan dalam SIG dapat berupa data grafis dan data atribut. Data grafis/spasial ini merupakan data yang merepresentasikan fenomena permukaan bumi yang memiliki referensi (koordinat) lazim berupa peta, foto udara, citra satelit dan

sebagainya atau hasil dari interpretasi data tersebut. Sedangkan data atribut misalnya data sensus penduduk, data survei, data statistik lainnya.

- Perangkat lunak SIG adalah program komputer yang dibuat khusus dan memiliki kemampuan pengolahan, penyimpanan, pemrosesan, analisis dan penayangan data spasial.
- Perangkat keras ini berupa seperangkat komputer, komputer sistem jaringan dengan server, komputer dengan jaringan global (internet) dan peripheral yang dapat mendukung pengoperasian perangkat lunak yang dipergunakan.

2.2.5 Fungsi SIG

Dijelaskan oleh Indarto tahun 2013, fungsi utama SIG adalah sebagai berikut (Indarto, 2013).

- Mengoleksi Data.
Data yang digunakan dalam SIG sering berasal dari berbagai tipe dan disimpan dengan cara yang berbeda. SIG menyediakan alat dan metode untuk mengintegrasikan data-data yang berbeda tersebut ke dalam sebuah format, sehingga data-data tersebut mudah untuk dibandingkan dan dianalisa. Sumber data SIG sebagian besar berasal dari hasil digitasi secara manual dan hasil *scanning* foto udara, peta kertas, atau data digital lain. Tetapi tidak menutup kemungkinan bahwa data satelit dapat juga dijadikan sebagai masukan untuk SIG.
- Memperbaharui dan mengelola *database*.
Setelah data dikoleksi dan diintegrasikan, SIG seharusnya mampu menyediakan fasilitas untuk menambah dan memelihara data. Manajemen data yang efektif memiliki arti yang cukup luas, yang mencakup aspek: keamanan, integrasi, penyimpanan dan pencarian data, serta kemampuan untuk pemeliharaan.

- Analisa Geografis.

Integrasi dan konversi data merupakan salah satu bagian dari tahap memasukkan data di dalam SIG. Langkah yang dibutuhkan selanjutnya adalah interpretasi dan analisa koleksi informasi tersebut secara kuantitatif dan kualitatif.

- Menampilkan dan mempresentasikan hasil.

Salah satu aspek yang menarik pada bidang teknologi SIG adalah bahwa: informasi yang beragam dapat ditampilkan sekaligus dalam suatu bidang gambar yang sama. Misalnya, data *Table* dan data grafik yang dihasilkan dari metode konvensional dapat dilengkapi dengan peta dan gambar tiga dimensi (3D) yang dihasilkan oleh SIG. Dalam hal ini, SIG akan menjadi alat komunikasi visual yang sangat mengagumkan karena tersedianya berbagai pilihan tampilan untuk menyajikan data yang telah diolah.

2.3 Pariwisata

2.3.1 Pengertian Pariwisata

Pariwisata adalah salah satu jenis industri baru yang mampu menghasilkan pertumbuhan ekonomi yang cepat dalam penyediaan lapangan kerja, peningkatan penghasilan, standar hidup serta menstimulasi sektor-sektor produktivitas lainnya. Sebagai sektor yang kompleks, pariwisata juga meliputi industri-industri klasik seperti kerajinan tangan dan cinderamata, penginapan, transportasi secara ekonomi juga dipancang sebagai industri (Wahab, 1985).

2.3.2 Pengertian Objek Wisata

Obyek wisata adalah salah satu komponen yang penting dalam industri pariwisata dan salah satu alasan pengunjung melakukan perjalanan (*something to see*). Di luar negeri obyek wisata disebut *tourist attraction* (atraksi wisata), sedangkan di Indonesia lebih dikenal dengan objek wisata (Wahab, 1985).

2.3.3 Pengertian Wisata

Wisatawan didefinisikan sebagai orang yang melakukan kegiatan wisata. Jadi menurut pengertian ini, “semua orang yang melakukan perjalanan wisata disebut “wisatawan” apapun tujuannya yang penting perjalanan itu bukan untuk menetap dan tidak untuk mencari nafkah di tempat yang dikunjungi”.

Menurut *IUOTO (International Union of Official Travel Organization)* dalam Damardjati (2001), kata *tourist* atau wisatawan haruslah diartikan sebagai ;

1. Orang yang bepergian untuk bersenang-senang (*pleasure*), untuk kepentingan keluarga, kesehatan dan lain sebagainya.
2. Orang-orang yang bepergian untuk kepentingan usaha.
3. Orang-orang yang datang dalam rangka perjalanan wisata walaupun mereka singgah kurang dari 24 jam (Undang-undang No. 9 tahun 1990).

2.4 Pariwisata Kabupaten Flores Timur

Segala urusan yang berkaitan dengan kepariwisataan, yang berada dalam kawasan Kabupaten Flores Timur merupakan tanggung jawab dari Dinas Kebudayaan dan Pariwisata Kabupaten Flores Timur. Instansi ini yang akan menangani semua urusan kebudayaan dan pariwisata di Kabupaten Flores Timur, termasuk pengembangan dan promosi, pembinaan sumber daya manusia, serta pengendalian kebudayaan dan pariwisata. Selain itu instansi ini bertanggung jawab atas semua izin akomodasi kepariwisataan di Flores Timur termasuk hotel, rumah makan, maupun tempat hiburan.

Adapun fungsi lainnya adalah sebagai berikut :

1. Penyusunan kebijakan teknis bidang pariwisata.
2. Pelaksanaan kebijakan promosi dan standarisasi pariwisata.
3. Perencanaan pembangunan, pengembangan dan promosi serta pengawasan objek daya tarik wisata.

4. Pelaksanaan *monitoring* dan evaluasi seni budaya.
5. Pelayanan umum di bidang budaya dan pariwisata.
6. Pelaksanaan rencana induk pengembangan pariwisata dan sumber daya manusia.
7. Pembinaan perizinan, pengawasan dan pengendalian usaha sarana dan jasa pariwisata.
8. Pelaksanaan penyuluhan dan pelatihan tenaga kerja pariwisata.
9. Pelaksanaan tugas lain yang diberikan atasan.

2.5 Objek Wisata di Kabupaten Flores Timur

Berdasarkan observasi yang dilakukan oleh Dinas Kebudayaan dan Pariwisata Kabupaten Flores Timur per-April 2019, data yang diperoleh berjumlah 28 tempat wisata terkemuka atau yang paling sering dikunjungi wisatawan, dan yang tersebar di daerah Kabupaten Flores Timur. Dan dikelompokkan menjadi 6 kategori wisata.

Berikut adalah data tempat objek wisata berdasarkan hasil observasi Dinas Kebudayaan dan Pariwisata Kabupaten Flores Timur :

Tabel 2.2 Daftar Objek Wisata

No.	Kategori	Nama Wisata	Jumlah
1.	Wisata Alam		
		Danau Waibelen (Danau Asmara)	
		Danau Kota Kaya	
		Pulau Konga	
		Pulau Waibalun	
		Padang Rumput Lamanabi	
		Pemandian Air Panas (Wai Platin)	6
2.	Wisata Pantai		
		Pantai Asam Satu Weri	
		Pantai Oa	
		Pantai Rako	

		Pantai Watotena	
		Pulau Pasir Meko	
		Pantai Kawaliwu	
		Pantai Ina Burak	7
3.	Wisata Budaya		
		Istana Raja Larantuka	
		Semana Santa	
		Prosesi Laut	
		Benteng Lohayong	4
4.	Wisata Belanja		
		Pasar Oka	1
5.	Wisata Kuliner		
		Asa Restaurant	
		Rumah Makan Senaren	2
6.	Wisata Rohani		
		Kapela Tuan Bediri Wureh	
		Kapela Tuan Ma	
		Kapela Tuan Ana	
		Gereja Khatedral Larantuka	
		Makam Mgr. Gabriel Manek	
		Kapela Tuan Meninu	
		Biara Trapist Lamanabi	
		Bukit Fatima	8
	TOTAL		28

Selain objek wisata, Dinas Kebudayaan dan Pariwisata Kabupaten Flores Timur juga melakukan observasi mengenai hotel serta penginapan yang ada di Kabupaten Flores Timur. Total ada 13 hotel termasuk

penginapan terkemuka dan yang paling sering dikunjungi wisatawan, dan yang tersebar di Kabupaten Flores Timur.

Berikut adalah data hotel dan penginapan berdasarkan hasil observasi Dinas Kebudayaan dan Kabupaten Flores Timur.

Tabel 2.3 Daftar Hotel

No.	Nama Hotel
1.	Asa Hotel & Restaurant
2.	Sunrise Hotel
3.	Panginapan Geo Permai
4.	Hotel Fortuna I
5.	Hotel Fortuna II
6.	Hotel Kartika
7.	Hotel Amoria
8.	Flores Cottage
9.	Larantuka Beach Apartements
10.	Gelekat Nara
11.	Hotel Tresna
12.	Hotel Lestari
13	Penginapan Budi Luhur

2.6 Google Map API

2.6.1 Application Programming Interface (API)

API (Application Programming Interface) bukan hanya satu *set class* dan *method* atau fungsi dan *signature* yang sederhana. *API* yang bertujuan utama untuk mengatasi ketidakpahaman dalam membangun *software* yang berukuran besar, berawal dari sesuatu yang sederhana sampai ke yang kompleks dan merupakan perilaku komponen yang sulit dipahami (Halim, 2011).

2.6.2 Definisi Google Map API

Melalui *Google Maps API*, pengguna mendapatkan *tools* pemrograman yang kuat untuk membuat peta yang interaktif, akses ke *database Google Earth* dan kesempatan untuk pengembangan antarmuka pengguna yang akan digunakan untuk melakukan operasi pencarian melalui *address geocoding*, visualisasi data geografis, menerapkan analisis dan konsultasi (Maldzhanski, 2015).

2.7 Android

2.7.1 Sejarah Android

Menurut (Safaat,2012) *Android* adalah sebuah sistem operasi yang sengaja diciptakan 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 perangkat ponsel pintar (*smartphone*). Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance (OHA)*, sebuah konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*.

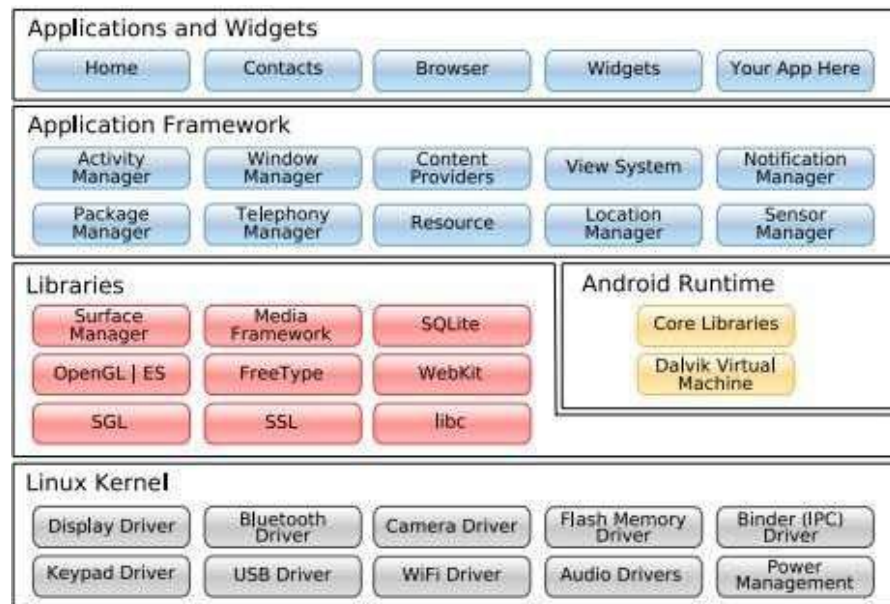
Pada saat perilisan perdana Android, 5 November 2007, *Android* bersama *OHA* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. Di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

Pada Juli 2005, *Google* bekerjasama dengan *Android Inc.*, perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri *Android Inc.* Bekerjasama pada *Google*, diantaranya Andy Rubin, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi *Android* hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa *Google* hendak memasuki pasar telepon seluler. Pada internal perusahaan *Google*, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan bahwa *Google* sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Sekitar September 2007, sebuah studi melaporkan bahwa *Google* mengajukan hak paten aplikasi telepon seluler. Akhirnya *Google* mengenalkan *Nexus One*, salah satu jenis telepon pintar *GSM* yang menggunakan *Android* pada sistem operasinya. Telepon seluler 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 Holding*, *Atheros Communications*, diproduksi oleh *Asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp.*, dan *Vodafone Group Plc*. Seiring pembentukan *OHA (Open Handset Alliance)*, *OHA* mengumumkan produk perdananya, *Android*, perangkat *mobile* yang merupakan modifikasi dari kernel Linux 2.6. Sejak *Android* dirilis, telah dilakukan pembaruan berupa pembaruan *bug* dan penambahan fitur baru. Pada penghujung tahun 2009, diperkirakan terdapat 18 jenis telepon seluler yang menggunakan sistem operasi *Android* (Safaat, 2012).

2.7.2 Arsitektur *Android*

Secara garis besar arsitektur android dapat dijelaskan dan digambarkan seperti yang terlihat pada gambar berikut :



Gambar 2.2 Arsitektur Android (Safaat, 2012)

1. *Application and Widgets*

Applications and Widgets ini adalah *layer* yang membatasi penggunaannya hanya untuk berhubungan dengan aplikasi saja, seperti halnya saat pengguna mengunduh aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut. Pada layer terdapat aplikasi ini termasuk klien *email*, program SMS, kalender, peta, *browser* atau kontak. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman Java.

2. *Application Framework.*

Android adalah “*Open Development Platform*”, yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang baik dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, serta lain sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang berkategori inti. Arsitektur

aplikasi dirancang supaya pengguna dapat dengan mudah menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga dapat dikatakan bahwa *Applications Framework* ini merupakan *layer* yang dapat digunakan pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang dijalankan di sistem operasi android, karena ada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content-provider* yang berupa SMS dan panggilan telepon. Komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut :

- a. *Views*.
- b. *Content Provider*.
- c. *Resource Manager*.
- d. *Notifications Manager*.
- e. *Activity Manager*.

3. *Libraries*.

Libraries ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel *Linux*, *layer* ini meliputi berbagai *library C/C++* inti seperti *Libc* dan *SSL*, serta :

- a. *libraries media* untuk pemutaran media audio dan video.
- b. *libraries* untuk manajemen tampilan.
- c. *libraries graphics* mencakup *SGL* dan *OpenSGL* untuk grafis 2D dan 3D.
- d. *libraries SQLite* untuk dukungan *database*.
- e. *libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*.
- f. *libraries LiveWebcote* mencakup modern *web browser* dengan *engine embedded web view*.
- g. *libraries 3D* yang mencakup implementasi *OpenGL ES 1.0 API's*.

4. *Android Run-Time.*

Android Run Time merupakan *layer* yang membuat aplikasi Android dapat dijalankan yang dalam prosesnya menggunakan implementasi *Linux*. *Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi Android. *Android Run Time* dibagi menjadi dua bagian, yaitu :

- a. *Core Libraries* yaitu aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai mesin virtualnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa Java/ C yang ditangani oleh *Core Libraries*.
- b. *Dalvik Virtual Machine*, yaitu mesin virtual berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dan merupakan pengembangan yang mampu membuat *Linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

5. Linux Kernel.

Linux Kernel adalah *layer* dimana inti dari operasi sistem dari Android itu berada. Berisi *file* sistem yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi Android lainnya (Safaat, 2012).

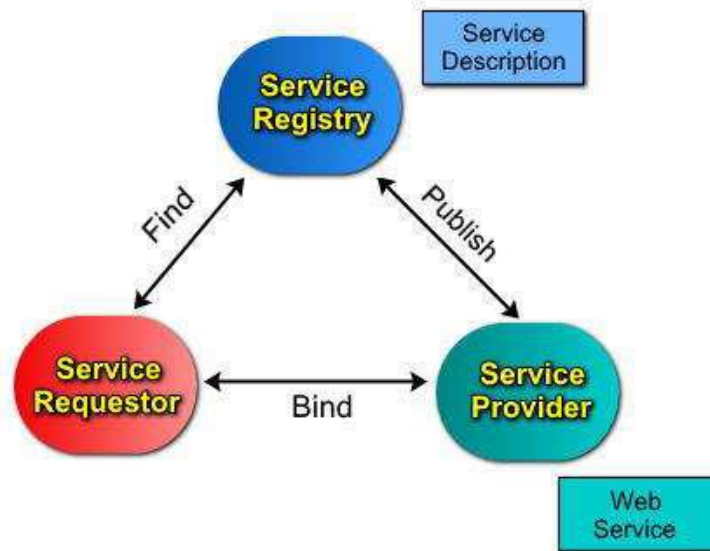
2.8 *Web Service*

2.8.1 *Pengertian Web Service*

Web service dapat diartikan sebagai sebuah antarmuka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet, dalam bentuk pesan (Yazdi, 2012). *Web service* menyimpan data informasi dalam format *JSON* atau *XML*, sehingga data ini dapat diakses oleh sistem

lain walaupun berbeda *platform*, sistem operasi, dan bahasa pemrograman (Rahman, dkk., 2013).

Berikut arsitektur *web service* secara umum (Kurniawan, 2014).



Gambar 2.3 Arsitektur Web Service (Kurniawan, 2014)

Pada gambar diatas ada 3 komponen utama dari *web service*, yaitu :

1. *Service Provider*, penyedia *web service* yang berfungsi menyediakan kumpulan *web services* yang dapat diakses oleh pengguna.
2. *Service Requestor*, aplikasi yang bertindak sebagai pengguna yang melakukan permintaan layanan (berupa *web services*) ke *service provider*.
3. *Service Registry*, tempat di mana *service provider* mempublikasikan layanannya. Pada arsitektur *web service*, *service registry* bersifat opsional.

2.8.2 *Javascript Object Notation (JSON)*

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran data. *JSON* terbuat dari dua struktur, yaitu kumpulan pasangan nama/nilai dan daftar nilai terurutkan (*an ordered list of values*) (Kasman, 2013).

2.9 *Database dan MySQL*

2.9.1 *Database*

Secara umum, *database* atau basis data berarti koleksi data yang saling terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusun data yang terstruktur yang disimpan dalam media penganal (*harddisk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat. Sesungguhnya ada beberapa jenis *database*, antara lain: *database* hierarkis, *database* jaringan, dan *database* relasional. *Database* relasional merupakan *database* yang populer saat ini dan telah diterapkan pada berbagai *platform*, dari *PC* hingga mini komputer. Sebuah *database* relasional tersusun atas sejumlah tabel (Kadir, 2008).

2.9.2 *MySQL*

Fungsi *MySQL* dapat dikatakan sebagai *interpreter query*, karena setiap menggunakan *query SQL* (perintah *SQL*) harus meletakkannya di dalam fungsi ini. Dengan kata lain, *SQL* tidak dapat dijadikan tanpa adanya fungsi *MySQL*. *MySQL* termasuk jenis

relational *database management system* (RDBMS). Sehingga istilah seperti tabel, baris dan kolom tetap digunakan dalam *MySQL*. Pada *MySQL*, *SQL* merupakan kependekan *Structured Query Language*. *SQL* digunakan untuk berkomunikasi dengan sebuah database. *SQL* adalah bahasa yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara, dan mengatur akses ke *database* serta digunakan untuk memanipulasi dan menampilkan data dari *database* (Puspitosari, 2011).

2.10 *Android Studio*

Android Studio adalah *Integrated Development Environment* (IDE) resmi dari *Android* untuk pengembangan aplikasi *Android*, diresmikan pada tanggal 16 Mei 2013 di *Google I/O conference*. *Android Studio* berbasis *IntelliJ IDEA* dari *JetBrains* dan menggunakan Bahasa *Java*. Selain *code editor* dan *tools* pengembangan *IntelliJ* yang kuat, *Android Studio* menawarkan lebih banyak fitur yang meningkatkan produktivitas ketika membangun aplikasi *Android*, seperti:

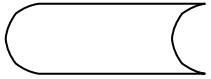
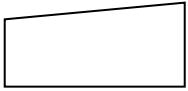
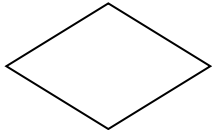
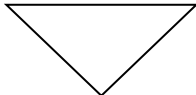
1. *Build system* berbasis *Gradle* yang fleksibel.
2. *Emulator* yang cepat dan kaya fitur.
3. Lingkungan terpadu yang dapat digunakan untuk mengembangkan aplikasi untuk semua perangkat *Android*.
4. Alat pengujian dan *framework* yang ekstensif.
5. *Instant Run* untuk menggabungkan perubahan pada aplikasi yang sedang berjalan tanpa membangun *APK* baru.
6. *Lint* untuk menangkap kinerja, kegunaan, kompatibilitas versi dan masalah lainnya.
7. *Template* kode dan integrasi *GitHub* untuk membantu membangun fitur aplikasi umum dan *import* contoh kode.
8. Mendukung *C++* dan *NDK*.
9. *Built-in support* untuk *Google Cloud Platform*.
(*developer.Android.com*).

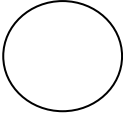
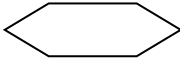
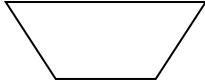
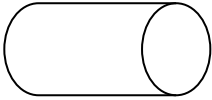
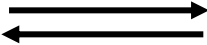
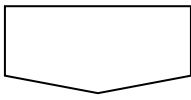
2.11 Diagram-diagram Perancangan Sistem

2.11.1 *Flowchart* (Diagram Alir)

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* Sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Dengan kata lain, *flowchart* ini merupakan deskripsi secara grafik dari urutan prosedur-prosedur yang terkombinasi yang membentuk suatu sistem. Data dan proses dalam *flowchart* sistem dapat digambarkan secara *online* (dihubungkan langsung dengan komputer) atau *offline* (tidak dihubungkan langsung dengan komputer, misalnya mesin tik, *cash register* atau kalkulator).

Tabel 2.4 Simbol-simbol *Flowchart*

No	Simbol	Keterangan
1.		Menyatakan permulaan atau akhir suatu program.
2.		Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
3.		Menyatakan input berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
4.		Memasukkan data secara manual.
5.		Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak.
6.		Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.

7.		Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
8.		Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
9.		Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
10.		Mencetak keluaran dalam bentuk dokumen cetak.
11.		Menyatakan input berasal dari kartu atau output ditulis ke kartu.
12.		Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
13.		Mencetak keluaran dalam layar monitor.
14.		<i>I/O</i> yang menggunakan pita kertas berlubang.
15.		<i>I/O</i> yang menggunakan drum maknetik.
16.		Menyatakan jalannya arus suatu proses.
17.		Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
18.		Basis data (<i>Database</i>).

2.11.2 *Unified Modelling Language (UML)*

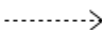
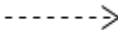
Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis 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 ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti *C++*, *Java*, *C#* atau *VB.NET*. Walaupun demikian, *UML* tetap dapat digunakan untuk modeling aplikasi prosedural dalam *VB* atau *C*.

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

1. *Use Case Diagram*

Use case diagram merupakan gambaran skenario dari interaksi antara pengguna dengan sistem. *Use case diagram* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.

Tabel 2.5 Simbol-simbol *Use Case Diagram*

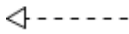
No.	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3.		<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>).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

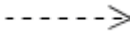
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. Class Diagram

Class Diagram merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.6 Simbol-simbol *Class Diagram*

No.	Gambar	Nama	Keterangan
1.		<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>).
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

3. Activity Diagram

Activity Diagram adalah representasi grafis dari seluruh tahapan alur kerja yang mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem.

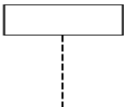
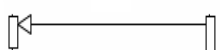
Tabel 2.7 Simbol-simbol Activity Diagram

No.	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* terhadap waktu.

Tabel 2.8 Simbol-simbol *Sequence Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.