

BAB II

LANDASAN TEORI

2.1 Pariwisata

Kata “pariwisata” sebenarnya baru populer di Indonesia setelah diselenggarakannya Msyawahar Nasional *Tourisme* di Tretes, Jawa Timur pada tanggal 12 s/d 14 juni 1958. Sebelumnya, sebagai kata pariwisata digunakan kata “*tourisme*” (bahasa Belanda) yang sering pula di Indonesia menjadi “*turisme*”.

Menurut pengertian ini, kata “pariwisata” yang berasal dari bahasa sangsekerta, sesungguhnya bukanlah berarti “*turisme*” (bahsa Belanda). Kata pariwisata menurut pengertian ini, sinonim dengan pengertian “*tour*”. Pendapat ini berdasarkan pemikiran sebagai berikut: kata pariwisata terdiri dari dua suku yaitu masing-masing kata “*pari*” dan “*sata*”.

- *Pari*, berarti banyak, berkali-kali, berputar-putar.
- *Wisata*, berarti perjalanan, berpergian yang dalam hal ini sinonim dengan kata “*travel*”.

Atas dasar itu, maka kata “pariwisata” seharusnya diartikan sebagai perjalanan yang dilakukan berkali-kali atau berputar-putar, dari suatu tempat ketempat lain, yang dalam bahasa Inggris disebut dengan kata “*tour*”, sedangkan untuk pengertian jamak , kata “kepariwisataan” dapat digunakan kata “*tourisme*” atau “*tourism*”.

Menurut ketentuan MPRS No.I – II Tahun 1960, kepariwisataan dalam dunia modern pada hakekatnya adalah suatu cara untuk memenuhi kebutuhan

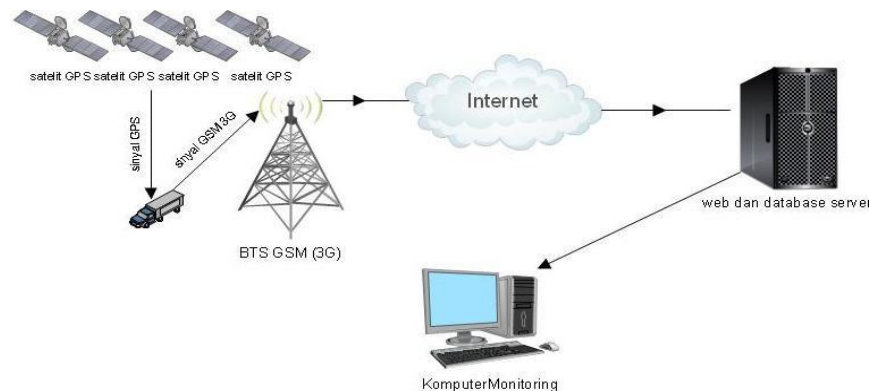
manusia dalam memberi liburan rohani dan jasmani setelah beberapa waktu bekerja serta mempunyai modal untuk melihat-lihat daerah lain baik dalam negeri maupun luar negeri. Unsur-unsur yang diperlukan calon wisatawan dalam rangka mengambil keputusan untuk melakukan perjalanan wisata:

Kemana hendak pergi?	Daerah tujuan wisata mana yang menjadi pilihan, dalam kota, luar kota, dalam negeri, atau keluar negeri.
Dimana akan menginap?	Kalau di hotel, hotel mana, berkemah dimana, rumah rakyat dimana.
Apa yang dapat dilakukan ditempat tujuan?	Fasilitas apa yang tersedia, olah raga, memancing, museum, upacara adat, keadaan geografis, dan lain-lain.
Kendaraan apa yang digunakan?	Memakai pesawat udara, kapal laut, kereta api, kendaraan pribadi, atau kendaran sewaan.
Dimaana pendaftaran dilakukan?	Travel agent atau <i>tour</i> operator mana yang menyelenggarakan kegiatan pariwisata.

2.2 Sistem Navigasi Satelit

Sistem navigasi satelit adalah system yang digunakan untuk menentukan posisi di Bumi, dengan menggunakan satelit. Sistem navigasi satelit mengirimkan data posisi (garis bujur dan lintang, dan ketinggian) dan sinyal waktu dari satelit, ke alat penerima di permukaan. Penerima di permukaan dapat mengetahui

posisinya, serta waktu yang tepat. (Wikipedia, 2010). Gambar 2.1 merupakan gambaran simulasi navigasi menggunakan satelit GPS.



Gambar 2.1. Simulasi Navigasi satelit Menggunakan Satelit GPS

Pada tahun 2007, sistem navigasi satelit yang berfungsi hanyalah NAVSTAR *Global Positioning System* (GPS) Amerika Serikat. GLONASS, sistem navigasi satelit Rusia sedang berada pada tahap perbaikan, dan diperkirakan akan selesai pada tahun 2010. Uni Eropa sedang dalam tahap meluncurkan sistem navigasi satelit baru bernama Galileo yang dijadwalkan selesai pada tahun 2013. Sistem navigasi satelit lain yang sedang dikembangkan adalah Beidou milik RRC dan IRNSS buatan India.

2.2.1 Cara Kerja

Sistem ini menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyalnya ke bumi dan ditangkap oleh sebuah alat penerima. Ada tiga bagian penting dari sistem ini, yaitu bagian kontrol, bagian angkasa, dan bagian pengguna.

1. Bagian Kontrol, bagian ini untuk mengontrol setiap satelit yang berada sedikit diluar orbit, sehingga bagian ini melacak orbit satelit, lokasi, ketinggian, dan kecepatan. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, dikoreksi, dan dikirimkan kembali ke satelit. Koreksi data lokasi yang tepat dari satelit ini disebut dengan data ephemeris, yang nantinya akan di kirimkan kepada alat navigasi kita.
2. Bagian Angkasa, Bagian ini terdiri dari kumpulan satelit-satelit yang berada di orbit bumi, sekitar 12.000 mil diatas permukaan bumi.

Kumpulan satelit-satelit ini diatur sedemikian rupa sehingga alat navigasi setiap saat dapat menerima paling sedikit sinyal dari empat buah satelit.

3. Bagian Pengguna, Bagian ini terdiri dari alat navigasi yang digunakan.

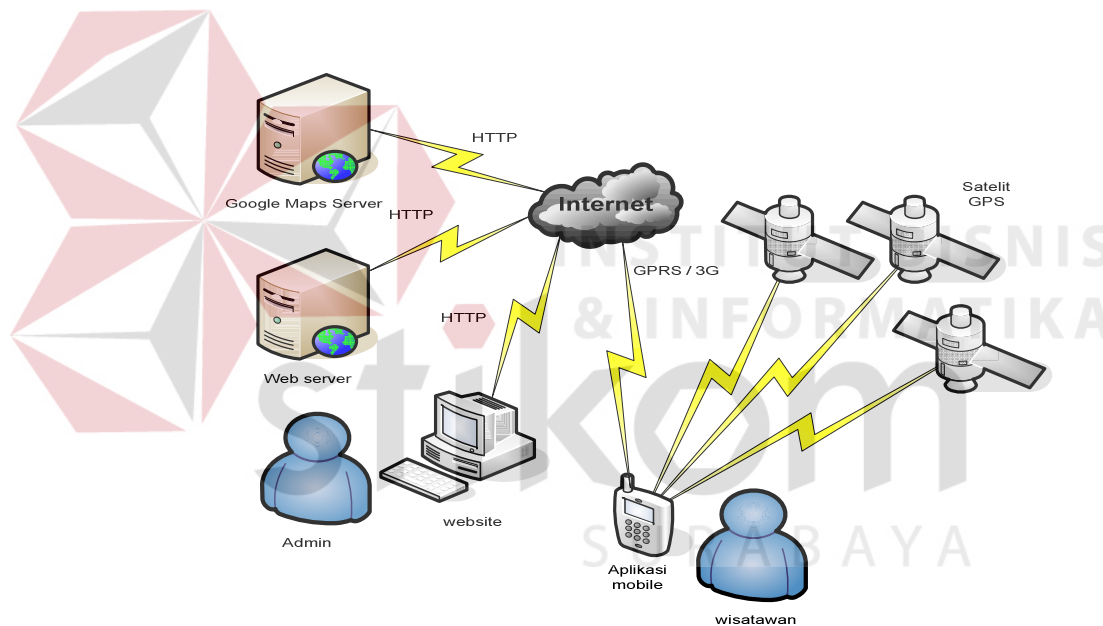
Karena alat navigasi ini bergantung penuh pada satelit, maka sinyal satelit menjadi sangat penting. Alat navigasi berbasis satelit ini tidak dapat bekerja maksimal ketika ada gangguan pada sinyal satelit. Ada banyak hal yang dapat mengurangi kekuatan sinyal satelit:

1. Kondisi geografis, seperti yang diterangkan diatas. Selama kita masih dapat melihat langit yang cukup luas, alat ini masih dapat berfungsi.
2. Hutan. Makin lebat hutannya, maka makin berkurang sinyal yang dapat diterima.
3. Air. Jangan berharap dapat menggunakan alat ini ketika menyelam.
4. Kaca film mobil, terutama yang mengandung metal.
5. Alat-alat elektronik yang dapat mengeluarkan gelombang elektromagnetik.

6. Gedung-gedung. Tidak hanya ketika di dalam gedung, berada di antara 2 buah gedung tinggi juga akan menyebabkan efek seperti berada di dalam lembah.
7. Sinyal yang memantul, misal bila berada di antara gedung-gedung tinggi, dapat mengacaukan perhitungan alat navigasi sehingga alat navigasi dapat menunjukkan posisi yang salah atau tidak akurat.

Arsitektur Sistem Navigasi Pariwisata di Jawa Timur pada Smartphone

Android dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur Sistem Navigasi Pariwisata Jawa Timur pada *Smartphone* Android

2.3 *Global Positioning System (GPS)*

Global Positioning System (GPS) adalah suatu sistem radio navigasi penentuan posisi dengan menggunakan satelit. GPS dapat memberikan posisi suatu objek di muka bumi secara cepat, akurat (tiga dimensi koordinat x, y, z) dan

memberikan informasi waktu serta kecepatan bergerak secara kontinyu di seluruh dunia pada setiap saat tanpa tergantung cuaca (Abidin, 2000). GPS merupakan sistem navigasi atau sistem penentuan posisi yang memiliki karakteristik dalam hal ketelitian. GPS memberi ketelitian posisi dari beberapa mm sampai beberapa meter dan ketelitian kecepatan sampai beberapa cm/detik serta ketelitian waktu sampai nanodetik.

Sistem GPS terdiri dari tiga segmen utama, yaitu segmen kontrol (control segment) yang terdiri dari stasiun-stasiun pemonitor dan pengontrol satelit, segmen satelit (space segment) yang terdiri dari satelit-satelit GPS, dan segmen pengguna (user segment) yang terdiri dari pengguna sipil dan militer. Satelit GPS mempunyai konstelasi 24 satelit dalam enam orbit yang mendekati lingkaran. Setiap orbit ditempati oleh 4 buah satelit dengan interval antara yang tidak sama. Orbit satelit GPS berinklinasi 55° terhadap bidang equator dengan ketinggian rata-rata dari permukaan bumi sekitar 20.200 km (Haryanto, 2004). Dengan adanya 24 satelit di angkasa, 4 sampai dengan 10 satelit GPS setiap saat akan selalu dapat diamati di seluruh permukaan bumi. Sinyal satelit GPS dipancarkan secara broadcast oleh satelit GPS secara kontinyu serta tidak tergantung cuaca. Dengan mengamati sinyal satelit menggunakan receiver GPS seseorang dapat menentukan posisi (lintang, bujur) di permukaan bumi. Informasi lainnya yang didapat dari satelit GPS selain posisi adalah kecepatan, arah, jarak, dan waktu.

2.3.1 Sejarah Perkembangan GPS

GPS dikembangkan pertama kali sebagai *NAVSTAR Global Positioning System* (GPS) juga dikenal sebagai *Navigation System with Timing And Ranging*

GPS. Sistem ini merupakan sistem penentuan posisi berbasis satelit, dan sekaligus merupakan tonggak revolusi bidang pengukuran posisi dan navigasi. Sistem GPS pada awalnya merupakan sistem navigasi militer yang dirancang, dilaksanakan, dibiayai, dan dikelola oleh Pertahanan Amerika Serikat Departement of Defence (DoD). Sistem ini dirancang oleh Amerika Serikat sejak tahun 1973. Sistem ini adalah hasil gabungan program U.S. Navy's Timation dan proyek U.S. Air Force's 621B di bawah tanggung jawab Joint Program Office (JPO). Satelit GPS yang pertama telah diluncurkan pada tahun 1978. Pada awalnya, penggunaan sistem ini ditujukan khusus bagi pihak militer Amerika Serikat. Namun setelah diluluskan pada Kongres Amerika Serikat, penggunaan sistem penentuan posisi ini terbuka untuk umum [Haryanto, 2004].

2.3.2 Sinyal GPS

Satelit GPS memancarkan sinyal yang pada prinsipnya untuk memberikan informasi kepada pengamat sinyal tentang posisi satelit GPS yang bersangkutan serta jaraknya dari pengamat lengkap dengan informasi waktunya (Abidin, 2000). Setiap sinyal dari satelit GPS membawa data yang diperlukan untuk mendukung proses penentuan posisi, kecepatan maupun waktu.

GPS memancarkan dua sinyal yaitu frekuensi L1 (1575.42 MHz) dan L2 (1227.60 MHz). Kedua Sinyal satelit GPS tersebut dikendalikan sesuai dengan jam atom rubidium atau cesium yang dipasang pada satelit GPS. Frekuensi jam atomic adalah 10.23 MHz (Haryanto, 2004). Sinyal L1 dimodulasikan dengan dua inyal pseudo-random yaitu kode P (Protected), kode C/A (coarse/aquisition) dan pesan navigasi sedangkan sinyal L2 hanya membawa kode P dan pesan navigasi.

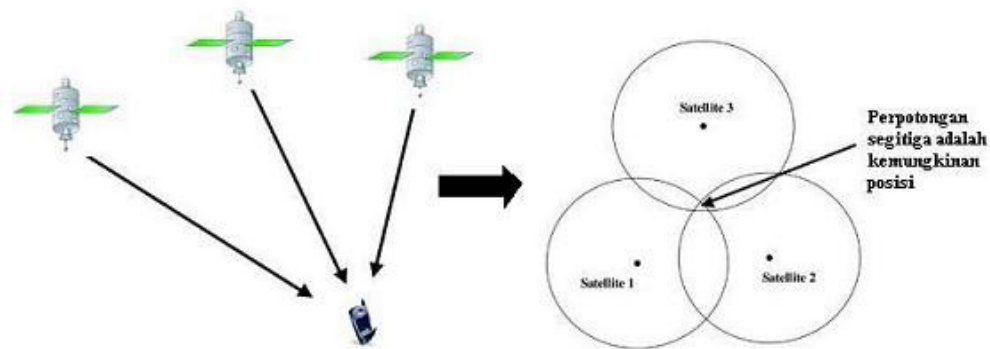
Setiap satelit mentransmisikan kode yang unik sehingga penerima (receiver GPS) dapat mengidentifikasi sinyal dari setiap satelit.

Ketika sinyal melalui lapisan atmosfer, maka sinyal tersebut akan terganggu oleh konten dari atmosfer tersebut. Besarnya gangguan di sebut bias. Bias sinyal yang ada utamanya terdiri dari 2 macam yaitu bias ionosfer dan bias troposfer. Bias ini harus diperhitungkan (dimodelkan atau diestimasi atau melakukan teknik differencing untuk metode diferensial dengan jarak baseline yang tidak terlalu panjang) untuk mendapatkan solusi akhir koordinat dengan ketelitian yang baik. Apabila bias diabaikan maka dapat memberikan kesalahan posisi sampai dengan orde meter.

2.3.3 Cara penentuan lokasi pada GPS

Satelit GPS mengorbit bumi dua kali dalam sehari dengan lintasan yang sangat presisi dan mentransmisikan sinyal informasi secara kontinu ke bumi. *GPS receiver* memanfaatkan informasi ini dengan berperan sebagai sebuah alat pengukur yang menghitung jarak antara antenna *receiver* dengan berbagai satelit GPS. Kemudian *GPS receiver* mendeduksi posisi melalui proses trilaterasi dengan mencari perpotongan tiap vektor satelit-satelit tersebut. Jarak antara antenna *receiver* dan satelit diukur dengan membandingkan waktu yang terdapat pada sinyal dengan waktu ketika sinyal diterima.

Sebuah *GPS receiver* setidaknya harus menangkap sinyal dari 3 buah satelit untuk menghitung posisi 2 dimensi (latitude dan longitude) dan pergerakannya. Dengan 4 buah satelit atau lebih, *receiver* dapat menghitung posisi 3 dimensi (latitude, longitude, dan altitude).



Gambar 2.3 Trilaterasi dalam GPS

Gambar 2.3 memberikan gambaran sederhana yang memodelkan proses trilaterasi dalam GPS untuk menentukan lokasi. Lingkaran melambangkan ruang wilayah cakupan sebuah satelit dan titik di tengahnya adalah satelit itu sendiri.

2.3.4 Akurasi GPS

GPS menyediakan posisi dengan ketepatan akurasi hingga 15 meter, yang berarti jika *GPS receiver* memberikan koordinat terhadap suatu lokasi tertentu, maka boleh diharapkan lokasi sebenarnya berada dalam radius 15 meter dari koordinat tersebut (El-Rabbany, 2002). Ketepatan GPS bergantung daripada lokasi *GPS receiver*-nya dan halangan terhadap sinyal satelit GPS. Meski secara umum, GPS menawarkan tingkat ketelitian 15 meter, namun akurasi ini dapat ditingkatkan dengan berbagai teknik, seperti *Assisted GPS (A-GPS)*, *Differential GPS (D-GPS)*, atau *Wide Area Augmentation System (WAAS)*. Pada kenyataan di lapangan, akurasi dapat bervariasi sesuai keadaan.

2.4 Android

Android adalah sistem operasi untuk smartphone yang dibuat oleh google corporation. Sistem operasi ini dikembangkan dengan memanfaatkan linux kernel.

(Google, 2010). Perencanaan Sistem dan Tori Penunjang Android). Versi stable terbaru dari Android adalah Honeycomb 3.1 yang di rilis pada bulan Mei 2011 dengan Linux versi 2.6.37. Pada versi ini fitur Android sudah dilengkapi dengan High performance WiFi lock, Download manager, Streaming audio formats dan juga browser Chrome versi 8 menggunakan HTML5. Pada versi 3.0 ke atas perangkat selular yang digunakan tidak lagi perangkat selular genggam melainkan menggunakan perangkat selular berbentuk tablet, yang memiliki ukuran lebih besar, kurang lebih dua kali lipat di banding perangkat selular biasa.

Dari segi arsitektur sistem, Android merupakan sekumpulan framework dan virtual machine yang berjalan di atas kernel Linux. Virtual machine Android bernama Dalvik Virtual Machine (DVM), engine ini berfungsi untuk menginterpretasikan dan menghubungkan seluruh kode mesin yang digunakan oleh setiap aplikasi dengan kernel Linux. Sementara untuk framework aplikasi sebagian besar dikembangkan oleh Google dan sebagian yang lain dikembangkan oleh pihak ketiga (developer). Beberapa framework yang dikembangkan oleh Android sendiri misalnya fungsi untuk telephony seperti panggilan telepon, SMS, video call. Untuk browser Android menggunakan Google Chrome.

Aplikasi yang dikembangkan di atas Android dibuat dengan menggunakan kode Java seperti halnya J2ME yang telah lama digunakan pada platform perangkat selular umumnya. Namun secara siklus program memiliki perbedaan mendasar antara J2ME dengan Java yang ada pada Android. Kode Java pada Android lebih dekat dengan J2SE. Hingga saat ini Android telah banyak digunakan pada produk smartphone seperti Samsung, LG, SonyEricsson, Nexian dan juga HTC. Dengan dukungan Software Development Kit (SDK) dan Application

Programming Interface (API) dari google memberikan kemudahan bagi pihak ketiga (developer) untuk membangun aplikasi yang dapat berjalan pada sistem operasi android.

2.4.1 Platform Google Android

Android merupakan suatu *software stack* untuk *mobile device* (Google Inc, 2003). Di dalamnya terdapat sistem operasi, *middleware*, dan *key application*. Aplikasi pada platform ini dikembangkan dalam bahasa pemrograman java. Android sendiri memiliki banyak fitur diantaranya adalah:

1. Merupakan sebuah *Application Framework* sehingga programmer dapat menggunakan beberapa fungsi yang telah disediakan.
2. *Dalvik virtual machine*. Tiap aplikasi dalam Android memiliki instance virtual machine yang dapat bekerja secara efisien dalam lingkungan memori yang terbatas.
3. *Integrated browser*. *Web browser* berbasis *WebKit engine* terdapat pada browser default Android atau pun dapat diintegrasikan dengan aplikasi lain.
4. *Optimized graphics*. *Library* grafis 2D yang kaya dan 3D berbasis OpenGL ES 1.0 yang mendukung akselerasi *hardware*.
5. SQLite. Basis data relasional yang ringan namun sangat *powerfull*.
6. *Media Support*. Mendukung berbagai format audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
7. *GSM Telephony*. Mendukung fungsi komunikasi GSM.
8. *Bluetooth, EDGE, 3G dan WiFi*. Mendukung komunikasi pada jaringan (tergantung hardware).

9. Kamera, GPS, kompas dan accelerometer. Mendukung berbagai fitur yang disediakan oleh hardware.
10. *Device emulator, tools* untuk *debugging*, profiling memori dan performa, plugin untuk Eclipse IDE.

2.4.2 Perkembangan Platform Android

Android merupakan system operasi yang dikembangkan oleh *google* bersama perusahaan yang tergabung dalam *open handset alliance* (Intel, NVIDIA, Texas Instrument) sejak tahun 2007. Hasil survey AC Nielsen yang kerap dijadikan sebagai acuan menyebutkan bahwa dalam 3 bulan terakhir penjualan Android mencetak angka 56 persen dari seluruh penjualan smartphone di dunia. Sedangkan Apple dengan iOS hanya menguasai 28 persen yang membuatnya berada di posisi kedua setelah Android. Sementara BlackBerry berada di tempat ketiga dengan menguasai 9 persen penjualan. Hasil survey yang dilakukan oleh AC Nielsen ini dikumpulkan dari 25.500 penjualan smartphone dalam 3 bulan terakhir. Dengan kata lain Android berhasil menjual 14.280 unit smartphone mengalahkan iOS yang hanya mampu menjual iPhone-nya sebanyak 7.140 unit saja. Angka ini akan semakin menunjang Google dalam melakukan pengembangan OS Android beserta aplikasinya untuk bisa digunakan oleh pasar yang kini sedang berkembang dengan pesat (TelSetNews.com, 2011).

2.4.3 Life Cycle Aplikasi Android

Pada banyak kasus, tiap aplikasi pada Android masing-masing memiliki Linux proses. Proses ini diciptakan untuk aplikasi tersebut pada saat kode program tersebut akan dieksekusi, akan terus berjalan sampai tidak dibutuhkan lagi dan sistem memerlukan memori untuk aplikasi lain.

Konsep dasar program Android adalah suatu proses pada aplikasi tidak langsung di atur oleh aplikasi tersebut. Proses tersebut ditentukan oleh sistem melalui kombinasi: informasi aplikasi bagi sistem, seberapa pentingnya aplikasi tersebut bagi user dan ketersediaan memori.

Contoh kasus *life-cycle bug* adalah *IntentReceiver* yang menciptakan *thread* pada saat menerima intent pada *method onReceiveIntent* dan selesai dari fungsi tersebut. Setelah fungsi tersebut selesai, sistem menganggap bahwa *IntentReceiver* tersebut menjadi tidak aktif sehingga ia memiliki proses yang tidak dibutuhkan (kecuali terdapat komponen aplikasi lain di dalamnya). Oleh karena itu, sistem dapat menghapus proses tersebut kapanpun apabila dibutuhkan. Hal tersebut akan menghapus semua thread yang berjalan pada proses tersebut. Solusi masalah ini adalah memakai *Service* dari *IntentReceiver*, sehingga sistem mengetahui bahwa ada suatu yang aktif dalam proses tersebut.

Sebagai programmer, kita harus mengetahui bagaimana cara yang tepat untuk merancang aplikasi. Kesalahan pada perancangan akan membuat suatu aplikasi akan dihentikan oleh sistem pada saat yang tidak diinginkan. Untuk menentukan proses mana yang akan dihapus pada situasi *low memory*, Android memiliki "*importance hierarchy*" berdasarkan komponen dan state yang sedang berjalan. Urutan berdasarkankepentingan tersebut adalah:

1. *Foreground process*. Adalah suatu proses yang dibutuhkan untuk berinteraksi dengan user saat ini. Berbagai komponen suatu aplikasi dapat mengakibatkan proses di dalamnya menjadi *foreground* proses. Hanya akan terdapat beberapa proses seperti itu dalam sistem, proses ini hanya akan dihapus apabila sistem

tidak memiliki pilihan lain. Contohnya apabila memori tinggal sedikit sehingga proses foreground tidak bisa lagi berjalan. Hal ini diperlukan untuk menjaga user interface tetap interaktif dengan user.

2. *Visible process.* Adalah proses yang memegang Activity yang dapat dilihat user pada layar tetapi tidak foreground (pada saat method `onPause()` dipanggil). Hal ini dapat terjadi, sebagai contoh activity foreground muncul dengan dialog sehingga activity sebelumnya terlihat di belakangnya. Proses ini sangat penting dan tidak akan di hapus kecuali untuk menjaga aplikasi foreground tetap berjalan.
3. *Service process.* Adalah proses yang memegang Service yang dimulai dengan method `startService()`. Meskipun proses ini secara langsung tidak dapat dilihat oleh user, biasanya proses ini menjalankan perintah dari user (seperti memutar lagu, men-download data dari jaringan dan lain-lain). Sistem akantetap membiarkan proses ini berjalan selama proses foreground dan visible masih bisa berjalan.
4. *Background process.* Adalah proses yang memegang Activity yang saat ini tidak dapat dilihat oleh user (method `onStop()` sedang dipanggil). Proses ini tidak secara langsung mempengaruhi user. Apabila activity life cycle telah dipenuhi dengan benar, sistem dapat menghapus proses ini sewaktu-waktu untuk memberikan memori tambahan. Biasanya banyak ditemukan proses semacam ini, sehingga proses tersebut disimpan dalam LRU (Least Recently Used) untuk menjaga agar proses yang baru saja dilihat oleh user akan dihapus paling akhir pada kondisi low memory.

5. *Empty Process*. Adalah proses yang tidak memegang suatu komponen yang sedang aktif. Alasan mengapa membiarkan proses semacam ini adalah untuk melakukan cache sehingga proses startup menjadi lebih cepat apabila aplikasitersebut akan dijalankan kembali. Sistem seringkali menghapus proses semacam ini untuk mencapai keseimbangan dari system resource antara cached processes dan kernel caches.

Berikut ini adalah tabel daftar rilis versi android dan juga kernel linux yang digunakan pada sistem operasi android. Secara garis besar kernel linux yang digunakan pada perangkat selular android hampir sama dengan kernel linux pada komputer.

Nama	Kode	Kernel Linux	Tahun Rilis
<i>Android</i>			
Cupcake 1.5		2.6.27	Februari 2009
Donut 1.6		2.6.29	September 2009
Éclair 2.1		2.6.29	Oktober 2009
Froyo 2.2		2.6.32	Mei 2010
GingerBread 2.3		2.6.34	Oktober 2010
HoneyComb 3.1		2.6.37	Mei 2011

Tabel 2.4.1 Daftar rilis versi android dan kernel linux

2.4.4 *The Dalvik Virtual Machine (DVM)*

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di dalam *Dalvik Virtual Machine* (DVM) bukan di *Java Virtual Machine* (JVM), sebenarnya banyak persamaannya dengan *Java Virtual*

Machine (JVM) seperti Java ME (Java mobile Edition), tetapi Android menggunakan *Virtual Machine* sendiri yang menurut saya dikustomisasi dan dirancang untuk memastikan bahwa beberapa *feature-feature* berjalan lebih efisien pada perangkat mobile.

Dalvik Virtual Machine (DVM) adalah "*register bases*" sementara Java *Virtual Machine* (JVM) adalah "*stack based*", DVM didesain oleh Dan Bornsten dan beberapa *engineers Google* lainnya. DVM menggunakan kernel Linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C / C + sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari kernel Linux yang digunakan dalam Android tersebut.

Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan tools yang disediakan dalam SDK Android.

2.4.5 Android SDK (Software Development Kit)

Android SDK adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java (Harahap, 2011). Android merupakan

subset perangkat lunak untuk *smartphone* yang meliputi sistem operasi, middleware dan aplikasi kunci yang *direlease* oleh *Google*.

Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, Android memberi Anda kesempatan untuk membuat aplikasi yang bukan merupakan aplikasi bawaan Smartphone.

Beberapa fitur-fitur Android yang paling penting adalah:

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat mobile.
3. *Integrated browser* berdasarkan engine open source WebKit.
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL ES 1.0 (Opsional akselerasi hardware).
5. SQLite untuk menyimpan data.
6. *Media Support* yang mendukung audio, video, dan gambar.
7. *Bluetooth, EDGE, dan WiFi* (tergantung hardware).
8. Kamera, GPS, kompas, dan *accelerometer* (tergantung *hardware*)
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat emulator, tool untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.4.6 Anatomi Aplikasi Android

Terdapat empat *building blocks* pada aplikasi Android yaitu:

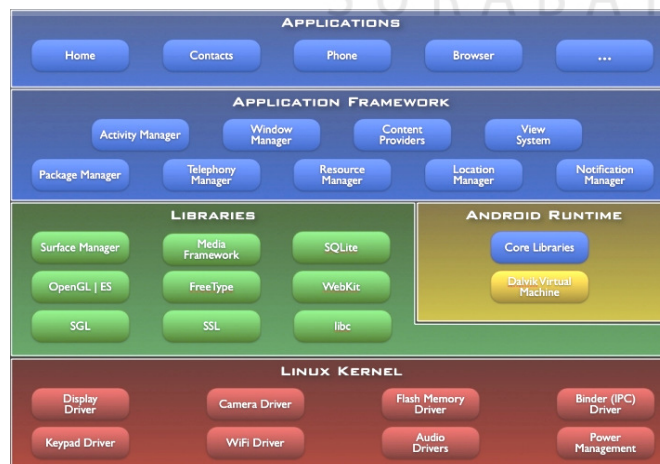
1. *Activity*

2. *Intent Receiver*
3. *Service*
4. *Content Provider*

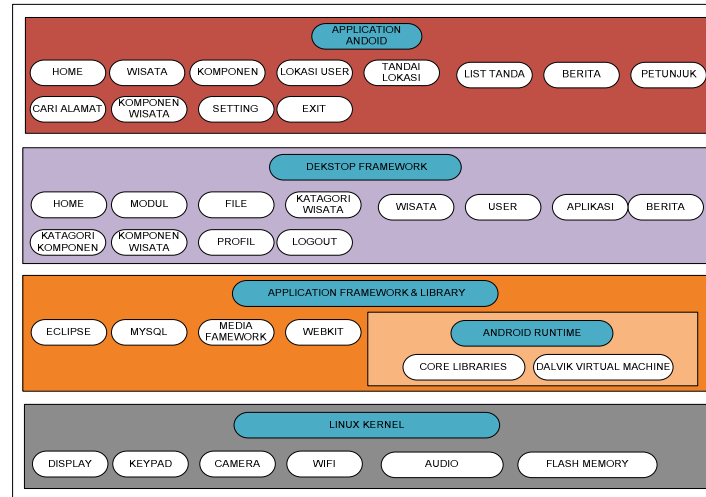
Tidak semua aplikasi membutuhkan keempat blok ini, tetapi suatu aplikasi dibuat menggunakan kombinasi beberapa blok ini. Setelah memutuskan blok komponen yang akan digunakan, blok tersebut didaftarkan pada suatu file yang disebut dengan `AndroidManifest.xml`. File XML ini digunakan untuk menyatakan komponen apa saja yang dibutuhkan oleh sebuah aplikasi dan kemampuan serta kebutuhan aplikasi tersebut.

2.4.7 Arsitektur Android

Arsitektur sistem terdiri atas 5 layer, pemisahan layer bertujuan untuk memberikan abstraksi sehingga memudahkan pengembangan aplikasi. *Layer-layer* tersebut adalah *layer* aplikasi, *layer framework* aplikasi, *layer libraries*, *layer runtime*, dan *layer kernel*. Gambar 2.5 memberikan gambaran umum komponen-komponen dalam arsitektur sistem operasi Android.



Gambar 2.5. Arsitektur Android



Gambar 2.6 Lapisan arsitektur sistem navigasi pariwisata pada Android.

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut:

1. *Applications dan Widgets*

Applications dan *Widgets* ini adalah layar dimana berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut, di layer inilah terdapat seperti aplikasi inti termasuk klien email, program sms, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman java.

2. *Aplikasi Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada penembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resource, menjalankan *service background*, mengatur alarm, dan menambahkan tambahan seperti status

notificaations, dan masih banyak lagi. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang berkategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan komponen yang sudah digunakan (*reuse*).

3. *Libraries*

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

4. *Android Runtime*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android.

5. *Linux Kernel*

Linux kernel adalah *layer* dimana inti dari operating sistem dari Android itu sendiri, berisi file-file *system* yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operating Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel release 2.6.

2.5 *Google Maps*

Google Maps adalah sebuah layanan gratis peta digital dari *Google* berbasis *web* yang dapat digunakan dan ditempatkan pada *website* tertentu dengan menggunakan *Google Maps API* (Wikipedia, 2010).

Google Maps sendiri mempunyai fitur-fitur antara lain navigasi peta dengan *dragging mouse*, *zoom-in* dan *zoom-out* untuk menunjukkan informasi peta secara detil memberi penanda pada peta dan memberi informasi tambahan. *Mode viewing* pada *Google Maps* berupa “*Map*” (peta topografi dan jalan), “*satelite*” (peta berupa foto satelit dan foto resolusi tinggi dari udara), “*Hybrid*” (peta berupa foto satelit dan peta jalan berada di atasnya) dan “*Street View*”, fasilitas ini secara resmi diperkenalkan oleh Google pada Mei 2007.

Google telah membuat *Google Maps API* untuk memfasilitasi para *developer* untuk mengintegrasikan *Google Maps* pada *websitenya*. Ini merupakan layanan gratis yang sementara tidak mengandung iklan, tetapi *Google* menyatakan pada perjanjian menggunakan layanan bahwa mereka berhak untuk menampilkan iklan dimasa yang akan datang.

Dengan menggunakan *Google Maps API* kita dapat menampilkan seluruh fasilitas *Google Maps* pada *website* kita. Dimulai dengan membuat *API key* (*API Key* ini berfungsi sebagai kunci akses untuk *website* kita) dan kita sudah dapat menggunakan fungsi-fungsi yang ada pada *Google Maps API* untuk *website* kita.