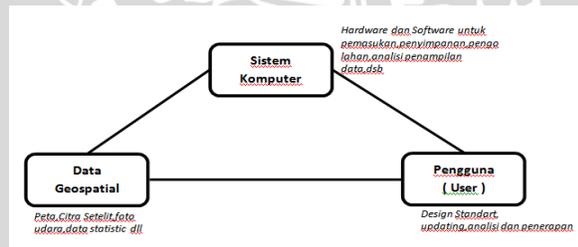


## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

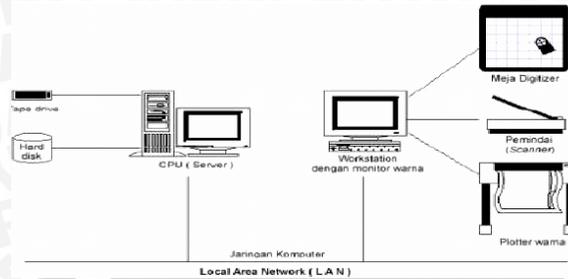
#### 2.1 GIS (*Geographic Information System*)

*GIS (Geographical Information System)* atau dikenal pula dengan SIG (*Sistem Informasi Geografis*) merupakan sistem informasi berbasis komputer yang menggabungkan antara unsur peta (geografis) dan informasinya tentang peta tersebut (data atribut) yang dirancang untuk mendapatkan, mengolah, memanipulasi, analisa, dan menampilkan data spasial untuk menyelesaikan perencanaan, mengolah dan meneliti permasalahan. Dengan definisi ini, maka terlihat bahwa aplikasi GIS di lapangan cukup luas terutama bagi bidang yang memerlukan adanya suatu sistem informasi tidak hanya menyimpan, menampilkan, dan menganalisa data atribut saja tetapi juga unsur geografisnya seperti PT. Telkom, Pertamina, Departemen Kelautan, Kehutanan, Bakosurtanal, Marketing, Perbankan, Perpajakan, dan yang lainnya. Komponen kunci dalam GIS adalah sistem komputer, data geospasial (data atribut) dan pengguna, yang dapat digambarkan sebagai berikut



**Gambar 2.1.** Komponen Kunci GIS [PRA-01]

Sistem komputer SIG terdiri dari perangkat keras (*hardware*), perangkat lunak (*software*) dan prosedur untuk penyusunan pemasukkan data, pengolahan, analisis, pemodelan (*modelling*), dan penayangan data *geospasial*.



**Gambar 2.2** Komponen Hardware GIS [PRA-01]

## 2.2 Sistem Informasi

Pada dasarnya sistem informasi bukanlah sistem yang selalu terkait dengan komputer. Adapun beberapa penjelasan terkait dengan sistem informasi adalah pengertian sistem, pengertian data, pengertian informasi, dan pengertian sistem informasi itu sendiri.

### 2.2.1. Pengertian Sistem

Pengertian sistem bisa dilihat dari dua pendekatan, yaitu berdasarkan prosedur dan komponennya. Sistem dilihat dari pendekatan prosedur berarti kumpulan prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu sedangkan sistem dilihat dari pendekatan komponennya berarti sistem terdiri dari komponen-komponen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

### 2.2.2. Pengertian Data

Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata. Atau data adalah : representasi dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, pelanggan), hewan, peristiwa, konsep, keadaan dll, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

### 2.2.3. Pengertian Informasi

Informasi merupakan data yang memiliki nilai lebih bagi penerimanya karena telah melewati proses pengolahan tertentu sehingga bisa dimanfaatkan untuk mengambil keputusan baik itu untuk saat ini ataupun untuk masa yang akan datang atau informasi didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

### 2.2.4. Pengertian Sistem Informasi

Berdasarkan pemaparan sebelumnya mengenai sistem, data, dan informasi maka dapat diambil kesimpulan bahwa pengertian dari sistem informasi adalah sekumpulan komponen-komponen dan prosedur-prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu. Berikut pengertian-pengertian lainnya dari sistem informasi:

1. Kumpulan dari orang-orang, data, prosedur, perangkat keras dan lunak yang bekerja bersama-sama untuk mencapai satu tujuan dalam manajemen informasi.
2. Keterkaitan komponen-komponen dalam pengumpulan (atau menerima), proses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan (decision making), mengkoordinasi serta mengontrol
3. Sistem informasi juga membantu, manajer dan para pegawai (pekerja) menganalisa masalah, menggambarkan “subyek” yang kompleks dan membuat (menghasilkan) produk-produk baru
4. Dasar (pondasi) dari bisnis pada umumnya.
5. Ada tiga aktifitas pada sistem informasi menghasilkan informasi pada organisasi yang diperlukan untuk membuat suatu keputusan, pengendalian, analisa masalah-masalah dan membuat produk baru atau layanan. Aktifitas itu adalah *input*, *process* dan *output*.
6. *Input* berarti menangkap atau mengumpulkan data mentah (*raw data*) didalam organisasi atau dari luar organisasi – *environment* / lingkungan. Proses diartikan dengan menterjemahkan data *input* kedalam bentuk yang lebih berarti. Sedangkan

*output* adalah mentransfer informasi yang telah diproses kepada manusia dan aktivitasnya.

## 2.3 Android

### 2.3.1. Pengertian Android

Android adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, *middleware*, dan aplikasi kunci. Android *Standart Development Kit* (SDK) menyediakan perlengkapan dan *Application Programming Interface* (API) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java.[[developer.android.com](http://developer.android.com)]

Android dikembangkan oleh Google bersama *Open Handset Allience* (OHA) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *Hardware*, *Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

### 2.3.2. Sejarah dan Perkembangan Android

Pada mulanya terdapat berbagai macam sistem operasi pada perangkat selular, diantaranya sistem operasi Symbian, Microsoft Windsos Mobile, Mobile Linux, iPhone, dan sistem operasi lainnya. Namun diantara sistem operasi yang ada belum mendukung standar dan penerbitan API yang dapat dimanfaatkan secara keseluruhan dan dengan biaya yang murah. Kemudian Google ikut berkecimpung didalamnya dengan platform Android, yang menjanjikan keterbukaan, keterjangkauan, *open source*, dan *framework* berkualitas.

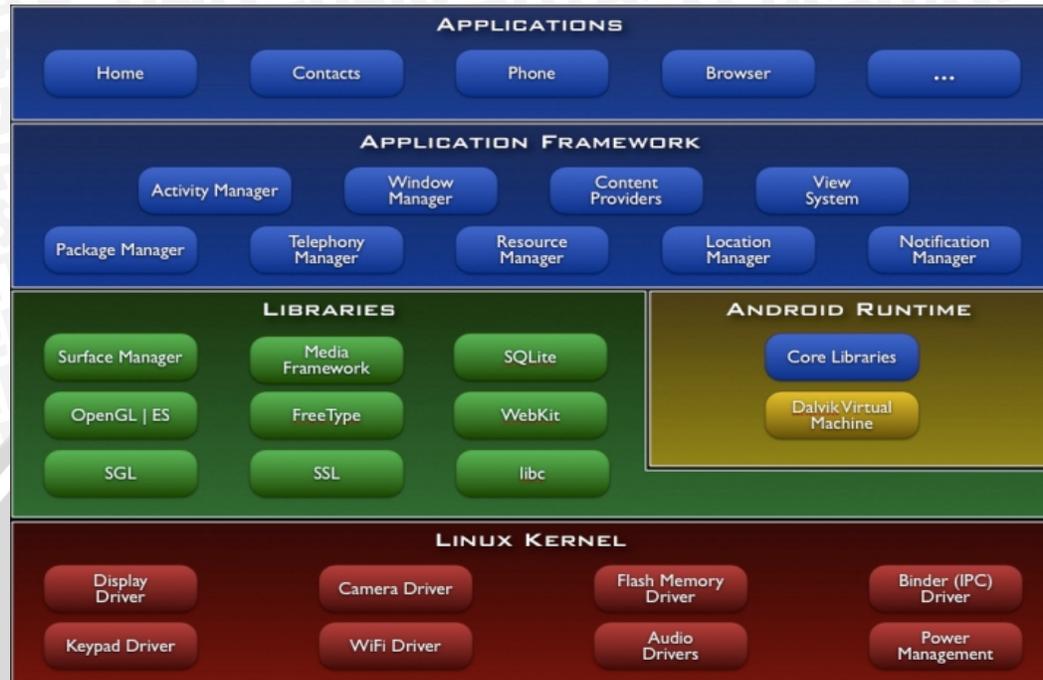
Pada tahun 2005, Google mengakuisisi perusahaan Android Inc. untuk memulai pengembangan platform android. Dimana terlibat dalam pengembangan ini Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada pertengahan 2007 sekelompok pemimpin industri bersama-sama membentuk aliansi perangkat selular terbuka, Open Handset Alliance (OHA). Bagian dari tujuan aliansi ini adalah berinovasi dengan cepat dan menanggapi kebutuhan konsumen dengan lebih baik,

dengan produk awalnya adalah platform Android. Dimana Android dirancang untuk melayani kebutuhan operator telekomunikasi, manufaktur handset, dan pengembang aplikasi. OHA berkomitmen untuk membuat android *open source* dengan lisensi Apache versi 2.0.

Android pertama kali diluncurkan pada 5 November 2007, dan *smartphone* pertama yang menggunakan sistem operasi android dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008. Hingga saat ini android telah dirilis beberapa versi android untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi android terdapat kode nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama *Cupcake*, versi 1.6 dirilis pada 15 September 2009 diberi nama *Donut*, dan versi terakhir 2.0 dirilis pada 26 Oktober 2009 diberi nama *Éclair*. [SAY-09]

### 2.3.3. Anatomi Android

Dalam paket sistem operasi android terdiri dari beberapa unsur seperti tampak pada gambar 2.4 secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.



**Gambar 2.3** Detail Anatomi Android [Google IO, Android Anatomy and Physiology]

### 2.3.3.1. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

### 2.3.3.2. Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

- *Media Library* untuk memutar dan merekam berbagai macam format audio dan video.
- *Surface Manager* untuk mengatur hak akses layer dari berbagai aplikasi.
- *Graphic Library* termasuk didalamnya *SGL* dan *OpenGL*, untuk tampilan 2D dan 3D.
- *SQLite* untuk mengatur relasi database yang digunakan pada aplikasi.
- *SSL* dan *WebKit* untuk browser dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit* (NDK).

### 2.3.3.3. Android Runtime

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam *Android Runtime* terdapat 2 bagian utama, diantaranya:

- Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin virtual Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android.
- Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah

perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (\*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas. [NIC]

#### 2.3.3.4. Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan *user interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut :

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan "Backstack" untuk navigasi penggunaan.
2. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
3. *Resource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan file layout.
4. *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
5. *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada *status bar*.

#### 2.3.3.5. Application Layer

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam

*Android runtime* dengan menggunakan kelas dan service yang tersedia pada *framework* aplikasi.

Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

#### 2.3.4. Komponen Aplikasi

Fitur penting android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan fitur *scroller* yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada.

Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi *main()* pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan.

##### 2.3.4.1. Activities

*Activity* merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan.

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh

posisinya dalam tumpukan *activity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi:

- 1) **Active**, setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.
- 2) **Paused**, dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukan dari pengguna. Dalam kasus ekstrim, android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.
- 3) **Stopped**, ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive*.

- 4) **Inactive**, kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

Kondisi transisi ini sepenuhnya ditangani oleh manajer memori android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

#### 2.3.4.2. Services

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

#### 2.3.4.3. Intents

*Intens* merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut.

#### 2.3.4.4. Broadcast Receivers

*Broadcast Receivers* merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *Broadcast* berasal dari sistem misalnya, Batre sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa *default*

pada perangkat. Sama halnya dengan *service*, *Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna.

#### 2.3.4.5. Content Providers

*Content Providers* digunakan untuk mengelola dan berbagi database. Data dapat disimpan dalam file sistem, dalam database *SQLite*, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

#### 2.3.5. Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada android :

##### 1. Foreground Activity

Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

##### 2. Background Service

Aplikasi yang memiliki interaksi terbatas dengan user, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms auto respon.

##### 3. Intermittent Activity

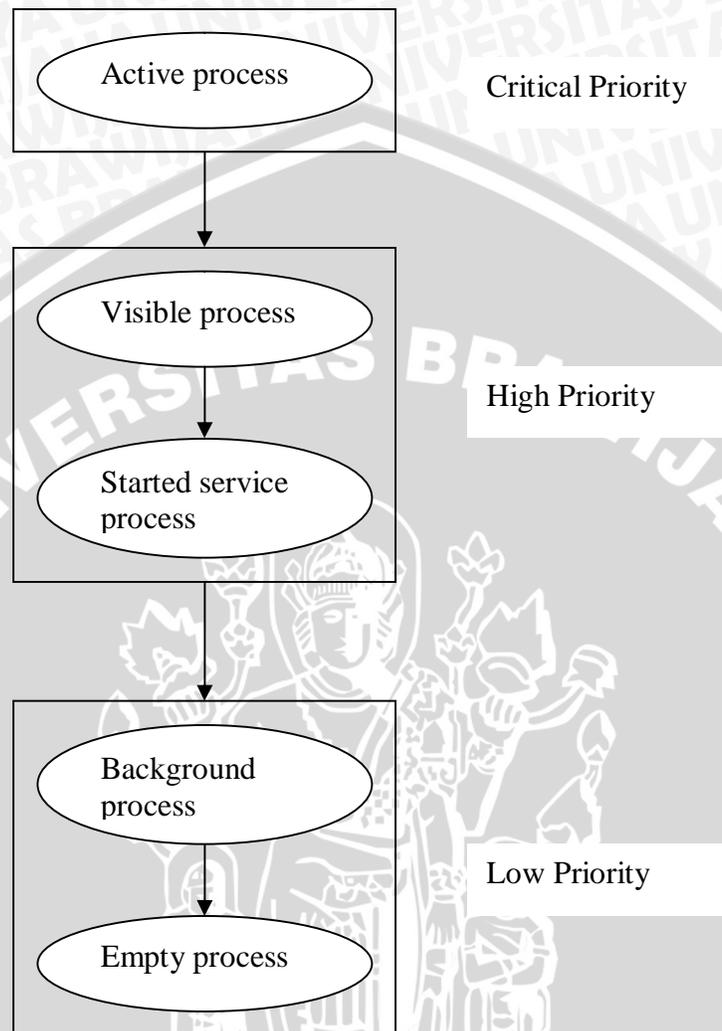
Aplikasi yang masih membutuhkan beberapa masukan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik.

Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

### 2.3.6. Siklus Hidup Aplikasi Android

Siklus hidup aplikasi android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan browser web, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan prioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil.



**Gambar 2.4** Prioritas aplikasi berdasarkan *activity*

### 2.3.7. Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiMO), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut:

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasis Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar royalty. Sementara pengembang software menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.
2. Arsitektur komponen dasar android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan service, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

## 2.4 Tools Implementasi

*Tools* implementasi merupakan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak. Adapun *tools* implementasi yang digunakan untuk membuat aplikasi *mobile* jejaring sosial Kampus *Online* Unikom berbasis Android adalah JDK sebagai perangkat pembangun Java, Eclipse IDE sebagai lingkungan pengembangan perangkat lunak, dan SQL sebagai sistem manajemen basis data.

### 2.4.1. JDK

JDK (*Java Development Kit*) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman Java. JDK berjalan diatas sebuah *virtual machine* yang dinamakan JVM (*Java Virtual Machine*). Dokumentasi JDK berisi spesifikasi API, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas JDK dan utilitas, demo, dan link ke informasi terkait. Dokumentasi ini juga tersedia dalam bundel yang dapat diunduh dan diinstal pada komputer.

### 2.4.2. Eclipse

Eclipse adalah sebuah komunitas proyek *open source* yang difokuskan pada membangun sebuah *platform* pengembangan yang dapat diperluas, *runtime* dan kerangka aplikasi untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh daur hidup perangkat lunak. Komunitas Eclipse memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori :

1. *Enterprise Development*
2. *Embedded and Device Development*
3. *Rich Client Platform*
4. *Rich Internet Applications*
5. *Application Frameworks*

6. *Application Lifecycle Management (ALM)*
7. *Service Oriented Architecture (SOA)*

Komunitas Eclipse juga didukung oleh ekosistem besar dan dinamis dari penyedia solusi teknologi informasi utama, *start-up* inovatif, universitas dan lembaga penelitian, dan individu yang memperluas, mendukung, dan melengkapi *platform* Eclipse.

#### 2.4.3. SQLite

SQLite merupakan sebuah perpustakaan proses yang menerapkan kemandirian, tanpa server, tidak adanya konfigurasi mesin, dan basis data SQL transaksional. Kode untuk SQLite berada dalam domain publik dengan demikian bebas untuk digunakan untuk tujuan apapun, komersial atau swasta. SQLite adalah mesin basis data SQL yang tertanam. Tidak seperti kebanyakan basis data SQL lainnya, SQLite tidak memiliki proses server yang terpisah. SQLite membaca dan menulis secara langsung ke *file disk* biasa. Basis data SQL lengkap dengan berbagai tabel, indeks, *triggers*, dan *views* terkandung dalam sebuah *file disk* tunggal.