

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Game***

##### **2.1.1 Pengertian *Game***

Kata *game* berasal dari bahasa Inggris. Dalam kamus bahasa Indonesia istilah *game* berarti permainan. Permainan merupakan bagian dari bermain dan bermain juga bagian dari permainan keduanya saling berhubungan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual yang juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Dalam *game*, ada target-target yang ingin dicapai pemainnya. Permainan adalah kegiatan yang kompleks yang di dalamnya terdapat peraturan, bermain dan budaya. Permainan adalah sebuah sistem di mana pemain terlibat dalam konflik buatan. Pemain akan berinteraksi dengan sistem dan konflik pada permainan yang merupakan rekayasa atau buatan. Dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Mendefinisikan apakah yang dimaksud dengan *game*, tidak cukup dengan hanya melihat kamus bahasa. Terdapat banyak makna dalam kata *game*. *Game* merupakan bagian dari kehidupan manusia. Makna sekilas dari *game* memberikan pengertian bahwa *game* merupakan suatu aktifitas yang tidak dilakukan dengan sungguh-sungguh. Untuk mengetahui apa yang sesungguhnya disebut dengan *game*, maka paling tidak dapat memahami dari adanya sejumlah pengertian *game* yang biasa dialami dalam kehidupan (Zamroni, et. al. 2013:489).

### 2.1.2 Klasifikasi *Game*

Menurut Martono (2015:23) Klasifikasi *game* dimaksudkan untuk memudahkan pengelompokan jenis *game*. Beberapa klasifikasi *game* adalah seperti berikut :

1. *Game as Game*, *game* yang dimaksud adalah *game* untuk kesenangan atau *fun*.
2. *Game as Media*, tujuan utama dari *game* as media adalah untuk menyampaikan pesan tertentu, menyampaikan pesan dari pembuat *game* tersebut.
3. *Game Beyond Game*, bisa disebut juga dengan istilah *gamification*. *Gamification* adalah penerapan konsep atau cara berpikir *game design* ke dalam lingkup *non-game*.

### 2.1.3 *Game* Edukasi

*Game* Edukasi sering disebut sebagai *edutainment*, *edutainment* berasal dari kata *education* dan *entertainment*. *Education* berarti pendidikan, sedangkan *entertainment* berarti hiburan. Jadi, dari segi bahasa, *edutainment* adalah pendidikan yang menghibur atau menyenangkan. Sementara itu, dari segi terminologi, *edutainment* adalah suatu proses pembelajaran yang didesain sedemikian rupa, sehingga muatan pendidikan dan hiburan bisa dikombinasikan secara harmonis untuk menciptakan pembelajaran yang menyenangkan (Hamid, 2011:17).

Menurut Riva (2012:12) ada beberapa fungsi permainan edukatif di antaranya adalah: 1) memberikan ilmu pengetahuan kepada anak melalui proses pembelajaran bermain sambil belajar, 2) merangsang pengembangan daya pikir, daya cipta, dan bahasa agar mampu menumbuhkan sikap, mental, serta akhlak yang baik, 3) menciptakan lingkungan bermain yang menarik, memberikan rasa aman, dan menyenangkan, 4) meningkatkan kualitas pembelajaran anak, bermain memiliki peranan yang penting dalam perkembangan anak pada hampir semua bidang perkembangan fisik-motorik, bahasa, intelektual, moral, sosial, maupun emosional.

#### **2.1.4 *Game Engine***

*Game Engine* adalah sebuah perangkat lunak yang dirancang untuk membuat sebuah *game*. Sebuah *Game Engine* biasanya dibangun dengan mengenkapsulasi beberapa fungsi standar yang umum digunakan dalam pembuatan sebuah *game*. Misalnya, fungsi *rendering*, pemanggilan suara, *network*, atau pembuatan partikel untuk *special effect*. Sebagian besar *Game Engine* umumnya berupa *library* atau sekumpulan fungsi-fungsi yang penggunaannya dipadukan dengan bahasa pemrograman (Roedavan, 2014:1).

#### **2.1.5 Tahapan Pembuatan *Game***

Pada tahapan pembuatan *game* secara umum terbagi menjadi lima tahap. Dan tahap - tahap tersebut yaitu :

### 1. Perencanaan

Tahap perencanaan adalah proses dasar memahami mengapa *game* harus dibangun dan menentukan bagaimana pembuat akan membangun sistem tersebut. Pada tahap ini pembuat merencanakan konsep *game*, sasaran-sasaran yang ingin dicapai dan jangka waktu pelaksanaan.

### 2. Analisa

Tahap ini merupakan kegiatan penguraian dari suatu *game* yang utuh ke dalam bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya serta mempelajari mekanisme *game* dan teori pendukung lainnya.

### 3. Desain Tampilan Permainan

Pada tahap ini membuat bagaimana *game* yang akan dibangun dapat mudah dimengerti oleh pengguna, dalam menampilkan penjelasan sistem serta memberikan panduan pemakaian sistem secara menyeluruh.

### 4. Desain Gambar

Pada tahap desain gambar ini adalah membuat macam-macam gambar sesuai dengan kebutuhan pada *game* seperti karakter utama, karakter prolog, dan gambar latar pada *game*.

### 5. Implementasi

Tahap terakhir adalah tahap implementasi, di mana *game* akan diimplementasikan pada perangkat dan uji coba kelayakan *game* tersebut.

## 2.2 Corona SDK

Corona SDK adalah *sdk* yang dibuat oleh Walter Luh yang merupakan *founder* dari CoronaLabsInc. Programmer Walter Luh memulai CoronaLabs setelah keluar dari Adobe pada tahun 2007, kemudian pada juni 2009 Luh mengeluarkan Corona SDK pertama kali versi beta. Corona menggunakan bahasa pemrograman lua yang sangat mudah dipelajari. CoronaLabs memiliki 3 buah produk yaitu Corona SDK, CoronaEnterprise, dan CoronaCoud (Fernandez, 2015:1).

### 2.2.1 Platform Corona SDK

Corona SDK merupakan *software* yang digunakan untuk membangun sebuah aplikasi *mobile* seperti Android, iPhone, dan iPad. Dengan menggunakan *source code* yang sama, Corona dapat melakukan *build* ke beberapa *platform* seperti Android, Iphone dan iPad, hanya saja untuk iPhone dan iPad membutuhkan *support build* berbayar.

### 2.2.2 Kelebihan dan Kekurangan Corona SDK

Kelebihan yang dimiliki Corona SDK dapat mengembangkan aplikasi 10 kali lebih cepat karena kuncinya ada di bahasa pemrograman yang mudah untuk dipahami oleh pengembang, *go cross-platform* maksudnya program hasil dari Corona SDK secara otomatis dapat terbaca pada platform Android dan iOS, bahkan saat ini sudah *support* Kindle dan NOOK, kelebihan lain yang dimiliki Corona SDK yaitu tanpa fragmentasi karena secara otomatis memilih aset yang

tepat berdasarkan resolusi layar dan skala *content smartphone*, tergantung pada ukuran layar perangkat pengguna. Selain memiliki kelebihan Corona SDK juga memiliki sisi kekurangan, di antaranya untuk melakukan *build* ke iOS harus menggunakan Mac dan berbayar, harus selalu terkoneksi internet setiap melakukan *build*, dan komputer yang digunakan memiliki resolusi layar yang tinggi (Fernandez, 2015:4).

## **2.3 Lua**

Lua adalah bahasa pemrograman yang dikembangkan pada tahun 1993 oleh Roberto Ierusalimschy, Luiz Henrique de Figueiredo, dan Waldemar Celes di Pontifical Catholic University dari Rio de Janeiro di Brazil. Lua adalah bahasa Portugis yang berarti bulan, pada awalnya pengucapan yang sebenarnya adalah "LOO-ah". Lua memiliki lisensi dari MIT, lisensi yang tidak terbatas yang pada dasarnya memungkinkan untuk menggunakan Lua kapan pun selama tetap pada pemberitahuan dan izin hak cipta (Emmerich, 2009:1).

### **2.3.1 Fitur pada Lua**

Secara umum Lua digambarkan sebagai bahasa pemrograman multi paradigma yang menyediakan seperangkat kecil atas fitur-fitur umum yang dapat dikembangkan lebih lanjut untuk memenuhi ragam kebutuhan yang berbeda-beda, karenanya Lua tidak menyediakan fitur yang lengkap dan kompleks yang hanya berfokus pada satu paradigma pemrograman. Sebagai contoh, Lua tidak dengan

jelas mendukung *inheritance* yang merupakan kemampuan untuk menurunkan sifat sebuah obyek dalam konsepsi pemrograman berorientasi obyek.

Secara umum, Lua berusaha untuk menyediakan fitur-fitur abstrak atau meta yang lebih fleksibel dan dapat dikembangkan jika dibutuhkan dibandingkan menyediakan seperangkat pustaka yang lengkap untuk memenuhi satu kebutuhan tertentu. Hal tersebut menjadikan Lua sebagai bahasa pemrograman yang ringkas dan dapat secara mudah diadaptasikan untuk memenuhi beragam jenis kebutuhan. Lua merupakan bahasa pemrograman dinamis yang ditujukan untuk digunakan sebagai bahasa skrip, dan cukup ringkas untuk disisipkan dalam berbagai jenis platform utama. Lua hanya mendukung beberapa jenis struktur data atomik seperti : *boolean*, *floating point*, serta *string*. Jenis-jenis tipe atau struktur data lainnya seperti : larik, set, ataupun *list* direpresentasikan dalam Lua melalui satu bentuk tipe data *table* (Setiasih, et. al. 2014:115-116).

### **2.3.2 Kelebihan dan Kekurangan Lua**

Pada umumnya, setiap bahasa pemrograman memiliki kelebihan dan kekurangan. Menurut Emmerich (2009:1) kelebihan yang dimiliki Lua adalah mudah diperpanjang, sederhana, dan portabel. Sangat mudah dalam penulisan modul untuk menambah fungsional, dan juga mudah untuk menanamkan Lua sebagai bahasa skrip di program lain seperti permainan. Bahasa skrip adalah bahasa pemrograman yang digunakan di bagian tertentu dari aplikasi yang sangat berpengaruh dalam hasil suatu aplikasi. Sedangkan untuk kekurangannya, Lua hanya mendukung beberapa jenis struktur data atomik seperti : *boolean*, *floating*

*point*, serta *string*. Jenis-jenis tipe dan struktur data lainnya seperti : larik, set, ataupun *list* direpresentasikan dalam Lua melalui satu bentuk tipe data yaitu *table*.

## 2.4 Pemodelan dan Elemen pada *Game*

Pada tahap ini adalah membuat setiap elemen pada *game*, seperti gambar yang diperlukan pada setiap tampilan *game*, membuat suara *background* dan efek, serta menjelaskan cara pembuatan pada setiap elemennya.

### 2.4.1 Adobe Photoshop

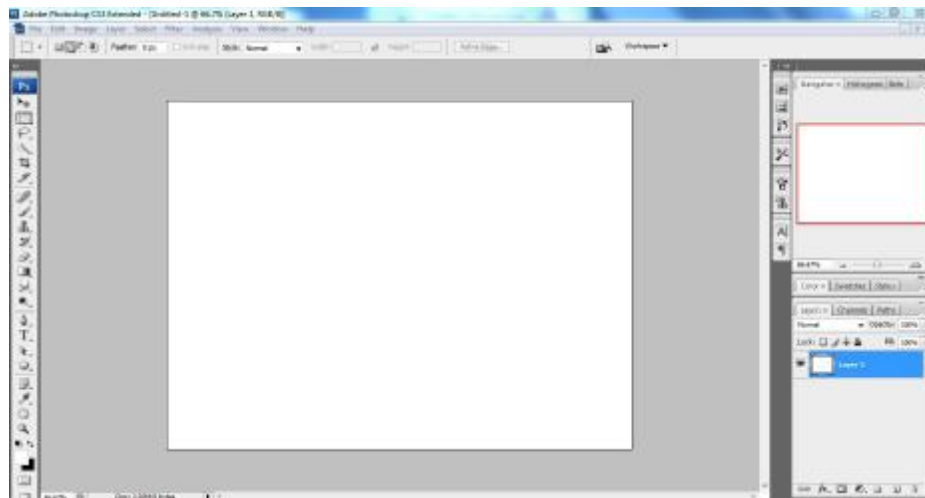
Adobe Photoshop adalah salah satu perangkat lunak canggih yang dapat digunakan untuk membuat, menyunting, dan memanipulasi tampilan termasuk mengoreksi warna dan memberikan efek tampilan atas sebuah gambar atau foto. Hasil dari program Adobe Photoshop merupakan sebuah gambar atau *image* yang di dalam program komputer grafis terbagi menjadi dua kelompok, yaitu gambar *bitmap* dan *vektor*.

Kualitas sebuah gambar Adobe Photoshop sangat tergantung pada jumlah *pixel* pada gambar tersebut. Oleh sebab itu *user* harus menentukan dimensi *pixel* yang diperlukan, resolusi gambar yang akan dibuat, serta menggunakan monitor dengan resolusi yang menunjang tampilan gambar.

Adobe Photoshop menyediakan berbagai peranti yang akan membantu *user* membuat gambar. *User* dapat memformat tampilan gambar tersebut dengan menggunakan filter yang telah disediakan oleh program ini. Dengan menggunakan peranti filter, *user* dapat memberikan efek-efek tertentu untuk objek



gambar, atau bahkan memberikan filter pada masing-masing *layer* sehingga dihasilkan sebuah gambar yang seperti *user* inginkan (Madcoms, 2005:1).



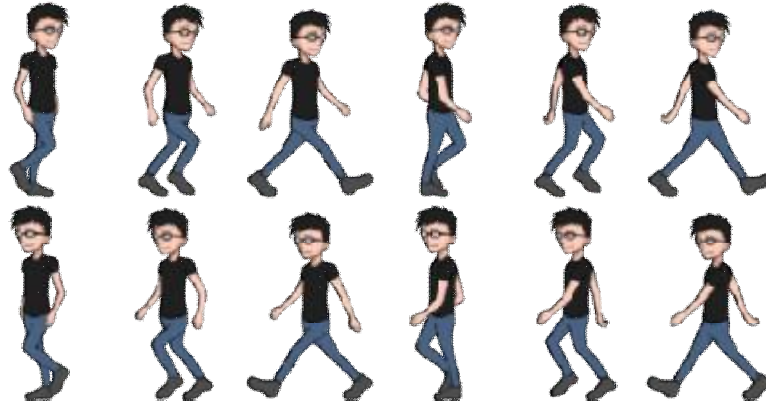
Gambar 2.1 Tampilan Adobe Photoshop

#### 2.4.2 *Sprite*

Animasi *sprite* serupa dengan teknik animasi tradisional, yaitu obyek yang diletakkan dan di animasikan pada bagian puncak grafik dengan latar belakang diam. *Sprite* adalah setiap bagian dari animasi yang bergerak secara mandiri, misalnya burung terbang, planet berotasi, bola memantul-mantul atau logo berputar. *Sprite* beranimasi dan bergerak sebagai obyek yang mandiri.

Dalam animasi *sprite*, sebuah gambar tunggal atau berurutan dapat ditempelkan dalam *sprite*. *Sprite* dapat di animasikan dalam satu tempat, seperti halnya planet berputar atau burung bergerak sepanjang garis lurus. Animasi *sprite* berbeda dengan animasi *frame*, dalam urutan masing-masing *frame*, *user* hanya dapat memperbaiki dari layar yang mengandung *sprite*. *User* tidak dapat

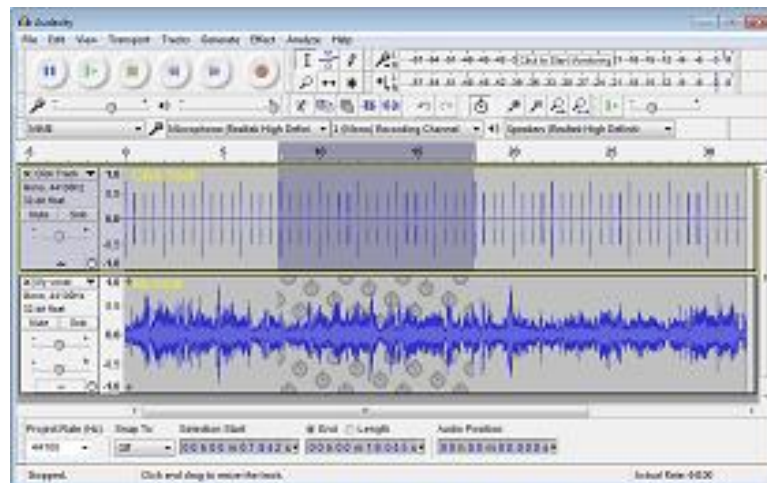
memperbaiki bagian dalam yang ditampilkan layar untuk masing-masing *frame*, seperti yang dapat dikerjakan pada animasi *frame* (Suyanto, 2007).



Gambar 2.2 Sprite (Sumber: [www.pirentest.com](http://www.pirentest.com))

### 2.4.3 Audacity

Audacity adalah perangkat lunak gratis, mudah digunakan, *multi-track*, perekam suara, *audio editor* untuk Windows, Mac OS X, GNU / Linux dan system operasi lainnya. Audacity juga menyediakan tampilan antarmuka yang diterjemahkan ke dalam banyak Bahasa (Audacity, 2008).



Gambar 2.3 Contoh Penggunaan Audacity

## 2.5 Target Pemain

*Game* yang dibangun ini ditujukan untuk siswa kelas enam Sekolah Dasar, *game* yang dibangun ini dikhususkan untuk mata pelajaran Ilmu Pengetahuan Alam, pada tahap pendidikan anak sekolah dasar, siswa akan cenderung lebih tertarik dengan permainan yang mudah dimainkan dan di dalamnya terdapat gambar animasi yang menarik perhatian. Dalam tahap ini siswa lebih mudah mengingat karena siswa berinteraksi langsung dengan bentuk atau karakter yang memiliki ciri warna menarik dan komunikatif sehingga lebih menyenangkan, sedangkan saat ini kebanyakan pembelajaran di sekolah masih menggunakan metode pengajaran konvensional dengan bantuan media yang sederhana dan buku pelajaran sebagai pedoman. Mata pelajaran Ilmu Pengetahuan Alam sekolah dasar selama ini juga menggunakan media yang kurang menarik bagi siswa sehingga menghilangkan semangat siswa untuk belajar serta keterbatasan pendidik dalam menggunakan media secara maksimal.

Ilmu Pengetahuan Alam merupakan salah satu mata pelajaran dalam salah satu Kurikulum Tingkat Satuan Pendidikan (KTSP) yang diberikan kepada siswa sekolah dasar untuk membekali siswa dengan cara mencari tahu tentang alam secara sistematis, sehingga Ilmu Pengetahuan Alam bukan hanya penguasaan kumpulan pengetahuan yang berupa fakta-fakta, konsep-konsep atau prinsip-prinsip saja tetapi juga merupakan suatu proses penemuan. Salah satu media yang dapat membangun kemampuan berpikir siswa dalam belajar Ilmu Pengetahuan Alam adalah dengan menggunakan *game* berbasis edukasi.

## 2.6 Android

Android merupakan sistem operasi untuk perangkat *mobile* yang berbasis Linux dan bersifat terbuka atau *open source* dengan lisensi GNU yang dimiliki Google (Wahana Komputer, 2013:iii). Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, GoogleInc. membeli AndroidInc. yang merupakan pendatang baru yang membuat peranti lunak untuk *smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *OpenHandset Alliance*, konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilis perdana Android, 5 November 2007. Android bersama *OpenHandset Alliance* 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 *GoogleMail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *OpenHandsetDistribution* (Hermawan, 2011:2).

### 2.6.1 Arsitektur Android

Menurut Hermawan (2011:5–6) secara garis besar arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

## 1. Application dan Widgets

*Application* dan *Widgets* ini adalah *layer* di mana pengguna berhubungan dengan aplikasi saja, di mana biasanya pengguna *download* aplikasi kemudian melakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain.

## 2. Application Frameworks

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

Sehingga dapat disimpulkan *Application Frameworks* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.

### 3. Libraries

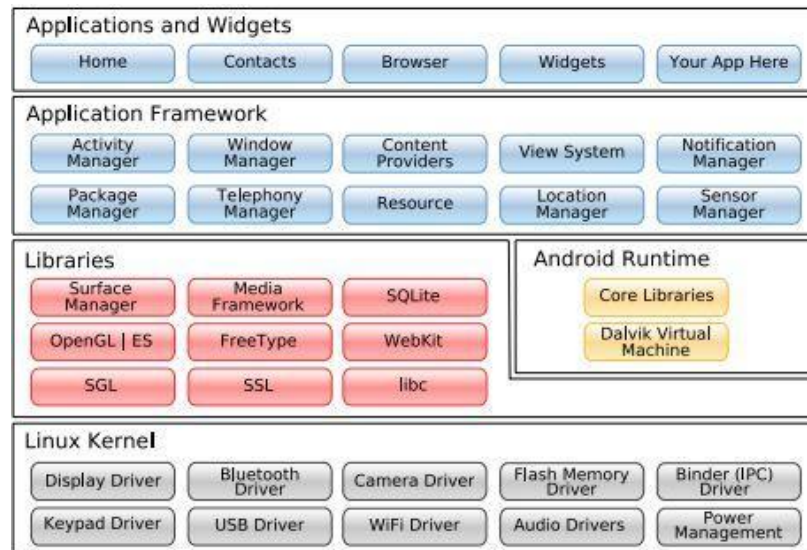
*Libraries* ini adalah *layer* di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

### 4. AndroidRun Time

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

### 5. Linux Kernel

*Linux Kernel* adalah *layer* di mana inti dari sistem operasi Android itu berada. Berisi *file-file* sistem yang mengatur sistem *processing*, *memory*, *resources*, *drivers*, dan sistem-sistem operasi Android lainnya.



Gambar 2.4 Arsitektur Android (Sumber: [www.windynovi.com](http://www.windynovi.com))

### 2.6.2 Software Development Kit (SDK)

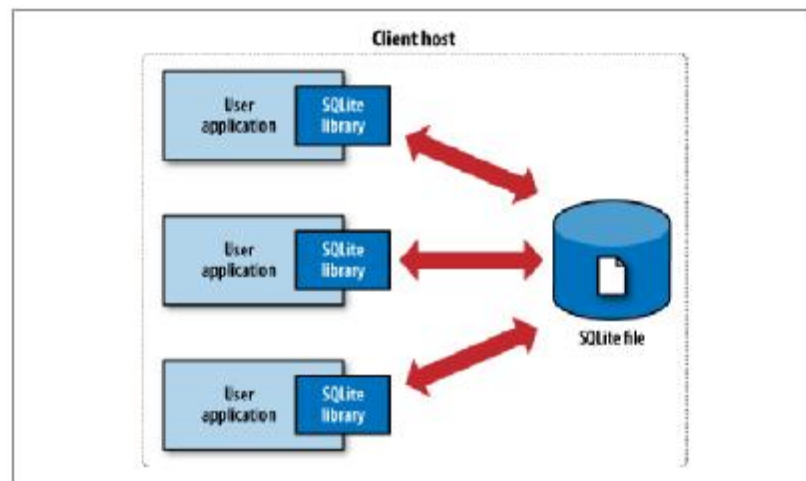
Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk *smartphone* yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di *release* 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 kesempatan untuk membuat aplikasi yang dibutuhkan dan bukan merupakan aplikasi bawaan *Handphone / Smartphone* (Safaat, 2011:14).

Menurut Safaat (2011:14–15) beberapa fitur-fitur Android yang paling penting adalah :

- *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- Mesin virtual *Dalvik* dioptimalkan untuk perangkat *mobile*.
- *Integrated Browser* berdasarkan *engine open source WebKit*.
- Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES 1.0*.
- SQLite untuk penyimpanan data.
- Media Support yang mendukung audio, video, gambar, dan GSM Telephony.
- Bluetooth, EDGE, 3G, dan WiFi.
- Kamera, GPS, kompas, dan *accelerometer*.
- Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging*, profil dan kinerja memori.

## 2.7 SQLite

SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi SQL *interface* dan penggunaan memory yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite di Android termasuk dalam Android runtime, sehingga setiap versi dari Android dapat membuat *database* dengan SQLite (Safaat, 2014:183–184).



Gambar 2.5 Arsitektur SQLite (Sumber: [www.dev.xbata.com](http://www.dev.xbata.com))

## 2.8 SublimeText Editor

SublimeText adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform* sistem operasi dengan menggunakan teknologi Python API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. SublimeText bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*)



dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki *linsensi* aplikasi gratis (Haughee, 2013:23).



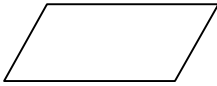
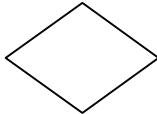

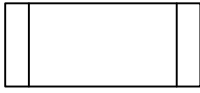
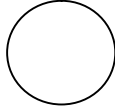

Pada tugas akhir ini penulis menggunakan SublimeText editor, Sublime dapat melakukan integrasi langsung dengan Corona SDK, dan untuk pencarian kata pada SublimeText juga relatif sangat mudah.

## 2.9 Flowchart

Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, di mana masing-masing simbol merepresentasikan suatu kegiatan tertentu. Flowchart diawali dengan penerimaan *input*, pemrosesan *input*, dan diakhiri dengan menampilkan *output*.

Dari pengertian tersebut, flowchart dapat diartikan sebagai gambaran urutan kerja sistem dengan menggunakan simbol-simbol tertentu yang digambarkan dalam bentuk tabel yang terpecah atas jalur-jalur berdasarkan *entity*. Selain data *flow* dan data *context*, flowchart atau diagram alir juga merupakan bagian dari data proses. Flowchart bisa dibuat sebelum maupun setelah pembuatan program, flowchart yang dibuat sebelum membuat program digunakan untuk mempermudah pembuat program untuk menentukan alur logika program, sedangkan yang dibuat setelah pembuatan program digunakan untuk menjelaskan alur program kepada orang lain. Berikut ini merupakan tabel dari simbol flowchart yang sering dipergunakan dalam menggambarkan rangkaian program dari awal sampai akhir (Wiryanto, 2005:12–13).

Tabel 2.1 Komponen pada Flowchart (Sumber: [www.zakwanirvani.blogspot.com](http://www.zakwanirvani.blogspot.com))

Simbol	Keterangan
	<b>Terminal</b> , sebagai awal ( <i>start</i> ) dan akhir ( <i>end</i> ) suatu bagan alir
	<b>Proccess</b> , sebagai himpunan penugasan yang akan dilaksanakan secara berurutan
	<b>Input</b> , sebagai penampung data sebelum dimasukkan ke dalam <i>memory</i> komputer
	<b>Decission</b> , sebagai cabang untuk memutuskan arah percabangan sesuai kondisi yang dipenuhi
	<b>Dokumen</b> , sebagai input yang berasal dari dokumen atau output yang dicetak ke kertas
	<b>Predefined Process</b> , Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain
	<b>Connector</b> , sebagai penghubung apabila bagan alir terputus, misal pergantian halaman
	<b>Flowline</b> , sebagai penunjuk arah proses