

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Huruf Arab**

##### **2.1.1 Sejarah Huruf Arab**

Menurut Azhar (2003) huruf arab sudah ada sejak berabad-abad yang lalu, ketika Al Quran pertama kali diturunkan menggunakan arab, karena pada kali pertama Al Quran diturunkan yaitu di Arab. Kenapa penting bagi kita mempelajari huruf Arab, karena huruf ini dipergunakan dalam penulisan Al Quran. Kita umat Islam wajib hukumnya untuk mempelajari Al Quran. Al Quran adalah kitab suci yang diwahyukan kepada Nabi Muhammad SAW. Yang menjadi petunjuk bagi seluruh umat manusia. Al Quran diturunkan untuk menjadi pegangan bagi mereka, yang ingin mencapai kebahagiaan dunia dan akhirat. Tidak diturunkan hanya untuk satu umat atau untuk satu abad, tapi untuk seluruh umat manusia dan untuk sepanjang masa, karena itu luas ajaran-ajarannya adalah sama dengan luasnya umat manusia. Maka dari itu, anak-anak harus mempelajari huruf arab agar dapat memahami kandungan Al Quran secara keseluruhan.

Huruf ini banyak dipakai di timur tengah. Berdasarkan kutipan dari situs internet, dengan narasumber Kiai Abdul Ghafur, bahwa setiap huruf Arab itu ada penjaganya, yaitu para Malaikat dan Sirullah. Seluruh huruf Arab itu bermula dari alif. Huruf alif bermula dari hamzahnya, dan selanjutnya bermula dari titik. Kemudian atas tujuh kasyfi dan ilhamy para sufi itu diberi kemampuan merangkai wifiq atau dalam istilah sehari-hari biasa disebut dengan rajah. Perkembangan berikutnya susunan wifiq itu bisa muncul sebagai khadam atau penjaga, bisa Jin

dan bisa Malaikat. Munculnya khadam Jin dan Malaikat itu disebabkan oleh dua hal: bisa karena metode pendekatan ruhaninya terhadap wifiq tersebut, atau karena memang struktur wifiq itu memunculkan sihir tertentu sehingga justru khadam Jin yang muncul, walaupun misalnya Jinnya Muslim. Huruf-huruf Arab adalah Rahasia nama Allah, dan sesungguhnya huruf itu adalah asma-asma Allah Ta'ala. Masing-masing huruf itu memiliki titik-titik organik di dalam jiwa spiritual, dan ibarat bintang-bintang bercahaya yang menerangi alam langit jiwa kita (Azhar:2003).

### **2.1.2 Perkembangan Huruf Arab**

Menurut Azhar (2003) huruf arab termasuk jenis huruf yang bersuara sebagaimana huruf latin yang telah kita kenal dan dipakai saat ini. Tidak seperti huruf China atau huruf Mesir kuno atau hieroglif dimana masing-masing hurufnya berupa suatu simbol mewakili gambar-gambar tertentu. Meskipun huruf arab memiliki kesamaan bentuk dengan huruf latin, namun berbeda dalam hal cara penulisannya. Perbedaan yang paling pokok adalah huruf arab ditulis dari kanan ke kiri sedangkan huruf latin yang kita kenal cara penulisannya dari kiri ke kanan dan mempunyai bentuk penulisan yang berbeda untuk masing-masing hurufnya tergantung letak dari penulisannya.

Huruf arab saat ini telah mengalami kemajuan, dari desain dan variasi hurufnya. Ini dapat dilihat dari karya seni seperti Kaligrafi yang memperindah huruf arab ini. Bahkan dari kumpulan-kumpulan huruf ini dapat terbentuk suatu benda atau sebuah tulisan lain. Huruf Arab terdapat 28 huruf, seperti pada Gambar 2.1.

خ Kho	ح Haa'	ج Jiim	ث Tsaa'	ت Taa'	ب Baa'	أ 'Alif
ص Saad	ش Shiin	س Siin	ز Zaayn	ر Raa'	ذ (Dzh)aal	د Daal
ق Qaaf	ف Faa'	ع Ghayn	ع 'Ayn	ظ (Zh)aa'	ط Taa'	ض Daad
ي Yaa'	و Waaw	ه Haa'	ن Nuun	م Miim	ل Laam	ك Kaaf

Gambar 2.1 Huruf Arab  
Sumber (Yunindra: 2012)

### 2.1.3 Cara Menulis Huruf Arab

Menurut Susiyanto (2002), penulisan huruf arab dapat mengikuti kaidah-kaidah berikut :

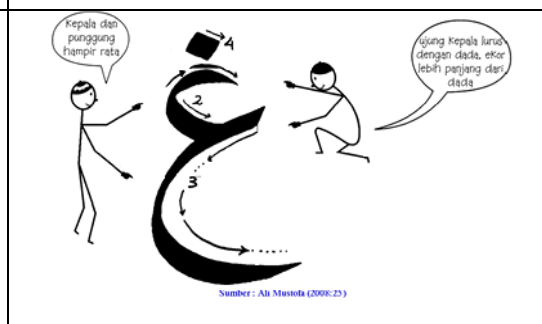
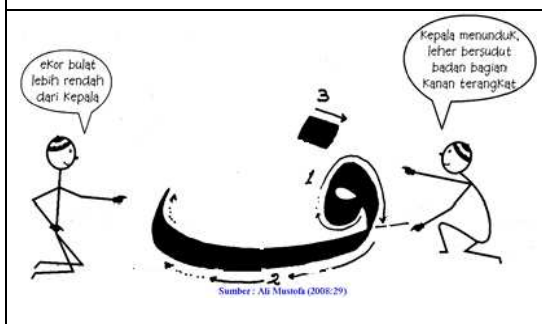
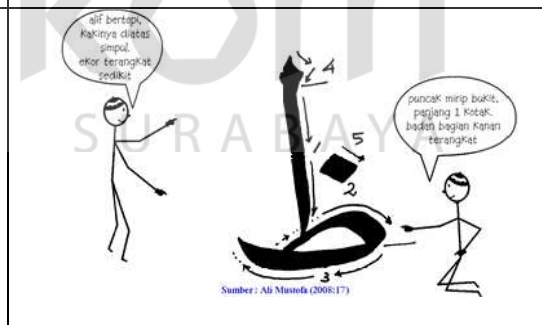
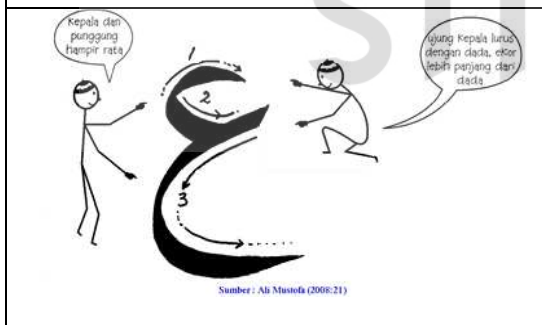
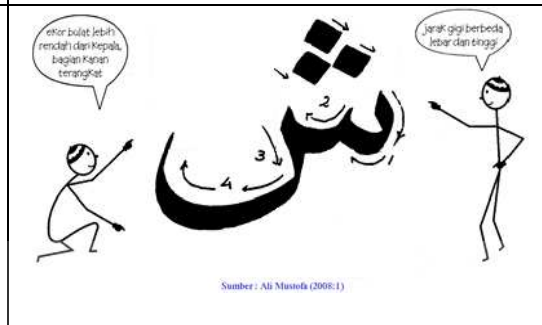
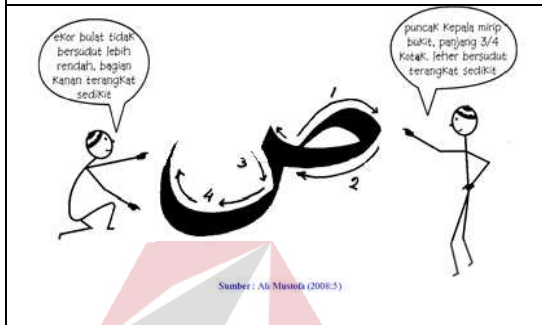
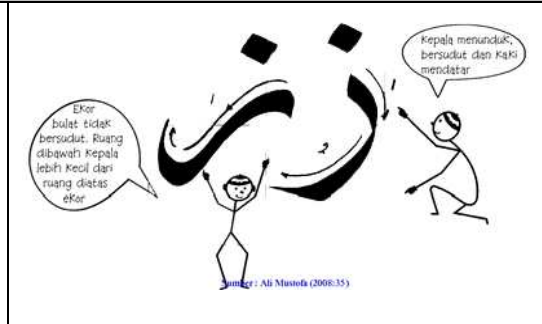
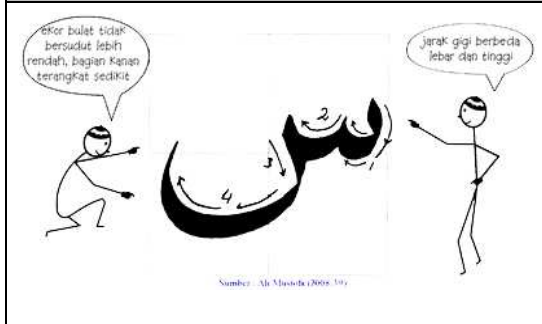
1. Penulisan huruf arab dimulai dari kanan ke kiri.
2. Jumlah huruf arab atau huruf hijaiyah ada 28 huruf.
3. Semua huruf Arab adalah konsonan, termasuk alif, wawu dan ya (sering disebut huruf *illat*), maka mereka memerlukan tanda vokal (sakal).

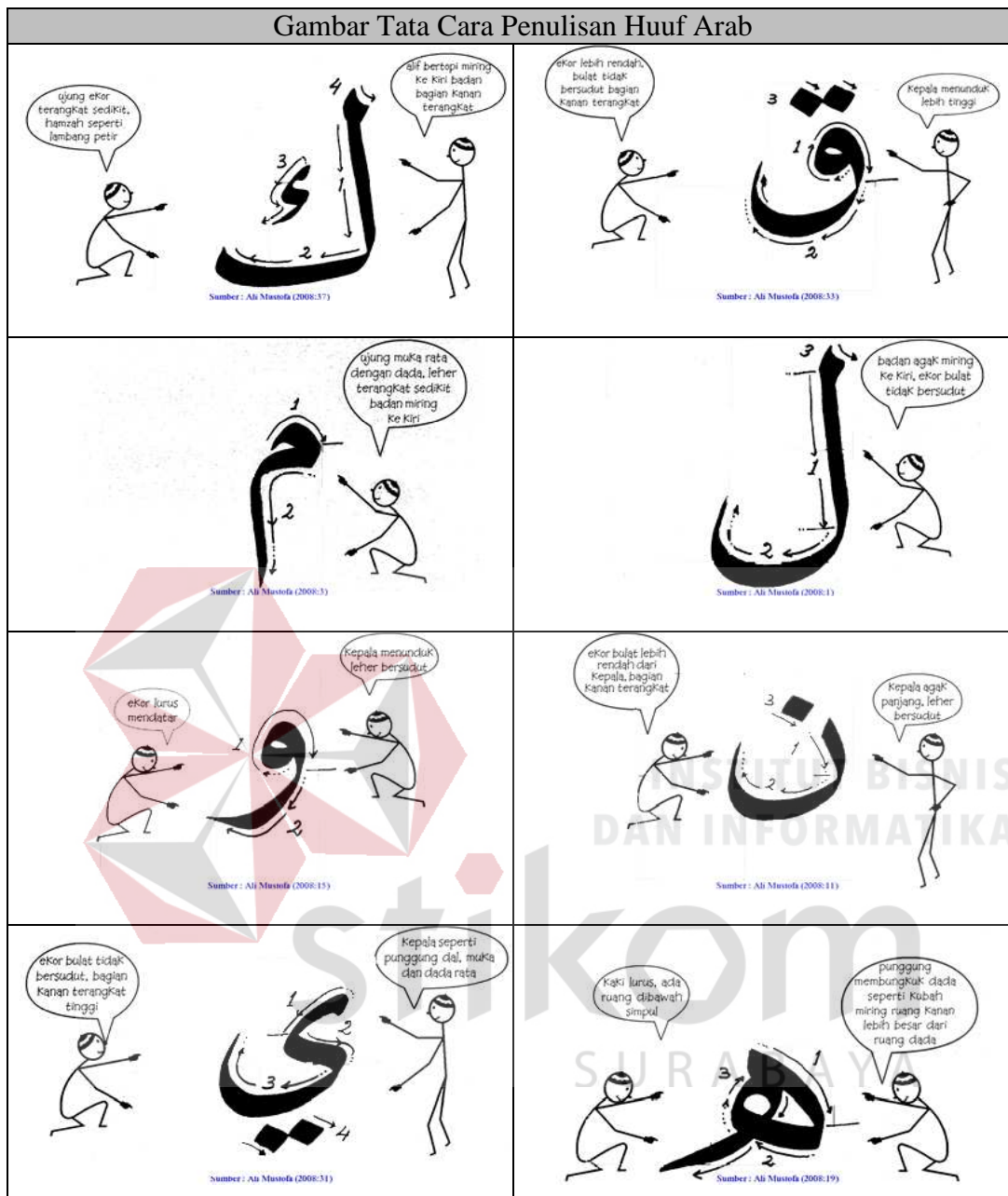
Menurut Mustofa (2008), metode menulis Al Qur'an (huruf hijaiyah atau arab) masih kurang mendapat perhatian, sehingga dibuatlah gambaran metodologi dengan kesempurnaan khot. Masing-masing 28 huruf arab mempunyai tata cara penulisan. Untuk jelasnya tata cara penulisannya seperti pada tabel 2.1.

Tabel 2.1 Gambar Tata Cara Penulisan Huuf Arab

Gambar Tata Cara Penulisan Huuf Arab	
<p>Sumber : Ali Mustofa (2008:3)</p>	<p>Sumber : Ali Mustofa (2008:1)</p>
<p>Sumber : Ali Mustofa (2008:3)</p>	<p>Sumber : Ali Mustofa (2008:5)</p>
<p>Sumber : Ali Mustofa (2008:15)</p>	<p>Sumber : Ali Mustofa (2008:11)</p>
<p>Sumber : Ali Mustofa (2008:23)</p>	<p>Sumber : Ali Mustofa (2008:19)</p>
<p>Sumber : Ali Mustofa (2008:31)</p>	<p>Sumber : Ali Mustofa (2008:27)</p>

Gambar Tata Cara Penulisan Huuf Arab





Menulis huruf arab ada suatu kaidah penulisannya, seperti arah dan urutan dalam menulis huruf arab. Seperti gambar yang ada pada tabel 2.1 adalah penulisan arab berdasarkan arah penulisan dan urutan penulisan huruf arab. Pada tabel tersebut terdapat tanda panah dan penomoran, jadi arah penulisannya berdasarkan urutan nomor arah panah. Aturan umum penulisan huruf arab

dimulai dari arah kanan ke kiri atau dari atas ke bawah. Seperti contoh gambar pada tabel 2.1 huruf arab "ba", arah panah nomor satu dari arah kanan kemudian ke kiri nomor dua kemudian pada huruf kha nomor satu dimulai dari atas kemudian kebawah nomor dua. Berikut diterangkan cara menulis huruf hijaiyah sesuai dengan gambar pada tabel 2.1. Pada aplikasi yang akan dibuat akan memberikan pengenalan mengenai tata cara penulisan huruf arab sesuai dengan kaidah yang benar dalam menulis huruf arab.

## 2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android dipuji sebagai *platform mobile* pertama yang lengkap, terbuka dan bebas (Safaat, 2012).

### a) Lengkap (*Complete Platform*)

Para desainer dapat melakukan pendekatan yang kompeherensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.

### b) Terbuka (*Open Source Platform*)

*Platform* Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.

### c) Bebas (*Free Platform*)

Android adalah aplikasi yang bebas untuk *develope*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya

kenggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Android merupakan generasi baru *platform mobile*, yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, General Public Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah *copyleft* lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh di bawah terms. Android didistribusikan di bawah lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua dan seterusnya.

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan, namun bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah:

- a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- b. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat mobile.
- c. *Integrated browser* berdasarkan *engine open source WebKit*.
- d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware).
- e. SQLite untuk penyimpanan data (database).



- f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
- g. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
- h. Kamera, GPS, kompas, dan accelerometer (tergantung hardware).

Lingkungan development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

*Android Development Tools* (ADT) adalah plugin yang didesain untuk IDE *Eclipse* yang memberikan kita kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE *Eclipse*. Dengan menggunakan ADT untuk *Eclipse* memudahkan kita dalam membuat aplikasi *project* Android, membuat GUI aplikasi dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui *eclipse*. Dengan ADT juga bisa melakukan pembuatan *package* Android (.apk) yang digunakan untuk distribusi aplikasi Android yang kita rancang.

### **2.2.1 Versi Android**

Perkembangan sistem operasi Android dari awal pertama dipakai hingga saat ini terdapat berbagai versi Android yang telah dirilis. Adapun versi Android yang telah dirilis adalah sebagai berikut (Safaat: 2012) :

- a. Android versi 1.1
- b. Android versi 1.5 (*Cupcake*)
- c. Android versi 1.6 (*Donut*)

- d. Android versi 2.0 / 2.1 (*Eclair*)
- e. Android versi 2.2 Froyo (*Frozen Yoghurt*)
- f. Android versi 2.3 (*Gingerbread*)
- g. Android versi 3.0/3.1 (*Honeycomb*)
- h. Android versi 4.0 ICS (*Ice Cream Sandwich*)
- i. Android versi 4.1/4.2/4.3(*Jelly Bean*)
- j. Android versi 4.4 (*KitKat*)

### 2.2.2 Android SDK (*Software Development Kit*)

Menurut Safaat (2012), Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan, namun bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah:

- a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- b. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat mobile.
- c. *Integrated browser* berdasarkan *engine open source WebKit*.
- d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware).
- e. SQLite untuk penyimpanan data (database).

- f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
- g. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
- h. Kamera, GPS, kompas, dan accelerometer (tergantung hardware).
- i. Lingkungan development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

### 2.2.3 Arsitektur Android

Menurut Safaat (2012), secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut:

#### a. Application dan Widgets

Application dan Widgets ini adalah *layer* yang berhubungan dengan aplikasi saja, biasanya dilakukan *download* kemudian di-*install* dan aplikasi tersebut dijalankan. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

#### b. Applications dan 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 *notifications*, dan

sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur dirancang agar dapat dengan mudah menggunakan kembali komponen yang sudah digunakan (*reuse*).

Jadi, *Applications Frameworks* ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer ini lah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telpon.

Komponen-komponen yang termasuk didalam *Applications Frameworks* adalah sebagai berikut:

a) *Views*

Digunakan untuk mengambil sekumpulan *button*, *list*, *grid*, dan *text box* yang digunakan didalam antarmuka pengguna.

b) *Content Provider*

Digunakan untuk memungkinkan aplikasi mengakses data dari aplikasi lain (seperti *contacts*) atau untuk membagikan data mereka sendiri.

c) *Resource Manager*

Digunakan untuk mengakses sumber daya yang bersifat bukan *code* seperti *string* lokal, *bitmap*, deskripsi dari *layout file* dan bagian eksternal lain dari aplikasi.

d) *Notification Manager*

Digunakan untuk mengatur tampilan peringatan dan fungsi-fungsi lain.

e) *Activity Manager*

Mengatur siklus dari aplikasi dan menyediakan navigasi *backstack* untuk aplikasi yang berjalan pada proses yang berbeda.

f) *Package Manager*

Untuk meacak aplikasi yang di-instal pada perangkat.

g) *Telephony Manager*

Berisi sekumpulan API yang diperlukan untuk memanggil aplikasi.

h) *Libraries*

Android memiliki sekumpulan library C/C++ yang digunakan oleh berbagai komponen dalam sistem Android. Kemampuan-kemampuan ini dilihat oleh para pengembang melalui kerangka kerja aplikasi. Berikut

beberapa *library* utama :

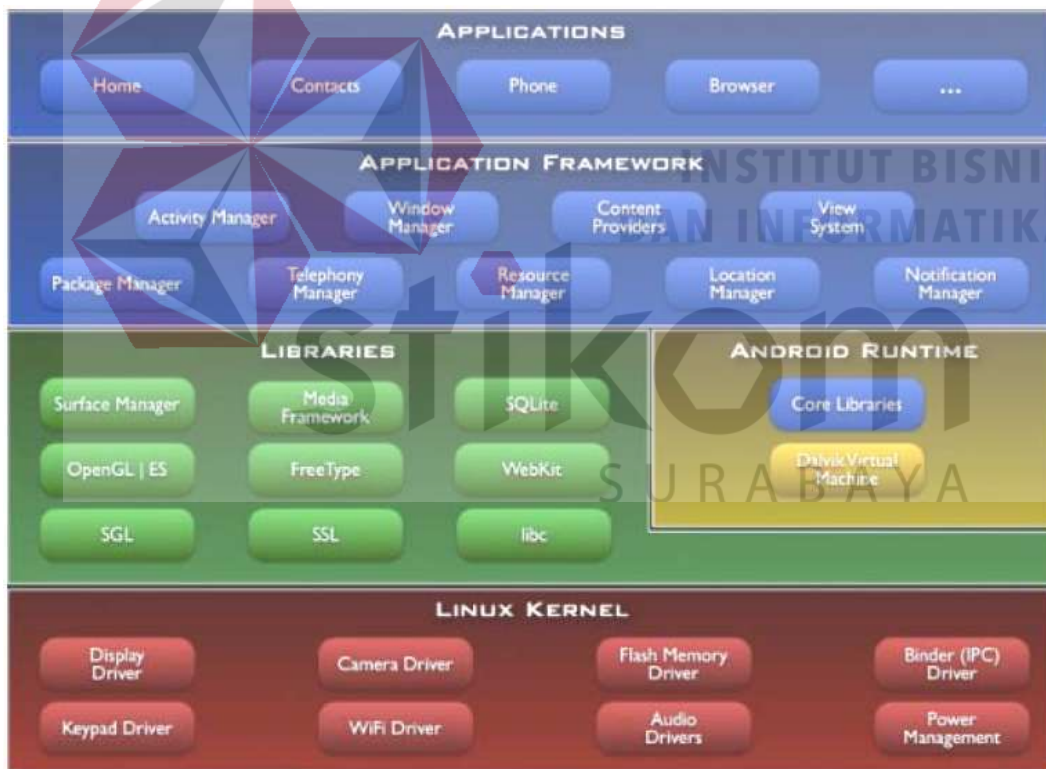
1) *System C Library*2) *Media Libraries*3) *Surface Manager*4) *LibWebCore*5) *Scalable Graphics Library*6) *3D Libraries*7) *Free Type Library*8) *SQLite*i) *Android Runtime*

Merupakan lokasi dimana komponen utama dari DVM ditempatkan. DVM dirancang secara khusus untuk Android pada saat dijalankan pada lingkungan yang terbatas, dimana baterai yang terbatas, CPU, memori dan penyimpanan data menjadi fokus utama. Android memiliki sebuah *tool* yang terintegrasi yaitu “dx” yang mengkonversi *generated byte code* dari (JAR) ke dalam file (DEX) sehingga *byte code* menjadi lebih efisien untuk dijalankan pada prosesor yang kecil. Hal ini

memungkinkan untuk memiliki beberapa jenis dari DVM berjalan pada suatu peralatan tunggal pada waktu yang sama. *Core libraries* ditulis dalam bahasa Java dan berisi kumpulan *class*, I/O dan peralatan lain.

j) Linux Kernel

Arsitektur Android berdasarkan pada Linux 2.6 kernel yang dapat digunakan untuk mengatur keamanan, manajemen memori, manajemen proses, *network stack*, dan *driver model*. Kernel juga bertindak sebagai lapisan abstrak antara perangkat keras dan seluruh *software stack*. Untuk lebih jelasnya, seperti pada Gambar 2.2 yang menunjukkan arsitektur dari Android.






Gambar 2.2 Arsitektur Android  
Sumber (elinux.org)





### 2.3 Gesture

*Gesture* adalah suatu bentuk ungkapan secara alami (*non-verbal*) berupa gerak-gerak atau isyarat untuk mengekspresikan atau menyampaikan suatu pesan (Haviland, 1998). Gerakan-gerakan yang dilakukan biasanya berupa gerakan tangan, wajah atau bagian lain dari tubuh.

*Gesture* memungkinkan pengguna untuk berinteraksi dengan aplikasi anda dengan memanipulasi objek layar anda. Tabel 2.2 menunjukkan set inti gerakan (*gesture*) yang didukung dalam Android (Android, 2010).

Tabel 2.2 *Gesture* pada Android

Jenis <i>Gestures</i>	Fungsi	Aksi
 <b>Touch</b>	Triggers fungsi <i>default</i> / standart untuk item tertentu	Press, Lift
 <b>Long Press</b>	Memasuki/ memilih modus data seleksi. Memungkinkan untuk memilih satu atau lebih item dalam pandangan dan bertindak berdasarkan data menggunakan panel tindakan kontekstual. Hindari menggunakan tekan lama untuk menunjukkan menu. Triggers kontekstual fungsi default untuk item tertentu.	Press, wait, lift
 <b>Swipe</b>	<i>Scroll overflowing</i> konten, atau menavigasi antara tampilan dalam hirarku yang sama.	Press, move, lift

Jenis <i>Gestures</i>	Fungsi	Aksi
 <p><b>Drag</b></p>	Menata kembali data dalam tampilan atau memindahkan data ke dalam wadah (folder misalnya pada <i>home screen</i> ).	Long Press, move, lift
 <p><b>Double touch</b></p>	Memperbesar ( <i>zoom</i> ) ke konten. Juga digunakan sebagai isyarat sekunder untuk pemilihan teks.	Two touches in quick succession
 <p><b>Pinch open</b></p>	Memperbesar ke konten.	2-finger press, move outwards, lift
 <p><b>Pinch close</b></p>	Memperbesar keluar dari konten/ memperkecil ( <i>zoom out</i> )	2-finger press, move inwards, lift

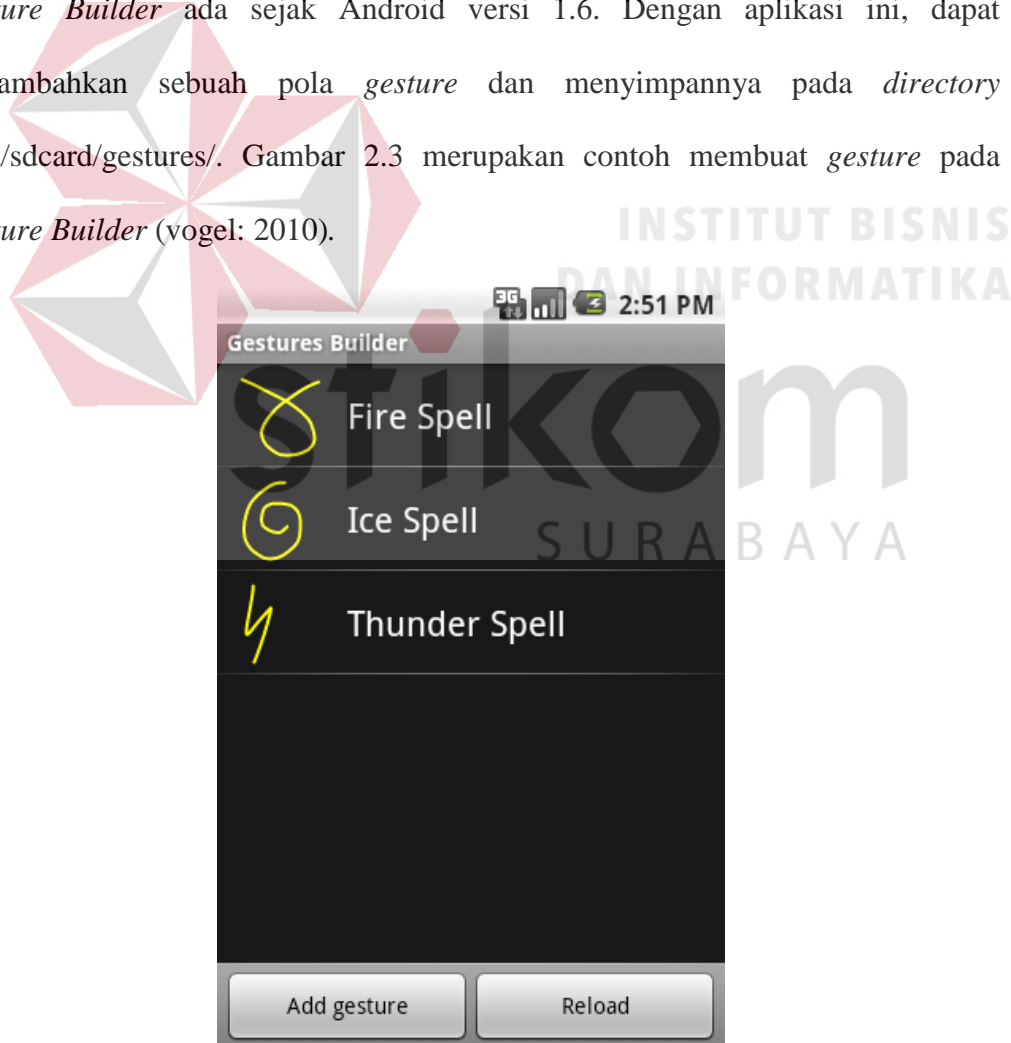
Pada Android terdapat fitur *gesture*, tiap view memiliki `onTouchEvent` yang diasosiasikan dengan view tersebut. Hal tersebut bisa digunakan dengan *gesture detector* yang bertugas mengeset *gesture listener*. Berikut beberapa *gesture* yang bisa dimungkinkan diantarmuka `OnGestureListener`:

- a) `onDown()`: memberitahu ketika *event tap down* terjadi.
- b) `onFling()`: memberitahu ketika *tap down* dan ada gerakan melempar terjadi.



- c) `onLongPress()`: memberitahu ketika klik panjang terjadi.
- d) `onScroll()`: memberitahukan ketika *scroll* terjadi.
- e) `onShowPress()`: memberitahukan ketika tap down terjadi sebelum adanya gerakan/dilepas.
- f) `onSingleTapUp()`: memberitahukan ketika *tap up* terjadi.

Sebelum membuat *project* yang akan menerapkan *gesture*, terlebih dahulu dibuat sebuah *library* dari pola-pola *gesture* yang akan didefinisikan. *Gesture Builder* adalah aplikasi tambahan yang ada dalam *Android Virtual Device* (AVD). *Gesture Builder* ada sejak Android versi 1.6. Dengan aplikasi ini, dapat menambahkan sebuah pola *gesture* dan menyimpannya pada *directory* `/mnt/sdcard/gestures/`. Gambar 2.3 merupakan contoh membuat *gesture* pada *Gesture Builder* (vogel: 2010).



Gambar 2.3 Membuat *Gesture*

### 2.3.1 Penggunaan Gesture

Gesture adalah bentuk digambar tangan pada paket `Android.gesture.screen`. Gesture menyediakan *libraries* untuk mengenali dan menangani dengan cara yang sederhana. Pertama, setiap SDK memiliki contoh program yang dapat digunakan untuk membangun sebuah koleksi gerakan (gesture) pada `platforms/Android-2.0/samples/GestureBuilder/` (Android: 2010).

Pada penggunaan tools yang ada pada Android yaitu *gestures builder* terdapat beberapa kekurangan, yaitu masih belum bisa mendeteksi lebih dari satu stroke dan mendeteksi pola vertikal. Hasil dari uji coba yang dilakukan, tidak bisa mendeteksi *gesture* yang berpola vertikal, hal itu juga sudah dibahas pada forum Android dan web developers Android.

Bila dibuat pola secara vertikal maka sistem tidak akan pernah mendeteksi, karena bila pola yang digambar diproses warna *gesture* adalah hitam, bila warna biru maka sistem tidak akan mendeteksi.

Bila dengan memakai tools *gestures builder* tidak berhasil, memakai metode \$N dalam permodelan *gestures* yang dikembangkan oleh seorang professor Sistem Informasi di *University of Washington* dengan asistennya Lisa Anthony. Metode \$N adalah *multistroke gesture recognizer* sederhana yang berbasis *template matching* pengembangan dari *unistroke* \$1. *Multistroke recognizer* adalah desain *stroke recognizer 2-D* yang dirancang untuk prototipe basis *gesture* antarmuka. \$N dibuat berdasarkan \$1 *unistroke recognizer*. Pengembang dan peneliti telah membuat dalam versi beberapa bahasa pemrograman Python, C#, C++, Objective-C, Java, JavaME dan JavaScript.

Tujuan dibuatnya \$N agar bisa lebih berguna, ringkas, mempermudah dalam penggabungan *multistroke recognizer deployable* di beberapa semua platform yang mendukung. \$N jauh lebih serbaguna dalam beberapa hal, yaitu :

1. Pengenalan *gestures* yang terdiri beberapa *stroke*.
2. Otomatis *generalizing* dari satu *multistroke template* untuk semua *multistroke*, dengan *alternative stroke order* dan *stroke direction*.
3. Mengenali dengan satu dimensi seperti garis.
4. Pendukung batas rotasi invarians (ketetapan).

Keterbatasan \$1 meliputi pada pembatasan dua dimensi *unistroke*, ketidakmampuan mengenali garis dan ketidakmampuan untuk mendukung orientasi yang saling bergantung. Semua itu telah diperbaiki oleh \$N.

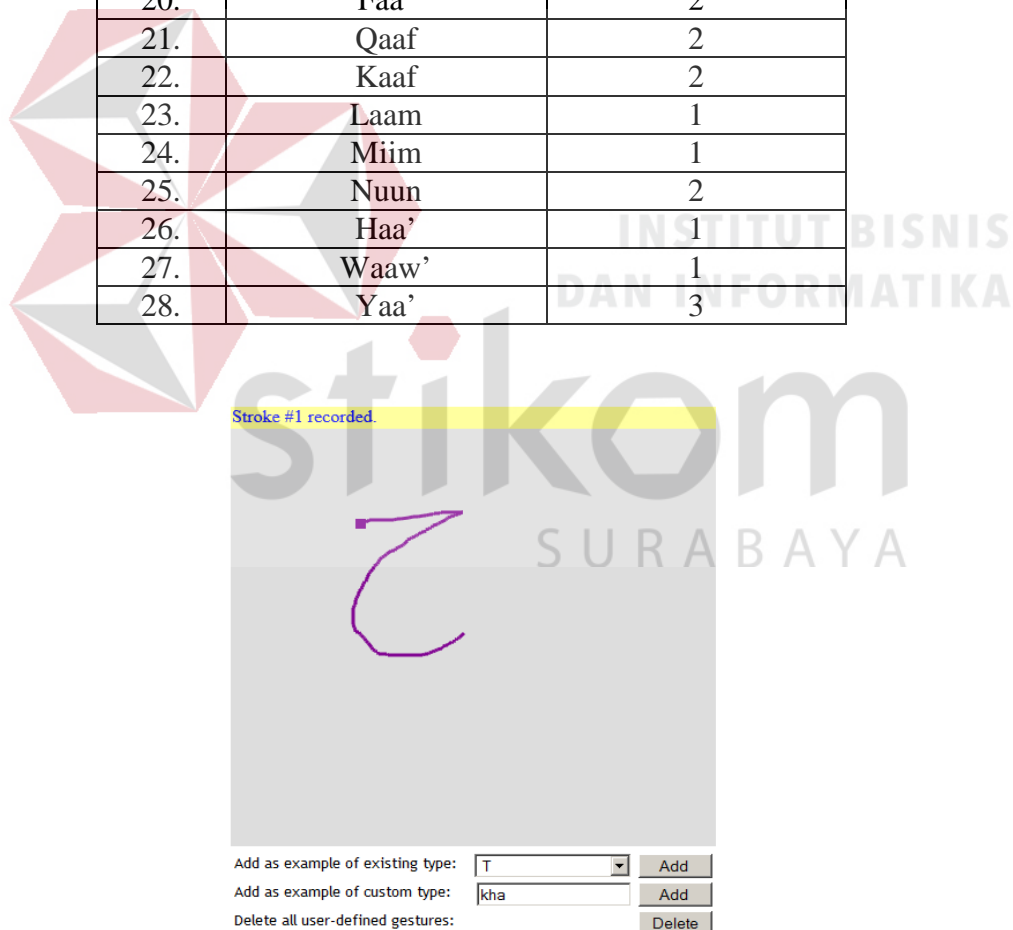
\$N secara otomatis *generalizes* contoh *multistrokes* untuk mencakup semua *stroke order* dan arah. Nantinya akan dapat membuat dan menentukan *mutistrokes* menggunakan urutan strokes dan arah yang diinginkan. Inti dari \$N adalah menyimpan setiap gerakan atau *gesture* berpindah

Pada metode \$N dapat mendeteksi jumlah *stroke*. Dikatakan satu *stroke* apabila awal gerakan tangan menyentuh layar kemudian akhir gerakan tangan menjauhi layar atau juga bisa dikatakan satu gerakan. Berikut adalah data huruf arab beserta jumlah *stroke* pada tabel 2.3, contoh gambar pada Gambar 2.4 dan Gambar 2.5.

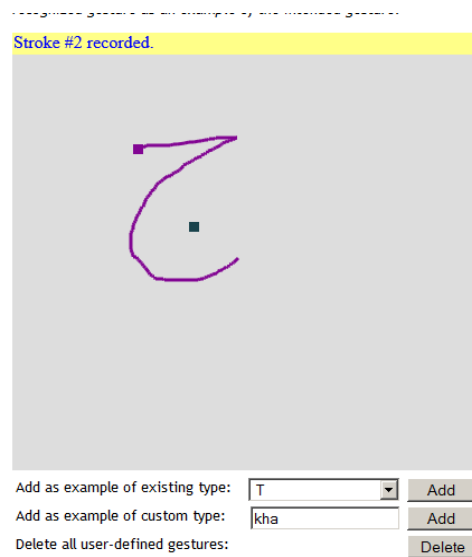
Tabel 2.3 Jumlah *Stroke* Huruf Arab

No	Huruf Arab	Jumlah Stroke
1.	'Alif	1
2.	Baa'	2
3.	Taa'	3
4.	Tsaa'	4
5.	Jiim	2

No	Huruf Arab	Jumlah Stroke
6.	Haa'	1
7.	Kho	2
8.	Daal	1
9.	(Dzh)aal	2
10.	Raa'	1
11.	Zaayn	2
12.	Siin	1
13.	Shiin	4
14.	Saad	1
15.	Daad	2
16.	Taa'	2
17.	(Zh)aa'	3
18.	'Ayn	1
19.	Ghayn	2
20.	Faa'	2
21.	Qaaf	2
22.	Kaaf	2
23.	Laam	1
24.	Miim	1
25.	Nuun	2
26.	Haa'	1
27.	Waaw'	1
28.	Yaa'	3



Gambar 2.4 Record Stroke Pertama Kha



Gambar 2.5 Record Stroke Kedua Titik

Metode \$N mempunyai beberapa fungsi sebagai metode pencocokan *gestures*. Cara kerja \$N mencocokkan nilai dari *input gestures* dengan data yang ada *gestures template*. Pencocokkan ini terdapat pada fungsi *Recognizes*.

## 2.4 Phonegapp

PhoneGap adalah sebuah *platform* aplikasi HTML 5 yang dapat membuat aplikasi dengan memanfaatkan teknologi web dan mendapatkan akses untuk APIs dan app stores (w3school: 2013). Dalam membuat aplikasi dengan menggunakan PhoneGap dapat menggunakan HTML, CSS, dan Javascript. PhoneGap merupakan solusi bagi *web developer* yang tertarik dalam pembuatan aplikasi di *smartphone*.

Aplikasi yang dikembangkan menggunakan PhoneGap dapat di-*deploy* ke beberapa *platform*, antara lain Android, iPhone, BlackBerry, Windows Mobile. Dukungan fitur *native* yang didukung oleh PhoneGap untuk beberapa platform berbeda-beda. Gambar 2.6 merupakan *list* fitur *native* untuk *platform* yang

didukung PhoneGap. Dari gambar tersebut dapat dilihat bahwa PhoneGap telah mendukung penuh secara utuh semua fitur pada iPhone 3GS keatas dan Android.

Pengembangan aplikasi *mobile* saat ini dianggap gagal, karena untuk membangun aplikasi untuk setiap perangkat (iPhone, Android, BlackBerry, dll) membutuhkan *frameworks* dan bahasa yang berbeda-beda pada setiap perangkat. Dengan adanya PhoneGap diharapkan untuk saat ini dapat menjadi sebuah jembatan antara aplikasi web dengan perangkat *mobile*. PhoneGap merupakan implementasi *open source* dari standar terbuka. Hal ini berarti pengembang dan perusahaan dapat menggunakan PhoneGap untuk aplikasi *mobile* yang *free*, komersial, *open source*, atau kombinasi dari semuanya.

	iOS iPhone / iPhone 3G	iOS iPhone 3GS and newer	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	hp WebOS	WP7	Symbian	bada
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
FILE	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
STORAGE	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗

Gambar 2.6 Fitur *Native* Phonegap (Android: 2010)

## 2.5 HTML5

HTML5 (*HyperText Markup Language*) merupakan generasi dari HTML sebelumnya, yang diciptakan untuk memperbaiki konten dan memperbarui teknologi multimedia yang ada pada HTML4. Standard HTML4 yang dijumpai banyak memiliki kelemahan untuk mendukung aplikasi web yang interaktif. Akibat hal ini banyak orang menambahkan fitur baru baik disisi aplikasi web ataupun disisi browser. Solusi ini dikenal dengan plugin dan salah satunya adalah *Flash* dan *Silverlight*.

HTML5 suatu spesifikasi atau standard yang dikeluarkan oleh W3C (*World Wide Web Consortium*) dan *Web Hypertext Application Technology Working Group* (WHATWG). WHATWG bekerja dengan bentuk web dan aplikasi, dan W3C bekerja dengan XHTML 2.0. Pada tahun 2006, mereka memutuskan untuk bekerja sama dan membuat versi baru dari HTML yaitu HTML5.

Alasan pembuatan HTML5 adalah:

1. Fitur baru harus didasarkan pada HTML, CSS, DOM, dan JavaScript.
2. Mengurangi kebutuhan untuk plugin eksternal (Seperti *Flash*).
3. Penanganan kesalahan yang lebih baik.
4. Lebih markup untuk menggantikan scripting.
5. HTML5 merupakan perangkat mandiri.
6. Proses pembangunan dapat terlihat untuk umum.

Fitur-fitur baru pada HTML5 adalah:

1. Unsur kanvas untuk menggambar.
2. Video dan elemen audio untuk media pemutaran.

3. Dukungan yang lebih baik untuk penyimpanan secara offline.
4. Elemen konten yang lebih spesifik, seperti artikel, footer, header, nav, section.
5. Bentuk kontrol form seperti kalender, tanggal, waktu, email, url, search.

## 2.6 Database Android

Android merupakan platform yang kompleks dan sangat bisa diandalkan. Tentu saja didalamnya ada *tool* untuk menyimpan data karena kebutuhan penyimpanan data sangat penting dalam sebuah aplikasi. Metode data storage pada Android ada banyak jenisnya, berikut beberapa di antaranya :

- a) *Shared Preferences*: untuk penyimpanan data sederhana, seperti *setting* aplikasi dan state pengaturan dari *user interface*.
- b) *Database SQLite* yang sudah *built-in* untuk menyimpan database yang kompleks seperti menyimpan record aplikasi.

*File standart java flat* untuk menyimpan data. Menggunakan method standar seperti *InputStream* dan *OutputStream*.