

BAB II

LANDASAN TEORI

2.1. Bahasa Inggris

Bahasa Inggris adalah bahasa Internasional selain yang digunakan untuk berhubungan antar negara, juga digunakan untuk memperdalam dan mengembangkan ilmu pengetahuan, karena sebagian besar buku ilmu pengetahuan didatangkan dari luar negeri.

(Izzan dan Mahfuddin, 2007:1)

2.1.1. Metode Belajar Bahasa Inggris

Izzan dan Mahfuddin (2007:3) mendefinisikan belajar sebagai suatu proses rumit yang menimbulkan kesulitan bagi orang muda maupun orang dewasa, tetapi berbeda dengan anak-anak dikarenakan proses belajar yang berbeda. Para ahli bahasa mengemukakan metode-metode pembelajaran bahasa sebagai berikut:

1. Metode *Gouin* (*La Methode Gouin*) oleh seorang Perancis yang bernama Gouin. Menurut Gouin pembelajaran bahasa seharusnya dimulai dimulai dengan bercakap-cakap dan membaca, serta tidak memulai dengan pelajaran tata bahasa (*Grammatical*). Kelebihan dari metode ini adalah untuk melatih berbicara dan mendengar.
2. *Berlitz Method*, yang disebut juga metode langsung (*the direct method*), metode alamiah (*natural method*), metode lisan (*oral method*), atau metode modern (*modern method*). Mengemukakan bahwa dalam mempelajari bahasa

jangan menggunakan terjemahan kecuali jika terpaksa. Metode ini hampir sama dengan metode *Gouin*, hanya benda atau barang harus diperhatikan.

3. *La Methode intuitive*, dinamakan juga metode panca indra, yaitu dengan pengajar mengucapkan kemudian didengar dan ditirukan oleh pelajar.
4. Metode elektik adalah kumpulan teknik penyajian bahan pelajaran yang disesuaikan dengan:
 - a. bahan pelajaran tertentu,
 - b. siswa tertentu, dan
 - c. hasil yang dituju.

Dalam metode ini pengajar yang menentukan teknik dalam membantu dan memudahkan siswa mempelajari bahan pelajaran dengan memperhatikan dan memperhitungkan kemampuan, keinginan, dan keadaan siswa.

5. Metode *all in one system* atau metode kesatuan adalah cara pembelajaran bahasa yang tidak boleh dipisah-pisah. Tujuan metode ini adalah siswa harus mengerti apa yang mereka dengar, katakan, dan baca. Metode ini sesuai dengan *Gestalt*, yaitu pengajar menyebutkan materi pelajaran, sedangkan pelajar mendengarkan dengan cermat, meniru, mempraktikan, membaca, kemudian menuliskan.

2.1.2. Langkah-Langkah Cara Mengajar Bahasa

Menurut Izzan dan Mahfuddin (2007:5), adapun langkah-langkah cara mengajar bahasa antara lain:

1. Langkah pertama
 - a. Tidak memberikan teks atau bacaan.
 - b. Tidak mengajarkan tata bahasa terlebih dahulu.

- c. Tidak menggunakan bahasa pengantar dengan bahasa asing (bagi para pemula) karena akan membuat para siswa bingung.
 - d. Mengajarkan bahasa Inggris atau bahasa asing lain yang komunikatif, mulai dari ucapan salam, ucapan terimakasih, berpisah, berkenalan, meminta maaf, menanyakan arah, menawarkan sesuatu, dan lain-lain.
2. Langkah kedua
- a. Pengajar menyampaikan kosakata dengan dibaca berulang-ulang, tetapi tidak ditulis kemudian pelajar menirukan apa yang dibaca oleh pengajar. Walaupun tidak berurutan, yang penting para siswa hafal kemudian menuliskannya.
 - b. Pada tahap pembelajaran tata bahasa, sebelumnya pengajar harus merangkum seluruh tata bahasa asing yang akan dipelajari agar para siswa mengetahui akhir pelajaran tata bahasa tersebut. Selanjutnya, pengajar menerangkan kepada pelajar dengan bahasa Indonesia. Apabila telah mengerti, pengajar dapat memberikan pelatihan percakapan maupun tulisan dan pekerjaan rumah.
 - c. Pelajar yang telah melalui dan memahami langkah-langkah diatas bisa melanjutkan ke pelajaran mengarang sederhana dan membaca teks-teks bahasa Inggris sederhana (bukan klasik).
3. Langkah ketiga

Langkah ketiga merupakan tahap akhir pembelajaran bahasa. Pelajar dapat mulai menggunakan buku pengantar berbahasa asing serta buku-buku ilmiah, novel, majalah, dan koran-koran berbahasa asing.

2.1.3. Kunci Ucapan dan Tekanan Bunyi

Menurut Peter (2006:XXI), pengucapan dan penekanan bunyi dalam pembelajaran melaftalkan bahasa Inggris ditunjukkan pada tabel 2.1 lambang bunyi dan tabel 2.2 tekanan bunyi.

Tabel 2.1 Lambang Bunyi

Vowels				
Lambang		Contoh		
British	Amerika	Kata	British	Amerika
i:	i	bee	bi:	bi
i	ɪ	hit	hit	hɪt
e	ɛ	yes	jes	jɛs
a	æ	sat	sat	sæt
ɑ:	ɑ	palm	pɑ:m	pam
ɔ	ɑ	ox	ɔks	aks
ɔ:	ɔ	law	lɔ:	lɔ
u	ʊ	book	buk	bʊk
u:	u	tool	tu:l	tul
ʌ	ə	cut	kʌt	kət
ə:	ə	bird	bə:d	bərd
e	ɛ	better	'betə	beder
Diphthongs				
ei	eɪ	name	nɛim	neɪm
ou	ou	no	nou	nou
ai	aɪ	my	mai	mai
au	aʊ	out	aut	avt
ɔɪ	ɔɪ	boy	bɔi	bɔɪ
iə	ɪə	mere	mɪə	mɪr
ɛə	ər	dare	dəə	dər
ɔə	ɔr	azyospore	'æzəspɔə	'æzəspər
uə	ʊr	tour	tuə	tʊr
ju:	ju	use	ju:s	jus

Konsonan

Konsonan		Contoh	
Lambang	Kata	British	Amerika
British/Amerika			
b	bob	bəb	bəb
d	dad	dæd	dæd
f	fife	faif	faif
g	gag	gæg	gæg
h	high	hai	haɪ
k	kick	kik	kɪk

Konsonan			
Lambang	Contoh		
British/Amerika	Kata	British	Amerika
l	lull	ɫʌl	ləl
m	mom	məm	mam
n	noon	nu:n	nun
p	pop	pɔp	pɑ
r	red	red	red
s	sauce	sɔ:s	sɔ:s
t	take'	teɪk	teɪk
v	vain	veɪn	veɪn
w	way	wei	weɪ
j	yell	jel	jel
z	zone	zoun	zoun
ŋ	sing	sɪŋ	sɪŋ
ʃ	shift	ʃift	ʃift
tʃ	church	tʃɔ:tʃ	tʃɔ:tʃ
ʒ	pleasure	'pleʒə	dʒus
dʒ	juice	dʒus:s	dʒus
θ	thin	θin	θɪn
ð	that	ðæt	ðæt

Tabel 2.2 Tekanan Bunyi

Tanda baca	Keterangan
'	Menunjukkan tekanan bunyi utama dan diletakkan di atas suku kata yang mendapat tekanan seperti dalam ballon (bə'lu:n, bə'lun)
,	Menunjukkan tekanan bunyi kedua dan diletakkan di bawah suku kata yang mendapat tekanan seperti dalam interfere (,intə'fiə, ɪnər'fɪr)

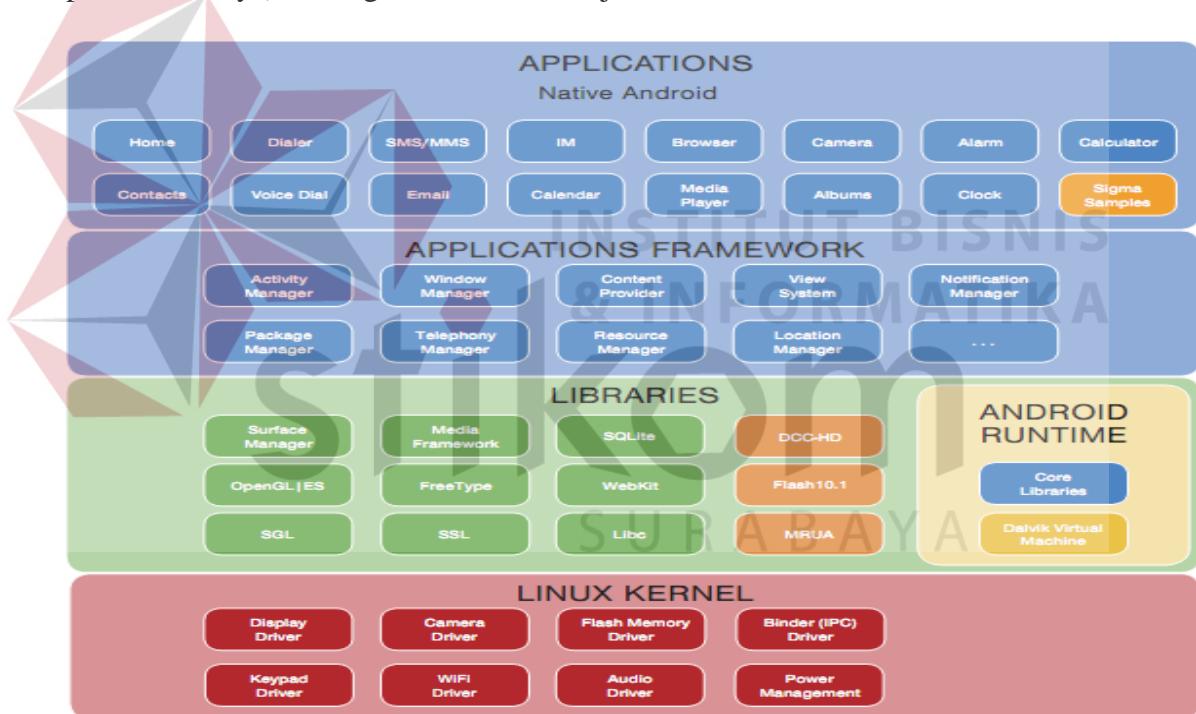
2.2. Android

Android dirilis pada tahun 2007 saat Google di bawah aliansi Handset.

Dengan dirilisnya Android, Google menyediakan berbagai *development kit* dan tutorial untuk membantu calon pengembang ke sistem baru. Berkas-berkas bantuan yaitu *software development kit* (SDK), dan bahkan komunitas

pengembang dapat ditemukan di website Google Android, <http://code.google.com/android>. (DiMarzio, 2008:5)

Android adalah sebuah sistem operasi berbasiskan *java*, yang berjalan pada linux 2.6 kernel (DiMarzio, 2008:7). Menurut *Blackduck* (2010:2), sistem operasi Android untuk peralatan *mobile* dibuat agar dapat digunakan pada *open source software* dibawah lisensi Apache 2008. Android memuat komponen dalam (*platform*) dan komponen luar (Linux kernel dan WebKit, dibawah lisensi GPLv2 dan LGPL serta variasi komponen lainnya atau projek, hak cipta oleh para pemilik lainnya). Pada gambar 2.1 menunjukkan arsitektur Android.



Gambar 2.1 Arsitektur Android

Android tidak hanya sebuah sistem operasi tetapi juga termasuk sebuah *middleware* dan sebuah *array of application* untuk pengguna. Beberapa dari fitur pendukung adalah Dalvik Virtual Machine, membangun pada *browser* dan mendukung *database*, media, kamera, GPS, map, dan fitur lainnya (Mohindra, 2008:3). Sistem ini sangat ringan dan memiliki banyak fitur,

antara lain *speech input*, *Text-to-Speech* (TTS), *accelerated 3D graphics engine* (berdasarkan pada dukungan *hardware*), mendukung *database* yang didukung oleh SQLite, dan sebuah *web browser* yang terintegrasi. (DiMarzio, 2008:7)

2.2.1. Versi Android

Menurut Developers, Android(2010), ketika seseorang mengembangkan aplikasi di Android, diharapkan memahami *platform* pada umumnya untuk manajemen perubahan API (*application programming interface*). Tingkat API diharapkan agar dikenali terlebih dahulu untuk memastikan kompatibilitas aplikasi dengan perangkat yang mungkin akan diinstal. Tingkat API adalah nilai *integer* yang secara unik mengidentifikasi kerangka revisi API yang ditawarkan oleh versi dari *platform* Android. Pembaruan kerangka API dirancang sehingga API baru tetap kompatibel dengan versi API sebelumnya, kecuali dalam kasus-kasus terisolasi dimana aplikasi menggunakan bagian dari API yang kemudian dihapus karena alasan tertentu. Setiap versi *platform* Android mendukung satu tingkat API.

Pada tabel 2.3 memperlihatkan tingkat API yang didukung oleh masing-masing versi dari *platform* Android.

Tabel 2.3 Tingkat API dan Versi Android

Platform Versi	Sebutan	Tingkat API
Android 2.3	Gingerbread	9
Android 2.2	Froyo	8
Android 2.1	Éclair	7
Android 2.0.1	Éclair	6
Android 2.0	Éclair	5
Android 1.6	Donut	4
Android 1.5	Cupcake	3
Android 1.1	-	2
Android 1.0	-	1

Pada tabel 2.4 memperlihatkan perbandingan kompatibilitas dari beberapa versi Android.

Tabel 2.4 Kompatibilitas dari beberapa versi Android

Package(P)-Interface(I)-Class(C)		versi 1.6	versi 2.1	versi 2.2
android.speech.RecognitionListener	I	x	x	v
android.speech.RecognitionService	C	x	x	v
android.speech.RecognizerIntent	C	v	v	v
android.speech.SpeechRecognizer	C	x	x	v
android.speech.tts	P	v	v	v
android.database.sqlite	P	v	v	v
android.database.sqlite.SQLiteTransactionListener	I	x	x	v

Adapun keterangan dari kompatibilitas yang ditunjukkan pada tabel 2.4

yaitu:

1. `android.speech.RecognitionListener` digunakan untuk menerima pemberitahuan dari `SpeechRecognizer` ketika `recognition` yang terkait terjadi.
2. `android.speech.RecognitionService` adalah *class* yang menyediakan *class* dasar untuk implementasi *recognition service*.
3. `android.speech.RecognizerIntent` adalah konstanta untuk mendukung *speech recognition* melalui pengawalan sebuah *Intent*.
4. `android.speech.SpeechRecognizer` adalah *class* yang menyediakan akses ke *speech recognition service*.
5. `android.speech.tts` adalah *package* yang mendukung *Text-to-Speech*
6. `android.database.sqlite` adalah *package* yang mendukung *database SQLite*
7. `android.database.sqlite.SQLiteTransactionListener` adalah sebuah *listener* untuk *transaction events*.

2.2.2. Android SDK

Android sendiri bukanlah bahasa pemrograman, melainkan suatu lingkungan yang digunakan untuk menjalankan aplikasi (DiMarzio, 2008:10).

Platform Android adalah *software stack* untuk perangkat *mobile*, termasuk sistem operasi, *middleware* dan *key applications*. Pengembang dapat membuat aplikasi untuk *platform* menggunakan Android SDK. Aplikasi yang ditulis menggunakan bahasa pemrograman *java* dan dijalankan pada dalvik, sebuah rancangan mesin *virtual* khusus untuk menangani fungsi dasar seperti *threading* dan manajemen *low-level memory*, yang berjalan di atas kernel Linux.

(DiMarzio, 2008:38)

Pada awal pembuatan proyek Android, diharuskan mengunduh dan menginstal Android SDK. Android SDK tidak berbeda dari SDK lain, dalam hal itu berisi semua kode *libraries java* yang diperlukan untuk membuat aplikasi yang berjalan secara khusus pada *platform* Android. SDK juga termasuk *file help*, dokumentasi, sebuah *Emulator* Android, dan sejumlah pengembang lainnya dan *debugging*. (DiMarzio, 2008:22)

Menurut DiMarzio (2008:37), sebagian besar dari Android SDK terdiri dari dokumentasi, pemograman APIs, *tools*, dan contoh-contoh yang terbagi menjadi tiga *folder* utama:

1. *Docs* : berisi seluruh dokumentasi yang menyertai Android
2. *Samples* : berisi enam contoh aplikasi yang dapat di-*compile* dan lainnya dari dalam Eclipse.
3. *Tools* : berisi semua *development*, *compiler*, dan alat *debugging* yang dibutuhkan selama proses pengembangan aplikasi Android.

2.2.3. Speech Recognition

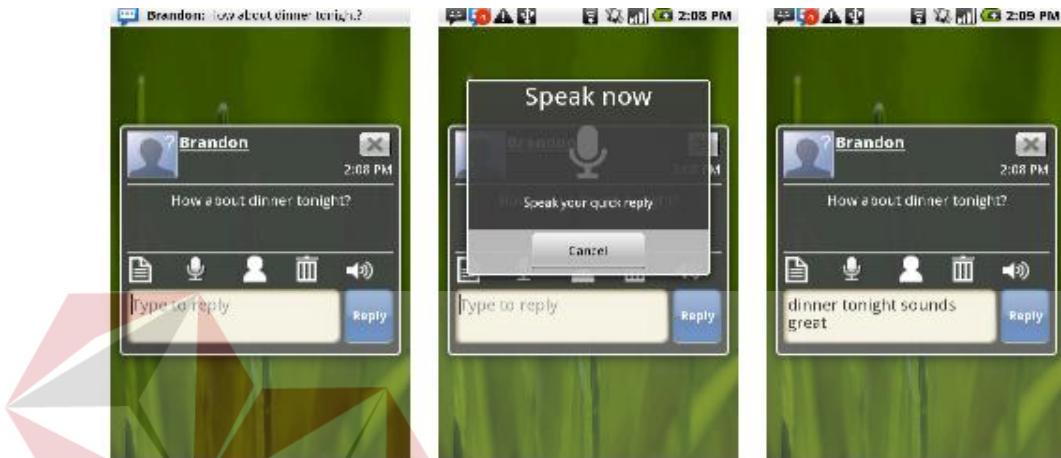
Speech recognition adalah proses identifikasi suara berdasarkan kata yang diucapkan. Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan *template database* yang tersedia. Sedangkan sistem pengenalan suara berdasarkan orang yang berbicara dinamakan *speaker recognition* (Melissa, 2008). *Speech recognition* yang dikenal sebagai *automatic speech recognition* atau *computer speech recognition* menerjemahkan perkataan yang diucapkan menjadi teks. Di dalam sistem *speech recognition* mengandung kata-kata tersebut membutuhkan perbandingan antara sinyal masuk dari kata dan bermacam-macam kata yang ada di dalam kamus (Sunny, 2009). Pada *platform* Android terdapat fitur *Speech Input* dan *Text-to-Speech* sebagai pendukung *voice recognition* pada aplikasi yang akan dibuat.

A. Speech Input

Menurut Developers, Android(2010), aplikasi Google *search voice* yang sudah terpasang pada banyak perangkat Android, termasuk fitur canggih seperti “*search by voice*” dan *shortcut* suara seperti “*navigate to*”. Android 2.1 memperkenalkan *voice-enabled keyboard*, yang membuatnya lebih mudah untuk tetap terhubung.

Pada gambar 2.2, menunjukkan salah satu contoh aplikasi yang mengintegrasikan *speech input* yaitu Handcent SMS. Android SDK memudahkan untuk mengintegrasikan *speech input* langsung ke dalam aplikasi, cukup *copy* dan *paste* dari contoh aplikasi untuk memulainya. Pada penggunaan *speech input* pertama kali aplikasi harus memverifikasi bahwa perangkat target mampu mengenali *speech input*:

```
// Check to see if a recognition activity is present
PackageManager pm = getPackageManager();
List activities = pm.queryIntentActivities(
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
if (activities.size() != 0) {
    speakButton.setOnClickListener(this);
} else {
    speakButton.setEnabled(false);
    speakButton.setText("Recognizer not present");
}
```



Gambar 2.2 Handscent SMS

Aplikasi kemudian menggunakan `startActivityForResult()` untuk menyampaikan maksud bahwa sedang meminta respon *voice recognition*, termasuk sebuah parameter tambahan yang menentukan salah satu dari dua model bahasa. Aplikasi *voice recognition* yang menangani hasil dari proses *voice input*, kemudian melewati pengenalan *string* kembali ke aplikasi dengan memanggil `onActivityResult()` serta panggilan ulang dirinya sendiri.

Android adalah *platform* terbuka, sehingga aplikasi dapat berpotensi menggunakan layanan dari *speech recognition* pada perangkat yang terdaftar untuk menerima `RecognizerIntent`. Aplikasi Google's *Voice Search*, yang merupakan pra-instal pada banyak perangkat Android, menanggapi `RecognizerIntent` dengan menampilkan dialog “*Speak now*” dan *streaming audio* menuju *server google -- server* yang sama digunakan ketika seorang

pengguna menekan tombol *microphone* pada pencarian *widget* atau *voice-enable keyboard*. *Voice Search* diinstal pada semua perangkat utama Amerika Serikat, dan tersedia pada Android Market.

Salah satu tip penting: untuk *speech input* agar seakurat mungkin, sangat membantu untuk memiliki gagasan tentang kata-kata apa yang mungkin untuk diucapkan, yaitu dengan memastikan *user* meminta model bahasa yang sesuai (*free form* untuk dikte atau *web search* untuk kata yang lebih pendek, seperti pencarian frase). *Server Google* mengembangkan model “*free form*” untuk meningkatkan akurasi dikte-an untuk *voice keyboard*, sedangkan model “*web search*” digunakan ketika pengguna ingin mencari dengan suara.

Server Google saat ini baru mendukung 3 bahasa yaitu Inggris, Jepang dan Cina, sedangkan yang paling utama telah dioptimalkan adalah bahasa Inggris. (Developers, Android., 2010)

B. Text-to-Speech

Menurut Developers, Android(2010), dimulai dengan Android 1.6 (API level 4), *platform* Android berisi kemampuan baru yaitu *Text-to-Speech* (TTS). Juga dikenal dengan “*speech synthesis*”, TTS memungkinkan ponsel Android untuk “*berbicara*” teks bahasa yang berbeda. Menurut Hashimi, Komatineni dan MacLean (2010:17), dimulai dari Android versi 2.0 (eclair), android telah memuat mesin Pico *Text To Speech*.

Menurut Developers, Android(2010), Adapun beberapa aspek yang akan penting untuk diaktifkan, sebelum menggunakan aplikasi TTS.

B.1. Bahasa dan sumber daya

TTS pada *platform* Android mendukung sejumlah bahasa yaitu: Inggris, Perancis, Jerman, Itali dan Spanyol. Selain itu, tergantung pada sisi mana *user* berada, Amerika dan aksen Inggris untuk kedua bahasa didukung.

TTS perlu mengetahui bahasa mana yang digunakan untuk berbicara, jadi suara dan kamus khusus sumber daya bahasa yang diperlukan, diambil sebelum aplikasi TTS dapat memulai untuk berbicara.

Beberapa perangkat memiliki penyimpangan yang terbatas dan mungkin kurang spesifik terhadap sumber daya *file* bahasa, walaupun semua perangkat berteknologi Android mendukung fungsi TTS. Jika seorang pengguna ingin menginstal sumber daya tersebut, TTS API memungkinkan aplikasi untuk melakukan *query platform* untuk ketersediaan *file* bahasa dan dapat melakukan *download* serta instalasi. Langkah pertama yang baik adalah untuk memeriksa keberadaan sumber daya TTS dengan maksud menyesuaikan:

```
Intent checkIntent = new Intent();
checkIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA);
startActivityForResult(checkIntent, MY_DATA_CHECK_CODE);
```

Sebuah cek yang berhasil akan ditandai dengan kode hasil dari *CHECK_VOICE_DATA_PASS*, yang menunjukkan perangkat ini siap untuk berbicara, setelah membuat objek *TextToSpeech*. Apabila menginginkan pengguna mengetahui bagaimana menginstal data yang dibutuhkan untuk perangkat, agar menjadi mesin berbicara dalam banyak bahasa, maka diharuskan mengunduh dan menginstal data. Pengunduhan dilakukan dengan mengarahkan kepada *ACTION_INSTALL_TTS_DATA*, yang akan membawa *user* ke Android Market, dan akan membiarkannya mengunduh. Instalasi data akan terjadi secara otomatis

setelah pengunduhan selesai. Berikut ini merupakan contoh tentang implementasi dari `onActivityResult()` akan terlihat seperti:

```
private TextToSpeech mTts;
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (requestCode == MY_DATA_CHECK_CODE) {
        if (resultCode ==
TextToSpeech.Engine.CHECK_VOICE_DATA_PASS) {
            // success, create the TTS instance
            mTts = new TextToSpeech(this, this);
        } else {
            // missing data, install it
            Intent installIntent = new Intent();
            installIntent.setAction(
                TextToSpeech.Engine.ACTION_INSTALL_TTS_DATA);
            startActivity(installIntent);
        }
    }
}
```

Dalam `constructor` dari contoh `TextToSpeech` akan melewati referensi ke `Context` yang akan digunakan (pada `Activity` ini), dan ke `OnInitListener` (pada `Activity` ini juga). `Listener` ini memungkinkan aplikasi untuk diberitahu ketika *Text-To-Speech* terisi penuh, maka dapat memulai mengkonfigurasi dan menggunakananya.

B.2. Bahasa dan Lokal

Google I/O 2009, menunjukkan contoh TTS di mana ia digunakan untuk berbicara hasil terjemahan dari dan ke salah satu dari 5 bahasa di TTS Android yang saat ini didukung. Agar dapat memuat dan mengatur bahasa ke bahasa Inggris, dapat dilakukan dengan:

```
mTts.setLanguage(Locale.US);
```

`locale` adalah cara untuk menentukan bahasa, dimana bahasa yang sama dapat bervariasi dari satu negara ke negara lain. Untuk mengetahui apakah `locale` tertentu didukung, maka dapat menggunakan `isLanguageAvailable()`, yang mengembalikan tingkat dukungan bagi `locale` yang diberikan serta

`TextToSpeech.LANG_MISSING_DATA` jika sumber daya yang dibutuhkan tidak diinstal untuk *query language*.

B.3. Membuat aplikasi berbicara

Pada saat `TextToSpeech` dengan benar diinisialisasi dan dikonfigurasi, dapat dimulai untuk membuat aplikasi berbicara dengan cara termudah yaitu menggunakan *method speak()*.

TTS mengelola antrian global semua entri untuk disintesis, yang juga dikenal sebagai "ucapan". Pada saat pertamakali `speak()` melakukan permintaan maka akan diidentifikasi ucapan mana yang akan mengganggu apa pun yang saat ini sedang disintesis (antrian mengalir dan ucapan baru antri, yang menempatkannya pada awal antrian). Ucapan kedua adalah antrian dan akan diputar setelah teks pertama telah selesai.

B.4. File rendering dan pemutar ulang

Pada saat *method speak()* digunakan untuk membuat Android melaflakan teks, terdapat kasus dimana akan menginginkan hasil dari sintesis disimpan dalam *file audio* sebagai gantinya. Pada saat ingin menghindari CPU *overhead* sintesis karena teks sering diucapkan maka dapat melakukan *render* sekali saja pada *file*, dan kemudian memutar ulang *file audio* kapan saja diperlukan. Pada `speak()`, dapat menggunakan opsional pengenal ucapan untuk diberitahukan pada penyelesaian dari sintesis ke *file*:

```
HashMap<String, String> myHashRender = new HashMap();
String wakeUpText = "Are you up yet?";
String destFileName = "/sdcard/myAppCache/wakeUp.wav";
myHashRender.put(TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID,
wakeUpText);
mTts.synthesizeToFile(wakeUpText, myHashRender, destFileName);
```

Setelah pemberitahuan tentang penyelesaian sintesis, pengguna dapat memutar *file output* sama seperti sumber daya *audio* lainnya dengan `android.media.MediaPlayer`.

Class TextToSpeech menawarkan cara lain untuk menggabungkan sumber daya *audio* dengan *speech*, karena telah memiliki sebuah *file WAV* yang berisi hasil sintesis dalam bahasa yang dipilih sebelumnya dan dapat diakses melalui *path*, atau melalui paket serta ID sumber dayanya, dengan menggunakan salah satu dari dua *method addSpeech()*:

```
mTts.addSpeech(exampleText, destFileName);
```

apabila *file* yang telah tersimpan hilang, maka *speak* akan bersikap seolah-olah *file audio* tidak ada, dan akan mensintesis dan memainkan string yang diberikan.

B.5. Bila tidak digunakan

Apabila telah selesai menggunakan TTS (tidak akan membutuhkan pelayanan TTS lagi) yaitu dengan memanggil `mTts.shutdown()`, dalam *activity method onDestroy()*.

2.2.4. SQLite

Menurut Developers, Android(2010), Android menyediakan dukungan penuh untuk SQLite *database*. Setiap *database* akan dapat diakses dengan nama untuk tiap *class* dalam aplikasi, tapi bukan di luar aplikasi. Android SDK berisi sqlite3 *database tools* yang memungkinkan dapat menelusuri isi tabel, menjalankan perintah SQL, dan melakukan fungsi-fungsi berguna lainnya pada *database* SQLite.

Menurut Developers, Android(2010), adapun spesifikasi dari SQLite pada *platform* Android yaitu:

1. Tanpa *server SQL database engine*
2. *Source code* adalah domain *public*
3. Ringan (kurang dari 300kb)
 - a. Berguna untuk *embedded application*
 - b. Berguna untuk perangkat *mobile*
4. Dukungan *cross platform*
5. *Read* dan *writes* sebuah *file disk* biasa

Tidak mendukung GRANT dan REVOKE karena hanya dapat mendukung hak akses *privileges file* normal.

6. Mengimplemen sebagian besar fitur SQL
SQLite mem-parsing *foreign key constraints* tetapi tidak menerapkannya.

2.3. Interaksi Manusia Komputer (IMK)

Menurut Rizky (2006:4), IMK berdasarkan deskripsi dari asosiasi pakar bidang IMK di Carnegie Melon *Univesity* (ACM) yaitu (Ghaoui, 2006) :

ilmu yang berhubungan dengan desain, evaluasi dan implementasi dari sistem komputasi interaksi untuk manusia dalam konteks sosial, dan juga mempelajari fenomena yang terjadi disekitarnya.

Menurut Sudartamawan dan Ariyus (2007:2), Model interaksi antara manusia dengan sistem melibatkan tiga komponen, yaitu pengguna, interaksi dan sistem itu sendiri.

2.3.1. Faktor Manusia

Menurut Sudartamawan dan Ariyus (2007:17), manusia mempunyai keterbatasan dalam memproses informasi melalui saluran yang terbatas, seperti *visual, auditory, haptic, movement*, dan informasi disimpan dalam memori.

A. Saluran Masukan-Keluaran

Saluran masukan (*input*) utama pada manusia terletak pada pancha indra. Dari saluran ini manusia bisa menerima informasi dari komputer. Semua informasi yang didapat akan diproses dan diolah di dalam memori hingga menghasilkan suatu keluaran (*output*).

Saluran *input* pada manusia antara lain:

1. Mata, berfungsi untuk melihat benda, ukuran, warna, bentuk, kepadatan dan tekstur.
2. Telinga, berfungsi sebagai input pendengaran, untuk mendengarkan nada, warna nada, pola titik nada, intensitas serta *frekuensi*.
3. Hidung, berfungsi untuk membedakan bau yang ada di sekeliling.
4. Lidah, sebagai indra perasa, untuk membedakan rasa manis, kecut, pahit, dan asin.
5. Kulit, yang membungkus tubuh manusia, berfungsi untuk merasakan tekanan dan suhu.

Saluran *output* pada manusia antara lain:

1. Jari-jari, tangan
2. Mata
3. Suara

B. Penglihatan

Penglihatan (*vision*) didapat dari mata. Dengan memiliki penglihatan yang baik maka manusia dapat membedakan informasi yang dilihat untuk diproses lebih lanjut. Beberapa ahli berpendapat bahwa mata manusia digunakan untuk menghasilkan persepsi yang terorganisasi terhadap gerakan, ukuran, bentuk, jarak, posisi relatif, tekstur dan warna.

Interpretasi Sinyal

Manusia mempunyai jarak pandang dan ukuran objek pandang yang terbatas. Ukuran dan kedalaman pandang manusia mengindikasikan seberapa banyak area dari objek pandang yang tertangkap yang berhubungan dengan ukuran dan jaraknya dari mata.

Ketajaman pandangan adalah kemampuan untuk mempersepsikan detail yang sangat baik, untuk memberikan gambaran dari objek yang dipandang jelas dan detail sehingga otak bisa memproses sinyal yang masuk hingga menghasilkan *output*. Selain itu, dipengaruhi oleh kecemerlangan cahaya (*brightness*) dan juga dipengaruhi oleh warna, maka dengan semakin besarnya kejelasan (*luminance*) dan kekeruhan pandangan akan mempengaruhi banyaknya kedipan yang terjadi.

C. Membaca

Membaca merupakan suatu topik yang kompleks. Dengan membaca akan timbul suatu persepsi untuk mengambil suatu kesimpulan. Pola atau jenis huruf, lingkungan dan latar belakang dari tulisan sangat menentukan proses membaca. Begitu juga halnya dengan membaca dilayar monitor. Efek kontras negatif dan huruf yang lebih terang akan memperbaiki proses membaca pada layar monitor.

D. Mendengar

Mendengar merupakan suatu proses penting yang dimiliki oleh manusia. Alat fisik untuk mendengar adalah telinga, telinga dibagi menjadi tiga bagian, yaitu:

1. Telinga bagian luar: berfungsi untuk melindungi telinga bagian dalam dan juga untuk menguatkan suara.
2. Telinga bagian tengah: mentransmisi gelombang suara dengan gendang telinga (*ear drum*) dalam bentuk getaran kemudian melewati telinga bagian dalam melalui *malleus*, *incus* dan *stapes*(*striprup*).
3. Telinga bagian dalam: merupakan transmitter kimiawi. Bergetarnya *stapes* menyebabkan gelombang tekanan dan *cochlea* yang menyediakan cairan mendeteksi adanya sinyal dan mengirimnya ke otak melalui syaraf pendengaran (*auditory nerve*).

E. Suara

Suara diakibatkan oleh getaran-getaran yang menimbulkan gelombang-gelombang tekanan. Bentuk dari suara tergantung dari panjang gelombang yang diakibatkan oleh getaran. Pada dasarnya suara memiliki tiga komponen, yaitu:

1. *Pitch* atau frekuensi suara.
2. *Loudness* atau *amplitude* (lebar atau kekuatan suara)
3. *Timbre* atau jenis dan kualitas suara. Kualitas suara tergantung dari tiga komponen:
 - a. *Attack*, pembangkit suara
 - b. *Sustain*, penopang suara
 - c. *Decay*, pengurangan suara