BAB II

LANDASAN TEORI

2.1 TOEFL

Test of English as a Foreign Languange yang disingkat TOEFL adalah ujian kemampuan berbahasa Inggris dengan logat Amerika yang diperlukan untuk masuk ke universitas di Amerika atau negara-negara lain di dunia. Ujian ini sangat diperlukan bagi pendaftar atau pembicara yang bahasa utamanya bukan bahasa Inggris.

Jenis tes bahasa Inggris TOEFL ini pada umumnya diperlukan untuk persyaratan masuk kuliah pada hampir semua universitas di Amerika Serikat dan Kanada baik untuk Program *Undergraduate* (S-1) maupun *graduate* (S-2 dan S-3), di Indonesia TOEFL juga diminta sebagai salah satu persyaratan kelulusan. Hasil TOEFL ini juga dipakai sebagai bahan pertimbangan mengenai kemampuan bahasa Inggris dari calon mahasiswa yang mendaftar ke universitas negara lain, termasuk Eropa dan Australia.

Nilai hasil TOEFL berkisar antara: 310 (nilai minimum) sampai 677 (nilai maksimum) untuk versi PBT (*paper-based test*). Sejak tahun 1998, TOEFL ini diadakan secara online dengan menggunakan komputer (*Computer-based Testing*/ CBT) dan sejak tahun 2005 disebut iBT (*Internet-based Test*). Di tempat-tempat yang belum bisa melaksanakan CBT atau iBT (karena belum ada fasilitas komputer dan jaringan internetnya), TOEFL masih tetap diadakan secara manual menggunakan kertas dan pensil (*paper-based test*). hasil TOEFL versi CBT

mempunyai nilai berkisar antara 0 sampai 300, sementara nilai untuk iBT adalah dari 0 sampai 120. (ETS, 2013)

Buku yang digunakan untuk aplikasi ini adalah Cambridge TOEFL dengan 3 set soal yang dirandom keluarnya. Pemilihan Cambridge TOEFL didasarkan pada pertimbangan bahwa Cambridge TOEFL berisikan ratusan pertanyaan untuk berlatih yang mencakup semua macam tipe tes. Cambridge TOEFL juga memberikan bagian yang bisa meningkatkan *grammar*, *vocabulary*, *Pronuncaiation* dan kemampuan belajar.

2.1.1 Jenis-jenis TOEFL

Saat ini, ada tiga jenis tes pilihan yang tersedia. Sedangkan materi yang dibahas pada tes akan sama, mereka berbeda dalam tingkat teknologi yang dibutuhkan untuk melakukan itu.

1. TOEFL PBT

   PBT singkatan *Paper Based Test*. Ini adalah pendekatan yang paling tradisional untuk mengambil tes TOEFL. Dalam kebanyakan kasus, Anda tidak akan mengambil tes dalam bentuk kertas pada CBT dan format iBT menawarkan keamanan yang lebih dan hasil scoring yang lebih cepat. Di beberapa lokasi di mana komputer tidak tersedia, PBT TOEFL akan terus digunakan.

2. TOEFL CBT

   CBT singkatan *Computer Based Test*. Sampai sekarang, ini adalah metode yang disukai, karena tes TOEFL anda bisa dicatat dalam cara yang paling seakurat mungkin, juga dapat menghilangkan dokumen yang tidak perlu. Di

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

banyak tempat di seluruh dunia, CBT TOEFL adalah format yang siswa internasional akan alami.

3. TOEFL iBT

iBT singkatan untuk *Internet-Based Test*. Sejak dimulai pada tahun 2005, iBT TOEFL telah menggantikan CBT sebagai metode yang paling disukai dalam mengambil tes TOEFL. Format iBT adalah satu-satunya bentuk tes TOEFL yang berisi komponen berbicara. Ini adalah alasan utama telah menjadi tes TOEFL disukai di kalangan perguruan tinggi di Amerika Serikat karena menyediakan representasi paling akurat dari kemampuan percakapan bahasa Inggris siswa (USA Education Guide, 2013)

The TOEFL ® test PBT, diberikan dalam format berbasis kertas, memakan waktu sekitar tiga dan satu setengah jam untuk menyelesaikan. Ada juga 30-menit tes menulis, disebut TWE ® test, yang dibutuhkan setiap orang yang mengambil tes berbasis kertas. Total waktu untuk mengambil kedua tes akan menjadi sekitar empat jam.

Seperti yang dapat dilihat pada tabel 2.1, tes TOEFL PBT memiliki tiga bagian ditambah tes menulis TWE, yang semuanya wajib. Setidaknya satu pertanyaan di setiap bagian harus dijawab dan satu esai harus ditulis untuk mendapatkan skor.

Tabel 2.1 TOEFL PBT (ETS, 2013)

Tes	Bagian	Deskripsi	Batas Waktu	Jumlah pertanyaan
TOEFL PBT	*Listening Comprehension*	Mengukur kemampuan untuk memahami pembicaraan bahasa inggris	30-40 menit	50
TOEFL PBT	*Structure and Written Expression*	Mengukur kemampuan untuk mengenali bahasa yabg sesuai standar tulisan inggris	25 menit	40

TOEFL PBT	*Reading Comprehenion*	Mengukur kemampuan untuk memahami bacaan	55 menit	50
TWE	*Test of Written English*™	Mengukur kemampuan untuk menulis bahasa inggris	30 menit	1 topik

Tabel 2.2 TOEFL CBT (ETS, 2013)

Tes	Bagian	Deskripsi	Batas Waktu	Jumlah pertanyaan
TOEFL CBT	*Listening*	Tes pada bagian ini menguji kemampuan berbicara bahasa inggris	40-60 menit	30-50
TOEFL CBT	*Structure*	Bagian ini mengukur kemampuan untuk mengenali grammar dengan benar, vocabulary dan penggunaan tulisan inggris dengan standart amerika utara	15-20 menit	20-25
TOEFL CBT	*Reading*	Bagian ini mengukur kemampuan peserta untuk membaca dan memahami cerita pendek seperti yang di gunakan pada universitas seperti di amerika atau Canada	70-90 menit	44-60
TOEFL CBT	*Writing*	Bagian ini mengukur kemampuan peserta untuk menulis esai menggunakan ide original dan contoh, fakta dan organisasi di inggris.	30 menit	1 essai

TOEFL iBT diberikan dalam bahasa inggris dan dijalankan melalui internet. Ada empat bagian (*listening, reading, speaking, and writing*) yang menggunakan total waktu 4 jam 30 menit. Mengkombinasikan 4 bagian: *Listening, Reading, Speaking dan Writing*. Dalam test, anda akan di minta untuk menjalankan tugas yang mengkombinasikan lebih dari satu bagian, seperti:

1. Baca, dengan dan memberikan jawaban dari pertanyaan
2. Dengarkan dan berikan jawaban dari pertanyaan
3. Baca, dengar dan tuliskan jawaban dari pertanyaan

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Tabel 2.3 TOEFL IBT (ETS, 2013)

Bagian	Batas waktu	Jumlah pertanyaan	Tugas
Reading	60-80 menit	36-56 pertanyaan	Baca 3-4 bagian dari teks dan jawab pertanyaan
Listening	60-90 menit	34-51 pertanyaan	Dengarkan percakapan dan jawab pertanyaan
Break	10 menit	-	-
Speaking	20 menit	6 tugas	Ekspresikan opini mengenai topic berbasis bacaan dan percakapan
Writing	50 menit	2 tugas	Tulis esai dari bacaan dan percakan yang memberikan support ke tulisan

2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware,* dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone*. (Safaat, 2011:1)

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android juga menjadi sistem operasi pada *Tablet PC*. Pesatnya pertumbuhan Android disebabkan oleh platformnya yang sangat lengkap baik dari sistem operasi, aplikasi, dan *tool* pengembangan, *market* aplikasi Android serta dukungan yang sangat tinggi dari komunitas *open source* di dunia.

2.2.1 Keunggulan Android

1. Lengkap (*Complete Platform*): Para desainer dapat melakukan pendekatan yang komprehensif 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.

2. 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.
3. *Free* (*Free Platform*): Android adalah *platform*/aplikasi yang bebas untuk *developer*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun. (Safaat, 2011:3)

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

2.2.2 Arsitektur Android

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

1. *Applications* dan *Widgets*

   Application dan Widgets ini adalah *layer* dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS. Kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Applications Frameworks*

   Applications Frameworks ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/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.

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

   a. *Views*
   b. *Content Provider*
   c. *Resource Manager*
   d. *Notification Manager*
   e. *Activity Manager*

3. *Libraries*

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

   a. *Libraries* media untuk pemutaran media *audio* dan *video.*
   b. *Libraries* untuk manajemen tampilan.
   c. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
   d. *Libraries* SQLite untuk dukungan database.
   e. *Libraries* SSL dan *WebKit* terintegrasi dengan *web browser* dan *security*.
   f. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embedded web view*.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

g. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's.

4. Android *Run Time*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya mengggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android *Run Time* dibagi menjadi dua bagian, yaitu:

a. *Core Libraries*: Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai *virtual* mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa java/c yang ditangani oleh *Core Libraries*.

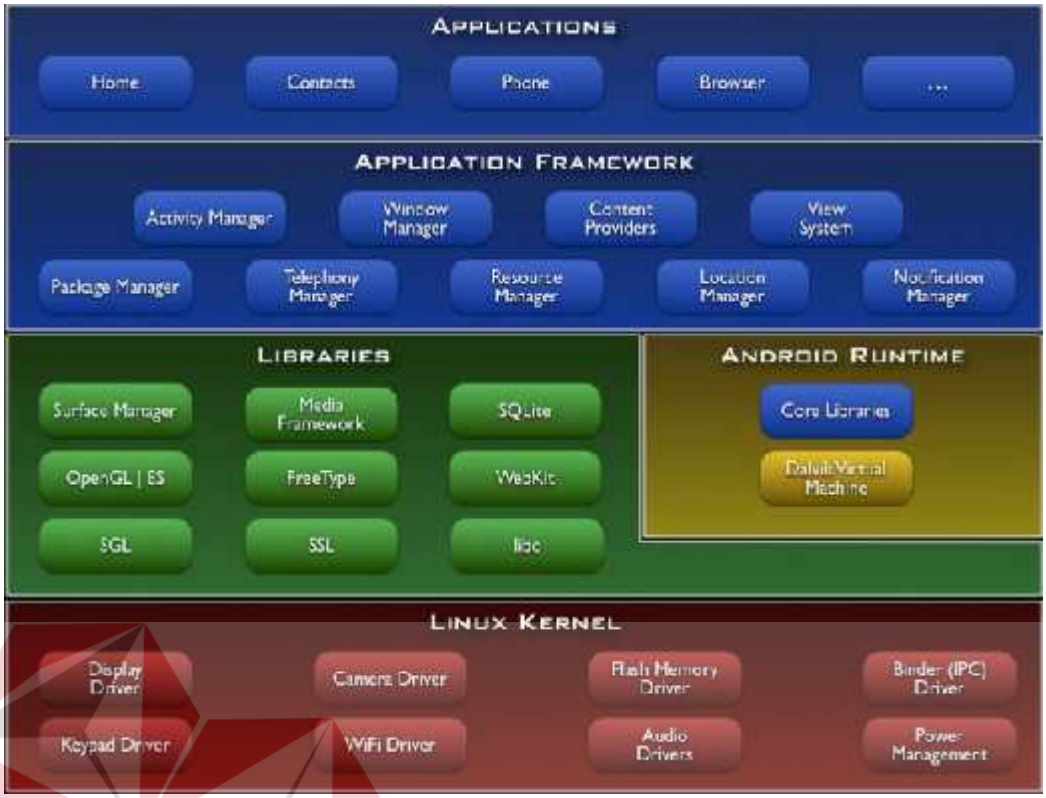
b. *Dalvik Virtual Machine*: *Virtual* mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

5. *Linux Kernel*

Linux kernel adalah layer dimana inti dari *operating system* dari Android itu berada. Berisi file-file sistem yang mengatur sistem *processing*, memori, *resource*, *drivers*, dan sistem-sistem operasi Android lainnya. *Linux kernel* yang digunakan Android adalah *linux kernel release 2.6.* (Safaat 2011:6-8)

Berikut merupakan gambar dari arsitektur Android:

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA



Gambar 2.1 Arsitektur Android (Nazruddin,2011:9)

2.2.3 Fundamental Aplikasi Android

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data file *resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh *tools* yang dinamakan *"apt-tools"* ke dalam paket Android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang kita sebut dengan aplikasi dan nantinya dapat di-*install* di perangkat *mobile*. Ada empat jenis komponen pada alikasi Android, yaitu:

a. *Activites*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bisa jadi hanya

memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.

b. *Service*

Service tidak memilki GUI, tetapi berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya, *media player* sedang memutar lagu dari list yang ada, aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan user untuk memilih lagu atau menulis SMS sambil *player sedang* jalan. Untuk menjaga musik tetap dapat dijalankan, *activity player* dapat menjalankan *service*.

c. *Broadcast Receiver*

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, dll.

Broadcast receiver tidak memiliki UI, tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau getaran perangkat, dan lain sebagainya.

d. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bias digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta, atau aplikasi yang membutuhkan untuk

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

mengakses data kontak dan navigasi, maka disinilah fungsi *content provider.* (Safaat, 2011:9-10)

2.3 Antarmuka atau interface

Antarmuka menurut Santosa diperlukan dalam membangun sebuah aplikasi. Dengan adanya antar muka yang *user-friendly*, diharapkan dapat memudahkan para pengguna sebagai user untuk menjalankan sistem tersebut.

Antarmuka juga dikenal dengan nama GUI (*Grapihcal User Interface*), yaitu program antarmuka yang berbasis grafis, dimana perintah-perintah tidak lagi diketik di *keyboard*, tetapi dengan cara melakukan interaksi secara langsung terhadap apa yang terlihat di layar, yang sebenarnya merupakan suatu abstraksi dari suatu perintah kepada komputer agar komputer mengerjakan apa yang pengguna inginkan.

Terdapat beberapa tipe komunikasi / interaksi antara manusia dengan mesin komputer, yaitu:

1. Dialog berbasis bahasa alami

   Pengguna dapat secara bebas dapat memberikan instruksinya. Dengan kebebasan yang dimiliki pengguna untuk memberikan sembarang instruksi, komputer harus mampu mengolah bahasa alami. Meskipun demikian, karena bahasa alami sering menimbulkan ambiguitas, maka dialog dengan bahasa alami tidak dapat diimplementasikan secara sempurna.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

2. Sistem Menu

   Sistem ini dilakukan dengan memilih pilihan-pilihan yang tersedia pada layar tampilan, atau dengan meng-klik pilihan-pilihan dari menu *pulldown* yang tersedia, maka komputer akan memproses instruksi tersebut.

3. *Form filling dialog*

   Pengguna seolah-olah mengisikan data ke dalam formulis elektronik menggunakan *keyboard*.

4. Dialog berbasis icon

   Tampilan layar menggunakan icon (gambar sederhana yang menunjukkan suatu aktifitas tertentu). Jadi dengan meng-klik gambar tersebut, maka komputer akan mengerjakan perintah dari maksud gambar tersebut.

5. Dialog berbasis jendela

   Tampilan pada layar terdapat jendela, yaitu terdapat bentuk empat persegi panjang dan dibatasi oleh suatu pembatas yang biasanya nampak. Hal ini memungkinkan pengguna untuk melihat banyak jendela yang berisi informasi yang dapat dilihat secara serempak.

6. Manipulasi Langsung

   Pengguna langsung berinteraksi dengan objek yang ada pada layar tampilan dengan mengarahkan *pointer* yang ada di layar, atau menekan tombol-tombol.

7. Interaksi grafis

   Pengguna seolah-olah berdialog dengan grafik yang dibuatnya. Pengguna mempunyai keleluasaan mengubah gambar yang ada pada layar tampilan.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

2.4. Kriteria User Friendly

Ada beberapa kriteria dalam menentukan apakah sistem atau aplikasi yang dibuat dikatakan *user friendly* menurut Nurhayati, yaitu:

1. Memiliki tampilan yang bagus.
   a. Memakai warna yang nyaman dilihat dalam waktu lama
   b. Memiliki kombinasi warna yang sesuai atau tidak mencolok
   c. Tampilannya tersusun dengan rapi
2. Mudah dioperasikan.
   a. Memilik tampilan yang jelas (tombol, teks, dll)
   b. Sistemnya stabil
3. Mudah dipelajari
   a. Alur jalannya program jelas
   b. Ada sistem bantuan.
4. Pengguna selalu merasa senang setiap kali menggunakan software tersebut.

(Nurhayati, 2010:9)

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

2.5. Simulasi

Simulasi ialah suatu metodologi untuk melaksanakan percobaan dengan menggunakan model dari satu sistem nyata (Siagian, 1987). Menurut Hasan (2002), simulasi merupakan suatu model pengambilan keputusan dengan mencontoh atau mempergunakan gambaran sebenarnya dari suatu sistem kehidupan dunia nyata tanpa harus mengalaminya pada keadaan yang sesungguhnya.