

**MEMBANGUN APLIKASI SIMULASI TOEFL MENGGUNAKAN APP  
INVENTOR**

**Naskah Publikasi**



diajukan oleh  
**Fatchurrijal Mufti**  
**08.11.22287**

kepada  
**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**  
**AMIKOM**  
**YOGYAKARTA**  
**2012**

NASKAH PUBLIKASI

MEMBANGUN APLIKASI SIMULASI TOEFL MENGGUNAKAN APP  
INVENTOR

disusun oleh

Fatchurrijal Mufti

08.11.2287

Dosen Pembimbing

Kusrini, Dr., M.Kom  
NIK. 190302106

Tanggal, 01 Februari 2012

Ketua Jurusan  
Teknik Informatika



Sudarmawan, MT.  
NIK. 190302035

**BUILDING TOEFL SIMULATION APPLICATION USING APP INVENTOR**  
**MEMBANGUN APLIKASI SIMULASI TOEFL MENGGUNAKAN APP INVENTOR**

Fatchurrijal Mufti  
Jurusan Teknik Informatika  
STMIK AMIKOM YOGYAKARTA

**ABSTRACT**

*The development of mobile-based applications are very dynamic increase of both types, segmentation and objectives. The development framework is the development of both operating systems, programming and application development provides plenty of options and an alternative presentation of business solutions and mobile-based communities.*

*Android is a mobile operating system that grows in the middle of other operating systems developed today. Other operating systems like Windows Mobile, I-Phone OS, Symbian, and many more that also offers a wealth of content and to optimally hardware runs on top of existing devices. Android offers a different environment for development.*

*TOEFL Simulations for Android is an application which developed on the Android Operating Systems, This application has the features needed for a simulation for TOEFL test, such as set of TOEFL listening test, and also the theory test, not only that this application also comes with the set theory of the english grammar and usage, tenses formula, a to z list of regular and irregular verbs and many other features, that can help you if you want to learn or improve your English knowledge before taking TOEFL Test. This application was made for those of you who want to do TOEFL Test so you have a good preparation when facing TOEFL Test.*

**Keywords:** *Android, Mobile Applications, TOEFL, Simulation*

## **1. Pendahuluan**

Perkembangan teknologi semakin pesat dan cepat, khususnya teknologi informasi dan komunikasi. Hal ini membuat manusia bagaikan tak terpisah oleh jarak ruang dan waktu. Dengan perkembangan teknologi yang kian maju, manusia dapat membuat berbagai macam peralatan sebagai alat bantu dalam menjalankan berbagai aktivitas untuk mendukung produktifitas. Dengan segala aktifitas yang kian padat menjadikan sebagian orang memiliki tingkat mobilitas yang tinggi. Terkadang hal yang ingin dilakukan menjadi terus terbengkalai karena faktor jarak dan waktu. Salah satunya TOEFL yang sulit untuk dilakukan. Salah satu faktor penyebabnya adalah terbatasnya informasi, jarak dan waktu. Misalkan ketika seseorang ingin melaksanakan latihan atau test TOEFL yang berada di suatu tempat atau daerah yang mana tempatnya terletak di daerah yang jauh dari tepat dia tinggal, hal itu menyulitkan jika kita ingin melaksanakan latihan atau test TOEFL. Dewasa ini sudah banyak bermunculan simulasi TOEFL ini yang menggunakan media komputer dan internet yang akan tetapi hal tersebut dinilai tidak terlalu efektif jika kita ingin melakukan simulasi TOEFL karena memerlukan media komputer dan internet jika kita ingin melakukan simulasi tes TOEFL.

Seiring dengan tingkat mobilitas yang tinggi, beberapa tahun terakhir tengah marak perangkat bergerak atau *mobile device*. Salah satu perangkat *mobile* yang paling pesat adalah *Handphone* dimana hampir setiap orang memiliki. *Handphone* yang sedianya sebagai alat komunikasi, saat ini sudah lebih dari fungsi dasarnya. Berbagai macam fitur telah ditanamkan, seperti pengolah gambar dan video, pengolah dokumen dan lain sebagainya. Hal ini tak lepas dari penggunaan Sistem Operasi pada *Handphone*. Layaknya pada komputer, *Handphone* pun dapat di instal berbagai macam aplikasi yang diinginkan.

Android sebagai Sistem Operasi berbasis linux yang dapat digunakan di berbagai perangkat *mobile*. Android memiliki tujuan utama untuk memajukan inovasi piranti telepon bergerak agar pengguna mampu mengeksplorasi kemampuan dan menambah pengalaman lebih dibandingkan dengan platform *mobile* lainnya. Hingga saat ini Android terus berkembang, baik secara sistem maupun aplikasinya.

## **2. Landasan Teori**

### **2.1 TOEFL**

TOEFL (*Test of English as a Foreign Language*) adalah tes untuk menguji kemampuan atau kompetensi seseorang dalam bahasa Inggris.(Priyonggo, 2004, h. 1). Terkadang TOEFL merupakan sebuah persyaratan untuk orang-orang yang non-native bahasa Inggris untuk masuk ke berbagai macam lembaga, seperti instansi pemerintah, badan perizinan, bisnis, atau program beasiswa mungkin membutuhkan tes ini. Nilai

TOEFL berlaku selama 2 tahun dan kemudian tidak akan lagi dilaporkan secara resmi, karena kemampuan bahasa kandidat bisa berubah signifikan sejak dia mengikuti test. Perguruan tinggi dan Universitas biasanya hanya mempertimbangkan nilai terakhir dari tes TOEFL. Tes TOEFL memiliki lisensi yang terdaftar dari *Educational Testing Service* (ETS) dan dikelola diseluruh dunia.

Kebijakan yang mengatur program TOEFL diformulasikan dengan saran dari 16 Anggota Dewan. Anggota dewan yang berafiliasi dengan sekolah-sekolah sarjana dan pasca sarjana. Institusi dan lembaga publik atau swasta dengan minat dalam pendidikan internasional. Anggota lain adalah spesialis dibidang bahasa Inggris sebagai bahasa asing atau bahasa kedua.

Komite pengujii TOEFL terdiri dari 12 spesialis linguistik, pengujian bahasa, pengajaran atau penelitian. Tanggung jawab utamanya adalah untuk memberikan saran tentang isi tes TOEFL. Komite membantu memastikan tes ini adalah ukuran valid yang bisa digunakan untuk mencerminkan kemampuan bahasa Inggris sebagai tren dan metodologi dewasa ini.

## 2.2 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan piranti telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.(Stephanus, 2011, h.1)

## 2.3 UML

*Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsi dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi obyek (OOP).(Fowler, 2004, h. 1).

UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.

Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri piranti lunak dan pengembangan sistem.

## 2.4 Google App Inventor

Google App Inventor adalah sebuah aplikasi web-based yang memungkinkan pengguna untuk membuat sebuah aplikasi perangkat lunak untuk OS Android. Google App Inventor menggunakan *graphical interface*, hampir mirip dengan graphical interface milik Scratch dan StarLogo TNG, yang memungkinkan pengguna untuk drag and drop objek visual untuk membuat aplikasi yang berjalan pada sistem Android yang sudah banyak digunakan diperangkat mobile. Aplikasi ini dibuat berdasarkan banyaknya permintaan untuk membuat developer tools untuk OS Android, dan aplikasi ini dirilis pada tanggal 15 Desember 2010.

Ada dua bagian utama untuk Google App Inventor ini : sebuah browser-based design screen dan sebuah java-based blocks editor (yang dijalankan dengan jendela terpisah dengan mengklik tombol “Open Blocks Editor”). Ada juga sebuah “Ekstra” aplikasi untuk mendownload dan menginstall yang berisi item atau data seperti sebuah emulator ponsel Android.

Didalam Browser-Based Google App Inventor untuk Android berisi daftar berbagai macam komponen seperti : tombol, suara, video, TinyDatabase, komponen jejaring sosial, sensor, text-to-speech dan masih banyak lagi, pada dasarnya Browser-based interface ini menyediakan bahan untuk membangun sebuah Blocks untuk membuat suatu aplikasi.

Untuk mulai membangun fungsi didalam program Android disediakan Java-based blocks editor. Blocks editor ini adalah built-in item yang tersedia untuk setiap aplikasi secara default. Disini, terdapat beberapa fungsi seperti : fungsi matematika, logika, lists, warna. Semua fungsi tersebut dihubungkan satu dengan yang lainnya untuk menciptakan sebuah “action”, seperti membuat aksi ketika tombol play diklik maka akan muncul suara. Blocks editor juga memiliki tombol yang berfungsi untuk menghubungkan ponsel Android yang telah terhubung ke komputer agar bisa digunakan untuk menjalankan program aplikasi tanpa harus menggunakan emulator.

## 2.5 Java

Java merupakan bahasa berorientasi objek (OOP) yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain. Java tidak

mengijinkan pewarisan jamak namun menyelesaikan kebutuhan pewarisan jamak dengan fasilitas antarmuka yang lebih elegan(Prasetyo, 2007, h. 1).

## 2.6 Smartphone

Telepon pintar (*smartphone*) adalah telepon gengam yang mempunyai mempunyai kemampuan tinggi dan dengan fungsi yang hampir menyerupai komputer. Belum ada definisi mutlak dari *smartphone* ini. Beberapa orang mengatakan, *smartphone* merupakan telepon yang bekerja menggunakan seluruh piranti lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, telepon pintar hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti *e-mail*, internet dan kemampuan membaca buku elektronik / *e-book* atau terdapat *keyboard* (baik *built-in* maupun eksternal) dan konektor VGA. Dengan kata lain, *smartphone* merupakan komputer mini yang mempunyai kapabilitas sebuah telepon.

## 3. Analisis

### 3.1 Analisis Sistem

Analisis dilakukan untuk mengetahui kebutuhan yang diperlukan oleh sistem untuk mengembangkan aplikasi. Analisis dilakukan dengan mencari dan menentukan beberapa kebutuhan seperti data masukan, fungsi-fungsi yang dibutuhkan, data hasil proses sistem dan desain antar muka sistem.

Konsep dasar ATOES ini aplikasi simulasi soal-soal mendekati tes TOEFL yang berjalan di *handset* yang memiliki sistem operasi Android

ATOES merupakan perangkat lunak simulasi TOEFL berbasiskan teknologi android. *User* bisa menggunakan aplikasi ini jika mempunyai perangkat yang memiliki *handset* bersistem operasikan android. Didalam aplikasi ini user bisa melakukan simulasi tes TOEFL dengan menggunakan *handset* Android, ketika *user* awal masuk *user* bisa langsung memilih untuk melakukan simulasi tes TOEFL yang terdiri dari tes *listening* dan tes teori, setelah user melakukan tes hasil dari simulasi yang telah dilakukan oleh user akan muncul perhitungan hasil atau skor TOEFL ini didasarkan kepada penilaian *paper based test* TOEFL.

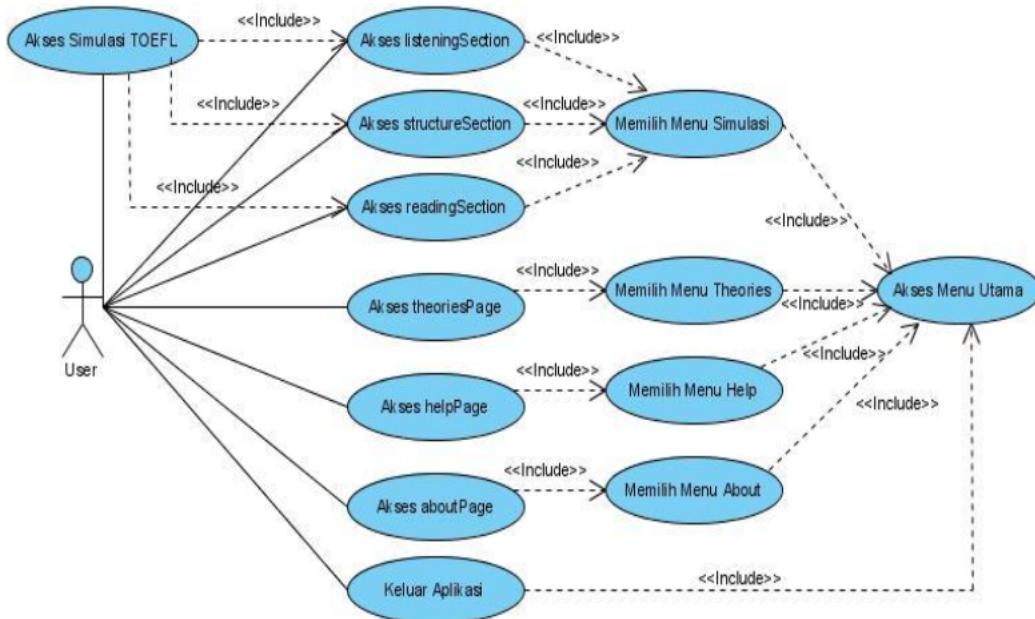
### 3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem sangat dibutuhkan guna menunjang penerapan sistem baru, apakah sistem baru yang akan diterapkan suda sesuai dengan kebutuhan. Sejalan dengan perancangan sistem yang akan dibuat dibutuhkan perangkat teknologi pendukungnya. Perangkat teknologi itu meliputi perangkat keras (*hardware*), perangkat lunak (*software*), serta pengguna (*brainware*). Analisis kebutuhan sistem ini bertujuan untuk mengetahui sistem seperti apa yang sesuai untuk diterapkan, perangkat keras dan

perangkat lunak apa yang sesuai, serta siapa saja pengguna yang akan menggunakan sistem ini.

### 3.3 Perancangan Sistem

Perancangan sistem yang digunakan untuk membangun aplikasi ATOES ini adalah dengan menggunakan metode UML (*Unified Modelling language*). UML adalah bahasa standar yang digunakan untuk menjelaskan dan menvisualisasikan artifak dari proses analisis dan desain berorientasi objek. UML memungkinkan developer melakukan pemodelan secara visual, yaitu penekanan pada penggambaran. Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari objek, mempermudah penggambaran interaksi antara elemen dalam sistem, dan mempertahankan konsistensi antara desain dan implementasi dalam pemrograman.

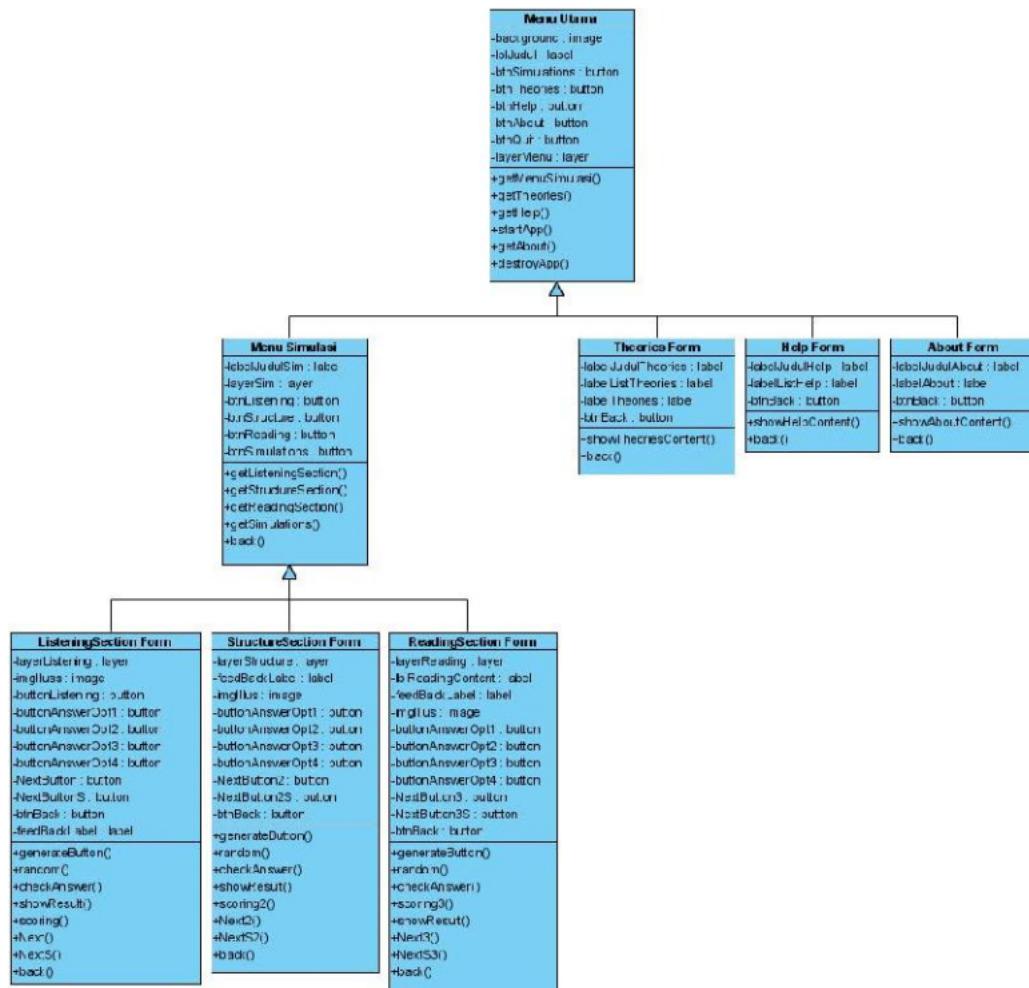


*Use case* diagram ATOES menceritakan tentang *user* yang menggunakan aplikasi ATOES secara optional *user* bisa memilih apa yang diinginkan. *User* disini adalah pengguna aplikasi yang ingin melakukan simulasi tes TOEFL. Ada beberapa aktifitas yang bisa dilakukan oleh pengguna di aplikasi ini yaitu :

1. Jika *user* memilih masuk aplikasi maka sistem akan meload semua data yang dibutuhkan untuk menjalankan aplikasi seperti data gambar, suara, fungsi-fungsi algoritma dan lain lain.
2. Jika *user* memilih menu simulasi maka muncul case baru yaitu masuk kedalam bagian *listeningSection*, *structureSection*, *readingSection*, dimana ketiga case ini merupakan sebuah *include* dari case "memilih menu simulasi"

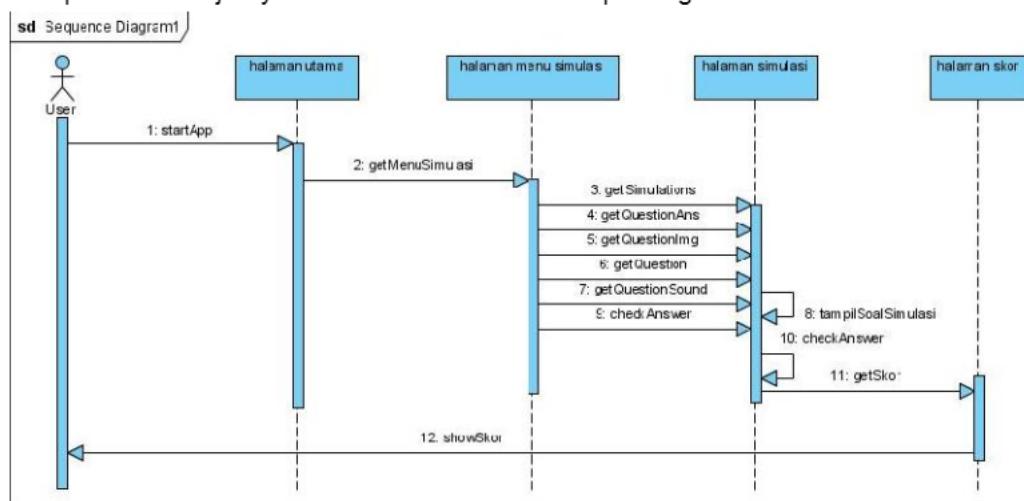
3. Jika *user* memilih menu *theories* maka akan muncul *case* baru yaitu masuk kedalam bagian teori-teori yang dibuat untuk menambah wawasan tentang bahasa Inggris
4. Jika *user* memilih menu *help* maka akan muncul *case* baru tentang cara-cara penggunaan aplikasi ATOES.
5. Jika *user* memilih menu *about* maka akan muncul *case* baru tentang riwayat dari aplikasi ATOES
6. Jika *user* memilih keluar aplikasi maka akan muncul *case* baru yaitu sistem akan *terminate* semua proses yang ada.

*Class diagram* merupakan diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang akan dibuat. *Class diagram* memberikan gambaran statis tentang sistem atau perangkat lunak yang kompleks. Aplikasi ini mempunyai *class diagram* dari sisi *user*. Berikut ini adalah gambar rancangan objek yang berupa *class diagram* tersebut.



*Sequence diagram* menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi

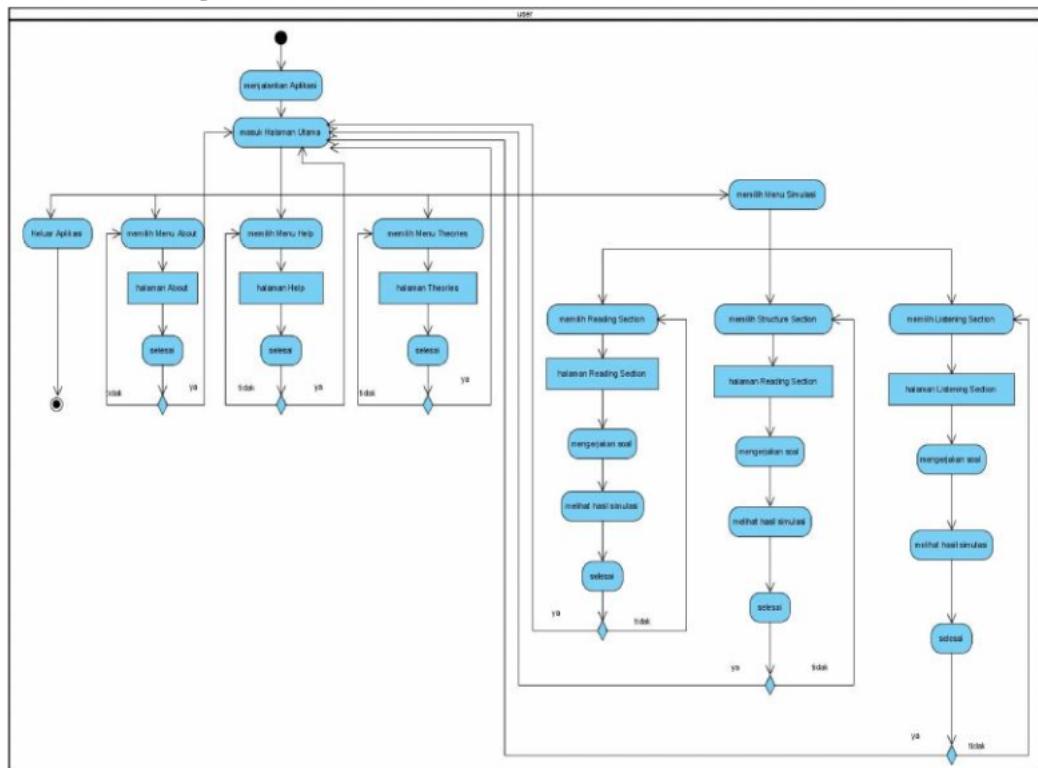
*Sequence diagram* simulasi simulasi tes di ATOES, dimulai ketika menjalankan aplikasi kemudian masuk kedalam halaman menu utama dengan berbagai macam pilihan fasilitas perangkat. Untuk melihat spesifikasi dari sebuah perangkat yang ada maka *user* harus memilih sebuah perangkat kemudian akan ditampilkan sub menu dari masing masing perangkat tersebut. Dari sub menu tersebut maka deskripsi simulasi akan ditampilkan. Selanjutnya adalah melakukan beberapa rangkaian tes simulasi toefl.



*Activity diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja(Fowler, 2004, h. 163). Berikut ini adalah Activity Diagram aplikasi ATOES.

*Activity Diagram* ATOES untuk simulasi TOEFL dimulai ketika *user* memulai dan menjalankan aplikasi, setelah itu sistem menampilkan halaman utama, setelah halaman utama muncul *user* dihadapkan kepada pilihan yaitu memilih menu simulasi, menu theories, menu help, menu about, dan keluar dari sistem. Jika *user* memilih menu simulasi maka sistem menampilkan menu dan halaman simulasi, ketika *user* sudah memilih tes simulasi sistem langsung menampilkan halaman tes simulasi, soal-soal, gambar, jawaban dan komponen-komponen lain yang dibutuhkan untuk tes simulasi. Setelah semua komponen ditampilkan *user* memilih jawaban, setelah *user* memilih jawaban sistem akan menganalisa, menyimpan data jawaban *user*, setelah itu sistem akan mengecek apakah jumlah soal sudah sesuai dengan batas maksimal, jika belum memenuhi sistem akan menampilkan soal lain, jika sudah memenuhi sistem akan menampilkan skor akhir hasil dari pekerjaan *user*, setelah itu muncul keputusan apakah *user* akan melakukan tes simulasi lagi atau *user* sudah selesai dengan tes simulasi

tersebut, jika sudah selesai maka *user* akan kembali ke halaman utama, jika *user* ingin melakukan tes lagi *user* akan kembali ke halaman simulasi.



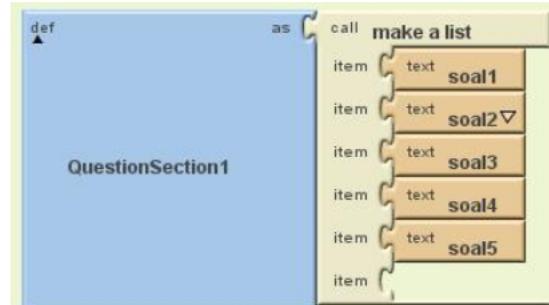
### 3.4 Rancangan Antarmuka

Rancangan yang akan dibuat harus memberikan gambaran dan penjelasan dari setiap gambar, teks dan navigasi. Rancangan tampilan ini menggambarkan keterkaitan setiap halaman dan juga menjelaskan arah komunikasinya. Rancangan tampilan ini bertujuan agar aplikasi yang dihasilkan terlihat lebih menarik, mudah dipahami dan dioperasikan.

### 3.5 Rancangan Struktur Blok

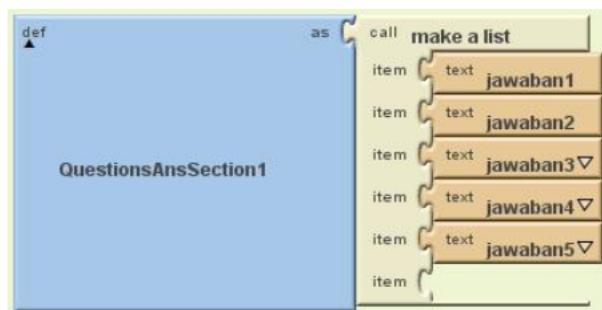
Untuk membangun sebuah aplikasi di App Inventor dibutuhkan koding-koding yang berbentuk blok agar aplikasi bisa berjalan atau berfungsi sesuai dengan rancangan. Berikut ini adalah rancangan struktur blok algoritma ATOES :

*Block Editor App Inventor* menyediakan berbagai macam fungsi yang bisa digunakan sesuai dengan kebutuhan user, misalnya untuk menyimpan data, user cukup menggunakan fungsi variable dengan fungsi list yang bisa digunakan untuk menyimpan data-data, seperti data soal, data jawaban, data gambar, data suara. Berikut ini contoh struktur penyimpanan data soal, jawaban, gambar dan suara :

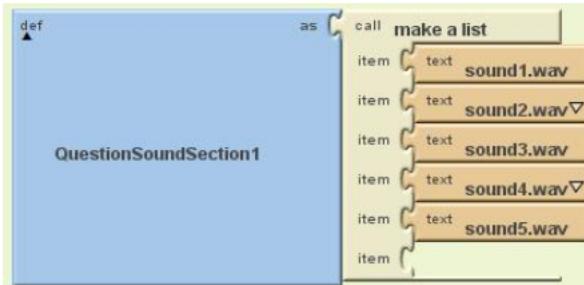


Gambar 3.16 Blok Variabel QuestionSection1

Seperti terlihat pada gambar diatas, struktur penyimpanan data pertanyaan dibuat didalam suatu variabel yang bernama `QuestionSection1` kemudian blok variabel tersebut digabungkan dengan fungsi `make a list` yang berfungsi menyimpan data pertanyaan yang berbentuk list, data pertanyaan yang dimasukan ke dalam list blok `QuestionSection1` ini diambil berdasarkan index list atau urutan didalam blok tersebut.



Seperti terlihat pada gambar diatas, struktur penyimpanan data jawaban dibuat didalam suatu variabel yang bernama `QuestionAnsSection1` kemudian blok variabel tersebut digabungkan dengan blok fungsi `make a list` yang berfungsi menyimpan data jawaban yang berbentuk list, data jawaban yang disimpan kedalam list blok `QuestionAnsSection1` ini diambil berdasarkan index atau urutan didalam blok tersebut.



Seperti terlihat pada gambar 3.18 diatas, struktur penyimpanan data sound / file suara dibuat didalam suatu variabel yang bernama `QuestionSoundSection1` kemudian blok variabel tersebut digabungkan dengan blok fungsi `make a list` yang berfungsi menyimpan data sound / file suara yang berbentuk list, data sound / file suara yang disimpan kedalam list blok `QuestionAnsSection1` ini diambil berdasarkan index atau urutan didalam blok tersebut.



Untuk data sound / file suara nama yang dibuat di dalam list harus sesuai dengan nama file yang sudah diupload kedalam App Inventor Server, jika terdapat perbedaan nama antara nama file dan nama yang tertera didalam list maka sound / file suara tidak bisa dijalankan di aplikasi App Inventor ini.

Seperti terlihat pada gambar diatas, struktur penyimpanan data gambar dibuat didalam suatu variabel yang bernama QuestionImgSection1 kemudian blok variabel tersebut digabungkan dengan blok fungsi make a list yang berfungsi menyimpan data gambar yang berbentuk list, data gambar yang disimpan kedalam list blok QuestionImgSection1 ini diambil berdasarkan index atau urutan didalam blok tersebut. Untuk data gambar suara nama yang dibuat di dalam list harus sesuai dengan nama file yang sudah diupload ke dalam App Inventor Server, jika terdapat perbedaan nama antara nama file dan nama yang tertera didalam blok, gambar tidak bisa ditampilkan di aplikasi App Inventor ini.

#### 4. Hasil Penelitian dan Pembahasan

##### 4.1 Hasil Penelitian

Implementasi merupakan tahapan setelah melakukan analisis dan perancangan sistem pada siklus rekayasa perangkat lunak, dimana aplikasi siap dioperasikan pada keadaan yang sebenarnya, sehingga dari sini akan diketahui apakah program atau aplikasi yang telah dibuat benar-benar dapat menghasilkan output yang sesuai dengan tujuan yang diinginkan.

Implementasi dan pembahasan perancangan aplikasi ATOES ini menggunakan emulator App Inventor dan handphone Samsung Galaxy Mini.

Batasan implementasi pada aplikasi android ini adalah tidak bisa menginput data-data aplikasi, baik data soal, pertanyaan, suara, gambar, dan lain-lain, penginputan data harus dilakukan secara manual.

Berikut ini adalah implementasi aplikasi ATOES

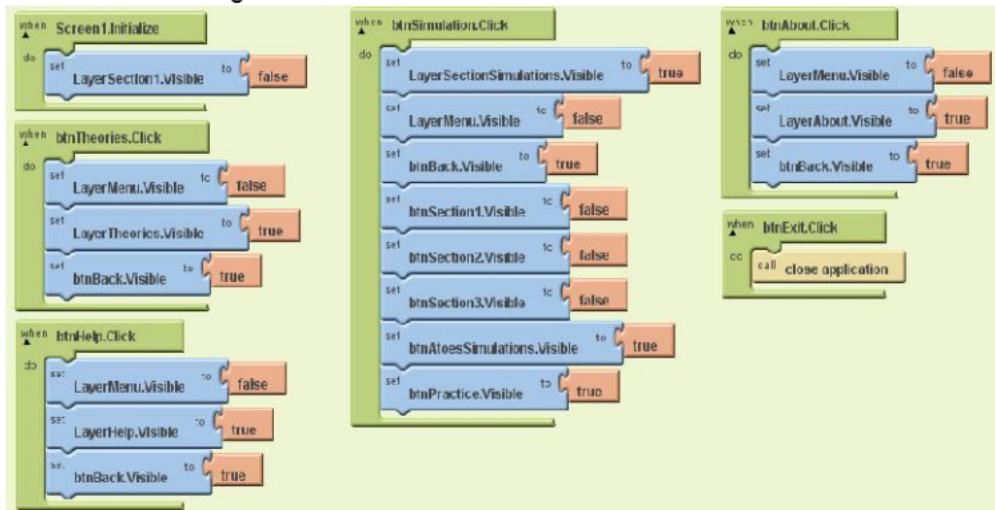
###### 1. Form Utama

Form utama ini merupakan halaman yang pertama kali muncul ketika aplikasi ini pertama dijalankan. Pada halaman ini terdapat 5 menu pilihan yang dapat dipilih oleh user yaitu Go to Simulations, Go to Theories, Help, About, dan Exit.

Form utama ini ini dibuat menggunakan elemen form, button, image, dan label.



Blok coding Halaman Utama

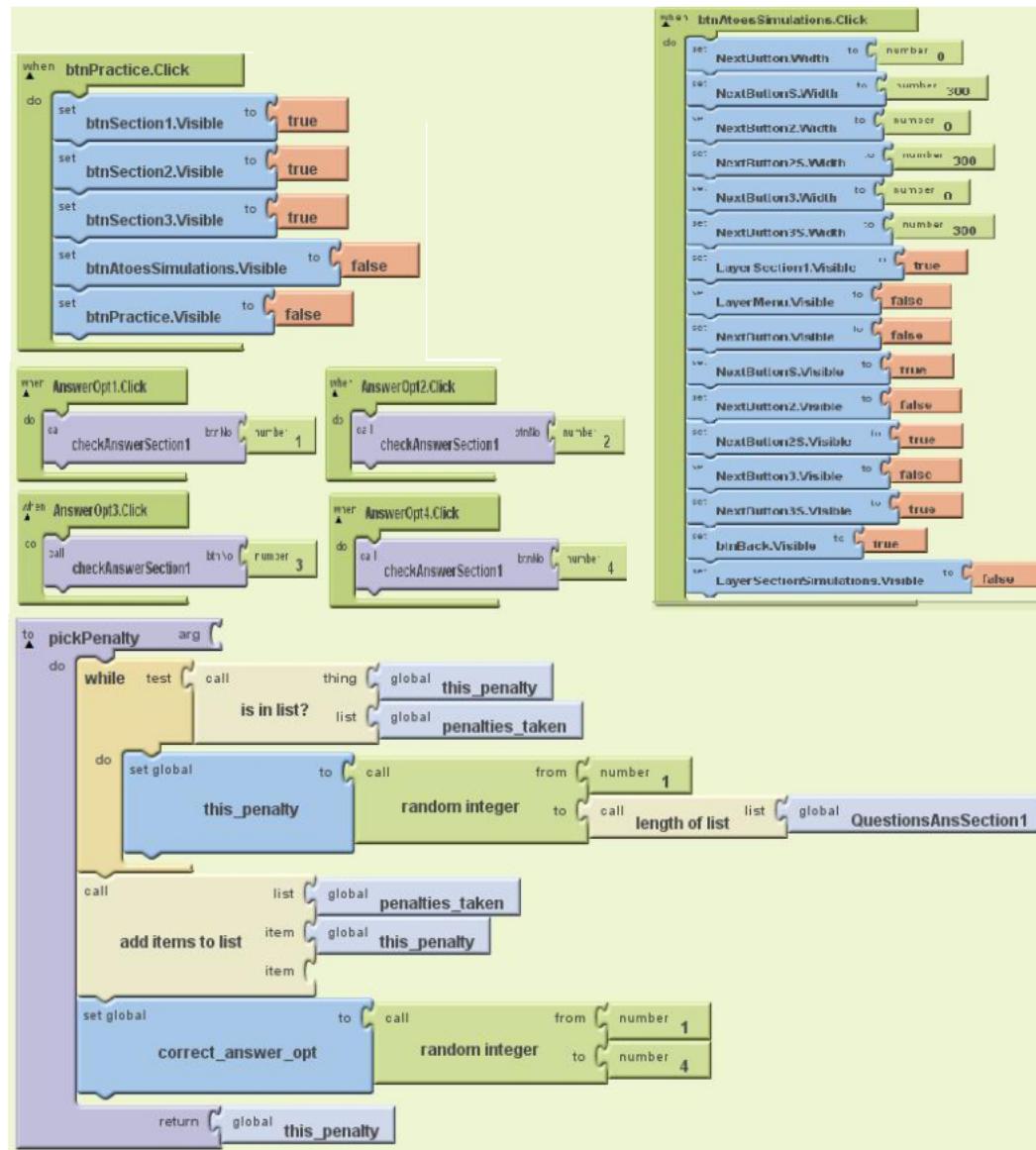


## 2. Form Halaman Menu Simulasi

Form ini akan muncul pada saat kita memilih menu Go to Simulations. Didalam menu ini terdapat 2 pilihan menu Practice dan Simulations. Form Halaman Menu Simulasi ini terdiri dari elemen screen, label, button, dan image



## Blok coding program halaman menu simulasi



## 4.2 Pembahasan dan Pengujian

Pengujian terhadap program dilakukan guna mengetahui lebih dini tentang kesiapan program dalam melakukan input data, proses pengolahan data dan output dari data yang dihasilkan, disamping itu juga dimaksudkan untuk mengetahui lebih lanjut masih adakah kesalahan-kesalahan atau dan kekurangan dari program. Untuk mengetahui keberhasilan dari program yang dibuat maka akan dilakukan pengujian program dengan metode pengujian Black Box Testing dan White Box Testing.

### 1. Black Box Testing

Black Box Testing merupakan tahap pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pada black box testing, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil

dari unit itu sesuai dengan proses bisnis yang diinginkan. Black box testing berusaha menemukan kesalahan dalam beberapa kategori yaitu :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan Interface
3. Kesalahan dalam struktur data / akses data

Langkah selanjutnya adalah melakukan pengecekan terhadap fungsi logika didalam aplikasi Atoes ini, dalam hal ini diambil sample untuk fungsi logika untuk skoring TOEFL asli dengan skoring simulasi TOEFL pada aplikasi ATOES. Untuk mengecek apakah terdapat kesalahan logika dalam aplikasi ini, maka dibuat percobaan perbandingan perhitungan manual kemudian dibandingkan dengan perhitungan yang dilakukan oleh aplikasi Atoes ini.

Contoh kasus perbandingan skoring manual dengan skoring aplikasi:

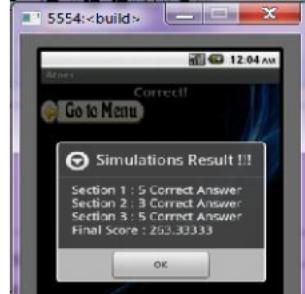
Seorang peserta melakukan tes TOEFL dan dia berhasil menjawab 5 jawaban benar pada bagian *listening comprehension*, 3 jawaban benar pada soal *structure & written expression*, dan 5 jawaban benar dari bagian *reading comprehension*. Maka perhitungan nilai TOEFL peserta tersebut adalah seperti pada tabel dibawah ini.

Jawaban Benar	Konversi Nilai dengan Tabel Konversi
Section I - Listening Comprehension : 5	29
Section II - Structure & Written Expression : 3	24
Section III - Reading Comprehension : 5	26
Skor yang telah dikonversi dijumlahkan	79
( Jumlah Skor Konversi * 10 ) / 3	263,3333333
Skor TOEFL	263,3333333

Setelah semua jawaban di konversi dan dilakukan perhitungan secara manual didapatkan nilai 263,333333 untuk skor TOEFL peserta tersebut.

Kemudian buat kondisi yang sama untuk di aplikasi Atoes ini yaitu 5 jawaban benar pada bagian *listening comprehension*, 3 jawaban benar pada soal *structure & written expression*, dan 5 jawaban benar dari bagian *reading comprehension*.

Setelah melakukan simulasi dengan kondisi yang sama didapatkan hasil seperti berikut ini.



Hasil skoring simulasi di aplikasi Atoes dengan kondisi yang sama menunjukkan skor yang sama antara skor perhitungan manual dengan skor perhitungan di aplikasi Atoes yaitu 263,3333. Dengan hasil perbandingan ini bisa dipastikan aplikasi Atoes ini sudah tidak memiliki *Logic Error* atau kesalahan logika.

## 2. White Box Testing

White-box testing adalah cara pengujian dengan melihat kedalam modul untuk meneliti kode-kode program yang ada dan menganalisis apakah ada kesalahan atau tidak. Jika terdapat modul yang menghasilkan kesalahan output yang tidak sesuai dengan proses bisnis yang dilakukan maka, blok-blok program, variabel dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian dipackage ulang.



## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Setelah melalui beberapa tahapan dalam menyelesaikan Aplikasi Android Simulasi TOEFL menggunakan App Inventor, dapat disimpulkan bahwa :

1. Aplikasi Atoes dapat membantu persiapan sebelum melaksanakan tes TOEFL
2. Tidak memerlukan banyak waktu untuk menjalankan aplikasi Atoes.
3. Simulasi TOEFL menjadi lebih mudah karena menggunakan handphone atau handset yang mudah dibawa kemana-mana sehingga dapat diakses kapanpun dan dimanapun
4. Dengan aplikasi Atoes ini pengenalan tentang TOEFL ini menjadi lebih mudah
5. Aplikasi Atoes ini mudah untuk digunakan
6. Performance aplikasi ini sudah tergolong cukup bagus
7. Aplikasi Atoes ini bisa dijadikan sarana alternatif pengenalan TOEFL
8. Dapat menghemat alokasi sumber daya, karena penggunaannya hanya menggunakan handphone yang menggunakan sumber daya yang kecil

9. Berdasarkan hasil pengujian program aplikasi Atoes ini sudah dapat dipastikan bebas dari *syntax error*, *runtime error*, dan *logic error*.
10. Dari hasil uji sistem dan uji program, aplikasi ini dapat digunakan sebagai alternatif pengenalan TOEFL yang menyajikan simulasi dan cara-cara tes TOEFL.

## 5.2 Saran

Pada penulisan skripsi ini tentu masih terdapat kekurangan yang dapat disempurnakan lagi pada pengembangan sistem berikutnya. Agar aplikasi ini dapat menjadi lebih baik terdapat beberapa saran yang dapat dipergunakan diantaranya :

1. Menambahkan fitur upload data dan hapus data, baik data soal, jawaban, suara, gambar dan lain lain, agar data-data aplikasi ini kualitasnya lebih bagus jika bisa digunakan oleh para ahli bahasa.
2. Membuat jawaban yang berkaitan, sehingga menambahkan tingkat kesulitan aplikasi ini agar dibuat semirip mungkin dengan tes TOEFL yang asli
3. Menambahkan fitur simpan skor hasil simulasi, yang mana fitur ini bisa digunakan untuk menjadi patokan perkembangan user.
4. Menambahkan fitur timer agar aplikasi ini bisa dibuat semirip mungkin dengan tes TOEFL.
5. Menambahkan fitur pilihan bahasa Indonesia, dan bahasa yang lain.
6. Desain aplikasi Atoes ini lebih dipercantik agar user tidak merasa bosan dengan interface aplikasi ini.
7. Aplikasi Atoes ini dikembangkan lagi dengan mengupdate data-data, baik data soal, jawaban, suara, gambar dan lain lain, agar pengguna aplikasi ini tidak merasa jemu.

### **Daftar Pustaka**

- Fowler, M. 2005. UML Distilled. Boston: Pearson Education
- Magnuson B. 2009. Building Blocks for Mobile Games: A Multiplayer Framework for App Inventor for Android. Massachusetts: Massachusetts Institute of Technology
- Mulyadi. 2010. Membuat Aplikasi untuk Android. Yogyakarta: Multimedia Center Publishing
- Munawar. 2005. Pemodelan Visual dengan UML. Yogyakarta: Graha Ilmu
- Prasetyo, D. 2007. 150 RAHASIA Pemrograman Java. Jakarta: PT Elex Media Komputindo.
- Priyonggo, A dan Fanani, A. 2004. Cara Mudah Menguasai TOEFL. Yogyakarta: Diglossia Media
- Riyanto, S. 2007. The 1<sup>st</sup> Student's Choice TOEFL [Test of English as a Foreign Language]. Yogyakarta: Pustaka Pelajar
- Sholiq. 2006. Pemodelan sistem Informasi Berorientasi Objek dengan UML. Yogyakarta: Graha Ilmu
- Stephanus, B.R. 2011. Mudah Membuat Aplikasi Android. Yogyakarta: C.V Andi Offset
- Wolber, D., dkk. 2011. App Inventor Create Your Own Android Apps. Canada: O'Reilly Media Inc.