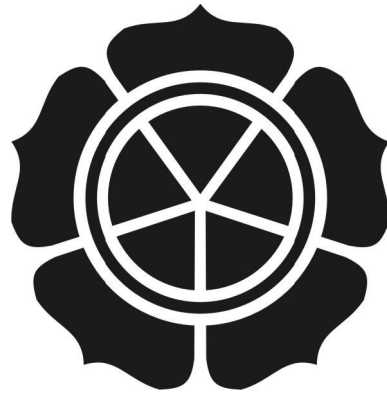


**RANCANG BANGUN GAME EDUKASI BAHASA INGGRIS
MENGUNAKAN APP INVENTOR**

Naskah Publikasi



diajukan oleh

Leo Adi Saputra

08.11.2252

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

NASKAH PUBLIKASI

**RANCANG BANGUN GAME EDUKASI BAHASA INGGRIS
MENGUNAKAN APP INVENTOR**

disusun oleh

Leo Adi Saputra

08.11.2252

Dosen Pembimbing

Kusrini, Dr., M.Kom

NIK. 190302106

Tanggal, 05 Maret 2012

**Ketua Jurusan
Teknik Informatika**



Sudarmawan, MT.

NIK. 190302035

DESIGN OF ENGLISH LANGUAGE EDUCATION GAME USING APP INVENTOR

**RANCANG BANGUN GAME EDUKASI BAHASA INGGRIS MENGGUNAKAN
APP INVENTOR**

Leo Adi Saputra
Jurusan Teknik Informatika
STMIK AMIKOM YOGYAKARTA

ABSTRACT

The development of android mobile application programming to an increasingly widespread application developers are racing to create interesting applications, one of which is gaming. But in Indonesia, as a medium of learning the game do not have a long tradition. In addition to resource problems, the problem is the uneven readiness of the infrastructure issues that make educational games have not been optimally utilized. Game image in the eyes of our society as a medium to entertain more than a medium of learning.

See the development of mobile applications and games are the author of lifting the title on the application Game Android-based English Language Education. This application uses the Google App Inventor in its programming. Google App Inventor is an application provided by Google that allows anyone to create software applications for the Android OS.

The results of this study is an educational game that is able to develop English language skills of children with simple, attractive and fun.

Keywords : *Android, App Inventor, Mobile applications, Education Games*

1. Pendahuluan

Perkembangan teknologi saat ini telah memberikan pengaruh yang sangat besar bagi dunia teknologi informasi. Munculnya beragam aplikasi memberikan pilihan dalam peningkatan kinerja suatu pekerjaan, baik yang bersifat desktop based, web based hingga yang sekarang ini munculnya aplikasi-aplikasi baru yang berjalan dalam telepon seluler/*handphone*.

Telepon seluler 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 telepon seluler. Layaknya pada komputer, telepon seluler pun dapat diinstal berbagai macam aplikasi yang diinginkan.

Android merupakan sistem operasi *mobile* yang tumbuh di tengah sistem operasi lainnya yang berkembang dewasa ini. Sistem operasi lainya seperti Windows Mobile, I-Phone OS, Symbian, dan masih banyak lagi yang juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat *hardware* yang ada. Android menawarkan sebuah lingkungan yang berbeda untuk pengembangan.

Android sebagai Sistem Operasi berbasis linux yang dapat digunakan di berbagai perangkat mobile. Android memiliki tujuan utama untuk memajukan inovasi piranti telepon seluler 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.

Pemilihan telepon seluler untuk salah satu pengembangan aplikasi selain lebih mudah dalam pengoperasiannya, sifat dari telepon seluler yang fleksibel menjadi salah satu alasannya. Sekarang ini telepon seluler seakan menjadi salah satu kebutuhan pokok bagi masyarakat. Hal ini dibuktikan terus meningkatnya pengguna telepon seluler di Indonesia. Jakarta (ANTARA News) - Jumlah pengguna seluler di Indonesia hingga Juni 2010 diperkirakan mencapai 180 juta pelanggan, atau 80 persen dari total penduduk Indonesia. "Jumlah pengguna seluler sebesar 180 juta nomor dicapai setelah 15 tahun layanan GSM beroperasi di tanah air," kata Ketua Asosiasi Telekomunikasi Selular Indonesia (ATSI), Sarwoto Atmosutarno pada *Indonesia Celular Show (ICS) 2010*, di Jakarta Convention Center.

Penggunaan telepon seluler selain sebagai media komunikasi, juga menjadikanya sebagai media hiburan. Beragam aplikasi hingga permainan telah banyak yang berjalan dalam telepon seluler. Peminat aplikasi maupun game yang bersifat *mobile* juga meningkat seiring dengan banyaknya aplikasi atau game yang diproduksi. Mulai dari game yang bertema petualangan, olah raga, hingga balap. Kebanyakan game tersebut hanya memberikan kesan hiburan dan bukan game yang bersifat edukasi.

Di era teknologi informasi sekarang ini perkembangan teknologi dan persaingan yang semakin ketat, menuntut kita agar dapat menguasai Bahasa Inggris yang merupakan Bahasa Internasional. Oleh karena itu, pendidikan Bahasa Inggris perlu diperkenalkan pada anak-anak sejak dini. Namun, pada umumnya anak-anak mengalami kesulitan dalam mempelajari bahasa asing, termasuk Bahasa Inggris. Menurut Diba Artsiyanti E.P., S.S. (2002), kesulitan anak dalam mempelajari Bahasa Inggris disebabkan karena Bahasa Inggris bukan merupakan bahasa mereka, sehingga mereka tidak terbiasa mendengar atau mengucapkan pelafalan dalam Bahasa Inggris. Selain itu kecenderungan pola belajar anak yang lebih suka bermain juga sangat mempengaruhi, sehingga pembelajaran secara teoritis saja kurang optimal untuk pembelajaran Bahasa Inggris pada anak.

Atas beberapa dasar itulah penulis memilih mengangkat permasalahan tentang perancangan aplikasi education game untuk membantu mempermudah pengajaran Bahasa Inggris. Aplikasi ini diharapkan mampu menerapkan sistem belajar sambil bermain yang sangat efektif untuk proses pembelajaran bagi anak-anak.

2. Landasan Teori

2.1. 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)

2.2. Google App Inventor

Google App Inventor adalah sebuah aplikasi web-based yang memungkinkan *user* 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 *user* 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.

App Inventor memiliki dua bagian utama yaitu *App Inventor Designer* dan *Block Editor*. *App Inventor designer* adalah bagian dari App Inventor untuk menambahkan komponen dan mendesain *layout* antarmuka *user* (*user interface*). *App Inventor designer*

ini berjalan di web browser (*web-based*). Pada *App Inventor designer* terdapat komponen palet yang menampilkan koleksi komponen dasar *user interface* seperti *button*, *label* dan *text box*. Untuk menambahkan komponen, *user* cukup dengan *drag-and-drop* komponen ke proyek.

Block editor adalah bagian dari App Inventor untuk merancang logika pemrograman untuk komponen-komponen yang telah dibuat sebelumnya di *App Inventor designer*. Pada *Block Editor user* dapat menggabungkan blok-blok kode yang terkait. Blok-blok tersebut terdapat pada *drawers* (laci) bagian sebelah kiri dari antarmuka *Block Editor*. *User* hanya perlu menyeret (*drag*) blok yang berada pada *drawers* ke dalam *workspace* dalam rangka untuk menambahkannya ke proyek mereka (Magnuson, 2010, h. 15).

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

2.4. Multimedia

Konsep multimedia telah banyak diterapkan dalam dunia pendidikan. Pembelajaran menggunakan multimedia interaktif berkembang atas dasar pembelajaran konvensional yang tidak bisa memenuhi kebutuhan peserta didik dalam pembelajaran (Sigit, dkk, 2008). Multimedia dapat menjadikan suatu aplikasi menjadi sangat interaktif dan menyajikan *interface* yang menarik.

Multimedia merupakan kombinasi dari teks, gambar, seni grafik, suara, animasi dan elemen-elemen video yang dimanipulasi secara digital. Tampilan dan cita rasa dari proyek multimedia harus menyenangkan, estetis, mengundang dan mengikat. Proyek harus memuat konsistensi visual, hanya dengan menggunakan elemen-elemen yang mendukung pesan keseluruhan dari program (Vaughan, 2004).

2.5. Game

Menurut Sibero (2009) Game atau permainan dapat diklasifikasikan menjadi dua bagian besar game fisik dan game elektronik. Game fisik mungkin sudah sering kita

lakukan dalam kehidupan sehari-hari sewaktu masih anak-anak. Seperti lompat tali, petak umpet dan sebagainya. Permainan di sini adalah berhubungan dengan gerak fisik. Dan game elektronik merupakan fenomena yang menarik saat ini. Bahkan dapat dikatakan bahwa hampir semua kalangan menyukai game elektronik. *Electronic game* saat ini berkembang sangat pesat sehingga berbagai konsol baru mulai bermunculan.

2.6. Education Game

Penerapan *education game* bermula dari perkembangan *video game* yang sangat pesat dan menjadikannya sebagai media efektif yang interaktif dan banyak dikembangkan di perindustrian. Melihat kepopuleran *game* tersebut, para pendidik berpikir bahwa mereka mempunyai kesempatan yang baik untuk menggunakan komponen rancangan *game* dan menerapkannya pada kurikulum dengan penggunaan industri berbasis *game*. *Game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan (Hurd dan Jenuings, 2009).

2.7. UML

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian 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.

3. Analisis

3.1. Analisis Aplikasi

Kebutuhan utama dalam perancangan Aplikasi *Education Game* Bahasa Inggris ini adalah berbagai file gambar, audio, animasi serta perangkat lunak yang digunakan untuk membangun aplikasi yaitu: *Google App Inventor*. Semua kebutuhan itu harus dapat dituangkan ke dalam perancangan agar hasil aplikasi dapat sesuai dengan konsep pembangunan perangkat lunak, multimedia, serta pada *education game* bahasa Inggris itu sendiri.

3.2. Perancangan Aplikasi

Perancangan aplikasi meliputi rancangan bentuk aplikasi secara keseluruhan, tahap-tahap yang digunakan dalam perancangan.

3.3.1. Rancangan Bentuk Aplikasi

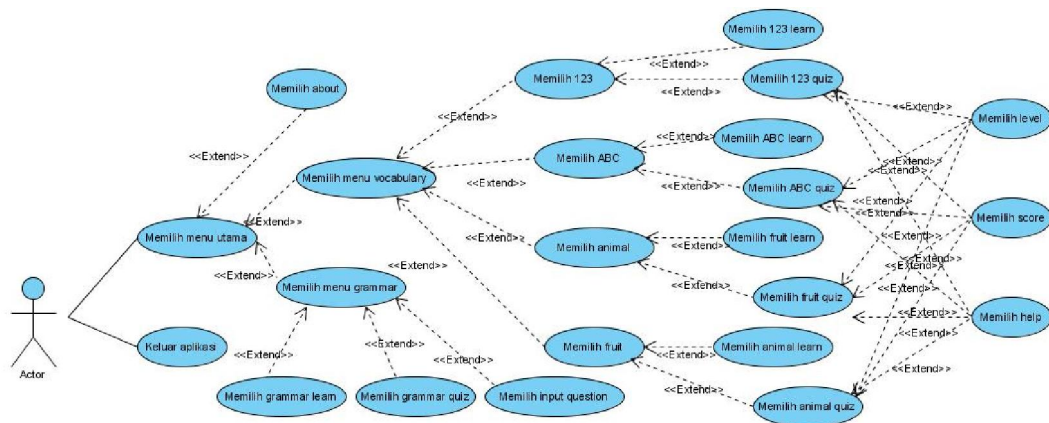
Rancangan bentuk aplikasi menjelaskan konsep aplikasi, bagaimana aplikasi berjalan yang merupakan tahap awal dalam perancangan aplikasi.

Aplikasi ini mempunyai 3 menu utama yaitu:

1. *Vocabulary*, berisi sub-sub menu untuk memainkan game *Vocabulary*. Di dalam setiap sub menu tersebut juga terdapat dua pilihan menu lagi yaitu *Learn* dan *Quiz*. *Learn* berisi tentang pengenalan kata-kata benda dalam bahasa Inggris dan bahasa Indonesia. *Quiz* berisi *game* menebak nama benda yang tertera pada layar. Selain itu di dalam menu *Quiz* terdapat tiga sub menu pilihan, yaitu *Level*, *Score* dan *Help*. Menu *Level* berfungsi untuk mengatur tingkat kesulitan pada *Quiz*. Ada tiga tingkat kesulitan yang dapat dipilih di menu *Level* ini, yaitu *Easy*, *Medium* dan *Hard*. Menu *Score* berisi informasi *score*/nilai yang didapat oleh *user*. Menu *Help* berisi informasi tentang tata cara bermain *Quiz*.
2. *Grammar*, di dalamnya terdapat dua sub menu yaitu *Learn* dan *Quiz*. *Learn* berisi tutorial tentang pengenalan struktur *Grammar* sederhana. *Quiz* berisi *game* menebak kata yang kosong dari sebuah kalimat. Pada menu *Quiz* juga terdapat tiga sub menu pilihan, yaitu *Level*, *Score*, dan *Help*. Menu *Level* berfungsi untuk mengatur tingkat kesulitan pada *Quiz*. Ada tiga tingkat kesulitan yang dapat dipilih di menu *Level* ini, yaitu *Easy*, *Medium* dan *Hard*. Menu *Score* berisi informasi *score*/nilai yang didapat oleh *user*. Menu *Help* berisi informasi tentang tata cara bermain *Quiz*.
3. *About*, berisi tutorial tentang pengenalan *education game* dan penjelasan singkat tentang penggunaan aplikasi dan *game* yang ada pada aplikasi.

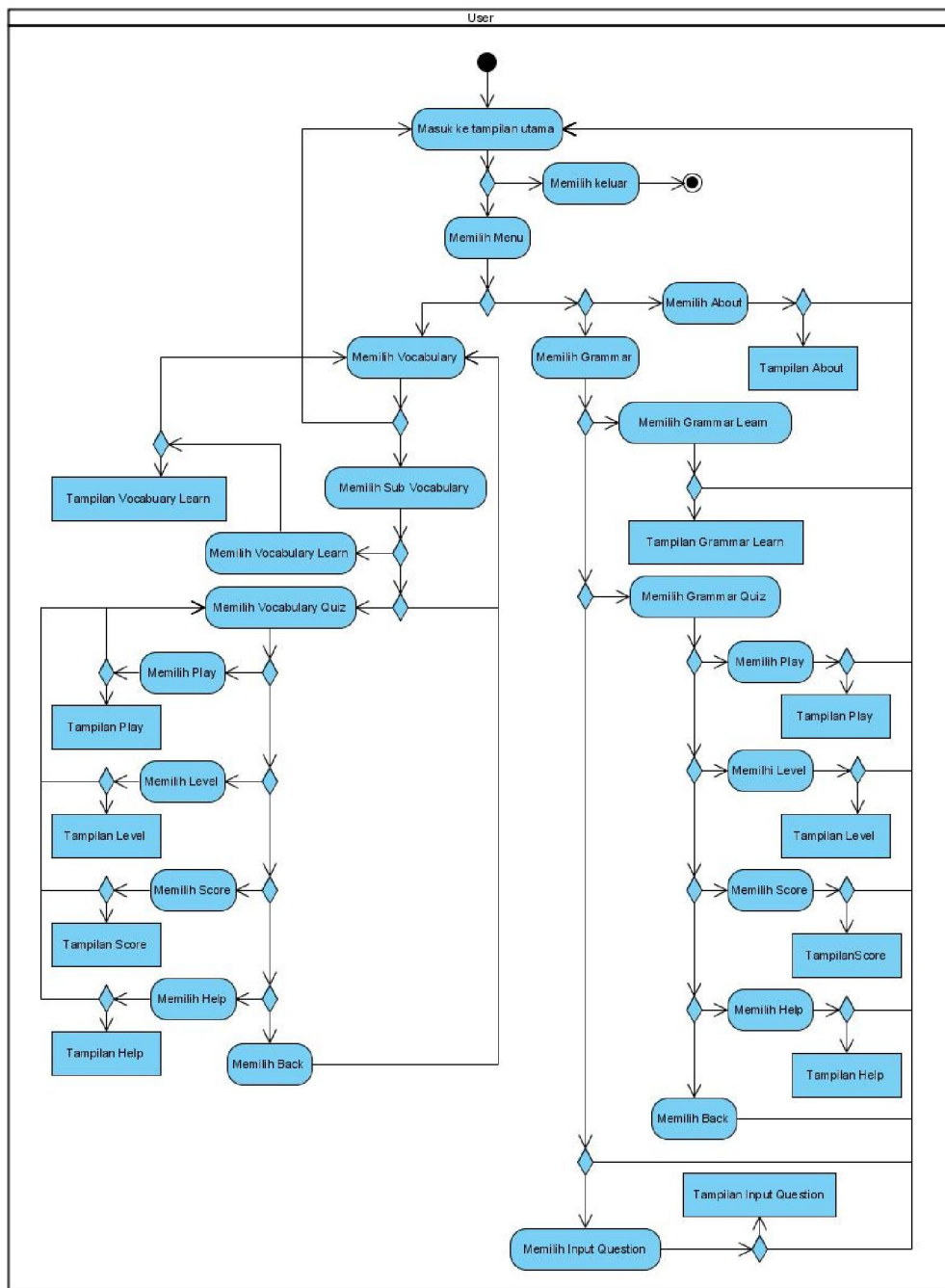
3.3.2. Rancangan Aplikasi Menggunakan UML

Aplikasi *game education* ini dirancang berbasis *object oriented* dengan menggunakan alat bantu tools atau model yaitu UML. Alasan mengapa saat ini pendekatan dalam pengembangan software dengan *object-oriented*, pertama adalah *scalability* dimana obyek lebih mudah dipakai untuk menggambarkan sistem yang besar dan kompleks. Kedua *dynamic modeling*, adalah dapat dipakai untuk permodelan sistem dinamis dan *real time*. UML yang digunakan oleh penulis adalah *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.



Actor adalah anak-anak sebagai pengguna aplikasi untuk belajar dan bermain game. Aktivitas yang dapat dilakukan, sebagai berikut:

1. Jika actor memilih menu utama maka akan muncul *activity* baru yaitu masuk ke tampilan main menu yang menampilkan tiga menu utama yaitu *Vocabulary*, *Grammar*, *About*.
2. Jika actor memilih menu *Vocabulary* maka akan muncul *activity* baru yaitu masuk ke tampilan menu *Vocabulary* yang menampilkan sub-sub menu yaitu 123, ABC, Fruit, Animal.
3. Jika actor memilih setiap sub menu dari menu *Vocabulary* maka akan muncul *activity* baru yaitu masuk ke tampilan dari sub menu yang dipilih yang menampilkan dua sub menu yaitu *Learn* dan *Quiz*.
4. Jika actor memilih menu *Grammar* maka akan muncul *activity* baru yaitu masuk ke tampilan menu *Grammar* yang menampilkan dua sub menu yaitu *Learn* dan *Quiz*.
5. Jika actor memilih menu *About* maka akan muncul *activity* baru yaitu masuk ke tampilan menu *About* yang berisi tentang aplikasi.
6. Jika actor memilih menu keluar maka actor akan keluar dari aplikasi.



Saat *actor* memulai aplikasi, pertama akan muncul dua pilihan menu yaitu ke menu utama dan keluar aplikasi. Menu utama adalah menu dari keseluruhan aplikasi yang ada, disini terdapat pilihan yaitu menu *Vocabulary*, menu *Grammar*, menu *About*.

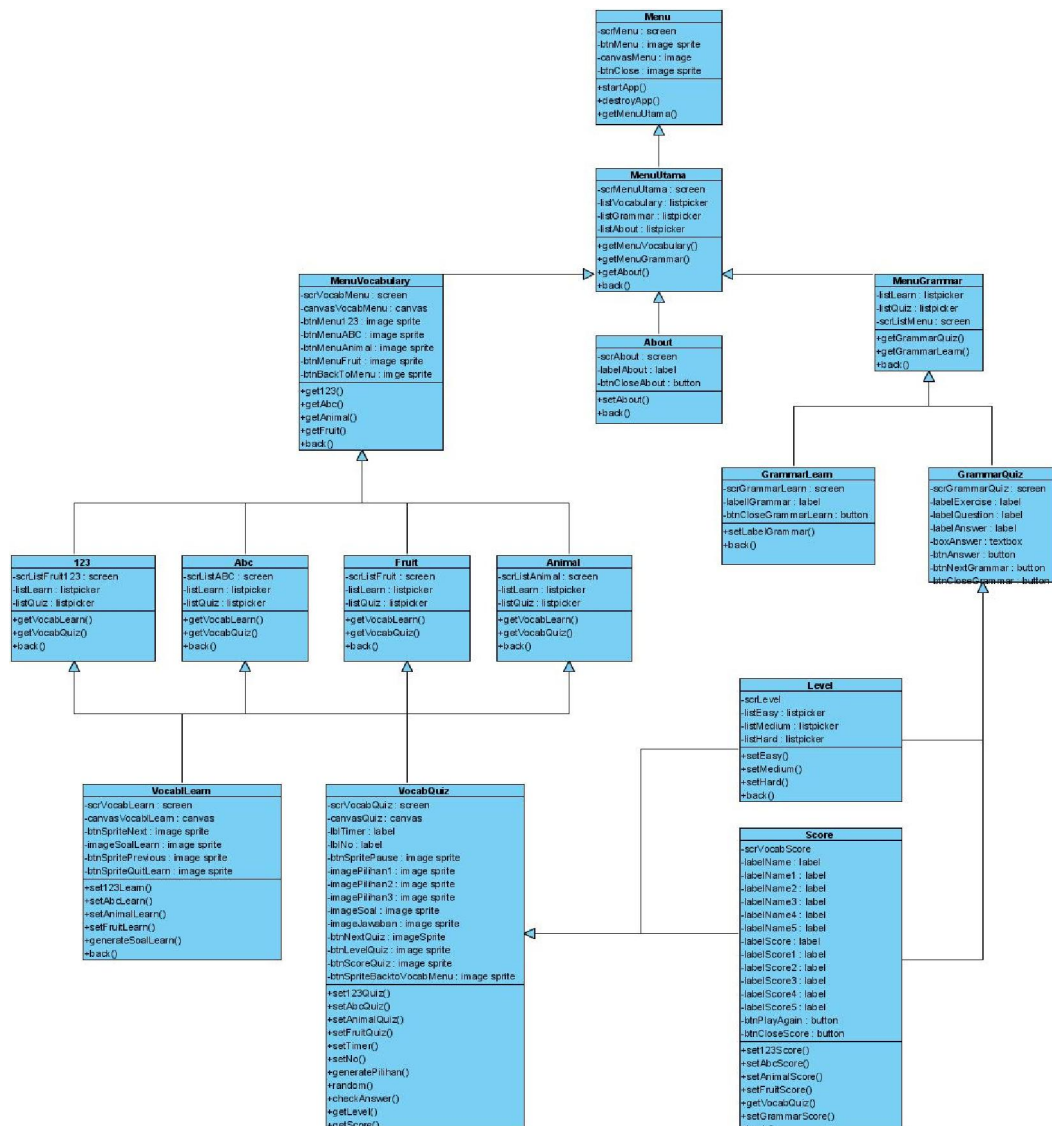
Pada saat *actor* memilih menu *Vocabulary* maka akan menuju ke menu *Vocabulary* yang menampilkan beberapa sub menu yaitu menu 123, ABC, Fruit, Animal, dan jika memilih salah satu dari sub menu tersebut akan muncul *listmenu* yang terdiri dari menu *Learn* dan *Quiz*, jika memilih salah satu dari *listmenu* tersebut akan membawa *actor* menuju tampilan *Learn* atau *Quiz* sesuai dengan *list* menu yang dipilih.

Pada tampilan *Learn* berisi pengenalan materi yang di ajarkan. Pada tampilan *Quiz* berisi lima sub menu pilihan yaitu *Play, Level, Score, Help dan Back to menu*. Isi dari tampilan *Learn* dan *Quiz* akan berbeda-beda sesuai dengan sub menu yang dipilih. Jika actor memilih *Play* maka akan memainkan *Quiz* yang dipilih. Jika actor memilih *Level* maka akan muncul tampilan *list* yang berisi tingkat kesulitan yang dapat dipilih. Jika actor memilih *Score* maka akan menampilkan informasi nilai yang telah diperoleh. Jika actor memilih *Back to menu* maka akan kembali ke *Vocabulary* menu.

Jika actor memilih menu *Grammar* maka akan muncul *listmenu* yang terdiri dari menu *Grammar Learn, Grammar Quiz, dan Input Question*. Jika actor memilih menu *Grammar Learn* maka akan menampilkan tentang materi struktur *Grammar*. Jika actor memilih menu dan *Grammar Quiz* akan menuju ke tampilan yang berisi lima sub menu pilihan yaitu *Level, Score, Help dan Back to menu*. Jika actor memilih *Level* maka akan muncul tampilan *list* yang berisi tingkat kesulitan yang dapat dipilih. Jika actor memilih *Score* maka akan menampilkan informasi nilai yang telah diperoleh. Jika actor memilih *Back to menu* maka akan kembali ke tampilan menu utama. Jika Actor memilih *Input Question* maka akan menampilkan tampilan *input question*.

Jika actor memilih menu *about* maka akan menuju ke screen *about* yang berisi tentang pengenalan aplikasi *education* ini.

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* dari aplikasi *education game*.



3.3. Rancangan Tampilan Aplikasi

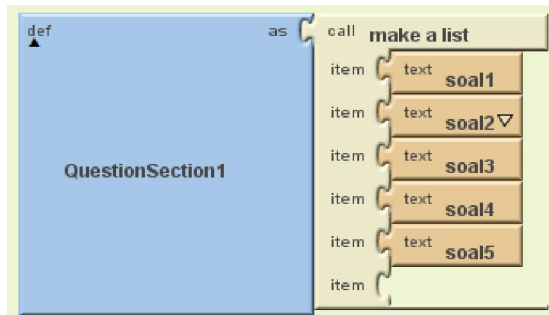
Rancangan yang akan dibuat harus memberikan gambaran dan penjelasan dari setiap gambar, teks dan navigasi. Rancangan tampilan ini menggambarkan keterkaitan setiap halaman dan juga menjelaskan arah komunikasinya. Rancangan tampilan ini bertujuan agar aplikasi yang dihasilkan terlihat lebih menarik, mudah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

3.4 Rancangan Struktur Blok

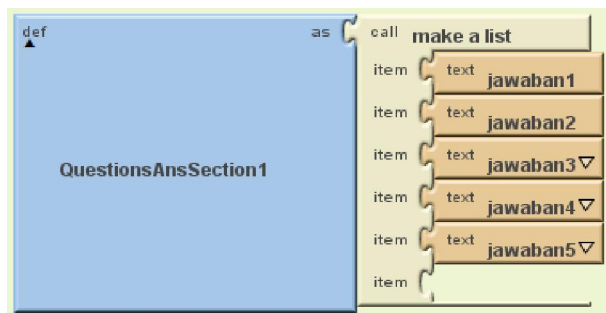
Untuk membangun sebuah aplikasi di App Inventor dibutuhkan koding-koding yang berbentuk blok agar aplikasi bisa berjalan atau berfungsi sesuai dengan rancangan. Berikut ini adalah rancangan struktur blok algoritma *Education Game*:

Block Editor App Inventor menyediakan berbagai macam fungsi yang bisa digunakan sesuai dengan kebutuhan user, misalnya untuk menyimpan data, user cukup

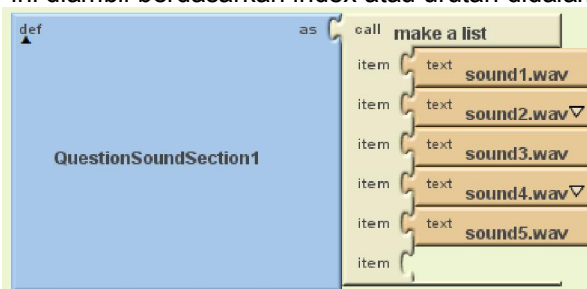
menggunakan fungsi variable dengan fungsi list yang bisa digunakan untuk menyimpan data-data, seperti data soal, data jawaban, data gambar, data suara. Berikut ini contoh struktur penyimpanan data soal, jawaban, gambar dan suara :



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.

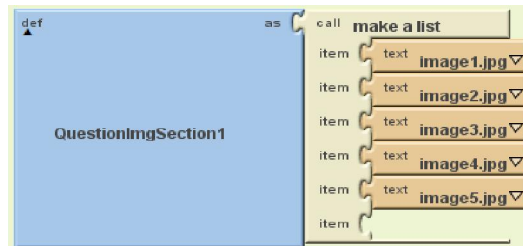


Seperti terlihat pada gambar diatas, struktur penyimpanan data jawaban dibuat didalam suatu variabel yang bernama QuestionAnsSection1 kemudian blok variabel tersebut digabungkan dengan blok fungsi make a list yang berfungsi menyimpan data jawaban yang berbentuk list, data jawaban yang disimpan kedalam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



Seperti terlihat pada gambar 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. Implementasi dan Pembahasan

4.1 Implementasi

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.

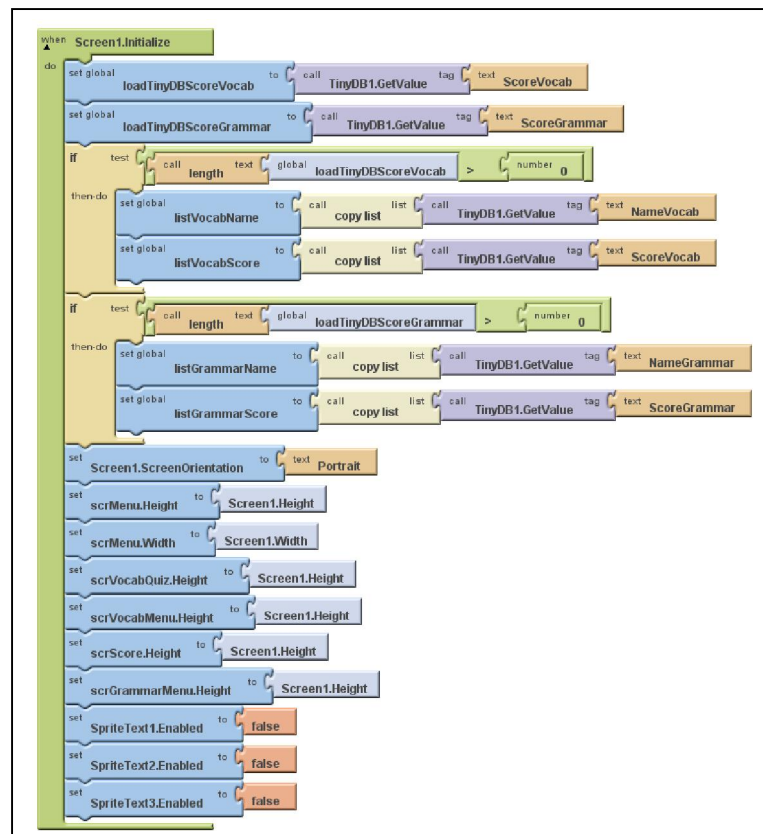
Implementasi dan pembahasan perancangan aplikasi *Education Game* ini menggunakan emulator App Inventor dan handphone Samsung Galaxy Young.

Berikut ini adalah implementasi aplikasi *Eduction Game*

1. Tampilan Awal Aplikasi



Tampilan awal adalah tampilan paling pertama atau pembuka aplikasi. Latar belakang (*background*) *game* menggunakan warna-warna cerah karena mempertimbangkan *user* adalah anak-anak. Dalam tampilan awal ini menampilkan dua menu yaitu: menu utama dan menu *close*. *Block Coding* dari tampilan awal aplikasi dapat dilihat pada gambar dibawah ini.



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 pada aplikasi *education game* ini, dalam hal ini diambil sample fungsi logika untuk skoring. Untuk mengecek apakah terdapat kesalahan logika dalam aplikasi ini, maka dibuat tabel percobaan sebagai berikut.

| Jawaban Benar | Score yang disimpan dan ditampilkan |
|---------------|-------------------------------------|
| 0 Soal | 0 |
| 1 Soal | 100 |
| 2 Soal | 200 |
| 3 Soal | 300 |
| 4 Soal | 400 |
| 5 Soal | 500 |
| 6 Soal | 600 |
| 7 Soal | 700 |
| 8 Soal | 800 |
| 9 Soal | 900 |
| 10 Soal | 1000 |

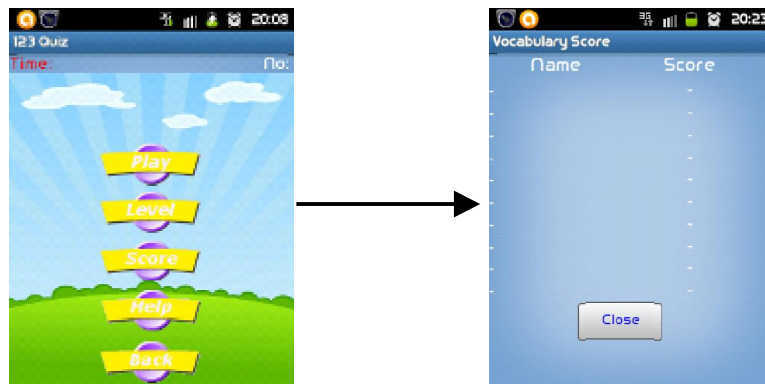
Pada tabel di atas menunjukkan hasil skoring pada aplikasi *education game* ini sesuai dengan jumlah jawaban yang benar. Untuk itu dapat dipastikan aplikasi *education game* ini sudah tidak memiliki *Logic Error* atau kesalahan logika.

2. White Box Testing

Pengetesan white box merupakan metode perancangan text case yang menggunakan struktur control dari perancangan procedural untuk mendapatkan text case. Tes ini dimaksudkan untuk meramalkan cara kerja perangkat lunak secara rinci. Karenanya logical path (jalur logika) perangkat lunak akan di tes dengan menyediakan test case yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara spesifik.

Pada pembuatan aplikasi *education game* ini pengujian *whitebox* dilakukan pada bagian *scoring*. Saat user telah menyelesaikan permainan maka akan muncul form untuk pengisian nama score. Setelah nama diinputkan, score akan disimpan ke dalam data base. Untuk itu diperlukan pengujian apakah score benar-benar tersimpan dalam database.

1. Langkah pertama masuk ke menu score untuk memastikan papan score benar-benar kosong.



2. Selanjutnya memainkan salah satu game hingga selesai dan akan muncul form pengisian nama score.



3. Inputkan nama dan nama akan muncul di papan score.



4. Langkah terakhir yaitu restart aplikasi dan masuk ke menu score.



Dari pengujian diatas dapat disimpulkan bahwa score benar-benar tersimpan dalam database.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Setelah melalui beberapa tahapan dalam menyelesaikan Aplikasi Android Game Edukasi Bahasa Inggris menggunakan App Inventor, dapat disimpulkan bahwa :

1. Aplikasi ini memiliki interface yang *user friendly* sehingga mudah digunakan oleh *user*.
2. Aplikasi ini menyajikan tampilan yang interaktif dan penentuan durasi *game* dengan penggunaan *timer*.
3. Aplikasi ini dapat menyimpan dan menampilkan *score* supaya *user* lebih termotivasi memainkannya.
4. Menu dan fitur yang disediakan dalam aplikasi ini sesuai dengan rancangan dan berfungsi dengan baik.
5. Menu Game yang disediakan dapat menarik minat belajar anak pada Materi pembelajaran bahasa Inggris.
6. Materi pembelajaran bahasa Inggris yang disediakan pada aplikasi ini sudah cukup sesuai dengan materi pembelajaran bahasa Inggris di sekolah.

7. Secara keseluruhan aplikasi Game Edukasi Bahasa Inggris ini nyaman digunakan.
8. Berdasarkan hasil pengujian program aplikasi Game Edukasi Bahasa Inggris ini sudah dapat dipastikan bebas dari *syntax error*, *runtime error*, dan *logic error*.

5.2 Saran – 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. *Game-game* dalam aplikasi ini diharapkan dapat lebih bervariasi lagi bentuknya agar anak-anak sebagai *user* semakin temotivasi memainkannya.
2. Penyempurnaan pengaturan score dengan menambah fitur total score dari semua permainan.
3. Menambahkan fitur update aplikasi dan soal melalui internet agar penambahan soal lebih efektif.
4. Penyempurnaan desain *interface* khususnya pada Menu Grammar agar lebih menarik lagi.
5. Penyempurnaan efek-efek suara pada aplikasi Game Edukasi Bahasa Inggris ini.
6. Penyempurnaan resolusi *screen* agar dirancang mendukung dengan semua device.

Daftar Pustaka

- Anonim.2008.Silabus,<http://litbangkemdiknas.net/detail.php?csection=kurikulum>, diakses tanggal 15 Desember 2011 pukul 20:30
- Artsiyanti, Diba, E,P., S.S. 2002. Bagaimana Meningkatkan Mutu Hasil Pelajaran Bahasa Inggris di Sekolah. Jakarta: PIKMI Ganesha.
- Fowler, M. 2005.*UML Distilled*. Boston: Pearson Education
- Hurd, Daniel dan Jenuings, Erin. 2009. *Standardized Educational Games Ratings: Suggested Criteria*. Karya Tulis Ilmiah.
- Idris, Husni. 2008. Perkembangan Multimedia Pembelajaran Berbantuan Komputer. <http://scribd.com/perkembanganmultimedia.htm>. Diakses tanggal: 5 Desember 2011 pukul 21:00
- Magnuson, B. 2009. *Building Blocks for Mobile Games: A Multiplayer Framework for App Inventor for Android*. Massachusetts: Massachusettes Institute of Techology. Karya Tulis Ilmiah
- 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
- Scott, Wendy A. & Ytreberg, Lisbeth H. 2006. *Teaching English to children*. London: Longman
- Sigit, Bambang, Joko. 2008. Pengembangan Pembelajaran dengan Menggunakan Multimedia Interaktif untuk Pembelajaran yang Berkualitas. Karya Tulis Ilmiah. Semarang, Indonesia: Universitas Negeri Semarang
- Sholiq. 2006. Pemodelan sistem Informasi Berorientasi Objek dengan UML. Yogyakarta: Graha Ilmu
- Sibero, Ivan C. 2009. Langkah Mudah Membuat Game 3D: MediaKom.
- Stephanus, B.R. 2011. Mudah Membuat Aplikasi Android. Yogyakarta: C.V Andi Offset
- Vaughan, Tay. 2004. *Multimedia: Making It Work*. Edisi 6. Yogyakarta: Penerbit Andi
- Wolber, D., dkk. 2011. *App Inventor Create Your Own Android Apps*. Canada: O'Reilly Media Inc