

**M – IQ TEST**  
**(Mobile Intellegence Quotient Test)**

**TUGAS AKHIR**

Oleh :

**Saut Jhonny Simatupang 3311111026**

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA**  
**POLITEKNIK NEGERI BATAM**

**BATAM**

**2014**

## **HALAMAN PENGESAHAN**

### **M - IQ TEST (Mobile Intelligence Quotient Test)**

**Oleh :**

**Saut Jhonny Simatupang (3311111026)**

Tugas Akhir ini telah diterima dan disahkan  
sebagai persyaratan untuk memperoleh gelar

Ahli Madya

di

**PROGRAM STUDI DIPLOMA 3 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BATAM**

Batam, 04 Juli 2014

Disetujui oleh;

Pembimbing,

**Riwinoto M.Kom**

**NIP. 1979080620121001**

## HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311111026

Nama : Saut Jhonny Simatupang

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

### **M - IQ TEST (Mobile Intelligence Quotient Test)**

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 04 Juli 2014

**Saut Jhonny Simatupang**  
3311111026

## **PERSEMBAHAN**

*Untuk Orangtua yang telah membesarkan dan meberikan restunya kepada kami,*

*Untuk seluruh teman-teman kami di Politeknik Negeri Batam,*

*Untuk seluruh anggota keluarga tercinta.*

## **KATA PENGANTAR**

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas segala rahmat, berkah, karunia dan ridho-Nya sehingga saya dapat menyelesaikan laporan Tugas Akhir yang berjudul “M-IQ TEST (Mobile Intelligence Quotient Test)”.

Dalam kesempatan ini, saya ingin menyampaikan ucapan ribuan terima kasih kepada pihak-pihak yang telah membantu saya dalam menyelesaikan laporan ini, baik dalam memberikan semangat, dukungan, bimbingan, waktu dan biaya, antara lain:

1. Orang tua dan keluarga yang telah memberikan dukungan baik moral maupun materi serta ribuan kata maaf jika dari dulu sampai sekarang sering mengecewakan dan menyia-nyiakan kesempatan yang telah diberikan.
2. Bapak Riwinoto selaku Pembimbing Tugas Akhir ini. Terima kasih atas bimbingan dan motivasi yang bapak berikan dalam penyelesaian laporan ini dan saya juga mohon maaf apabila terdapat kesalahan selama proses bimbingan.
3. Bapak Dwi ely selaku Koordinator Tugas Akhir.
4. Dosen penguji yang telah bersedia memberikan kritik dan saran sebagai masukan bagi penulis.
5. Buat sahabat dan teman-teman yang tidak dapat penyusun sebutkan satu per satu yang telah membantu dalam menyelesaikan laporan ini. Terima kasih atas diskusi-diskusi serta kritik dan saran yang telah diberikan.

Penyusun juga mengucapkan terima kasih kepada pihak-pihak lainnya yang sumbernya berupa artikel, tulisan, dan buku yang telah penyusun jadikan referensi guna penyusunan laporan ini, semoga dapat terus berkarya, guna menghasilkan tulisan-tulisan yang mengacu terwujudnya generasi masa depan yang lebih baik.

Penyusun berharap, semoga informasi yang ada dalam laporan ini dapat berguna bagi penyusun khususnya dan bagi para pembaca pada umumnya.

Pepatah mengatakan “Tak ada gading yang tak retak”, penyusun menyadari bahwa laporan ini masih jauh dari sempurna, jikalau banyak kekurangan dan kesalahan disana-sini, kami selaku penyusun laporan sebelumnya memohon maaf.

Penyusun setulus hati menerima kritik dan saran membangun, guna penyempurnaan laporan ini.

Batam, 04 Juli 2014

Saut Jhonny Simatupang

## **ABSTRAK**

### **M - IQ TEST (Mobile Intelligence Quotient Test)**

Kecerdasan atau yang biasa dikenal dengan IQ dalam bahasa Inggris Intelligence Quotient adalah istilah umum yang digunakan untuk menjelaskan sifat pikiran yang mencakup sejumlah kemampuan, seperti kemampuan menalar, merencanakan, memecahkan masalah, berpikir abstrak, memahami gagasan, menggunakan bahasa dan belajar. Karenanya Tes IQ ini sering dihubungkan dengan kecerdasan seseorang. Model, materi, dan bidang yang diuji dalam Tes IQ sebagian besar merujuk kepada tes GRE. Tes GRE menjadi standar internasional syarat penerimaan mahasiswa Perguruan Tinggi. Game ini untuk mengukur Potensi Intelligence Quotient (IQ) pengguna.

Aplikasi ini terdiri dari lima set yang berbeda masing-masing terdapat 20 pertanyaan yang dirancang untuk mengevaluasi IQ. Aplikasi ini bertujuan agar anda dapat mengenal bentuk dan jenis soal yang biasa muncul pada ujian tes IQ.

## **ABSTRACT**

### **M - IQ TEST (Mobile Intelligence Quotient Test)**

Intelligence or commonly known in English IQ: Intelligence Quotient is a general term used to describe the nature of the mind which includes a number of capabilities, such as the ability to reason, plan, solve problems, think abstractly, comprehend ideas, to use language and learning. IQ Test is therefore often associated with intelligence. Models, materials, and field tested in a IQ Test mostly refer to the GRE test. GRE test became the international standard university admission requirements. This game measures your potential intelligence quotient (IQ), which will allow you to know the strength of your own mind.

This application consists of five different sets each containing 20 questions designed to evaluate your Intelligence Quotient (IQ). This application aims to make you can recognize shapes and types of problems that commonly appear on the exam IQ test.

## DAFTAR ISI

<a href="#">Halaman Judul</a> .....	i
<a href="#">Halaman Pengesahan</a> .....	ii
<a href="#">Halaman Pernyataan</a> .....	iii
<a href="#">Halaman Persembahan</a> .....	iv
<a href="#">Kata Pengantar</a> .....	v
<a href="#">Halaman Abstrak</a> .....	vii
Bab I Pendahuluan.....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	3
I.3 Batasan Masalah .....	3
I.4 Tujuan.....	4
I.5 Sistematika Penulisan .....	4
Bab II Landasan Teori .....	6
II.1 Unified Modeling Language (UML).....	6
II.2 Diagram Usecase .....	7
II.3 Rekayasa Perangkat Lunak Berbasis Object Oriented .....	7
II.3.1 Spesifikasi Kebutuhan dan Diagram Usecase.....	8
II.3.1.1 Notasi Diagram Usecase .....	9
II.3.2 Diagram Robustness .....	10
II.3.2.1 Notasi Robustness Diagram .....	11
II.3.3 Diagram Sequence .....	12
II.3.3.1 Notasi Sequence Diagram .....	13
II.3.4 Diagram Kelas Analisis.....	15
II.3.4.1 Notasi Diagram Kelas .....	16
II.4 Adobe Flash.....	18
II.4.1 ActionScript .....	18
II.5 XML (eXtensible Markup Language) .....	18
II.6 Android.....	19
II.6.1 Sejarah Singkat Sistem Operasi Android.....	19
II.6.2 Bahasa Pemrograman Java di Android .....	21
II.6.3 Arsitektur Android .....	22

II.6.4	Struktur Aplikasi Android.....	23
II.6.5	Versi Android.....	23
II.6.6	Pengembangan Android.....	25
II.7	Gambaran Umum pada Aplikasi Sejenis.....	26
II.8	Rumus Cara Penilaian dan Cara Pengacuannya.....	28
II.8.1	Pengolahan Nilai.....	28
II.8.2	Skala Penilaian (Skala 1-100).....	28
II.9	Klasifikasi Perhitungan IQ (Intellegent Quotient).....	29
II.9.1	IQ (Intellegent Quotient).....	29
II.9.2	Ukuran Tingkat IQ Manusia.....	29
II.9.3	Cara Perhitungan Nilai Pada Aplikasi M-TPA.....	31
II.9.4	Deskripsi IQ Score.....	32
Bab III	Analisis dan Perancangan.....	33
III.1	Analisis Aplikasi Tes IQ.....	33
III.1.1	Deskripsi Umum Sistem.....	33
III.1.2	Analisis Karakteristik Pengguna.....	34
III.1.3	Analisis Kebutuhan Sistem pada Computer dan Mobile.....	35
III.1.4	Analisis Kebutuhan Fungsional.....	35
III.2	Perancangan Aplikasi Tes IQ.....	36
III.2.1	Diagram Use Case Tes IQ.....	36
III.2.2	Skenario Use Case Tes IQ.....	37
III.2.3	Analisis Kelas Tes IQ.....	40
III.2.4	Diagram Sequence.....	41
III.2.5	Diagram Kelas Tes IQ.....	50
III.3	Perancangan Tampilan.....	51
III.3.1	Rancangan Tampilan Layar Utama.....	51
III.3.2	Rancangan Tampilan Home Page Menu.....	52
III.3.3	Rancangan Tampilan Login.....	53
III.3.4	Rancangan Tampilan Question Set.....	54
III.3.5	Rancangan Menampilkan Notice.....	55
III.3.6	Rancangan Tampilan Grade.....	56

III.3.7	Rancangan Review Question Answered .....	57
III.3.8	Rancangan Tampilan Answer Questions .....	58
Bab IV	Implementasi dan Pengujian.....	59
IV.1	Implementasi .....	59
IV.1.1	Struktur Objek M-IQ Test.....	59
IV.1.2	File Action Script .....	62
IV.1.3	Implementasi Antarmuka .....	63
IV.2	Pengujian dan Hasil .....	75
IV.2.1	Hasil Pengujian Fungsionalitas seluruh Menu pada Aplikasi.....	75
IV.2.2	Pengujian Aplikasi Pada Pengguna.....	79
Bab V	Kesimpulan dan Saran .....	81
V.1	Kesimpulan.....	81
V.2	Saran .....	81
	<a href="#">Daftar Pustaka</a> .....	82
	<a href="#">Lampiran</a> .....	3

## DAFTAR GAMBAR

Gambar II-1 Simbol-simbol pada diagram <i>Robustness</i> .....	10
Gambar II-2 Contoh Notasi Diagram Sequence .....	13
Gambar II-3 Contoh Diagram Kelas .....	16
Gambar II-4 Arsitektur Android .....	22
Gambar III-1 Deskripsi Umum Sistem .....	33
Gambar III-2 Diagram Use Case Tes IQ.....	36
Gambar III-3 Analisis Kelas Tes IQ .....	40
Gambar III-4 Sequence GUI-Layar Utama.....	41
Gambar III-5 Sequence GUI-Start .....	42
Gambar III-6 Sequence GUI-Login .....	43
Gambar III-7 Sequence GUI-Question Set .....	44
Gambar III-8 Sequence GUI-Grade .....	45
Gambar III-9 Sequence GUI-Answer .....	46
Gambar III-10 Sequence GUI-Instruction .....	47
Gambar III-11 Sequence GUI-About.....	48
Gambar III-12 Sequence GUI-History.....	49
Gambar III-13 Diagram Kelas Tes IQ .....	50
Gambar III-14 Tampilan Layar Utama .....	51
Gambar III-15 Tampilan Layar Home Page Menu .....	52
Gambar III-16 Tampilan Layar Login .....	53
Gambar III-17 Tampilan Layar Question Set .....	54
Gambar III-18 Menampilkan Notice.....	55
Gambar III-19 Menampilkan Grade.....	56
Gambar III-20 Tampilan Review Question Answered .....	57
Gambar III-21 Tampilan Answer Questions.....	58
Gambar IV-1 Halaman Utama Aplikasi.....	63
Gambar IV-2 Login Aplikasi M-IQ Test .....	64
Gambar IV-3 Input Name dan Age .....	65

Gambar IV-4 Choose Question Set.....	66
Gambar IV-5 Tampilan Soal M-IQ Test.....	67
Gambar IV-6 Menampilkan Soal Setelah di Jawab .....	68
Gambar IV-7 Tampilan Grade IQ .....	69
Gambar IV-8 Tampilan Review Soal.....	70
Gambar IV-9 Tampilan Menu Instruction .....	71
Gambar IV-10 Tampilan Menu About.....	72
Gambar IV-11 Tampilan Menu History Sebelum Menjawab Soal.....	73
Gambar IV-12 Menu History Setelah Menjawab Soal .....	74

## DAFTAR TABEL

Tabel 1 Notasi Diagram Use Case .....	9
Tabel 2 Notasi <i>Robustness Diagram</i> .....	11
Tabel 3 Notasi Sequence Diagram.....	13
Tabel 4 Notasi Diagram Kelas .....	16
Tabel 5 Kategori Penilaian.....	31
Tabel 6 Deskripsi IQ Score .....	32
Tabel 7 Karakteristik Pengguna .....	34
Tabel 8 Analisis Kebutuhan Perangkat Lunak pada Komputer dan Mobile.....	35
Tabel 9 Fungsi Layar Utama.....	51
Tabel 10 Fungsi Menu Home Page.....	52
Tabel 11 Fungsi Login .....	53
Tabel 12 Fungsi Tampilan Question Set.....	54
Tabel 13 Fungsi Tampilan Notice.....	55
Tabel 14 Fungsi Tampilan Grade.....	56
Tabel 15 Fungsi Tampilan Review Answers .....	57
Tabel 16 Tampilan Answers Questions .....	58
Tabel 17 Struktur Objek M-IQ Test.....	59
Tabel 18 Struktur File M-IQ Test .....	62
Tabel 19 Pengujian Fungsionalitas Seluruh Menu.....	75
Tabel 20 Pengujian Aplikasi pada Pengguna.....	79

# BAB I

## PENDAHULUAN

Seiring dengan berkembangnya teknologi telepon genggam (*handphone*) dan perangkat bergerak (*gadget mobile*), maka penulis berusaha untuk dapat menambahkan satu fungsi penggunaan pada perangkat tersebut. Diharapkan dapat dimanfaatkan dalam dunia pendidikan dengan memanfaatkan teknologi informasi dan jaringan komputer yang saling terhubung.

### I.1 Latar Belakang

Dalam era digitalisasi saat ini, perkembangan teknologi komunikasi sungguh sangat pesat. Bahkan teknologi komunikasi berperan penting dalam kehidupan manusia. *Handphone* (telepon genggam) yang pada awal ditemukannya hanya untuk telepon dan kirim pesan, namun sekarang *handphone* sudah sangat canggih. Di mana yang dulunya hanya dapat dilakukan komputer sekarang dapat dilakukan oleh *handphone*, yang sering disebut dengan julukan *smartphone* atau telepon pintar.

Perusahaan-perusahaan *Operating System* (OS) besar seperti Apple, BlackBerry, Nokia, Windows dan Google berlomba-lomba menciptakan OS yang canggih untuk kebutuhan *smartphone* tersebut. Salah satunya adalah OS Android yang diciptakan serta dikembangkan oleh Google dengan berbasis Linux.

Teknologi Android akhir-akhir ini sudah menjamur dikalangan remaja, baik yang masih sekolah/kuliah maupun yang sudah bekerja. Fenomena ini disebabkan oleh murahnya teknologi Android, sehingga semua kalangan mampu untuk membelinya. Android disukai oleh pengguna karena sifatnya yang terbuka */open source code* sehingga bisa diubah-ubah sesuai keinginan pengguna, serta tampilannya yang tidak kalah menarik. Bahkan lebih menarik daripada OS yang lain, misal iOS, OS BB, Windows Mobile serta Symbian [1].

Android bisa dimanfaatkan dalam berbagai bidang, salah satu contoh penggunaannya bisa dipergunakan dalam bidang pendidikan, karena dapat memudahkan pengguna untuk mengakses data dan belajar secara cepat seperti melakukan tes potensi akademik. Beberapa aplikasi serupa yang telah dikembangkan dalam berbagai device telah banyak yang dikembangkan banyak orang misalnya pada website versi online [1].

Tes IQ (Intellegent Quotient) merupakan angka yang menjelaskan tingkat kecerdasan seseorang yang dibandingkan dengan sesamanya dalam satu populasi. Dimana berbagai macam tes untuk mengetahui bakat dan kemampuan seseorang telah banyak dikembangkan salah satunya Tes potensi akademik yang sudah banyak diterapkan hampir di seluruh Universitas saat ujian penerimaan mahasiswa baru. Hal ini juga diterapkan pada universitas Politeknik Negeri Batam pada saat ujian penerimaan mahasiswa baru. Hasil tes potensi akademik akan membantu para penguji dalam mengambil keputusan untuk memprediksi kelulusan calon mahasiswa terhadap program studi yang dipilih. Sekaligus hasil tes yang didapat oleh pengguna aplikasi akan mengubah nilai menjadi tingkat IQ (*intelligence quotients*) dari pengguna, sehingga pengguna mengetahui tingkat IQ yang dimiliki [2].

Kelulusan tes masuk pada saat ujian penerimaan mahasiswa baru berdasarkan hasil tes potensi akademik dan mata pelajaran lainnya seperti bahasa Indonesia, bahasa inggris, dan lainnya yang diterapkan oleh Politeknik Negeri Batam menjadi tolak ukur bagi penulis untuk mengimplementasikannya ke dalam suatu aplikasi berbasis android untuk memudahkan calon mahasiswa untuk proses belajar secara cepat dan praktis.

Dengan system ini diharapkan calon mahasiswa dapat melakukan tes akademik secara *mobile* dengan menggunakan perangkat android yang telah terpasang aplikasi M-IQ TEST. Aplikasi diharapkan dapat memberikan soal, jawaban dan lain sebagainya secara nyata (*realtime*) guna proses kelangsungan tes akademik.

## **I.2 Rumusan Masalah**

Berdasarkan latar belakang permasalahan di atas, maka permasalahan yang akan dibahas pada tugas akhir ini dapat dirumuskan sebagai berikut:

1. Bagaimana membangun sebuah aplikasi sebagai sarana Tes IQ pengguna.
2. Bagaimana menambahkan satu aplikasi yang berfungsi untuk melakukan tes IQ pada *smartphone*.
3. Bagaimana aplikasi Tes IQ dapat digunakan pada komunitas Internasional
4. Bagaimana aplikasi Tes IQ ini dapat menyajikan data hasil tes secara benar?

## **I.3 Batasan Masalah**

Dari latar belakang masalah dan rumusan masalah yang telah dijelaskan di atas, maka dibuat batasan-batasan masalah yang menjadi ruang lingkup penyusunan tugas akhir agar semakin terarah dan dapat memperjelas permasalahan. Adapun batasan-batasan masalah tersebut adalah:

1. Aplikasi Tes IQ dapat melakukan autotentifikasi data untuk setiap pengguna.
2. Setelah *client* mengirimkan (*submit*) jawaban maka sistem akan memeriksa dan secara cepat memberikan jawaban hasil tes.
3. Menampilkan nilai hasil tes berupa tingkat IQ (*intelligence quotients*) yang dimiliki pengguna.
4. Pengguna dapat melakukan review terhadap soal dan jawaban yang telah dikerjakan.
5. Aplikasi ini menggunakan 4 model Tes yaitu tes kemampuan kata (verbal), tes angka (*numeric*), tes daya nalar (*analogy*), dan tes gambar (*spatial*) sebagai variable masukan (*input*) dalam bahasa inggris.

Aplikasi ini tidak menangani beberapa hal dibawah ini:

1. Tes dalam bentuk essay.
2. Aplikasi tidak memberikan pembahasan soal secara rinci.

## **I.4 Tujuan**

Adapun tujuan yang akan di capai dalam penelitian ini yaitu:

1. Dapat membangun aplikasi media pembelajaran berupa latihan Tes IQ berbasis android.
2. Aplikasi dapat menyajikan tingkat IQ (*intelligence quotients*) dari pengguna.
3. Aplikasi menyajikan soal dengan menggunakan bahasa inggris.
4. Aplikasi tes IQ dengan segala fasilitasnya diharapkan dapat memberikan hasil tes dengan cepat dan benar.

## **I.5 Sistematika Penulisan**

Sistematika penulisan laporan Tugas Akhir Tes IQ adalah sebagai berikut:

### **BAB I: Pendahuluan**

Berisi tentang latar belakang pembuatan tugas akhir, tujuan yang akan dicapai, pembahasan, metodologi dan sistematika pembelajaran.

### **BAB II: Landasan Teori**

Meliputi uraian dari teori-teori yang akan digunakan sebagai dasar pada perencanaan dan pembuatan tugas akhir.

### **BAB III: Analisis dan Perancangan**

Berisi tentang analisis dan perancangan suatu program simulasi.

### **BAB IV: Implementasi dan pengujian**

Membahas tentang implementasi dan pengujian dari perangkat lunak yang telah dibuat serta analisisnya.

### **BAB V: Kesimpulan dan Saran**

Berisi tentang kesimpulan dan saran yang didapatkan selama proses perencanaan dari sistem serta rencana pengembangan dari tugas akhir dimasa yang akan datang.

### **Daftar Pustaka**

Berisi tentang referensi-referensi yang telah digunakan sebagai landasan selama pembuatan tugas akhir.

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini berisi tentang teori-teori yang berhubungan dengan perancangan aplikasi Mobile Intelligence Quotient Test (M-IQ Test) dan aplikasi penunjang proses pembuatan aplikasi.

#### **II.1 Unified Modeling Language (UML)**

*UML (Unified Modeling Language)* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented).

Pemecahan masalah utama dari Object Oriented biasanya dengan penggambaran dalam bentuk model. Model abstrak (semu) merupakan gambaran detail dari inti masalah yang ada, umumnya sama seperti refleksi dari problem yang ada pada kenyataan. Beberapa modeling tool yang dipakai adalah bagian dari dasar UML, kependekan dari *United Modeling Language*.

UML terdiri atas beberapa diagram, yaitu :

1. Diagram Use Case
2. Diagram Class
3. Diagram Package
4. Diagram Sequence
5. Diagram Collaboration
6. Diagram StateChart
7. Diagram Activity
8. Diagram Deployment

Semakin kompleks bentukan sistem yang akan dibuat, maka semakin sulit komunikasi antara orang-orang yang saling terkait dalam pembuatan dan pengembangan software yang akan dibuat. Pada masa lalu, UML mempunyai peranan sebagai software blueprint (gambaran) language untuk analisis sistem,

designer, dan programmer. Sedangkan pada saat ini, merupakan bagian dari software trade (bisnis software). UML memberikan jalur komunikasi dari sistem analis kemudian designer, lalu programmer mengenai rancangan software yang akan dikerjakan.

## II.2 Diagram Usecase

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu *apa yang dilakukan* bukan *bagaimana melakukannya*. Diagram Use Case dekat kaitannya dengan kejadian-kejadian. Kejadian (scenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem.

Diagram Use Case berguna dalam tiga hal :

1. Menjelaskan fasilitas yang ada (requirements)  
Use Case baru selalu menghasilkan fasilitas baru ketika sistem di analisa, dan design menjadi lebih jelas.
2. Komunika dengan klien  
Penggunaan notasi dan simbol dalam diagram Use Case membuat pengembang lebih mudah berkomunikasi dengan klien-kliennya.
3. Membuat test dari kasus-kasus secara umum  
Kumpulan dari kejadian-kejadian untuk Use Case bisa dilakukan test kasus layak untuk kejadian-kejadian tersebut.

## II.3 Rekayasa Perangkat Lunak Berbasis *Object Oriented*

Dalam aplikasi-aplikasi bisnis, metodologi-metodologi pengembangan aplikasi yang menggunakan bahasa pemrograman terstruktur (*structure programming language*), DFD dan ERD kurang dapat beradaptasi dengan kebutuhan dan harapan pengguna (*user's need and expectations*). Dengan alasan itu, para pakar dibidang perancangan perangkat lunak pada sekitar tahun 1980-1990 mulai bekerja dengan bahasa pemrograman yang berorientasi objek atau lebih dikenal dengan OOP (*Object Oriented Programming*) seperti C++ dan Java.

Dengan demikian diperlukan metodologi dan perkakas-perkakas (*tools*) yang lebih sesuai. Dalam hal ini, UML (*Unified Modelling Language*) yang merupakan metodologi kolaborasi antara metode-metode Booch, OMT (*Object Modelling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metode lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek (OOP) [5].

### **II.3.1 Spesifikasi Kebutuhan dan Diagram Use Case**

Sebuah sistem/perangkat lunak pada umumnya dikembangkan berdasarkan analisis kebutuhan (*requirements analysis*). Analisis kebutuhan ini adalah tahap konseptualisasi, yaitu suatu tahap yang mengharuskan analis dan perancang sistem/perangkat lunak untuk berusaha tahu secara pasti mengenai hal-hal yang menjadi kebutuhan dan pengguna sehingga kelak aplikasi yang dibuat memang akan digunakan oleh pengguna (*user*) serta akan memuaskan kebutuhan dan harapannya.

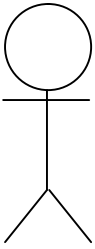
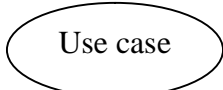
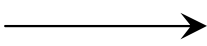
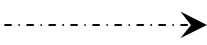
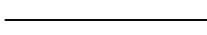

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *Use Case diagram* yang merupakan deskripsi peringkat tinggi bagaimana perangkat lunak atau aplikasi akan digunakan oleh penggunanya. Selanjutnya *Use Case diagram* tidak hanya sangat penting pada tahap analisis, tetapi juga sangat penting untuk perancangan (*design*), untuk mencari (mencoba menemukan) kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (*testing*).

*Use Case diagram* menggambarkan fungsi fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem [6].

### II.3.1.1 Notasi Diagram Use Case

*Use case* merupakan teknik berdasarkan skenario yang mendeskripsikan model system berorientasi objek, yang mengidentifikasi aktor yang terlibat dalam interaksi dan nama dari tipe interaksi tersebut.

**Tabel 1:** Notasi Diagram Use Case

Simbol	Nama	Keterangan
	<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> .
	<i>Use Case</i>	Aksi yang akan dilakukan <i>actor</i> .
	<i>Generalization</i> Dependensi	Generalisasi dari masing-masing <i>use case</i> .
	<i>Extends/Include</i> Dependensi	Hubungan antara dua <i>use case</i> .
	<i>Association</i>	Menghubungkan antara <i>actor</i> dengan <i>use case</i> .
	<i>System</i>	Lingkungan kerja dari perangkat lunak yang akan dibuat.

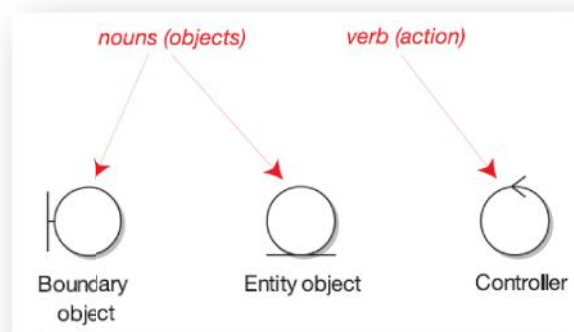
Cara penggunaan masing-masing dependensi:

1. *Generalization* digunakan ketika menggambarkan sebuah variasi pada sebuah *behaviour* dan ingin digambarkan secara sederhana (*casually*).
2. *Include* digunakan apabila terdapat repetisi pada dua atau lebih *use case* yang terpisah.
3. *Extend* digunakan saat menggambarkan variasi pada *behaviour* dan diharapkan ada bentuk-bentuk pengendalian yang lebih [5,6]

### II.3.2 Diagram Robustness

Menurut Rosenberg dan Stephens, diagram *Robustness* menjembatani tahap analisis dan desain. Jika tahap analisis (*Use Cases*) dianggap sebagai “apa” dan desain sebagai “bagaimana”, maka analisis *Robustness* merupakan desain sistem yang sangat awal. Pada tahap ini dibuat asumsi awal tentang desain [5].

Jadi, tahap ini dapat menjadi bagian dari tahap analisis dan dapat menjadi bagian dari tahap desain. Diagram *Robustness* merupakan representasi bergambar dari perilaku (*behavior*) yang dideskripsikan oleh *Use Cases*. Diagram *Robustness* menunjukkan perilaku dari kelas-kelas dan perilaku dari perangkat lunak. Pada diagram ini tidak digambarkan kelas mana yang bertanggung jawab terhadap perilaku tertentu. Walaupun demikian, diagram *Robustness* dapat dibaca seperti diagram aktivitas (*activity diagram*) atau sebagai sebuah *flowchart* dalam arti suatu objek “berbicara” dengan objek lainnya. Simbol-simbol yang terlibat dalam diagram *Robustness* ditunjukkan oleh gambar di bawah ini :



**Gambar II-1: Simbol-Simbol pada Diagram Robustness**

*Boundary object* adalah antarmuka antara sistem dengan segala sesuatu di luar sistem. Contohnya adalah layar atau halaman web [6].

*Entity object* merupakan kelas-kelas dari domain model. *Controller* adalah penghubung antara *boundary* dan *entity object*. Aturan-aturan dalam menggambar diagram *Robustness* adalah (Rosenberg & Stephens):

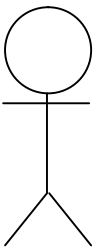
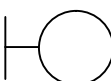

1. Objek (*noun*) dapat berbicara kepada *controller* (*verb*) dan sebaliknya.
2. Objek tidak dapat berbicara kepada objek lainnya.


*Controller* dapat berbicara kepada *controller* lainnya.

### II.3.2.1 Notasi Robustness Diagram

Ide dasarnya adalah bahwa untuk menganalisa langkah-langkah dari *use case* untuk memvalidasi logika bisnis di dalamnya dan untuk memastikan bahwa terminology yang konsisten dengan kasus penggunaan lainnya yang sebelumnya dianalisis. Penggunaan lainnya adalah untuk mengidentifikasi objek potensial atau tanggung jawab objek untuk mendukung logika yang disebut dalam *use case*, efektif sebagai jembatan untuk antar diagram lainnya.

**Tabel 2:**Notasi *Robustness Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> .
	<i>Boundary</i>	<i>Boundary</i> terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori ini.
	<i>Control</i>	<i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau

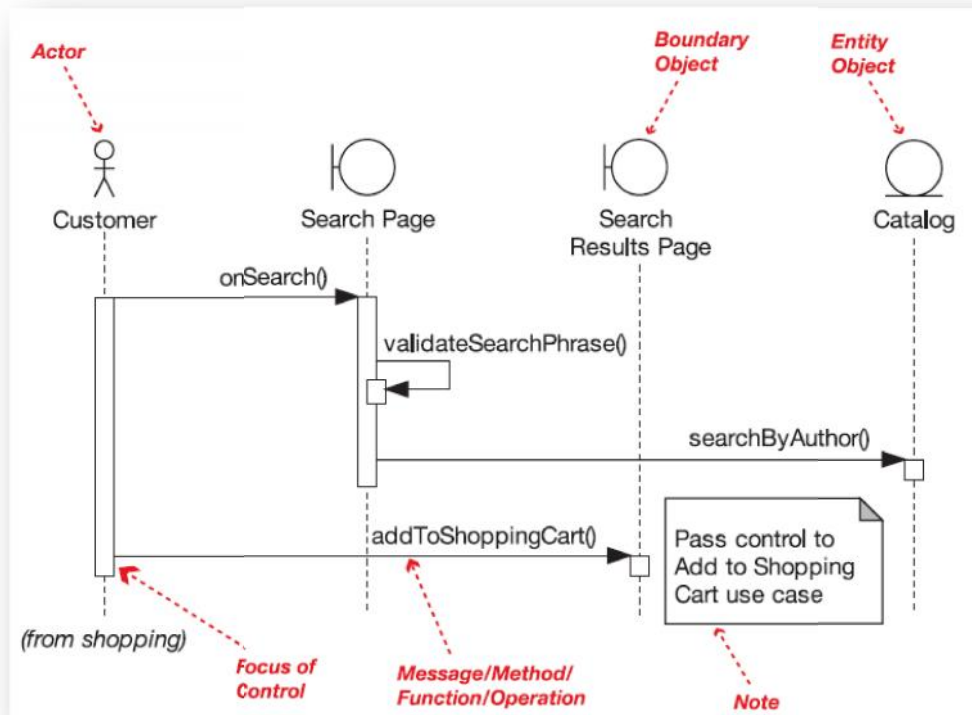
Simbol	Nama	Keterangan
		penanganan kesalahan.
	<i>Entity</i>	<i>Entity</i> digunakan menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data.

### II.3.3 Diagram Sequence

Diagram *Sequence* adalah sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah sistem. Interaksi tersebut berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri dari dimensi horizontal (objek-objek) dan dimensi vertikal (waktu) [7].

Karakteristik diagram *Sequence* :

- 1 Menggambarkan alur kejadian sebuah aktivitas
- 2 Lebih detail dalam menggambarkan aliran data, termasuk data atau *behaviour* yang dikirimkan atau diterima.
- 3 Kurang mampu menjelaskan detail dari sebuah algoritma (*loop, branching*).

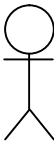


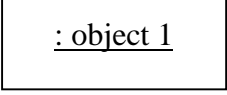



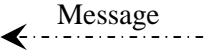

**Gambar II-2** Contoh Notasi Diagram Sequence


### II.3.3.1 Notasi Sequence Diagram

Diagram *Sequence* atau disebut juga diagram interaksi. Jelas bahwa diagram ini digunakan untuk menggambarkan beberapa jenis interaksi antara unsur-unsur yang berbeda dalam model. Jadi interaksi ini adalah bagian dari perilaku dinamis dari system [5].

**Tabel 3:** Notasi Sequence Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.

Simbol	Nama	Keterangan
	<i>Object</i> (Partisipan)	<i>Object</i> atau biasa juga disebut partisipan merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.
	<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> .
	<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang akan melakukan sebuah aksi.
	<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> .
	<i>Return Message</i>	<i>Return Message</i> , digambarkan dengan anak panah horizontal dan garis putus-putus antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi balik antara <i>object-object</i> .
	<i>Self-Message</i>	<i>Self-message</i> atau panggilan mandiri mengindikasikan komunikasi kembali ke dalam sebuah <i>object</i> itu sendiri.

Simbol	Nama	Keterangan
	<i>Loop</i>	Operator <i>loop</i> adalah <i>fragmen</i> yang dapat mengeksekusi berulang kali dan penjaga menunjukkan dasar iterasi.

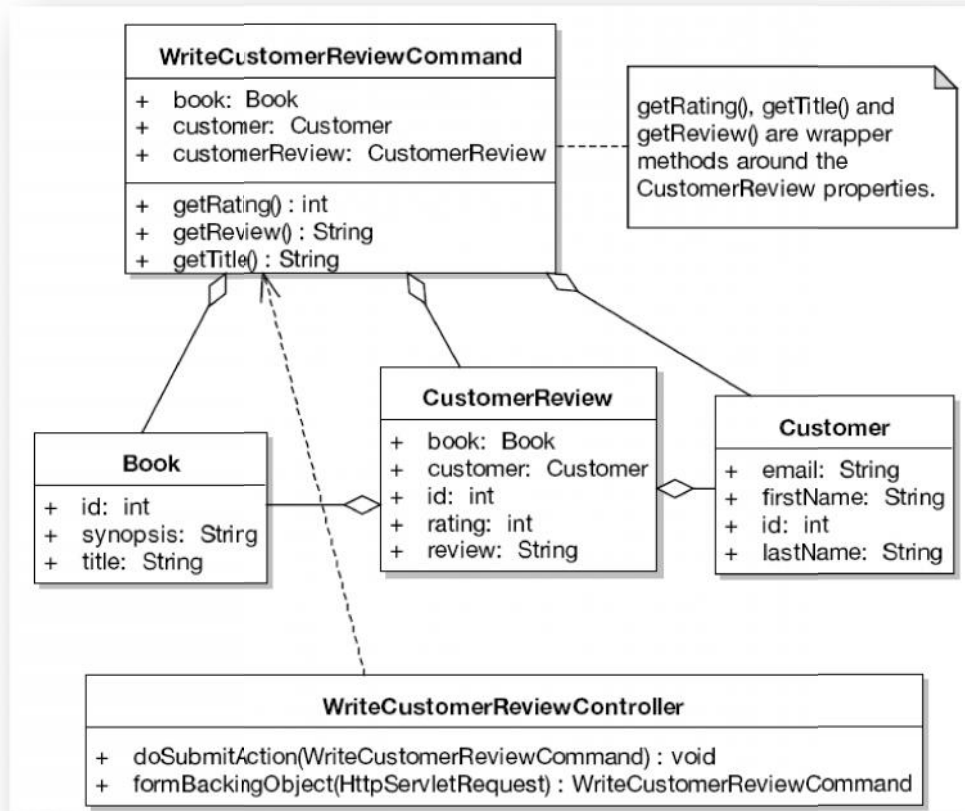
### II.3.4 Diagram Kelas Analisis

Diagram *class* sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

*Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain [6].

Komponen-komponen diagram kelas analisis antara lain :

1. Objek adalah abstraksi dari sebuah entitas nyata atau tidak nyata yang informasinya harus diingat atau disimpan.
2. *Class* adalah deskripsi lebih dari satu atau lebih objek dengan sejumlah atribut dan layanan yang sama termasuk deskripsi tentang cara membuat objek dari kelas tersebut.
3. Atribut adalah *variable* data, yang dapat memberikan informasi keadaan di mana tiap objek dari suatu kelas mempunyai nilai tersendiri.
4. Metoda adalah prosedur atau fungsi yang menjadi perilaku kelas dan objek dan menjadi tanggung jawab objek tersebut [5,6].



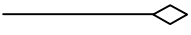
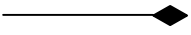
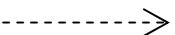
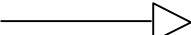
Gambar II-3 Contoh Diagram Kelas

### II.3.4.1 Notasi Diagram Kelas

Diagram Kelas merupakan sebuah statik diagram. Ini merepresentasikan tampilan yang tetap dari sebuah aplikasi. Diagram kelas tidak hanya untuk memvisualisasikan, menggambarkan dan mendokumentasikan berbagai aspek dari sistem, tetapi juga untuk membangun kode *executable* dari aplikasi perangkat lunak [5].

Tabel 4: Notasi Diagram Kelas

Simbol	Nama	Keterangan			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><b>ClassName</b></td> </tr> <tr> <td>-Attribute</td> </tr> <tr> <td>-Operation()</td> </tr> </table>	<b>ClassName</b>	-Attribute	-Operation()	<i>Class</i>	<i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi objek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut
<b>ClassName</b>					
-Attribute					
-Operation()					

Simbol	Nama	Keterangan
		<i>class</i> . Bagian akhir mendefinisikan method-method/operasi dari sebuah <i>class</i> .
	<i>The Aggregation (containing)</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.
	<i>The Aggregation (composition)</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.
	<i>The Aggregation (dependency)</i>	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.
<u>1..n Owned by 1</u>	<i>The Relationship (association)</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> (Contoh: One-to-one, one-to-many, many-to-many).
	<i>Generalization</i>	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi objek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “parent”-nya/induknya.

## **II.4 Adobe Flash**

Adobe flash yang sebelumnya disebut sebagai *Macromedia Flash*. Merupakan software animasi 2D yang dikembangkan Adobe System. Menurut Gaiy, Adobe Flash sangat populer dalam menciptakan berbagai iklan, game, media pembelajaran dan halaman web. Fitur-fitur khususnya adalah fleksibilitas dalam format *Output*, integrasi dengan *third party programs* lainnya dan gambar *vector* [8].

### **II.4.1 ActionScript**

Menurut Moock ActionScript adalah bahasa scripting yang digunakan dalam adobe flash. Hal ini digunakan untuk meningkatkan animasi dengan menggunakan logika pemrograman. Hal ini juga dapat digunakan untuk mengontrol animasi 2D vector menjadi sebuah animasi yang lebih kompleks. Ada tiga versi action script saat ini, yaitu ActionScript 1.0, ActionScript 2.0, dan ActionScript 3.0. ActionScript 1.0 adalah bahasa script yang pertamakali di implementasikan dalam adobe flash ActionScript 1.0 memiliki fungsi kecil dan kompatibilitas sedikit dan jarang digunakan. ActionScript 2.0 lebih baik dari pendahulunya karena dapat berfungsi dengan baik dan dapat digugat dalam berbagai cara untuk meningkatkan animasi.

Sedangkan ActionScript 3.0 adalah bahasa scripting terbaru yang dikembangkan untuk adobe flash. Action script 3.0 ini cocok untuk membuat aplikasi internet dan aplikasi smart phone yang handal serta aplikasi game berbasis 2D [9].

## **II.5 XML (eXtensible Markup Language)**

XML (eXtensible Markup Language) merupakan bahasa web turunan dari SGML (Standart Generalized Markup Language) yang ada sebelumnya. XML hampir sama dengan HTML, dimana keduanya sama-sama turunan dari SGML. Teknologi XML dikembangkan mulai tahun 1966 dan mendapat pengakuan dari

*World Wide Web Consortium (W3C)* pada February 1998. Sedangkan SGML itu sendiri telah dikembangkan pada awal tahun 1980-an. Pada saat HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting SGML dan dengan berpedoman pada pengembangan HTML menghasilkan bahasa markup yang tidak kalah hebatnya dengan SGML [10].

Keuntungan XML

1. Extensibilitas : dapat ditukar/ digabung dengan dokumen XML lain
2. Memungkinkan pemrograman yang lebih baik : maka dibuat suatu software pengolah xml.
3. Memisahkan data dan presentasi : yang akan direpresentasikan dalam XML dan XSLT.
4. Pencarian data cepat karena XML merupakan data dalam format terstruktur.
5. Plain text dan platform independent untuk pertukaran data.

## **II.6 Android**

### **II.6.1 Sejarah Singkat Sistem Operasi 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. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel (*smartphone*). Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorium dari 34 perusahaan peranti keras, piranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [10].

Pada saat perilis perdana Android (5 November 2007), Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *Open Source* pada perangkat *mobile*. Dilain pihak, Google merilis kode-kode Android di bawah

lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusi tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). Sekitar bulan September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasi. Telepon seluler ini diproduksi oleh HTC Cooration dan tersedia di pasaran pada 5 Januari 2008.

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Co dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana merek Android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6.

Sejak Android dirilis telah dilakukan berbagai pembaharuan berupa perbaikan *bug* dan penambahan fitur baru. Pada masa saat ini, kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis Android. Vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus, dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi Android. Banyak vendor yang memproduksi *smartphone* berbasis Android karena Android adalah sistem operasi yang *Open Source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android juga menjadi pesaing utama dari Apple pada sistem operasi Tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan di atas adalah karena Android merupakan *platform* yang sangat lengkap baik dari sistem operasinya, aplikasi dan

*tools* pengembangan, *market* aplikasi Android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga Android berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

## II.6.2 Bahasa Pemrograman Java di Android

Java merupakan perangkat lunak produksi *Sun Microsystem Inc.* untuk pemrograman beberapa tujuan (*multi puose*), *multi platform* dapat berjalan di beberapa sistem operasi, mudah dipelajari dan *powerfull*. Aplikasi-aplikasi yang dapat dibuat dengan java meliputi *web programming* (pemrograman web), *desktop programming* (pemrograman desktop), *mobile programming* (pemrograman *handphone*).

Perangkat lunak java sintaknya (tulisanannya) mirip dengan bahasa pemrograman C karena bahasa Java memakai bahasa pemrograman C, tetapi bahasa Java menyempurnakan kekurangan bahasa pemrograman C. Saat pertama kali rilis, Java disebut JDK (*Java Development Kit*), hingga JDK Versi 2 atau dikenal dengan Java 2, dibagi menjadi 3 edisi, yaitu J2SE (*Java 2 Standard Edition*), J2EE (*Java 2 Enterise Edition*) dan J2ME (*Java 2 Micro Edition*) [11].

J2SE merupakan edisi atau teknologi untuk pemrograman *desktop* atau aplikasi layar (*console*). J2SE juga merupakan perangkat lunak dasar yang harus dipasang (*install*) sebelum memakai J2EE dan J2ME. J2EE merupakan edisi atau teknologi untuk pemrograman *enterise*, seperti pemrograman *database*, JSP, Beans dan lain-lainnya. J2ME merupakan edisi atau teknologi untuk pemrograman *mobile/handphone* atau peralatan kecil (*small device*). Setelah Java teasang, maka di dalam sistem komputer terdapat JVM (*Java Virtual Machine*). Di dalam JVM tersebut terdapat JRE (*Java Runtime Environment*). Program yang diketik dengan bahasa Java memiliki ekstensi *.java*, akan menghasilkan *file.class* jika dikompilasi. *File.class* dapat dijalankan dengan memanfaatkan JRE.

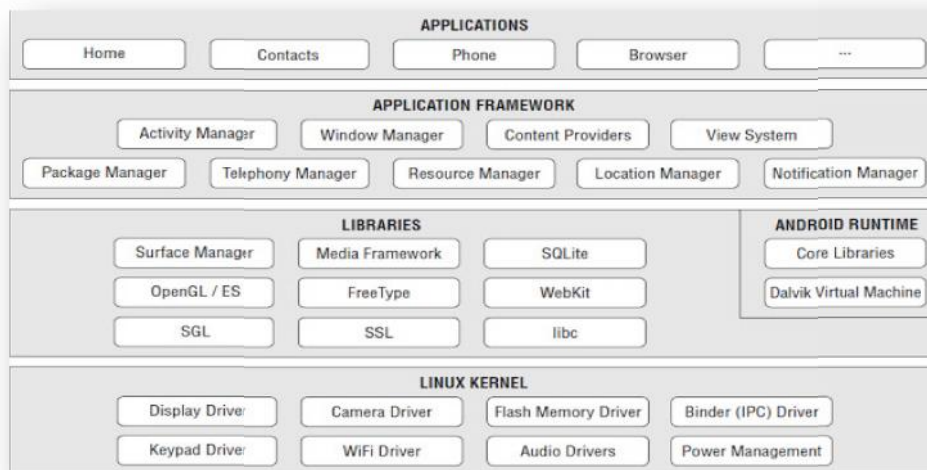
Android hanya menyediakan lingkungan *runtime* sebagai *intereter*, di mana kode sumber yang telah *dicompile* dengan *compiler* Java akan dioptimasi

oleh Dalvik, yaitu sebuah *virtual machine* yang memang dibuat khusus untuk menjalankan kode program yang dibuat dengan bahasa pemrograman Java. Kemudian oleh *dex tools* (merupakan bagian dari DVM) mengubah Java *Class* yang telah *compile* oleh *compiler* Java ke lingkungan *native* yang berbentuk *.dexformat* (Dalvik executable), yang teroptimasi untuk lingkungan perangkat keras dengan komputasi yang rendah.

### II.6.3 Arsitektur Android

Secara garis besar arsitektur Android terdiri atas *Applications*, *Applications Frameworks*, *Libraries*, *Android Run Time*, dan *Linux Kernel*.

1. *Applications* merupakan *layer* (lapis), di mana *user* berhubungan dengan aplikasi saja [11].



**Gambar II-4 Arsitektur Android**

2. *Applications Frameworks* merupakan *Open Development Platform* yang ditawarkan Android untuk dapat dikembangkan guna membangun aplikasi. Pengembang memiliki akses penuh menuju *API Frameworks* seperti yang dilakukan oleh aplikasi kategori inti. Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, dan *Activity Manager*.

3. *Libraries* merupakan *layer*, di mana fitur-fitur Android berada.
4. *Android Run Time* merupakan *layer* yang membuat aplikasi Android dapat dijalankan, di mana dalam prosesnya menggunakan implementasi Linux.
5. *Linux Kernel* merupakan *layer* inti dari sistem operasi Android berada.

#### **II.6.4 Struktur Aplikasi Android**

Struktur aplikasi Android atau Fundamental Aplikasi ditulis dalam bahasa pemrograman Java. Kode Java ini dikompilasi bersama dengan *fileresource* yang dibutuhkan oleh aplikasi, di mana prosesnya *dipackage* oleh *tools* yang disebut “*apt-tools*” ke dalam paket Android, sehingga menghasilkan *file* dengan ekstensi *.apk*. *File .apk* ini yang disebut dengan aplikasi, dan nantinya dapat dijalankan pada *device/peralatan mobile* [11,12]

Ada 4 komponen pada aplikasi Android, antara lain :

1. *Activities* merupakan komponen untuk menyajikan *user Tampilan* (tampilan program) kepada pengguna.
2. *Service* merupakan komponen yang tidak memiliki *user Tampilan* (tampilan program), tetapi *service* berjalan secara *backgrounds*.
3. *Broadcast Receiver* merupakan komponen yang berfungsi menerima dan beraksi untuk menyimpan notifikasi.
4. *Contact Provider* merupakan komponen untuk membuat aplikasi data secara spesifik sehingga dapat digunakan oleh aplikasi lain.

#### **II.6.5 Versi Android**

Banyak *smartphone* dan PC Tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versinya, fitur yang digunakan pun semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream yang dirilis pada tanggal 22 Oktober 2008. Beberapa uraian versi Android antara lain [11,12] :

1. **Android Versi 1.1** pertama kali dirilis oleh Google pada tanggal 9 Maret 2009. Versi ini memiliki fitur-fitur: jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail dan pemberitahuan *e-mail*.
2. **Android Versi 1.5** (Cupcake) merupakan versi yang dirilis oleh Google pada pertengahan Mei 2009. Fitur-fitur yang dimiliki versi ini adalah merekam dan menonton video dengan modus kamera, *upload* video ke Youtube dan gambar Picasa langsung dari telepon, *bluetooth* A2DP, terhubung langsung ke headset *bluetooth*, animasi layar, *keyboard* pada layar yang disesuaikan sistem.
3. **Android Versi 1.6** (Donut) merupakan versi yang dirilis pada September 2009 dengan fitur-fitur: proses pencarian yang lebih baik dibandingkan versi sebelumnya, penggunaan baterai indikator dan *control applet* VPN, galeri memilih foto yang dihapus, kamera, *camcorder*, CDMA/EVDo, 802.1x, Gestures, *text-to-speech engine*, dan *dial contact*.
4. **Android Versi 2.0/2.1** (Eclair) merupakan versi yang dirilis Google pada tanggal 3 Desember 2009, memiliki fitur-fitur : mengoptimalkan *hardware* versi sebelumnya, Google Maps 3.1.2, perubahan *user Tampilan*, *browser* baru, HTML5, daftar kontak baru, *flash* untuk kamera 3,2MP, *digital zoom*, *bluetooth* 2.1.
5. **Android Versi 2.2** (Froyo : Frozen Yoghurt) merupakan versi yang dirilis oleh Google pada Mei 2010. Versi pertama (Rev 1) memiliki fitur-fitur : dapat menghapus komponen, DVM dioptimalkan, *graphic* 2D dan 3D, *SQLite*, Media Audio dan Video berbagai *format*, GSM, *bluetooth*, EDGE, 3G, WiFi, *camera*, GPS (*Global Positioning System*), kompas, accelerometer (tergantung *hardware*).
6. **Android Versi 2.3** (Gingerbread) merupakan versi yang dirilis oleh Google pada Desember 2010, memiliki fitur-fitur : *Sip-based VoIP*, NFC (*Near Field Communications*), *Gyroscope* dan Sensor, *Multiple Cameras Supprot*, *Mixable Audio Effects* dan *Download Manager*.

7. **Android Versi 3.0/Versi 3.1** (HoneyComb) merupakan versi yang dirilis oleh Google khusus untuk PC Tablet. Beberapa *smartphone* tidak dapat menggunakan karena versi ini memang banyak dipakai bukan untuk *voice call*. Memiliki fitur-fitur : aksesoris API terbuka, USB *host* API, *keyboard external* dan perangkat penunjuk, *joystick* dan *gamepads*, WiFi, perluasan daftar terbaru Apps, layar *home widgetresizeable*.
8. **Android Versi 4.0** (ICS: Ice Cream Sandwich) diumumkan pada tanggal 19 Oktober 2011. Android versi 4.0 ini membawa fitur Honeycomb untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, teadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline* dan berbagai informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.
9. **Android Versi 4.1 – 4.2** (Jelly Bean) diluncurkan pada acara Google I/O yang membawa sejumlah keunggulan dan fitur baru. Penambahan baru tersebut diantaranya meningkatkan *input keyboard*, *desain* baru fitur pencarian, *user* Tampilan yang baru dan pencarian melalui *voice search* yang lebih cepat. Selain itu terdapat pembaharuan pada *Google Now*. *Google Now* memberikan informasi yang tepat pada waktu yang tepat juga. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yaitu Google Nexus 7.

## II.6.6 Pengembangan Android

Android adalah sistem operasi *mobile* berbasis Linux. Awalnya Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel pintar/*smartphone*. Beberapa hal yang harus kita kerjakan sebelum memulai untuk membuat aplikasi Android sebagai berikut:

1. Java SE

Install Java SE versi 1.5 atau 1.6 ke atas untuk dapat mengkompilasi Android.

2. Andoird SDK

Android SDK adalah tools API (*Application Programing Tampilans*) yang di perlukan untuk mengembangkan aplikasi pada *platform* Android.

## II.7 Gambaran Umun Pada Aplikasi Sejenis

Secara garis besar aplikasi Tes Potensi Akademik bertujuan untuk mengetahui bakat dan kemampuan seseorang dibidang keilmuan atau akademis. Karenanya TPA ini sering dihubungkan dengan kecerdasan seseorang, dengan begitu telah bayak pengembang aplikasi sejenis, misalnya di Politeknik Negeri Batam, sudah banyak Mahasiswa yang membahas TugasAkhirnya tentang simulasi tes, baik itu tentang system ujian online, simulasi ujian SPMB dan lain sebagainya. Maka penulis membuat gambaran umum pada pengembangan sebelumnya [13].

Deskripsi umum pada Aplikasi Tes Online yang dibuat oleh Muhammad Tarmizi (2005) terdiri dari:

1. Aplikasi mampu menampilkan soal tes secara acak untuk tiap *user*.
2. Aplikasi mampu menyimpan hasil tes peserta.
3. Aplikasi mampu menghitung nilai tes peserta.

Dari deskripsi umum Aplikasi Tes Online yang dibuat oleh Muhammad Tarmizimasih memiliki kekurangan, seperti:

1. Pemilihan level pada soal yang masih statis.
2. Pengaturan soal ujian yang belum teratur menurut level.
3. Penyajian nilai ujian yang masih statis.

Deskripsi umum pada Aplikasi Tes UMPB Berbasis Komputer yang dibuat oleh Karmila (2008) terdiri dari:

1. Aplikasi dapat menampilkan soal-soal ujian secara acak.
2. Aplikasi dapat menentukan jumlah soal yang akan dikerjakan oleh peserta.
3. Aplikasi dapat menghitung nilai dari hasil tes UMPB secara cepat.
4. Aplikasi dapat melihat peserta-peserta yang diterima dan ditolak pada program studi tertentu berdasarkan hasil tes.
5. Aplikasi dapat menampilkan report/laporan hasil tes UMPB.
6. Aplikasi dapat melakukan pengelolaan data soal yang terdiri dari insert, update dan delete soal.
7. Aplikasi dapat melakukan penambahan mata pelajaran.
8. Aplikasi mampu mengelola data soal bertipe suara dan gambar.
9. Aplikasi dapat melakukan pengelolaan data peserta yang terdiri dari insert, update dan delete peserta ujian.
10. Aplikasi dapat melakukan pencarian *user* berdasarkan nama, *id\_user*, dan program studi.

Dari deskripsi umum Aplikasi Tes UMPB Berbasis Komputer yang dibuat oleh Karmila masih memiliki kekurangan, seperti:

1. Tidak memiliki bank soal.
2. Penyediaan level masih statis.
3. Pengaturan nilai ujian yang masih statis.

Selain itu deskripsi umum aplikasi sejenis yang ada pada Google Play Store yang di design oleh Androlite dengan judul Tes Potensi Akademik memiliki karakteristik sebagai berikut:

1. Aplikasi menyediakan berbagai macam jenis soal, seperti soal persamaan kata, lawan kata, aritmatika, gambar dan lain sebagainya
2. Aplikasi memberikan jawaban secara statis

Dari deskripsi aplikasi Tes Potensi Akademik yang di design oleh Androlite diatas, masih terdapat beberapa kekurangan, seperti:

1. Aplikasi hanya menyajikan soal dan jawaban secara statis.
2. Pada aplikasi tidak ada interaksi yang dilakukan oleh pengguna
3. Aplikasi tidak memiliki fitur untuk menjawab soal

## **II.8 Rumus Cara Penilaian dan Cara Pengacuannya**

### **II.8.1 Pengolahan Nilai**

Proses penilaian adalah suatu proses membandingkan skor yang diperoleh tiap pengguna aplikasi dengan acuan yang dipakai penilaian aturan patokan atau penilaian aturan normal (PAP atau PAN), yang hasilnya berbentuk nilai dengan skala 0 – 100.

### **II.8.2 Skala Penilaian (Skala 1-100)**

Dengan menggunakan skala 1-10 maka bilangan bulat yang ada masih menunjukkan penilaian yang agak kasar. Ada sebenarnya hasil prestasi yang berada diantara kedua angka bulat itu [14]. Untuk itu dengan menggunakan skala 1-100, dimungkinkan melakukan penilaian yang lebih halus karena terdapat 100 bilangan bulat. Nilai 5,5 dan 6,4 dalam skala 1-10 yang biasanya dibulatkan menjadi 6, dalam skala 1-100 ini boleh dituliskan dengan 55 dan 64.

$$N = \frac{\sum N \quad s_i \quad y \quad b}{\sum N \quad T \quad S} \times 100$$

**Persamaan 1**

Misalnya si A mendapat mengerjakan soal dengan benar sebanyak 100 soal dari 150 soal yang ada, maka nilai yang dia dapat adalah:

$$N = \frac{100}{150} \times 100$$

$$N = 66$$

## **II.9 Klasifikasi Perhitungan IQ (Intellegent Quotient)**

### **II.9.1 IQ (Intellegent Quotient)**

IQ (Intellegent Quotient) merupakan angka yang menjelaskan tingkat kecerdasan seseorang yang dibandingkan dengan sesamanya dalam satu populasi. Untuk mengukur IQ seseorang dapat dilakukan dengan dua cara, yaitu Tes Verbal (Pengetahuan Umum) dan Non-Verbal (pengorganisasian gambar dan angka secara abstrak).

Klasifikasi IQ berbeda untuk setiap metode test yang digunakan. Stanford-Binet mengklasifikasikan nilai IQ normal yang berkisar diantara 85-115. Lewis Terman mengklasifikasikan nilai IQ normal pada kisaran 90-109. Lebih jauh lagi, Wechsler mengklasifikasikan IQ normal pada angka 100 dengan nilai toleransi 15 (berarti 85-115). Dikarenakan perbedaan ini, maka selain nilai IQ yang didapat, harus diperhatikan pula metode test apa yang dipergunakan [15].

Untuk klasifikasi umum, saat kita tidak mengetahui metode apa yang digunakan, bisa menggunakan klasifikasi dibawah ini (hasil kompromi ketiga metode diatas).

70-79	: tingkat IQ rendah atau keterbelakangan mental
80-90	:tingkat IQ rendah yang masih dalam kategori normal (dull normal)
91-110	: tingkat IQ normal atau rata-rata
111-120	: tingkat IQ tinggi dalam kategori normal (bright normal)
120-130	: tingkat IQ superior
131-140 atau lebih	: tingkat IQ sangat superior atau jenius

### **II.9.2 Ukuran Tingkat IQ Manusia**

#### **1. Idiot (0-29) [16]**

Manusia dengan tingkat IQ idiot merupakan individu terbelakang paling rendah. Tidak dapat berbicara atau hanya dapat mengucapkan beberapa kata saja, biasanya tidak dapat mengurus dirinya sendiri seperti mandi, berpakaian, makan dan sebagainya.

2. Imbecile (30-40)

Kemampuan kecerdasannya sama dengan anak- anak umur 3-7 tahun. Bisa mandiri dan berkata-kata tetapi dengan pengawasan dan perlu sekolah khusus karena tetap bergantung pada orang lain.

3. Moron atau Debil IQ / Mentaly Retarted(50-69)

Bisa membaca, menulis dan perhitungan sederhana, bisa diberikan tugas tanpa perencanaan dan pemecahan. Biasanya mereka sekolah di sekolah luar biasa.

4. Kelompok bodoh IQ dull / Bordeline (70-79)

Kelompok ini dengan susah payah untuk menyelesaikan pekerjaannya. Mereka biasanya bisa sampai SMP tetapi dengan susah payah menyelesaikan pelajaran kelas 3, kelompok ini diatas terbelakang dan dibawah kelompok normal.

5. Normal Rendah / Below Average (80-89)

Kelompok ini termasuk normal sedang paling bawah. Mereka biasanya bisa lulus SMA tetapi agak kesulitan di SMA.

6. Normal Sedang (90-109)

Kelompok ini merupakan kelompok normal rata-rata atau sedang. Kelompok ini paling besar dalam populasi penduduk.

7. Normal Tinggi / above average (110-119)

Kelompok ini merupakan kelompok individu yang normal tetapi berada pada tingkat yang tinggi.

8. Cerdas / Superior (120-129)

Kelompok ini berhasil pada prestasi akademik

9. Sangat Cerdas / very superior(gifted) (130-139)

Kelompok ini lebih cakap dalam membaca dan mempunyai pengetahuan sangat baik tentang bilangan. Perbendaharaan kata yang banyak dan memahami hal yang abstrak. Factor ketangkasan, kekuatan, dan kesehatan lebih menonjol daripada anak normal.

10. Genius (>140)

Kelompok ini sangat luar biasa mereka mempunyai kemampuan untuk menyelesaikan masalah dan menemukan sesuatu yang baru meskipun tidak

bersekolah. Mereka berada disemua ras dan disemua tingkatan baik ekonomi maupun jenis kelamin. Contohnya adalah Einstein dan Thomas Alva Edison(160).

### II.9.3 Cara Perhitungan Nilai Pada Aplikasi M-TPA

Rumus untuk menghitung nilai yang didapat oleh pengguna aplikasi yaitu dengan cara sebagai berikut:

Rumus :

$$X = \text{Jumlah Point} - \text{Minpoint}$$

$$Y = X/\text{Selisih}$$

$$Z = Y * \text{Toleran}$$

$$\text{Maka, Hasil} = \text{min IQ} + Z$$

**Tabel 5:** Kategori Penilaian

kategori	Point	selisih	iq range	Toleran
1	20	0	140	0
2	16-19	3	130-139	9
3	10-15	5	115-129	14
4	6-9	3	102-114	12
5	3-5	2	90-101	11
6	0-2	2	70-89	19

contoh:

jika jumlah soal yang benar = 7, maka masuk kategori 4

$$7 - 6 = 1$$

$$1/3 = 0.3333$$

$$0.33333 * 12 = 3,9 = 4$$

$$102 + 4 = 106$$

jadi total iq yang di dapat = 106

## II.9.4 Deskripsi IQ Score

**Tabel 6:** Deskripsi IQ Score [17]

Points	Level	IQ Range	Description
20	Genius	140+	you are probably a genius
16-19	Gifted	130-139	you have a unique imagination and logical mind
10-15	Above Average	115-129	you have a good imagination and logical mind
6-9	Higher Average	102-114	you have an average imagination and logical mind
3-5	Lower Average	90-101	you have an average imagination and logical mind with dull normal
0-2	Below Average	70-89	you have a below average imagination and logical mind

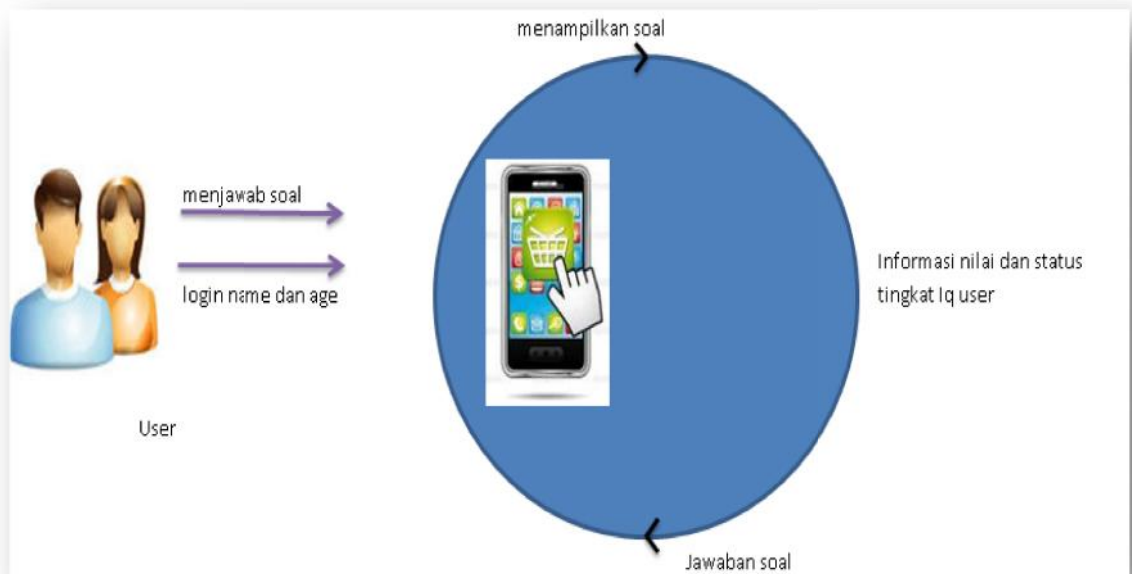
## BAB III

### ANALISIS DAN PERANCANGAN

Pada tugas akhir ini akan dirancang sebuah aplikasi Tes IQ berbasis android, dimana tes IQ ini diperuntukkan untuk setiap orang maupun untuk umum. Aplikasi ini menyajikan soal yang umum guna mengetahui tingkat potensi akademik oleh si pengguna.

#### III.1 Analisis Aplikasi Tes IQ

##### III.1.1 Deskripsi Umum Sistem



**Gambar III-1 Deskripsi Umum Sistem**

Keterangan:

Input 1 : User harus melakukan login terlebih dahulu dengan menggunakan name dan age (tampa menggunakan password)

Input 2 : Setelah user login, secara otomatis sistem akan menampilkan soal yang tersedia untuk dijawab oleh user.

Input 3 : User menjawab soal-soal yang tersedia, jawaban dari user akan diproses oleh sistem dari aplikasi tes IQ.

Input 4 : Sistem memberikan informasi nilai yang di dapat oleh user setelah mengerjakan soal, dan nilai yang didapat diakumulasikan oleh sistem menjadi nilai berupa tingkat IQ user.

### III.1.2 Analisis Karakteristik Pengguna

Karakteristik dari pengguna aplikasi Tes IQ ini terbagi atas 2 bagian, seperti yang terlihat pada tabel 5, di bawah ini:

**Tabel 7:**Karakteristik Pengguna

Kategori pengguna	Tugas	Hak akse pada aplikasi
User	<ul style="list-style-type: none"><li>- Mengerjakan soal-soal yang tersedia, setelah itu mengumpulkannya (submit)</li><li>- Dapat mereview hasil jawaban yang telah di kumpulkan (submit).</li></ul>	<ul style="list-style-type: none"><li>- Dapat melihat soal dan dapat mengerjakan serta mengumpulkannya.</li><li>- Dapat melihat jawaban yang benar dari system dan jawaban yang salah dari user.</li></ul>
Sistem	<ul style="list-style-type: none"><li>- Mengoreksi hasil jawaban dari user yang telah di submit.</li><li>- Memberikan hasil jawaban yang benar dari soal yang dikerjakan.</li></ul>	Menampilkan nilai tingkat IQ dan result dari user.

### III.1.3 Analisis Kebutuhan Sistem pada Computer dan Mobile

Dalam pembuatan tugas akhir ini memiliki spesifikasi kebutuhan perangkat lunak yang dibutuhkan untuk membangun aplikasi ini sebagai berikut:

**Tabel 8:** Analisis Kebutuhan Perangkat Lunak pada Computer dan Mobile

Perangkat	Hardware	Software
Computer	<ul style="list-style-type: none"><li>- Processor minimal intel core 2 Duo atau setara.</li><li>- Kapasitas Memori 2 GB</li><li>- Hardisk minimal 80 GB</li></ul>	<ul style="list-style-type: none"><li>- Operating system windows 7.</li><li>- Adobe Flash CS 6</li><li>- Emulator android SDK</li><li>- Plugin ADT 16.1.0</li></ul>
Mobile	<ul style="list-style-type: none"><li>- Memori eksternal minimal 2 GB</li><li>- Kabel data USB</li><li>- Processor minimal ARMv6</li></ul>	<ul style="list-style-type: none"><li>- System operating system android minimal Gingerbread 2.3</li></ul>

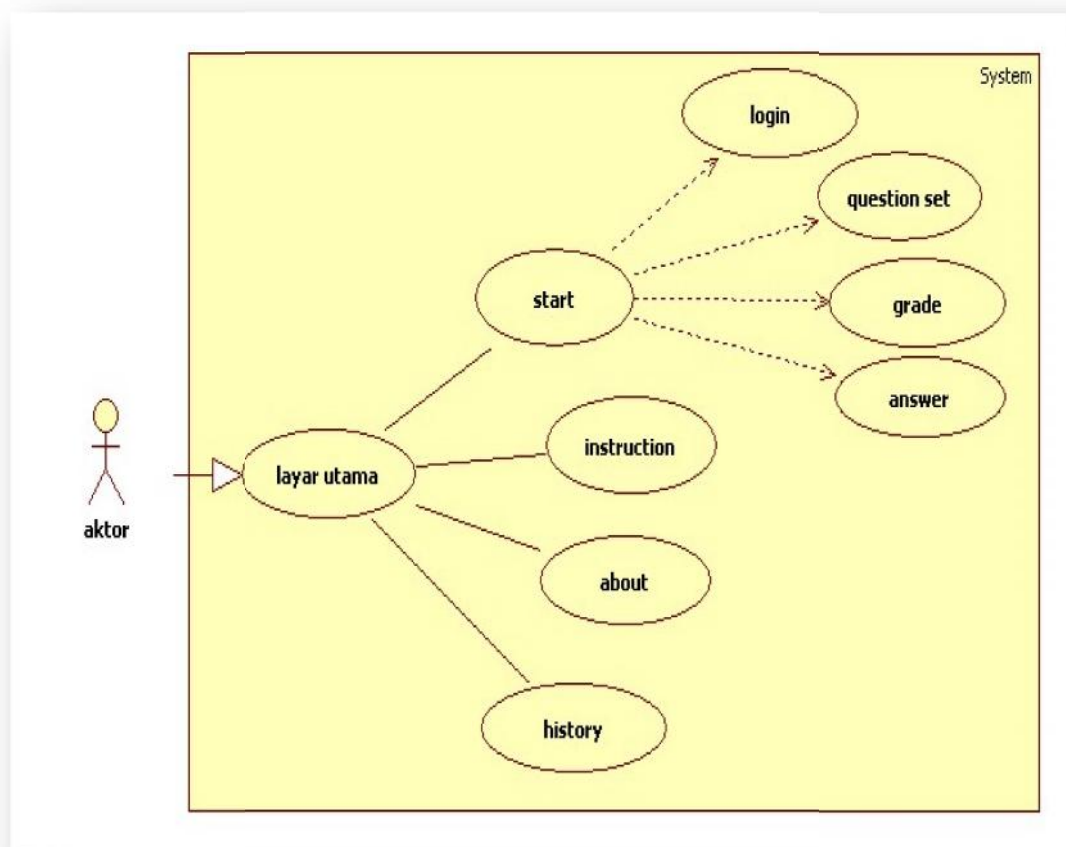
### III.1.4 Analisis Kebutuhan Fungsional

1. Pembuatan login dengan memasukkan nama dan umur untuk autentifikasi data dari pengguna agar dapat mengakses ke aplikasi.
2. Aplikasi mampu memilih, mengubah jawaban yang akan dimasukkan.
3. Aplikasi akan menampilkan hasil submit jawaban dari user dengan menampilkan jawaban yang benar dan jawaban yang salah.
4. Aplikasi mampu menampilkan result nilai yang didapat oleh user
5. Aplikasi mampu menentukan tingkat IQ dari user setelah mendapat nilai, dan akan ditampilkan pada result nilai oleh pengguna.
6. Nilai yang akan diberikan berupa percentage yang akan ditampilkan bersamaan dengan nama dan umur dari hasil login.
7. Aplikasi akan menampilkan soal berupa bahasa inggris.

## III.2 Perancangan Aplikasi Tes IQ

### III.2.1 Diagram Use Case Tes IQ

*Use case* diagram aplikasi ini digunakan untuk menggambarkan hubungan sejumlah eksternal aktor dengan *use case* yang terdapat dalam aplikasi. *Use case* diagram ini hanya menggambarkan keadaan lingkungan aplikasi yang dapat dilihat dari luar oleh aktor.



Gambar III-2 Diagram Use Case Tes IQ.

## **III.2.2 Skenario Use Case Tes IQ**

### **III.2.2.1 Skenario Usecase Menampilkan Layar Utama**

aktor : user

kondisi awal : user install aplikasi

scenario :user menginstal aplikasi tes IQ dan menjalankannya pada android yang dimiliki oleh user, maka akan muncul layar utama berupa nama aplikasi.

Kondisi akhir : layar utama aplikasi (judul) ditampilkan.

### **III.2.2.2 Skenario Usecase Menu Start**

Actor : user

Kondisi awal : klik atau pilih menu start.

Scenario :setelah masuk pada layar utama maka akan muncul layar selanjutnya yaitu beberapa menu yang tersedia pada aplikasi salah satunya menu start. Pilih menu start untuk mulai proses login.

Kondisi akhir : menu start akan tampil dengan menampilkan menu login untuk memulai memainkan aplikasi.

### **III.2.2.3 Skenario Usecase Login**

Actor : user

Kondisi awal : menu login tampil dengan isi tampilan name, age dan submit berupa button.

Scenario : user login dengan memasukkan nama dan umur dari si user, dan setelah dimasukkan maka harus di submit untuk melanjutkan pada proses selanjutnya.

Kondisi akhir : user telah login setelah di submit.

#### **III.2.2.4 Skenario Usecase Question Set**

Actor : user

Kondisi awal :setelah proses login berhasil maka layar question set akan ditampilkan.

Scenario : - user akan memilih question set yang ingin dikerjakan terlebih dahulu, dengan jumlah 1 question set sebanyak 20 soal.

: -setelah dipilih maka akan muncul sebuah notice berisi pertanyaan apakah yakin untuk memulai menjawab pertanyaan.

: - user memulai proses menjawab soal.

Kondisi akhir :user menjawab soal-soal dari question set yang telah di pilih selama 20 menit.

#### **III.2.2.5 Skenario Usecase Grade**

Actor : user

Kondisi awal : user menjawab soal-soal yang diberikan.

Scenario : - user memilih jawaban dari soal-soal yang disediakan dengan limit waktu sebanyak 20 menit untuk menjawab sebanyak 20 soal.

: - setelah selesai mengerjakan maka hasil jawaban di submit, agar system pada aplikasi mulai untuk mengoreksi hasil jawaban.

Kondisi akhir :akan muncul layar hasil jawaban dari si user yaitu percentage dan tingkat IQ dari si user.

#### **III.2.2.6 Skenario Usecase Answer**

Actor : user

Kondisi awal : memilih tombol answer pada layar grade

Scenario : user dapat mereview hasil jawaban yang telah di kerjakan, dan dapat melihat jawaban yang benar dan salah dari soal tersebut.

Kondisi akhir : dapat melihat review hasil jawaban.

### **III.2.2.7 Skenario Usecase Menu Instruction**

Actor : user

Kondisi awal : user menekan tombol menu instruction

Scenario : sistem menampilkan instruction cara menggunakan aplikasi, supaya user lebih paham pada pilihan menu-menu aplikasi.

Kondisi akhir : layar menu instruction tampil.

### **III.2.2.7 Skenario Usecase Menu About**

Actor : user

Kondisi awal : user memilih/menekan tombol menu about.

Scenario : sistem menampilkan dan menjelaskan pengertian dari aplikasi Tes IQ tersebut.

Kondisi akhir : layar menu about tampil.

### **III.2.2.8 Skenario Usecase Menu History**

Actor : user

Kondisi awal : user menekan tombol menu history

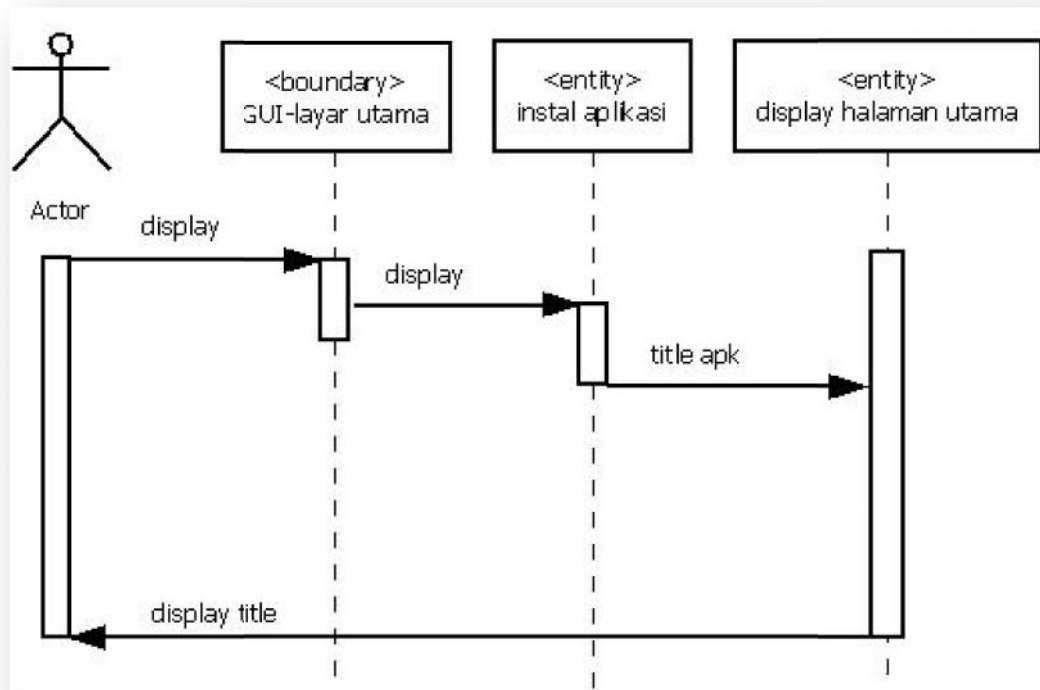
Scenario : menu history akan menampilkan data nama, umur dari user beserta result dari nilai yang didapat.

Kondisi akhir : result nilai di tampilkan.



## III.2.4 Diagram Sequence

### III.2.4.1 Diagram Sequence GUI-Layar Utama

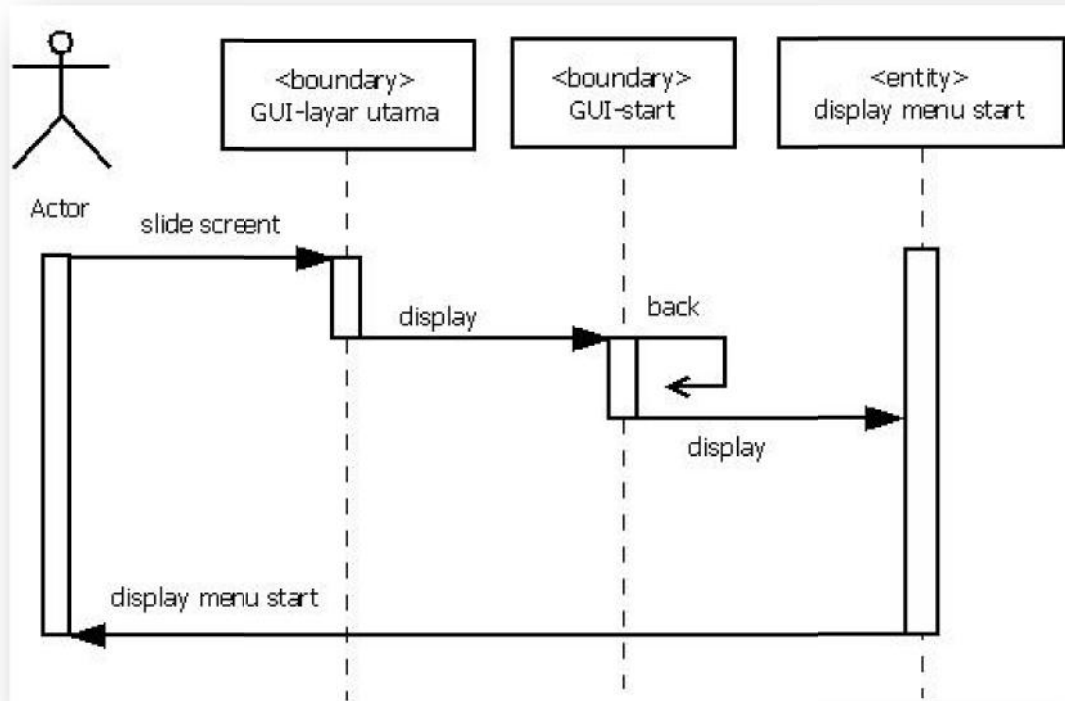


**Gambar III-4 Sequence GUI-Layar Utama.**

Penjelasan proses menu layar utama

- User terlebih dahulu membuka aplikasi pada android dari aplikasi tes potensi akademik
- Setelah success di buka maka aplikasi akan menampilkan layar utama
- Display yang di tampilkan yaitu nama dari aplikasi.

### III.2.4.2 Diagram Sequence GUI-Start

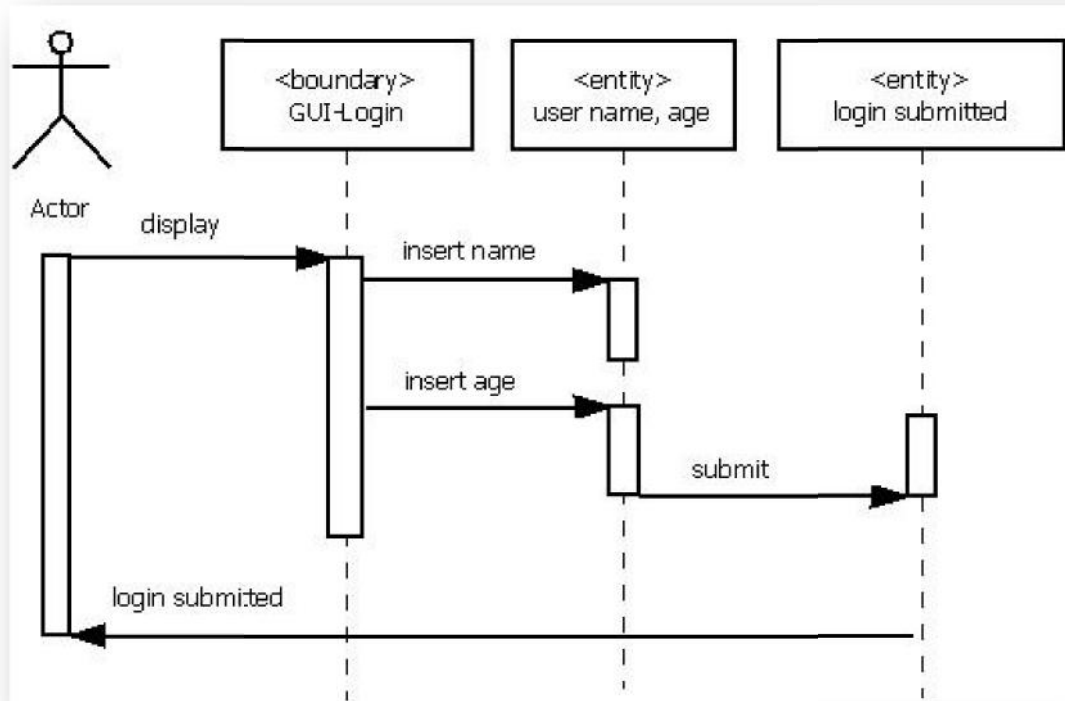


**Gambar III-5 Sequence GUI-Start.**

Penjelasan proses menu start

- Sebelumnya sistem telah menampilkan halaman layar utama
- Maka dilanjutkan pada halaman home page
- Sistem menampilkan beberapa menu pada home page yaitu menu start, instruction, about, dan history.
- Choose menu start
- Sistem menampilkan menu start yaitu login untuk proses selanjutnya.

### III.2.4.3 Diagram Sequence GUI-Login

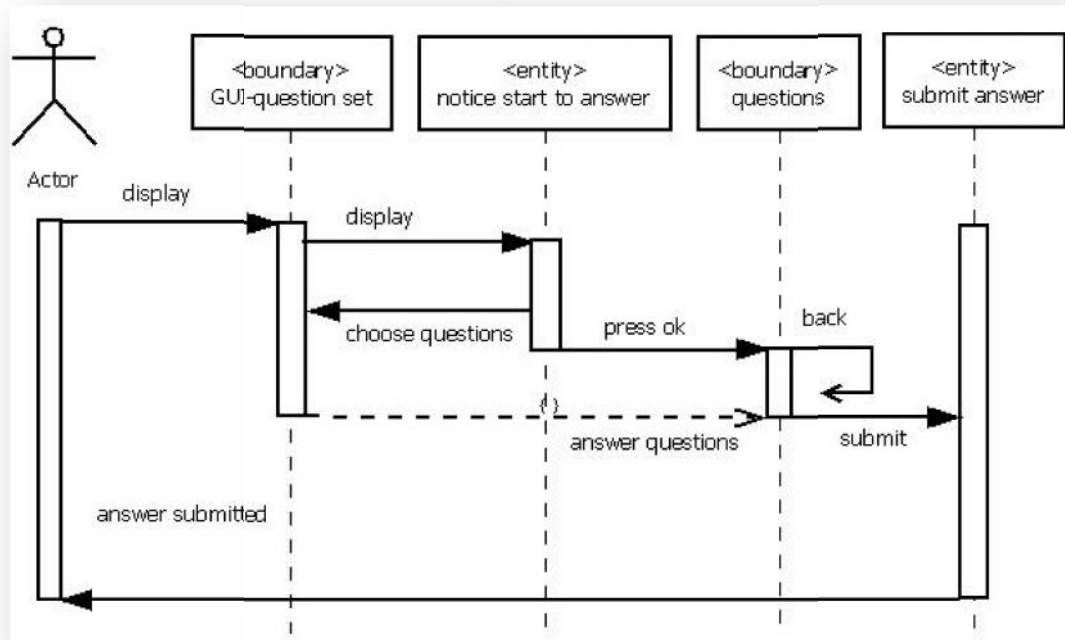


**Gambar III-6 Sequence GUI-Login.**

Penjelasan proses login

- System menampilkan layar menu login
- Layar login terdapat kotak masukan untuk nama dan umur dari user
- User insert name dan age kemudian klik tombol submit untuk masuk pada aplikasi
- System akan memvalidasi name dan age dari hasil insert user
- Jika user valid maka system akan menampilkan ulang insert name dan age
- Jika login telah submitted, system akan menuju layar berikutnya.

### III.2.4.4 Diagram Sequence GUI-Question Set

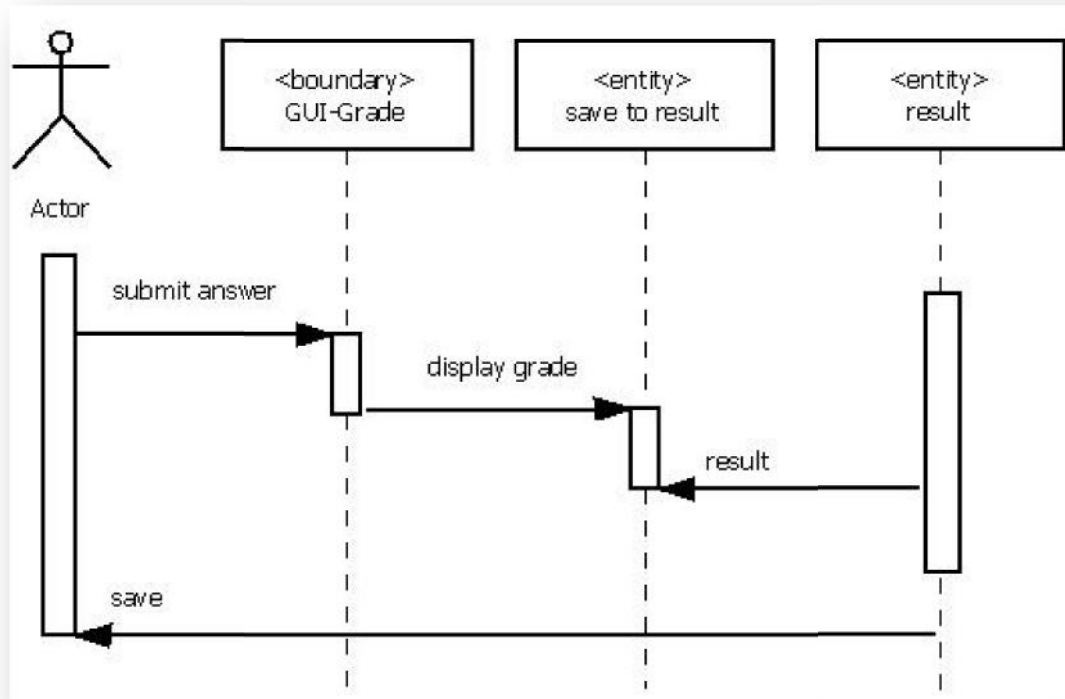


Gambar III-7 Sequence GUI-Question Set.

Penjelasan proses question set

- System menampilkan layar question set
- Untuk memulai aplikasi, pilih question set yang ingin di kerjakan
- System akan menampilkan notice start to answer, maka press ok
- System menampilkan Question, maka user dapat memulai untuk menjawab questions
- Pada saat proses answer questions, user dapat back, review, dan next question
- Submit answer, maka system akan submitted.

### III.2.4.5 Diagram Sequence GUI-Grade

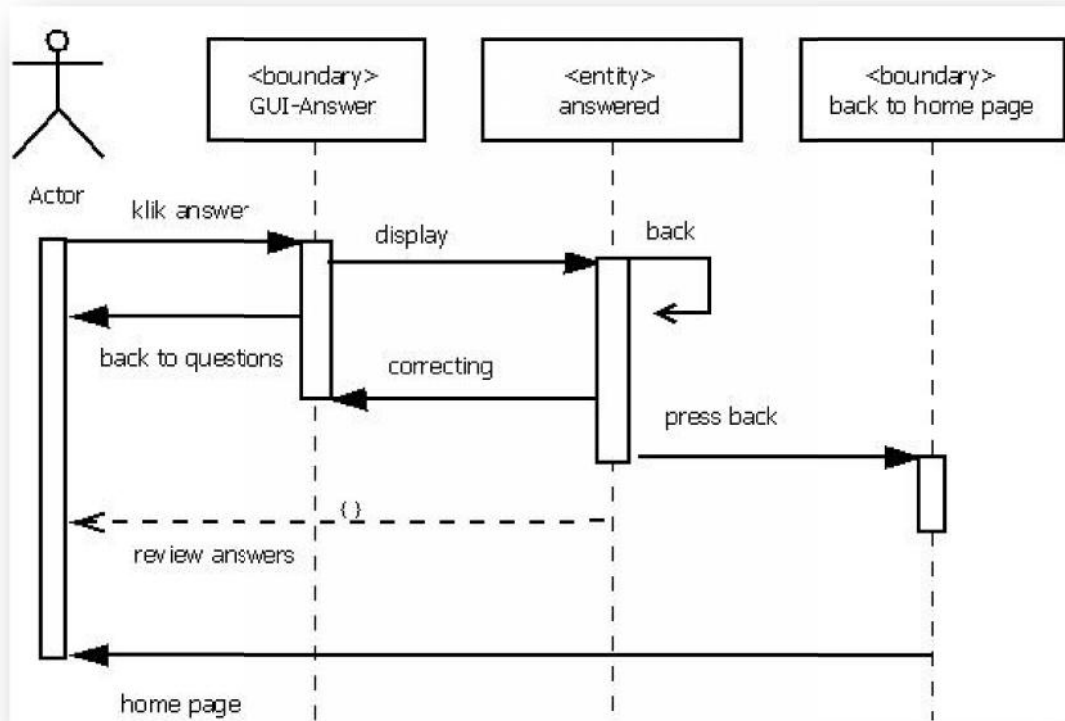


Gambar III-8 Sequence GUI-Grade.

Penjelasan proses mendapat grade

- Setelah system telah submitted answer, system akan melakukan proses guna mengoreksi hasil jawaban.
- System menampilkan grade user
- Display yang ditampilkan pada grade berupa percentage jawaban yang benar dari jumlah soal.
- Hasil grade akan di simpan pada result nilai pada menu history.

### III.2.4.6 Diagram Sequence GUI-Answer

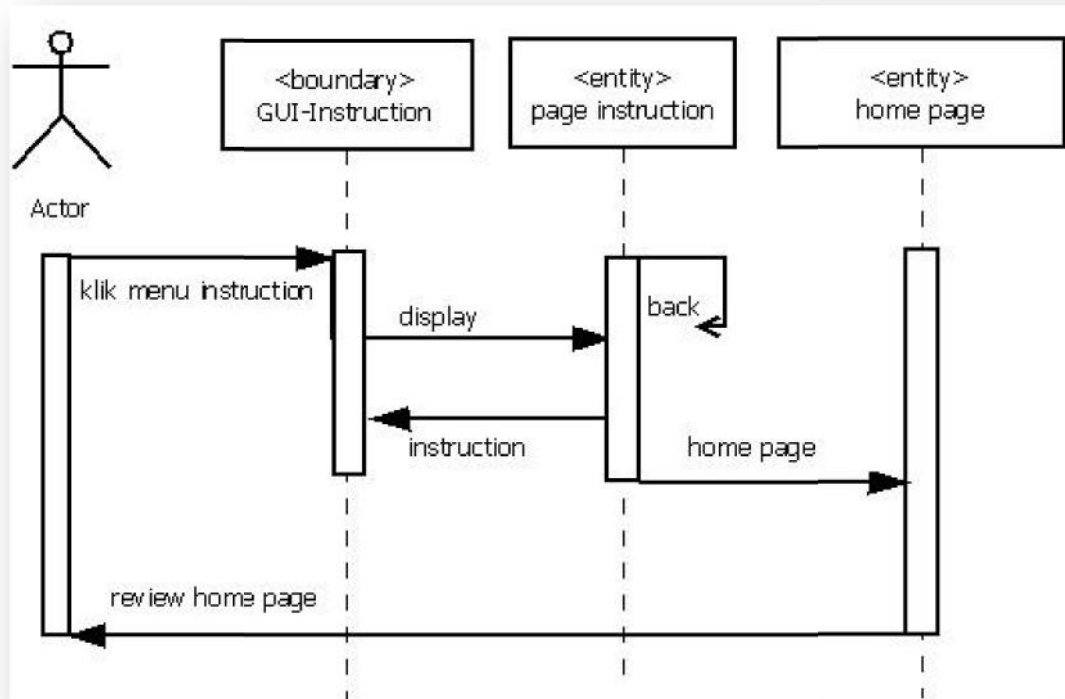


Gambar III-9 Sequence GUI-Answer.

Penjelasan proses answer

- Press tombol answer pada halaman grade
- Halaman answer di tampilkan
- System akan kembali pada halaman questions
- User dapat rivew hasil dari pembahasan soal, dan system akan menampilkan jawaban yang salah dan yang benar.
- Setelah selesai melakukan pengecekan jawaban, user dapat meninggalkan halaman dan menuju halaman home page.

### III.2.4.7 Diagram Sequence GUI-Instruction

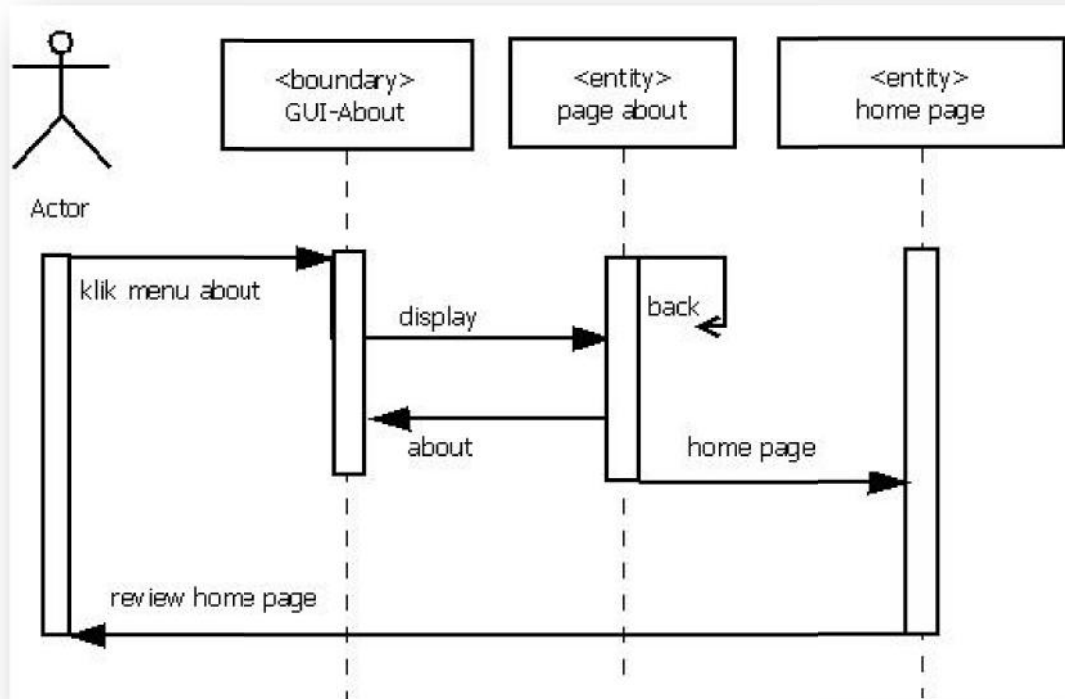


**Gambar III-10** Sequence GUI-Instruction.

Penjelasan proses menu Instruction

- Pada halaman home page user dapat menekan tombol menu instruction
- System menampilkan halaman instruction
- System menjelaskan instruction penggunaan aplikasi, dimana user dapat membaca petunjuk tersebut.
- User dapat kembali pada halaman home page dengan menekan tombol back
- Halaman home page ditampilkan.

### III.2.4.8 Diagram Sequence GUI-About

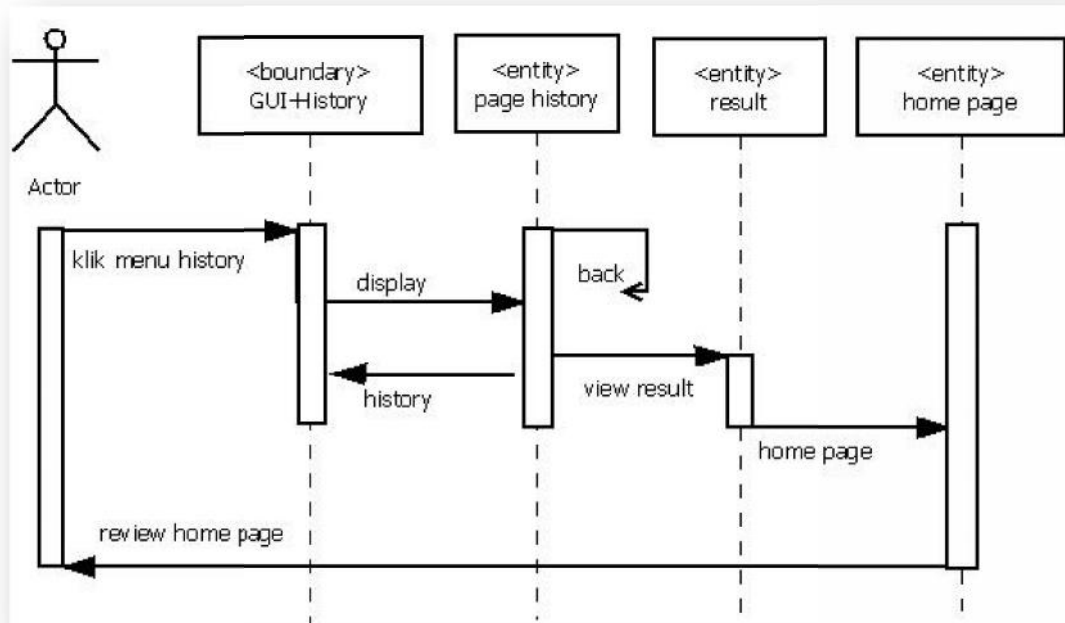


**Gambar III-11 Sequence GUI-About.**

Penjelasan proses menu about

- Pada halaman home page user dapat menekan tombol menu about
- System menampilkan halaman about
- Halaman about akan menjelaskan sebagian kecil definisi dan tujuan dari aplikasi tersebut.
- User dapat kembali pada halaman home page dengan menekan tombol back
- Halaman home page ditampilkan.

### III.2.4.9 Diagram Sequence GUI-History

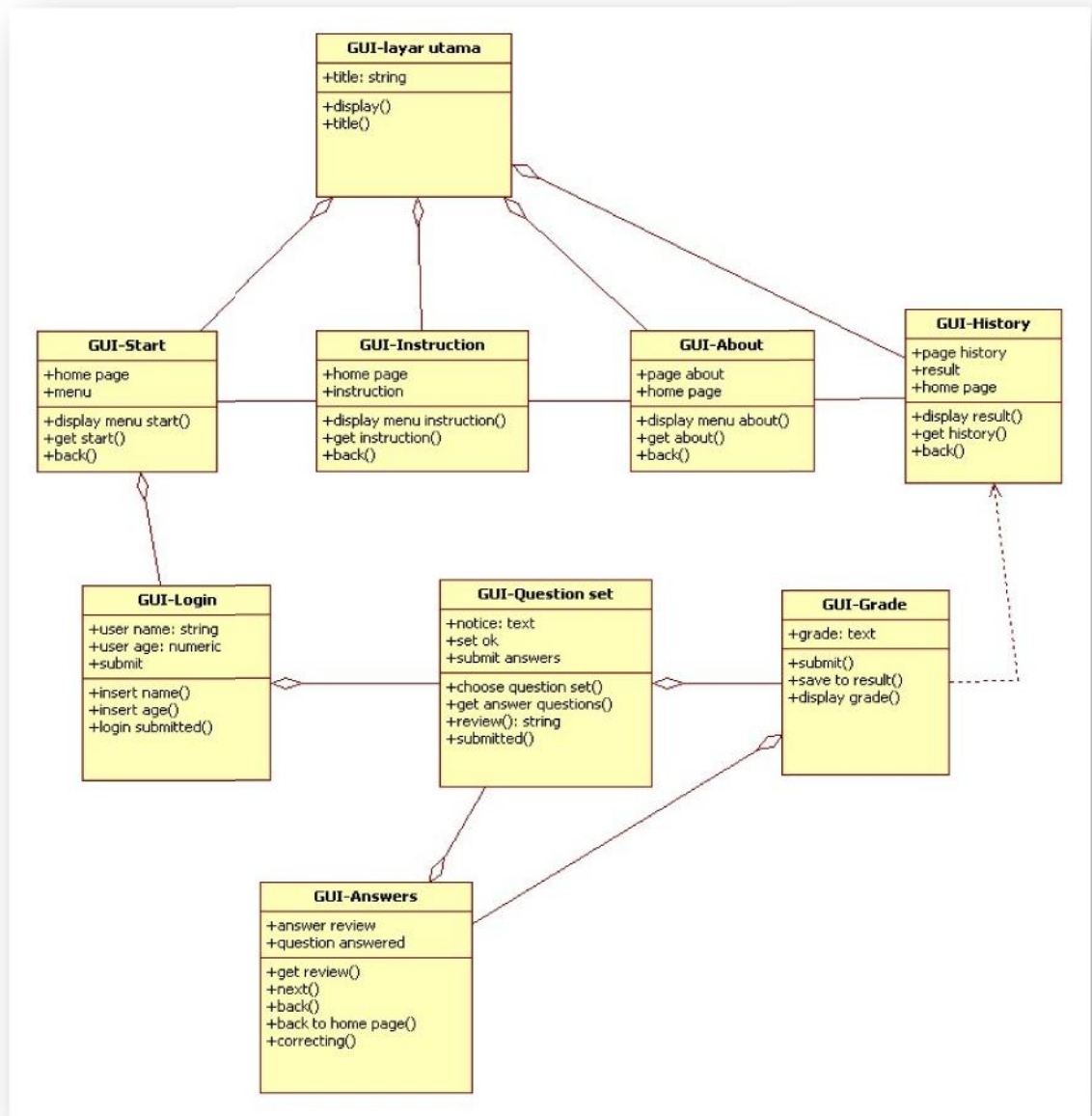


Gambar III-12 Sequence GUI-History.

Penjelasan proses menu history

- Pada halaman home page, user dapat menekan tombol menu history
- System menampilkan halaman history
- Pada halaman history sistem menampilkan result nilai dari user
- Result didapat pada proses grade user
- User dapat kembali pada halaman home page dengan menekan tombol back
- Halaman home page ditampilkan.

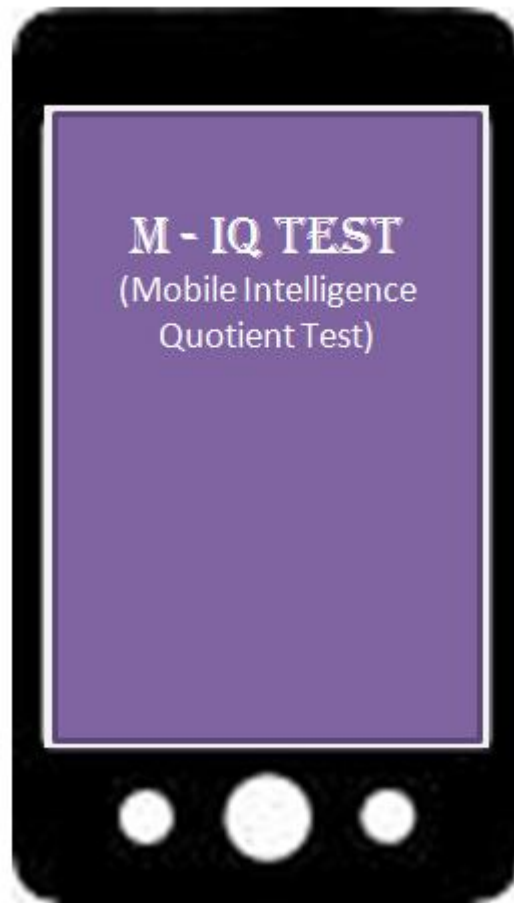
### III.2.5 Diagram Kelas Tes IQ



Gambar III-13 Diagram Kelas Tes IQ.

### III.3 Perancangan Tampilan

#### III.3.1 Rancangan Tampilan Layar Utama

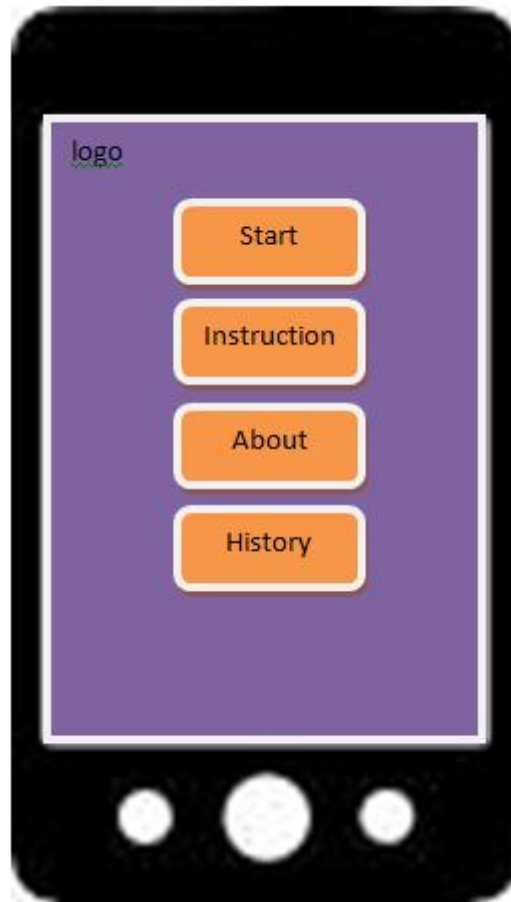


**Gambar III-14 Tampilan Layar Utama.**

**Tabel 9:** Fungsi Layar Utama

Nama	Keterangan
Layar utama	Berfungsi untuk menampilkan nama dari aplikasi. Halaman ini muncul ketika user pertama kali membuka aplikasi mobile intelligence quotient test (M-IQ Test)

### III.3.2 Rancangan Tampilan Home Page Menu

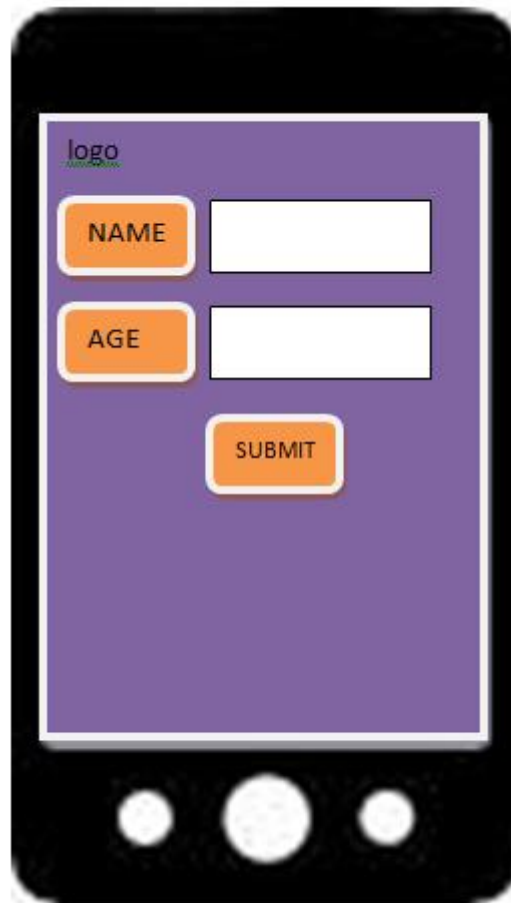


Gambar III-15 Tampilan Layar Home Page Menu.

Tabel 10: Fungsi Menu Home Page

Nama	Keterangan
Tampilan home page	<ul style="list-style-type: none"><li>- Menu start berfungsi untuk memulai aktifitas</li><li>- Menu instruction berfungsi untuk menjelaskan cara menggunakan aplikasi</li><li>- Menu about menjelaskan definisi aplikasi</li><li>- Menu history sebagai halaman result nilai dari user.</li></ul>

### III.3.3 Rancangan Tampilan Login

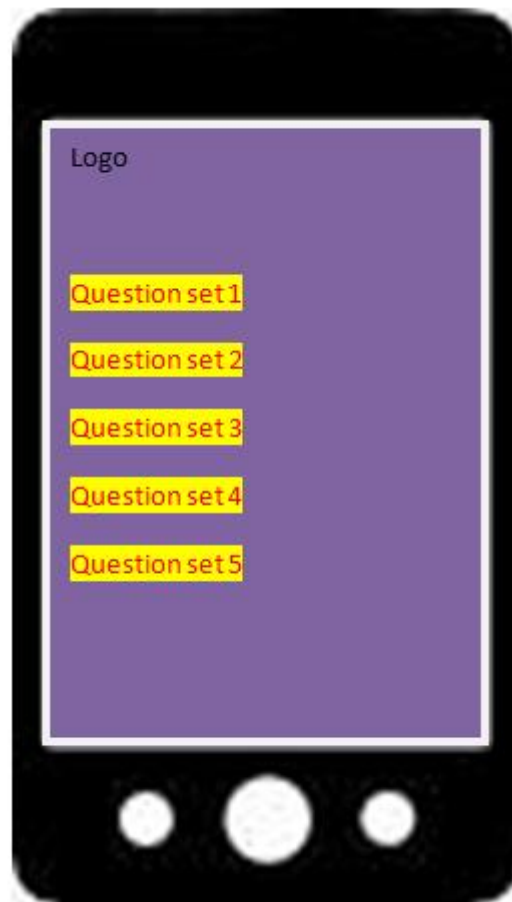


Gambar III-16 Tampilan Layar Login.

Tabel 11:Fungsi Login

Nama	Keterangan
login	Login berfungsi sebagai konfirmasi data dari user dengan memasukkan data berupa nama dan umur pengguna. Data yang dimasukkan oleh user akan ditampilkan pada halaman history sebagai result nilai yang telah didapat. Tombol submit untuk menyimpan data yang telah dimasukkan.

### III.3.4 Rancangan Tampilan Question Set

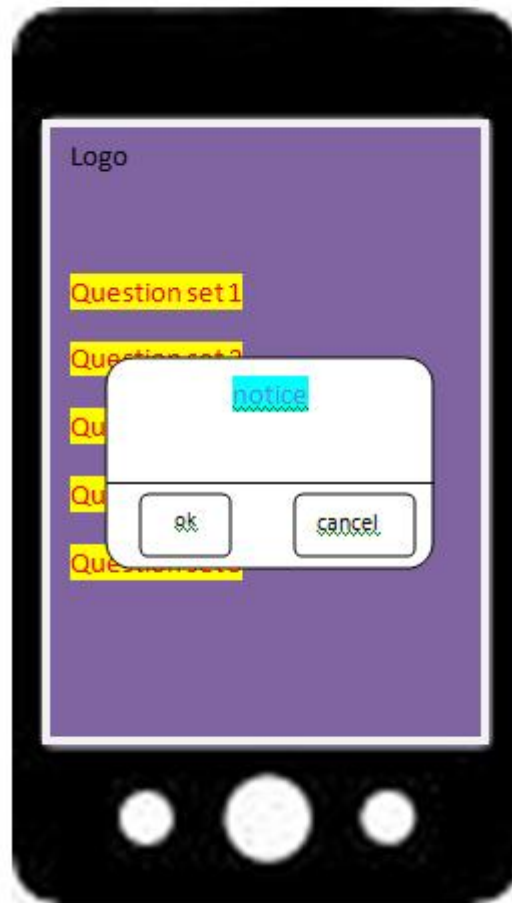


Gambar III-17 Tampilan Layar Question Set.

Tabel 12: Fungsi Tampilan Question Set

Nama	Keterangan
Question set	Pada halaman question set terdapat beberapa pertanyaan, user dapat memilih salah satu pertanyaan yang akan dijawab dimana 1 question set terdapat 20 soal. Setelah dipilih user mulai untuk mengerjakan soal.

### III.3.5 Rancangan Menampilkan Notice

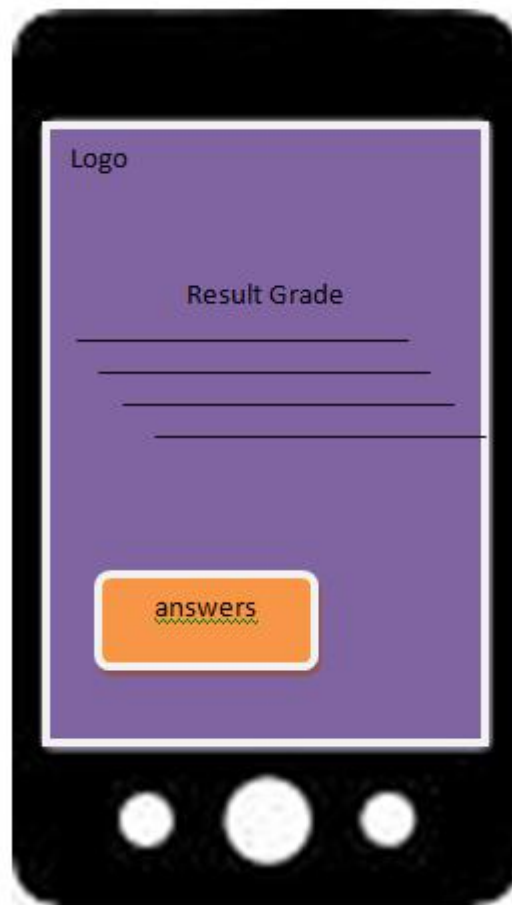


**Gambar III-18 Menampilkan Notice.**

**Tabel 13:**Fungsi Tampilan Notice

Nama	Keterangan
Notice	Sebelum mengerjakan soal, system akan menampilkan notice dimana notice tersebut menanyakan pada user apakah user yakin untuk memulai menjawab pertanyaan.

### III.3.6 Rancangan Tampilan Grade

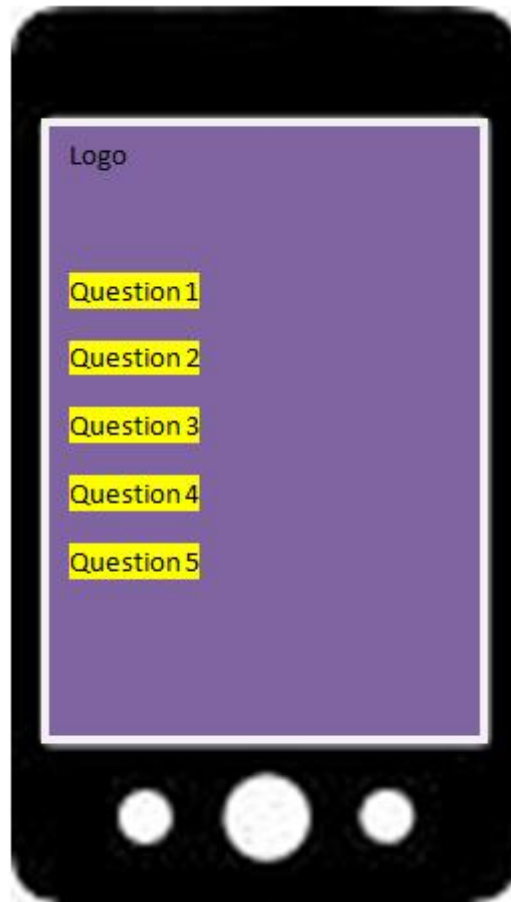


**Gambar III-19 Menampilkan Grade.**

**Tabel 14: Fungsi Tampilan Grade**

Nama	Keterangan
Grade	Grade berfungsi untuk mengetahui hasil yang didapat oleh user setelah mengerjakan soal. Nilai yang diberikan berupa percentage hasil keseluruhan jawaban yang benar.

### III.3.7 Rancangan Review Question Answered



Gambar III-20 Tampilan Review Question Answered.

Tabel 15: Fungsi Tampilan Review Answers

Nama	Keterangan
Answers	<ul style="list-style-type: none"><li>- Tombol answer berfungsi untuk review hasil jawaban.</li><li>- Review yang di tampilkan yaitu jawaban dari system dan jawaban dari user</li><li>- Setelah selesai review user dapat kembali pada home page Tes IQ.</li></ul>

### III.3.8 Rancangan Tampilan Answer Questions



Gambar III-21 Tampilan Answer Questions.

Tabel 16: Tampilan Answers Questions

Nama	Keterangan
Answer questions	<ul style="list-style-type: none"><li>- Rancangan tampilan answer question menunjukkan soal dan pilihan jawaban.</li><li>- User dapat next untuk soal berikutnya</li><li>- Review untuk melihat keseluruhan soal yang telah di jawab</li><li>- Submit untuk menyimpan hasil jawaban supaya system dapat memproses hasil.</li></ul>

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

Setelah sistem dianalisis dan didesain secara rinci, maka akan menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehingga pengguna dapat memberikan masukan kepada pembangun sistem. Untuk mengetahui apakah aplikasi tersebut dapat melakukan fungsi sesuai dengan perencanaan, maka diperlukan pengujian-pengujian.

#### IV.1 Implementasi

##### IV.1.1 Struktur Objek M-IQ Test

Fungsi-fungsi yang telah dirancang kemudian diimplementasikan. Adapun object dan file suara yang terdapat pada aplikasi game M-IQ Test ini dapat dilihat pada table dibawah ini.

**Tabel 17:** Struktur Objek M-IQ Test

Nama Tampilan	Nama Objek	Jenis Objek	Keterangan
Tampilan awal	Title	Movie Clip	Tampilan awal yang akan menampilkan menu pada home page
	StartBtn	Button	melanjutkan ke login
	exitBtn	Button	keluar dari game
	instructionBtn	Button	menampilkan instruksi
	aboutBtn	Button	menampilkan about
	historyBtn	Button	menampilkan history
	Klik	Sound	mengeluarkan suara pada setiap button

Nama Tampilan	Nama Objek	Jenis Objek	Keterangan
Start	Login	Movie Clip	
	setName	set button	login user dengan memasukkan nama
	setAge	set button	login user dengan memasukkan Age
	Kilk	Sound	sound button
Instruction	Intruksi	Movie Clip	masuk pada menu instruksi
	backBtn3	Button	kembali pada halaman utama
	Kilk	Sound	sound button
About	About	Movie Clip	masuk pada menu about
	backBtn3	Button	kembali pada halaman utama
	Kilk	Sound	sound button
Choose Question Set	Set	Button	memilih Question Set
	btnSet1	Button	memulai permainan question set 1
	btnSet2	Button	memulai permainan question set 2
	btnSet3	Button	memulai permainan question set 3
	btnSet4	Button	memulai permainan question set 4
	btnSet5	Button	memulai permainan question set 5
	backBtn	Button	kembali pada halaman utama
	Kilk	Sound	sound button
Start Game	prevBtn	Button	untuk menampilkan kembali soal yang telah di jawab

Nama Tampilan	Nama Objek	Jenis Objek	Keterangan
	nextBtn	Button	untuk menampilkan soal selanjutnya
	submitBtn	Button	untuk menyimpan jawaban dari si user
	Time	Timer	waktu dalam hitungan mundur
Result	Grade	Text	menampilkan hasil nilai
	backBtn2	Button	kembali pada halaman utama
	btnReview	Button	menampilkan kembali hasil jawaban dari sistem dan user
	Kilk	Sound	sound button
Review	showReview	Button	mereview kembali soal
	btnBack4	Button	kembali pada halaman utama
	prevBtn2	Button	kembali ke nomor soal sebelumnya
	nextBtn2	Button	menuju review soal berikutnya
	Kilk	Sound	sound button
History	txtHistory	Text	menampilkan grade dan hasil result dari si user
	backBtn6	Button	kembali pada halaman utama
	upBtn	Button	up page
	downBtn	Button	down page
	Kilk	Sound	sound button

#### IV.1.2 File Action Script

File *Action Script* yang telah diimplementasikan untuk sistem adalah sebagai berikut:

**Tabel 18:** Struktur File M-IQ Test

Column	Type	Null	Default	Keterangan
dataSoal	XML	No		
soalKe	Int	No	0	
soalTerjawab	Int	No	0	iqGrade
indexSoal	Int	No		
jawabanBenar	String	No		
Pilihan	Varchar	No		
waktuJawab	Timer	No	Current_timestamp	TimeEvent
arrJawab	Varchar	No	0	
numSoal	Int	No		
iqValue	Int	No		
iqGrade	String	No		
Count	int	No	Current_timestamp	1200
Menit	int	No	Current_timestamp	19
Detik	int	No	Current_timestamp	60
arrSoal	Varchar	No		
arrNama	Varchar	Yes		Nama:String
arrUmur	Varchar	Yes		Umur:String

Column	Type	Null	Default	Keterangan
arrGrade	Varchar	No		
arrIq	Varchar	No		
sKlik	Klik	Yes		Sound

#### IV.1.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap halaman aplikasi yang dibuat dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi antarmuka yang telah dibuat dalam dua format:



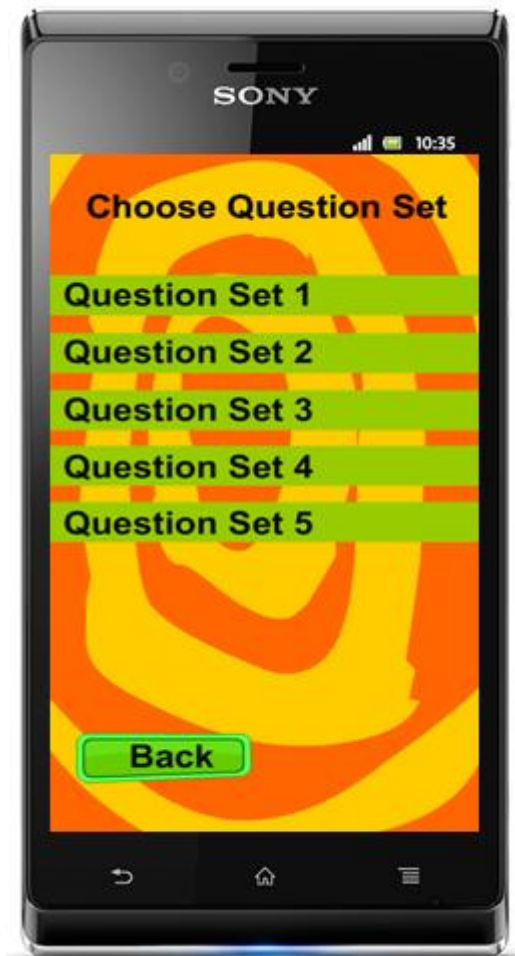
Gambar IV-1 Halaman Utama Aplikasi



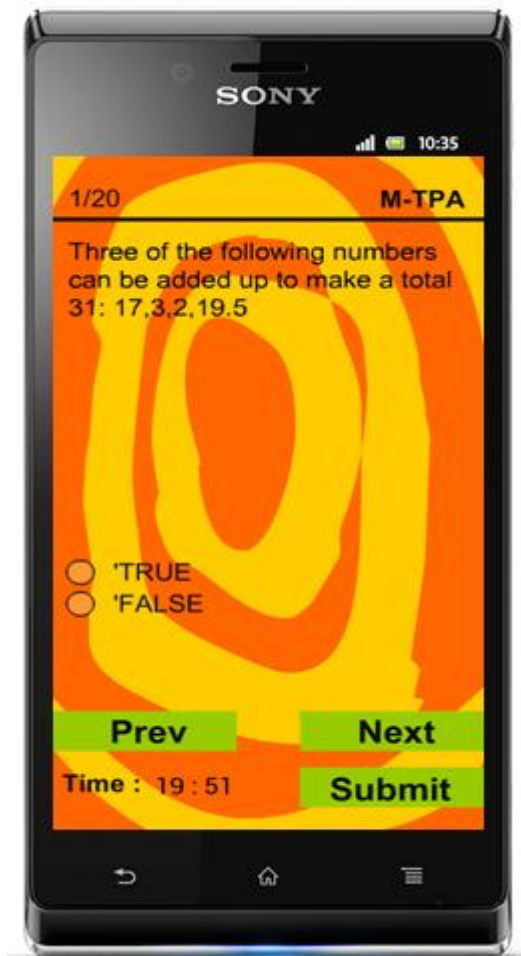
**Gambar IV-2 Login Aplikasi M-IQ Test**



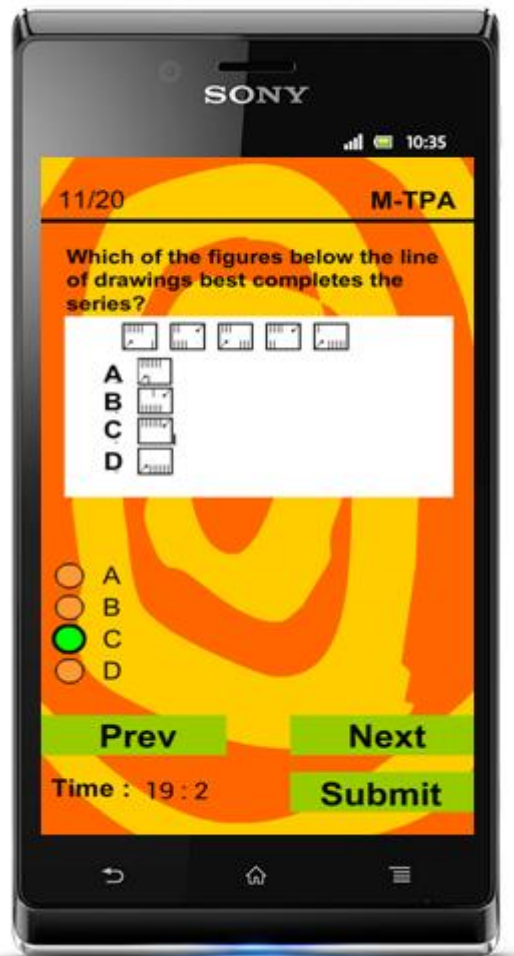
Gambar IV-3 Input Name dan Age



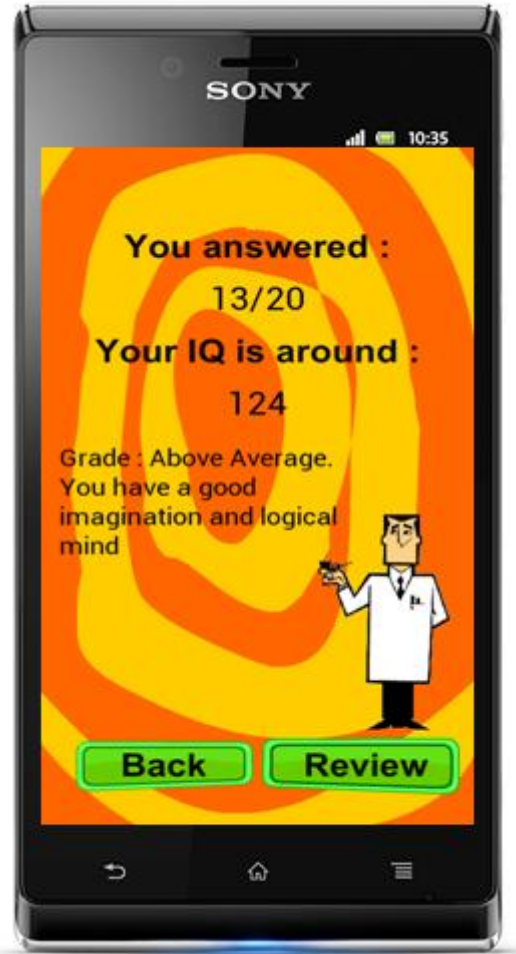
**Gambar IV-4 Choose Question Set**



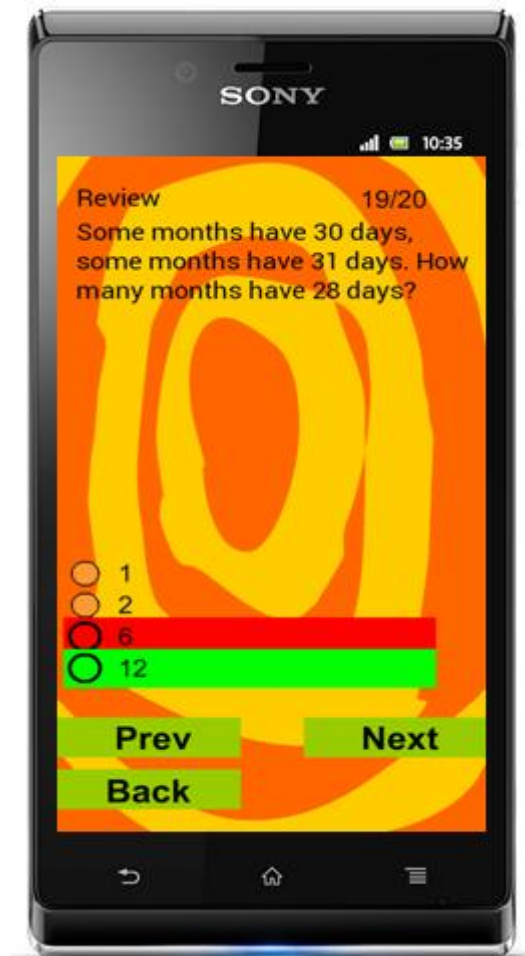
Gambar IV-5 Tampilan Soal M-IQ Test



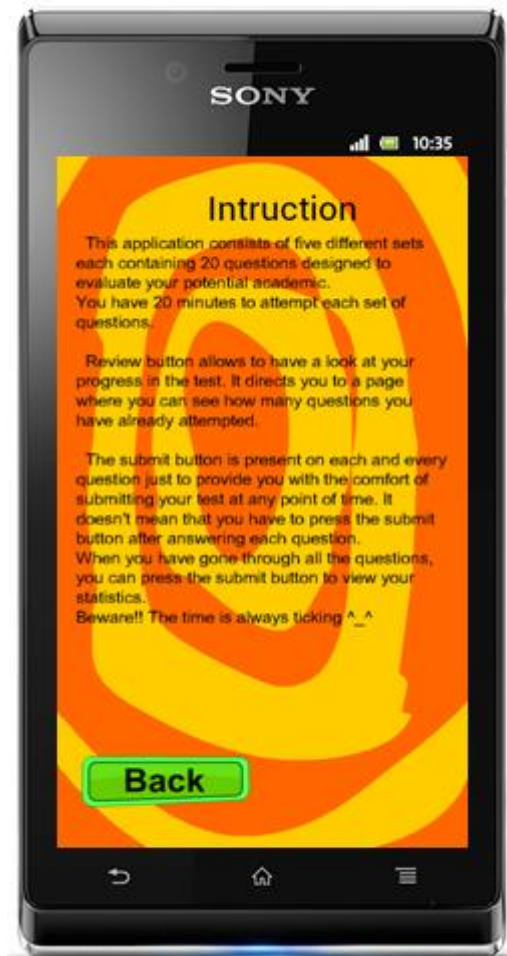
Gambar IV-6 Menampilkan Soal Setelah di Jawab



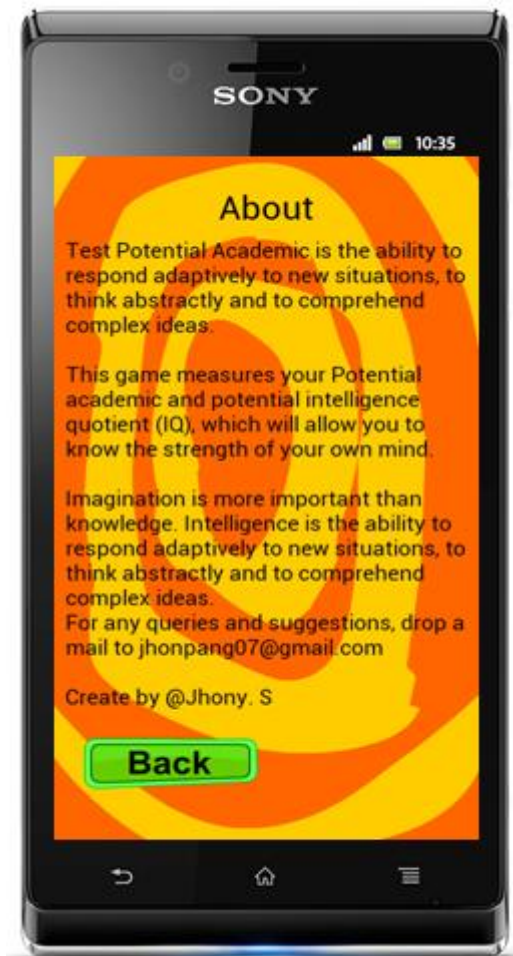
Gambar IV-7 Tampilan Grade IQ



**Gambar IV-8 Tampilan Review Soal**



**Gambar IV-9 Tampilan Menu Instruction**



**Gambar IV-10 Tampilan Menu About**



**Gambar IV-11 Tampilan Menu History Sebelum Menjawab Soal**



**Gambar IV-12 Menu History Setelah Menjawab Soal**

## IV.2 Pengujian dan Hasil

Strategi *testing* yang penulis lakukan menggunakan strategi *Black Box*. *Black Box testing* merupakan strategi *testing* di mana hanya memperhatikan atau memfokuskan kepada faktor fungsionalitas dan spesifikasi perangkat lunak. Berbeda dengan *white box*, *black box testing* tidak membutuhkan pengetahuan mengenai alur internal (*internal path*), struktur atau implementasi dari *software under test* (SUT). Karena itu uji coba *black box* memungkinkan untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional program [18].

### IV.2.1 Hasil Pengujian Fungsionalitas Seluruh Menu pada Aplikasi M-IQ Test

**Table 19:** Pengujian Fungsionalitas Seluruh Menu

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	
				Berhasil	Gagal
1.	Masuk pada aplikasi awal, dengan menampilkan judul aplikasi		Sistem menampilkan judul aplikasi berupa animasi movieclip	✓	
2.	Mengklik tombol start untuk masuk pada mulai game	Kode pengguna: 1	System akan menampilkan menu login setelah user/pengguna menekan tombol start	✓	
3.	Mengklik tombol Login dengan memasukkan Nama dan Umur sebagai identifikasi dari pengguna	arrNama.push(nama) arrUmur.push(umur)	Setelah pengguna memasukkan nama dan umur, maka system akan menyimpan data	✓	

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	
				Berhasil	Gagal
			pengguna sebagai output grade IQ / nilai IQ yang didapat pengguna		
4.	Mengklik tombol Login Tanpa memasukkan Nama dan Umur sebagai identifikasi dari pengguna	Start	System akan memulai game tanpa ada input nama dan umur dari pengguna. Namun hasil akhir pada history tidak akan ditampilkan nama pengguna	✓	
5.	Memasuki tahap memulai game dengan memilih Question Set yang disediakan sebanyak 5 set	Quiz: -- Question Set--	System menampilkan Question sesuai yang dipilih oleh pengguna	✓	
		Start Game Timer : Set timer: Count : 1200/detik Menit : 20	System menampilkan timer untuk batas waktu pengerjaan soal-soal	✓	
		PrevQuestion	System dapat menampilkan kembali soal yang telah dijawab oleh pengguna	✓	
		Next	System akan melanjutkan untuk	✓	

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	
				Berhasil	Gagal
			menampilkan soal-soal berikutnya yang belum terjawab oleh pengguna game.		
6.	Mengakhiri permainan dengan Submit jawaban	SimpanJawaban: Jawab (indexSoal) = 0 soalTerjawab =0	System akan menyimpan jawaban setelah submit	✓	
7.	Mengakhiri permainan Tanpa Submit jawaban		System tidak akan menyimpan jawaban karena pengguna tidak submit untuk mengetahui nilai akhir		
8.	Mengulang kembali untuk memulai game dengan question set yang sama	dataSoal = xml randomSoal()	System akan menampilkan question dengan spesifikasi yang telah di rancang dengan soal Random	✓	
9.	Mengklik submit untuk mengetahui result Grade yang di dapat pengguna	Grade: minValue=70 minPoint=0 rangeGrade=2 tirValue=19 iqGrade=	System menampilkan rancangan dari result grade berupa Point Grade dan IQ grade, dengan spesifikasi grade yang telah	✓	

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	
				Berhasil	Gagal
		grade : Below average. You have a below average imagination and logical mind	dirancang pada system.		
10.	Mengklasifikasi jawaban yang benar dan yang salah setelah mendapat grade dengan cara review Question	tampilSoalReview	System menampilkan kembali soal dengan ketentuan terdapat nilai jawaban yang benar oleh system dan jawaban salah hasil input dari pengguna	✓	
11.	Mengklik tombol back untuk kembali ke menu halaman utama, setelah review question	Back to home page	Aplikasi akan kembali ke halaman utama setelah menekan tombol back	✓	
12.	Mengklik tombol instruction untuk masuk pada menu		System aplikasi akan menampilkan text berupa instruction cara penggunaan aplikasi	✓	
13.	Mengklik tombol about untuk masuk pada menu		System aplikasi akan menampilkan text about berupa penjelasan dari pengertian aplikasi	✓	

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	
				Berhasil	Gagal
14.	Mengklik tombol History untuk mengetahui hasil result grade	txtHistory: “Nama: “Umur: “IQ: “Grade:	System menampilkan hasil dari tes, berupa : Nama : Umur , setelah pengguna menginputkan data pada saat login. IQ : Grade : , system akan menampilkan nilai IQ berupa percentage, dan Grade berupa tingkatan IQ.	✓	
15.	Mengklik setiap button yang ada pada aplikasi	Sound : klik	System akan menghasilkan suara setelah button di klik	✓	

#### IV.2.2 Pengujian Aplikasi Pada Pengguna

Tabel 20: Pengujian Aplikasi Pada Pengguna.

Kode Pengguna	(0) Kurang (X)	(1) Cukup (Y)	(2) Sangat (Z)
Pengguna 1	X	✓	X
Pengguna 2	X	✓	X
Pengguna 3	X	X	V
Pengguna 4	X	X	✓
Pengguna 5	X	✓	X

- Rumus uji kepuasan *tester* :

Dik :

- Total pengguna = 30 orang
- Jumlah pemilih Kurang = 1 orang (A)
- Jumlah pemilih Cukup = 22 orang (B)
- Jumlah pemilih Baik = 7 orang (C)

Dit : persentase kepuasan

Jawab : Rumus (  $A/TxX + B/TxY + C/TxZ$  ) = G

$$1/30x0 + 22/30x1 + 7/30x2 = 1.19$$

Ket :

- G = hasil
- A = jumlah pemilih kurang
- B = jumlah pemilih cukup
- C = jumlah pemilih baik
- X = bobot kurang (0)
- Y = bobot cukup (1)
- Z = bobot baik (2)
- T = total pengguna

Keterangan Rumus:

- Didapatkan 30 orang *tester*, apabila semua tester tersebut memilih kategori baik *otomatis* jumlah pemilih baik menjadi 30, sedangkan kategori baik memiliki nilai bobot 2. Apabila dikalikan  $30x2 = 60$  (jumlah orang yang memilih baik x bobot baik), maka untuk mendapatkan *point* 100, harus dikalikan dengan 10.

Maka didapat kesimpulan bahwa Aplikasi M-IQ Test mendapat 70% dari hasil tes yang dilakukan sehingga aplikasi sudah cukup baik, dimana hasil perhitungan terdapat pada lampiran. Karena aplikasi memberikan soal berupa bahasa inggris sehingga pengguna terasa kesulitan untuk mengartikan soal bagi pengguna yang belum paham akan bahasa inggris.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **V.1 Kesimpulan**

Berdasarkan Tugas Akhir yang telah dibangun maka kesimpulan yang dapat diambil adalah:

1. Aplikasi media pembelajaran berupa latihan Tes IQ berbasis android telah dibangun sesuai perancangan.
2. Aplikasi dapat menyajikan tingkat IQ (*intelligence quotients*) dari pengguna.
3. Aplikasi menyajikan soal dengan menggunakan bahasa inggris.
4. Aplikasi Tes IQ dapat digunakan pada kumunitas Internasional.
5. Aplikasi Tes IQ dengan segala fasilitasnya dapat memberikan hasil tes dengan cepat dan benar.

#### **V.2 Saran**

Berdasarkan kesimpulan yang telah diperoleh dalam tugas akhir ini, diajukan beberapa saran yang perlu dipertimbangkan sebagai berikut:

1. Untuk memperbaiki sistem, diperlukan Update versi aplikasi
2. Menambahkan lebih banyak soal-soal tes.
3. Menambahkan fitur- fitur interaktif lainnya untuk menunjang pengguna tidak cepat bosan.

## DAFTAR PUSTAKA

1. Effendi, F. R. 2013. *Android Sistem Operasi pada Smartphone*. Universitas Surabaya, Surabaya.
2. Azwar, Sifuddin. 2007. *Kualitas Tes Potensi Akademik Versi 07A*: Universitas Gadjah Mada.UGM. <http://azwar.staff.ugm.ac.id/files/2010/04/Evaluasi-TPA-Versi-07A.pdf>
3. Dimanthara, Agus. 2013. *Pembanguna Aplikasi Simulasi TRY OUT Ujian Nasional SMA Berbasis Mobil*; S1 thesis, UAJY.
4. Dinata, Nyoman Trisna. 2010 *Pembangunan Aplikasi Tes Potensi Akademik Menggunakan Framework CodeIgniter*: S1 thesis, UAJY.
5. Betts, M. 17 November 2003, *The Almanac: Hot Tech*, ComputerWorld 52.
6. Berkhin, P. 2002, *Survey of Clustering Data Mining Techniques*, Accrue Software, 1045 Forest Knoll Dr., San Jose.
7. Fowler,M., Scott, K.1999. *UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language*; Addison Wesley.
8. Agus, Nugroho. 2012. *Pengenalan Flash dan ActionScript 3.0*: Lecturer. <http://lecturer.ukdw.ac.id/cnuq/wp-content/uploads/animasi/bab1.pdf>
9. Fandy. 2011. *Pengantar XML*: Universitas Sriwijaya. Jurnal.
10. Safaat, Nazruddin. 2011 *ANDROID – Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
11. Sudjana, Nana. 1991. *Penilaian Hasil Proses Belajar Mengajar*, cetakan ketiga. Bandung: PT Remaja RosdaKarya.
12. Kurniawan Khannedy ,Eko 2012.*Membuat Android Sederhana*.Bandung: StripBandunk.com
13. Tarmizi, Muhammad. 2005, *Aplikasi Tes Online*, Tugas Akhir, Batam: Politeknik Batam.
14. Mitna, Kiki, 2012. *Using Adobe Flash Professional CS5 (Interactive Video) to Teach Listening in Senior High School*. Universitas Negeri Padang. [ejournal.unp.ac.id/index.php/jelt/article/download/jelt1a5275](http://ejournal.unp.ac.id/index.php/jelt/article/download/jelt1a5275)

15. Syahirul, Muhammad Huda. 2012. *Aplikasi Tes IQ Berbasis Mobile: Sekolah Tinggi Manajemen Informatika dan Komputer*. Amikom Yogyakarta.
16. Nur'aeni, 2012. *Tes Psikologi: Tes Inteligensi dan Tes Bakat*: Yogyakarta, Universitas Muhammadiyah. <http://digilib.ump.ac.id/files/disk121/jhptump-ump-gdl-nuraenisps-1031-1-fulltek-u.pdf>.
17. Putra, Ismadi. 2014. *Min 2 Tanjungbalai-Tes IQ Anda disini*: Tanjungbalai.
18. Rosenberg, Doug and Stephens, Matt. 2007. *Use Case Driven Object Modeling with UML: Theory and Practice*. APress.
19. Riwinoto. 2014. *Pengembangan Game Home Sweet Home dalam Pembelajaran Kosakata Bahasa Inggris untuk Anak Usia Dini menggunakan Diagram Unified Modelling Language, Scriptwriting dan Storyboard*. Batam <http://journal.uui.ac.id/index.php/Snati/article/view/3277>
20. Lia, K. 2006. *Mahir dan terampil berkomputer*. Jakarta: Grafindo Media Pratama.
21. Supryadi, Agus and Edi Purnomo. 2013. *MOBIK-Aplikasi Kuis Generator Berbasis Android: Tugas Akhir*. Politeknik Negeri Batam.