

SKRIPSI

APLIKASI PSIKOTES BERBASIS MOBILE MENGGUNAKAN J2ME



Disusun Oleh :

AHMAD KHOLISIN

NIM 06.12.592



**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011**

LEMBAR PERSETUJUAN

APLIKASI PSIKOTES BERBASIS MOBILE MENGGUNAKAN J2ME

SKRIPSI

Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai gelar sarjana teknik

Disusun Oleh :

AHMAD KHOLISIN

NIM : 06.12.592



Mengetahui
Ketua Program Studi Teknik Elektro S-1

[Signature]
Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Dosen Pembimbing I

I Komang Somawirata, ST, MT
NIP.P. 1030100361

Dosen Pembimbing II

Sandy Nataly Mantja, Skom
NIP.P.1030800418

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

ABSTRAKSI

APLIKASI PSIKOTES BERBASIS MOBILE MENGGUNAKAN J2ME

Ahmad Kholisin, NIM 0612592

**Dosen Pembimbing : I Komang Somawirata, ST, MT, MT dan Sandy Nataly
Mantja, Skom**

Psikotes adalah suatu tes kecerdasan yang dilakukan oleh perusahaan, sekolah, dll, untuk mendapatkan kriteria ideal orang yang akan direkrut karena dengan mengikuti psikotes maka emosi, kecerdasan dan intuisi dari setiap orang akan diketahui. banyak orang menganggap psikotes sulit ditaklukkan maka Skripsi ini memuat Konsep Aplikasi Psikotes menggunakan Java dengan harapan semua orang dapat belajar langsung Psikotes dengan cepat, tanpa harus bingung mencari Materi diinternet membeli buku atau yang lain, tetapi cukup hanya dengan menggunakan Aplikasi Psikotes berbasis mobile, maka secara otomatis semua orang dapat langsung belajar Psikotes dimana saja dan kapan saja.

Semoga Aplikasi ini banyak memberikan manfaat bagi semua orang dikehidupan sekarang dan yang akan datang

Kata kunci: aplikasi, psikotes, java

KATA PENGANTAR

Dengan mengucapkan syukur kehadiran Allah SWT yang dengan segala Kasih dan Anugerah Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul :

“APLIKASI PSIKOTES BERBASIS MOBILE MENGGUNAKAN J2ME “

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata-1 di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE., selaku Rektor ITN Malang.
2. Bapak Ir. Sidik Noertjahyono, MT., selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT., selaku Ketua Jurusan Teknik Elektro S – 1 ITN Malang.
4. Bapak I Komang Somawirata, ST, MT, selaku Dosen Pembimbing I.
5. Ibu Sandy Nataly Mantja, Skom, selaku Dosen Pembimbing II.
6. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua. Amin.

Malang, Februari 2011

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	1
1.3. Tujuan dan Manfaat.....	1
1.4. Batasan Masalah.....	2
1.5. Metodologi	2
1.6. Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	4
2.1. Pengertian dasar sistem	4
2.1.1. Karakteristik sistem	4
2.1.2. Klasifikasi sistem.....	6
2.2. Sejarah java.....	7
2.2.1. Sebuah Bahasa Pemrograman.....	9
2.2.2. Sebuah Development Environment.....	9
2.2.3. Sebuah Aplikasi Java	9
2.2.4. Deployment environment	9
2.2.5. Karakteristik Java	9
2.2.6. Sebagian Fitur Java.....	11
2.2.6.1. Java Virtual Machine (JVM).....	11
2.2.6.2. Code Security	11
2.2.6.3. Fase – Fase Pemrograman Java.....	12

2.3. J2me.....	12
2.4. Data Flowchart	13
2.4.1. Pedoman – Pedoman Dalam Membuat Flowchart	14
2.4.2. Simbol – Simbol Flowchart.....	15
2.5. Psikotest.....	17
2.5.1. Macam – Macam Psikotest.....	18
2.6. Rms.....	19
BAB III ANALISA DAN PERANCANGAN SISTEM	20
3.1. Gambaran Umum Psikotest.....	20
3.1.1. Gambaran Umum Program.....	20
3.1.2. Algoritma Program.....	20
3.2. Analisa Sistem.....	22
3.2. Perencanaan Perangkat Keras	22
3.4. Perancangan Sistem	23
3.4.1. Flowchart	23
3.4.2. Penjelasan Flowchart.....	23
3.5. Desain Antarmuka Aplikasi.....	24
3.5.1. Desain Form Utama.....	24
3.5.2. Desain Form Tampilan Main Menu.....	25
3.5.3. Desain Form Main Menu.....	25
3.5.4. Desain Form Menu.....	26
3.5.4. Desain Form Tampilan.....	26
BAB IV IMPLEMENTASI DAN PENGUJIAN	28
4.1. Kebutuhan Perangkat Lunak.....	28
4.1.1. Kebutuhan Perangkat Keras	28
4.1.2. Setup Program	28
4.2. Impelementasi Sitem	28
4.3. Pengujian Hasil	29
4.3.1. Tampilan Main Menu	29
4.3.2. Tampilan Mulai Test	30
4.3.3. Tampilan Menu	31
4.3.4. Tampilan Menu Test Antonim	31

4.3.5. Tampilan Menu Test Sinonim	32
4.3.6. Tampilan Menu Test Korelasi	32
4.3.7. Tampilan Menu Test Kuantitatif	33
4.3.8. Tampilan Menu Test Logika	33
4.3.9. Tampilan Menu Hasil	34
4.3.10. Tampilan Menu History	35
4.3.10. Tampilan Menu Update	36
4.3.11. Tampilan Menu Exit	37
4.4. Tabel Pengujian Aplikasi	38
4.4.1. Tabel Skoring	39
4.4.2. Tabel Perbandingan Sistem	40
4.4.2. Analisa Sistem	42
BAB V PENUTUP	43
5.1. Kesimpulan	43
5.2. Saran	43
DAFTAR PUSTAKA	44
LAMPIRAN – LAMPIRAN	

DAFTAR GAMBAR

BAB II DASAR TEORI

Gambar 2.1. Karakteristik Sistem.....	6
Gambar 2.2. Arsitektur Java.....	8
Gambar 2.3. Fase – Fase Java.....	12
Gambar 2.4. Sistem J2me.....	13
Gambar 2.5. Symbol Flowchart.....	15

BAB III PERANCANGAN DAN PEMBUATAN SISTEM

Gambar 3.1. Design Menu Aplikasi Psikotes Berbasis J2me.....	21
Gambar 3.2. Flowchart Aplikasi Psikotes.....	23
Gambar 3.3. Desain Form utama.....	25
Gambar 3.4. Desain Form Tampilan Main Menu.....	25
Gambar 3.5. Desain Form Main Menu.....	26
Gambar 3.6. Desain Form Menu.....	26
Gambar 3.7. Desain Form Tampilan.....	27

BAB IV IMPLEMENTASI DAN PENGUJIAN HASIL

Gambar 4.1. Tampilan Utama.....	29
Gambar 4.2. Tampilan Main Menu.....	30
Gambar 4.3. Tampilan Mulai Test.....	30
Gambar 4.4. Tampilan Menu.....	31
Gambar 4.5. Tampilan Menu Test Antonim.....	31
Gambar 4.6. Tampilan Menu Test Sinonim.....	32
Gambar 4.7. Tampilan Menu Test Korelasi.....	32
Gambar 4.8. Tampilan Menu Test Kuantitatif.....	33
Gambar 4.9. Tampilan Menu Test Logika.....	33
Gambar 4.10. Menu Hasil.....	34
Gambar 4.11. Tampilan Menu Hasil.....	34
Gambar 4.12. Tampilan Menu History.....	35
Gambar 4.13. Tampilan Menu Hasil History.....	35
Gambar 4.14. Tampilan Menu Update.....	36
Gambar 4.15. Tampilan Informasi Menu Update.....	36

Gambar 4.16. Tampilan Menu Exit.....	37
Gambar 4.17. Tampilan Exit.....	37

DAFTAR TABEL

BAB IV IMPLEMENTASI DAN PENGUJIAN HASIL

Tabel 4.1. Tabel Pengujian Aplikasi.....	38
Tabel 4.2. Tabel Skoring.....	39
Tabel 4.3. Tabel Perbandingan Sistem.....	41

BAB I PENDAHULUAN

1.1 LATAR BELAKANG

Psikotes memang satu hal yang menarik untuk dibahas karena banyak orang merasa bahwa psikoteslah yang menggagalkan keinginan mereka untuk bergabung dengan sebuah perusahaan, merasa psikotes adalah satu hal yang sulit ditaklukkan dan membutuhkan persiapan special. Bagi sebagian orang psikotes merupakan satu hal yang menakutkan atau minimal membuat mereka bertanya - tanya apa yang harus dilakukan ketika menghadapinya.

Ketika mendapatkan panggilan untuk melaksanakan psikotes, beragam respon yang dikeluarkan seperti berburu buku-buku sukses melaksanakan psikotes, bertanya kepada orang yang pernah menjalaninya, bertanya kepada ahlinya (psikolog) terkait apa yang perlu dipersiapkan atau dilakukan ketika psikotes sampai meminta bocoran soal-soal psikotes yang "biasanya" diberikan.

Pertanyaannya adalah perlukah kita melakukan hal-hal tersebut sebelum menghadapi psikotes, Karena pada umumnya orang yang sudah memahami arti dari psikotes, mereka justru tidak terlalu ribet dengan "ritual-ritual" tersebut melainkan menyiapkan fisik dan mentalnya agar bisa lebih siap dengan psikotes yang akan dihadapi.

Oleh karna itu pada skripsi ini, memuat Konsep Psikotes berbasis mobile menggunakan j2me dengan harapan semua orang dapat belajar langsung Psikotes dengan cepat, tanpa harus bingung mencari Materi diinternet membeli buku atau yang lain, tetapi cukup hanya dengan menggunakan aplikasi psikotes berbasis mobile, maka secara otomatis semua orang dapat langsung belajar psikotes dimana saja dan kapan saja,

1.2 RUMUSAN MASALAH

Adapun rumusan masalah yang akan dibahas dalam pembuatan skripsi ini adalah bagaimana membuat pengguna belajar psikotes dengan senang hati,

1.3 TUJUAN DAN MANFAAT

Tujuan dari pembuatan skripsi ini adalah memudahkan pengguna untuk mendapatkan latihan tentang psikotes dengan cepat dan akurat. harapannya dapat melakukan Tes Potensi Akademik tanpa harus bingung mencari materi, cukup dengan

memainkan psikotes berbasis mobile, maka secara otomatis pengguna langsung belajar psikotes,

1.4 BATASAN MASALAH

Agar permasalahan yang diangkat tidak meluas dan terarah sesuai tujuan maka aplikasi psikotes ini hanya mencakup tentang:

1. Tes Kemampuan Verbal – Antonim.
2. Tes Kemampuan Verbal - Sinonim.
3. Tes Kemampuan Verbal - Korelasi Makna.
4. Tes Kemampuan Kuantitatif - Deret Angka atau Huruf.
5. Tes Kemampuan Spasial – Logika Gambar.
6. Perangkat lunak yang digunakan dalam aplikasi ini adalah java.
7. Perangkat lunak dijalankan didalam handphone.
8. aplikasi tes potensi akademik ini membahas soal – soal yang sering keluar dalam seleksi pegawai.
9. Soal – soal psikotes yang ada mencakup seluruh bidang umum.

1.5 METODOLOGI

Data merupakan sumber atau bahan mentah yang sangat penting dalam proses menghasilkan sebuah informasi. Oleh sebab itu dalam pengambilan atau pengumpulan data perlu dilakukan penanganan secara cermat dan hati-hati, sehingga data yang diperoleh dapat bermanfaat dan berkualitas

Dalam pengumpulan data penyusun menggunakan metode sebagai berikut:

1. Studi Lapangan

Dengan metode ini data-data diperoleh langsung dari sumber yang bersangkutan, dimana peneliti berhadapan langsung dengan obyek yang diteliti, yang dilakukan dengan cara

a. Survey

Teknik pengumpulan data dengan cara terjun secara langsung dan mencatat secara sistematis terhadap obyek masalah.

b. Wawancara / Interview

Teknik pengumpulan data dengan jalan mengadakan komunikasi langsung dengan Masyarakat tentang Psikotes yang telah dilaksanakan

2. Study Literature / Pustaka

Pengumpulan data dilakukan dengan cara mencari bahan-bahan di toko buku sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan obyek penelitian.

1.6 SISTEMATIKA PENULISAN

Untuk mempermudah dan memahami pembahasan skripsi ini, maka peneliti menyajikan secara sistematis sebagai berikut :

Bab I : Pendahuluan

Bab ini merupakan dasar penyusunan laporan Skripsi yang di dalamnya berisi tentang latar belakang masalah, rumusan masalah, tujuan Skripsi, batasan masalah, metodologi pengembangan sistem, dan sistematika pembahasan Skripsi.

Bab II : Landasan Teori

Bab ini berisi tentang permasalahan yang berhubungan dengan penelitian yang dilakukan pada Skripsi ini.

Bab III: Analisis dan Perancangan Sistem

Bab ini berisi tentang tahap analisis yaitu identifikasi dan analisis masalah dan analisis kebutuhan sistem untuk menyelesaikan masalah yang dihadapi. Bab ini juga berisi hasil perancangan yaitu proses kelanjutan dari tahap analisis meliputi proses akusisi pengetahuan.

Bab IV :Implementasi dan Pengujian Sistem

Bab ini berisi tentang implementasi hasil perancangan dan penyesuaian kebutuhan sistem serta meliputi hasil pengujian sistem.

Bab V : Penutup

Bab ini berisi tentang kesimpulan dan saran dari hasil penyusunan laporan Skripsi yang telah disusun.

BAB II

LANDASAN TEORI

2.1. Pengertian Dasar Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum pertama kali diuraikan oleh Kenneth Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem.

Teori sistem mengatakan bahwa setiap unsur pembentuk organisasi adalah penting dan harus mendapat perhatian yang utuh. Unsur atau komponen pembentuk organisasi di sini bukan hanya bagian-bagian yang tampak secara fisik, tetapi juga hal-hal yang mungkin bersifat abstrak atau konseptual, seperti misi, pekerjaan, kegiatan, kelompok informal dan lain sebagainya.

Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (input), pengolahan (processing), dan keluaran (output). Di samping itu suatu sistem dapat pula dikembangkan hingga menyertakan media penyimpanan. Sistem dapat terbuka dan tertutup. Sistem informasi biasanya adalah sistem terbuka, yang berarti menerima berbagai masukan dari lingkungan sekitarnya.

2.1.1. Karakteristik Sistem

Sebuah sistem terdiri atas bagian-bagian yang saling berkaitan dan bervariasi bersama-sama untuk mencapai beberapa sasaran dan maksud. Sebuah sistem bukanlah seperangkat unsur yang tersusun secara teratur, tetapi terdiri atas unsur yang dapat dikenal yang saling melengkapi karena suatu maksud, tujuan dan sasaran.

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu :

a. **Komponen Sistem (*Components*)**

Suatu sistem terdiri atas sejumlah komponen yang saling berinteraksi, yang bekerjasama membentuk suatu kesatuan.

b. **Batas Sistem (*Boundary*)**

Batas sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan.

c. **Lingkungan Luar Sistem (*Environment*)**

Lingkungan luar dari sistem adalah apapun di luar ruang lingkup sistem yang mempengaruhi operasi sistem. Lingkungan luar dapat bersifat menguntungkan dan dapat pula bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

d. Penghubung Sistem (*Interface*)

Interface merupakan media penghubung antara suatu subsistem dengan subsistem lainnya. Penghubung ini memungkinkan sumber daya mengalir dari suatu subsistem ke subsistem lainnya. Keluaran (*Output*) dari suatu subsistem akan menjadi masukan (*Input*) untuk subsistem lainnya dengan melalui penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, "program" adalah maintenance input yang digunakan untuk mengoperasikan komputer sementara "data" adalah signal input yang akan diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

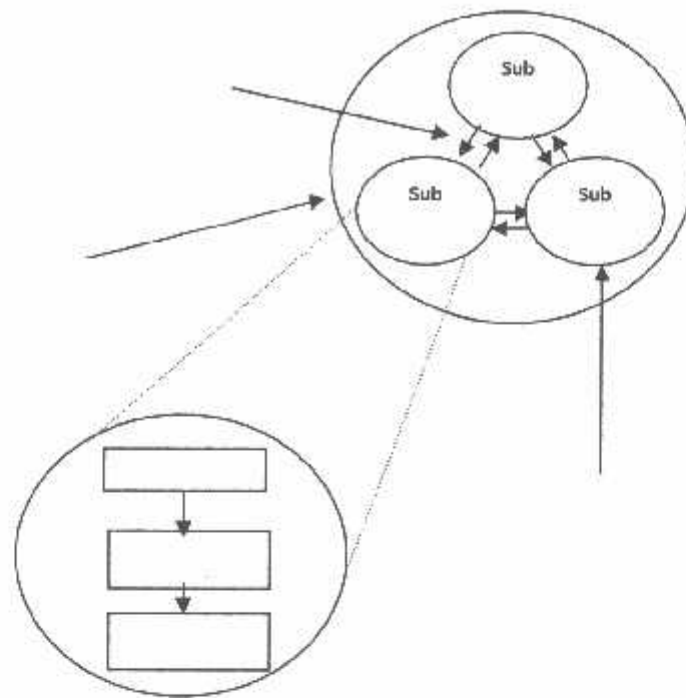
Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan untuk subsistem lain.

g. Pengolah Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objective*)

Suatu sistem mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya



Gambar 2.1 Karakteristik Sistem

Sumber : Sutabri, Tata, *Analisa Sistem Informasi*, Andi, Yogyakarta, 2004

2.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandang, seperti :

a. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan. Sedangkan sistem fisik merupakan sistem yang ada secara fisik, seperti sistem komputer, sistem penjualan, dan lain sebagainya.

b. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang dan malam. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan *human machine system*. Sistem informasi berbasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

c. Sistem deterministik dan sistem probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

d. Sistem terbuka dan sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak dipengaruhi oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

2.2. Sejarah Java

Pada 1991, sekelompok insinyur Sun dipimpin oleh Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat konsumen seperti *cable TV Box*. Karena perangkat tersebut tidak memiliki banyak memori, bahasa harus berukuran kecil dan mengandung kode yang liat. Juga karena manufaktur – manufaktur berbeda memilih *processor* yang berbeda pula, maka bahasa harus bebas dari manufaktur manapun. Proyek diberi nama kode "Green".

Kebutuhan untuk fleksibilitas, kecil, liat dan kode yang netral terhadap *platform* mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta bahasa Pascal telah merancang bahasa portabel yang menghasilkan *intermediate code* untuk mesin hipotesis. Mesin ini sering disebut dengan mesin maya (*virtual machine*). Kode ini kemudian dapat digunakan di sembarang mesin yang memiliki *interpreter*. Proyek Green menggunakan mesin maya untuk mengatasi isu utama tentang netral terhadap arsitektur mesin.

Karena orang – orang di proyek Green berbasis C++ dan bukan Pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan prosedural. Mulanya bahasa yang diciptakan diberi nama "Oak" oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada pada seberang kantornya, namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan JAVA. Nama JAVA sendiri

terinspirasi pada saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal biji kopi. Akhirnya mereka sepakat untuk memberikan nama bahasa pemrograman tersebut dengan nama Java.

Bahasa/Alat pengembangan	Arsitektur Program			
	Modul Web Server	Scripting Web Server	Modul Web Browser	Scripting Web Browser
Java	servlet	JSP	Applet	Javascript
C++	CGI exe		ActiveX*	
Perl	CGI script			
Python	CGI script			
PHP		PHP script		
Visual Basic		ASP*	ActiveX*	VB Script*

*) Hanya di landasan Windows, tidak bisa di Linux.

Gambar 2.2 Arsitektur Java

Java telah mengakomodasi hampir seluruh fitur penting bahasa – bahasa pemrograman yang ada semenjak perkembangan komputasi modern manusia :

1. Dari SIMULA, bahasa pada tahun 65-an, bahasa yang paling mempengaruhi Java sekaligus C++. Dari bahasa ini diadopsi bentukan – bentukan dasar dari pemrograman berorientasi objek.
2. Dari LISP – bahasa tahun 55-an. Diadopsi fasilitas *garbage collection*, serta kemampuan untuk meniru *generic list processing*, meski fasilitas ini jarang yang memanfaatkannya.
3. Dari Algol – bahasa pada tahun 60-an, diambil struktur kendali yang dimilikinya.
4. Dari C++, diadopsi sintaks, sebagian semantiks dan *exception handling*
5. Dari bahasa Ada, diambil *strongly type*, dan *exception handling*.
6. Dari Objective C, diambil fasilitas interface.
7. Dari bahasa SmallTalk, diambil pendekatan *single-root class hiérarchie*, dimana objek adalah satu kesatuan hirarki pewarisan
8. Dari bahasa Eiffel, fasilitas *assertion* yang mulai diterapkan di sebagian JDK 1.4

2.2.1. Sebuah Bahasa Pemrograman

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop*, *web* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

2.2.2. Sebuah Development Environment

Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak *tools* : *compiler*, *interpreter*, penyusun dokumentasi, paket kelas dan sebagainya.

2.2.3. Sebuah Aplikasi Java

Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki *Java Runtime Environment* (JRE).

2.2.4. Deployment Environment

Terdapat dua komponen utama dari *Deployment Environment*. Yang pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas-kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada Web Browser. Hampir seluruh Web Browser komersial menyediakan *interpreter* dan *runtime environment* dari teknologi Java.

2.2.5. Karakteristik Java

Berdasarkan *white paper* resmi dari SUN, Java memiliki karakteristik berikut :

1. Sederhana (*Simple*)

Bahasa pemrograman Java menggunakan Sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

2. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman

berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

3. Terdistribusi (*Distributed*)

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada Java.

4. Interpreted

Program Java dijalankan menggunakan interpreter yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada platform yang berbeda-beda

5. Robust

Java mempunyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi error pada pemrograman.

6. Secure

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

2. Architecture Neutral

Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform berbeda dengan *Java Virtual Machine*.

3. Portable

Source code maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

4. Performance

Performance pada Java sering dikatakan kurang tinggi. Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers* (JIT).

5. Multithreaded

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

6. Dynamic

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

2.2.6. Sebagian Fitur Java

2.2.6.1. Java Virtual Machine (JVM)

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi hardware dan platform dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh JVM.

Aplikasi program Java diciptakan dengan *file* teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

2.2.6.2. Code Security

Code Security terimplementasi pada Java melalui penggunaan Java Runtime Environment (JRE). Java menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*.

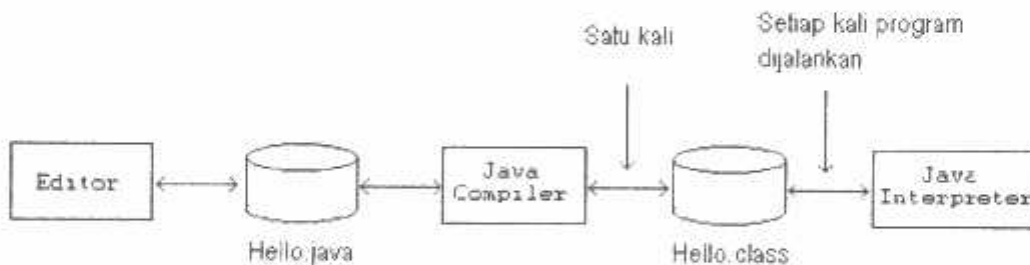
1. Pertama, *class-loader* menangani pemuatan kelas Java ke *runtime interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelas – kelas yang berasal dari *local disk* dengan kelas – kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas – kelas yang berasal dari *local disk* yang dimuat terlebih dahulu.
2. Kedua, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan – aturan dasar bahasa Java.
3. Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, *port* jaringan, proses eksternal dan sistem *windowing*.

Setelah seluruh proses tersebut selesai dijalankan, barulah kode program di eksekusi. java juga menyediakan beragam teknik pengamanan lain :

1. Bahasa dirancang untuk mempersulit eksekusi kode perusak. Peniadaan *pointer* merupakan langkah besar pengamanan. Java tidak mengenal operasi *pointer*. Di tangan pemrogram handal, operasi *pointer* merupakan hal yang luar biasa untuk optimasi dan pembuatan program yang efisien serta mengagumkan. Namun mode ini dapat menjadi petaka di hadapan pemrogram jahat. *Pointer* merupakan sarana luar biasa untuk pengaksesan tak diotorisasi. Dengan peniadaan operasi *pointer*, Java dapat menjadi bahasa yang lebih aman.
2. Java memiliki beberapa pengamanan terhadap *applet*. Untuk mencegah program bertindak mengganggu media penyimpanan, maka *applet* tidak diperbolehkan melakukan *open*, *read* ataupun *write* terhadap berkas secara sembarangan. Karena Java *applet* dapat membuka jendela *browser* yang baru, maka jendela mempunyai logo Java dan teks identifikasi terhadap jendela yang dibuka. Hal ini mencegah jendela *pop-up* menipu sebagai permintaan keterangan *username* dan *password*.

2.2.6.3. Fase – Fase Pemrograman Java

Gambar dibawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program Java :



Gambar 2.3 Fase – Fase Java

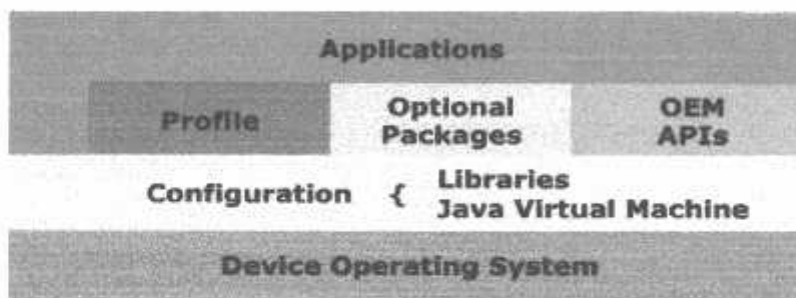
2.3. J2me

J2ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan bandwidth jaringan yang rendah. dengan perkembangbiakan perangkat mobile konsumen dari telepon, PDA, kotak permainan ke

peralatan-peralatan rumah, Java menyediakan suatu lingkungan yang portable untuk mengembangkan dan menjalankan aplikasi pada perangkat ini.

Program J2ME, seperti semua program JAVA adalah diterjemahkan oleh VM. Program-program tersebut dikompil ke dalam bytecode dan diterjemahkan dengan Java Virtual Machine(JVM).Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat.

J2ME menyediakan suatu interface yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus dikompil ulang supaya mampu dijalankan pada mesin yang berbeda. Inti dari J2ME terletak pada configuration dan profile-profile. Suatu configuration menggambarkan lingkungan runtime dasar dari suatu sistem J2ME. Ia menggambarkan core library, virtual machine, fitur keamanan dan jaringan.



Gambar 2.4 Sistem J2me

Sebuah profile memberikan library tambahan untuk suatu kelas tertentu pada sebuah perangkat. profile-profile menyediakan user interface(UI) API, persistence, messaging library, dan sebagainya. Satu set library tambahan atau package tambahan menyediakan kemampuan program tambahan. Pemasukan package ini ke dalam perangkat J2ME dapat berubah-ubah karena tergantung pada kemampuan sebuah perangkat. Sebagai contoh, beberapa perangkat MIDP tidak memiliki Bluetooth built-in, sehingga Bluetooth API tidak disediakan dalam perangkat ini.

2.4. Data Flowchat

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.


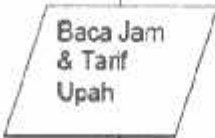

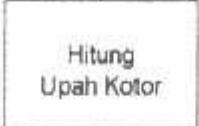

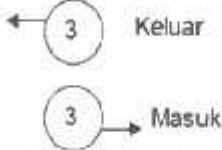

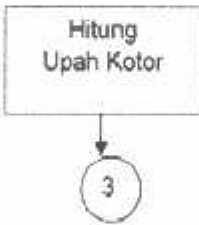



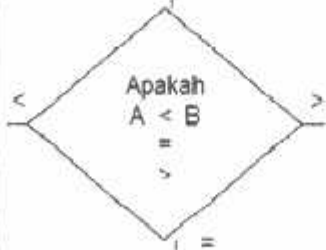
2.4.1. Pedoman - Pedoman Dalam Membuat Flowchart



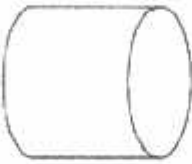
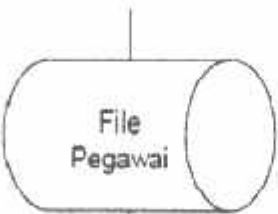
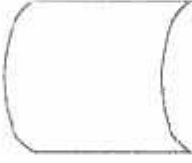

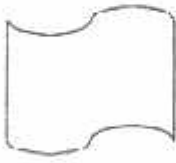



Bila seorang analis dan programmer akan membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, seperti :







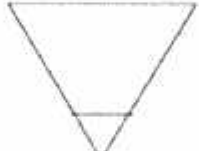
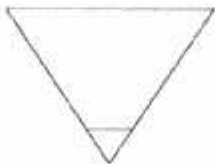
- a. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
 - b. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
 - c. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
 - d. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan MENGHITUNG PAJAK PENJUALAN.
 - e. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
 - f. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
 - g. Gunakan simbol-simbol flowchart yang standar.
-

2.4.2. Simbol-Simbol Flowchart

Simbol-simbol flowchart yang biasanya dipakai adalah simbol-simbol flowchart standar yang dikeluarkan oleh ANSI dan ISO.

SIMBOL	ARTI	CONTOH
Input / Output 	Merepresentasikan Input data atau Output data yang diproses atau Informasi.	
Proses 	Mempresentasikan operasi	
Penghubung 	Keluar ke atau masuk dari bagian lain flowchart khususnya halaman yang sama	
Anak Panah 	Merepresentasikan alur kerja	
Penjelasan 	Digunakan untuk komentar tambahan	
Keputusan 	Keputusan dalam program	

SIMBOL	ARTI	CONTOH
<p>Magnetic Disk</p> 	<p>I/O yang menggunakan disk magnetik</p>	
<p>Magnetic Drum</p> 	<p>I/O yang menggunakan drum magnetik</p>	
<p>On-line Storage</p> 	<p>I/O yang menggunakan penyimpanan akses langsung</p>	
<p>Punched Tape</p> 	<p>I/O yang menggunakan pita kertas berlubang</p>	
<p>Manual Input</p> 	<p>Input yang dimasukkan secara manual dari keyboard</p>	

SIMBOL	ARTI	CONTOH
Display 	Output yang ditampilkan pada terminal	
Manual Operation 	Operasi Manual	
Communication Link 	Transmisi data melalui channel komunikasi, seperti telepon	
Off-line Storage 	Penyimpanan yang tidak dapat diakses oleh komputer secara langsung	

Gambar 2.5 simbol flowchart

2.5. PSIKOTES

Psikotes adalah tes yang dilakukan untuk mengukur aspek individu secara psikis. *Tes ini dapat berbentuk tertulis, proyektif, atau evaluasi secara verbal yang teradministrasi untuk mengukur fungsi atau kemampuan kognitif dan emosional seseorang.*

Tujuan dari dilaksanakannya tes ini adalah untuk mengukur berbagai kemungkinan atas bermacam kemampuan orang secara mental dan factor-faktor yang mendukungnya, termasuk prestasi dan kemampuan, kepribadian, dan intelegensi. Jadi sederhananya, psikotes adalah serangkaian kegiatan yang dilakukan untuk mengetahui gambaran seseorang mulai dari kemampuan kognitifnya, kondisi emosinya, kecenderungan-kecenderungan sikap dan hal-hal yang mempengaruhi kecenderungan

tersebut. Jadi dalam psikotes, kemampuan yang diukur tidak melulu terkait dengan IQ seseorang. Selain tes IQ ada juga tes kepribadian, dan wawancara.

Dari integrasi tes-tes tersebut, maka akan diperoleh gambaran mengenai orang yang di tes yang kemudian disesuaikan dengan kebutuhan perusahaan.

2.5.1. Macam – Macam Psikotes

1. Tes Kemampuan Verbal – Antonim (Lawan kata)

Tes kemampuan verbal – antonim (lawan kata) bertujuan untuk melihat kebenaran secara terbalik, sekaligus melihat wawasan seseorang. Yang dimaksud kebenaran secara terbalik, bahwa seseorang mengetahui sesuatunya benar atau salah tidak hanya secara fenomenologis, tetapi dapat juga secara dialektis,

2. Tes Kemampuan Verbal – Sinonim (Padaan Kata)

Tes kemampuan verbal – sinonim persamaan makna bertujuan untuk mengukur tingkat kewaspadaan dan kecermatan terhadap suatu indikasi yang sama, sekaligus mengukur wawasannya, salah satunya, bahwa seseorang akan lebih cepat, lebih efektif, dan lebih efisien dalam mengambil keputusan, ketika dihadapkan suatu permasalahan yang memiliki prinsip serta tipe yang sama dengan permasalahan yang pernah dihadapinya,

3. Tes Analogi Verbal – Korelasi Makna

Tes analogi verbal – korelasi makna bertujuan untuk melihat pemahaman anda terhadap hubungan antar kata. Dampak positifnya adalah kemampuan memahami permasalahan. Pada tes ini diminta kemampuan mengerti akan arti kata, fungsi kata, dan pemakaian serta padanan fungsi dengan kata lain,

4. Tes Kemampuan Kuantitatif – Deret Angka Atau Huruf

Tes kemampuan kuantitatif – deret angka atau huruf ini bertujuan untuk mengukur tingkat kecerdasan, kecermatan, sekaligus ketelitian seseorang, dalam memandang permasalahan secara terpadu, sistematis dan menyeluruh dari berbagai sisi.

5. Tes Kemampuan Spasial – Logika Gambar

Tes Kemampuan Spasial – Logika Gambar bertujuan untuk ketelitian dan kecerdasan, dengan apa yang akan dilakukan, dalam pekerjaan yang dibebankan,

2.6. RMS

RMS merupakan implementasi database sederhana digunakan dalam MIDlet Java yang berorientasi record. Jika memanfaatkan fasilitas RMS ini, kita harus mengimport paket yang menyediakan fungsi-fungsi untuk manajemen RMS ini, yaitu `javax.microedition.rms.*`. Data yang ada di dalam database RMS ini dikenal dengan `RecordStore`. Fungsi-fungsi dalam paket `javax.microedition.rms.*` menyediakan fungsi-fungsi untuk menghapus, menambah, mengubah data dan memungkinkan dua atau lebih MIDlet dalam satu aplikasi untuk berbagi data `RecordStore`. `RecordStore` dapat memiliki nama unik dengan batasan 32 karakter dan bersifat case sensitive. Dalam satu project MIDlet kita tidak bisa membuat dua atau lebih `RecordStore`.

Perangkat bergerak, sama halnya dengan PC, memiliki media untuk menyimpan program dan datanya. Media yang dimaksud adalah memori, baik yang bersifat volatile maupun non-volatile. Sehubungan dengan kebutuhan untuk menyimpan dan mengelola data di perangkat bergerak, platform J2ME menyediakan Record Management System (RMS). Kemampuan RMS di platform J2ME serupa dengan kemampuan JDBC di platform J2SE atau J2EE, tentunya dengan keterbatasan dalam lingkungan yang memiliki sumber daya rendah. Dengan adanya RMS, data aplikasi tidak perlu selalu disimpan di database server melalui koneksi Internet

Informasi yang disimpan di record store tergantung pada aplikasi yang dibuat. Ketika berkerja dengan `RecordStore`, pembuat aplikasi perlu memperhatikan limitasi RMS.

- RMS menyimpan record dalam format byte array (`byte[]`).
 - RMS memberikan id yang unik untuk setiap record. Pada saat data disimpan di dalam obyek `RecordStore`, setiap record mendapatkan id yang unik. Id yang diberikan dimulai 1 (satu) dan id tersebut tidak dapat digunakan kembali. Hal ini berarti setelah record dihapus dari obyek `RecordStore`, id yang telah diisi untuk record tersebut tidak lagi valid untuk digunakan.
-

BAB III ANALISA DAN PERANCANGAN SISTEM

3.1. Gambaran Umum Psikotest

Diera global ini persaingan sekarang ini, diperlukan kemampuan akademik (psikotes) yang mumpuni untuk memenangkan persaingan kerja yang diperebutkan oleh banyak lulusan. Banyak lulusan dengan beragam kemampuan dengan ip-nya yang tinggi. Di situlah kejelian diperlukan. Ip tinggi saja tidak cukup, tetapi kemampuan mengerjakan TPA (tes potensi akademik) sangat diperlukan. Karena perusahaan – perusahaan besar dan menengah akan menjaring SDMnya lewat TPA. Karena pintu masuk memasuki dunia kerja baik dipemerintahan maupun swasta salah satunya adalah kemampuan TPAnya. Oleh karena itu keterampilan mengerjakan TPA hukumnya menjadi wajib. Dengan kemampuan menyelesaikan TPA yang cerdas, tepat, dan cepat menjadi salah satu kunci keberhasilan.

Aplikasi psikotes ini memberikan solusi bagi keberhasilan anda, dengan rajin latihan dan menyelesaikan aplikasi psikotes, pintu kesuksesan semakin terbuka

3.1.1. Gambaran Umum Program

Pembuatan program aplikasi ini menggunakan java. Dimana unit digunakan sebagai penghubung antara *handphone* dengan komputer yang terkoneksi. skripsi ini mempunyai tujuan memudahkan masyarakat untuk mendapatkan latihan tentang psikotes.

3.1.2 Algoritma Program

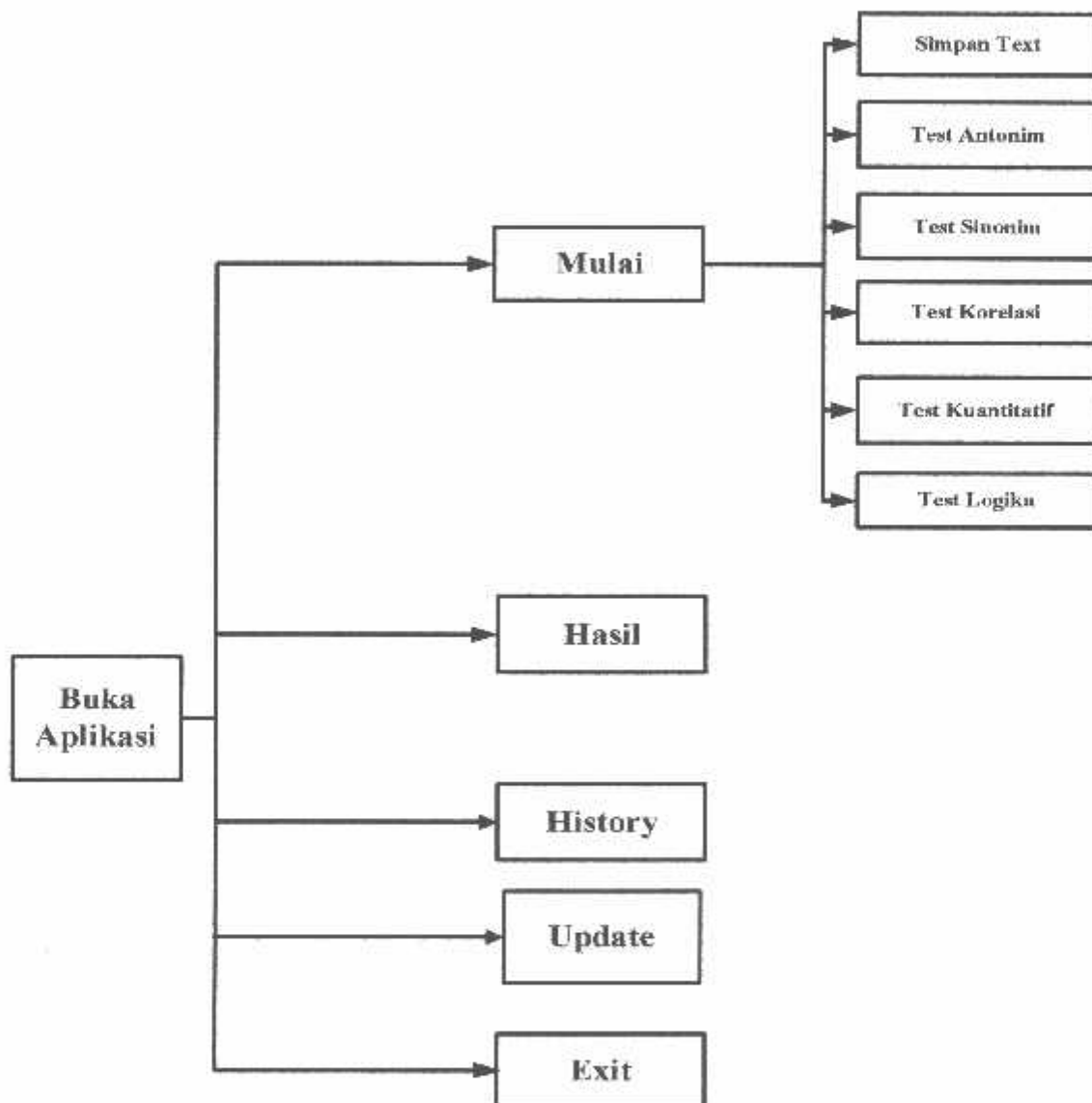
Program untuk menjalankan aplikasi psikotes dengan menggunakan java, program java yang sudah berbentuk aplikasi akan dikirim ke *handphone* melalui kabel data menurut tipe *handphone* yang sudah disediakan, dimana data dari komputer yang berupa program dengan format tertentu dipindah ke *handphone* untuk dijalankan dalam bentuk jar dan lalu diinstal di *handphone*, Jika penginstalan sukses maka program siap dijalankan, didalam aplikasi ini disediakan beberapa soal – soal yang sudah disiapkan, sitem aplikasi ini adalah untuk menseleksi masyarakat yang ingin mendaftar pekerjaan, didalam *handphone*, aplikasi ini juga disediakan beberapa tombol,

Tombol mulai test, kegunaan tombol ini adalah untuk menampilkan soal – soal yang ada pada aplikasi psikotes. Tombol hasil, kegunaan dari tombol ini adalah untuk menampilkan score dari hasil tes soal – soal yang sudah dikerjakan, Tombol history,

tombol ini berguna untuk kapan hari tanggal dan waktu kita terakhir kali menggunakan aplikasi ini. Tombol update kegunaan dari tombol ini adalah untuk memperbarui soal – soal yang sudah ada dengan cara mengupdate disitus www.kholisin.com, Tombol exit, untuk mengakhiri aplikasi.

Didalam aplikasi ini, juga ada tombol menu, dimana tombol menu tersebut mencakup beberapa tombol, tombol simpan, untuk menyimpan data aplikasi yang sedang berjalan dan 5 tombol lain untuk memilih soal psikotes.

Dibawah ini adalah gambaran dari design menu yang ada pada aplikasi psikotes berbasis java:



Gambar 3.1. Design Menu Aplikasi Psikotes Berbasis Java

3.2. Analisa Sistem

Agar program dapat dijalankan sesuai dengan tujuannya maka harus menyiapkan beberapa hal yang dibutuhkan dan pengaksesan data yaitu sebagai berikut:

a. JAVA

JAVA adalah perangkat lunak yang sangat penting dalam perancangan aplikasi ini, sesuai dengan prinsip tulis sekali dan dijalankan dimana saja, dengan kelebihan ini pemrograman cukup menulis sebuah program java dan dikompilasi (diubah dari bahasa manusia ke bahasa mesin/bytecode) sekali, lalu hasilnya dapat dijalankan diatas platform tanpa perubahan, jika aplikasi ini ingin digunakan dalam handphone maka handphone tersebut harus berbasis java,

b. RMS

RMS dalam aplikasi ini berfungsi sebagai implementasi database RecordStore adalah data yang ada didalam database RMS aplikasi ini

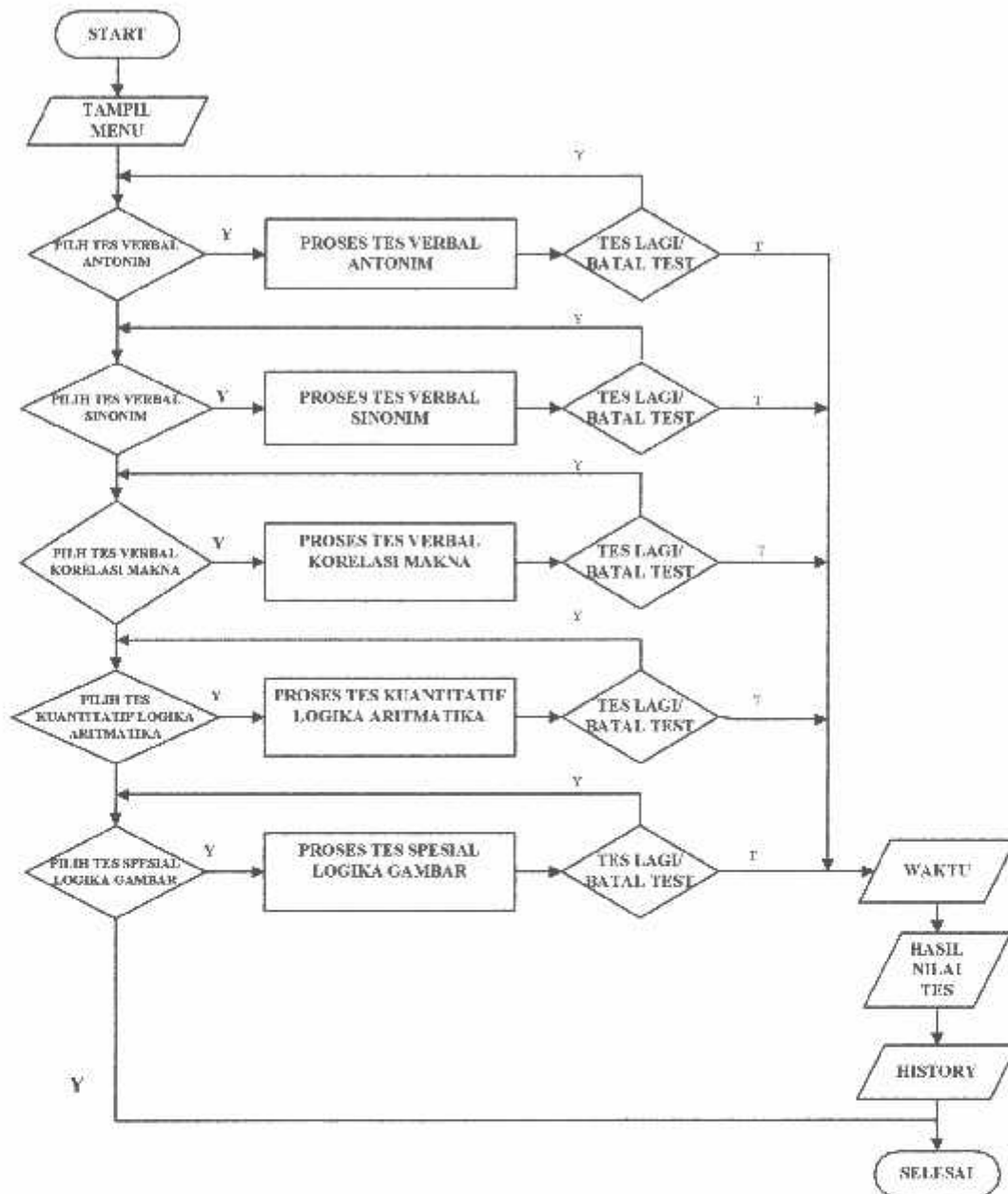
3.3. Perencanaan Perangkat Keras

Perangkat keras yang digunakan dalam sistem ini adalah perangkat keras yang mendukung proses aplikasi yang dibuat

1. Perangkat telepon seluler yang berfungsi untuk mengambil data aplikasi yang sudah jadi pada computer, Dalam Skripsi ini digunakan *handphone* Samsung type C6625 valencia sebagai devicenya.
 2. Perangkat komputer dengan system operasi windows *XP service pack 2*.
 3. Satu kabel data samsung, sebagai penghubung antara *handphone* dengan PC.
-

3.4 Perancangan Sistem

3.4.1 flowchart



Gambar 3.2 Flowchart Aplikasi Psikotes

3.4.2 Penjelasan Flowchart:

Start

Memulai aplikasi ini

Tampil Menu

Menu yang ada pada aplikasi ini akan muncul jika pengguna sudah memulai aplikasi psikotes, ada 5 menu yang ada dalam aplikasi ini, yaitu:

- a. Test Antonim
- b. Test Sinonim
- c. Test Korelasi Makna
- d. Test Kuantitatif
- e. Test Logikan Gambar

Jika pengguna hanya memilih 1 menu dari ke 5 menu diatas maka pengguna dapat membatalkan aplikasi psikotes ini atau lanjut ke menu selanjutnya.

Waktu

Waktu dalam aplikasi psikotes ini adalah 50 menit

Hasil Nilai Tes

Hasil dari aplikasi psikotes akan muncul jika pengguna sudah selesai mengerjakan soal – soal yang ada pada aplikasi ini.

History

History akan menampilkan waktu terakhir aplikasi ini digunakan.

Selesai

Jika pengguna sudah memainkan aplikasi ini maka hasilnya bisa diketahui, dan aplikasi bisa di exit.

Dalam flowchart diatas bahwa setiap pengguna harus memilih bagian mana yang akan dikerjakan dengan dibatasi waktu. Diaplikasi psikotes ini para pengguna dapat melihat waktu terakhir mereka menyelesaikan aplikasi ini karena didalam menu aplikasi ini ada tombol history, para pengguna juga dapat langsung dapat memilih tes apa yang akan diselesaikan.

3.5 Desain Antarmuka Aplikasi

Sesuai dengan spesifikasi sistem di atas, aplikasi psikotes ini diharapkan dapat digunakan dengan mudah oleh pengguna yang mengoperasikan aplikasi. Untuk itu dibuat desain antarmuka yang tidak terlalu rumit dan semenarik mungkin agar mudah dipahami oleh pengguna.

3.5.1 Desain Form Utama

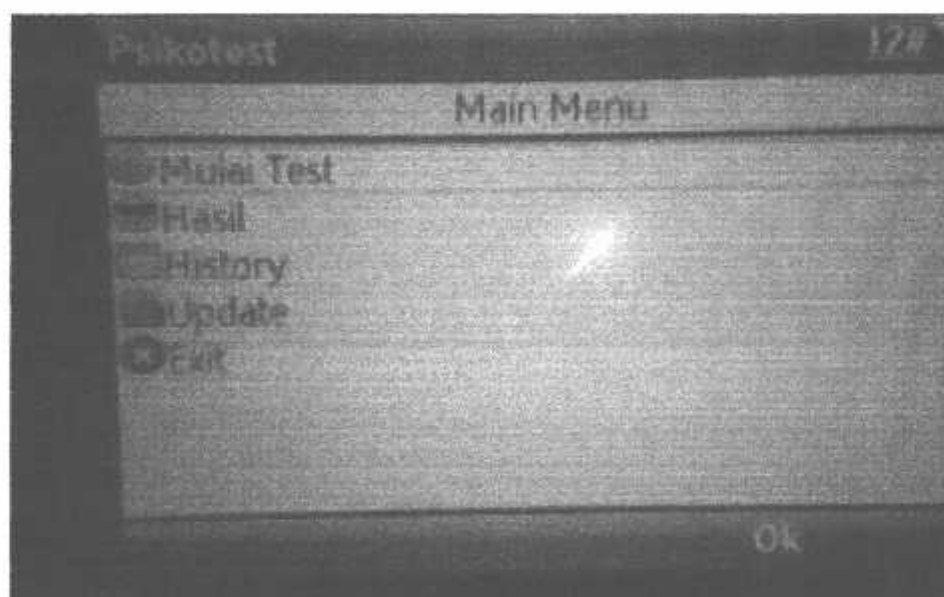
Form Utama berisikan tampilan utama dapat diakses oleh pengguna dan tampilan disesuaikan dengan tujuan pembuatan aplikasi Psikotes. Berikut desain form utama ditunjukkan pada gambar dibawah ini.



Gambar 3.3. Desain Form utama

3.5.2 Desain Form Tampilan Main Menu

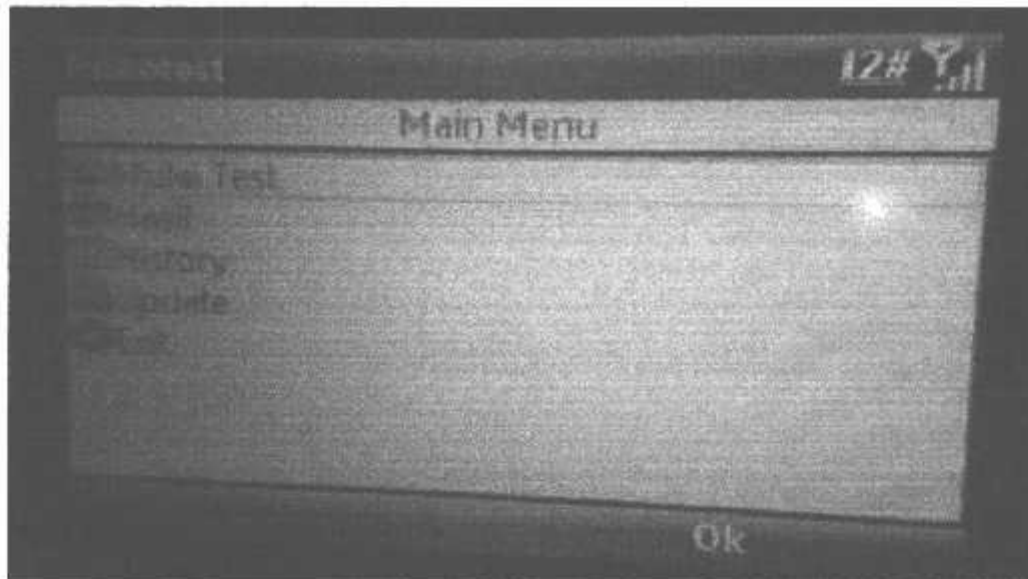
Form main menu ini terdiri atas menu utama yang berbentuk tegak (*vertikal*).



Gambar 3.4. Desain Form Tampilan Main Menu

3.5.3 Desain Form Main Menu

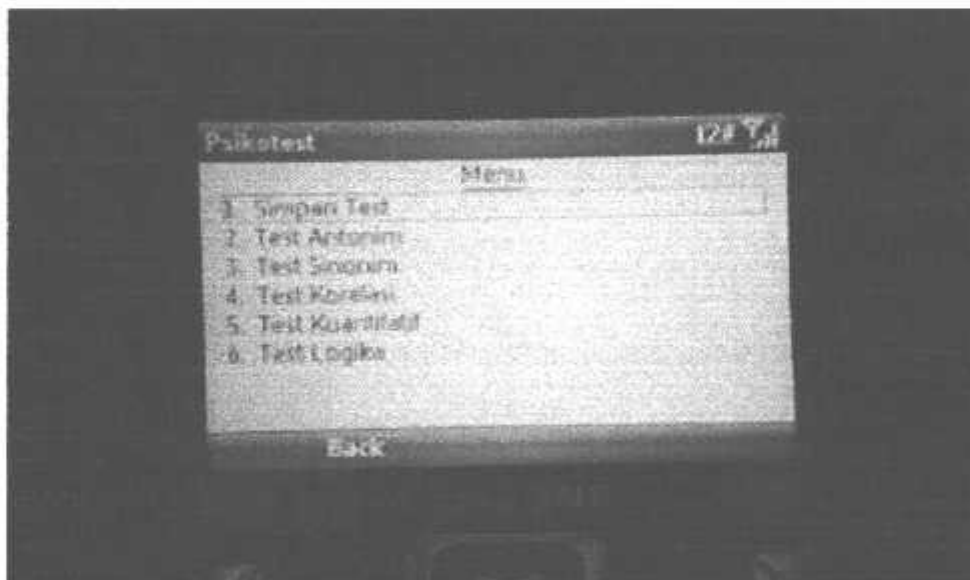
Form menu ini adalah menu yang berbentuk tegak (*vertikal*) terdiri dari bagian utama menu dalam aplikasi yang menyediakan beberapa pilihan tampilan yang terdiri dari mulai test, hasil, history, dan exit. Tampilan tersebut di desain dengan tujuan mempermudah pengguna memahami aplikasi ini



Gambar 3.5. Desain Form Main Menu

3.5.4 Desain Form Menu

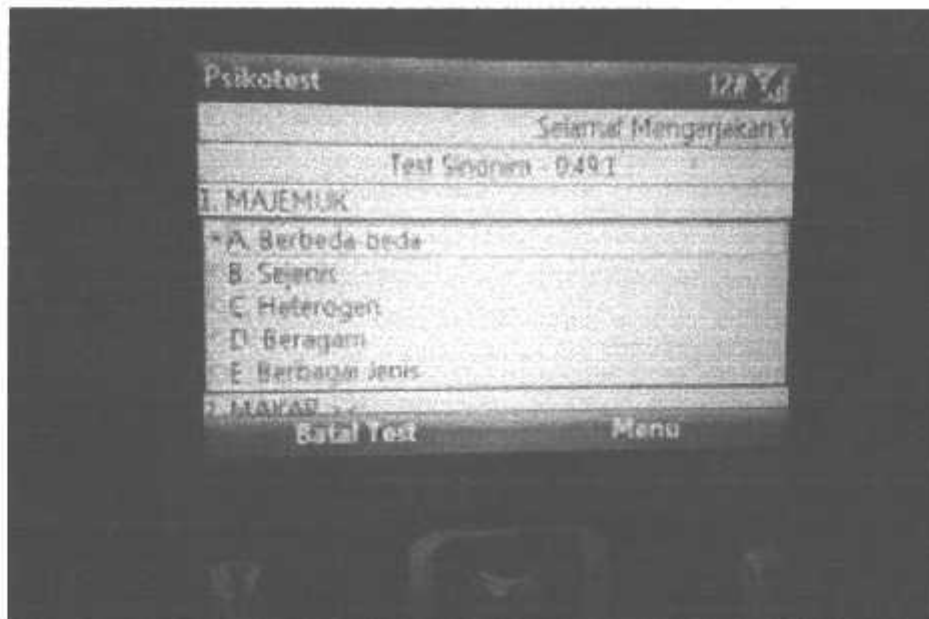
Form menu ini dipakai disaat aplikasi dimulai, menu ini berbentuk tegak (vertikal) yang akan ditampilkan jika kita mengklik menu kanan bawah di tampilan form disaat aplikasi dimulai, menu ini terdiri dari bagian simpan test, test antonim, test sinonim, dan lain - lain



Gambar 3.6. Desain Form Menu

3.5.5 Desain Form Tampilan

Form tampilan ini berisi dari hasil proses pemilihan tampilan mulai test Dibuat sedemikian rupa dengan tujuan mempermudah pengguna memahami aplikasi ini.



Gambar 3.7. Desain Form Tampilan

Demikianlah perancangan dan pembuatan Psikotes Berbasis Java, aplikasi ini dibuat untuk mempermudah dan membuat nyaman Masyarakat dalam mempelajari psikotes.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan menjelaskan tentang implementasi dan pengujian terhadap kinerja aplikasi yang telah dibuat. Dengan demikian akan diketahui tingkat kekurangan dan tingkat kelebihan dari sistem atau aplikasi yang telah dibuat dalam program aplikasi psikotes ini.

4.1 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan untuk menjalankan sistem ini adalah :

1. Microsoft Windows XP Service Pack 2.
2. JAVA (TM) ME Platform SDK 3.0

4.1.1 Kebutuhan Perangkat Keras

Konfigurasi minimum perangkat keras yang digunakan adalah :

1. Perangkat telepon seluler beserta nomor telepon seluler yang berfungsi untuk mengambil data yang sudah berbentuk jar dengan cara dikoneksikan ke PC . Dalam Skripsi ini digunakan *handphone* samsung C6625 Valencia sebagai devicenya.
2. Perangkat komputer dengan system operasi windows *XP service pack2*.
3. Satu kabel data samsung, sebagai penghubung antara *handphone* dengan PC.

4.1.2 Setup Program

1. Instal Microsoft Windows Xp Service Park 2
2. Jalankan Java (TM) ME Platform SDK 3.0

4.2 Implementasi Sistem

Tahap implementasi pengembangan perangkat lunak merupakan proses pengubahan spesifikasi sistem menjadi sistem yang dapat dijalankan. Tahap ini merupakan lanjutan dari proses perancangan, yaitu proses pemrograman perangkat lunak sesuai dengan spesifikasi dan desain sistem.

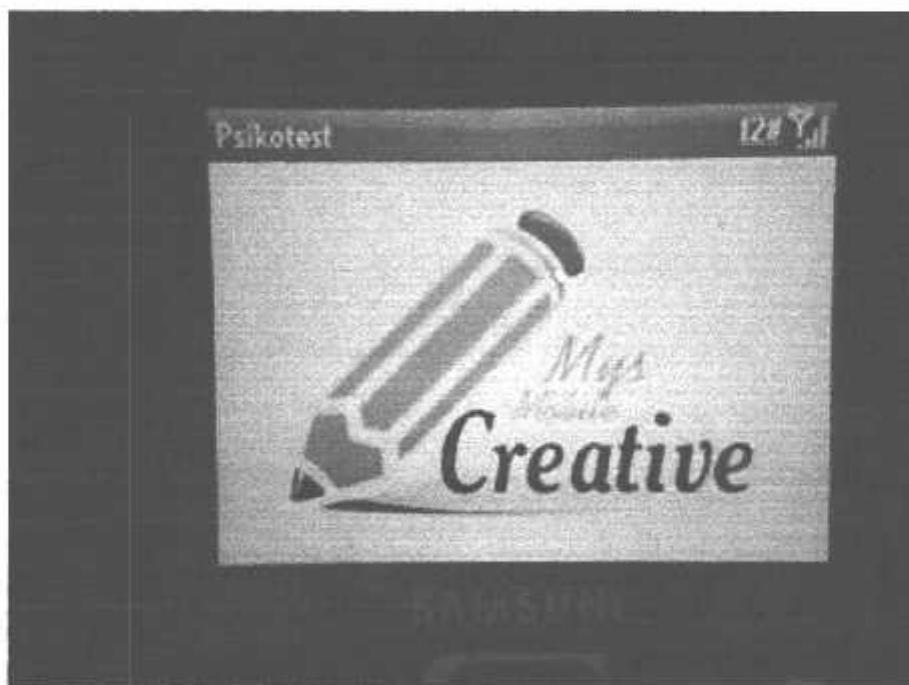
Dalam hal ini dapat di lihat cara kerja program lebih detail tentang layanan yang disediakan. Sebelum menjalankan program ini perlu melakukan koneksi *handphone* ke PC dan menjalankan aplikasi berbasis java.

4.3 Pengujian Hasil

Dalam pengujian hasil skripsi ini, penulis menampilkan hasil tampilan dari tampilan aplikasi psikotes yang berhasil dijalankan dan bagaimana aplikasi psikotes ini digunakan. Adapun tampilan pengujian hasil ini sebagai berikut.

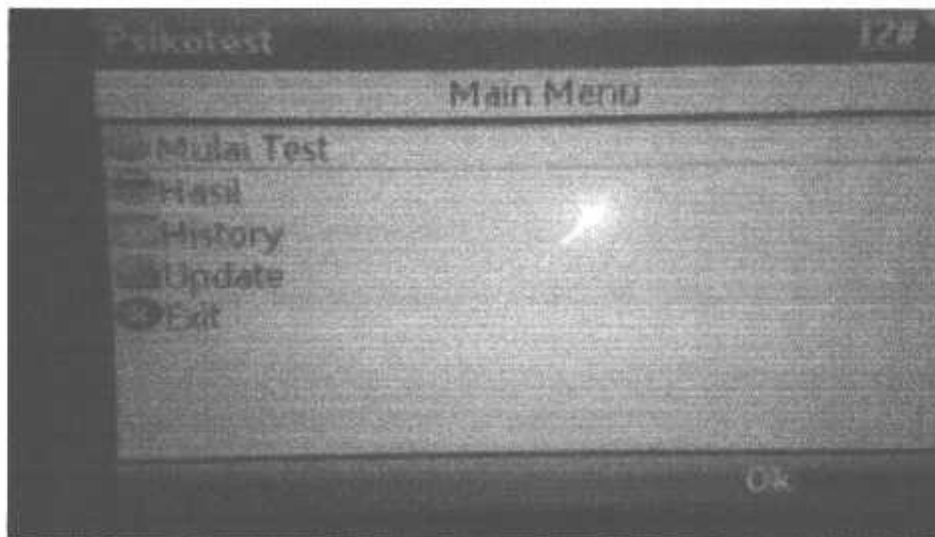
4.3.1 Tampilan Main Menu

Tampilan gambar dibawah ini adalah tampilan utama dan tampilan main menu didalam main menu, tampilan ini mempunyai beberapa pilihan menu dimana pengguna dapat memilih salah satu dari menu yang akan dipilih, dimenu ini ada 4 menu, yaitu : Mulai Test, Hasil, History, Update dan Exit, menu ini dapat kita lihat pada gambar dibawah.



Gambar 4.1 Tampilan Utama

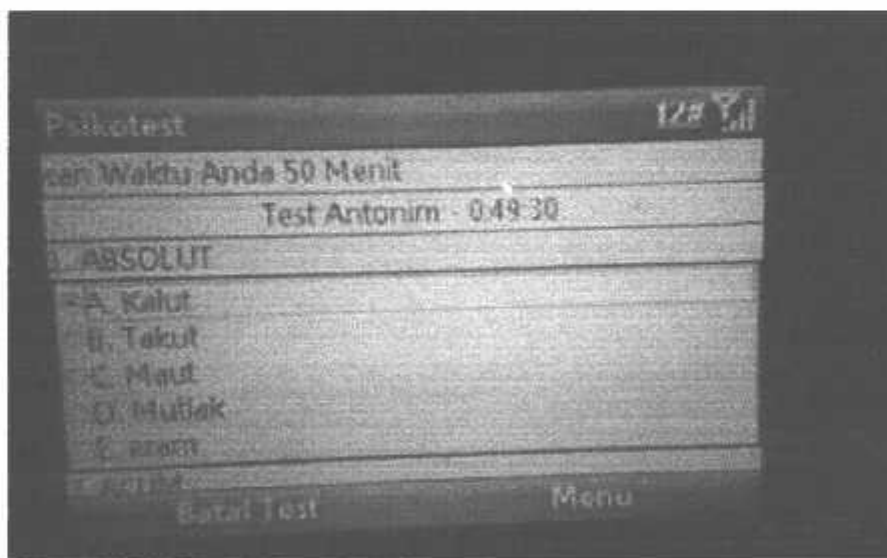
Gambar dibawah ini adalah tampilan main menu



Gambar 4.2 Tampilan Main Menu

4.3.2 Tampilan Mulai Test

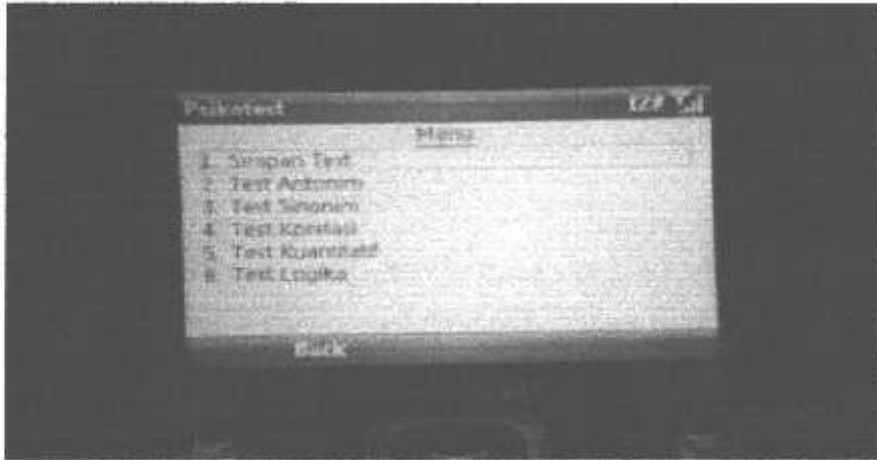
Didalam tampilan mulai test kita akan memulai aplikasi psikotes, dimana dalam pengguna ini kita diberi waktu 50 menit untuk menyelesaikan aplikasi psikotes jika kita melebihi batas dari waktu yang sudah disediakan, maka kita akan dinyatakan gagal dalam aplikasi ini, dan dibawah ini tampilan menunya.



Gambar 4.3 Tampilan Mulai Test

4.3.3 Tampilan Menu

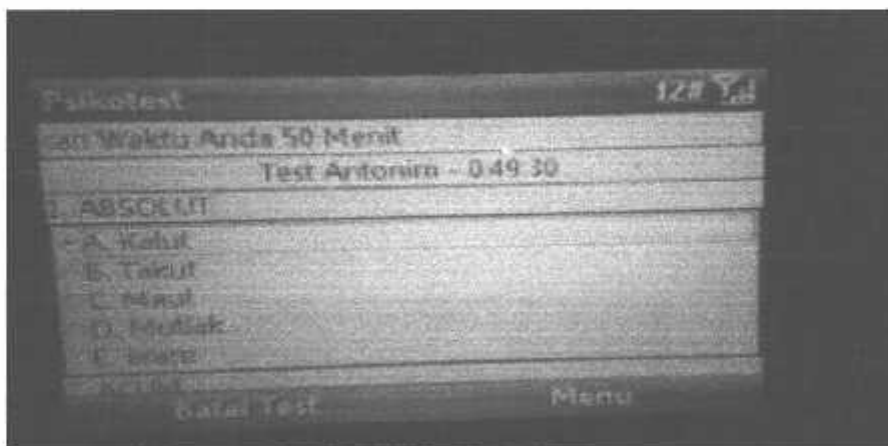
Didalam tampilan menu ini, ada 6 pilihan yaitu simpan test, test antonim, test sinonim, test korelasi, test kuantitatif dan test logika, berikut ini adalah tampilan menunya.



Gambar 4.4 Tampilan Menu

4.3.4 Tampilan Menu Test Antonim

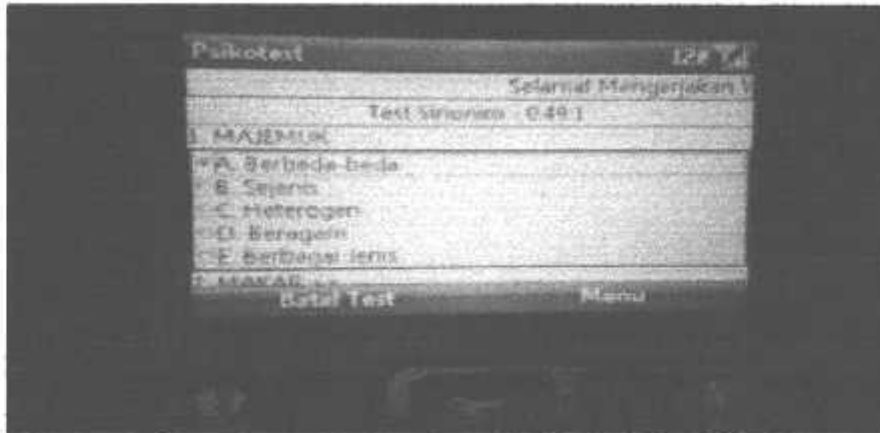
Pada tampilan menu Test antonim, terdapat 10 soal yang harus diselesaikan pengguna aplikasi, Tes kemampuan antonim (lawan kata) bertujuan untuk melihat kebenaran secara terbalik, berikut ini adalah gambar tampilan test antonim..



Gambar 4.5 Tampilan Menu Test Antonim

4.3.5 Tampilan Menu Test Sinonim

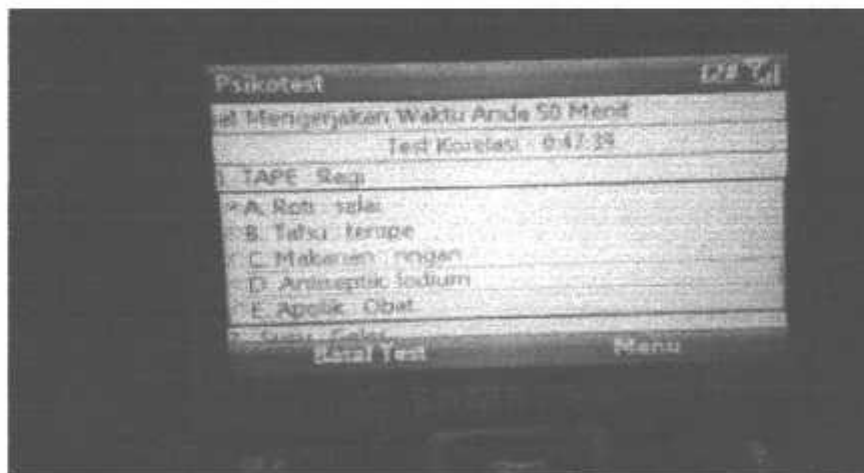
Pada tampilan menu test sinonim, terdapat 10 soal yang harus diselesaikan pengguna aplikasi, Tes kemampuan sinonim persamaan makna bertujuan untuk mengukur tingkat kewaspadaan dan kecermatan terhadap suatu indikasi yang sama, berikut ini adalah tampilan gambar test sinonim.



Gambar 4.6 Tampilan Menu Test Sinonim

4.3.6 Tampilan Menu Test Korelasi

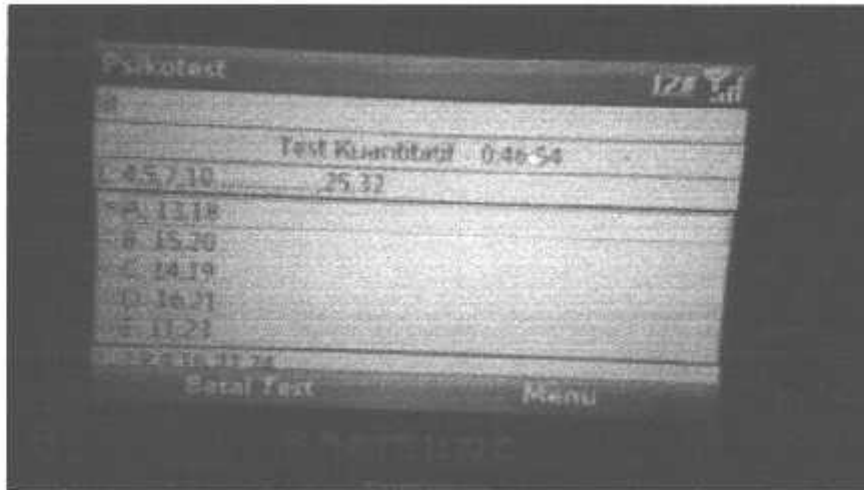
Pada tampilan menu test korelasi, terdapat 10 soal yang harus diselesaikan pengguna aplikasi, Tes korelasi bertujuan untuk melihat pemahaman anda terhadap hubungan antar kata, berikut ini adalah tampilan gambar test korelasi.



Gambar 4.7 Tampilan Menu Test Korelasi

4.3.7 Tampilan Menu Test Kuantitatif

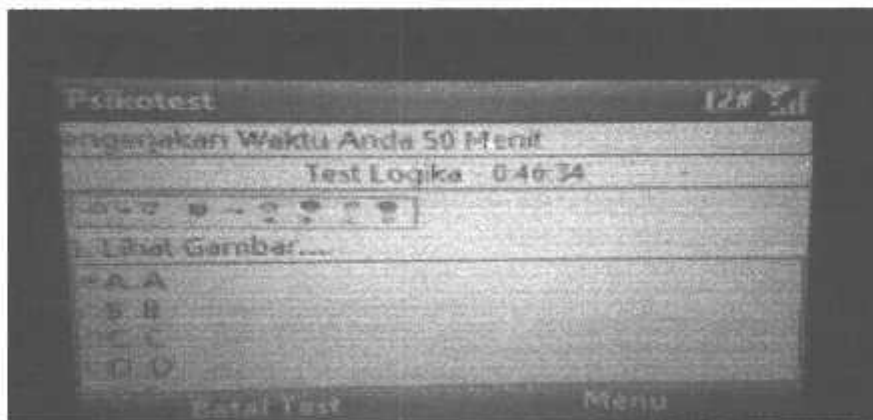
Pada tampilan menu test kuantitatif, terdapat 10 soal yang harus diselesaikan pengguna aplikasi, Tes kemampuan kuantitatif ini bertujuan untuk mengukur tingkat kecerdasan, kecermatan, berikut ini adalah tampilan gambar test kuantitatif.



Gambar 4.8 Tampilan Menu Test Kuantitatif

4.3.8 Tampilan Menu Test logika

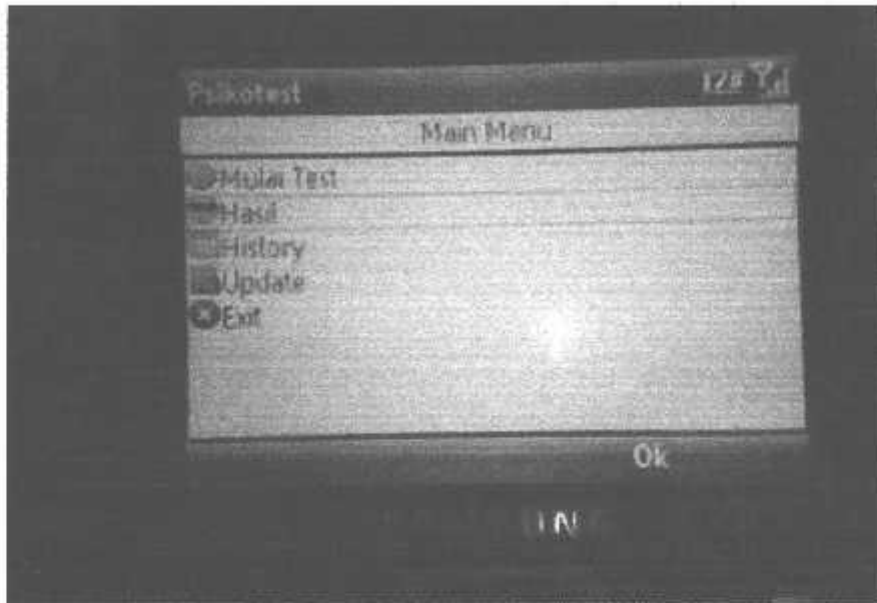
Pada tampilan menu test logika, terdapat 10 soal yang harus diselesaikan pengguna aplikasi, Tes Kemampuan Logika bertujuan untuk ketelitian dan kecerdasan, kecermatan berikut ini adalah tampilan gambar test logika.



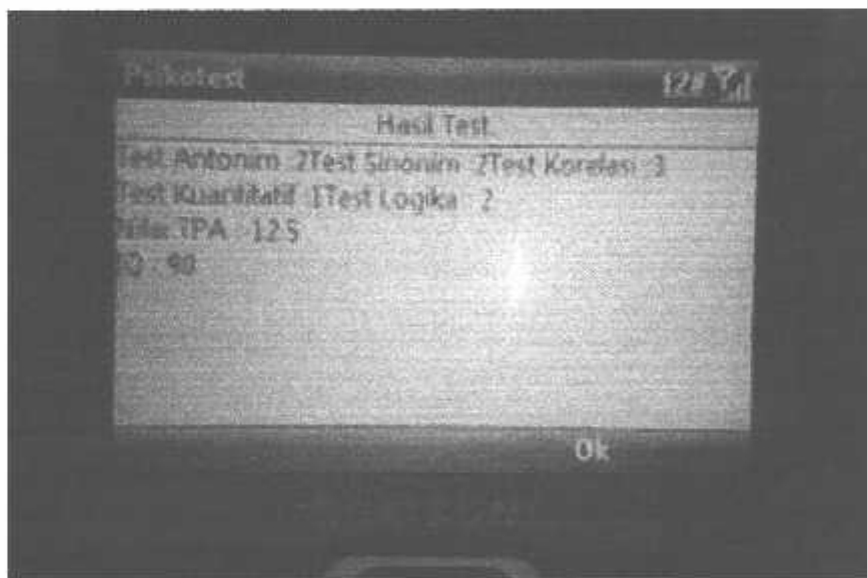
Gambar 4.9 Tampilan Menu Test Logika

4.3.9 Tampilan Menu Hasil

Tampilan menu hasil ini adalah tampilan dimana disaat pemain sudah menyelesaikan pengguna aplikasi, tampilan menu ini akan menampilkan berapa hasil yang sudah kita dapatkan dalam permainan ini, menu ini dapat dilihat pada gambar dibawah ini.



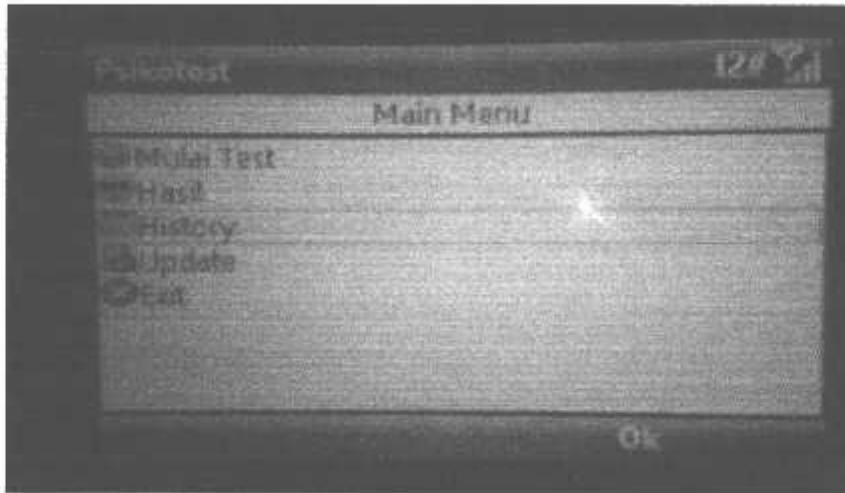
Gambar 4.10 Menu Hasil
Gambar dibawah ini bagian dari tombol hasil



Gambar 4.11 Tampilan Menu Hasil.

4.3.10 Tampilan Menu History

Tampilan menu history ini adalah tampilan menu jam, bulan, tanggal, dan tahun, dimana tampilan ini memperlakukan kita, kapan terakhir kali kita menggunakan aplikasi ini, menu ini kita dapat lihat pada tampilan dibawah ini.



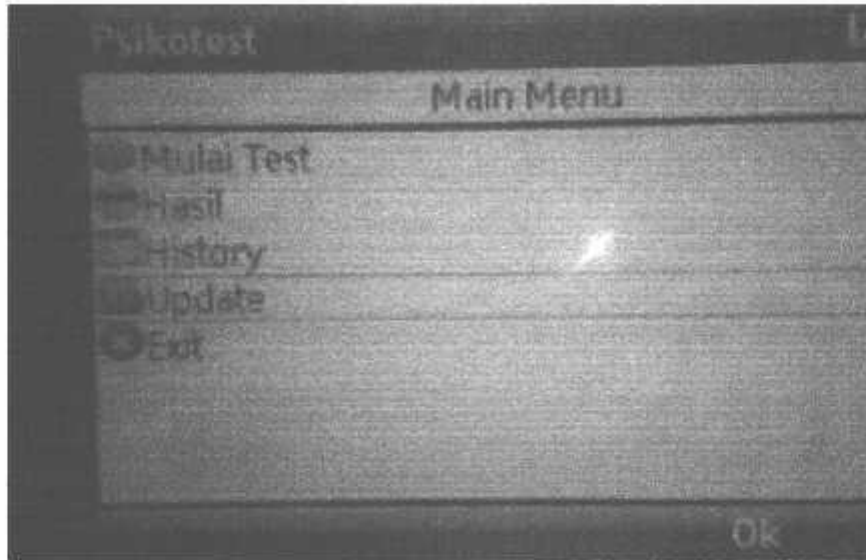
Gambar 4.12 Tampilan Menu History
Gambar dibawah ini bagian dari menu history



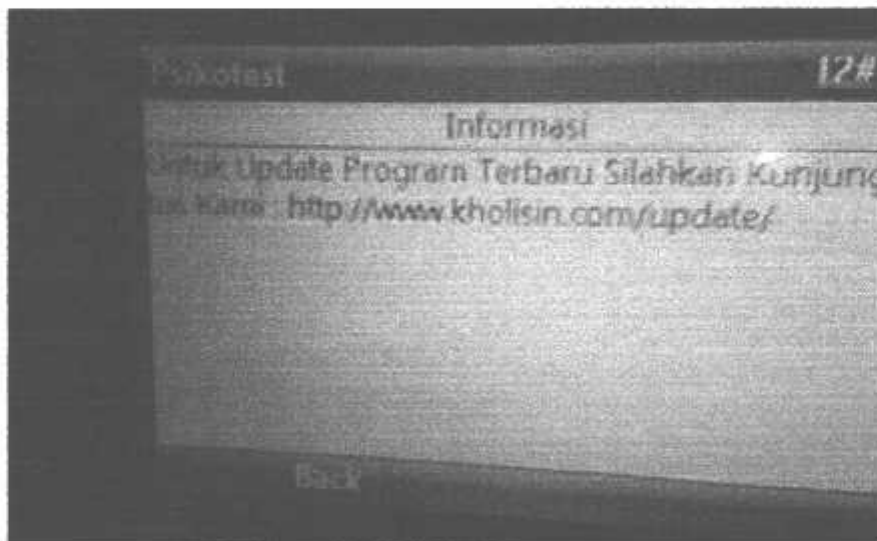
Gambar 4.13 Tampilan Menu Hasil Test History

4.3.11 Tampilan Menu Update

Tampilan ini digunakan apabila pengguna aplikasi ingin mengupdate soal soal terbaru, soal yang disediakan dalam aplikasi ini dapat diupdate disitus www.kholisin.com, dan berikut ini adalah tampilannya.



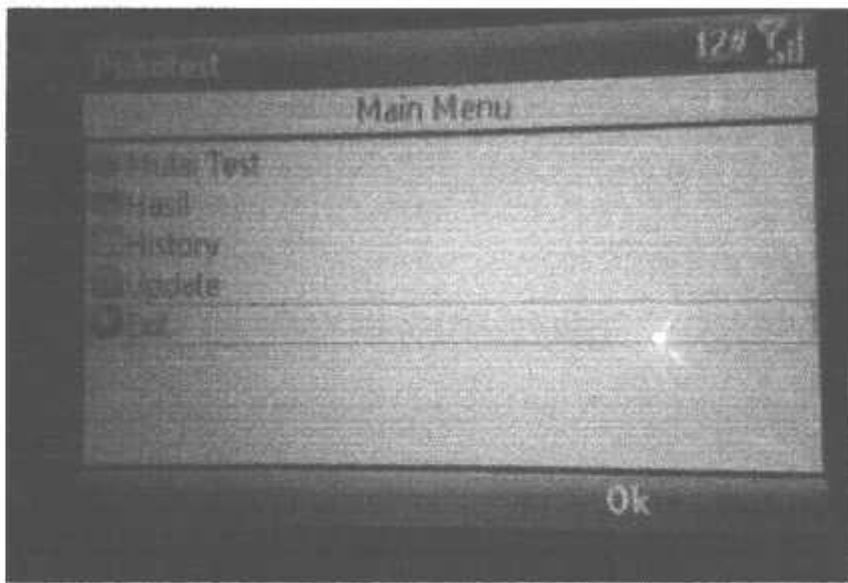
Gambar 4.14 Tampilan Menu Update
Gambar dibawah ini adalah bagian dari menu update



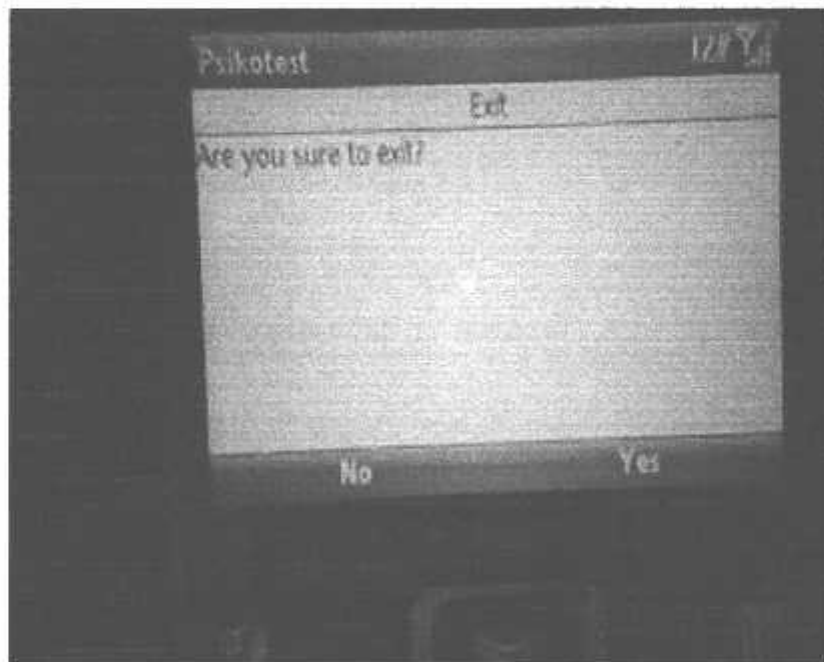
Gambar 4.15 Tampilan Informasi Menu Update

4.3.12 Tampilan Menu Exit

Tampilan menu ini digunakan apabila kita ingin menghentikan aplikasi psikotes, menu dapat kita lihat pada tampilan menu gambar dibawah ini.



Gambar 4.16 Tampilan Menu Exit
Gambar dibawah ini bagian dari tombol exit



Gambar 4.17 Tampilan Exit

4.4 Tabel Pengujian Aplikasi

Setelah pengujian terhadap sistem dilakukan, maka hasil dari pengujian dapat dilihat dalam bentuk tabel seperti berikut ini :

Tabel 4.1 Pengujian Aplikasi

No	Ruang Lingkup Aplikasi Psikotes	Tampilan
1	Main Menu	Berhasil
2	Mulai test	Berhasil
3	Hasil	Berhasil
4	History	Berhasil
5	Update	Berhasil
6	Exit	Berhasil
7	Simpan Test	Berhasil
8	Test Antonim	Berhasil
9	Test sinonim	Berhasil
10	Test Korelasi	Berhasil
11	Test Kuantitatif	Berhasil
12	Test Logika	Berhasil
13	Batal Test	Berhasil
14	Menu Utama	Berhasil

Keterangan :

- ✚ Berhasil pada tampilan ialah bahwa tampilan sesuai dengan submenu ataupun popupmenu yang telah dipilih oleh pengguna aplikasi.
- ✚ Tampilan diatas menunjukkan beberapa menu yang dijalankan, dan diatas adalah nama menu pada aplikasi psikotes yang dirancang.
- ✚ Tabel diatas menunjukkan kinerja aplikasi psikotes yang dijalankan.

4.4.1 Tabel Skoring

Setelah didapatkan hasil keseluruhan dari aplikasi, maka perhitungan dapat dilihat pada tabel dibawah ini

Tabel 4.2 Skoring

Nilai Mentah TPA	Perkiraan	
	GMAT/GRE	IQ
275	795	140>
270	790	140
265	785	
260	780	137
255	765	
250	750	135
245	735	
240	720	130
235	705	
230	690	
225	675	125
220	660	
215	645	
210	630	120
205	620	
200	610	115
195	600	
190	590	110
185	575	
180	560	105
170	540	
160	520	
150	500	100

140	480	
130	460	95
120	440	
110	420	
100	400	90
090	370	
080	350	
<080	<350	<90

Keterangan :

Sistem skoring TPA ini menyerupai GMT dan GRE, pada test ini ada 5 sub yaitu antonim, sinonim, korelasi, kuantitatif, logika, seluruh soal ada 50 butir, untuk skoring pakailah cara sebagai berikut :

1. Jumlahkan semua soal yang benar
2. Kurangi dengan jawaban yang salah dengan cara
 - a. Jumlahkan jawaban yang salah
 - b. Kalikan jumlah yang salah dengan $\frac{1}{4}$
 - c. Kurangi jawaban yang benar (1) dengan (b)
3. bulatkan nilai itu menjadi nilai mentah
4. nilai itu dapat dikonversi menjadi nilai GMAT/GRE dan nilai IQ

4.4.2 Tabel Perbandingan Sistem

Setelah pengujian terhadap sistem dilakukan, maka perbandingan sistem psikotes yang lama dengan sistem psikotes yang baru, akan dilakukan agar diketahui perbandingan antara psikotes lama dengan psikotes baru, tabel dibawah ini adalah perbandingan antara psikotes lama dan psikotes baru maka hasil dari perbandingan dapat dilihat dalam bentuk tabel seperti berikut ini :

Tabel 4.2 Perbandingan Sistem

No	Proses	Sistem Lama	Sistem Baru
1	Media yang digunakan untuk psikotes	Media yang digunakan untuk psikotes, pada sistem lama menggunakan media kertas	Media yang digunakan untuk psikotes, pada sistem yang baru menggunakan media handphone
2	Isi soal pada psikotes	Isi soal Pada psikotes sistem lama tidak dapat diupdate karena berbentuk kertas yang dapat digunakan sekali saja	Isi soal pada psikotes sistem baru bisa diupdate karena menggunakan media mobile sehingga dapat mengupdate soal kapanpun sesuai kehendak progremer .
3	Batas waktu pada psikotes	Batas waktu pada psikotes sistem lama tergantung pada acara yang diselenggarakan / tidak tentu setiap acaranya	Batas waktu pada aplikasi psikotes adalah 1 soal 1 menit
4	Isi soal yang ada pada waktu tes potensi akademik	Isi soal yang ada pada sistem lama Terdiri dari beberapa soal yang tidak tentu jumlahnya	Isi soal pada sistem baru Dibatasi hanya 50 soal saja

Keterangan :

- ✦ Table diatas menjelaskan antara perbandingan sistem antara psikotes sistem lama dengan sistem psikotes yang baru,

4.4.3 Analisa Sistem

Komunikasi antara komputer dan *handphone* untuk tiap-tiap tipe *handphone* itu berbeda. Tiap generasi *handphone* mempunyai fungsi-fungsi yang sangat mencolok yang tidak dimiliki oleh generasi sesudahnya maupun sebelumnya dan hal ini yang mempengaruhi perintah-perintah untuk melakukan komunikasi antara komputer terhadap *handphone*. Dan untuk mengatasi masalah itu dapat menggunakan sistem database fungsi-fungsi dari tiap generasi *handphone* yang dikeluarkan oleh suatu perusahaan *handphone*. Perintah-perintah yang digunakan untuk berkomunikasi antara komputer terhadap *handphone* untuk tiap-tiap *handphone* juga sama, sehingga proses yang dilakukan tidak begitu sulit. Dengan demikian program ini memiliki kelebihan mampu menyimpan data yang telah diselesaikan oleh pengguna aplikasi, memudahkan psikotes tanpa harus membeli buku dan user mendapat informasi tentang soal – soal yang biasanya diujikan didalam ujian psikotes.

BAB V PENUTUP

Berdasarkan pada hasil pengujian dan analisa terhadap hasil yang didapatkan, maka dapat diambil suatu kesimpulan dan saran untuk kemungkinan pengembangan sistem yaitu :

5.1. Kesimpulan

1. Pada aplikasi ini terdapat fasilitas hasil test, dari fasilitas ini pengguna dapat mengetahui hasil test yang sudah dikerjakan. Karena Pada umumnya fasilitas hasil test ini sangat dibutuhkan untuk pengguna.
2. Aplikasi psikotes ini terdapat 5 macam soal psikotes, soal – soal yang terdapat didalam aplikasi ini, biasanya sering keluar didalam tes potensi, aplikasi ini sangat membantu pengguna yang akan melaksanakan tes potensi akademik.
3. Soal psikotes yang ada didalam aplikasi psikotes ini ditujukan untuk test umum.
4. Aplikasi psikotes ini dijalankan didalam handphone yang menggunakan OS java

5.2. Saran

Mengingat masih banyaknya hal-hal yang belum dapat diimplementasikan pada skripsi ini, maka kami mempertimbangkan beberapa saran untuk perbaikan-perbaikan proyek akhir ini dalam hal :

Hal – hal yang dapat dikembangkan atau diberikan sebagai saran berkaitan dengan perancangan dan pembuatan sistem ini antara lain:

1. Dapat memberikan fasilitas load pada soal yang sudah tersimpan tujuanya untuk dapat mempermudah pengguna aplikasi untuk melanjutkan soal yang tersimpan.
2. Adanya tambahan fasilitas update soal yg dihubungkan dengan internet.
3. Adanya fasilitas yg dapat mengetahui tingkat kecerdasan dari pengguna.

6. DAFTAR PUSTAKA

1. Daril-McBride, *Java Development on PDAs: Building Application for Pocket PC and Palm Devices*, Addison Wesley, 2003.
 2. Michael Juntao Yuan & Kevin Shrps, *Developing Scalable Series 40 Application: A Guide for Java Developers*, Addison Wesley Profesional 2004.
 3. Nokia Corpotion, *Designes Java Application for Series 60*, 2002.
 4. Paul Tremblett, *Instant Wireless Java with J2ME Technology and MIDP*, sun Microsystems Press and Prentice – Hall, 2002
 5. Sugeng, DTdkk(2008) TPA: Tes Potensi Akademik, Yogyakarta Progesif
 6. Tim Psikolog Hariwijaya Group,(2006),Strategi Lolos Psikotes dan Tes Potensi Akademik,Toko bukumila Banten.
-





PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo. Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**


Nama : Ahmad Kholisin
Nim : 0612592
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Judul : APLIKASI PSIKOTES BERBASIS MOBILE MENGGUNAKAN J2ME

Dipertahankan dihadapan Tim Pengujian Skripsi jenjang Program Strata Satu (S-1)

Pada Hari : Jum'at
Tanggal : 18 Februari 2011
Dengan Nilai : B+ (73,33)


Panitia Ujian Skripsi

Ketua Majelis Penguji



Ir. Yusuf Ismail Nakhoda, MT.
NIP.Y.1018800189

Sekretaris Majelis Penguji



Dr. Aryuanto Soetedjo, ST. MT.
NIP.P.1030800417

Anggota Penguji

Penguji Pertama



Dr. Aryuanto Soetedjo, ST. MT.
NIP.P.1030800417

Penguji Kedua



Sotyohadi, ST
NIP.Y.103970309



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Ahmad Kholisin
 NIM : 06 125 92
 Jurusan : Teknik Elektro S-1
 Konsentrasi : Teknik Komputer & Informatika
 Masa Bimbingan : 13 Desember 2011 s/d 13 Juni 2011
 Judul Skripsi : APLIKASI PSIKOTES BERBASIS MOBILE
 MENGGUNAKAN J2ME

Penguji/Tanggal	Uraian	Paraf
Penguji I 18 Februari 2011	Judul "game" dihilangkan Teori perhitungan nilai tidak ada dilaporan Ditambahkan bank soal yang dimunculkan tiap session berbeda, (misal : ada 100 soal, ditambahkan 10 soal secara acak setiap running)	
Penguji II 18 Februari 2011		

Dosen Penguji,
Penguji I

Dr Eng Aryuanto Soetedjo, ST, MT
 NIP.P.1030800417

Penguji II

Sotyohadi, ST
 NIP.Y.103970309

Mengetahui,

Dosen Pembimbing I

(I Komang Somawirata, ST, MT)
 NIP.P.1030100361

Dosen Pembimbing II

(Sandy Nataly Mantja, Skom)
 NIP.P.1030800418



FORMULIR BIMBINGAN SKRIPSI

Nama : Ahmad Kholisin
Nim : 06.12.592
Masa Bimbingan : 13 Desember 2010 s/d 13 Juni 2011
Judul Skripsi : Aplikasi Psikotes Berbasis Mobile Menggunakan J2me

No	Tanggal	Uraian	Paraf Pembimbing
1	2/2/11	Perbaiki bab I dan bab II	
2	3/2/11	Bab I dan bab II ok	
3	4/2/11	Perbaiki bab III bab IV	
4	5/2/11	Bab III dan bab IV ok	
5	6/2/11	Perbaiki makalah seminar hasil pada gambar	
6	7/2/11	Makalah seminar hasil pada gambar ok	
7			
8			
9			
10			

Malang,
Dosen Pembimbing I

(I Komang Somawirata, ST, MT)
NIP. 1030100361



FORMULIR BIMBINGAN SKRIPSI

Nama : Ahmad Kholisin
Nim : 06.12.592
Masa Bimbingan : 13 Desember 2010 s/d 13 Juni 2011
Judul Skripsi : Aplikasi Psikotes Berbasis Mobile Menggunakan J2me

No	Tanggal	Uraian	Paraf Pembimbing
1	31/1/11	Perbaiki Bab I, III	
2	2/2/11	Bab I, II, III, IV ok siap demo	
3	5/2/11	Makalah pada kata – kata diperbaiki	
4	6/2/11	Makalah sudah acc	
5			
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing II

(Sandy Nataly Mantja, Skom)
NIP.P.1030800418

LAMPIRAN

> Data.java

```
public class data {

    soal [] sVerbal;
    soal [] sVerbal2;
    soal [] sKorelasi;
    soal [] sKuantif;
    soal [] sLogika;
    int jml =10;

    private void Init(){
        sVerbal = new soal[jml];
        sVerbal2 = new soal[jml];
        sKorelasi=new soal[jml];
        sKuantif = new soal[jml];
        sLogika = new soal[jml];
    }

    public int getJml() {
        return jml;
    }

    public soal[] getsKorelasi() {
        return sKorelasi;
    }

    public soal[] getsLogika() {
        return sLogika;
    }

    public soal[] getsVerbal() {
        return sVerbal;
    }

    public soal[] getsVerbal2() {
        return sVerbal2;
    }

    public soal[] getsKuantif() {
        return sKuantif;
    }

    private void isiSoal(){

        sVerbal[0] = new soal("ABSOLUT", "", "Kadut", "Takut", "Maut", "Mutlak", "eram", "Mutlak");
        sVerbal[1] = new soal("ACUM", "", "Rujukan", "Muka", "Depan", "Belakang", "Panggung", "Rujukan");
        sVerbal[2] = new
soal("ADAPTASI", "", "Transportasi", "Rekreasi", "Pemberian", "Adopsi", "Penyesuaian", "Penyesuaian");
        sVerbal[3] = new
soal("AKREDITASI", "", "Kredit", "Meditasi", "Pengakuan", "Mulla", "Pencelantaran", "Pengakuan");
        sVerbal[4] = new soal("AKSELERASI", "", "Aksi", "Percepatan", "Reaksi", "Terasi", "Top Speed", "Percepatan");
        sVerbal[5] = new soal("AKURAT", "", "Seksama", "Bijaksana", "Ceria", "Durjana", "Tepat guna", "Seksama");
        sVerbal[6] = new soal("ALGORITMA", "", "Aljabar", "Prosedur", "Ritme", "Dugaan", "Momentum", "Prosedur");
        sVerbal[7] = new soal("ALTERNATIF", "", "Tipuan", "Harapan", "Tinjauan", "Jalan pintas", "Cara lain", "Cara
lain");
        sVerbal[8] = new soal("AMBIGUITAS", "", "Andalan", "Makna ganda", "Aktivitas", "Tanpa
Makna", "Bermakna", "Makna ganda");
        sVerbal[9] = new soal("ANDAL", "", "Dashyat", "Luar biasa", "Tangguh", "Mental", "Teruji", "Tangguh");

        sVerbal2[0] = new soal("MAJEMUK", "", "Berbeda-beda", "Sejenis", "Heterogen", "Beragam", "Berbagai
Jenis", "Sejenis");
        sVerbal2[1] = new soal("MAKAR ><", "", "Sekongkol", "Muslihat", "Kudeta", "Militer", "Milisi", "Muslihat");
        sVerbal2[2] = new soal("MANDIRI", "", "Independen", "
Merdeka", "Wiraswasta", "Wirauyaha", "Dependen", "Dependen");
```

```

sVerbal2[3] = new soal("MAYOR ><", "", "Kopral", "Jendral", "Minor", "Menor", "Kapten", "Minor");
sVerbal2[4] = new soal("MENTAL ><", "", "Fisik", "Materi", "Sosial", "Ekonomi", "Budaya", "Fisik");
sVerbal2[5] = new soal("MENYAMBUNG
><", "", "Menolong", "Mencuri", "Memutus", "Menolak", "Merusak", "Memutus");
sVerbal2[6] = new soal("MITRA ><", "", "Penolong", "Saingan", "Kerjasama", "Kerja", "Bisnis", "Saingan");
sVerbal2[7] = new soal("MODERN ><", "", "Kuno", "Klasik", "Tempo dulu", "Antik", "Bekas", "Kuno");
sVerbal2[8] = new soal("MODIFIKASI
><", "", "Mengubah", "Mengganti", "Standar", "Menambah", "Mengurangi", "Standar");
sVerbal2[9] = new soal("MONOTEIS ><", "", "Politik", "Politek-nik", "Politeis", "Poligami", "Poligon", "Politeis");

sKorelasi[0] = new soal("TAPE : Ragi", "", "Roti : selai", "Tahu : tempe", "Makanan : ringan", "Antiseptik:
Iodium", "Apolik : Obat", "Antiseptik: Iodium");
sKorelasi[1] = new soal("Susu : Gelas", "", "Bubur : Piring", "Nasi : kbul", "Susu: Sapi", "Rendang: padang", "Bubur
: Kacang hijau", "Bubur : Piring");
sKorelasi[2] = new soal("Bunga : Taman", "", "Siswa : Sisiw", "Dosen : Universitas", "Guru : Murid", "Mahasiswa :
mahasiswa", "Matahari : bulan", "Dosen : Universitas");
sKorelasi[3] = new soal("Padi : Petani", "", "Pedagang : pasar", "Petani : jagung", "Padi : Panen", "Penyair :
Puisi", "Pilot : Pesawat", "Penyair : Puisi");
sKorelasi[4] = new soal("Presiden : Negara", "", "Presiden : Indonesia", "Ayah : Keluarga", "Kerajaan :
Majapahit", "Negara : bangsa", "Anak : Sekolah", "Ayah : Keluarga");
sKorelasi[5] = new soal("Teluk : Laut", "", "Sernenanjung: Daratan", "Laut : Jaws", "Laut : Atlantik", "Tanjung :
Harapan", "Teluk Persia", "Sernenanjung: Daratan");
sKorelasi[6] = new soal("Adagio : Allegro", "", "Malas : bodoh", "Cantik : Cakep", "Lambat : Cepat", "Pandal :
Rajin", "Negro : Bole", "Lambat : Cepat");
sKorelasi[7] = new soal("Marah : Cemburu", "", "Marah : Emosi", "Cemburu : buta", "Fanatik : Suku", "Tidak
Toleransi : Fanatik", "Senang : bahagia", "Tidak Toleransi : Fanatik");
sKorelasi[8] = new soal("Serut : Kayu", "", "Cangkul : Kebun", "Kayu : lapis", "Cangkul : Kayu", "Kebun :
piwang", "Kebun : tebu", "Cangkul : Kebun");
sKorelasi[9] = new soal("Kosong : Hampa", "", "Cair : Encer", "Panas : api", "Rumah : kosong", "Hampa :
udara", "Keras : Kepala", "Panas : api");

sKuantif[0] = new soal("4,5,7,10,.....,25,32", "", "13,18", "15,20", "14,19", "16,21", "11,23", "11,23");
sKuantif[1] = new soal("3,8,9,16,27,24.....", "", "36,32", "36,50", "36,90", "36,79", "34,56", "36,79");
sKuantif[2] = new soal("60,62,50,52,40,42.....", "", "20,22", "30,32", "40,32", "30,34", "20,30", "30,32");
sKuantif[3] = new soal("16,35,14,30,12,25.....", "", "29,36", "20,35", "20,10", "30,20", "30,10", "30,10");
sKuantif[4] = new soal("4,9,.....,36,59", "", "15,30", "13,30", "15,25", "16,25", "14,24", "15,25");
sKuantif[5] = new soal("1,2,4,8,.....,64", "", "16,23", "20,25", "16,32", "32,48", "32,30", "32,48");
sKuantif[6] = new soal("A,D,H,M,.....", "", "P", "Q", "R", "S", "T", "S");
sKuantif[7] = new soal("B,G,K,N,.....", "", "P", "Q", "R", "S", "T", "Q");
sKuantif[8] = new soal("A,C,....G,I", "", "B", "D", "E", "F", "H", "F");
sKuantif[9] = new soal("A,C,F,J,O,....", "", "P", "Q", "R", "S", "T", "P");

sLogika[0] = new soal("Lihat Gambar....", "1", "A", "B", "C", "D", "E", "C");
sLogika[1] = new soal("Lihat Gambar....", "2", "A", "B", "C", "D", "E", "B");
sLogika[2] = new soal("Lihat Gambar....", "3", "A", "B", "C", "D", "E", "A");
sLogika[3] = new soal("Lihat Gambar....", "4", "A", "B", "C", "D", "E", "D");
sLogika[4] = new soal("Lihat Gambar....", "5", "A", "B", "C", "D", "E", "E");
sLogika[5] = new soal("Lihat Gambar....", "6", "A", "B", "C", "D", "E", "F");
sLogika[6] = new soal("Lihat Gambar....", "7", "A", "B", "C", "D", "E", "D");
sLogika[7] = new soal("Lihat Gambar....", "8", "A", "B", "C", "D", "E", "A");
sLogika[8] = new soal("Lihat Gambar....", "9", "A", "B", "C", "D", "E", "E");
sLogika[9] = new soal("Lihat Gambar....", "10", "A", "B", "C", "D", "E", "D");

}

public data() {
    Init();
    Isisoal();
}

public class soal {
    private String soal;
    private String gambar;
    private String jawabA, jawabB, jawabC, jawabD, jawabE;
    private String jawaban;
}

```

```

    public soal(String soal, String gambar, String jawabA, String jawabB, String jawabC, String jawabD, String jawabE,
String jawaban) {
        this.soal = soal;
        this.gambar = gambar;
        this.jawabA = jawabA;
        this.jawabB = jawabB;
        this.jawabC = jawabC;
        this.jawabD = jawabD;
        this.jawabE = jawabE;
        this.jawaban = jawaban;
    }

    public String getGambar() {
        return gambar;
    }

    public String getJawabA() {
        return jawabA;
    }

    public String getJawabB() {
        return jawabB;
    }

    public String getJawabC() {
        return jawabC;
    }

    public String getJawabD() {
        return jawabD;
    }

    public String getJawabE() {
        return jawabE;
    }

    public String getSoal() {
        return soal;
    }

    public String getJawaban() {
        return jawaban;
    }

    public boolean isBenar(String jawab){
        boolean hasil=false;
        if (jawab == null ? jawaban == null : jawab.equals(jawaban)){
            hasil =true;
        }
        return hasil;
    }
}

```

➤ Hasil java

```

import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;

```

/*

* To change this template, choose Tools : Templates
* and open the template in the editor.

```

*/
/**
 *
 * @author Mbah Marijan Okek
 */
public class Hasil extends Form implements CommandListener{
    Psikotes parent;

    public Hasil(Psikotes parent,int[] hasil ){
        super("Hasil Test.");
        this.parent = parent;

        append(new StringItem(" Test Antonim :", String.valueOf(hasil[0] ));
        append(new StringItem(" Test Sinonim :", String.valueOf(hasil[1] ));
        append(new StringItem(" Test Korelasi :", String.valueOf(hasil[2] ));
        append(new StringItem(" Test Kuantitatif :", String.valueOf(hasil[3] ));
        append(new StringItem(" Test Logika :", String.valueOf(hasil[4] ));
        this.setCommandListener(this);

    }

    public void commandAction(Command c, Displayable d) {
        if (c.getLabel().equals("Ok")){

            parent.display.setCurrent(parent.menuList);
        }else{
            parent.fHasil=null;
            parent.display.setCurrent(parent.fHistory);
        }
    }
}
}

```

➤ History Java

```

import java.io.IOException;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Font;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.List;
import javax.microedition.lcdui.StringItem;
import javax.microedition.rms.RecordStoreException;

/**
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Mbah Marijan Okek
 */
public class History extends List implements CommandListener {
    private Psikotes parent;
    private Command cDel,cBack,cBuka, cBatal;

    public History (Psikotes parent) {
        super("Hasil Test", IMPLICIT);
        setFitPolicy(TEXT_WRAP_ON);
        try {

```

```

        this.parent = parent;

        cDel = new Command("Hapus", Command.OK, 2);
        cBack = new Command("Kembali", Command.BACK, 1);
        cBuka = new Command("Detail", Command.OK, 1);

        this.addCommand(cDel);
        this.addCommand(cBack);
        this.addCommand(cBuka);
        fill();

        this.setCommandListener(this);
    } catch (Exception ex) {
    }
}

private void tampilDetail(){
    try {

        byte[] b = parent.rmsHasil.record.getRecord(getSelectedIndex() + 1);
        String record = new String(b);
        String a;
        String c;
        a = record.substring(0, record.indexOf("@1#"));
        if (a.indexOf(":") != -1){
            a = a.substring(0, a.indexOf(":"));
        }
        int [] jawab= new int[5];

        jawab[0] = Integer.parseInt( record.substring(record.indexOf("@1#") + 3, record.indexOf("@2#")));
        jawab[1] = Integer.parseInt( record.substring(record.indexOf("@2#") + 3, record.indexOf("@3#")));
        jawab[2] = Integer.parseInt( record.substring(record.indexOf("@3#") + 3, record.indexOf("@4#")));
        jawab[3] = Integer.parseInt( record.substring(record.indexOf("@4#") + 3, record.indexOf("@5#")));
        jawab[4] = Integer.parseInt( record.substring(record.indexOf("@5#") + 3, record.indexOf("@6#")));
        parent.fHasil = new Hasil(parent, jawab);
        parent.fHasil.addCommand( new Command("Kembali", Command.OK, 0));
        parent.display.setCurrent(parent.fHasil);

    } catch (RecordStoreException ex) {
    }
}

public void fill(){
    try {
        deleteAll();
        String a;
        String c;

        for (int i = 1; i <= parent.rmsHasil.record.getNumRecords(); i++) {
            byte[] b = parent.rmsHasil.record.getRecord(i);
            String record = new String(b);

            a = record.substring(0, record.indexOf("@1#"));
            c = record.substring(record.indexOf("@2#") + 3, record.indexOf("@3#"));
            append(a , Image.createImage("/Images/new.png"));

        }
    } catch (IOException ex) {
    } catch (RecordStoreException ex) {
    }
}
}

```



```

public Form waitForm ;
public Splash splash;
public List menuList;

private Gauge wait;
public Display display;
public List FrmRecent;
public Hasil fHasil;
soal fsoal;
public int modeSoal=0;
public Rms rmsHasil;
public History fHistory;

public Psikotes() {
    Exit = new Command("Exit", Command.EXIT, 0);
    Back = new Command("Back", Command.BACK, 0);
    Detail = new Command("Detail", Command.OK, 0);
    OK = new Command("Ok", Command.OK, 0);

    splash = new Splash(this);
    menuList = new List("Main Menu", List.IMPLICIT);
    try {
        for (int i = 0; i < strMenu.length; i++) {
            menuList.append(strMenu[i] + "\n", Image.createImage("/images/" + String.valueOf(i) + ".png"));
        }
    } catch (Exception ex) {}
    menuList.addCommand(OK);
    if (this.rmsHasil == null) {
        this.rmsHasil = new Rms("HasilRms");
    }
    this.rmsHasil.openRecord();
}

public void startApp() {
    menuList.setCommandListener(this);
    display = Display.getDisplay(this);
    splash.show();
}

public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
}

public void commandAction(Command c, Displayable d) {
    if (c.getLabel().equals("Yes")) {
        notifyDestroyed();
    } else if (c.getLabel().equals("Ok") || c == List.SELECT_COMMAND ) {
        switch (menuList.getSelectedIndex()) {
            case 0:
                modeSoal = 0;
                if (fsoal != null) fsoal = null;
                fsoal = new soal(this);
                fsoal.show();

                break;
            case 1:
                if (fHasil == null) {
                    Alert alert = new Alert("Info", "Data Hasil Tidak Ditemukan", null, AlertType.INFO);
                    alert.setTimeout(2000);
                    alert.addCommand(new Command("Kembali", Command.BACK, 1));
                    alert.setCommandListener(this);
                    display.setCurrent(alert);
                }
            }
        }
    }
}

```

```

        }else{
            display.setCurrent(fHasil);
        }

        break;
    case 2:
        fHistory = new IHistory(this);
        this.display.setCurrent(fHistory);
        break;
    case 3:
        Alert alert = new Alert("Exit", "Are you sure to exit?", null, AlertType.CONFIRMATION);
        alert.setTimeout(2000);
        alert.addCommand(new Command("Yes", Command.OK, 0));
        alert.addCommand(new Command("No", Command.BACK, 1));
        alert.setCommandListener(this);
        display.setCurrent(alert);
        break;
    }
} else{
    display.setCurrent(menuList);
}
}
}

```

> Rms java

```

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import javax.microedition.rms.RecordStore;
import javax.microedition.rms.RecordStoreException;
import javax.microedition.sensor.Data;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Mbah Marijan Okek
 */
public class Rms {
    public RecordStore record;

    public String REC_STORE = "";

    public Rms(String REC_STORE) {
        this.REC_STORE = REC_STORE;
    }

    public void openRecord() {
        try{
            record = RecordStore.openRecordStore(REC_STORE, true);
        }catch (Exception e){
            System.err.println(e.toString());
        }
    }

    public void closeRecord(){
        try{
            record.closeRecordStore();
        }catch (Exception c){
            System.err.println(e.toString());
        }
    }
}

```

```

    }
}

public void deleteRecord(){
    if (RecordStore.listRecordStores() != null){
        try{
            RecordStore.deleteRecordStore(REC_STORE);
        } catch (Exception e){
            System.err.println(e.toString());
        }
    }
}

public void writeRecord(String str){
    byte[] rec = str.getBytes();
    try{
        record.addRecord(rec, 0, rec.length);
    } catch (Exception e){
        System.err.println(e.toString());
    }
}

public void writeInbox(String nomer, String pesan, String tanggal, String status){
    try {
        byte[] temps = null;
        ByteArrayOutputStream outputStream = null;
        DataOutputStream dos = new DataOutputStream(outputStream);
        dos.writeUTF(nomer);
        dos.writeUTF(pesan);
        dos.writeUTF(tanggal);
        dos.writeUTF(status);
        temps = outputStream.toByteArray();
        record.addRecord(temps, 0, temps.length);
    } catch (RecordStoreException ex) {
    } catch (IOException ex) {
    }
}

public void readRecord(){
    try{
        if (record.getNumRecords() > 0){
            for (int i=1; i<= record.getNumRecords();i++){
                byte b[] = record.getRecord(i);
                String str = new String(b);
                System.out.println(" " + str);
            }
        }
    } catch (Exception e){
        System.err.println(e.toString());
    }
}

public String getData(int idx){
    String Data="";
    try{
        if (record.getNumRecords() > 0){
            byte b[] = record.getRecord(idx);
            Data = new String(b);
        }
    } catch (Exception e){
        System.err.println(e.toString());
    }
    return Data;
}
}
}

```

➤ Splash Java

```
import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Graphics;
import javax.microedition.lcdui.Image;

/**
 * @author Mbah Marijan Okek
 */
public class Splash extends Canvas {
    private Psikotes parent;
    private Image iSplash;

    public Splash(final Psikotes parent) {

        this.parent =parent ;
        try {
            iSplash = Image.createImage("/images/logo.png");
        } catch (Exception e) {

        }
        setFullScreenMode(true);
    }
    public void show() {
        parent.display.setCurrent(this);
        try {
            Thread.sleep(2000);
        } catch (InterruptedException ex) {
            ex.printStackTrace();
        }
        parent.display.setCurrent(parent.menuList);
    }

    public void paint(Graphics g) {
        g.setColor(255,255,255);
        g.fillRect(0, 0, getWidth(), getHeight());
        g.drawImage(iSplash, (getWidth()-iSplash.getWidth())/2, (getHeight()-iSplash.getHeight())/2, 0);
    }
}
```