



**RANCANG BANGUN APLIKASI TES PSIKOLOGI
“HoTest” (*HOLLAND ADVANCED-STUDY TEST*)
DENGAN MENERAPKAN ALAT UKUR SDS
(*SELF-DIRECTED SEARCH*)**

SKRIPSI

**diajukan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana
Pendidikan Program Studi Teknik Informatika dan Komputer**

Oleh

Inayah Kurniawati NIM. 5302411169

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS NEGERI SEMARANG

2015

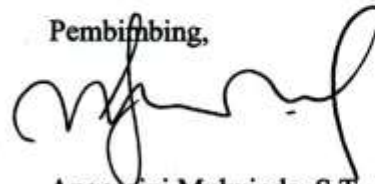
PERSETUJUAN PEMBIMBING

Nama : Inayah Kurniawati
NIM : 5302411169
Program Studi : S-1 Pendidikan Teknik Informatika dan Komputer
Judul Skripsi : Rancang Bangun Aplikasi Tes Psikologi "HoTest" (*Holland Advanced-Study Test*) dengan Menerapkan Alat Ukur SDS (*Self-Directed Search*)

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi Program Studi S-1 Pendidikan Teknik Informatika dan Komputer FT. UNNES.

Semarang, 01 Juni 2015

Pembimbing,



Anggraini Mulwinda, S.T., M.Eng.

NIP. 19781226 200501 2 002

HALAMAN PENGESAHAN

Skripsi dengan judul “Rancang Bangun Aplikasi Tes Psikologi “HoTest” (*Holland Advanced-Study Test*) dengan Menerapkan Alat Ukur SDS (*Self-Directed Search*)” telah dipertahankan di depan sidang Panitia Ujian Skripsi Fakultas Teknik Universitas Negeri Semarang pada tanggal 24 Juni 2015.

Oleh :


Nama : Inayah Kurniawati

NIM : 5302411169


Program Studi : Pendidikan Teknik Informatika dan Komputer

Panitia:


Ketua Panitia


Drs. Sarwono, M.T.
NIP. 195503161985031001

Sekretaris


Feddy Setio Pribadi, S.Pd., M.T.
NIP. 197808222003121002

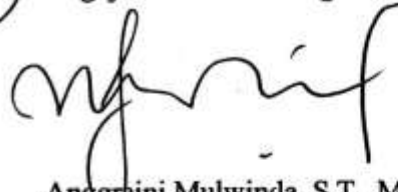
Penguji I


Dr. H. Noor Hudallah, M.T.
NIP. 196410161989011001

Penguji II


Drs. Sri Sukamta, M.Si.
NIP. 196505081991031003


Penguji III/ Pembimbing


Angraini Mulwinda, S.T., M.Eng.
NIP. 197812262005012002

Mengetahui,

Dekan Fakultas Teknik,




Drs. Muhammad Harlanu, M.Pd.
NIP. 196602151991021001

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa :

1. Skripsi ini, adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, magister, dan/atau doktor), baik di Universitas Negeri Semarang (UNNES) maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan Pembimbing dan masukan Tim Penguji.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi ini.

Semarang, 29 Mei 2015



Inayah Kurniawati
NIM. 5302411169

MOTTO DAN PERSEMBAHAN

1. *“...Barangsiapa bertaqwa kepada Allah, niscaya Dia akan mengadakan baginya jalan keluar.”(QS At-Talaq:2)*
2. *Jangan pernah menjadikan kesusahan dan kesedihanmu sebagai tema pembicaraan. Karena dengan demikian, engkau akan menjadikannya sebagai penghalang antara dirimu dan kebahagiaan. (Dr. Aidh al-Qarni)*
3. *Kita tidak akan pernah kecewa jika kita selalu mengendalikan harapan. Mau secanggih apapun orang lain memupuk pesonanya, menimbun perhatiannya, kalau kita sempurna mengendalikan hati, no problem at all. (Darwis Tere Liye)*

Dengan mengucapkan syukur Alhamdulillah, kupersembahkan karya kecilku ini untuk orang-orang yang kusayangi :

- Bapak Ibu tercinta, motivator terhebat dalam hidupku yang tak pernah lelah memberikan doa, pengorbanan, dukungan, dan kesabarannya hingga mengantarkanku sampai saat ini.
- Saudaraku, Kakak dan Adikku tersayang yang selalu memberikan semangat dan doa.
- Sahabat-sahabatku seperjuangan dan teman-teman PTIK 2011 yang selalu membantu. Terimakasih atas waktu dan dukungannya.

ABSTRAK

Kurniawati, Inayah. 2015. “Rancang Bangun Aplikasi Tes Psikologi “HoTest” (*Holland Advanced-Study Test*) dengan Menerapkan Alat Ukur SDS (*Self-Directed Search*)”. *Skripsi*. Pendidikan Teknik Informatika dan Komputer. Jurusan Elektro: Fakultas Teknik. Universitas Negeri Semarang.

Pembimbing : Anggraini Mulwinda, S.T., M.Eng.

Kata Kunci : Aplikasi Tes Psikologi, Tes SDS (*Self-Directed Search*), Berbasis *desktop*, Java, SQLite, Model *Waterfall*, UML (*Unified Modelling Language*), Analisis PIECES.

Dalam praktiknya, tes psikologi dalam memahami suatu objek yaitu manusia dengan segala sikap dan tingkah lakunya masih menggunakan metode lama. Tes psikologi masih menggunakan lembar questioners atau serangkaian pertanyaan yang diberikan kepada objek terkait dan penghitungan hasil atau skoring masih dilakukan secara manual. Kesalahan dalam skoring akan berpengaruh terhadap hasil sehingga akan menimbulkan persepsi yang tidak sesuai. Penelitian ini bertujuan untuk menghasilkan perangkat lunak berupa Aplikasi Tes Psikologi *HoTest (Holland Advanced-Study Test)* berbasis *desktop* menggunakan Java dan SQLite serta untuk mengetahui validitas skoring dalam menentukan 3 kode ringkas tertinggi beserta saran studi berdasarkan kamus Holland yang diujikan di Laboratorium Psikologi Universitas Negeri Semarang. Pada penelitian ini pengujian dilaksanakan dengan membandingkan hasil penghitungan menggunakan aplikasi *HoTest (Holland Advanced-Study Test)* dan penghitungan secara manual.

Metode yang digunakan dalam penelitian adalah Metode *Waterfall*. Model *waterfall* disebut juga sebagai model *sekuensial linier* yang mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial dimulai dari analisis, desain, kode, pengujian, dan pemeliharaan. Dalam menganalisis sistem, langkah yang dilakukan adalah studi pendahuluan yaitu tahap pengumpulan data dan identifikasi masalah serta kebutuhan pengguna, prinsip dan konsep analisis, dan model analisis untuk menggambarkan kebutuhan pelanggan serta membangun dasar bagi pembuatan desain perangkat lunak. Dalam pembuatan model analisis digunakan UML (*Unified Modelling Language*). Tahap desain adalah tahap pembuatan desain data, desain arsitektur, dan desain *interface*. Tahap kode adalah penerjemahan desain dalam bentuk mesin yang dapat dibaca. Pada tahap pengujian sistem digunakan metode pengujian analisis sistem yaitu PIECES.

Pengujian dilakukan di Laboratorium Psikologi Universitas Negeri Semarang. Pengujian dilakukan oleh satu orang ahli yaitu Kepala Laboratorium dikarenakan adanya unsur kerahasiaan isi yang tidak boleh disebarluaskan. Dari hasil analisis sistem menggunakan metode PIECES, aplikasi *HoTest (Holland Advanced-Study Test)* memiliki fungsionalitas yang cukup baik dengan informasi yang dihasilkan merupakan informasi yang akurat, tepat pada waktunya, dan relevan. Kemudian hasil *review* dari pengujian menyebutkan bahwa aplikasi *HoTest* memiliki tingkat akurasi penghitungan yang baik dan dapat digunakan dalam pelaksanaan tes minat SDS (*Self-Directed Search*) berbasis komputer.

KATA PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT yang telah melimpahkan nikmat, rahmat, taufik, dan hidayah-Nya, sehingga penulisan skripsi ini dapat diselesaikan dengan baik.

Laporan skripsi ini disusun guna memenuhi salah satu syarat untuk menyelesaikan Program Studi Pendidikan Teknik Informatika dan Komputer Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang.

Keberhasilan penulis dalam menyusun skripsi ini tidak terlepas dari bantuan, dorongan, serta doa dari berbagai pihak baik secara langsung maupun tidak langsung. Pada kesempatan ini, dengan segala kerendahan hati penulis menyampaikan terima kasih yang setulus-tulusnya kepada :

1. Bapak Prof. Dr. Fathur Rokhman, M.Hum., sebagai Rektor Universitas Negeri Semarang atas kesempatan yang diberikan kepada penulis untuk menempuh studi di Universitas Negeri Semarang.
2. Bapak Drs. Muhammad Harlanu, M.T., sebagai Dekan Fakultas Teknik Universitas Negeri Semarang yang telah memberikan bimbingan dengan menerima kehadiran penulis setiap saat disertai kesabaran, ketelitian, masukan-masukan yang berharga.
3. Bapak Drs. Suryono, M.T., sebagai Ketua Jurusan Teknik Elektro yang telah memberikan bimbingan dengan menerima kehadiran penulis setiap saat disertai kesabaran, ketelitian, dan masukan-masukan yang berharga untuk menyelesaikan karya ini.

4. Bapak Feddy Setio Pribadi S.Pd, M.T., sebagai Ketua Program Studi Pendidikan Teknik Informatika dan Komputer yang telah memberi bimbingan dan dorongan dengan menerima kehadiran penulis setiap saat disertai kesabaran, ketelitian, serta masukan-masukan yang berharga untuk menyelesaikan karya ini.
5. Ibu Anggraini Mulwinda, S.T, M.Eng., sebagai dosen pembimbing utama yang telah memberikan arahan, bimbingan, dan saran kepada penulis disertai kesabaran dan ketelitian.
6. Ibu Liftiah, S.Psi., M.Si., sebagai Kepala Laboratorium Psikologi Universitas Negeri Semarang yang telah memberikan bantuan serta arahan.
7. Seluruh Dosen dan staf karyawan jurusan Teknik Elektro.
8. Kedua Orang tua tercinta, Bapak dan Ibu yang selalu memberikan doa, semangat, serta dorongan yang tiada hentinya.
9. Rekan-rekan PTIK 2011 yang selalu memberikan bantuan serta semangat dalam penyusunan skripsi ini.

Semoga laporan skripsi ini dapat bermanfaat bagi semua pihak sebagaimana yang diharapkan. Amin.

Penulis,

DAFTAR ISI

HALAMAN JUDUL	i
PERSETUJUAN PEMBIMBING	ii
HALAMAN PENGESAHAN	iii
PERNYATAAN KEASLIAN	iv
MOTTO DAN PERSEMBAHAN	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	8
1.3 Tujuan Penelitian	9
1.4 Manfaat Penelitian	9
1.5 Batasan Masalah	10
1.6 Sistematika Penulisan Skripsi	11
BAB II KAJIAN PUSTAKA	14
2.1 Komputer	14
2.1.1 Pemrograman Komputer	14
2.1.2 Bahasa Pemrograman	17
2.1.3 Java	20
2.1.4 Basis Data (<i>Database</i>)	23
2.1.5 Pemodelan Perangkat Lunak	25
2.1.6 Model <i>Waterfall</i>	29
2.1.7 UML (<i>Unified Modelling Language</i>)	32
2.2 Psikologi	39
2.2.1 Tes Psikologi	39

2.2.2 Tes Minat.....	50
2.2.3 Teori Holland.....	52
2.2.4 SDS (<i>Self-Directed Search</i>).....	58
2.3 Kerangka Berfikir	62
BAB III METODE PENELITIAN	66
3.1 Analisis Perangkat Lunak	66
3.1.1 Studi Pendahuluan	66
3.1.2 Prinsip dan Konsep Analisis.....	73
3.1.3 Model Analisis.....	74
3.2 Desain Perangkat Lunak	90
3.2.1 Desain Data atau <i>Class</i>	90
3.2.2 Desain Arsitektur	96
3.2.3 Desain <i>Interface</i>	97
3.3 Pengkodean.....	102
3.4 Pengujian	106
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	112
4.1 Aplikasi <i>HoTest</i> (<i>Holland Advanced-Study Test</i>).....	112
4.2 Analisis Pemanfaatan Aplikasi <i>HoTest</i>	118
4.2.1 Analisis Kinerja (<i>Performance</i>).....	119
4.2.2 Analisis Informasi (<i>Information</i>).....	122
4.2.3 Analisis Ekonomi (<i>Economics</i>)	124
4.2.4 Analisis Pengendalian (<i>Control</i>)	125
4.2.5 Analisis Efisiensi (<i>Efficiency</i>)	126
4.2.6 Analisis Layanan (<i>Service</i>).....	126
4.3 Pembahasan	130
BAB V PENUTUP.....	133
5.1 Simpulan	133
5.2 Saran	134
DAFTAR PUSTAKA	136
LAMPIRAN.....	139

DAFTAR TABEL

Tabel 2.1 Tipe Diagram UML	33
Tabel 2.2 Elemen-Elemen dalam <i>Use Case Diagram</i>	34
Tabel 2.3 Elemen-Elemen dalam <i>Activity Diagram</i>	36
Tabel 3.1 Identifikasi Masalah dan Kebutuhan Pengguna.....	70
Tabel 3.2 Spesifikasi Perangkat Keras (<i>Hardware</i>).....	73
Tabel 3.3 Spesifikasi Perangkat Lunak	74
Tabel 3.4 Proses-Proses yang Terlibat dalam DFD Level 1	82
Tabel 3.5 Daftar Entitas dan Atribut pada Aplikasi <i>HoTest</i>	88
Tabel 3.6 <i>Source Code</i> Aplikasi <i>HoTest</i>	102
Tabel 3.7 Perbedaan Sistem Lama dan Sistem Baru.....	106
Tabel 4.1 Hasil Pengujian Kinerja Hak Akses <i>Testee</i> (Peserta Tes).....	119
Tabel 4.2 Hasil Pengujian Kinerja Hak Akses <i>Test-taker</i>	121
Tabel 4.3 Analisis Informasi Aplikasi <i>HoTest</i>	123

DAFTAR GAMBAR

Gambar 2.1 <i>Assembler</i> Menerjemahkan Instruksi-Instruksi Bahasa <i>Assembly</i> ke Kode Mesin	18
Gambar 2.2 Alur Pemrograman Bahasa Tingkat-Tinggi	20
Gambar 2.3 Lapisan-lapisan Konseptual sebuah Basis Data	25
Gambar 2.4 Model Sekuensial Linier	30
Gambar 2.5 Contoh Notasi <i>Class Diagram</i>	39
Gambar 2.6 Heksagon dari Holland	56
Gambar 2.7 Kerangka Berfikir Pengembangan Aplikasi	65
Gambar 3.1 Model Sekuensial Linier	66
Gambar 3.2 Alur Pelaksanaan Tes	68
Gambar 3.3 <i>Use Case Diagram</i> Tes Psikologi <i>HoTest</i>	75
Gambar 3.4 <i>Activity Diagram</i> “Start Test” Aplikasi Tes Psikologi <i>HoTest</i>	76
Gambar 3.5 <i>Activity Diagram</i> Prosedur Tes Aplikasi Tes Psikologi <i>HoTest</i>	77
Gambar 3.6 <i>Activity Diagram</i> About Test Aplikasi Tes Psikologi <i>HoTest</i>	78
Gambar 3.7 <i>Activity Diagram</i> “Form Test-taker” Aplikasi <i>HoTest</i>	79
Gambar 3.8 Diagram Konteks Aplikasi Tes Psikologi <i>HoTest</i>	81
Gambar 3.9 DFD Level 1	82
Gambar 3.10 <i>Sequence Diagram</i> Pengerjaan Soal Aplikasi <i>HoTest</i>	85
Gambar 3.11 <i>Sequence Diagram</i> Cetak Hasil Aplikasi <i>HoTest</i>	86
Gambar 3.12 <i>Analysis Packages</i> Aplikasi Tes Psikologi <i>HoTest</i>	87
Gambar 3.13 Relasi <i>Database</i> Aplikasi <i>HoTest</i>	89
Gambar 3.14 Tabel Identitas	90
Gambar 3.15 Tabel Jenis Tes	91
Gambar 3.16 Tabel Tema Tes	91
Gambar 3.17 Tabel Soal	92

Gambar 3.18 Tabel Rekap.....	92
Gambar 3.19 Tabel Total	93
Gambar 3.20 Tabel Rekap Akhir	93
Gambar 3.21 Tabel Kode Ringkas	94
Gambar 3.22 Tabel Userpass (<i>Username dan Password</i>)	94
Gambar 3.23 Tabel Minat Ekspresif	95
Gambar 3.24 Tabel Saran Studi	95
Gambar 3.25 Tabel Saran.....	96
Gambar 3.26 Arsitektur Desain Menu <i>Testee</i>	96
Gambar 3.27 Arsitektur Desain Menu <i>Test-taker</i>	97
Gambar 3.28 Desain <i>Interface</i> Halaman Awal Aplikasi.....	98
Gambar 3.29 Desain <i>Interface</i> Halaman Prosedur Pengerjaan Aplikasi	98
Gambar 3.30 Desain <i>Interface</i> Halaman Menu Aplikasi	99
Gambar 3.31 Desain <i>Interface</i> Halaman Pengisian Data Diri	99
Gambar 3.32 Desain <i>Interface</i> Halaman Pengisian Data Diri	100
Gambar 3.33 Desain <i>Interface</i> Halaman Login <i>Test-taker</i>	100
Gambar 3.34 Desain <i>Interface</i> Halaman Data <i>Testee</i> dan Cetak Hasil.....	101
Gambar 3.35 Desain <i>Interface</i> Halaman Input Hasil Pengerjaan	101
Gambar 3.36 Flowchart Penghitungan Hasil Aplikasi <i>HoTest (Holland Advanced-Study Test)</i>	111
Gambar 4.1 Halaman Awal Aplikasi <i>HoTest</i>	113
Gambar 4.2 Halaman Prosedur Penggunaan Aplikasi <i>HoTest</i>	113
Gambar 4.3 Halaman Menu Aplikasi <i>HoTest</i>	114
Gambar 4.4 Halaman Pengisian Identitas Diri.....	115
Gambar 4.5 Halaman Soal Aplikasi <i>HoTest</i>	116
Gambar 4.6 Halaman Menu <i>Test-taker</i> Input Hasil Pengerjaan	117
Gambar 4.7 Halaman Menu <i>Test-taker</i> Cetak Hasil	117

Gambar 4.8 <i>Database</i> Aplikasi <i>HoTest</i>	118
Gambar 4.9 Hasil Penghitungan Data secara Manual.....	128
Gambar 4.10 Hasil Penghitungan Data Menggunakan Aplikasi <i>HoTest</i>	129

DAFTAR LAMPIRAN

Lampiran 1. Surat Usulan Topik Skripsi.....	140
Lampiran 2. Surat Usulan Pembimbing Skripsi.....	141
Lampiran 3. Surat Keputusan Dosen Pembimbing	142
Lampiran 4. Surat Penelitian.....	143
Lampiran 5. Surat Persetujuan Kerahasiaan	144
Lampiran 6. Surat <i>Review</i> Hasil Pengujian.....	145
Lampiran 7. Surat Selesai Penelitian dan Pengujian.....	146
Lampiran 8. Dokumentasi.....	148

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini, teknologi dan komunikasi berkembang semakin cepat dari waktu ke waktu. Khususnya perkembangan teknologi komputer dalam penggunaan dan pemanfaatannya diberbagai bidang kehidupan yang mempermudah masyarakat pada umumnya dan individu pada khususnya dalam menunjang kegiatan sehari-hari. Salah satu contoh perkembangan teknologi adalah penggunaan dan pemanfaatan teknologi komputer dalam beberapa cabang ilmu pengetahuan salah satunya adalah psikologi (Budiman, 2009: 2).

Menurut William James (1980), psikologi merupakan ilmu pengetahuan yang mempelajari tentang kondisi mental. Menurut John B. Watson (1919), psikologi merupakan ilmu alam tentang tingkah laku manusia berupa perbuatan dan ucapan sebagai pokok masalah. Sedangkan menurut Richard Meyer (1981), psikologi merupakan analisa ilmiah tentang proses mental dan struktur daya ingat yang digunakan dalam memahami tingkah laku manusia. Sehingga di dalam bukunya, Atkinson, et al (1997: 18-19) mendefinisikan psikologi sebagai studi ilmiah terkait proses perilaku dan mental manusia.

Pada dasarnya ilmu psikologi bertujuan untuk memahami sesama manusia sehingga merupakan ilmu pengetahuan yang sangat luas. Dengan kondisi tersebut, tidak menutup kemungkinan untuk penerapan teknologi dalam cabang ilmu tersebut. Namun penggunaan teknologi dalam cabang ilmu tersebut dirasakan masih sangat kurang (Budiman, 2009: 2)

Penerapan teknologi dalam cabang ilmu psikologi salah satunya adalah dalam tes psikologi. Tes psikologi merupakan sebuah instrumen untuk mengukur kemampuan psikologi suatu subyek (Fitriani, 2012: 190). Tes psikologi berfungsi untuk mengukur beberapa perbedaan antar individu atau reaksi individu yang sama terhadap situasi yang berbeda dan bertujuan untuk mengukur tingkah laku seseorang (Anastasi, Urbina, 2007: 3).

Dalam praktiknya, tes psikologi dalam memahami suatu objek yaitu manusia dengan segala sikap dan tingkah lakunya masih menggunakan metode lama. Tes psikologi masih menggunakan lembaran-lembaran questioner atau serangkaian pertanyaan yang diberikan kepada objek terkait. Objek diminta untuk menjawab pertanyaan-pertanyaan yang ada kemudian dikumpulkan kembali dan dihitung serta ditarik sebuah kesimpulan dari hasil pengerjaan. Tentunya hal tersebut akan memakan waktu yang lama dan kurang efektif (Budiman, 2009: 2).

Tes yang dikomputerisasikan memberikan keuntungan yang lebih besar dibandingkan dengan tes yang menggunakan pensil dan kertas atau tes secara manual. Keuntungan yang didapatkan dari kondisi tersebut juga didiskusikan dalam beberapa artikel dan paper (Olson-Buchanan & Drasgrow, 1999, for an overview). Selain itu, tes yang dikomputerisasikan dapat meningkatkan kualitas dari segi keamanan (Drasgrow, 2002; Potenza, 2002; Way, Steffen, & Anderson, 2002). Sebagai contoh, tes yang dikomputerisasikan akan memberikan keuntungan dalam pemeliharaan isi yang dapat digunakan sewaktu-waktu ketika dibutuhkan (Foster, 2010: 208). Manfaat lain dari penggunaan komputer telah

menggambarkan peningkatan yang luar biasa dalam hal kecepatan proses analisis data dan skoring yang dijalankan (Anastasi, Urbina, 2007: 83).

Tes psikologi yang dikomputerisasikan tidak hanya seperti game yang dapat digunakan tanpa dibatasi oleh waktu dan tempat. Dari beberapa keuntungan yang ada yaitu dalam segi bahan yang dapat tersimpan atau tidak sekali pakai, penggunaan yang sepenuhnya otomatis, hasil yang dapat disajikan dengan lebih cepat, skor yang dapat disimpan dan dapat digunakan sewaktu-waktu untuk penelitian, dan pengambil tes mendapatkan keuntungan dalam menerima umpan balik secara langsung dari hasil tes yang dikerjakan, terdapat beberapa hal yang harus diperhatikan yaitu berupa batasan dalam pembuatan tes yang dikomputerisasikan (Barak dan English, 2002: 67-68).

Beberapa hal yang harus diperhatikan dalam pembuatan tes yang dikomputerisasikan diantaranya adalah semua tes tidak bisa diterapkan dalam komputer terutama dalam pengambilan point perfomansi yaitu tidak dalam bentuk *multiple choice* tetapi dari psikiater langsung karena dalam penskoran harus melihat kondisi responden secara langsung. Permasalahan lain dalam pembuatan tes yang dikomputerisasikan adalah terkait akurasi dalam penafsiran skor yang disesuaikan dengan model tes yang dilakukan (Barak dan English, 2002: 68).

Beberapa permasalahan tersebut menjelaskan bahwa dalam pembuatan tes yang dikomputerisasikan harus memperhatikan etika dalam pembuatannya. Dalam arti, penggunaan dan pembuatan tes-tes psikologi harus dikendalikan. Seperti yang ditulis dalam buku Anastasi dan Urbina (2007: 11) terdapat suatu komentar yang melatarbelakangi pengendalian tes-tes psikologi salah satunya, "Teman

sekamar saya sedang belajar psikologi. Ia memberi saya tes kepribadian dan hasilnya saya disebut neurotik. Setelah itu saya malu masuk kelas”. Komentar tersebut menggambarkan kemungkinan dalam penyalahgunaan tes-tes psikologi sehingga seseorang akan menyatakan tes tersebut akan melukai seseorang. Tes-tes psikologi harus digunakan dengan tepat dan efektif. Salah satunya oleh penguji yang memenuhi syarat agar tidak berdampak buruk. Pengendalian harus dilakukan dalam penggunaan tes-tes psikologi untuk memastikan bahwa tes diberikan oleh penguji yang memenuhi syarat dan mencegah kedekatan orang dengan isi tes yang dapat membuat tes tersebut tidak valid.

Permasalahan lain dalam pembuatan aplikasi *psychological testing* adalah tes harus lengkap dengan *security* dan tetap dalam pengawasan seorang ahli. Tes psikologi berbasis *online* tidak memiliki keamanan yang terjamin. Kecurangan sangat mungkin terjadi dan mudah untuk dilakukan. Isi dapat dengan mudah ditangkap dan disebarluaskan. Masalah etika tersebut disebabkan karena tes secara *online* disajikan dengan keamanan yang relatif kecil bahkan tidak ada sama sekali (Foster, 2010: 213).

Jenis tes psikologi sangat beragam dengan fungsi dan kegunaan yang berbeda. Berdasarkan aspek yang diukur tes psikologi terbagi atas tes inteligensi, tes bakat, tes kepribadian, tes prestasi, dan tes minat (Nur'aeni, 2012: 22). Menurut Anastasi dan Urbina (2007: 426), tes minat menjadi sangat penting dikarenakan hakikat dan kekuatan dari minat dan sikap seseorang merupakan aspek yang penting dari kepribadian. Tes minat mengungkapkan reaksi seseorang terhadap berbagai situasi yang secara keseluruhan akan mencerminkan minatnya.

Minat yang terungkap melalui tes akan menunjukkan minat yang lebih mewakili daripada minat yang hanya sekedar dinyatakan yang biasanya bukan minat dari seseorang yang sesungguhnya. (Nur'aeni, 2012: 23).

Individu akan dapat berkembang dengan maksimal apabila berada dalam lingkungan kerja yang tepat yaitu lingkungan pekerjaan yang sesuai dengan kepribadiannya. Holland (Sharf, 2006) mengatakan bahwa individu mengekspresikan diri, minat, dan nilai melalui pilihan pekerjaan maupun pengalaman yang mereka lalui. Sehingga pemilihan karir yang tepat akan memaksimalkan potensi yang ada dalam diri seseorang (Lubis, 2008: 1).

Karakteristik dari tes minat secara material memengaruhi prestasi pendidikan dan pekerjaan, hubungan antarpribadi, kesenangan yang didapatkan dari seseorang dari aktivitas di waktu luang, dan fase-fase utama lainnya dari kehidupan sehari-hari. Tes minat sangat penting karena berhubungan dengan minat-minat pekerjaan khususnya bagi usia sekolah. Studi tentang minat mendapatkan dorongan terkuat dari penaksiran pendidikan dan karir. Penaksiran karir dimaksudkan untuk membantu individu untuk memilih karir yang paling tepat dari segi kemampuan, minat, sasaran, nilai, dan temperamen pribadi dan juga persyaratan dari suatu pekerjaan. Menurut Super & Sverko (1995) dalam Anastasi dan Urbina (2007: 579), beberapa area kehidupan sama pentingnya bagi seseorang seperti pekerjaan yang tidak hanya dikarenakan oleh jumlah waktu yang diluangkan orang pada waktu bekerja namun juga dikarenakan pekerjaan memberikan kesempatan untuk mendapatkan ganjaran intrinsik ataupun ekstrinsik. Dari sudut pandang pengetesan, instrumen yang paling dapat

diterapkan dalam konseling karir individu yaitu inventori minat. Karena inventori minat pada dasarnya menaksir sistem nilai individu, inventori ini semakin lama dianggap sebagai fokus perencanaan karir yang efektif (Anastasi dan Urbina, 2007: 579).

Tes minat menjadi salah satu dari beberapa jenis tes psikologi yang penting karena digunakan pada anak-anak usia sekolah dan usia pekerja yang pada umumnya digunakan dalam 4 bidang terapan yaitu konseling karir bagi siswa sekolah lanjutan, konseling pekerjaan bagi karyawan, penjurusan siswa sekolah lanjutan atau mahasiswa, dan perencanaan bacaan dalam pendidikan dan latihan. Sehingga dengan tes minat tersebut dapat membantu individu baik untuk usia sekolah lanjutan ataupun pekerja agar dapat memilih karir yang paling tepat sesuai dengan kemampuan dan kepribadian yang dimiliki (Nur'aeni, 2012: 34).

Remaja yang telah menyelesaikan pendidikan Sekolah Menengah Atas (SMA) memiliki kebutuhan untuk menentukan jurusan yang ditempuh untuk dapat mencapai karir yang sesuai. Erikson (Sharf, 2006) menyatakan bahwa perkembangan psikososial pada masa remaja akhir akan berada pada sebuah kondisi yang membawa individu pada penetapan jati diri atau *role confusion*. Pada masa tersebut individu akan berada dalam pencarian identitas diri yang jelas serta peran yang akan dibawa nantinya. Seiring dengan perkembangan fisiknya, remaja perlu menyadari bahwa perlunya penentuan karir yang akan berpengaruh pada pada seluruh sisa hidup yang akan dijalannya. Bimbingan karir pada masa ini meliputi pengukuran minat, kemampuan, dan nilai yang dimiliki oleh masing-masing individu. Pada umumnya, untuk bimbingan karir tersebut remaja akan

mendatangi psikolog atau konselor pendidikan untuk mendapatkan bantuan melalui diskusi terkait dunia kerja sehingga konselor menemukan pengukuran dan konseling yang dapat membantunya (Lubis, 2008: 1).

Saat ini banyak ditemukan alat ukur psikologis yang dapat membantu individu dalam masa tersebut. Salah satu alat ukur minat dalam pengembangan karir adalah dari Holland yang mengembangkan alat ukur minat dengan dasar teori Heksagonal yang dapat membantu psikolog ataupun praktisi pendidikan dalam membantu untuk menemukan karir yang sesuai (Lubis, 2008: 1-2).

Sebagian besar inventori minat dirancang untuk menaksir minat individu dalam berbagai bidang pekerjaan. Selain itu, sejumlah inventori juga memberikan analisis minat dalam kurikulum pendidikan atau bidang studi, yang pada gilirannya terkait dengan keputusan karier yang akan dipilihnya. Salah satu inventori minat yang tersedia dewasa ini adalah SDS (*Self-Directed Search*). SDS merupakan salah satu dari pendekatan terhadap penaksiran minat-minat pekerjaan yang dikembangkan oleh Holland. SDS atau *Self-Directed Search* lebih banyak digunakan karena keringkasan dan kesederhanannya yang dapat memperluas pilihan karir individu. SDS dirancang sebagai instrumen konseling pekerjaan yang dapat dilaksanakan sendiri, diskor sendiri, dan diinterpretasikan sendiri. Meskipun SDS dirancang untuk bersifat bisa menentukan skor sendiri, namun tetap harus dilakukan pengawasan dan pemeriksaan skor. Sehingga kembali kepada permasalahan awal terkait penggunaan teknologi dalam tes psikologi bahwa segala bentuk tes psikologi harus tetap dalam pengawasan seorang ahli agar tidak

terjadi kecurangan dan kesalahan penafsiran hasil (Anastasi dan Urbina, 2007: 441).

Aplikasi dibuat dalam pemrograman berbasis desktop dikarenakan dapat mempermudah pengambil tes dalam mengendalikan proses berlangsungnya tes dan meminimalisir tindak kecurangan karena pada dasarnya aplikasi yang dibuat untuk mempermudah kedua belah pihak yaitu pengambil tes dan responden terkait dalam pelaksanaan tes kepribadian tersebut. Diharapkan pembuatan aplikasi tes minat tersebut dapat memecahkan permasalahan yang ada dalam cabang ilmu psikologi tersebut.

1.2 Rumusan Masalah

1. Bagaimana membangun aplikasi tes psikologi “HoTest” dengan menerapkan alat ukur SDS (*Self-Directed Search*) untuk mengukur minat karir individu?
2. Bagaimana validitas skoring aplikasi tes psikologi “HoTest” dengan menerapkan alat ukur SDS (*Self-Directed Search*) untuk menaksir minat karir individu berbasis komputer berdasarkan pengujian aplikasi menggunakan metode analisis PIECES (*Performance, Information, Economic, Control, Efficiency, and Service*)?

1.3 Tujuan Penelitian

Berdasarkan permasalahan yang diungkapkan di atas, maka dapat dirumuskan tujuan penelitian yang akan dilakukan. Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang dan membangun aplikasi dalam pengimplementasian SDS (*Self-Directed Search*) untuk tes minat ke dalam sebuah aplikasi berbasis java desktop.
2. Menguji validitas skoring dari aplikasi tes minat yang menerapkan alat ukur *Self-Directed Search* (SDS) berdasarkan analisis pemanfaatan aplikasi menggunakan metode analisis PIECES (*Performance, Information, Economic, Control, Efficiency, and Service*).

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi:

1. Bagi *test-taker*, aplikasi *HoTest* dengan menggunakan SDS dapat mempermudah pengambil tes dalam mengukur minat-minat pekerjaan dalam hal ini minat studi lanjut sehingga dapat membantu individu dalam memilih pendidikan yang paling tepat sesuai dengan kemampuan, minat, sasaran, nilai, dan kepribadiannya.
2. Bagi responden, bahwa aplikasi tes minat nantiya dapat digunakan untuk membantu individu dalam memilih studi lanjut yang paling tepat sesuai dengan kemampuan, minat, dan kepribadiannya dengan mudah. Tes minat dalam bentuk aplikasi akan lebih mudah digunakan dengan hasil

yang didapatkan akan lebih cepat karena penghitungan hasil penskoran secara otomatis dapat dilakukan. Selain itu, aplikasi tes minat yang dibuat akan distandarkan dengan ketentuan dan etika dalam pelaksanaan tes minat yang ada sehingga hasil yang didapatkan akan valid.

1.5 Batasan Masalah

Pada perencanaan pembuatan aplikasi tes minat berbasis java desktop dengan mengimplementasikan SDS (*Self-Directed Search*) dalam tes untuk pengukuran minat pekerjaan tersebut sangat kompleks. Sehingga dibutuhkan batasan masalah dari penelitian yang akan dilakukan. Berikut beberapa masalah yang dibatasi:

1. Tes minat yang diterapkan dalam aplikasi berbasis java desktop terdiri atas soal-soal yang sudah distandarkan dan memiliki validitas sebagai alat ukur yang didapatkan dari Laboratorium Psikologi Universitas Negeri Semarang. Pemberian skor dan hasil tes juga disesuaikan dengan standar penghitungan yang ada pada tes minat pada umumnya.
2. Memfokuskan pada aplikasi berupa pengerjaan soal-soal yaitu pernyataan-pernyataan dan penghitungan hasil pengerjaan dengan output program berupa dokumen hasil pengerjaan. Dokumen hasil penskoran kasar untuk menjadi rahasia penguji yang dapat digunakan sewaktu-waktu untuk penelitian dan yang diberikan kepada responden berupa hasil yang sudah mengalami penghitungan dan sudah digabungkan dengan tes-tes yang lain. Jadi aplikasi tes minat yang dibuat tetap disesuaikan dengan standar dalam tes pengukuran minat pada umumnya sehingga dalam pembuatan aplikasi

tersebut juga mendapat pendampingan dari pakar psikologi secara langsung.

3. Hasil penelitian hanya sampai pada perancangan, pembuatan, dan pengujian untuk mengetahui validitas skoring aplikasi *HoTest (Holland Advanced-Study Test)*.

1.6 Sistematika Penulisan Skripsi

Secara garis besar, sistematika skripsi ini terbagi menjadi tiga bagian, yaitu bagian awal, bagian isi dan bagian akhir.

1. Bagian awal berisi halaman judul, lembar persetujuan pembimbing, lembar pengesahan, lembar pernyataan keaslian, motto dan persembahan, abstrak, kata pengantar, daftar isi, daftar tabel, daftar gambar, dan daftar lampiran.
2. Bagian isi skripsi terdiri atas lima bab, yaitu :

BAB I : PENDAHULUAN

Bab ini merupakan bagian pendahuluan yang menjelaskan latar belakang dibuatnya Aplikasi Tes Psikologi yaitu tes minat dengan menerapkan alat ukur SDS (*Self-Directed Search*), rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan skripsi.

BAB II : KAJIAN PUSTAKA

Bab ini berisi tentang konsep dasar perancangan dan pembangunan perangkat lunak dalam hal ini aplikasi tes psikologi yaitu tes minat dengan menerapkan alat ukur SDS (*Self-Directed*

Search) yang terdiri atas teori pemrograman komputer, bahasa pemrograman, basis data, pemodelan perangkat lunak yaitu model *waterfall*, pemodelan sistem dengan menggunakan bahasa pemodelan UML (*Unified Modelling Language*), teori psikologi, teori tes minat, teori Holland, dan alat ukur SDS (*Self-Directed Search*). Di akhir bab II dijelaskan tentang kerangka berpikir dalam pembangunan aplikasi.

BAB III : METODE PENELITIAN

Bab ini berisi metode penelitian yang digunakan yang terdiri atas tahap analisis, desain, pengkodean, dan pengujian. Pada tahap analisis terdiri atas studi pendahuluan yang berisi waktu dan tempat penelitian, metode pengumpulan data dengan menggunakan menggunakan metode wawancara, observasi, dan dokumentasi, identifikasi masalah dan kebutuhan pengguna di Laboratorium Psikologi Universitas Negeri Semarang, perangkat keras dan perangkat lunak yang digunakan dalam pembuatan aplikasi, dan model analisis yang merupakan tahap pemodelan sistem menggunakan UML (*Unified Modelling Language*) yang terdiri atas *Scenario-Based Modelling* dengan membuat *Use Case Diagram* dan *Activity Diagram*, *Flow-Oriented Modelling* dengan membuat DFD (*Data Flow Diagram*), *Class-Oriented Modelling* dengan membuat *analysis packages*, serta *Behavioral Modelling* dengan membuat *Sequence Diagram*. Tahap selanjutnya adalah

desain perangkat lunak yang akan dibangun yang terdiri atas desain data, desain arsitektur, dan desain *interface*. Setelah tahap desain adalah tahap pengkodean yaitu penerjemahan desain ke dalam mesin menggunakan NetBeans 7.4 dan SQLite untuk pembuatan *database*. Tahap terakhir adalah pengujian yang dilakukan menggunakan metode analisis pemanfaatan sistem yaitu metode PIECES.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi tentang hasil penelitian yaitu pengujian yang dilakukan dengan menggunakan metode analisis pemanfaatan aplikasi yaitu metode PIECES dengan menguji aplikasi menggunakan 6 variabel yaitu *Performance, Information, Economics, Control, Efficiency, dan Service*. Pada analisis pemanfaatan aplikasi *Service*, dilakukan pengujian di Laoratorium Psikologi Universitas Negeri Semarang untuk mengetahui validitas skoring aplikasi.

BAB V : PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian dan saran untuk perbaikan dalam pengembangan aplikasi selanjutnya.

3. Bagian akhir berisi Daftar Pustaka dan Lampiran-Lampiran.

BAB II

KAJIAN PUSTAKA

2.1 Komputer

2.1.1 Pemrograman Komputer

Komputer merupakan *device* elektronik yang dapat menyimpan dan mengolah data. Komputer terdiri atas perangkat keras dan perangkat lunak. Perangkat lunak menyediakan instruksi-instruksi untuk mengendalikan elemen-elemen fisik komputer. Menulis instruksi-instruksi yang dapat memerintahkan komputer untuk menjalankan sesuatu disebut memprogram komputer (Sianipar, 2013: 2).

Di dalam buku karya Pressman (1997: 10), terdapat beberapa definisi perangkat lunak sebagai berikut:

- a. Perangkat lunak adalah perintah (program komputer) yang apabila dieksekusi akan memberikan fungsi dan unjuk kerja seperti yang diinginkan.
- b. Perangkat lunak adalah struktur data yang memungkinkan program memanipulasi informasi secara proporsional.
- c. Perangkat lunak adalah dokumen yang menggambarkan operasi dan kegunaan program.

Menurut Pressman (1997: 10-13), perangkat lunak memiliki beberapa karakteristik atau ciri yang berbeda dari perangkat keras sebagai berikut:

- a. Perangkat lunak dibangun dan dikembangkan, bukan dibuat dalam bentuk yang klasik.

- b. Perangkat lunak tidak pernah usang. Dalam hal ini, perangkat lunak tidak rentan terhadap pengaruh lingkungan yang merusak yang menyebabkan perangkat keras akan menjadi usang.
- c. Sebagian besar perangkat lunak dibuat secara *custom-built*, serta tidak dapat dirakit dari komponen yang sudah ada. Dalam hal ini memang memungkinkan untuk memesan perangkat lunak secara terpisah, tetapi tetap merupakan satu kesatuan yang lengkap bukan sebagai komponen yang dapat dipasangkan dalam program-program baru.

Menurut Fauzi (2008: 82), perangkat lunak (software) komputer secara umum memiliki fungsi sebagai berikut:

- a. Melakukan aktivitas bersama dengan *hardware*.
- b. Menyediakan semua sumber daya yang dapat digunakan pada sebuah komputer.
- c. Bertindak sebagai perantara antara *user* dengan *hardware* untuk melakukan aktivitas dengan perintah yang harus dilakukan dalam *software* komputer.

Secara garis besar software komputer dibagi ke dalam dua bagian yaitu *software* sistem operasi dan *software* aplikasi. Software aplikasi digolongkan menjadi bahasa pemrograman, program aplikasi, program paket, dan program utilitas (Fauzi, 2008: 82-83).

Software aplikasi merupakan *software* program yang mempunyai aktifitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna

dengan tujuan tertentu. Untuk mendukung operasi perangkat lunak, tugas pengguna komputer dibagi ke dalam beberapa bagian yaitu (Fauzi, 2008: 90-91)

- a. Analis sistem, yaitu orang yang bertugas mengidentifikasi kebutuhan, studi kelayakan, dan melakukan batasan-batasan analisis terhadap sistem yang akan diimplementasikan dalam suatu pemrograman.
- b. *Programmer*, yaitu orang yang bertugas membuat atau mengimplementasikan sistem yang dirancang dalam bentuk pemrograman komputer.
- c. Operator, yaitu orang yang bertugas mengoperasikan atau menjalankan komputer sesuai dengan sistem yang telah dirancang.
- d. *Administrator database* adalah orang yang bertugas melakukan definisi dan manipulasi *software database*.
- e. Administrator jaringan yaitu orang yang bertugas mengelola *software* sistem operasi, *software* aplikasi yang digunakan dalam jaringan komputer.

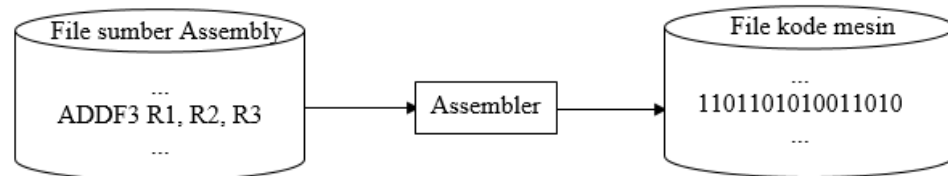
Program komputer atau dikenal sebagai *software* merupakan sekumpulan instruksi yang mendeskripsikan segala sesuatu yang harus dilakukan oleh komputer. Komputer tidak dapat memahami bahasa manusia sehingga dibutuhkan bahasa komputer dalam program. Pemrograman adalah pembuatan suatu program yang dapat dieksekusi oleh komputer sehingga dapat melaksanakan tugas yang diperintahkan. Bahasa *native* komputer merupakan bahasa mesin. Bahasa mesin merupakan sekumpulan instruksi primitif yang memiliki format kode biner. Sehingga dalam memberikan instruksi pada mesin untuk melakukan sesuatu, *user*

harus memberikan kode biner. Pemrograman menggunakan bahasa mesin ini sangat susah dipahami dan dimodifikasi (Sianipar, 2013: 8-9).

2.1.2 Bahasa Pemrograman

Komponen perangkat lunak dibangun menggunakan bahasa pemrograman yang memiliki kosakata yang terbatas, sebuah tata bahasa yang dibatasi secara eksplisit, dan aturan-aturan *syntax* dan semantik yang dibentuk secara baik. Pada tingkat yang paling rendah, bahasa tersebut mencerminkan serangkaian instruksi perangkat keras. Pada tingkat sedang, bahasa pemrograman seperti C dipakai untuk membuat deskripsi prosedural dari program. Pada tingkat yang paling tinggi, bahasa tersebut menggunakan ikon grafik atau simbol lain untuk mewakili kebutuhan untuk sebuah pemecahan. Instruksi-instruksi yang dapat dieksekusi dibuat secara otomatis (Pressman, 1997: 14).

Bahasa *Assembly* adalah bahasa pemrograman tingkat-rendah yang menggunakan instruksi berupa bahasa mesin. Bahasa *assembly* digunakan untuk mempermudah pemrograman. Komputer tidak bisa memahami bahasa *assembly* sehingga dibuat suatu program untuk mengonversi bahasa *assembly* ke kode-kode mesin. Namun karena bahasa mesin tergantung dari jenis mesin yang digunakan, program *assembly* tidak bisa dieksekusi ke semua jenis mesin. Sehingga berkembanglah bahasa tingkat-tinggi yang mengurangi ketergantungan terhadap mesin dan memberikan keuntungan dalam pemrograman atau pembuatan *software* yang akan lebih mudah.



Gambar 2.1 *Assembler* Menerjemahkan Instruksi-Instruksi Bahasa *Assembly* ke Kode Mesin (Sianipar, 2013: 9)

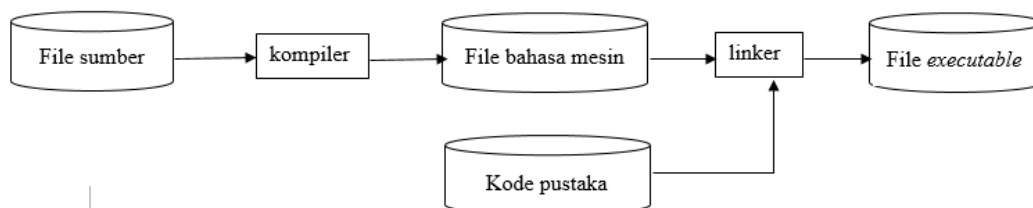
Terdapat banyak bahasa pemrograman tingkat-tinggi yang diciptakan. Dari seratus bahasa pemrograman tingkat-tinggi, beberapa bahasa pemrograman yang paling dikenal adalah sebagai berikut (Sianipar, 2013: 8-11):

- a. COBOL (*Common Business Oriented Language*), dirancang untuk pengolahan data bisnis.
- b. FORTRAN (*FORmula TRANslation*), didesain untuk komputasi matematik dan digunakan untuk komputasi numerik.
- c. BASIC (*Beginner's All-purpose Symbolic Instruction Code*), diciptakan agar dapat dipelajari dan dipahami dengan mudah.
- d. Pascal (dinamai dari ilmuwan Blaise Pascal).
- e. Ada (dinamai dari Ada Lovelace), dikembangkan untuk Departemen Pertahanan Amerika dan digunakan dalam proyek-proyek pertahanan dan militer.
- f. Bahasa C, menggabungkan kekuatan bahasa *Assembly* dengan kemudahan dan portabilitas bahasa tingkat-tinggi.
- g. Visual basic merupakan bahasa visual seperti-Basic dikembangkan oleh *Microsoft* yang digunakan untuk mengembangkan *interface* pengguna

grafikal (GUI, *graphical User Interface*) dan sebagai bahasa tingkat-tinggi yang berkembang dengan sangat cepat.

- h. Delphi, merupakan bahasa visual seperti Pascal yang dikembangkan oleh Borland dan digunakan untuk mengembangkan *interface* pengguna grafikal (GUI, *graphical User Interface*) dan sebagai bahasa tingkat-tinggi yang berkembang dengan sangat cepat.
- i. C++, merupakan bahasa pemrograman berorientasi objek yang dikembangkan dari C. C++ sangat populer untuk banyak proyek *software* seperti kompil器和 sistem operasi seperti Microsoft Windows yang dikembangkan dari C++.
- j. C# merupakan bahasa seperti Java yang dikembangkan oleh *Microsoft* untuk aplikasi-aplikasi yang berbasis *Microsoft.NET*.
- k. JAVA merupakan bahasa pemrograman yang dikembangkan oleh *Microsystem* dan digunakan untuk aplikasi internet.

Program yang menggunakan bahasa pemrograman tingkat-tinggi disebut kode sumber atau program sumber. Dikarenakan komputer tidak memahami kode sumber, sehingga digunakan kompil器和 untuk menterjemahkan menjadi bahasa mesin. Kemudian dari program bahasa mesin dihubungkan ke kode pustaka pendukung sehingga dihasilkan suatu file yang dieksekusi. *File executable* tersebut berekstensi .exe (Sianipar, 2013: 11).



Gambar 2.2 Alur Pemrograman Bahasa Tingkat-Tinggi (Sianipar, 2013: 11)

Dari beberapa macam bahasa pemrograman tingkat-tinggi di atas, Java merupakan salah satu bahasa pemrograman yang berkembang dengan cepat. Pemrograman Java tidak hanya digunakan untuk membuat dan mengembangkan aplikasi internet saja, namun juga digunakan untuk menciptakan dan mengembangkan aplikasi *desktop* maupun *mobile* (Sianipar, 2013: 14).

2.1.3 Java

Java dikembangkan oleh tim di Sun Microsystem yang diketuai oleh James Gosling. Java awal mulanya dikenal dengan Oak yang didesain pertama kali untuk chip peralatan elektronik pada tahun 1991. Kemudian pada tahun 1995 untuk mengembangkan aplikasi-aplikasi internet, Oak didesain ulang yang kemudian diberi nama Java. Java merupakan salah satu pemrograman yang sangat populer. Java berkembang dengan sangat cepat dan diterima oleh pengguna salah satunya dapat dilihat dari karakteristik perancangannya. Karakteristik perancangan yang dijanjikan oleh pengembang java adalah bahwa ketika menciptakan suatu program maka program tersebut dapat dijalankan dimana saja. Seperti yang dikutip dari Sun, “*Java is simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, high performance, multithreaded, and dynamic*”. Java memiliki banyak fitur dengan bahasa pemrograman yang dapat digunakan untuk

mengembangkan aplikasi-aplikasi tingkat tinggi. Saat ini Java tidak hanya digunakan untuk pemrograman Web, namun juga digunakan untuk menciptakan dan mengembangkan aplikasi *standalone*, *desktop*, maupun *mobile* (Sianipar, 2013: 14-15).

Menurut Tasmawati (2008: 4-6), beberapa kelebihan yang disebutkan sebelumnya dijelaskan sebagai berikut:

- a. *Simple* (sederhana dan ampuh), yaitu Java sangat mudah dipelajari terutama bagi programmer yang sudah mengenal pemrograman C/C++.
- b. *Object-oriented* (berorientasi objek), menjelaskan bahwa paradigma pemrograman berorientasi objek adalah paradigma pemrograman masa depan. Java bukan merupakan turunan dari bahasa pemrograman manapun. Java memiliki keseimbangan dalam menyediakan peng-class-an sederhana.
- c. *Secure* (aman), yaitu Java dirancang menjadi bahasa pemrograman yang handal dan aman. Java dikenal handal karena memiliki manajemen memori yang baik, dan dikatakan secure karena Java tidak memiliki fasilitas dengan keamanan yang rendah yang perlu ditangani secara khusus.
- d. *Robust* (kokoh), yaitu Java membatasi beberapa kunci sehingga saat mengembangkan program, kesalahan yang terjadi dapat ditemukan dengan cepat. Program akan dicek ketika dituliskan dan saat dijalankan sehingga akan menghemat waktu. Java adalah bahasa pemrograman yang luwes dan kokoh dibandingkan dengan bahasa pemrograman yang lain.

- e. Interaktif, yaitu Java memiliki kemampuan yang memungkinkan program dapat melakukan beberapa hal dalam waktu bersamaan.
- f. *Architecture neutral*, yaitu dalam menggunakan bahasa pemrograman Java, user dapat sepenuhnya percaya dengan pemrograman Java karena “tulis sekali, jalan dimana saja, kapan saja, dan selamanya”.
- g. *Interpreted & High performance*, yaitu Java dilengkapi dengan lintas-platform dengan kompilasi ke dalam representasi langsung yang disebut kode-byte Java (*Java byte-code*). Sehingga dapat diterjemahkan oleh sistem manapun yang memiliki program Java di dalamnya.

Java merupakan bahasa pemrograman yang tangguh dan handal pada banyak aplikasi. Terdapat tiga edisi Java yaitu Java SE (*Java Standard Edition*), Java EE (*Java Enterprise Edition*), dan Java ME (*Java Micro Edition*). Java SE digunakan dalam mengembangkan aplikasi-aplikasi applet. Java EE digunakan untuk mengembangkan aplikasi pada sisi server. Java ME digunakan untuk mengembangkan aplikasi pada *device* bergerak seperti *mobile device* (Sianipar, 2013: 19).

Dalam instalasi java, terdapat dua paket aplikasi yaitu JRE (*Java Runtime Environment*) dan JDK (*Java Development Kit*). Paket JRE (*Java Runtime Environment*) terdiri atas semua aplikasi yang diperlukan agar aplikasi Java dapat berjalan seperti *library* dan JVM (*Java Virtual Machine*). Paket JDK (*Java Development Kit*) merupakan paket yang terdiri atas JRE ditambah dengan perlengkapan untuk membuat aplikasi Java seperti *Java archive* atau *jar*, *Java documentation* atau *javadoc*, dan *Java compiler* atau *javac* (Bima, 2011).

2.1.4 Basis Data (*Database*)

Bahasa basis data pada umumnya ditempelkan dalam bahasa pemrograman lain seperti ditempelkan ke dalam bahasa pemrograman Java, C/C++, Pascal, Basic, Fortran, Ada, dan lain sebagainya. Bahasa tempat ditempelkannya instruksi bahasa dari basis data disebut dengan inang (*host language*). Dalam inang (*host language*) tempat menempelnya bahasa basis data, ketika *source code* program dikompilasi maka *source code* program tersebut dilewatkan dahulu ke *pre-compiler* yang menerjemahkan bahasa basis data ke bahasa inang (bahasa asli pemrograman inang). Kemudian kode-kode yang sudah berbentuk bahasa inang dikompilasi menggunakan kompilator sehingga menghasilkan kode-kode biner yang menghubungkan instruksi-instruksi bahasa basis data ke *library* yang berfungsi menghubungkan ke DBMS (*Database Management System*). Setelah itu, hasil instruksi bahasa basis data ditangkap oleh bahasa inang dan diolah sesuai dengan keperluan aplikasi (Setiyadi, 2010: 1).

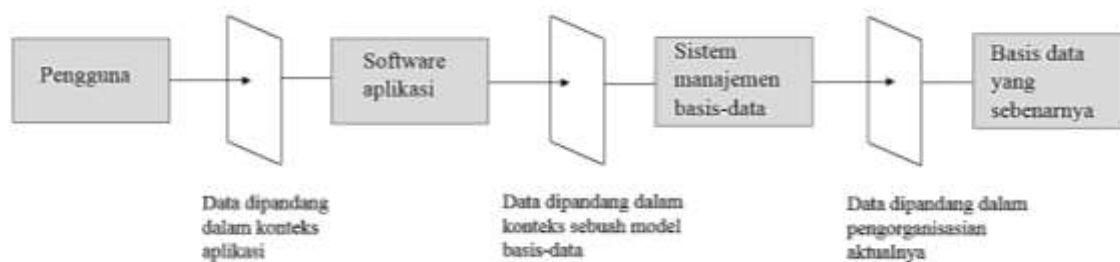
Secara historis, basis data telah berkembang menjadi sebuah sarana untuk mengintegrasikan sistem-sistem penyimpanan data. Bersamaan dengan semakin berkembangnya mesin-mesin komputer dalam manajemen informasi, setiap aplikasi cenderung diimplementasikan sebagai sebuah sistem yang terpisah dengan datanya sendiri (Brookshear, 2003: 392).

Dilihat dari beberapa sudut pandang, *database* memiliki beberapa pengertian sebagai berikut :

- Basis data merupakan kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengkonsolidasi banyak catatan yang sebelumnya disimpan dalam file yang terpisah (Indrajani, 2009: 2).
- Basis data merupakan suatu kumpulan data yang saling berhubungan secara logis dan deskripsi data, yang dirancang untuk memenuhi kebutuhan informasi yang dibutuhkan oleh suatu organisasi. Dalam arti, basis data merupakan tempat penyimpanan data yang besar yang dapat digunakan oleh banyak pengguna (Indrajani, 2009: 2).
- Menurut Kusri (2006) dalam Rusdianto (2012), basis data merupakan himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga nantinya dapat dimanfaatkan dengan cepat dan mudah.

Sebuah sistem basis data yang tipikal memuat dua lapisan *software*, lapisan aplikasi, dan lapisan manajemen basis data. *Software* aplikasi menangani komunikasi dengan pengguna. Sehingga *software* aplikasi yang menentukan karakter eksternal dari suatu sistem secara keseluruhan. *Software* aplikasi tidak secara langsung dalam memanipulasi basis data. Manipulasi aktual dari basis data dilakukan oleh lapisan *software* yang lain yang disebut *Database Management System* (DBMS). Setelah *software* aplikasi menentukan tindakan yang diminta oleh pengguna, *software* tersebut menggunakan DBMS sebagai perangkat abstrak untuk melaksanakan permintaan tersebut. Seperti permintaan menambahkan atau menghapus data, maka tugas DBMS untuk melaksanakan perintah tersebut. Dikotomi antara *software* aplikasi dan basis data membawa sejumlah manfaat.

Salah satunya adalah memungkinkan pembuatan dan pengguna perangkat abstrak, yang telah diketahui bersama dapat berperan sebagai sebuah konsep dalam menyederhanakan proses perancangan *software*.



Gambar 2.3 Lapisan-lapisan Konseptual sebuah Basis Data

(Brookshear, 2003: 395)

2.1.5 Pemodelan Perangkat Lunak

Rekayasa perangkat lunak (*software*) merupakan disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap spesifikasi sistem hingga pemeliharaan sistem setelah digunakan. Menurut Sommerville (2003: 7), terdapat dua istilah kunci untuk definisi tersebut:

- a. 'disiplin rekayasa', Perekayasa membuat suatu alat bekerja. Perekayasa menerapkan teori, metode, dan alat bantu yang sesuai.
- b. 'semua aspek produksi perangkat lunak' bahwa rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak, namun juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode, dan teori untuk mendukung produksi perangkat lunak.

Proses perangkat lunak merupakan serangkaian kegiatan dan hasil-hasil relevannya yang menghasilkan perangkat lunak. Kegiatan ini biasa dilaksanakan

oleh perancang perangkat lunak. Menurut Sommerville (2003:8), terdapat empat proses dasar yang umum bagi seluruh kegiatan proses perangkat lunak yaitu:

- a. Spesifikasi perangkat lunak. Fungsionalitas perangkat lunak dan batasan kemampuan operasinya harus didefinisikan.
- b. Pengembangan perangkat lunak. Perangkat lunak yang memenuhi spesifikasi harus diproduksi.
- c. Validasi perangkat lunak. Perangkat lunak harus divalidasi sehingga menjamin bahwa perangkat lunak memenuhi segala sesuatu yang diinginkan pelanggan.
- d. Evolusi perangkat lunak. Perangkat lunak harus berkembang untuk memenuhi kebutuhan pelanggan.

Model proses perangkat lunak merupakan deskripsi yang disederhanakan dari proses perangkat lunak yang dipresentasikan dengan sudut pandang tertentu. Model, sesuai sifatnya merupakan penyederhanaan sehingga merupakan abstraksi dari proses sebenarnya yang dideskripsikan. Menurut Sommerville (2003: 8-9), beberapa jenis model proses perangkat lunak yang dapat dihasilkan diantaranya adalah:

- a. Model aliran kerja (*workflow*), menunjukkan urutan kegiatan pada proses bersama dengan input, output, dan ketergantungannya.
- b. Model aliran data (*data flow*) atau kegiatan, merepresentasikan proses sebagai satu set kegiatan yang masing-masing melakukan transformasi data. Model ini menunjukkan input ke proses.

- c. Model peran/aksi, merepresentasikan peran orang yang terlibat pada proses perangkat lunak dan kegiatan yang menjadi tanggung jawab orang tersebut.

Menurut Sommerville (2003: 9), model atau paradigma umum dalam pengembangan perangkat lunak yaitu sebagai berikut:

- a. Pendekatan air terjun (*waterfall*). Cara ini menggunakan kegiatan-kegiatan dan merepresentasikannya sebagai fase proses yang berbeda seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian, dan seterusnya. Setelah setiap tahap didefinisikan, tahap tersebut 'diakhiri' dan pengembangan ke tahap berikutnya.
- b. Pengembangan evolusioner. Pendekatan ini bertumpu dengan kegiatan spesifikasi, pengembangan, dan validasi. Sistem awal dikembangkan dengan cepat dari spesifikasi yang sangat abstrak yang kemudian disempurnakan dengan masukan dari pelanggan untuk menghasilkan sistem yang memenuhi kebutuhan pelanggan. Sistem kemudian diserahkan. Sebagai alternatif, sistem tersebut dapat diimplementasikan ulang dengan menggunakan pendekatan yang lebih terstruktur untuk menghasilkan sistem yang berbobot dan dapat dipelihara.
- c. Transformasi formal. Pendekatan ini menghasilkan suatu sistem matematis yang formal dan mentransformasikan spesifikasi ini, menggunakan metode matematik menjadi program. Sehingga pendekatan ini 'mempertahankan kebenaran'.

- d. Perakitan (*Assembling*) sistem dari komponen-komponen yang dapat dipakai ulang. Teknik ini menganggap bahwa bagian-bagian sistem sudah ada. Proses pengembangan sistem terfokus pada pengintegrasian bagian-bagian ini dan bukan pengembangan dari awal.

Menurut Sommerville (2001: 48-49), Semua model tersebut di atas memiliki kelebihan dan kekurangan, Sehingga untuk pembuatan sistem yang besar perlu digunakan berbagai pendekatan untuk berbagai bagian sistem, sehingga harus digunakan model hibrid. Terdapat dua model hibrid yang mendukung pendekatan yang berbeda terhadap pengembangan dan secara eksplisit telah dirancang untuk mendukung iterasi proses. Model-model tersebut adalah:

- a. Pengembangan inkremental, meliputi spesifikasi, perancangan, dan implementasi perangkat lunak dibagi ke serangkaian inkremen yang dikembangkan secara bergantian;
- b. Pengembangan spiral, yaitu pengembangan sistem seolah beralur membentuk spiral ke luar dari garis awal sampai sistem pengembangan akhir.

Model sekuensial linier atau *waterfall* merupakan paradigma yang paling luas dipakai dan paling tua. Paradigma siklus kehidupan klasik memiliki tempat terbatas namun penting di dalam kerja rekayasa perangkat lunak. Paradigma tersebut memberikan template yang menunjukkan bahwa metode analisis, desain, pengkodean, pengujian, dan pemeliharaan bisa dilakukan. Siklus kehidupan klasik tetap menjadi model bagi rekayasa perangkat lunak yang paling banyak

digunakan. Sekalipun memiliki kelemahan, secara signifikan model tersebut lebih baik daripada pendekatan yang sifatnya sembrono dalam pengembangan perangkat lunak (Pressman, 1997: 39).

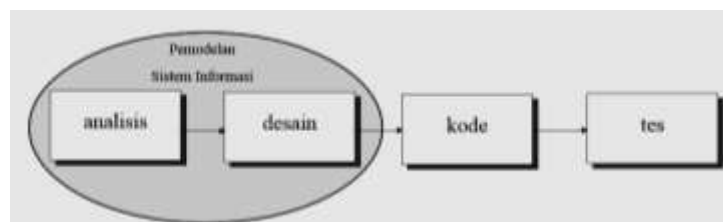
2.1.6 Model *Waterfall*

Tahapan dari fase pengembangan siklus hidup perangkat lunak adalah analisis, perancangan, implementasi, dan pengujian. Analisis merupakan fase pengembangan dalam siklus hidup perangkat lunak yang sarannya adalah mengidentifikasi beberapa hal yang harus mampu dilakukan oleh sistem yang diajukan. Perancangan merupakan tahap pembangunan dari sistem perangkat lunak. Implementasi merupakan tahap yang melibatkan penulisan aktual dari program-program, pembuatan file-file, maupun pembuatan *database*. Tahap terakhir adalah pengujian yang sangat erat kaitannya dengan implementasi karena setiap modul sistem biasanya diuji ketika diimplementasikan (Brookshear, 2003: 286-287).

Menurut Royce (1970), model pertama yang diterbitkan untuk proses pengembangan perangkat lunak diambil dari proses rekayasa lain. Karena penurunan dari fase ke fase lain, model ini disebut sebagai ‘model air terjun’ atau siklus hidup perangkat lunak. Model *waterfall* mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi dan evolusi, dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti persyaratan, perancangan perangkat lunak, implementasi, pengujian, dan seterusnya (Sommerville, 2003: 42-43).

Menurut Pressman (2005: 79), *“The waterfall model, sometimes called the classic life cycle, suggest a systematic, sequential approach to software development that begins with customer specifications of requirements and progresses through planning, modeling, construction, and deployment, culminating in on-going support of the completed software”*.

Model *waterfall* disebut juga sebagai model sekuensial linier atau model siklus kehidupan klasik. Sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Model sekuensial linier digambarkan sebagai berikut:



Gambar 2.4 Model Sekuensial Linier (Pressman, 1997: 37)

Tahapan-tahapan tersebut dijelaskan sebagai berikut :

1. Analisis

Proses pengumpulan kebutuhan difokuskan khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perencana harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan (Pressman, 1997: 38).

2. Desain

Desain perangkat lunak adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda yaitu struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat atau kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum pemunculan kode. Desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak (Pressman, 1997: 38).

3. Generasi Kode

Dalam tahap ini, desain diterjemahkan dalam bentuk mesin yang dapat dibaca. Langkah pembuatan kode melakukan tugas ini. Apabila desain dilakukan dengan cara yang lengkap, maka pembuatan kode dapat diselesaikan secara mekanis (Pressman, 1997: 28).

4. Tes atau Pengujian

Sekali kode dibuat, pengujian program dimulai. Pengujian difokuskan pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil yang aktual sesuai dengan hasil yang dibutuhkan (Pressman, 1997: 28).

Prinsip utama dari model *waterfall* yaitu bahwa dalam pengerjaan proyek dibagi ke dalam beberapa fase yang saling berurutan. Penekanan dilakukan saat perencanaan pembuatan, dan jadwal, deadline, biaya, serta implementasi sistem dilakukan sekaligus. Model *waterfall* mempunyai kemampuan untuk lebih mudah

dimengerti, mudah untuk digunakan, syarat dari sistem yang bersifat stabil, baik dalam manajemen kontrol, dan bekerja lebih baik. Hal tersebut dikarenakan kualitas lebih diutamakan dibandingkan biaya dan waktu deadline (Fahrurozi dan Azhari, 2008).

2.1.7 UML (*Unified Modelling Language*)

Rekayasa sistem merupakan sebuah proses pemodelan. Dalam membangun sebuah model sistem, perancang harus mempertimbangkan sejumlah faktor pembatasan sebagai berikut (Pressman, 1997: 280-281):

- a. Asumsi, yang mengurangi jumlah permutasi dan variasi yang mungkin, sehingga memungkinkan sebuah model mencerminkan masalah dengan cara yang dapat dipertanggungjawabkan.
- b. Penyederhanaan, yang memungkinkan model diciptakan dalam waktu yang tepat.
- c. Pembatasan yang membantu membatasi sistem.
- d. Batasan yang menunjukkan cara dari model tersebut diciptakan dan pendekatan dilakukan pada saat model diimplementasikan.
- e. Preferensi yang menunjukkan arsitektur yang dipilih untuk semua data, fungsi, dan teknologi.

UML (*Unified Modelling Language*) digunakan untuk pemodelan sistem. Seperti yang dikemukakan oleh Pressman dalam buku *Software Engineering : A Practitioner's Approach* (2005:167) Chapter 6, "*UML provides a wide array of diagrams that can be used for analysis and design at both the system and the software level*". Dalam kalimat tersebut dapat disimpulkan bahwa UML (*Unified*

Modelling Language) menyediakan beragam diagram yang dapat digunakan untuk analisis dan desain pada sistem dan perangkat lunak.

UML merupakan bahasa pemodelan yang konsisten, dengan sistem arsitektur yang bekerja dalam *Object Oriented Analysis and Design* (OOAD) untuk menentukan visualisasi, konstruksi, dan mendokumentasikan *artifact* dari sistem *software* (Sumarta et al., 2004: 2).

Menurut Whitten et al. (2007) dalam Putra (2014: 27), UML merupakan blue print dari suatu rancangan sistem. Fowler dan Scott (1999) menyatakan bahwa “*UML is the successor to the wave of object-oriented analysis and design (OOA&D) methods that appeared in the late ‘80s and early ‘90s’*”.

UML menyediakan banyak diagram yang dapat digunakan untuk mendefinisikan suatu aplikasi. Berikut ini merupakan tipe diagram UML menurut Munawar (2005) dalam Budiyati (2009) :

Tabel 2.1 Tipe Diagram UML

Diagram	Tujuan
Use Case	Bagaimana user berinteraksi dengan sebuah sistem.
Activity	Perilaku prosedural dan paralel.
Class	Class, fitur, dan relasinya.
Package	Struktur hierarki saat kompilasi.
Collaboration	Menggambarkan interaksi antar objek seperti sequence diagram tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message.
Sequence	Interaksi diantara objek, lebih menekankan pada urutan.

Component	Struktur dan koneksi dari komponen.
Deployment	Penyebaran atau instalasi ke klien.


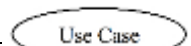
Sumber : Munawar (2005) dalam Budiwati (2009).

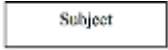

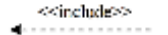
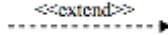

Dalam UML versi 2 terdapat beberapa diagram sebagai berikut :

a. *Use Case Diagram*

Merupakan diagram yang menggambarkan secara sederhana fungsi-fungsi utama dari sistem dan berbagai *user* yang akan berinteraksi dengan sistem tersebut. Berikut ini merupakan elemen-elemen dari *use case diagram* menurut Dennis, et al. (2010) dalam Desanti, et al. (2010).

Tabel 2.2 Elemen-elemen dalam Use Case Diagram

No.	Nama Elemen	Fungsi	Notasi
1	Actor	Menggambarkan tokoh atau sistem yang memperoleh keuntungan dan berada diluar sistem. <i>Actor</i> dapat berasosiasi dengan <i>actor</i> lainnya dengan menggunakan <i>specialization/superclass association</i> . <i>Actor</i> diletakkan di luar <i>object boundary</i> .	 Actor/Role
2	Use Case	Mewakili sebuah bagian dari fungsionalitas sistem dan	 Use Case

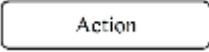
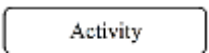
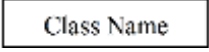


		ditempatkan dalam sistem <i>boundary</i> .	
3	Subject Boundary	Menyatakan lingkup dari <i>subject</i> .	
4	Association Relationship	Menghubungkan <i>actor</i> untuk berinteraksi dengan <i>use case</i> .	
5	Include Relationship	Menunjukkan <i>inclusion</i> fungsionalitas dari sebuah <i>use case</i> dengan <i>use case</i> lainnya. Arah panah dari <i>base use case</i> ke <i>included use case</i> .	
6	Extend Relationship	Menunjukkan <i>extension</i> dari sebuah <i>use case</i> untuk menambahkan <i>optional behaviour</i> . Arah panah dari <i>extension use case</i> ke <i>base use case</i> .	
7	Generalization Relationship	Menunjukkan generalisasi dari <i>use case</i> khusus ke yang lebih umum.	




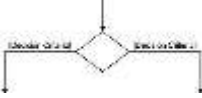
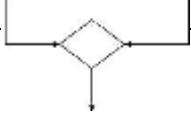
Sumber : Dennis, Alan, Wixom, Barbara H., Tegarden, David, 2010, *System Analysis and Design with UML An Object Oriented Approach*, 3rd ed, hal.174, dalam Desanti, et al. (2010).

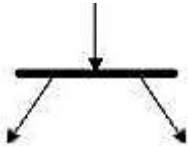
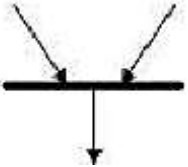

b. Activity Diagram

Activity Diagram digunakan untuk memodelkan perilaku. *Activity Diagram* dapat dilihat sebagai sebuah *sophisticated data flow diagram* (DFD) yang digunakan dalam analisis struktural. Namun berbeda dengan DFD, *activity diagram* mempunyai notasi untuk memodelkan aktivitas yang berlangsung secara paralel, bersamaan, dan juga proses pengambilan keputusan yang kompleks. Berikut ini merupakan elemen-elemen dari *activity diagram* menurut Dennis, et al. (2010) dalam Desanti, et al. (2010).

Tabel 2.3 Elemen-elemen dalam Activity Diagram

No.	Nama Elemen	Fungsi	Notasi
1.	Action	Untuk menggambarkan perilaku yang sederhana dan bersifat <i>non-decomposable</i> .	
2.	Activity	Untuk mewakili kumpulan aksi (<i>action</i>)	
3.	Object Mode	Untuk mewakili objek yang terhubung dengan kumpulan <i>object flow</i> .	
4.	Control Flow	Menunjukkan rangkaian pelaksanaan.	
5.	Object flow	Menunjukkan aliran	

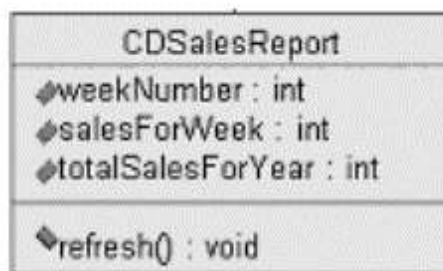
		sebuah objek dari sebuah aktivitas atau aksi ke aktivitas atau aksi lainnya.	
6.	Initial Node	Menandakan awal dari kumpulan aksi atau aktivitas.	
7.	Final-Activity Node	Untuk menghentikan seluruh <i>control flows</i> atau <i>object flows</i> pada sebuah aktivitas (atau aksi).	
8.	Final-Flow Node	Untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu.	
9.	Decision Node	Untuk mewakili suatu kondisi pengujian yang bertujuan untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya menuju ke satu arah.	
10.	Merge Node	Untuk menyatukan	

		kembali <i>decision path</i> yang dibuat dengan <i>decision node</i> .	
11.	Fork Node	Untuk memisahkan perilaku menjadi kumpulan aktivitas yang berjalan secara paralel atau bersamaan.	
12.	Join Node	Untuk menyatukan kembali kumpulan aktivitas yang berjalan paralel atau bersamaan.	
13.	Swimlane	Untuk membagi sebuah <i>activity diagram</i> menjadi kolom guna menempatkan aktivitas atau aksi tertentu pada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas atau aksi tersebut.	

Sumber : Dennis, Alan, Wixom, Barbara H., Tegarden, David, 2010, System Analysis and Design with UML An Object Oriented Approach, 3rd ed, hal.160-164, dalam Desanti, et al. (2010).

c. Class Diagram

Class Diagram merupakan notasi yang paling mendasar pada UML. *Class Diagram* berupa notasi untuk merepresentasikan suatu *class* beserta atribut dan operasinya (Haviluddin, 2011: 6).



Gambar 2.5 Contoh Notasi Class Diagram

d. Sequence Diagram

Sequence Diagram merupakan diagram yang menjelaskan interaksi objek yang disusun berdasar urutan waktu. *Sequence Diagram* menggambarkan tahapan termasuk urutan perubahan yang dilakukan untuk menghasilkan sesuatu yang sesuai dengan *use case diagram* (Haviluddin, 2011: 5).

2.2 Psikologi

2.2.1 Tes Psikologi

Psikologi merupakan salah satu cabang ilmu pengetahuan yang sangat berpengaruh terhadap segala aspek dalam kehidupan manusia. Menurut William

James (1980), psikologi merupakan ilmu pengetahuan yang mempelajari tentang kondisi mental. Menurut John B. Watson (1919), psikologi merupakan ilmu alam tentang tingkah laku manusia berupa perbuatan dan ucapan sebagai pokok masalah. Sedangkan menurut Richard Meyer (1981), psikologi merupakan analisa ilmiah tentang proses mental dan struktur daya ingat yang digunakan dalam memahami tingkah laku manusia (Atkinson et al., 1997: 19). Sehingga di dalam bukunya, Atkinson et al. (1997: 18) mendefinisikan psikologi sebagai studi ilmiah terkait proses perilaku dan mental manusia.

Psikologi memberikan peranan yang sangat penting untuk memberikan solusi dalam memecahkan berbagai permasalahan manusia yang sangat beraneka ragam. Berbagai permasalahan tersebut dapat berupa permasalahan umum dan permasalahan khusus. Permasalahan secara umum salah satunya terkait cara mencegah penyakit jiwa maupun cara membesarkan anak agar anak nantinya dapat menjadi manusia dewasa yang bahagia. Permasalahan secara khusus salah satu contohnya adalah tindakan terbaik untuk menangani permasalahan merokok dan kegemukan. Psikologi mencakup segala aspek kehidupan manusia dan menjadi sangat penting tidak hanya untuk para psikolog namun juga semua orang yang tidak bermaksud untuk memperdalam ilmu tersebut untuk sedikit memahami beberapa fakta dasar maupun metode penelitian psikologi (Atkinson et al., 1997: 5-6).

Terdapat beberapa metode penelitian yang digunakan di dalam psikologi. Metode-metode tersebut adalah metode eksperimental, metode pengamatan, metode survei, metode tes, dan riwayat kasus. Metode tes merupakan instrumen

penelitian yang penting dalam psikologi masa kini atau kontemporer. Metode ini digunakan untuk mengukur kemampuan, minat, sikap, dan hasil kerja tanpa peralatan laboratorium yang rumit (Atkinson et al.,1997: 30).

Tes psikologi merupakan alat yang digunakan untuk mengukur perbedaan-perbedaan antara individu atau perbedaan reaksi individu yang sama terhadap situasi yang berbeda. Salah satu masalah yang membawa terhadap perkembangan tes-tes psikologi adalah identifikasi terhadap orang-orang dengan kondisi keterbelakangan mental. Penggunaan secara klinis dari tes-tes psikologi mencakup pemeriksaan terhadap orang-orang dengan gangguan emosi yang parah serta berbagai permasalahan perilaku. Perkembangan awal penggunaan tes-tes psikologi didapatkan dari kebutuhan dalam dunia pendidikan. Dalam hal ini, sekolah menjadi pengguna tes terbesar, salah satunya digunakan untuk mengklasifikasikan anak-anak berdasarkan kemampuan mereka menyerap instruksi di dalam kelas. Selain dalam dunia pendidikan, tes-tes psikologi juga banyak digunakan dalam bidang industri seperti digunakan dalam penerimaan karyawan maupun pemutusan hubungan kerja (Anastasi dan Urbina, 2007: 2-3).

Penggunaan tes-tes dalam konseling individu menyangkut rencana pendidikan dan pekerjaan hingga seluruh aspek kehidupan. Konseling tersebut menyangkut ketentraman emosi dan hubungan intrapersonal individu. Selain itu, penggunaan tes psikologi digunakan dalam pemahaman dan pengembangan diri. Sehingga skor yang didapatkan dalam tes merupakan informasi yang diberikan kepada individu, yang digunakan sebagai alat bantu dalam proses pengambilan keputusan (Anastasi dan Urbina, 2007: 3-4).

Tes psikologi merupakan alat ukur yang objektif dan dibakukan atas sampel perilaku. Nilai diagnostik atau prediktif dari suatu tes tergantung pada sejauh mana tes tersebut dapat menjadi indikator dari bidang perilaku yang relatif luas. Tes psikologi, seperti yang didefinisikan sebelumnya merupakan alat ukur yang dibakukan. Standarisasi, dalam hal ini merupakan keseragaman dalam hal penyelenggaraan dan penskoran dari tes yang dilakukan. Sehingga dalam menjamin keseragaman dalam penyelenggaraan tes, penyusun tes haruslah menyediakan petunjuk yang rinci dan jelas seperti batas waktu, instruksi lisan, demonstrasi awal, cara menjawab pertanyaan, dan beberapa hal lain dalam situasi *testing* (Anastasi dan Urbina, 2007: 4-6).

Tes psikologi sebagai salah satu metode psikodiagnostik atau teknik untuk melakukan pemeriksaan psikologis bertujuan untuk mengadakan klasifikasi, deskripsi, interpretasi, dan prediksi. Beberapa tujuan klasifikasi dari tes psikologi membantu mengatasi berbagai permasalahan yang berhubungan dengan :

- a. Pendidikan, yaitu permasalahan yang berhubungan dengan tingkat inteligensi, minat dan bakat, kesulitan belajar, dan lain sebagainya.
- b. Perkembangan anak, yaitu permasalahan yang berhubungan dengan hambatan perkembangan psikis maupun sosial.
- c. Klinis yaitu permasalahan yang berhubungan dengan beberapa gangguan psikis dalam tingkat berat atau ringan.
- d. Industri, yaitu permasalahan yang berhubungan dengan seleksi karyawan seperti pemilihan individu untuk menduduki sebuah jabatan tertentu yang sesuai, evaluasi seperti pemeriksaan psikologis untuk mengidentifikasi

kesesuaian jabatan karyawan dalam suatu perusahaan, dan promosi yaitu pemeriksaan psikologis untuk menilai kemampuan sebelum menaikkan posisi jabatan.

Tes psikologis sebenarnya tidak hanya digunakan untuk klasifikasi berbagai gangguan psikis saja atau *diagnose*, tetapi lebih diarahkan kepada pendeskripsian atau pemahaman secara intensif dari subjek. Tingkah laku atau kepribadian individu merupakan sebuah produk dari aspek sosiobiopsikologis sehingga tes psikologi bertujuan untuk memperoleh deskripsi dari individu dan aspek-aspek tersebut. Tes psikologi juga bertujuan sebagai prediksi yaitu digunakan untuk meramalkan atau mengidentifikasi perkembangan individu selanjutnya (Nur'aeni, 2012: 18-19).

Tes psikologi sangat banyak ragamnya sehingga dibutuhkan klasifikasi agar dapat memiliki orientasi yang baik dan jelas. Klasifikasi tersebut adalah sebagai berikut (Nur'aeni, 2012: 20-22):

- a. Berdasarkan banyaknya tes terdiri atas :
 - 1) Tes individu, yaitu saat berlangsungnya tes, tester hanya menghadapi satu *testee* saja.
 - 2) Tes kelompok, yaitu saat berlangsungnya tes, tester menghadapi sekelompok *testee*.
- b. Berdasarkan cara menyelesaikannya terdiri atas:
 - 1) Tes Verbal, yaitu dalam mengerjakan tes, *testee* memberikan jawaban dalam bentuk kata-kata seperti keterangan, lawan kata suatu gambar, dan mengataka kekurangan dari suatu gambar.

- 2) Tes Non-Verbal, yaitu dalam mengerjakan tes, *testee* tidak menggunakan respon dalam bentuk bahasa namun lebih kepada melakukan suatu kegiatan tertentu seperti menyusun balok.
- c. Berdasarkan atas cara penilaian tes terdiri atas :
- 1) Tes alternative, yaitu penilaian atas jawaban benar dan salah.
 - 2) Tes gradual, yaitu penilaian dengan beberapa tingkatan seperti diberikan nilai dalam rentang 1-5.
- d. Berdasarkan fungsi psikis yang dijadikan sasaran *testing* terdiri atas tes perhatian, tes fantasi, tes ingatan, dan tes kemauan.
- e. Berdasarkan tipe tes yang berhubungan dengan isi dan waktu yang disediakan terdiri atas:
- 1) *Speed test*, maksudnya adalah bahwa dalam tes lebih mengutamakan kecepatan dan ketepatan kerja.
 - 2) *Power test*, maksudnya adalah bahwa dalam tes tidak mengutamakan kecepatan sehingga tidak ada pembatasan waktu.
- f. Berdasarkan atas materi tes yang berhubungan dengan latar belakang teorinya terdiri atas:
- 1) Tes proyektif, yaitu tes yang disusun berdasarkan atas mekanisme proyektif seperti materi tes berupa objek yang belum jelas strukturnya.
 - 2) Tes non-proyektif, yaitu tes yang tidak mempertimbangkan mekanisme proyektif.

- g. Berdasarkan bentuknya terdiri atas tes benar salah, tes pilihan ganda, tes isian, tes mencari pasangan, tes penyempurnaan, tes mengatur objek, tes deret angka, dan tes perancangan balok.
- h. Berdasarkan penciptanya terdiri atas Tes Roscharch, Tes Wechsler, Binet-Simon, Tes Kraepelin, dan Tes Raven.
- i. Berdasarkan aspek yang diukur terdiri atas tes kecerdasan, tes bakat, tes kepribadian, dan tes minat.

Menurut HIMPSI (2002) dalam Nur'aeni (2012: 23), klasifikasi tes yang diuraikan di atas, secara garis besar dapat dikasifikasikan ke dalam tes intelegensi, tes bakat, tes kepribadian, dan tes minat.

a. Tes Intelegensi

Tes ini mengungkapkan intelegensi atau kecerdasan individu dalam mengetahui sejauh mana kemampuan suatu individu, sehingga dapat diperkirakan kebutuhan individu untuk pendidikan atau pelatihan tertentu yang dapat diberikan kepadanya. Nilai intelegensi dikaitkan dengan umur yang menghasilkan IQ sehingga dapat diketahui kedudukan individu tersebut dalam kelompok sebayanya.

- b. Tes Bakat, digunakan untuk mengungkap kemampuan khusus seseorang dalam bidang tertentu.
- c. Tes Kepribadian, digunakan untuk mencoba mengungkap ciri-ciri kepribadian seseorang seperti introversi, kemampuan sosial, dan hal lain yang menyangkut kepribadian seseorang.

- d. Tes Minat, digunakan untuk mengungkap reaksi seseorang dalam berbagai situasi yang secara keseluruhan akan mengungkapkan minatnya.

Dalam praktiknya, dalam proses mempelajari sisi psikologis manusia dengan segala sikap dan tingkah lakunya, masih menggunakan metode lama. Metode tersebut menggunakan lembaran-lembaran questioner atau serangkaian pertanyaan yang akan diberikan kepada objek yang akan dipelajari. Kemudian questioner-questioner tersebut diisi dan dikumpulkan kembali. Hasil dari pengerjaan questioner tersebut disajikan dalam bentuk nilai sehingga akan didapatkan sebuah kesimpulan dari nilai tersebut. Hal tersebut kurang efisien dan memakan waktu lama (Budiman, 2009: 2).

Tes psikologi yang dikomputerisasikan merupakan salah satu perkembangan teknologi dalam cabang ilmu psikologi yang masih tergolong baru. Perkembangan dari tes yang dikomputerisasikan berkembang sangat cepat dan memberikan keuntungan yang lebih besar dibandingkan dengan tes yang menggunakan pensil dan kertas atau tes secara manual. Keuntungan yang didapatkan dari kondisi tersebut juga didiskusikan dalam beberapa artikel dan *paper* (Olson-Buchanan & Drasgrow, 1999, for an overview). Beberapa keuntungan dari tes yang dikomputerisasikan diantaranya adalah penskoran dan laporan dari hasil pengerjaan yang dapat dilaksanakan secara langsung, fleksibilitas dalam penjadwalan tes, dan tes yang dikomputerisasikan memberikan pengukuran yang lebih baik terhadap pengetahuan dan kemampuan dari *testee*. Pengetesan yang menggunakan teknologi, memberikan alternatif baru dalam pengetesan yang dilakukan secara manual yaitu menggunakan pensil dan kertas

dan standar pertanyaan multiple-choice. Selain itu, tes yang dikomputerisasikan dapat meningkatkan kualitas dari segi keamanan (Drasgrow, 2002; Potenza, 2002; Way, Steffen, & Anderson, 2002). Sebagai contoh tes yang dikomputerisasikan akan memberikan keuntungan dalam pemeliharaan isi yang dapat digunakan sewaktu-waktu ketika dibutuhkan (Foster, 2010: 208).

Komputer memiliki pengaruh yang besar dalam setiap tahap pengetesan yaitu dari penyusunan tes hingga administrasi, penskoran, pelaporan, dan interpretasi (F.B. Baker, 1989; Butcher, 1987; Gutkin & Wise, 1991; Roid, 1986). Manfaat yang jelas terlihat dari penggunaan komputer telah menggambarkan peningkatan yang luar biasa dalam hal kecepatan proses analisis data dan skoring yang dijalankan (Anastasi dan Urbina, 2007: 83).

Tes yang dikomputerisasikan sekarang jelas memberikan beberapa keuntungan daripada tes dengan menggunakan kertas dan pensil. Pertama, bahan uji dapat disimpan atau tidak sekali pakai. Keuntungan ini berhubungan dengan biaya yaitu memberikan kemudahan dalam administrasi tes dan lingkup keamanan. Kedua, dikarenakan tes sepenuhnya otomatis, sehingga dibutuhkan standarisasi dalam penyediaan instruksi tes dan waktu. Ketiga, skor dapat dihitung dengan sangat cepat, sehingga dapat mengurangi kesalahan penghitungan yang dapat dilakukan oleh manusia dan dapat disimpan dalam waktu yang lama. Keempat, skor secara otomatis dan mudah disimpan dalam database untuk membenarkan norma dan dapat digunakan dalam penelitian. Kelima, pengambil tes mendapatkan keuntungan dalam menerima umpan balik secara langsung dari hasil tes yang dikerjakan (Barak dan English, 2002: 67-68).

Dari beberapa keuntungan dalam penggunaan teknologi untuk *testing*, terdapat kekurangan dan pembatasan dalam tes yang dikomputerisasikan. Pertama, untuk pengelola tes diharuskan memiliki dasar pengetahuan dalam operasi komputer dan juga dalam pemrograman. Kedua, untuk pengambil tes, terdapat beberapa indikasi terkait kemampuan penggunaan komputer yaitu dalam hal kecepatan mengetik akan berpengaruh pada keberhasilan tes (Russel, 1999). Sehingga perlu diperhatikan bahwa dalam point tes performansi tetap harus dilaksanakan dalam tes yang bersifat terbuka atau dilaksanakan secara langsung oleh ahli dan tidak berbentuk *multiple-choice* (Russel & Haney, 1997). Ketiga, terjadi perdebatan dalam hal akurasi dalam penafsiran skor yang disesuaikan dengan model tes yang dilakukan (Barak dan English, 2002: 68).

Beberapa permasalahan di atas menjelaskan bahwa dalam pelaksanaan tes harus memperhatikan etika baik tes yang menggunakan pensil dan kertas ataupun tes yang dikomputerisasikan. Perdebatan muncul salah satunya terkait etika yang berhubungan dengan keamanan (Chapman & Webster, 2003). Teknologi yang berpengaruh dalam kegiatan tes mencakup penggunaan alat yang secara umum dapat digunakan untuk berbuat curang, mengambil pertanyaan-pertanyaan tanpa ijin, dan melakukan komunikasi menggunakan internet (Foster, 2010: 209).

Perubahan dalam pelaksanaan tes, dengan lebih menggunakan teknologi dan internet dan sebagai program yang berkembang secara global, penting bagi para psikolog dan psychometricians untuk memahami dan mematuhi kode etik yang berlaku. Berhubungan dengan kondisi tersebut, kode 9.11 dalam *Ethical Principles of Psychologists and Code of Conduct* (APA, 2002) menjelaskan

bahwa “*psychologists make reasonable efforts to maintain the integrity and security of test materials*” (p.14) (Foster, 2010: 213).

Dalam pembuatan aplikasi *psychological testing* harus lengkap dengan *security* dan tetap dalam pengawasan seorang ahli. Tes psikologi berbasis online tidak memiliki keamanan yang terjamin. Kecurangan sangat mungkin terjadi dan mudah untuk dilakukan. Isi dapat dengan mudah ditangkap dan disebarluaskan. Masalah etika tersebut disebabkan karena tes secara online disajikan dengan keamanan yang relatif kecil bahkan tidak ada sama sekali (Foster, 2010: 213).

Beberapa permasalahan terkait dengan etika tersebut menjelaskan bahwa dalam pembuatan tes yang dikomputerisasikan harus memperhatikan etika dalam pembuatannya. Dalam arti, penggunaan dan pembuatan tes-tes psikologi harus dikendalikan. Seperti yang ditulis dalam buku Anastasi dan Urbina (2007: 11) terdapat suatu komentar yang melatarbelakangi pengendalian tes-tes psikologi salah satunya, “Teman sekamar saya sedang belajar psikologi. Ia memberi saya tes kepribadian dan hasilnya saya disebut neurotik. Setelah itu saya malu masuk kelas”. Komentar tersebut menggambarkan kemungkinan dalam penyalahgunaan tes-tes psikologi sehingga seseorang akan menyatakan tes tersebut akan melukai seseorang. Tes-tes psikologi harus digunakan dengan tepat dan efektif. Salah satunya oleh penguji yang memenuhi syarat agar tidak berdampak buruk. Penguji yang memenuhi syarat diperlukan dalam pelaksanaan tes yang mencakup tiga aspek utama yaitu seleksi tes, administrasi dan penskoran, dan interpretasi skor. Pengendalian harus dilakukan dalam penggunaan tes-tes psikologi untuk memastikan bahwa tes diberikan oleh penguji yang memenuhi syarat dan

mencegah kedekatan orang dengan isi tes yang dapat membuat tes tersebut tidak valid.

2.2.2 Tes Minat

Hakikat dan kekuatan dari minat dan sikap seseorang merupakan aspek penting kepribadian. Karakteristik ini secara material memengaruhi prestasi pendidikan dan pekerjaan, hubungan antarpribadi, kesenangan yang didapatkan seseorang dan aktivitas di waktu luang, dan fase-fase utama lainnya dari kehidupan sehari-hari. Studi tentang minat mendapatkan dorongan terkuat dari penaksiran pendidikan dan karir (Anastasi dan Urbina, 2007: 426).

Tes minat mengungkapkan reaksi seseorang terhadap berbagai situasi yang secara keseluruhan akan mencerminkan minatnya. Minat yang terungkap melalui tes minat ini seringkali menunjukkan minat yang lebih mewakili daripada minat yang sekedar dinyatakan yang biasanya bukanlah minat yang sesungguhnya. Pada umumnya, tes minat digunakan dalam 4 bidang terapan yaitu konseling karir bagi siswa sekolah lanjutan, konseling pekerjaan bagi karyawan, penjurusan siswa sekolah lanjutan atau mahasiswa, dan perencanaan bacaan dalam pendidikan dan latihan. Penjelasan lebih lanjut dijabarkan sebagai berikut (Nur' aeni, 2012: 34).

a. Konseling Karir

Hasil tes minat digunakan dalam konseling karir untuk siswa sekolah, khususnya sekolah umum (SMU) pada tahun-tahun pertama bersekolah. Namun demikian, konseling karir juga digunakan bagi siswa-siswi yang merencanakan bekerja setelah lulus. Selain itu konseling karir juga dapat digunakan bagi orang-orang yang putus sekolah untuk membantu memilih

pekerjaan yang cocok dalam waktu dekat. Kegunaan tes minat bagi siswa SMA adalah membantu untuk menunjukkan bidang pekerjaan yang secara umum dan luas agar segera mempersempit bidang pekerjaan yang sesuai dan memfokuskan diri pada bidang pekerjaan yang jelas.

b. Konseling Pekerjaan

Hasil tes minat juga digunakan dalam konseling pekerjaan untuk karyawan-karyawan yang bekerja dalam perusahaan atau bidang pekerjaan yang lain. Dalam hal ini fungsi tes minat adalah untuk melihat konsistensi antara tugas pekerjaan yang telah dijalani dengan pekerjaan yang disukai. Tes minat juga berfungsi untuk peningkatan efisiensi pekerjaan dan kepuasan kerja karyawan.

c. Penjurusan Siswa

Penjurusan siswa di sekolah lanjutan merupakan penempatan siswa pada jurusan-jurusan atau program-program studi yang tersedia. Namun apabila program studi hanya terdiri atas dua atau tiga pilihan maka tidak disarankan menggunakan tes minat karena mengukur minat seseorang secara luas.

d. Perencanaan Bacaan Pendidikan

Buku-buku bacaan yang ada di sekolah maupun perguruan tinggi kadang-kadang tidak disukai oleh para siswa dan mahasiswa karena tidak sesuai dengan bidang minatnya. Dalam sistem pendidikan klasikal tes minat dapat digunakan untuk menentukan bahan bacaan yang sesuai untuk meningkatkan prestasi belajar. Tes minat ini berfungsi untuk memilih bahan

bacaan yang sesuai yaitu bahan bacaan yang disukai oleh mayoritas siswa maupun mahasiswa.

Sebagian besar dari inventori minat dirancang untuk menaksir minat individu dalam berbagai bidang pekerjaan. Sejumlah inventori juga memberikan analisis minat dalam kurikulum pendidikan atau bidang studi, yang pada gilirannya terkait dengan keputusan karir (Anastasi dan Urbina, 2007:427).

2.2.3 Teori Holland

Personality Type Theory atau Teori Tipe Kepribadian merupakan teori yang dikemukakan oleh John Holland yang menyatakan bahwa sebuah usaha harus dilakukan untuk mencocokkan pilihan karir individu dengan kepribadiannya (Holland, 1973, 1987 dalam Santrock, 2001). Menurut Holland, apabila individu menemukan karir yang sesuai dengan kepribadiannya, maka individu tersebut akan menikmati karirnya tersebut dan dapat bertahan cukup lama. Holland mengemukakan ada 6 kepribadian dasar yang harus dipertimbangkan ketika menyesuaikan keadaan psikologis individu dan karir. Keenam kepribadian dasar tersebut adalah sebagai berikut (Lubis, 2008: 4-6):

1. *Realistic*

- a. Kemampuan mekanikal, psikomotor, dan atletik yang baik.
- b. Jujur.
- c. Setia
- d. Suka kegiatan-kegiatan diluar
- e. Lebih suka bekerja dengan mesin, alat-alat, tumbuhan, dan hewan.
- f. Lebih menyukai kegiatan-kegiatan yang bersifat fisik.

- g. Lebih menyukai kegiatan kongkrit.
- h. Tidak terlalu suka bersosialisai.
- i. Tidak suka hal-hal yang kompleks (lebih menyukai kesederhanaan)

Individu dengan kepribadian tersebut lebih baik memilih karir-karir yang bersifat praktis, seperti buruh, bertani, supir truk, dan konstruksi.

2. *Investigative*

- a. Kemampuan memecahkan masalah dan analitis yang baik.
- b. Cenderung berfikir matematis.
- c. Suka mengobservasi, mempelajari, dan mengevaluasi.
- d. Lebih suka bekerja sendiri.
- e. Pemberi ide.
- f. Hati-hati, kritis, dan selalu ingin tahu
- g. Suka kedisiplinan.
- h. Berorientasi tugas.
- i. Sistematis

Individu dengan kepribadian ini lebih berorientasi pada konsep dan teori. Mereka lebih berperan sebagai pemikir daripada sebagai pelaksana. Mereka sering menghindari adanya hubungan interpersonal dan lebih sesuai dengan karir-karir yang berkaitan dengan matematika dan pengetahuan.

3. *Artistic*

- a. Berpikir abstrak.
- b. Menyukai estetika (keindahan).
- c. Kreatif, suka hal-hal kompleks, emosional, intuitif, ideal.

- d. Lebih suka bekerja secara mandiri.
- e. Suka menyanyi, berakting, menulis, dan melukis.
- f. Imaginatif.
- g. Tidak suka hal-hal yang konvensional.
- h. Tidak dapat diduga.
- i. Tidak suka keteraturan.

Individu dengan kepribadian tersebut lebih suka berinteraksi dengan dunianya melalui ekspresi artistik, cenderung menghindari situasi yang konvensional dan interpersonal. Individu tersebut lebih baik berorientasi pada karir seperti seni dan menulis.

4. *Social*

- a. Komunikatif.
- b. Bersahabat, mudah bergaul.
- c. Suka memberi dan membantu.
- d. Baik, impulsive.
- e. Bertanggung jawab.
- f. Berjiwa kelompok.
- g. Mempunyai toleransi yang cukup baik.
- h. Dapat memahami.
- i. Kemampuan verbal dan personal yang baik.

Individu ini memiliki kemampuan verbal dan hubungan interpersonal yang baik. Mereka lebih cocok memasuki dunia karir yang berhubungan dengan manusia seperti mengajar, pekerja sosial, konseling, dan sebagainya.

5. *Enterprising*

- a. Percaya diri, bersikap asertif.
- b. Mudah beradaptasi.
- c. Ambisiusus.
- d. Kemampuan berbicara dan memimpin yang baik.
- e. Suka menggunakan pengaruh seseorang.
- f. Kemampuan interpersonal yang cukup baik.
- g. Penuh energi, ekstrovert, optimis, persuasif.
- h. Suka mengambil resiko, spontan.
- i. Suka mengontrol.

Individu ini menggunakan kemampuan verbalnya untuk memimpin orang lain, mendominasi individu, dan menjual produk atau hal yang lain. Individu ini lebih sesuai dalam karir seperti sales, bidang politik, dan manajemen.

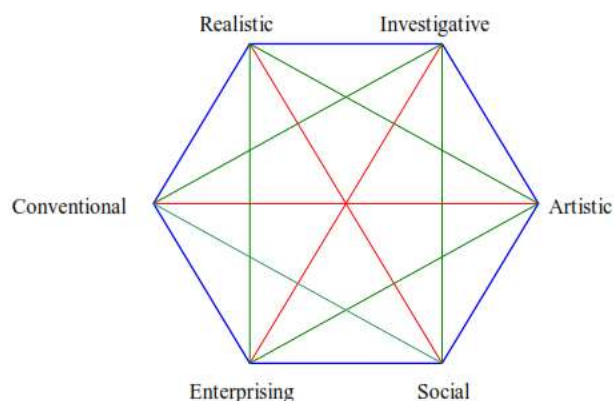
6. *Conventional*

- a. Tergantung pada orang lain.
- b. Tidak kreatif.
- c. Suka kedisiplinan dan ketepatan.
- d. Suka memperhatikan detail.
- e. Efisien.
- f. Melaksanakan tugas secara teratur.
- g. Kemampuan klerikel dan numerical yang baik.
- h. Terorganisir.

i. Bersifat tradisional.

Individu ini lebih menyukai kegiatan yang bersifat terstruktur. Mereka lebih suka pekerjaan yang berupa bawahan seperti pegawai bank, sekretaris, dan petugas arsip.

Holland mengembangkan kuesioner *Vocational Preference Inventory* yang berisi 160 pekerjaan. Responden memilih pekerjaan yang disukai atau tidak disukainya dan jawaban digunakan untuk membentuk profil kepribadiannya. Cara untuk mengidentifikasi karir digambarkan dengan sebuah heksagon. Sebuah heksagon menggambarkan kesamaan dan perbedaan karakteristik diantara orang, diantara pekerjaan, dan diantara orang dan pekerjaan. Hanya sedikit orang yang memiliki satu tipe. Biasanya setiap orang memiliki dua tipe atau lebih. Tipe data heksagon yang bersebelahan langsung merupakan kombinasi tipe yang erat hubungannya. Tipe yang berseberangan merupakan tipe yang paling tidak berhubungan. Contohnya tipe *Realistic* dan *Investigative* serupa, namun *Realistic* dan *Social* bertolak belakang.



Gambar 2.6 Heksagon dari Holland (Lubis, 2008: 7)

Dikutip dari salah satu hasil test yang dilakukan oleh PAR Inc. Amerika, dari 160 jenis pekerjaan yang dikemukakan oleh Holland, sebagai contoh 5 kombinasi *SDS Code* diantaranya adalah sebagai berikut:

- a. Kombinasi kode ringkas **SEI** dengan saran pekerjaan *Psychologist* dan *Counseling*.
- b. Kombinasi kode **ISC** dengan saran pekerjaan sebagai berikut:
 - *Computer Network Support Specialist*
 - *Dialysis Technician*
 - *Linguist*
 - *Market Research Analyst*
 - *Microbiologist*
 - *Physician*
 - *Research Asssistant*
 - *Translator*
- c. Kombinasi kode **CSI** dengan saran pekerjaan
 - *Accounting and Computer Sciences*
 - *Computer Support Specialist*
 - *Health Information/Medical Records Technology/Technician*
- d. Kombinasi kode **SCI** dengan saran pekerjaan sebagai *Community Health Worker* dan *Optometri Assistant*
- e. Kombinasi kode **SEC** dengan saran pekerjaan sebagi *Social Worker*.

2.2.4 SDS (*Self-Directed Search*)

Di antara banyak inventori minat yang tersedia dewasa ini, empat inventori telah dipilih untuk dibahas satu demi satu karena masing-masing menggambarkan segi yang patut diperhatikan dalam orientasi teoritis, metodologi, atau jenis populasi yang untuknya inventory tersebut dirancang. Salah satunya adalah SDS (*Self-Directed Search*) (Anastasi dan Urbina, 2007: 437).

SDS (*Self-Directed Search*) merupakan pendekatan terhadap penaksiran minat-minat pekerjaan. Instrumen ini dikembangkan oleh Holland, yang model segi-enamnya tentang tema-tema pekerjaan umum telah menarik perhatian luas dan dimasukkan dalam berbagai inventori yang ada sekarang (Holland, 1985/1992; Holland, Fritzsche & Powell, 1994; Holland & Gottfredson, 1976; Holland, Powell & Fritzsche, 1994). SDS dirancang sebagai instrumen konseling pekerjaan yang bisa dilakukan sendiri, diskor sendiri, dan diinterpretasikan sendiri. Meskipun disusun di seputar minat, prosedur ini juga memerlukan peringkat-diri kemampuan dan kompetensi yang dilaporkan. Individu mengisi Buklet berhubungan dengan tema model Holland (Realistis, Investigative, Artistik, Sosial, Bersifat Wirausaha, dan Konvensional) (Anastasi dan Urbina, 2007: 441).

SDS digunakan secara luas dalam berbagai lingkungan dan telah menghasilkan banyak penelitian. SDS telah mengalami revisi beberapa kali untuk menyederhanakan prosedur dan mengurangi bias jenis kelamin dalam keputusan-keputusan karir. Daya tarik dari alat ukur ini adalah dari keringkasannya dan kesederhanaannya. Selain bentuk reguler dari inventori ini (Bentuk R), tiga versi

lain telah tersedia yaitu Formulir E (*Easy*), yang dikembangkan untuk individu dengan ketrampilan membaca yang terbatas; Formulir CP (*Career Planning*), yang dirancang untuk orang dewasa yang berada di tengah-tengah transisi karirnya; Versi Penjelajah Karir, yang ditujukan pada siswa sekolah menengah dan sekolah menengah pertama (Anastasi dan Urbina, 2007: 441).

J.L Holland (1966, 1985/1992) mengaitkan dirinya dengan jelas bersama mereka yang memandang pilihan pekerjaan sebagai pilihan cara hidup yaitu pilihan yang mencerminkan konsep-diri individu dan kepribadian utama. Masing-masing tema dari tema pekerjaan Holland sesuai dengan “tipe” kepribadian. Juga bersesuaian dengan model lingkungan. Dalam kaitannya dengan model lingkungan, lingkungan pekerjaan yang berbeda dapat dikarakteristikan. Tidak hanya berupa segi-segi jasmani dan tuntutan kerja, namun juga dengan jenis orang dengan siapa individu tersebut bekerja (rekan kerja, penyelia, pelanggan, klien, siswa). Menurut Holland, individu mencari lingkungan yang sesuai dengan tipe kepribadian mereka dan kesesuaian tersebut meningkatkan kepuasan kerja, stabilitas pekerjaan, dan prestasi (Anastasi dan Urbina, 2007: 442).

Menurut Holland, seseorang dapat digolongkan ke dalam salah satu tipe dari 6 jenis kepribadian. Lingkungan hidup dari orang-orang tersebut juga dapat digolongkan ke dalam 6 jenis juga. Model orientasi yang telah dijelaskan di atas dijabarkan lagi dalam penjelasan sebagai berikut (Febriana, 2013: 8-10):

- a. *Realistic*, preferensinya pada aktivitas-aktivitas yang memerlukan manipulasi eksplisit, teratur, atau sistematis terhadap obyek-obyek, alat-alat, mesin-mesin, dan binatang-binatang. Tidak menyukai aktivitas

pemberian bantuan dan pendidikan. Preferensi-preferensi membawa kepada pengembangan kompetensi-kompetensi dalam bekerja dengan benda-benda, binatang-binatang, alat-alat dan perlengkapan teknik, dan mengabaikan kompetensi-kompetensi sosial dan pendidikan. Menganggap diri baik dalam kegiatan mekanikal dan atletik dan tidak cakap dalam kemampuan-kemampuan sosial. Menilai tinggi benda-benda nyata seperti uang dan kekuasaan. Ciri-ciri khususnya adalah praktikalitas, stabilitas, konformitas. Mungkin lebih menyukai ketrampilan-ketrampilan dan okupasi-okupasi teknik.

- b. *Investigative*, memiliki preferensi untuk aktivitas-aktivitas yang memerlukan penyelidikan observasional, simbolik, sistematis, dan kreatif terhadap fenomena fisik, biologis, dan kultural agar dapat memahami dan mengontrol fenomena tersebut, dan tidak menyukai aktivitas persuasif, sosial, dan repetitif. Contoh-contoh dari jenis ini adalah ahli kimia dan fisika.
- c. *Artistic*, lebih menyukai aktivitas-aktivitas yang bersifat ambigu, bebas, dan tidak tersistematisasi untuk menciptakan produk-produk artistik seperti lukisan, drama, dan karangan. Tidak menyukai aktivitas yang sistematis, teratur, dan rutin. Kompetensi dalam upaya-upaya artistik dikembangkan dan kemampuan yang rutin, sistematis, dan klerikal diabaikan. Memandang diri sebagai ekspresif, murni, independen, dan memiliki kemampuan artistik. Ciri khususnya adalah emosional,

imaginatif, impulsif, dan murni. Okupasi-okupasi dari artistik seperti seni pahat, akting, lukisan, dan karangan.

- d. *Social*, menyukai aktivitas-aktivitas yang berhubungan dengan orang lain seperti mengajar, membantu, dan menyediakan bantuan. Tidak menyukai aktivitas rutin dan sistematis. Kompetensi sosial dikembangkan dan yang bersifat manual dan teknik diabaikan. Menganggap diri berkompeten dalam mengajar dan membantu orang lain dan menyukai aktivitas sosial. Ciri khususnya kerjasama, bersahabat, persuasif, dan bijaksana. Okupasi sosial mencakup pekerjaan seperti mengajar, konseling, dan pekerjaan sosial.
- e. *Enterprising*, lebih menyukai aktivitas-aktivitas yang melibatkan manipulasi terhadap orang lain. Tidak menyukai aktivitas yang sistematis, abstrak, dan ilmiah. Kompetensi kepemimpinan, persuasif, dan bersifat supervisi dikembangkan dan yang ilmiah diabaikan. Memandang diri sebagai agresif, populer, percaya diri, dan memiliki kemampuan memimpin. Ciri-ciri khususnya adalah ambisius, dominasi, optimisme, dan sosiabilitas.
- f. *Conventional*, lebih menyukai aktivitas-aktivitas yang memerlukan manipulasi data secara eksplisit, teratur, dan sistematis guna memberikan kontribusi pada tujuan-tujuan organisasi. Menyukai aktivitas yang tidak bebas, sistematis, dan pasti. Kompetensi dikembangkan dalam bidang klerikal, komputasional, dan sistem usaha dan aktivitas artistik diabaikan. Memandang diri teratur, mudah menyesuaikan diri, memiliki kemampuan

klerikal dan numerical. Ciri khasnya efisiensi, keteraturan, praktikalitas, dan kontrol diri. Okupasi yang sesuai seperti bankir, penaksir harga, ahli pajak, dan pemegang buku.

Menurut Winkel dan Hastuti (2005) dalam Febriana (2013: 11), kelebihan tes karir Holland adalah sebagai berikut :

- a. Alat tes ini arahnya sudah jelas yaitu terfokus pada mengukur minat seseorang.
- b. Dengan alat tes ini dapat diketahui karakteristik dari individu.
- c. Menunjuk pada taraf inteligensi yang memungkinkan tingkat pendidikan sekolah tertentu.
- d. Pandangan Holland sangat relevan bagi bimbingan karir dan konseling karir di institusi pendidikan untuk jenjang pendidikan menengah dan awal pendidikan tinggi.

2.3 Kerangka Berfikir

Uma Sekaran (1992) dalam Sugiyono (2011) mengemukakan bahwa, kerangka berfikir merupakan model konseptual tentang hubungan antara teori dengan berbagai faktor yang telah diidentifikasi sebagai masalah penting. Kerangka berfikir dalam pembuatan aplikasi tes psikologi dalam hal ini tes minat dimulai dengan analisis. Analisis meliputi studi pendahuluan, konsep dan prinsip analisis, serta model analisis. Studi pendahuluan meliputi metode pengumpulan data dan identifikasi masalah serta kebutuhan pelanggan. Konsep dan prinsip analisis merupakan analisis penggunaan perangkat keras dan perangkat lunak dalam pembuatan aplikasi. Kemudian model analisis meliputi *Scenario-based*

modelling, Flow oriented modelling, Class-based modelling, dan Creating a behavioral model (Pressman, 2005: 218-248).

Setelah dilaksanakan analisis, pembuatan aplikasi dilanjutkan dengan tahap desain yang mencakup desain data atau *class*, desain arsitektur, dan desain *interface* (Pressman, 2005: 260). Tahap selanjutnya adalah tahap implementasi. Tahap ini dibagi ke dalam dua tahap lagi yaitu tahap implementasi sistem dan tahap pengujian sistem. Tahap implementasi sistem merupakan tahap penentuan lingkungan sistem, batasan sistem, tahapan implementasi, dan implementasi *interface*. Sedangkan tahap pengujian merupakan tahap dilaksanakannya pengujian analisis sistem untuk mengetahui kelemahan dan kemampuan sistem baru yang ada.

Pada tahap pengujian analisis sistem digunakan metode analisis PIECES yaitu uji terhadap *Performance, Information, Economics, Control, Efficiency, dan Service*. Metode PIECES menurut Wukil Ragil (2010) dalam Puspitasari (2015: 41-43) adalah metode yang digunakan untuk memperoleh pokok-pokok permasalahan dengan lebih spesifik. Metode ini menggunakan enam variabel evaluasi yaitu:

1. *Performance* (Kinerja)

Variabel ini digunakan untuk menilai apakah proses atau prosedur yang ada masih mungkin untuk ditingkatkan kinerjanya dan melihat sejauh mana dan seberapa handal suatu sistem dalam melakukan proses untuk mencapai tujuan. Dalam hal ini kinerja diukur dari:

- a. *Throughput*, yaitu jumlah pekerjaan /*output/deliverables* yang dapat dilakukan/ dihasilkan pada saat tertentu.
- b. *Response time*, yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan pekerjaan/*output/deliverables* tertentu.

2. *Information* (Informasi)

Digunakan untuk menilai apakah prosedur saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan semakin baik. Dalam hal ini informasi dapat diukur dari:

- a. Keluaran (*output*) yaitu sistem dalam memproduksi keluaran.
- b. Masukan (*input*) yaitu dalam memasukkan suatu data sehingga dapat diolah untuk menghasilkan informasi yang berguna.

3. *Economics* (Ekonomi)

Digunakan untuk menilai apakah prosedur yang ada pada saat ini masih dapat ditingkatkan manfaatnya dan diturunkan biaya penyelenggaraannya.

4. *Control* (Pengendalian)

Digunakan untuk menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian semakin baik, dan kemampuannya dalam mendeteksi kecurangan atau kesalahan menjadi semakin baik pula.

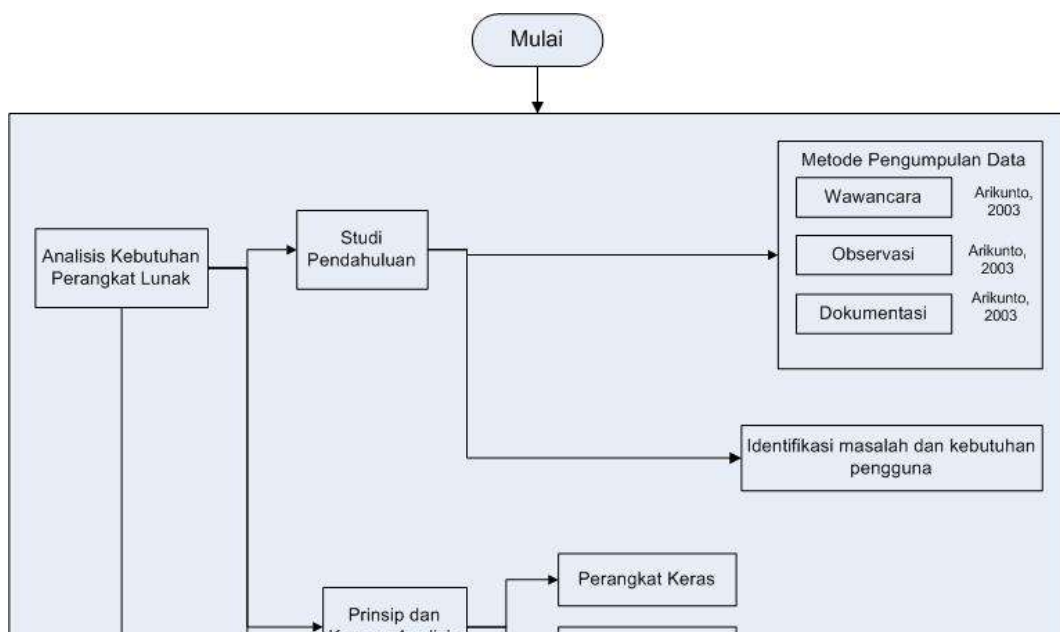
5. *Efficiency* (Efisiensi)

Digunakan untuk menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya sehingga tercapai peningkatan efisiensi operasi dan harus lebih unggul daripada sistem manual.

6. *Service* (Layanan)

Digunakan untuk menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Sehingga kualitas layanan perlu dibuat *user friendly* untuk pengguna sehingga pengguna mendapatkan layanan yang baik.

Kerangka berfikir yang digunakan untuk penelitian pengembangan aplikasi digambarkan pada gambar 2.7 sebagai berikut :

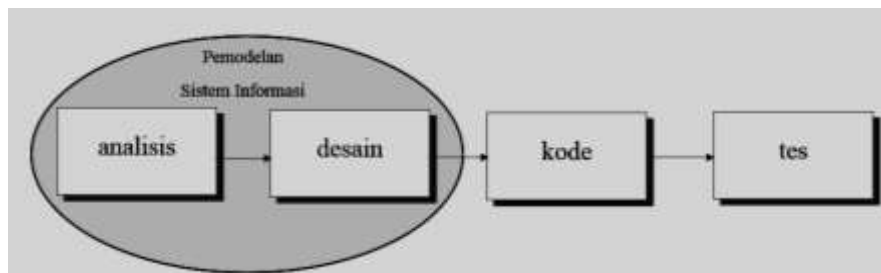


Gambar 2.7 Kerangka Berfikir Pengembangan Aplikasi

BAB III

METODE PENELITIAN

Metode yang digunakan dalam pembuatan aplikasi untuk skripsi ini adalah metode sekuensial linier atau yang sering disebut sebagai “siklus kehidupan klasik” atau “metode air terjun”. Pada penelitian ini dilakukan beberapa tahap sesuai dengan alur yang ada di dalam metode sekuensial linier untuk rekayasa perangkat lunak seperti pada gambar 3.1 sebagai berikut:



Gambar 3.1 Model Sekuensial Linier (Pressman, 1997: 37)

3.1 Analisis Perangkat Lunak

Analisis dibutuhkan sebelum dilakukan pembuatan ataupun pengembangan suatu produk. Analisis dalam pembuatan aplikasi, dalam hal ini aplikasi *HoTest* dibutuhkan untuk mengidentifikasi kebutuhan-kebutuhan dalam pembangunan aplikasi tersebut.

3.1.1 Studi Pendahuluan

Studi pendahuluan merupakan tahap awal dalam pengembangan aplikasi untuk mengetahui masalah, tujuan, dan kelayakan pengembangan aplikasi tes minat dari aplikasi untuk cabang ilmu psikologi.

3.1.1.1 Waktu dan Tempat Penelitian

Penelitian Skripsi Rancang Bangun Aplikasi Tes SDS (*Self-Directed Search*) dalam rangka studi pendahuluan, dilaksanakan pada tanggal 12 Januari 2015 - 01 April 2015 di Laboratorium Psikologi Universitas Negeri Semarang.

3.1.1.2 Metode Pengumpulan Data

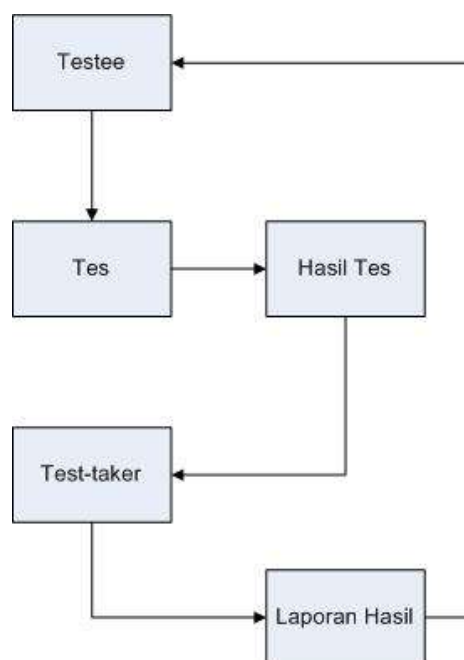
Untuk mendapatkan data yang diperlukan dalam penyusunan skripsi, digunakan metode sebagai berikut:

1. Wawancara

Dalam hal ini, dilakukan *interview* atau wawancara langsung dengan Kepala Laboratorium Psikologi Universitas Negeri Semarang untuk memperoleh data dan informasi terkait tes psikologi dalam hal ini tes minat dan alat ukur yang akan diterapkan dalam aplikasi serta proses pelaksanaan tes secara manual sesuai dengan kebutuhan informasi yang diperlukan.

2. Observasi

Dalam hal ini, peneliti melakukan pengamatan secara langsung terhadap pengolahan data hasil tes di Laboratorium Psikologi Universitas Negeri Semarang. Observasi ini dilakukan bersamaan dengan kegiatan wawancara dengan Kepala Laboratorium Universitas Negeri Semarang. Dari hasil wawancara dan observasi yang dilakukan alur dari pelaksanaan tes hingga olah data hasil tes seperti pada gambar 3.2 sebagai berikut :



Gambar 3.2 Alur Pelaksanaan Tes

3. Dokumentasi

Dalam penelitian ini, data diperoleh dari Laboratorium Psikologi Universitas Negeri Semarang berupa alat ukur SDS (*Self-Directed Search*), lembar jawab tes, dan data program studi menurut kamus Holland dengan mengajukan surat peminjaman ke Laboratorium yang bersangkutan. Data-data tersebut sangat membantu untuk melakukan analisis perancangan maupun sebagai bahan pembuatan aplikasi serta dalam penarikan kesimpulan.

3.1.1.3 Identifikasi Masalah dan Kebutuhan Pengguna

Permasalahan yang ada dalam bidang psikologi khususnya tes psikologi adalah bagaimana membuat suatu aplikasi tes psikologi dalam hal ini tes untuk mengukur minat yang dapat memudahkan proses pelaksanaan tes, penghitungan, maupun penyampaian hasil kepada responden atau *testee* serta tetap memiliki

tingkat kerahasiaan yang cukup baik karena berada dalam pengawasan seorang ahli. Hal tersebut dikarenakan, dalam praktiknya tes psikologi dalam hal ini tes minat yang dilaksanakan secara konvensional terdapat beberapa poin permasalahan yang menjadi perhatian. Dari hasil identifikasi kelemahan pelaksanaan tes minat secara manual, secara garis besar permasalahan yang ada adalah:

1. Proses pelaksanaan tes membutuhkan waktu kurang lebih 2 minggu dari mulai tes dilaksanakan sampai keluarnya hasil pengerjaan.
2. Ditinjau dari segi bahan dalam hal ini soal maupun lembar jawaban yang hanya sekali pakai sehingga akan memakan banyak waktu untuk mempersiapkan bahan-bahan tersebut untuk pelaksanaan tes dan juga memakan biaya yang lebih banyak.
3. Dengan cara manual atau konvensional, segala sesuatu yang berhubungan dengan pelaksanaan tes harus dilakukan secara manual dari persiapan hingga penyajian hasil pengerjaan.
4. Informasi dari skor yang akan sulit ditemukan apabila sewaktu-waktu akan digunakan untuk penelitian karena berhubungan dengan sulitnya penyimpanan untuk data-data hasil pengerjaan.
5. Tes yang dilaksanakan secara online masih belum memenuhi standar keamanan dalam tes karena isi tes dapat dengan mudah ditangkap dan disebarluaskan.

Penyebab terjadinya permasalahan tersebut dipengaruhi oleh beberapa faktor diantaranya adalah :

1. Pelaksanaan tes minat yang masih manual atau menggunakan kertas.
2. Secara umum, aplikasi-aplikasi yang sudah ada belum memenuhi standar dari segi keamanan dan soal maupun penghitungan untuk hasil belum disesuaikan dengan model alat ukur apa yang digunakan.
3. Berdasarkan studi pendahuluan yang dilakukan di Laboratorium Psikologi Universitas Negeri Semarang, aplikasi tes minat yang ada masih sedikit dibandingkan dengan banyaknya alat ukur yang ada di dalam tes psikologi.

Untuk lebih rinci, identifikasi masalah dan kebutuhan pengguna yang akan diterapkan dalam aplikasi tes minat nantinya setelah dilaksanakan studi pendahuluan, disajikan dalam tabel berikut :

Tabel 3.1 Identifikasi Masalah dan Kebutuhan Pengguna

No.	Identifikasi Masalah	Kebutuhan Pengguna
1.	<ul style="list-style-type: none"> • Pelaksanaan tes minat sebagian besar masih menggunakan cara manual yaitu dengan menggunakan pensil dan kertas dalam pengerjaannya. Sehingga dibutuhkan waktu yang cukup lama dari mulai dilaksanakannya pengerjaan tes, proses penghitungan, hingga keluarnya hasil tes. 	<ul style="list-style-type: none"> • Lembaga penyedia tes psikologi dalam hal ini tes minat membutuhkan sebuah aplikasi yang dapat mempermudah pelaksanaan tes baik dalam proses pelaksanaan tes, penghitungan, hingga keluarnya hasil tes.

2.	<ul style="list-style-type: none"> • Penyampaian informasi terkait pengerjaan tes masih harus dilaksanakan secara manual. • Informasi hasil penskoran akan sulit ditemukan apabila akan digunakan dikemudian hari karena hasil pengerjaan berupa dokumen-dokumen <i>hardfile</i> yang akan sulit untuk disimpan dalam waktu yang lama. • Penghitungan masih menggunakan cara manual sehingga sangat mungkin terjadi <i>human error</i> sehingga penghitungan hasil pengerjaan kurang akurat dan memakan banyak waktu. 	<ul style="list-style-type: none"> • Lembaga penyedia tes psikologi dalam hal ini tes minat membutuhkan aplikasi yang dapat memberikan informasi dalam pelaksanaan tes dengan lebih cepat, dokumen dapat tersimpan dengan baik dan dapat mudah dicari ketika akan digunakan sewaktu-waktu, serta penghitungan untuk hasil tes dapat dilakukan dengan cepat dan akurat.
3.	<ul style="list-style-type: none"> • Ditinjau dari segi biaya, pelaksanaan tes secara manual akan memakan biaya yang lebih banyak, dikarenakan <i>test taker</i> harus menyediakan bahan-bahan yang digunakan dalam 	<ul style="list-style-type: none"> • Lembaga penyedia tes psikologi dalam hal ini tes minat membutuhkan aplikasi yang dapat mempermudah penyediaan bahan-bahan untuk pelaksanaan dan pelaporan hasil

	<p>pelaksanaan tes seperti kertas untuk lembar soal, lembar jawab, maupun laporan hasil pengerjaan yang hanya sekali pakai.</p>	<p>tes. Bahan tersebut dapat tersimpan dengan baik dan dapat digunakan berulang kali, serta hasil pengerjaan dapat dicetak sewaktu-waktu apabila dibutuhkan.</p>
4.	<ul style="list-style-type: none"> • Dalam pelaksanaan tes psikologi, ditinjau dari segi keamanan bahwa pelaksanaan tes psikologi harus tetap berada dalam pengawasan. Apabila tidak dilakukan pengawasan, kecurangan sangat mungkin terjadi. • Hasil dari pengerjaan memuat suatu kerahasiaan bahwa hasil kasar hanya boleh diketahui oleh <i>test-taker</i>. 	<p>Lembaga penyedia tes psikologi dalam hal ini tes minat membutuhkan aplikasi yang dapat membantu proses pelaksanaan tes dengan lebih mudah, namun tetap dalam pengawasan seorang ahli. Hasil dari pengerjaan juga harus dapat dirahasiakan yaitu dengan menyediakan hak akses tersendiri bagi <i>test-taker</i> sehingga pengendalian terhadap pemberian hasil tes dapat dilakukan.</p>
5.	<ul style="list-style-type: none"> • Penggunaan kertas dan pelaksanaan tes yang masih serba manual akan menyebabkan pemborosan baik 	<ul style="list-style-type: none"> • Lembaga penyedia tes psikologi dalam hal ini tes minat, membutuhkan aplikasi yang dapat membantu pelaksanaan

	dalam segi penyediaan bahan maupun waktu.	tes yang segala sesuatunya dapat dilaksanakan dengan lebih otomatis sehingga dapat menghemat waktu dan biaya.
--	---	---

Sumber : Hasil Olahan Penulis, 2015

3.1.2 Prinsip dan Konsep Analisis

Dalam pembuatan suatu sistem, peran keseluruhan perangkat lunak diidentifikasi selama rekayasa sistem. Sehingga untuk mengidentifikasi dengan benar harus mengikuti serangkaian konsep yang akan mendasari prinsip-prinsipnya.

3.1.2.1 Perangkat Keras yang Digunakan

Dalam pembuatan aplikasi tes minat SDS (*Self-Directed Search*), dibutuhkan perangkat keras agar aplikasi dapat berjalan dengan baik. Spesifikasi dari perangkat keras yang digunakan dalam pembangunan aplikasi tes minat SDS yaitu aplikasi *HoTest (Holland Advanced-Study Test)* adalah sebagai berikut:

Tabel 3.2 Spesifikasi Perangkat Keras (Hardware)

No.	Spesifikasi Hardware	Kebutuhan	Ketersediaan
1.	Processor	Intel Core i3 1,90 GHz	✓
2.	Memori (RAM)	2,00 GB DDR3	✓
3.	Harddisk	500 GB	✓
4.	Monitor	14''	✓
5.	Mouse	Mouse Optic Port USB	✓

Sumber : Hasil Olahan Penulis, 2015

3.1.2.2 Perangkat Lunak yang Digunakan

Untuk merancang dan membangun aplikasi tes minat dengan menerapkan alat ukur SDS (*Self-Directed Search*) dalam aplikasi *HoTest (Holland Advanced-Study Test)*, spesifikasi perangkat lunak yang digunakan dalam pembangunan aplikasi tersebut adalah sebagai berikut :

Tabel 3.3 Spesifikasi Perangkat Lunak

No.	Spesifikasi Software	Kebutuhan	Ketersediaan
1.	Sistem Operasi	Windows 8.1 Pro	✓
2.	Databse	SQLite Database	✓
3.	Desain Aplikasi	Netbeans IDE 7.4	✓
4.	Desain Grafis	Corel Draw X4	✓
5.	Pemodelan UML	ArgoUML 0.34	✓
6.	Script Engine	Java	✓

Sumber : Hasil Olahan Penulis, 2015

3.1.3 Model Analisis

Model analisis harus dapat mencapai tiga tujuan utama yaitu untuk menggambarkan kebutuhan pelanggan, untuk membangun dasar bagi pembuatan desain perangkat lunak, dan untuk membatasi serangkaian persyaratan yang dapat divalidasi ketika perangkat lunak dibangun (Pressman, 1997:353). Sehingga model analisis yang ditarik selama analisis terstruktur berlangsung adalah sebagai berikut:

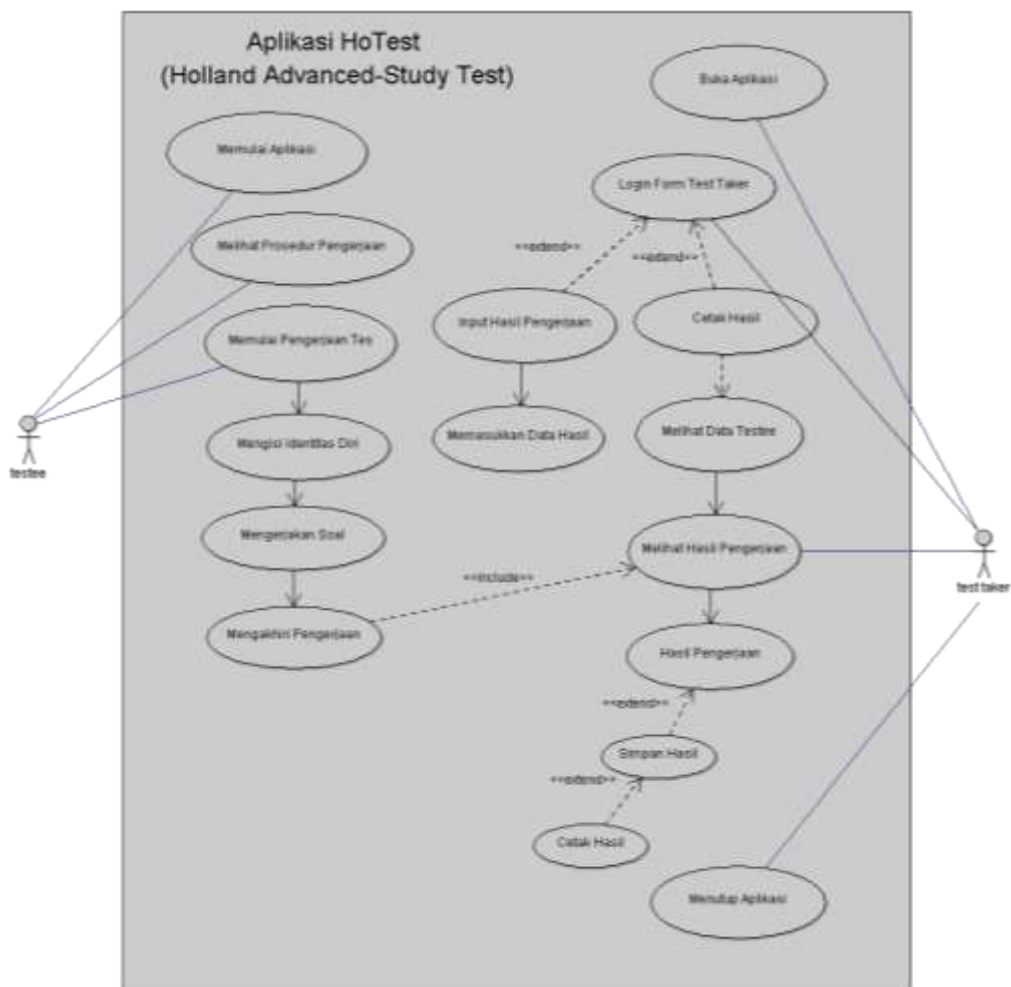
3.1.3.1 Scenario-Based Modelling

Model analisis menggunakan UML (*Unified Modelling Language*) dimulai dengan membuat skenario dalam bentuk *Use case diagram*, *Activity diagram*, dan

Swimlane diagram. Dalam pembuatan aplikasi *HoTest* skenario dibuat dalam bentuk *Use case Diagram* dan *Activity Diagram*.

a. *Use-Case Diagram*

Use case diagram untuk aplikasi tes minat dengan menerapkan alat ukur SDS (*Self-Directed Search*) digambarkan pada gambar 3.3 sebagai berikut :

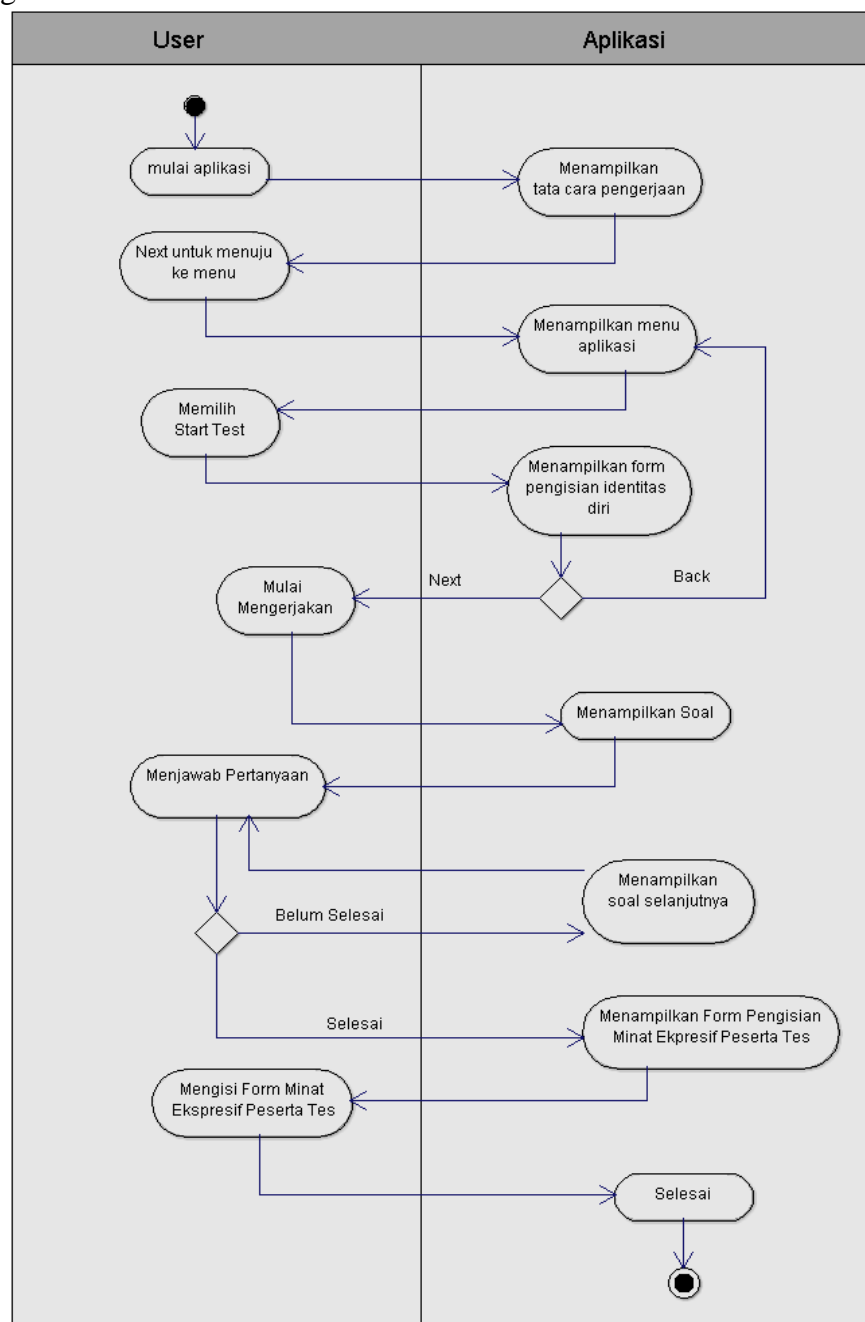


Gambar 3.3 *Use Case Diagram* Tes Psikologi *HoTest*

b. *Activity Diagram*

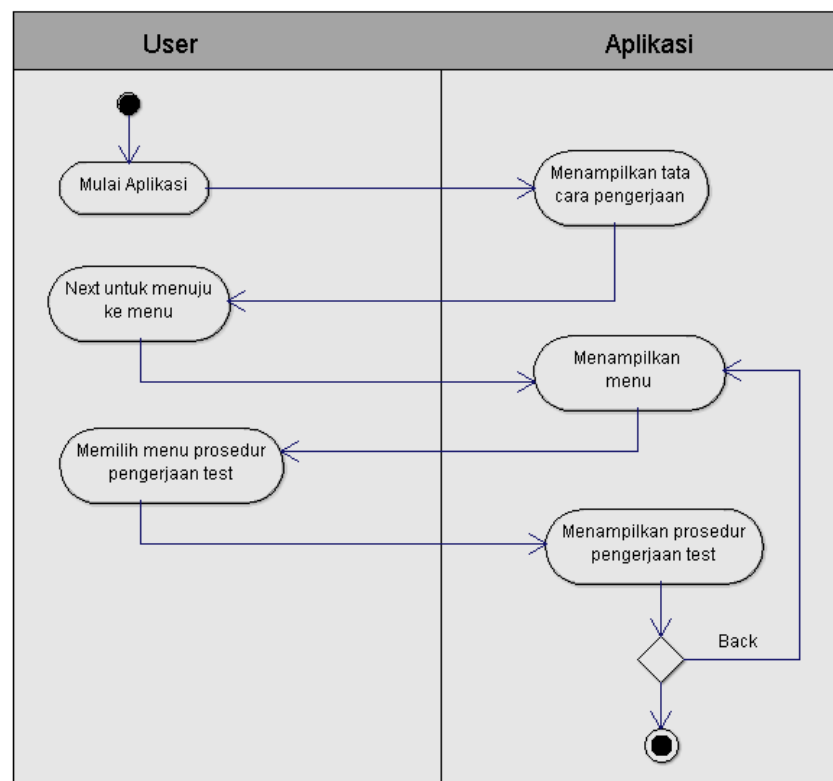
Activity Diagram untuk menggambarkan alur interaksi dengan skenario yang lebih spesifik. *Activity Diagram* mirip dengan *flowchart* yang

menggunakan bentuk oval untuk menggambarkan fungsi sistem secara spesifik, tanda panah untuk merepresentasikan alur sistem, simbol belah ketupat untuk menggambarkan percabangan keputusan, garis horisontal untuk menunjukkan kegiatan *parallel* yang terjadi. *Activity Diagram* untuk aplikasi *HoTest* digambarkan dalam gambar 3.4, gambar 3.5, gambar 3.6, dan gambar 3.7 sebagai berikut :



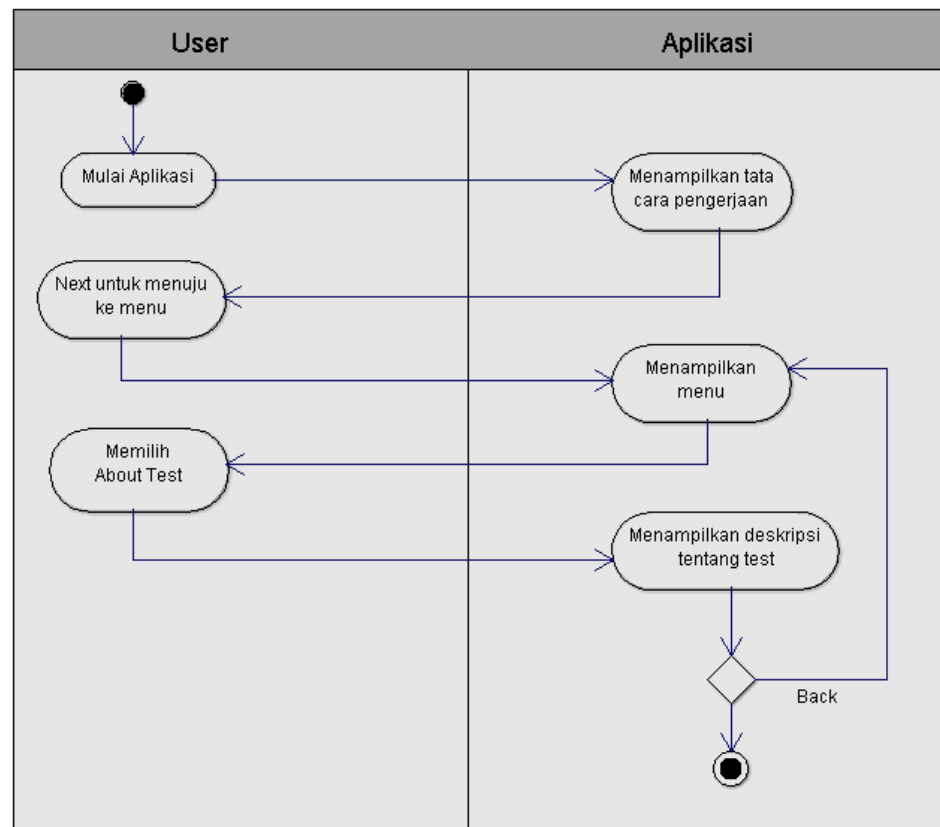
Gambar 3.4 *Activity Diagram* "Start Test" Aplikasi Tes Psikologi *HoTest*

Activity Diagram pada gambar 3.4 menggambarkan alur dalam pengerjaan soal. Alur pengerjaan soal oleh *testee* (peserta tes) dimulai dengan memilih *button* “Start Tes” pada halaman menu. Pengerjaan soal dimulai dengan pengisian data diri oleh *testee* dan dilanjutkan dengan pengerjaan soal. Soal akan ditampilkan sesuai dengan jenis soal pada Tes SDS (*Self-Directed Search*) untuk setiap halaman yang terdiri atas 4 jenis soal yaitu Aktivitas, Kompetensi, Pekerjaan, dan Penilaian Diri, dengan 6 tema soal yaitu (R, I, A, S, E, C). Setelah pengerjaan soal selesai *testee* akan dibawa ke halaman pengisian minat ekspresif peserta tes. Dalam halaman minat ekspresif peserta tes, akan terdapat *button* selesai. Apabila *testee* memilih *button* selesai, maka menandakan tes telah selesai dilaksanakan.



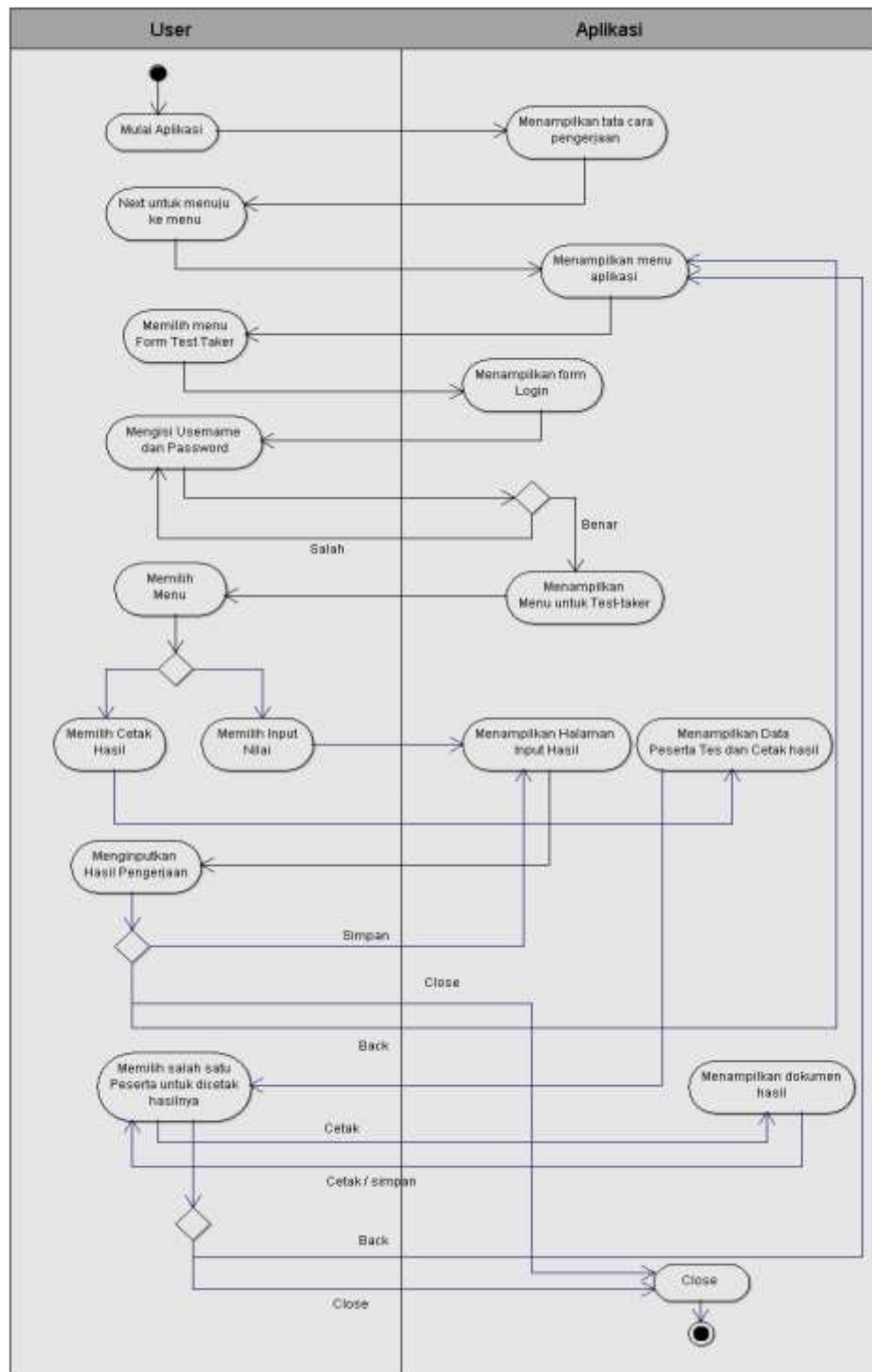
Gambar 3.5 *Activity Diagram* Prosedur Tes Aplikasi Tes Psikologi *HoTest*

Activity Diagram pada gambar 3.5 menggambarkan alur untuk masuk ke dalam menu “Prosedur Tes”. Dalam menu tersebut berisi prosedur pelaksanaan tes yang akan diikuti oleh *testee*. Jika dari halaman “Prosedur Pengerjaan” *user* memilih *button* “Back” maka *user* akan kembali ke dalam menu pilihan dan jika *user* memilih *close* maka program akan ditutup.



Gambar 3.6 *Activity Diagram* “About Test” Aplikasi Tes Psikologi *HoTest*

Activity Diagram pada gambar 3.6 menggambarkan alur untuk masuk ke dalam menu “About Test”. Dalam menu tersebut berisi profil pengembang dan sedikit teori tentang Tes SDS (*Self-Directed Search*). Jika dari halaman “About Test” *user* memilih *button* “Back” maka *user* akan kembali ke dalam menu pilihan dan jika *user* memilih *close* maka program akan ditutup.



Gambar 3.7 Activity Diagram "Form Test-taker" Aplikasi HoTest

Activity Diagram pada gambar 3.7 menggambarkan alur untuk masuk ke dalam menu “Form Test-taker”. Dalam menu tersebut *test-taker* harus *login* dengan mengisikan *username* dan *password*. Kemudian apabila *username* dan *password* benar, maka *test-taker* akan dibawa ke halaman menu untuk *test-taker* yang berisi menu “Input Nilai” dan “Cetak Hasil”. Apabila *test-taker* memilih menu “Input Nilai” maka akan muncul halaman pengisian nilai. Setelah mengisikan nilai atau hasil, apabila *test-taker* memilih *button* simpan maka data nilai yang diinputkan akan tersimpan dan form pengisian akan kembali kosong. Apabila dari menu tersebut *test-taker* memilih *button* “Back” maka akan kembali ke menu test-taker dan apabila memilih *button close* maka program akan berhenti atau selesai.

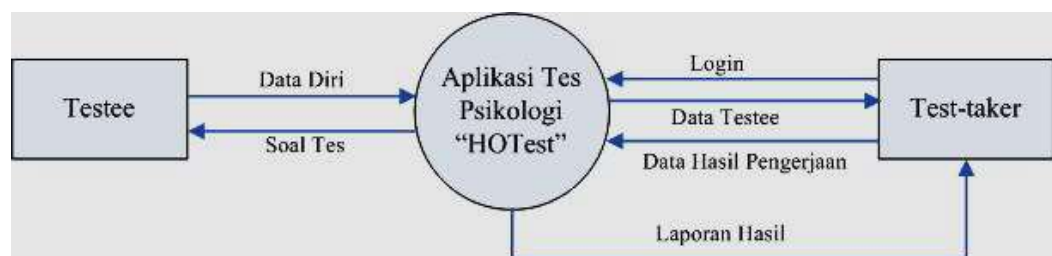
Dari menu *test-taker*, apabila *test-taker* memilih menu “Cetak Hasil” maka akan tampil halaman cetak hasil yang berisi semua data peserta. Untuk mencetak laporan hasil, *test-taker* harus memilih salah satu data peserta kemudian memilih *button* “Cetak Laporan Hasil”, maka akan muncul dokumen hasil pengerjaan. Pada halaman ini akan terdapat *button* “Back” untuk kembali ke menu *test-taker*. Apabila *test-taker* memilih *close* maka program akan berhenti atau selesai.

3.1.3.2 Flow-Oriented Modelling

Pemodelan aliran data merupakan salah satu yang paling banyak digunakan dalam analisis saat ini. Model aliran data digunakan untuk mengetahui bagaimana data mengalir melalui serangkaian langkah pemrosesan. Meskipun DFD (*Data Flow Diagram*) bukan merupakan bagian dari UML, namun dapat

digunakan untuk melengkapi diagram UML (*Unified Modelling Language*). Data Flow Diagram diawali dengan pembuatan diagram konteks yang digambarkan pada gambar 3.9 dilanjutkan dengan diagram level 1 pada gambar 3.10 sebagai berikut :

1. Diagram Konteks



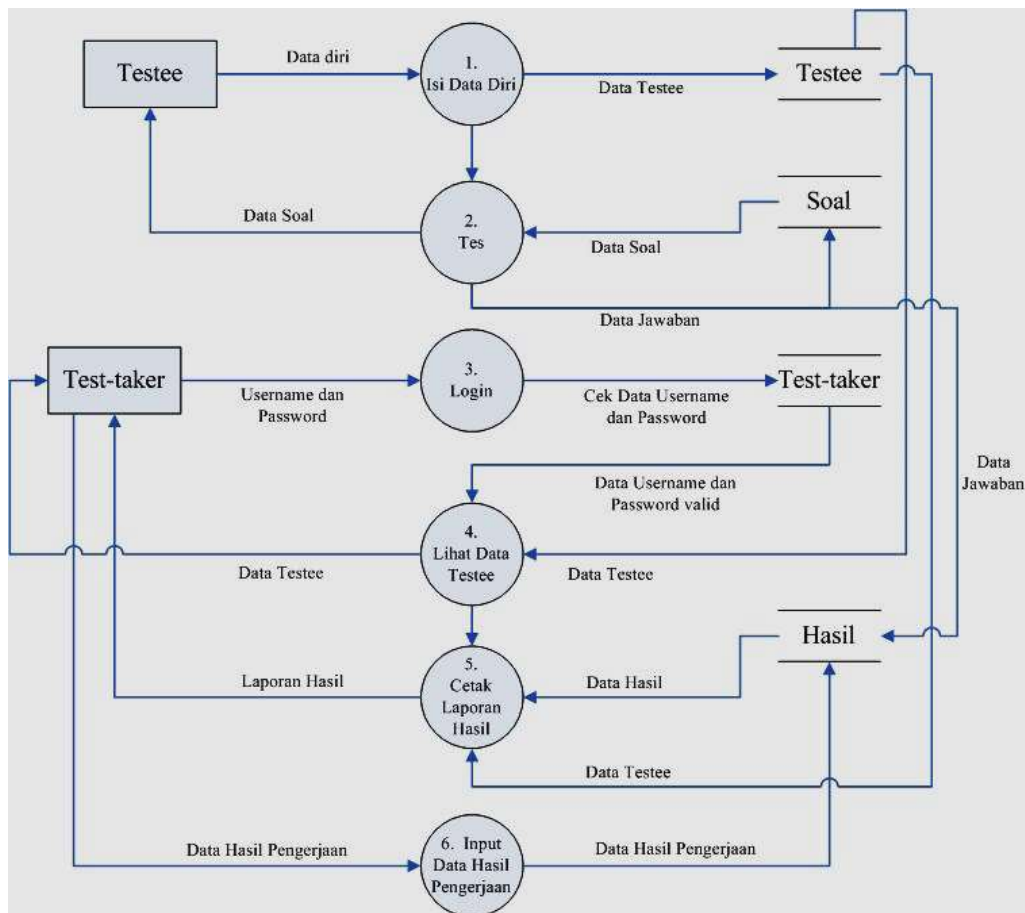
Gambar 3.8 Diagram Konteks Aplikasi Tes Psikologi *HoTest*

Diagram konteks pada gambar 3.7 menggambarkan proses besar dalam aplikasi *HoTest* yang nantinya akan didekomposisi menjadi proses-proses yang lebih detail. Diagram konteks yang diusulkan memiliki 2 buah entitas yaitu *Testee* dan *Test-taker*. Keterangan:

1. *Testee* dapat memulai mengerjakan tes diawali dengan mengisi data diri terlebih dahulu.
2. *Test-taker* sebagai pengelola aplikasi atau dapat dikatakan sebagai admin yang dapat melihat data *testee* yang melaksanakan tes dan laporan hasil pengerjaan tes dengan Login terlebih dahulu.

2. Diagram Level 1

DFD Level 1 pada gambar 3.8 menggambarkan semua aliran data proses pelaksanaan tes psikologi dalam hal ini tes minat yang menerapkan alat ukur SDS (*Self-Directed Search*) ke dalam aplikasi *HoTest*. Gambar DFD Level 1 digambarkan sebagai berikut:



Gambar 3.9 DFD Level 1

Proses-proses yang terlibat dalam DFD Level 1 untuk aplikasi *HoTest* dijelaskan dalam tabel berikut:

Tabel 3.4 Proses-proses yang Terlibat dalam DFD Level 1

No.	Nama Proses	Aliran Data Masuk (Input)	Aliran Data Keluar (Output)	Keterangan
1.	Isi Data Diri	<i>Testee</i>	<i>Testee</i>	<i>Testee</i> harus mengisi data diri terlebih dahulu

				sebelum melaksanakan tes.
2.	Tes	Data Soal	Jawaban	Untuk memulai mengerjakan soal diperlukan data soal dan hasil pengerjaan berupa jawaban. Dalam memulai mengerjakan soal, <i>testee</i> sudah mengisi data diri.
3.	Login	<i>Test-taker</i>	<i>Test-taker</i>	Untuk dapat melihat data <i>testee</i> yang mengikuti tes <i>test-taker</i> harus Login.
4.	Lihat Data <i>Testee</i> dan Hasil	Data <i>Testee</i> , Data Hasil	Laporan Hasil	<i>Test taker</i> melihat data <i>testee</i> yang sudah mengikuti tes setelah Login dan dapat melihat,

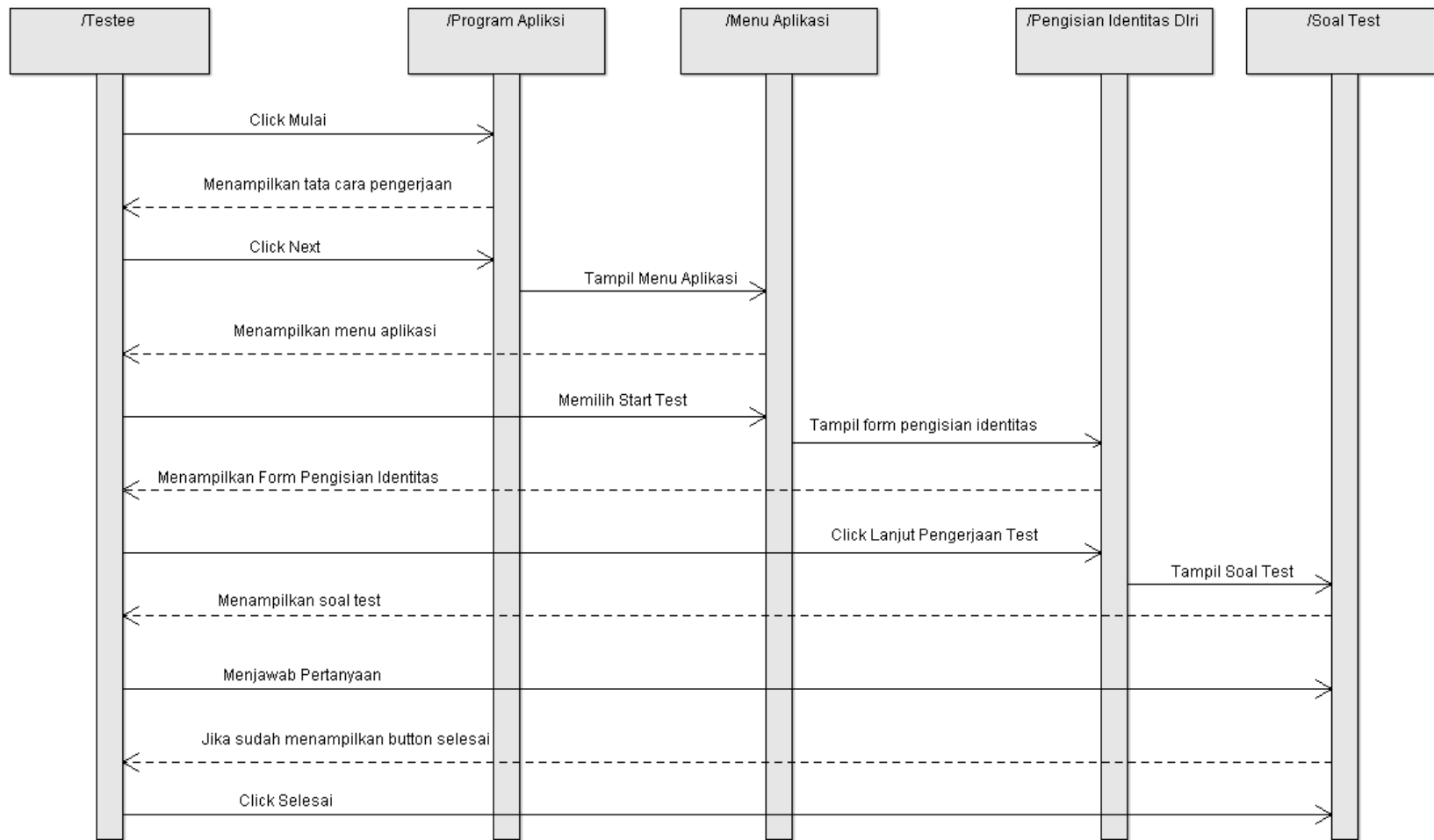
				menyimpan, ataupun mencetak laporan hasil tes.
5.	Input Hasil Pengerjaan	Data <i>Testee</i> , Data Hasil	Hasil pengerjaan	Test-taker menginputkan data <i>testee</i> ke dalam program apabila dilaksanakan tes secara manual. Program berfungsi untuk menghitung dan menginputkan laporan hasil

Sumber : Hasil Olahan Penulis, 2015

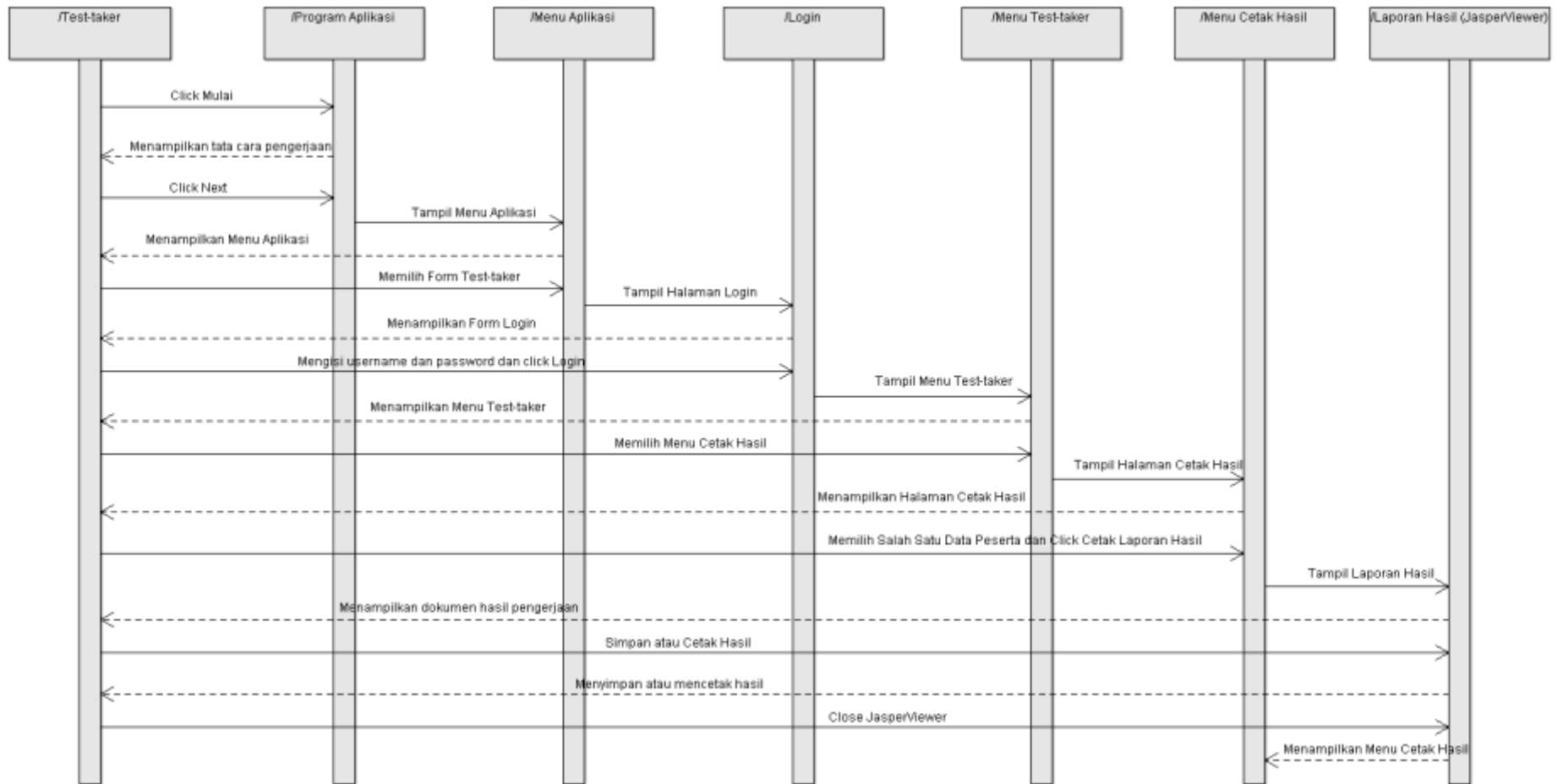
3.1.3.3 Behavioral Modelling

Bentuk kedua untuk merepresentasikan tingkah laku selain *state diagram* adalah *sequence diagram* dalam UML (*Unified Modelling Language*) yang menunjukkan peristiwa transisi dari suatu objek ke objek lain. *State diagram* untuk aplikasi *HoTest* yaitu untuk menunjukkan peristiwa transisi oleh *testee* dalam proses pelaksanaan tes atau pengerjaan soal digambarkan pada gambar 3.10

dan proses pencetakan hasil pengerjaan yang menjadi hak akses *test-taker* digambarkan pada gambar 3.11 sebagai berikut :



Gambar 3.10 *Sequence Diagram* Pengerjaan Soal Aplikasi Tes Psikologi *HoTest*

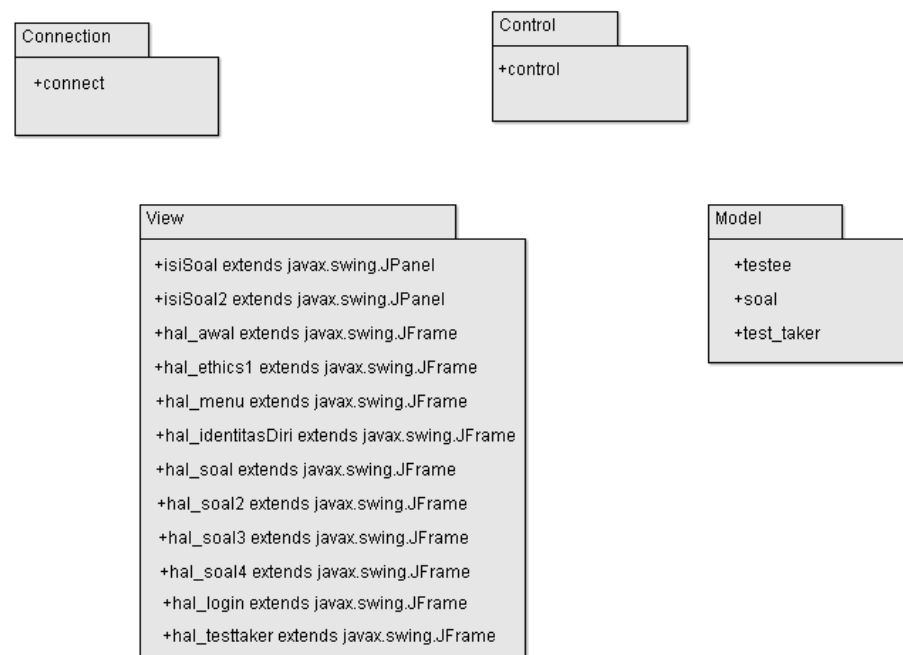


Gambar 3.11 *Sequence Diagram* Cetak Hasil Aplikasi Tes Psikologi *HoTest*

3.1.3.4 Class-based environment

1. Analysis Packages

Bagian terpenting dalam pemodelan analisis adalah *categorization*. Berbagai elemen pada pemodelan analisis dikategorisasikan ke dalam *packages* sehingga diperlukan analisis terhadap *package*. Dalam setiap *package* terdapat analisis dari nama *class* yang dapat menunjukkan hubungan antar *package* yaitu suatu *package* dapat diakses oleh *packages* yang lain atau tidak. *Analysis packages* aplikasi *HoTest* digambarkan pada gambar 3.12 sebagai berikut :



Gambar 3.12 *Analysis Packages* Aplikasi Tes Psikologi *HoTest*

2. Analysis Database

Analisis *database* digunakan untuk menganalisis kebutuhan *database* yang diperlukan dalam pengembangan aplikasi. Analisis kebutuhan *database* untuk aplikasi *HoTest* sebagai berikut :

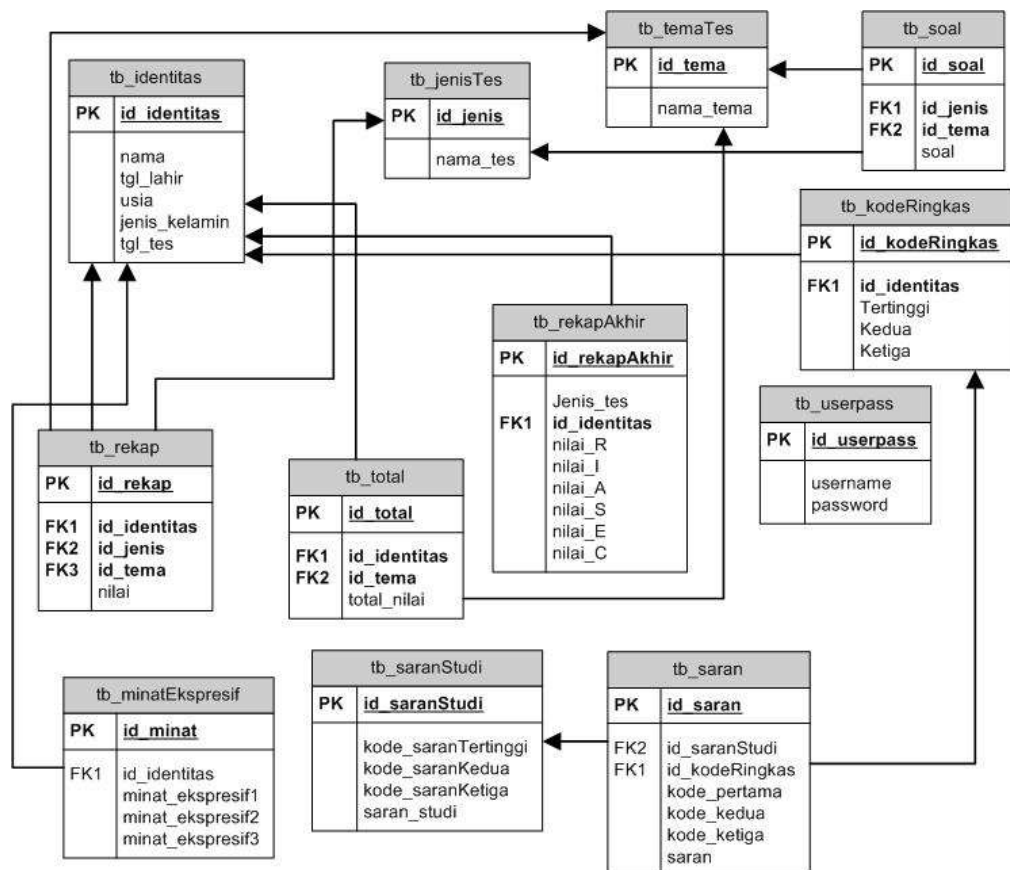
Tabel 3.5 Daftar Entitas dan Atribut pada Aplikasi *HoTest*

No.	Entitas	Atribut
1.	tb_identitas	id_identitas , nama, tgl_lahir usia, jenis_kelamin tgl_tes
2.	tb_jenisTes	id_jenis , nama_tes
3.	tb_temaTes	id_tema , nama_tema
4.	tb_soal	id_soal , id_jenis, id_tema, soal
5.	tb_rekap	id_rekap , id_identitas, id_jenis, id_tema, nilai
6.	tb_total	id_total , id_identitas, id_tema, total_nilai
7.	tb_rekapAkhir	id_rekapAkhir , Jenis_tes, id_identitas, nilai_R, nilai_I, nilai_A, nilai_S, nilai_E, nilai_C
8.	tb_kodeRingkas	id_kodeRingkas , id_identitas, Tertinggi, Kedua, Ketiga
9.	tb_userpass	id-userpass , username, password
10.	tb_minatEkspresif	id_minat , id_identitas, minat_ekspresif1,

		minat_ekspresif2, minat_ekspresif3
11.	tb_saranStudi	id_saranStudi , kode_saranTertinggi, kode_saranKedua, kode_saranKetiga, saran_studi
12.	tb_saran	id_saran , id_identitas, kode_tertinggi, kode_kedua, kode_ketiga, saran

Sumber: Hasil Olahan Penulis, 2015

Relasi untuk database di atas digambarkan pada gambar 3.13 sebagai berikut :



Gambar 3.13 Relasi Database Aplikasi HoTest

3.2 Desain Perangkat Lunak

Pada tahap ini dilakukan desain untuk aplikasi tes SDS (*Self-Directed Search*) yang meliputi desain data atau *class*, desain arsitektur, dan desain *interface*.

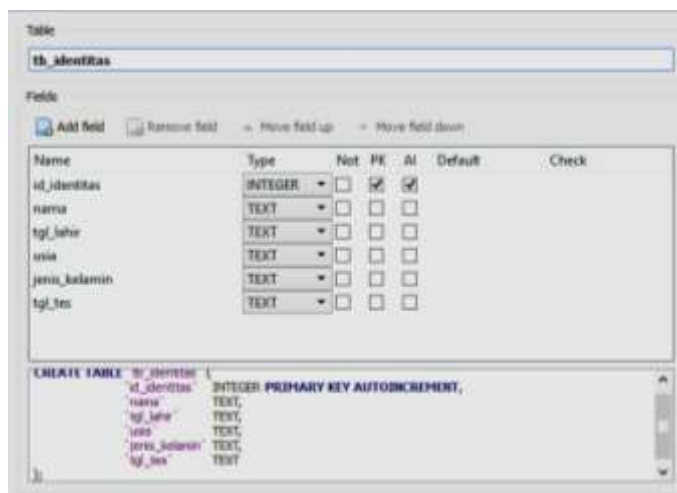
3.2.1 Desain Data atau *Class*

Desain data mentransformasikan model domain informasi yang dibuat selama analisis ke dalam struktur data yang akan diperlukan untuk melaksanakan perangkat lunak. Bagian desain data dapat terjadi dengan hubungannya dengan desain arsitektur. Desain data dalam *database* sebagai berikut:

1. Tabel Identitas

Primary key : `id_identitas`

Fungsi : Menyimpan data *testee* atau peserta tes dan digunakan dalam pengolahan nilai.

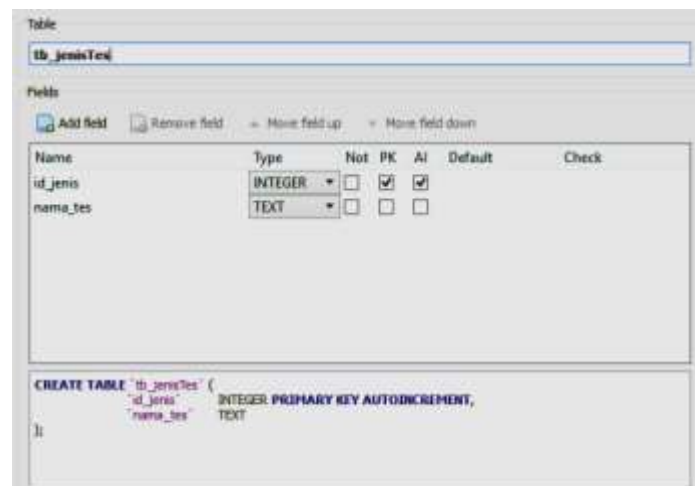


Gambar 3.14 Tabel Identitas

2. Tabel Jenis Tes

Primary key : `id_jenisTes`

Fungsi : Mendefinisikan jenis tes untuk aplikasi *HoTest*.

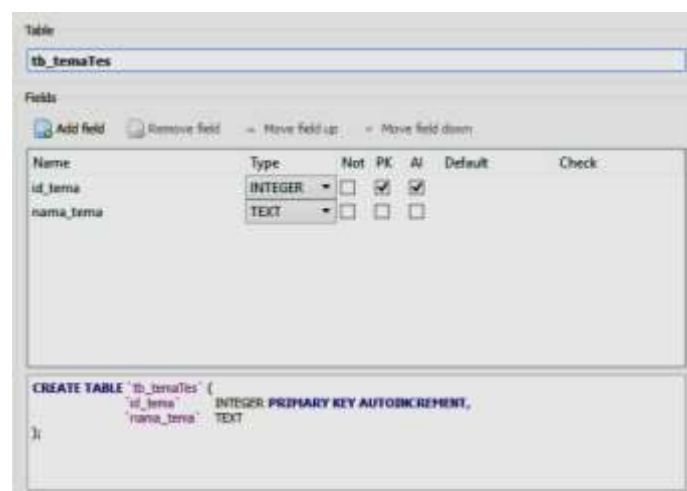


Gambar 3.15 Tabel Jenis Tes

3. Tabel Tema Tes

Primary key : id_temaTes

Fungsi : Mendefinisikan tema tes untuk aplikasi *HoTest*.



Gambar 3.16 Tabel Tema Tes

4. Tabel Soal

Primary key : id_soal

Fungsi : Mengambil soal untuk ditampilkan dalam halaman soal (tempat penyimpanan soal).

The screenshot shows a database management interface for a table named 'tb_soal'. It displays a list of fields with their respective types and attributes. Below the list, there is a 'CREATE TABLE' statement for 'tb_soal'.

Name	Type	Not	PK	AI	Default	Check
id_soal	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
id_jenis	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_tema	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
soal	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

CREATE TABLE "tb_soal" (
  "id_soal" INTEGER PRIMARY KEY AUTOINCREMENT,
  "id_jenis" TEXT,
  "id_tema" TEXT,
  "soal" TEXT
);

```

Gambar 3.17 Tabel Soal

5. Tabel Rekap

Primary key : id_rekap

Fungsi : Menyimpan nilai hasil pengerjaan.

The screenshot shows a database management interface for a table named 'tb_rekap'. It displays a list of fields with their respective types and attributes. Below the list, there is a 'CREATE TABLE' statement for 'tb_rekap'.

Name	Type	Not	PK	AI	Default	Check
id_rekap	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
id_identitas	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_jenis	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_tema	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
nilai	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

CREATE TABLE "tb_rekap" (
  "id_rekap" INTEGER PRIMARY KEY AUTOINCREMENT,
  "id_identitas" INTEGER,
  "id_jenis" INTEGER,
  "id_tema" INTEGER,
  "nilai" INTEGER
);

```

Gambar 3.18 Tabel Rekap

6. Tabel Total

Primary key : id_total

Fungsi : Menyimpan nilai total berdasarkan jenis dan tema soal.

The screenshot shows a database management interface for a table named 'tb_total'. The 'Fields' section contains the following data:

Name	Type	Not	PK	AI	Default	Check
id_total	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
id_identitas	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_tema	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
total_nilai	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Below the fields table, the SQL code for creating the table is displayed:

```
CREATE TABLE `tb_total` (
  `id_total` INTEGER PRIMARY KEY AUTOINCREMENT,
  `id_identitas` INTEGER,
  `id_tema` INTEGER,
  `total_nilai` INTEGER
);
```

Gambar 3.19 Tabel Total

7. Tabel Rekap Akhir

Primary key : id_rekapAkhir

Fungsi : Menyimpan hasil dan total nilai dari setiap tema tes.

The screenshot shows a database management interface for a table named 'tb_rekapAkhir'. The 'Fields' section contains the following data:

Name	Type	Not	PK	AI	Default	Check
id_rekapAkhir	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
jenis_tes	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_identitas	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
nilai_R	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
nilai_I	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
nilai_A	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
nilai_S	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Below the fields table, the SQL code for creating the table is displayed:

```
CREATE TABLE `tb_rekapAkhir` (
  `id_rekapAkhir` INTEGER PRIMARY KEY AUTOINCREMENT,
  `jenis_tes` TEXT,
  `id_identitas` INTEGER,
  `nilai_R` INTEGER,
  `nilai_I` INTEGER,
  `nilai_A` INTEGER,
  `nilai_S` INTEGER,
  `nilai_C` INTEGER
);
```

Gambar 3.20 Tabel Rekap Akhir

8. Tabel Kode Ringkas

Primary key : id_kodeRingkas

Fungsi : Menyimpan tema tes berdasarkan tiga nilai tertinggi dari hasil pengerjaan.

The screenshot shows a database management interface for a table named 'tb_koderingkas'. It displays a list of fields with their data types and constraints. Below the list is a SQL 'CREATE TABLE' statement.

Name	Type	Null	PK	AI	Default	Check
id_koderingkas	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
id_identitas	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Tertinggi	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Kedua	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Ketiga	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

CREATE TABLE 'tb_koderingkas' (
  id_koderingkas INTEGER PRIMARY KEY AUTOINCREMENT,
  id_identitas INTEGER,
  tertinggi TEXT,
  kedua TEXT,
  ketiga TEXT
);

```

Gambar 3.21 Tabel Kode Ringkas

9. Tabel User (*Test-taker*)

Primary key : id_userpass

Fungsi : Menyimpan *username* dan *password* bagi *test-taker* untuk *login*.

The screenshot shows a database management interface for a table named 'tb_userpass'. It displays a list of fields with their data types and constraints. Below the list is a SQL 'CREATE TABLE' statement.

Name	Type	Null	PK	AI	Default	Check
id_userpass	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
username	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
password	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

CREATE TABLE 'tb_userpass' (
  id_userpass INTEGER PRIMARY KEY AUTOINCREMENT,
  username TEXT,
  password TEXT
);

```

Gambar 3.22 Tabel Userpass (*Username dan Password*)

10. Tabel Minat Ekspresif

Primary key : id_minat

Fungsi : Menyimpan minat ekspresif peserta tes.

Name	Type	Not	PK	AI	Default	Check
id_minat	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
id_identitas	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
minat_ekspresif1	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
minat_ekspresif2	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
minat_ekspresif3	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

CREATE TABLE `tb_minatEkspresif` (
  `id_minat` INTEGER PRIMARY KEY AUTOINCREMENT,
  `id_identitas` INTEGER,
  `minat_ekspresif1` TEXT,
  `minat_ekspresif2` TEXT,
  `minat_ekspresif3` TEXT
)

```

Gambar 3.23 Tabel Minat Ekspresif

11. Tabel Saran Studi

Primary key : id_saranStudi

Fungsi : Menyimpan saran studi untuk mengidentifikasi minat peserta tes berdasarkan hasil tes.

Name	Type	Not	PK	AI	Default	Check
id_saranStudi	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
kode_saranTerltinggi	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
kode_saranKedua	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
kode_saranKetiga	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
saran_studi	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

CREATE TABLE `tb_saranStudi` (
  `id_saranStudi` INTEGER PRIMARY KEY AUTOINCREMENT,
  `kode_saranTerltinggi` TEXT,
  `kode_saranKedua` TEXT,
  `kode_saranKetiga` TEXT,
  `saran_studi` TEXT
)

```

Gambar 3.24 Tabel Saran Studi

12. Tabel Saran

Primary key : id_saran

Fungsi : Menyimpan hasil saran berdasarkan kode ringkas dan saran studi.

Name	Type	Not	PK	AI	Default	Check
id_saran	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
id_kodeRingkas	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
id_saranStudi	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
kode_sertinggg	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
kode_kelua	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
kode_kelga	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
saran	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

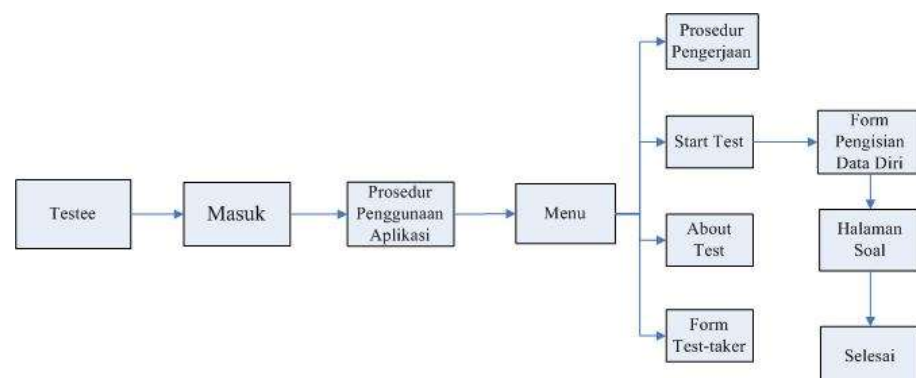
Gambar 3.25 Tabel Saran

3.2.2 Desain Arsitektur

3.2.2.1 Arsitektur Design Menu Testee

Desain arsitektur menu untuk *testee* menunjukkan alur penggunaan aplikasi oleh *testee* atau peserta tes dari mulai masuk hingga pengerjaan soal.

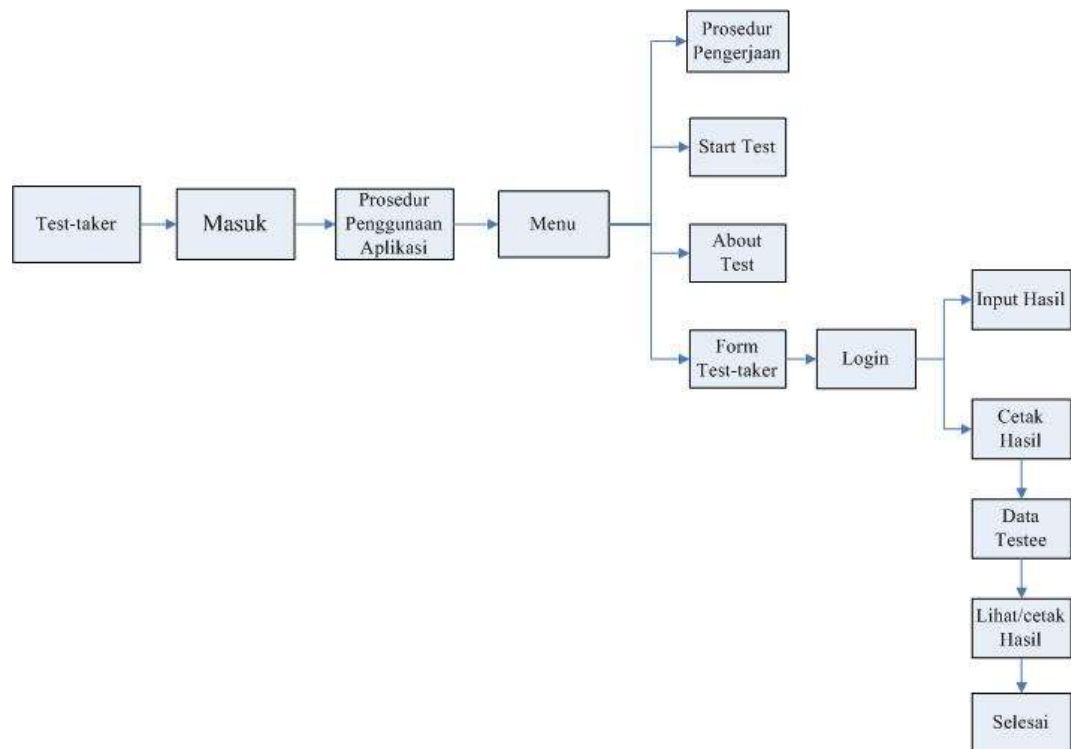
Desain arsitektur menu *testee* digambarkan pada gambar 3.26 sebagai berikut :



Gambar 3.26 Arsitektur Desain Menu *Testee*

3.2.2.2 Arsitektur Design Menu Test-taker

Desain arsitektur menu untuk *test-taker* digambarkan pada gambar 3.27 sebagai berikut :

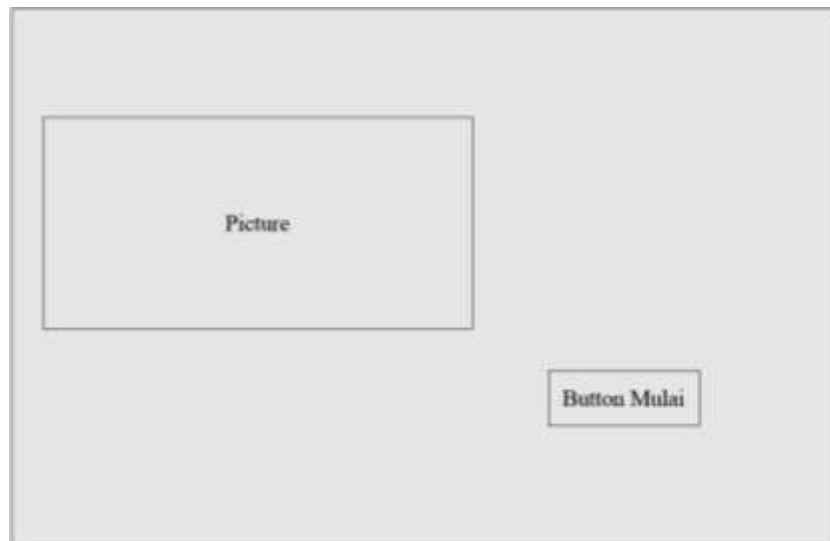


Gambar 3.27 Arsitektur Desain Menu *Test-taker*

3.2.3 Desain Interface

Dari hasil *Scenario-based Modelling*, *Flow-Oriented Modelling*, dan *Behavioral Modelling* dapat dibuat perancangan untuk desain *interface*. Desain *interface* untuk aplikasi *HoTest (Holland Advanced-Study Test)* digambarkan sebagai berikut:

1. Desain Halaman Awal



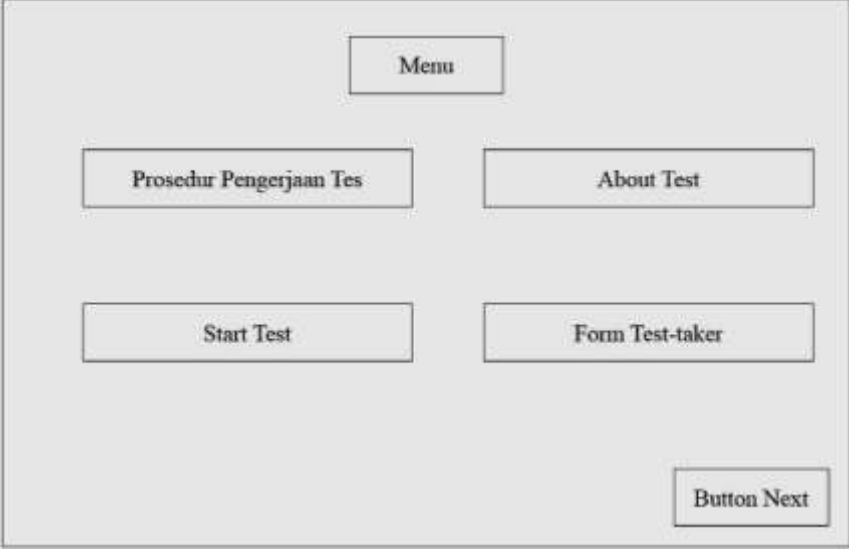
Gambar 3.28 Desain *Interface* Halaman Awal Aplikasi

2. Desain Halaman Prosedur Penggunaan Aplikasi



Gambar 3.29 Desain *Interface* Halaman Prosedur Pengerjaan Aplikasi


3. Desain Halaman Menu



The screenshot shows a menu interface with a central title box labeled "Menu". Below the title, there are four buttons arranged in a 2x2 grid: "Prosedur Pengerjaan Tes" (top-left), "About Test" (top-right), "Start Test" (bottom-left), and "Form Test-taker" (bottom-right). A "Button Next" is located in the bottom right corner of the interface.

Gambar 3.30 Desain *Interface* Halaman Menu Aplikasi

4. Desain Halaman Pengisian Data Diri



The screenshot shows a self-data entry form titled "Form Pengisian Data Diri". It contains four input fields, each with a label and a colon separator: "Nama :", "Usia :", "Jenis Kelamin :", and "Tanggal Tes :". A "Button Next" is located in the bottom right corner of the form area.

Gambar 3.31 Desain *Interface* Halaman Pengisian Data Diri

5. Desain Halaman Soal

The wireframe shows a rectangular frame containing several elements. At the top left is a box labeled 'Jenis Tes'. At the top right is a box labeled 'Id Testee :'. Below these is a horizontal bar with a slider and an arrow pointing right, labeled 'Tema Tes'. Inside a larger central box, there are two stacked input fields: the top one is labeled 'Soal' and the bottom one is labeled 'Pilihan Jawaban'. Below these fields is the text 'dan seterusnya..'. At the bottom right of the main frame is a box labeled 'Button Next'.

Gambar 3.32 Desain *Interface* Halaman Pengisian Data Diri

6. Desain Halaman Login *Test-taker*

The wireframe shows a rectangular frame with a central box titled 'Login Test-taker'. Inside this box are two input fields: the top one is labeled 'Username :' and the bottom one is labeled 'Password :'. At the bottom right of the main frame is a box labeled 'Button Next'.

Gambar 3.33 Desain *Interface* Halaman Login *Test-taker*

7. Desain Halaman Cetak Hasil *Test-taker*

Cetak Laporan Hasil Pengerjaan

No.	Nama	Usia	Jenis Kelamin	Tanggal Tes
1.				
2.				

dan seterusnya..

Button Refresh

Button Cetak Edit Data

Button Hapus

Button Cetak Laporan Pengerjaan

Gambar 3.34 Desain *Interface* Halaman Data *Testee* dan Cetak Hasil

8. Desain Halaman Input Hasil Pengerjaan

Input Hasil Pengerjaan

A. Identitas Diri

Nama

Tanggal Lahir

B. Hasil Pengerjaan

Jenis Tes 1 Nilai Nilai
 Tema 1 Tema 2 Tema 3 Tema ..

Jenis Tes 2
 Tema ..

C. Minat Ekspresif

1.

2.

3.

Button Simpan

Gambar 3.35 Desain *Interface* Halaman Input Hasil Pengerjaan

3.3 Pengkodean

Pada tahap ini dilakukan pembuatan aplikasi menggunakan Netbeans 7.4. Pengkodean dilakukan terhadap rancangan-rancangan baik rancangan aplikasi maupun rancangan tampilan. Bahasa pemrograman yang digunakan dalam pengembangan aplikasi ini adalah menggunakan Java.

SQLite digunakan sebagai *database* untuk menyimpan data dengan menggunakan versi 3.0. SQLite bersifat *portable* sehingga *database* yang dibuat dalam SQLite di *Export* dalam bentuk SQL File. Untuk *editor* atau pengkodean aplikasi dan tampilan, digunakan Netbeans dengan menggunakan versi 7.4. Dalam pembuatan aplikasi *HoTest (Holland Advanced-Study Test)* juga digunakan *report* atau laporan sehingga dalam Netbeans ditambahkan *plugin JasperReport* dengan versi 3.5.2. Berikut ditampilkan beberapa *source code* untuk aplikasi *HoTest (Holland Advanced-Study Test)* pada tabel 3.6.

Tabel 3.6 *Source Code* Aplikasi *HoTest (Holland Advanced-Study Test)*

Package	Fungsi	Source Code
Connection	Koneksi ke database	<pre>public class connect { private Connection koneksi; public Connection connect(){ try{ Class.forName("org.sqlite.JDBC"); }catch(ClassNotFoundException ex){ JOptionPane.showMessageDialog(null, "Tidak ada Driver!\n" + ex); } try{ String url="jdbc:sqlite:db/db_HoTest"; koneksi=DriverManager.getConnection(url); System.out.println("Berhasil koneksi"); }catch(SQLException se){ System.out.println("Gagal koneksi "+se); } } }</pre>

		<pre> JOptionPane.showMessageDialog(null,"Gagal Koneksi Database","Peringatan",JOptionPane.WARNING_ MESSAGE); } return koneksi; } public Connection getConnection(){ return koneksi; } } </pre>
Control	Input data diri peserta tes	<pre> public void insert(){ String nama, usia, jenis_kelamin = null, tgl_tes; nama=hal_id.getTxt_nama().getText(); usia=hal_id.getTxt_usia().getText(); if (hal_id.getRB_P().isSelected()){ jenis_kelamin=hal_id.getRB_P().getText(); }else if(hal_id.getRB_L().isSelected()){ jenis_kelamin=hal_id.getRB_L().getText(); } DateFormat df=new SimpleDateFormat("dd MMMM yyyy"); tgl_tes=df.format(hal_id.getJDate().getDate()); testee user=new testee(); user.setNama(nama); user.setUsia(usia); user.setJenis_kelamin(jenis_kelamin); user.setTgl_tes(tgl_tes); user.getId_identitas(); user.insertUser(); } </pre>
Model	Ambil soal dari <i>database</i>	<pre> public void soaltesR() { try{ int panjang=0; GridLayout gr; Connection koneksi = new connect().connect(); String sql ="SELECT soal FROM tb_soal WHERE id_jenis=? AND id_tema=?"; java.sql.PreparedStatement ps = koneksi.prepareStatement(sql); ps.setInt(1, 1); ps.setInt(2, 1); </pre>

		<pre> java.sql.ResultSet res=ps.executeQuery(); while(res.next()){ panjang+=1; gr = new GridLayout(panjang, 0); sol = new IsiSoal(); sol.getTxtSoal().setText(res.getString("soal")); listR.add(sol); hs.getPanelSoal().add(sol); hs.getPanelSoal().setLayout(gr); } } catch (SQLException se){ JOptionPane.showMessageDialog(null,"Data Gagal disimpan!" + "\nDengan Pesan : " + se.getMessage()); } </pre>
Model	Login	<pre> public void login(){ try{ Connection koneksi = new connect().connect(); String sql="select *from tb_userpass where "+"username="+hl.getTxt_username().getText().to String()+"""; java.sql.PreparedStatement ps = koneksi.prepareStatement(sql); java.sql.ResultSet res=ps.executeQuery(); if(res.next()){ if (hl.getTxt_password().getText().toString().equals(re s.getString("password"))){ this.getPassword(); hal_testtaker hal=new hal_testtaker(); tampilData(); hal.getTabel().setModel(model); hal.show(); hl.setVisible(false); }else{ JOptionPane.showMessageDialog(null,"Password salah"); hl.getTxt_password().setText(""); hl.getTxt_password().requestFocus(); } }else{ JOptionPane.showMessageDialog(null, </pre>

		<pre> "Username tidak ditemukan"); hl.getTxt_username().setText(""); hl.getTxt_password().setText(""); hl.getTxt_username().requestFocus(); } } catch (SQLException se){ OptionPane.showMessageDialog(null,"Data Gagal disimpan!" + "\nDengan Pesan : " + se.getMessage()); } } </pre>
	<p>Input data diri peserta tes ke <i>database</i></p>	<pre> public void insertUser() { try{ Connection koneksi = new connect().connect(); String sql ="INSERT INTO tb_identitas(nama,usia,jenis_kelamin,tgl_tes)+" VALUES (?,?,,?)"; java.sql.PreparedStatement ps = koneksi.prepareStatement(sql); ps.setString(1, nama); ps.setString(2, usia); ps.setString(3, jenis_kelamin); ps.setString(4, tgl_tes); ps.executeUpdate(); JOptionPane.showMessageDialog(null, "Data berhasil disimpan"); } catch (SQLException se){ OptionPane.showMessageDialog(null,"Data Gagal disimpan!" + "\nDengan Pesan : " + se.getMessage()); } } </pre>
View	<p><i>Button</i> simpan hasil soal Aktivitas</p>	<pre> private void bt_simpanActionPerformed(java.awt.event.ActionE vent evt) { // TODO add your handling code here: con.hasil1(); hal_soal2 s = null; try { s = new hal_soal2(); } catch (SQLException ex) { Logger.getLogger(hal_soal.class.getName()).log(Le </pre>

		<pre> vel.SEVERE, null, ex); } con =new control(s); con.tampil2(); s.setCon(con); s.show(); this.setVisible(false); } </pre>
--	--	--

Sumber : Hasil Olahan Penulis, 2015

3.4 Pengujian

Pada tahap ini dilakukan pengujian terhadap aplikasi yang sudah dibuat. Pengujian yang dilakukan menggunakan pengujian analisis sistem. Metode yang digunakan untuk menganalisis sistem ini yaitu metode PIECES yang dapat dilihat pada tabel 3.7 sebagai berikut:

Tabel 3.7 Perbedaan Sistem Lama dan Sistem Baru

Jenis Analisis	Sistem Lama	Sistem yang Diajukan
Performance	<p>Pelaksanaan tes minat sebagian besar masih menggunakan cara manual yaitu dengan menggunakan pensil dan kertas dalam pengerjaannya. Sehingga dibutuhkan waktu yang cukup lama dari mulai dilaksanakannya pengerjaan tes, proses penghitungan, hingga</p>	<p>Aplikasi yang dibuat ditinjau dari segi kemampuan aplikasi dalam menyediakan sarana bagi pelaksanaan tes dengan lebih otomatis yaitu pengerjaan, penghitungan, dan keluarnya hasil tes akan dilaksanakan secara komputerisasi.</p>

	keluarnya hasil tes.	
Information	<p>Input :</p> <p>Informasi hasil penskoran yang sulit didapatkan khususnya apabila akan digunakan sewaktu-waktu untuk penelitian.</p> <p>Output :</p> <p>Data masih manual dan rentan terjadi suatu masalah. Penghitungan data hasil pengerjaan dilaksanakan secara manual sehingga sangat mungkin terjadi human error.</p>	<p>Input :</p> <p>Data akan aman dan mudah disimpan serta dapat digunakan kapan saja.</p> <p>Output :</p> <p>Kemudahan dalam mendapatkan informasi hasil pengerjaan dengan data penghitungan yang akurat sehingga menghasilkan informasi yang akurat.</p>
Economic	Penggunaan kertas yang hanya sekali pakai untuk soal, lembar jawab, dan laporan akan memakan lebih banyak biaya.	Dengan sistem komputerisasi akan menghemat waktu dan penggunaan kertas serta alat tulis lainnya.
Control	Dengan sistem pengerjaan	Sistem berbasis komputer

	<p>tes secara manual atau sistem lama akan dibutuhkan waktu yang lama dalam hal penyimpanan data, pencarian data, pembuatan laporan, dan dengan sistem lama sangat mungkin terjadi kehilangan atau kerusakan data baik lembar jawab ataupun laporan hasil pengerjaan. Sehingga dalam hal ini sistem lama sangat sulit dilakukan control.</p>	<p>dapat memudahkan control terhadap data sehingga kemungkinan kesalahan atau kehilangan dapat ditekan.</p>
Efficiency	<p>Sistem lama dalam pelaksanaan tes kurang efisien karena memakan waktu cukup lama dari mulai pengerjaan hingga keluarnya hasil atau laporan.</p>	<p>Lebih cepat dan efisien karena pelayanan dalam pelaksanaan tes dilaksanakan secara komputerisasi.</p>
Services	<p>Pelayanan pada peserta tes</p>	<p>Pelayanan terhadap peserta tes</p>

	akan memakan banyak waktu karena harus menunggu pemrosesan data hasil pengerjaan.	dapat lebih cepat karena pemrosesan hasil pengerjaan dan pengecekan data dilaksanakan menggunakan komputer.
--	---	---

Sumber : Hasil Olahan Penulis, 2015

Dalam menganalisis pemanfaatan aplikasi pada poin *Services* atau layanan, analisis yang dilakukan bertujuan untuk mengetahui validitas skoring dari aplikasi *HoTest*. Pengujian akan dilakukan di Laboratorium Psikologi Universitas Negeri Semarang.

Untuk mengetahui validitas skoring tersebut, pengujian dilakukan dengan memasukkan 10 data hasil tes yang telah dihitung secara manual ke dalam aplikasi *HoTest*. Dalam tahap pengujian ini, hasil yang didapatkan dari aplikasi *HoTest* akan dibandingkan dengan data yang telah dihitung secara manual. Dalam aplikasi *HoTest* juga ditambahkan saran studi yang didasarkan pada kamus Holland.

Sesuai dengan kebutuhan pengguna yang memfokuskan pembuatan aplikasi ini untuk peminatan studi lanjut, maka kamus Holland yang berisi 160 jenis pekerjaan dianalisis kembali bersama Laboratorium Psikologi Universitas Negeri Semarang untuk menyesuaikan jenis pekerjaan tersebut dengan jurusan atau program studi yang ada di Indonesia. Untuk lebih memperjelas tahap pengujian tersebut, berikut 3 contoh kombinasi kode beserta jenis pekerjaan menurut kamus Holland:

1. Kombinasi kode **SIA**

- Saran Pekerjaan: *Psychologist, Counseling*
- Saran Studi: Jurusan Psikologi, Jurusan Bimbingan dan Konseling

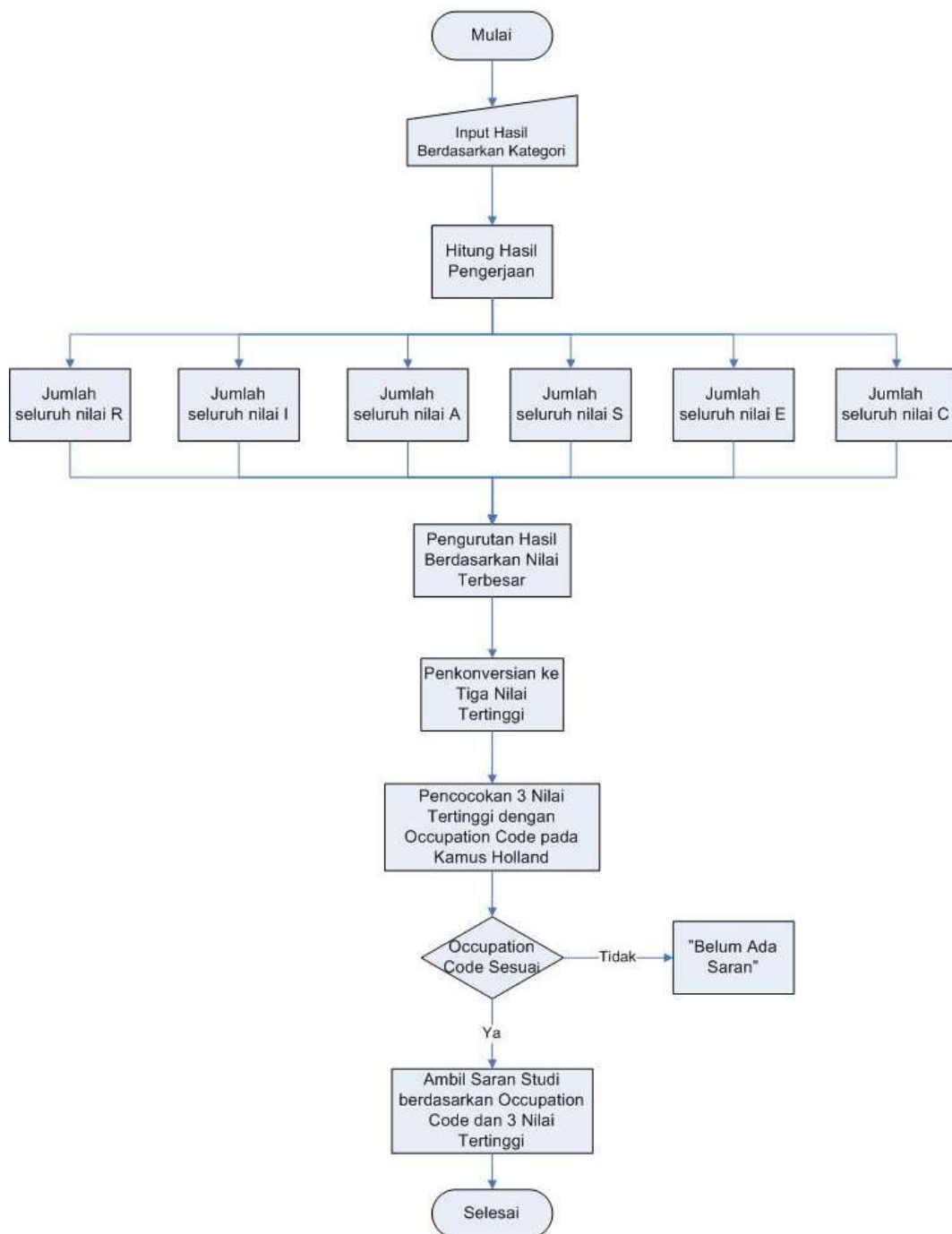
2. Kombinasi kode **CSI**

- Saran Pekerjaan: *Accounting and Computer Sciences, Computer Support Specialist, Health Information/Medical Record Technology*
- Saran Studi: Akuntansi, Teknik Komputer, Rekam Medis, dan lain sebagainya.

3. Kombinasi kode **SIC**

- Saran Pekerjaan: *Nurse, Critical Care, Informatics Nurse Specialist*
- Saran Studi: Keperawatan, Rekam Medis, dan lain sebagainya.

Contoh kombinasi kode di atas beserta saran pekerjaan yang dikemukakan oleh Holland dan kombinasi kode lain yang tidak bisa dicantumkan karena adanya pembatasan dalam hal kerahasiaan, dijadikan sebagai patokan untuk memberikan hasil pengujian dalam penentuan validitas skoring dengan hasil berupa tiga kode tertinggi dan saran studi dari aplikasi *HoTest (Holland Advanced-Study Test)*. Untuk memperjelas hasil dari penghitungan dalam penentuan validitas skoring dalam pengujian yang dilakukan, berikut merupakan tahap penghitungan hingga sampai pada penentuan saran studi dari aplikasi *HoTest*:



Gambar 3.36 Flowchart Penghitungan Hasil Aplikasi *HoTest* (*Holland Advanced-Study Test*)

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan hasil penelitian dari Aplikasi *HoTest (Holland Advanced-Study Test)* di Laboratorium Psikologi Universitas Negeri Semarang, dapat disimpulkan bahwa :

1. Aplikasi *HoTest (Holland Advanced-Study Test)* dirancang dan dibangun untuk mempermudah pelaksanaan tes minat SDS (*Self-Directed Search*) dan membantu skoring dari hasil pengerjaan. Aplikasi *HoTest (Holland Advanced-Study Test)* terdiri atas menu utama yang di bagi ke dalam dua hak akses yaitu *testee* (peserta tes) dan *test-taker*. Menu yang diperuntukkan bagi *testee* (peserta tes) terdiri atas menu untuk pelaksanaan tes yang merupakan menu utama bagi *testee* (peserta tes), menu prosedur pengerjaan berisi tata cara pengerjaan tes, dan menu “About test” yang berisi informasi *developer* dan informasi terkait tes. Menggunakan aplikasi *HoTest*, *testee* (peserta tes) dapat melaksanakan tes secara langsung secara komputerisasi tanpa menggunakan *paper-and-pencil*. Menu untuk *test-taker* merupakan menu yang dapat digunakan untuk input nilai apabila tes dilaksanakan secara manual karena kurangnya ketersediaan komputer, dan menu cetak hasil yang digunakan untuk edit data peserta dan mencetak laporan hasil skoring dari tes yang sudah dilaksanakan.

2. Berdasarkan pengujian dengan hasil berupa *review* dari penguji di Laboratorium Universitas Negeri Semarang dan analisis sistem menggunakan metode

PIECES, dapat disimpulkan bahwa aplikasi *HoTest (Holland Advanced-Study Test)* memiliki fungsionalitas yang cukup baik dengan informasi yang dihasilkan merupakan informasi yang akurat, tepat pada waktunya, dan relevan. Hasil uji kinerja juga menunjukkan bahwa aplikasi *HoTest* memiliki kinerja yang baik dengan *response time* yang ditunjukkan kurang dari 1 menit. Hasil *review* dari Laboratorium Psikologi Universitas Negeri Semarang memberikan hasil bahwa aplikasi *HoTest* memiliki validitas skoring yang baik dengan hasil penghitungan yang disajikan memiliki hasil yang sama dengan penghitungan secara manual. Selain itu, saran studi dapat otomatis dimunculkan sehingga akan memudahkan *test-taker* dalam mempercepat pencarian saran studi berdasarkan kamus Holland yang ada. Dengan adanya aplikasi *HoTest (Holland Advanced-Study Test)* ini diharapkan dapat meminimalisir penggunaan biaya dalam penyediaan soal dan lembar jawab.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran bagi pengembangan aplikasi selanjutnya diantaranya adalah :

1. Aplikasi perlu dikembangkan tidak hanya untuk membantu proses skoring dan menghasilkan laporan skoring saja namun hingga pembuatan laporan yang dapat langsung diserahkan ke *testee*.
2. Dalam mengimplementasikan aplikasi tersebut, dibutuhkan penyesuaian lebih lanjut karena menggantikan sistem lama dengan sistem yang baru membutuhkan persiapan yang tidak mudah.

3. Seiring dengan berkembangnya teknologi yang ada saat ini, perlu dikembangkan aplikasi untuk jenis-jenis tes yang lain sehingga dapat memudahkan proses pelaksanaan tes.
4. Penelitian lebih lanjut terkait pada pengembangan aplikasi dalam bidang psikologi dalam hal ini dalam kegiatan tes, perlu lebih ditekankan dalam hal keamanan yang masih harus ditingkatkan.

DAFTAR PUSTAKA

- Anastasi, Anna., Susana Urbina. 2007. *Tes Psikologi*. Edisi Ketujuh. (Terjemahan Robertus Hariono S. Imam. Jakarta: Indeks).
- Atkinson, Rita L., Richard C. Atkinson, Ernest R . Hilgard. 1997. *Pengantar Psikologi*. Edisi Kedelapan. (Terjemahan Agus Dharma dan Michael Adryanto. Jakarta: Erlangga).
- Barak, Azy, dan Nicole English. 2002. *Prospect and Limitations of Psychological Testing on the Internet*. *Journal of Technology in Human Services* 19(2-3): 65-89.
- Bima, Ifnu. 2011. *Java Desktop: Aplikasi POS Berarsitektur Three Tier Menggunakan Swing, Hibernate, dan Spring*. Jakarta: Artivisi Intermedia.
- Brookshear, J. Glenn. 2003. *Computer Science : Suatu Pengantar*. (Terjemahan Irzan Hardiansyah. Jakarta: Erlangga).
- Budiman, Irfan. 2009. *Pembuatan Aplikasi Tes Kepribadian Berbasis Sistem Pakar Menggunakan Visual Studio.Net 2008*. Universitas Gunadarma. http://www.gunadarma.ac.id/library/articles/graduate/computer-science/2009/Artikel_10104875.pdf. 13 Januari 2015.
- Budiyati, Ika. 2009. *Analisis dan Perancangan Sistem Pengolahan Data Absensi dan Lembur Karyawan pada Kopegtel Dinasti Jakarta Timur dengan Pendekatan Berorientasi Objek Menggunakan UML*. http://www.gunadarma.ac.id/library/articles/graduate/computer-science/2009/Artikel_11105847.pdf. 22 Februari 2015 (09:43).
- Desanti, R.I., Suryasari, Greecia P.G. 2010. *Analisa Proses Bisnis Sistem Penggajian dan Pinjaman Pegawai Studi Kasus Perusahaan Industri Kertas PT UNIPA DAYA*. *Seminar Nasional Informatikai 2010 (SemnasIF 2010)*. UPN "Veteran" Yogyakarta. 157-166.
- Fahrurozi, Imam., Azhari SN. 2008. *Proses Pemodelan Software Dengan Metode Waterfall dan Extreme Programming: Studi Perbandingan*. Universitas Gadjah Mada. http://jurnal.stmikelrahma.ac.id/assets/file/Imam%20Fahrurrozi,%20Azhari%20SN_stmikelrahma.pdf. 13 Februari 2015 (13:09).
- Fauzi, Akhmad. 2008. *Pengantar Teknologi Informasi*. Yogyakarta: Graha Ilmu.
- Febriana, Ely. 2013. *Pengembangan Aplikasi Test Karir Holland (SDS) dengan Menggunakan PHP dan MYSQL*.

http://repository.uksw.edu/bitstream/123456789/3374/3/T1_132009031_BAB%20II.pdf. 02 Maret 2015 (22:35).

- Fitriani, Wahidah. 2012. *Bias Budaya Dalam Tes Psikologi Ditinjau dari Aspek Testee Dan Alternatif Solusinya*. *Jurnal Ta'dib* 15(2): 189-198.
- Foster, David F. 2010. *Worldwide Tetsing and Test Security Issues: Ethical Challenges and Solutions*. *Journal Ethics & Behaviour* 20(3-4): 207-228.
- Fowler, Martin dan Kendall Scott. 1999. *UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language*. Second Edition. Addison Wesley.
- Haviluddin. 2011. *Memahami Penggunaan UML (Unified Modelling Language)*. *Jurnal Informatika Mulawarman* 16(1): 1-15.
- Indrajani. 2009. *Sistem Basis Data dalam Paket Five in One*. Jakarta: PT. Elex Media Komputindo.
- Lubis, Fitriani Yustikasari. 2008. *Pengembangan Alat Ukur Minat untuk Pengembangan Karir pada Lulusan Sekolah Menengah Atas*. *Laporan Penelitian*. Universitas Padjadjaran. Bandung.
- Nur'aeni. 2012. *Tes Psikologi : Tes Inteligensi dan Tes Bakat*. Universitas Muhammadiyah Purwokerto Press. Purwokerto.
- Pressman, Roger S. 2005. *Software Engineering : A Practitioner's Approach*. Sixth Edition. New York: McGraw-Hill, Inc.
- _____.1997. *Software Engineering : A Practitioner's Approach*. McGraw-Hill, Inc. New York (Terjemahan LN Harnaningrum. Yogyakarta: ANDI).
- Puspitasari, Fitriana Eki. 2015. *Rancang Bangun Sistem Informasi Sertifikasi Tanah Kantor Pertanahan Kabupaten Kebumen*. *Skripsi*. Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang.
- Putra, A.R. Pradana. 2014. *Implementasi Hack and Slash untuk Peningkatan Kualitas Battle System pada Game RPG*. *Skripsi*. Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang.
- Respati, Ragil Bayu. 2013. *Persepsi Pengguna Terhadap Kinerja Online Public Access Catalog (OPAC) Badan Perpustakaan dan Kearsipan Propinsi Jawa Timur (Studi Deskriptif Analisa Sistem Informasi OPAC Badan*

- Perpustakaan dan Kearsipan Propinsi Jawa Timur dengan Menggunakan Analisis Pieces*). Universitas Airlangga. <http://journal.unair.ac.id/filerPDF/jurnal%20ragil.pdf>. 06 Mei 2015 (15:25).
- Rusdianto, T. Ilham. 2012. *Analisis dan Perancangan Aplikasi Desktop Untuk Pengolahan Data Siswa pada Manajemen Prakerin SMK Yosonegoro Magetan*. Naskah publikasi. Jurusan Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer (AMIKOM) Yogyakarta.
- Setiyadi, Didik. 2010. *Bahasa Basis Data (Database Language)*. STMIK Bani Saleh. <https://helpmeups.files.wordpress.com/2012/07/modul-dewa89s-bab-3-bahasa-basis-data.pdf>. 21 Februari 2015 (08:15).
- Sianipar, R.H. 2013. *Teori dan Implementasi Java*. Bandung: Informatika.
- Sommerville, Ian. 2003. *Software Engineering (Rekayasa Perangkat Lunak)*. Sixth Edition. (Terjemahan Yuhilza Hanum. Jakarta: Erlangga).
- Sugiyono. 2011. *Metode Penelitian Kuantitatif, Kualitatif dan R & D*. Bandung: Alfabeta.
- Sumarta, T., B. Siswoyo., N. Juhana. 2004. *Perancangan Model Berorientasi Objek Menggunakan Unified Modelling Language (UML) Studi Kasus Pengolahan Parkir pada PT. TRIKARYA ABADI*. Universitas Komputer Indonesia. http://rio_wirawan.staff.gunadarma.ac.id/Downloads/files/29293/jbptunikompp-gdl-s1-2004-tsumartani-792-JURNAL.pdf. 05 Februari 2015 (19:43).
- Tasmawati, Mila. 2008. *Aplikasi Konversi Regular Grammar menjadi Ekspresi Regular dengan Menggunakan Bahasa Pemrograman Java*. Universitas Gunadarma. http://www.gunadarma.ac.id/library/articles/graduate/industrial-technology/2008/Artikel_50404875.pdf. 01 Februari 2015 (14:01).
- Widiastuti. 2008. *Analisis dan Perancangan Sistem Pengadaan Barang pada PT. Indo Mitra Pratama*. Universitas Gunadarma. http://www.gunadarma.ac.id/library/articles/graduate/industrial-technology/2008/Artikel_12103034.pdf. 06 Mei 2015 (14:25).

LAMPIRAN

Lampiran 1. Surat Usulan Topik Skripsi

Formulir Usulan Topik Skripsi
FM-1-AKD-24/rev.00
UNIVERSITAS NEGERI SEMARANG


Usulan topik skripsi ini diajukan oleh:

Nama : INAYAH KURNIAWATI
NIM : 5302411169
Jurusan : Teknik Elektro
Program Studi : Pendidikan Teknik Informatika dan Komputer, S1
Topik : Rancang Bangun Aplikasi Psychological Testing



DR. JAYONO, M.T.
NIP. 196503161985031001

Semarang, 12 Maret 2015
Yang mengajukan,


INAYAH KURNIAWATI
NIM. 5302411169



Lampiran 2. Surat Usulan Pembimbing Skripsi



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI SEMARANG
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
 Gedung E6 lt 2, Kampus Sekaran, Gunungpati, Semarang 50229
 Telepon: 8508104
 Laman: www.te.unnes.ac.id, surel:

Nomor : 259/TE/0/2015
 Lamp. :
 Hal : Usulan Pembimbing

Yth. Dekan Fakultas Teknik
 Universitas Negeri Semarang

Merujuk Keputusan Rektor Unnes Nomor 164/O/2004 tentang Pedoman Penyusunan Skripsi Mahasiswa Program S1 pasal 7 mengenai penentuan pembimbing, dengan ini saya usulkan

Nama : ANGGRAINI MULWINDA, S.T., M.Eng.
 NIP : 197812262005012002
 Pangkat/Golongan : III/A
 Jabatan Akademik : Asisten Ahli
 Sebagai Dosen Pembimbing

Dalam penyusunan Skripsi/Tugas Akhir untuk mahasiswa

Nama : INAYAH KURNIAWATI
 NIM : 5302411169
 Program Studi : Pendidikan Teknik Informatika dan Komputer, S1
 Topik : Rancang Bangun Aplikasi Psychological Testing

Untuk itu, mohon diterbitkan surat penetapannya.





Sejarah 22 Maret 2015
 Ketua Jurusan
 Drs. H. ... M.T.
 NIP. 195003181985031001

Lampiran 3. Surat Keputusan Dosen Pembimbing



**KEPUTUSAN
DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG**
Nomor: *419/FT-UNNES/2015*
Tentang
**PENETAPAN DOSEN PEMBIMBING SKRIPSI/TUGAS AKHIR SEMESTER
GASAL/GENAP
TAHUN AKADEMIK 2014/2015**

Menimbang : Bahwa untuk memperlancar mahasiswa Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Fakultas Teknik membuat Skripsi/Tugas Akhir, maka perlu menetapkan Dosen-dosen Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Fakultas Teknik UNNES untuk menjadi pembimbing

Mengingat : 1. Undang-undang No.20 Tahun 2003 tentang Sistem Pendidikan Nasional (Tambahan Lembaran Negara RI No.4301, penjelasan atas Lembaran Negara RI Tahun 2003, Nomor 78)
2. Peraturan Rektor No. 21 Tahun 2011 tentang Sistem Informasi Skripsi UNNES
3. SK. Rektor UNNES No. 164/O/2004 tentang Pedoman penyusunan Skripsi/Tugas Akhir Mahasiswa Strata Satu (S1) UNNES.
4. SK Rektor UNNES No.162/O/2004 tentang penyelenggaraan Pendidikan UNNES.

Menimbang : Usulan Ketua Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Tanggal 27 Maret 2015

MEMUTUSKAN

Menetapkan
PERTAMA

Menunjuk dan menugaskan kepada:

Nama : ANGGRAINI MULWINDA, S.T., M.Eng
NIP : 197812262005012002
Pangkat/Golongan : III/A
Jabatan Akademik : Asisten Ahli
Sebagai Pembimbing

Untuk membimbing mahasiswa penyusun skripsi/Tugas Akhir :

Nama : INAYAH KURNIAWATI
NIM : 5302411169
Jurusan/Prodi : Teknik Elektro/Pend. Teknik Informatika dan Komputer
Topik : Rancang Bangun Aplikasi Psychological Testing

KEDUA : Keputusan ini mulai berlaku sejak tanggal ditetapkan.

DITETAPKAN DI : SEMARANG
PADA TANGGAL : 30 Maret 2015



Dr. Muhammad Harlanu, M.Pd
NIP. 196602151991021001

Tembusan
1. Pembantu Dekan Bidang Akademik
2. Ketua Jurusan
3. Petinggal

UNNES
5302411169
FM-03-AKD-24/Rev. 00



Lampiran 4. Surat Penelitian


KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI SEMARANG
FAKULTAS TEKNIK
 Gedung E1Kampus Sekaran Gunungpati Semarang 50229
 Telepon/Fax (024) 8508101 – 8508009
 Laman : <http://www.ft.unnes.ac.id>, email: ft_unnes@yahoo.com

Nomor : *2978* /UN37.1.5/DT/2015
 Lampiran : -
 Hal : **Permohonan Izin Penelitian**

Yth : Kepala Laboratorium Psikologi, Jurusan Psikologi
 Fakultas Ilmu Pendidikan UNNES Semarang, Gd.A1

Dengan ini kami mohonkan ijin Penelitian di Laboratorium Psikologi Unnes, dalam rangka
 Penyusunan Skripsi mahasiswa kami :

N a m a : Inayah Kurniawati
 N I M : 5302411169
 Program Studi : SI PTIK
 Jurusan : Teknik Elektro
 Judul Skripsi : Rancang Bangun Aplikasi Tes Psikologi “ Hotest “ (Holland Career Test) dengan
 Menerapkan Alat Ukur SDS (*Self- Directed Search*) untuk Mengukur Minst Ksrir
 Individu

Waktu Penelitian : Mulai tanggal 15 April 2015 s/d Selesai

Atas bantuannya kami ucapkan terima kasih

Semarang, 14 April 2015
 A.n Dekan
 P. Asmanto Dekan Bidang Akademik


Joko Adi Widodo, M.T
 NIP. 195909271986011001



Tembusan
 1. Rektor Universitas Negeri Semarang
 2. Ketua Jurusan TE

Lampiran 5. Surat Persetujuan Kerahasiaan



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS NEGERI SEMARANG
 FAKULTAS ILMU PENDIDIKAN
 JURUSAN PSIKOLOGI
 Gedung A1, Kampus Sekaran, Gunungpati, Semarang 50229
 Telepon/Fax : (024) 8508022
 Laman : <http://psikologi.unnes.ac.id>

SURAT PERSETUJUAN

Yang bertanda tangan di bawah ini :

Nama : Inayah Kurniawati
 NIM : 5302411169
 Program Studi : Pendidikan Teknik Informatika dan Komputer, S1
 Jurusan : Teknik Elektro

Menyetujui dan bersedia mengikuti tata tertib yang ada di Laboratorium Psikologi Universitas Negeri Semarang dengan bersedia tidak menyebarluaskan alat ataupun melakukan penggandaan terhadap alat ukur tes psikologi serta menjaga kerahasiaan alat ukur tes tersebut dengan penuh tanggung jawab.

Mengetahui,
 Kepala Laboratorium Psikologi UNNES,

Liffiah S. Psi, M.Si
 NIP. 196904151997032002

Mahasiswa,

Inayah Kurniawati
 NIM. 5302411169

Lampiran 6. Surat Review Hasil Pengujian

REVIEW APLIKASI HOTEST

Nama Aplikasi : HOTest (*Holland Career Test*)

Nama Penguji : Lifiyah S.Psi, M.Si
 NIP : 196904151997032002

Pangkat/ Gol. Ruang : III/c-Penata
 Jabatan : Kepala Laboratorium (Kepala Laboratorium Jurusan Psikologi FIP)

Developer : Inayah Kurniawati
 NIM : 5302411169
 Program Studi : Pendidikan Teknik Informatika dan Komputer, S1
 Jurusan : Teknik Elektro

Review Hasil Pengujian

1. Aplikasi HoTest dapat membantu untuk mempercepat skoring, mengurangi tingkat kesalahan penghitungan.
2. Hasil skoring (kode ringkas) sudah sesuai (valid) dengan penghitungan secara manual.
3. Aplikasi HoTest membantu pemeriksa untuk melakukan verifikasi antara minat yang diekspresikan oleh subjek dengan minat yang ada di kamus.
4. Aplikasi HoTest dapat digunakan dalam pelaksanaan tes minat berbasis komputer.

Demikian review dari hasil pengujian aplikasi *HoTest* yang dapat kami sampaikan, agar dapat dipergunakan sebagaimana mestinya.

Semarang, 15 Mei 2015
 Kepala Laboratorium Psikologi UNNES



Lifiyah S.Psi, M.Si
 NIP. 196904151997032002

Lampiran 7. Surat Selesai Penelitian dan Pengujian



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI SEMARANG
FAKULTAS ILMU PENDIDIKAN
JURUSAN PSIKOLOGI

Gedung A1, Kampus Sekaran, Gunungpati, Semarang 50229

Telepon/Fax : (024) 8508022

Laman : <http://psikologi.unnes.ac.id>

SURAT KETERANGAN

Nomor : **42/UN37.1.1.5/KM/2015**

Yang bertanda tangan di bawah ini :

Nama : Liftiah S.Psi, M.Si
NIP : 196904151997032002
Pangkat/ Gol. Ruang : III/c-Penata
Jabatan : Kepala Laboratorium (Kepala Laboratorium Jurusan Psikologi FIP)

Dengan ini menyatakan bahwa :

Nama : Inayah Kurniawati
NIM : 5302411169
Program Studi : Pendidikan Teknik Informatika dan Komputer, S1
Jurusan : Teknik Elektro

Telah melaksanakan penelitian dan pengujian aplikasi HoTest (*Holland Career Test*) di Laboratorium Psikologi Universitas Negeri Semarang terhitung mulai tanggal 05 April 2015 s.d 15 Mei 2015. Dari pengujian aplikasi yang telah dilaksanakan dapat kami simpulkan bahwa:

1. Aplikasi HoTest dapat membantu untuk mempercepat skoring, mengurangi tingkat kesalahan penghitungan.
2. Hasil skoring (kode ringkas) sudah sesuai (valid) dengan penghitungan secara manual.
3. Aplikasi HoTest membantu pemeriksa untuk melakukan verifikasi antara minat yang diekspresikan oleh subjek dengan minat yang ada di kamus.
4. Aplikasi HoTest dapat digunakan dalam pelaksanaan tes minat berbasis komputer.

Demikian surat keterangan ini kami buat dengan sesungguhnya, agar dapat dipergunakan sebagaimana mestinya.

Semarang, 15 Mei 2015

Kepala Laboratorium Psikologi UNNES



NIP. 196904151997032002

Lampiran 8. Dokumentasi

