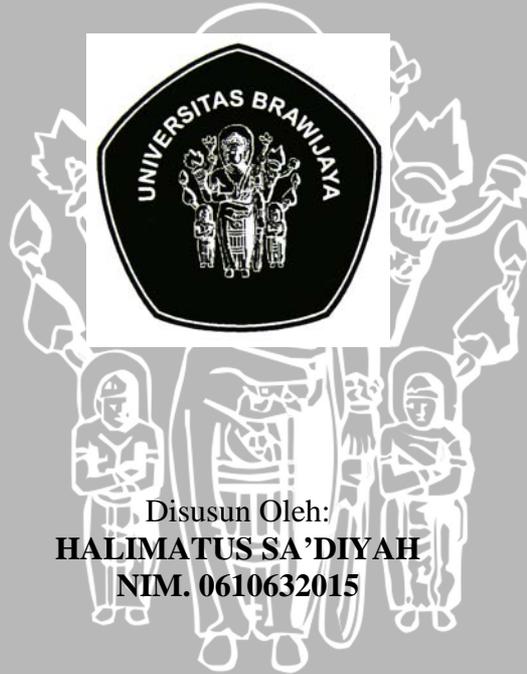


**PENERAPAN SMARTY TEMPLATE ENGINE
DALAM PERANCANGAN DAN PEMBUATAN
SOFTWARE PSIKOTES ONLINE**

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun Oleh:
HALIMATUS SA'DIYAH
NIM. 0610632015

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2009**

**PENERAPAN SMARTY TEMPLATE ENGINE
DALAM PERANCANGAN DAN PEMBUATAN
SOFTWARE PSIKOTES ONLINE**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

UNIVERSITAS BRAWIJAYA



Disusun Oleh:
HALIMATUS SA'DIYAH
NIM. 0610632015

Telah diperiksa dan disetujui oleh :

DOSEN PEMBIMBING

Herman Tolle, ST.,MT.
NIP. 132 283 206

Amir Hasan Ramli, Drs., Psy., M.Si.
NIP. 130 704 066

LEMBAR PENGESAHAN

**PENERAPAN SMARTY TEMPLATE ENGINE DALAM PERANCANGAN
DAN PEMBUATAN SOFTWARE PSIKOTES ONLINE**

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh:

HALIMATUS SA'DIYAH

NIM. 0610632015

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 09 Pebruari 2009

Majelis Penguji:

Ir. Muhammad Aswin, MT
NIP. 131 879 045

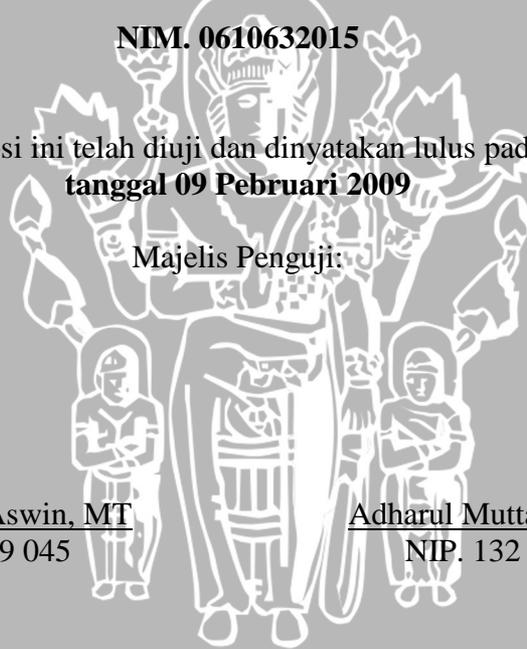
Adharul Muttaqin, ST., MT
NIP. 132 311 886

Ir. Heru Nurwarsito, M.Kom
NIP. 131 879 033

Mengetahui,
Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, M.Kom
NIP. 131 879 033

UNIVERSITAS BRAWIJAYA



ABSTRAK

HALIMATUS SA'DIYAH. 2009. : Penerapan *Smarty Template Engine* dalam Perancangan dan Pembuatan *Software* Psikotes *Online*. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing : Herman Tolle,ST.,MT dan Amir Hasan Ramli,Drs.,Psy.,M.Si.

Psikologi merupakan salah satu bidang yang juga memperoleh dampak akibat berkembangnya internet. Banyak situs – situs yang menawarkan adanya uji kepribadian secara *online*. Dengan adanya uji kepribadian secara *online* ini akan membantu peselancar untuk mengetahui kecerendungan dari sifat masing – masing. Para konsultan psikologi dapat memanfaatkan teknologi yang sama sehingga ketika melakukan psikotes terhadap pelamar dari instansi yang menjadi clientnya dapat mengetahui hasilnya lebih cepat dan dapat memodifikasi soal – soal psikotes kapan saja dengan cepat.

Proses pengembangan perangkat lunak harus melalui suatu proses rekayasa perangkat lunak (RPL), begitu pula dengan perangkat lunak psikotes *online* yang ditujukan bagi para jasa konsultan psikologi. Pengembangan *software* ini menggunakan paradigma *object oriented* (OO) dengan model proses *waterfall* dan pemodelan UML. Implementasi sistem ini akan menggunakan *PHP* (*Hypertext Preprocessor*), *Smarty template engine*, dan database *MySQL*. Penerapan *template engine* dilakukan karena diperlukan adanya pemisahan *template code* dan *logic code*. *Smarty* dipilih karena merupakan *template engine* yang kinerjanya paling cepat, akurat, dan efisien. *PHP* dipilih karena dapat membuat website menjadi lebih dinamis dan interaktif terhadap pengunjung. *MySQL* dipilih karena database ini cocok untuk sistem yang membutuhkan database besar, selain itu juga *MySQL* adalah suatu bahasa berbasis *SQL* yang dikembangkan secara *free* dan *open source*

Kualitas aplikasi didapat dengan melakukan pengujian unit, integrasi dan validasi pada sistem psikotes *online*. Hasil pengujian menunjukkan bahwa sistem yang dibangun adalah valid sesuai dengan kebutuhan dari *user*.

Kata Kunci : *Object Oriented*, *template engine*, *smarty*, Rekayasa Perangkat Lunak (RPL), psikotes, Sistem Informasi

PENGANTAR

Puji syukur penulis panjatkan ke hadapan Allah SWT karena berkat rahmat dan hidayahNya-lah penulis dapat menyelesaikan Tugas Akhir yang berjudul: **“Penerapan Smarty Template Engine Dalam Perancangan Dan Pembuatan Software Psikotes Online”**. Hanya kepada-Nya kita menyembah dan memohon. Teriring doa keselamatan untuk Rasulullah Muhammad SAW, keluarga, sahabat serta seluruh umatnya. Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Konsentrasi Teknik Informatika dan Komputer Fakultas Teknik Universitas Brawijaya Malang.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar besarnya kepada semua pihak yang telah memberikan bantuan baik material maupun non-material sehingga Tugas Akhir ini dapat terselesaikan dengan baik dan tepat pada waktunya. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada:

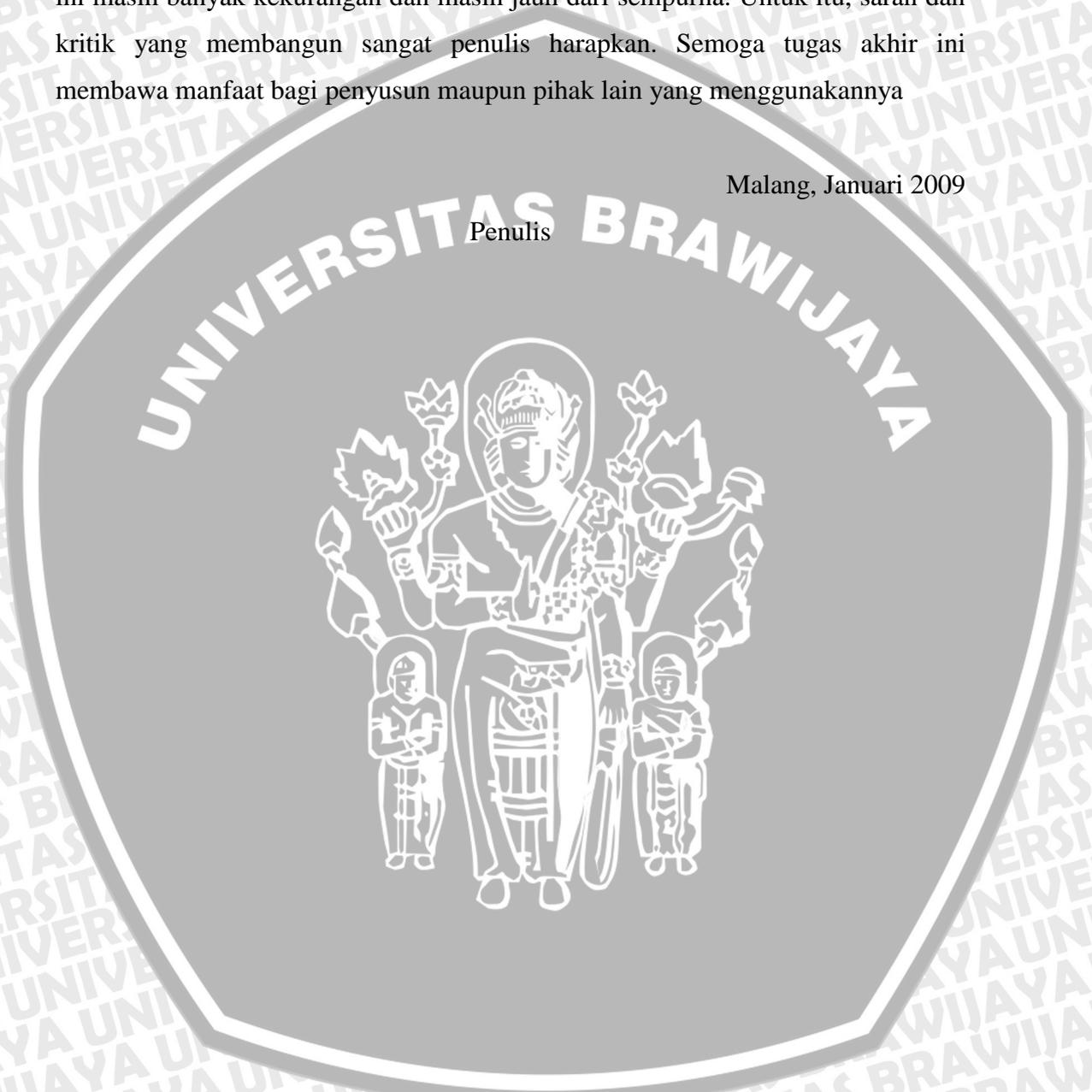
1. Kedua Orang Tua penulis dan seluruh keluarga yang senantiasa tiada henti hentinya memberikan do'a demi terselesainya tugas akhir ini.
2. Bapak Ir. Heru Nurwarsito, M.Kom dan Bapak Rudy Yuwono, ST., M.Sc. selaku Ketua dan Sekretaris Jurusan Teknik Elektro serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya
3. Bapak Herman Tolle, ST., MT. Dan Bapak Amir Hasan Ramli, Drs.,Psy.,M.Si. selaku pembimbing penulis dalam menyelesaikan tugas akhir ini.
4. M. Desdrianton Islamy TE UB '01 terima kasih atas bantuan dan *supportnya*.
5. Semua Asisten, Ka. Lab, serta Laboran dari Laboratorium Sistem Informasi yang telah memberikan banyak bantuan dan dukungan dalam menyelesaikan tugas akhir ini.
6. Teman-teman Ijen Perkasa terima kasih atas segala bantuannya
7. Teman – teman kertosari 13 khususnya Ria, Icha, dan Dana terima kasih atas bantuannya.

8. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesainya tugas akhir ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya

Malang, Januari 2009

Penulis



DAFTAR ISI

	Halaman
ABSTRAK	i
PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xxii
I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan	4
1.5. Manfaat	4
1.6 Sistematika Penulisan	4
II TINJAUAN PUSTAKA	
2.1. Kajian Pustaka	6
2.2. Dasar Teori	8
2.2.1. Teori Dasar Psikotes	9
2.2.1.1. Komponen Psikotes	9
2.2.2. Teori Dasar RPL	10
2.2.2.1. Proses-proses RPL	11
2.2.2.2. Model Proses Perangkat Lunak	12
2.2.2.3. Paradigma RPL	13
2.2.3. Unified Modeling Language	14
2.2.3.1. Diagram Use Case	17
2.2.3.2. Diagram Klas	24
2.2.3.3. Diagram Sequence	31
2.2.4. Pengujian (Testing)	34
2.2.4.1. Level Pengujian	34
2.2.4.2. Jenis Pengujian	35
2.2.5. PHP (Hypertext Preprocessor)	37
2.2.6. Template Engine	38



2.2.6.1. Smarty	39
2.2.7. Basis Data.....	39
2.2.7.1. Diagram Entity Relationship.....	39
2.2.7.2. MySQL.....	42
III METODE PENELITIAN	
3.1. Studi Literatur.....	44
3.2. Analisis dan Perancangan Sistem Aplikasi.....	44
3.3. Implementasi Sistem Aplikasi.....	45
3.4. Pengujian Sistem Aplikasi	45
3.5. Pengambilan Kesimpulan dan Saran.....	45
IV PERANCANGAN	
4.1. Analisis Kebutuhan	47
4.1.1. Analisis Sistem yang Telah Ada	48
4.1.2. Analisis Sistem yang Akan Dirancang.....	49
4.1.2.1. Daftar Kebutuhan.....	50
4.1.2.2. Use Case Diagram.....	59
4.1.2.2.1. Use Case Diagram untuk Modul Pendukung Sistem Psikotes Online.....	59
4.1.2.2.2. Use Case Diagram untuk Modul Pengaturan Instansi Client.....	68
4.1.2.2.3. Use Case Diagram untuk Modul Pengaturan Soal Tes.....	75
4.1.2.2.4. Use Case Diagram untuk Modul Pengaturan Penyelenggaraan Tes.....	103
4.1.2.2.5. Use Case Diagram untuk Modul Pengaturan Peserta Tes.....	109
4.1.2.2.6. Use Case Diagram untuk Modul Perhitungan Nilai dan Hasil Penilaian.....	122
4.1.2.3. Class Diagram	126
4.1.2.3.1. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.administrator.....	126
4.1.2.3.2. Class Diagram untuk Class-class dari Paket	

psikotes.LogicCode.dataSoal	127
4.1.2.3.3. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.hasilTes	127
4.1.2.3.4. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.home	128
4.1.2.3.5. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.instansiClient	128
4.1.2.3.6. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.pelaksanaanTes	129
4.1.2.3.7. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.pengaturanSoal	129
4.1.2.3.8. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.pesertaLokasiTes	130
4.1.2.3.9. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.pesertaTes	130
4.1.2.3.10. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.petugas	131
4.1.2.3.11. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.pilihanJawaban	131
4.1.2.3.12. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.subKategori	132
4.1.2.3.13. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.urutanKategoriSoal .	133
4.1.2.3.14. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.client	133
4.1.2.3.15. Class Diagram untuk Class-class dari Paket psikotes.LogicCode.client.soal	134
4.1.2.4. Sequence Diagram	134
4.1.2.4.1. Sequence Diagram untuk method setSoalList(\$db,\$smarty,\$ps)	135
4.1.2.4.2. Sequence Diagram untuk method setJawaban(\$db,\$smarty,\$ps)	135

4.1.2.4.3. Sequence Diagram untuk method simpanJawaban(\$db,\$smarty,\$ps).....	136
4.2. Perancangan Basis Data	136
4.2.1. Data Object Description untuk Tabel tbl_petugas	139
4.2.2. Data Object Description untuk Tabel tbl_instansi_client.....	139
4.2.3. Data Object Description untuk Tabel tbl_kategori_soal	140
4.2.4. Data Object Description untuk Tabel Tbl_urutan_kategori_soal.....	140
4.2.5. Data Object Description untuk Tabel tbl_lokasi_tes.....	141
4.2.6. Data Object Description untuk Tabel tbl_soal	141
4.2.7. Data Object Description untuk Tabel tbl_peserta	141
4.2.8. Data Object Description untuk Tabel tbl_lokasi_peserta.....	142
4.2.9. Data Object Description untuk Tabel tbl_jawaban	142
4.2.10. Data Object Description untuk Tabel tbl_nilai.....	143
4.2.11. Data Object Description untuk Tabel tbl_sub_kategori	143
4.2.12. Data Object Description untuk Tabel tbl_pilihan_jawaban	144
V IMPLEMENTASI	
5.1. Spesifikasi Sistem.....	145
5.1.1. Spesifikasi Perangkat Keras	145
5.1.2. Spesifikasi Perangkat Lunak	146
5.2. Implementasi Konfigurasi Smarty Template Engine	146
5.2.1. Implementasi Penggunaan Smarty Template Engine.....	147
5.3. Implementasi Algoritma.....	150
5.3.1. Pseudocode method setSoalList(\$db,\$smarty,\$ps) dari Klas soal	152
5.3.2. Implementasi Template Code Penggunaan Smarty untuk Pseudocode method setSoalList	153
5.3.3. Pseudocode method setJawaban(\$db,\$smarty,\$ps) dari Klas soal	152
5.3.4. Implementasi Template Code Penggunaan Smarty untuk Pseudocode method setJawaban.....	154
5.3.5. Pseudocode method simpanJawaban(\$db,\$smarty,\$ps) dari	



Klas kategoriList	154
5.4. Implementasi Antarmuka Kebutuhan Fungsional.....	155
5.4.1. Implementasi Antarmuka Login.....	155
5.4.2. Implementasi Antarmuka Home	156
5.4.3. Implementasi Antarmuka Petugas.....	156
5.4.4. Implementasi Antarmuka Tambah Petugas.....	157
5.4.5. Implementasi Antarmuka Edit Petugas	158
5.4.6. Implementasi Antarmuka Detail Petugas	159
5.4.7. Implementasi Antarmuka Hapus Petugas.....	159
5.4.8. Implementasi Antarmuka Cari Petugas.....	160
5.4.9. Implementasi Antarmuka Instansi Client.....	161
5.4.10. Implementasi Antarmuka Tambah Instansi Client.....	161
5.4.11. Implementasi Antarmuka Edit Instansi Client	162
5.4.12. Implementasi Antarmuka Detail Instansi Client	163
5.4.13. Implementasi Antarmuka Hapus Instansi Client.....	163
5.4.14. Implementasi Antarmuka Cari Instansi Client	164
5.4.15. Implementasi Antarmuka Kategori Soal	164
5.4.16. Implementasi Antarmuka Tambah Kategori Soal.....	165
5.4.17. Implementasi Antarmuka Edit Kategori Soal	165
5.4.18. Implementasi Antarmuka Detail Kategori Soal	166
5.4.19. Implementasi Antarmuka Cari Kategori Soal	166
5.4.20. Implementasi Antarmuka Halaman Sub Kategori.....	167
5.4.21. Implementasi Antarmuka Tambah Sub Kategori Soal.....	168
5.4.22. Implementasi Antarmuka Edit Sub Kategori Soal	169
5.4.23. Implementasi Antarmuka Detail Sub Kategori	169
5.4.24. Implementasi Antarmuka Hapus Sub Kategori.....	169
5.4.25. Implementasi Antarmuka Cari Sub Kategori	170
5.4.26. Implementasi Antarmuka Soal	171
5.4.27. Implementasi Antarmuka Tambah Soal.....	172
5.4.28. Implementasi Antarmuka Edit Soal	173
5.4.29. Implementasi Antarmuka Detail Soal	173
5.4.30. Implementasi Antarmuka Hapus Soal.....	174



5.4.31. Implementasi Antarmuka Cari Soal	174
5.4.32. Implementasi Antarmuka Pilihan Jawaban	175
5.4.33. Implementasi Antarmuka Tambah Pilihan Jawaban	176
5.4.34. Implementasi Antarmuka Edit Pilihan Jawaban	176
5.4.35. Implementasi Antarmuka Detail Pilihan Jawaban	177
5.4.36. Implementasi Antarmuka Hapus Pilihan Jawaban.....	177
5.4.37. Implementasi Antarmuka Urutan Kategori Soal.....	178
5.4.38. Implementasi Antarmuka Tambah Urutan Kategori Soal..	178
5.4.39. Implementasi Antarmuka Edit Urutan Kategori Soal	179
5.4.40. Implementasi Antarmuka Detail Urutan Kategori Soal	179
5.4.41. Implementasi Antarmuka Hapus Urutan Kategori Soal.....	180
5.4.42. Implementasi Antarmuka Cari Urutan Kategori Soal	180
5.4.43. Implementasi Antarmuka Pelaksanaan Tes.....	181
5.4.44. Implementasi Antarmuka Tambah Pelaksanaan Tes.....	182
5.4.45. Implementasi Antarmuka Edit Pelaksanaan Tes	182
5.4.46. Implementasi Antarmuka Detail Pelaksanaan Tes	183
5.4.47. Implementasi Antarmuka Hapus Pelaksanaan Tes	183
5.4.48. Implementasi Antarmuka Cari Pelaksanaan Tes.....	184
5.4.49. Implementasi Antarmuka Data Peserta Tes	185
5.4.50. Implementasi Antarmuka Tambah Peserta Tes.....	185
5.4.51. Implementasi Antarmuka Edit Peserta Tes	186
5.4.52. Implementasi Antarmuka Detail Peserta Tes	186
5.4.53. Implementasi Antarmuka Hapus Peserta Tes.....	187
5.4.54. Implementasi Antarmuka Cari Peserta Tes	187
5.4.55. Implementasi Antarmuka Data Peserta di Lokasi Tes	188
5.4.56. Implementasi Antarmuka Tambah Peserta di Lokasi Tes..	189
5.4.57. Implementasi Antarmuka Edit Peserta di Lokasi Tes	189
5.4.58. Implementasi Antarmuka Detail Peserta di Lokasi Tes	190
5.4.59. Implementasi Antarmuka Hapus Peserta di Lokasi Tes.....	190
5.4.60. Implementasi Antarmuka Cari Peserta di Lokasi Tes	191
5.4.61. Implementasi Antarmuka Hasil Tes	192
5.4.62. Implementasi Antarmuka Edit Hasil Tes	192

5.4.63. Implementasi Antarmuka Detail Hasil Tes	193
5.4.64. Implementasi Antarmuka Hapus Hasil Tes.....	193
5.4.65. Implementasi Antarmuka Cari Hasil Tes	194
5.4.66. Implementasi Antarmuka Login Peserta	195
5.4.67. Implementasi Antarmuka Tes Kategori Pertama	196
5.4.68. Implementasi Antarmuka Menjawab Soal Kategori Pertama	196
5.4.69. Implementasi Antarmuka Tes Kategori Kedua.....	197
5.4.70. Implementasi Antarmuka Menjawab Soal Kategori Kedua	197
5.4.71. Implementasi Antarmuka Peringatan Jika Menjawab Soal Melebihi Batas Waktu yang Ditentukan	198
5.4.72. Halaman Terakhir Peserta	198
VI PENGUJIAN	
6.1. Pengujian Unit.....	199
6.2. Pengujian Integrasi.....	201
6.3. Pengujian Validasi.....	205
6.3.1. Kasus Uji Validasi.....	205
6.3.1.1. Kasus Uji Login.....	206
6.3.1.2. Kasus Uji Logout.....	206
6.3.1.3. Kasus Uji Menambah Data Petugas	207
6.3.1.4. Kasus Uji Melihat Data Petugas.....	208
6.3.1.5. Kasus Uji Mengedit Data Petugas.....	208
6.3.1.6. Kasus Uji Menghapus Data Petugas	209
6.3.1.7. Kasus Uji Mencari Data Petugas.....	210
6.3.1.8. Kasus Uji Menambah Data Instansi Client	211
6.3.1.9. Kasus Uji Melihat Data Instansi Client.....	212
6.3.1.10. Kasus Uji Mengedit Data Instansi Client.....	212
6.3.1.11. Kasus Uji Menghapus Data Instansi Client.....	213
6.3.1.12. Kasus Uji Mencari Data Instansi Client.....	214
6.3.1.13. Kasus Uji Menambah Kategori Soal.....	215
6.3.1.14. Kasus Uji Melihat Kategori Soal.....	215



6.3.1.15. Kasus Uji Mengedit Kategori Soal.....	216
6.3.1.16. Kasus Uji Menghapus Kategori Soal	217
6.3.1.17. Kasus Uji Mencari Kategori Soal.....	217
6.3.1.18. Kasus Uji Menambah Urutan Kategori Soal	218
6.3.1.19. Kasus Uji Mengedit Urutan Kategori Soal.....	219
6.3.1.20. Kasus Uji Menghapus Urutan Kategori Soal	220
6.3.1.21. Kasus Uji Mencari Urutan Kategori Soal.....	220
6.3.1.22. Kasus Uji Melihat Urutan Kategori Soal	221
6.3.1.23. Kasus Uji Menambah Data Soal Psikotes	221
6.3.1.24. Kasus Uji Melihat Daftar Soal	222
6.3.1.25. Kasus Uji Mengedit Data Soal	222
6.3.1.26. Kasus Uji Menghapus Data Soal.....	223
6.3.1.27. Kasus Uji Mencari Data Soal	223
6.3.1.28. Kasus Uji Menjawab Data Soal.....	224
6.3.1.29. Kasus Uji Menambah Sub Kategori Soal.....	224
6.3.1.30. Kasus Uji Melihat Sub Kategori Soal	225
6.3.1.31. Kasus Uji Mengedit Sub Kategori Soal	226
6.3.1.32. Kasus Uji Menghapus Sub Kategori Soal	227
6.3.1.33. Kasus Uji Mencari Sub Kategori Soal	227
6.3.1.34. Kasus Uji Menambah Pilihan Jawaban Soal.....	228
6.3.1.35. Kasus Uji Melihat Pilihan Jawaban Soal	229
6.3.1.36. Kasus Uji Mengedit Pilihan Jawaban Soal.....	229
6.3.1.37. Kasus Uji Menghapus Pilihan Jawaban Soal	230
6.3.1.38. Kasus Uji Menambah Data Waktu dan Tempat Pelaksanaan Tes	231
6.3.1.39. Kasus Uji Melihat Data Waktu dan Tempat Pelaksanaan Tes	231
6.3.1.40. Kasus Uji Mengedit Data Waktu dan Tempat Pelaksanaan Tes	232
6.3.1.41. Kasus Uji Menghapus Data Waktu dan Tempat Pelaksanaan Tes	233
6.3.1.42. Kasus Uji Mencari Data Waktu dan Tempat	

	Pelaksanaan Tes	233
6.3.1.42.	Kasus Uji Menambah Data Peserta Tes	234
6.3.1.44.	Kasus Uji Melihat Data Peserta Tes	235
6.3.1.45.	Kasus Uji Mengedit Data Peserta Tes	235
6.3.1.46.	Kasus Uji Menghapus Data Peserta Tes	236
6.3.1.47.	Kasus Uji Mencari Data Peserta Tes	236
6.3.1.48.	Kasus Uji Menambah Data Peserta di Lokasi Tes	237
6.3.1.49.	Kasus Uji Melihat Data Peserta di Lokasi Tes	238
6.3.1.50.	Kasus Uji Mengedit Peserta di Lokasi Tes	238
6.3.1.51.	Kasus Uji Menghapus Peserta di Lokasi Tes	239
6.3.1.52.	Kasus Uji Mencari Peserta di Lokasi Tes	240
6.3.1.53.	Kasus Uji Mengenerate Password Login Peserta Tes	241
6.3.1.54.	Kasus Uji Menampilkan Nilai Peserta	241
6.3.1.55.	Kasus Uji Mencari Nilai Peserta	241
6.3.1.56.	Kasus Uji Menambah Penilaian Konsultan terhadap Nilai yang Diperoleh untuk Setiap Sub Kategori ...	242
6.3.1.57.	Kasus Uji Melihat Penilaian Konsultan terhadap Nilai yang Diperoleh untuk Setiap Sub Kategori ...	243
6.3.2.	Hasil Pengujian Validasi	243
VII	PENUTUP	
7.1.	Kesimpulan	254
7.2.	Saran	254
	DAFTAR PUSTAKA	255

DAFTAR TABEL

	Halaman
Tabel 2.1. Diagram UML.....	15
Tabel 2.2. Contoh Use Case Diagram.....	18
Tabel 2.3. Notasi Use Case.....	19
Tabel 2.4. Bentuk-bentuk Multiplicity.....	27
Tabel 4.1. Deskripsi Aktor.....	50
Tabel 4.2. Daftar Kebutuhan Fungsional.....	51
Tabel 4.3. Modul dan Kebutuhan Fungsional.....	57
Tabel 4.4. Daftar Kebutuhan Non Fungsional Sistem Psikotes Online.....	58
Tabel 4.5. Use Case Spesification Login.....	60
Tabel 4.6. Use Case Spesification Logout.....	61
Tabel 4.7. Use Case Spesification Menambah Daftar Petugas.....	62
Tabel 4.8. Use Case Spesification Melihat Daftar Petugas.....	64
Tabel 4.9. Use Case Spesification Mengedit Daftar Petugas.....	64
Tabel 4.10. Use Case Spesification Menghapus Daftar Petugas.....	66
Tabel 4.11. Use Case Spesification Mencari Daftar Petugas.....	67
Tabel 4.12. Use Case Spesification Menambah Data Instansi Client.....	69
Tabel 4.13. Use Case Spesification Mencari Data Instansi Client.....	70
Tabel 4.14. Use Case Spesification Melihat Data Instansi Client.....	71
Tabel 4.15. Use Case Spesification Mengedit Data Instansi Client.....	72
Tabel 4.16. Use Case Spesification Menghapus Data Instansi Client.....	74
Tabel 4.17. Use Case Spesification Menambah Data Kategori Soal.....	76
Tabel 4.17. Use Case Spesification Menambah Data Kategori Soal.....	76
Tabel 4.18. Use Case Spesification Melihat Data Kategori Soal.....	77
Tabel 4.19. Use Case Spesification Mengedit Data Kategori Soal.....	78
Tabel 4.20. Use Case Spesification Menghapus Data Kategori Soal.....	79
Tabel 4.21. Use Case Spesification Mencari Data Kategori Soal.....	80
Tabel 4.22. Use Case Spesification Menambah Urutan Kategori Soal.....	81
Tabel 4.23. Use Case Spesification Mengedit Urutan Kategori Soal.....	83
Tabel 4.24. Use Case Spesification Menghapus Urutan Kategori Soal.....	84



Tabel 4.25. Use Case Spesification Mencari Urutan Kategori Soal	85
Tabel 4.26. Use Case Spesification Melihat Urutan Kategori Soal	86
Tabel 4.27. Use Case Spesification Menambah Data Soal Psikotes	87
Tabel 4.28. Use Case Spesification Melihat Data Soal Psikotes.....	88
Tabel 4.29. Use Case Spesification Mengedit Data Soal Psikotes.....	89
Tabel 4.30. Use Case Spesification Menghapus Data Soal Psikotes	90
Tabel 4.31. Use Case Spesification Mencari Data Soal Psikotes.....	91
Tabel 4.32. Use Case Spesification Menjawab Data Soal Psikotes	92
Tabel 4.33. Use Case Spesification Menambah Sub Kategori Soal.....	93
Tabel 4.34. Use Case Spesification Melihat Sub Kategori Soal	94
Tabel 4.35. Use Case Spesification Mengedit Sub Kategori Soal	95
Tabel 4.36. Use Case Spesification Menghapus Sub Kategori Soal.....	96
Tabel 4.37. Use Case Spesification Mencari Sub Kategori Soal	97
Tabel 4.38. Use Case Spesification Menambah Pilihan Jawaban Soal.....	98
Tabel 4.39. Use Case Spesification Melihat Pilihan Jawaban Soal	100
Tabel 4.40. Use Case Spesification Mengedit Pilihan Jawaban Soal	100
Tabel 4.41. Use Case Spesification Menghapus Pilihan Jawaban Soal	102
Tabel 4.42. Use Case Spesification Menambah Data Waktu dan Tempat Pelaksanaan Tes	103
Tabel 4.43. Use Case Spesification Mencari Data Waktu dan Tempat Pelaksanaan Tes	105
Tabel 4.44. Use Case Spesification Mengedit Data Waktu dan Tempat Pelaksanaan Tes	106
Tabel 4.45. Use Case Spesification Menghapus Data Waktu dan Tempat Pelaksanaan Tes	108
Tabel 4.46. Use Case Spesification Melihat Data Waktu dan Tempat Pelaksanaan Tes	108
Tabel 4.47. Use Case Spesification Menambah Data Peserta Psikotes.....	110
Tabel 4.48. Use Case Spesification Melihat Data Peserta Psikotes	112
Tabel 4.49. Use Case Spesification Mengedit Data Peserta Psikotes	112
Tabel 4.50. Use Case Spesification Menghapus Data Peserta Psikotes	114
Tabel 4.51. Use Case Spesification Mencari Data Peserta Psikotes	114



Tabel 4.52. Use Case Spesification Mencari Data Peserta pada Lokasi Tes Psikotes.....	116
Tabel 4.53. Use Case Spesification Melihat Data Peserta pada Lokasi Tes Psikotes.....	117
Tabel 4.54. Use Case Spesification Mengedit Data Peserta pada Lokasi Tes Psikotes.....	118
Tabel 4.55. Use Case Spesification Menghapus Data Peserta pada Lokasi Tes Psikotes.....	119
Tabel 4.56. Use Case Spesification Mencari Data Peserta pada Lokasi Tes Psikotes.....	120
Tabel 4.57. Use Case Spesification Mengenerate Password Login Peserta Tes	121
Tabel 4.58. Use Case Spesification Melihat Nilai Peserta untuk Setiap sub Kategori Soal.....	122
Tabel 4.59. Use Case Spesification Mencari Nilai Peserta.....	123
Tabel 4.60. Use Case Spesification Menambah Penilaian Konsultan Terhadap Nilai yang Diperoleh Peserta untuk Setiap Sub Kategori	124
Tabel 4.61. Use Case Spesification Melihat Penilaian Konsultan Terhadap Nilai yang Diperoleh Peserta untuk Setiap Sub Kategori	125
Tabel 4.62. Data Object Description untuk Tabel tbl_petugas	139
Tabel 4.63. Data Object Description untuk Tabel tbl_instansi_client	139
Tabel 4.64. Data Object Description untuk Tabel tbl_kategori_soal.....	140
Tabel 4.65. Data Object Description untuk Tabel tbl_urutan_kategori_soal	140
Tabel 4.66. Data Object Description untuk Tabel tbl_lokasi_tes.....	141
Tabel 4.67. Data Object Description untuk Tabel tbl_soal	141
Tabel 4.68. Data Object Description untuk Tabel tbl_peserta	141
Tabel 4.69. Data Object Description untuk Tabel tbl_lokasi_peserta.....	142
Tabel 4.70. Data Object Description untuk Tabel tbl_jawaban	142
Tabel 4.71. Data Object Description untuk Tabel tbl_nilai	143
Tabel 4.72. Data Object Description untuk Tabel tbl_sub_kategori.....	143
Tabel 4.73. Data Object Description untuk Tabel tbl_pilihan_jawaban	144
Tabel 5.1. Spesifikasi Perangkat Keras	146

Tabel 5.2. Spesifikasi Perangkat Lunak	146
Tabel 5.3. Pengaturan File Logic Code dan Template Code.....	148
Tabel 5.4. Kelas dalam Package Logic Code.....	150
Tabel 6.1. Test Case untuk Pengujian Integrasi method create_session()	201
Tabel 6.2. Test Case untuk Pengujian Integrasi method setSoalList	203
Tabel 6.3. Test Case untuk Pengujian Integrasi method setJawaban.....	204
Tabel 6.4. Test Case untuk Pengujian Integrasi method simpanJawaban.....	205
Tabel 6.5a. Kasus Uji Login.....	206
Tabel 6.5b. Kasus Uji Login dimana pasangan username dan password tidak ada dalam sistem	206
Tabel 6.6. Kasus Uji Logout	206
Tabel 6.7a. Kasus Uji Menambah Data Petugas	207
Tabel 6.7b. Kasus uji menambah data petugas dimana masukan nama petugas, Username atau password kosong	207
Tabel 6.7c. Kasus uji menambah data petugas dimana masukan username Telah ada dalam sistem	207
Tabel 6.8. Kasus Uji Melihat Daftar Petugas.....	208
Tabel 6.9a. Kasus Uji Mengedit Data Petugas.....	208
Tabel 6.9b. Kasus Uji Mengedit Data Petugas Dimana Masukan nip atau nama Petugas atau username atau password kosong	209
Tabel 6.9c. Kasus Uji Mengedit Data Petugas dimana username petugas telah Ada dalam Sistem.....	209
Tabel 6.10. Kasus Uji Menghapus data Petugas	209
Tabel 6.11a. Kasus Uji Mencari Petugas	210
Tabel 6.11b. Kasus Uji Mencari Petugas dimana Masukan Keyword Kosong.....	210
Tabel 6.11c. Kasus Uji Mencari Petugas dan Menekan Tombol Refresh.....	210
Tabel 6.12a. Kasus Uji Menambah data Instansi Client	211
Tabel 6.12b. Kasus Uji Menambah Data Instansi Client dimana Masukan Nama instansi, bidang usaha, contact person, dan telepon Kosong.....	211
Tabel 6.12c. Kasus Uji Menambah Data Instansi Client dimana masukan nama	

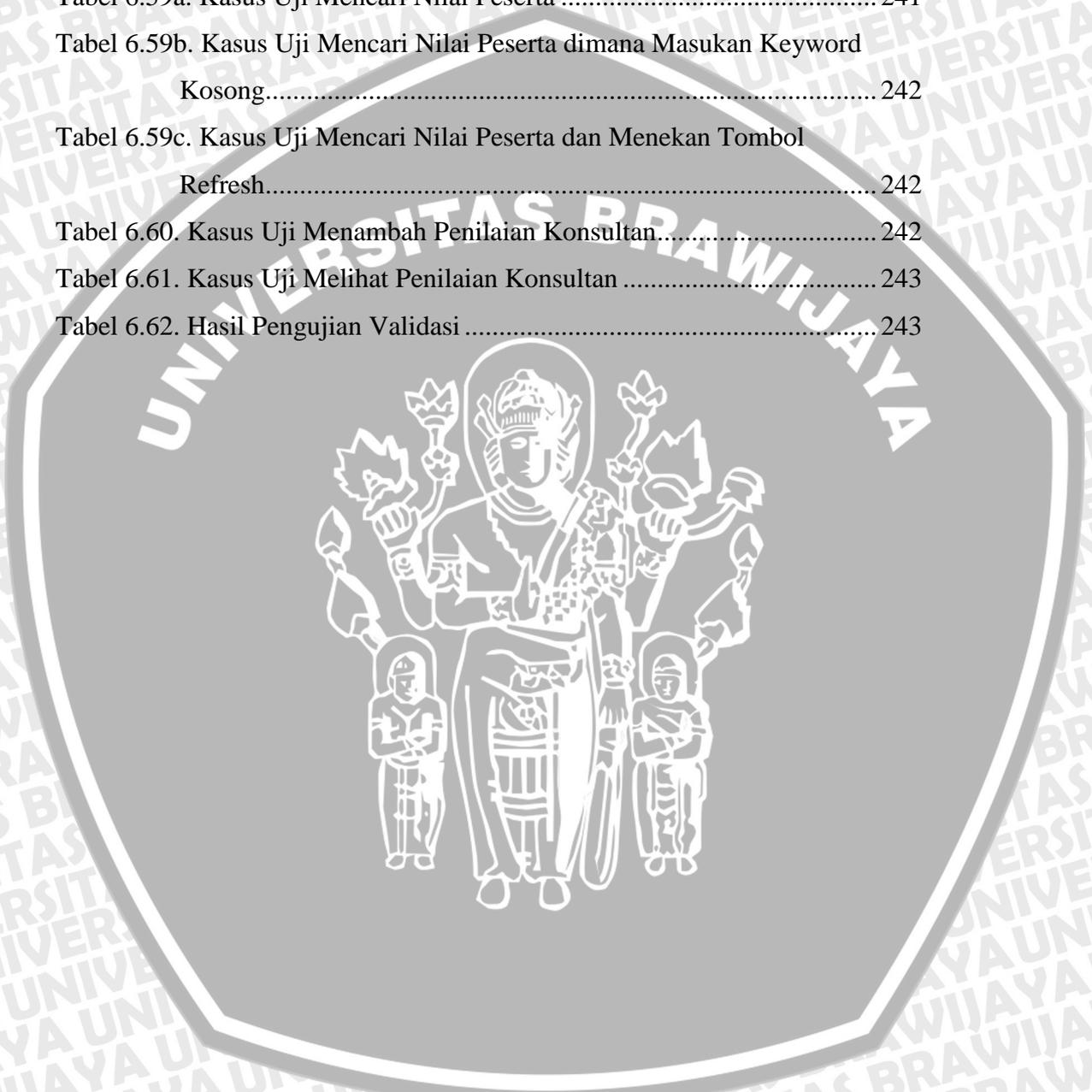
dan alamat telah Ada dalam Sistem	211
Tabel 6.13. Kasus Uji Melihat Daftar Instansi Client	212
Tabel 6.14a. Kasus Uji Mengedit Data Instansi Client	212
Tabel 6.14b. Kasus Uji Mengedit Data Instansi Client dimana masukan nama Instansi, alamat, contact person, telepon kosong	213
Tabel 6.14c. Kasus Uji Mengedit Data Instansi Client dimana Masukan nama Dan alamat Instansi Client telah Ada dalam Sistem	213
Tabel 6.15. Kasus Uji Menghapus Instansi Client	213
Tabel 6.16a. Kasus Uji Mencari Instansi Client.....	214
Tabel 6.16b. Kasus Uji Mencari Instansi Client dimana Masukan Keyword Kosong.....	214
Tabel 6.16c. Kasus Uji Mencari Instansi Client dan Menekan Tombol Refresh.....	214
Tabel 6.17a. Kasus Uji Menambah Kategori Soal	215
Tabel 6.17b. Kasus Uji Menambah Kategori Soal dimana Masukan nama Kategori kosong	215
Tabel 6.18. Kasus Uji Melihat Daftar Kategori Soal	215
Tabel 6.19a. Kasus Uji Mengedit Kategori Soal.....	216
Tabel 6.19b. Kasus Uji Mengedit Kategori Soal dimana Masukan Nama Kategori Kosong.....	216
Tabel 6.19c. Kasus Uji Mengedit Kategori Soal dimana Nama Kategori Telah Ada di dalam Sistem	216
Tabel 6.20. Kasus Uji Menghapus Kategori Soal	217
Tabel 6.21a. Kasus Uji mencari Kategori Soal.....	217
Tabel 6.21b. Kasus Uji Mencari Kategori Soal dimana Masukan Keyword Kosong.....	217
Tabel 6.21c. Kasus Uji Mencari Kategori Soal dan Menekan Tombol Refresh.....	218
Tabel 6.22a. Kasus Uji Menambah Urutan Kategori Soal	218
Tabel 6.22b. Kasus Uji Menambah Urutan Kategori Soal dimana Masukan Lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan Lama pengerjaan kosong.....	218

Tabel 6.23a. Kasus Uji Mengedit Urutan Kategori Soal.....	219
Tabel 6.23b. Kasus Uji Mengedit Urutan Kategori Soal dimana Masukan Lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan Lama pengerjaan kosong.....	219
Tabel 6.24. Kasus Uji Menghapus Urutan Kategori Soal	220
Tabel 6.25a. Kasus Uji Mencari Urutan Kategori Soal.....	220
Tabel 6.25b. Kasus Uji Mencari Urutan Kategori Soal dimana Masukan Keyword Kosong.....	220
Tabel 6.25c. Kasus Uji Mencari Urutan Kategori Soal dan Menekan Tombol Refresh Kosong	221
Tabel 6.26. Kasus Uji Melihat Daftar Urutan Kategori Soal	221
Tabel 6.27a. Kasus Uji Menambah Data Soal Psikotes	221
Tabel 6.27b. Kasus Uji Menambah Data Soal dimana Masukan Soal Kosong.....	222
Tabel 6.28. Kasus Uji Melihat Daftar Soal	222
Tabel 6.29a. Kasus Uji Mengedit Data Soal Psikotes	222
Tabel 6.29b. Kasus Uji Mengedit Data Soal Psikotes dimana Masukan Soal Kosong.....	223
Tabel 6.30. Kasus Uji Menghapus Data Soal.....	223
Tabel 6.31a. Kasus Uji Mencari Data Soal	223
Tabel 6.31b. Kasus Uji Mencari Data Soal dimana Masukan Keyword Kosong.....	224
Tabel 6.31c. Kasus Uji Mencari Data Soal dan Menekan Tombol Refresh.....	224
Tabel 6.32. Kasus Uji Menjawab Soal Psikotes.....	224
Tabel 6.33a. Kasus Uji Menambah Sub Kategori Soal.....	225
Tabel 6.33b. Kasus Uji Menambah Sub Kategori Soal dimana Masukan Nama Sub Kategori Kosong	225
Tabel 6.34. Kasus Uji Melihat Daftar Sub Kategori Soal	225
Tabel 6.35a. Kasus Uji Mengedit Sub Kategori Soal.....	226
Tabel 6.35b. Kasus Uji Mengedit Sub Kategori dimana Masukan Nama Sub Kategori Kosong	226

Tabel 6.35c. Kasus Uji Mengedit Sub Kategori dimana Nama Sub Kategori telah Ada dalam Sistem.....	226
Tabel 6.36. Kasus Uji Menghapus Sub Kategori Soal.....	227
Tabel 6.37a. Kasus Uji Mencari Sub Kategori Soal.....	227
Tabel 6.37b. Kasus Uji Mencari Sub Kategori Soal dimana Masukan Keyword Kosong.....	227
Tabel 6.37c. Kasus Uji Mencari Sub Kategori Soal dan Menekan Tombol Refresh.....	228
Tabel 6.38a. Kasus Uji Menambah Pilihan Jawaban Soal.....	228
Tabel 6.38b. Kasus Uji Menambah Pilihan Jawaban Soal dimana Masukan Huruf Pilihan, Jawaban, dan Status Jawaban Kosong.....	228
Tabel 6.39. Kasus Uji Melihat Daftar Pilihan Jawaban.....	229
Tabel 6.40a. Kasus Uji Mengedit Pilihan Jawaban.....	229
Tabel 6.40b. Kasus Uji Mengedit Pilihan Jawaban dimana masukan huruf Pilihan, Jawaban, dan Status Jawaban Kosong.....	229
Tabel 6.40c. Kasus Uji Mengedit Pilihan Jawaban dimana Huruf Pilihan Telah Ada dalam Sistem.....	230
Tabel 6.41. Kasus Uji Menghapus Pilihan Jawaban.....	230
Tabel 6.42a. Kasus Uji Menambah Data Waktu dan Tempat Pelaksanaan Tes.....	231
Tabel 6.42b. Kasus Uji Menambah Data Waktu dan Tempat Pelaksanaan Tes dimana Masukan Waktu, Tempat, dan Instansi Client Kosong.....	231
Tabel 6.43. Kasus Uji Melihat Data Waktu dan Tempat Pelaksanaan Tes.....	232
Tabel 6.44a. Kasus Uji Mengedit Data Waktu dan Tempat Pelaksanaan Tes.....	232
Tabel 6.44b. Kasus Uji Mengedit Data Waktu dan Tempat Pelaksanaan Tes dimana Masukan Waktu, Tempat, dan Instansi Client Kosong.....	232
Tabel 6.45. Kasus Uji Menghapus Data Waktu dan Tempat Pelaksanaan Tes.....	233

Tabel 6.46a. Kasus Uji Mencari Data Waktu dan Tempat Pelaksanaan Tes	233
Tabel 6.46b. Kasus Uji Mencari Data Waktu dan Tempat Pelaksanaan Tes dimana Masukan Keyword Kosong	233
Tabel 6.46c. Kasus Uji Mencari Data Waktu dan Tempat Pelaksanaan Tes dan Menekan Tombol Refresh	234
Tabel 6.47a. Kasus Uji Menambah Data Peserta Tes	234
Tabel 6.47b. Kasus Uji Menambah Data Peserta Tes dimana Masukan id peserta, Nama, alamat, pendidikan terakhir, jurusan, dan telepon Kosong.....	235
Tabel 6.48. Kasus Uji Melihat Data Peserta Tes.....	235
Tabel 6.49a. Kasus Uji Mengedit Peserta Tes.....	235
Tabel 6.49b. Kasus Uji Mengedit Peserta Tes dimana Masukan id peserta, Nama, alamat, pendidikan terakhir, jurusan, dan telepon Kosong.....	236
Tabel 6.50. Kasus Uji Menghapus Peserta Tes.....	236
Tabel 6.51a. Kasus Uji Mencari Peserta Tes.....	236
Tabel 6.51b. Kasus Uji Mencari Peserta Tes dimana Masukan Keyword Kosong.....	237
Tabel 6.51c. Kasus Uji Mencari Peserta Tes dan Menekan Tombol Refresh.....	237
Tabel 6.52a. Kasus Uji Menambah Data Peserta di Lokasi Tes	237
Tabel 6.52b. Kasus Uji Menambah Data Peserta di Lokasi Tes dimana Masukan Id peserta, tempat tes, dan waktu tes kosong	238
Tabel 6.53. Kasus Uji Melihat Data Peserta di Lokasi Tes.....	238
Tabel 6.54a. Kasus Uji Mengedit Data Peserta di Lokasi Tes	239
Tabel 6.54b. Kasus Uji Mengedit Data Peserta di Lokasi Tes dimana Masukan Id peserta, tempat tes, dan waktu tes kosong	239
Tabel 6.55. Kasus Uji Menghapus Peserta di Lokasi Tes	239
Tabel 6.56a. Kasus Uji Mencari Data Peserta di Lokasi Tes	240
Tabel 6.56b. Kasus Uji Mencari Data Peserta di Lokasi Tes dimana Masukan Keyword Kosong.....	240

Tabel 6.56c. Kasus Uji Mencari Data Peserta di Lokasi Tes dan Menekan Tombol Refresh.....	240
Tabel 6.57. Kasus Uji Mengenerate Password Login Peserta Tes.....	241
Tabel 6.58. Kasus Uji Melihat Nilai Peserta.....	241
Tabel 6.59a. Kasus Uji Mencari Nilai Peserta.....	241
Tabel 6.59b. Kasus Uji Mencari Nilai Peserta dimana Masukan Keyword Kosong.....	242
Tabel 6.59c. Kasus Uji Mencari Nilai Peserta dan Menekan Tombol Refresh.....	242
Tabel 6.60. Kasus Uji Menambah Penilaian Konsultan.....	242
Tabel 6.61. Kasus Uji Melihat Penilaian Konsultan.....	243
Tabel 6.62. Hasil Pengujian Validasi.....	243



DAFTAR GAMBAR

	Halaman
Gambar 2.1. Model Proses Waterfall	13
Gambar 2.2. Klasifikasi Jenis Diagram UML Versi 2	15
Gambar 2.3. Contoh Use Case Diagram	18
Gambar 2.4. Asosiasi Antara Aktor dengan Use Case	20
Gambar 2.5. Generalisasi dari Aktor	21
Gambar 2.6. Contoh Generalisasi Aktor	22
Gambar 2.7. Generalisasi dari Use Case	22
Gambar 2.8. Notasi <<include>>	23
Gambar 2.9. Contoh include antara dua buah use case	23
Gambar 2.10. Notasi <<extend>>	24
Gambar 2.11. Notasi Klas	25
Gambar 2.12. Contoh Notasi Klas dengan Tiga Buah Compartment	26
Gambar 2.13. Contoh Notasi Klas dengan Inlined Attribute	26
Gambar 2.14. Notasi Dependensi	29
Gambar 2.15. Notasi Asosiasi	29
Gambar 2.16. Notasi Agregasi	30
Gambar 2.17. Notasi Komposisi	30
Gambar 2.18. Notasi Generalisasi	31
Gambar 2.19. Contoh Diagram Sequence	32
Gambar 2.20. Notasi Lifelines	32
Gambar 2.21. Notasi Message	33
Gambar 2.22. Notasi Synchronous Message	33
Gambar 2.23. Notasi Return Message	33
Gambar 2.24. Notasi Grafik Alir	37
Gambar 2.25. Diagram Ilustrasi Cara Kerja Template Engine	38
Gambar 2.26. Notasi Entitas	40
Gambar 2.27. Notasi Atribut	40
Gambar 2.28. Contoh Hubungan Himpunan Entitas dengan Atribut	41
Gambar 2.29. Notasi Himpunan Relasi	41

Gambar 4.1. Diagram Pohon Analisis Kebutuhan dan Perancangan.....	47
Gambar 4.2. Penyelenggaraan Psikotes Saat Ini	48
Gambar 4.3. Penyelenggaraan Psikotes Online	49
Gambar 4.4. Generalisasi Aktor-aktor Pengguna Sistem Psikotes Online....	51
Gambar 4.5. Use Case Diagram untuk Modul Pendukung Sistem Psikotes Online	60
Gambar 4.6. Use Case Diagram untuk Modul Pengaturan Instansi Client Sistem Psikotes Online.....	68
Gambar 4.7. Use Case Diagram untuk Modul Pengaturan Soal Tes	75
Gambar 4.8. Use Case Diagram untuk Modul Pengaturan Penyelenggaraan Tes	103
Gambar 4.9. Use Case Diagram untuk Modul Pengaturan Peserta Tes.....	110
Gambar 4.10. Use Case Diagram untuk Modul Perhitungan Nilai dan Hasil Penilaian.....	122
Gambar 4.11. Class Diagram psikotes.LogicCode.administrator	127
Gambar 4.12. Class Diagram psikotes.LogicCode.dataSoal.....	127
Gambar 4.13. Class Diagram psikotes.LogicCode.hasilTes	128
Gambar 4.14. Class Diagram psikotes.LogicCode.home.....	128
Gambar 4.15. Class Diagram psikotes.LogicCode.instansiClient.....	129
Gambar 4.16. Class Diagram psikotes.LogicCode.pelaksanaanTes	129
Gambar 4.17. Class Diagram psikotes.LogicCode.pengaturanSoal.....	130
Gambar 4.18. Class Diagram psikotes.LogicCode.pesertaLokasiTes.....	130
Gambar 4.19. Class Diagram psikotes.LogicCode.pesertaTes	131
Gambar 4.20. Class Diagram psikotes.LogicCode.petugas	131
Gambar 4.21. Class Diagram psikotes.LogicCode.pilihanJawaban.....	132
Gambar 4.22. Class Diagram psikotes.LogicCode.subKategori	132
Gambar 4.23. Class Diagram psikotes.LogicCode.urutanKategoriSoal	133
Gambar 4.24. Class Diagram psikotes.LogicCode.client.....	133
Gambar 4.25. Class Diagram psikotes.LogicCode.client.soal	134
Gambar 4.26. Sequence Diagram untuk Method setSoalList	135
Gambar 4.27. Sequence Diagram untuk Method setJawaban.....	135
Gambar 4.28. Sequence Diagram untuk Method simpanJawaban.....	136

Gambar 4.29. ER Diagram Sistem Psikotes Online.....	137
Gambar 4.30. Conceptual Data Model Sistem Psikotes Online.....	138
Gambar 5.1. Diagram Pohon Implementasi Sistem	145
Gambar 5.2. Konfigurasi Smarty Template Engine	146
Gambar 5.3. Notasi Smarty untuk Menunjuk Template Code	147
Gambar 5.4. Pseudocode Algoritma Method setSoalList(db,smarty,ps)	153
Gambar 5.5. Template Code untuk Method setSoalList(db,smarty,ps)	153
Gambar 5.6. Pseudocode Algoritma Method setJawaban(db,smarty,ps).....	154
Gambar 5.7. Template Code untuk Method setJawaban (db,smarty,ps).....	154
Gambar 5.8. Pseudocode Algoritma Method simpanJawaban(db,smarty,ps).....	155
Gambar 5.9. Halaman Login	156
Gambar 5.10. Halaman Home.....	156
Gambar 5.11. Halaman Petugas	157
Gambar 5.12. Legend Detail Biodata Petugas pada Form Tambah Petugas.	157
Gambar 5.13. Legend Detail Login Petugas pada Form Tambah Petugas....	158
Gambar 5.14. Legend Detail Biodata Petugas pada Form Edit Petugas	158
Gambar 5.15. Legend Detail Login pada Form Edit Petugas	159
Gambar 5.16. Legend Detail Login pada Form Detail Petugas	159
Gambar 5.17. Legend Detail Biodata pada Form Detail Petugas	159
Gambar 5.18. Konfirmasi Hapus Petugas	160
Gambar 5.19. Halaman Pencarian Jika yang Dicari Ada dalam Database....	160
Gambar 5.20. Halaman Pencarian Jika yang Dicari Tidak Ada dalam Database	161
Gambar 5.21. Konfirmasi Instansi Client.....	161
Gambar 5.22. Halaman Tambah Instansi Client	162
Gambar 5.23. Halaman Edit Instansi Client.....	162
Gambar 5.24. Halaman Detail Instansi Client.....	163
Gambar 5.25. Halaman Hapus Instansi Client	163
Gambar 5.26. Halaman Cari Instansi Client Jika yang Dicari Ada dalam Database	164
Gambar 5.27. Halaman Cari Instansi Client Jika yang Dicari Tidak Ada dalam	

Database	164
Gambar 5.28. Halaman Kategori Soal.....	165
Gambar 5.29. Halaman Tambah Kategori Soal	165
Gambar 5.30. Halaman Edit Kategori Soal.....	166
Gambar 5.31. Halaman Detail Kategori Soal.....	166
Gambar 5.32. Halaman Cari Kategori Soal Jika yang Dicari Ada dalam Database	167
Gambar 5.33. Halaman Cari Kategori Soal Jika yang Dicari Tidak Ada dalam Database	168
Gambar 5.34. Letak Link Sub Kategori	168
Gambar 5.35. Halaman Sub Kategori	168
Gambar 5.36. Halaman Tambah Sub Kategori	168
Gambar 5.37. Halaman Edit Sub Kategori.....	169
Gambar 5.38. Halaman Detail Sub Kategori.....	169
Gambar 5.39. Halaman Hapus Sub Kategori	170
Gambar 5.40. Halaman Cari Sub Kategori Jika yang Dicari Ada dalam Database	170
Gambar 5.41. Halaman Cari Sub Kategori Jika yang Dicari Tidak Ada dalam Database	171
Gambar 5.42. Letak Link Soal	172
Gambar 5.43. Halaman Soal.....	172
Gambar 5.44. Halaman Tambah Soal	173
Gambar 5.45. Halaman Edit Soal.....	173
Gambar 5.46. Halaman Detail Soal.....	173
Gambar 5.47. Halaman Hapus Soal	174
Gambar 5.48. Halaman Cari Soal Jika yang Dicari Ada dalam Database	174
Gambar 5.49. Halaman Cari Soal Jika yang Dicari Tidak Ada dalam Database	175
Gambar 5.50. Letak Link Pilihan Jawaban	175
Gambar 5.51. Halaman Pilihan Jawaban.....	176
Gambar 5.52. Halaman Tambah Pilihan Jawaban	176
Gambar 5.53. Halaman Edit Pilihan Jawaban.....	177



Gambar 5.54. Halaman Detail Pilihan Jawaban.....	177
Gambar 5.55. Halaman Hapus Pilihan Jawaban	178
Gambar 5.56. Halaman Urutan Kategori Soal	178
Gambar 5.57. Halaman Tambah Urutan Kategori Soal	179
Gambar 5.58. Halaman Edit Urutan Kategori Soal.....	179
Gambar 5.59. Halaman Detail Urutan Kategori Soal.....	180
Gambar 5.60. Halaman Hapus Urutan Kategori Soal	180
Gambar 5.61. Halaman Cari Urutan Kategori Soal Jika yang Dicari Ada dalam Database	181
Gambar 5.62. Halaman Cari Urutan Kategori Soal Jika yang Dicari Tidak Ada dalam Database.....	181
Gambar 5.63. Halaman Pelaksanaan Tes	182
Gambar 5.64. Halaman Tambah Pelaksanaan Tes	182
Gambar 5.65. Halaman Edit Pelaksanaan Tes	183
Gambar 5.66. Halaman Detail Pelaksanaan Tes	183
Gambar 5.67. Halaman Hapus Pelaksanaan Tes.....	184
Gambar 5.68. Halaman Cari Pelaksanaan Tes Jika yang Dicari Ada dalam Database	184
Gambar 5.69. Halaman Cari Pelaksanaan Tes Jika yang Dicari Tidak Ada dalam Database	185
Gambar 5.70. Halaman Data Peserta Tes.....	185
Gambar 5.71. Halaman Tambah Peserta Tes	186
Gambar 5.72. Halaman Edit Peserta Tes.....	186
Gambar 5.73. Halaman Detail Peserta Tes.....	187
Gambar 5.74. Halaman Hapus Peserta Tes	187
Gambar 5.75. Halaman Cari Peserta Tes Jika yang Dicari Ada dalam Database	188
Gambar 5.76. Halaman Cari Peserta Tes Jika yang Dicari Tidak Ada dalam Database	188
Gambar 5.77. Halaman Data Peserta di Lokasi Tes.....	189
Gambar 5.78. Halaman Tambah Data Peserta di Lokasi Tes.....	189
Gambar 5.79. Halaman Edit Data Peserta di Lokasi Tes	190

Gambar 5.80. Halaman Detail Data Peserta di Lokasi Tes.....	190
Gambar 5.81. Halaman Hapus Data Peserta di Lokasi Tes.....	191
Gambar 5.82. Halaman Cari Data Peserta di Lokasi Tes Jika yang Dicari Ada dalam Database.....	191
Gambar 5.83. Halaman Cari Data Peserta di Lokasi Tes Jika yang Dicari Tidak Ada dalam Database.....	192
Gambar 5.84. Halaman Hasil Tes	192
Gambar 5.85. Halaman Edit Hasil Tes.....	193
Gambar 5.86. Halaman Detail Hasil Tes.....	193
Gambar 5.87. Halaman Hapus Hasil Tes	194
Gambar 5.88. Halaman Cari Hasil Tes Jika yang Dicari Ada dalam Database	194
Gambar 5.89. Halaman Cari Hasil Tes Jika yang Dicari Tidak Ada dalam Database	195
Gambar 5.90. Login Peserta.....	195
Gambar 5.91. Halaman Urutan Kategori Tes Pertama.....	196
Gambar 5.92. Halaman Soal Kategori Tes Pertama.....	197
Gambar 5.93. Halaman Urutan Kategori Tes Kedua.....	197
Gambar 5.94. Halaman Soal Kategori Tes Kedua.....	197
Gambar 5.95. Halaman Peringatan.....	198
Gambar 5.96. Halaman Terakhir Psikotes.....	198
Gambar 6.1. Diagram Pohon Pengujian.....	199
Gambar 6.2. Klas Diagram private_sessions.....	200
Gambar 6.3. Pemodelan Operasi create_session() ke dalam Flow Graph	201
Gambar 6.4. Pemodelan Operasi setSoalList(db,smarty,ps) ke dalam Flow Graph.....	202
Gambar 6.5. Pemodelan Operasi setJawaban(db,smarty,ps) ke dalam Flow Graph.....	203
Gambar 6.6. Pemodelan Operasi simpanJawaban(db,smarty,ps) ke dalam Flow Graph.....	204

BAB I PENDAHULUAN

1.1 Latar Belakang

Website merupakan jembatan bagi para pemburu informasi di dunia maya untuk berselancar di internet. Pembuatan *website* oleh suatu lembaga seringkali dikerjakan dalam sebuah tim yang nantinya akan terbagi – bagi sesuai dengan tugasnya antara lain, *system analist*, *template designer*, *database designer*, dan *programmer*.

Bekerja dalam suatu tim tidak mudah, harus memerlukan koordinasi dan komunikasi yang cukup baik, terutama dalam menyelesaikan sebuah *website*. Jika seorang *template designer* akan merubah *template codenya*, sebisa mungkin *logic code* milik *programmer* tidak terjamah sedikitpun. Demikian pula dengan *programmer*, jika sedang melakukan perubahan terhadap *logic codenya*, sebisa mungkin *template code* tidak ada yang terhapus atau terganti baik sengaja maupun tidak sengaja. Karena alasan inilah, perlu digunakan sebuah *template engine* untuk memisahkan antara *template code* dan *logic code*, sehingga *template designer* tidak akan dapat menghapus dan merubah *logic code*, dan *programmer* dapat dengan leluasa merubah *logic codenya*.

Macam – macam *template engine* antara lain : Smarty, Heyes Template Class, FastTemplate, VarPage Class, Powerful PHP Templates, dll. *Template engine* yang banyak digunakan saat ini salah satunya adalah *Smarty* yang terkenal dengan kecepatan dan fleksibilitasnya dibandingkan dengan *template engine* lainnya. *Smarty* diciptakan khusus sebagai *template engine* pada *PHP*. Menggunakan teknologi *Smarty* akan memberikan beberapa keuntungan tersendiri, diantaranya :

1. Kinerja web menjadi lebih cepat [SMA-07:02].
2. Lebih efisien karena tag *PHP* dan *HTML* terbuat secara terpisah [SMA-07:02].
3. Dapat berfungsi sebagai compiler *template* [SMA-07:02].
4. Dapat membuat fungsi-fungsi *logic* di *PHP* tanpa mempengaruhi *template* [SMA-07:02].

PHP (Hypertext Preprocessor) dapat membuat *website* menjadi lebih dinamis dan interaktif terhadap pengunjung. *PHP* tidak mempunyai *script editor*, sehingga kita akan tetap bisa menggunakan *text editor* apapun. Namun selain itu *PHP* juga tidak memiliki *Compiler*, karena *Script PHP* akan *dicompile* oleh *web server* saat digunakan [PHP-07:04].

MySQL adalah suatu bahasa berbasis *SQL* yang dikembangkan secara *free* dan *open source*. *MySQL* membuat pemrogram dan seorang administrator *database* dapat melakukan hal-hal berikut :

1. Membuat dan mengubah struktur sebuah *database* [SQL-06:08].
2. Mengubah pengaturan keamanan sistem [SQL-06:08].
3. Memberikan hak akses kepada pengguna untuk mengakses *database* atau tabel [SQL-06:08].
4. Memperoleh informasi dari *database* [SQL-06:08].
5. Mengupdate isi *database* [SQL-06:08].

Keunggulan *PHP*, *MySQL* dan teknologi *Smarty* yang telah diuraikan menjadi alasan bagi penyusun untuk menggunakan *PHP* sebagai bahasa pemrograman dan teknologi *Smarty* sebagai *template engine*nya untuk membuat *software psikotes online*.

Psikologi merupakan salah satu bidang yang juga memperoleh dampak akibat berkembangnya internet. Banyak situs – situs yang menawarkan adanya uji kepribadian secara *online*. Dengan adanya uji kepribadian secara *online* ini akan membantu peselancar untuk mengetahui kecerendungan dari sifat masing – masing. Selain itu juga dapat dijadikan bahan acuan untuk mengikuti psikotes yang sesungguhnya dalam tes untuk melamar sebuah pekerjaan. Dengan demikian, metode semacam ini dapat dimodifikasi oleh konsultan psikologi yang biasanya disewa oleh instansi – instansi untuk menyelenggarakan psikotes bagi para pelamarnya dan melakukan psikotes secara *online*.

Para konsultan psikologi dapat memanfaatkan teknologi yang sama sehingga ketika melakukan psikotes terhadap pelamar dari instansi yang menjadi clientnya dapat mengetahui hasilnya lebih cepat dan dapat memodifikasi soal – soal psikotes kapan saja dengan cepat.

Psikologi memiliki tiga fungsi sebagai ilmu yaitu:

1. Menjelaskan
Yaitu mampu menjelaskan apa, bagaimana, dan mengapa tingkah laku itu terjadi. Hasilnya penjelasan berupa deskripsi atau bahasan yang bersifat deskriptif [PSI-05:03].
2. Memprediksikan
Yaitu mampu meramalkan atau memprediksikan apa, bagaimana, dan mengapa tingkah laku itu terjadi. Hasil prediksi berupa prognosa, prediksi atau estimasi [PSI-05:03].
3. Pengendalian
Yaitu mengendalikan tingkah laku sesuai dengan yang diharapkan. Perwujudannya berupa tindakan atau treatment [PSI-05:03].

Ketiga fungsi diatas dapat menjelaskan bahwa ilmu psikologi sangat penting bagi seseorang untuk mengetahui kepribadian masing – masing sehingga dapat bersosialisasi dengan orang lain dengan baik. Maka dari itu, dibutuhkan sebuah tes untuk menguji kepribadian tersebut. Dengan demikian, akan sangat membantu sekali jika terdapat software yang dapat merealisasikan fungsi tersebut.

1.2 Rumusan Masalah

pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka rumusan masalah dapat disusun sebagai berikut :

1. Bagaimana cara menerapkan teknologi *Smarty* pada bahasa pemrograman *PHP* untuk merancang dan membuat sistem psikotes *online*?
2. Bagaimana merancang dan membuat sistem psikotes *online*?
3. Bagaimana mengimplementasikan dan menguji sistem psikotes *online*?

1.3 Batasan Masalah

Pembatasan masalah yang diajukan dalam penyusunan skripsi ini adalah sebagai berikut:

1. Pembahasan difokuskan kepada pembuatan sistem psikotes *online*.
2. *Content* sistem psikotes *online* hanya berisi soal – soal yang berbentuk textual.
3. Sistem psikotes *online* tidak menyediakan fasilitas untuk tes kepribadian, bakat, dan minat.

4. Sistem psikotes *online* sifatnya membantu konsultan untuk memberikan penilaian terhadap nilai yang diperoleh peserta, bukan menggantikan penilaian konsultan.
5. Pembahasan metode pembuatan sistem psikotes *online* menggunakan bahasa pemrograman *PHP*, database *MySQL*, dan *template engine Smarty*.
6. Sistem yang dibuat merupakan aplikasi yang berbasis web.

1.4 Tujuan

Tujuan penulisan ini adalah merancang dan mengimplementasikan sistem psikotes *online* yang dibuat menggunakan *PHP*, *Smarty*, dan *database MySQL*.

1.5 Manfaat

Manfaat yang diharapkan dapat diperoleh melalui pengerjaan skripsi ini adalah :

1. Bagi penyusun
 - a) Penyusun dapat mengembangkan perangkat lunak berbasis web dengan menggunakan *PHP*, *Smarty*, database *MySQL* sebagai sarana untuk pembuatannya.
 - b) Memperoleh pemahaman mengenai perancangan dan pengembangan menggunakan metodologi desain teknologi *Smarty* sebagai sarana untuk pembuatannya.
 - c) Menambah bekal pengetahuan dan wawasan baru serta dapat belajar memecahkan masalah dengan efisien dan berfikir secara lebih kritis dan tepat dalam membuat sebuah keputusan.
 - d) Mengimplementasikan ilmu yang telah didapat pada kegiatan perkuliahan.
2. Bagi pengguna
 - a) Memberikan alternatif bagi para konsultan psikologi untuk menyelenggarakan psikotes secara *online*.

1.6 Sistematika Pembahasan

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika pembahasan.

BAB II Dasar Teori

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan sistem.

BAB III Metodologi

Berisi tentang metode penelitian yang digunakan dalam perancangan dan pengujian *software*.

BAB IV Analisis kebutuhan dan Perancangan

Membahas tentang analisa kebutuhan dari sistem dan kemudian merancang hal-hal yang berhubungan dengan analisa tersebut. Membahas tentang perencanaan dan pembuatan sistem.

BAB V Implementasi

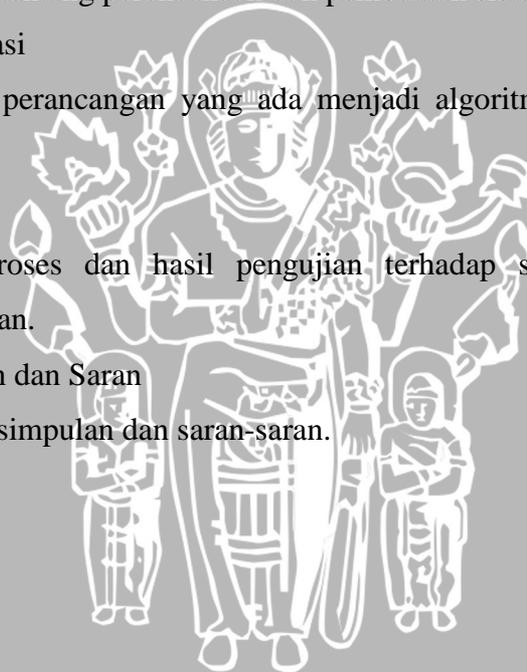
Mengubah perancangan yang ada menjadi algoritma dan kode-kode program

BAB VI Pengujian

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VII Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran.



BAB II TINJAUAN PUSTAKA

Bab ini membahas kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai penerapan *Smarty template engine* dalam perancangan dan pembuatan *software* psikotes *online*.

2.1 Kajian Pustaka

Psikologi adalah ilmu yang luas dan ambisius, dilengkapi oleh biologi pada kaitannya dengan ilmu alam dan dilengkapi oleh sosiologi dan antropologi pada kaitannya dengan ilmu social. Beberapa kajian ilmu psikologi diantaranya adalah:

1. Psikologi perkembangan

Adalah bidang studi psikologi yang mempelajari perkembangan manusia dan faktor-faktor yang membentuk perilaku seseorang sejak lahir sampai lanjut usia. Psikologi perkembangan berkaitan erat dengan psikologi sosial, karena sebagian besar perkembangan terjadi dalam konteks adanya interaksi sosial. Dan juga berkaitan erat dengan psikologi kepribadian, karena perkembangan individu dapat membentuk kepribadian khas dari individu tersebut [PSU-04:10].

2. Psikologi sosial

Mempunyai 3 ruang lingkup, yaitu :

1. studi tentang pengaruh sosial terhadap proses individu, misalnya : studi tentang persepsi, motivasi proses belajar, atribusi (sifat) [PSU-04:10].
2. studi tentang proses-proses individual bersama, seperti bahasa, sikap sosial, perilaku meniru dan lain-lain [PSU-04:10].
3. studi tentang interaksi kelompok, misalnya : kepemimpinan, komunikasi hubungan kekuasaan, kerjasama, persaingan, konflik [PSU-04:10].

3. Psikologi kepribadian

Adalah bidang studi psikologi yang mempelajari tingkah laku manusia dalam menyesuaikan diri dengan lingkungannya, psikologi kepribadian berkaitan erat dengan psikologi perkembangan dan psikologi sosial, karena kepribadian

adalah hasil dari perkembangan individu sejak masih kecil dan bagaimana cara individu itu sendiri dalam berinteraksi sosial dengan lingkungannya [PSU-04:10].

4. Psikologi kognitif

Adalah bidang studi psikologi yang mempelajari kemampuan kognisi, seperti: Persepsi, proses belajar, kemampuan memori, atensi, kemampuan bahasa dan emosi [PSU-04:10].

Psikologi merupakan salah satu bidang yang juga memperoleh dampak akibat berkembangnya internet. Banyak situs – situs yang menawarkan adanya uji kepribadian secara *online*. Dengan adanya uji kepribadian secara *online* ini akan membantu peselancar untuk mengetahui kecerendungan dari sifat masing – masing. Selain itu juga dapat dijadikan bahan acuan untuk mengikuti psikotes yang sesungguhnya dalam tes untuk melamar sebuah pekerjaan. Dengan demikian, metode semacam ini dapat dimodifikasi oleh konsultan psikologi yang biasanya disewa oleh instansi – instansi untuk menyelenggarakan psikotes bagi para pelamarnya dan melakukan psikotes secara *online*.

Para konsultan psikologi dapat memanfaatkan teknologi yang sama sehingga ketika melakukan psikotes terhadap pelamar dari instansi yang menjadi clientnya dapat mengetahui hasilnya lebih cepat dan dapat memodifikasi soal – soal psikotes kapan saja dengan cepat. Penerapan *software* psikotes *online* ini nantinya akan sangat bermanfaat dan dapat menjadi sebuah alternatif bagi para konsultan untuk menyelenggarakan psikotes secara *online*.

Proses pengembangan perangkat lunak harus melalui suatu proses rekayasa perangkat lunak (RPL), begitu pula dengan perangkat lunak psikotes *online* yang ditujukan bagi para jasa konsultan psikologi. Rekayasa perangkat lunak ini dimaksudkan untuk memberi landasan sehingga pengembangan perangkat lunak dilakukan secara sistematis dengan sasaran memperoleh produk akhir yang berkualitas [HAR-04:3]. Perangkat lunak berkualitas tinggi adalah program yang handal, efisien, mudah dipahami, mudah dimodifikasi dan mudah dipelihara [HAR-04:3].

Pengembangan *software* ini menggunakan paradigma *object oriented* (OO) dengan model proses *waterfall* dan pemodelan UML. Paradigma OO

digunakan karena paradigma RPL terstruktur tidak dapat mengatasi masalah-masalah penting, seperti *encapsulation*, *reusability*, dan *abstraction* [RYA-03:4]. Model proses *waterfall* digunakan karena model ini mewajibkan kedisiplinan dalam setiap fase RPL, sehingga suatu fase harus telah selesai sebelum dapat ke fase berikutnya.

Implementasi sistem ini akan menggunakan *PHP* (*Hypertext Preprocessor*), *Smarty template engine*, dan database *MySQL*. Penerapan *template engine* dilakukan karena diperlukan adanya pemisahan *template code* dan *logic code*. *Smarty* dipilih karena merupakan *template engine* yang kinerjanya paling cepat, akurat, dan efisien [SMA-07:02]. *PHP* dipilih karena dapat membuat website menjadi lebih dinamis dan interaktif terhadap pengunjung. *PHP* tidak mempunyai *script editor*, sehingga kita akan tetap bisa menggunakan *text editor* apapun. Namun selain itu *PHP* juga tidak memiliki *Compiler*, karena *Script PHP* akan *dcompile* oleh *web server* saat digunakan [PHP-07:04]. Database *MySQL* dipilih karena database ini cocok untuk sistem yang membutuhkan database besar, selain itu juga *MySQL* adalah suatu bahasa berbasis *SQL* yang dikembangkan secara *free* dan *open source* [SQL-06:08].

2.2 Dasar Teori

Berdasarkan kajian pustaka di atas, maka untuk penyelesaian skripsi ini dibutuhkan dasar teori sebagai berikut :

1. Teori dasar psikotes
2. Teori dasar RPL, meliputi :
 - a. Proses-proses RPL
 - b. Model proses perangkat lunak
 - c. Paradigma RPL
3. UML, meliputi :
 - a. *Use case diagram*
 - b. *Class diagram*
 - c. *Sequence diagram*
4. Pengujian
 - a. Teknik pengujian

- b. Strategi pengujian
5. Teknologi *Template Engine* dan *PHP*
 - a. *Smarty template Engine*
 - b. *PHP (Hypertext Preprocessor)*
6. Basis data

2.2.1 Teori Dasar Psikotes

Kompetensi profesional oleh kebanyakan orang sering disebut dengan “*skill*”. Padahal, lebih tepatnya, kompetensi ini termasuk dalam dimensi “*hard skill*”. Untuk meraih kesuksesan dalam suatu pekerjaan, sebenarnya sangat diperlukan kompetensi intrapersonal dan interpersonal. Kedua kompetensi ini termasuk dalam dimensi “*soft skill*” yang merupakan salah satu bagian dari kajian psikologi. Bagaimana kita mampu mengaplikasikannya dalam kehidupan keseharian sehingga bisa termaknakan dan menjadi kepribadian kita yang sesungguhnya. Sejauh mana kita bisa mengaplikasikannya, akan tergambar dan tergali melalui psikotes [IKO:08].

Kompetensi sebenarnya terbagi dalam tiga bagian sebagai berikut :

1. Kompetensi profesional, yaitu kemampuan kita untuk menjalankan profesi tertentu [IKO:08].
2. Kompetensi intrapersonal, yaitu kemampuan kita untuk mengenal dan mengendalikan diri sendiri [IKO:08].
3. Kompetensi interpersonal, yaitu kemampuan kita untuk bergaul dan berinteraksi dengan orang lain [IKO:08].

2.2.1.1 Komponen Psikotes

Aspek – aspek psikologis yang biasanya dinilai dalam berbagai tes terdiri [IKO:08] :

1. Motivasi berprestasi, yaitu dorongan dan keinginan individu untuk melakukan pekerjaan sebaik mungkin [IKO:08].
2. Stabilisasi emosi, yaitu kemampuan individu untuk mengendalikan diri atau mengontrol emosi dalam berbagai situasi, baik situasi yang normal maupun situasi yang penuh dengan tekanan dan gejolak [IKO:08].

3. Hubungan sosial, yaitu kemampuan individu dalam menjalin dan membina hubungan interpersonal atau berhubungan dengan orang lain, yakni dalam hal senang bergaul dan terlibat dalam suatu kegiatan dengan orang lain [IKO:08].
 4. Penyesuaian diri, yaitu kemampuan individu untuk menyesuaikan sikap terhadap orang lain dan lingkungan sekitar, luwes, dan cepat atau mudah menyesuaikan diri di lingkungan sosial yang baru [IKO:08].
 5. Kemandirian, yaitu kemampuan dan keinginan individu untuk menentukan semua tindakannya sendiri tanpa tergantung pada orang lain [IKO:08].
 6. Komunikasi, yaitu kemampuan individu untuk mengutarakan atau menyampaikan pikiran, ide, dan pendapatnya secara sistematis sehingga buah pikirannya dapat dimengerti orang lain, serta mampu menangkap maksud pikiran dan ide orang lain [IKO:08].
 7. Kerja sama, yaitu kemampuan dan keinginan individu untuk berinteraksi dan melibatkan diri atau berpartisipasi secara aktif dalam pencapaian tujuan kelompok, serta mampu bekerja sama dengan orang lain dalam suatu kelompok [IKO:08].
 8. Kepemimpinan, yaitu kemampuan individu untuk melakukan koordinasi dalam menyelesaikan suatu pekerjaan, menggunakan keterampilan interpersonal dalam mengarahkan atau mengatur, serta membimbing orang lain (bawahan) atau kelompok dalam rangka mencapai tujuan [IKO:08].
- Persoalan yang akan disajikan dalam psikotes bentuknya beraneka ragam. Ada yang berupa kalimat, hitungan, menggambar dan sebagainya.

2.2.2 Teori Dasar RPL

Pengertian perangkat lunak dalam arti sempit adalah program yang dijalankan di dalam suatu pemroses [HAR-04]. Pengertian perangkat lunak dalam arti luas adalah program yang dieksekusi komputer, beserta dokumen-dokumen pendukung program [HAR-04].

Program berkualitas tinggi adalah program yang handal (*reliable*), efisien dan mudah dipahami, dimodifikasi dan dipelihara [HAR-04]. Untuk membuat program yang berkualitas tinggi, diperlukan adanya rekayasa.

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan [SOM-04]. RPL dimaksudkan untuk memberi landasan sehingga pengembangan perangkat lunak dilakukan secara sistematis dengan sasaran memperoleh produk yang berkualitas [HAR-04].

2.2.2.1 Proses-Proses RPL

Rekayasa perangkat lunak memiliki proses-proses. Proses-proses rekayasa perangkat lunak adalah serangkaian kegiatan dan hasil-hasil relevannya yang menghasilkan perangkat lunak [SOM-04].

Rekayasa perangkat lunak memiliki empat proses dasar [SOM-04], yaitu:

1. Spesifikasi perangkat lunak.
2. Pengembangan perangkat lunak
3. Validasi perangkat lunak
4. Evolusi perangkat lunak.

a. Proses Spesifikasi Perangkat Lunak

Proses spesifikasi perangkat lunak adalah proses yang bertujuan untuk menetapkan layanan apa yang dituntut dari sistem dan batasan pada operasi dan pengembangan sistem [SOM-04]. Proses spesifikasi perangkat lunak sering disebut sebagai rekayasa persyaratan [SOM-04].

b. Proses Pengembangan Perangkat Lunak

Proses pengembangan perangkat lunak adalah proses perubahan spesifikasi sistem menjadi sistem yang dapat dijalankan [SOM-04]. Proses ini selalu mencakup dua hal [SOM-04], yaitu :

1. Perancangan perangkat lunak.
2. Pemrograman perangkat lunak.

Perancangan perangkat lunak adalah deskripsi struktur perangkat lunak yang akan diimplementasikan, data yang merupakan bagian sistem, *interface* antara komponen-komponen sistem, dan kadang-kadang algoritma yang digunakan [SOM-04]. Pemrograman perangkat lunak adalah proses lanjutan dari perancangan perangkat lunak. Pemrograman merubah suatu rancangan menjadi kode-kode program.

c. Proses Validasi Perangkat Lunak

Proses validasi perangkat lunak adalah proses yang ditujukan untuk menunjukkan bahwa sistem sesuai dengan spesifikasi dan bahwa sistem memenuhi harapan pelanggan yang membelinya [SOM-04].

d. Proses Evolusi Perangkat Lunak

Perangkat lunak selalu diharapkan untuk dapat berevolusi dalam menghadapi perubahan kebutuhan sistem di masa-masa yang akan datang. Suatu sistem yang telah ada saat ini mungkin dapat memenuhi kebutuhan pengguna di saat ini, akan tetapi belum tentu sistem tersebut dapat memenuhi kebutuhan pengguna di masa datang.

Proses evolusi adalah suatu proses yang sangat penting. Proses evolusi menyebabkan perangkat lunak menjadi fleksibel dalam menghadapi tantangan perubahan kebutuhan sistem. Menurut [SOM-04], dalam perangkat lunak yang dirancang untuk waktu hidup yang sangat lama, proses evolusi memakan biaya tiga sampai dengan empat kali biaya proses pengembangan sistem.

2.2.2.2 Model Proses Perangkat Lunak

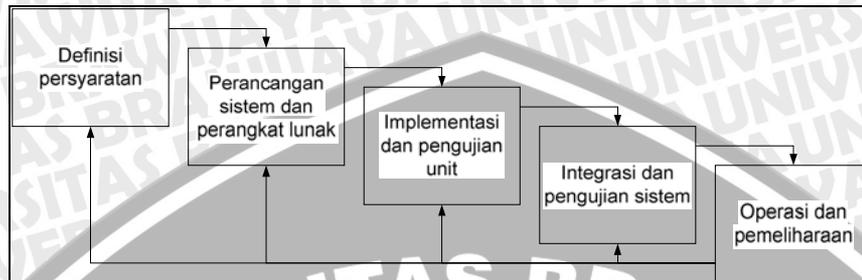
Model proses perangkat lunak adalah representasi abstrak dari proses perangkat lunak [SOM-04]. Salah satu model proses yang sering digunakan adalah model proses air terjun (*waterfall*).

Model proses *waterfall* adalah model yang menyarankan pendekatan pengembangan secara sekuensial dan sistemik untuk mengembangkan perangkat lunak [SOM-04]. Model proses *waterfall* dimulai dari proses analisis dan definisi persyaratan sampai dengan pemeliharaan. Tahap-tahap utama dari model *waterfall* adalah [SOM-04] :

1. Analisis dan definisi persyaratan
2. Perancangan sistem dan perangkat lunak
3. Implementasi dan pengujian unit
4. Integrasi dan pengujian sistem
5. Operasi dan pemeliharaan.

Model *waterfall* dimulai dari proses definisi persyaratan dilanjutkan dengan perancangan sistem dan perangkat lunak, implementasi dan pengujian

unit, integrasi dan pengujian sistem, dan terakhir adalah operasi dan pemeliharaan. Hasil dari setiap proses adalah masukan untuk proses selanjutnya, sehingga suatu proses tidak dapat dimulai bila proses sebelumnya belum selesai. Model *waterfall* ini diperlihatkan pada Gambar 2.1.



Gambar 2.1 : Model Proses *Waterfall*
Sumber : [SOM-04]

2.2.2.3 Paradigma RPL

Paradigma rekayasa perangkat lunak adalah cara pandang spesifik dalam usaha pencarian solusi, yaitu menyangkut konseptualisasi permasalahan dan solusi yang tidak berkaitan dengan urutan aktivitas penyelesaian [HAR-04]. Beberapa paradigma rekayasa perangkat lunak adalah :

1. Paradigma konvensional
2. Paradigma berorientasi objek

a. Paradigma Konvensional

Paradigma konvensional adalah paradigma yang menggunakan strategi dekomposisi berdasarkan algoritma atau fungsional [HAR-04]. Paradigma ini telah meliputi seluruh proses pengembangan perangkat lunak secara lengkap, mulai dari analisis kebutuhan, analisis domain masalah, perancangan, pemrograman, pengujian, metriks, dan sebagainya.

b. Paradigma Berorientasi Objek

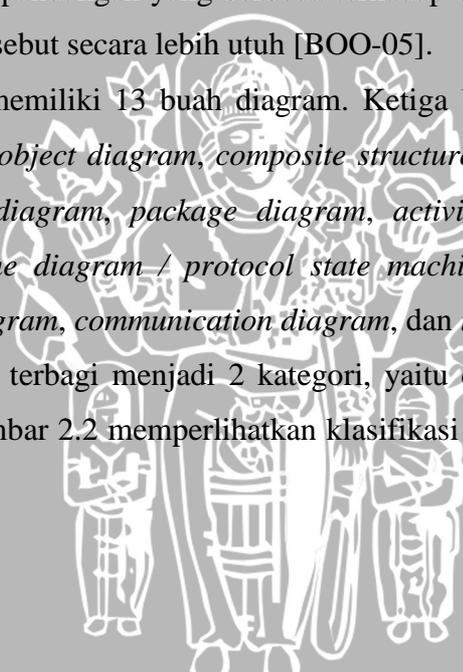
Paradigma rekayasa perangkat lunak adalah cara pandang spesifik dalam usaha pencarian solusi, yaitu menyangkut konseptualisasi permasalahan dan solusi yang tidak berkaitan dengan urutan aktivitas penyelesaian [HAR-04]. Paradigma berorientasi objek adalah paradigma yang berdasarkan pada pola pikir berorientasi objek [HAR-04]. Pola pikir berorientasi objek adalah pola pikir yang memandang segala sesuatu sebagai objek.

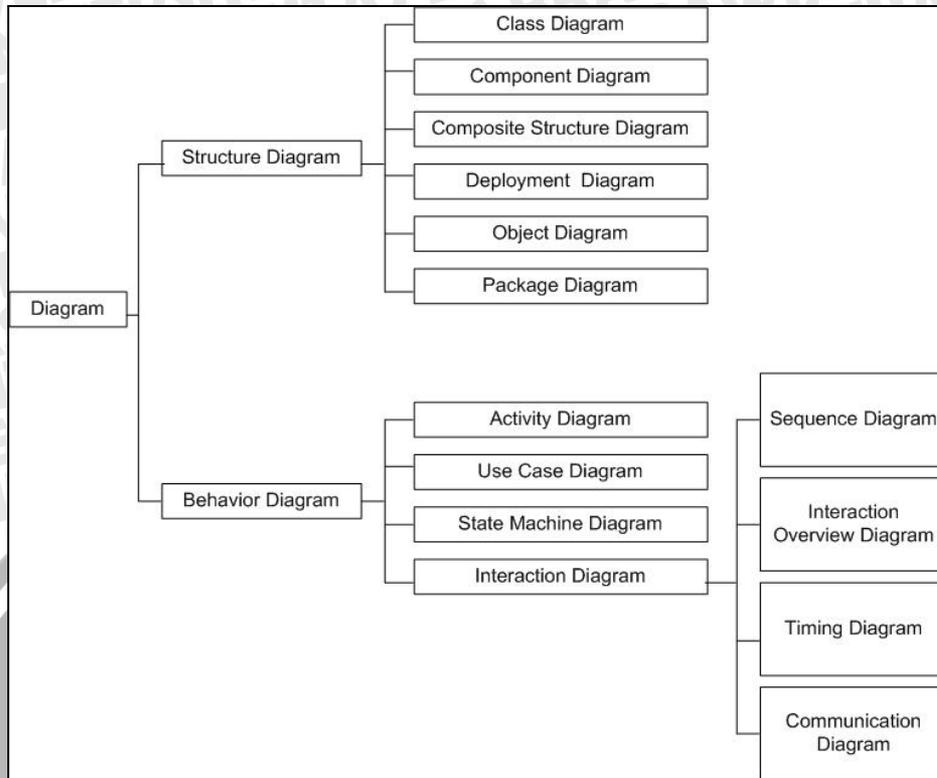
2.2.3 Unified Modeling Language

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang terdiri dari kumpulan diagram yang terintegrasi. UML dikembangkan untuk membantu pengembang perangkat lunak dan pengembang sistem menyelesaikan tugas-tugas, seperti: spesifikasi, visualisasi, desain arsitektural, konstruksi, simulasi, dan dokumentasi [CHO:03].

UML memenuhi sebuah kebutuhan penting dalam pengembangan perangkat lunak dan sistem, yaitu pemodelan. Pemodelan memungkinkan sebuah sistem ataupun sebuah perangkat lunak dapat dipandang dengan berbagai macam cara yang berbeda. Setiap pandangan tersebut adalah berbeda, akan tetapi berbagai macam pandangan tersebut akan lebih memberikan kejelasan akan sistem tersebut. Setiap pandangan yang berbeda terhadap suatu sistem akan lebih menjelaskan sistem tersebut secara lebih utuh [BOO-05].

UML versi 2 memiliki 13 buah diagram. Ketiga belas diagram tersebut adalah *class diagram*, *object diagram*, *composite structure diagram*, *deployment diagram*, *component diagram*, *package diagram*, *activity diagram*, *use case diagram*, *state machine diagram / protocol state machine diagram*, *overview diagram*, *sequence diagram*, *communication diagram*, dan *timing diagram*. Ketiga belas diagram tersebut terbagi menjadi 2 kategori, yaitu diagram struktural dan diagram *behavior*. Gambar 2.2 memperlihatkan klasifikasi diagram-diagram yang dimiliki UML.





Gambar 2.2 : Klasifikasi Jenis Diagram UML Versi 2
Sumber : [FOW-04]

Diagram struktural dipergunakan untuk menunjukkan blok bangunan dari sistem, yaitu fitur-fitur yang tidak berubah seiring dengan waktu. Diagram *behavior* dipergunakan untuk menunjukkan bagaimana sistem merespon sesuatu seiring dengan waktu. Diagram interaksi adalah salah satu tipe dari diagram *behavior* yang dipergunakan untuk melukiskan pertukaran pesan dalam sebuah kolaborasi objek-objek.

Ketiga belas diagram UML dapat dirangkum kegunaannya dalam Tabel 2.1.

Tabel 2.1 : Diagram UML

No	Kategori	Nama Diagram	Keterangan
1	<i>Structure diagram</i>	<i>Class diagram</i>	<i>Class diagram</i> menunjukkan entitas dari dunia nyata, elemen-elemen dari analisis dan desain, atau klas-klas hasil implementasi dan hubungannya.
2	<i>Structure diagram</i>	<i>Object diagram</i>	<i>Object diagram</i> menunjukkan contoh spesifik atau contoh

			ilustratif dari objek-objek dan hubungan yang mereka miliki.
3	<i>Structure diagram</i>	<i>Composite structure diagram</i>	<i>Composite structure diagram</i> menunjukkan bagaimana sesuatu dibuat.
4	<i>Structure diagram</i>	<i>Deployment diagram</i>	<i>Deployment diagram</i> menunjukkan arsitektur <i>run-time</i> dari sistem, <i>platforms</i> dari perangkat keras, lingkungan perangkat lunak (seperti sistem operasi dan <i>virtual machines</i>).
5	<i>Structure diagram</i>	<i>Component diagram</i>	<i>Component diagram</i> menunjukkan organisasi dan hubungan diantara sistem.
6	<i>Structure diagram</i>	<i>Package diagram</i>	<i>Package diagram</i> dipergunakan untuk mengorganisasi elemen-elemen model dan menunjukkan ketergantungan diantara mereka.
7	<i>Behavioral diagram</i>	<i>Activity diagram</i>	<i>Activity diagram</i> menunjukkan aliran data dan atau aliran kontrol dari objek-objek yang saling bekerja sama.
8	<i>Behavioral diagram</i>	<i>Use case diagram</i>	<i>Use case diagram</i> menunjukkan jenis-jenis layanan yang dapat diminta oleh aktor dari sistem.
9	<i>Behavioral diagram</i>	<i>State machine diagram / Protocol state machine diagram</i>	<i>State machine diagram / Protocol state machine diagram</i> menunjukkan daur hidup suatu objek khusus atau rangkaian dari sebuah objek yang harus didukung oleh <i>interface</i> .
10	<i>Interaction diagram</i>	<i>Overview diagram</i>	<i>Overview diagram</i> menunjukkan beberapa skenario interaksi dari suatu kolaborasi.
11	<i>Interaction diagram</i>	<i>Sequence diagram</i>	<i>Sequence diagram</i> dipergunakan untuk memfokuskan pada pertukaran pesan antara suatu grup objek dan urutan dari pesan-pesan tersebut.

12	<i>Interaction diagram</i>	<i>Communication diagram</i>	<i>Communication diagram</i> dipergunakan untuk memfokuskan pada pesan-pesan antara sebuah grup dari objek-objek dan hubungan yang mendasari objek-objek.
13	<i>Interaction diagram</i>	<i>Timing diagram</i>	<i>Timing diagram</i> menunjukkan perubahan dan hubungan mereka dalam <i>real-time system</i> atau <i>embedded system</i> .

Sumber : [CHO-03]

Beberapa diagram UML memiliki tingkat kegunaan yang lebih dibandingkan dengan diagram UML yang lainnya, hal ini tergantung pada sifat dari sistem yang akan dibangun [BOO-05]. Beberapa diagram UML yang paling berguna dalam pengembangan sistem adalah [BEL-03] :

1. Diagram *use case*
2. Diagram Klas
3. Diagram *Sequence*

2.2.3.1 Diagram Use Case

Diagram *use case* adalah suatu cara untuk menangkap fungsionalitas dan kebutuhan dari sistem dalam UML [PIL-05]. Diagram *use case* menekankan “apa” yang dapat diperbuat sistem, bukan “bagaimana” hal tersebut dilakukan. Diagram *use case* sangat membantu dalam penyusunan kebutuhan sebuah sistem, mengkomunikasikan rancangan dengan pengguna sistem, dan merancang pengujian untuk semua fitur yang ada pada sistem.

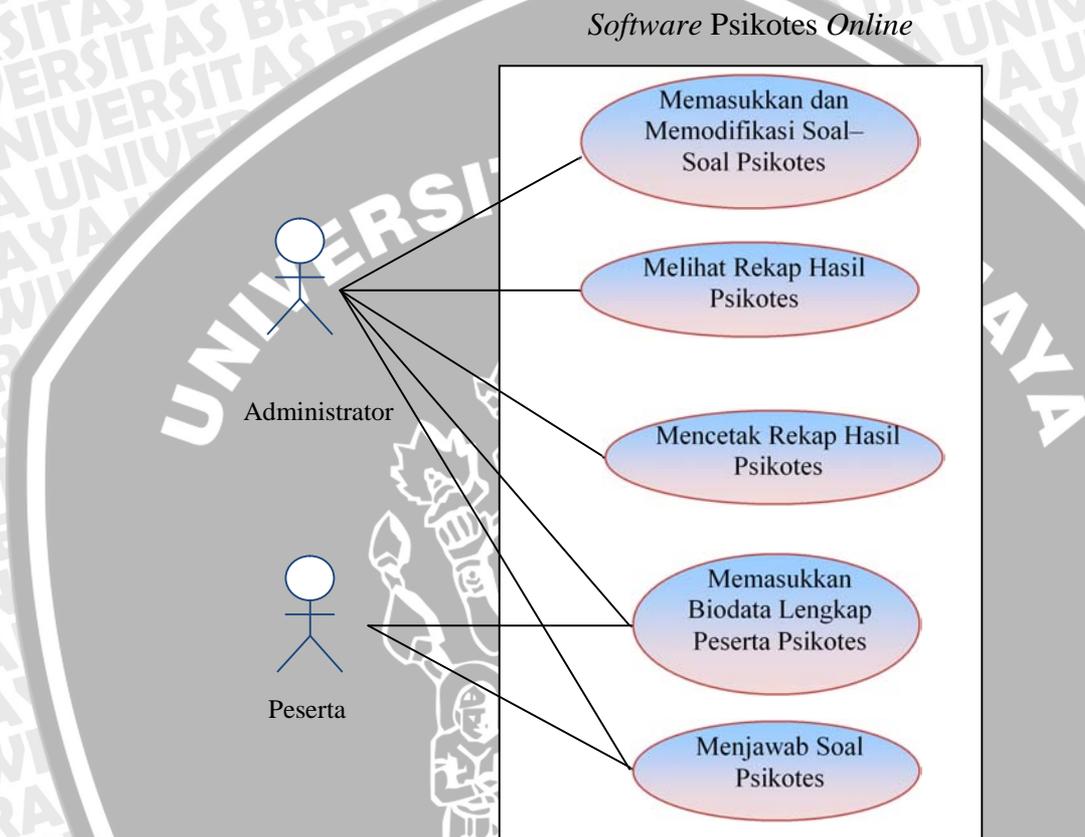
Tiga konsep utama yang berhubungan dengan *use case diagram* adalah [OMG-04] :

1. Aktor
2. *Use case*
3. Subjek

Subjek adalah suatu sistem yang menjadi perhatian dimana *use case* diterapkan [OMG-04]. Selain ketiga konsep utama di atas, terdapat konsep hubungan (*relationship*).

Gambar 2.3 adalah contoh dari diagram *use case*. Contoh ini

menggambarkan dua buah aktor (Peserta dan Administrator) dan lima buah *use case* (Memasukkan dan memodifikasi soal psikotes, melihat rekap hasil psikotes, mencetak rekap hasil psikotes, memasukkan biodata peserta dan menjawab soal psikotes). Aktor peserta dapat mengakses *use case* memasukkan biodata peserta dan menjawab soal psikotes. Aktor Administrator dapat mengakses semua *use case*. Subjek dari contoh ini adalah *software* psikotes *online*.



Gambar 2.3 : Contoh *Use Case* Diagram

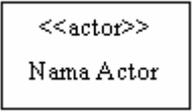
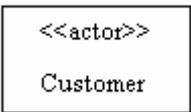
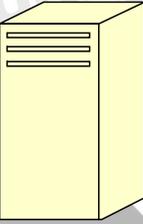
a) Aktor

Aktor adalah pengguna sistem ataupun sistem lain yang berinteraksi dengan subjek [OMG-04]. Aktor memodelkan entitas yang berada di luar subjek [OMG-04]. Aktor dapat berupa manusia, perangkat keras ataupun sistem (subjek lain [OMG-04]).

Aktor dapat dinotasikan dengan 3 cara [OMG-04]:

Tabel 2.2 : Notasi aktor

No	Cara	Notasi	Keterangan	Contoh
1	<i>Stick man</i>		Aktor dinotasikan dengan " <i>stick man</i> ".	

		 Nama actor	Nama aktor biasanya diletakkan di bawah atau di atas “ <i>stick man</i> ”.	 Customer
2	<i>Class rectangle</i>	 Nama Actor	Aktor dinotasikan dengan kotak klas, dengan kata kunci <<actor>>.	 Customer
3	<i>Icon</i>		<i>Icon</i> dapat digunakan sebagai penotasian aktor, misalnya bila aktor adalah suatu perangkat keras.	 Database Server

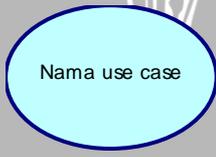
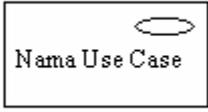
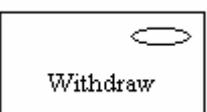
Sumber : [OMG-04]

b) Use Case

Use case adalah perilaku atau fitur dari subjek yang berusaha untuk dimodelkan [OMG-04]. *Use case* merepresentasikan fungsionalitas unik dari sistem [PIL-05]. Setiap *use case* menyatakan suatu perilaku yang dapat dilakukan oleh subjek dalam kolaborasinya dengan satu atau lebih aktor [OMG-04].

Use case dapat dinotasikan dengan 2 cara [OMG-04] :

Tabel 2.3 : Notasi *Use Case*

No	Cara	Notasi	Keterangan	Contoh
1	<i>Ellipse</i>	 Nama use case	<i>Use case</i> dapat dinotasikan dengan gambar elips, dengan nama <i>use case</i> berada di dalam elips ataupun di bawah elips.	 Withdraw
2	<i>Class rectangle</i>	 Nama Use Case	<i>Use case</i> dapat pula dinotasikan dengan kotak klas dengan <i>icon</i> elips di pojok kanan atas.	 Withdraw

Sumber : [OMG-04]

Perilaku suatu *use case* secara detail dapat dideskripsikan dengan suatu teks



document ataupun dengan diagram UML lainnya (*interaction diagram, activity diagram, state machines diagram*) [OMG-04].

c) Relationship

Relationship dalam diagram *use case* menyatakan hubungan yang terjadi antara suatu aktor dengan aktor lainnya, suatu aktor dengan *use case*, atau suatu *use case* dengan *use case* lainnya. Beberapa *relationship* di dalam *use case* diagram adalah :

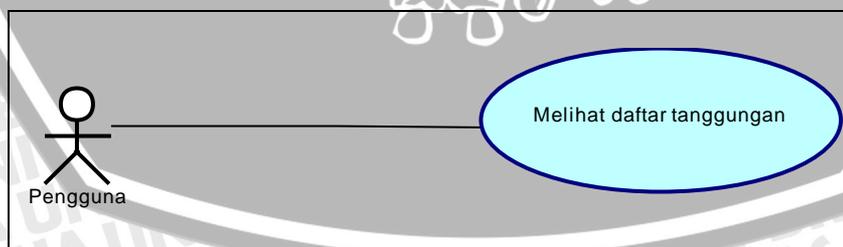
1. Asosiasi antara aktor dengan *use case*.
2. Generalisasi dari aktor.
3. Generalisasi dari *use case*.
4. *Include* antara dua buah *use case*.
5. *Extend* antara dua buah *use case*.

1. Asosiasi antara Aktor dengan *use case*

Asosiasi antara aktor dengan *use case* menyatakan salah satu dari hal berikut [PIL-05] :

1. Aktor dapat memulai suatu *use case*.
2. Suatu *use case* menyediakan sesuatu untuk suatu aktor.
3. Aktor dapat memulai suatu *use case* dan *use case* tersebut menyediakan atau menghasilkan sesuatu untuk aktor tersebut.

Asosiasi antara aktor dengan *use case* dinotasikan dengan garis lurus dari aktor ke *use case*. Gambar 2.4 adalah contoh asosiasi antara aktor dengan *use case*.



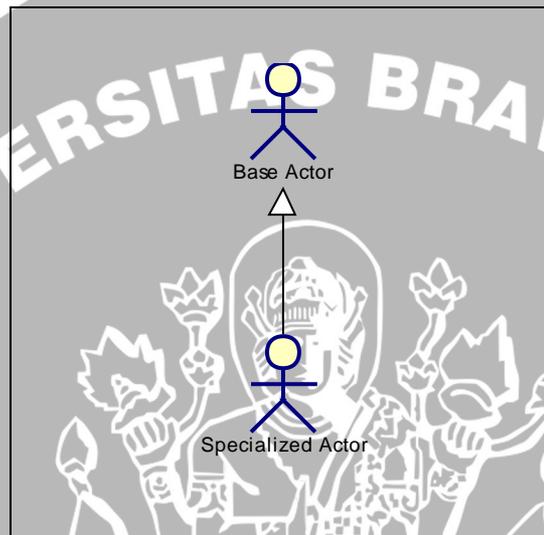
Gambar 2.4 : Asosiasi antara aktor dengan *use case*

Sebuah aktor dapat diasosiasikan dengan lebih dari satu *use case* dalam sebuah diagram *use case*. Ketika satu aktor diasosiasikan dengan lebih dari satu *use case*, maka aktor tersebut dimaksudkan dapat menggunakan semua *use case*-

use case tersebut [OMG-04].

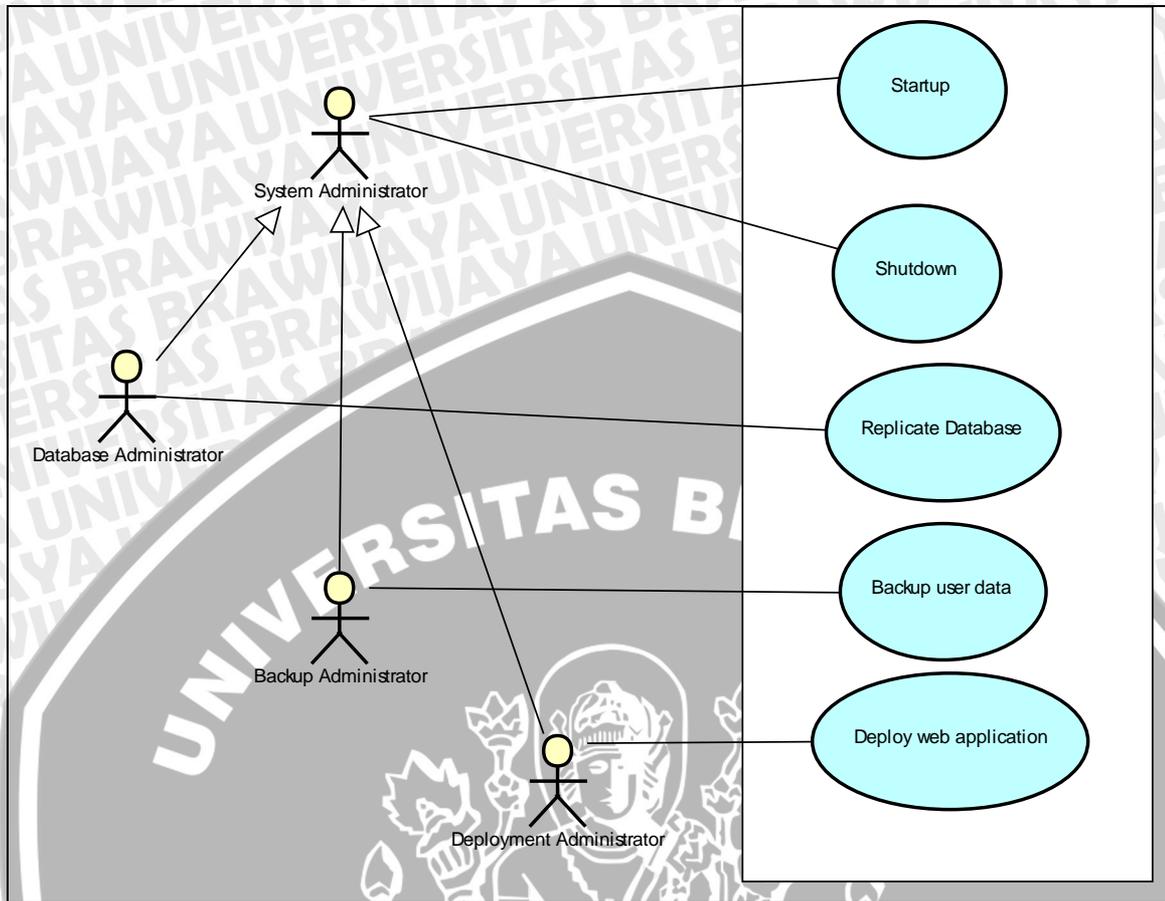
2. Generalisasi dari Aktor

Suatu aktor dapat digeneralisasi. Pengertian generalisasi dari aktor adalah aktor yang memiliki sifat-sifat umum dari aktor yang lainnya. Generalisasi aktor dinotasikan dengan garis lurus dengan kepala panah tertutup dimana kepala panah menunjuk ke aktor yang lebih umum [PIL-05]. Gambar 2.5 menyatakan generalisasi dari aktor.



Gambar 2.5 : Generalisasi dari aktor

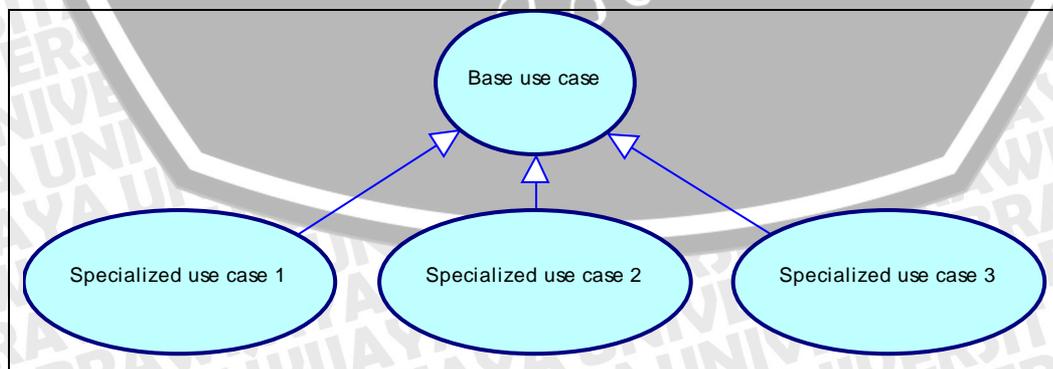
Gambar 2.5 memberikan contoh dari generalisasi aktor. Diagram berikut memiliki empat aktor, yaitu *System Administrator*, *Database Administrator*, *Backup Administrator*, dan *Deployment Administrator*. Aktor umum dari diagram berikut adalah *System Administrator*. Aktor selain *System Administrator* akan memiliki sifat-sifat dari *System Administrator* ditambah dengan sifat khusus yang ada dalam aktor itu sendiri.



Gambar 2.6 : Contoh generalisasi aktor
Sumber : [PIL-05]

3. Generalisasi dari Use Case

Use case dapat digeneralisasi. Pengertian generalisasi dari *use case* adalah *use case* yang memiliki sifat-sifat umum dari *use case* lainnya. *Use case generalisasi* dinotasikan sebagai anak panah dengan kepala panah tertutup, dimana kepala panah menunjuk ke *use case* yang lebih umum [PIL-05].

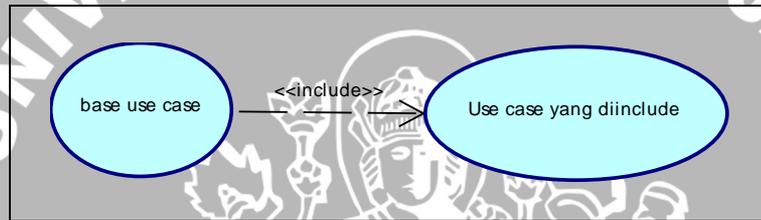


Gambar 2.7 : Generalisasi dari *use case*
Sumber : [PIL-05]

4. Include antara Dua Buah Use Case

Include mendefinisikan bahwa sebuah *use case* mengandung perilaku yang didefinisikan di dalam *use case* lainnya [OMG-04]. Pengertian mengandung perilaku adalah *use case* yang menginclude (*base use case*) pasti menjalankan *use case* yang diinclude. *Base use case* adalah *use case* yang tidak lengkap, dan membutuhkan *use case* yang diincludeny. Tujuan dari *include* adalah untuk guna ulang, sehingga lebih dari satu buah *use case* dapat berbagi sebuah *use case* yang sama.

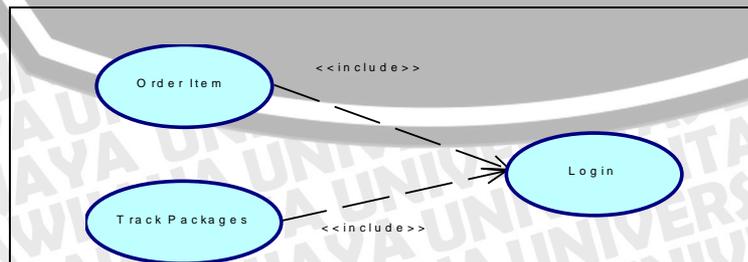
Include dinotasikan dengan anak panah dengan garis putus-putus dengan kepala panah terbuka, dimana anak panah menunjuk ke *use case* yang diinclude, dan pada badan panah diberi keyword <<include>> [OMG-04].



Gambar 2.8 : Notasi <<include>>

Sumber : [PIL-05]

Contoh dari *include* pada *use case* diperlihatkan pada Gambar 2.9. Gambar tersebut mengandung dua buah *base use case*, yaitu Order Item dan Track Packages. Kedua *use case* tersebut menginclude *use case* Login, yang berarti *use case* Order Item dan Track Packages membutuhkan *use case* Login agar kedua *use case* tersebut (*use case* Order Item dan *use case* Track Package) dapat berjalan. Arah panah dari *use case* Order Item dan *use case* Track Package menuju *use case* Login, hal ini menunjukkan bahwa *use case* Order Item dan *use case* Track Package adalah *base use case*.



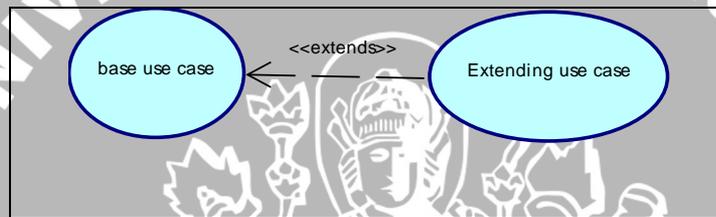
Gambar 2.9 : Contoh *include* antara dua buah *use case*

Sumber : [PIL-05]

5. Extend antara Dua Buah Use Case

Hubungan ini menyatakan bahwa perilaku dari sebuah *use case* dapat diextend (diperpanjang/diperluas) oleh perilaku dari *use case* yang lain [OMG-04]. *Extended use case (base use case)* adalah *independen* (tidak tergantung) pada *extending use case* [OMG-04]. *Extending use case* tidak berarti bila berdiri sendiri [OMG-04]. *Extending use case* memberikan fungsi tambahan terhadap *base use case* [OMG-04]. *Extending use case* dapat dijalankan ataupun tidak dijalankan, tergantung apakah suatu kondisi tercapai pada *base use case*.

Extend di notasikan dengan anak panah dengan garis putus-putus dengan kepala panah terbuka, panah menunjuk dari *extending use case* ke *base use case*, pada badan panah terdapat *keyword* <<extends>>.



Gambar 2.10 : Notasi <<extend>>

Sumber : [PIL-05]

2.2.3.2 Diagram Klas

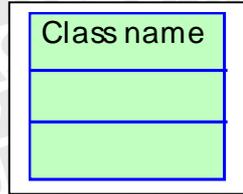
Diagram klas adalah diagram dalam UML yang digunakan untuk menangkap hubungan statis dari suatu perangkat lunak [PIL-05]. Diagram klas memiliki beberapa hal penting, yaitu klas dan *relationship* [BOO-05].

a) Klas

Klas adalah kumpulan objek-objek yang memiliki spesifikasi fitur, batasan dan semantik yang sama [OMG-04]. Fitur dari klas ada 2, yaitu [OMG-04]:

1. Atribut
2. Operasi

Klas dinotasikan dengan gambar kotak, nama klas diletakkan di tengah kotak. Gambar 2.11 menyatakan pernotasian sebuah klas.



Gambar 2.11 : Notasi klas
Sumber : [PIL-05]

Penamaan klas dapat dilakukan dengan dua cara, yaitu :

1. *Simple name*
2. *Qualified name*

Qualified name adalah nama klas yang didahului dengan nama paket tempat klas tersebut berada, antara nama-nama paket dan nama klas dipisahkan dengan dua buah titik dua. *Simple name* adalah nama klas yang tanpa didahului dengan nama paket tempat klas tersebut berada.

Klas dapat memiliki beberapa *compartment*. *Compartment* adalah daerah dalam notasi klas yang dipisahkan oleh garis horisontal. Notasi klas yang umum memiliki tiga buah *compartment*, yaitu :

1. *Compartment* pertama berisi nama klas [PIL-05]. *Compartment* ini sifatnya wajib.
2. *Compartment* kedua berisi daftar atribut [PIL-05]. *Compartment* ini sifatnya opsional.
3. *Compartment* ketiga berisi daftar operasi [PIL-05]. *Compartment* ini sifatnya opsional.

Gambar 2.12 menunjukkan notasi klas dengan tiga buah *compartment*, yaitu *compartment* nama klas, *compartment* daftar atribut dan *compartment* daftar operasi. Mobil adalah nama dari klas. Klas ini memiliki dua buah atribut, yaitu warna dan tahunKeluaran, dan memiliki empat buah operasi, yaitu maju, mundur, belokKanan, dan belokKiri.

Mobil	
- warna	: string
- tahunKeluaran	: date
<hr/>	
+ maju ()	: int
+ mundur ()	: int
+ belokKanan ()	: int
+ belokKiri ()	: int

Gambar 2.12 : Contoh notasi klas dengan tiga buah *compartment*
Sumber : [PIL-05]

Penotasian sebuah klas dapat tidak menampilkan *compartment* atribut atau *compartment* operasi. Tidak adanya suatu *compartment* dalam notasi klas tidak dapat diasumsi bahwa klas tersebut tidak memiliki atribut atau operasi. Peniadaan *compartment* dimaksudkan untuk mempermudah pembacaan suatu klas. [PIL-05]

1. Atribut

Atribut adalah properti dari suatu klas yang mendeskripsikan suatu rentang nilai yang mungkin dimiliki oleh properti dari objek tersebut [BOO-05]. Atribut dapat dinotasikan dengan 2 cara, yaitu [PIL-05]:

1. *Inlined attribute*.
2. *Relationship* antar klas.

Inlined attribute adalah atribut yang dituliskan dalam *compartment* daftar atribut. Berikut ini adalah sintak dari penulisan *inlined attribute* :

visibility namaAtribut : type multiplicity = default Value

Gambar 2.13 adalah contoh dari *inlined attribute*. Notasi klas pada Gambar 2.13 memiliki tiga buah *inlined attribute*, yaitu nim, nama, dan jenisKelamin.

Mahasiswa	
- nim	: string
- nama	: string
- jenisKelamin	: string = pria
<hr/>	

Gambar 2.13 : Contoh notasi klas dengan *inlined attribute*
Sumber : [PIL-05]

Visibility dari *inlined attribute* menyatakan kenampakan atribut objek tersebut dari objek-objek lainnya. Berikut ini adalah contoh *visibility* dari *inlined attribute* :

- + nama : String
- alamat : String
- # jenisKelamin : String
- ~ nim : Integer

namaAtribut dari *inlined attribute* adalah nama dari atribut. namaAtribut adalah kata benda. Huruf pertama namaAtribut biasanya adalah huruf kecil dan setiap huruf pertama dari kata selanjutnya biasanya adalah huruf besar. Contoh :

1. jenisKelamin
2. luas
3. nilaiAkhirMataKuliah

type dari *inlined attribute* adalah tipe data yang dapat disimpan pada atribut tersebut. *Type* dapat berupa :

1. Nama kelas
2. Nama *interface*
3. Tipe data *built-in*

multiplicity dari *inlined attribute* menyatakan berapa banyak nilai yang dapat ditampung oleh suatu variabel. *multiplicity* diletakkan di dalam pasangan tanda kurung siku “[]”. *multiplicity* dapat dituliskan dalam 3 bentuk, yaitu :

Tabel 2.4 : Bentuk-bentuk *multiplicity*

No	Multiplicity	Keterangan
1	[x]	Menyatakan bahwa variabel tersebut dapat menyimpan tepat x buah nilai. x dapat bernilai *, yang berarti variabel tersebut dianggap dapat menyimpan nol sampai dengan tak terhingga buah nilai.
2	[x,y,z]	Menyatakan bahwa variabel tersebut dapat menyimpan x atau y atau z buah nilai.
3	[x..y]	Menyatakan bahwa variabel tersebut dapat menyimpan x sampai dengan y buah nilai.

Sumber : [CHO-03]

Bila di dalam penulisan atribut tidak disebutkan *multiplicity*, maka variabel tersebut akan dianggap memiliki *multiplicity* 1.

defaultValue dari *inlined attribute* adalah nilai default untuk variabel



tersebut. defaultValue harus cocok dengan *type* dari atribut tersebut [CHO-03].

Penjelasan tentang atribut menggunakan *relationship* dapat dilihat pada penjelasan *relationship* antar klas.

2. Operasi

Operasi adalah fitur dari klas yang digunakan untuk menyatakan bagaimana menjalankan suatu perilaku [PIL-05]. Daftar operasi diletakkan di dalam *compartment* operasi dari suatu klas. Sintak penulisan operasi adalah sebagai berikut : visibility name (parameters) : return-type.

visibility dari operasi memiliki kesamaan dengan visibility dari atribut. name adalah nama dari operasi. name pada operasi biasanya berupa kata kerja. Huruf pertama dari nama operasi direkomendasikan oleh UML agar huruf kecil, sedangkan huruf pertama dari kata operasi berikutnya disarankan berupa huruf besar. return-type dari operasi menyatakan tipe data kembalian dari operasi tersebut.

b) Relationship

Relationship adalah hubungan antar klas yang satu dengan klas yang lain dalam diagram klas [BOO-05]. UML menyediakan beberapa cara untuk merepresentasikan *relationship* antar klas. Beberapa jenis *relationship* dalam diagram klas adalah :

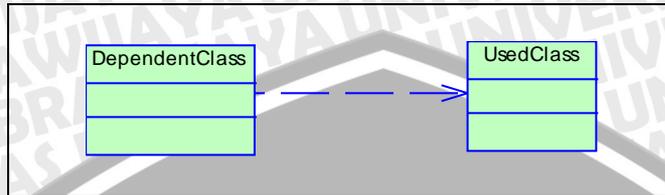
1. Dependensi
2. Asosiasi
3. Agregasi
4. Komposisi
5. Generalisasi

1. Dependensi

Dependensi adalah *relationship* yang menyatakan bahwa suatu klas menggunakan informasi dan layanan dari klas lain, tapi tidak harus sebaliknya [BOO-05]. Dependensi sering digunakan untuk menunjukkan keadaan ketika suatu klas menggunakan operasi atau variabel dari klas lain. Hubungan dependensi sering dibaca dengan "... use a ..." [PIL-05]. Hubungan dependensi

adalah hubungan paling umum yang dapat terjadi antar kelas [PIL-05].

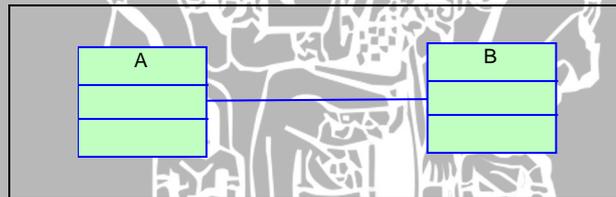
Dependensi dinotasikan sebagai anak panah dengan badan panah berupa garis putus-putus, kepala panah terbuka dan menunjuk dari kelas *dependent* ke kelas yang digunakan.



Gambar 2.14 : Notasi dependensi
Sumber : [PIL-05]

2. Asosiasi

Asosiasi adalah *structural relationship* yang menyatakan bahwa objek dari suatu kelas terhubung dengan objek dari kelas lain [BOO-05]. Hubungan asosiasi sering dibaca dengan “... *has a* ...” [PIL-05]. Hubungan asosiasi dianggap lebih khusus bila dibandingkan dengan hubungan dependensi [PIL-05]. Asosiasi dinotasikan sebagai garis lurus antara dua kelas yang saling berasosiasi. Gambar 2.15 adalah penotasian asosiasi.



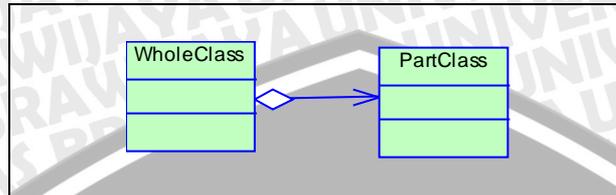
Gambar 2.15 : Notasi asosiasi
Sumber : [PIL-05]

3. Agregasi

Agregasi adalah *relationship* yang menyatakan bahwa suatu kelas mengandung kelas lainnya. Hubungan agregasi sering dibaca dengan “... *owns a* ...” [PIL-05]. Agregasi adalah jenis khusus dari asosiasi [BOO-05]. Hubungan agregasi dianggap lebih kuat bila dibandingkan dengan hubungan asosiasi [PIL-05].

Perbedaan antara agregasi dengan asosiasi adalah pada kepemilikan. Ketika dua buah objek saling terhubung, maka hubungan keduanya dapat disebut sebagai asosiasi. Pada agregasi, hubungan antara dua kelas lebih pada suatu kelas memiliki atau tersusun dari kelas yang lainnya.

Agregasi dinotasikan sama seperti asosiasi, yaitu berupa garis lurus, hanya pada salah satu ujung garis lurus tersebut berupa gambar belah ketupat [BOO-05]. Gambar belah ketupat diletakkan pada sisi kelas yang dianggap memiliki kelas lainnya.



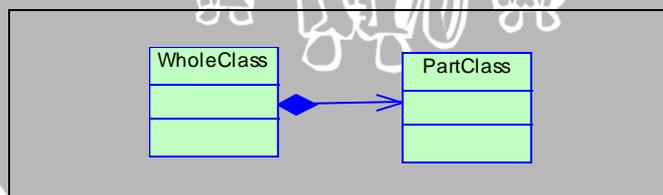
Gambar 2.16 : Notasi agregasi
Sumber : [PIL-05]

4. Komposisi

Komposisi adalah suatu *relationship* yang menyatakan hubungan *whole-part*. Hubungan komposisi sering dibaca dengan “... is part of ...” [PIL-05]. Komposisi adalah jenis khusus dari agregasi. Komposisi dianggap memiliki hubungan yang lebih kuat bila dibanding dengan agregasi.

Perbedaan antara komposisi dan agregasi terletak pada objek yang dimiliki. Pada agregasi, bila suatu objek A dimiliki oleh objek B, maka objek C dapat pula memiliki objek A tersebut. Sedangkan pada komposisi tidak demikian, bila objek A dimiliki oleh objek B, maka hanya objek B saja yang dapat memiliki objek A tersebut.

Komposisi dinotasikan sama seperti agregasi, perbedaannya adalah bila belah ketupat pada agregasi tidak diarsir, maka pada komposisi, belah ketupat tersebut diarsir.



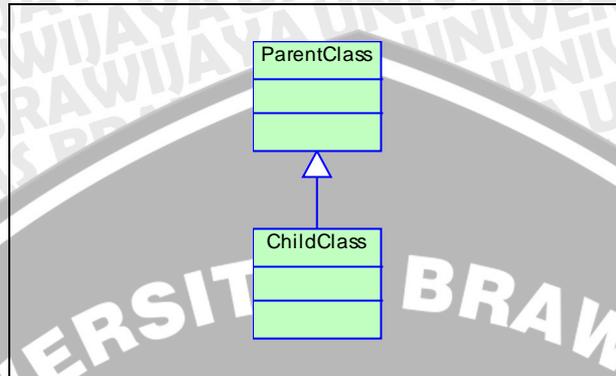
Gambar 2.17 : Notasi komposisi
Sumber : [PIL-05]

5. Generalisasi

Generalisasi adalah *relationship* yang menyatakan hubungan antara suatu kelas yang lebih umum dengan kelas yang lebih khusus. Hubungan generalisasi sering dianggap sebagai hubungan antara kelas orang tua dengan kelas anak.

Hubungan generalisasi sering dibaca dengan “... is a ... “ [PIL-05].

Generalisasi dinotaskan sebagai anak panah dengan badan anak panah lurus, dan kepala panah tertutup menunjuk dari kelas yang lebih khusus (klas anak) ke kelas yang lebih umum (klas orang tua).



Gambar 2.18 : Notasi *Generalisasi*

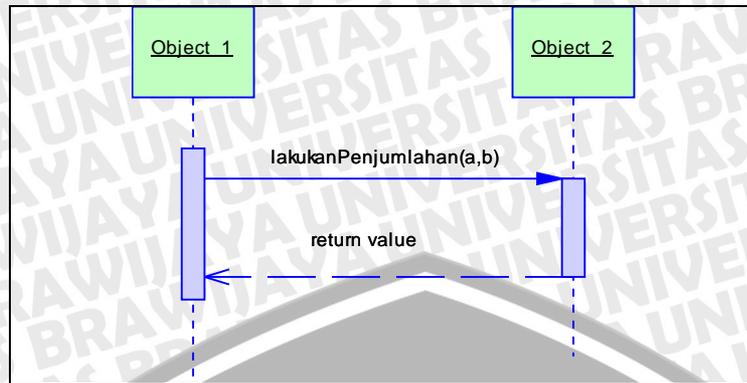
Sumber : [PIL-05]

2.2.3.3 Diagram *Sequence*

Diagram *sequence* adalah diagram interaksi yang menitik-beratkan pada urutan waktu dari suatu *messages* [BOO-05]. *Sequence diagram* memiliki beberapa konsep penting, diantaranya adalah :

1. *Lifelines*
2. *Message*

Gambar 2.19 adalah contoh dari sebuah diagram *sequence*. Diagram *sequence* ini memiliki dua buah *lifelines* yang berupa objek, yaitu Client dan Penghitung. Penghitung memiliki operasi lakukanPenjumlahan, dan operasi tersebut dijalankan oleh *lifelines* Client, setelah operasi lakukanPenjumlahan(a,b) dikerjakan oleh Penghitung, maka hasil dari operasi tersebut dikembalikan ke Client.

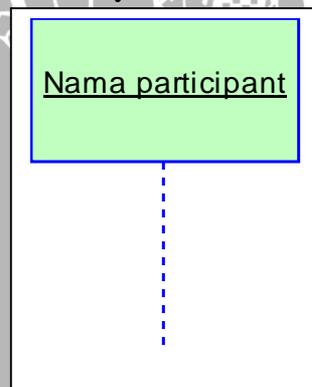


Gambar 2.19 : Contoh *diagram sequence*
Sumber : [PIL-05]

a) Lifelines

Lifelines merepresentasikan partisipan tunggal di dalam interaksi [OMG-04]. Partisipan disini biasanya adalah sebuah objek ataupun sebuah kelas yang ikut dalam suatu interaksi yang hendak ditangkap oleh *diagram sequence*.

Lifelines dinotasikan dengan gambar persegi yang memiliki garis titik-titik yang turun dari kotak [PIL-05]. Nama partisipan diletakkan di dalam kotak. Garis titik-titik yang turun dari gambar persegi menunjukkan berapa lama suatu objek atau kelas 'hidup' [PIL-05] dalam suatu interaksi yang hendak ditangkap oleh *diagram sequenc*. Gambar 2.20 menyatakan notasi sebuah *lifelines*.



Gambar 2.20 : Notasi *lifelines*
Sumber : [PIL-05]

b) Message

Message menyatakan suatu komunikasi yang terjadi antara *lifelines* dalam suatu interaksi [OMG-04]. Contoh komunikasi antar *lifeline* diantaranya adalah pemanggilan *method*, pengiriman sinyal, pembuatan sebuah objek, penghapusan sebuah objek, dan lain sebagainya. Penggunaan paling umum dari *message* adalah untuk menunjukkan pemanggilan *method* antara dua buah objek [PIL-05].



Beberapa jenis *message* adalah :

1. *Synchronous message*
2. *Return message*
3. *Self message*

Message dinotasikan sebagai gambar anak panah, tetapi bentuk badan anak panah, kepala panah dan arah anak panah ditentukan oleh jenis *message*.

Sedangkan sintak dari *message* diletakkan di atas badan panah.

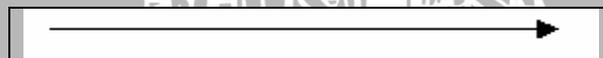


Gambar 2.21 : Notasi *message*
Sumber : [PIL-05]

1. *Synchronous Message*

Synchronous message adalah *message* yang dikirim oleh suatu *lifeline* ke *lifeline* yang lainnya, dimana *lifeline* pengirim menunggu *lifeline* penerima untuk menyelesaikan *message* yang diberikan kepadanya.

Synchronous message dinotasikan dengan gambar anak panah dengan badan anak panah garis lurus, kepala panah terisi dan tertutup, arah anak panah mengarah dari *lifeline* pengirim ke *lifeline* penerima *message*. Gambar 2.22 menyatakan penotasian sebuah *synchronous message*.



Gambar 2.22 : Notasi *synchronous message*
Sumber : [PIL-05]

2. *Return Message*

Pemanggilan *method* dapat mengirimkan suatu nilai kembalian. Nilai kembalian ini yang disebut dengan *return message*.

Return message dinotasikan dengan gambar anak panah, badan panah berupa *dashed line*, kepala panah terbuka, arah panah dari *lifeline* yang dipanggil *methodnya* menuju ke *lifeline* yang memanggil *method* tersebut [PIL-05].



Gambar 2.23 : Notasi *return message*
Sumber : [PIL-05]

3. Self Message

Self message adalah pemanggilan *method* pada objek yang sama [SPA-06]. *Self message* terjadi bila suatu objek memanggil *method* yang ada pada dirinya sendiri.

2.2.4 Pengujian (*Testing*)

Definisi pengujian (*testing*) menurut IEEE 610.12-1990 adalah proses menjalankan suatu sistem atau komponen dalam suatu kondisi tertentu, mengamati atau mencatat hasil, dan membuat evaluasi terhadap sistem atau komponen [IEE-03]. Pengujian pada perangkat lunak merupakan elemen yang sangat penting untuk menjamin kualitas perangkat lunak, serta menggambarkan peninjauan terhadap spesifikasi, desain dan *coding*. Pengujian dapat pula menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang melandasinya.

2.2.4.1 Level Pengujian

Pengujian memiliki 4 level, yaitu [COP-04]:

1. Pengujian unit (*Unit testing*)
2. Pengujian integrasi (*Integration testing*)
3. Pengujian sistem (*System testing*)
4. Pengujian penerimaan (*Acceptance test*)

Pengujian seharusnya dimulai dari level pengujian paling kecil (pengujian unit) menuju level pengujian paling tinggi (pengujian penerimaan).

a. Pengujian Unit

Unit adalah bagian terkecil dari suatu perangkat lunak yang mungkin dapat dibuat [COP-04]. Unit biasanya adalah hasil kerja dari satu orang programmer. Setiap bahasa pemrograman memiliki unit yang berbeda-beda. Unit dalam bahasa pemrograman java dan C++ adalah kelas, sedangkan pada bahasa pemrograman C adalah fungsi [COP-04].

b. Pengujian Integrasi

Pengujian integrasi adalah pengujian yang dilakukan pada beberapa unit yang telah digabungkan. Pengujian ini dilakukan untuk melihat apakah beberapa unit yang saling digabung menghasilkan sesuatu yang diharapkan atau tidak.

Suatu unit dapat lolos dalam pengujian unit, akan tetapi unit tersebut dapat tidak lolos dalam pengujian integrasi.

c. Pengujian Sistem

Pengujian sistem adalah pengujian yang dilakukan untuk menguji kebenaran beberapa subsistem yang saling digabung. Salah satu bentuk pengujian sistem yang sangat penting adalah pengujian *use case* (*use case testing*), hal ini karena *use case* menggambarkan secara lengkap tentang apa yang dapat dikerjakan suatu sistem [DOC-05].

d. Pengujian Penerimaan

Pengujian penerimaan dilakukan oleh pengguna perangkat lunak. Dalam pengujian penerimaan, pengguna menjalankan perangkat lunak untuk mengetahui apakah mereka menerima perangkat lunak itu sebelum perangkat lunak tersebut direlease [DOC-05].

2.2.4.2 Jenis Pengujian

Terdapat 3 jenis pengujian [COP-04], yaitu :

1. Pengujian kotak hitam (*Black box testing*)
2. Pengujian kotak putih (*White box testing*)
3. Pengujian kotak abu-abu (*Gray box testing*)

Pengujian kotak hitam adalah suatu strategi pengujian yang hanya berdasarkan pada spesifikasi dan kebutuhan [COP-04]. Pengujian kotak hitam tidak membutuhkan pengetahuan tentang jalur, struktur atau implementasi dari perangkat lunak yang diuji [COP-04]. Pengujian kotak hitam dapat dilakukan pada semua level pengujian (Pengujian unit, pengujian integrasi, pengujian sistem, dan pengujian penerimaan). Pengujian kotak hitam memiliki beberapa teknik [COP-04], diantaranya adalah :

1. Pengujian nilai batas (*Boundary value testing*)
2. Pengujian use case (*Use case testing*)

Pengujian kotak putih adalah suatu strategi pengujian yang berdasarkan pada jalur, struktur atau implementasi perangkat lunak yang diuji [COP-04]. Pengujian kotak putih suatu perangkat lunak didasarkan pada pengamatan yang teliti terhadap detail prosedural. Jalur-jalur logika yang melewati perangkat lunak

diuji dengan memberikan *test case* yang manguji serangkaian kondisi dan atau *loop* tertentu. Pengujian kotak putih memiliki beberapa teknik, salah satu teknik yang sering dipergunakan adalah *basis path testing*.

Pengujian kotak abu-abu (*gray box testing*) adalah gabungan pengujian kotak putih dan kotak hitam [DOC-04]. Pengujian ini mengharuskan kita untuk mengetahui jalur, struktur atau implementasi suatu perangkat lunak dan kemudian mengujinya dengan pengujian kotak hitam.

a. Pengujian Nilai Batas pada Pengujian Kotak Hitam

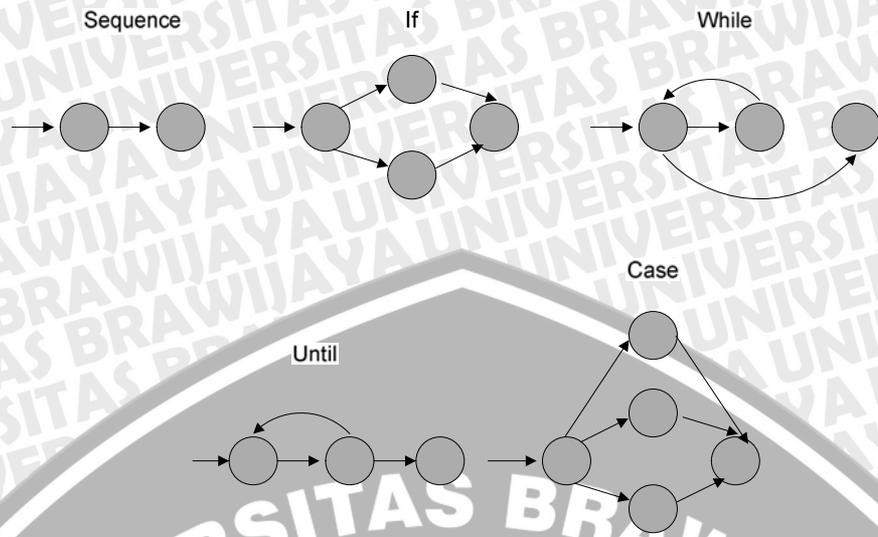
Pengujian nilai batas adalah suatu teknik pengujian yang digunakan untuk mengurangi jumlah kasus uji dengan tidak mengurangi mutu pengujian. Pengujian nilai batas memfokuskan pada nilai-nilai batas, karena pada nilai-nilai batas tersebut sering terjadi kesalahan. Bila suatu fungsi bekerja mulai nilai x , maka pengujian nilai batas akan memasukkan nilai $x-1$, x , dan $x+1$ sebagai kasus uji dari fungsi tersebut.

b. Pengujian Use Case pada Pengujian Kotak Hitam

Pengujian *use case* dilakukan pada implementasi sistem. Pengujian *use case* ini dilakukan terhadap setiap skenario yang ada dari suatu *use case*, baik *main success scenario* maupun *extensions scenario*.

c. Basis Path Testing pada Pengujian Kotak Putih

Metode *basis path testing* ini memungkinkan perancang *test case* mengukur kompleksitas logis dari desain prosedural dan *basis set* dari jalur eksekusi. *Test case* yang dilakukan untuk menggunakan *basis set* tersebut dijamin untuk menggunakan setiap pernyataan di dalam program paling tidak sekali selama pengujian. Teknik pengujian ini memerlukan penelusuran terhadap kontrol logika untuk menentukan *test case* dalam proses pengujian. Penggambaran aliran kontrol logika dapat menggunakan diagram alir (grafik alir) dengan notasi ditunjukkan pada Gambar 2.24.



Gambar 2.24 : Notasi grafik alir
Sumber: PRE-01:446

Kompleksitas siklomatis merupakan matrik perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis atas suatu program. Penentuan kompleksitas siklomatis bisa dilakukan dengan beberapa cara, diantaranya:

1. Jumlah *region* grafik alir sesuai dengan kompleksitas siklomatis.
2. Kompleksitas siklomatis $V(G)$, untuk grafik G adalah $V(G) = E - N + 2$, dimana E adalah jumlah *edge*, dan N adalah jumlah *node*.
3. $V(G) = P + 1$, dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir G .

2.2.5 PHP (Hypertext Preprocessor)

PHP adalah sebuah bahasa dalam pemrograman *web* yang bekerja pada sisi *server*. PHP tidak mempunyai *script editor*, sehingga kita akan tetap bisa menggunakan *text editor* apapun. Namun selain itu PHP juga tidak memiliki *Compiler*, karena *Script PHP* akan *dicompile* oleh *web server* saat digunakan. Berikut contoh sederhana dari skrip PHP [PHP-07:06].

```
<?php
echo "Hello world";
?>
```



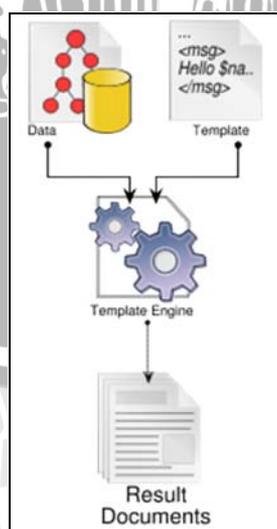
Script ini adalah suatu *script* sederhana untuk menampilkan kata “*Hello world*”. Secara struktur bahasa, *PHP* memiliki kesamaan dengan bahasa C, karena memang *PHP* adalah bahasa C yang disederhanakan untuk pemrograman *web* [PHP-07:06].

Setelah berhasil menulis suatu *script PHP* maka harus disimpan dengan ekstensi *file *.php*. Karena *PHP* adalah bahasa pemrograman *web* yang berjalan pada *server side*, maka tidak dapat dijalankan dari komputer yang bersifat *stand alone*. Namun hal ini dapat diatasi dengan membuat *virtual web server*. *Virtual web server* dibuat dengan program bernama *apache*. *Apache* sendiri sebenarnya adalah *HTTP server* yang dibuat untuk sistem operasi *UNIX*. Dengan demikian, dibuat seolah komputer *connect* dengan *IP* tujuan 127.0.0.1, dimana *IP* ini adalah alamat *apache* yang merupakan *localhost* [PHP-07:07].

2.2.6 Template Engine

Template Engine merupakan *framework* yang berfungsi untuk memisahkan *logic code* dan *template code*. Macam – macam *template engine* antara lain : *Smarty*, *Heyes Template Class*, *FastTemplate*, *VarPage Class*, *Powerful PHP Templates*, dll. Saat ini predikat *template engine* terbaik diraih oleh *Smarty* karena terkenal dengan kecepatan dan fleksibilitasnya [TPL-08:01].

Jika diilustrasikan pemisahan *template code* dan *logic code* secara sederhana akan seperti diagram dibawah ini :



Gambar 2.25. Diagram Ilustrasi Cara Kerja *Template Engine*

Sumber : [TEM - 08]

Pada ilustrasi diatas dapat disimpulkan bahwa file yang akan dibuat jika menggunakan template engine adalah dua macam, yaitu file data yang berisi *logic code* dan *template* yang berisi *template code* yang nantinya file – file ini dicompile oleh *template engine* dan akan menghasilkan 1 file lagi (*result document*) yang akan ditampilkan ke *browser* .

2.2.6.1 Smarty

Smarty merupakan sebuah *template engine* untuk bahasa pemrograman *PHP*. Fasilitas yang disediakan oleh *Smarty* ialah adanya metode untuk mengatur pemisahan antara *script* yang bersifat logika dengan *template*. Hal ini dapat memudahkan bagi pengerjaan sebuah software yang mana pada umumnya orang yang mengerjakan *script* dan *template designer* adalah orang yang berbeda, sehingga ketika dilakukan sebuah pemodifikasian pada salah satu bagian, tidak akan mengganggu bagian lainnya [SMA-07:02].

Salah satu konsep yang menarik pada *Smarty* ialah adanya metode *template compiling*. Artinya *Smarty* dapat membaca file *template* dan menciptakan *script PHP* didalamnya. Ketika file hasil *compile*-nya telah terbuat, maka halaman web yang akan dibuka akan mengacu pada file hasil *compile* tadi. Dengan kata lain, file hasil *compile* merupakan *chace* dari halaman yang telah dibuka sebelumnya [SMA-07:02].

2.2.7 Basis Data

Basis data adalah himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan dengan cepat dan mudah [FAT-02].

2.2.7.1 Diagram Entity Relationship

Diagram *entity relationship* adalah diagram yang menggambarkan hubungan antara entitas basis data. Dua hal utama diagram *entity relationship* (diagram ER) adalah entitas (*entity*) dan relasi (*relationship*).

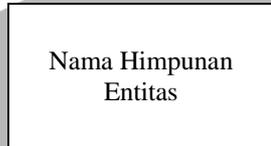
a) Entitas

Entitas adalah individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dengan sesuatu yang lain [FAT-02]. Himpunan entitas adalah

sekelompok entitas yang sejenis dan berada dalam lingkup yang sama [FAT-02].
 Contoh himpunan entitas dan entitas adalah sebagai berikut :

1. Himpunan entitas mobil dengan entitas Suzuku, Toyota, Honda.
2. Himpunan entitas Binatang dengan entitas kambing, gajah, ayam.

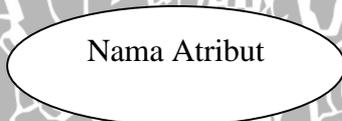
Himpunan entitas di dalam diagram ER dinotasikan dengan persegi panjang dengan nama himpunan entitas diletakkan di tengah persegi panjang.



Gambar 2.26 : Notasi entitas
Sumber : [FAT-02]

b) Atribut

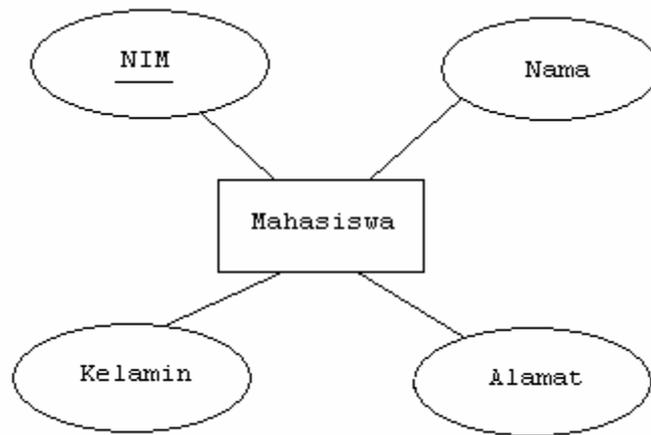
Atribut adalah karakter (properti) dari suatu entitas atau relasi. Atribut di dalam diagram ER dinotasikan sebagai elips dengan nama atribut diletakkan di tengah elips tersebut. Atribut yang berfungsi sebagai kunci dari himpunan entitas atau himpunan relasi tersebut diberi garis bawah.



Gambar 2.27 : Notasi atribut
Sumber : [FAT-02]

Gambar 2.28 adalah contoh hubungan antara himpunan entitas dengan atribut. Himpunan entitas Mahasiswa memiliki empat buah atribut, yaitu : NIM, Nama, Kelamin, dan Alamat. Atribut NIM berfungsi sebagai kunci dari himpunan entitas Mahasiswa.

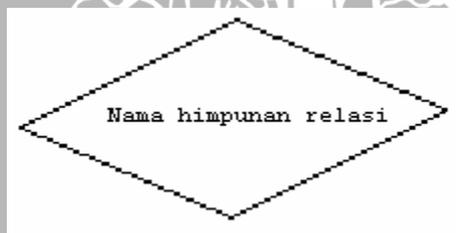




Gambar 2.28 : Contoh hubungan himpunan entitas dengan atribut
Sumber : [FAT-02]

c) Relasi

Relasi adalah hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda [FAT-02]. Himpunan relasi adalah kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan entitas [FAT-02]. Himpunan relasi dalam diagram ER dinyatakan dengan belah ketupat dengan nama himpunan relasi berada ditengah belah ketupat.



Gambar 2.29 : Notasi himpunan relasi
Sumber : [FAT-02]

Relasi dapat memiliki derajat relasi atau kardinalitas relasi. Derajat relasi adalah jumlah maksimal entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang dapat terjadi di antara dua himpunan entitas dapat berupa :

1. Satu ke Satu (*One to One*)

Setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, begitu pula sebaliknya.



2. Satu ke Banyak (*One to Many*)

Setiap entitas pada himpunan entitas A berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya. Setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

3. Banyak ke Satu (*Many to One*)

Setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya. Setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A. Derajat relasi *many to one* adalah sama dengan derajat relasi *one to many*.

4. Banyak ke Banyak (*Many to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, begitu pula sebaliknya.

Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka 1 dan 1 untuk relasi satu ke satu 1 dan N untuk relasi satu ke banyak. N dan N untuk relasi banyak ke banyak.

2.2.7.2 MySQL

MySQL adalah suatu bahasa berbasis SQL yang dikembangkan secara *free* dan *open source*. Perintah-perintah MySQL dikelompokkan menjadi lima macam:

1. *Data Definition Language (DDL)* [SQL-06;08].

Adalah perintah MySQL yang digunakan untuk menjelaskan objek dari data base. Dengan kata lain, *DDL* digunakan untuk mendefinisikan kerangka database. Perintahnya adalah:

- *Create*: untuk membuat/menciptakan objek *database*
- *Alter*: untuk memodifikasi/mengubah objek *database*
- *Drop*: untuk menghapus objek *database*
- Objek *database* yang dimaksud terdiri dari *database, table, index*, dan *view*

2. *Data Manipulation Language (DML)* [SQL-06;08].

Adalah perintah yang digunakan untuk mengoperasikan atau

memanipulasi isi *database*. *MySQL* menyediakan 4 perintah *DML*:

- *Select*: digunakan untuk mengambil data dari *database*
- *Delete*: digunakan untuk menghapus data dari *database*
- *Insert*: menambahkan data ke *database*
- *Update*: memodifikasi data pada *database*

3. *Security*

Adalah perintah-perintah yang digunakan untuk menjamin keamanan data.

Antara lain terdiri atas:

- *Grant*: memberi akses kepada user tertentu untuk akses ke *database*
- *Revoke*: mencabut hak akses dari *user*

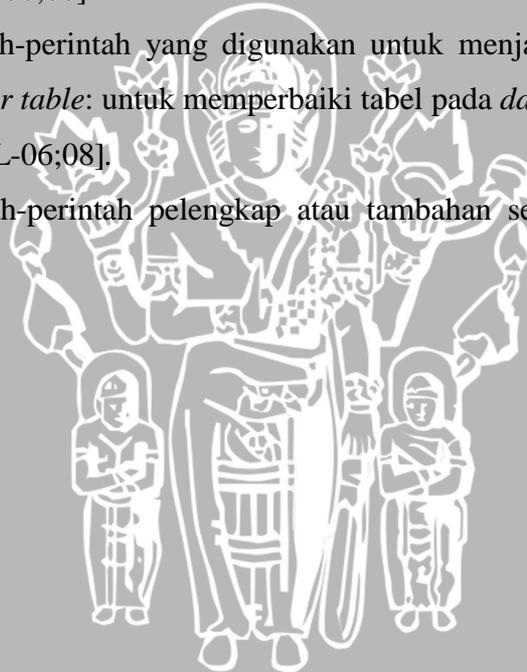
4. *Integrity* [SQL-06;08].

Adalah perintah-perintah yang digunakan untuk menjaga kesatuan data.

Contoh: *recover table*: untuk memperbaiki tabel pada *database*

5. *Auxilliary* [SQL-06;08].

Adalah perintah-perintah pelengkap atau tambahan seperti: *unload* dan *rename*.



BAB III

METODE PENELITIAN

Bab ini menjelaskan tentang langkah-langkah yang akan dilakukan untuk merealisasikan aplikasi perangkat lunak yang akan dibuat. Langkah-langkah yang diperlukan adalah studi literatur, analisis perangkat lunak, perancangan perangkat lunak, implementasi perangkat lunak, pengujian, serta pengambilan kesimpulan dan saran.

3.1 Studi Literatur

Studi literatur menjelaskan kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

1. Teori dasar psikotes
2. Teori dasar RPL, meliputi :
 - a. Proses-proses RPL
 - b. Model proses perangkat lunak
 - c. Paradigma RPL
3. UML, meliputi :
 - a. *Use case diagram*
 - b. *Class diagram*
 - c. *Sequence diagram*
4. Pengujian
 - a. Teknik pengujian
 - b. Strategi pengujian
5. Teknologi *Template Engine* dan *PHP*
 - a. *Smarty template Engine*
 - b. *PHP (Hypertext Preprocessor)*
6. Basis data

3.2 Analisis dan Perancangan Sistem Aplikasi

Tahap ini akan dilakukan analisis kebutuhan dan perancangan yang sesuai dengan prinsip-prinsip RPL. Tahap analisis dan perancangan ini akan

menggunakan metode berorientasi objek dan dibantu dengan UML sebagai bahasa pemodelannya.

Analisis yang pertama kali dilakukan adalah membuat diagram UML yang berupa *use case diagram* dan *use case specification* dari kebutuhan. *Use case specification* akan lebih menjelaskan setiap *use case* yang ada.

Perancangan dimulai dengan perancangan basis data keperustakaan. Basis data keperustakaan akan menyimpan data-data antara lain koleksi perpustakaan, anggota, sirkulasi, dan lain sebagainya. Perancangan dilanjutkan dengan membuat diagram-diagram UML yang berupa *class diagram* dan *sequence diagram*. Pembuatan *class diagram* dan *sequence diagram* dibuat untuk membantu dalam fase implementasi sistem.

3.3 Implementasi Sistem Aplikasi

Implementasi mengubah perancangan yang ada menjadi algoritma dan kode-kode program. Metode yang digunakan adalah pemrograman berbasis obyek. Alat bantu yang digunakan untuk pembuatan sistem aplikasi adalah *WebBuilder* dengan bahasa pemrograman *PHP 5.0*.

3.4 Pengujian Sistem Aplikasi

Pengujian dilakukan untuk menjamin dan memastikan bahwa sistem yang dirancang dapat berjalan seperti yang diharapkan. Tolak ukur keberhasilan uji coba adalah semua kebutuhan telah diwujudkan dan berjalan dengan baik. Beberapa rencana pengujian yang mendasar adalah sebagai berikut:

Pengujian *software*, yaitu berisi pengujian pada *client* untuk memastikan apakah psikotes *online* berjalan dengan baik, diantaranya adalah menguji menjawab soal – soal psikotes yang tersedia dan hasilnya sesuai dengan nilai jawaban yang benar.

3.5 Pengambilan Kesimpulan dan Saran

Setelah melakukan pengujian, dilakukan pengambilan saran untuk pengembangan sistem lebih lanjut. Pengambilan kesimpulan didasarkan pada kesesuaian antara teori dan praktek. Berdasarkan performa rancangan sistem

aplikasi yang berjalan apakah dapat diterapkan secara layak di lingkungan konsultan psikologi yang sesungguhnya.



BAB IV PERANCANGAN

Bab ini membahas analisis kebutuhan dan perancangan aplikasi. Tahap analisis kebutuhan menggunakan pemodelan *use case diagram* beserta *use case specification*nya. Tahap perancangan aplikasi berisi perancangan sistem, *class diagram*, *sequence diagram*, dan perancangan basis data.



Gambar 4.1 : Diagram Pohon Analisis Kebutuhan dan Perancangan

Gambar 4.1 menunjukkan langkah-langkah proses analisis dan perancangan yang akan digunakan pada bab ini. Metode analisis dan perancangan yang dipergunakan adalah metode analisis dan perancangan berorientasi objek, dengan UML sebagai bahasanya pemodelannya.

4.1 Analisis Kebutuhan

Analisis kebutuhan perangkat lunak adalah aktifitas rekayasa perangkat lunak yang menjembatani antara kebutuhan ditingkat sistem yang sudah ada dengan yang dirancang pada perancangan perangkat lunak [PRE-02]. Analisis kebutuhan perangkat lunak dapat juga diartikan proses yang digunakan untuk

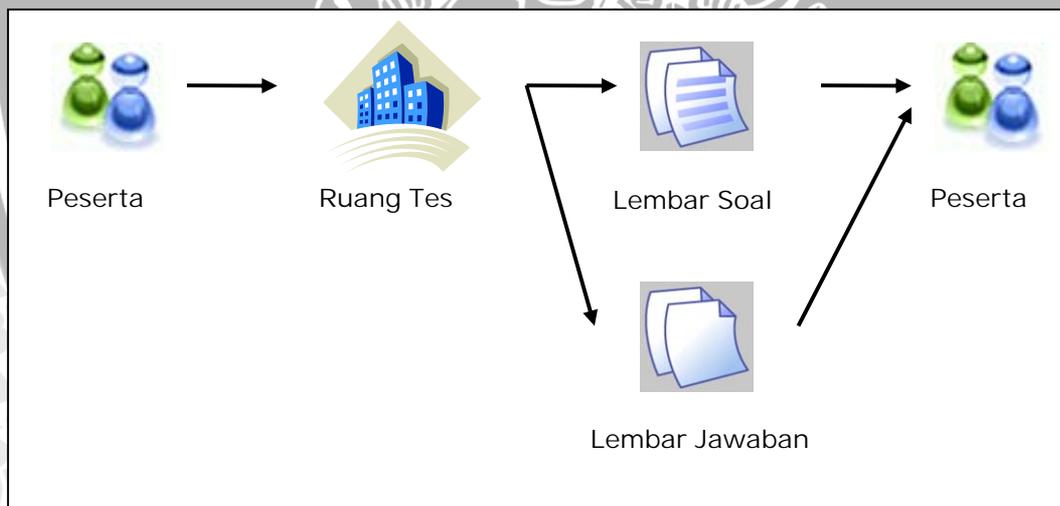
mendapatkan, menganalisis, dan memvalidasi kebutuhan-kebutuhan sistem [SOM-02].

Analisis kebutuhan perangkat lunak menggunakan bahasa pemodelan UML. Tahap analisis kebutuhan menggunakan pemodelan *use case diagram* beserta *use case specification*-nya.

4.1.1 Analisis Sistem yang Telah Ada

Tujuan dari pengembangan sebuah sistem adalah menghasilkan sesuatu yang dapat memenuhi kebutuhan dari pengguna yang akan menggunakannya. Tujuan ini dapat dipenuhi dalam suatu pengembangan sistem dengan cara melakukan analisis kebutuhan pada sistem tersebut.

Proses penyelenggaraan psikotes yang telah ada selama ini yaitu dengan melakukan tes secara kolektif dan dipandu oleh salah satu petugas dari konsultan psikologi supaya pengerjaan tes dapat dilakukan secara serentak. Ilustrasi penyelenggaraan psikotes saat ini dapat dilihat pada Gambar 4.2.



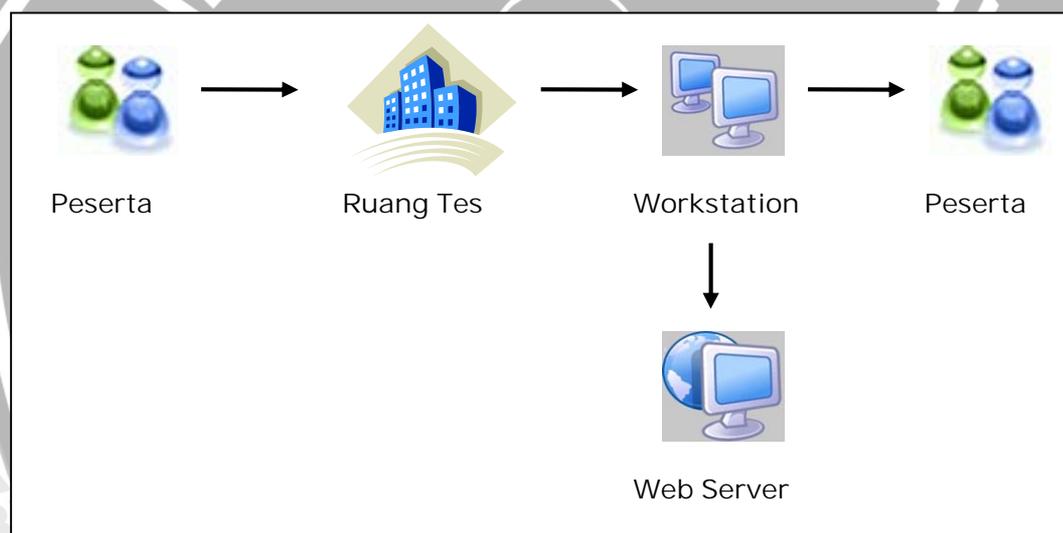
Gambar 4.2 : Penyelenggaraan Psikotes Saat Ini

Penyelenggaraan psikotes *online* memiliki kelebihan antara lain:

1. Konsultan psikologi dapat mengatur petugas pada saat penyelenggaraan psikotes dengan cepat.
2. Konsultan psikologi dapat mengatur instansi *client* dengan cepat
3. Konsultan psikologi dapat menentukan kategori soal psikotes dengan cepat

4. Konsultan psikologi dapat menentukan sub kategori soal psikotes dengan cepat
5. Konsultan psikologi dapat menentukan urutan kategori soal psikotes dengan cepat
6. Konsultan psikologi dapat memasukkan dan memodifikasi soal-soal psikotes dengan cepat
7. Konsultan psikologi dapat melihat rekap hasil psikotes dengan cepat

Penyelenggaraan psikotes setelah menggunakan perangkat lunak dapat dilihat pada Gambar 4.3. Peserta memasuki ruangan tes, kemudian menggunakan perangkat lunak yang telah tersedia untuk menjawab soal-soal psikotes sesuai petunjuk yang dijelaskan dalam perangkat lunak.



Gambar 4.3 : Penyelenggaraan Psikotes *Online*
Sumber : [Analisis]

4.1.2 Analisis Kebutuhan Sistem yang akan Dirancang

Pengembangan sebuah perangkat lunak bertujuan untuk menghasilkan perangkat lunak yang dapat memenuhi kebutuhan *user*. Setiap pengembangan sebuah sistem perangkat lunak memerlukan adanya dokumentasi terhadap kebutuhan-kebutuhan *user* agar tujuan tersebut tercapai. Tahap analisis kebutuhan dilakukan dengan memodelkan kebutuhan ke dalam *use case diagram*. *Use case diagram* bertujuan untuk menggambarkan kebutuhan-kebutuhan fungsional yang harus disediakan oleh sistem psikotes *online* agar dapat

memecahkan permasalahan yang dihadapi oleh pengguna.

4.1.2.1 Daftar Kebutuhan

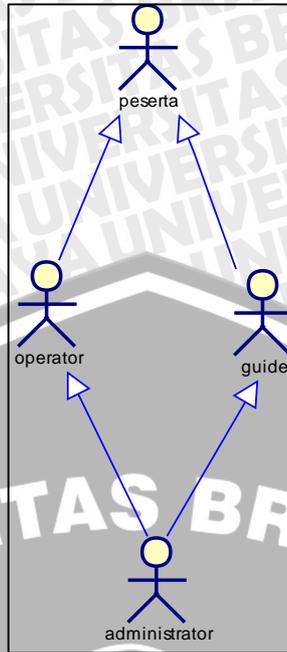
Daftar kebutuhan merupakan daftar yang menguraikan kebutuhan-kebutuhan pengguna yang harus disediakan oleh perangkat lunak baik kebutuhan fungsional maupun non fungsional. Proses yang dilakukan sebelum menentukan daftar kebutuhan adalah mengidentifikasi aktor yang menggunakan *software* psikotes *online*. Tabel 4.1 memperlihatkan daftar aktor beserta penjelasannya yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 : Deskripsi Aktor

No	Aktor	Keterangan
1	Peserta	Aktor yang menggunakan sistem psikotes <i>online</i> untuk menjawab soal – soal psikotes.
2	Operator	Aktor yang bertugas melakukan pemeriksaan terhadap peserta dan memberikan <i>password</i> login untuk peserta.
3	Guide	Aktor yang bertugas memandu ketika pelaksanaan tes berlangsung.
4	Administrator	Aktor yang bertugas mengadministrasi soal – soal dan data-data pendukung penyelenggaraan psikotes <i>online</i> .

Sumber : [Analisis]

Aktor operator dan administrator adalah petugas-petugas dari konsultan psikologi. Gambar 4.4 adalah *use case diagram* untuk generalisasi dari keseluruhan aktor dari Tabel 4.1.



Gambar 4.4 : Generalisasi Aktor-Aktor Pengguna Sistem Psikotes *Online*
Sumber : [Analisis]

Penyusunan daftar kebutuhan fungsional dari sistem psikotes *online* dilakukan setelah identifikasi aktor. Daftar kebutuhan fungsional sistem disertai dengan nama *use case* yang merepresentasikan fungsionalitas dari kebutuhan tersebut. Daftar kebutuhan fungsional tersebut ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Fungsional

No	Kebutuhan	Use case
1	Sistem harus mampu memberikan fasilitas untuk login, sehingga hanya pengguna tertentu yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu (sebagai suatu aktor).	Login
2	Sistem harus menyediakan fasilitas agar pengguna yang telah login dapat keluar dari sistem.	Logout
3	Sistem harus menyediakan fasilitas untuk menambah petugas. Data yang harus dicatat dari penambahan petugas adalah nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.	Menambah data petugas
4	Sistem harus dapat menampilkan semua petugas yang tercatat di dalam sistem. Data yang ditampilkan berupa nip, nama, jenis kelamin, telepon, <i>expired</i> , dan jenis petugas.	Melihat daftar petugas

5	Sistem harus menyediakan fasilitas untuk mengedit data petugas yang telah dimasukkan. Data yang dapat diedit adalah nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.	Mengedit data petugas
6	Sistem harus menyediakan fasilitas untuk menghapus data petugas tertentu.	Menghapus data petugas
7	Sistem harus menyediakan fasilitas untuk mencari data petugas yang ada di dalam sistem. Pencarian petugas berdasarkan nip, dan nama petugas.	Mencari data petugas
8	Sistem harus menyediakan fasilitas untuk menambah data instansi <i>client</i> pengguna jasa. Data yang harus dicatat dari penambahan instansi adalah nama instansi, alamat, bidang usaha, contact person, telepon, email, fax.	Menambah data instansi <i>client</i>
9	Sistem harus dapat menampilkan daftar instansi <i>client</i> pengguna jasa. Data yang ditampilkan berupa nama instansi, alamat, bidang usaha, contact person, telepon, fax, dan website.	Melihat daftar instansi <i>client</i>
10	Sistem harus menyediakan fasilitas untuk mengedit data instansi <i>client</i> yang telah dimasukkan. Data yang dapat diedit adalah nama instansi, alamat, bidang usaha, contact person, telepon, email, fax, dan website.	Mengedit data instansi <i>client</i>
11	Sistem harus menyediakan fasilitas untuk menghapus data instansi tertentu.	Menghapus data instansi
12	Sistem harus menyediakan fasilitas untuk mencari data suatu instansi yang ada di dalam sistem. Pencarian data instansi berdasarkan nama instansi.	Mencari data instansi
13	Sistem harus menyediakan fasilitas untuk menambah kategori soal. Data yang harus dicatat dari penambahan kategori adalah nama kategori, tujuan, dan keterangan.	Menambah data kategori soal
14	Sistem harus dapat menampilkan kategori soal yang tercatat di dalam sistem. Data yang ditampilkan berupa nama kategori, tujuan, dan keterangan.	Melihat daftar kategori soal
15	Sistem harus menyediakan fasilitas untuk mengedit data kategori soal yang telah dimasukkan. Data yang dapat diedit adalah nama kategori, tujuan, dan keterangan.	Mengedit data kategori soal
16	Sistem harus menyediakan fasilitas untuk menghapus data kategori tertentu.	Menghapus data kategori
17	Sistem harus menyediakan fasilitas untuk mencari data kategori soal yang ada di dalam sistem. Pencarian data kategori soal berdasarkan nama kategori.	Mencari kategori soal
18	Sistem harus menyediakan fasilitas untuk menambah sub kategori soal. Data yang harus dicatat dari penambahan sub	Menambah data sub

	kategori adalah nama sub kategori.	kategori
19	Sistem harus dapat menampilkan sub kategori soal yang tercatat di dalam sistem. Data yang ditampilkan berupa nama sub kategori.	Melihat daftar sub kategori soal
20	Sistem harus menyediakan fasilitas untuk mengedit data sub kategori soal yang telah dimasukkan. Data yang dapat diedit adalah nama sub kategori.	Mengedit data sub kategori soal
21	Sistem harus menyediakan fasilitas untuk menghapus data sub kategori tertentu.	Menghapus data sub kategori
22	Sistem harus menyediakan fasilitas untuk mencari data sub kategori soal yang ada di dalam sistem. Pencarian data sub kategori soal berdasarkan nama sub kategori.	Mencari sub kategori soal
23	Sistem harus menyediakan fasilitas untuk menentukan urutan kategori soal yang akan diberikan pada peserta. Data yang dimasukkan adalah lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.	Menambah urutan kategori soal yang akan diberikan
24	Sistem harus menyediakan fasilitas untuk mengedit data urutan kategori soal yang telah dimasukkan. Data yang dapat diedit adalah lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.	Mengedit urutan kategori soal yang akan diberikan
25	Sistem harus dapat menampilkan urutan kategori soal yang tercatat di dalam sistem untuk pelamar instansi tertentu. Data yang ditampilkan berupa lokasi tes, waktu tes, kategori, sub kategori, nomor urut, lama pengerjaan, dan jumlah soal.	Menampilkan daftar urutan kategori soal yang akan diberikan
26	Sistem harus menyediakan fasilitas untuk menghapus nomor urut kategori tertentu.	Menghapus urutan kategori soal yang akan diberikan
27	Sistem harus menyediakan fasilitas untuk mencari data urutan kategori soal yang akan diberikan pada pelamar untuk instansi tertentu. Pencarian data berdasarkan nama lokasi tes.	Mencari data urutan kategori soal yang akan diberikan pada pelamar instansi tertentu
28	Sistem harus menyediakan fasilitas untuk menambah data waktu dan tempat pelaksanaan psikotes untuk peserta. Data yang dapat ditambahkan adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.	Menambah data waktu dan tempat pelaksanaan psikotes
29	Sistem harus menyediakan fasilitas untuk menampilkan data waktu dan tempat pelaksanaan psikotes untuk peserta. Data	Melihat data waktu dan

	yang dapat ditampilkan adalah tanggal, waktu, tempat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.	tempat pelaksanaan psikotes
30	Sistem harus menyediakan fasilitas untuk mengedit data waktu dan tempat pelaksanaan psikotes untuk peserta. Data yang dapat diedit adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.	Mengedit data waktu dan tempat pelaksanaan psikotes
31	Sistem harus menyediakan fasilitas untuk menghapus data waktu dan tempat pelaksanaan psikotes.	Menghapus data waktu dan tempat pelaksanaan psikotes
32	Sistem harus menyediakan fasilitas untuk mencari data waktu dan tempat pelaksanaan psikotes untuk peserta. Pencarian data berdasar tempat, dan nama instansi.	Mencari data waktu dan tempat pelaksanaan psikotes
33	Sistem harus menyediakan fasilitas untuk menambah data soal – soal psikotes per seb kategori yang diberikan pada peserta. Data yang dapat ditambahkan adalah soal.	Menambah data soal – soal psikotes
34	Sistem harus menyediakan fasilitas untuk menampilkan data soal – soal psikotes yang diberikan pada pelamar. Data yang dapat ditampilkan adalah daftar soal berdasar sub kategori tertentu.	Menampilkan data soal – soal psikotes
35	Sistem harus menyediakan fasilitas untuk mengedit data soal – soal psikotes yang diberikan pada pelamar. Data yang dapat diedit adalah soal.	Mengedit data soal – soal psikotes
36	Sistem harus menyediakan fasilitas untuk menghapus data soal – soal psikotes yang diberikan pada peserta tes.	Menghapus data soal – soal psikotes
37	Sistem harus menyediakan fasilitas untuk mencari data soal – soal psikotes yang diberikan pada pelamar. Pencarian data berdasarkan kutipan kalimat soal.	Mencari data soal – soal psikotes
38	Sistem harus menyediakan fasilitas untuk menambah pilihan jawaban soal psikotes per sub kategori yang diberikan pada peserta. Data yang dapat ditambahkan adalah huruf pilihan, jawaban, dan status jawaban.	Menambah pilihan jawaban soal psikotes
39	Sistem harus menyediakan fasilitas untuk menampilkan pilihan jawaban soal psikotes per sub kategori. Data yang dapat ditampilkan adalah huruf pilihan, jawaban, status berdasar soal tertentu.	Menampilkan pilihan jawaban soal psikotes
40	Sistem harus menyediakan fasilitas untuk mengedit pilihan jawaban soal psikotes per sub kategori. Data yang dapat diedit adalah huruf pilihan, jawaban, dan status jawaban.	Mengedit pilihan jawaban soal psikotes
41	Sistem harus menyediakan fasilitas untuk menghapus pilihan jawaban soal psikotes yang diberikan pada peserta	Menghapus pilihan

	tes.	jawaban soal psikotes
42	Sistem harus menyediakan fasilitas untuk menambah biodata peserta psikotes. Data yang dapat ditambahkan adalah id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin	Menambah data peserta psikotes
43	Sistem harus menyediakan fasilitas untuk menampilkan biodata peserta psikotes. Data yang dapat ditampilkan adalah id peserta, nama, alamat, pendidikan terakhir, jurusan, jenis kelamin	Menampilkan data peserta psikotes
44	Sistem harus menyediakan fasilitas untuk mengedit biodata peserta psikotes. Data yang dapat diedit adalah nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin	Mengedit data peserta psikotes
45	Sistem harus menyediakan fasilitas untuk menghapus data peserta psikotes yang ada dalam database.	Menghapus data peserta psikotes
46	Sistem harus menyediakan fasilitas untuk mencari biodata peserta psikotes. nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin Pencarian data berdasarkan nama, dan id peserta.	Mencari data peserta psikotes
47	Sistem harus menyediakan fasilitas untuk menambah peserta pada tempat dan waktu pelaksanaan psikotes. Data yang ditambahkan adalah lokasi tes, dan nama peserta.	Menambah data peserta pada lokasi psikotes
48	Sistem harus menyediakan fasilitas untuk menampilkan peserta pada tempat dan waktu pelaksanaan psikotes. Data yang ditampilkan adalah nama peserta, id peserta, tempat tes, waktu tes, dan password.	Menampilkan data peserta pada lokasi psikotes
49	Sistem harus menyediakan fasilitas untuk mengedit data peserta pada tempat dan waktu pelaksanaan psikotes. Data yang diedit adalah lokasi tes, dan nama peserta.	Mengedit data peserta pada lokasi psikotes
50	Sistem harus menyediakan fasilitas untuk menghapus data peserta pada lokasi dan tempat pelaksanaan psikotes.	Menghapus data peserta pada lokasi psikotes
51	Sistem harus menyediakan fasilitas untuk mencari data peserta psikotes pada tempat dan waktu pelaksanaan tes yang ditentukan. Pencarian data berdasarkan nama, id peserta, dan lokasi tes.	Mencari data peserta pada lokasi psikotes
52	Sistem harus menyediakan fasilitas untuk menampilkan nilai psikotes peserta untuk setiap sub kategori. Data yang dihitung adalah nilai total dari setiap sub kategori dengan menjumlahkan nilai jawaban benar. Data yang ditampilkan adalah id peserta, nama, lokasi tes, waktu tes, nilai total per sub kategori, dan nilai konsultan.	Menampilkan nilai peserta tes untuk setiap kategori soal
53	Sistem harus menyediakan fasilitas untuk menjawab soal	Menjawab

	psikotes berdasar sub kategori soal.	soal psikotes
54	Sistem harus menyediakan fasilitas untuk menambah hasil penilain konsultan berdasar nilai yang diperoleh peserta untuk setiap sub kategori. Data yang ditambahkan adalah nilai konsultan.	Menambah penilain konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori soal.
55	Sistem harus menyediakan fasilitas untuk menampilkan hasil penilaian konsultan berdasar nilai yang diperoleh peserta untuk setiap kategori. Data yang ditampilkan adalah id peserta, nama peserta, tempat tes, waktu tes, sub kategori soal, nilai, dan penilaian konsultan.	Menampilkan penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori soal.
56	Sistem harus menyediakan fasilitas untuk mencari data nilai peserta psikotes pada tempat dan waktu pelaksanaan tes yang ditentukan. Pencarian data berdasarkan nama, id peserta, dan lokasi tes.	Mencari nilai peserta tes
57	Sistem harus menyediakan fasilitas untuk <i>generate password login</i> untuk peserta tes.	<i>Generate password login</i> peserta tes

Sumber : [Analisis]

Keseluruhan kebutuhan fungsionalitas di atas dibagi menjadi 6 modul untuk lebih mempermudah pemahaman dan perancangan sistem. Keenam modul tersebut adalah sebagai berikut :

1. Modul pendukung sistem
2. Modul pengaturan instansi *client*
3. Modul pengaturan peserta tes
4. Modul pengaturan soal psikotes
5. Modul pengaturan penyelenggaraan tes
6. Modul perhitungan nilai dan hasil penilaian

Tabel 4.3 berikut menyatakan pembagian modul-modul dan *use case* yang termasuk dalam modul tersebut :

Tabel 4.3 : Modul dan kebutuhan fungsionalitas

No	Modul	Kebutuhan fungsionalitas
1	Modul pendukung sistem	<ol style="list-style-type: none"> 1. Login 2. Logout 3. Melihat daftar petugas 4. Menambah data petugas 5. Mengedit data petugas 6. Menghapus data petugas 7. Mencari data petugas
2	Modul pengaturan instansi <i>client</i>	<ol style="list-style-type: none"> 1. Menambah data instansi <i>client</i> 2. Melihat data instansi <i>client</i> 3. Mengedit data instansi <i>client</i> 4. Menghapus data instansi <i>client</i> 5. Mencari data instansi <i>client</i>
3	Modul pengaturan soal	<ol style="list-style-type: none"> 1. Menambah data kategori soal 2. Melihat data kategori soal 3. Mengedit data kategori soal 4. Menghapus data kategori soal 5. Mencari data kategori soal 6. Menambah data sub kategori soal 7. Melihat data sub kategori soal 8. Mengedit sub kategori soal 9. Menghapus data sub kategori soal 10. Mencari data sub kategori soal 11. Menentukan urutan kategori soal yang akan diberikan 12. Mengedit urutan kategori soal yang akan diberikan 13. Menampilkan urutan kategori soal yang akan diberikan 14. Menghapus urutan kategori soal yang akan diberikan 15. Mencari urutan kategori soal yang akan diberikan 16. Menambah data soal psikotes 17. Melihat data soal psikotes 18. Mengedit data soal psikotes 19. Menghapus data soal psikotes 20. Mencari data soal psikotes 21. Menambah pilihan jawaban soal psikotes 22. Melihat pilihan jawaban soal psikotes 23. Mengedit pilihan jawaban soal psikotes 24. Menghapus pilihan jawaban soal psikotes 25. Menjawab soal psikotes

4	Modul pengaturan penyelenggaraan psikotes	<ol style="list-style-type: none"> 1. Menambah data waktu dan tempat pelaksanaan tes 2. Melihat data waktu dan tempat pelaksanaan tes 3. Mengedit data waktu dan tempat pelaksanaan tes 4. Menghapus data waktu dan tempat pelaksanaan tes 5. Mencari data waktu dan tempat pelaksanaan tes
5	Modul pengaturan peserta tes	<ol style="list-style-type: none"> 1. Menambah data peserta psikotes 2. Menampilkan data peserta psikotes 3. Mengedit data peserta psikotes 4. Menghapus data peserta psikotes 5. Mencari data peserta psikotes 6. Menambah data peserta pada lokasi psikotes 7. Menampilkan data peserta pada lokasi psikotes 8. Mengedit data peserta pada lokasi psikotes 9. Menghapus data peserta pada lokasi psikotes 10. Mencari data peserta pada lokasi psikotes 11. Mengenerate <i>password login</i> peserta tes
6	Modul Perhitungan nilai dan hasil penilaian	<ol style="list-style-type: none"> 1. Menampilkan nilai peserta tes untuk setiap sub kategori soal 2. Menambah penilaian konsultan terhadap nilai yang diperoleh peserta tes untuk setiap sub kategori 3. Melihat penilaian konsultan terhadap nilai yang diperoleh peserta tes untuk setiap sub kategori 4. Mencari nilai peserta tes

Sumber : [Analisis]

Daftar kebutuhan non fungsional *software* psikotes *online* pada Tabel 4.4.

Tabel 4.4 : Daftar Kebutuhan Non Fungsional Sistem Psikotes *Online*

No	Kebutuhan non fungsional
1	Sistem dikembangkan dengan menggunakan bahasa php.
2	Sistem dikembangkan dengan menggunakan <i>template engine</i> .
3	Sistem harus dapat diakses melalui web browser.

Sumber: [Analisis]

4.1.2.2 Use Case Diagram

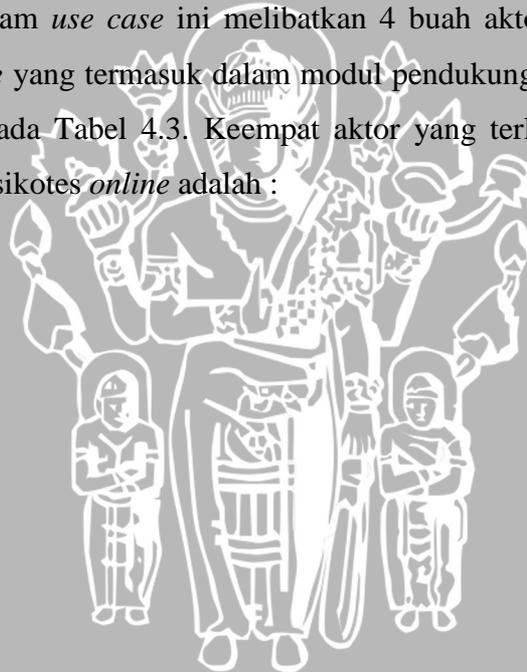
Use case diagram merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing *use case diagram* menunjukkan sekumpulan *use case*, aktor dan hubungannya.. *Use case* merupakan fungsionalitas dari sistem yang diinisiasi oleh aktor. Aktor adalah sebuah entitas-entitas luar yang berkomunikasi dengan sistem [HAR-04:267].

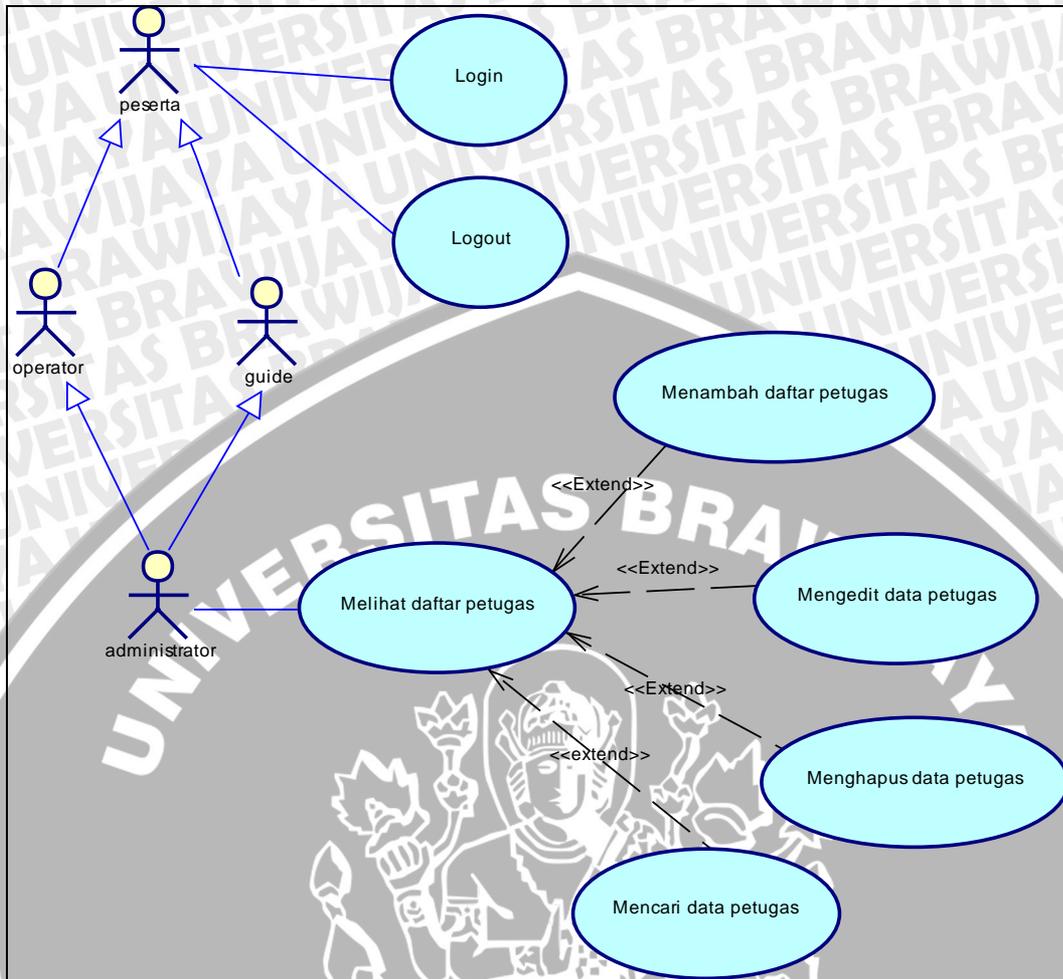
Pemodelan dalam *use case diagram* yang menggambarkan fungsionalitas yang disediakan oleh sistem psikotes *online* dibagi menjadi tujuh buah diagram yang bersesuaian dengan modul dalam sistem psikotes *online*.

4.1.2.2.1 Use Case Diagram untuk Modul Pendukung Sistem Psikotes Online

Gambar 4.5 adalah *use case diagram* untuk modul pendukung *software* psikotes *online*. Diagram *use case* ini melibatkan 4 buah aktor dan 7 buah *use case*. Ketujuh *use case* yang termasuk dalam modul pendukung *software* psikotes *online* dapat dilihat pada Tabel 4.3. Keempat aktor yang terlibat dalam modul pendukung *software* psikotes *online* adalah :

1. User
2. Operator
3. Guide
4. Administrator





Gambar 4.5 : Use Case Diagram untuk Modul Pendukung Sistem Psikotes Online

Sumber : [Analisis]

1. Use Case Specification Login

Tabel 4.5 : Use Case Specification Login

Nama use case	Login
Aktor	Peserta, operator, guide, administrator
Deskripsi	Sistem harus mampu memberikan fasilitas untuk login, sehingga hanya peserta dan petugas tertentu yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu (sebagai suatu aktor).
Pra-kondisi	Sistem telah berjalan.
Pasca-kondisi	User telah login sebagai aktor peserta, aktor operator, aktor



	guide, atau aktor administrator.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. User memasukkan username dan password dan menekan tombol login.	2. Sistem memvalidasi username dan password..
<p>Aliran Alternatif 1: Pasangan <i>username</i> dan <i>password</i> tidak ada di dalam sistem.</p> <p>(User pada langkah nomer 1 aliran utama tidak memasukkan <i>username</i> atau <i>password</i> yang sesuai.)</p>	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Menampilkan pesan bahwa pasangan username dan password tidak cocok.
	2. Kembali pada langkah nomer 1 aliran utama.

Sumber : [Analisis]

2. Use Case Specification Logout

Tabel 4.6: Use Case Specification Logout

Nama use case	Logout
Aktor	Peserta, guide, operator, administrator.
Deskripsi	Sistem harus menyediakan fasilitas agar pengguna yang telah login keluar dari sistem.
Pra-kondisi	User telah login.
Pasca-kondisi	User telah logout.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem

1. Aktor menekan tombol "logout"	2. Sistem melakukan logout untuk aktor tersebut.
----------------------------------	--

Sumber : [Analisis]

3. Use Case Specification Menambah Data Petugas

Tabel 4.7: Use Case Specification Menambah Daftar Petugas

Nama use case	Menambah data petugas	
Aktor	Administrator	
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah petugas. Data yang harus dicatat dari penambahan petugas adalah nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.	
Pra-kondisi	Administrator telah menjalankan <i>use case</i> Melihat daftar petugas	
Pasca-kondisi	Data petugas telah ditambahkan ke dalam sistem	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator memilih <i>link</i> "Tambah petugas".	2. Sistem menampilkan <i>form</i> masukan yang berupa <i>field</i> nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.	
3. Administrator memasukkan data nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> ,	4. Sistem memvalidasi masukan nip, nama petugas, <i>username</i> , <i>password</i> tidak boleh kosong dan <i>username</i> tersebut belum ada dalam sistem.	

<p><i>password</i>, jenis petugas, agama, dan keterangan.</p>	
	<p>5. Sistem menjalankan use case melihat daftar petugas.</p>
<p>Aliran Alternatif 1: Masukan nip, username, atau password kosong (Administrator pada langkah nomer 3 aliran utama tidak memasukkan nip atau nama petugas atau username atau password.)</p>	
<p>Aksi dari Aktor</p>	<p>Tanggapan dari Sistem</p>
	<p>1. Sistem menampilkan pesan kesalahan bahwa nip atau nama petugas atau username atau password tidak boleh kosong.</p>
	<p>2. Kembali pada langkah nomer 2 aliran utama.</p>
<p>Aliran Alternatif 2: Username telah ada dalam sistem (Administrator pada langkah nomer 3 aliran utama memasukkan <i>username</i> yang telah ada dalam sistem.)</p>	
<p>Aksi dari Aktor</p>	<p>Tanggapan dari Sistem</p>
	<p>1. Sistem menampilkan pesan kesalahan bahwa username tersebut telah ada dalam sistem.</p>
	<p>2. Kembali pada langkah nomer 2 aliran utama.</p>
<p>Aturan khusus</p> <p>1. Masukan nip, nama petugas, username, atau password tidak boleh kosong.</p>	

Sumber : [Analisis]



4. Use Case Specification Melihat Daftar Petugas

Tabel 4.8: Use Case Specification Melihat Daftar Petugas

Nama use case	Melihat daftar petugas
Aktor	Administrator
Deskripsi	Sistem harus dapat menampilkan semua data petugas yang tercatat di dalam sistem.
Pra-kondisi	Sistem telah berjalan dan aktor user telah login sebagai administrator.
Pasca-kondisi	Sistem menampilkan daftar petugas.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "data petugas".	2. Sistem menampilkan semua petugas yang telah ada di dalam sistem. Data petugas yang ditampilkan adalah nip, nama, jenis kelamin, telepon, <i>expired</i> , dan jenis petugas.

Sumber : [Analisis]

5. Use Case Specification Mengedit Data Petugas

Tabel 4.9: Use Case Specification Mengedit Data Petugas

Nama use case	Mengedit data petugas
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data petugas yang telah dimasukkan. Data yang dapat diedit adalah nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> Melihat daftar petugas

Pasca-kondisi	Data petugas yang telah diedit disimpan ke dalam sistem
----------------------	---

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "edit" salah satu baris dari daftar petugas yang ditampilkan.	2. Sistem menampilkan nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan. dari baris petugas yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.	4. Sistem memvalidasi masukan bahwa nip, nama petugas, username dan password tidak boleh kosong dan username tersebut belum ada dalam sistem.
	5. Sistem menjalankan use case melihat daftar petugas.

Aliran Alternatif 1: Masukan nip atau nama petugas atau *username* atau *password* kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan nip atau nama petugas atau *username* atau *password*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nip atau nama patugas atau <i>username</i> atau <i>password</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2



	aliran utama.
Aliran Alternatif 2: Username petugas telah ada dalam sistem	
(Administrator pada langkah nomer 3 aliran utama memasukkan <i>username</i> petugas yang telah ada dalam sistem.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa <i>username</i> petugas tersebut telah ada dalam sistem.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukan nip, nama petugas, <i>username</i> , atau <i>password</i> tidak boleh kosong.	

Sumber : [Analisis]

6. Use Case Specification Menghapus Data Petugas

Tabel 4.10 : Use Case Specification Menghapus Data Petugas

Nama use case	Menghapus data petugas
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data petugas tertentu.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> Melihat daftar petugas.
Pasca-kondisi	Data petugas dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "delete" salah satu baris dari daftar petugas yang ditampilkan dari <i>use case</i> Melihat	2. Sistem menjalankan <i>use case</i> menampilkan daftar petugas.

daftar petugas.	
-----------------	--

Sumber : [Analisis]

7. Use Case Specification Mencari Data Petugas

Tabel 4.11 : Use Case Specification Mencari Data Petugas

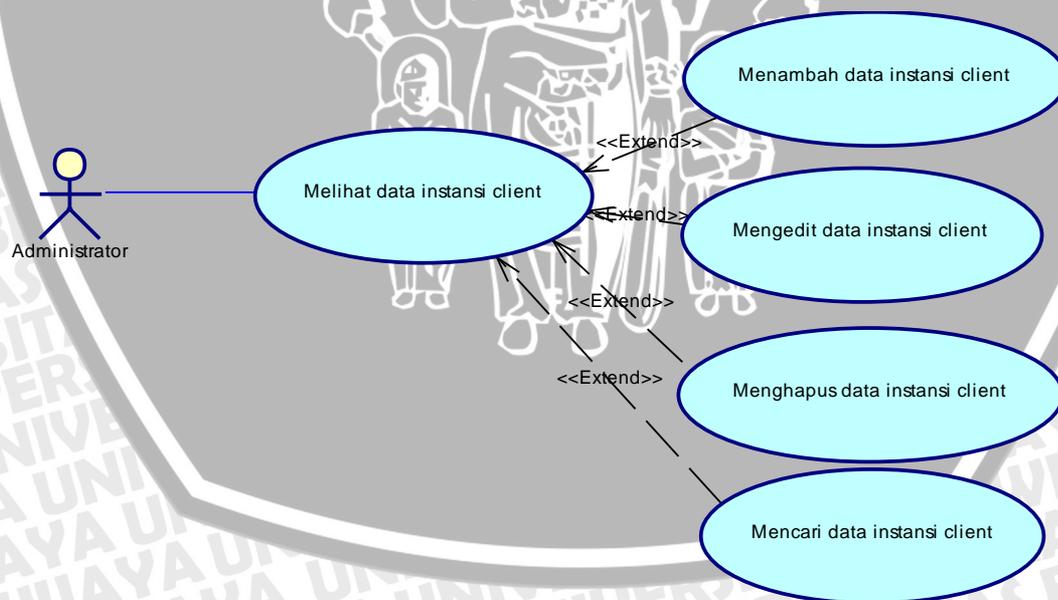
Nama use case	Mencari data data petugas	
Aktor	Administrator	
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data petugas yang ada di dalam sistem. Pencarian data petugas berdasarkan nip dan nama petugas.	
Pra-kondisi	Aktor user telah login sebagai aktor administrator.	
Pasca-kondisi	Sistem menampilkan data petugas yang dicari oleh aktor administrator.	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Aktor user memilih menu "Petugas".	2. Sistem menjalankan use case melihat daftar petugas beserta form pencarian petugas.	
3. User memasukkan data untuk masukan "keyword" dan mengklik tombol <i>Find</i> .	4. Sistem menampilkan data petugas yang dicari.	
Aliran Alternatif 1: Masukan keyword kosong		
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan <i>keyword</i> .)		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. Sistem menampilkan pesan kesalahan bahwa <i>keyword</i> tidak boleh kosong.	
	2. Sistem kembali ke langkah nomer 2 dari aliran utama.	

Aliran Alternatif 2: Menekan link refresh petugas	
(Administrator pada langkah nomer 4 aliran utama menekan tombol refresh.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1 Sistem mererefresh tampilan dari daftar petugas.
Aturan khusus	
1. Masukan keyword tidak boleh kosong.	

Sumber : [Analisis]

4.1.2.2 Use Case Diagram untuk Modul Pengaturan Instansi Client

Gambar 4.6 adalah *use case diagram* untuk modul pengaturan instansi *client* sistem psikotes *online*. Diagram *use case* ini melibatkan 1 buah aktor dan 5 buah *use case*. Kelima *use case* yang termasuk dalam modul pengaturan instansi *client* sistem psikotes *online* dapat dilihat pada Tabel 4.3. Aktor yang terlibat dalam modul pengaturan instansi *client* sistem psikotes *online* adalah Administrator.



Gambar 4.6 : Use Case Diagram untuk Modul Pengaturan Instansi Client Sistem Psikotes Online

Sumber : [Analisis]

1. Case Specification Menambah Data Instansi Client

Tabel 4.12 : Case Specification Menambah Data Instansi Client

Nama use case	Menambah data instansi <i>client</i>	
Aktor	Administrator	
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah data instansi <i>client</i> pengguna jasa. Data yang harus dicatat dari penambahan instansi adalah nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, <i>email</i> , fax, dan <i>website</i> .	
Pra-kondisi	Aktor user telah login sebagai aktor administrator.	
Pasca-kondisi	Data instansi <i>client</i> telah ditambahkan ke dalam sistem	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator menu "Tambah Instansi Client".	2. Sistem menampilkan <i>form</i> masukan yang berisi nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, <i>email</i> , fax, , dan <i>website</i> .	
3. Administrator memasukkan data untuk nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, <i>email</i> , fax, , dan <i>website</i> .	4. Sistem memvalidasi masukan nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon tidak kosong dan tidak ada di dalam sistem.	
	5. Sistem menjalankan <i>use case</i> melihat daftar instansi <i>client</i>	
Aliran Alternatif 1: Masukan nama instansi, alamat, bidang usaha, <i>contact person</i>, dan telepon kosong		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon	

	tidak kosong.
2. Kembali pada langkah nomer 2 aliran utama.	
Aliran Alternatif 2: nama dan alamat telah ada dalam sistem	
(Administrator pada langkah nomer 3 aliran utama memasukkan nama instansi dan alamat yang telah ada dalam sistem.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan nama instansi dan alamat tersebut telah ada dalam sistem.
2. Kembali pada langkah nomer 2 aliran utama.	
Aturan khusus	
1. Masukan nama instansi dan alamat tidak boleh kosong.	

Sumber : [Analisis]

2. Use Case Specification Mencari Data Instansi Client

Tabel 4.13 : Use Case Specification Mencari Data Instansi Client

Nama use case	Mencari data data instansi <i>client</i>
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data suatu instansi yang ada di dalam sistem. Pencarian data instansi berdasarkan nama instansi.
Pra-kondisi	Aktor user telah login sebagai aktor administrator.
Pasca-kondisi	Sistem menampilkan data instansi yang dicari oleh aktor administrator.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem

1. Aktor user memilih menu "Instansi Client".	2. Sistem menjalankan use case melihat daftar instansi <i>client</i> beserta form pencarian instansi <i>client</i> .
3. User memasukkan data untuk masukan "keyword" dan mengklik tombol "Find".	4. Sistem menampilkan data instansi yang dicari.

Aliran Alternatif 1: Masukan *keyword* kosong
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan *keyword*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa <i>keyword</i> tidak boleh kosong.
	2. Sistem kembali ke langkah nomer 2 dari aliran utama.

Aliran Alternatif 2: Menekan tombol refresh
(Administrator pada langkah nomer 4 aliran utama menekan tombol refresh.)

Aksi dari Aktor	Tanggapan dari Sistem
	1 Sistem mererefresh tampilan dari daftar instansi.

Aturan khusus

1. Masukan *keyword* tidak boleh kosong.

Sumber : [Analisis]

3. Use Case Specification Melihat Data Instansi Client

Tabel 4.14 : Use Case Specification Melihat Data Instansi Client

Nama use case	Melihat data instansi <i>client</i>
----------------------	-------------------------------------

Aktor	Administrator
Deskripsi	Sistem harus dapat menampilkan daftar instansi <i>client</i> pengguna jasa. Data yang ditampilkan berupa nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, fax, dan website.
Pra-kondisi	Aktor user telah login sebagai aktor administrator.
Pasca-kondisi	Sistem menampilkan data instansi <i>client</i>

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik menu "Instansi <i>Client</i> ".	2. Sistem menampilkan daftar nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, fax, dan website.

Sumber : [Analisis]

4. Use Case Specification Mengedit Data Instansi Client

Tabel 4.15 : Use Case Specification Mengedit Data Instansi Client

Nama use case	Mengedit data instansi <i>client</i>
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data instansi <i>client</i> yang telah dimasukkan. Data yang dapat diedit adalah nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, <i>email</i> , fax, dan <i>website</i> .
Pra-kondisi	Administrator telah menjalankan <i>use case</i> Melihat daftar instansi <i>client</i>
Pasca-kondisi	Data instansi <i>client</i> yang telah diedit disimpan ke dalam sistem

Aliran Utama



Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari daftar koleksi yang ditampilkan.	2. Sistem menampilkan nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, <i>email</i> , fax, dan <i>website</i> yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, <i>email</i> , fax, dan <i>website</i> .	4. Sistem memvalidasi masukan bahwa nama instansi atau alamat, atau <i>contact person</i> atau telepon tidak kosong dan tidak ada dalam sistem.
	5. Sistem menjalankan <i>use case</i> melihat daftar instansi <i>client</i> .
Aliran Alternatif 1: Masukan nama instansi, alamat, <i>contact person</i>, telepon kosong	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama instansi, alamat, <i>contact person</i> , telepon tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
Aliran Alternatif 2: nama instansi dan alamat telah ada dalam sistem	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama instansi dan alamat tersebut telah ada dalam sistem.

	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukkan nama instansi, alamat, <i>contact person</i> , telepon tidak boleh kosong.	

Sumber : [Analisis]

5. Use Case Specification Menghapus Data Instansi Client

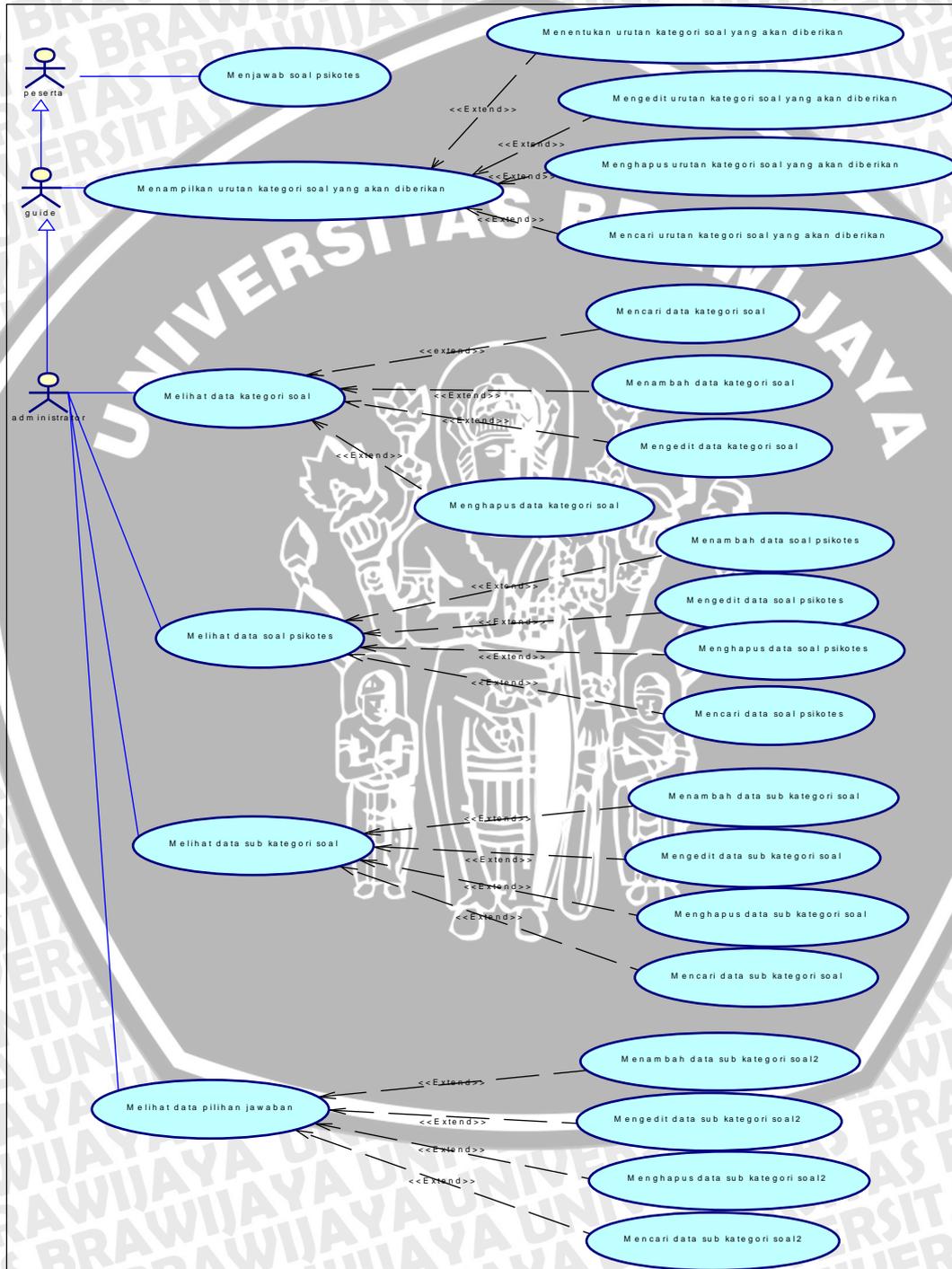
Tabel 4.16 : Use Case Specification Menghapus Data Instansi Client

Nama use case	Menghapus data instansi <i>client</i>
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data instansi tertentu.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> Mencari data instansi <i>client</i> atau telah menjalankan <i>use case</i> Melihat daftar instansi <i>client</i> .
Pasca-kondisi	Data instansi dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "delete" salah satu baris dari daftar instansi yang ditampilkan dari <i>use case</i> Mencari data instansi atau mengklik tombol "delete" saat menjalankan <i>use case</i> Menampilkan daftar instansi <i>client</i> .	2. Sistem menampilkan pesan bahwa data instansi berhasil dihapus.

Sumber : [Analisis]

4.1.2.2.3 Use Case Diagram untuk Modul Pengaturan Soal Tes

Gambar 4.7 adalah *use case diagram* untuk modul pengaturan soal tes pada sistem psikotes *online*. Diagram *use case* ini melibatkan tiga buah aktor dan 25 buah *use case*. Dua puluh lima *use case* yang termasuk dalam pengaturan soal tes pada sistem psikotes *online* dapat dilihat pada Tabel 4.3.



Gambar 4.7 : Use Case Diagram untuk Modul Pengaturan Soal Tes

Sumber : [Analisis]

1. Use Case Specification Menambah Data Kategori Soal

Tabel 4.17 : Use Case Specification Menambah Data Kategori Soal

Nama use case	Menambah data kategori soal	
Aktor	Administrator	
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah kategori soal. Data yang harus dicatat dari penambahan kategori adalah nama kategori, tujuan, dan keterangan.	
Pra-kondisi	Administrator telah menjalankan <i>use case</i> menampilkan daftar kategori soal.	
Pasca-kondisi	Data kategori soal telah ditambahkan ke dalam sistem	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator memilih link "Tambah Kategori".	2. Sistem menampilkan <i>form</i> masukan yang berisi nama kategori, tujuan, dan keterangan.	
3. Administrator memasukkan data untuk nama kategori, tujuan, dan keterangan.	4. Sistem memvalidasi masukan nama kategori tidak boleh kosong dan nama kategori tersebut belum ada dalam sistem.	
	5. Sistem menjalankan <i>use case</i> melihat daftar kategori soal.	
Aliran Alternatif 1: Masukan nama kategori kosong		
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan nama kategori)		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. Sistem menampilkan pesan kesalahan bahwa nama kategori tidak boleh kosong.	

	2. Kembali pada langkah nomer 2 aliran utama.
Aliran Alternatif 2: Nama kategori telah ada dalam sistem	
(Administrator pada langkah nomer 3 aliran utama memasukkan nama kategori yang telah ada dalam sistem.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama kategori tersebut telah ada dalam sistem.
2. Kembali pada langkah nomer 2 aliran utama.	
Aturan khusus	
1. Masukan nama kategori tidak boleh kosong.	

Sumber : [Analisis]

2. Use Case Specification Melihat Data Kategori Soal

Tabel 4.18 : Use Case Specification Melihat Daftar Kategori Soal

Nama use case	Melihat daftar kategori soal
Aktor	Administrator
Deskripsi	Sistem harus dapat menampilkan kategori soal yang tercatat di dalam sistem. Data yang ditampilkan berupa nama kategori, tujuan, dan keterangan.
Pra-kondisi	Sistem telah berjalan dan aktor user telah login sebagai administrator.
Pasca-kondisi	Sistem menampilkan daftar kategori soal
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "kategori	2. Sistem menampilkan kategori soal

soal”.	yang tercatat di dalam sistem. Data yang ditampilkan berupa nama kategori, tujuan, dan keterangan.
--------	--

Sumber : [Analisis]

3. Use Case Specification Mengedit Data Kategori Soal

Tabel 4.19 : Use Case Specification Mengedit Data Kategori Soal

Nama use case	Mengedit data kategori soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data kategori soal yang telah dimasukkan. Data yang dapat diedit adalah nama kategori, tujuan, dan keterangan.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> Melihat data kategori soal.
Pasca-kondisi	Data kategori soal yang telah diedit disimpan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari daftar kategori soal yang ditampilkan.	2. Sistem menampilkan nama kategori, tujuan, dan keterangan dari kategori soal yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk nama kategori, tujuan, dan keterangan.	4. Sistem memvalidasi masukan bahwa nama kategori tidak boleh kosong dan nama kategori tersebut belum ada dalam sistem.
	5. Sistem menjalankan <i>use case</i> melihat daftar kategori soal.

Aliran Alternatif 1: Masukan nama kategori kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan nama kategori.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama kategori tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Aliran Alternatif 2: Nama kategori telah ada dalam sistem

(Administrator pada langkah nomer 3 aliran utama memasukkan nama kategori yang telah ada dalam sistem.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama kategori tersebut telah ada dalam sistem.
	2. Kembali pada langkah nomer 2 aliran utama.

Aturan khusus

1. Masukan nama kategori tidak boleh kosong.

Sumber : [Analisis]

4. Use Case Specification Menghapus Data Kategori Soal

Tabel 4.20 : Use Case Specification Menghapus Data Kategori Soal

Nama use case	Menghapus data kategori soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus suatu data kategori tertentu.
Pra-kondisi	Administrator telah menjalankan use case melihat data



	kategori soal
Pasca-kondisi	Data kategori yang dipilih telah dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "delete" salah satu baris dari daftar kategori soal yang ditampilkan dari <i>use case</i> .	2. Sistem menjalankan <i>use case</i> melihat daftar kategori soal.

Sumber : [Analisis]

5. Use Case Specification Mencari Data Kategori Soal

Tabel 4.21 : Use Case Specification Mencari Data Kategori Soal

Nama use case	Mencari data kategori soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data kategori soal. Pencarian data berdasarkan nama kategori.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> melihat kategori soal.
Pasca-kondisi	Sistem menampilkan data kategori soal yang dicari oleh administrator.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Kategori Soal".	2. Sistem menampilkan <i>form</i> pencarian dan daftar keseluruhan data kategori soal yang telah ada di dalam sistem dan sebuah tombol refresh. <i>Form</i> pencarian memiliki masukan untuk "keyword" kategori soal.

3. Administrator memasukkan data untuk masukan “keyword” mengklik tombol “Finds”.	4. Sistem menampilkan data kategori soal hasil pencarian
Aliran Alternatif 1: Masukan <i>keyword</i> kategori soal kosong (Administrator pada langkah nomer 3 aliran utama tidak memasukkan <i>keyword</i> .)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan ” <i>keyword</i> ” tidak boleh kosong.
Aliran Alternatif 2: Menekan tombol refresh (Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1 Sistem menampilkan data kategori soal yang telah <i>direfresh</i> .
Aturan khusus	
1. Masukan <i>keyword</i> kategori soal tidak boleh kosong.	

Sumber : [Analisis]

6. Use Case Specification Menambah Urutan Kategori Soal Yang Diberikan

Tabel 4.22 : Use Case Specification Menambah Urutan Kategori Soal yang diberikan

Nama use case	Menambah urutan kategori soal yang diberikan
Aktor	Administrator, guide
Deskripsi	Sistem harus menyediakan fasilitas untuk menentukan urutan kategori soal yang akan diberikan pada peserta. Data yang dimasukkan adalah lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> melihat daftar urutan kategori soal.

Pasca-kondisi	Sistem menyimpan urutan kategori soal.	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator memilih menu "data urutan kategori soal".	2. Sistem menampilkan <i>form</i> pengurutan kategori soal berupa masukan lokasi tes, nomorurut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.	
3. Administrator memasukkan data lokasi tes, nomorurut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.	4. Sistem memvalidasi masukan data lokasi tes, nomorurut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.	
Aliran Alternatif 1: Masukan lokasi tes, nomorurut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong		
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan lokasi tes, nomorurut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan)		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. Sistem menampilkan seluruh data urutan kategori soal yang ada dalam sistem.	
Aturan khusus		
1. Masukan lokasi tes, nomorurut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.		

Sumber : [Analisis]

7. Use Case Specification Mengedit Urutan Dan Bobot Kategori Soal yang Diberikan

Tabel 4.23 : Use Case Specification Mengedit Urutan Kategori Soal yang Diberikan

Nama use case	Mengedit urutan kategori soal yang diberikan
Aktor	Administrator, guide
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data urutan kategori soal yang telah dimasukkan. Data yang dapat diedit adalah lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> melihat daftar urutan kategori soal atau <i>use case</i> mencari urutan kategori soal.
Pasca-kondisi	Data urutan kategori soal yang telah diedit disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari daftar urutan kategori soal yang ditampilkan.	2. Sistem menampilkan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan yang telah dipilih oleh administrator.
3. Administrator mengedit data lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.	4. Sistem memvalidasi masukan bahwa lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tersebut tidak boleh kosong.
	5. Sistem menjalankan <i>use case</i> melihat daftar urutan kategori soal.
Aliran Alternatif 1: Masukan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong	

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Aturan khusus

1. Masukan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.

Sumber : [Analisis]

8. Use Case Specification Menghapus Data Urutan Kategori Soal yang Diberikan

Tabel 4.24 : Use Case Specification Menghapus Data Urutan Kategori Soal yang Diberikan

Nama use case	Menghapus data urutan kategori soal yang diberikan
Aktor	Administrator, guide
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus suatu data urutan kategori soal yang diberikan tertentu.
Pra-kondisi	Administrator telah menjalankan use case Mencari data urutan kategori soal atau telah menjalankan use case Melihat data urutan kategori soal yang diberikan.
Pasca-kondisi	Data urutan kategori soal dihapus dari dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
-----------------	-----------------------

1. Administrator mengklik tombol "delete" salah satu baris dari daftar urutan kategori soal yang diberikan yang ditampilkan dari <i>use case</i> Mencari data urutan kategori soal yang diberikan atau mengklik tombol "delete" saat menjalankan <i>use case</i> melihat daftar urutan kategori soal yang diberikan.	2. Sistem menjalankan <i>use case</i> melihat daftar urutan kategori soal.
--	--

Sumber : [Analisis]

9. Use Case Specification Mencari Data Urutan Kategori Soal yang Diberikan

Tabel 4.25 : *Use Case Specification* Mencari Data Urutan Kategori Soal yang Diberikan

Nama use case	Mencari data urutan kategori soal yang diberikan
Aktor	Administrator, guide
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data urutan kategori soal yang akan diberikan pada pelamar untuk instansi tertentu. Pencarian data berdasarkan nama instansi.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> melihat urutan kategori soal yang diberikan
Pasca-kondisi	Sistem menampilkan data urutan kategori soal yang dicari oleh aktor.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Urutan Kategori Soal".	2. Sistem menampilkan <i>form</i> pencarian dan daftar keseluruhan urutan kategori soal yang telah ada di dalam sistem dan sebuah tombol



	refresh. <i>Form</i> pencarian memiliki masukan untuk “keyword” lokasi tes atau instansi <i>client</i> .
3. Administrator memasukkan data untuk masukan “keyword” mengklik tombol “ <i>Find</i> ”.	4. Sistem menampilkan daftar urutan kategori soal hasil pencarian

Aliran Alternatif 1: Masukan *keyword* daftar urutan kategori soal tes kosong
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan *keyword*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan seluruh urutan kategori soal yang ada di dalam sistem.

Aliran Alternatif 2: Menekan tombol refresh
(Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)

Aksi dari Aktor	Tanggapan dari Sistem
	1 Sistem menampilkan daftar urutan kategori soal yang telah direfresh.

Aturan khusus	
1. Masukan keyword tempat atau instansi <i>client</i> penyelenggaraan tes tidak boleh kosong.	

Sumber : [Analisis]

10. Use Case Specification Melihat Data Urutan Kategori Soal yang Diberikan

Tabel 4.26 : Use Case Specification Melihat Data Urutan Kategori Soal yang Diberikan

Nama use case	Melihat data urutan kategori soal yang diberikan
Aktor	Administrator, guide

Deskripsi	Sistem harus dapat menampilkan urutan kategori soal yang tercatat di dalam sistem untuk pelamar instansi tertentu. Data yang ditampilkan berupa lokasi tes,waktu, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.
Pra-kondisi	Sistem telah berjalan dan <i>user</i> telah <i>login</i> sebagai Administrator
Pasca-kondisi	Sistem menampilkan data lokasi tes,waktu, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan yang telah ada dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik menu "urutan kategori soal"	2. Sistem menampilkan data lokasi tes, waktu, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.

Sumber : [Analisis]

11. Use Case Specification Menambah Data Soal Psikotes

Tabel 4.27 : Use Case Specification Menambah Data Soal Psikotes

Nama use case	Menambah data soal psikotes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah data soal – soal psikotes yang diberikan pada pelamar. Data yang dapat ditambahkan adalah soal.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> melihat data soal psikotes.
Pasca-kondisi	Data soal psikotes telah ditambahkan ke dalam sistem

Aliran Utama



Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih link "Tambah Soal".	2. Sistem menampilkan <i>form</i> masukan yang berisi soal
3. Administrator memasukkan data untuk soal.	4. Sistem menjalankan <i>use case</i> melihat daftar data soal psikotes.

Aliran Alternatif 1: Masukan soal kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan soal dan pilihan jawaban.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa soal tidak boleh kosong.
2. Kembali pada langkah nomer 2 aliran utama.	

Aturan khusus

- Masukan soal dan pilihan jawaban tidak boleh kosong.

Sumber : [Analisis]

17. Use Case Specification Melihat Data Soal Psikotes

Tabel 4.28 : *Use Case Specification* Melihat Data Soal Psikotes

Nama use case	Melihat data soal psikotes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menampilkan data soal – soal psikotes yang diberikan pada pelamar. Data yang dapat ditampilkan adalah soal.
Pra-kondisi	Sistem telah berjalan dan <i>user</i> telah <i>login</i> sebagai Administrator
Pasca-kondisi	Sistem menampilkan data soal.

Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "soal"	2. Sistem menampilkan soal per sub kategori.

Sumber : [Analisis]

18. Use Case Specification Mengedit Data Soal Psikotes

Tabel 4.29 : Use Case Specification Mengedit Data Soal Psikotes

Nama use case	Mengedit data soal psikotes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data soal – soal psikotes yang diberikan pada pelamar. Data yang dapat diedit adalah soal.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> mencari data soal psikotes atau menampilkan data soal psikotes.
Pasca-kondisi	Data soal psikotes yang telah diedit disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari data soal yang ditampilkan.	2. Sistem menampilkan soal.
3. Administrator mengedit soal.	4. Sistem memvalidasi masukan soal tersebut tidak boleh kosong.
	5. Sistem menjalankan <i>use case</i> melihat daftar soal psikotes.
Aliran Alternatif 1: Masukan soal kosong (Administrator pada langkah nomer 3 aliran utama tidak memasukkan soal)	

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan soal tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukan soal tidak boleh kosong.	

Sumber : [Analisis]

19. Use Case Specification Menghapus Data Soal Psikotes

Tabel 4.30 : Use Case Specification Menghapus Data Soal Psikotes

Nama use case	Menghapus data soal psikotes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data soal – soal psikotes yang diberikan pada peserta tes.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> Mencari data soal psikotes atau telah menjalankan <i>use case</i> Menampilkan data soal psikotes.
Pasca-kondisi	Data soal psikotes dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "delete" salah satu baris dari daftar soal psikotes yang diberikan yang ditampilkan dari <i>use case</i> mencari data soal psikotes atau mengklik tombol "delete" saat menjalankan <i>use case</i> menampilkan data soal psikotes	2. Sistem menjalankan <i>use case</i> melihat daftar data soal psikotes.

Sumber : [Analisis]

20. Use Case Specification Mencari Data Soal Psikotes

Tabel 4.31 : Use Case Specification Mencari Data Soal Psikotes

Nama use case	Mencari data soal psikotes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data soal – soal psikotes yang diberikan pada pelamar. Pencarian data berdasarkan kutipan kalimat soal.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> melihat data soal psikotes.
Pasca-kondisi	Sistem menampilkan data soal psikotes yang dicari oleh administrator.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "data soal psikotes".	2. Sistem menampilkan <i>form</i> pencarian dan daftar keseluruhan data soal psikotes yang telah ada di dalam sistem dan sebuah tombol refresh. <i>Form</i> pencarian memiliki masukan untuk "keyword" soal.
3. Administrator memasukkan data untuk masukan "keyword" mengklik tombol "Finds".	4. Sistem menampilkan data soal psikotes hasil pencarian
Aliran Alternatif 1: Masukan <i>keyword</i> soal tes kosong (Administrator pada langkah nomer 3 aliran utama tidak memasukkan <i>keyword</i> .)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan "keyword" tidak boleh kosong.

Aliran Alternatif 2: Menekan tombol refresh	
(Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1 Sistem menampilkan data soal psikotes yang telah direfresh.
Aturan khusus	
1. Masukan keyword soal tidak boleh kosong.	

Sumber : [Analisis]

21. Use Case Specification Menjawab Soal Psikotes

Tabel 4.32 : Use Case Specification Menjawab Soal Psikotes

Nama use case	Menjawab soal psikotes
Aktor	Peserta
Deskripsi	Sistem harus menyediakan fasilitas untuk menjawab soal psikotes berdasar sub kategori soal.
Pra-kondisi	Peserta telah menjalankan <i>use case</i> melihat data soal psikotes.
Pasca-kondisi	Data soal psikotes telah terjawab dan tersimpan di dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Peserta menekan tombol "Mulai".	2. Sistem menampilkan <i>form</i> masukan yang berisi soal, <i>radio button</i> pilihan jawaban.
3. Peserta menjawab soal dengan memilih salah satu pilihan jawaban yang tersedia.	4. Sistem menyimpan jawaban soal.

Sumber : [Analisis]

22. Use Case Specification Menambah Data Sub Kategori Soal

Tabel 4.33 : Use Case Specification Menambah Data Sub Kategori Soal

Nama use case	Menambah data sub kategori soal	
Aktor	Administrator	
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah sub kategori soal. Data yang harus dicatat dari penambahan sub kategori.	
Pra-kondisi	Administrator telah menjalankan <i>use case</i> menampilkan daftar sub kategori soal.	
Pasca-kondisi	Data sub kategori soal telah ditambahkan ke dalam sistem	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator memilih link "Tambah Sub Kategori".	2. Sistem menampilkan <i>form</i> masukan yang berisi sub kategori.	
3. Administrator memasukkan data untuk sub kategori.	4. Sistem memvalidasi masukan sub kategori tidak boleh kosong dan sub kategori tersebut belum ada dalam sistem.	
	5. Sistem menjalankan <i>use case</i> melihat daftar sub kategori soal.	
Aliran Alternatif 1: Masukan nama sub kategori kosong		
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan sub kategori)		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. Sistem menampilkan pesan kesalahan bahwa sub kategori tidak boleh kosong.	
	2. Kembali pada langkah nomer 2	

	aliran utama.
Aliran Alternatif 2: Sub kategori telah ada dalam sistem	
(Administrator pada langkah nomer 3 aliran utama memasukkan sub kategori yang telah ada dalam sistem.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa sub kategori tersebut telah ada dalam sistem.
2. Kembali pada langkah nomer 2 aliran utama.	
Aturan khusus	
1. Masukan sub kategori tidak boleh kosong.	

Sumber : [Analisis]

23. Use Case Specification Melihat Data Sub Kategori Soal

Tabel 4.34 : Use Case Specification Melihat Daftar Sub Kategori Soal

Nama use case	Melihat daftar sub kategori soal
Aktor	Administrator
Deskripsi	Sistem harus dapat menampilkan sub kategori soal yang tercatat di dalam sistem. Data yang ditampilkan berupa sub kategori.
Pra-kondisi	Sistem telah berjalan dan aktor user telah login sebagai administrator.
Pasca-kondisi	Sistem menampilkan daftar sub kategori soal

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Sub Kategori".	2. Sistem menampilkan sub kategori soal yang tercatat di dalam sistem.

	Data yang ditampilkan berupa sub kategori.
--	--

Sumber : [Analisis]

24. Use Case Specification Mengedit Data Sub Kategori Soal

Tabel 4.35 : Use Case Specification Mengedit Data Sub Kategori Soal

Nama use case	Mengedit data sub kategori soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data sub kategori soal yang telah dimasukkan. Data yang dapat diedit adalah sub kategori.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> melihat data sub kategori soal.
Pasca-kondisi	Data sub kategori soal yang telah diedit disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari daftar sub kategori soal yang ditampilkan.	2. Sistem menampilkan sub kategori dari sub kategori soal yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk sub kategori.	4. Sistem memvalidasi masukan bahwa sub kategori tidak boleh kosong dan sub kategori tersebut belum ada dalam sistem.
	5. Sistem menjalankan <i>use case</i> melihat daftar sub kategori soal.
Aliran Alternatif 1: Masukan nama sub kategori kosong (Administrator pada langkah nomer 3 aliran utama tidak memasukkan sub kategori.)	

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa sub kategori tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Aliran Alternatif 2: Sub kategori telah ada dalam sistem

(Administrator pada langkah nomer 3 aliran utama memasukkan sub kategori yang telah ada dalam sistem.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa sub kategori tersebut telah ada dalam sistem.
	2. Kembali pada langkah nomer 2 aliran utama.

Aturan khusus

1. Masukan sub kategori tidak boleh kosong.

Sumber : [Analisis]

25. Use Case Specification Menghapus Data Sub Kategori Soal

Tabel 4.36 : Use Case Specification Menghapus Data Sub Kategori Soal

Nama use case	Menghapus data sub kategori soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus suatu data sub kategori tertentu.
Pra-kondisi	Administrator telah menjalankan use case melihat data sub kategori soal
Pasca-kondisi	Data sub kategori yang dipilih telah dihapus dari dalam

sistem	
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "delete" salah satu baris dari daftar sub kategori soal yang ditampilkan.	2. Sistem menjalankan <i>use case</i> melihat daftar sub kategori soal.

Sumber : [Analisis]

27. Use Case Specification Mencari Data Sub Kategori Soal

Tabel 4.37 : Use Case Specification Mencari Data Sub Kategori Soal

Nama use case	Mencari data sub kategori soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data sub kategori soal. Pencarian data berdasarkan sub kategori.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> melihat sub kategori soal.
Pasca-kondisi	Sistem menampilkan data sub kategori soal yang dicari oleh administrator.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Sub Kategori".	2. Sistem menampilkan <i>form</i> pencarian dan daftar keseluruhan data sub kategori yang telah ada di dalam sistem dan sebuah tombol refresh. <i>Form</i> pencarian memiliki masukan untuk "keyword" sub kategori soal.
3. Administrator memasukkan data untuk masukan "keyword" mengklik tombol	4. Sistem menampilkan data sub kategori soal hasil pencarian

"Finds".	
Aliran Alternatif 1: Masukan keyword kategori soal kosong (Administrator pada langkah nomer 3 aliran utama tidak memasukkan <i>keyword</i> .)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan " <i>keyword</i> " tidak boleh kosong.
Aliran Alternatif 2: Menekan tombol refresh (Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan data sub kategori soal yang telah <i>direfresh</i> .
Aturan khusus	
1. Masukan <i>keyword</i> sub kategori soal tidak boleh kosong.	

Sumber : [Analisis]

28. Use Case Specification Menambah Pilihan Jawaban Soal

Tabel 4.38 : Use Case Specification Menambah Pilihan Jawaban Soal

Nama use case	Menambah pilihan jawaban soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah pilihan jawaban soal. Data yang harus dicatat dari penambahan adalah huruf pilihan, jawaban, dan status jawaban.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> menampilkan daftar pilihan jawaban soal.
Pasca-kondisi	Pilihan jawaban soal telah ditambahkan ke dalam sistem
Aliran Utama	

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih link "Tambah Pilihan Jawaban".	2. Sistem menampilkan <i>form</i> masukan yang berisi huruf pilihan, jawaban, dan status jawaban.
3. Administrator memasukkan data untuk huruf pilihan, jawaban, dan status jawaban.	4. Sistem memvalidasi masukan huruf pilihan, jawaban, dan status jawaban tidak boleh kosong dan huruf pilihan tersebut belum ada dalam sistem.
	5. Sistem menjalankan <i>use case</i> melihat pilihan jawaban soal.
<p align="center">Aliran Alternatif 1: Masukan huruf pilihan, jawaban, dan status jawaban kosong</p> <p>(Administrator pada langkah nomer 3 aliran utama tidak memasukkan huruf pilihan, jawaban, dan status jawaban)</p>	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
<p align="center">Aliran Alternatif 2: huruf pilihan telah ada dalam sistem</p> <p>(Administrator pada langkah nomer 3 aliran utama memasukkan huruf pilihan yang telah ada dalam sistem.)</p>	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa huruf pilihan

	tersebut telah ada dalam sistem.
2. Kembali pada langkah nomer 2 aliran utama.	
Aturan khusus	
1. Masukkan huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.	

Sumber : [Analisis]

29. Use Case Specification Melihat Pilihan Jawaban Soal

Tabel 4.39 : Use Case Specification Melihat Pilihan Jawaban Soal

Nama use case	Melihat daftar pilihan jawaban soal
Aktor	Administrator
Deskripsi	Sistem harus dapat menampilkan pilihan jawaban soal yang tercatat di dalam sistem. Data yang ditampilkan berupa huruf pilihan, jawaban, dan status jawaban.
Pra-kondisi	Sistem telah berjalan dan aktor user telah login sebagai administrator.
Pasca-kondisi	Sistem menampilkan daftar pilihan jawaban soal
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Pilihan Jawaban".	2. Sistem menampilkan pilihan jawaban soal yang tercatat di dalam sistem. Data yang ditampilkan berupa huruf pilihan, jawaban, dan status jawaban.

Sumber : [Analisis]

30. Use Case Specification Mengedit Pilihan Jawaban Soal

Tabel 4.40 : Use Case Specification Mengedit Pilihan Jawaban Soal

Nama use case	Mengedit data pilihan jawaban soal
----------------------	------------------------------------

Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit pilihan jawaban soal yang telah dimasukkan. Data yang dapat diedit adalah huruf pilihan, jawaban, dan status jawaban.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> melihat pilihan jawaban soal
Pasca-kondisi	huruf pilihan, jawaban, dan status jawaban soal yang telah diedit disimpan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik <i>link</i> "edit" salah satu baris dari daftar pilihan jawaban soal yang ditampilkan.	2. Sistem menampilkan pilihan jawaban soal yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk huruf pilihan, jawaban, dan status jawaban.	4. Sistem memvalidasi masukan bahwa huruf pilihan, jawaban, dan status jawaban tidak boleh kosong dan huruf pilihan tersebut belum ada dalam sistem.
	5. Sistem menjalankan <i>use case</i> melihat pilihan jawaban soal.

Aliran Alternatif 1: Masukan huruf pilihan, jawaban, dan status jawaban kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan huruf pilihan, jawaban, dan status jawaban)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa huruf pilihan, jawaban, dan status jawaban tidak

	boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
Aliran Alternatif 2: huruf pilihan telah ada dalam sistem	
(Administrator pada langkah nomer 3 aliran utama memasukkan huruf pilihan yang telah ada dalam sistem.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa huruf pilihan tersebut telah ada dalam sistem.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukan huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.	

Sumber : [Analisis]

31. Use Case Specification Menghapus Pilihan Jawaban Soal

Tabel 4.41 : Use Case Specification Menghapus Pilihan Jawaban Soal

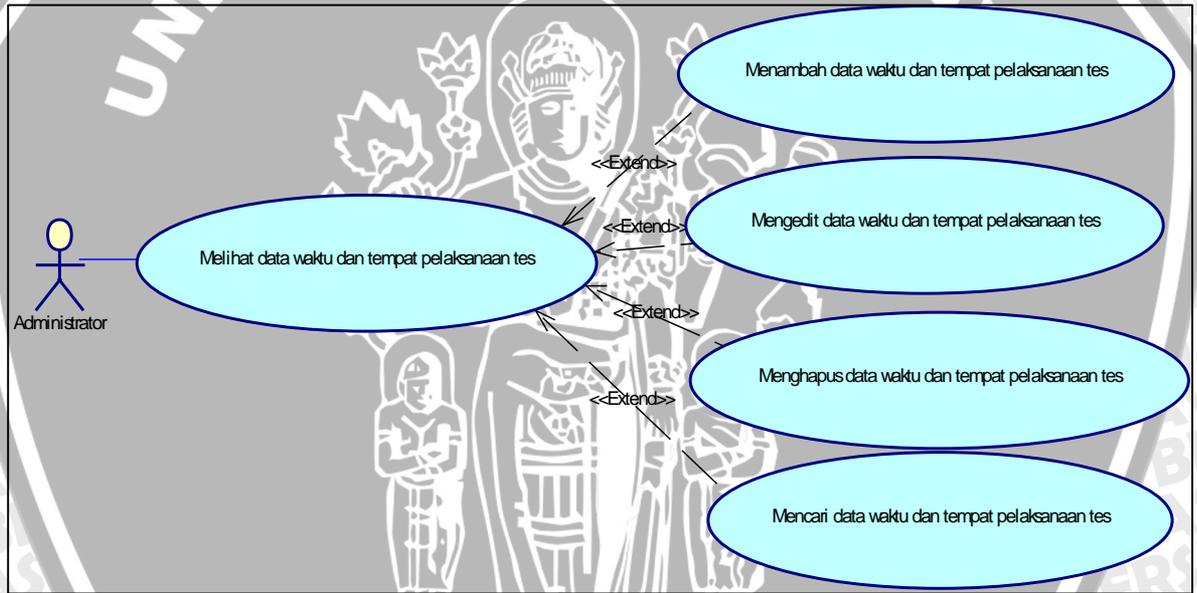
Nama use case	Menghapus pilihan jawaban soal
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus pilihan jawaban tertentu.
Pra-kondisi	Administrator telah menjalankan use case melihat pilihan jawaban soal
Pasca-kondisi	Pilihan jawaban yang dipilih telah dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem

1. Administrator mengklik tombol "delete" salah satu baris dari daftar pilihan jawaban soal yang ditampilkan.	2. Sistem menjalankan <i>use case</i> melihat daftar pilihan jawaban soal.
---	--

Sumber : [Analisis]

4.1.2.2.4 Use Case Diagram untuk Modul Pengaturan Penyelenggaraan Tes

Gambar 4.8 adalah *use case diagram* untuk modul pengaturan penyelenggaraan tes *software* psikotes online. Diagram *use case* ini melibatkan sebuah aktor dan 5 buah *use case*. Kelima *use case* yang termasuk dalam modul pengaturan penyelenggaraan tes *software* psikotes online dapat dilihat pada Tabel 4.3. Aktor yang terlibat dalam modul pengaturan penyelenggaraan tes *software* psikotes online adalah administrator.



Gambar 4.8 : *Use Case Diagram* untuk modul pengaturan penyelenggaraan tes

Sumber : [Analisis]

1. Use Case Specification Menambah Data Waktu dan Tempat Pelaksanaan Tes

Tabel 4.42 : *Use Case Specification* Menambah Data Waktu dan Tempat Pelaksanaan Tes

Nama use case	Menambah data waktu dan tempat pelaksanaan tes
Aktor	Administrator



Deskripsi	Sistem harus menyediakan fasilitas untuk menambah data waktu dan tempat pelaksanaan psikotes untuk peserta. Data yang dapat ditambahkan adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> mencari data waktu dan tempat pelaksanaan tes atau melihat data waktu dan tempat pelaksanaan tes.
Pasca- kondisi	Data waktu dan tempat pelaksanaan tes telah ditambahkan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih link "Tambah Pelaksanaan Tes".	2. Sistem menampilkan <i>form</i> masukan yang berupa <i>field</i> tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.
3. Administrator memasukkan data untuk tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.	4. Sistem memvalidasi masukan waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.
	5. Sistem menjalankan <i>use case</i> melihat daftar data waktu dan tempat pelaksanaan tes.

Aliran Alternatif 1: Masukan waktu, tempat, dan instansi *client* kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan waktu, tempat, dan instansi *client*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan



	kesalahan bahwa waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukkan waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.	

Sumber : [Analisis]

2. Use Case Specification Mencari Data Waktu dan Tempat Pelaksanaan Tes

Tabel 4.43 : Use Case Specification Mencari Data Waktu dan Tempat Pelaksanaan Tes

Nama use case	Mencari data waktu dan tempat pelaksanaan tes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data waktu dan tempat pelaksanaan psikotes untuk peserta. Pencarian data berdasar tempat dan nama instansi.
Pra- kondisi	Administrator telah menjalankan <i>use case</i> melihat data waktu dan tempat pelaksanaan tes.
Pasca- kondisi	Sistem menampilkan data waktu dan tempat pelaksanaan tes yang dicari administrator.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Pelaksanaan Tes".	2. Sistem menampilkan <i>form</i> . <i>Form</i> pencarian memiliki masukan untuk "keyword".
3. Administrator memasukkan data untuk masukan "keyword" dan mengklik tombol " <i>Find</i> ".	4. Sistem menampilkan data tanggal, tempat, alamat, dan nama instansi hasil pencarian.



Aliran Alternatif 1: Masukan <i>keyword</i> kosong	
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan <i>keyword</i> .)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan bahwa <i>keyword</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
Aliran Alternatif 2: Menekan tombol refresh	
(Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem mererefresh data waktu dan tempat pelaksanaan psikotes.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukan <i>keyword</i> tidak boleh kosong.	

Sumber : [Analisis]

3. Use Case Specification Mengedit Data Waktu dan Tempat Pelaksanaan Psikotes

Tabel 4.44 : Use Case Specification Mengedit Data Waktu dan Tempat Pelaksanaan Psikotes

Nama use case	Mengedit data waktu dan tempat pelaksanaan psikotes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data waktu dan tempat pelaksanaan psikotes untuk peserta. Data yang dapat diedit adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.

Pra- kondisi	Administrator telah menjalankan <i>use case</i> mencari atau melihat data waktu dan tempat pelaksanaan tes
Pasca- kondisi	Data waktu dan tempat pelaksanaan tes yang telah diedit disimpan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari data waktu dan tempat pelaksanaan tes yang ditampilkan.	2. Sistem menampilkan tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2 yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.	4. Sistem memvalidasi masukan bahwa waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.
	5. Sistem menjalankan <i>use case</i> menampilkan daftar data waktu dan tempat pelaksanaan tes

Aliran Alternatif 1: Masukan waktu, tempat, dan instansi *client* kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan waktu, tempat, dan instansi *client*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.



Aturan khusus

1. Masukan waktu, tempat, dan instansi *client* tidak boleh kosong.

Sumber : [Analisis]

4. Use Case Specification Menghapus Data Waktu dan Tempat Pelaksanaan Psikotes

Tabel 4.45 : Use Case Specification Menghapus Data Waktu dan Tempat Pelaksanaan Psikotes

Nama use case	Menghapus data waktu dan tempat pelaksanaan psikotes
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data waktu dan tempat pelaksanaan psikotes.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> melihat data waktu dan tempat pelaksanaan tes atau telah menjalankan <i>use case</i> mencari data waktu dan tempat pelaksanaan tes.
Pasca-kondisi	Data waktu dan tempat pelaksanaan tes dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "delete" salah satu baris dari data waktu dan tempat pelaksanaan tes yang ditampilkan dari <i>use case</i> melihat waktu dan tempat pelaksanaan tes.	2. Sistem menjalankan <i>use case</i> melihat daftar waktu dan tempat pelaksanaan tes.

Sumber : [Analisis]

5. Use case specification Melihat Data Waktu dan Tempat Pelaksanaan Tes

Tabel 4.46 : Use Case Specification Melihat Data Waktu dan Tempat Pelaksanaan Tes

Nama use case	Melihat data waktu dan tempat pelaksanaan tes
----------------------	---

Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menampilkan data waktu dan tempat pelaksanaan psikotes untuk peserta. Data yang dapat ditampilkan adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.
Pra-kondisi	User telah <i>login</i> sebagai administrator.
Pasca-kondisi	Sistem menampilkan data waktu dan tempat pelaksanaan tes.

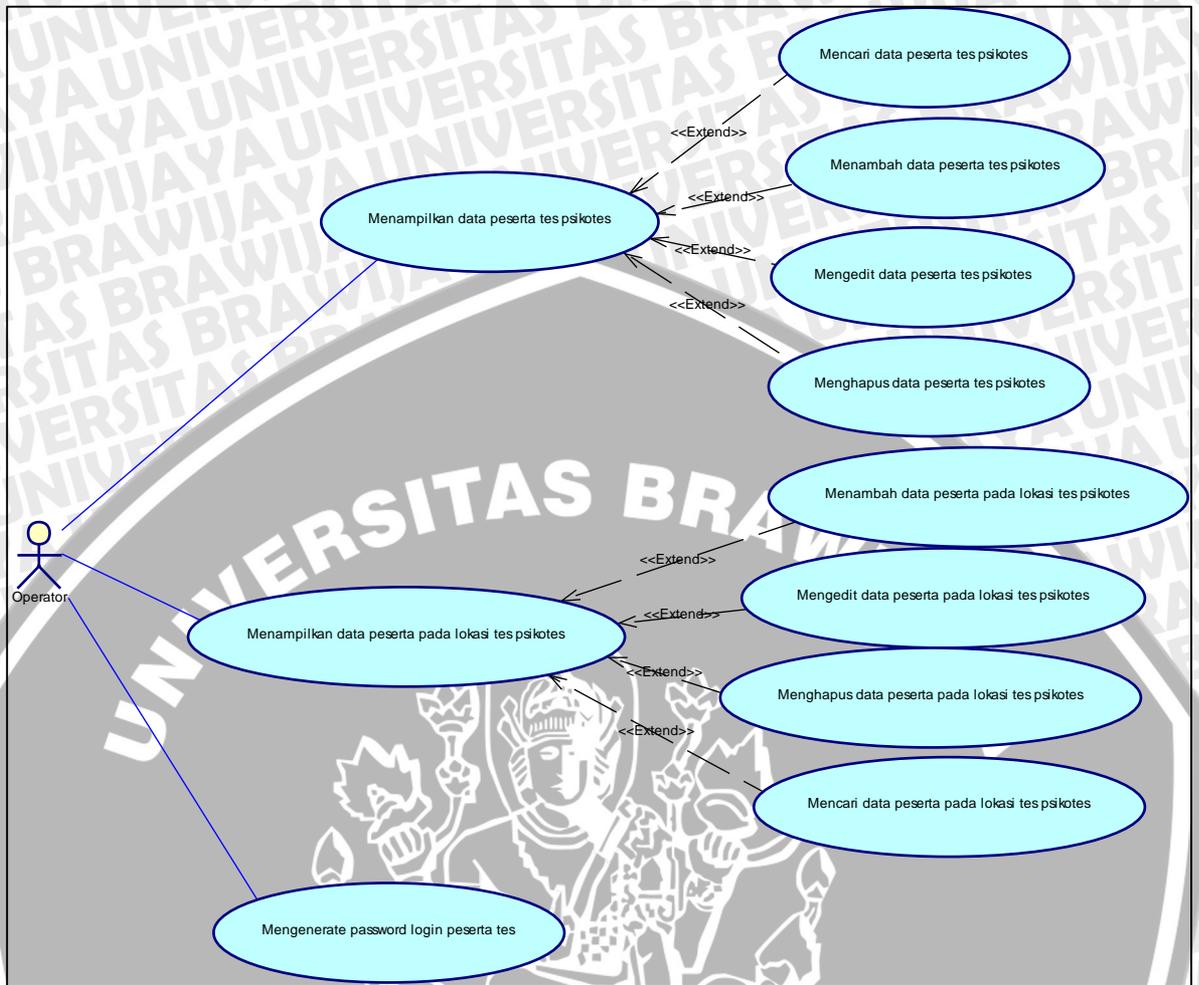
Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "Pelaksanaan Tes"	2. Sistem menampilkan data waktu dan tempat pelaksanaan psikotes untuk peserta. Data yang dapat ditampilkan adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.

Sumber : [Analisis]

4.1.2.2.5 Use Case Diagram untuk Modul Pengaturan Peserta Tes

Gambar 4.9 adalah *use case diagram* untuk modul pengaturan peserta tes *software* psikotes online. Diagram *use case* ini melibatkan dua aktor dan 11 buah *use case*. Sebelas *use case* yang termasuk dalam modul pengaturan peserta tes *software* psikotes online dapat dilihat pada Tabel 4.3. Aktor yang terlibat dalam modul pengaturan peserta tes *software* psikotes online adalah administrator dan operator.



Gambar 4.9 : Use Case Diagram untuk Modul Pengaturan Peserta Tes
Sumber : [Analisis]

1. Use Case Specification Menambah Data Peserta Psikotes

Tabel 4.47 : Use Case Specification Menambah Data Peserta Psikotes

Nama use case	Menambah data peserta psikotes
Aktor	Operator, administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah biodata peserta psikotes. Data yang dapat ditambahkan adalah id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin
Pra- kondisi	Administrator telah menjalankan use case mencari data peserta psikotes atau melihat data peserta psikotes

Pasca- kondisi	Data peserta psikotes telah ditambahkan ke dalam sistem	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator memilih link "Tambah Peserta".	2. Sistem menampilkan <i>form</i> masukan yang berupa <i>field</i> id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin	
3. Administrator memasukkan data untuk id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin	4. Sistem memvalidasi masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.	
	5. Sistem menjalankan <i>use case</i> melihat data peserta.	
<p>Aliran Alternatif 1: Masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon kosong</p> <p>(Administrator pada langkah nomer 3 aliran utama tidak memasukkan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon.)</p>		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. Sistem menampilkan pesan kesalahan bahwa id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.	
	2. Kembali pada langkah nomer 2 aliran utama.	
Aturan khusus		
1. Masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon		

tidak boleh kosong.

Sumber : [Analisis]

2. Use Case Specification Menampilkan Data Peserta Psikotes

Tabel 4.48 : Use Case Specification Menampilkan Data Peserta Psikotes

Nama use case	Menampilkan data peserta psikotes	
Aktor	Operator, administrator	
Deskripsi	Sistem harus menyediakan fasilitas untuk menampilkan biodata peserta psikotes. Data yang dapat ditampilkan adalah id peserta, nama, alamat, pendidikan terakhir, jurusan, dan jenis kelamin.	
Pra-kondisi	User telah <i>login</i> sebagai administrator.	
Pasca-kondisi	Sistem menampilkan data id peserta, nama, alamat, pendidikan terakhir, jurusan, dan jenis kelamin.	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator mengklik tombol "Data Peserta Tes"	2. Sistem menampilkan data id peserta, nama, alamat, pendidikan terakhir, jurusan, dan jenis kelamin.	

Sumber : [Analisis]

3. Use Case Specification Mengedit Data Peserta Psikotes

Tabel 4.49 : Use Case Specification Mengedit Data Peserta Psikotes

Nama use case	Mengedit data peserta psikotes
Aktor	Operator, administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit biodata peserta psikotes. Data yang dapat diedit adalah id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin.

Pra- kondisi	Administrator telah menjalankan <i>use case</i> mencari atau menampilkan data peserta psikotes
Pasca- kondisi	data peserta psikotes yang telah diedit disimpan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari data peserta psikotes yang ditampilkan.	2. Sistem menampilkan id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin.	4. Sistem memvalidasi masukan bahwa nama, alamat, pendidikan terakhir, jurusan, telepon, dan jenis kelamin tidak boleh kosong.
	5. Sistem menjalankan <i>use case</i> melihat daftar peserta tes.

Aliran Alternatif 1: Masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, dan jenis kelamin kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan nama, alamat, pendidikan terakhir, jurusan, telepon, dan jenis kelamin)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, dan jenis kelamin tidak boleh kosong.
	2. Kembali pada langkah nomer 2

	aliran utama.
Aturan khusus	
1. Masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, dan jenis kelamin tidak boleh kosong.	

Sumber : [Analisis]

4. Use Case Specification Menghapus Data Peserta Psikotes

Tabel 4.50 : Use Case Specification Menghapus Data Peserta Psikotes

Nama use case	Menghapus data peserta psikotes
Aktor	Operator, administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data peserta psikotes yang ada dalam database.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> menampilkan data peserta psikotes atau telah menjalankan <i>use case</i> mencari data peserta psikotes.
Pasca-kondisi	Data peserta psikotes dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "delete" salah satu baris dari data peserta psikotes yang ditampilkan dari <i>use case</i> menampilkan data peserta psikotes.	2. Sistem menjalankan <i>use case</i> melihat daftar peserta tes.

Sumber : [Analisis]

5. Use Case Specification Mencari Data Peserta Psikotes

Tabel 4.51 : Use Case Specification Mencari Data Peserta Psikotes

Nama use case	Mencari data peserta psikotes
Aktor	Operator, administrator

Deskripsi	Sistem harus menyediakan fasilitas untuk mencari biodata peserta psikotes. Pencarian data berdasarkan nama dan id peserta.
Pra- kondisi	Administrator telah menjalankan <i>use case</i> menampilkan data peserta psikotes.
Pasca- kondisi	Sistem menampilkan data peserta psikotes yang dicari administrator.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Data Peserta Tes".	2. Sistem menampilkan <i>form</i> . <i>Form</i> pencarian memiliki masukan untuk "keyword".
3. Administrator memasukkan data untuk masukan "keyword" dan mengklik tombol "Find".	4. Sistem menampilkan data peserta hasil pencarian.

Aliran Alternatif 1: Masukan keyword kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan *keyword*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan bahwa <i>keyword</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Aliran Alternatif 2: Menekan tombol refresh

(Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem mererefresh data peserta psikotes.
	2. Kembali pada langkah nomer 2

	aliran utama.
Aturan khusus	
1. Masukan <i>keyword</i> tidak boleh kosong.	

Sumber : [Analisis]

6. Use Case Specification Menambah Data Peserta pada Lokasi Psikotes

Tabel 4.52 : Use Case Specification Menambah Data Peserta Pada Lokasi Psikotes

Nama use case	Menambah data peserta pada lokasi psikotes	
Aktor	Administrator, operator	
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah peserta pada tempat dan waktu pelaksanaan psikotes. Data yang ditambahkan adalah id peserta,tempat tes, dan waktu tes.	
Pra- kondisi	Administrator telah menjalankan <i>use case</i> mencari data peserta pada lokasi psikotes atau melihat data peserta pada lokasi psikotes	
Pasca- kondisi	Data peserta pada lokasi psikotes telah ditambahkan ke dalam sistem	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Administrator memilih <i>link</i> "Tambah Peserta pada Lokasi Tes".	2. Sistem menampilkan <i>form</i> masukan yang berupa <i>field</i> id peserta, tempat tes, dan waktu tes.	
3. Administrator memasukkan data untuk id peserta,tempat tes, dan waktu tes.	4. Sistem memvalidasi masukan id peserta,tempat tes, dan waktu tes tidak boleh kosong.	
	5. Sistem menjalankan <i>use case</i> melihat data peserta di lokasi tes.	

Aliran Alternatif 1: Masukan id peserta,tempat tes, dan waktu tes kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan id peserta,tempat tes, dan waktu tes.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa id peserta,tempat tes, dan waktu tes tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Aturan khusus

1. Masukan id peserta, tempat tes, dan waktu tes tidak boleh kosong.

Sumber : [Analisis]

7. Use Case Specification Melihat Data Peserta pada Lokasi Psikotes

Tabel 4.53 : Use Case Specification Menampilkan Data Peserta Pada Lokasi Psikotes

Nama use case	Melihat data peserta pada lokasi psikotes
Aktor	Administrator, operator
Deskripsi	Sistem harus menyediakan fasilitas untuk menampilkan peserta pada tempat dan waktu pelaksanaan psikotes. Data yang ditampilkan adalah nama peserta, id peserta, tempat tes, waktu tes, dan <i>password login</i> peserta.
Pra-kondisi	User telah <i>login</i> sebagai administrator atau operator.
Pasca-kondisi	Sistem menampilkan data nama peserta, id peserta, tempat tes, waktu tes, dan <i>password login</i> peserta.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
-----------------	-----------------------

1. Administrator mengklik tombol "Data Peserta Tes pada Lokasi Tes"	2. Sistem menampilkan data nama peserta, id peserta, tempat tes, waktu tes, dan <i>password login</i> peserta.
---	--

Sumber : [Analisis]

8. Use Case Specification Mengedit Data Peserta pada Lokasi Psikotes

Tabel 4.54 : Use Case Specification Mengedit Data Peserta pada Lokasi Psikotes

Nama use case	Mengedit data peserta pada lokasi psikotes
Aktor	Administrator, operator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data peserta pada tempat dan waktu pelaksanaan psikotes. Data yang diedit adalah id peserta, tempat tes, dan waktu tes.
Pra- kondisi	Administrator telah menjalankan <i>use case</i> mencari atau menampilkan data peserta pada lokasi psikotes.
Pasca- kondisi	data peserta pada lokasi psikotes yang telah diedit disimpan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "edit" salah satu baris dari data peserta pada lokasi psikotes.	2. Sistem menampilkan data peserta pada lokasi psikotes yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk id peserta, tempat tes, dan waktu tes.	4. Sistem memvalidasi masukan bahwa id peserta, tempat tes, dan waktu tes tidak boleh kosong.
	5. Sistem menjalankan <i>use case</i> melihat data peserta di lokasi tes.

Aliran Alternatif 1: Masukan id peserta, tempat tes, dan waktu tes kosong
 (Administrator pada langkah nomer 3 aliran utama tidak memasukkan id

peserta,tempat tes,dan waktu tes)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa id peserta,tempat tes,dan waktu tes tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukan id peserta,tempat tes,dan waktu tes tidak boleh kosong.	

9. Use Case Specification Menghapus Data Peserta pada Lokasi Psikotes

Tabel 4.55 : Use Case Specification Menghapus Data Peserta pada Lokasi Psikotes

Nama use case	Menghapus data peserta pada lokasi psikotes
Aktor	Administrator, operator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data peserta pada lokasi dan tempat pelaksanaan psikotes.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> menampilkan data peserta pada lokasi psikotes atau telah menjalankan <i>use case</i> mencari data peserta pada lokasi psikotes.
Pasca-kondisi	Data peserta pada lokasi psikotes dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik tombol "hapus" salah satu baris dari data peserta pada lokasi psikotes yang ditampilkan dari <i>use case</i>	2. Sistem menampilkan pesan bahwa data peserta pada lokasi psikotes berhasil dihapus.

menampilkan data peserta pada lokasi psikotes.	
--	--

Sumber : [Analisis]

10. Use Case Specification Mencari Data Peserta pada Lokasi Psikotes

Tabel 4.56 : Use Case Specification Mencari Data Peserta pada Lokasi Psikotes
Sumber : [Analisis]

Nama use case	Mencari data peserta pada lokasi psikotes
Aktor	Administrator, operator
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari data peserta psikotes pada tempat dan waktu pelaksanaan tes yang ditentukan.
Pra- kondisi	Administrator telah menjalankan <i>use case</i> menampilkan data peserta pada lokasi psikotes.
Pasca- kondisi	Sistem menampilkan data peserta pada lokasi psikotes yang dicari administrator dan operator.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Data Peserta Psikotes".	2. Sistem menampilkan <i>form</i> . <i>Form</i> pencarian memiliki masukan untuk "keyword". Pencarian berdasar id peserta, nama, atau lokasi tes
3. Administrator memasukkan data untuk masukan "keyword" dan mengklik tombol "Find".	4. Sistem menampilkan data id peserta, tempat tes, dan waktu tes hasil pencarian.

Aliran Alternatif 1: Masukan keyword kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan *keyword*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan bahwa



	<i>keyword</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Aliran Alternatif 2: Menekan tombol refresh
(Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem mererefresh data peserta pada lokasi psikotes.
	2. Kembali pada langkah nomer 2 aliran utama.

Aturan khusus
1. Masukan <i>keyword</i> tidak boleh kosong.

Sumber : [Analisis]

11. Use Case Specification Mengenerate Passwod Login Peserta Tes

Tabel 4.57 : Use Case Specification Mengenerate Password Login Peserta Psikotes

Nama use case	Mengenerate <i>password login</i> peserta psikotes
Aktor	Administrator ,operator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengenerate <i>password login</i> untuk peserta tes.
Pra-kondisi	Aktor telah menjalankan <i>use case</i> menampilkan data peserta pada lokasi psikotes.
Pasca-kondisi	<i>Password login</i> peserta tes telah berhasil digenerate.

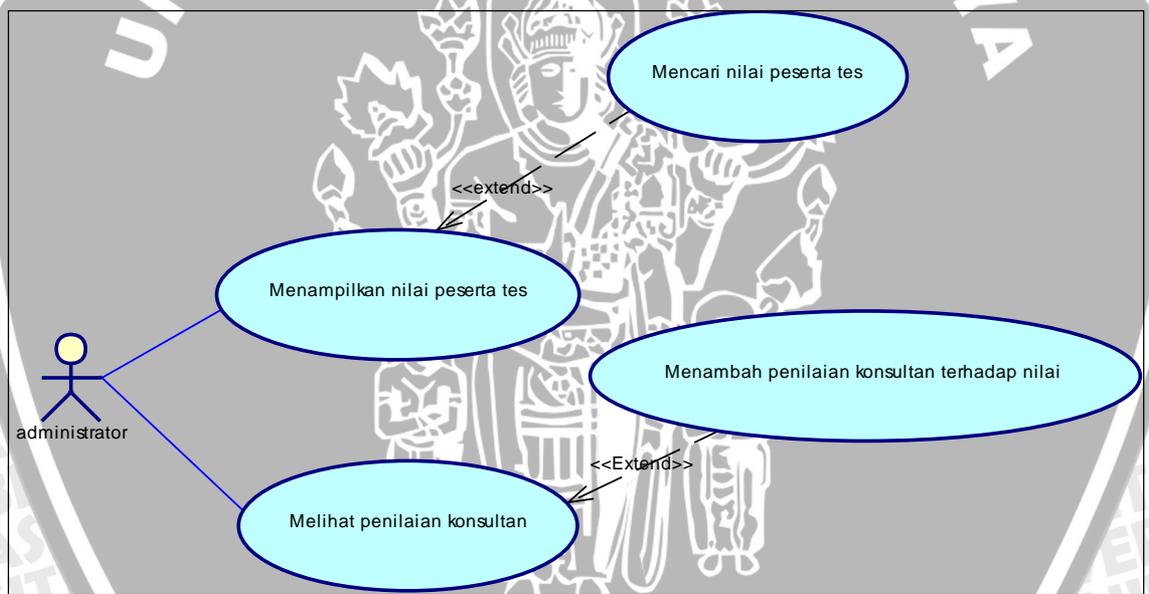
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "edit".	2. Sistem menampilkan <i>form</i> id peserta, tempat tes, waktu tes, <i>username, password</i> .

3. Aktor menekan tombol “Simpan”	4. Sistem menampilkan id peserta, tempat tes, waktu tes dan <i>password</i> yang telah digenerate.
----------------------------------	--

Sumber : [Analisis]

4.1.2.2.6 Use Case Diagram untuk Modul Perhitungan Nilai dan Hasil Penilaian

Gambar 4.10 adalah *use case diagram* untuk modul perhitungan dan hasil penilaian *software* psikotes *online*. Diagram *use case* ini melibatkan sebuah aktor dan 4 buah *use case*. Empat *use case* yang termasuk dalam modul perhitungan dan hasil penilaian *software* psikotes *online* dapat dilihat pada Tabel 4.3. Aktor yang terlibat dalam modul perhitungan dan hasil penilaian *software* psikotes *online* adalah administrator.



Gambar 4.10 : *Use Case Diagram* untuk Modul Perhitungan Nilai dan Hasil Penilaian

Sumber : [Analisis]

1. Use Case Specification Menampilkan Nilai Peserta untuk Setiap Sub Kategori Soal

Tabel 4.58 : *Use Case Specification* Menampilkan Nilai Peserta untuk Setiap Sub Kategori Soal

Nama <i>use case</i>	Menghitung nilai peserta untuk setiap sub kategori soal
----------------------	---



Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menampilkan nilai psikotes peserta untuk setiap sub kategori. Data yang dihitung adalah nilai total dari setiap sub kategori.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> melihat hasil tes.
Pasca-kondisi	Sistem menampilkan hasil tes peserta untuk masing-masing sub kategori soal

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu “Hasil Tes”.	2. Sistem menampilkan daftar tempat, waktu, id peserta, nama, kategori, sub kategori, nilai total, nilai konsultan untuk setiap sub kategori soal.

Sumber : [Analisis]

2. Use Case Specification Mencari Nilai Peserta

Tabel 4.59 : *Use Case Specification* Mencari Nilai Peserta

Nama use case	Mencari nilai peserta
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mencari nilai peserta psikotes.
Pra- kondisi	Administrator telah menjalankan <i>use case</i> menampilkan nilai peserta.
Pasca- kondisi	Sistem menampilkan nilai peserta pada lokasi psikotes yang dicari administrator.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
-----------------	-----------------------

1. Administrator memilih menu "Hasil Tes".	2. Sistem menampilkan <i>form</i> . <i>Form</i> pencarian memiliki masukan untuk "keyword". Pencarian berdasar id peserta, nama, atau lokasi tes
3. Administrator memasukkan data untuk masukan " <i>keyword</i> " dan mengklik tombol " <i>Find</i> ".	4. Sistem menampilkan data id peserta, tempat tes, dan waktu tes hasil pencarian.

Aliran Alternatif 1: Masukan *keyword* kosong
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan *keyword*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan bahwa <i>keyword</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Aliran Alternatif 2: Menekan tombol refresh
(Administrator pada langkah nomer 2 aliran utama menekan tombol refresh.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem mererefresh nilai peserta.
	2. Kembali pada langkah nomer 2 aliran utama.

Aturan khusus	
1. Masukan <i>keyword</i> tidak boleh kosong.	

Sumber : [Analisis]

3. Use Case Specification Menambah Penilaian Konsultan Terhadap Nilai yang Diperoleh Peserta untuk Setiap Sub Kategori

Tabel 4.60 : Use Case Specification Menambah Penilaian Konsultan Terhadap Nilai yang Diperoleh Peserta untuk Setiap Sub Kategori

Nama use case	Menambah penilaian konsultan terhadap nilai yang
----------------------	--

	diperoleh peserta untuk setiap kategori
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah hasil penilain konsultan berdasar nilai yang diperoleh peserta untuk setiap sub kategori. Data yang ditambahkan adalah nilai konsultan.
Pra-kondisi	Sistem telah menjalankan <i>use case</i> menampilkan nilai peserta untuk setiap kategori.
Pasca-kondisi	Hasil penilaian konsultan dapat disimpan ke dalam sistem.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih link “ <i>edit</i> ”	2. Sistem menampilkan id peserta, nama, kategori, sub kategori, nilai sub kategori soal yang dipilih dan masukan <i>text area</i> nilai konsultan.
3. Administrator menekan tombol “Simpan”	4. Sistem menjalankan <i>use case</i> melihat daftar nilai peserta untuk setiap sub kategori.

Sumber : [Analisis]

4. Use case specification Melihat Penilaian Konsultan terhadap Nilai yang Diperoleh Peserta Tes untuk Setiap Sub Kategori

Tabel 4.61 : *Use Case Specification* Melihat Penilaian Konsultan Terhadap Nilai yang Diperoleh Peserta Tes untuk Setiap Kategori

Nama use case	Menampilkan penilaian konsultan terhadap nilai yang diperoleh peserta tes untuk setiap Sub kategori
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menampilkan hasil penilaian konsultan berdasar nilai yang diperoleh peserta untuk setiap kategori. Data yang ditampilkan adalah

	id peserta, nama peserta, tempat tes, waktu tes, kategori soal, sub kategori, nilai sub kategori, dan penilaian konsultan.
Pra-kondisi	User telah <i>login</i> sebagai administrator.
Pasca-kondisi	Sistem menampilkan data id peserta, nama peserta, tempat tes, waktu tes, kategori soal, sub kategori, nilai sub kategori, dan penilaian konsultan.

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik menu "Hasil Tes"	2. Sistem menampilkan data id peserta, nama peserta, tempat tes, waktu tes, kategori soal, sub kategori, nilai sub kategori, dan penilaian konsultan.

Sumber : [Analisis]

4.1.2.3 Class Diagram

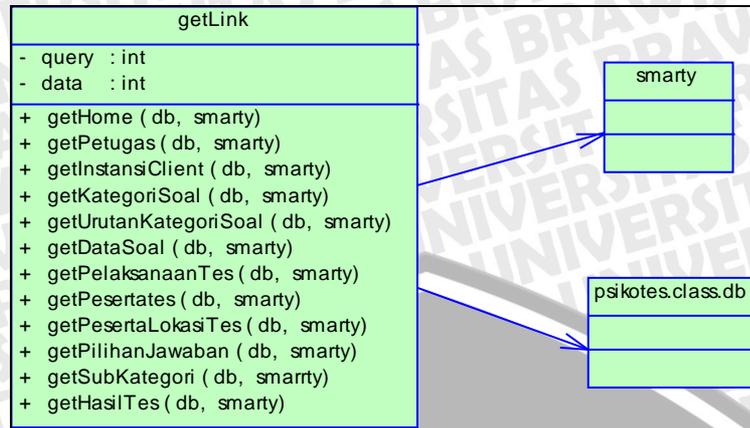
Diagram klas memberikan gambaran pemodelan elemen-elemen klas yang membentuk sebuah perangkat lunak. Klas bisa didapatkan dengan menganalisis secara lebih detail terhadap *use case* yang telah dimodelkan dan merealisasikannya (*Use Case Realization*).

4.1.2.3.1 Class Diagram untuk Class-Class dari Paket

`Psikotes.LogicCode.administrator`

Klas `Psikotes.LogicCode.administrator` memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain. Gambar 4.11 adalah *class diagram* `Psikotes.LogicCode.administrator`.



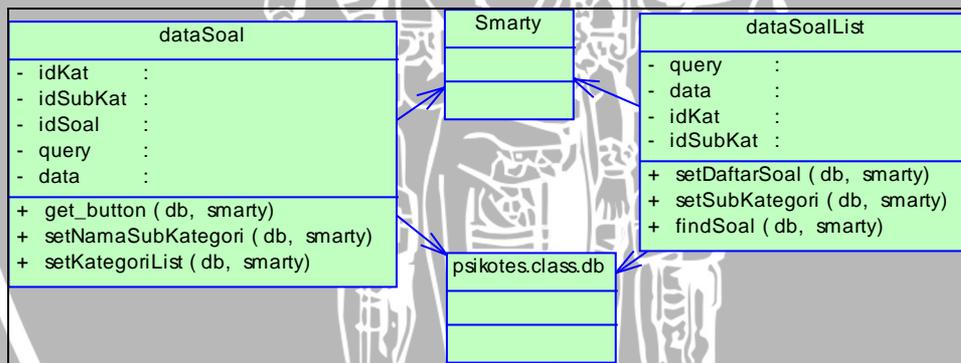


Gambar 4.11 : *Class Diagram* Psikotes.LogicCode.administrator
Sumber : [Perancangan]

4.1.2.3.2 Class Diagram untuk Class-Class dari Paket

Psikotes.LogicCode.dataSoal

Klas Psikotes.LogicCode.dataSoal memodelkan soal-soal pada psikotes. Gambar 4.12 adalah *class diagram* Psikotes.LogicCode.dataSoal.

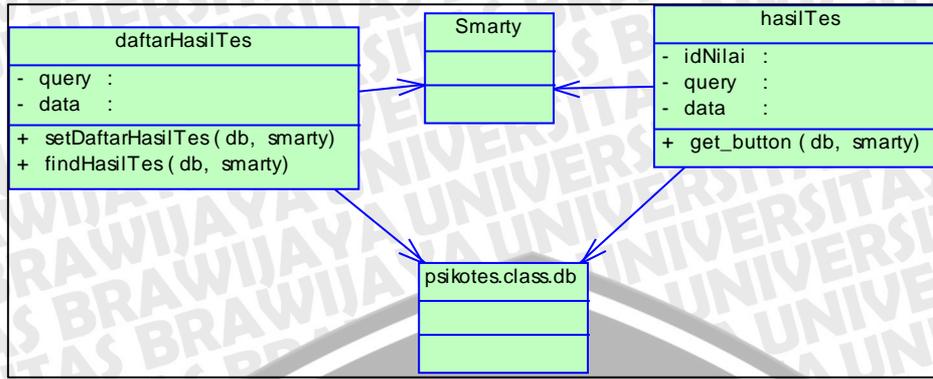


Gambar 4.12 : *Class Diagram* Psikotes.LogicCode.dataSoal
Sumber : [Perancangan]

4.1.2.3.3 Class Diagram untuk Class-Class dari Paket

Psikotes.LogicCode.hasilTes

Klas Psikotes.LogicCode.hasilTes memodelkan hasil tes pada psikotes. Gambar 4.13 adalah *class diagram* Psikotes.LogicCode.hasilTes.

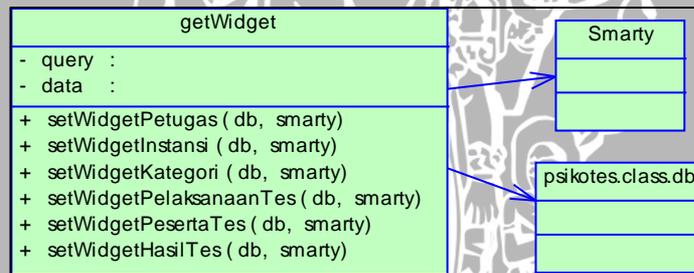


Gambar 4.13 : *Class Diagram* Psikotes.LogicCode.hasilTes
Sumber : [Perancangan]

4.1.2.3.4 *Class Diagram* untuk *Class-Class* dari Paket

Psikotes.LogicCode.home

Klas Psikotes.LogicCode.home memodelkan halaman awal pada *control panel administrator*. Gambar 4.14 adalah *class diagram* Psikotes.LogicCode.home.



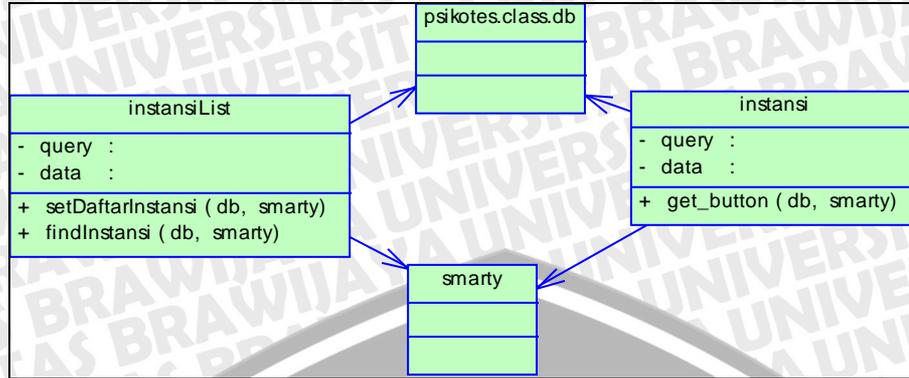
Gambar 4.14 : *Class Diagram* Psikotes.LogicCode.home
Sumber : [Perancangan]

4.1.2.3.5 *Class Diagram* untuk *Class-Class* dari Paket

Psikotes.LogicCode.instansiClient

Klas Psikotes.LogicCode.instansiClient memodelkan *instansi client* yang menyewa konsultan psikologi. Gambar 4.15 adalah *class diagram* Psikotes.LogicCode.instansiClient.



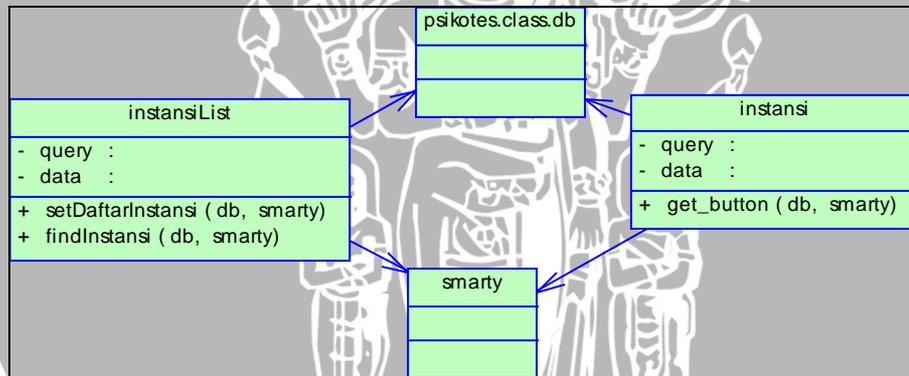


Gambar 4.15 : *Class Diagram* Psikotes.LogicCode.instansiClient
Sumber : [Perancangan]

4.1.2.3.6 *Class Diagram* untuk *Class-Class* dari Paket

Psikotes.LogicCode.pelaksanaanTes

Klas Psikotes.LogicCode.pelaksanaanTes memodelkan waktu dan tempat pelaksanaan psikotes. Gambar 4.16 adalah *class diagram* Psikotes.LogicCode.pelaksanaanTes.

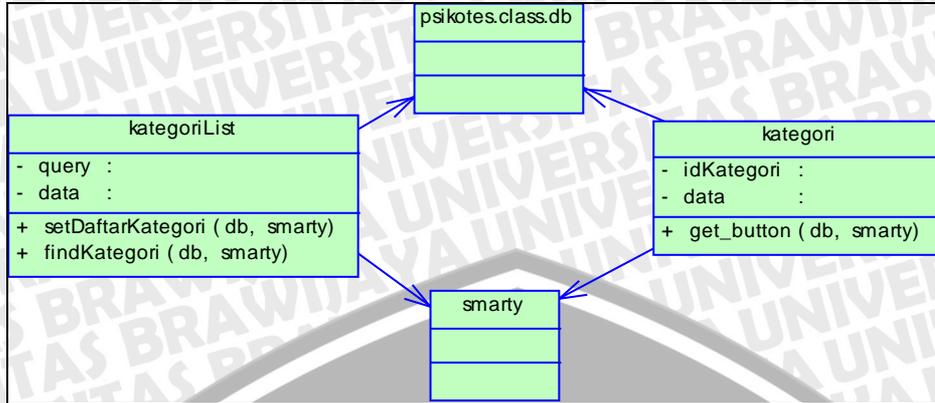


Gambar 4.16 : *Class Diagram* Psikotes.LogicCode.pelaksanaanTes
Sumber : [Perancangan]

4.1.2.3.7 *Class Diagram* untuk *Class-Class* dari Paket

Psikotes.LogicCode.pengaturanSoal

Klas Psikotes.LogicCode.pengaturanSoal memodelkan kategori soal pada psikotes. Gambar 4.17 adalah *class diagram* Psikotes.LogicCode.pengaturanSoal.

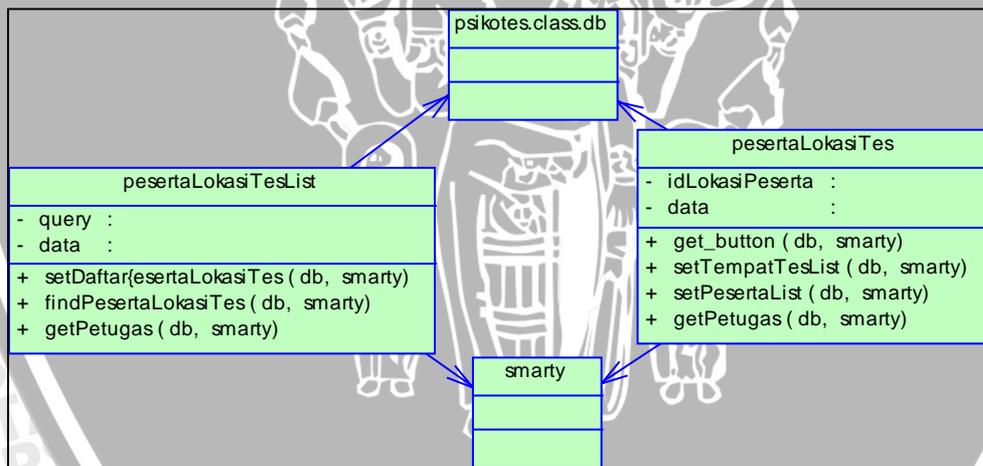


Gambar 4.17 : *Class Diagram* Psikotes.LogicCode.pengaturanSoal
Sumber : [Perancangan]

4.1.2.3.8 Class Diagram untuk Class-Class dari Paket

Psikotes.LogicCode.pesertaLokasiTes

Klas Psikotes.LogicCode.pesertaLokasiTes memodelkan peserta pada suatu lokasi psikotes. Gambar 4.18 adalah *class diagram* Psikotes.LogicCode.pesertaLokasiTes.



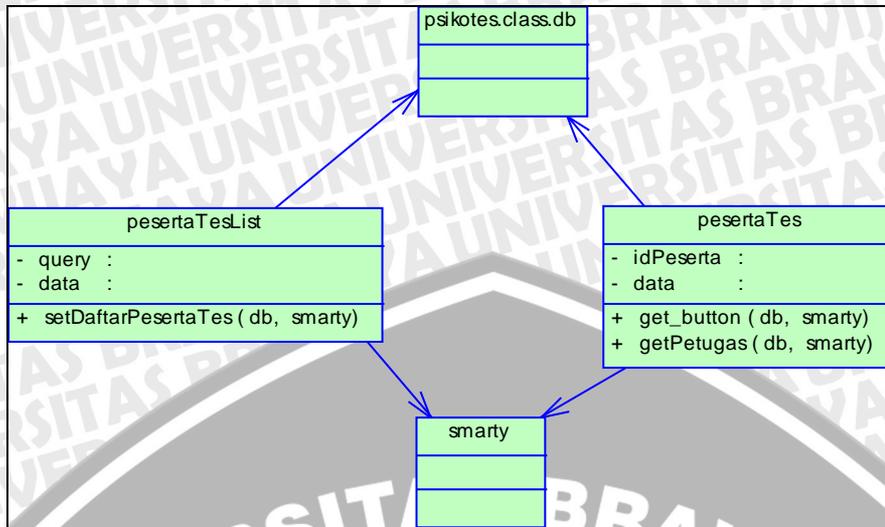
Gambar 4.18 : *Class Diagram* Psikotes.LogicCode.pesertaLokasiTes
Sumber : [Perancangan]

4.1.2.3.9 Class Diagram untuk Class-Class dari Paket

Psikotes.LogicCode.pesertaTes

Klas Psikotes.LogicCode.pesertaTes memodelkan peserta psikotes. Gambar 4.19 adalah *class diagram* Psikotes.LogicCode.pesertaTes.



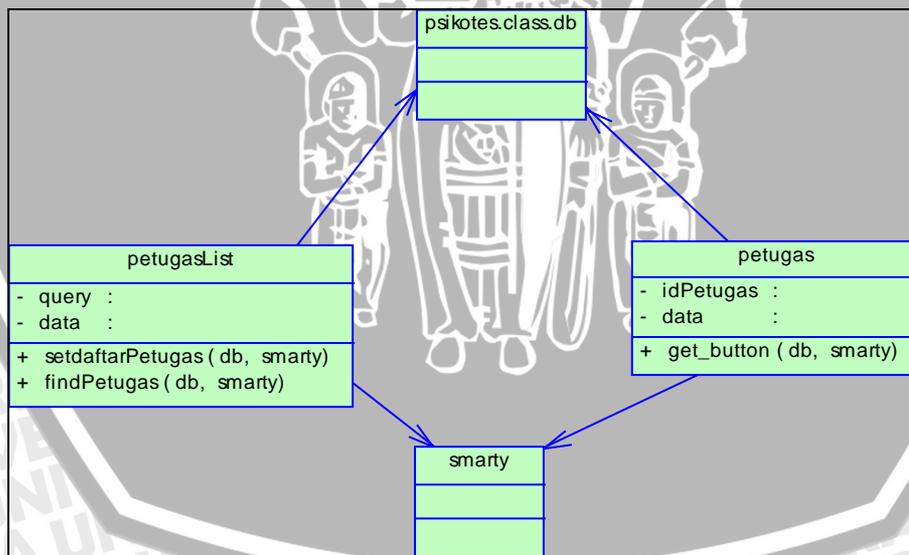


Gambar 4.19 : *Class Diagram* Psikotes.LogicCode.pesertaTes
 Sumber : [Perancangan]

4.1.2.3.10 Class Diagram untuk Class-Class dari Paket

Psikotes.LogicCode.petugas

Klas Psikotes.LogicCode.petugas memodelkan petugas di suatu lokasi psikotes. Gambar 4.20 adalah *class diagram* Psikotes.LogicCode.petugas.



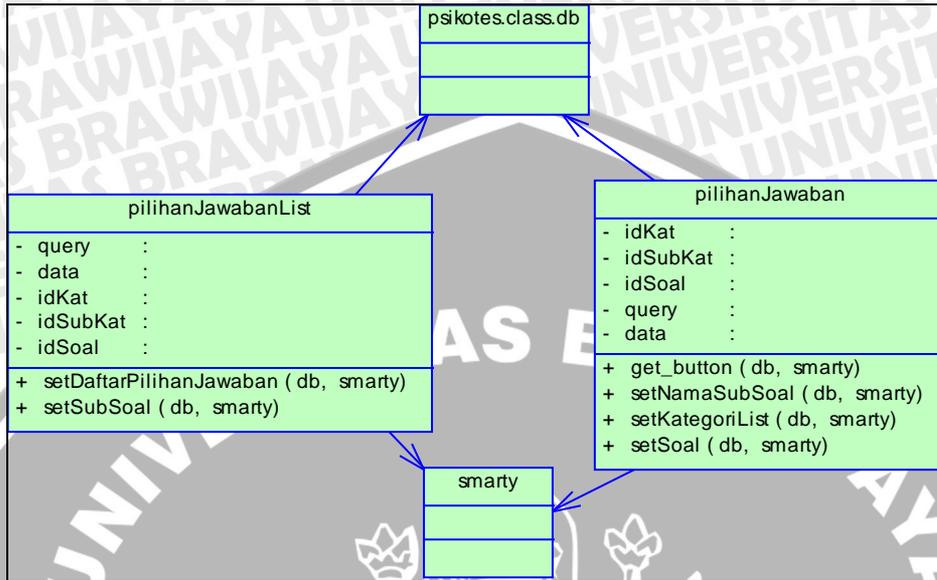
Gambar 4.20 : *Class Diagram* Psikotes.LogicCode.petugas
 Sumber : [Perancangan]

4.1.2.3.11 Class Diagram untuk Class-Class dari Paket

Psikotes.LogicCode.pilihanJawaban

Klas Psikotes.LogicCode.pilihanJawaban memodelkan pilihan

jawaban soal-soal psikotes. Gambar 4.21 adalah *class diagram* Psikotes.LogicCode.pilihanJawaban.

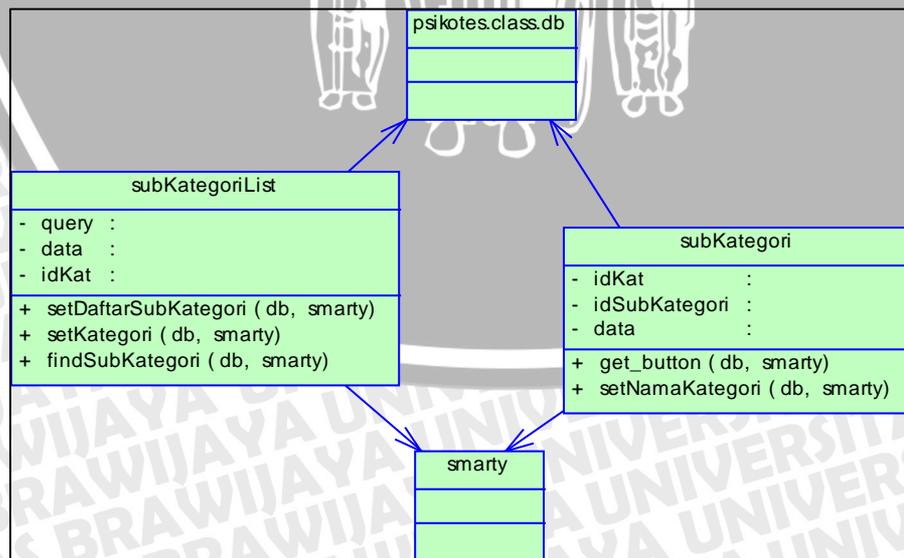


Gambar 4.21 : *Class Diagram* Psikotes.LogicCode.pilihanJawaban
Sumber : [Perancangan]

4.1.2.3.12 *Class Diagram* untuk *Class-Class* dari Paket

Psikotes.LogicCode.subKategori

Klas Psikotes.LogicCode.subKategori memodelkan sub kategori soal-soal psikotes. Gambar 4.22 adalah *class diagram* Psikotes.LogicCode.subKategori.

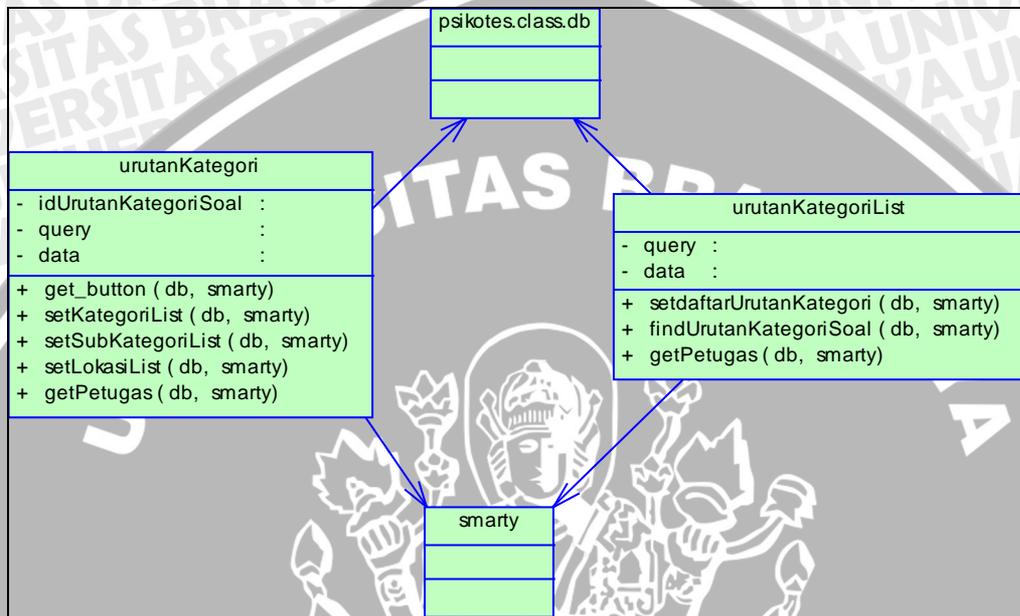


Gambar 4.22 : *Class Diagram* Psikotes.LogicCode.subKategori
Sumber : [Perancangan]

4.1.2.3.13 Class Diagram untuk Class-Class dari Paket

`Psikotes.LogicCode.urutanKategoriSoal`

Klas `Psikotes.LogicCode.urutanKategoriSoal` memodelkan urutan kategori soal-soal psikotes. Gambar 4.23 adalah *class diagram* `Psikotes.LogicCode.urutanKategoriSoal`.

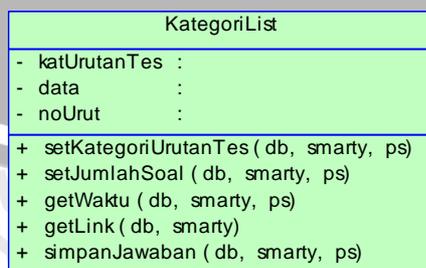


Gambar 4.23 : *Class Diagram* `Psikotes.LogicCode.urutanKategoriSoal`
Sumber : [Perancangan]

4.1.2.3.14 Class Diagram untuk Class-Class dari Paket

`Psikotes.LogicCode.client`

Klas `Psikotes.LogicCode.client` memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain pada sisi *client*. Gambar 4.24 adalah *class diagram* `Psikotes.LogicCode.client`.

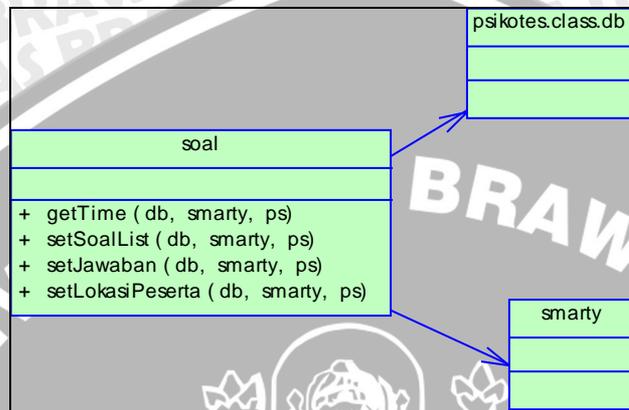


Gambar 4.24 : *Class Diagram* `Psikotes.LogicCode.client`
Sumber : [Perancangan]

4.1.2.3.15 Class Diagram untuk Class-Class dari Paket

`Psikotes.LogicCode.client.soal`

Klas `Psikotes.LogicCode.client.soal` memodelkan soal-soal psikotes di sisi *client*. Gambar 4.25 adalah *class diagram* `Psikotes.LogicCode.client.soal`.



Gambar 4.25 : *Class Diagram* `Psikotes.LogicCode.client.soal`

Sumber : [Perancangan]

4.1.2.4 Sequence Diagram

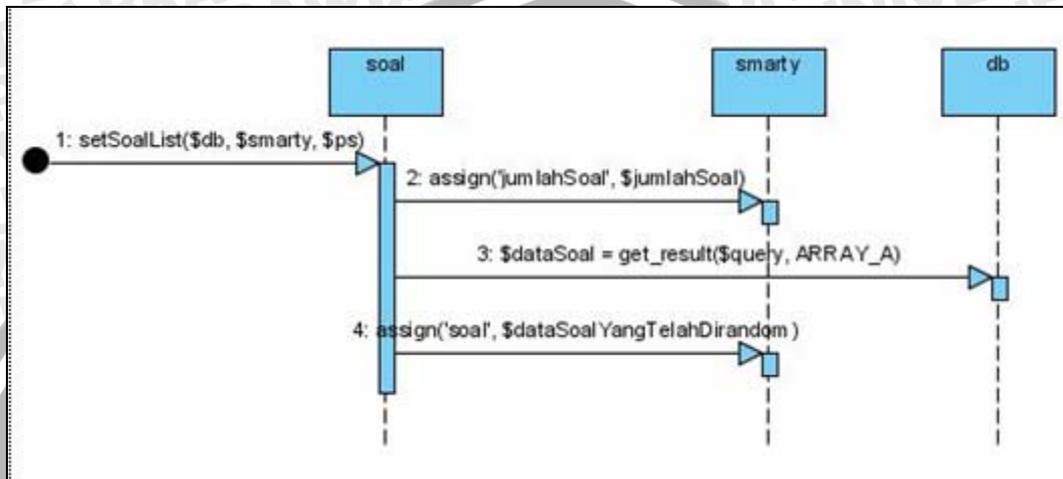
Hubungan antar klas yang digambarkan pada diagram klas termasuk dalam pemodelan statis, tanpa disertai dengan gambaran aliran proses atau data antarklas yang satu dengan klas yang lain. *Sequence diagram* digunakan untuk memodelkan aliran jalannya proses yang ditunjukkan dengan interaksi antar objek atau klas, dan disusun berdasarkan urutan waktu. *Sequence diagram* disusun dengan mengambil acuan pada *use case* dan klas-klas yang telah diturunkan untuk membentuk fungsionalitas yang digambarkan pada *use case* tersebut.

Subbbab 4.1.2.4 ini tidak memodelkan keseluruhan *sequence diagram* untuk tiap *method*, akan tetapi diambil beberapa contoh *sequence diagram* untuk *method* tertentu. Subbab 4.1.2.4 ini memodelkan *sequence diagram* untuk *method* - *method* berikut :

1. *Method* `setSoalList($db,$smarty,$ps)`
2. *Method* `setJawaban($db,$smarty,$ps)`
3. *Method* `simpanJawaban($db,$smarty,$ps)`

4.1.2.4.1 *Sequence Diagram* untuk *method* `setSoalList($db,$smarty,$ps)`

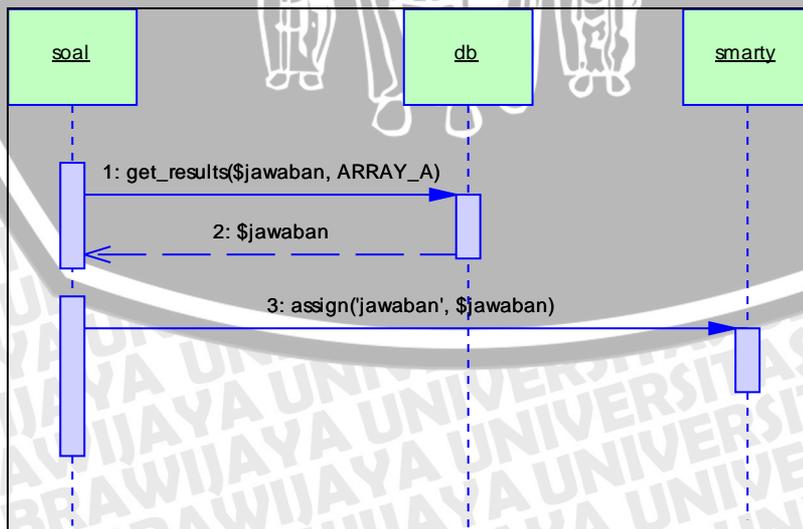
Gambar 4.26 adalah *sequence diagram* untuk *method* `setSoalList`. *Method* ini dipergunakan untuk menampilkan daftar soal yang harus dikerjakan oleh peserta. *Sequence diagram* dari *method* `setSoalList` adalah sebagai berikut :



Gambar 4.26 : *Sequence diagram* untuk *method* `setSoalList`
 Sumber : [Perancangan]

4.1.2.4.2 *Sequence Diagram* untuk *Method* `setJawaban($db,$smarty,$ps)`

Gambar 4.27 adalah *sequence diagram* untuk *method* `setJawaban`. *Method* ini dipergunakan untuk menampilkan daftar jawaban dari masing-masing soal yang harus dikerjakan oleh peserta. *Sequence diagram* dari *method* `setJawaban` adalah sebagai berikut :



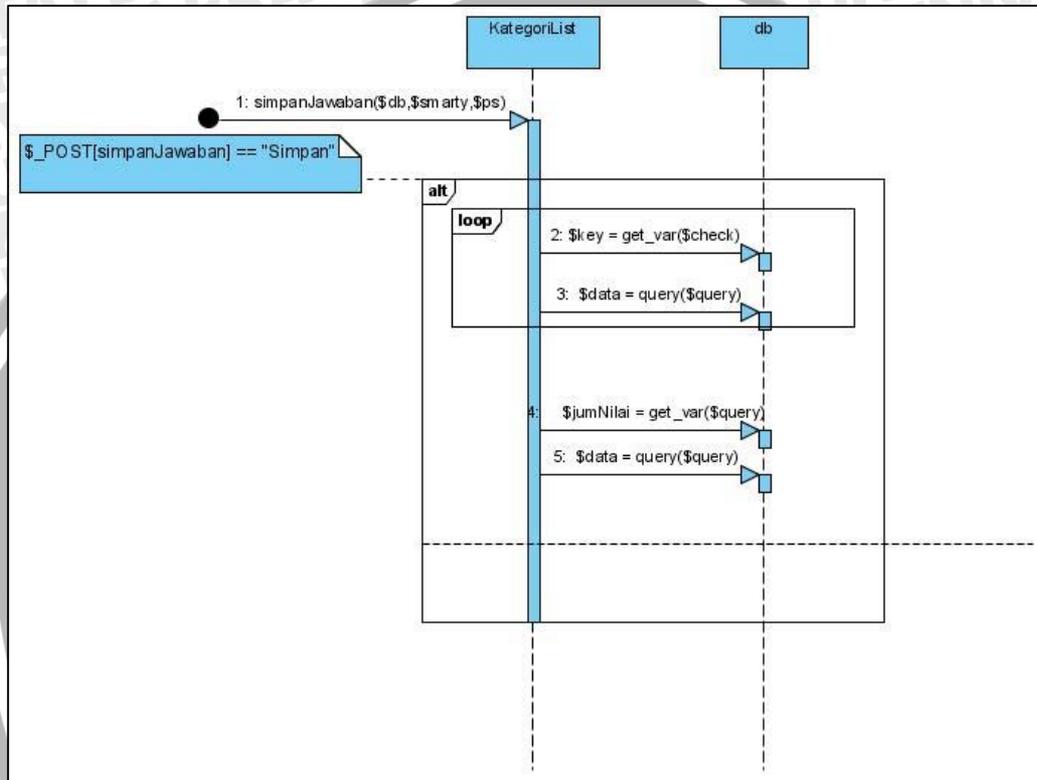
Gambar 4.27 : *Sequence diagram* untuk *Method* `setJawaban`
 Sumber : [Perancangan]

4.1.2.4.3 Sequence Diagram untuk method

`simpanJawaban($db,$smarty,$ps)`

Gambar 4.28 adalah *sequence diagram* untuk *method* `simpanJawaban`.

Method ini dipergunakan untuk menyimpan jawaban dari masing-masing soal yang telah dikerjakan oleh peserta dan menghitung nilainya. *Sequence diagram* dari *method* `simpanJawaban` adalah sebagai berikut :

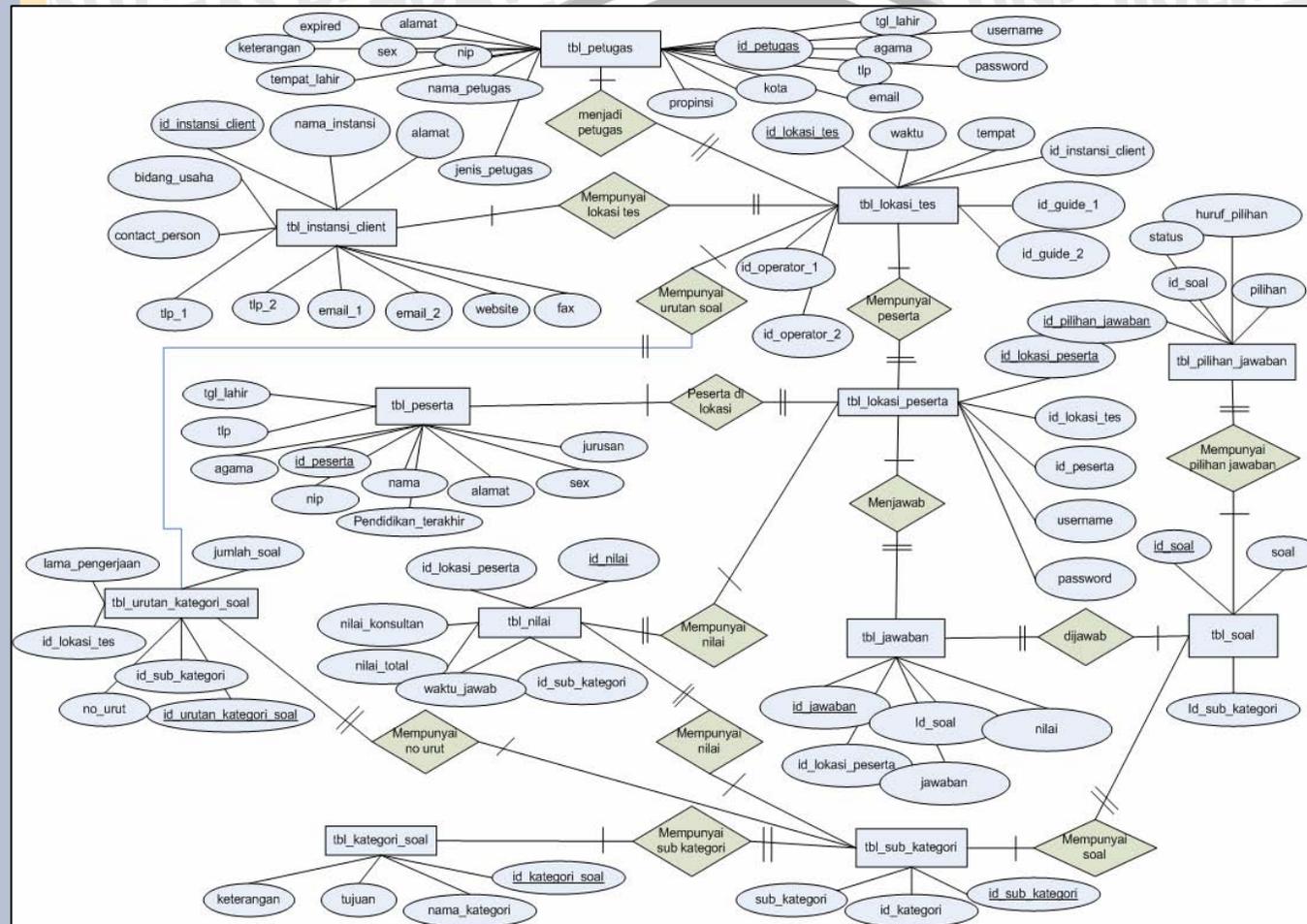


Gambar 4.28 : *Sequence diagram* untuk *method* `simpanJawaban`

Sumber : [Perancangan]

4.2 Perancangan Basis Data

Sistem psikotes *online* memerlukan sebuah basis data. Basis data berfungsi sebagai tempat menyimpan data. Pemodelan basis data dapat dilakukan dengan menggunakan *Entity-Relationship diagram*. Gambar 4.35 adalah gambar ER diagram dari database sistem psikotes *online*.



Gambar 4.29 : ER Diagram Database Sistem Psikotes Online
Sumber : [Perancangan]

4.2.1. Data Object Description untuk Tabel tbl_petugas

Tabel 4.62 : Data Object Description untuk Tabel tbl_petugas

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_petugas	int	
	nip	varchar	5
	nama_petugas	varchar	30
	alamat	varchar	30
	kota	varchar	50
	propinsi	varchar	50
	tgl_lahir	date	
	agama	varchar	15
	tlp	varchar	15
	email	varchar	15
	sex	enum('L','P')	
	tempat_lahir	varchar	20
	jenis_petugas	enum('admin','guide','operator')	
	keterangan	varchar	30
	expired	enum('Y','T')	
	username	varchar	50
	password	varchar	10

Sumber : [Perancangan]

Tabel tbl_petugas mempunyai tujuh belas *field*. *Field* id_petugas merupakan *primary key* dari tbl_petugas. *Field* id_petugas tidak boleh kosong. Tabel tbl_petugas berisi data-data tentang petugas dalam penyelenggaraan psikotes.

4.2.2 Data Object Description untuk Tabel tbl_instansi_client

Tabel 4.63 : Data Object Description untuk Tabel tbl_instansi_client

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_instansi_client	int	
	nama_instansi	varchar	30
	alamat	varchar	30
	bidang_usaha	varchar	20
	contact_person	varchar	50
	tlp_1	varchar	15
	tlp_2	varchar	15
	email_1	varchar	30
	email_2	varchar	30
	fax	varchar	15
	website	varchar	20

Sumber : [Perancangan]

Tabel `tbl_instansi_client` mempunyai dua belas *field*. *Field* `id_instansi_client` merupakan *primary key* dari tabel `tbl_jenis_anggota`. Tabel `tbl_jenis_anggota` berisi data-data tentang instansi *client* yang menyewa konsultan tersebut untuk menyelenggarakan psikotes.

4.2.3 Data Object Description untuk Tabel `tbl_kategori_soal`

Tabel 4.64 : Data Object Description untuk Tabel `tbl_kategori_soal`

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	<code>id_kategori_soal</code>	int	
	<code>nama_kategori</code>	varchar	30
	<code>tujuan</code>	varchar	30
	<code>keterangan</code>	varchar	50

Sumber : [Perancangan]

Tabel `tbl_kategori_soal` mempunyai empat *field*. *Field* `id_kategori_soal` merupakan *primary key* dari tabel `tbl_kategori_soal`. Tabel `tbl_kategori_soal` berisi kategori soal yang digunakan untuk psikotes.

4.2.4 Data Object Description untuk Tabel `tbl_urutan_kategori_soal`

Tabel 4.65 : Data Object Description untuk Tabel `tbl_urutan_kategori_soal`

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	<code>id_urutan_kategori_soal</code>	int	
FK	<code>id_sub_kategori</code>	int	
	<code>no_urut</code>	Int	
FK	<code>id_lokasi_tes</code>	int	
	<code>lama_pengerjaan</code>	int	
	<code>jumlah_soal</code>	int	

Sumber : [Perancangan]

Tabel `tbl_urutan_kategori_soal` mempunyai enam *field*. *Field* `id_urutan_kategori_soal` merupakan *primary key* dari tabel `tbl_urutan_kategori_soal`. *Field* `id_sub_kategori` merupakan *foreign key* dari tabel `tbl_sub_kategori`. *Field* `id_lokasi_tes` merupakan *foreign key* dari tabel `tbl_lokasi_tes`. *Field* `id_urutan_kategori_soal`, `id_sub_kategori`, dan `id_lokasi_tes` tidak boleh kosong. Tabel `tbl_urutan_kategori_soal` berisi data-data tentang urutan soal per kategori pada suatu lokasi psikotes.

4.2.5 Data Object Description untuk Tabel tbl_lokasi_tes

Tabel 4.66 : Data Object Description untuk Tabel tbl_lokasi_tes

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_lokasi_tes	int	
	waktu	datetime	
	tempat	varchar	50
	alamat	varchar	100
FK	id_instansi_client	int	
FK	id_guide_1	int	
FK	id_guide_2	int	
FK	id_operator_1	int	
FK	id_operator_2	int	

Sumber : [Perancangan]

Tabel tbl_lokasi_tes mempunyai sembilan *field*. *Field* id_lokasi_tes merupakan *primary key* dari tabel tbl_lokasi_tes. *Field* id_instansi_client merupakan *foreign key* dari tabel tbl_instansi_client. *Field* id_guide_1, id_guide_2, id_operator_1, id_operator_2 merupakan *foreign key* dari tabel tbl_petugas. *Field* id_lokasi_tes, id_instansi_client, id_guide_1 dan id_operator_1 tidak boleh kosong. Tabel tbl_lokasi_tes berisi data-data tentang lokasi dan waktu penyelenggaraan psikotes.

4.2.6 Data Object Description untuk Tabel tbl_soal

Tabel 4.67 : Data Object Description untuk Tabel tbl_soal

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_soal	int	
	soal	text	
FK	id_sub_kategori	int	

Sumber : [Perancangan]

Tabel tbl_soal mempunyai tiga *field*. *Field* id_soal merupakan *primary key* dari tabel tbl_soal. *Field* id_sub_kategori merupakan *foreign key* dari tabel tbl_sub_kategori. *Field* id_soal dan id_sub_kategori tidak boleh kosong. Tabel tbl_soal berisi soal yang diberikan pada psikotes.

4.2.7 Data Object Description untuk Tabel tbl_peserta

Tabel 4.68 : Data Object Description untuk Tabel tbl_peserta

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_peserta	int	
	nip	varchar	10
	nama	varchar	50

	alamat	alamat	100
	pendidikan_terakhir	varchar	5
	jurusan	varchar	30
	tlp	varchar	15
	agama	varchar	15
	tgl_lahir	date	
	sex	enum('L','P')	

Sumber : [Perancangan]

Tabel tbl_soal mempunyai sepuluh *field*. *Field* id_peserta merupakan *primary key* dari tabel tbl_peserta. *Field* id_peserta tidak boleh kosong. Tabel tbl_peserta berisi tentang data-data peserta psikotes.

4.2.8 Data Object Description untuk Tabel tbl_lokasi_peserta

Tabel 4.69 : Data Object Description untuk Tabel tbl_lokasi_peserta

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_lokasi_peserta	int	
FK	id_lokasi_tes	int	
FK	id_peserta	int	
	username	varchar	10
	password	varchar	10

Sumber : [Perancangan]

Tabel tbl_lokasi_peserta mempunyai lima *field*. *Field* id_lokasi_peserta merupakan *primary key* dari tabel tbl_lokasi_peserta. *Field* id_lokasi_tes merupakan *foreign key* dari tabel tbl_lokasi_tes. *Field* id_peserta merupakan *foreign key* dari tabel tbl_peserta. *Field* id_lokasi_peserta, id_lokasi_tes dan id_peserta tidak boleh kosong. Tabel tbl_lokasi_peserta berisi data-data pembagian peserta pada lokasi tes tertentu.

4.2.9 Data Object Description untuk Tabel tbl_jawaban

Tabel 4.70 : Data Object Description untuk Tabel tbl_jawaban

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_jawaban	int	
FK	id_lokasi_peserta	int	
FK	id_soal	int	
	jawaban	char	1
	nilai	int	

Sumber : [Perancangan]

Tabel tbl_jawaban mempunyai lima *field*. *Field* id_jawaban merupakan *primary key* dari tabel tbl_jawaban. *Field* id_lokasi_peserta merupakan *foreign*

key dari tabel tbl_lokasi_peserta. *Field* id_soal merupakan *foreign key* dari tabel tbl_soal. *Field* id_jawaban, id_lokasi_peserta dan id_soal tidak boleh kosong. Tabel tbl_jawaban berisi jawaban soal dari peserta tes.

4.2.10 Data Object Description untuk Tabel tbl_nilai

Tabel 4.71 : Data Object Description untuk Tabel tbl_nilai

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_nilai	int	
FK	id_lokasi_peserta	int	
FK	id_sub_kategori	int	
	nilai_total	int	
	nilai_konsultan	text	
	waktu_jawab	int	

Sumber : [Perancangan]

Tabel tbl_nilai mempunyai enam *field*. *Field* id_nilai merupakan *primary key* dari tabel tbl_nilai. *Field* id_lokasi_peserta merupakan *foreign key* dari tabel tbl_lokasi_peserta. *Field* id_sub_kategori merupakan *foreign key* dari tabel tbl_sub_kategori. *Field* id_nilai, id_lokasi_peserta dan id_sub_kategori tidak boleh kosong. Tabel tbl_nilai berisi nilai total dari peserta tes.

4.2.11 Data Object Description untuk Tabel tbl_sub_kategori

Tabel 4.72 : Data Object Description untuk Tabel tbl_sub_kategori

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_sub_kategori	int	
FK	id_kategori	int	
	sub_kategori	varchar	100

Sumber : [Perancangan]

Tabel tbl_sub_kategori mempunyai tiga *field*. *Field* id_sub_kategori merupakan *primary key* dari tabel tbl_sub_kategori. *Field* id_kategori merupakan *foreign key* dari tabel tbl_kategori_soal. *Field* id_sub_kategori merupakan *foreign key* dari tabel tbl_sub_kategori. *Field* id_sub_kategori dan id_kategori tidak boleh kosong. Tabel tbl_sub_kategori berisi sub kategori soal dari kategori soal yang ada.

4.2.12 Data Object Description untuk Tabel tbl_pilihan_jawaban

4.73 : Data Object Description untuk Tabel tbl_pilihan_jawaban

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	id_pilihan_jawaban	int	
FK	id_soal	int	
	huruf_pilihan	char	1
	pilihan	text	
	Status	enum('Benar','Salah','Netral')	

Sumber : [Perancangan]

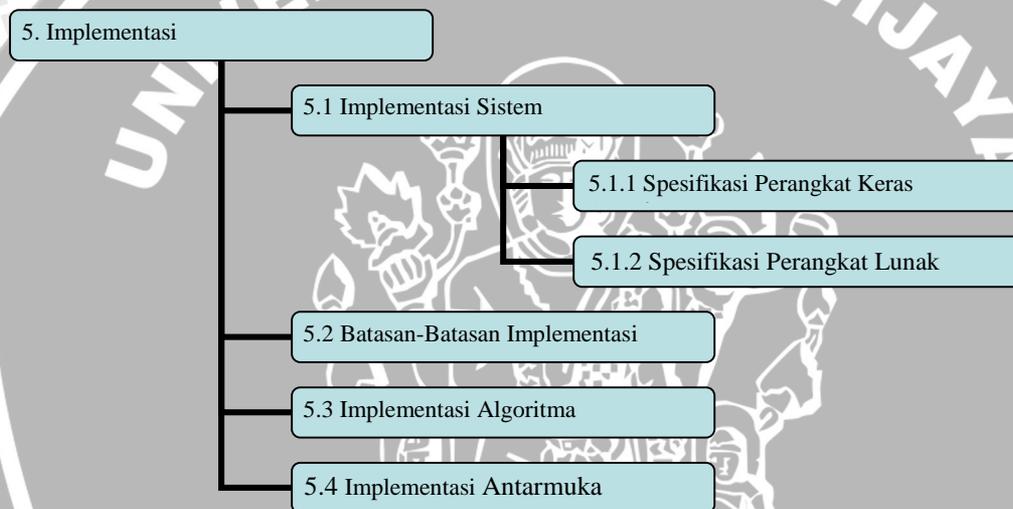
Tabel tbl_pilihan_jawaban mempunyai lima field. Field id_pilihan_jawaban merupakan *primary key* dari tabel tbl_pilihan_jawaban. Field id_soal merupakan *foreign key* dari tabel tbl_soal. Field id_pilihan_jawaban dan id_soal tidak boleh kosong. Tabel tbl_pilihan_jawaban berisi pilihan jawaban yang telah tersedia.



BAB V IMPLEMENTASI

Bab ini membahas tentang implementasi perangkat lunak psikotes *online* berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak. Bab ini terdiri dari pembahasan implementasi algoritma dan implementasi antarmuka, ditambah dengan penjelasan tentang spesifikasi lingkungan dimana sistem diimplementasikan.

Gambar 5.1 menunjukkan langkah-langkah proses implementasi yang digunakan pada bab ini.



Gambar 5.1 : Diagram Pohon Implementasi Sistem
Sumber : Implementasi

5.1 Spesifikasi Sistem

Dari hasil analisis kebutuhan dan perancangan yang telah diuraikan pada Bab 4, dilakukan implementasi menjadi sebuah sistem RBTEOnline yang nyata agar bisa berfungsi sesuai dengan kebutuhan. Sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Sistem psikotes *online* merupakan jenis aplikasi *client-server*. Dalam skripsi ini, baik *client* maupun *server* terdapat dalam satu komputer yang sama.

Pengembangan sistem psikotes *online* menggunakan sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Nama Komponen	Spesifikasi
Prosesor	Genuine Intel T2060 @ 1,6 GHz
Memori (RAM)	768 MB
Hardisk	80 GB
VGA Card	VIA Chrome 9 HC 256 MB

Sumber : [Implementasi]

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan sistem psikotes *online* menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak

Spesifikasi Perangkat Lunak	
Sistem operasi	Windows XP Service Pack 2
Bahasa pemrograman	PHP 5
Web Server	Apache 2.0
Database	MySQL 5.0.37
Framework	Smarty Template Engine 2. 6.14
IDE (<i>Integrated Development Environment</i>)	WebBuilder 2008

Sumber : [Implementasi]

5.2 Implementasi Konfigurasi *Smarty Template Engine*

Penggunaan *smarty* dalam PHP perlu dikonfigurasi terlebih dahulu. Hal ini berguna untuk mengatur letak penyimpanan dari file *template code* terhadap *logic code* yang telah dibuat. Konfigurasi yang digunakan dalam sistem psikotes *online* adalah sebagai berikut :

```
require_once ('Smarty/libs/Smarty.class.php');
$smarty = new Smarty();
$smarty->compile_check = true;
$smarty->debugging = false;
$smarty->template_dir = 'TemplateCode';
$smarty->compile_dir = 'templates_c';
$smarty->cache_dir = 'cache';
$smarty->config_dir = 'configs';
```

Gambar 5.2 : Konfigurasi *Smarty Template Engine*

Sumber : [Implementasi]

Pada konfigurasi diatas, letak *template code* akan disimpan dalam folder TemplateCode dan untuk chace dari gabungan antara *logic code* dan *template code* akan disimpan pada folder templates_c. *File – file* yang tersimpan dalam folder TemplateCode adalah *.tpl, sehingga ketika dibuat suatu file *logic code* yang didalamnya sama sekali tidak ada pengaturan tampilan, file ini harus menunjuk salah satu file *.tpl yang akan menjadi pasangan *template codenya* dengan notasi sebagai berikut :

```
$smarty->display('administrator/dataSoal/tambahDataSoal.tpl');
```

Gambar 5.3 : Notasi *smarty* untuk Menunjuk *Template Code*

Sumber : [Implementasi]

Jika *file logic code* (*.php) telah menunjuk salah satu file *.tpl maka ketika dijalankan, dua file tersebut akan di *compile* dan akan tersimpan dalam file templates_c dengan nama file yang telah dikonversi oleh *smarty*. Jika salah satu dari file dalam templates_c dibuka, isinya adalah merupakan gabungan dari *logic code* dan *template code*.

5.2.1 Implementasi Penggunaan *Smarty Template Engine*

Seperti telah disebutkan sebelumnya, bahwa satu file *.php harus menunjuk satu file *.tpl sebagai tampilannya. Dalam sistem aplikasi psikotes *online*, file php disimpan dalam folder LogicCode dan folder TemplateCode untuk menyimpan file tpl. Berikut tabel dari masing – masing pasangan file php dan tpl yang digunakan dalam sistem psikotes *online*.

Tabel 5.3 : Pengaturan File Logic Code dan Template Code

No	Logic Code	Template Code
1.	LogicCode/administrator/tambahDataSoal.php	TemplateCode/administrator/tambahDataSoal.tpl
2.	LogicCode/administrator/dataSoalList.php	TemplateCode/administrator/dataSoalList.tpl
3.	LogicCode/administrator/tambahHasilTes.php	TemplateCode/administrator/tambahHasilTes.tpl
4.	LogicCode/administrator/hasilTesList.php	TemplateCode/administrator/tambahHasilTes.tpl
5.	LogicCode/administrator/home.php	TemplateCode/administrator/home.tpl
6.	LogicCode/administrator/instansiList.php	TemplateCode/administrator/instansiList.tpl
7.	LogicCode/administrator/tambahInstansi.php	TemplateCode/administrator/tambahInstansi.tpl
8.	LogicCode/administrator/tambahPelaksanaanTes.php	TemplateCode/administrator/tambahPelaksanaanTes.tpl
9.	LogicCode/administrator/pelaksanaanTesList.php	TemplateCode/administrator/pelaksanaanTesList.tpl
10.	LogicCode/administrator/kategoriList.php	TemplateCode/administrator/kategoriList.tpl
11.	LogicCode/administrator/tambahKategori.php	TemplateCode/administrator/tambahKategori.tpl
12.	LogicCode/administrator/tambahPesertaLokasiTes.php	TemplateCode/administrator/tambahPesertaLokasiTes.tpl
13.	LogicCode/administrator/pesertaLokasiTes.php	TemplateCode/administrator/pesertaLokasiTes.tpl
14.	LogicCode/administrator/pesertaTesList.php	TemplateCode/administrator/pesertaTesList.tpl
15.	LogicCode/administrator/tambahPesertaTes.php	TemplateCode/administrator/tambahPesertaTes
16.	LogicCode/administrator/petugasList.php	TemplateCode/administrator/petugasList.tpl
17.	LogicCode/administrator/tambahPetugas.php	TemplateCode/administrator/tambahPetugas.tpl
18.	LogicCode/administrator/pilihanJawaban.php	TemplateCode/administrator/pilihanJawaban.tpl
19.	LogicCode/administrator/tambahPilihanJawaban.php	TemplateCode/administrator/tambahPilihanJawaban.tpl
20.	LogicCode/administrator/subKategoriList.php	TemplateCode/administrator/subKategoriList.tpl

21.	LogicCode/administrator/tambahSubKategori.php	TemplateCode/administrator/subKategoriList.tpl
22.	LogicCode/administrator/urutanKategoriSoalList.php	TemplateCode/administrator/urutanKategoriSoalList.tpl
23.	LogicCode/administrator/tambahUrutanKategoriSoal.php	TemplateCode/administrator/tambahUrutanKategoriSoal.tpl
24.	LogicCode/administrator/index.php	TemplateCode/administrator/index.tpl
25.	LogicCode/client/index.php	TemplateCode/client/index.tpl
26.	LogicCode/client/soal/soal.php	TemplateCode/client/soal/soal.tpl
27.	LogicCode/global/index.php	TemplateCode/global/index.tpl

Sumber : [Implementasi]



5.3 Implementasi Algoritma

Pada *package logic code* untuk aplikasi psikotes *online* ini memiliki 28 klas, yaitu :

Tabel 5.4 : Klas dalam *Package Logic Code*

No	Klas	Method
1.	dataSoal	get_button(\$db,\$smarty)
2.		setNamaSubKategori(\$db,\$smarty)
3.		setKategoriList(\$db,\$smarty)
4.	dataSoalList	setDaftarSoal(\$db,\$smarty)
5.		setSubKategori(\$db,\$smarty)
6.		findSoal(\$db,\$smarty)
7.	hasilTesList	setDaftarHasilTes(\$db,\$smarty)
8.		findHasilTes(\$db,\$smarty)
9.	hasilTes	get_button(\$db,\$smarty)
10.	getWidget	setWidgetPetugas(\$db,\$smarty)
11.		setWidgetInstansi(\$db,\$smarty)
12.		setWidgetKategori(\$db,\$smarty)
13.		setWidgetPelaksanaanTes(\$db,\$smarty)
14.		setWidgetPesertaTes(\$db,\$smarty)
15.		setWidgetHasilTes(\$db,\$smarty)
16.		instansiList
17.	findInstansi(\$db,\$smarty)	
18.	instansi	get_button(\$db,\$smarty)
19.	pelaksanaanTesList	setDaftarPelaksanaanTes(\$db,\$smarty)
20.		findPelaksanaanTes(\$db,\$smarty)
21.		getPetugas(\$db,\$smarty)
22.	pelaksanaanTes	pelaksanaanTes(\$db,\$smarty)
23.		setOperatorList(\$db,\$smarty)
24.		setGuideList(\$db,\$smarty)
25.		setInstansiList(\$db,\$smarty)
26.		getPetugas(\$db,\$smarty)
27.	kategoriList	setDaftarKategori(\$db,\$smarty)
28.		findKategori(\$db,\$smarty)
29.	kategori	get_button(\$db,\$smarty)
30.	pesertaLokasiTesList	setDaftarPesertaLokasiTes(\$db,\$smarty)
31.		findPesertaLokasiTes(\$db,\$smarty)
32.		getPetugas(\$db,\$smarty)

33.		get_button(\$db,\$smarty)
34.	pesertaLokasiTes	setTempatTesList(\$db,\$smarty)
35.		setPesertaList(\$db,\$smarty)
36.		getPetugas(\$db,\$smarty)
37.	pesertaTes	get_button(\$db,\$smarty)
38.		getPetugas(\$db,\$smarty)
39.	pesertaTesList	setDaftarPesertaTes(\$db,\$smarty)
40.		findPesertaTes(\$db,\$smarty)
41.	petugas	get_button(\$db,\$smarty)
42.	petugasList	setDaftarPetugas(\$db,\$smarty)
43.		findPetugas(\$db,\$smarty)
44.	pilihanJawaban	setNamaSubSoal(\$db,\$smarty)
45.		setKategoriList(\$db,\$smarty)
46.		setSoal(\$db,\$smarty)
47.		get_button(\$db,\$smarty)
48.	pilihanJawabanList	setDaftarPilihanJawaban(\$db,\$smarty)
49.		setSubSoal(\$db,\$smarty)
50.	subKategori	setNamaKategori(\$db,\$smarty)
51.		get_button(\$db,\$smarty)
52.	subKategoriList	setDaftarSubKategori(\$db,\$smarty)
53.		setKategori(\$db,\$smarty)
54.		findSubKategori(\$db,\$smarty)
55.	urutanKategoriList	setDaftarUrutanKategori(\$db,\$smarty)
56.		findUrutanKategoriSoal(\$db,\$smarty)
57.		getPetugas(\$db,\$smarty)
58.	urutanKategori	setLokasiList(\$db,\$smarty)
59.		setKategoriList(\$db,\$smarty)
60.		setSubKategoriList(\$db,\$smarty)
61.		get_button(\$db,\$smarty)
62.		getPetugas(\$db,\$smarty)
63.	getLink	getHome(\$db,\$smarty)
64.		getPetugas(\$db,\$smarty)
65.		getInstansiClient(\$db,\$smarty)
66.		getKategoriSoal(\$db,\$smarty)
67.		getUrutanKategoriSoal(\$db,\$smarty)
68.		getDataSoal(\$db,\$smarty)
69.		getPelaksanaanTes(\$db,\$smarty)
70.		getPesertaTes(\$db,\$smarty)

71.		getPesertaLokasiTes(\$db,\$smarty)
72.		getPilihanJawaban(\$db,\$smarty)
73.		getSubKategori(\$db,\$smarty)
74.		getHasilTes(\$db,\$smarty)
75.		getTime(\$db,\$smarty,\$ps)
76.	soal	setSoalList(\$db,\$smarty,\$ps)
77.		setJawaban(\$db,\$smarty,\$ps)
78.		setLokasiPeserta(\$db,\$smarty,\$ps)
79.		setKategoriUrutanTes(\$db,\$smarty,\$ps)
80.		setJumlahSoal(\$db,\$smarty,\$ps)
81.	kategoriList	getWaktu(\$db,\$smarty,\$ps)
82.		getLink(\$db,\$smarty,\$ps)
83.		simpanJawaban(\$db,\$smarty,\$ps)
84.	db	query(\$query)
85.		get_var(\$query)
86.		get_results(\$query=null,\$output=OBJECT)
87.		selection_session()
88.	private_sessions	logout()
89.		create_session(\$buttonName,\$tableName)

Sumber : [Implementasi]

5.3.1 Pseudocode method setSoalList(\$db,\$smarty,\$ps) dari Klas soal

Method setSoalList(\$db,\$smarty,\$ps) ini dipergunakan untuk menampilkan daftar soal yang harus dikerjakan oleh peserta. Pseudocode dari algoritma method setSoalList adalah sebagai berikut :

METHOD : setSoalList PARAMETER db,smarty,ps

DECLARATION :

db
 smarty
 ps
 jumlahSoal
 indexSoalYgDipilih
 dataSoalYangTelahDirandom
 controlLooping
 indexYangDirandom
 nilai
 indexSoalYgDipilih
 dataSoal

DESCRIPTION :

```

1  Get jumlahSoal
2  Query dataSoal limit jumlahSoal
3  indexSoalYgDipilih = array()
4  dataSoalYangTelahDirandom = array(array())
5      FOR i = 0 TO i < jumlahSoal
6
```

```

7         controlLooping = true
8     WHILE (controlLooping)
9         indexYangDirandom = random(0,jumlahSoal-1)
10        nilai=in_array(indexYangDirandom,
11                        indexSoalYgDipilih)
12    IF nilai = 1 THEN
13        controlLooping = true
14    ELSE
15        controlLooping = false
16    END IF
17
18    IF controlLooping = false
19        indexSoalYgDipilih[$i] = indexYangDirandom;
20        dataSoalYangTelahDirandom[$i]=
21            dataSoal[$indexYangDirandom]
22    END IF
23    END WHILE
24 END FOR

```

Gambar 5.4 : Pseudocode Algoritma Method *setSoalList* (db, smarty, ps)

Sumber : [Implementasi]

Pertama-tama *method* harus mendapatkan nilai dari *jumlahSoal* yang didapat dari pengembalian *variable page* yang sebelumnya. Jika nilai *jumlahSoal* telah didapat maka dilakukan operasi penyeleksian *query sql* dari *tbl_soal* dengan *limit* sebanyak *jumlahSoal*. Kemudian dari hasil penyeleksian, dilakukan *random* terhadap *index* soal yang dipilih sehingga soal yang diberikan dapat berbeda-beda.

5.3.2 Implementasi Template Code Penggunaan Smarty untuk Pseudocode

method setSoalList

Implementasi *template code* untuk menampilkan *return* dari *method setSoalList* adalah sebagai berikut :

```

{section name=dataSoal loop=$soal}
    {counter}.$soal[dataSoal].soal}
{/section}

```

Gambar 5.5 : Template Code untuk Method *setSoalList* (db, smarty, ps)

Sumber : [Implementasi]

Perintah *section* dalam *smarty* berfungsi untuk melakukan perintah berulang dari sebuah array yang didapat dari *variable soal* dan diberi nama dengan nama *dataSoal*.

5.3.3 Pseudocode method *setJawaban*(\$db,\$smarty,\$ps) dari Klas soal

Method setJawaban(\$db,\$smarty,\$ps) ini dipergunakan untuk menampilkan daftar jawaban dari masing-masing soal yang harus dikerjakan oleh

peserta. *Pseudocode* dari algoritma *method* `setJawaban` adalah sebagai berikut :

METHOD : `setSoallist PARAMETER db,smarty,ps`

DECLARATION :

`db`
`smarty`
`ps`

DESCRIPTION :

1 Query dataJawaban

Gambar 5.6 : *Pseudocode* Algoritma *Method* `setJawaban (db, smarty, ps)`

Sumber : [Implementasi]

Pada *method* ini hanya dilakukan penyeleksian terhadap seluruh jawaban soal. Sedangkan untuk pencocokan dengan *index* soalnya akan dilakukan di sisi *template* dengan *smarty*.

5.3.4 Implementasi *Template Code* Penggunaan *Smarty* untuk *Pseudocode*

method `setJawaban`

Implementasi *template code* untuk menampilkan *return* dari *method* `setJawaban` adalah sebagai berikut :

```
{section name=dataJawaban loop=$jawaban}
  {if $soal[dataSoal].id_soal eq $jawaban[dataJawaban].id_soal}
    {$jawaban[dataJawaban].huruf_pilihan}
    {$jawaban[dataJawaban].pilihan}
  {/if}
{/section}
```

Gambar 5.7 : *Template Code* untuk *Method* `setJawaban (db, smarty, ps)`

Sumber : [Implementasi]

Perintah *section* dalam *smarty* berfungsi untuk melakukan perintah berulang dari sebuah array yang didapat dari variable `jawaban` dan diberi nama dengan nama `dataJawaban`.

5.3.5 *Pseudocode method* `simpanJawaban($db,$smarty,$ps)` dari Kelas `kategoriList`

Method `simpanJawaban($db,$smarty,$ps)` ini dipergunakan untuk menyimpan jawaban dari masing-masing soal yang telah dikerjakan oleh peserta sekaligus menghitung nilainya. *Pseudocode* dari algoritma *method* `simpanJawaban` adalah sebagai berikut :

METHOD : `simpanJawaban PARAMETER db,smarty,ps`

DECLARATION :

```

db
smarty
ps
simpanJawaban
idSoal
Jawaban
key
Nilai

```

DESCRIPTION :

```

1   GET simpanJawaban
2   FOREACH idSoal
3       GET Jawaban[pilihan][idSoal]
4       Query jawabanBenar
5       GET key
6       IF key = jawaban[pilihan][idSoal]
7           Nilai = 1
8       ELSE
9           Nilai = 0
10      END IF
11      Query insertJawaban
12  END FOREACH
13  Query jumNilai
14  Query insertNilai

```

Gambar 5.8 : Pseudocode Algoritma *Method* simpanJawaban (db, smarty, ps)

Sumber : [Implementasi]

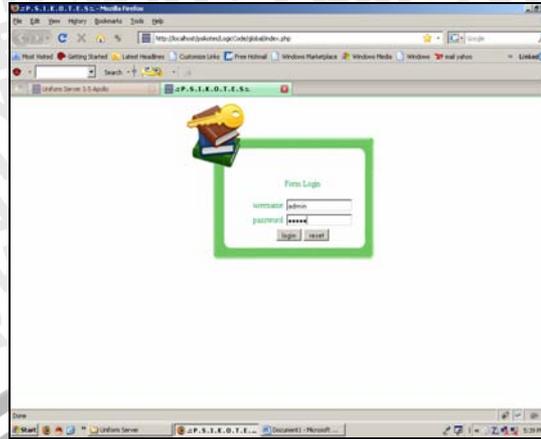
Pertama-tama *method* akan mendapatkan *value* dari simpanJawaban, kemudian untuk masing-masing soal diambil jawaban dari peserta. Selanjutnya *method* melakukan penyeleksian terhadap jawaban benar dari idSoal tersebut. Kemudian diseleksi terhadap jawaban benar dari database dan jawaban peserta. Jika hasilnya sama, maka akan bernilai 1, jika salah akan bernilai 0. Selanjutnya dilakukan penyimpanan ke dalam database baik jawabannya maupun jumlah nilainya.

5.4 Implementasi Antarmuka Kebutuhan Fungsional

Subbab 5.4 membahas semua implementasi antarmuka yang didapat dari daftar kebutuhan fungsional.

5.4.1 Implementasi Antarmuka Login

. Gambar berikut ini merupakan adalah contoh login sebagai 'admin'.



Gambar 5.9 : Halaman Login
Sumber : [Implementasi]

5.4.2 Implementasi Antarmuka Home

Halaman home untuk administrator berisi link-link untuk mengatur seluruh kebutuhan penyelenggaraan psikotes.

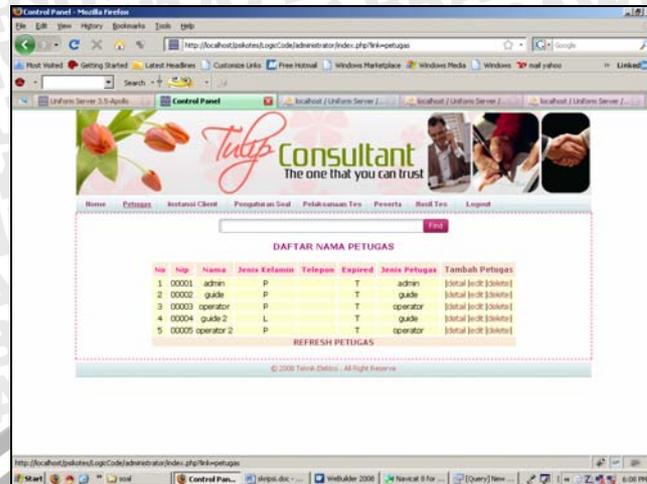


Gambar 5.10 : Halaman Home (Halaman Awal)
Sumber : [Implementasi]

5.4.3 Implementasi Antarmuka Petugas

Halaman petugas berisi daftar-daftar petugas yang terdaftar dalam suatu konsultan. Petugas dibagi menjadi tiga macam, yaitu admin, guide, dan operator. Perbedaan ketiga fungsi petugas tersebut adalah pada hak aksesnya dalam program aplikasi dan fungsi masing-masing disaat hari penyelenggaraan psikotes.





Gambar 5.11 : Halaman Petugas
Sumber : [Implementasi]

5.4.4 Implementasi Antarmuka Tambah Petugas

Halaman tambah petugas berisi form untuk menambahkan petugas baru. Form tambah petugas terdiri dari dua legend, yaitu legend detail login dan legend detail biodata petugas.

Detail Biodata Petugas

NIP: 00006

Nama: Halimatus Sa'diyah

Alamat: Jl. Arif Rahman Hakim 21 B

Kota: Malang

Propinsi: Jawa Timur

Jenis Kelamin: Perempuan Laki - laki

Tempat Lahir: Situbondo

Tanggal Lahir: 2008-12-31

Telepon: 08170476086

Email: i_tus@yahoo.com

Expired: Tidak

Jenis Petugas: guide

Agama: Islam

Keterangan: guide

Tambah

Gambar 5.12 : Legend Detail Biodata Petugas pada Form Tambah Petugas
Sumber : [Implementasi]

FORM TAMBAH DATA PETUGAS

Detail Login

Username

Password

Gambar 5.13 : Legend Detail Login pada Form Tambah Petugas
Sumber : [Implementasi]

5.4.5 Implementasi Antarmuka Edit Petugas

Halaman edit petugas berisi form yang sama dengan form tambah petugas dan berfungsi untuk mengubah biodata petugas yang telah tersimpan.

Detail Biodata Petugas

NIP

Nama

Alamat

Kota

Propinsi

Jenis Kelamin Perempuan Laki - laki

Tempat Lahir

Tanggal Lahir

Telepon

Email

Expired

Jenis Petugas

Agama

Keterangan

Gambar 5.14 : Legend Detail Biodata Petugas pada Form Edit Petugas
Sumber : [Implementasi]

FORM EDIT DATA PETUGAS

Detail Login

Username

Password

Gambar 5.15 : Legend Detail Login pada Form Edit Petugas
Sumber : [Implementasi]



5.4.6 Implementasi Antarmuka Detail Petugas

Halaman detail petugas berisi form yang sama dengan form tambah petugas dan berfungsi untuk melihat biodata petugas yang telah tersimpan tanpa mengubahnya.

Gambar 5.16 : Legend Detail Login pada Form Detail Petugas
Sumber : [Implementasi]

Gambar 5.17 : Legend Detail Biodata Petugas pada Form Detail Petugas
Sumber : [Implementasi]

5.4.7 Implementasi Antarmuka Hapus Petugas

Halaman Hapus petugas berisi konfirmasi delete petugas jika terdapat petugas yang ingin dihapus.





Gambar 5.18 : Konfirmasi Hapus Petugas
Sumber : [Implementasi]

5.4.8 Implementasi Antarmuka Cari Petugas

Form pencarian yang terletak pada halaman petugas berfungsi untuk mencari petugas berdasar nip dan nama petugas.



Gambar 5.19 : Halaman Pencarian Jika yang Dicari ada dalam Database
Sumber : [Implementasi]



Gambar 5.20 : Halaman Pencarian Jika yang Dicari Tidak Ada dalam Database
Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link ‘REFRESH PETUGAS’ untuk kembali ke halaman petugas.

5.4.9 Implementasi Antarmuka Instansi Client

Halaman instansi client berisi daftar-daftar instansi client yang menyewa pihak konsultan psikologi untuk menyelenggarakan psikotes.



Gambar 5.21 : Halaman Instansi Client
Sumber : [Implementasi]

5.4.10 Implementasi Antarmuka Tambah Instansi Client

Halaman tambah instansi client berisi form untuk menambahkan instansi client baru.



FORM TAMBAH DATA INSTANSI CLIENT

Detail Instansi Client

Nama Instansi	PT. Sampoerna Tbk
Alamat	Jl. Jakarta 15
Bidang Usaha	Perdagangan
Contact Person	Tanti
Telepon 1	034158598958
Telepon 2	
Email 1	sampoerna@yahoo.com
Email 2	
Fax	
Website	www.sampoerna.com

Gambar 5.22 : Halaman Tambah Instansi Client

Sumber : [Implementasi]

5.4.11 Implementasi Antarmuka *Edit Instansi Client*

Halaman *edit* instansi *client* berisi *form* yang sama dengan *form* tambah instansi *client* dan berfungsi untuk mengubah instansi *client* yang telah tersimpan.

FORM EDIT DATA INSTANSI CLIENT

Detail Instansi Client

Nama Instansi	PT. Sampoerna Tbk
Alamat	Jl. Jakarta 15
Bidang Usaha	Perdagangan
Contact Person	Tanti
Telepon 1	034158598958
Telepon 2	
Email 1	sampoerna@yahoo.com
Email 2	
Fax	
Website	www.sampoerna.com

Gambar 5.23 : Halaman *Edit Instansi Client*

Sumber : [Implementasi]

5.4.12 Implementasi Antarmuka *Detail* instansi *Client*.

Halaman *detail* instansi *client* berisi *form* yang sama dengan *form* tambah instansi *client* dan berfungsi untuk melihat instansi *client* yang telah tersimpan tanpa mengubahnya.

FORM DETAIL DATA INSTANSI CLIENT

Detail Instansi Client

Nama Instansi	<input type="text" value="PT. Sampoerna Tbk"/>
Alamat	<input type="text" value="Jl. Jakarta 15"/>
Bidang Usaha	<input type="text" value="Perdagangan"/>
Contact Person	<input type="text" value="Tanti"/>
Telepon 1	<input type="text" value="034158598958"/>
Telepon 2	<input type="text"/>
Email 1	<input type="text" value="sampoerna@yahoo.com"/>
Email 2	<input type="text"/>
Fax	<input type="text"/>
Website	<input type="text" value="www.sampoerna.com"/>

Gambar 5.24 : Halaman *Detail* Instansi *Client*
Sumber : [Implementasi]

5.4.13 Implementasi Antarmuka Hapus Instansi *Client*

Halaman Hapus instansi *client* berisi konfirmasi hapus instansi *client* jika terdapat instansi *client* yang ingin dihapus.

The screenshot shows the 'Tulip Consultant' web application interface. At the top, there is a navigation menu with items: Home, Petugas, Instansi Client, Pengaturan Soal, Pelaksanaan Tes, Peserta, Hasil Tes, and Logout. Below the menu is a search bar with a 'Find' button. The main content area is titled 'DAFTAR NAMA INSTANSI CLIENT' and contains a table with the following data:

No	Nama Instansi	Alamat	Bidang Usaha	Contact Person	Telepon	Fax	Website	Tambah Instansi
1	Cv. Ijen Perkasa	Jl. Bantaran Barat 11 No. 4				80001	www.ijenperkasa.net	[detail] [edit] [delete]
2	Cv. Artamas	Jl. Surabaya No. 15				9393	www.artamas.com	[detail] [edit] [delete]
3	PT. Bank Central Asia Tbk	Jl. Arif Rahman Hakim 21 B						[detail] [edit] [delete]
4	PT. Pertamina Tbk	Jl. Arif Rahman Hakim 21 A						[detail] [edit] [delete]
5	PT. Bank Indonesia Tbk		Perbankan					[detail] [edit] [delete]
6	PT. Sampoerna Tbk	Jl. Jakarta 15	Perdagangan	Tanti	034158598958		www.sampoerna.com	[detail] [edit] [delete]

Below the table is a 'REFRESH INSTANSI CLIENT' button. A confirmation dialog box is overlaid on the table, asking 'Apakah anda yakin akan menghapus record ini?' with 'OK' and 'Cancel' buttons. The dialog also shows a message: 'The page at http://localhost says: x|80001 www.ijenperkasa.net'.

© 2008 Telink Elektro - All Right Reserve

Gambar 5.25 : Halaman Hapus Instansi *Client*
Sumber : [Implementasi]

5.4.14 Implementasi Antarmuka Cari Instansi Client

Form pencarian yang terletak pada halaman instansi *client* berfungsi untuk mencari instansi client berdasar dan nama instansi.



Gambar 5.26 : Halaman Cari Instansi Client Jika yang Dicari Ada dalam Database

Sumber : [Implementasi]



Gambar 5.27 : Halaman Cari Instansi Client Jika yang Dicari Tidak Ada dalam Database

Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link 'REFRESH INSTANSI CLIENT' untuk kembali ke halaman instansi client.

5.4.15 Implementasi Antarmuka Kategori Soal

Halaman kategori soal berisi daftar-daftar kategori soal yang akan diberikan pada penyelenggaraan psikotes. Link 'kategori soal' terletak dibawah link 'Pengaturan Soal'



Gambar 5.28 : Halaman Kategori Soal
Sumber : [Implementasi]

5.4.16 Implementasi Antarmuka Tambah Kategori Soal

Halaman tambah kategori soal berisi *form* untuk menambahkan kategori soal baru.



Gambar 5.29 : Halaman Tambah Kategori Soal
Sumber : [Implementasi]

5.4.17 Implementasi Antarmuka Halaman *Edit* Kategori Soal.

Halaman *edit* kategori soal berisi *form* yang sama dengan *form* tambah kategori soal dan berfungsi untuk mengubah kategori soal yang telah tersimpan.



FORM EDIT KATEGORI SOAL

Detail Kategori Soal

Nama Kategori

Tujuan

Keterangan

Gambar 5.30 : Halaman Edit Kategori Soal
Sumber : [Implementasi]

5.4.18 Implementasi Antarmuka *Detail* Kategori Soal

Halaman *detail* kategori soal berisi *form* yang sama dengan *form* tambah kategori soal dan berfungsi untuk melihat kategori soal yang telah tersimpan tanpa mengubahnya.

FORM DETAIL KATEGORI SOAL

Detail Kategori Soal

Nama Kategori

Tujuan

Keterangan

Gambar 5.31 : Halaman *Detail* Kategori Soal
Sumber : [Implementasi]

5.4.19 Implementasi Antarmuka Cari Kategori Soal

Form pencarian yang terletak pada halaman kategori soal berfungsi untuk mencari kategori soal berdasar dan nama kategori.



Gambar 5.32 : Halaman Cari Kategori Soal Jika yang Dicari Ada dalam Database

Sumber : [Implementasi]



Gambar 5.33 : Halaman Cari Kategori Soal Jika yang Dicari Tidak Ada dalam Database

Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link 'REFRESH KATEGORI SOAL' untuk kembali ke halaman kategori soal.

5.4.20 Implementasi Antarmuka Halaman Sub Kategori

Halaman sub kategori soal berisi daftar-daftar sub kategori soal yang akan diberikan pada penyelenggaraan psikotes. Link 'sub kategori' terletak pada masing-masing kategori soal yang ada.

DAFTAR KATEGORI SOAL				
No	Nama Kategori	Tujuan	Keterangan	Tambah Kategori
1	Tes Verbal			Sub Kategori detail edit delete
2	Tes Bilangan			Sub Kategori detail edit delete
3	Tes Kepribadian			Sub Kategori detail edit delete
4	kategori 1	xxxxxx	xxxxxxxx	Sub Kategori detail edit delete
5	kategori 2			Sub Kategori detail edit delete
6	Tes Logika	Menguji logika	Penting untuk tes kecerdasan	Sub Kategori detail edit delete

REFRESH KATEGORI SOAL

Gambar 5.34 : Letak Link Sub Kategori
Sumber : [Implementasi]

The screenshot shows the 'Tulus Consultant' website interface. At the top, there is a navigation menu with links: Home, Petugas, Instansi Client, Pengaturan Soal, Pelaksanaan Tes, Peserta, Hasil Tes, and Logout. Below the menu is a search bar with a 'Find' button. The main content area displays the title 'DAFTAR SUB KATEGORI UNTUK KATEGORI TES VERBAL' and a table with the following data:

No	Nama Sub Kategori	Tambah Sub Kategori
1	Tes Sinonim	Soal detail edit delete
2	Tes Antonim	Soal detail edit delete
3	Tes Kelompok Kata	Soal detail edit delete
4	Padanan Hubungan Kata	Soal detail edit delete
5	Tes Peribendaharaan Kata	Soal detail edit delete
6	Analogi	Soal detail edit delete
7	Tes Informasi	Soal detail edit delete
8	Tes Similanti	Soal detail edit delete
9	Tes Kemampuan Vokabuler	Soal detail edit delete
10	Tes Bahasa Inggris	Soal detail edit delete

Below the table are two buttons: 'Kembali Ke Halaman Kategori' and 'Refresh Sub Kategori Soal'. At the bottom of the page, there is a copyright notice: '© 2009 Teknik Elektro . All Right Reserve'.

Gambar 5.35 : Halaman Sub Kategori
Sumber : [Implementasi]

Gambar diatas merupakan daftar sub kategori soal untuk kategori soal verbal.

5.4.21 Implementasi Antarmuka Tambah Sub Kategori Soal

Halaman tambah sub kategori soal berisi *form* untuk menambahkan sub kategori soal baru.

TAMBAH SUB KATEGORI UNTUK KATEGORI TES VERBAL

Detail Sub Kategori Soal

Nama Sub Kategori

Gambar 5.36 : Halaman Tambah Sub Kategori
Sumber : [Implementasi]



5.4.22 Implementasi Antarmuka *Edit* Sub Kategori Soal

Halaman *edit* sub kategori soal berisi *form* yang sama dengan *form* tambah sub kategori soal dan berfungsi untuk mengubah sub kategori soal yang telah tersimpan.



Gambar 5.37 :Halaman *Edit* Sub Kategori
Sumber : [Implementasi]

5.4.23 Implementasi Antarmuka Detail Sub Kategori Soal

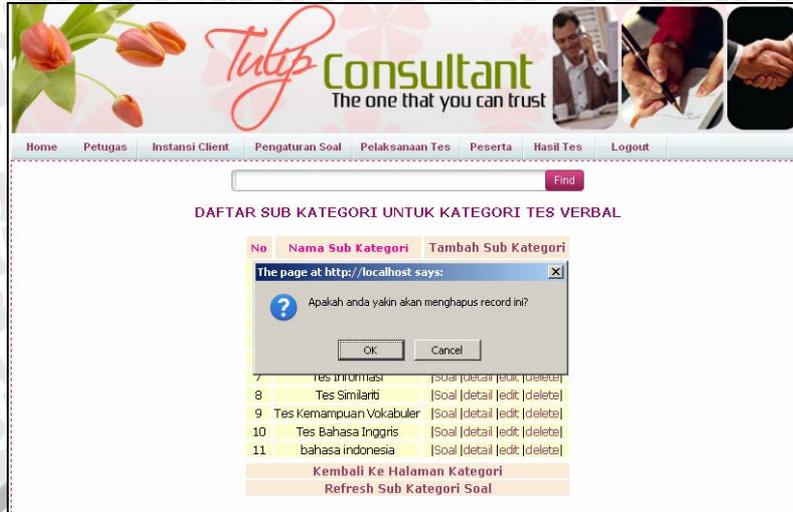
Halaman detail sub kategori soal berisi *form* yang sama dengan *form* tambah sub kategori soal dan berfungsi untuk melihat sub kategori soal yang telah tersimpan tanpa mengubahnya.



Gambar 5.38 :Halaman Detail Sub Kategori
Sumber : [Implementasi]

5.4.24 Implementasi Antarmuka Hapus Sub Kategori Soal

Halaman Hapus sub kategori soal berisi konfirmasi hapus sub kategori soal jika terdapat sub kategori soal yang ingin dihapus.



Gambar 5.39 :Halaman Hapus Sub Kategori
Sumber : [Implementasi]

5.4.25 Implementasi Antarmuka Cari Sub Kategori Soal

Form pencarian yang terletak pada halaman sub kategori soal berfungsi untuk mencari sub kategori soal berdasar dan nama sub kategori.



Gambar 5.40 :Halaman Cari Sub Kategori Jika yang Dicari Ada dalam Database
Sumber : [Implementasi]





Gambar 5.41 :Halaman Cari Sub Kategori Jika yang Dicari Tidak Ada dalam Database

Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link ‘Refresh Sub Kategori Soal’ untuk kembali ke halaman sub kategori soal, dan dapat menggunakan link ‘Kembali ke Halaman Kategori’ untuk kembali ke halaman kategori soal.

5.4.26 Implementasi Antarmuka Soal

Halaman soal berisi daftar-daftar soal yang akan diberikan pada penyelenggaraan psikotes. Link ‘soal’ terletak pada masing-masing sub kategori soal yang ada.

DAFTAR SUB KATEGORI UNTUK KATEGORI TES VERBAL

No	Nama Sub Kategori	Tambah Sub Kategori
1	Tes Sinonim	Soal detail edit delete
2	Tes Antonim	Soal detail edit delete
3	Tes Kelompok Kata	Soal detail edit delete
4	Padanan Hubungan Kata	Soal detail edit delete
5	Tes Perbendaharaan Kata	Soal detail edit delete
6	Analogi	Soal detail edit delete
7	Tes Informasi	Soal detail edit delete
8	Tes Similariti	Soal detail edit delete
9	Tes Kemampuan Vokabuler	Soal detail edit delete
10	Tes Bahasa Inggris	Soal detail edit delete
11	bahasa indonesia	Soal detail edit delete

[Kembali Ke Halaman Kategori](#)
[Refresh Sub Kategori Soal](#)

Gambar 5.42 :Letak Link Soal
Sumber : [Implementasi]

The screenshot shows the website interface for Tulip Consultant. At the top, there is a navigation menu with links: Home, Petugas, Instansi Client, Pengaturan Soal, Pelaksanaan Tes, Peserta, Hasil Tes, and Logout. Below the menu is a search bar with a 'Find' button. The main content area displays the title 'DATA SOAL PSIKOTES KATEGORI TES VERBAL SUB KATEGORI TES SINONIM' and a table of test questions.

No	Soal	Tambah Soal
1	BAKU.....	Pilihan Jawaban detail edit delete
2	ANDAL.....	Pilihan Jawaban detail edit delete
3	SIKLUS.....	Pilihan Jawaban detail edit delete
4	ASA.....	Pilihan Jawaban detail edit delete
5	RELATIF.....	Pilihan Jawaban detail edit delete
6	RABAT.....	Pilihan Jawaban detail edit delete
7	CITRA.....	Pilihan Jawaban detail edit delete
8	PROTEKSI.....	Pilihan Jawaban detail edit delete
9	INSOMNIA.....	Pilihan Jawaban detail edit delete
10	ASUMSI.....	Pilihan Jawaban detail edit delete
11	RANCU.....	Pilihan Jawaban detail edit delete
12	IMAJINASI.....	Pilihan Jawaban detail edit delete
13	OVAL.....	Pilihan Jawaban detail edit delete
14	PRIMER.....	Pilihan Jawaban detail edit delete

Gambar 5.43 :Halaman Soal
Sumber : [Implementasi]

Gambar diatas merupakan daftar soal untuk sub kategori tes sinonim dan kategori verbal.

5.4.27 Implementasi Antarmuka Tambah Soal

Halaman tambah soal berisi form untuk menambahkan soal baru.



Gambar 5.44 :Halaman Tambah Soal
Sumber : [Implementasi]

5.4.28 Implementasi Antarmuka *Edit Soal*

Halaman *edit* soal berisi *form* yang sama dengan *form* tambah soal dan berfungsi untuk mengubah soal yang telah tersimpan.

Gambar 5.45: Halaman *Edit Soal*
Sumber : [Implementasi]

5.4.29 Implementasi Antarmuka *Detail Soal*

Halaman detail soal berisi *form* yang sama dengan *form* tambah soal dan berfungsi untuk melihat soal yang telah tersimpan tanpa mengubahnya.

Gambar 5.46: Halaman Detail Soal
Sumber : [Implementasi]

5.4.30 Implementasi Antarmuka Hapus Soal

Halaman Hapus soal berisi konfirmasi hapus soal jika terdapat soal yang ingin dihapus.



Gambar 5.47: Halaman Hapus Soal
Sumber : [Implementasi]

5.4.31 Implementasi Antarmuka Cari Soal

Form pencarian yang terletak pada halaman soal berfungsi untuk mencari soal berdasar cuplikan soal.



Gambar 5.48: Halaman Cari Soal Jika yang Dicari Ada dalam Database
Sumber : [Implementasi]



Gambar 5.49: Halaman Cari Soal Jika yang Dicari Tidak Ada dalam Database
Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link ‘Refresh Soal’ untuk kembali ke halaman soal, dan dapat menggunakan link ‘Kembali ke Halaman Sub Kategori’ untuk kembali ke halaman sub kategori soal.

5.4.32 Implementasi Antarmuka Pilihan Jawaban.

Halaman pilihan jawaban berisi daftar-daftar pilihan jawaban yang akan diberikan pada penyelenggaraan psikotes. Link ‘pilihan jawaban’ terletak pada masing-masing soal yang ada.

No	Soal	Tambah Soal
1	BAKU.....	Pilihan Jawaban detail edit delete
2	ANDAL.....	Pilihan Jawaban detail edit delete
3	SIKLUS.....	Pilihan Jawaban detail edit delete
4	ASA.....	Pilihan Jawaban detail edit delete
5	RELATIF.....	Pilihan Jawaban detail edit delete
6	RABAT.....	Pilihan Jawaban detail edit delete
7	CITRA.....	Pilihan Jawaban detail edit delete
8	PROTEKSI.....	Pilihan Jawaban detail edit delete
9	INSOMNIA.....	Pilihan Jawaban detail edit delete
10	ASUMSI.....	Pilihan Jawaban detail edit delete
11	RANCU.....	Pilihan Jawaban detail edit delete
12	IMAJINASI.....	Pilihan Jawaban detail edit delete
13	OVAL.....	Pilihan Jawaban detail edit delete
14	PRIMER.....	Pilihan Jawaban detail edit delete
15	DIKOTOMI.....	Pilihan Jawaban detail edit delete
16	INSPEKSI.....	Pilihan Jawaban detail edit delete
17	DINAMIS.....	Pilihan Jawaban detail edit delete
18	RELASI.....	Pilihan Jawaban detail edit delete
19	SUPERVISI.....	Pilihan Jawaban detail edit delete
20	KORELASI.....	Pilihan Jawaban detail edit delete

Gambar 5.50: Letak Link Pilihan Jawaban
Sumber : [Implementasi]



Gambar 5.51: Halaman Pilihan Jawaban
Sumber : [Implementasi]

5.4.33 Implementasi Antarmuka Tambah Pilihan Jawaban

Halaman tambah pilihan jawaban berisi form untuk menambahkan pilihan jawaban baru.

The screenshot shows the "FORM TAMBAH PILIHAN JAWABAN" interface. It contains the following fields and controls:

- Detail Jawaban Soal:** A section header for the form.
- Soal:** A text input field containing the word "BAKU".
- Huruf Pilihan:** A dropdown menu currently showing "E".
- Pilihan:** A text input field containing the word "KECIL".
- Status Jawaban:** A dropdown menu currently showing "Salah".
- Tambah:** A button to submit the form.

Gambar 5.52 : Halaman Tambah Pilihan Jawaban
Sumber : [Implementasi]

5.4.34 Implementasi Antarmuka Edit Pilihan Jawaban

Halaman edit pilihan jawaban berisi form yang sama dengan form tambah pilihan jawaban dan berfungsi untuk mengubah pilihan jawaban yang telah tersimpan.

Gambar 5.53 : Halaman *Edit* Pilihan Jawaban
 Sumber : [Implementasi]

5.4.35 Implementasi Antarmuka Detail Pilihan Jawaban

Halaman detail pilihan jawaban berisi form yang sama dengan form tambah pilihan jawaban dan berfungsi untuk melihat pilihan jawaban yang telah tersimpan tanpa mengubahnya.

Gambar 5.54 : Halaman *Detail* Pilihan Jawaban
 Sumber : [Implementasi]

5.4.36 Implementasi Antarmuka Hapus Pilihan Jawaban

Halaman Hapus pilihan jawaban berisi konfirmasi hapus pilihan jawaban jika terdapat pilihan jawaban yang ingin dihapus.



Gambar 5.55 : Halaman Hapus Pilihan Jawaban
Sumber : [Implementasi]

5.4.37 Implementasi Antarmuka Urutan Kategori Soal

Halaman urutan kategori soal berisi daftar-daftar urutan kategori soal yang akan diberikan pada penyelenggaraan psikotes di suatu tempat. Link ‘Urutan Kategori Soal’ terletak di bawah link ‘Pengaturan Soal’.



Gambar 5.56 : Halaman Urutan Kategori Soal
Sumber : [Implementasi]

5.4.38 Implementasi Antarmuka Tambah Urutan Kategori Soal

Halaman tambah urutan kategori soal berisi form untuk menambahkan urutan kategori soal baru.



FORM TAMBAH URUTAN KATEGORI SOAL

Detail Urutan Kategori Soal

Lokasi Tes: 2008-12-10 08:00:00

Kategori Soal:

Sub Kategori:

No Urut:

Lama Pengerjaan: menit

Jumlah Soal:

Gambar 5.57 : Halaman Tambah Urutan Kategori Soal
Sumber : [Implementasi]

5.4.39 Implementasi Antarmuka *Edit* Urutan Kategori Soal

Halaman *edit* urutan kategori soal berisi *form* yang sama dengan *form* tambah urutan kategori soal dan berfungsi untuk mengubah urutan kategori soal yang telah tersimpan.

FORM EDIT URUTAN KATEGORI SOAL

Detail Urutan Kategori Soal

Lokasi Tes: 2008-12-10 08:00:00

Kategori Soal:

Sub Kategori:

No Urut:

Lama Pengerjaan: menit

Jumlah Soal:

Gambar 5.58 : Halaman *Edit* Urutan Kategori Soal
Sumber : [Implementasi]

5.4.40 Implementasi Antarmuka *Detail* Urutan Kategori Soal

Halaman *detail* urutan kategori soal berisi *form* yang sama dengan *form* tambah urutan kategori soal dan berfungsi untuk melihat urutan kategori soal yang telah tersimpan tanpa mengubahnya.

FORM DETAIL URUTAN KATEGORI SOAL

Detail Urutan Kategori Soal

Lokasi Tes:

Kategori Soal:

Sub Kategori:

No Urut:

Lama Pengerjaan: menit

Jumlah Soal:

Gambar 5.59 : Halaman Detail Urutan Kategori Soal
Sumber : [Implementasi]

5.4.41 Implementasi Antarmuka Hapus Urutan Kategori Soal

Halaman hapus urutan kategori soal berisi konfirmasi hapus urutan kategori soal jika terdapat urutan kategori soal yang ingin dihapus.

The screenshot shows a web application interface for 'Tulip Consultant'. The main content is a table titled 'DAFTAR URUTAN KATEGORI SOAL'. A confirmation dialog box is overlaid on the table, asking 'Apakah anda yakin akan menghapus record ini?' (Are you sure you want to delete this record?). The dialog has 'OK' and 'Cancel' buttons. The table contains the following data:

No	Lokasi Tes	Tanggal	Waktu	Kategori	Sub Kategori	No Urut Kategori	Lama Pengerjaan (menit)	Jumlah Soal	Tambah Urutan Kategori Soal
1	SMU 3 Malang	2008-12-10	08:00:00	Tes Verbal	Tes Sinonim	1	25	20	[detail] [edit] [delete]
2	SMU 3 Malang	2008-12-10	08:00:00	The page at http://localhost says:				10	[detail] [edit] [delete]
3	SMU 1 Malang	2008-12-12	07:00:00	Apakah anda yakin akan menghapus record ini?				10	[detail] [edit] [delete]
4	SMU 1 Malang	2008-12-12	07:00:00					15	[detail] [edit] [delete]
5	SMU 1 Malang	2008-12-12	07:00:00					10	[detail] [edit] [delete]
6	SMU 2 Malang	2008-12-13	07:00:00	Tes Verbal	Tes Sinonim	1	25	25	[detail] [edit] [delete]
7	SMU 2 Malang	2008-12-13	07:00:00	Tes Verbal	Tes Antonim	2	30	30	[detail] [edit] [delete]

Gambar 5.60 : Halaman Hapus Urutan Kategori Soal
Sumber : [Implementasi]

5.4.42 Implementasi Antarmuka Cari Urutan Kategori Soal

Form pencarian yang terletak pada halaman urutan kategori soal berfungsi untuk mencari urutan kategori soal berdasar tempat pelaksanaan tes.



The screenshot shows the Tulip Consultant website interface. At the top, there is a navigation menu with items: Home, Petugas, Instansi Client, Pengaturan Soal, Pelaksanaan Tes, Peserta, Hasil Tes, and Logout. Below the menu is a search bar containing 'smu 3' and a 'Find' button. The main content area displays the title 'DAFTAR URUTAN KATEGORI SOAL' and a table with the following data:

No	Lokasi Tes	Tanggal	Waktu	Kategori	Sub Kategori	No Urut Kategori	Lama Pengerjaan (menit)	Jumlah Soal	Tambah Urutan Kategori Soal
1	SMU 3 Malang	2008-12-10	08:00:00	Tes Verbal	Tes Sinonim	1	25	20	[detail] [edit] [delete]
2	SMU 3 Malang	2008-12-10	08:00:00	Tes Verbal	Tes Sinonim	2	1	10	[detail] [edit] [delete]

Below the table, there is a 'REFRESH URUTAN KATEGORI SOAL' button and a copyright notice: '© 2008 Teknik Elektro . All Right Reserve'.

Gambar 5.61 : Halaman Cari Urutan Kategori Soal Jika yang Dicari Ada dalam Database

Sumber : [Implementasi]

The screenshot shows the Tulip Consultant website interface. At the top, there is a navigation menu with items: Home, Petugas, Instansi Client, Pengaturan Soal, Pelaksanaan Tes, Peserta, Hasil Tes, and Logout. Below the menu is a search bar containing 'smu 5' and a 'Find' button. The main content area displays the title 'DAFTAR URUTAN KATEGORI SOAL' and a message: 'DATA TIDAK ADA'. Below this message, there is a 'REFRESH URUTAN KATEGORI SOAL' button and a copyright notice: '© 2008 Teknik Elektro . All Right Reserve'.

Gambar 5.62 : Halaman Cari Urutan Kategori Soal Jika yang Dicari Tidak Ada dalam Database

Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link 'REFRESH URUTAN KATEGORI SOAL' untuk kembali ke halaman urutan kategori soal.

5.4.43 Implementasi Antarmuka Pelaksanaan Tes

Halaman pelaksanaan tes berisi daftar-daftar tempat pelaksanaan tes beserta petugas yang akan bertugas disaat hari pelaksanaan tes.

No	Tanggal	Waktu	Tempat	Instansi Client	Guide 1	Guide 2	Operator 1	Operator 2	Tambah Pelaksanaan Tes
1	2008-12-10	08:00:00	SMU 3 Malang	Cv. Artamas	guide	guide	operator	operator	[detail] [edit] [delete]
2	2008-12-12	07:00:00	SMU 1 Malang	Cv. Artamas	guide	guide	operator	operator	[detail] [edit] [delete]
3	2008-12-13	07:00:00	SMU 2 Malang	Cv. Ijen Perkasa	guide	guide	operator	operator	[detail] [edit] [delete]
4	2008-12-15	07:00:00	SMU 5 Malang	Cv. Artamas	guide	guide	operator	operator	[detail] [edit] [delete]
5	2008-12-17	07:30:00	SMU 10 Malang	PT. Bank Indonesia Tbk	guide 2	guide	operator 2	operator	[detail] [edit] [delete]

Gambar 5.63 : Halaman Pelaksanaan Tes
Sumber : [Implementasi]

5.4.44 Implementasi Antarmuka Tambah Pelaksanaan Tes

Halaman tambah pelaksanaan tes berisi *form* untuk menambahkan tempat pelaksanaan tes baru.

Gambar 5.64 : Halaman Tambah Pelaksanaan Tes
Sumber : [Implementasi]

5.4.45 Implementasi Antarmuka Edit Pelaksanaan Tes

Halaman *edit* pelaksanaan tes berisi *form* yang sama dengan *form* tambah pelaksanaan tes dan berfungsi untuk mengubah pelaksanaan tes yang telah tersimpan.



FORM EDIT PELAKSANAAN PSIKOTES

Detail Pelaksanaan Tes

Tanggal (yyyy-mm-dd)

Waktu (hh:mm:ss)

Tempat

Alamat

Instansi Client

Guide 1

Guide 2

Operator 1

Operator 2

Gambar 5.65 : Halaman *Edit Pelaksanaan Tes*
Sumber : [Implementasi]

5.4.46 Implementasi Antarmuka *Detail Pelaksanaan Tes*

Halaman detail pelaksanaan tes berisi *form* yang sama dengan *form* tambah pelaksanaan tes dan berfungsi untuk melihat pelaksanaan tes yang telah tersimpan tanpa mengubahnya.

FORM DETAIL PELAKSANAAN TES PSIKOTES

Detail Pelaksanaan Tes

Tanggal (yyyy-mm-dd)

Waktu (hh:mm:ss)

Tempat

Alamat

Instansi Client

Guide 1

Guide 2

Operator 1

Operator 2

Gambar 5.66 : Halaman *Detail Pelaksanaan Tes*
Sumber : [Implementasi]

5.4.47 Implementasi Antarmuka *Hapus Pelaksanaan Tes*

Halaman hapus pelaksanaan tes berisi konfirmasi hapus pelaksanaan tes jika terdapat pelaksanaan tes yang ingin dihapus.



Gambar 5.67 : Halaman Hapus Pelaksanaan Tes
Sumber : [Implementasi]

5.4.48 Implementasi Antarmuka Cari Pelaksanaan Tes

Form pencarian yang terletak pada halaman pelaksanaan tes berfungsi untuk mencari pelaksanaan tes berdasar tempat pelaksanaan tes atau instansi client.



Gambar 5.68 : Halaman Cari Pelaksanaan Tes Jika yang Dicari Ada dalam Database
Sumber : [Implementasi]





Gambar 5.69 : Halaman Cari Pelaksanaan Tes Jika yang Dicari Tidak Ada dalam Database

Sumber : [Implementasi]

5.4.49 Implementasi Antarmuka Data Peserta Tes

Halaman data peserta tes berisi daftar-daftar peserta tes. Link ‘Data Peserta Tes’ berada dibawah Link ‘Peserta Tes’.



Gambar 5.70 : Halaman Data Peserta Tes

Sumber : [Implementasi]

5.4.50 Implementasi Antarmuka Tambah Peserta Tes

Halaman tambah peserta tes berisi *form* untuk menambahkan peserta tes baru.

FORM TAMBAH DATA PESERTA TES

Detail Biodata Peserta

NIP: 00006

Nama: Halimatus Sa'diyah

Alamat: Jl. Arif Rahman Hakim 21 B

Pendidikan Terakhir: S1

Jurusan: Teknik Elektro

Jenis Kelamin: Perempuan Laki - laki

Tanggal Lahir: 1984-12-31

Telepon: 08170476086

Agama: Islam

Tambah

Gambar 5.71 : Halaman Tambah Peserta Tes
Sumber : [Implementasi]

5.4.51 Implementasi Antarmuka *Edit* Peserta Tes

Halaman edit peserta tes berisi *form* yang sama dengan *form* tambah peserta tes dan berfungsi untuk mengubah biodata peserta tes yang telah tersimpan.

FORM EDIT DATA PESERTA

Detail Biodata Peserta

NIP: 00006

Nama: Halimatus Sa'diyah

Alamat: Jl. Arif Rahman Hakim 21 B

Pendidikan Terakhir: S1

Jurusan: Teknik Elektro

Jenis Kelamin: Perempuan Laki - laki

Tanggal Lahir: 1984-12-31

Telepon: 08170476086

Agama: Islam

Simpan

Gambar 5.72 : Halaman *Edit* Peserta Tes
Sumber : [Implementasi]

5.4.52 Implementasi Antarmuka Detail Peserta Tes

Halaman detail peserta tes berisi *form* yang sama dengan *form* tambah peserta tes dan berfungsi untuk melihat peserta tes yang telah tersimpan tanpa mengubahnya.

FORM DETAIL DATA PESERTA

Detail Biodata Peserta

NIP: 00006

Nama: Halimatus Sa'diyah

Alamat: Jl. Arif Rahman Hakim 21 B

Pendidikan Terakhir: S1

Jurusan: Teknik Elektro

Jenis Kelamin: Perempuan Laki - laki

Tanggal Lahir: 1984-12-31

Telepon: 08170476086

Agama: Islam

Gambar 5.73 : Halaman Detail Peserta Tes
Sumber : [Implementasi]

5.4.53 Implementasi Antarmuka Hapus Peserta Tes

Halaman hapus peserta tes berisi konfirmasi hapus peserta tes jika terdapat peserta tes yang ingin dihapus.

The screenshot shows the 'DAFTAR NAMA PESERTA' (Participant List) page. A table lists participants with columns for No, Nip, Nama, Alamat, Jenis Kelamin, Pendidikan Terakhir, Jurusan, and Tambah Peserta. A confirmation dialog box is overlaid on the table, asking 'Apakah anda yakin akan menghapus record ini?' (Are you sure you want to delete this record?).

No	Nip	Nama	Alamat	Jenis Kelamin	Pendidikan Terakhir	Jurusan	Tambah Peserta
1	00001	Ria Wulandari				Teknik Elektro	[detail] [edit] [delete]
2	00002	Budi Hartanto					[detail] [edit] [delete]
3	00003	Halimatus					[detail] [edit] [delete]
4	00004	Adita Riski P					[detail] [edit] [delete]
5	00005	Mayang Rekyandisa					[detail] [edit] [delete]
6	00006	Halimatus Sa'diyah	Jl. Arif			Teknik Elektro	[detail] [edit] [delete]

Gambar 5.74: Halaman Hapus Peserta Tes
Sumber : [Implementasi]

5.4.54 Implementasi Antarmuka Cari Peserta Tes

Form pencarian yang terletak pada halaman peserta tes berfungsi untuk mencari peserta tes berdasar nama peserta atau nomor peserta.





Gambar 5.75: Halaman Cari Peserta Tes Jika yang Dicari Ada dalam Database
Sumber : [Implementasi]



Gambar 5.76: Halaman Cari Peserta Tes Jika yang Dicari Tidak Ada dalam Database
Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link ‘REFRESH PESERTA TES’ untuk kembali ke halaman urutan kategori soal.

5.4.55 Implementasi Antarmuka Data Peserta di Lokasi Tes

Halaman data peserta di lokasi tes berisi daftar-daftar peserta tes di suatu lokasi psikotes. Link ‘Data Peserta di Lokasi Tes’ berada dibawah Link ‘Peserta Tes’.



Gambar 5.77: Halaman Data Peserta di Lokasi Tes
Sumber : [Implementasi]

5.4.56 Implementasi Antarmuka Tambah Peserta di Lokasi Tes

Halaman tambah peserta di lokasi tes berisi *form* untuk menambahkan peserta di lokasi tes baru.

Gambar 5.78: Halaman Tambah Data Peserta di Lokasi Tes
Sumber : [Implementasi]

5.4.57 Implementasi Antarmuka *Edit* Peserta di Lokasi Tes

Halaman *edit* peserta di lokasi tes berisi *form* yang sama dengan *form* tambah peserta di lokasi tes dan berfungsi untuk mengubah peserta di lokasi tes yang telah tersimpan atau untuk memberi login peserta di lokasi tes.



Gambar 5.79: Halaman *Edit* Data Peserta di Lokasi Tes
Sumber : [Implementasi]

5.4.58 Implementasi Antarmuka *Detail* Peserta di Lokasi Tes

Halaman *detail* peserta di lokasi tes berisi *form* yang sama dengan *form* tambah peserta di lokasi tes dan berfungsi untuk melihat peserta di lokasi tes yang telah tersimpan tanpa mengubahnya.

Gambar 5.80: Halaman *Detail* Data Peserta di Lokasi Tes
Sumber : [Implementasi]

5.4.59 Implementasi Antarmuka *Hapus* Peserta di Lokasi Tes

Halaman *hapus* peserta di lokasi tes berisi konfirmasi *hapus* peserta di lokasi tes jika terdapat peserta di lokasi tes yang ingin dihapus.



Gambar 5.81: Halaman Hapus Data Peserta di Lokasi Tes
Sumber : [Implementasi]

5.4.60 Implementasi Antarmuka Cari Peserta di Lokasi Tes

Form pencarian yang terletak pada halaman peserta di lokasi tes berfungsi untuk mencari peserta di lokasi tes berdasar nama peserta, nomor peserta, atau lokasi tes.



Gambar 5.82: Halaman Cari Data Peserta di Lokasi Tes Jika yang Dicari Ada dalam Database

Sumber : [Implementasi]



Gambar 5.83: Halaman Cari Data Peserta di Lokasi Tes Jika yang Dicari Tidak Ada dalam Database
Sumber : [Implementasi]

Setelah melakukan pencarian, maka dapat menggunakan link 'REFRESH PESERTA DI LOKASI TES' untuk kembali ke halaman urutan kategori soal.

5.4.61 Implementasi Antarmuka Hasil Tes

Halaman hasil tes berisi daftar-daftar hasil tes peserta.



Gambar 5.84: Halaman Hasil Tes
Sumber : [Implementasi]

5.4.62 Implementasi Antarmuka Edit Hasil Tes

Halaman *edit* hasil tes berfungsi untuk menambah atau mengubah nilai konsultan.

FORM EDIT DATA HASIL TES PESERTA

Detail Peserta

NIP	<input type="text" value="00002"/>
Nama	<input type="text" value="Budi Hartanto"/>
Kategori Soal	<input type="text" value="Tes Verbal"/>
Sub Kategori Soal	<input type="text" value="Tes Antonim"/>
Nilai	<input type="text" value="0"/>

Detail Nilai Konsultan

Nilai Konsultan

Gambar 5.85: Halaman *Edit* Hasil Tes
Sumber : Implementasi

5.4.63 Implementasi Antarmuka Detail Hasil Tes

Halaman detail hasil tes berfungsi untuk melihat hasil tes yang telah tersimpan tanpa mengubahnya.

FORM DETAIL DATA HASIL TES PESERTA

Detail Peserta

NIP	<input type="text" value="00002"/>
Nama	<input type="text" value="Budi Hartanto"/>
Kategori Soal	<input type="text" value="Tes Verbal"/>
Sub Kategori Soal	<input type="text" value="Tes Antonim"/>
Nilai	<input type="text" value="0"/>

Detail Nilai Konsultan

Nilai Konsultan

Gambar 5.86: Halaman Detail Hasil Tes
Sumber : [Implementasi]

5.4.64 Implementasi Antarmuka Hapus Hasil Tes

Halaman hapus hasil tes berisi konfirmasi hapus hasil tes jika terdapat hasil tes yang ingin dihapus.



Gambar 5.87: Halaman Hapus Hasil Tes
Sumber : [Implementasi]

5.4.65 Implementasi Antarmuka Cari Hasil Tes

Form pencarian yang terletak pada halaman hasil tes berfungsi untuk mencari hasil tes berdasar nama peserta, nomor peserta, atau lokasi tes.



Gambar 5.88: Halaman Cari Hasil Tes Jika yang Dicari Ada dalam Database
Sumber : [Implementasi]



Gambar 5.89 : Halaman Cari Hasil Tes Jika yang Dicari Tidak Ada dalam Database

Sumber : [Implementasi]

5.4.66 Implementasi Antarmuka Login Peserta

Gambar dibawah ini merupakan contoh login peserta tes.



Gambar 5.90 : Login Peserta

Sumber : [Implementasi]

5.4.67 Implementasi Antarmuka Tes Kategori Pertama

Berikut merupakan gambar daftar kategori pertama yang harus dikerjakan oleh peserta.



Gambar 5.91 : Halaman Urutan Kategori Tes Pertama
Sumber : [Implementasi]

5.4.68 Implementasi Antarmuka Menjawab Soal Kategori Pertama

Berikut merupakan daftar soal dan jawaban yang disediakan untuk kategori pertama dengan batas waktu yang telah ditentukan.



Gambar 5.92 : Halaman Soal Kategori Tes Pertama
Sumber : [Implementasi]

5.4.69 Implementasi Antarmuka Tes Kategori Kedua

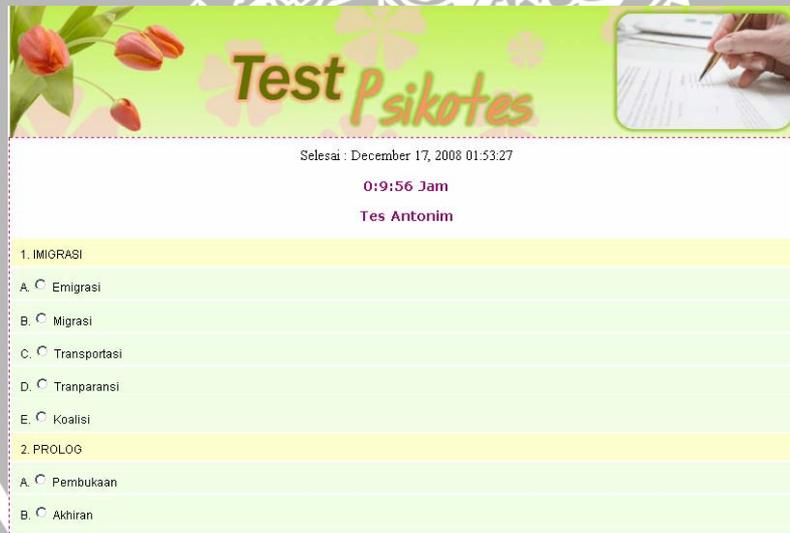
Berikut merupakan gambar daftar kategori kedua yang harus dikerjakan oleh peserta.



Gambar 5.93 : Halaman Urutan Kategori Tes Kedua
Sumber : [Implementasi]

5.4.70 Implementasi Antarmuka Menjawab Soal Kategori Kedua

Berikut merupakan daftar soal dan jawaban yang disediakan untuk kategori kedua dengan batas waktu yang telah ditentukan.



Gambar 5.94 : Halaman Soal Kategori Tes Kedua
Sumber : [Implementasi]

5.4.71 Implementasi Antarmuka Peringatan Jika Menjawab Soal Melebihi Batas Waktu yang Ditentukan.

Peringatan akan muncul jika peserta mengerjakan soal melebihi batas waktu yang ditentukan, kemudian sistem akan langsung menuju daftar kategori selanjutnya.



Gambar 5.95 : Halaman Peringatan
Sumber : [Implementasi]

5.4.72 Halaman Terakhir Peserta

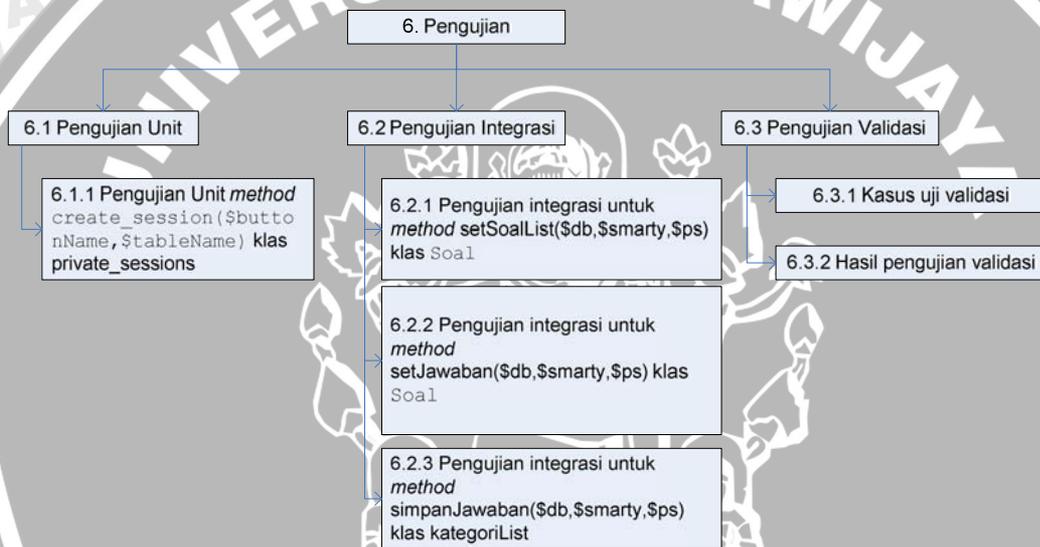
Berikut merupakan gambar bahwa peserta telah selesai mengerjakan keseluruhan tes.



Gambar 5.96 : Halaman Terakhir Psikotes
Sumber : [Implementasi]

BAB VI PENGUJIAN

Bab ini membahas proses pengujian terhadap sistem psikotes *online* yang telah dibangun. Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian unit, pengujian integrasi dan pengujian validasi. Pengujian unit dan pengujian integrasi menggunakan teknik pengujian *white box* (*white box Testing*). Pengujian validasi menggunakan teknik pengujian *black box* (*black box Testing*). Gambar 6.1 menunjukkan langkah-langkah pengujian yang digunakan pada bab ini.



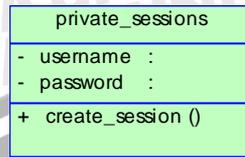
Gambar 6.1 : Diagram Pohon Pengujian
Sumber : Pengujian

6.1 Pengujian Unit

Sistem yang dibangun dengan paradigma pemrograman berorientasi objek menerapkan pengujian unit untuk suatu metode (operasi) dari suatu klas. Pada pengujian unit sistem psikotes *online* ini, digunakan teknik pengujian *White Box* (*White Box Testing*) dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan. Penulisan laporan skripsi ini

hanya dicantumkan hasil pengujian unit untuk algoritma dari beberapa metode (operasi) saja (tidak untuk keseluruhan metode).

Klas uji untuk keseluruhan *method* yang diuji adalah sama. Diagram klas untuk klas uji tersebut dapat dilihat pada gambar 6.2.



Gambar 6.2 : Klas Diagram *private_sessions*
Sumber : [Pengujian]

a. Pengujian Unit untuk *Method* `create_session()` klas *private_sessions*

Method `create_session` ini dipergunakan untuk mendapat session.

Gambar 6.3 memperlihatkan proses pemodelan dalam *flow graph* pada *method* `create_session()`.



Gambar 6.3 : Pemodelan Operasi `create_session()` ke dalam *flow graph*
Sumber : [Pengujian]

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* `create_session()` menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 0 - 1 + 2 \\
 &= 1
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 1 ditentukan sebuah basis set dari jalur independen yaitu:

Jalur 1 : 1

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk kasus uji dijelaskan pada Tabel 6.1.



Tabel 6.1 Test case untuk Pengujian Integrasi *method* `create_session ()`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	<code>create_session ()</code>	Method mengembalikan session.	Method mengembalikan session.

Sumber: [Pengujian]

Berdasar tabel 6.1. didapat bahwa semua jalur tidak memiliki kesalahan dan mendapatkan hasil sesuai dengan yang diharapkan.

6.2 Pengujian Integrasi

Pengujian integrasi diterapkan pada proses yang mengintegrasikan fungsionalitas dari beberapa klas untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi yang dijadikan sebagai obyek uji adalah klas-klas yang menggabungkan kinerja dari klas-klas yang lain. Klas-klas yang mengintegrasikan beberapa klas yang lain tersebut akan berperan juga sebagai *test driver* yang mengendalikan proses pengujian sehingga bisa memperlihatkan status valid atau tidaknya hasil integrasi. Dalam melakukan pengujian integrasi terhadap sistem perangkat lunak ini digunakan strategi *bottom-up*.

Pada pengujian integrasi perangkat lunak psikotes *online* ini, digunakan teknik pengujian *White Box (White Box Testing)* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan. Penulisan laporan skripsi ini hanya dicantumkan hasil pengujian integrasi untuk algoritma dari beberapa metode (operasi) saja (tidak untuk keseluruhan metode).

a. Pengujian Integrasi untuk *Method* `setSoalList(db,smarty,ps)` klas `soal`

Method `setSoalList(db,smarty,ps)` ini ini dipergunakan untuk menampilkan daftar soal yang harus dikerjakan oleh peserta. Gambar 6.4 memperlihatkan proses pemodelan dalam *flow graph* pada *method* `setSoalList(db,smarty,ps)`.

METHOD : setSoalList PARAMETER db,smarty,ps

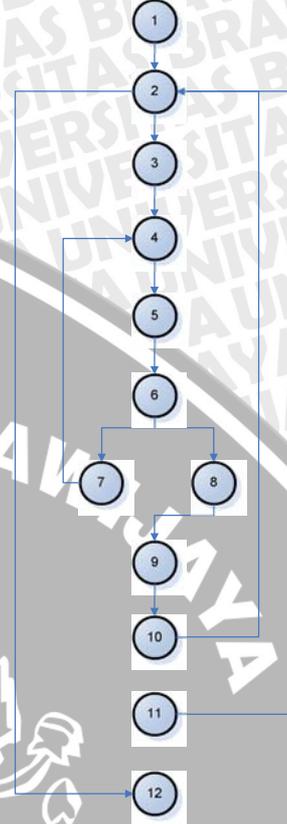
DECLARATION :

db
smarty
ps
jumlahSoal
indexSoalYgDipilih
dataSoalYangTelahDirandom
controlLooping
indexYangDirandom
nilai
indexSoalYgDipilih
dataSoal

DESCRIPTION :

```
1 Get jumlahSoal
2 ① Query dataSoal limit jumlahSoal
3   indexSoalYgDipilih = array()
4   dataSoalYangTelahDirandom =
5   array(array())
6   ② FOR i = 0 TO i < jumlahSoal
7     ③ controlLooping = true
8     ④ WHILE (controlLooping)
9       indexYangDirandom =
10      random(0, jumlahSoal-1)
11
12      nilai=in_array(indexYangDirandom,
13      indexSoalYgDipilih)
14      ⑥ IF nilai = 1 THEN
15        controlLooping = true ⑦
16      ⑧ ELSE
17        controlLooping = false
18      END IF
19
20      ⑨ IF controlLooping = false
21        indexSoalYgDipilih[$i]=
22        indexYangDirandom
23        dataSoalYangTelahDirandom[$i]=
24        dataSoal[$indexYangDirandom] ⑩
25      END IF
26 ⑫ END WHILE ⑪
27 END FOR
```

Node (N) = 12
Edge (E) = 13



Gambar 6.4 : Pemodelan operasi setSoalList (db,smarty,ps) ke dalam flow graph

Sumber : [Pengujian]

Pemodelan ke dalam flow graph yang telah dilakukan terhadap method setSoalList. Jumlah kompleksitas siklomatis (cyclomatic complexity) melalui persamaan $V(G) = E - N + 2$.

$$V(G) = 13 - 12 + 2 \\ = 3$$

Dari nilai cyclomatic complexity yang telah dihasilkan dari perhitungan yaitu 3 ditentukan dua buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-12

Jalur 2 : 1-2-3-4-5-6-7-4-5-6-7...8-9-10-2-...-12

Jalur 3 : 1-2-3-4-5-6-8-9-10-2-....-12

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah dijelaskan pada Tabel 6.2.

Tabel 6.2 *Test case* untuk pengujian integrasi *method* `setSoalList`

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	<code>jumlahSoal = 0</code>	Tidak ada soal	Tidak ada soal
2	<code>controlLooping</code> bernilai false	Soal dimasukkan kedalam array <code>dataSoal</code>	Soal dimasukkan kedalam array <code>dataSoal</code>
3	<code>controlLooping</code> bernilai true	Soal tidak dimasukkan kedalam array <code>dataSoal</code> dan index soal diacak kembali	Soal tidak dimasukkan kedalam array <code>dataSoal</code> dan index soal diacak kembali

Sumber: [Pengujian]

Berdasar tabel 6.2. didapat bahwa semua jalur tidak memiliki kesalahan dan mendapatkan hasil sesuai dengan yang diharapkan.

b. Pengujian Integrasi untuk *Method* `setJawaban(db, smarty, ps)` Kelas `soal`

Method `setJawaban(db, smarty, ps)` ini dipergunakan untuk menampilkan daftar jawaban dari soal yang harus dikerjakan oleh peserta. Gambar 6.5 memperlihatkan proses pemodelan dalam *flow graph* pada *method* `setJawaban(db, smarty, ps)`.



Gambar 6.5 : Pemodelan operasi `setJawaban(db, smarty, ps)` ke dalam *flow graph*

Sumber: [Pengujian]

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* `setJawaban`. Jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$.

$$V(G) = 0 - 1 + 2 = 1$$



Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.3.

Tabel 6.3 Test case untuk pengujian integrasi *method* `setJawaban`

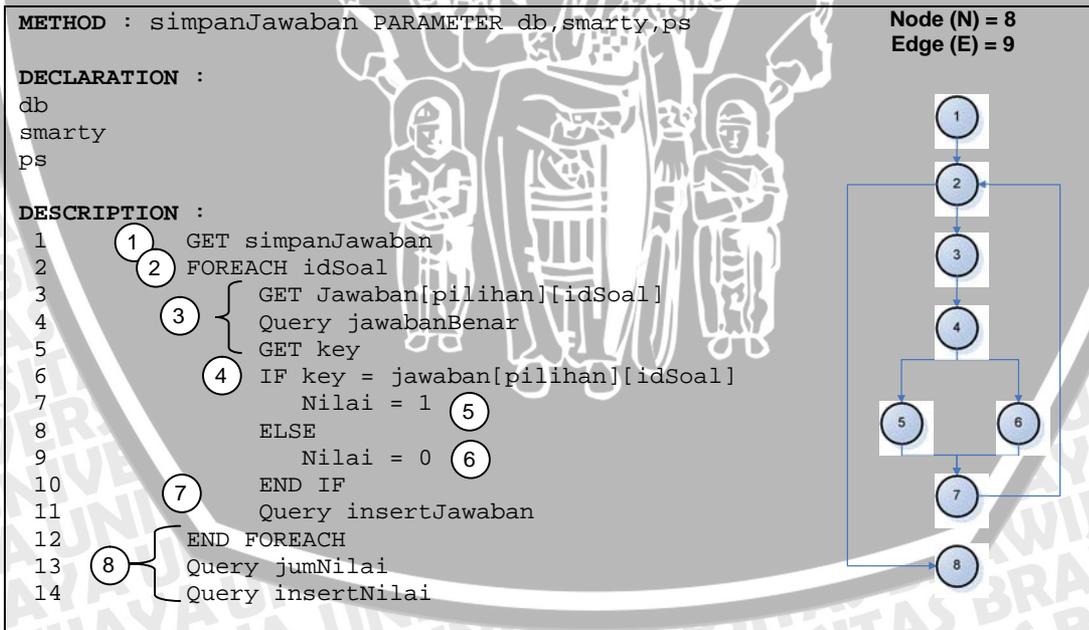
Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Query dataJawaban	Menampilkan hasil seleksi jawaban soal	Menampilkan hasil seleksi jawaban soal

Sumber: [Pengujian]

Berdasar tabel 6.3. didapat bahwa semua jalur tidak memiliki kesalahan dan mendapatkan hasil sesuai dengan yang diharapkan.

c. Pengujian Integrasi untuk *Method* `simpanJawaban(db, smarty, ps)` Kelas kategoriList

Method `simpanJawaban(db, smarty, ps)` ini dipergunakan untuk menyimpan jawaban dari masing-masing soal yang telah dikerjakan oleh peserta. Gambar 6.6 memperlihatkan proses pemodelan dalam *flow graph* pada *method* `simpanJawaban(db, smarty, ps)`.



Gambar 6.6 : Pemodelan operasi `simpanJawaban(db, smarty, ps)` ke dalam *flow graph*

Sumber : [Pengujian]

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* `simpanJawaban`. Jumlah kompleksitas siklomatis (*cyclomatic complexity*)

melalui persamaan $V(G) = E - N + 2$.

$$\begin{aligned} V(G) &= 9 - 8 + 2 \\ &= 3 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu 2 ditentukan dua buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-3-4-5-7-2-...-7-2-8

Jalur 2 : 1-2-3-4-6-7-2-...-7-2-8

Jalur 3 : 1-2-8

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah dijelaskan pada Tabel 6.4.

Tabel 6.4 *Test case* untuk pengujian integrasi *method* simpanJawaban

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Nilai = 1	Jawaban tersimpan dengan nilai 1	Jawaban tersimpan dengan nilai 1
2	Nilai = 0	Jawaban tersimpan dengan nilai 0	Jawaban tersimpan dengan nilai 0
3	idSoal = 0	Tidak ada soal yang terjawab, nilai 0	Tidak ada soal yang terjawab, nilai 0

Sumber: [Pengujian]

Berdasar tabel 6.4. didapat bahwa semua jalur tidak memiliki kesalahan dan mendapatkan hasil sesuai dengan yang diharapkan.

6.3 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak memerlukan untuk berkonsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan.

6.3.1 Kasus Uji Validasi

Untuk mengetahui kesesuaian antara kebutuhan dengan kinerja sistem, pada setiap kebutuhan (*requirement*) dilakukan proses pengujian dengan kasus uji masing-masing.

6.3.1.1 Kasus Uji Login

Tabel 6.5a dan tabel 6.5b Menunjukkan Kasus Uji *login*.

Tabel 6.5a : Kasus uji *login*

Nama kasus uji	Kasus uji <i>login</i>
Obyek uji	Kebutuhan nomer 1 (aliran utama)
Tujuan pengujian	Menguji bahwa hanya pengguna tertentu yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu(sebagai suatu aktor tertentu). Pengguna sistem yang tidak login hanya akan menjadi aktor user.
Prosedur uji	<ol style="list-style-type: none">1. Sistem telah berjalan.2. Memasukkan <i>username</i> dan <i>password</i> pada kotak isian (<i>username</i>="admin", <i>password</i> = "admin")3. Menekan tombol login.
Hasil yang diharapkan	Pengguna berhasil login pada sistem.

Sumber : [Pengujian]

Tabel 6.5b : Kasus uji *login* dimana pasangan *username* dan *password* tidak ada dalam sistem

Nama kasus uji	Kasus uji <i>login</i> dimana pasangan <i>username</i> dan <i>password</i> tidak ada dalam sistem
Obyek uji	Kebutuhan nomer 1 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>username</i> atau <i>password</i> tidak ada.
Prosedur uji	<ol style="list-style-type: none">1. Sistem telah berjalan.2. Memasukkan <i>username</i> dan <i>password</i> pada kotak isian pada pojok atas kanan yang tidak ada dalam sistem (<i>username</i>="zz", <i>password</i> = "zz").3. Menekan tombol login.
Hasil yang diharapkan	Menampilkan pesan bahwa <i>username</i> atau <i>password</i> tidak ada.

Sumber : [Pengujian]

6.3.1.2 Kasus Uji Logout

Tabel 6.6 menunjukkan kasus uji *logout*

Tabel 6.6 : Kasus uji *logout*

Nama kasus uji	Kasus uji <i>logout</i>
Obyek uji	Kebutuhan nomer 2
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas agar pengguna yang telah login dapat keluar dari sistem.
Prosedur uji	<ol style="list-style-type: none">1. Sistem telah berjalan dan pengguna telah <i>login</i>.2. Pengguna menekan link <i>logout</i>.
Hasil yang diharapkan	Pengguna berhasil <i>logout</i> dari sistem.

Sumber : [Pengujian]

6.3.1.3 Kasus Uji Menambah Data Petugas

Tabel 6.7a, tabel 6.7b dan tabel 6.7c menunjukkan kasus uji menambah data petugas.

Tabel 6.7a : Kasus uji menambah data petugas

Nama kasus uji	Kasus uji menambah data petugas
Obyek uji	Kebutuhan nomer 3 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah petugas. Data yang harus dicatat dari penambahan petugas adalah nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar petugas telah dijalankan 2. Memilih <i>link</i> tambah petugas. 3. Memasukkan data-data pada <i>form</i> (nama="operator", username = "operator", password = "operator"). 4. Menekan tombol tambah.
Hasil yang diharapkan	Data petugas berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.7b : Kasus uji menambah data petugas dimana masukan nama petugas, *username* atau *password* kosong

Nama kasus uji	Kasus uji menambah data petugas dimana masukan nip, nama petugas, username atau password kosong.
Obyek uji	Kebutuhan nomer 3 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nip atau nama petugas atau <i>username</i> atau <i>password</i> tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar petugas telah dijalankan 2. Memilih <i>tab</i> tambah petugas. 3. Memasukkan data-data pada <i>form</i> (nama="operator"). 4. Menekan tombol tambah.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nip nama petugas atau <i>username</i> atau <i>password</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.7c : Kasus uji menambah data petugas dimana masukan username telah ada di dalam sistem

Nama kasus uji	Kasus uji menambah data petugas dimana masukan username telah ada di dalam sistem.
Obyek uji	Kebutuhan nomer 3 (aliran alternatif 2).
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan kesalahan bahwa <i>username</i> tersebut telah ada di dalam

	sistem bila <i>username</i> tersebut memang telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar petugas telah dijalankan 2. Memilih <i>link</i> tambah petugas. 3. Memasukkan data-data pada <i>form</i> (nama="operator", <i>username</i>=" operator" dan <i>password</i> = "operator", dimana telah ada petugas dengan <i>username</i> "operator"). 4. Menekan tombol tambah.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa <i>username</i> tersebut telah ada di dalam sistem.

Sumber : [Pengujian]

6.3.1.4 Kasus Uji Melihat Daftar Petugas

Tabel 6.8 menunjukkan kasus uji melihat daftar petugas.

Tabel 6.8 : Kasus uji melihat daftar petugas

Nama kasus uji	Kasus uji melihat daftar petugas
Obyek uji	Kebutuhan nomer 4 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua petugas yang tercatat di dalam sistem. Data yang ditampilkan berupa nip, nama, jenis kelamin, telepon, <i>expired</i> , dan jenis petugas.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu petugas.
Hasil yang diharapkan	Semua daftar petugas dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.5 Kasus Uji Mengedit Data Petugas

Tabel 6.9a, tabel 6.9b dan tabel 6.9c menunjukkan kasus uji mengedit data petugas.

Tabel 6.9a: Kasus uji mengedit data petugas

Nama kasus uji	Kasus uji mengedit data petugas
Obyek uji	Kebutuhan nomer 5 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data petugas yang telah dimasukkan. Data yang dapat diedit adalah nip, nama, alamat, kota, propinsi, jenis kelamin, tanggal lahir, tempat lahir, telepon, email, <i>expired</i> , <i>username</i> , <i>password</i> , jenis petugas, agama, dan keterangan.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar petugas telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> (nama = "coba edit", <i>username</i> = "coba edit", <i>password</i> = "coba edit"). 4. Menekan tombol simpan.

Hasil yang diharapkan	Data petugas dapat diedit dan disimpan.
-----------------------	---

Sumber : [Pengujian]

Tabel 6.9b: Kasus uji mengedit data petugas dimana masukan nip atau nama petugas atau *username* atau *password* kosong

Nama kasus uji	Kasus uji mengedit data petugas dimana masukan nip atau nama petugas atau <i>username</i> atau <i>password</i> kosong.
Obyek uji	Kebutuhan nomer 5 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa data nip atau nama atau <i>username</i> atau <i>password</i> tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar petugas telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> (data nama kosong, <i>username</i> kosong, <i>password</i> kosong). 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa data nip atau nama atau <i>username</i> atau <i>password</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.9c: Kasus uji mengedit data petugas dimana *username* petugas telah ada di dalam sistem

Nama kasus uji	Kasus uji mengedit data petugas dimana <i>username</i> petugas telah ada di dalam sistem
Obyek uji	Kebutuhan nomer 5 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa suatu data <i>username</i> telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar petugas telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> (data nama kosong = “coba”, <i>username</i> = “admin”, <i>password</i> kosong, dimana telah ada <i>username</i> = “administrator” di dalam sistem). 4. Menekan tombol <i>simpan</i>.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa data <i>username</i> telah ada dalam sistem.

Sumber : [Pengujian]

6.3.1.6 Kasus Uji Menghapus Data Petugas

Tabel 6.10 menunjukkan kasus uji menghapus data petugas.

Tabel 6.10 : Kasus uji menghapus data petugas

Nama kasus uji	Kasus uji menghapus data petugas
Obyek uji	Kebutuhan nomer 6 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus data petugas tertentu.

Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar petugas telah dijalankan 2. Memilih tombol <i>delete</i> pada salah satu baris dimana petugas tersebut tidak direferensi oleh data lain.
Hasil yang diharapkan	Data petugas dapat dihapus.

Sumber : [Pengujian]

6.3.1.7 Kasus Uji Mencari Petugas

Tabel 6.11a, tabel 6.11b dan tabel 6.11c menunjukkan kasus uji mencari petugas.

Tabel 6.11a : Kasus uji mencari petugas

Nama kasus uji	Kasus uji mencari petugas
Obyek uji	Kebutuhan nomer 7 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu petugas telah ada di dalam sistem. Pencarian data petugas dapat berdasarkan pada nip atau nama petugas.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih menu petugas. 3. Memasukkan <i>keyword</i>. 4. Menekan tombol <i>find</i>.
Hasil yang diharapkan	Suatu koleksi berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.11b : Kasus uji mencari petugas dimana masukan keyword kosong

Nama kasus uji	Kasus uji mencari petugas dimana masukan keyword kosong
Obyek uji	Kebutuhan nomer 7 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih menu petugas. 3. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.11c : Kasus uji mencari petugas dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari petugas dan menekan tombol refresh
Obyek uji	Kebutuhan nomer 7 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar petugas yang ditampilkan.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih menu petugas. 3. Memasukkan <i>keyword</i>.

	<ol style="list-style-type: none"> 4. Menekan tombol <i>find</i>. 5. Menekan tombol <i>refresh</i> data petugas.
Hasil yang diharapkan	Sistem menampilkan daftar petugas yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.8 Kasus Uji Menambah Data Instansi *Client*

Tabel 6.12a, tabel 6.12b dan tabel 6.12c menunjukkan kasus uji menambah data instansi *client*.

Tabel 6.12a : Kasus uji menambah data instansi *client*

Nama kasus uji	Kasus uji menambah data instansi <i>client</i>
Obyek uji	Kebutuhan nomer 8 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah data instansi <i>client</i> . Data yang harus dicatat dari penambahan instansi <i>client</i> adalah nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, email, fax.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar instansi <i>client</i> telah dijalankan 2. Memilih <i>link</i> tambah petugas. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol tambah.
Hasil yang diharapkan	Data instansi <i>client</i> berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.12b : Kasus uji menambah data instansi *client* dimana masukan nama instansi, alamat, bidang usaha, *contact person*, dan telepon kosong

Nama kasus uji	Kasus uji menambah data instansi <i>client</i> dimana masukan nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon kosong.
Obyek uji	Kebutuhan nomer 8 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar instansi <i>client</i> telah dijalankan 2. Memilih <i>tab</i> tambah intansi <i>client</i>. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.12c : Kasus uji menambah data instansi *client* dimana masukan nama dan alamat telah ada di dalam sistem

Nama kasus uji	Kasus uji menambah data instansi <i>client</i> dimana
----------------	---

	masukan nama dan alamat telah ada di dalam sistem.
Obyek uji	Kebutuhan nomer 8 (aliran alternatif 2).
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan kesalahan bahwa nama dan alamat tersebut telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar instansi <i>client</i> telah dijalankan 2. Memilih <i>link</i> tambah instansi <i>client</i>. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nama dan alamat tersebut telah ada di dalam sistem.

Sumber : [Pengujian]

6.3.1.9 Kasus Uji Melihat Daftar Instansi *Client*

Tabel 6.13 menunjukkan kasus uji melihat daftar instansi *client*.

Tabel 6.13 : Kasus uji melihat daftar instansi *client*

Nama kasus uji	Kasus uji melihat daftar instansi <i>client</i>
Obyek uji	Kebutuhan nomer 9 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua instansi <i>client</i> yang tercatat di dalam sistem. Data yang ditampilkan berupa nama instansi, alamat, bidang usaha, contact person, telepon, fax, dan website.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu instansi <i>client</i>.
Hasil yang diharapkan	Semua daftar instansi <i>client</i> dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.10 Kasus Uji Mengedit Data Instansi *Client*

Tabel 6.14a, tabel 6.14b dan tabel 6.14c menunjukkan kasus uji mengedit data instansi *client*.

Tabel 6.14a: Kasus uji mengedit data instansi *client*

Nama kasus uji	Kasus uji mengedit data instansi <i>client</i>
Obyek uji	Kebutuhan nomer 10 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data instansi <i>client</i> yang telah dimasukkan. Data yang dapat diedit adalah nama instansi, alamat, bidang usaha, <i>contact person</i> , telepon, fax, dan <i>website</i>
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar instansi <i>client</i> telah dijalankan 2. Memilih <i>link edit</i> pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol Simpan.
Hasil yang diharapkan	Data instansi <i>client</i> dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.14b: Kasus uji mengedit data instansi *client* dimana masukan nama instansi, alamat, *contact person*, telepon kosong

Nama kasus uji	Kasus uji mengedit data instansi <i>client</i> dimana masukan nama instansi, alamat, <i>contact person</i> , telepon kosong.
Obyek uji	Kebutuhan nomer 10 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa data nama instansi, alamat, <i>contact person</i> , telepon tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar instansi <i>client</i> telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa data nama instansi, alamat, <i>contact person</i> , telepon tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.14c: Kasus uji mengedit data instansi *client* dimana nama dan alamat instansi *client* telah ada di dalam sistem

Nama kasus uji	Kasus uji mengedit data instansi <i>client</i> dimana nama dan alamat instansi <i>client</i> telah ada di dalam sistem
Obyek uji	Kebutuhan nomer 10 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa suatu data nama dan alamat instansi <i>client</i> telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar instansi <i>client</i> telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol <i>simpan</i>.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa data nama dan alamat instansi <i>client</i> telah ada dalam sistem.

Sumber : [Pengujian]

6.3.1.11 Kasus Uji Menghapus Data Instansi *Client*

Tabel 6.15 menunjukkan kasus uji menghapus data instansi *client*.

Tabel 6.15 : Kasus uji menghapus data instansi *client*

Nama kasus uji	Kasus uji menghapus data instansi <i>client</i>
Obyek uji	Kebutuhan nomer 11 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus data instansi <i>client</i> tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar instansi <i>client</i> telah dijalankan

	2. Memilih tombol <i>delete</i> pada salah satu baris instansi <i>client</i> .
Hasil yang diharapkan	Data instansi <i>client</i> dapat dihapus.

Sumber : [Pengujian]

6.3.1.12 Kasus Uji Mencari Instansi *Client*

Tabel 6.16a, tabel 6.16b dan tabel 6.16c menunjukkan kasus uji mencari instansi *client*.

Tabel 6.16a : Kasus uji mencari instansi *client*

Nama kasus uji	Kasus uji mencari instansi <i>client</i>
Obyek uji	Kebutuhan nomer 12 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu instansi <i>client</i> telah ada di dalam sistem. Pencarian data instansi <i>client</i> dapat berdasarkan pada nama instansi.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih menu instansi <i>client</i>. 3. Memasukkan <i>keyword</i>. 4. Menekan tombol <i>find</i>.
Hasil yang diharapkan	Suatu instansi <i>client</i> berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.16b : Kasus uji mencari instansi *client* dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari instansi <i>client</i> dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 12 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih menu instansi <i>client</i>. 3. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.16c : Kasus uji mencari instansi *client* dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari instansi <i>client</i> dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 12 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar instansi <i>client</i> yang ditampilkan.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih menu instansi <i>client</i>. 3. Memasukkan <i>keyword</i>. 4. Menekan tombol <i>find</i>. 5. Menekan tombol <i>refresh</i> data instansi <i>client</i>.

Hasil yang diharapkan	Sistem menampilkan daftar instansi <i>client</i> yang telah di- <i>refresh</i> .
-----------------------	--

Sumber : [Pengujian]

6.3.1.13 Kasus Uji Menambah Kategori Soal

Tabel 6.17a dan tabel 6.17b menunjukkan kasus uji menambah kategori soal.

Tabel 6.17a : Kasus uji menambah kategori soal

Nama kasus uji	Kasus uji menambah kategori soal
Obyek uji	Kebutuhan nomer 13 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah kategori soal. Data yang harus dicatat dari penambahan kategori soal adalah nama kategori, tujuan, keterangan.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar kategori soal telah dijalankan 2. Memilih <i>link</i> tambah kategori soal. 3. Memasukkan data-data pada <i>form</i>. 4. Menekan tombol tambah.
Hasil yang diharapkan	Kategori soal berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.17b : Kasus uji menambah kategori soal dimana masukan nama kategori kosong

Nama kasus uji	Kasus uji menambah kategori soal dimana masukan nama kategori kosong.
Obyek uji	Kebutuhan nomer13 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nama kategori tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar kategori soal telah dijalankan 2. Memilih <i>tab</i> tambah kategori soal. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan nama kategori tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.14 Kasus Uji Melihat Daftar Kategori Soal

Tabel 6.18 menunjukkan kasus uji melihat daftar kategori soal.

Tabel 6.18 : Kasus uji melihat daftar kategori soal

Nama kasus uji	Kasus uji melihat daftar kategori soal
Obyek uji	Kebutuhan nomer 14 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua kategori soal yang tercatat di dalam sistem. Data yang ditampilkan berupa nama kategori, tujuan, keterangan.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login

	sebagai administrator. 2. Memilih menu kategori soal.
Hasil yang diharapkan	Semua daftar kategori soal dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.15 Kasus Uji Mengedit Kategori Soal

Tabel 6.19a, tabel 6.19b dan tabel 6.19c menunjukkan kasus uji mengedit kategori soal.

Tabel 6.19a: Kasus uji mengedit kategori soal

Nama kasus uji	Kasus uji mengedit data kategori soal
Obyek uji	Kebutuhan nomer 15 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data kategori soal yang telah dimasukkan. Data yang dapat diedit adalah nama kategori, tujuan, keterangan.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar kategori soal telah dijalankan 2. Memilih <i>link edit</i> pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol Simpan.
Hasil yang diharapkan	Data kategori soal dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.19b: Kasus uji mengedit kategori soal dimana masukan nama kategori kosong

Nama kasus uji	Kasus uji mengedit kategori soal dimana masukan nama kategori kosong.
Obyek uji	Kebutuhan nomer 15 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa nama kategori tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar kategori soal telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa nama kategori tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.19c: Kasus uji mengedit data kategori soal dimana nama kategori telah ada di dalam sistem

Nama kasus uji	Kasus uji mengedit kategori soal dimana nama kategori telah ada di dalam sistem
Obyek uji	Kebutuhan nomer 15 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa suatu data nama kategori telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar kategori soal telah dijalankan 2. Memilih tombol edit pada salah satu baris.

	3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol <i>simpan</i> .
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa data nama kategori telah ada dalam sistem.

Sumber : [Pengujian]

6.3.1.16 Kasus Uji Menghapus Kategori Soal

Tabel 6.20 menunjukkan kasus uji menghapus kategori soal.

Tabel 6.20 : Kasus uji menghapus kategori soal

Nama kasus uji	Kasus uji menghapus kategori soal
Obyek uji	Kebutuhan nomer 16 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus kategori soal tertentu.
Prosedur uji	1. Use case melihat daftar kategori soal telah dijalankan 2. Memilih tombol <i>delete</i> pada salah satu baris kategori soal.
Hasil yang diharapkan	Data kategori soal dapat dihapus.

Sumber : [Pengujian]

6.3.1.17 Kasus Uji Mencari Kategori Soal

Tabel 6.21a, tabel 6.21b dan tabel 6.21c menunjukkan kasus uji mencari kategori soal.

Tabel 6.21a : Kasus uji mencari kategori soal

Nama kasus uji	Kasus uji mencari kategori soal
Obyek uji	Kebutuhan nomer 17 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu kategori soal telah ada di dalam sistem. Pencarian data kategori soal dapat berdasarkan pada nama kategori.
Prosedur uji	5. Sistem telah berjalan. 6. Memilih menu kategori soal. 7. Memasukkan <i>keyword</i> . 8. Menekan tombol <i>find</i> .
Hasil yang diharapkan	Suatu kategori soal berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.21b : Kasus uji mencari kategori soal dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari kategori soal dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 17 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	4. Sistem telah berjalan. 5. Memilih menu kategori soal.

	6. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.21c : Kasus uji mencari kategori soal dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari kategori soal dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 17 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar kategori soal yang ditampilkan.
Prosedur uji	6. Sistem telah berjalan. 7. Memilih menu pengaturan soal, kategori soal. 8. Memasukkan <i>keyword</i> . 9. Menekan tombol <i>find</i> . 10. Menekan tombol <i>refresh</i> kategori soal.
Hasil yang diharapkan	Sistem menampilkan daftar kategori soal yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.18 Kasus Uji Menambah Urutan Kategori Soal

Tabel 6.22a dan tabel 6.22b menunjukkan kasus uji menambah urutan kategori soal.

Tabel 6.22a : Kasus uji menambah urutan kategori soal

Nama kasus uji	Kasus uji menambah urutan kategori soal
Obyek uji	Kebutuhan nomer 18 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah urutan kategori soal. Data yang harus dicatat dari penambahan urutan kategori soal adalah lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.
Prosedur uji	1. Use case melihat daftar urutan kategori soal telah dijalankan 2. Memilih <i>link</i> tambah urutan kategori soal. 3. Memasukkan data-data pada <i>form</i> . 4. Menekan tombol tambah.
Hasil yang diharapkan	Urutan kategori soal berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.22b : Kasus uji menambah urutan kategori soal dimana masukan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong

Nama kasus uji	Kasus uji menambah urutan kategori soal dimana masukan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong.
Obyek uji	Kebutuhan nomer 18 (aliran alternatif 1).

Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar urutan kategori soal telah dijalankan 2. Memilih <i>tab</i> tambah urutan kategori soal. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.19 Kasus Uji Mengedit Urutan Kategori Soal

Tabel 6.23a dan tabel 6.23b menunjukkan kasus uji mengedit urutan kategori soal.

Tabel 6.23a: Kasus uji mengedit urutan kategori soal

Nama kasus uji	Kasus uji mengedit data urutan kategori soal
Obyek uji	Kebutuhan nomer 19 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data urutan kategori soal yang telah dimasukkan. Data yang dapat diedit adalah lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar urutan kategori soal telah dijalankan 2. Memilih <i>link edit</i> pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol Simpan.
Hasil yang diharapkan	Data urutan kategori soal dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.23b: Kasus uji mengedit urutan kategori soal dimana masukan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong

Nama kasus uji	Kasus uji mengedit urutan kategori soal dimana masukan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong.
Obyek uji	Kebutuhan nomer 19 (aliran ystem tive 1)
Tujuan pengujian	Menguji bahwa ystem dapat menampilkan bahwa lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar urutan kategori soal telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.

Hasil yang diharapkan	Sistem menampilkan peringatan bahwa lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.
-----------------------	---

Sumber : [Pengujian]

6.3.1.20 Kasus Uji Menghapus Urutan Kategori Soal

Tabel 6.24 menunjukkan kasus uji menghapus urutan kategori soal.

Tabel 6.24 : Kasus uji menghapus urutan kategori soal

Nama kasus uji	Kasus uji menghapus urutan kategori soal
Obyek uji	Kebutuhan nomer 20 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus urutan kategori soal tertentu.
Prosedur uji	1. Use case melihat daftar urutan kategori soal telah dijalankan 2. Memilih tombol <i>delete</i> pada salah satu baris urutan kategori soal.
Hasil yang diharapkan	Data urutan kategori soal dapat dihapus.

Sumber : [Pengujian]

6.3.1.21 Kasus Uji Mencari Urutan Kategori Soal

Tabel 6.25a, tabel 6.25b dan tabel 6.25c menunjukkan kasus uji mencari urutan kategori soal.

Tabel 6.25a : Kasus uji mencari urutan kategori soal

Nama kasus uji	Kasus uji mencari urutan kategori soal
Obyek uji	Kebutuhan nomer 21 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu kategori soal telah ada di dalam sistem. Pencarian data kategori soal dapat berdasarkan pada nama instansi.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih menu urutan kategori soal. 3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> .
Hasil yang diharapkan	Suatu urutan kategori soal berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.25b : Kasus uji mencari urutan kategori soal dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari urutan kategori soal dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 21 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih menu urutan kategori soal.

	3. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.25c : Kasus uji mencari urutan kategori soal dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari urutan kategori soal dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 21 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar urutan kategori soal yang ditampilkan.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih menu pengaturan soal, urutan kategori soal. 3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> . 5. Menekan tombol <i>refresh</i> urutan kategori soal.
Hasil yang diharapkan	Sistem menampilkan daftar urutan kategori soal yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.22 Kasus Uji Melihat Daftar Urutan Kategori Soal

Tabel 6.26 menunjukkan kasus uji melihat daftar urutan kategori soal.

Tabel 6.26 : Kasus uji melihat daftar urutan kategori soal

Nama kasus uji	Kasus uji melihat daftar urutan kategori soal
Obyek uji	Kebutuhan nomer 21 (aliran utama)
Tujuan pengujian	Menguji bahwa system dapat menampilkan semua urutan kategori soal yang tercatat di dalam sistem. Data yang ditampilkan berupa lokasi tes, waktu, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan.
Prosedur uji	1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu pengaturan soal, data urutan kategori soal.
Hasil yang diharapkan	Semua daftar urutan kategori soal dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.23 Kasus Uji Menambah Data Soal Psikotes

Tabel 6.27a dan tabel 6.27b menunjukkan kasus uji menambah data soal psikotes.

Tabel 6.27a : Kasus uji menambah data soal psikotes

Nama kasus uji	Kasus uji menambah data soal psikotes
Obyek uji	Kebutuhan nomer 22 (aliran utama)
Tujuan pengujian	Menguji bahwa system dapat menyediakan fasilitas

	untuk menambah data soal psikotes. Data yang harus dicatat dari penambahan data soal psikotes adalah soal.
Prosedur uji	1. Use case melihat daftar soal telah dijalankan 2. Memilih <i>link</i> tambah soal. 3. Memasukkan data-data pada <i>form</i> . 4. Menekan tombol tambah.
Hasil yang diharapkan	Soal berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.27b : Kasus uji menambah soal dimana masukan soal kosong

Nama kasus uji	Kasus uji menambah soal dimana masukan soal kosong.
Obyek uji	Kebutuhan nomer 22 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa soal tidak boleh kosong.
Prosedur uji	1. Use case melihat daftar soal telah dijalankan 2. Memilih <i>tab</i> tambah soal. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan soal tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.24 Kasus Uji Melihat Daftar Soal

Tabel 6.28 menunjukkan kasus uji melihat daftar soal.

Tabel 6.28 : Kasus uji melihat daftar soal

Nama kasus uji	Kasus uji melihat daftar soal
Obyek uji	Kebutuhan nomer 23 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua soal yang tercatat di dalam sistem. Data yang ditampilkan berupa soal.
Prosedur uji	1.Sistem telah berjalan dan pengguna telah login sebagai administrator. 2.Memilih <i>link</i> soal.
Hasil yang diharapkan	Semua daftar soal dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.25 Kasus Uji Mengedit Data Soal

Tabel 6.29a dan tabel 6.29b menunjukkan kasus uji mengedit data soal.

Tabel 6.29a: Kasus uji mengedit data soal

Nama kasus uji	Kasus uji mengedit data soal
Obyek uji	Kebutuhan nomer 24 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data soal yang telah dimasukkan. Data yang dapat diedit adalah soal.
Prosedur uji	1.Use case melihat daftar soal telah dijalankan 2.Memilih <i>link edit</i> pada salah satu baris.

	3.Memasukkan data-data pada <i>form</i> 4.Menekan tombol Simpan.
Hasil yang diharapkan	Data soal dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.29b: Kasus uji mengedit soal dimana masukan soal kosong

Nama kasus uji	Kasus uji mengedit soal dimana masukan soal kosong.
Obyek uji	Kebutuhan nomer 24 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa soal tidak boleh kosong.
Prosedur uji	1. Use case melihat daftar soal telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa soal tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.26 Kasus Uji Menghapus Data Soal

Tabel 6.30 menunjukkan kasus uji menghapus data soal.

Tabel 6.30 : Kasus uji menghapus data soal

Nama kasus uji	Kasus uji menghapus data soal
Obyek uji	Kebutuhan nomer 25 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus data soal tertentu.
Prosedur uji	1. Use case melihat daftar soal telah dijalankan 2. Memilih tombol <i>delete</i> pada salah satu baris soal.
Hasil yang diharapkan	Data soal dapat dihapus.

Sumber : [Pengujian]

6.3.1.27 Kasus Uji Mencari Data Soal

Tabel 6.31a, tabel 6.31b dan tabel 6.31c menunjukkan kasus uji mencari soal.

Tabel 6.31a : Kasus uji mencari data soal

Nama kasus uji	Kasus uji mencari data soal
Obyek uji	Kebutuhan nomer 26 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data soal telah ada di dalam sistem.Pencarian data soal dapat berdasarkan potongan kalimat soal.
Prosedur uji	1.Sistem telah berjalan. 2.Memilih menu soal. 3.Memasukkan <i>keyword</i> . 4.Menekan tombol <i>find</i> .
Hasil yang diharapkan	Suatu soal berhasil ditampilkan berdasarkan pada

	<i>keyword</i> tertentu.
--	--------------------------

Sumber : [Pengujian]

Tabel 6.31b : Kasus uji mencari soal dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari soal dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 26 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	4. Sistem telah berjalan. 5. Memilih <i>link</i> soal. 6. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.31c : Kasus uji mencari soal dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari soal dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 26 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar soal yang ditampilkan.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih menu soal. 3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> . 5. Menekan tombol <i>refresh</i> soal.
Hasil yang diharapkan	Sistem menampilkan daftar soal yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.28 Kasus Uji Menjawab Soal Psikotes

Tabel 6.32 menunjukkan kasus uji menjawab soal psikotes.

Tabel 6.32 : Kasus uji menjawab soal psikotes

Nama kasus uji	Kasus uji menjawab soal psikotes
Obyek uji	Kebutuhan nomer 27 (aliran utama)
Tujuan pengujian	Menguji bahwa system dapat menyediakan fasilitas untuk menjawab soal psikotes. Data yang harus disimpan dari menjawab soal psikotes adalah jawaban.
Prosedur uji	1. Memilih <i>button</i> mulai. 2. Menjawab soal pada <i>form</i> . 3. Menekan tombol simpan.
Hasil yang diharapkan	Soal berhasil dijawab dan jawaban berhasil disimpan.

Sumber : [Pengujian]

6.3.1.29 Kasus Uji Menambah Sub Kategori Soal

Tabel 6.33a dan tabel 6.33b menunjukkan kasus uji menambah sub

kategori soal.

Tabel 6.33a : Kasus uji menambah sub kategori soal

Nama kasus uji	Kasus uji menambah sub kategori soal
Obyek uji	Kebutuhan nomer 28 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah sub kategori soal. Data yang harus dicatat dari penambahan sub kategori soal adalah nama sub kategori.
Prosedur uji	1. Use case melihat daftar sub kategori soal telah dijalankan 2. Memilih <i>link</i> tambah sub kategori soal. 3. Memasukkan data-data pada <i>form</i> . 4. Menekan tombol tambah.
Hasil yang diharapkan	Sub kategori soal berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.33b : Kasus uji menambah sub kategori soal dimana masukan nama sub kategori kosong

Nama kasus uji	Kasus uji menambah sub kategori soal dimana masukan sub kategori kosong.
Obyek uji	Kebutuhan nomer 28 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa sub kategori tidak boleh kosong.
Prosedur uji	1. Use case melihat daftar sub kategori soal telah dijalankan 2. Memilih <i>tab</i> tambah sub kategori soal. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan nama sub kategori tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.30 Kasus Uji Melihat Daftar Sub Kategori Soal

Tabel 6.34 menunjukkan kasus uji melihat daftar sub kategori soal.

Tabel 6.34 : Kasus uji melihat daftar sub kategori soal

Nama kasus uji	Kasus uji melihat daftar sub kategori soal
Obyek uji	Kebutuhan nomer 29 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua sub kategori soal yang tercatat di dalam sistem. Data yang ditampilkan berupa sub kategori.
Prosedur uji	1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu sub kategori soal.
Hasil yang diharapkan	Semua daftar sub kategori soal dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.31 Kasus Uji Mengedit Sub Kategori Soal

Tabel 6.35a, tabel 6.35b dan tabel 6.35c menunjukkan kasus uji mengedit sub kategori soal.

Tabel 6.35a: Kasus uji mengedit sub kategori soal

Nama kasus uji	Kasus uji mengedit data sub kategori soal
Obyek uji	Kebutuhan nomer 30 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data sub kategori soal yang telah dimasukkan. Data yang dapat diedit adalah nama sub kategori.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar sub kategori soal telah dijalankan 2. Memilih <i>link edit</i> pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol Simpan.
Hasil yang diharapkan	Data sub kategori soal dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.35b: Kasus uji mengedit sub kategori soal dimana masukan nama sub kategori kosong

Nama kasus uji	Kasus uji mengedit sub kategori soal dimana masukan nama sub kategori kosong.
Obyek uji	Kebutuhan nomer 30 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa sub kategori tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 5. Use case melihat daftar sub kategori soal telah dijalankan 6. Memilih tombol edit pada salah satu baris. 7. Memasukkan data-data pada <i>form</i> 8. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa sub kategori tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.35c: Kasus uji mengedit data sub kategori soal dimana sub kategori telah ada di dalam sistem

Nama kasus uji	Kasus uji mengedit sub kategori soal dimana sub kategori telah ada di dalam sistem
Obyek uji	Kebutuhan nomer 30 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa suatu data sub kategori telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar sub kategori soal telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol <i>simpan</i>.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa data sub

	kategori telah ada dalam sistem.
--	----------------------------------

Sumber : [Pengujian]

6.3.1.32 Kasus Uji Menghapus Sub Kategori Soal

Tabel 6.36 menunjukkan kasus uji menghapus sub kategori soal.

Tabel 6.36 : Kasus uji menghapus sub kategori soal

Nama kasus uji	Kasus uji menghapus sub kategori soal
Obyek uji	Kebutuhan nomer 31 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus sub kategori soal tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar sub kategori soal telah dijalankan 2. Memilih tombol <i>delete</i> pada salah satu baris sub kategori soal.
Hasil yang diharapkan	Data sub kategori soal dapat dihapus.

Sumber : [Pengujian]

6.3.1.33 Kasus Uji Mencari Sub Kategori Soal

Tabel 6.37a, tabel 6.37b dan tabel 6.37c menunjukkan kasus uji mencari sub kategori soal.

Tabel 6.37a : Kasus uji mencari sub kategori soal

Nama kasus uji	Kasus uji mencari sub kategori soal
Obyek uji	Kebutuhan nomer 32 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu sub kategori soal telah ada di dalam sistem. Pencarian data sub kategori soal dapat berdasarkan pada sub kategori.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih menu sub kategori soal. 3. Memasukkan <i>keyword</i>. 4. Menekan tombol <i>find</i>.
Hasil yang diharapkan	Suatu sub kategori soal berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.37b : Kasus uji mencari sub kategori soal dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari sub kategori soal dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 33 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih <i>link</i> sub kategori soal. 3. Menekan tombol <i>find (keyword</i> kosong).

Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.
-----------------------	--

Sumber : [Pengujian]

Tabel 6.37c : Kasus uji mencari sub kategori soal dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari sub kategori soal dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 34 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar sub kategori soal yang ditampilkan.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan. 2. Memilih <i>link</i> sub kategori soal. 3. Memasukkan <i>keyword</i>. 4. Menekan tombol <i>find</i>. 5. Menekan tombol <i>refresh</i> sub kategori soal.
Hasil yang diharapkan	Sistem menampilkan daftar sub kategori soal yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.34 Kasus Uji Menambah Pilihan Jawaban Soal

Tabel 6.38a dan tabel 6.38b menunjukkan kasus uji menambah pilihan jawaban soal.

Tabel 6.38a : Kasus uji menambah pilihan jawaban soal

Nama kasus uji	Kasus uji menambah pilihan jawaban soal
Obyek uji	Kebutuhan nomer 35 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah pilihan jawaban soal. Data yang harus dicatat dari penambahan pilihan jawaban soal adalah huruf pilihan, jawaban, dan status jawaban.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar pilihan jawaban soal telah dijalankan 2. Memilih <i>link</i> tambah pilihan jawaban. 3. Memasukkan data-data pada <i>form</i>. 4. Menekan tombol tambah.
Hasil yang diharapkan	Pilihan jawaban soal berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.38b : Kasus uji menambah pilihan jawaban soal dimana masukan huruf pilihan, jawaban, dan status jawaban kosong

Nama kasus uji	Kasus uji menambah pilihan jawaban soal dimana masukan huruf pilihan, jawaban, dan status jawaban kosong.
Obyek uji	Kebutuhan nomer 35 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.

Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar pilihan jawaban soal telah dijalankan 2. Memilih <i>tab</i> tambah pilihan jawaban. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.35 Kasus Uji Melihat Daftar Pilihan Jawaban

Tabel 6.39 menunjukkan kasus uji melihat daftar pilihan jawaban.

Tabel 6.39 : Kasus uji melihat daftar pilihan jawaban

Nama kasus uji	Kasus uji melihat daftar pilihan jawaban
Obyek uji	Kebutuhan nomer 36 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua pilihan jawaban yang tercatat di dalam sistem. Data yang ditampilkan berupa huruf pilihan, jawaban, dan status jawaban.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu pilihan jawaban.
Hasil yang diharapkan	Semua daftar pilihan jawaban dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.36 Kasus Uji Mengedit Pilihan Jawaban

Tabel 6.40a, tabel 6.40b dan tabel 6.40c menunjukkan kasus uji mengedit pilihan jawaban.

Tabel 6.40a: Kasus uji mengedit pilihan jawaban

Nama kasus uji	Kasus uji mengedit pilihan jawaban
Obyek uji	Kebutuhan nomer 37 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit pilihan jawaban yang telah dimasukkan. Data yang dapat diedit adalah huruf pilihan, jawaban, dan status jawaban.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar pilihan jawaban telah dijalankan 2. Memilih <i>link edit</i> pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol Simpan.
Hasil yang diharapkan	Data pilihan jawaban dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.40b: Kasus uji mengedit pilihan jawaban dimana masukan huruf pilihan, jawaban, dan status jawaban kosong

Nama kasus uji	Kasus uji mengedit pilihan jawaban dimana masukan
----------------	---

	huruf pilihan, jawaban, dan status jawaban kosong.
Obyek uji	Kebutuhan nomer 37 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar pilihan jawaban telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.40c: Kasus uji mengedit pilihan jawaban dimana huruf pilihan telah ada di dalam sistem

Nama kasus uji	Kasus uji mengedit pilihan jawaban dimana huruf pilihan telah ada di dalam sistem
Obyek uji	Kebutuhan nomer 37 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa suatu huruf pilihan telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar pilihan jawaban telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol <i>simpan</i>.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa huruf pilihan telah ada dalam sistem.

Sumber : [Pengujian]

6.3.1.37 Kasus Uji Menghapus Pilihan Jawaban

Tabel 6.41 menunjukkan kasus uji menghapus pilihan jawaban.

Tabel 6.41 : Kasus uji menghapus pilihan jawaban

Nama kasus uji	Kasus uji menghapus pilihan jawaban
Obyek uji	Kebutuhan nomer 38 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus pilihan jawaban tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar pilihan jawaban telah dijalankan 2. Memilih tombol <i>delete</i> pada salah satu baris sub pilihan jawaban.
Hasil yang diharapkan	Pilihan jawaban dapat dihapus.

Sumber : [Pengujian]

6.3.1.38 Kasus Uji Menambah Data Waktu dan Tempat Pelaksanaan Tes

Tabel 6.42a dan tabel 6.42b menunjukkan kasus uji menambah data waktu dan tempat pelaksanaan tes.

Tabel 6.42a : Kasus uji menambah data waktu dan tempat pelaksanaan tes

Nama kasus uji	Kasus uji menambah data waktu dan tempat pelaksanaan tes
Obyek uji	Kebutuhan nomer 39 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah data waktu dan tempat pelaksanaan tes. Data yang harus dicatat dari penambahan data waktu dan tempat pelaksanaan tes adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar data waktu dan tempat pelaksanaan tes telah dijalankan 2. Memilih <i>link</i> tambah pelaksanaan tes. 3. Memasukkan data-data pada <i>form</i>. 4. Menekan tombol tambah.
Hasil yang diharapkan	Pelaksanaan tes soal berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.42b : Kasus uji menambah data waktu dan tempat pelaksanaan tes dimana masukan waktu, tempat, dan instansi *client* kosong

Nama kasus uji	Kasus uji menambah data waktu dan tempat pelaksanaan tes dimana waktu, tempat, dan instansi <i>client</i> kosong.
Obyek uji	Kebutuhan nomer 39 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa data waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar data waktu dan tempat pelaksanaan tes telah dijalankan 2. Memilih <i>tab</i> tambah pelaksanaan tes. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.39 Kasus Uji Melihat Data Waktu dan Tempat Pelaksanaan Tes

Tabel 6.43 menunjukkan kasus uji melihat daftar data waktu dan tempat pelaksanaan tes.

Tabel 6.43 : Kasus uji melihat daftar data waktu dan tempat pelaksanaan tes

Nama kasus uji	Kasus uji melihat daftar waktu dan tempat pelaksanaan tes
Obyek uji	Kebutuhan nomer 40 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua waktu dan tempat pelaksanaan tes yang tercatat di dalam sistem. Data yang ditampilkan berupa tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu pelaksanaan tes.
Hasil yang diharapkan	Semua daftar waktu dan tempat pelaksanaan tes dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.40 Kasus Uji Mengedit Data Waktu dan Tempat Pelaksanaan Tes

Tabel 6.44a dan tabel 6.44b menunjukkan kasus uji mengedit waktu dan tempat pelaksanaan tes.

Tabel 6.44a: Kasus uji mengedit waktu dan tempat pelaksanaan tes

Nama kasus uji	Kasus uji mengedit data waktu dan tempat pelaksanaan tes
Obyek uji	Kebutuhan nomer 41 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data waktu dan tempat pelaksanaan tes yang telah dimasukkan. Data yang dapat diedit adalah tanggal, waktu, tempat, alamat, instansi <i>client</i> , guide 1, guide 2, operator 1, operator 2.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar data waktu dan tempat pelaksanaan tes telah dijalankan 2. Memilih <i>link edit</i> pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol Simpan.
Hasil yang diharapkan	Data pelaksanaan tes dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.44b: Kasus uji mengedit waktu dan tempat pelaksanaan tes dimana masukan waktu, tempat, dan instansi *client* kosong

Nama kasus uji	Kasus uji mengedit waktu dan tempat pelaksanaan tes dimana masukan waktu, tempat, dan instansi <i>client</i> kosong.
Obyek uji	Kebutuhan nomer 41 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar waktu dan tempat pelaksanaan tes telah dijalankan 2. Memilih tombol edit pada salah satu baris.

	3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.41 Kasus Uji Menghapus Data Waktu dan Tempat Pelaksanaan

Tes

Tabel 6.45 menunjukkan kasus uji menghapus waktu dan tempat pelaksanaan tes.

Tabel 6.45 : Kasus uji menghapus waktu dan tempat pelaksanaan tes

Nama kasus uji	Kasus uji menghapus waktu dan tempat pelaksanaan tes
Obyek uji	Kebutuhan nomer 42 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus waktu dan tempat pelaksanaan tes tertentu.
Prosedur uji	3. Use case melihat daftar waktu dan tempat pelaksanaan tes telah dijalankan 4. Memilih tombol <i>delete</i> pada salah satu baris waktu dan tempat pelaksanaan tes.
Hasil yang diharapkan	Waktu dan tempat pelaksanaan tes dapat dihapus.

Sumber : [Pengujian]

6.3.1.42 Kasus Uji Mencari Data Waktu dan Tempat Pelaksanaan Tes

Tabel 6.46a, tabel 6.46b dan tabel 6.46c menunjukkan kasus uji mencari Waktu dan tempat pelaksanaan tes.

Tabel 6.46a : Kasus uji mencari sub waktu dan tempat pelaksanaan tes

Nama kasus uji	Kasus uji mencari waktu dan tempat pelaksanaan tes
Obyek uji	Kebutuhan nomer 43 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu waktu dan tempat pelaksanaan tes telah ada di dalam sistem. Pencarian data waktu dan tempat pelaksanaan tes dapat berdasarkan pada tempat dan nama instansi.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih menu pelaksanaan tes. 3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> .
Hasil yang diharapkan	Suatu waktu dan tempat pelaksanaan tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.46b : Kasus uji mencari waktu dan tempat pelaksanaan tes dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari waktu dan tempat pelaksanaan tes dimana masukan <i>keyword</i> kosong
----------------	---

Obyek uji	Kebutuhan nomer 43 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih <i>link</i> pelaksanaan tes. 3. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.46c : Kasus uji mencari waktu dan tempat pelaksanaan tes dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari waktu dan tempat pelaksanaan tes dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 43 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar waktu dan tempat pelaksanaan tes yang ditampilkan.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih <i>link</i> pelaksanaan tes. 3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> . 5. Menekan tombol <i>refresh</i> pelaksanaan tes.
Hasil yang diharapkan	Sistem menampilkan daftar waktu dan tempat pelaksanaan tes yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.43 Kasus Uji Menambah Data Peserta Tes

Tabel 6.47a dan tabel 6.47b menunjukkan kasus uji menambah data peserta tes.

Tabel 6.47a : Kasus uji menambah data peserta tes

Nama kasus uji	Kasus uji menambah data peserta tes
Obyek uji	Kebutuhan nomer 44 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah data peserta tes. Data yang harus dicatat dari penambahan data peserta tes adalah id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin.
Prosedur uji	1. Use case melihat daftar peserta tes telah dijalankan 2. Memilih <i>link</i> tambah peserta. 3. Memasukkan data-data pada <i>form</i> . 4. Menekan tombol tambah.
Hasil yang diharapkan	Peserta berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.47b : Kasus uji menambah data peserta tes dimana masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon kosong

Nama kasus uji	Kasus uji menambah data peserta tes dimana id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon kosong.
Obyek uji	Kebutuhan nomer 44 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar data peserta tes telah dijalankan 2. Memilih <i>tab</i> tambah peserta. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol tambah.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.44 Kasus Uji Melihat Data Peserta Tes

Tabel 6.48 menunjukkan kasus uji melihat data peserta tes.

Tabel 6.48 : Kasus uji melihat data peserta tes

Nama kasus uji	Kasus uji melihat daftar peserta tes
Obyek uji	Kebutuhan nomer 45 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua peserta tes yang tercatat di dalam sistem. Data yang ditampilkan berupa id peserta, nama, alamat, pendidikan terakhir, jurusan, dan jenis kelamin.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu data peserta tes.
Hasil yang diharapkan	Semua daftar peserta tes dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.45 Kasus Uji Mengedit Peserta Tes

Tabel 6.49a dan tabel 6.49b menunjukkan kasus uji mengedit peserta tes.

Tabel 6.49a: Kasus uji mengedit peserta tes

Nama kasus uji	Kasus uji mengedit data peserta tes
Obyek uji	Kebutuhan nomer 45 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data peserta tes yang telah dimasukkan. Data yang dapat diedit adalah id peserta, nama, alamat, pendidikan terakhir, jurusan, telepon, agama, tanggal lahir, jenis kelamin.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar data peserta tes telah dijalankan

	<ol style="list-style-type: none"> 2. Memilih <i>link edit</i> pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol Simpan.
Hasil yang diharapkan	Data peserta tes dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.49b: Kasus uji mengedit peserta tes dimana id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon kosong

Nama kasus uji	Kasus uji mengedit peserta tes dimana masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon kosong.
Obyek uji	Kebutuhan nomer 45 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar peserta tes telah dijalankan 2. Memilih tombol edit pada salah satu baris. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.46 Kasus Uji Menghapus Data Peserta Tes

Tabel 6.50 menunjukkan kasus uji menghapus peserta tes.

Tabel 6.50 : Kasus uji menghapus peserta tes

Nama kasus uji	Kasus uji menghapus peserta tes
Obyek uji	Kebutuhan nomer 46 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus peserta tes tertentu.
Prosedur uji	<ol style="list-style-type: none"> 5. Use case melihat daftar peserta tes telah dijalankan 6. Memilih tombol <i>delete</i> pada salah satu baris peserta tes.
Hasil yang diharapkan	peserta tes dapat dihapus.

Sumber : [Pengujian]

6.3.1.47 Kasus Uji Mencari Data Peserta Tes

Tabel 6.51a, tabel 6.51b dan tabel 6.51c menunjukkan kasus uji mencari peserta tes.

Tabel 6.51a : Kasus uji mencari peserta tes

Nama kasus uji	Kasus uji mencari peserta tes
Obyek uji	Kebutuhan nomer 47 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu peserta

	tes telah ada di dalam sistem. Pencarian data peserta tes dapat berdasarkan pada id peserta atau nama peserta.
Prosedur uji	5. Sistem telah berjalan. 6. Memilih menu peserta tes. 7. Memasukkan <i>keyword</i> . 8. Menekan tombol <i>find</i> .
Hasil yang diharapkan	Peserta tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.51b : Kasus uji mencari peserta tes dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari peserta tes dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 47 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	4. Sistem telah berjalan. 5. Memilih <i>link</i> peserta tes. 6. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.51c : Kasus uji mencari peserta tes dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari peserta tes dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 47 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar peserta tes yang ditampilkan.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih <i>link</i> peserta tes. 3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> . 5. Menekan tombol <i>refresh</i> peserta tes.
Hasil yang diharapkan	Sistem menampilkan daftar peserta tes yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.48 Kasus Uji Menambah Data Peserta pada Lokasi Tes

Tabel 6.52a dan tabel 6.52b menunjukkan kasus uji menambah data peserta pada lokasi tes.

Tabel 6.52a : Kasus uji menambah data peserta pada lokasi tes

Nama kasus uji	Kasus uji menambah data peserta pada lokasi tes
Obyek uji	Kebutuhan nomer 48 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah data peserta pada lokasi tes. Data yang

	harus dicatat dari penambahan data peserta pada lokasi tes adalah id peserta, tempat tes, dan waktu tes.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar peserta pada lokasi tes telah dijalankan 2. Memilih <i>link</i> tambah peserta di lokasi tes. 3. Memasukkan data-data pada <i>form</i>. 4. Menekan tombol tambah.
Hasil yang diharapkan	Peserta di lokasi tes berhasil ditambah.

Sumber : [Pengujian]

Tabel 6.52b : Kasus uji menambah data peserta tes dimana masukan id peserta, tempat tes, dan waktu tes kosong

Nama kasus uji	Kasus uji menambah data peserta di lokasi tes dimana id peserta, tempat tes, dan waktu tes kosong.
Obyek uji	Kebutuhan nomer 48 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa id peserta, tempat tes, dan waktu tes tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Use case melihat daftar data peserta di lokasi tes telah dijalankan 2. Memilih <i>tab</i> tambah peserta di lokasi tes. 3. Memasukkan data-data pada <i>form</i> 4. Menekan tombol tambah.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan id peserta, tempat tes, dan waktu tes tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.49 Kasus Uji Melihat Data Peserta di Lokasi Tes

Tabel 6.53 menunjukkan kasus uji melihat data peserta di lokasi tes.

Tabel 6.53 : Kasus uji melihat data peserta di lokasi tes

Nama kasus uji	Kasus uji melihat daftar peserta di lokasi tes
Obyek uji	Kebutuhan nomer 49 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua peserta tes yang tercatat di dalam sistem. Data yang ditampilkan berupa nama peserta, id peserta, tempat tes, waktu tes, dan <i>password login</i> peserta.
Prosedur uji	<ol style="list-style-type: none"> 3. Sistem telah berjalan dan pengguna telah login sebagai administrator. 4. Memilih menu data peserta di lokasi tes.
Hasil yang diharapkan	Semua daftar peserta di lokasi tes dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.50 Kasus Uji Mengedit Peserta di Lokasi Tes

Tabel 6.54a dan tabel 6.54b menunjukkan kasus uji mengedit peserta di lokasi tes.

Tabel 6.54a: Kasus uji mengedit peserta di lokasi tes

Nama kasus uji	Kasus uji mengedit data peserta di lokasi tes
Obyek uji	Kebutuhan nomer 50 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengedit data peserta di lokasi tes yang telah dimasukkan. Data yang dapat diedit adalah id peserta,tempat tes, dan waktu tes.
Prosedur uji	5. Use case melihat daftar data peserta di lokasi tes telah dijalankan 6. Memilih <i>link edit</i> pada salah satu baris. 7. Memasukkan data-data pada <i>form</i> 8. Menekan tombol Simpan.
Hasil yang diharapkan	Data peserta di lokasi tes dapat diedit dan disimpan.

Sumber : [Pengujian]

Tabel 6.54b: Kasus uji mengedit peserta di lokasi tes dimana id peserta, tempat tes, dan waktu tes kosong

Nama kasus uji	Kasus uji mengedit peserta di lokasi tes dimana masukan id peserta, tempat tes, dan waktu tes kosong.
Obyek uji	Kebutuhan nomer 50 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa id peserta, tempat tes, dan waktu tes tidak boleh kosong.
Prosedur uji	5. Use case melihat daftar peserta di lokasi tes telah dijalankan 6. Memilih tombol edit pada salah satu baris. 7. Memasukkan data-data pada <i>form</i> 8. Menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa id peserta, tempat tes, dan waktu tes tidak boleh kosong.

Sumber : [Pengujian]

6.3.1.51 Kasus Uji Menghapus Data Peserta di Lokasi Tes

Tabel 6.55 menunjukkan kasus uji menghapus peserta di lokasi tes.

Tabel 6.55 : Kasus uji menghapus peserta di lokasi tes

Nama kasus uji	Kasus uji menghapus peserta di lokasi tes
Obyek uji	Kebutuhan nomer 51 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus peserta di lokasi tes tertentu.
Prosedur uji	1. Use case melihat daftar peserta di lokasi tes telah dijalankan 2. Memilih tombol <i>delete</i> pada salah satu baris peserta di lokasi tes.
Hasil yang diharapkan	peserta di lokasi tes dapat dihapus.

Sumber : [Pengujian]

6.3.1.52 Kasus Uji Mencari Data Peserta di Lokasi Tes

Tabel 6.56a, tabel 6.56b dan tabel 6.56c menunjukkan kasus uji mencari peserta di lokasi tes.

Tabel 6.56a : Kasus uji mencari peserta di lokasi tes

Nama kasus uji	Kasus uji mencari peserta di lokasi tes
Obyek uji	Kebutuhan nomer 52 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu peserta di lokasi tes telah ada di dalam sistem. Pencarian data peserta di lokasi tes dapat berdasarkan pada id peserta atau nama peserta.
Prosedur uji	<ol style="list-style-type: none">1. Sistem telah berjalan.2. Memilih menu peserta di lokasi tes.3. Memasukkan <i>keyword</i>.4. Menekan tombol <i>find</i>.
Hasil yang diharapkan	Peserta di lokasi tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.56b : Kasus uji mencari peserta di lokasi tes dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari peserta di lokasi tes dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 52 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none">1. Sistem telah berjalan.2. Memilih <i>link</i> peserta di lokasi tes.3. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.56c : Kasus uji mencari peserta di lokasi tes dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari peserta di lokasi tes dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 52 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar peserta di lokasi tes yang ditampilkan.
Prosedur uji	<ol style="list-style-type: none">1. Sistem telah berjalan.2. Memilih <i>link</i> peserta di lokasi tes.3. Memasukkan <i>keyword</i>.4. Menekan tombol <i>find</i>.5. Menekan tombol <i>refresh</i> peserta di lokasi tes.
Hasil yang diharapkan	Sistem menampilkan daftar peserta di lokasi tes yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.53 Kasus Uji Mengenerate Password Login Peserta Tes

Tabel 6.57 menunjukkan kasus uji *mengenerate password login* peserta tes.

Tabel 6.57 : Kasus uji *mengenerate password login* peserta tes

Nama kasus uji	Kasus uji <i>mengenerate password login</i> peserta tes
Obyek uji	Kebutuhan nomer 53 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat <i>mengenerate password login</i> peserta tes.
Prosedur uji	1. Use case melihat daftar peserta pada lokasi tes telah dijalankan 2. Memilih <i>link edit</i> . 3. Menekan tombol simpan.
Hasil yang diharapkan	<i>Password</i> berhasil di <i>generate</i>

Sumber : [Pengujian]

6.3.1.54 Kasus Uji Menampilkan Nilai Peserta

Tabel 6.58 menunjukkan kasus uji melihat nilai peserta.

Tabel 6.58 : Kasus uji melihat nilai peserta

Nama kasus uji	Kasus uji melihat daftar nilai peserta
Obyek uji	Kebutuhan nomer 54 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua nilai peserta tes yang tercatat di dalam sistem. Data yang ditampilkan berupa daftar tempat, waktu, id peserta, nama, kategori, sub kategori, nilai total, nilai konsultan untuk setiap sub kategori soal.
Prosedur uji	1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu hasil tes.
Hasil yang diharapkan	Semua daftar nilai peserta dapat ditampilkan.

Sumber : [Pengujian]

6.3.1.55 Kasus Uji Mencari Nilai Peserta

Tabel 6.59a, tabel 6.59b dan tabel 6.59c menunjukkan kasus uji mencari nilai peserta.

Tabel 6.59a : Kasus uji mencari nilai peserta

Nama kasus uji	Kasus uji mencari nilai peserta
Obyek uji	Kebutuhan nomer 55 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari data suatu nilai peserta telah ada di dalam sistem. Pencarian data nilai peserta dapat berdasarkan pada id peserta, nama peserta, atau lokasi tes.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih menu hasil tes.

	3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> .
Hasil yang diharapkan	Nilai peserta tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.

Sumber : [Pengujian]

Tabel 6.59b : Kasus uji mencari nilai peserta dimana masukan *keyword* kosong

Nama kasus uji	Kasus uji mencari nilai peserta dimana masukan <i>keyword</i> kosong
Obyek uji	Kebutuhan nomer 55 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa <i>keyword</i> pencarian tidak boleh kosong.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih <i>link</i> hasil tes. 3. Menekan tombol <i>find</i> (<i>keyword</i> kosong).
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.

Sumber : [Pengujian]

Tabel 6.59c : Kasus uji mencari nilai peserta dan menekan tombol *refresh*

Nama kasus uji	Kasus uji mencari nilai peserta dan menekan tombol <i>refresh</i>
Obyek uji	Kebutuhan nomer 55 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem dapat me- <i>refresh</i> daftar nilai peserta yang ditampilkan.
Prosedur uji	1. Sistem telah berjalan. 2. Memilih menu hasil tes. 3. Memasukkan <i>keyword</i> . 4. Menekan tombol <i>find</i> . 5. Menekan tombol <i>refresh</i> hasil tes.
Hasil yang diharapkan	Sistem menampilkan daftar hasil tes yang telah di- <i>refresh</i> .

Sumber : [Pengujian]

6.3.1.56 Kasus Uji Menambah Penilaian Konsultan Terhadap Nilai yang Diperoleh Peserta untuk Setiap Sub Kategori

Tabel 6.60 menunjukkan kasus uji menambah penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori.

Tabel 6.60 : Kasus uji menambah penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori

Nama kasus uji	Kasus uji menambah penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori
Obyek uji	Kebutuhan nomer 56 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah penilaian konsultan terhadap nilai

	yang diperoleh peserta untuk setiap sub kategori. Data yang harus dicatat dari penambahan adalah penilaian konsultan.
Prosedur uji	1. Use case melihat daftar hasil tes telah dijalankan 2. Memilih <i>link edit</i> . 3. Memasukkan data-data pada <i>form</i> . 4. Menekan tombol simpan.
Hasil yang diharapkan	Penilaian konsultan berhasil ditambah.

Sumber : [Pengujian]

6.3.1.57 Kasus Uji Melihat Penilaian Konsultan Terhadap Nilai yang Diperoleh Peserta untuk Setiap Sub Kategori

Tabel 6.61 menunjukkan kasus uji melihat penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori.

Tabel 6.61 : Kasus uji melihat penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori

Nama kasus uji	Kasus uji melihat daftar penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori
Obyek uji	Kebutuhan nomer 57 (aliran utama)
Tujuan pengujian	Menguji bahwa system dapat menampilkan daftar penilaian konsultan terhadap nilai yang diperoleh peserta untuk setiap sub kategori.
Prosedur uji	1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu hasil tes.
Hasil yang diharapkan	Semua daftar penilaian konsultan dapat ditampilkan.

Sumber : [Pengujian]

6.3.2 Hasil Pengujian Validasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian pada sub pokok bahasan 6.3.1, didapatkan hasil seperti ditunjukkan pada Tabel 6.62.

Tabel 6.62 : Hasil Pengujian Validasi

No	Nama kasus uji	Hasil yang didapat	Status validitas
1	Kasus uji <i>login</i>	Pengguna berhasil login pada system.	Valid
2	Kasus uji <i>login</i> dimana pasangan username dan password tidak ada dalam sistem	Menampilkan pesan bahwa pasangan <i>username</i> atau <i>password</i> tidak ada dalam sistem.	Valid

3	Kasus uji <i>logout</i>	Pengguna berhasil <i>logout</i> dari system.	Valid
4	Kasus uji menambah data petugas	Data petugas berhasil ditambah.	Valid
5	Kasus uji menambah data petugas dimana masukan nama petugas, username atau password kosong.	Sistem menampilkan pesan kesalahan bahwa nama petugas atau <i>username</i> atau <i>password</i> tidak boleh kosong.	Valid
6	Kasus uji menambah data petugas dimana masukan username telah ada di dalam system.	Sistem menampilkan pesan kesalahan bahwa <i>username</i> tersebut telah ada di dalam sistem.	Valid
7	Kasus uji melihat daftar petugas	Semua daftar petugas dapat ditampilkan	Valid
8	Kasus uji mengedit data petugas	Data petugas dapat diedit dan disimpan.	Valid
9	Kasus uji mengedit data petugas dimana masukan nama petugas atau username atau password kosong.	Sistem menampilkan peringatan bahwa data nama atau <i>username</i> atau <i>password</i> tidak boleh kosong.	Valid
10	Kasus uji mengedit data petugas dimana username petugas telah ada di dalam sistem	Sistem menampilkan peringatan bahwa data username petugas telah ada di dalam system.	Valid
11	Kasus uji menghapus data petugas	Data petugas dapat dihapus.	Valid
12	Kasus uji mencari data petugas	Petugas berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
13	Kasus uji	Sistem menampilkan peringatan bahwa	Valid

	mencari petugas dimana masukan keyword kosong	<i>keyword</i> tidak boleh kosong.	
14	Kasus uji mencari petugas dan menekan tombol refresh	Sistem menampilkan daftar petugas yang telah di- <i>refresh</i> .	Valid
15	Kasus uji menambah instansi <i>client</i>	Data instansi <i>client</i> dapat ditambahkan.	Valid
16	Kasus uji menambah data instansi <i>client</i> dimana masukan nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon kosong	Sistem menampilkan peringatan bahwa masukan nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon tidak boleh kosong.	Valid
17	Kasus uji menambah data instansi <i>client</i> dimana nama instansi dan alamat telah ada dalam sistem	Sistem menampilkan peringatan bahwa nama instansi dan alamat tersebut telah ada di dalam system.	Valid
18	Kasus uji mencari instansi <i>client</i>	Suatu instansi <i>client</i> berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
19	Kasus uji mencari instansi <i>client</i> dimana masukan keyword kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
20	Kasus uji mengedit instansi <i>client</i>	Data suatu instansi <i>client</i> dapat diedit dan disimpan.	Valid
21	Kasus uji mengedit data instansi <i>client</i> dimana	Sistem menampilkan peringatan bahwa dimana instansi <i>client</i> dimana masukan nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon tidak boleh kosong.	Valid

	masukan dimana instansi <i>client</i> dimana masukan nama instansi, alamat, bidang usaha, <i>contact person</i> , dan telepon kosong		
22	Kasus uji mengedit data instansi <i>client</i> dimana nama instansi dan alamat telah ada di dalam sistem	Sistem menampilkan peringatan bahwa nama instansi dan alamat tersebut telah ada di dalam system.	Valid
23	Kasus uji menghapus instansi <i>client</i> .	Data suatu instansi <i>client</i> dapat dihapus.	Valid
24	Kasus uji menambah kategori soal	Data kategori soal dapat ditambahkan.	Valid
25	Kasus uji menambah data kategori soal dimana kategori soal kosong	Sistem menampilkan peringatan bahwa kategori soal tidak boleh kosong.	Valid
26	Kasus uji mencari kategori soal	Suatu kategori soal berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
27	Kasus uji mencari kategori soal dimana masukan keyword kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
28	Kasus uji mengedit kategori soal	Data suatu kategori soal dapat diedit dan disimpan.	Valid
29	Kasus uji mengedit data kategori soal dimana masukan kategori soal	Sistem menampilkan peringatan bahwa kategori soal tidak boleh kosong.	Valid

	kosong		
30	Kasus uji menghapus kategori soal.	Data suatu kategori soal dapat dihapus.	Valid
31	Kasus uji menambah urutan kategori soal	Data urutan kategori soal dapat ditambahkan.	Valid
32	Kasus uji menambah data urutan kategori soal dimana lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong	Sistem menampilkan peringatan bahwa lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.	Valid
33	Kasus uji mencari urutan kategori soal	Suatu urutan kategori soal berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
34	Kasus uji mencari urutan kategori soal dimana masukan <i>keyword</i> kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
35	Kasus uji mengedit urutan kategori soal	Data suatu urutan kategori soal dapat diedit dan disimpan.	Valid
36	Kasus uji mengedit data urutan kategori soal dimana masukan lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan kosong	Sistem menampilkan peringatan bahwa lokasi tes, nomor urut, kategori soal, sub kategori, jumlah soal, dan lama pengerjaan tidak boleh kosong.	Valid
37	Kasus uji	Data suatu urutan kategori soal dapat dihapus.	Valid

	menghapus urutan kategori soal.		
38	Kasus uji menambah data soal	Data soal dapat ditambahkan.	Valid
39	Kasus uji menambah data data soal dimana soal kosong	Sistem menampilkan peringatan bahwa soal tidak boleh kosong.	Valid
40	Kasus uji mencari soal	soal berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
41	Kasus uji mencari soal dimana masukan <i>keyword</i> kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
42	Kasus uji mengedit soal	Data suatu soal dapat diedit dan disimpan.	Valid
43	Kasus uji mengedit data soal dimana masukan soal kosong	Sistem menampilkan peringatan bahwa soal tidak boleh kosong.	Valid
44	Kasus uji menghapus soal.	Data soal dapat dihapus.	Valid
45	Kasus uji menjawab soal	Jawaban soal berhasil disimpan	Valid
46	Kasus uji menambah sub kategori soal	sub kategori soal dapat ditambahkan.	Valid
47	Kasus uji menambah data sub kategori soal dimana sub kategori soal kosong	Sistem menampilkan peringatan bahwa sub kategori soal tidak boleh kosong.	Valid
48	Kasus uji mencari sub kategori soal	sub kategori soal berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
49	Kasus uji mencari sub kategori soal dimana	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid

	masukan keyword kosong		
50	Kasus uji mengedit sub kategori soal	Data sub kategori soal dapat diedit dan disimpan.	Valid
51	Kasus uji mengedit data sub kategori soal dimana masukan sub kategori soal kosong	Sistem menampilkan peringatan bahwa sub kategori soal tidak boleh kosong.	Valid
52	Kasus uji menghapus sub kategori soal.	Data sub kategori soal dapat dihapus.	Valid
53	Kasus uji menambah pilihan jawaban	pilihan jawaban dapat ditambahkan.	Valid
54	Kasus uji menambah data pilihan jawaban dimana huruf pilihan, jawaban, dan status jawaban kosong	Sistem menampilkan peringatan bahwa huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.	Valid
55	Kasus uji mengedit pilihan jawaban	Data pilihan jawaban dapat diedit dan disimpan.	Valid
56	Kasus uji mengedit data pilihan jawaban dimana masukan huruf pilihan, jawaban, dan status jawaban kosong	Sistem menampilkan peringatan bahwa huruf pilihan, jawaban, dan status jawaban tidak boleh kosong.	Valid
57	Kasus uji menghapus pilihan jawaban.	Data pilihan jawaban dapat dihapus.	Valid
58	Kasus uji menambah data waktu dan tempat pelaksanaan tes	data waktu dan tempat pelaksanaan tes dapat ditambahkan.	Valid

59	Kasus uji menambah data waktu dan tempat pelaksanaan tes dimana waktu, tempat, dan instansi <i>client</i> kosong	Sistem menampilkan peringatan bahwa waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.	Valid
60	Kasus uji mencari data waktu dan tempat pelaksanaan tes	data waktu dan tempat pelaksanaan tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
61	Kasus uji mencari data waktu dan tempat pelaksanaan tes dimana masukan <i>keyword</i> kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
62	Kasus uji mengedit data waktu dan tempat pelaksanaan tes	data waktu dan tempat pelaksanaan tes dapat diedit dan disimpan.	Valid
63	Kasus uji mengedit data waktu dan tempat pelaksanaan tes dimana masukan waktu, tempat, dan instansi <i>client</i> kosong	Sistem menampilkan peringatan bahwa waktu, tempat, dan instansi <i>client</i> tidak boleh kosong.	Valid
64	Kasus uji menghapus data waktu dan tempat pelaksanaan tes.	Data waktu dan tempat pelaksanaan tes dapat dihapus.	Valid
65	Kasus uji menambah data peserta tes	data peserta tes dapat ditambahkan.	Valid
66	Kasus uji	Sistem menampilkan peringatan bahwa id	Valid

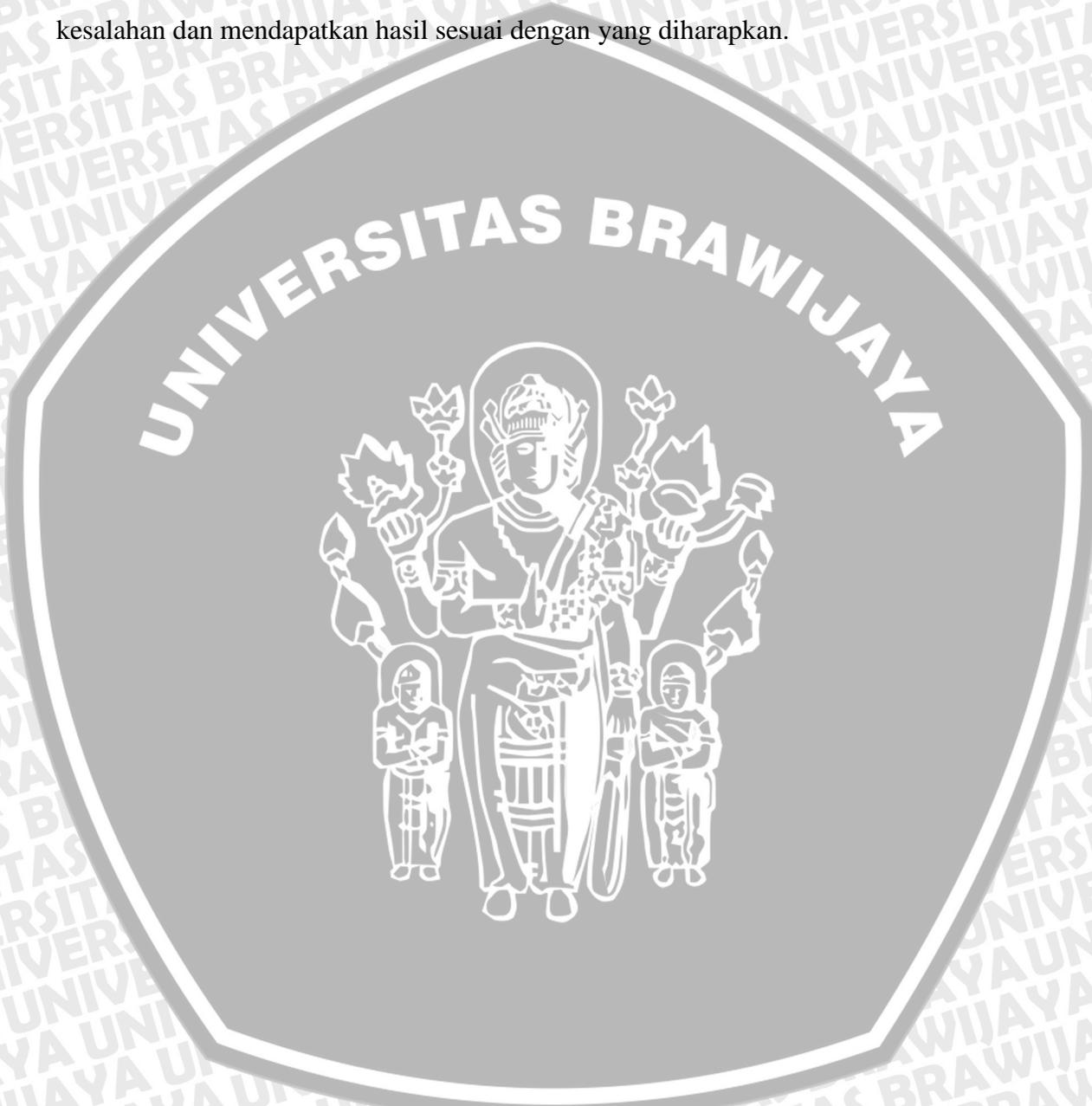
	menambah data peserta tes dimana id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon kosong	peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.	
67	Kasus uji mencari data peserta tes	data peserta tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
68	Kasus uji mencari data peserta tes dimana masukan <i>keyword</i> kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
69	Kasus uji mengedit peserta tes	data peserta tes dapat diedit dan disimpan.	Valid
70	Kasus uji mengedit data peserta tes dimana masukan id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon kosong	Sistem menampilkan peringatan bahwa id peserta, nama, alamat, pendidikan terakhir, jurusan, dan telepon tidak boleh kosong.	Valid
71	Kasus uji menghapus data peserta tes.	Data peserta tes dapat dihapus.	Valid
72	Kasus uji menambah data peserta di lokasi tes	data peserta di lokasi tes dapat ditambahkan.	Valid
73	Kasus uji menambah data peserta di lokasi tes dimana id peserta, tempat tes, dan waktu tes kosong	Sistem menampilkan peringatan bahwa id peserta, tempat tes, dan waktu tes tidak boleh kosong.	Valid

74	Kasus uji mencari data peserta di lokasi tes	data peserta di lokasi tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
75	Kasus uji mencari data peserta di lokasi tes dimana masukan <i>keyword</i> kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
76	Kasus uji mengedit peserta di lokasi tes	data peserta di lokasi tes dapat diedit dan disimpan.	Valid
77	Kasus uji mengedit data peserta di lokasi tes dimana masukan id peserta, tempat tes, dan waktu tes kosong	Sistem menampilkan peringatan bahwa id peserta, tempat tes, dan waktu tes tidak boleh kosong.	Valid
78	Kasus uji menghapus data peserta di lokasi tes.	Data peserta di lokasi tes dapat dihapus.	Valid
79	Kasus uji mengenerate <i>password login</i>	<i>Password</i> berhasil digenerate	Valid
80	Kasus uji menampilkan hasil tes	Hasil tes dapat ditampilkan	Valid
81	Kasus uji mencari data hasil tes	hasil tes berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
82	Kasus uji mencari hasil tes dimana masukan <i>keyword</i> kosong	Sistem menampilkan peringatan bahwa <i>keyword</i> tidak boleh kosong.	Valid
83	Kasus uji menambah penilaian konsultan	penilaian konsultan dapat ditambahkan.	Valid

84	Kasus uji melihat penilaian konsultan	penilaian konsultan dapat ditampilkan	Valid
----	---------------------------------------	---------------------------------------	-------

Sumber : [Pengujian]

Berdasar tabel 6.62. didapat bahwa semua uji validitas tidak memiliki kesalahan dan mendapatkan hasil sesuai dengan yang diharapkan.



BAB VII PENUTUP

7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Berdasar kebutuhan fungsional yang telah dirumuskan dalam perancangan, sistem psikotes *online* membutuhkan empat buah aktor dan 6 modul perancangan sistem.
2. Pada hasil pengujian unit terhadap sebuah *method* `create_session()` menggunakan metode *white box* terbukti bahwa seluruh jalur pada *method* ini telah sesuai dengan yang diharapkan.
3. Pada hasil pengujian integrasi terhadap *method* `setSoalList(db, smarty, ps)`, `setJawaban(db, smarty, ps)`, dan `simpanJawaban(db, smarty, ps)` terbukti bahwa ketiga *method* ini dapat melakukan fungsional utama yang dibutuhkan sistem.
4. Pengujian validasi yang dilakukan dengan menggunakan metode *black box* terbukti bahwa seluruh uji penerimaan yang dibutuhkan *user* adalah valid.
5. Berdasar implementasi sistem menggunakan *template engine* terbukti bahwa *template engine* dapat membantu pengorganisasian *file* bagi seorang *template designer* dan *programmer*.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan system psikotes *online* menggunakan *smarty template engine*:

1. Fitur – fitur pendukung psikotes *online* ditambahkan dan dibuat lebih bervariasi terutama untuk soal – soal yang berbentuk gambar.
2. Bentuk representasi hasil tes dapat lebih dikembangkan menjadi lebih baik, misalnya ditampilkan dalam bentuk grafik yang lebih mudah dipahami.
3. Sistem psikotes *online* dapat dikembangkan lagi sehingga mempunyai kemampuan untuk melakukan tes kepribadian, bakat, dan minat.

DAFTAR PUSTAKA

- [BOO-05] Booch, Grady , James Rumbaugh, dan Ivar Jacobson, 2005, “The Unified Modeling Language User Guide Second Edition”
- [BEL-03] Bell, Donald, 2003 “UML basics: An introduction to the Unified Modeling Language. 2003”,<http://www-128.ibm.com/developerworks/rational/library/769.html>, diakses tanggal 9 Juli 2008
- [CHO-03] Chonoles, Michael Jesse dan James A Schardt , 2003, ”UML 2 for Dumies”
- [COP-04] Copeland, Lee, 2004, “A Practitioner’s Guide to Software Test Design”,Artech House
- [DOC-05] Docherty, Mike O, 2005, “Object-Oriented Analysis and Design Understanding System Development with UML 2.0”
- [FAT-02] Fathansyah, 2002, “Basis Data”, Informatika, Bandung,
- [FOW-04] Fowler, Martin, 2004, “UML Distilled”
- [HAR-04] Hariyanto, Bambang, MT, 2004, Rekayasa Sistem Berorientasi Objek, Jakarta.
- [IEE-03] IEEE 610.12-2003,“IEEE Standard Glossary of Software Engineering Terminology”
- [IKO-08] Zaman Saeful. Spsi, 2008, “Memahami dan Mengerjakan Psikotes dengan Benar dan Tepat Waktu”, Visimedia, Jakarta.
- [OMG-04] OMG,2004, “Unified Modeling Language: Superstructure version 2.0 formal“
- [OTS-04] Budiman Arief, 2004, “Panduan Psikotes”,Pustaka Grafika, Jakarta
- [PIL-05] Pilone, Dan, Neil Pitman, 2005, “UML 2.0 in a Nutshell”
- [PHP-07] Stig Sæther Bakken, Egon Schmid, 2007, “PHP Manual”, PHP Documentation Group.
- [PSI-05] Chaplin, James P., 2005, Kamus Lengkap Psikologi, Rajawali Press, Jakarta.
- [PSU-04] Sudarsono, 2004, Pengantar Kuliah Psikologi Umum, Fak. psikologi Unas Pasim, Jakarta.

- [RYA-03] Ryan, Caspar, 2003, A Methodology for the Empirical Study of Object-Oriented Designers, Jakarta.
- [SIK-05] Sumarjono Sujono, 2005, "Buku Pintar Psikotes", Diva Press, Jakarta.
- [SMA-07] Andrei Zmievski, Monte Ohrt, 2007, "Smarty Manual", New Digital Group, Inc.
- [SOM-04] Sommerville, Ian, 2004, "Software Engineering"
- [SQL-06] MySQL, 2006, "MySQL 5.1 Reference Manual", MySQL AB.
- [TEM-08] anonymous, 2008, Template Processor,
http://en.wikipedia.org/wiki/Template_processor, diakses tanggal 13 Mei 2008
- [TPL-08] PHPSlum, 2008, Top 10 Template Engine,
<http://www.harshvardhans.com/best-php-template-engines/>, diakses tanggal 6 Mei 2008.

