

repository.ub.ac.id

Perancangan dan Pembuatan Sistem Informasi Job Placement Center di Politeknik Negeri Malang

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh :

ACHMAD MUFTI
NIM. 0610632001

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG
2010**

LEMBAR PERSETUJUAN

Perancangan dan Pembuatan Sistem Informasi Job Placement Center di Politeknik Negeri Malang

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

ACHMAD MUFTI
NIM. 0610632001- 63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Suprpto, ST, MT
NIP. 19720425199903 1 002

Himawat Aryadita, ST, M.Sc
NIP. 19570325198701 1 001



LEMBAR PENGESAHAN

**Perancangan dan Pembuatan Sistem Informasi
Job Placement Center di Politeknik Negeri Malang**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
ACHMAD MUFTI
0610632001- 63

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 4 Pebruari 2010

Menyetujui,
Dosen Penguji

Ir. M. Aswin, MT
NIP. 19640626 199002 1 001

Waru Djuriatno, ST., MT.
NIP. 19690725 199702 1 001

R. Arief Setyawan, ST., MT.
NIP. 19750819 199903 1 001

Mengetahui,
Ketua Jurusan Teknik Elektro

Rudy Yuwono, ST., M.Sc
NIP. 19710615 199802 1 003

KATA PENGANTAR

Segala Puji bagi Allah SWT, berjuta rasa syukur saya panjatkan kehadirat Allah SWT. Berkat rahmat dan hidayah-Nya, penyusunan skripsi dengan judul “Sistem Informasi Job Placement Center Berbasis *Web*” ini dapat diselesaikan pada waktunya. Saya menyadari bahwa skripsi ini tidak akan mencapai titik akhir penyelesaian tanpa bantuan berbagai pihak. Penulis mengucapkan terima kasih kepada:

1. Keluarga besar Bapak H. Mohammad Wahdi, untuk seluruh doa dan dukungannya yang telah diberikan kepada Ananda selama studi hingga terselesaikannya skripsi ini.
2. Bapak Rudy Yuwono, ST, MSc selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
3. Bapak M. Aziz Muslim, ST.,MT.,Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
4. Bapak Suprpto, ST, MT dan Himawat Aryadita, ST, M.Sc. selaku dosen pembimbing dalam penyusunan skripsi ini.
5. Bapak dan Ibu dosen serta karyawan Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
6. Rekan-rekan Asisten di Laboratorium Komputasi dan Jaringan Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya untuk saran, masukan, dan semangat selama proses penyusunan skripsi.
7. Teman, sahabat dan saudaraku yang telah bersama menjalani hari dengan suka dan duka. Terima kasih atas pelajaran hidup yang sangat berharga, semua kenangan ini tidak akan bisa terlupakan.
8. Serta semua pihak yang tak dapat disebutkan satu persatu yang telah turut membantu baik secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Hidup adalah perjuangan yang tiada henti-hentinya. Di setiap akhir perjalanan, di sana ada awal perjuangan. Keberhasilan kita ditentukan oleh bagaimana cara kita menyikapi setiap ujian. Hanya kepada Allah kita semua akan kembali.

Tersadar bahwa skripsi ini sangat jauh dari kesempurnaan, karenanya, segala kritik dan saran yang sifatnya membangun akan diterima dengan senang hati.

Akhir kata, saya berharap semoga skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Malang, 4 Pebruari 2010

Penyusun



DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xiii
RINGKASAN	xviii
BAB I PENDAHULUAN	1
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
BAB II TINJAUAN PUSTAKA	6
2.1 Sistem Informasi	6
2.2 Dasar Teori	7
2.2.1 Sekilas Tentang Politeknik	8
2.2.2 Teori Dasar RPL	10
2.2.2.1 Proses-Proses RPL	10
2.2.2.2 Model Proses Perangkat Lunak	11
2.2.2.3 Paradigma RPL	12
2.2.3 <i>Unified Modeling Language</i>	13
2.2.3.1 Diagram <i>Use Case</i>	16
2.2.3.2 Diagram Klas	23
2.2.3.3 Diagram <i>Sequence</i>	30
2.2.4 Pengujian (<i>Testing</i>).....	32
2.2.4.1 Level Pengujian	32
2.2.4.2 Jenis Pengujian	33
2.2.5 Jaringan Komputer.....	36

2.2.5.1	Topologi Jaringan Komputer	36
2.2.5.2	TCP/IP (<i>Transmission Control Protocol/Internet Protocol</i>).....	38
2.2.5.3	Manfaat Jaringan Komputer	40
2.2.6	Basis Data	40
2.2.6.1	<i>Entity-Relationship Diagram</i> (Diagram E-R).....	42
2.2.6.2	SQL (<i>Structured Query Language</i>)	44
2.2.6.3	MySQL	44
2.2.6.3.1	Jenis-jenis Kolom Pada MySQL.....	46
2.2.7	Teori Dasar Web	47
2.2.7.1	HTML (<i>Hyper Text Markup Language</i>).....	47
2.2.7.2	PHP (<i>PHP Hypertext Preprocessor</i>).....	48
BAB III METODOLOGI PENELITIAN		55
3.1	Studi Literatur	55
3.2	Analisis dan Perancangan Sistem Aplikasi.....	56
3.3	Implementasi Sistem Aplikasi	56
3.4	Pengujian Sistem Aplikasi	56
3.5	Pengambilan Kesimpulan dan Saran	56
BAB IV ANALISIS KEBUTUHAN DAN PERANCANGAN		57
4.1	Analisis Kebutuhan	57
4.1.1	Analisis Sistem yang Telah Ada	58
4.1.2	Analisis Kebutuhan Sistem yang akan dirancang	60
4.1.2.1	Daftar Kebutuhan	60
4.1.2.2	<i>Use Case Diagram</i>	65
4.1.2.2.1	<i>Use Case Diagram</i> untuk Modul Pengaturan Administator Sistem <i>Job Placement Center Online</i>	65
4.1.2.2.2	<i>Use Case Diagram</i> untuk Modul Pengaturan Alumni Sistem <i>Job Placement Center Online</i>	81
4.1.2.2.3	<i>Use Case Diagram</i> untuk Modul Pengaturan Perusahaan Sistem <i>Job Placement Center Online</i>	88
4.1.2.3	<i>Class Diagram</i>	105
4.1.2.3.1	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Admin	105

4.1.2.3.2	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Input Alumni.....	106
4.1.2.3.3	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Input Perusahaan..	107
4.1.2.3.4	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Input Berita	107
4.1.2.3.5	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Alumni	108
4.1.2.3.6	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket <i>Test Online</i>	108
4.1.2.3.7	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Kirim Lamaran.....	108
4.1.2.3.8	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Perusahaan	109
4.1.2.3.9	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Lihat Lowongan... ..	109
4.1.2.3.10	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Input Lowongan ..	110
4.1.2.3.11	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Lihat Kategori Soal	110
4.1.2.3.12	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Input Kategori Soal	111
4.1.2.3.13	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Lihat Soal	111
4.1.2.3.14	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Input Soal	112
4.1.2.3.15	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Lihat Peserta Test	112
4.1.2.3.16	<i>Class Diagram</i> untuk <i>Class-Class</i> dari Paket Input Peserta Test	113
4.1.2.4	<i>Sequence Diagram</i>	113
4.1.2.4.1	<i>Sequence Diagram</i> untuk <i>Method</i> setLowongan (\$db)	114
4.1.2.4.2	<i>Sequence Diagram</i> untuk <i>Method</i> setKategoriSoal (\$db)	114
4.1.2.4.3	<i>Sequence Diagram</i> untuk <i>Method</i> setJawaban(\$db)	115
4.1.2.4.4	<i>Sequence Diagram</i> untuk <i>Method</i> simpanJawaban(\$db) ...	115
4.2	Perancangan Basis Data	116
4.2.1	Data Object Description untuk Tabel Demo_Admin	117
4.2.2	Data Object Description untuk Tabel Demo_Alumni	117
4.2.3	Data Object Description untuk Tabel Demo_Berita	118
4.2.4	Data Object Description untuk Tabel Demo_CV.....	118
4.2.5	Data Object Description untuk Tabel Demo_Login.....	118
4.2.6	Data Object Description untuk Tabel Demo_Lowongan	119
4.2.7	Data Object Description untuk Tabel Demo_Perusahaan	119
4.2.8	Data Object Description untuk Tabel Demo_Peserta.....	119
4.2.9	Data Object Description untuk Tabel Demo_Soal	120
4.2.10	Data Object Description untuk Tabel Demo_Soal_Kategori	120



BAB V IMPLEMENTASI	121
5.1 Spesifikasi Sistem.....	121
5.1.1 Spesifikasi Perangkat Keras	121
5.1.2 Spesifikasi Perangkat Lunak	122
5.2 Implementasi Basis Data MySQL.....	122
5.3 Implementasi Antarmuka Kebutuhan Fungsional.....	126
5.3.1 Implementasi Antarmuka Login.....	126
5.3.2 Implementasi Antarmuka Home.....	128
5.3.3 Implementasi Antarmuka Administrator.....	128
5.3.4 Implementasi Antarmuka Input Alumni.....	129
5.3.5 Implementasi Antarmuka Lihat Alumni.....	131
5.3.6 Implementasi Antarmuka Input Perusahaan.....	132
5.3.7 Implementasi Antarmuka Lihat Perusahaan.....	133
5.3.8 Implementasi Antarmuka Input Berita	134
5.3.9 Implementasi Antarmuka Lihat Berita	135
5.3.10 Implementasi Antarmuka Alumni	136
5.3.11 Implementasi Antarmuka Edit Data Alumni.....	137
5.3.12 Implementasi Antarmuka Kirim Lamaran.....	137
5.3.13 Implementasi Antarmuka Ikuti <i>Test Online</i>	138
5.3.14 Implementasi Antarmuka Perusahaan	138
5.3.15 Implementasi Antarmuka Edit Perusahaan	139
5.3.16 Implementasi Antarmuka Input Lowongan.....	139
5.3.17 Implementasi Antarmuka Lihat Lowongan.....	141
5.3.18 Implementasi Antarmuka Input Kategori Soal.....	142
5.3.19 Implementasi Antarmuka Lihat Kategori Soal.....	142
5.3.20 Implementasi Antarmuka Input Soal.....	143
5.3.21 Implementasi Antarmuka Lihat Soal.....	144
5.3.22 Implementasi Antarmuka Input Peserta Test	145
5.3.23 Implementasi Antarmuka Lihat Peserta Test	146
5.3.24 Implementasi Antarmuka Lihat Lowongan.....	146
5.3.25 Implementasi Antarmuka <i>Detail</i> Lihat Lowongan.....	147
5.3.26 Implementasi Antarmuka Cari Perusahaan	147

5.3.27	Implementasi Antarmuka Detail Lihat perusahaan	148
5.3.28	Implementasi Antarmuka Cari Alumni	149
5.3.29	Implementasi Antarmuka Detail Lihat Alumni	150
5.3.30	Implementasi Antarmuka Lihat Berita	151
BAB VI PENGUJIAN DAN ANALISIS		152
6.1	Pengujian	152
6.1.1	Pengujian Unit	152
6.1.2.1	Pengujian Integrasi	154
6.1.3	Pengujian Koneksi Basis Data dan Web Server	157
6.1.4	Pengujian Validasi	160
6.1.4.1.1	Kasus Uji Validasi	160
6.1.4.1.1	Kasus Uji <i>Login</i>	160
6.1.4.1.2	Kasus Uji <i>Logout</i>	161
6.1.4.1.3	Kasus Uji Menambah Data Alumni	161
6.1.4.1.4	Kasus Uji Melihat Daftar Alumni	162
6.1.4.1.5	Kasus Uji Melihat Daftar Alumni Secara Detail	162
6.1.4.1.6	Kasus Uji Mengubah Data Alumni	163
6.1.4.1.7	Kasus Uji Menghapus Data Alumni	164
6.1.4.1.8	Kasus Uji Menambah Data Perusahaan	164
6.1.4.1.9	Kasus Uji Melihat Daftar Perusahaan	165
6.1.4.1.10	Kasus Uji Melihat Daftar Perusahaan Secara Detail	166
6.1.4.1.11	Kasus Uji Mengubah Profil Perusahaan	166
6.1.4.1.12	Kasus Uji Menghapus Profil Perusahaan	167
6.1.4.1.13	Kasus Uji Menambah Data Berita	168
6.1.4.1.14	Kasus Uji Melihat Daftar Berita	168
6.1.4.1.15	Kasus Uji Melihat Daftar Berita Secara Detail	168
6.1.4.1.16	Kasus Uji Mengubah Data Berita	169
6.1.4.1.17	Kasus Uji Menghapus Berita	170
6.1.4.1.18	Kasus Uji Mereset Password	170
6.1.4.1.19	Kasus Uji Mengubah Profil Perusahaan	171
6.1.4.1.20	Kasus Uji Menambah Data Lowongan	171

6.1.4.1.21 Kasus Uji Melihat Daftar Lowongan	172
6.1.4.1.22 Kasus Uji Melihat Daftar Lowongan Secara Detail	173
6.1.4.1.23 Kasus Uji Mengubah Data Lowongan	173
6.1.4.1.24 Kasus Uji Menghapus Data Lowongan.....	174
6.1.4.1.25 Kasus Uji Melihat Daftar Alumni Yang Mengirim Lamaran	175
6.1.4.1.26 Kasus Uji Men-Download Berkas Lamaran.....	175
6.1.4.1.27 Kasus Uji Menambah Data Kategori Soal.....	176
6.1.4.1.28 Kasus Uji Melihat Daftar Kategori Soal	176
6.1.4.1.29 Kasus Uji Mengubah Kategori Soal	177
6.1.4.1.30 Kasus Uji Menghapus Kategori Soal	177
6.1.4.1.31 Kasus Uji Menambah Data Soal.....	178
6.1.4.1.32 Kasus Uji Melihat Daftar Soal	179
6.1.4.1.33 Kasus Uji Mengubah Daftar Soal.....	179
6.1.4.1.34 Kasus Uji Menghapus Daftar Soal	180
6.1.4.1.35 Kasus Uji Menambah Data Peserta Test	181
6.1.4.1.36 Kasus Uji Melihat Daftar Peserta <i>Test Online</i>	182
6.1.4.1.37 Kasus Uji Menghapus Daftar Peserta <i>Test Online</i>	182
6.1.4.1.38 Kasus Uji Mengubah Data Diri Alumni.....	183
6.1.4.1.39 Kasus Uji Ikuti <i>Test Online</i>	184
6.1.4.1.40 Kasus Uji Mencari Data Alumni	185
6.1.4.1.41 Kasus Uji Mencari Data Perusahaan	186
6.1.4.2 Hasil Pengujian Validasi	186
BAB VII PENUTUP	190
7.1 Kesimpulan.....	190
7.2 Saran.....	190
DAFTAR PUSTAKA	191

DAFTAR GAMBAR

	Halaman
1. Gambar 2.1 Model Proses <i>Waterfall</i>	12
2. Gambar 2.2 Klasifikasi Jenis Diagram UML Versi 2.....	14
3. Gambar 2.3 Contoh <i>Use Case Diagram</i>	17
4. Gambar 2.4 Asosiasi antara Aktor dengan <i>Use Case</i>	19
5. Gambar 2.5 Generalisasi dari Aktor.....	20
6. Gambar 2.6 Contoh Generalisasi Aktor.....	21
7. Gambar 2.7 Generalisasi dari <i>Use Case</i>	21
8. Gambar 2.8 Notasi <<include>>.....	22
9. Gambar 2.9 Contoh <i>Include</i> antara dua buah <i>Use Case</i>	22
10. Gambar 2.10 Notasi <<extend>>.....	23
11. Gambar 2.11 Notasi klas	23
12. Gambar 2.12 Contoh Notasi Klas dengan tiga buah <i>Compartment</i>	24
13. Gambar 2.13 Contoh Notasi klas dengan <i>Inlined Attribute</i>	25
14. Gambar 2.14 Notasi Dependensi.....	27
15. Gambar 2.15 Notasi Asosiasi.....	28
16. Gambar 2.16 Notasi Agregasi.....	28
17. Gambar 2.17 Notasi Komposisi	29
18. Gambar 2.18 Notasi <i>Generalisasi</i>	29
19. Gambar 2.19 Contoh Diagram <i>Sequence</i>	30
20. Gambar 2.20 Notasi <i>Lifelines</i>	31
21. Gambar 2.21 Notasi <i>Message</i>	31
22. Gambar 2.22 Notasi <i>Synchronous Message</i>	32
23. Gambar 2.23 Notasi <i>Return Message</i>	32
24. Gambar 2.24 Notasi Grafik Alir.....	35
25. Gambar 2.25 Topologi Bus	36
26. Gambar 2.26 Topologi Ring	37
27. Gambar 2.27 Topologi Star.....	37
28. Gambar 2.28 <i>Layer TCP/IP</i>	38
29. Gambar 2.29 Contoh Skema Dalam Basis Data	41



30. Gambar 2.30	Skema Konsep Kerja PHP.....	49
31. Gambar 4.1	Diagram Pohon Analisis Kebutuhan dan Perancangan.....	57
32. Gambar 4.2	Blok Diagram Sistem Informasi <i>Job Placement Center</i>	58
33. Gambar 4.3	Arsitektur Jaringan Sistem Informasi <i>Job Placement Center</i>	60
34. Gambar 4.4	Generalisasi Aktor-Aktor <i>Job Placement Center Online</i>	61
35. Gambar 4.5	<i>Use Case Diagram</i> untuk Modul Pengaturan Administrator <i>Job Placement Center Online</i>	66
36. Gambar 4.6	<i>Use Case Diagram</i> untuk Modul Pengaturan Alumni Sistem <i>Job Placement Center Online</i>	82
37. Gambar 4.7	<i>Use Case Diagram</i> untuk Modul Pengaturan Perusahaan Sistem <i>Job Placement Center Online</i>	88
38. Gambar 4.8	<i>Class Diagram</i> Alumni-Polinema.admin	106
39. Gambar 4.9	<i>Class Diagram</i> Alumni-Polinema.inputAlumni	106
40. Gambar 4.10	<i>Class Diagram</i> Alumni-Polinema.inputPerusahaan	107
41. Gambar 4.11	<i>Class Diagram</i> Alumni-Polinema.inputBerita	107
42. Gambar 4.12	<i>Class Diagram</i> alumni-polinema.alumni	108
43. Gambar 4.13	<i>Class Diagram</i> Alumni-Polinema.testOnline	108
44. Gambar 4.14	<i>Class Diagram</i> Alumni-Polinema.kirimLamaran	109
45. Gambar 4.15	<i>Class Diagram</i> Alumni-Polinema.Perusahaan	109
46. Gambar 4.16	<i>Class Diagram</i> Alumni-Polinema.lihatLowongan.....	109
47. Gambar 4.17	<i>Class Diagram</i> Alumni-Polinema.inputLowongan.....	110
48. Gambar 4.18	<i>Class Diagram</i> Alumni-Polinema.inputKategoriSoal.....	110
49. Gambar 4.19	<i>Class Diagram</i> Alumni-Polinema.inputKategoriSoal.....	111
50. Gambar 4.20	<i>Class Diagram</i> Alumni-Polinema.lihatSoal.....	111
51. Gambar 4.21	<i>Class Diagram</i> Alumni-Polinema.inputSoal.....	112
52. Gambar 4.22	<i>Class Diagram</i> Alumni-Polinema.lihatPesertaTest	112
53. Gambar 4.23	<i>Class Diagram</i> Alumni-Polinema.inputPesertaTest	113
54. Gambar 4.24	<i>Sequence Diagram</i> untuk <i>method</i> setLowongan	114
55. Gambar 4.25	<i>Sequence Diagram</i> untuk <i>method</i> setKategoriSoal.....	114
56. Gambar 4.26	<i>Sequence Diagram</i> untuk <i>Method</i> setJawaban.....	115
57. Gambar 4.27	<i>Sequence Diagram</i> untuk <i>method</i> simpanJawaban	115
58. Gambar 4.28	<i>ER Diagram</i> Database Sistem <i>Job Placement Center Online</i>	116

59. Gambar 5.1	Diagram Pohon Implementasi Sistem	121
60. Gambar 5.2	Implementasi Basis Data <i>job placement center online</i> pada MySQL oleh phpMyAdmin.....	126
61. Gambar 5.3	Halaman Login.....	126
62. Gambar 5.4	Halaman Home (Halaman Awal).....	128
63. Gambar 5.5	Halaman Administrator.....	128
64. Gambar 5.6	Legend Input Data Alumni.....	129
65. Gambar 5.7	Halaman Lihat Alumni.....	131
66. Gambar 5.8	Halaman Alumni Secara Detail.....	131
67. Gambar 5.9	Legend Input Perusahaan	132
68. Gambar 5.10	Halaman Lihat Perusahaan.....	133
69. Gambar 5.11	Halaman Perusahaan Secara Detail.....	134
70. Gambar 5.12	Legend Input Berita.....	134
71. Gambar 5.13	Halaman Lihat Berita	135
72. Gambar 5.14	Halaman Berita Secara Detail	135
73. Gambar 5.15	Halaman Alumni	136
74. Gambar 5.16	Legend Detail Form Edit Data Alumni.....	137
75. Gambar 5.17	Halaman Kirim Lamaran.....	138
76. Gambar 5.18	Halaman Ikuti <i>Test Online</i>	138
77. Gambar 5.19	Halaman Perusahaan	139
78. Gambar 5.20	Form Edit Perusahaan	139
79. Gambar 5.21	Legend Detail pada Form Input Lowongan	140
80. Gambar 5.22	Halaman Lihat Lowongan.....	141
81. Gambar 5.23	Halaman Lihat Pelamar	142
82. Gambar 5.24	Legend Form Input Kategori Soal.....	142
83. Gambar 5.25	Halaman Lihat Kategori Soal.....	142
84. Gambar 5.26	Legend Form Input Soal.....	143
85. Gambar 5.27	Halaman Lihat Soal.....	144
86. Gambar 5.28	Legend Form Input Peserta Test	145
87. Gambar 5.29	Halaman Lihat Peserta Test.....	146
88. Gambar 5.30	Konfirmasi Lihat Lowongan	147
89. Gambar 5.31	Halaman <i>Detail</i> Lihat Lowongan.....	147



90. Gambar 5.32	Halaman Pencarian Perusahaan Jika yang dicari ada dalam Database	148
91. Gambar 5.33	Halaman Pencarian Perusahaan Jika yang dicari tidak ada dalam Database	148
92. Gambar 5.34	Halaman <i>Detail</i> Lihat Perusahaan.....	149
93. Gambar 5.35	Halaman Pencarian Alumni Jika yang dicari ada dalam Database	149
94. Gambar 5.36	Halaman Pencarian Alumni Jika yang dicari tidak ada dalam Database	150
95. Gambar 5.37	Halaman <i>Detail</i> Lihat Alumni.....	150
96. Gambar 5.38	Halaman Berita.....	151
97. Gambar 6.1	Diagram Pohon Pengujian.....	152
98. Gambar 6.2	Klas Diagram <code>private_sessions</code>	153
99. Gambar 6.3	Pemodelan operasi <code>create_session()</code> ke dalam <i>flow graph</i> ...	153
100. Gambar 6.4	Pemodelan operasi <code>setInputSoal(\$db)</code> ke dalam <i>flow graph</i>	154
101. Gambar 6.5	Pemodelan operasi <code>setJawaban(\$db)</code> ke dalam <i>flow graph</i> ..	155
102. Gambar 6.6	Pemodelan operasi <code>simpanJawaban(\$db)</code> ke dalam <i>flow graph</i> ...	156
103. Gambar 6.7	Perintah <code>netstat -an</code> pada komputer <i>server</i> sebelum ada koneksi.	158
104. Gambar 6.8	Perintah <code>netstat -an</code> pada komputer <i>client</i> sebelum ada koneksi..	158
105. Gambar 6.9	Perintah <code>netstat -an</code> pada komputer <i>client</i> setelah ada koneksi....	159
106. Gambar 6.10	Perintah <code>netstat -an</code> pada komputer <i>server</i> setelah ada koneksi...	159



DAFTAR TABEL

	Halaman
1. Tabel 2.1 Diagram UML.....	14
2. Tabel 2.2 Notasi Aktor.....	18
3. Tabel 2.3 Notasi <i>Use Case</i>	18
4. Tabel 2.4 Bentuk-Bentuk Multiplicity.....	26
5. Tabel 2.5 Jenis Kolom Numerik.....	46
6. Tabel 2.6 Jenis Kolom Tanggal dan Waktu.....	47
7. Tabel 2.7 Jenis Kolom Karakter.....	47
8. Tabel 2.8 Parameter <i>Cookie</i>	53
9. Tabel 4.1 Deskripsi Aktor.....	61
10. Tabel 4.2 Daftar Kebutuhan Fungsional.....	61
11. Tabel 4.3 Daftar Kebutuhan Non Fungsional.....	65
12. Tabel 4.4 <i>Use Case Specification</i> Login.....	66
13. Tabel 4.5 <i>Use Case Specification</i> Logout.....	67
14. Tabel 4.6 <i>Use Case Specification</i> Menambah Data Alumni.....	68
15. Tabel 4.7 <i>Use Case Specification</i> Melihat Biodata Alumni Secara Detail.....	69
16. Tabel 4.8 <i>Use Case Specification</i> Mengedit Data Alumni.....	70
17. Tabel 4.9 <i>Use Case Specification</i> Menghapus Data Alumni.....	72
18. Tabel 4.10 <i>Use Case Specification</i> Menambah Daftar Perusahaan.....	73
19. Tabel 4.11 <i>Use Case Specification</i> Melihat Profil Perusahaan Secara Detail.....	74
20. Tabel 4.12 <i>Use Case Specification</i> Mengedit Profil Perusahaan.....	75
21. Tabel 4.13 <i>Use Case Specification</i> Menghapus Daftar Perusahaan.....	76
22. Tabel 4.14 <i>Use Use Case Specification</i> Menambah Daftar Berita.....	77
23. Tabel 4.15 <i>Use Case Specification</i> Melihat Berita Secara Detail.....	78
24. Tabel 4.16 <i>Use Case Specification</i> Mengedit Berita.....	79
25. Tabel 4.17 <i>Use Case Specification</i> Menghapus Daftar Berita.....	80
26. Tabel 4.18 <i>Use Case Specification</i> Mereset Password.....	81
27. Tabel 4.19 <i>Use Case Specification</i> Login.....	82
28. Tabel 4.20 <i>Use Case Specification</i> Logout.....	83
29. Tabel 4.21 <i>Use Case Specification</i> Ubah Data Diri.....	83



30. Tabel 4.22 <i>Use Case Specification</i> Lowongan Pekerjaan.....	85
31. Tabel 4.23 <i>Use Case Specification</i> Melihat Lowongan Kerja Secara Detail	85
32. Tabel 4.24 <i>Use Case Specification</i> Mengirim Berkas Lamaran	86
33. Tabel 4.25 <i>Use Case Specification</i> Ikuti Test Online	87
34. Tabel 4.26 <i>Use Case Specification</i> Login	88
35. Tabel 4.27 <i>Use Case Specification</i> Logout	89
36. Tabel 4.28 <i>Use Case Specification</i> Ubah Data Diri Perusahaan.....	90
37. Tabel 4.29 <i>Use Case Specification</i> Input Lowongan Pekerjaan	91
38. Tabel 4.30 <i>Use Case Specification</i> Melihat Lowongan Pekerjaan	92
39. Tabel 4.31 <i>Use Case Specification</i> Melihat Lowongan Secara Detail	93
40. Tabel 4.32 <i>Use Case Specification</i> Mengubah Data Lowongan.....	93
41. Tabel 4.33 <i>Use Case Specification</i> Menghapus Data Lowongan	95
42. Tabel 4.34 <i>Use Case Specification</i> Lihat Pelamar.....	95
43. Tabel 4.35 <i>Use Case Specification</i> Input Kategori Soal.....	96
44. Tabel 4.36 <i>Use Case Specification</i> Lihat Kategori Soal.....	97
45. Tabel 4.37 <i>Use Case Specification</i> Mengubah Kategori Soal	98
46. Tabel 4.38 <i>Use Case Specification</i> Menghapus Kategori Soal.....	99
47. Tabel 4.39 <i>Use Case Specification</i> Input Soal.....	99
48. Tabel 4.40 <i>Use Case Specification</i> Lihat Soal.....	100
49. Tabel 4.41 <i>Use Case Specification</i> Mengubah Soal	101
50. Tabel 4.42 <i>Use Case Specification</i> Menghapus Soal.....	102
51. Tabel 4.43 <i>Use Case Specification</i> Input Peserta Test.....	103
52. Tabel 4.44 <i>Use Case Specification</i> Lihat Peserta Test.....	104
53. Tabel 4.45 <i>Use Case Specification</i> Menghapus Peserta Test	105
54. Tabel 4.46 <i>Data Object Description</i> untuk Tabel demo_admin.....	117
55. Tabel 4.47 <i>Data Object Description</i> untuk Tabel demo_alumni.....	117
56. Tabel 4.48 <i>Data Object Description</i> untuk Tabel demo_berita.....	118
57. Tabel 4.49 <i>Data Object Description</i> untuk Tabel demo_cv	118
58. Tabel 4.50 <i>Data Object Description</i> untuk Tabel demo_login.....	118



59. Tabel 4.51 <i>Data Object Description</i> untuk Tabel demo_Lowongan ..	119
60. Tabel 4.52 <i>Data Object Description</i> untuk Tabel Perusahaan.....	119
61. Tabel 4.53 <i>Data Object Description</i> untuk Tabel demo_Peserta.....	119
62. Tabel 4.54 <i>Data Object Description</i> untuk Tabel demo_Soal	120
63. Tabel 4.55 <i>Data Object Description</i> untuk Tabel demo_Soal_Kategori	120
64. Tabel 4.56 Spesifikasi Perangkat Keras	122
65. Tabel 4.57 Spesifikasi Perangkat Lunak	122
66. Tabel 6.1 <i>Test Case</i> untuk Pengujian Integrasi <i>method</i> create_session()..	153
67. Tabel 6.2 <i>Test Case</i> untuk pengujian integrasi <i>method</i> setInputSoal	155
68. Tabel 6.3 <i>Test Case</i> untuk pengujian integrasi <i>method</i> setJawaban	155
69. Tabel 6.4 <i>Test Case</i> untuk pengujian integrasi <i>method</i> simpanJawaban ..	156
70. Tabel 6.5a Kasus uji <i>login</i> (aliran utama)	160
71. Tabel 6.5b Kasus uji <i>login</i> (aliran alternatif 1)	160
72. Tabel 6.6 Kasus uji <i>logout</i>	161
73. Tabel 6.7a Kasus uji menambah data alumni (aliran utama)	161
74. Tabel 6.7b Kasus uji menambah data alumni (aliran alternatif 1)	161
75. Tabel 6.7c Kasus uji menambah data alumni (aliran alternatif 2)	162
76. Tabel 6.8 Kasus uji melihat daftar alumni (aliran utama)	162
77. Tabel 6.9a Kasus uji melihat daftar alumni secara detail (aliran utama)	162
78. Tabel 6.9b Kasus uji melihat daftar alumni secara detail (aliran alternatif 1)	163
79. Tabel 6.10a Kasus uji mengubah data alumni (aliran utama).....	163
80. Tabel 6.10b Kasus uji mengubah data alumni (aliran alternatif 1).....	163
81. Tabel 6.11a Kasus uji menghapus data alumni (aliran utama).....	164
82. Tabel 6.11b Kasus uji menghapus data alumni (aliran alternatif 1)	164
83. Tabel 6.12a Kasus uji menambah data perusahaan (aliran utama)	164
84. Tabel 6.12b Kasus uji menambah data perusahaan (aliran alternatif 1)	165
85. Tabel 6.12c Kasus uji menambah data perusahaan (aliran alternatif 2)	165
86. Tabel 6.13 Kasus uji melihat daftar perusahaan (aliran utama)	165
87. Tabel 6.14a Kasus uji melihat daftar perusahaan secara detail (aliran utama)	166
88. Tabel 6.14b Kasus uji menambah data perusahaan (aliran alternatif 1)	166
89. Tabel 6.15a Kasus uji mengubah profil perusahaan (aliran utama).....	166
90. Tabel 6.15b Kasus uji mengubah profil perusahaan (aliran alternatif 1)	167

91. Tabel 6.16a Kasus uji menghapus profil perusahaan (aliran utama) ...	167
92. Tabel 6.16b Kasus uji menghapus profil perusahaan (aliran alternatif 1) ...	167
93. Tabel 6.17 Kasus uji menambah data berita (aliran utama)	168
94. Tabel 6.18 Kasus uji melihat daftar berita (aliran utama)	168
95. Tabel 6.19a Kasus uji melihat daftar berita secara detail (aliran utama)	168
96. Tabel 6.19b Kasus uji melihat daftar berita secara detail (aliran alternatif 1)	169
97. Tabel 6.20a Kasus uji mengubah data alumni (aliran utama)	169
98. Tabel 6.20b Kasus uji mengubah data berita (aliran alternatif 1)	169
99. Tabel 6.21a Kasus uji menghapus berita (aliran utama)	170
100. Tabel 6.21b Kasus uji menghapus berita (aliran alternatif 1)	170
101. Tabel 6.22 Kasus uji mengubah data perusahaan (aliran utama)	170
102. Tabel 6.23a Kasus uji mengubah data perusahaan (aliran utama)	171
103. Tabel 6.23b Kasus uji mengubah profil perusahaan (aliran alternatif 1) ..	171
104. Tabel 6.24a Kasus uji menambah data lowongan (aliran utama)	171
105. Tabel 6.24b Kasus uji menambah data lowongan (aliran alternatif 1)	172
106. Tabel 6.24c Kasus uji menambah data lowongan (aliran alternatif 2)	172
107. Tabel 6.25 Kasus uji melihat daftar lowongan (aliran utama)	172
108. Tabel 6.26a Kasus uji melihat daftar lowongan secara detail (aliran utama)	173
109. Tabel 6.26b Kasus uji melihat daftar lowongan (aliran alternatif 1).	173
110. Tabel 6.27a Kasus uji mengubah data lowongan (aliran utama)	173
111. Tabel 6.27b Kasus uji mengubah data lowongan (aliran alternatif 1)	174
112. Tabel 6.28a Kasus uji menghapus data lowongan (aliran utama)	174
113. Tabel 6.28b Kasus uji menghapus data lowongan (aliran alternatif 1)	174
114. Tabel 6.29 Kasus uji melihat daftar pelamar (aliran utama)	175
115. Tabel 6.30a Kasus uji men-download berkas lamaran (aliran utama)	175
116. Tabel 6.30b Kasus uji men-download berkas lamaran (aliran alternatif 1)	175
117. Tabel 6.31a Kasus uji menambah data kategori soal (aliran utama)	176
118. Tabel 6.31b Kasus uji menambah data kategori soal (aliran alternatif 1) .	176
119. Tabel 6.32 Kasus uji melihat daftar kategori soal (aliran utama)	176
120. Tabel 6.33a Kasus uji mengubah kategori soal (aliran utama)	177
121. Tabel 6.33b Kasus uji mengubah kategori soal (aliran alternatif 1)	177
122. Tabel 6.34a Kasus uji menghapus kategori soal (aliran utama)	177

123. Tabel 6.34b Kasus uji menghapus kategori soal (aliran alternatif 1)	178
124. Tabel 6.35a Kasus uji menambah data soal (aliran utama)	178
125. Tabel 6.35b Kasus uji menambah data soal (aliran alternatif 1)	178
126. Tabel 6.36a Kasus uji melihat daftar soal (aliran utama)	179
127. Tabel 6.36b Kasus uji melihat daftar soal (aliran alternatif 1)	179
128. Tabel 6.37a Kasus uji mengubah daftar soal (aliran utama).....	179
129. Tabel 6.37b Kasus uji mengubah daftar soal (aliran alternatif 1).....	180
130. Tabel 6.38a Kasus uji menghapus daftar soal (aliran utama).....	180
131. Tabel 6.38b Kasus uji menghapus daftar soal (aliran alternatif 1)	180
132. Tabel 6.39a Kasus uji menambah data peserta test (aliran utama).....	181
133. Tabel 6.39b Kasus uji menambah data peserta test (aliran alternatif 1)....	181
134. Tabel 6.39c Kasus uji menambah data peserta test (aliran alternatif 2)....	181
135. Tabel 6.40 Kasus uji melihat daftar peserta test <i>online</i> (aliran utama)....	182
136. Tabel 6.41a Kasus uji menghapus daftar peserta test (aliran utama).....	182
137. Tabel 6.41b Kasus uji menghapus daftar peserta test (aliran alternatif 1).	182
138. Tabel 6.42a Kasus uji mengubah data diri alumni (aliran utama).....	183
139. Tabel 6.42b Kasus uji mengubah data diri alumni (aliran alternatif 1)	183
140. Tabel 6.42c Kasus uji mengubah data diri alumni (aliran alternatif 2)	183
141. Tabel 6.43a Kasus uji ikuti <i>test online</i> (aliran utama).....	184
142. Tabel 6.43b Kasus uji ikuti <i>test online</i> (aliran alternatif 1).....	184
143. Tabel 6.43c Kasus uji ikuti <i>test online</i> (aliran alternatif 2).....	184
144. Tabel 6.43d Kasus uji ikuti <i>test online</i> (aliran alternatif 3).....	185
145. Tabel 6.44a Kasus uji mencari data alumni (aliran utama)	185
146. Tabel 6.44b Kasus uji mencari data alumni (aliran alternatif 1).....	185
147. Tabel 6.45a Kasus uji mencari data perusahaan (aliran utama)	186
148. Tabel 6.45b Kasus uji mencari data perusahaan (aliran alternatif 1)	186
149. Tabel 6.45b Hasil Pengujian Validasi.....	186



RINGKASAN

ACHMAD MUFTI. 2009. : Perancangan dan Pembuatan Sistem Informasi Job Placement Center di Politeknik Negeri Malang. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing : Suprpto, ST., MT dan Himawat Aryadita, ST., MT., M.Sc.

Job placement center merupakan salah satu bidang yang juga memperoleh dampak akibat berkembangnya internet. Banyak situs – situs yang menawarkan lowongan pekerjaan secara *online*. Dengan adanya lowongan pekerjaan secara *online* ini akan membantu para alumni untuk memperoleh pekerjaan sesuai dengan bidang yang diinginkan. Selain itu juga para alumni tidak harus datang ke perusahaan untuk memperoleh informasi lowongan pekerjaan, begitu juga bagi perusahaan tidak harus datang ketempat instansi-instansi untuk memperoleh informasi para alumni yang dibutuhkan oleh perusahaan.

Job placement center online dapat menggunakan teknologi ini untuk mempertemukan para alumni dengan perusahaan-perusahaan sebagai client. Dimana para alumni dapat memperoleh informasi lowongan pekerjaan dan mengirim berkas lamaran secara online serta mengikuti test perekrutan secara online. Bagi perusahaan dapat menginputkan informasi lowongan pekerjaan secara online serta melakukan test perekrutan secara online. sehingga data dapat diperoleh dengan cepat dan akurat.

Proses pengembangan perangkat lunak harus melalui suatu proses rekayasa perangkat lunak (RPL), begitu pula dengan perangkat lunak *job placement center online* yang ditujukan bagi para alumni akademik dan perusahaan sebagai client. Rekayasa perangkat lunak ini dimaksudkan untuk memberi landasan sehingga pengembangan perangkat lunak dilakukan secara sistematis dengan sasaran memperoleh produk akhir yang berkualitas. Perangkat lunak berkualitas tinggi adalah program yang handal, efisien, mudah dipahami, mudah dimodifikasi dan mudah dipelihara.

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*. 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*), dan database *MySQL*. Karena *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 *dcompile* oleh *web server* saat digunakan. 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*

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



BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada saat ini di Indonesia khususnya angka pengangguran setiap tahun semakin meningkat. Banyak faktor yang mempengaruhi angka pengangguran di Indonesia yang semakin lama terus bertambah diantaranya, kurangnya daya minat masyarakat untuk menempuh pendidikan serta minimnya informasi yang diperoleh masyarakat maupun para lulusan akademik tentang informasi-informasi lowongan pekerjaan dari instansi-instansi yang membutuhkan tenaga kerja. Rendahnya pendidikan serta minimnya orang-orang yang mempunyai keahlian dan keterampilan dibidang tertentu, maka mereka akan sulit untuk bersaing di dunia kerja dengan orang-orang yang memiliki pendidikan tinggi dan mempunyai keahlian serta keterampilan di dalam bidang tertentu.

Website merupakan jembatan bagi para pemburu informasi di dunia maya untuk berselancar di internet. Beberapa dekade terakhir perkembangan informasi melalui internet sangat pesat. Internet bukan lagi menjadi sebuah barang baru di masyarakat, namun sudah bergeser menjadi sebuah kebutuhan. Internet seperti halnya sebuah perpustakaan raksasa dengan website sebagai bukunya. Website tidak hanya dapat dijadikan sebagai media informasi, namun dapat juga digunakan sebagai media entertainment. Dari beberapa hal tersebut, kemampuan untuk dapat membangun sebuah situs web dirasa sangat perlu.

Dalam pembuatan *website* dapat menggunakan beberapa metode diantaranya dengan *system analyst*, *template designer*, *database designer*, dan *programmer*. Dalam pembuatan website memerlukan koordinasi dan komunikasi yang cukup baik. Untuk merancang dan mendesain website harus mempunyai wawasan dan informasi yang luas serta butuh keterampilan agar website yang kita buat dapat dikunjungi oleh pemakai jasa internet.



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* dan *MySQL* yang telah diuraikan menjadi alasan bagi penyusun untuk menggunakan *PHP* sebagai bahasa pemrograman untuk membuat *software job placement center online*.

Perusahaan dan para lulusan akademik merupakan salah satu bidang yang juga memperoleh dampak akibat berkembangnya internet. Banyak situs – situs perusahaan yang menawarkan lowongan pekerjaan secara *online*, tetapi belum terorganisasi dengan baik. Dengan ini kami akan merancang sebuah sistem informasi *job Placement center* yang dapat mempertemukan perusahaan dengan para lulusan akademik secara *online*. Metode semacam ini dapat menjadi solusi untuk membantu pihak perusahaan memperoleh tenaga kerja ahli yang sesuai dengan bidang yang dibutuhkan oleh perusahaan, serta pihak akademik dapat menyalurkan para lulusannya untuk mendapatkan pekerjaan di sebuah perusahaan. Begitu juga dengan para lulusan akademik dapat mengetahui perusahaan-perusahaan yang menyediakan lowongan pekerjaan sehingga para lulusan akademik dengan mudah mendapat informasi-informasi lowongan kerja serta dapat bekerja sesuai dengan minat dan kemampuan dibidangnya.

Dengan memanfaatkan sistem informasi *job placement center*, maka dapat membantu pemerintah untuk mengurangi jumlah angka pengangguran, kemiskinan, dan meningkatkan kualitas pendidikan.

1.2. Rumusan Masalah

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

1. Bagaimana cara membuat web yang dapat menghubungkan perusahaan dengan lulusan akademik dengan bahasa pemrograman *PHP dan MySQL* sebagai database?
2. Bagaimana cara merancang dan membuat *software Job Placement Center* secara *online*?
2. Bagaimana cara mengimplementasikan dan menguji *software Job Placement Center* secara *online*?

1.3 Batasan Masalah

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

1. Sistem Informasi ini dibuat dengan menggunakan sistem operasi Microsoft Windows XP Professional Service Pack 2.
2. Program aplikasi yang digunakan adalah Web Server Apache 2.2.4 (Win32), database MySQL version 5.0.37, PhpMyAdmin dan PHP Version 5.2.1 yang terbungkus dalam program XAMPP for Windows Version 1.7.0.
3. Browser yang digunakan adalah semua browser yang mendukung pengembangan bahasa HTML seperti: Internet Explore, Netscape, Opera, Neoplanet dan lain-lain.
4. Pembahasan difokuskan kepada pembuatan *software* yang dapat mempertemukan perusahaan dengan para lulusan secara *online*?
5. Pembahasan metode pembuatan *software Job Placement Center* secara *online*?difokuskan pada *PHP dan MySQL*.
6. *Software* yang dibuat berbasis *website*.

1.4. Maksud dan Tujuan

Tujuan penulisan adalah untuk merancang dan pembuatan software untuk menghubungkan perusahaan dengan lulusan akademik secara online, sehingga data dapat diakses oleh beberapa user yang diberi hak akses. Sistem informasi *job placement center* diharapkan dapat memberikan manfaat bagi penulis, Politeknik Negeri Malang, para alumni dan perusahaan sebagai client.

Manfaat untuk penulis antara lain:

- Dapat menerapkan ilmu yang diperoleh dari sistem perkuliahan pada Teknik Elektro Konsentrasi Sistem Informatika dan Komputer Universitas Brawijaya.
- Dapat merancang sistem informasi *job placement center* yang sesuai dengan tujuannya.

Manfaat untuk Politeknik Negeri Malang, Alumni dan Perusahaan:

- Dengan sistem informasi *job placement center* yang terpusat maka memudahkan pihak Politeknik Negeri Malang dalam melihat data alumni sesuai jurusan dan angkatan.
- Para alumni dapat memperoleh informasi tentang lowongan pekerjaan dari perusahaan-perusahaan sebagai client dari Politeknik Negeri Malang yang membutuhkan tenaga kerja.
- Bagi perusahaan dapat menginformasikan lowongan pekerjaan dan melihat data para lulusan yang dibutuhkan oleh perusahaan serta memudahkan perusahaan melakukan test perekrutan secara *online*.

1.5. Sistematika Penulisan

Penulisan skripsi ini akan diberikan uraian singkat dengan susunan penulisan sebagai berikut :

BAB I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat, dan sistematika penulisan.

BAB II Dasar Teori

Berisi tentang teori dasar dan teori penunjang mengenai perancangan sistem informasi *job placement center*, jaringan komputer, basis data dan teori dasar *web*.

BAB III Metodologi Penelitian

Berisi berbagai metode yang digunakan dalam penulisan skripsi dan pengerjaan perancangan, baik dalam tahap pengambilan data, proses perancangan, pengujian dan juga pengambilan kesimpulan.

BAB IV Analisa Kebutuhan dan Perancangan Perangkat Lunak

Berisi tentang pembahasan mengenai analisis kebutuhan, perancangan sistem, perancangan proses, perancangan basis data dan perancangan tampilan dari sistem informasi *job placement center*.

BAB V Implementasi Perangkat Lunak

Berisi tentang pembahasan mengenai implementasi perancangan sistem informasi *job placement center*.

BAB VI Pengujian Perangkat Lunak

Berisi tentang pembahasan mengenai pengujian implementasi sistem informasi *job placement center*.

BAB VII Kesimpulan dan Saran

Menjelaskan tentang beberapa kesimpulan dari hasil yang telah dicapai serta saran-saran untuk pengembangan software lebih lanjut.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai teori dasar yang digunakan untuk menunjang penulisan skripsi. Teori dasar yang digunakan yaitu, perancangan sistem informasi *Job Placement Center*, jaringan komputer, basis data dan teori dasar *web*. Teori dasar perancangan sistem informasi berbasis Client-server meliputi pemodelan sistem yang berupa *Use Case Diagram*, serta *Entity-Relationship Diagram (ER Diagram)*. Teori dasar jaringan komputer meliputi topologi jaringan, konsep TCP/IP dan manfaat dari jaringan komputer. Teori dasar basis data meliputi pengertian basis data, bahasa basis data SQL (*Structured Query Language*) dan basis data MySQL. Teori dasar *web* meliputi HTML (*Hyper Text Markup Language*) dan PHP (*PHP Hypertext Preprocessor*).

2.1 Sistem Informasi

Sistem informasi dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses, dan menyimpan serta mendistribusikan informasi [DHA-02:11]. Sistem informasi merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi.

Pembuatan sistem informasi memerlukan perangkat pemodelan sistem informasi. Perangkat pemodelan berfungsi sebagai media yang memberikan penjelasan tentang sistem informasi yang dibuat. Perangkat pemodelan dapat berupa diagram, *flowchart*, maupun gambar.

Tiga alasan yang menyebabkan kita sebaiknya melakukan pemodelan sistem, yaitu [POH-97:9] :

- Dapat memfokuskan perhatian pada hal-hal penting dalam sistem tanpa mesti terlibat terlalu jauh.
- Mendiskusikan perubahan dan koreksi terhadap kebutuhan pemakai dengan resiko dan biaya minimal.

- Menguji pengertian penganalisa sistem terhadap kebutuhan pemakai dan membantu pendesain sistem dan pemrogram membangun sistem.

Pemodelan sistem dapat dipresentasikan dengan berbagai cara antara lain, *Use-Case Diagram*, *Data Flow Diagram Context Level*, *Data Flow Diagram Levelled*, *State-Transition Diagram*, *Entity-Relationship Diagram (ER Diagram)* dan banyak model lainnya.

2.2 Dasar Teori

Berdasarkan sistem informasi di atas, maka untuk penyelesaian skripsi ini dibutuhkan dasar teori sebagai berikut :

1. Sekilas tentang Politeknik
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. Jaringan komputer
 - a. Topologi jaringan komputer
 - b. TCP/IP (*Transmission Control Protocol/Internet Protocol*)
 - c. Manfaat jaringan komputer
6. Basis data dan MySQL
 - a. *Entity-RelationshipDiagram* (Diagram E-R)
 - b. SQL (*Structured Query Language*)
 - c. Basis data MySQL
7. Teori dasar web
 - a. HTML (*Hyper Text Markup Language*)
 - b. PHP (*PHP Hypertext Preprocessor*)

2.2.1 Sekilas Tentang Politeknik

Politeknik Negeri Malang adalah salah satu bentuk perguruan tinggi yang mengemban tugas dan fungsi khusus yaitu menyelenggarakan pendidikan profesional dalam sejumlah bidang pengetahuan dan teknologi terapan dengan mengutamakan peningkatan kemampuan penerapannya. Menurut sejarah pendiriannya, Politeknik Negeri Malang yang sebelumnya bernama Politeknik Universitas Brawijaya merupakan satu dari 6 (enam) Politeknik perintis yang didirikan oleh Pemerintah Indonesia pada tahun 1982 dengan bantuan Bank Dunia. Dalam kedudukannya sebagai perguruan tinggi, Politeknik Negeri Malang merupakan bagian dari sistem pendidikan nasional yang bertujuan menyiapkan mahasiswa menjadi anggota masyarakat yang memiliki kemampuan profesional yang dapat menerapkan, mengembangkan dan menyebarkan ilmu pengetahuan dan teknologi serta mengupayakan penggunaannya untuk meningkatkan taraf kehidupan masyarakat dan kesejahteraan umat manusia serta memperkaya kebudayaan nasional.

Politeknik merupakan pendidikan profesional yang diarahkan pada kesiapan penerapan keahlian tertentu. Selain daripada itu, Politeknik merupakan perguruan tinggi yang menyelenggarakan pendidikan terapan dalam sejumlah bidang pengetahuan khusus. Guna mencapai maksud itu, Politeknik memberikan pengalaman belajar dan latihan yang memadai untuk pembentukan kemampuan profesional dibidang ilmu pengetahuan dan teknologi. Hal ini telah diwujudkan dalam komposisi jumlah jam pelajaran teori dan praktek dengan perbandingan 45% dan 55%, dalam masa pendidikan selama 6 semester yang menggunakan sistem paket. Disamping itu juga didukung oleh staf pengajar yang benar-benar profesional, yang berasal dari lulusan perguruan tinggi dalam negeri dan lulusan luar negeri (Amerika, Australia, Filipina, Inggris dan Swiss), sehingga Politeknik Negeri Malang merupakan sistem pendidikan profesional dan diharapkan mampu menghasilkan lulusan yang siap pakai. Hal ini sesuai dengan tuntutan sektor industri di negara kita dewasa ini. Sudah sewajarnya jika lulusan Politeknik Negeri Malang selalu menjadi incaran dunia industri.

Politeknik Negeri Malang menyelenggarakan program Diploma Tiga Politeknik. Politeknik ini mempunyai empat jurusan, yaitu:

1. Jurusan Teknik Elektronika
2. Jurusan Teknik Listrik
3. Jurusan Teknik Mesin
4. Jurusan Teknik Sipil

Mengingat kebutuhan industri yang semakin beragam dan pesatnya pembangunan nasional maka pada tahun 1986 Politeknik menambah tiga jurusan tahun 1987 menambah satu jurusan, dan mulai tahun 2004 ini Polinema menambah satu jurusan lagi, yaitu:

1. Jurusan Teknik Telekomunikasi
2. Jurusan Akutansi
3. Jurusan Administrasi Niaga
4. Jurusan Teknik Kimia
5. Jurusan Manajemen Informatika

Tujuan Politeknik adalah :

1. Terselenggaranya proses peningkatan mutu pendidikan dan pengajaran yang relevan dengan kebutuhan dunia usaha dan industri.
2. Terselenggaranya proses pendidikan yang didukung fasilitas akademik yang dapat dijadikan andalan proyeksi perkembangan jangka pendek dan jangka panjang.
3. Tersedianya sumber daya manusia untuk menyelenggarakan tugas fungsional sehingga mampu menghasilkan lulusan yang relevan dengan kebutuhan pasar.
4. Terselenggaranya bentuk penelitian yang bermutu dan bermanfaat untuk meningkatkan efisiensi dan produktivitas dalam proses produksi.
5. Terselenggaranya bentuk pengabdian kepada masyarakat yang bermanfaat dan dapat meningkatkan kesejahteraan masyarakat.
6. Terselenggaranya kerjasama kemitraan dengan organisasi profesi, industri dan instansi terkait untuk menghasilkan tenaga profesional.
7. Terciptanya sistem manajemen pendidikan yang dinamis dan berorientasi pada motifasi kerja.
8. Menyiapkan pemberdayaan otonomi Perguruan Tinggi.

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 pengubahan 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*).

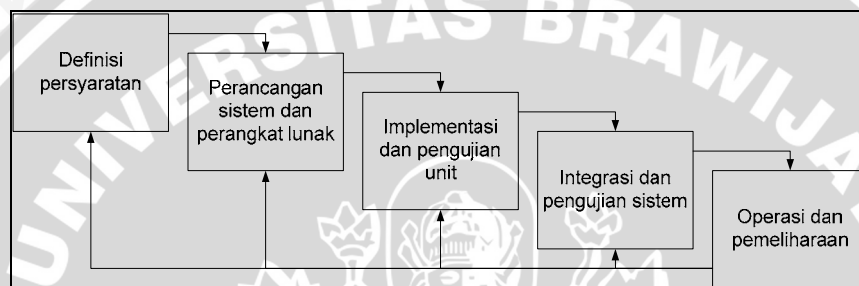
Model proses *waterfall* adalah model yang menyarankan pendekatan pengembangan secara sekuen dan sistemik untuk mengembangkan perangkat lunak [SOM-04]. Model proses *waterfall* dimulai dari proses analisis dan definisi persyaratan sampai dengan pemeliharaan. Tahap-tahap utama dari model *waterfall* adalah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

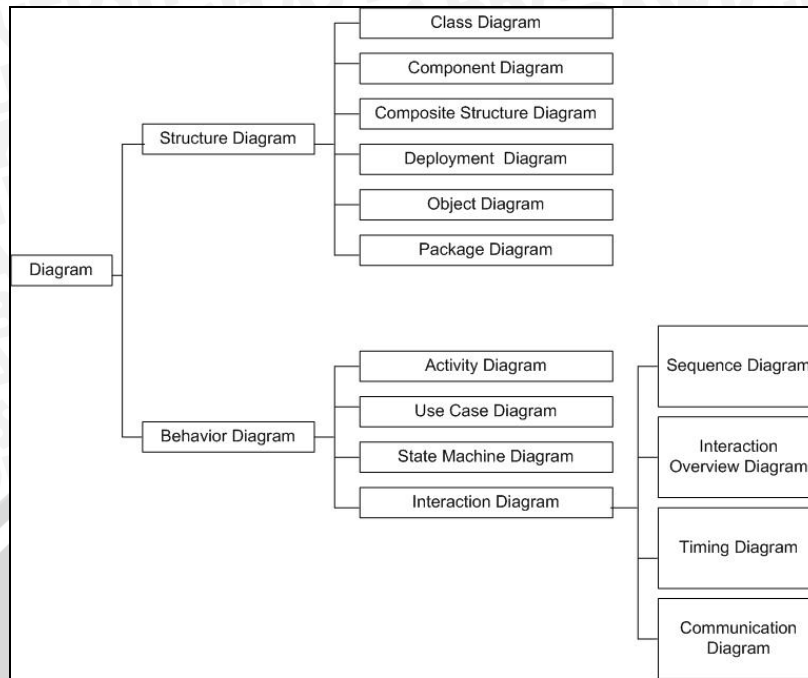
Paradigma rekayasa perangkat lunak adalah cara pandang spesifik dalam usaha pencarian solusi, yaitu menyangkut konseptualisasi permasalahan dan solusi yang tidak berkaitan dengan urutan aktivitas penyelesaian [HAR-04]. Paradigma berorientasi objek adalah paradigma yang berdasarkan pada pola pikir berorientasi 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 Kelas
3. Diagram *Sequence*

2.2.3.1 Diagram Use Case

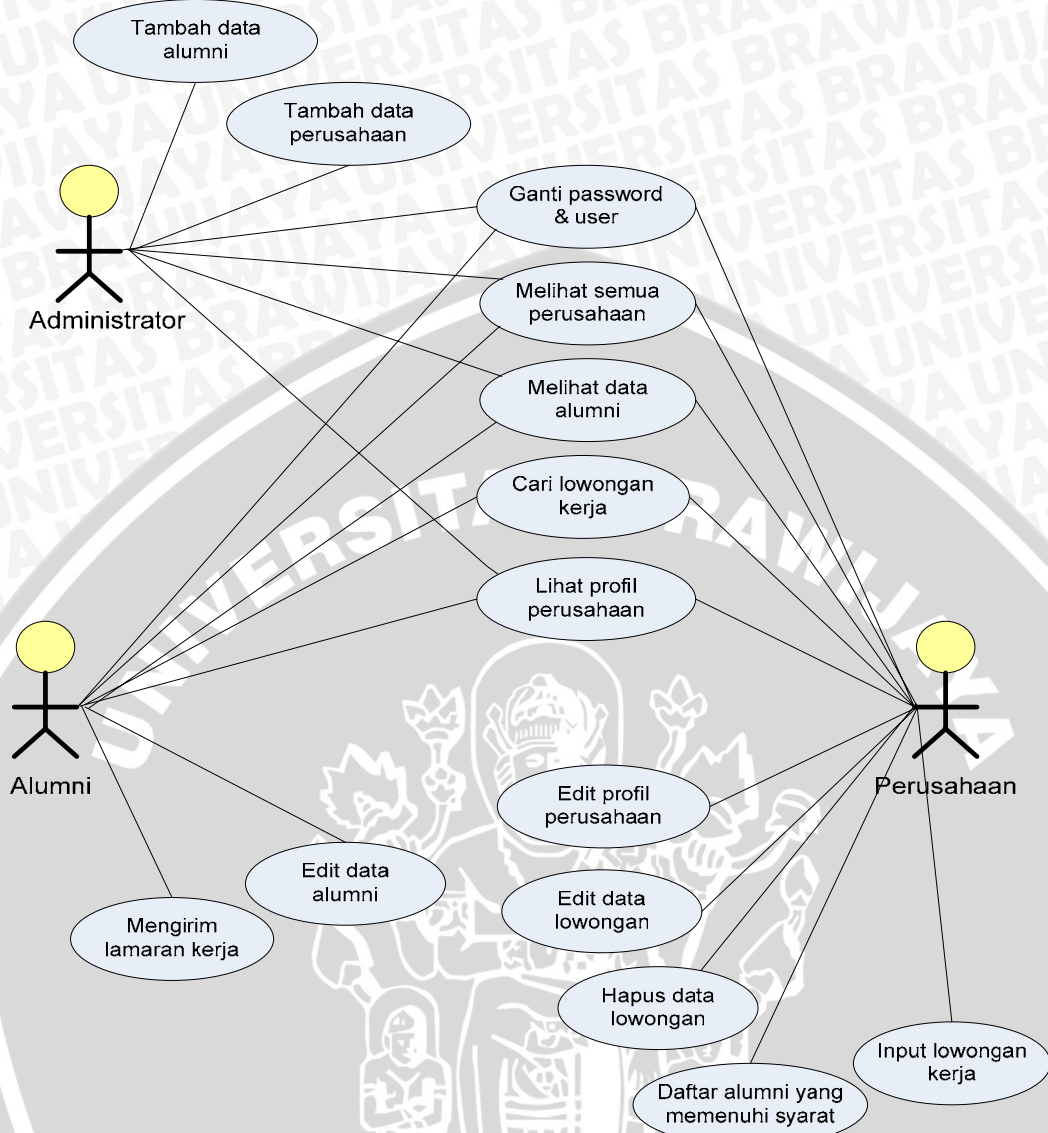
Diagram *use case* adalah suatu cara untuk menangkap fungsionalitas dan kebutuhan dari sistem dalam UML [PIL-05]. Diagram *use case* menekankan “apa” yang dapat diperbuat sistem, bukan “bagaimana” hal tersebut dilakukan. Diagram *use case* sangat membantu dalam penyusunan kebutuhan sebuah sistem, mengkomunikasikan 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 tiga buah aktor (Administrator, Alumni dan Perusahaan) dan delapan buah *use case* (Memasukkan dan memodifikasi data *job placement center*, melihat rekap data alumni dan perusahaan, memasukkan biodata alumni dan perusahaan). Aktor alumni dan perusahaan dapat mengakses *use case* memasukkan biodata. Aktor Administrator dapat mengakses semua *use case*. Subjek dari contoh ini adalah *software job placement center online*.





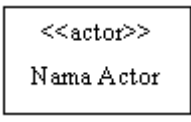
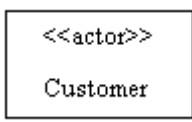

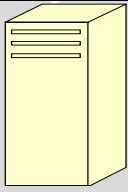
Gambar 2.3 : Contoh *Use Case Diagram*

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

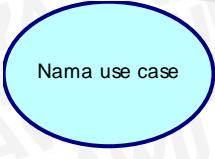

Sumber : [OMG-04]


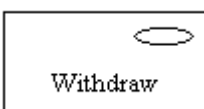
b) Use Case

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>		<i>Use case</i> dapat dinotasikan dengan gambar elips, dengan nama <i>use case</i> berada di dalam elips ataupun di bawah elips.	

2	Class rectangle	 Nama Use Case	Use case dapat pula dinotasikan dengan kotak klas dengan icon 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 :

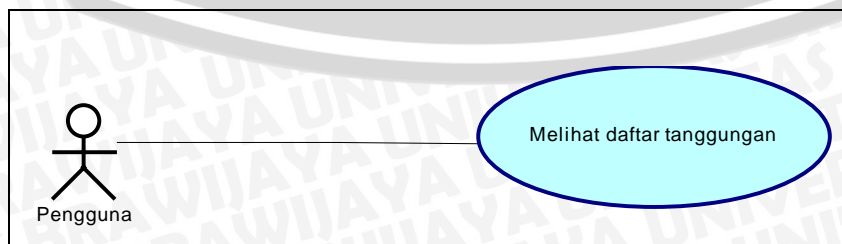
1. Asosiasi antara aktor dengan *use case*.
2. Generalisasi dari aktor.
3. Generalisasi dari *use case*.
4. *Include* antara dua buah *use case*.
5. *Extend* antara dua buah *use case*.

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*.



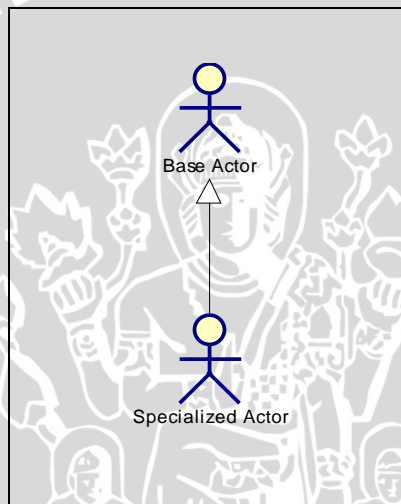
Gambar 2.4 : Asosiasi antara Aktor dengan *Use Case*



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].

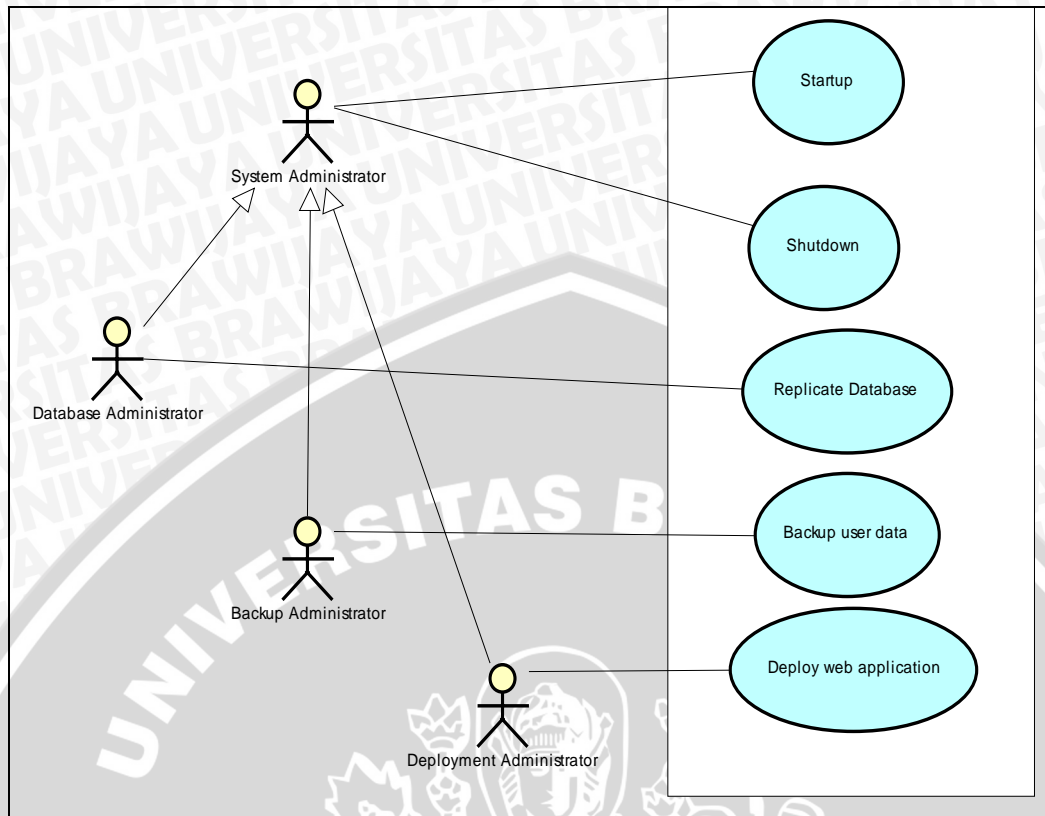
2. Generalisasi dari Aktor

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

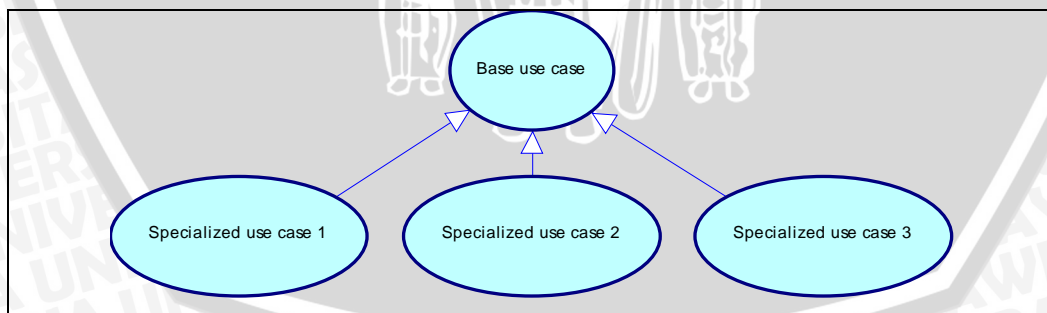
Gambar 2.5 memberikan contoh dari generalisasi aktor. Diagram berikut memiliki empat aktor, yaitu *System Administrator*, *Database Administrator*, *Backup Administrator*, dan *Deployment Administrator*. Aktor umum dari diagram berikut adalah *System Administrator*. Aktor selain *System Administrator* 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].



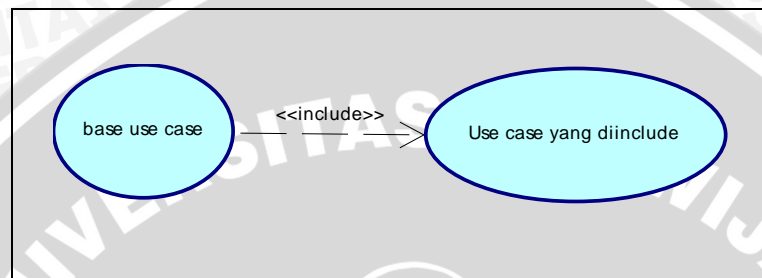
Gambar 2.7 : Generalisasi dari Use Case
 Sumber : [PIL-05]

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.

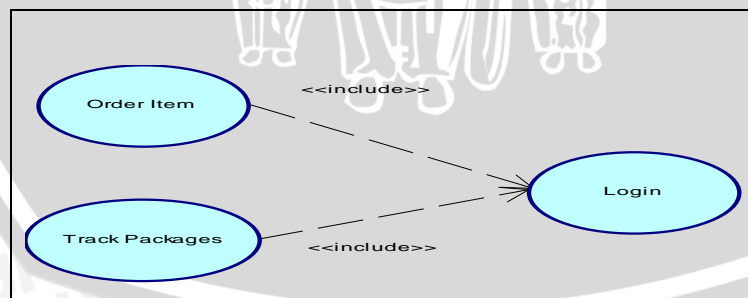
Include dinotasikan dengan anak panah dengan garis putus-putus dengan kepala panah terbuka, dimana anak panah menunjuk ke *use case* yang diinclude, dan pada badan panah diberi *keyword* <<include>> [OMG-04].



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*

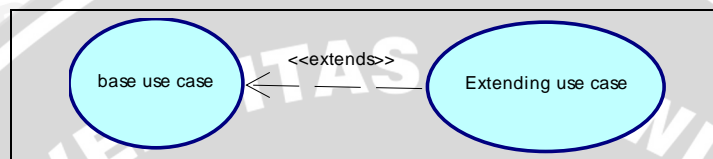
Sumber : [PIL-05]

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

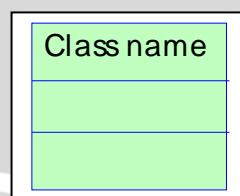
Diagram klas adalah diagram dalam UML yang digunakan untuk menangkap hubungan statis dari suatu perangkat lunak [PIL-05]. Diagram klas memiliki beberapa hal penting, yaitu klas dan *relationship* [BOO-05].

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
+ 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 menampakkan *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

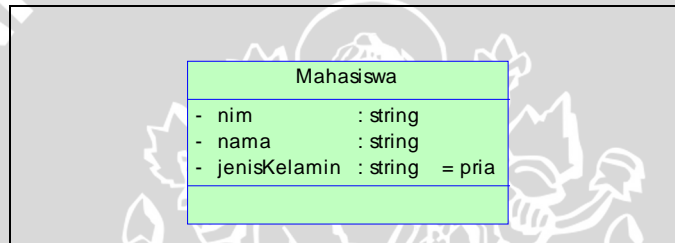
Atribut adalah properti dari suatu kelas yang mendeskripsikan suatu rentang nilai yang mungkin dimiliki oleh properti dari objek tersebut [BOO-05]. Atribut dapat dinotasikan dengan 2 cara, yaitu [PIL-05]:

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

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

visibility namaAtribut : type multiplicity = defaultValue

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



Gambar 2.13 : Contoh Notasi kelas 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 klas
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

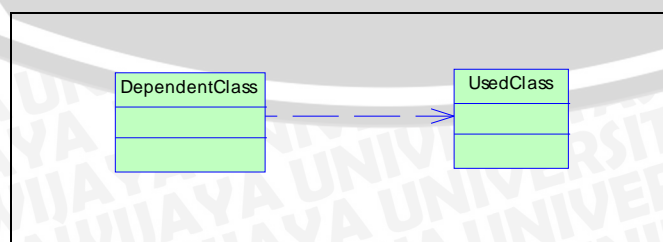
Relationship adalah hubungan antar klas yang satu dengan klas yang lain dalam diagram klas [BOO-05]. UML menyediakan beberapa cara untuk merepresentasikan *relationship* antar klas. Beberapa jenis *relationship* dalam diagram klas adalah :

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 klas [PIL-05].

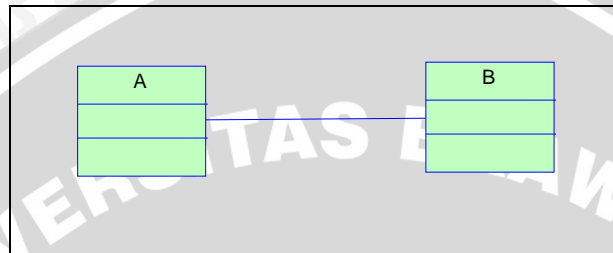
Dependensi dinotasikan sebagai anak panah dengan badan panah berupa garis putus-putus, kepala panah terbuka dan menunjuk dari klas *dependent* ke klas yang digunakan.



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

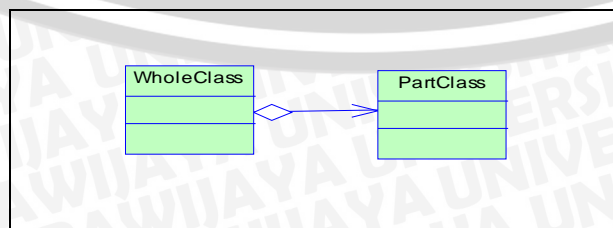
Sumber : [PIL-05]

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

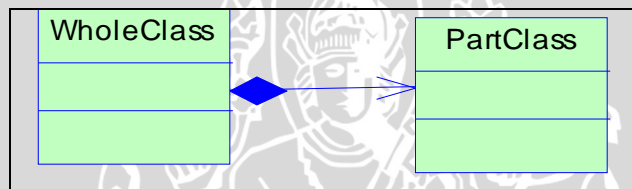
Sumber : [PIL-05]

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

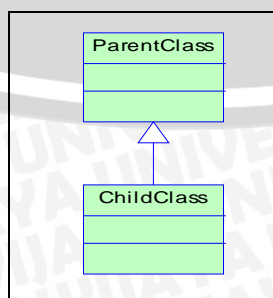


Gambar 2.17 : Notasi Komposisi
Sumber : [PIL-05]

5. Generalisasi

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



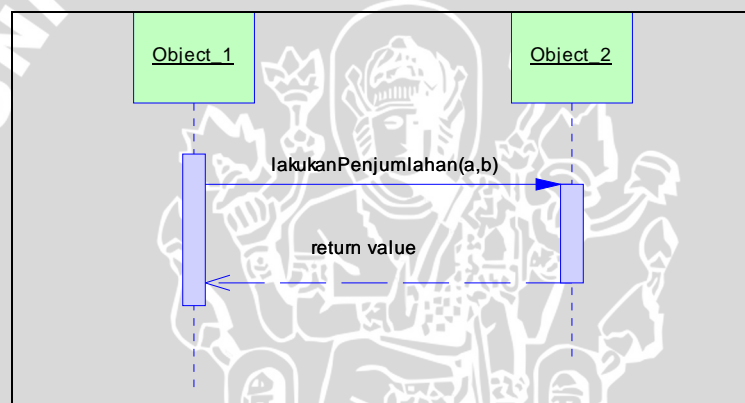
Gambar 2.18 : Notasi Generalisasi
Sumber : [PIL-05]

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.



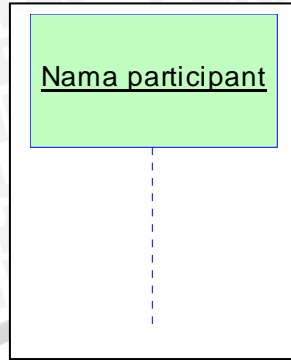
Gambar 2.19 : Contoh Diagram *Sequence*

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*.

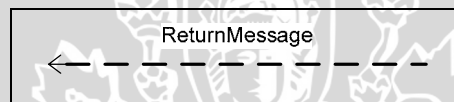


Gambar 2.22 : Notasi *Synchronous Message*
Sumber : [PIL-05]

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

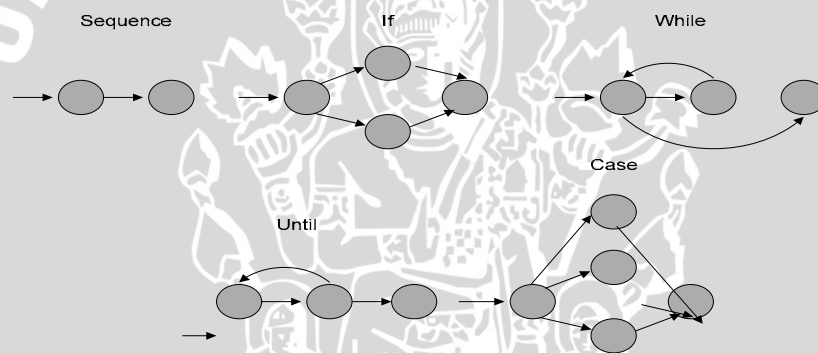
Pengujian nilai batas adalah suatu teknik pengujian yang digunakan untuk mengurangi jumlah kasus uji dengan tidak mengurangi mutu pengujian. Pengujian nilai batas memfokuskan pada nilai-nilai batas, karena pada nilai-nilai batas tersebut sering terjadi kesalahan. Bila suatu fungsi bekerja mulai nilai 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 Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer, dan peralatan lainnya yang terhubung dalam satu kesatuan [DHA-02:61]. Informasi dan data ditransmisikan melalui kabel-kabel atau tanpa kabel (*wireless*) sehingga memungkinkan pengguna komputer untuk bertukar informasi. Jaringan komputer juga dapat digunakan untuk mencetak dokumen pada printer yang sama. Setiap perangkat yang terhubung dalam jaringan dinamakan dengan *node*.

Jaringan komputer memerlukan penentuan terhadap topologi jaringan komputer yang akan digunakan. Hal ini untuk mendapatkan sebuah jaringan yang tepat dan hemat dalam pembuatannya.

2.2.5.1 Topologi Jaringan Komputer

Topologi adalah suatu cara untuk menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk sebuah jaringan. Setiap topologi yang digunakan untuk membentuk suatu jaringan mempunyai kelebihan dan kekurangan masing-masing. Beberapa jenis topologi antara lain [DHA-02:67] :

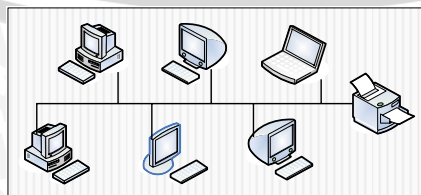
1. Topologi *Bus*

Topologi *Bus* menggunakan media kabel *coaxial* sebagai media transmisi yang ditunjukkan dalam Gambar 2.25. Keuntungan dari penggunaan topologi *bus* adalah:

- Hemat kabel.
- *Layout* kabel sederhana.
- Mudah dikembangkan.

Kekurangan penggunaan topologi *bus* adalah:

- Kesulitan untuk mendeteksi kerusakan, apabila terjadi kerusakan dalam jaringan.
- Kepadatan lalu lintas.
- Seluruh jaringan akan tidak berfungsi, jika terjadi kerusakan pada kabel utama.
- Diperlukan *repeater* untuk jarak jauh.



Gambar 2.25 Topologi Bus

Sumber: [DHA-02:67]

2. Topologi *Ring*

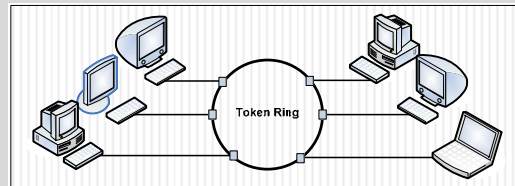
Topologi *ring* adalah cara menghubungkan komputer sehingga membentuk lingkaran seperti ditunjukkan dalam Gambar 2.26. Setiap informasi yang diperoleh diperiksa alamatnya oleh *terminal* yang dilewatinya. Jika informasi yang dikirim bukan untuk terminal tersebut maka informasi akan dilewatkan sampai menemukan alamat tujuan.

Keuntungan penggunaan *topologi ring* adalah:

- Mudah untuk mencari kerusakan dalam jaringan.
- Tidak terdapat *collision*.

Kekurangan dari penggunaan *topologi ring* adalah:

- Apabila satu komputer terganggu, komputer lainnya akan terganggu.
- Pengembangan jaringan lebih kaku.



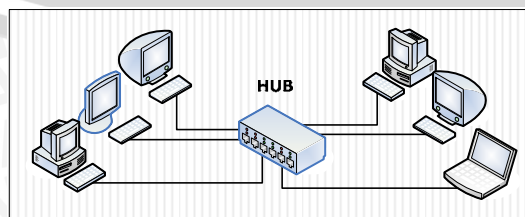
Gambar 2.26 Topologi Ring

Sumber: [DHA-02:68]

3. Topologi *Star*

Topologi *star* adalah cara menghubungkan komputer secara langsung pada *server* atau *hub* seperti yang ditunjukkan dalam Gambar 2.27. Karena setiap komputer yang memiliki hubungan dengan *server* melalui kabel tersendiri maka *bandwith* atau lebar jalur data dalam jaringan akan semakin lebar. Penggunaan *topologi star* akan meningkatkan kinerja jaringan secara keseluruhan. Keuntungan lain dari penggunaan topologi *star* adalah apabila terdapat gangguan di suatu jalur kabel maka jaringan tidak akan terganggu secara keseluruhan.

Topologi *star* memerlukan jumlah kabel yang lebih banyak sehingga biaya untuk menggunakan topologi ini juga akan meningkat.



Gambar 2.27 Topologi Star

Sumber: [DHA-02:67]

2.2.5.2 TCP/IP (*Transmission Control Protocol/Internet Protocol*)

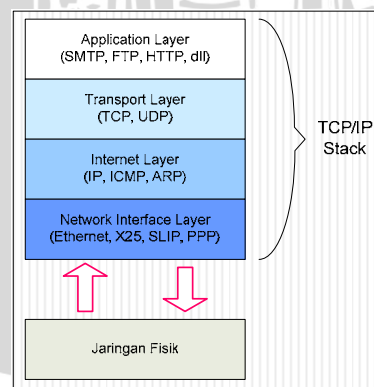
TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data *Wide Area Network* (WAN). TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung-jawab atas bagian-bagian tertentu dari komunikasi data. Prinsip ini menjelaskan tugas masing-masing protokol secara sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain selama masih bisa saling mengirim dan menerima data.

Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan. Sebagian besar isi kumpulan protokol TCP/IP tidak spesifik terhadap satu komputer atau peralatan tertentu. TCP/IP dapat berjalan di atas *interface* tertentu dengan melakukan perubahan pada protokol yang berhubungan dengan *interface* jaringan.

TCP/IP terdiri atas empat lapis kumpulan protokol yang bertingkat. Keempat lapis / *layer* tersebut adalah:

- *Network Interface Layer*
- *Internet Layer*
- *Transport Layer*
- *Application Layer*

Protokol TCP/IP dimodelkan dengan empat *layer* TCP/IP yang ditunjukkan dalam Gambar 2.28.



Gambar 2.28 *Layer* TCP/IP

Sumber: [ONN-01:23]

TCP/IP melakukan penyampaian data dari protokol yang berada di satu *layer* ke protokol yang berada di *layer* yang lain. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data.

Jika suatu protokol menerima data dari protokol lain di *layer* atasnya maka protokol penerima data akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu, data akan diteruskan ke protokol pada *layer* di bawahnya.

Hal yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada *layer* di bawahnya. Jika data valid maka protokol akan melepas informasi tambahan tersebut. Protokol tersebut kemudian meneruskan data itu ke protokol lain yang berada pada *layer* di atasnya.

Network Interface Layer bertanggung-jawab mengirim dan menerima data ke dan dari media fisik. Media fisiknya dapat berupa kabel, serat optik, atau gelombang radio. Protokol pada *layer* ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer.

Internet Layer bertanggung-jawab dalam proses pengiriman paket ke alamat yang tepat. Pada *layer* ini terdapat tiga macam protokol yaitu IP, ARP, dan ICMP.

IP (*Internet Protocol*) berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan untuk menentukan alamat *hardware* dari *host*/komputer yang terletak pada *network* yang sama. ICMP (*Internet Control Message Protocol*) ialah protokol yang digunakan untuk mengirimkan pesan dan melaporkan kegagalan pengiriman data.

Transport Layer berisi protokol yang bertanggung-jawab untuk mengadakan komunikasi antara dua *host*/komputer. Kedua protokol tersebut adalah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*) [ONN-01:22-25].

TCP (*Transmission Control Protocol*) merupakan protokol yang menyediakan *service* yang dikenal sebagai *connection oriented*, *reliable*, *byte stream service*. *Connection oriented* berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan hubungan (*handshake*) terlebih dahulu. *Reliable* berarti TCP menerapkan proses deteksi kesalahan paket dan retransmisi. *Byte stream service* berarti paket dikirimkan dan sampai ke tujuan secara berurutan [ONN-01:51].

UDP (*User Datagram Protocol*) merupakan protokol transport yang sederhana. Berbeda dengan TCP yang connection oriented, UDP bersifat connectionless. UDP tidak terjadi pengurutan kembali (sequencing) paket yang datang, acknowledgement terhadap paket yang datang, atau retransmisi jika paket mengalami masalah dalam pengiriman [ONN-01:55].

Pada *Application layer* terletak semua aplikasi yang menggunakan protokol TCP/IP [ONN-01:25].

2.2.5.3 Manfaat Jaringan Komputer

Beberapa manfaat dengan adanya jaringan komputer antara lain [DHA-02:72] :

1. *Resource Sharing*

Jaringan komputer dapat digunakan untuk menggunakan sumber daya yang ada secara bersama-sama.

2. Reliabilitas Tinggi

Jaringan komputer dapat digunakan untuk menyediakan penyimpanan data alternatif.

3. Menghemat Uang

Penggunaan jaringan komputer dapat menghemat pengeluaran pada suatu organisasi. penghematan ini disebabkan karena manfaat dari penggunaan sumber daya secara bersama-sama.

2.2.6 Basis Data

Basis data, menurut Stephen dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basis data, dimodifikasi, dan dihapus [SIM-06].

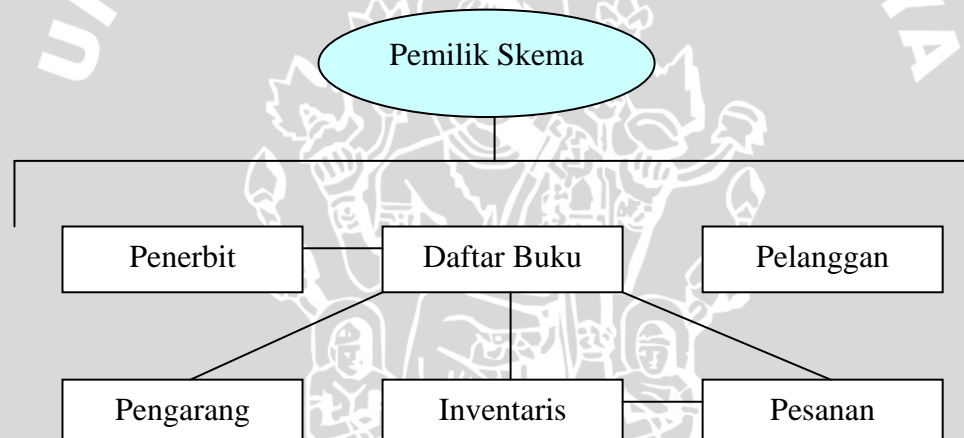
Basis Data mempunyai komponen penyusun sebagai berikut:

- Skema basis data

Pada dasarnya, skema adalah sekelompok obyek dalam basis data yang saling berhubungan. Dalam skema, obyek yang dihubungkan memiliki relasi satu sama lain. Ada satu pemilik skema, yaitu yang memiliki akses memanipulasi struktur semua obyek dalam skema.

- Tabel

Tabel adalah unit penyimpanan fisik utama untuk data dalam basis data. Ketika mengakses basis data, pengguna mengacu tabel untuk data yang diinginkan. Basis data bisa terdiri dari banyak tabel sehingga harus terdapat relasi antar tabel. Karena menyimpan data, tabel membutuhkan penyimpanan fisik pada host komputer untuk basis data. Berikut contoh skema basis data seperti Gambar 2.29.



Gambar 2.29 Contoh Skema Dalam Basis Data

Sumber: [SIM-06]

- Kolom

Kolom atau field adalah kategori informasi yang terdapat dalam tabel. Dengan kata lain, ketika model bisnis diubah menjadi model basis data, entitas menjadi tabel dan atribut menjadi kolom. Entitas adalah obyek yang dapat direalisasikan dalam dunia nyata, misalnya mahasiswa, mata kuliah, ruang kuliah, fakultas, dan lain-lain. Sedangkan atribut adalah sifat-sifat yang melekat dalam entitas tersebut. Kolom adalah struktur penyimpanan logis dalam sebuah basis data. Masing-masing kolom dalam tabel memiliki tipe data. Jika tabel divisualisasikan, sebuah kolom adalah struktur vertikal dalam tabel dan mengandung nilai untuk setiap baris data yang berhubungan dengan kolom tersebut.

- Baris

Satu baris data adalah kumpulan semua kolom yang berhubungan dengan kejadian tunggal. Dengan kata lain, satu baris data adalah satu record dalam tabel. Sebagai contoh, jika 25.000 judul buku, akan ada 25.000 record atau baris dalam tabel judul buku. Jumlah baris dalam tabel akan berubah ketika judul buku ditambah atau dikurangi.

- Tipe Data

Tipe data menentukan jenis data yang dapat disimpan dalam kolom suatu tabel. Meskipun ada banyak tipe data, hanya tiga tipe data yang umum digunakan, yaitu: Alfanumerik, numerik, dan tanggal dan waktu. Tipe data alfanumerik digunakan untuk menyimpan karakter, angka, karakter khusus, atau hampir semua kombinasi. Jika nilai numerik disimpan dalam kolom alfanumerik, maka nilai ini diperlakukan sebagai karakter, bukan angka. Dengan kata lain, Anda tidak boleh melakukan fungsi aritmatika pada nilai numerik yang disimpan dalam field alfanumerik. Tipe data numerik digunakan hanya untuk menyimpan nilai numerik. Tipe data tanggal dan waktu digunakan untuk menyimpan nilai tanggal dan waktu.

2.2.6.1 Entity-Relationship Diagram (Diagram E-R)

Diagram *Entity-Relationship* menjelaskan semesta data yang ada pada dunia nyata. Diagram *Entity-Relationship* menterjemahkan atau mentransformasikan data dengan memanfaatkan sejumlah perangkat konseptual [POH-97:35]. Komponen-komponen pembentuk diagram *Entity-Relationship*, antara lain:

1. Entitas dan Himpunan Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas. Entitas menunjuk pada individu suatu objek, sedangkan himpunan entitas menunjuk pada rumpun dari individu tersebut. Himpunan entitas digambarkan dengan persegi panjang dalam diagram *Entity-Relationship*.

2. Atribut

Entitas pasti memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal penting lain dalam pembentukan model data. Atribut digambarkan dengan elips dalam diagram *Entity-Relationship*.

3. Relasi dan Himpunan Relasi

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan entitas tersebut membentuk suatu himpunan relasi. Himpunan relasi digambarkan dengan belah ketupat dalam diagram *Entity-Relationship*.

4. Kardinalitas atau Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain. Kardinalitas relasi yang terjadi di antara dua himpunan entitas (misal A dan B) dapat berupa:

- Satu ke Satu (*One to One*)

Entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B dan sebaliknya.

- Satu ke Banyak (*One to Many*)

Entitas pada himpunan entitas A dapat 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.

- Banyak ke Satu (*Many to One*)

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.

- Banyak ke Banyak (*Many to Many*)

Entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B dan sebaliknya.

Diagram E-R selalu dibuat bertahap. Tahapan pertama yang dapat dilakukan adalah dengan membuat diagram E-R awal. Tujuan dari pentahapan ini adalah untuk mendapatkan sebuah rancangan basis data minimal yang dapat mengakomodasi kebutuhan penyimpanan data terhadap sistem yang sedang ditinjau. Langkah-langkah yang harus dilakukan antara lain:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut key dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada beserta *foreign key*-nya.
4. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non-key*).

2.2.6.2 SQL (*Structured Query Language*)

SQL adalah sebuah bahasa basis data standar untuk membuat, meng-*update*, dan melakukan *query* pada sistem manajemen basis data relasional. SQL mengalami beberapa perubahan. SQL mula-mula dibuat pada tahun 1970 dengan Sequel. Standarisasi yang pertama dibuat pada tahun 1986 oleh ANSI (*The American National Standard Institute*) dan ISO (*International Standard Organization*), yang disebut sebagai SQL-86. SQL-86 diperbarui pada tahun 1989 menjadi SQL-89. Standar terakhir yang dibuat dan digunakan hingga saat ini adalah SQL-92 yang dikeluarkan pada tahun 1992 [FAT-02:220]. SQL dapat digunakan untuk membuat sebuah *query* dan digabungkan dalam sebuah aplikasi sebagai instruksi-instruksi yang digunakan untuk menangani data di dalam basis data.

2.2.6.3 MySQL

MySQL merupakan *Database Management System* (DBMS) yang bersifat *open source* dan dikembangkan, serta didistribusikan oleh MySQL AB [RAM-05]. MySQL memiliki bagian berupa *Structured Query Language* (SQL) yang digunakan untuk mengolah database-database relasional yang ada di dalamnya. Sejak dahulu, MySQL merupakan pasangan serasi PHP dalam pembuatan web

yang dinamis. MySQL dapat digunakan untuk membangun aplikasi basis data berbasis *client-server*. MySQL memiliki beberapa keistimewaan antara lain [PRA-03:2]:

- *Portability*
MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Solaris* dan lain sebagainya.
- *Open Source*
MySQL didistribusikan secara *open source* sehingga digunakan secara gratis.
- *Multi-User*
MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- *Performance Tuning*
MySQL memiliki kecepatan yang tinggi dalam menangani sebuah *Query*.
- *Column Types*
MySQL mendukung tipe kolom yang banyak di antaranya *signed/unsigned integer, float, double, char, varchar*.
- *Command dan Functions*
MySQL mendukung penuh perintah *select* dan *where* dalam *query*.
- *Scalability dan Limits*
MySQL mampu menangani basis data dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.
- *Interface*
MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- *Connectivity*
Client dapat terhubung dengan MySQL Server menggunakan *socket* TCP/IP dari *platform* apapun. Pada sistem operasi Microsoft Windows keluarga NT (NT, 2000, XP, atau 2003), *client* dapat terhubung dengan *pipe*. Pada sistem Unix, *client* dapat terhubung dengan menggunakan *file-file socket domain* Unix.

- Antarmuka *Connector*/ODBC (MyODBC)
MyODBC memberikan dukungan MySQL untuk program-program *client* yang menggunakan koneksi ODBC (*Open Database Connectivity*).

2.2.6.2.1 Jenis-jenis Kolom Pada MySQL

Jenis kolom merupakan rincian dari jenis data yang diterapkan pada setiap kolom pada saat membuat sebuah tabel dengan perintah `CREATE TABLE`. Jenis kolom numerik pada basis data MySQL ditunjukkan dalam Tabel 2.5.

Tabel 2.5 Jenis Kolom Numerik

Jenis Kolom	Bentuk Dasar Penulisan	Keterangan	Ukuran
TINYINT	TINYINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -128 s/d 127 <i>Unsigned</i> : 0 s/d 255	1 byte
SMALLINT	SMALLINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -32768 s/d 32767 <i>Unsigned</i> : 0 s/d 65535	2 byte
MEDIUMINT	MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -8388608 s/d 8388607 <i>Unsigned</i> : 0 s/d 1677215	3 byte
INT	INT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -128 s/d 127 <i>Unsigned</i> : 0 s/d 255	4 byte
BIGINT	BIGINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -9223372036854775808 s/d 9223372036854775807 <i>Unsigned</i> : 0 s/d 18446744073709551615	8 byte
FLOAT	FLOAT[(M, D)] [UNSIGNED] [ZEROFILL] atau FLOAT(4)	Bilangan pecahan presisi tunggal dengan kisaran nilai: Nilai minimum (tanpa nilai nol): $\pm 1.175494351E-38$ Nilai maksimum (tanpa nilai nol): $\pm 6.402823466E+38$	4 byte
DOUBLE	DOUBLE[(M, D)] [UNSIGNED] [ZEROFILL] atau FLOAT(8)	Bilangan pecahan presisi ganda dengan kisaran nilai: Nilai minimum (tanpa nilai nol): $\pm 2.2250738580720E-308$ Nilai maksimum (tanpa nilai nol): $\pm 1.7976931348623157E+308$	8 byte
DECIMAL	DECIMAL[(M, D)]	Bilangan desimal, dengan nilai tergantung pada besaran M dan D.	M byte

Sumber: [ARB-04:71-72]

Jenis kolom tanggal dan waktu pada basis data MySQL ditunjukkan dalam Tabel 2.6.

Tabel 2.6 Jenis Kolom Tanggal dan Waktu

Jenis Kolom	Keterangan	Ukuran
DATE	Digunakan untuk tanggal dengan format YYYY-MM-DD.	3 byte
DATETIME	Digunakan untuk tanggal dan waktu dengan format YYYY-MM-DD hh:mm:ss.	8 byte
TIME	Digunakan untuk waktu dengan format hh:mm:ss.	3 byte
TIMESTAMP	Digunakan untuk penulisan waktu dan tanggal, dengan format YYYYMMDDhhmmss.	4 byte
YEAR	Digunakan untuk penulisan tahun dengan format YYYY.	1 byte

Sumber: [ARB-04:80]

Jenis kolom Karakter basis data MySQL ditunjukkan dalam Tabel 2.7.

Tabel 2.7 Jenis Kolom Karakter

Jenis Kolom	Bentuk Dasar Penulisan	Keterangan	Ukuran
CHAR	Char (M)	Data string dengan lebar data yang tetap (M).	M byte
VARCHAR	Varchar (M)	Data string dengan lebar data bervariasi (M).	M byte

Sumber: [ARB-04:75]

Jenis kolom dapat dijelaskan sebagai berikut (Kode M merupakan nilai maksimum yang diperkenankan (1-255) dan D merupakan jumlah digit setelah tanda titik pada bilangan pecahan (kisaran nilai 0-30) [ARB-04:70].

2.2.7 Teori Dasar Web

Aplikasi berbasis web membutuhkan bahasa pemrograman dan program editor HTML agar web lebih dinamis dan interaktif. Bahasa pemrograman yang digunakan adalah HTML (Hyper Text Markup Language), PHP (PHP Hypertext Preprocessor).

2.2.7.1 HTML (*Hyper Text Markup Language*)

HTML adalah suatu format data yang digunakan untuk membuat dokumen *hypertext* yang dapat dibaca dari satu *platform* komputer ke *platform* komputer lain tanpa perlu melakukan perubahan apapun. Dokumen HTML disebut *markup language* karena mengandung tanda-tanda tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen [ANO-04:6].

HTML merupakan bahasa standar untuk membuat tampilan *web*. HTML tidak dibuat untuk mempublikasikan informasi di halaman *web*. HTML kemudian dipilih untuk mendistribusikan informasi di halaman *web* karena kemudahan dan kesederhanaan dalam penggunaannya [SYU-99:9].

HTML adalah sebuah *plain text* yang berisi HTML *tags* dan *user-defined text fields*. Perintah-perintah HTML ditandai dengan menggunakan *tag* yang berupa karakter “<” dan “>” dan disimpan dalam file berekstensi *.html.

Struktur HTML pada dasarnya dibagi menjadi dua bagian, yaitu *header* dan *body*. Masing-masing ditandai dengan pasangan *container tag* <head> </head> dan <body> </body>. Bagian *head* berisi judul dokumen dan informasi-informasi dasar lainnya, sedangkan bagian *body* berisi data dokumennya. Pengaturan format teks dan pembentukan *links* dilakukan terhadap obyeknya langsung dengan ditandai oleh *tag-tag* HTML. Contoh perintah-perintah HTML:

```
<html>
<head>Tulisan ini muncul juga lho...
<title>Teks ini ditampilkan di Title bar-nya Browser </title>
</head>
<body>
Teks (isi web) + script diletakkan disini didalam tag body
</body>
</html>
```

File HTML hanya berisi teks, tetapi dapat disisipi *tag* untuk *image*, *sound*, animasi dan objek ActiveX. File multimedia tersebut berdiri sendiri dan harus di-*upload* juga bersama file html-nya. File HTML dapat dibuka atau dilihat dengan aplikasi internet *browser* seperti Microsoft Internet Explorer, Netscape, Mozilla Fireworks, Opera dan lain-lain.

HTML diatur dalam *World Wide Web Consortium (W3C)*. Perubahan atas standar bahasa HTML harus disahkan dahulu oleh konsorsium ini.

2.2.7.2 PHP (*PHP Hypertext Preprocessor*)

Aplikasi berbasis *web* dibangun tidak cukup hanya menggunakan HTML. HTML memiliki kelemahan yaitu tidak dapat mengakses dan memanipulasi data di dalam *server* basis data. Kelemahan HTML dapat diatasi dengan bahasa pemrograman yang mempunyai kemampuan untuk mengakses dan memanipulasi data di dalam *server* basis data [KRI-05].

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada *server*, sedangkan yang dikirimkan ke *browser* hanya hasilnya saja [WID-03:1].

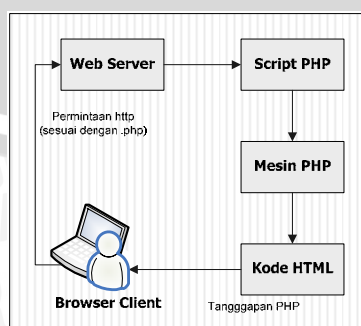
PHP dirancang secara khusus untuk membentuk *web* dinamis. PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini. PHP mempunyai fungsi yang sama dengan *script-script* seperti ASP (*Active Server Page*), Cold Fusion ataupun Perl.

PHP dirancang pada awalnya untuk diintegrasikan dengan *web server* Apache. Namun belakangan, PHP juga dapat bekerja dengan *web server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Services*) dan Xitami (Kadir, 2001:2).

Untuk dapat lebih memahami tentang PHP, maka akan dibahas lebih lanjut mengenai konsep kerja PHP, variabel pada PHP, *Script* PHP, *tag* PHP, *cookie*, *session* dan keunggulan dari PHP itu sendiri.

a. Konsep Kerja PHP

Model kerja PHP diawali dengan permintaan suatu halaman *web* oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari berkas yang diminta dan apabila sudah didapatkan maka *web server* akan mengirimkannya ke mesin PHP [KAD-01:5]. Mesin inilah yang kemudian akan memproses dan memberikan hasilnya (berupa kode HTML) ke *web server*. Selanjutnya *web server* akan menyampaikan hasil tersebut ke *client*. Skema konsep kerja PHP ditunjukkan dalam Gambar 2.30.



Gambar 2.30 Skema Konsep Kerja PHP

Sumber: [KAD-01:5]

b. Variabel Pada PHP

Setiap variabel dalam PHP selalu dimulai dengan tanda dolar (\$) dan harus dimulai dengan huruf atau garis bawah (_) dan kemudian dapat diikuti oleh huruf, angka ataupun garis bawah. `$warpspeed`, `$impuls_speed`, `$LCAR` dan `$Dilithium1` adalah contoh penamaan variabel PHP yang benar.

Jenis suatu variabel pada PHP ditentukan pada saat jalannya program (*runtime*). PHP tidak memerlukan pendeklarasian variabel terlebih dahulu karena PHP mempunyai kemampuan untuk membedakan jenis variabel secara otomatis berdasarkan konteks yang sedang berlaku bagi variabel tersebut dan tipe datanya bisa diubah sesuai dengan keinginan.

PHP mendukung beberapa jenis variabel sebagai berikut:

1. *Integer*

Variabel berjenis *integer* bertujuan untuk menyimpan bilangan bulat.

2. *Double*

Untuk menyimpan bilangan bernilai pecahan dan juga bilangan pemangkatan.

3. *String*

String merupakan jenis data karakter yang disimpan sebagai nomor pada memori komputer. Nilai yang disimpan adalah nilai ASCII karakter *string* tersebut.

4. *Array*

Array adalah sebuah set variabel yang mempunyai jenis data sama. *Array* mengandung komponen yang disebut elemen dan disimpan pada lokasi tertentu pada memori.

5. *Object*

Jenis variabel objek adalah berdasarkan gambaran objek pada dunia nyata yang mempunyai status dan tingkah laku. Sebuah variabel objek menyimpan statusnya dalam bentuk variabel dan tingkah lakunya dalam bentuk parameter.

c. *Script PHP*

Terdapat dua cara yang sering digunakan untuk menulis *script* PHP, yaitu:

1. *Embedded Script*

Cara ini dilakukan dengan meletakkan *script* PHP di antara *tag-tag* HTML.

```
<html>
<head>
<title> embedded script </title>
</head>
<body>
<? echo "Ini adalah contoh embedded script"; ?>
</body>
</html>
```

2. *Non-Embedded Script*

Non-embedded script merupakan pembuatan program murni PHP, dimana *tag-tag* HTML yang diletakkan didalamnya. Berikut contoh penggunaannya:

```
<? echo"<html>";
echo"<head>";
echo"<title> non embedded script </title>";
echo"</head>";
echo"<body>";
echo"Ini adalah contoh non embedded script";
echo"</body>";
echo"</html>";
?>
```

d. *Tag PHP*

Penulisan program PHP dapat dilakukan dengan menggunakan berbagai *tag*. Hal ini tidak akan mempengaruhi hasil program yang dibuat [PRA-03:6].

Tag yang dapat digunakan adalah sebagai berikut:

1. *Style Standar*

Style standar PHP sangat mirip dengan penulisan program XML, yakni diawali dengan `<?php` dan diakhiri dengan `?>`. Contoh penulisannya adalah

```
<html>
<head>
<title> style standar </title>
</head>
<body>
<?php echo "Ini adalah contoh style standar PHP"; ?>
</body>
</html>
```


2. Short Style

Penulisan *script* pada *style* ini cukup praktis bila dibandingkan dengan *style* sebelumnya karena hanya diawali dengan `<?` dan diakhiri dengan `?>`.

```
<html>
<head>
<title> short style </title>
</head>
<body>
<? echo "Ini adalah contoh short style PHP"; ?>
</body>
</html>
```

3. Style JavaScript

Penulisan programnya diawali dengan `<SCRIPT LANGUAGE = 'PHP'>` dan diakhiri dengan `</SCRIPT>`.

```
<html>
<head>
<title> style JavaScript </title>
</head>
<body>
<SCRIPT LANGUAGE = 'PHP'>
echo "Ini adalah contoh style Javascript";
</SCRIPT>
</body>
</html>
```

4. Style ASP

Diawali dengan `<%` dan diakhiri dengan `%>`. Contoh penulisannya adalah:

```
<html>
<head>
<title> style ASP </title>
</head>
<body>
<% echo "Ini adalah contoh style ASP"; %>
</body>
</html>
```

e. Cookie

Cookie adalah sepotong data yang disimpan pada *harddisk* lokal milik pengunjung (*client*) dan digunakan oleh halaman *web* dalam mengingat sesuatu informasi [KAD-03:407]. Data pada *harddisk* lokal ini dikenali oleh *server* untuk mendapatkan kembali informasi yang pernah dikirimkan ke *client*.

Cookie bersifat sementara, artinya ketika *browser* ditutup atau waktu penyimpanan *cookie* tersebut telah habis. *Cookie* juga akan secara otomatis dihapus. Dengan demikian, ketika *browser* kembali dibuka atau melewati jangka waktu tertentu, maka nilai *cookie* tersebut akan kosong.

Sebuah *client* hanya dapat memegang 300 *cookie* pada satu saat dan sebuah *server* hanya dapat mengirim 20 *cookie* ke sebuah *client*. Sebuah *cookie* hanya bisa berukuran sampai dengan 4 kilobyte, sehingga total memori *harddisk* yang digunakan *cookie* hanya mencapai 1,2 megabyte.

Web server mengirim *cookie* ke *client* melalui judul HTTP, yang dikirim sebelum teks HTML. Melalui judul pada HTTP inilah *client* bisa mengetahui apakah ia perlu mengirimkan *cookie* yang ada pada *harddisk* lokal ke *web server* atau tidak. Seperti halnya dengan *web server*, *client* juga mengirimkan kembali *cookie* ke *web server* melalui judul HTTP. Di dalam judul HTTP, *cookie* diatur melalui *set-cookie*. *set-cookie* mempunyai informasi nama *cookie* dan nilainya, batas kadaluarsa, *path*, *domain* dan parameter keamanan. Informasi nama variabel dan nilai dari *cookie* ditunjukkan dalam Tabel 2.8.

Tabel 2.8 Parameter *Cookie*

Informasi	Keterangan
<i>Path</i>	Digabungkan dengan <i>domain</i> , nilai <i>path</i> menentukan direktori pada <i>web server</i> yang dapat menggunakan <i>cookie</i> . Jika informasi <i>path</i> dan URL yang diminta tidak cocok, maka <i>client</i> tidak akan mengirimkan <i>cookie</i> . Nilai bawaan pada <i>path</i> berupa “/” yang berarti bahwa <i>cookie</i> valid untuk semua direktori pada <i>web server</i> .
<i>Domain</i>	Menyatakan <i>domain</i> (alamat) <i>server</i> yang mendefinisikan <i>cookie</i> . <i>Client</i> tidak akan mengirimkan <i>cookie</i> apabila <i>path</i> dan URL yang diminta tidak cocok dengan <i>domain</i> ini. Salah satu pemanfaatannya adalah untuk berbagi <i>cookie</i> pada beberapa <i>server</i> .
<i>Expires</i>	Menyatakan batas waktu kadaluarsa. Bawaannya, <i>cookie</i> hanya berlaku sampai <i>browser</i> ditutup.
<i>Secure</i>	Untuk menentukan pengiriman <i>cookie</i> hanya apabila protokol HTTPS (<i>HTTP Secure</i>) digunakan.

Sumber: [KAD-03:408-409]

f. *Session*

Session dapat digunakan untuk membentuk interaksi antara sebuah *client* dengan *web server* dalam selang waktu tertentu. PHP menyediakan pustaka yang berguna untuk membentuk sebuah *session* dengan menggunakan *session*. Berbeda dengan *cookie* yang menyimpan data pada *client*, *session* diimplementasikan dengan menyimpan data pada *server*. Dengan demikian tidak perlu ada komunikasi bolak-balik antara *web server* dan *client* ketika *web server* membutuhkan data tersebut.

Session akan dimulai ketika *client* mulai masuk sebuah situs dan akan berakhir begitu *client* tersebut menutup halaman situs yang telah dibukanya. *Client* akan mendapat variabel yang terus ada selama ia melakukan kunjungannya tersebut. Setiap kali suatu *session* dibentuk, maka akan terdapat referensi yang menunjuk ke *session* yang bersangkutan. Referensi ini dikenal dengan SID (*Session Identifier*). SID ini akan disimpan sebagai *cookie* apabila fasilitas *cookie* dalam keadaan dihidupkan pada *browser*, sedangkan data *session*-nya akan disimpan di *server* [KAD-03:437].

g. Keunggulan PHP

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya. Kelebihan PHP yaitu [SYU-99:36] :

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* bisa berkonsentrasi langsung pada penampilan dokumen *web*-nya.
2. Tidak ada proses *compiling* dan *linking* dan berorientasi obyek (*object oriented*).
3. Sintaks pemrogramannya mudah dipelajari, sangat menyerupai C dan Perl.
4. Integrasi yang sangat luas ke berbagai *server* basis data. Menulis *web* yang terhubung ke basis data menjadi sangat sederhana. Basis data yang didukung oleh PHP antara lain Oracle, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, dBase, UNIX dbm dan sebagainya.

BAB III

METODOLOGI 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 dalam penulisan ini dimaksudkan untuk mempelajari teori dari komponen yang akan digunakan dalam penulisan ini, studi literatur dalam penulisan ini disusun sebagai berikut:

1. Teori dasar RPL, meliputi :
 - a. Proses-proses RPL
 - b. Model proses perangkat lunak
 - c. Paradigma RPL
2. UML, meliputi :
 - a. *Use case diagram*
 - b. *Class diagram*
 - c. *Sequence diagram*
3. Pengujian
 - a. Teknik pengujian
 - b. Strategi pengujian
4. Basis data dan MySQL
 - d. *Entity-RelationshipDiagram* (Diagram E-R)
 - e. SQL (*Structured Query Language*)
 - f. Basis data MySQL
8. Teori dasar web
 - c. HTML (*Hyper Text Markup Language*)
 - d. PHP (*PHP Hypertext Preprocessor*)

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 keperpustakaan. Basis data keperpustakaan 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 XAMPP for Windows Version 1.6.1, Macromedia Dreamweaver MX, MySQL-Front, DBDesigner 4, Opera versi 9.5 Beta 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

3.5 Pengambilan Kesimpulan dan Saran

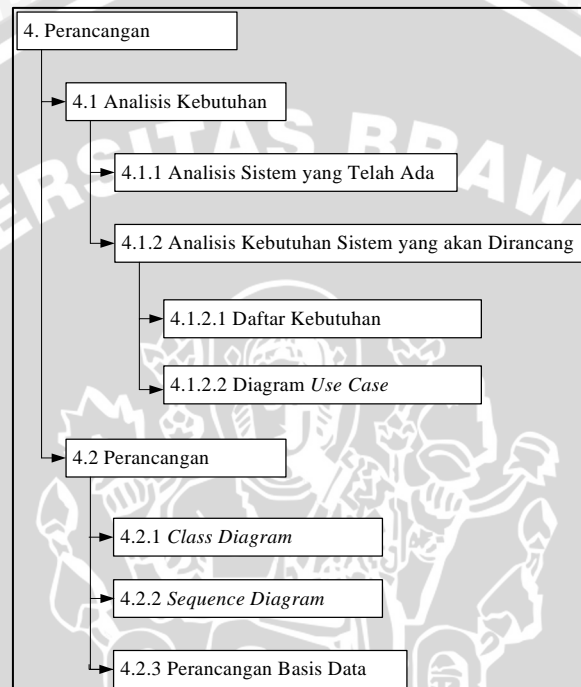
Tahap berikutnya dari penulisan adalah pengambilan kesimpulan dari perangkat lunak yang dibuat. Pengambilan kesimpulan ini didasarkan pada kesesuaian antara sistem informasi yang dihasilkan terhadap rumusan masalah dan tujuan awal dari perancangan sistem tersebut.

Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penulisan.

BAB IV

ANALISIS KEBUTUHAN DAN 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 bahasa pemodelannya.

4.1 Analisis Kebutuhan

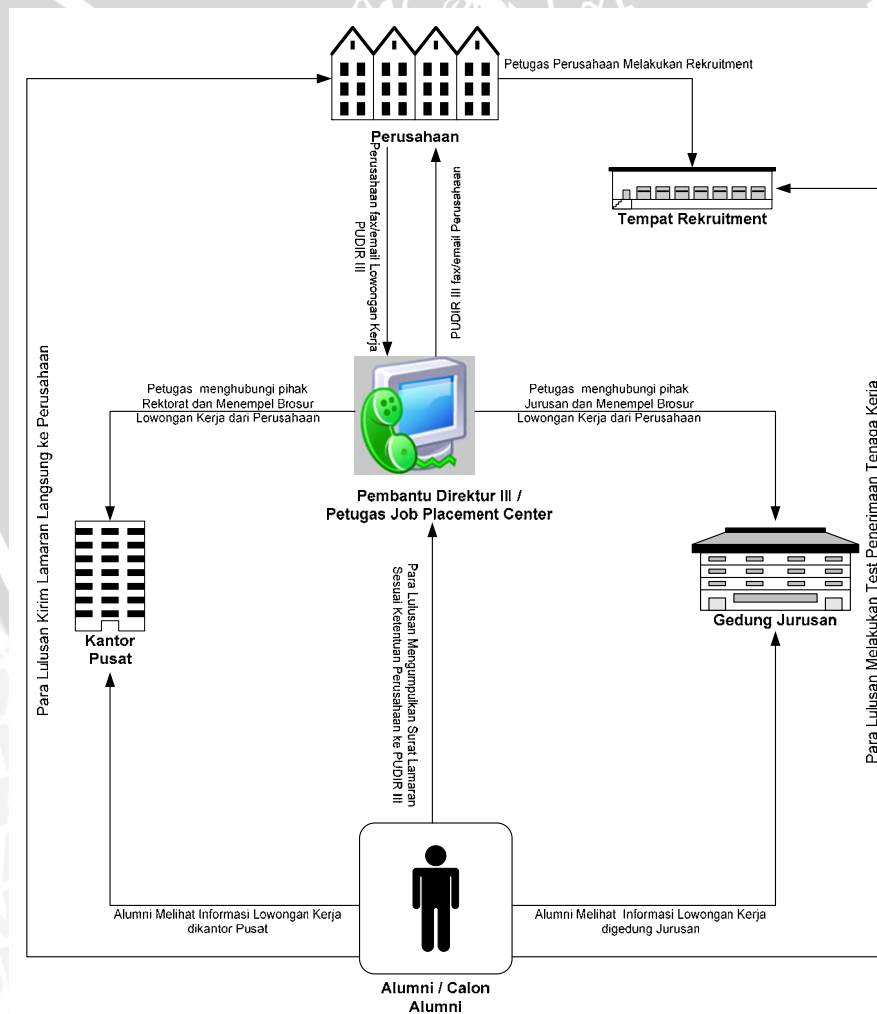
Analisis kebutuhan perangkat lunak adalah aktifitas rekayasa perangkat lunak yang menjembatani antara kebutuhan ditingkat sistem yang sudah ada dengan yang dirancang pada perancangan perangkat lunak [PRE-02]. Analisis kebutuhan perangkat lunak dapat juga diartikan proses yang digunakan untuk mendapatkan, 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 penginformasian yang telah ada selama ini yaitu dengan cara pihak perusahaan menghubungi perusahaan-perusahaan, atau sebaliknya pihak perusahaan yang menghubungi perusahaan-perusahaan yang membutuhkan tenaga kerja. Ilustrasi *job placement center* saat ini dapat dilihat pada Gambar 4.2.

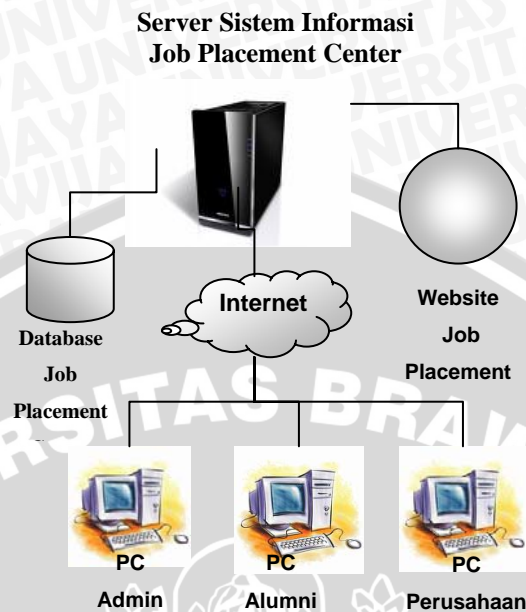


Gambar 4.2 : Blok Diagram Sistem Informasi *Job Placement Center*

Job placement center online memiliki kelebihan antara lain:

1. Politeknik Negeri Malang dapat menyalurkan para alumninya untuk memperoleh pekerjaan sesuai bidangnya dengan cepat.
2. Politeknik Negeri Malang dapat menginformasikan info-info terbaru dengan cepat.
3. Politeknik Negeri Malang dapat melihat data para alumni sesuai jurusan dan angkatan dengan cepat dan akurat.
4. Politeknik Negeri Malang dapat bekerjasama dengan perusahaan-perusahaan sebagai client dengan cepat.
5. Alumni dapat mengetahui informasi-informasi lowongan pekerjaan dengan cepat dan akurat.
6. Alumni dapat mengetahui biodata para alumni sesuai dengan jurusan dan angkatan dengan cepat.
7. Alumni dapat mengetahui informasi-informasi terbaru dari Politeknik Negeri Malang dengan cepat.
8. Alumni dapat mengetahui biodata perusahaan-perusahaan yang menjadi client Politeknik Negeri Malang dengan cepat.
9. Alumni dapat mengirim berkas lamaran pekerjaan secara online dengan cepat dan tepat.
10. Perusahaan dapat melihat para alumni sesuai dengan jurusan dan angkatan serta biodata para alumni.
11. Perusahaan dapat mengetahui berita-berita terbaru Politeknik Negeri Malang.
12. Perusahaan dapat menginformasikan lowongan-lowongan pekerjaan dengan cepat.
13. Perusahaan dapat mengetahui alumni-alumni yang memenuhi persyaratan sesuai dengan kebutuhan perusahaan dengan cepat dan akurat.
14. Perusahaan dapat menginputkan soal-soal untuk test online sesuai dengan perusahaan masing-masing.
15. Perusahaan dapat melakukan test perekrutan tenaga kerja secara *online* dengan cepat.

Perancangan *job placement center* setelah menggunakan perangkat lunak secara *online* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Arsitektur Jaringan (*Network*) Sistem Informasi *Job Placement Center*
Sumber: Perancangan *Job Placement Center* Politeknik Negeri Malang

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 *job placement center* agar dapat membantu para alumni untuk mendapatkan pekerjaan, perusahaan dapat memperoleh tenaga kerja yang sesuai dengan kebutuhan perusahaan.

4.1.2.1 Daftar Kebutuhan

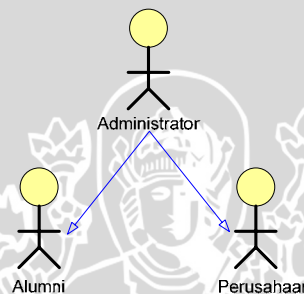
Daftar kebutuhan merupakan daftar yang menguraikan kebutuhan-kebutuhan pengguna yang harus disediakan oleh perangkat lunak baik kebutuhan fungsional maupun non fungsional. Proses yang dilakukan sebelum menentukan daftar kebutuhan adalah mengidentifikasi aktor yang menggunakan *software Job Placement Center* secara *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	Administrator	Aktor yang bertugas mengadministrasi penginputan data alumni dan perusahaan serta memberikan login dan <i>password</i> untuk alumni dan perusahaan
2	Alumni	Aktor yang menggunakan sistem <i>job placement center</i> untuk mengetahui informasi lowongan pekerjaan dari perusahaan serta mengirim berkas lamaran secara <i>online</i> .
3	Perusahaan	Aktor yang bertugas untuk menginputkan informasi lowongan pekerjaan dan melakukan test secara <i>online</i> .

Sumber : [Analisis]

Aktor administrator adalah petugas dari *job placement center*. Gambar 4.4 adalah *use case diagram* untuk generalisasi dari keseluruhan aktor dari Tabel 4.1.

**Gambar 4.4** : Generalisasi Aktor-Aktor Pengguna Sistem *Job Placement Center Online*

Sumber : [Analisis]

Penyusunan daftar kebutuhan fungsional dari sistem *job placement center online* dilakukan setelah identifikasi aktor. Daftar kebutuhan fungsional sistem disertai 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	<i>Use case</i>
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 data alumni. Data yang harus dicatat dari input data alumni adalah no.induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto, <i>username</i> , <i>password</i> .	Input data alumni

4	Sistem harus menyediakan fasilitas untuk menambah data perusahaan pengguna jasa. Data yang harus dicatat dari input data perusahaan adalah nama perusahaan, alamat, email, telepon, foto, <i>username</i> , <i>password</i> .	Input data perusahaan
5	Sistem harus menyediakan fasilitas untuk input berita tentang Politeknik Negeri Malang. Data yang harus dicatat dari input berita adalah judul, isi berita, foto.	Input berita
6	Sistem harus menyediakan fasilitas untuk mereset <i>password</i> apabila user lupa akan <i>password</i> nya. Data yang harus diinputkan adalah nama user dan <i>password</i> yang baru.	Mereset <i>password</i>
7	Sistem harus menyediakan fasilitas untuk mengedit data alumni yang telah dimasukkan. Data yang dapat diedit adalah no.induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto.	Mengedit data alumni
8	Sistem harus menyediakan fasilitas untuk mengganti <i>password</i> alumni yang terdiri dari <i>password</i> lama, <i>password</i> baru.	Mengganti <i>password</i> alumni
9	Sistem harus menyediakan fasilitas untuk mengedit data perusahaan yang telah dimasukkan. Data yang dapat diedit adalah nama, alamat, email, foto, telepon.	Mengedit data perusahaan
10	Sistem harus menyediakan fasilitas untuk mengganti <i>password</i> untuk user perusahaan yang terdiri dari <i>password</i> lama, <i>password</i> baru.	Mengganti <i>password</i> perusahaan
11	Sistem harus menyediakan fasilitas untuk input lowongan pekerjaan. Data yang harus dicatat dari input lowongan kerja adalah posisi, deskripsi lowongan, batas waktu, jurusan, IPK minimum, batas umur.	Menginput data lowongan pekerjaan
12	Sistem harus menyediakan fasilitas untuk menginputkan kategori soal dan soal-soal untuk aktor perusahaan itu sendiri. Data yang harus dicatat dari input soal adalah pertanyaan, pilihan A, pilihan B, Pilihan C, pilihan D, dan jawaban.	Menginput kategori soal dan soal-soal
13	Sistem harus dapat menampilkan daftar perusahaan-perusahaan <i>client</i> . Data yang ditampilkan berupa id, nama perusahaan, alamat, foto, telepon.	Melihat daftar profil perusahaan
14	Sistem harus menyediakan fasilitas untuk mengedit data perusahaan <i>client</i> yang telah dimasukkan. Data yang dapat diedit adalah posisi, deskripsi lowongan, batas waktu, jurusan, IPK minimum, batas umur.	Mengedit data informasi lowongan
15	Sistem harus menyediakan fasilitas untuk menghapus informasi lowongan pekerjaan yang sudah expired.	Menghapus lowongan pekerjaan
16	Sistem harus menyediakan fasilitas untuk mencari data para alumni yang terdaftar di dalam sistem <i>job placement center</i> . Pencarian data alumni berdasarkan nama, alamat, jurusan, angkatan dan email.	Mencari data alumni

17	Sistem harus menyediakan fasilitas untuk mencari daftar perusahaan yang terdaftar di dalam sistem <i>job placement center</i> . Pencarian daftar perusahaan berdasarkan nama, alamat dan email.	Mencari daftar perusahaan
18	Sistem harus menyediakan fasilitas untuk menampilkan daftar berita-berita yang diinputkan oleh administrator.	Melihat daftar berita
19	Sistem harus menyediakan fasilitas untuk menampilkan data alumni yang terdaftar dalam <i>job placement center</i> . Data yang ditampilkan adalah no induk nama, tempat lahir, tanggal lahir, jenis kelamin, alamat, agama, alumnus, angkatan, jurusan, tanggal lulus, IPK, lokasi kerja, email, telepon, foto.	Melihat biodata para alumni yang terdaftar
20	Sistem harus menyediakan fasilitas untuk melihat daftar perusahaan yang menyediakan lowongan pekerjaan. Data yang harus ditampilkan adalah nama perusahaan, posisi, deskripsi lowongan, batas waktu, jurusan, IPK minimum, batas umur.	Melihat daftar perusahaan yang menyediakan lowongan
21	Sistem harus menyediakan fasilitas untuk meng uplode berkas lamaran untuk dikirim ke perusahaan yang menyediakan lowongan pekerjaan.	Meng uplode berkas lamaran
22	Sistem harus menyediakan fasilitas untuk melakukan test secara <i>online</i> terhadap alumni yang memenuhi persyaratan administratif.	Melakukan test secara online

Sumber : [Analisis]

Dari proses kegiatan yang telah dijelaskan, terdapat tiga pengguna yang dapat mengakses sistem informasi *Job Placement Center*. Tiga diantaranya mempunyai hak untuk *login* dan melakukan pemrosesan data dalam basis data sesuai otoritasnya. Tiga pengguna itu adalah Administrator, Alumni dan Perusahaan. Sedangkan masyarakat selaku pengguna umum tidak diijinkan untuk *login* dan hanya mempunyai hak untuk melihat profil Politeknik Negeri Malang.

Administrator dalam sistem ini mempunyai beberapa otoritas, yaitu:

- Menginput data para alumni.
- Melihat seluruh data dalam basis data kecuali data *password user*.
- Melihat data para alumni secara detail
- Mengedit dan menghapus biodata alumni
- Menginput data perusahaan sebagai client.
- Melihat, mengedit dan menghapus data profil perusahaan
- Menginput berita-berita.
- Melihat, mengedit dan menghapus daftar berita
- Mereset *password* dari semua user.

Alumni mempunyai otoritas sebagai berikut:

- Melihat seluruh data para alumni.
- Melihat berita dan profil Politeknik Negeri Malang.
- Meng-*update* data pribadi dan mengubah *password*-nya.
- Menampilkan semua data perusahaan sebagai client.
- Melihat informasi-informasi lowongan pekerjaan dari semua perusahaan.
- Mengirim berkas lamaran kerja keperusahaan yang menyediakan lowongan pekerjaan.
- Mengikuti test secara *online*.

Perusahaan mempunyai otoritas sebagai berikut:

- Mengubah data diri perusahaan.
- Menginput data lowongan pekerjaan
- Melihat lowongan pekerjaan dari perusahaan secara detail
- Mengedit dan menghapus data lowongan.
- Menginput kategori soal
- Melihat, mengedit dan menghapus kategori soal
- Menginput soal-soal untuk *test online*.
- Melihat, mengedit dan menghapus soal-soal.
- Menginput peserta *test online*
- Melihat dan menghapus peserta *test online*

Setiap data yang dimasukkan oleh *user* akan membawa identitas pengisi data yang berupa *id_user* yang bersifat unik untuk setiap *user*. Penghapusan dan perubahan data oleh operator dapat dilakukan bila data tersebut diisikan oleh *administrator*. Dengan kata lain, data yang dimasukkan oleh seorang administrator tidak dapat dihapus atau diubah oleh operator lain.

Untuk penambahan *user* baru, identitas *user* baru tersebut akan ditambahkan ke dalam basis data oleh Administrator. Setelah itu, *user* baru tersebut akan mendapatkan *username* dan *password* sesuai permintaan (asalkan belum digunakan oleh *user* lain) dari Administrator. Setelah itu *user* baru tersebut harus melakukan *login* dan mengganti *password*-nya. Hal ini dilakukan agar Administrator tidak mengetahui *password user* baru tersebut, sehingga otorisasi penggunaan web tetap terjaga.

Administrator mengisikan data utama yang berupa data kode alumni atau perusahaan dan data *user*. Alumni dan perusahaan berhak memasukkan biodata kedalam database. Semua operator berhak melihat seluruh biodata alumni dan perusahaan dalam basis data.

Daftar kebutuhan non fungsional *software job placement center online* pada Tabel 4.3.

Tabel 4.3 : Daftar Kebutuhan Non Fungsional

No	Kebutuhan non fungsional
1	Sistem dikembangkan dengan menggunakan bahasa php.
2	Sistem dikembangkan dengan menggunakan <i>mysql</i> sebagai <i>database</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].

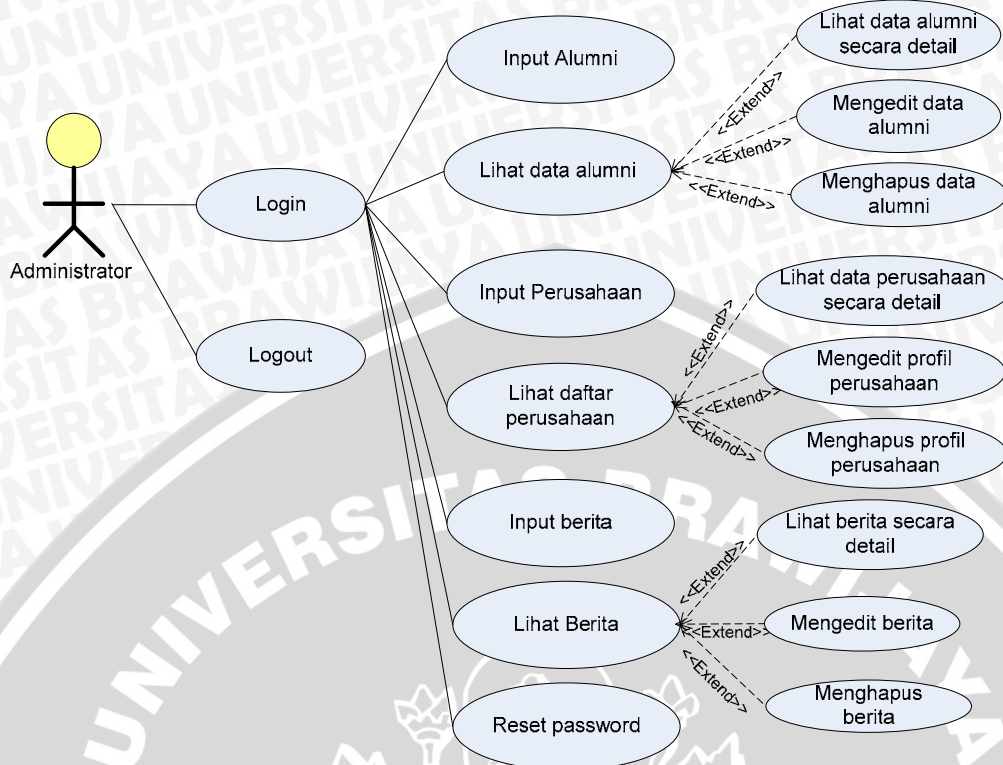
Pemodelan dalam *use case diagram* yang menggambarkan fungsionalitas yang disediakan oleh sistem *job placement center online* dibagi menjadi tiga buah diagram yang bersesuaian dengan modul dalam sistem *job placement center*

4.1.2.2.1 Use Case Diagram untuk Modul Administator Sistem Job Placement Center Online

Gambar 4.5 adalah *use case diagram* untuk modul administrator pendukung *software* sistem *job placement center online*. Diagram *use case* ini melibatkan 3 buah aktor dan 18 buah *use case*. Kedelapan belas *use case* yang termasuk dalam modul pendukung *software* sistem *job placement center online*.

Ketiga aktor yang terlibat dalam modul pendukung *software* sistem *job placement center online* adalah :

1. Administrator
2. Alumni
3. Perusahaan



Gambar 4.5 : Use Case Diagram untuk Administrator Sistem Job Placement Center Online
 Sumber : [Analisis]

1. Use Case Specification Login

Tabel 4.4 : Use Case Specification Login

Nama use case	Login
Aktor	Administrator
Deskripsi	Sistem harus mampu memberikan fasilitas untuk login, sehingga hanya administrator yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu (sebagai suatu aktor).
Pra-kondisi	Sistem telah berjalan.
Pasca-kondisi	User telah login sebagai 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. (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 login gagal, silahkan mengulangi kembali.
	2. Kembali pada langkah nomer 1 aliran utama.

Sumber : [Analisis]

2. Use Case Specification Logout

Tabel 4.5: Use Case Specification Logout

Nama use case	Logout
Aktor	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 Alumni

Tabel 4.6: Use Case Specification Menambah Data Alumni

Nama use case	Menambah data alumni
Aktor	Administrator



Deskripsi	Sistem harus menyediakan fasilitas untuk menambah Alumni. Data yang harus dicatat dari penambahan alumni adalah no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto <i>username</i> , <i>password</i> .
Pra-kondisi	Administrator telah menjalankan <i>use case</i> input alumni
Pasca-kondisi	Data alumni telah ditambahkan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih <i>link</i> "Input Alumni".	2. Sistem menampilkan <i>form</i> masukan yang berupa <i>field</i> no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto <i>username</i> , <i>password</i> .
3. Administrator memasukkan data no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto <i>username</i> , <i>password</i> .	4. Sistem memvalidasi masukan no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto <i>username</i> , <i>password</i> .
	6. Sistem menjalankan <i>use case</i> lihat alumni.

Aliran Alternatif 1: Masukan no induk, nama, username atau password kosong

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan no induk, nama, username atau *password*.)

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa no induk, nama, username atau <i>password</i> tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.
Aliran Alternatif 2: Username telah ada dalam sistem (Administrator pada langkah nomer 3 aliran utama memasukkan <i>username</i> yang telah ada dalam sistem.)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa username tersebut telah ada dalam sistem.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus 1. Masukan no induk, nama, <i>username</i> atau <i>password</i> tidak boleh kosong.	

Sumber : [Analisis]

4. Use Case Specification Melihat Biodata Alumni Secara Detail

Tabel 4.7 : Use Case Specification Melihat Biodata Alumni Secara Detail

Nama use case	Melihat biodata alumni
Aktor	Administrator, Alumni, Perusahaan
Deskripsi	Sistem harus dapat menampilkan biodata data alumni yang tercatat di dalam sistem. Data yang ditampilkan adalah no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto.

Pra-kondisi	Sistem telah berjalan dan aktor user telah login sebagai administrator serta memilih alumni yang akan dilihat.
Pasca-kondisi	Sistem menampilkan biodata alumni
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih menu "Alumni" Pencarian data alumni berdasarkan nama, alamat, jurusan, angkatan dan email.	2. Sistem menampilkan biodata alumni yang tercatat di dalam sistem. Data yang ditampilkan adalah no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto

Sumber : [Analisis]

5. Use Case Specification Mengedit Biodata Alumni

Tabel 4.8: Use Case Specification Mengedit Data Alumni

Nama use case	Mengedit biodata alumni
Aktor	Administrator, Alumni
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit data petugas yang telah dimasukkan. Data yang dapat diedit adalah no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, lokasi kerja, email, telepon, foto.
Pra-kondisi	Administrator telah menjalankan use case lihat alumni
Pasca-kondisi	Data alumni yang telah diedit disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "Ubah"	2. Sistem menampilkan no induk,



salah satu baris dari daftar alumni yang ditampilkan.	nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto. Dari data alumni yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, lokasi kerja, email, telepon, foto.	4. Sistem memvalidasi no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, lokasi kerja, email, telepon, foto.
	5. Sistem menjalankan use case melihat perubahan profil.
Aliran Alternatif 1: Masukan no induk atau nama	
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan no induk atau nama).	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa no induk, nama tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

6. Use Case Specification Menghapus Data Alumni

Tabel 4.9: Use Case Specification Menghapus Data Alumni

Nama use case	Menghapus data alumni
Aktor	Administrator

Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data alumni yang telah terdaftar dalam sistem <i>job placement center</i> .
Pra-kondisi	Administrator telah menjalankan <i>use case</i> lihat alumni
Pasca-kondisi	Data alumni yang telah dihapus akan hilang dari sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "Lihat Alumni" pilih salah satu baris dari daftar alumni yang ditampilkan.	2. Sistem menampilkan no induk, nama, tempat lahir, tanggal lahir, jurusan, angkatan, tanggal lulus. Dari data alumni yang telah dipilih oleh administrator.
3. Administrator menghapus data alumni dari sistem	4. Sistem memvalidasi masukan bahwa "anda yakin akan menghapus". 'OK'
Aliran Alternatif 1: Belum memilih alumni (Administrator pada langkah nomer 3 aliran utama administrator belum memilih data alumni yang akan dihapus)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa "anda harus memilih data terlebih dahulu"
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

7. Use Case Specification Menambah Daftar Perusahaan

Tabel 4.10: Use Case Specification Menambah Daftar Perusahaan

Nama use case	Menambah daftar perusahaan
----------------------	----------------------------

Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah Perusahaan. Data yang harus dicatat dari penambahan perusahaan adalah nama, alamat, email, telepon, foto, <i>username</i> , <i>password</i> .
Pra-kondisi	Administrator telah menjalankan <i>use case</i> input perusahaan
Pasca-kondisi	Daftar perusahaan telah ditambahkan ke dalam sistem

Aliran Utama

Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih <i>link</i> "Input Perusahaan".	2. Sistem menampilkan <i>form</i> masukan yang berupa <i>field</i> nama, alamat, email, telepon, foto, <i>username</i> , <i>password</i> .
3. Administrator memasukkan data nama, alamat, email, telepon, foto, <i>username</i> , <i>password</i> .	4. Sistem memvalidasi masukan nama, alamat, email, telepon, foto, <i>username</i> , <i>password</i> .
5. Aktor memilih "keyword" mengklik tombol "Simpan Data".	6. Sistem menjalankan <i>use case</i> melihat daftar perusahaan.

Aliran Alternatif 1: Masukan nama, *user name* atau *password* kosong

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

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

Aliran Alternatif 2: Username telah ada dalam sistem

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

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa <i>username</i> tersebut telah ada dalam sistem.
	2. Kembali pada langkah nomer 2 aliran utama.
Aturan khusus	
1. Masukan nama, <i>username</i> atau <i>password</i> tidak boleh kosong.	

Sumber : [Analisis]

8. Use Case Specification Melihat Profil Perusahaan Secara Detail

Tabel 4.11: Use Case Specification Melihat Profil Perusahaan Secara Detail

Nama use case	Melihat profil perusahaan
Aktor	Administrator, Alumni, Perusahaan
Deskripsi	Sistem harus dapat menampilkan profil perusahaan yang tercatat di dalam sistem. Data yang ditampilkan adalah nama, alamat, email, foto, telepon.
Pra-kondisi	Aktor telah menjalankan <i>use case</i> lihat perusahaan
Pasca-kondisi	Sistem menampilkan profil perusahaan
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Lihat Perusahaan".	2. Sistem menampilkan semua perusahaan yang telah ada di dalam sistem.
3. Aktor memilih "keyword" mengklik tombol "Detail".	4. Sistem menampilkan data perusahaan atau profil perusahaan

	yang telah ada di dalam sistem. Data yang ditampilkan berupa nama, alamat, email, foto, telepon.
--	---

Sumber : [Analisis]

9. Use Case Specification Mengedit Profil Perusahaan

Tabel 4.12: Use Case Specification Mengedit Profil Perusahaan

Nama use case	Mengedit profil perusahaan
Aktor	Administrator, Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit proffil perusahaan yang telah dimasukkan. Data yang dapat diedit adalah nama, alamat, email, foto, telepon.
Pra-kondisi	Administrator atau perusahaan telah menjalankan <i>use case</i> lihat perusahaan.
Pasca-kondisi	Data administrator atau perusahaan yang telah diedit disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "Ubah" salah satu baris dari daftar perusahaan yang ditampilkan.	2. Sistem menampilkan nama, alamat, email, foto, telepon. Dari data perusahaan yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk nama, alamat, email, foto, telepon.	4. Sistem memvalidasi masukan bahwa nama tidak boleh kosong.
	5. Sistem menjalankan use case melihat perubahan profil perusahaan.
Aliran Alternatif 1: Masukan nama (Administrator pada langkah nomer 3 aliran utama tidak memasukkan nama).	

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

Sumber : [Analisis]

10. Use Case Specification Menghapus Daftar Perusahaan

Tabel 4.13: Use Case Specification Menghapus Daftar Perusahaan

Nama use case	Menghapus daftar perusahaan
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus daftar perusahaan yang telah dimasukkan. Data yang dapat dihapus adalah Daftar perusahaan.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> lihat perusahaan
Pasca-kondisi	Daftar perusahaan yang telah dihapus akan hilang dari sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "Lihat Perusahaan" pilih salah satu baris dari daftar perusahaan yang ditampilkan.	2. Sistem menampilkan nama, alamat, email, telepon. Dari data perusahaan yang telah dipilih oleh administrator.
3. Administrator menghapus data perusahaan dari sistem	4. Sistem memvalidasi masukan bahwa "anda yakin akan menghapus". 'OK'
Aliran Alternatif 1: Belum memilih daftar perusahaan	

(Administrator pada langkah nomer 3 aliran utama administrator belum memilih daftar perusahaan yang akan dihapus)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa "anda harus memilih data terlebih dahulu"
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

11. Use Case Specification Menambah Daftar Berita

Tabel 4.14: Use Case Specification Menambah Daftar Berita

Nama use case	Menambah daftar Berita
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menambah berita. Data yang harus dicatat dari penambahan berita adalah judul, isi berita, foto.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> input berita
Pasca-kondisi	Daftar berita telah ditambahkan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator memilih <i>link</i> "Input Berita".	2. Sistem menampilkan <i>form</i> masukan yang berupa <i>field</i> judul, isi berita, foto.
3. Administrator memasukkan data judul, isi berita, foto.	4. Sistem memvalidasi masukan judul, isi berita, foto.
5. Aktor memilih "keyword" mengklik tombol "Simpan Berita".	6. Sistem menjalankan <i>use case</i> melihat daftar berita.
Aliran Alternatif 1: Masukan judul atau isi berita kosong	

(Administrator pada langkah nomer 3 aliran utama tidak memasukkan judul atau isi berita).

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

Sumber : [Analisis]

12. Use Case Specification Melihat Berita Secara Detail

Tabel 4.15: Use Case Specification Melihat Berita Secara Detail

Nama use case	Melihat berita secara detail
Aktor	Administrator, Alumni, Perusahaan
Deskripsi	Sistem harus dapat menampilkan berita yang tercatat di dalam sistem. Data yang ditampilkan adalah judul, isi berita, foto.
Pra-kondisi	Aktor telah menjalankan <i>use case</i> lihat berita
Pasca-kondisi	Sistem menampilkan daftar berita
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Lihat Berita".	2. Sistem menampilkan semua berita yang telah ada di dalam sistem.
3. Aktor memilih "keyword" mengklik tombol "Detail".	4. Sistem menampilkan berita yang telah ada di dalam sistem. Data yang ditampilkan berupa judul, isi berita, foto.

Sumber : [Analisis]

13. Use Case Specification Mengedit Berita

Tabel 4.16: Use Case Specification Mengedit Berita

Nama use case	Mengedit berita
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mengedit berita yang telah dimasukkan. Data yang dapat diedit adalah judul, isi berita, foto.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> lihat berita.
Pasca-kondisi	Data berita yang telah diedit disimpan kedalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "Ubah" salah satu baris dari daftar berita yang ditampilkan.	2. Sistem menampilkan judul, isi berita, foto. Dari daftar berita yang telah dipilih oleh administrator.
3. Administrator mengedit data untuk judul, isi berita, foto.	4. Sistem memvalidasi masukan bahwa judul atau isi berita tidak boleh kosong.
	5. Sistem menjalankan use case melihat perubahan berita.
Aliran Alternatif 1: Masukan nama	
(Administrator pada langkah nomer 3 aliran utama tidak memasukkan judul atau isi berita).	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama atau isi berita tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

14. Use Case Specification Menghapus Daftar Berita

Tabel 4.17: Use Case Specification Menghapus Daftar Berita

Nama use case	Menghapus daftar berita
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus daftar berita yang telah dimasukkan. Data yang dapat dihapus adalah Daftar berita.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> melihat berita
Pasca-kondisi	Daftar berita yang telah dihapus akan hilang dari sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "Hapus" salah satu baris dari daftar berita yang ditampilkan.	2. Sistem menampilkan judul, waktu. Dari data berita yang telah dipilih oleh administrator.
3. Administrator menghapus berita dari sistem	4. Sistem memvalidasi masukan bahwa "anda yakin akan menghapus". 'OK'
Aliran Alternatif 1: Belum memilih daftar berita	
(Administrator pada langkah nomer 3 aliran utama administrator belum memilih daftar berita yang akan dihapus)	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa "anda harus memilih data terlebih dahulu"
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

15. Use Case Specification Mereset Password

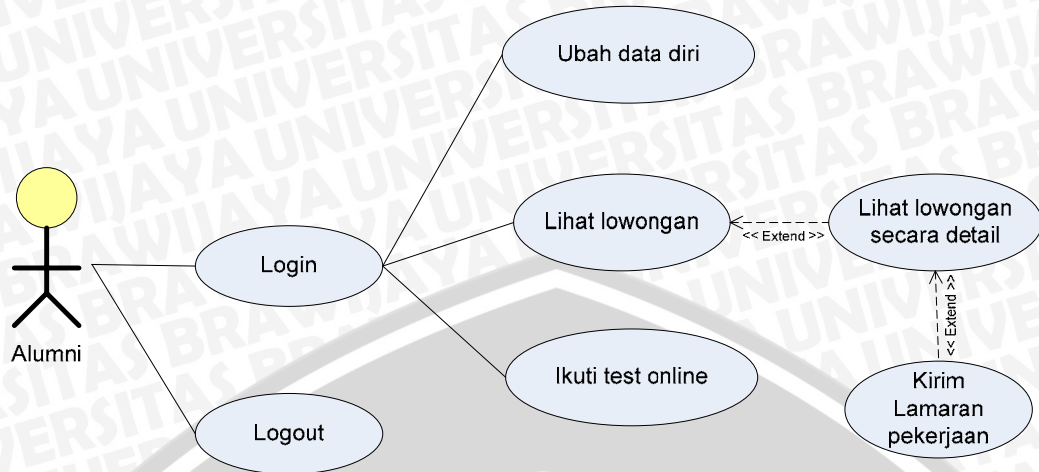
Tabel 4.18: Use Case Specification Mereset Password

Nama use case	Mereset <i>password</i>
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas untuk mereset password untuk semua aktor yang terdiri dari user name lama, password baru.
Pra-kondisi	Administrator telah menjalankan <i>use case</i> login sebagai administrator
Pasca-kondisi	<i>Password</i> aktor yang telah direset disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Administrator mengklik link "Password Reset".	2. Sistem menampilkan user name dan <i>password</i> baru. Dari password reset yang telah dipilih oleh administrator.
3. Administrator mengganti <i>password</i> baru.	4. Sistem memvalidasi masukan bahwa <i>password</i> telah diubah.
	5. Sistem menjalankan <i>use case</i> perubahan <i>password</i> .

Sumber : [Analisis]

4.1.2.2.2 Use Case Diagram untuk Modul Alumni Sistem Job Placement Center Online

Gambar 4.6 adalah *use case diagram* untuk modul alumni sistem *job placement center online*. Diagram *use case* ini melibatkan 3 buah aktor dan 7 buah *use case*. Ketujuh *use case* yang termasuk dalam modul pengaturan Alumni Sistem *job placement center*. Aktor yang terlibat dalam modul pengaturan alumni sistem *job placement center online* adalah seperti gambar berikut:



Gambar 4.6 : Use Case Diagram untuk Modul Pengaturan Alumni Job Placement Center Online
Sumber : [Analisis]

1. Use Case Specification Login

Tabel 4.19 : Use Case Specification Login

Nama use case	Login	
Aktor	Alumni	
Deskripsi	Sistem harus mampu memberikan fasilitas untuk login, sehingga hanya alumni tertentu yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu (sebagai suatu aktor).	
Pra-kondisi	Sistem telah berjalan.	
Pasca-kondisi	User telah login sebagai alumni	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. User memasukkan username dan password dan menekan tombol login.	2. Sistem memvalidasi username dan password.	
Aliran Alternatif 1: Pasangan username dan password tidak ada di dalam sistem.		
(User pada langkah nomer 1 aliran utama tidak memasukkan username atau password yang sesuai.)		

Aksi dari Aktor	Tanggapan dari Sistem
	1. Menampilkan pesan bahwa login gagal, silahkan mengulangi kembali.
	2. Kembali pada langkah nomer 1 aliran utama.

Sumber : [Analisis]

2. Use Case Specification Logout

Tabel 4.20: Use Case Specification Logout

Nama use case	Logout
Aktor	Alumni
Deskripsi	Sistem harus menyediakan fasilitas agar pengguna yang telah login keluar dari sistem.
Pra-kondisi	User telah login sebagai alumni.
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 Ubah Data Diri

Tabel 4.21: Use Case Specification Ubah Data Diri

Nama use case	Ubah data diri
Aktor	Administrator, Alumni
Deskripsi	Sistem harus menyediakan fasilitas untuk ubah data diri yang telah dimasukkan. Data yang dapat diedit adalah nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, lokasi kerja, email, telepon, foto, <i>password</i> .

Pra-kondisi	Alumni telah menjalankan <i>use case</i> ubah data diri
Pasca-kondisi	Data alumni yang telah diedit disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Alumni mengklik link "Ubah Data Diri".	2. Sistem menampilkan no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto, <i>username</i> , <i>password</i> .
3. Alumni mengedit data untuk nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, lokasi kerja, email, telepon, foto, <i>password</i> .	4. Sistem memvalidasi masukan no induk, nama, jenis kelamin, tempat lahir, tanggal lahir, alamat, agama, alumnus, tanggal lulus, jurusan, angkatan, IPK, lokasi kerja, email, telepon, foto, <i>username</i> , <i>password</i> .
	5. Sistem menjalankan <i>use case</i> ubah data diri.
Aliran Alternatif 1: Masukan no induk, nama atau <i>username</i> kosong (Alumni pada langkah nomer 3 aliran utama tidak memasukkan no induk, nama, <i>username</i>).	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa no induk, nama, <i>use name</i> . Tidak boleh kosong.
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

4. Use Case Specification Lowongan Pekerjaan

Tabel 4.22 : Use Case Specification Lowongan Pekerjaan

Nama use case	Lowongan pekerjaan
Aktor	Administrator, Alumni, Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk menampilkan lowongan-lowongan pekerjaan yang diinputkan oleh perusahaan yang ada di dalam sistem.
Pra-kondisi	Aktor telah menjalankan <i>use case</i> lowongan pekerjaan
Pasca-kondisi	Sistem menampilkan data lowongan pekerjaan dari perusahaan-perusahaan sebagai client.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor user memilih menu "Lowongan Pekerjaan".	2. Sistem menampilkan daftar perusahaan-perusahaan yang menyediakan lowongan pekerjaan.

Sumber : [Analisis]

5. Use Case Specification Melihat Lowongan Kerja Secara Detail

Tabel 4.23 : Use Case Specification Melihat Lowongan Kerja Secara Detail

Nama use case	Melihat lowongan kerja secara detail
Aktor	Administrator, Alumni, Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk melihat lowongan kerja secara detail yang ada di dalam sistem. Sesuai dengan nama perusahaan
Pra-kondisi	Aktor user telah login sebagai aktor administrator, alumni, perusahaan.
Pasca-kondisi	Sistem menampilkan data lowongan secara detail yang dicari oleh aktor administrator, alumni, perusahaan.
Aliran Utama	

Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor user memilih menu ” Lowongan Pekerjaan”.	2. Sistem menampilkan daftar perusahaan-perusahaan yang menyediakan lowongan pekerjaan.
3. Aktor mengklik link “Detail” pilih salah satu daftar lowongan yang ditampilkan.	4. Sistem menampilkan lowongan kerja secara detail. Data lowongan yang ditampilkan adalah nama perusahaan, nama lowongan, deskripsi, IPK minimum, jurusan, posisi, waktu, batas waktu.

Sumber : [Analisis]

6. Use Case Specification Mengirim Berkas Lamaran

Tabel 4.24 : Use Case Specification Mengirim Berkas Lamaran

Nama use case	Mengirim berkas lamaran
Aktor	Alumni
Deskripsi	Sistem harus dapat mengirim berkas lamaran dari alumni perusahaan yang menyediakan lowongan pekerjaan dan tercatat di dalam sistem. Data yang dikirimkan adalah berkas lamaran yang dibutuhkan oleh perusahaan berupa file <i>Microsoft word</i> .
Pra-kondisi	Alumni telah menjalankan <i>use case</i> lihat lowongan
Pasca-kondisi	Sistem dapat mengirimkan berkas lamaran berupa file <i>Microsoft word</i> .
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Alumni mengklik link “Lowongan Pekerjaan” .	2. Sistem menampilkan daftar lowongan dari perusahaan.
3. Alumni mengklik link “Detail” pilih salah satu daftar lowongan yang ditampilkan.	4. Sistem akan menampilkan lowongan pekerjaan dari perusahaan secara detail.

5. Alumni mengklik link “Kirim Lamaran”	6. Sistem akan menampilkan <i>form</i> untuk mengirim berkas lamaran.
---	---

Sumber : [Analisis]

7. Use Case Specification Ikuti Test Online

Tabel 4.25: Use Case Specification Ikuti Test Online

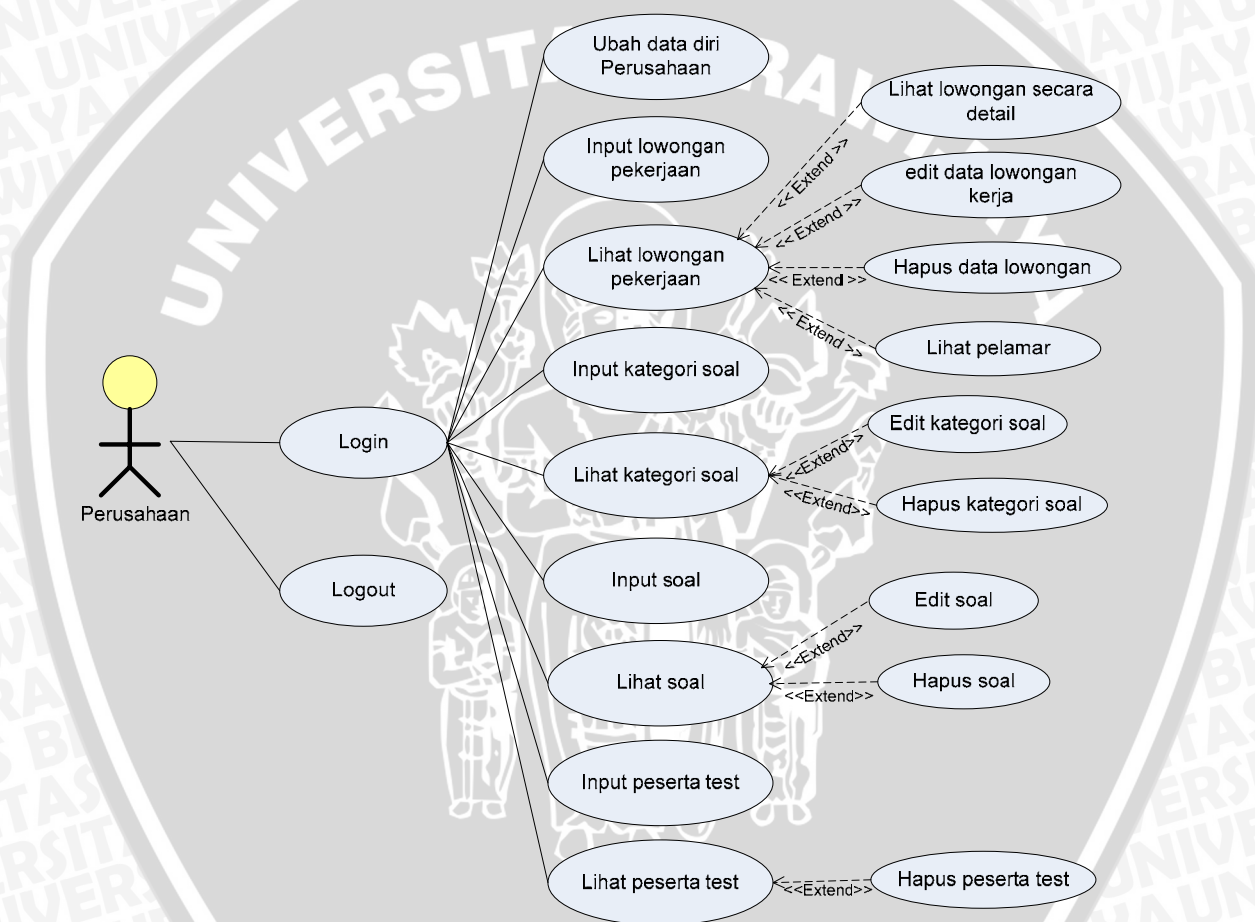
Nama use case	Ikuti test online	
Aktor	Alumni	
Deskripsi	Sistem harus menyediakan fasilitas untuk mengikuti test online untuk alumni yang terdiri dari no, kategori soal, waktu ujian, jam ujian, lama ujian.	
Pra-kondisi	Alumni telah menjalankan <i>use case</i> ikuti test online	
Pasca-kondisi	Alumni mengerjakan soal dari perusahaan secara online	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. Alumni mengklik link ”Ikuti Test Online”.	2. Sistem menampilkan no, kategori soal, waktu ujian, jam ujian, lama ujian.	
3. Alumni mengklik link “Ikuti Test”	4.Sistem memvalidasi dan menampilkan soal-soal dari perusahaan.	
	5. Sistem menjalankan use case ikuti test online.	
Aliran Alternatif 1: Link ikuti test tidak muncul		
(Alumni pada langkah nomer 3 aliran utama tidak muncul link ikuti test).		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. Alumni belum mengirimkan berkas lamaran pada perusahaan yang menyediakan lowongan pekerjaan.	
	2. Perusahaan belum mengaktifkan alumni untuk mengikuti test online.	

Sumber : [Analisis]



4.1.2.2.3 Use Case Diagram untuk Modul Pengaturan Perusahaan Sistem Job Placement Center Online

Gambar 4.7 adalah *use case diagram* untuk modul pengaturan perusahaan *client* sistem job placement center online. Diagram *use case* ini melibatkan 3 buah aktor dan 20 buah *use case*. Keduapuluh *use case* yang termasuk dalam modul pengaturan perusahaan *client* sistem job placement center online. Aktor yang terlibat dalam modul pengaturan perusahaan *client* sistem job placement center online adalah Administrator, Alumni dan Perusahaan.



Gambar 4.7 : Use Case Diagram untuk Modul Pengaturan Perusahaan Sistem Job Placement Center Online
Sumber : [Analisis]

1. Use Case Specification Login

Tabel 4.26 : Use Case Specification Login

Nama use case	Login
Aktor	Perusahaan

Deskripsi	Sistem harus mampu memberikan fasilitas untuk login, sehingga hanya perusahaan tertentu yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu (sebagai suatu aktor).	
Pra-kondisi	Sistem telah berjalan.	
Pasca-kondisi	User telah login sebagai perusahaan	
Aliran Utama		
Aksi dari Aktor	Tanggapan dari Sistem	
1. User memasukkan <i>username</i> dan <i>password</i> dan menekan tombol login.	2. Sistem memvalidasi <i>username</i> dan <i>password</i> ..	
Aliran Alternatif 1: Pasangan <i>username</i> dan <i>password</i> tidak ada di dalam sistem. (User pada langkah nomer 1 aliran utama tidak memasukkan <i>username</i> atau <i>password</i> yang sesuai.)		
Aksi dari Aktor	Tanggapan dari Sistem	
	1. Menampilkan pesan bahwa login gagal, silahkan mengulangi kembali.	
	2. Kembali pada langkah nomer 1 aliran utama.	

Sumber : [Analisis]

2. Use Case Specification Logout

Tabel 4.27: Use Case Specification Logout

Nama use case	Logout
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas agar pengguna yang telah login keluar dari sistem.
Pra-kondisi	User telah login sebagai perusahaan.
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 Ubah Data Diri Perusahaan

Tabel 4.28: Use Case Specification Ubah Data Diri Perusahaan

Nama use case	Ubah data diri perusahaan
Aktor	Administrator, Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk ubah data diri perusahaan yang telah dimasukkan. Data yang dapat diedit adalah nama perusahaan, alamat, email, telepon, foto, <i>password</i> .
Pra-kondisi	perusahaan telah menjalankan <i>use case</i> ubah data diri
Pasca-kondisi	Data perusahaan yang telah diedit disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link "Ubah Data Diri".	2. Sistem menampilkan nama perusahaan, alamat, email, telepon, foto, <i>username</i> , <i>password</i> .
3. Perusahaan mengedit data untuk nama perusahaan, alamat, email, telepon, foto, <i>password</i> .	4. Sistem memvalidasi masukan nama perusahaan, alamat, email, telepon, foto, <i>username</i> , <i>password</i> .
	5. Sistem menjalankan use case ubah data diri.
<p>Aliran Alternatif 1: Masukan nama perusahaan kosong (Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan nama perusahaan).</p>	

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

Sumber : [Analisis]

4. Use Case Specification Input Lowongan Pekerjaan

Tabel 4.29: Use Case Specification Input Lowongan Pekerjaan

Nama use case	Input lowongan pekerjaan
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk Menginput lowongan pekerjaan. Data yang dapat diinputkan adalah nama lowongan, deskripsi, posisi, batas umur, IPK minimum, jurusan, batas waktu.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> input lowongan.
Pasca-kondisi	Data lowongan pekerjaan telah ditambahkan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan memilih menu "Input Lowongan".	2. Sistem menampilkan <i>form</i> untuk input lowongan pekerjaan.
3. Perusahaan memasukkan nama lowongan, deskripsi, posisi, batas umur, IPK minimum, jurusan, batas waktu.	4. Sistem memvalidasi masukan nama lowongan, deskripsi, posisi, batas umur, IPK minimum, jurusan, batas waktu.
	5. Sistem menjalankan use case input lowongan
Aliran Alternatif 1: Masukan nama, deskripsi, posisi, IPK minimum, batas waktu kosong	

(Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan nama, deskripsi, posisi, IPK minimum, batas waktu).

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama, deskripsi, posisi, IPK minimum atau batas waktu harus diisikan
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

5. Use Case Specification Melihat Lowongan Pekerjaan

Tabel 4.30 : Use Case Specification Melihat Lowongan Pekerjaan

Nama use case	Melihat lowongan pekerjaan
Aktor	Administrator, Alumni, Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk melihat lowongan kerja yang ada di dalam sistem. Sesuai dengan yang diinputkan perusahaan.
Pra-kondisi	Aktor user telah login sebagai aktor administrator, alumni, perusahaan.
Pasca-kondisi	Sistem menampilkan daftar lowongan yang diinputkan aktor perusahaan.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor user memilih menu "Lihat Lowongan".	2. Sistem menjalankan use case lihat lowongan dari perusahaan yang menginputkan.
	3. Sistem menampilkan lowongan pekerjaan. Daftar lowongan yang

	ditampilkan adalah nama lowongan, posisi, batas umur, jurusan, IPK minimum.
--	---

Sumber : [Analisis]

6. Use Case Specification Melihat Lowongan Pekerjaan Secara Detail

Tabel 4.31: Use Case Specification Melihat Lowongan Pekerjaan Secara Detail

Nama use case	Melihat lowongan pekerjaan secara detail
Aktor	Administrator, Alumni, Perusahaan
Deskripsi	Sistem harus dapat menampilkan lowongan pekerjaan yang tercatat di dalam sistem. Data yang ditampilkan adalah nama perusahaan, nama lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu.
Pra-kondisi	Aktor telah menjalankan use case lihat lowongan secara detail
Pasca-kondisi	Sistem menampilkan lowongan pekerjaan secara detail.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Lihat Lowongan".	2. Sistem menampilkan semua lowongan yang telah ada di dalam sistem.
3. Aktor memilih "keyword" mengklik tombol "Detail".	4. Sistem menampilkan daftar lowongan yang telah ada di dalam sistem. Data yang ditampilkan berupa nama perusahaan, nama lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu.

Sumber : [Analisis]

7. Use Case Specification Mengubah Data lowongan

Tabel 4.32: Use Case Specification Mengubah Data Lowongan

Nama use case	Mengubah data lowongan
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk mengubah data lowongan yang telah diinputkan. Data yang dapat diubah adalah nama perusahaan, nama lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> ubah lowongan
Pasca-kondisi	Data lowongan yang telah diubah disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link “Lihat Lowongan” pilih salah satu baris dari daftar lowongan yang ditampilkan.	2. Sistem menampilkan <i>form</i> ubah lowongan.
3. Perusahaan mengubah data untuk nama lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu.	4. Sistem memvalidasi masukan bahwa nama lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu. tidak boleh kosong.
5. Perusahaan mengklik tombol “Simpan Data”	6. Sistem menjalankan use case. Bahwa data berhasil disimpan “oke”. Melihat perubahan data lowongan.
<p>Aliran Alternatif 1: Masukan nama, lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu kosong</p> <p>(Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan nama, lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu).</p>	

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama, lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu harus diisikan
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

8. Use Case Specification Menghapus Data Lowongan

Tabel 4.33 : Use Case Specification Menghapus Data Lowongan

Nama use case	Menghapus data Lowongan
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data lowongan tertentu.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> lihat lowongan.
Pasca-kondisi	Data lowongan pekerjaan dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link “Lihat Lowongan” pilih salah satu baris dari daftar lowongan yang ditampilkan.	2. Sistem memvalidasi dan menampilkan pesan ”anda yakin akan menghapus”.”Oke”
3. Perusahaan menghapus data untuk lowongan kerja	4.

Sumber : [Analisis]

9. Use Case Specification Lihat Pelamar

Tabel 4.34: Use Case Specification Lihat Pelamar

Nama use case	Lihat pelamar
Aktor	Perusahaan
Deskripsi	Sistem harus dapat menampilkan daftar alumni yang melamar dalam sebuah lowongan yang tercatat di dalam sistem. Data yang ditampilkan adalah nama lowongan, posisi, batas umur, IPK minimal, jurusan waktu dan jumlah pelamar. Serta data alumni yang ditampilkan adalah no, nama, jurusan, angkatan, IPK, tanggal kelulusan dan download cv.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> lihat lowongan
Pasca-kondisi	Sistem menampilkan lowongan pekerjaan dan jumlah alumni yang telah mengirimkan lamaran.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link “Lihat Lowongan” pilih salah satu baris dari daftar lowongan yang ditampilkan.	2. Sistem menampilkan Daftar alumni yang telah mengirimkan berkas lamaran. Data yang ditampilkan adalah no, nama, jurusan, angkatan, IPK, tanggal kelulusan dan download CV.

Sumber : [Analisis]

10. Use Case Specification Input Kategori Soal

Tabel 4.35: Use Case Specification Input Kategori Soal

Nama use case	Input kategori soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk Menginput kategori soal. Data yang dapat diinputkan adalah nama, tanggal, jam, durasi.

Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> input kategori soal.
Pasca-kondisi	Data kategori soal telah ditambahkan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan memilih menu "Input Kategori Soal".	2. Sistem menampilkan <i>form</i> untuk input kategori soal.
3. Perusahaan memasukkan nama, tanggal, jam, durasi.	4. Sistem memvalidasi masukan nama, tanggal, jam, durasi.
	5. Sistem menjalankan <i>use case</i> input kategori soal
Aliran Alternatif 1: Masukan nama, tanggal, jam, durasi kosong (Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan nama, tanggal, jam, durasi).	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama, lowongan, deskripsi, IPK minimum, jurusan, posisi, batas umur, batas waktu harus diisikan
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

11. Use Case Specification Lihat Kategori Soal

Tabel 4.36 : Use Case Specification Lihat Kategori Soal

Nama use case	Lihat kategori soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk melihat kategori soal yang ada di dalam sistem. Sesuai dengan yang diinputkan perusahaan. Data yang ditampilkan adalah nama, tanggal, waktu, durasi.
Pra-kondisi	Aktor telah login sebagai aktor perusahaan.
Pasca-kondisi	Sistem menampilkan kategori soal yang diinputkan aktor perusahaan.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor user memilih menu "Lihat Kategori Soal".	2. Sistem menjalankan use case lihat kategori soal dari perusahaan yang menginputkan.
	3. Sistem menampilkan kategori soal. Kategori soal yang ditampilkan adalah nama, tanggal, waktu, durasi.

Sumber : [Analisis]

12. Use Case Specification Mengubah Kategori Soal

Tabel 4.37: Use Case Specification Mengubah Kategori Soal

Nama use case	Mengubah kategori soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk mengubah kategori soal yang telah diinputkan. Data yang dapat diubah adalah nama, tanggal, waktu, durasi.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> ubah kategori soal
Pasca-kondisi	Kategori soal yang telah diubah disimpan ke dalam sistem

Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link “Lihat Kategori Soal” pilih salah satu baris dari daftar kategori soal yang ditampilkan.	2. Sistem menampilkan <i>form</i> ubah kategori soal.
3. Perusahaan mengubah data kategori soal untuk nama, tanggal, waktu, durasi.	4. Sistem memvalidasi masukan bahwa nama, tanggal, waktu, durasi. tidak boleh kosong.
5. Perusahaan mengklik tombol “Simpan Data”	6. Sistem menjalankan use case. Bahwa data berhasil disimpan “Oke”. Melihat perubahan data kategori soal.
Aliran Alternatif 1: Masukan nama, tanggal, waktu, durasi kosong (Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan nama, tanggal, waktu, durasi.).	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa nama, tanggal, waktu, durasi. Harus diisi
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

13. Use Case Specification Menghapus Kategori Soal

Tabel 4.38 : Use Case Specification Menghapus Kategori Soal

Nama use case	Menghapus kategori soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data kategori soal tertentu.

Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> lihat kategori soal
Pasca-kondisi	Kategori soal dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link “Lihat Kategori Soal” pilih salah satu baris dari daftar kategori soal yang ditampilkan.	2. Sistem memvalidasi dan menampilkan pesan anda yakin akan menghapus. Tekan ”oke”
	3. Sistem menghapus data untuk kategori soal.

Sumber : [Analisis]

14. Use Case Specification Input Soal

Tabel 4.39: Use Case Specification Input Soal

Nama use case	Input soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk Menginput soal. Data yang dapat diinputkan adalah kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> input soal.
Pasca-kondisi	Data soal telah ditambahkan ke dalam sistem dan ditampilkan.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan memilih menu ”Input Soal”.	2. Sistem menampilkan <i>form</i> untuk input soal.

3. Perusahaan memasukkan kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban.	4. Sistem memvalidasi masukan kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban.
	5. Sistem menjalankan use case input soal
<p>Aliran Alternatif 1: Masukan kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban kosong</p> <p>(Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban).</p>	
Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban harus diisi
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

15. Use Case Specification Lihat Soal

Tabel 4.40: Use Case Specification Lihat Soal

Nama use case	Lihat soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk melihat soal yang ada di dalam sistem, sesuai dengan yang diinputkan perusahaan itu sendiri. Data yang ditampilkan adalah kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban.
Pra-kondisi	Aktor telah login sebagai aktor perusahaan.

Pasca-kondisi	Sistem menampilkan soal yang diinputkan oleh aktor perusahaan itu sendiri.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor user memilih menu "Lihat Soal".	2. Sistem menjalankan <i>use case</i> lihat soal dari perusahaan yang menginputkan.
	3. Sistem menampilkan soal dari perusahaan masing-masing. soal yang ditampilkan adalah kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban

Sumber : [Analisis]

16. Use Case Specification Mengubah Soal

Tabel 4.41: Use Case Specification Mengubah Soal

Nama use case	Mengubah soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk mengubah soal yang telah diinputkan. Data soal yang dapat diubah adalah pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> ubah soal
Pasca-kondisi	Kategori soal-soal yang telah diubah disimpan ke dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link "Lihat Soal" pilih salah satu baris dari daftar kategori soal yang ditampilkan.	2. Sistem menampilkan <i>form</i> ubah soal.

3. Perusahaan dapat mengubah data soal untuk pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban	4. Sistem memvalidasi masukan bahwa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban. tidak boleh kosong.
5. Perusahaan mengklik tombol "Simpan Data"	6. Sistem menjalankan use case. Bahwa data berhasil disimpan.

Aliran Alternatif 1: Masukan kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban.

(Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban).

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem menampilkan pesan kesalahan bahwa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban. Harus diisikan
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

17. Use Case Specification Menghapus Soal

Tabel 4.42 : Use Case Specification Menghapus Soal

Nama use case	Menghapus soal
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data soal tertentu.
Pra-kondisi	Perusahaan telah menjalankan use case lihat soal.
Pasca-kondisi	Data soal dihapus dari dalam sistem
Aliran Utama	

Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link “Lihat Soal” pilih salah satu baris dari daftar soal yang ditampilkan.	2. Sistem memvalidasi dan menampilkan pesan anda yakin akan menghapus. Tekan ”Oke”
	3. Sistem menghapus data soal.

Sumber : [Analisis]

18. Use Case Specification Input Peserta Test

Tabel 4.43 : Use Case Specification Input Peserta Test

Nama use case	Input peserta test
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk Menginputkan peserta test. Data yang dapat diinputkan adalah alumni yang telah mengirimkan berkas lamaran dan data yang ditampilkan adalah kategori soal, nama lowongan, nama peserta yang terdiri dari nim dan nama.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> input peserta test.
Pasca-kondisi	Peserta test dapat ditambahkan ke dalam sistem dan ditampilkan.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan memilih menu ”Input Peserta Test”.	2. Sistem menampilkan <i>form</i> untuk input peserta test.
3. Perusahaan memasukkan kategori soal, nama lowongan.	4. Sistem memvalidasi masukan kategori soal, nama lowongan, nama peserta yang terdiri dari nim dan nama.
	5. Sistem menjalankan use case input soal

Aliran Alternatif 1: Masukan kategori soal, nama lowongan kosong

(Perusahaan pada langkah nomer 3 aliran utama tidak memasukkan kategori soal, nama lowongan).

Aksi dari Aktor	Tanggapan dari Sistem
	1. Sistem tidak dapat menampilkan peserta test.
	2. Kembali pada langkah nomer 2 aliran utama.

Sumber : [Analisis]

19. Use Case Specification Lihat Peserta Test

Tabel 4.44 : Use Case Specification Lihat Peserta Test

Nama use case	Lihat peserta test
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk melihat data peserta test yang ada di dalam sistem. Data yang ditampilkan adalah kategori soal, nim, nama peserta test, tanggal, jam, benar, salah.
Pra-kondisi	Perusahaan telah menjalankan <i>use case</i> lihat peserta test.
Pasca-kondisi	Sistem dapat menampilkan peserta test yang telah mengirimkan berkas lamaran dan telah diaktifkan oleh perusahaan.
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor user memilih menu "Lihat Peserta Test".	2. Sistem menampilkan peserta test yang telah mengirimkan berkas lamaran dan telah diaktifkan oleh perusahaan. Data yang ditampilkan adalah kategori, no induk, nama, tanggal, jam, benar, salah.

	3. Sistem menampilkan peserta test dari perusahaan masing-masing. peserta yang ditampilkan adalah kategori soal, nim, nama peserta test, tanggal, jam, benar, salah.
--	--

Sumber : [Analisis]

20. Use Case Specification Menghapus Peserta Test

Tabel 4.45 : Use Case Specification Menghapus Peserta Test

Nama use case	Menghapus peserta test
Aktor	Perusahaan
Deskripsi	Sistem harus menyediakan fasilitas untuk menghapus data peserta test.
Pra-kondisi	Perusahaan telah menjalankan use case lihat lihat peserta test.
Pasca-kondisi	Data peserta test dihapus dari dalam sistem
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. Perusahaan mengklik link “Lihat Peserta Test” pilih salah satu baris dari daftar peserta test yang ditampilkan.	2. Sistem memvalidasi dan menampilkan pesan anda yakin akan menghapus. Tekan ”Oke”
	3. Sistem menghapus data peserta test.

Sumber : [Analisis]

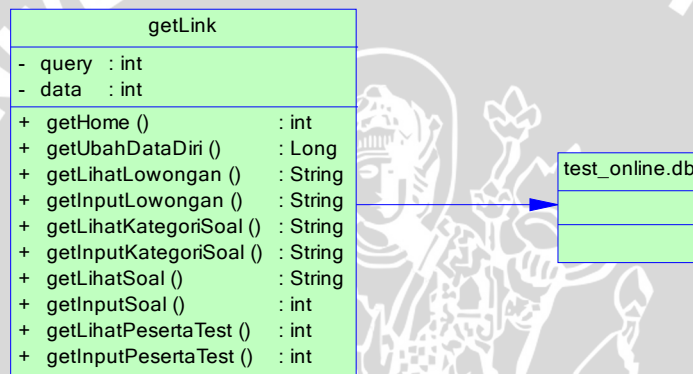
4.1.2.3 Class Diagram

Diagram kelas memberikan gambaran pemodelan elemen-elemen kelas yang membentuk sebuah perangkat lunak. Kelas 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

Alumni-Polinema.admin

Klas alumni-polinema.admin memiliki beberapa *method* yang harus diimplementasi pada kelas model yang lain. Gambar 4.8 adalah *class diagram* alumni-polinema.admin.

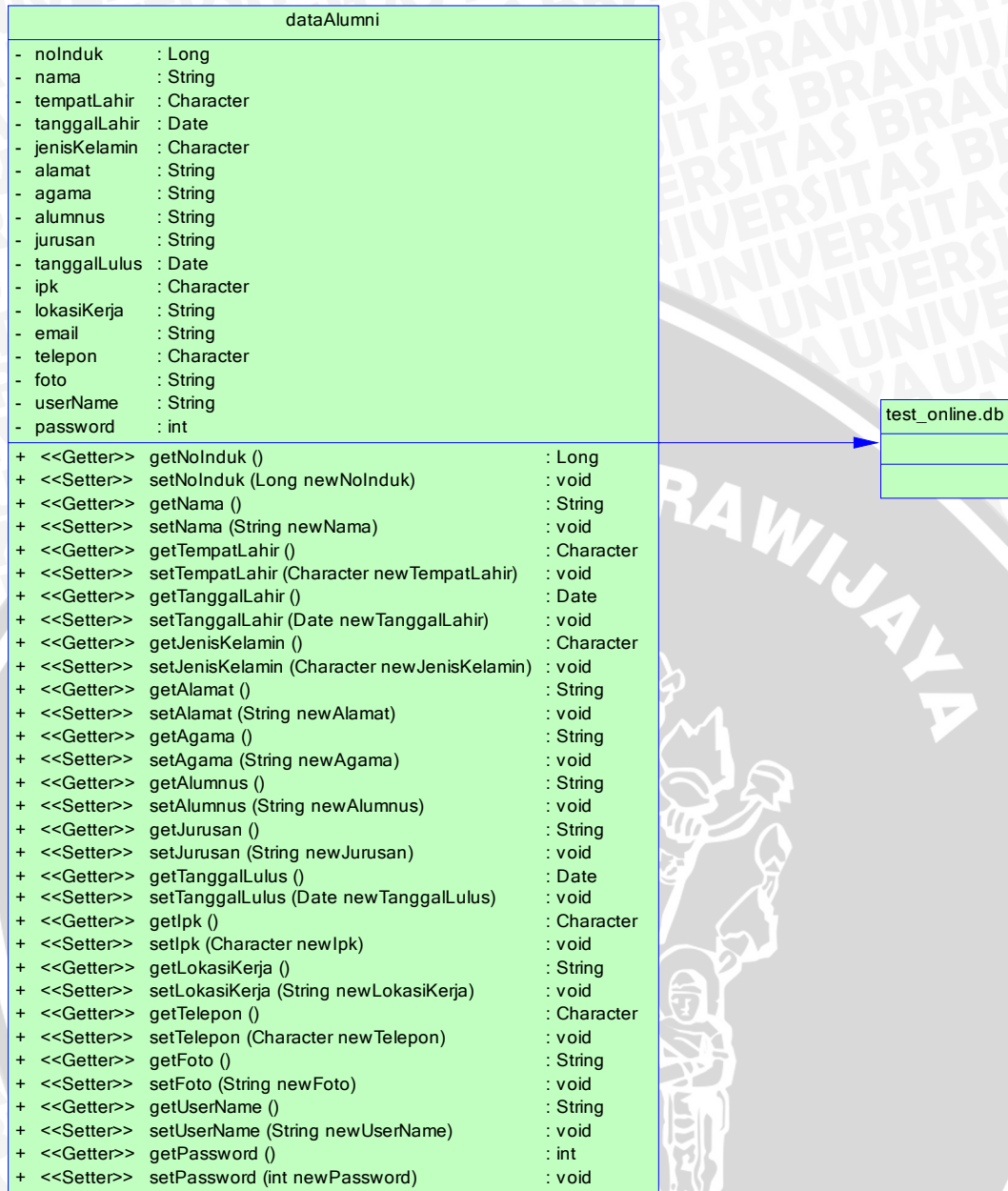


Gambar 4.8 : *Class Diagram* alumni-polinema.admin
Sumber : [Perancangan]

4.1.2.3.2 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema.inputAlumni

Klas Alumni-Polinema.inputAlumni memodelkan data-data alumni yang terdaftar dalam *job placement center* Politeknik Negeri Malang. Gambar 4.9 adalah *class diagram* Alumni-Polinema.inputAlumni.



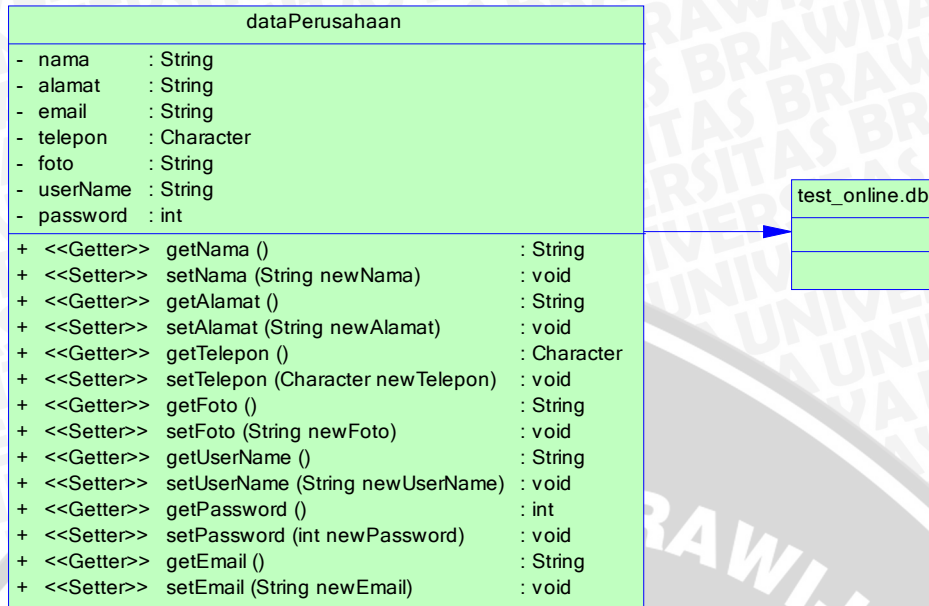
Gambar 4.9 : *Class Diagram* Alumni-Polinema .inputAlumni

Sumber : [Perancangan]

4.1.2.3.3 *Class Diagram* untuk *Class-Class* dari Paket

Alumni-Polinema .inputPerusahaan

Klas Alumni-Polinema .inputPerusahaan memodelkan data-data perusahaan yang menjadi *client job placement center* Politeknik Negeri Malang. Gambar 4.10 adalah *class diagram* Alumni-Polinema .inputPerusahaan.

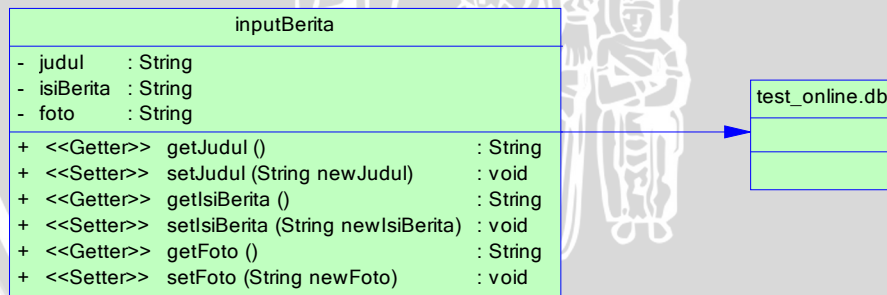


Gambar : 4.10 : *Class Diagram* Alumni-Polinema.inputPerusahaan
Sumber : [Perancangan]

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

Alumni-Polinema.inputBerita

Klas Alumni-Polinema.inputBerita memodelkan berita-berita tentang *job placement center* Politeknik Negeri Malang. Gambar 4.11 adalah *class diagram* Alumni-Polinema.inputBerita.

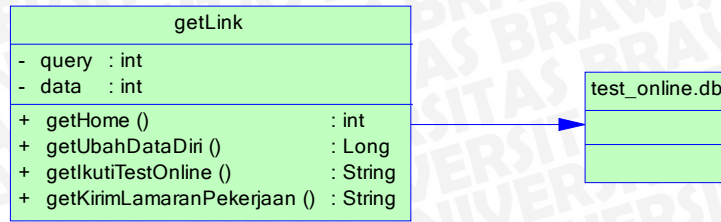


Gambar : 4.11 : *Class Diagram* Alumni-Polinema.inputBerita
Sumber : [Perancangan]

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

Alumni-Polinema.alumni

Klas alumni-polinema.alumni memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain. Gambar 4.12 adalah *class diagram* alumni-polinema.alumni.

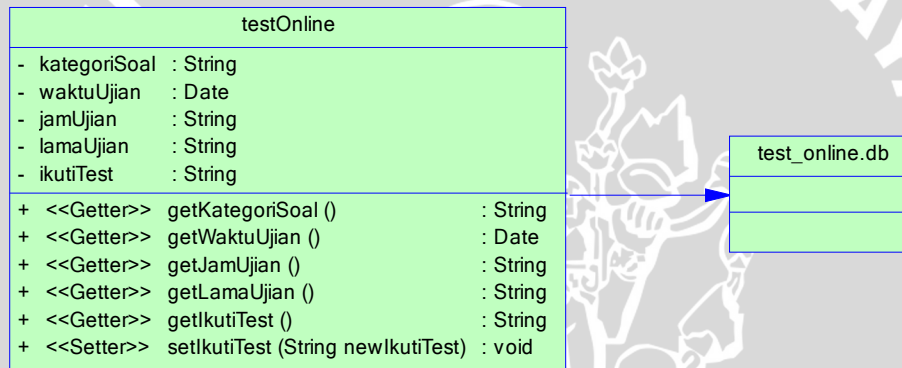


Gambar 4.12 : *Class Diagram* alumni-polinema.alumni
Sumber : [Perancangan]

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

Alumni-Polinema.testOnline

Klas Alumni-Polinema.testOnline memodelkan waktu dan pelaksanaan tes online *job placement center* Politeknik Negeri Malang. Gambar 4.13 adalah *class diagram* Alumni-Polinema.testOnline.

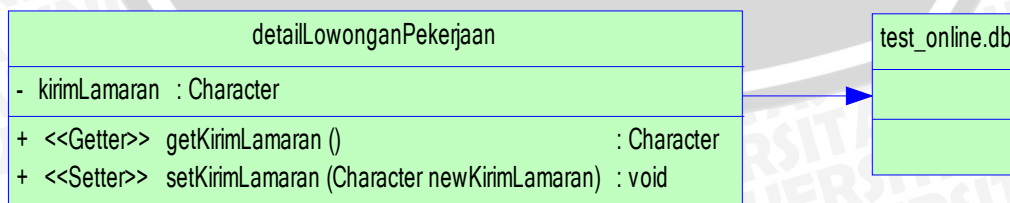


Gambar 4.13: *Class Diagram* Alumni-Polinema.testOnline
Sumber : [Perancangan]

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

Alumni-Polinema.kirimLamaran

Klas Alumni-Polinema.kirimLamaran memodelkan waktu dan pengiriman berkas lamaran *job placement center* Politeknik Negeri Malang. Gambar 4.14 adalah *class diagram* Alumni-Polinema.kirimLamaran.

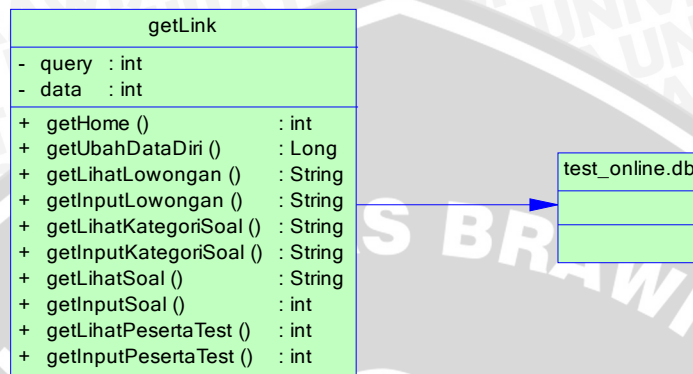


Gambar 4.14 : *Class Diagram* Alumni-Polinema.kirimLamaran
Sumber : [Perancangan]

4.1.2.3.8 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema.perusahaan

Klas alumni-polinema.perusahaan memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain. Gambar 4.15 adalah *class diagram* alumni-polinema.perusahaan.

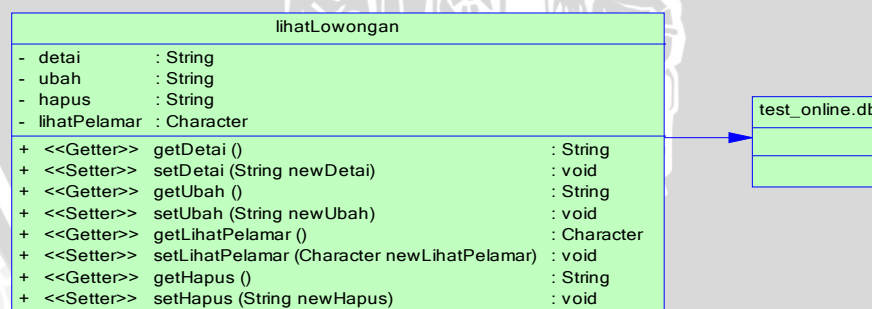


Gambar 4.15 : Class Diagram Alumni-Polinema.Perusahaan
Sumber : [Perancangan]

4.1.2.3.9 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema.lihatLowongan

Klas alumni-polinema.lihatLowongan memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain. Gambar 4.16 adalah *class diagram* alumni-polinema.lihatLowongan.



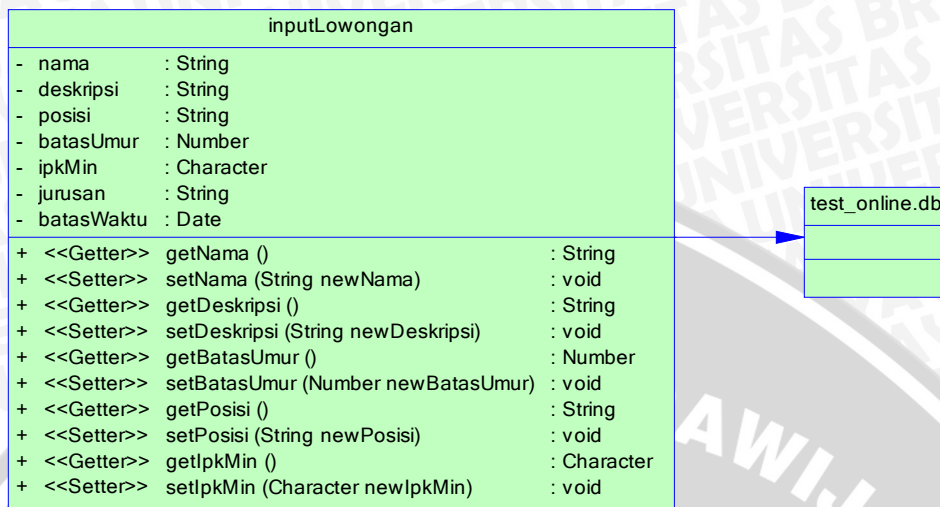
Gambar 4.16 : Class Diagram Alumni-Polinema.lihatLowongan
Sumber : [Perancangan]

4.1.2.3.10 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema.inputLowongan

Klas Alumni-Polinema.inputLowongan memodelkan data-data lowongan pekerjaan dari perusahaan yang menjadi *client job placement center*

Politeknik Negeri Malang. Gambar 4.17 adalah *class diagram* Alumni-Polinema.inputLowongan.

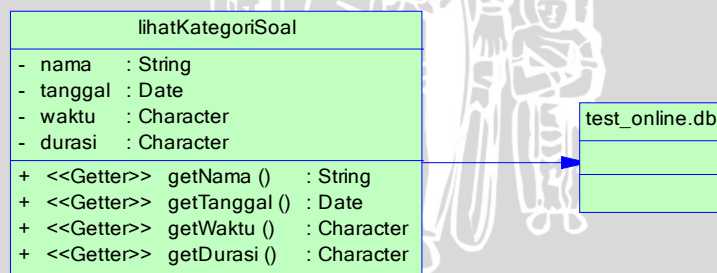


Gambar 4.17 : *Class Diagram* Alumni-Polinema.inputLowongan
Sumber : [Perancangan]

4.1.2.3.11 *Class Diagram* untuk *Class-Class* dari Paket

Alumni-Polinema. lihatKategoriSoal

Klas alumni-polinema.lihatKategoriSoal memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain. Gambar 4.18 adalah *class diagram* alumni-polinema.lihatKategoriSoal.

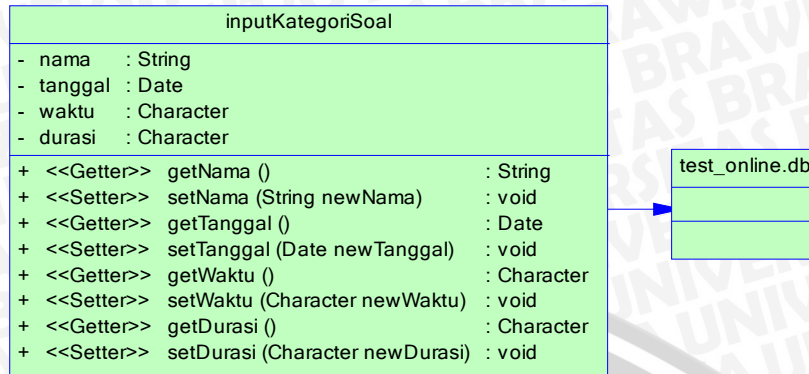


Gambar 4.18 : *Class Diagram* Alumni-Polinema.inputKategoriSoal
Sumber : [Perancangan]

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

Alumni-Polinema.inputKategoriSoal

Klas Alumni-Polinema.inputKategoriSoal memodelkan soal-soal dari perusahaan yang menjadi *client job placement center* Politeknik Negeri Malang. Gambar 4.19 adalah *class diagram* Alumni-Polinema.inputKategoriSoal.

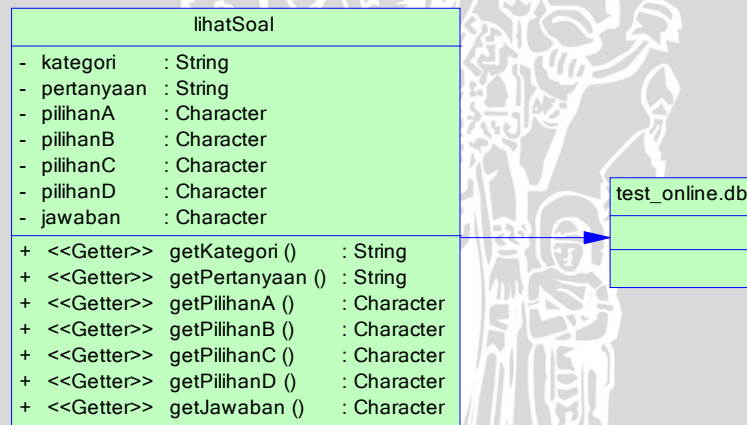


Gambar 4.19 : *Class Diagram* Alumni-Polinema.inputKategoriSoal
Sumber : [Perancangan]

4.1.2.3.13 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema. lihatSoal

Klas alumni-polinema.lihatSoal memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain. Gambar 4.20 adalah *class diagram* alumni-polinema.lihatSoal



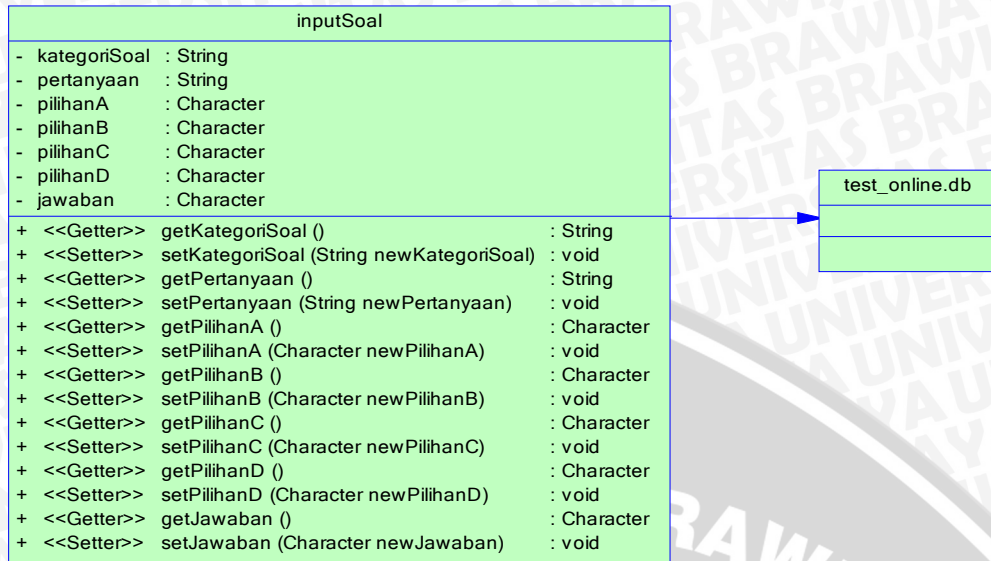
Gambar 4.20 : *Class Diagram* Alumni-Polinema.lihatSoal
Sumber : [Perancangan]

4.1.2.3.14 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema.inputSoal

Klas Alumni-Polinema.inputSoal memodelkan soal-soal dari perusahaan yang menjadi *client job placement center* Politeknik Negeri Malang.

Gambar 4.21 adalah *class diagram* Alumni-Polinema.inputSoal.

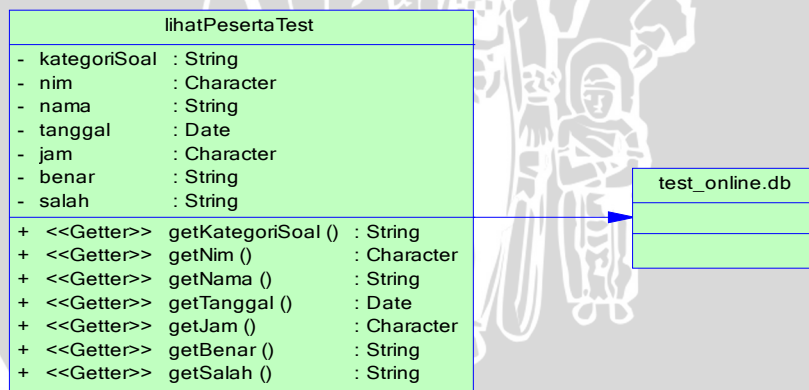


Gambar 4.21 : Class Diagram Alumni-Polinema.inputSoal
Sumber : [Perancangan]

4.1.2.3.15 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema. lihatPesertaTest

Klas alumni-polinema.lihatPesertaTest memiliki beberapa *method* yang harus diimplementasi pada klas model yang lain. Gambar 4.22 adalah *class diagram* alumni-polinema.lihatPesertaTest



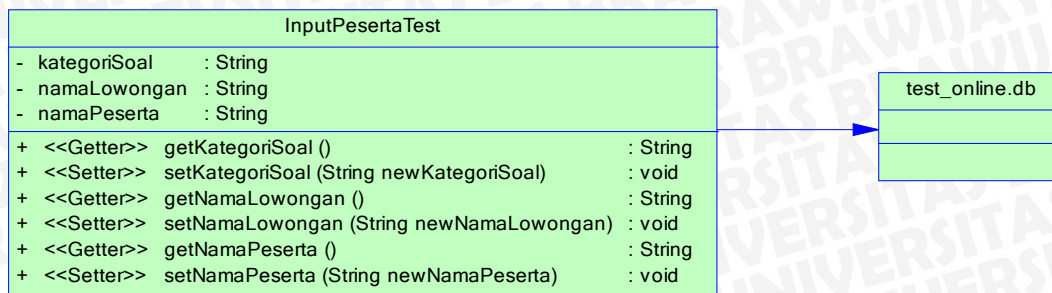
Gambar 4.22 : Class Diagram Alumni-Polinema.lihatPesertaTest
Sumber : [Perancangan]

4.1.2.3.16 Class Diagram untuk Class-Class dari Paket

Alumni-Polinema.inputPesertaTest

Klas Alumni-Polinema.inputPesertaTest memodelkan alumni-alumni yang menjadi client *job placement center* Politeknik Negeri Malang. Gambar 4.23 adalah *class diagram* Alumni-Polinema.inputPesertaTest.





Gambar 4.23 : *Class Diagram* Alumni-Polinema.inputPesertaTest
Sumber : [Perancangan]

4.1.2.4 *Sequence Diagram*

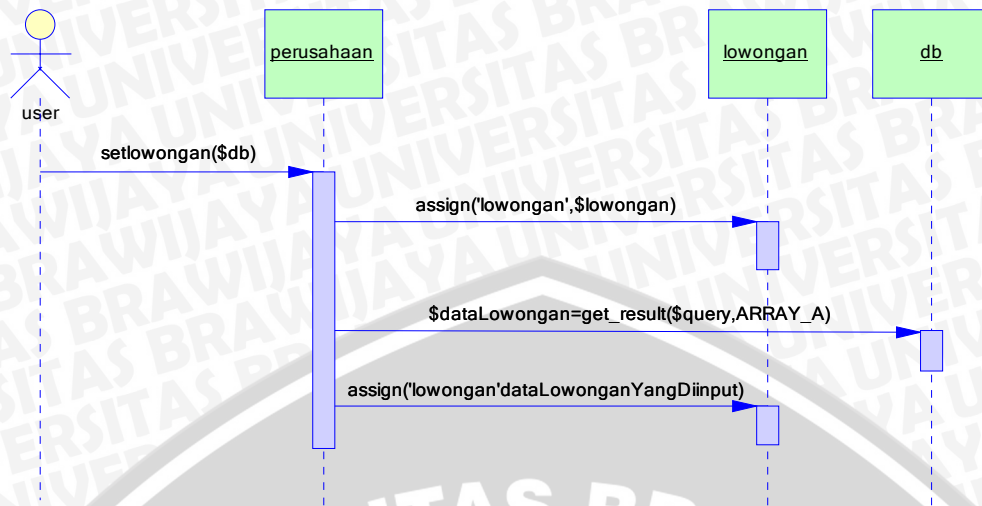
Hubungan antar klas yang digambarkan pada diagram klas termasuk dalam pemodelan statis, tanpa disertai dengan gambaran aliran proses atau data antarklas yang satu dengan klas yang lain. *Sequence diagram* digunakan untuk memodelkan aliran jalannya proses yang ditunjukkan dengan interaksi antar objek 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 setLowongan(\$db)*
2. *Method setKategoriSoal(\$db)*
3. *Method setJawaban(\$db)*
4. *Method simpanJawaban(\$db)*

4.1.2.4.1 *Sequence Diagram untuk method setLowongan (\$db)*

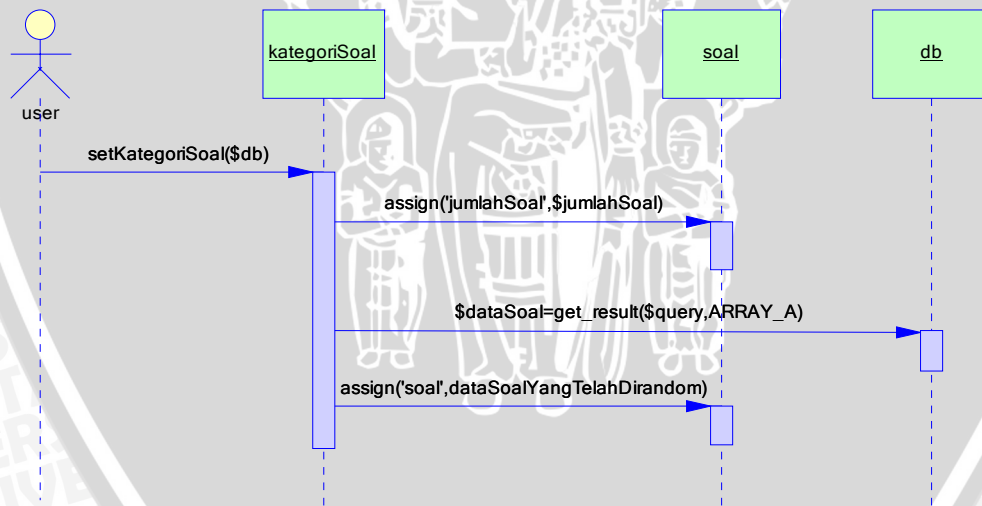
Gambar 4.24 adalah *sequence diagram* untuk *method setLowongan*. *Method* ini dipergunakan untuk menampilkan daftar lowongan kerja yang dapat dilamar oleh alumni yang memenuhi syarat. *Sequence diagram* dari *method setLowongan* adalah sebagai berikut:



Gambar 4.24 : *Sequence Diagram* untuk *method* setLowongan
Sumber : [Perancangan]

4.1.2.4.2 Sequence Diagram untuk method setKategoriSoal (\$db)

Gambar 4.25 adalah *sequence diagram* untuk *method* setKategoriSoal. *Method* ini dipergunakan untuk menampilkan daftar soal yang harus dikerjakan oleh peserta. *Sequence diagram* dari *method* setKategoriSoal adalah sebagai berikut:

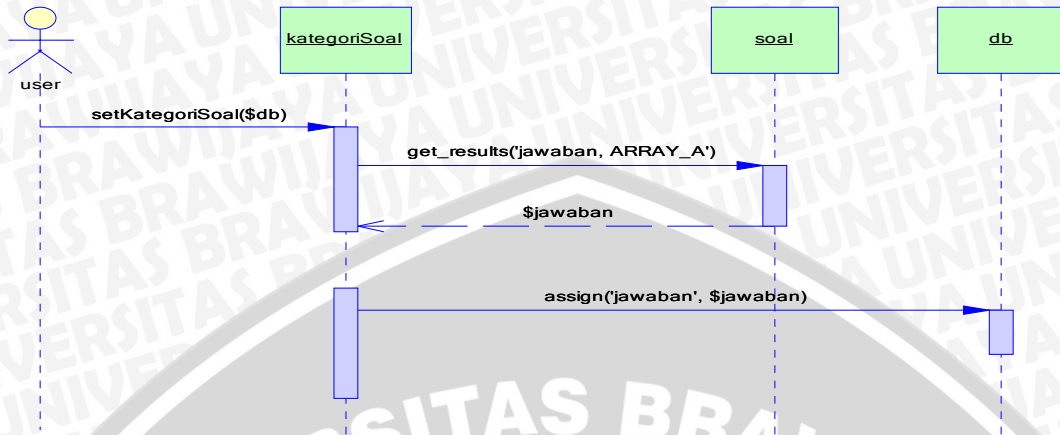


Gambar 4.25 : *Sequence Diagram* untuk *method* setKategoriSoal
Sumber : [Perancangan]

4.1.2.4.3 Sequence Diagram untuk Method setJawaban (\$db)

Gambar 4.26 adalah *sequence diagram* untuk *method* setJawaban. *Method* ini dipergunakan untuk menampilkan daftar jawaban dari masing-masing

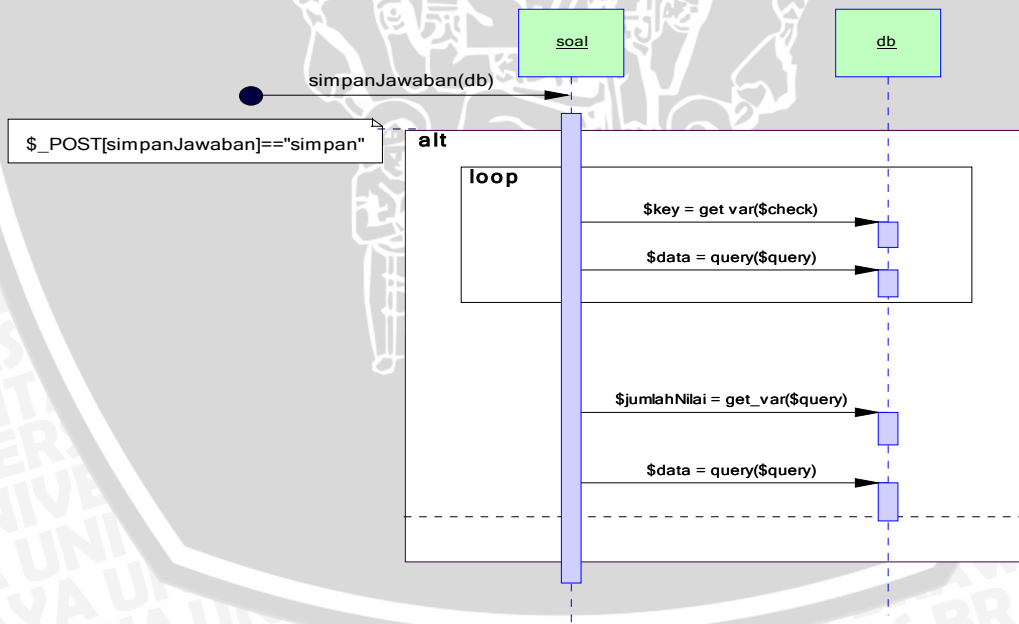
soal yang harus dikerjakan oleh peserta. *Sequence diagram* dari *method* *setJawaban* adalah sebagai berikut :



Gambar 4.26 : *Sequence Diagram* untuk *Method* *setJawaban*
Sumber : [Perancangan]

4.1.2.4.4 Sequence Diagram untuk method *simpanJawaban(\$db)*

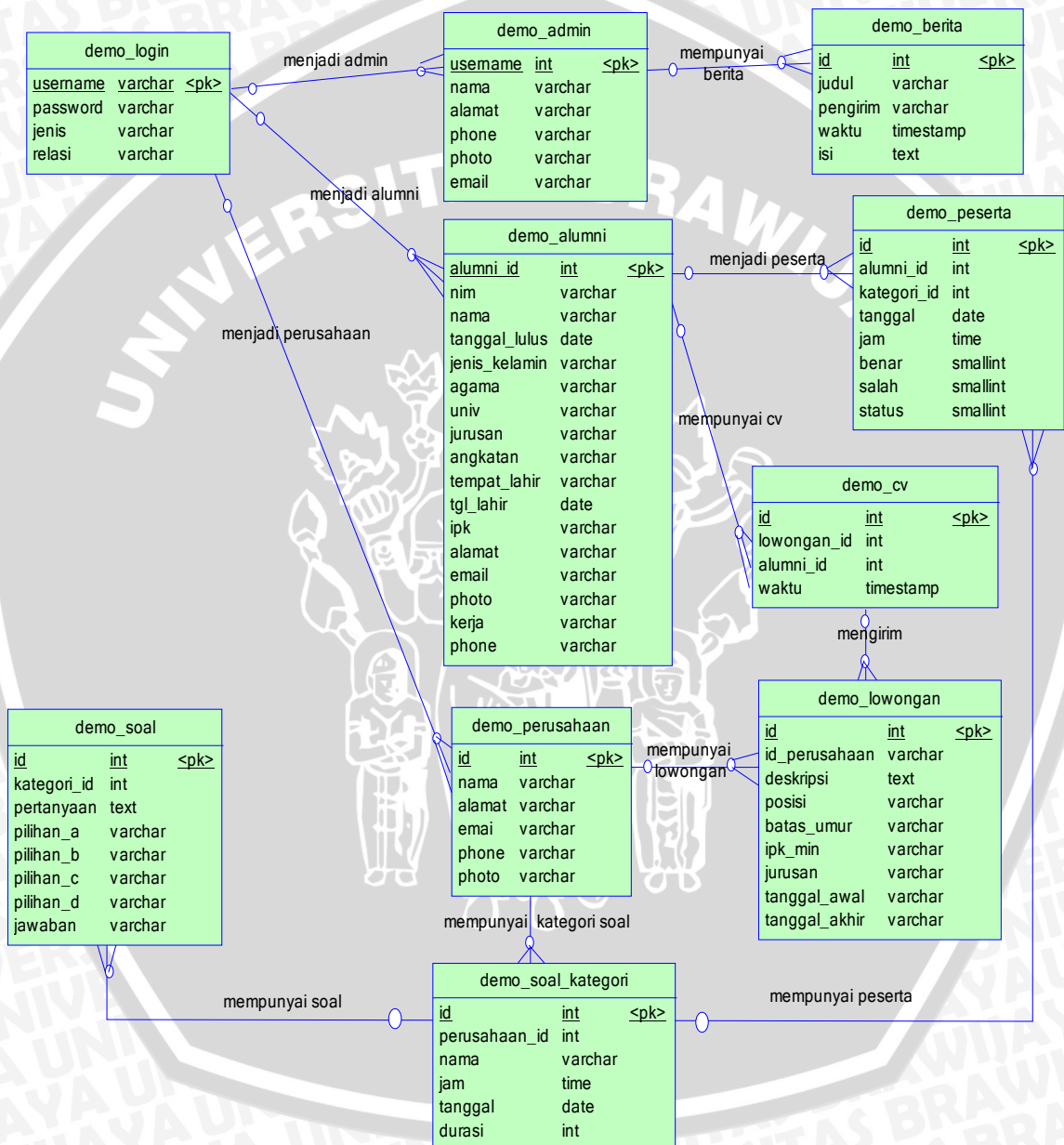
Gambar 4.27 adalah *sequence diagram* untuk *method* *simpanJawaban*. *Method* ini dipergunakan untuk menyimpan jawaban dari masing-masing soal yang telah dikerjakan oleh peserta dan menghitung nilainya. *Sequence diagram* dari *method* *simpanJawaban* adalah sebagai berikut :



Gambar 4.27 : *Sequence Diagram* untuk *method* *simpanJawaban*
Sumber : [Perancangan]

4.2 Perancangan Basis Data

Sistem *job placement center online* memerlukan sebuah basis data. Basis data berfungsi sebagai tempat menyimpan data. Pemodelan basis data dapat dilakukan dengan menggunakan *Entity-Relationship diagram*. Gambar 4.28 adalah gambar ER diagram dari database sistem *job placement center online*.



Gambar 4.28 : ER Diagram Database Sistem Job Placement Center Online

Sumber : [Perancangan]

4.2.1 Data Object Description untuk Tabel Demo_Admin

Tabel 4.46 : *Data Object Description* untuk Tabel demo_admin

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Username	varchar	30
	Nama	varchar	30
	Alamat	varchar	50
	Phone	varchar	20
	Foto	varchar	100
	Email	varchar	50

Sumber : [Perancangan]

Tabel demo_admin mempunyai enam *field*. *Field* id merupakan *primary key* dari table admin. *Field* id tidak boleh kosong. Tabel demo_admin berisi data-data tentang identitas admin dalam *job placement center online*.

4.2.2 Data Object Description untuk Tabel Demo_Alumni

Tabel 4.47 : *Data Object Description* untuk Tabel demo_alumni

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Alumni_Id	int	10
	NIM	varchar	20
	Nama	varchar	50
	Tanggal_lulus	date	
	Jenis_kelamin	enum('L','P')	
	Agama	varchar	20
	Univ	varchar	100
	Jurusan	varchar	30
	Angkatan	varchar	5
	Tempat_lahir	varchar	20
	Tgl_lahir	date	
	IPK	float	
	Alamat	varchar	100
	Email	varchar	50
	Foto	varchar	100
	Kerja	varchar	20
	Phone	varchar	30

Sumber : [Perancangan]

Tabel demo_alumni mempunyai enam belas *field*. *Field* alumni_id merupakan *primary key* dari table demo_alumni. *Field* alumni_id tidak boleh kosong. Tabel demo_alumni berisi data-data tentang biodata alumni dalam *job placement center online*.

4.2.3 Data Object Description untuk Tabel Demo_Berita

Tabel 4.48 : *Data Object Description* untuk Tabel demo_berita

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Id	int	10
	Judul	varchar	50
	Pengirim	varchar	20
	Waktu	timestamp	
	Isi	text	

Sumber : [Perancangan]

Tabel demo_berita mempunyai lima *field*. *Field* id merupakan *primary key* dari tabel demo_berita. Tabel demo_berita berisi data-data tentang berita-berita tentang Politeknik Negeri Malang.

4.2.4 Data Object Description untuk Tabel Demo_CV

Tabel 4.49 : *Data Object Description* untuk Tabel demo_cv

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Id	int	10
	Lowongan_id	int	11
	Alumni_id	int	11
	Waktu	timestamp	

Sumber : [Perancangan]

Tabel demo_cv mempunyai empat *field*. *Field* id merupakan *primary key* dari tabel demo_cv. Tabel demo_cv berisi data-data tentang berkas lamaran yang dikirim alumni perusahaan yang menjadi *client job placement center* Politeknik Negeri Malang.

4.2.5 Data Object Description untuk Tabel Demo_Login

Tabel 4.50 : *Data Object Description* untuk Tabel demo_login

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Username	varchar	30
	Password	varchar	100
	Jenis	varchar	20
	Relasi	varchar	5

Sumber : [Perancangan]

Tabel demo_login mempunyai empat *field*. Tabel demo_login berisi data-data tentang user-user yang menjadi *client job placement center online* Politeknik Negeri Malang.

4.2.6 Data Object Description untuk Tabel Demo_Lowongan

Tabel 4.51 : *Data Object Description* untuk Tabel demo_Lowongan

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Id	int	7
	Id_perusahaan	varchar	50
	Deskripsi	text	
	Posisi	varchar	50
	Batas_umur	varchar	5
	IPK_min	varchar	5
	Jurusan	varchar	30
	Tanggal_awal	varchar	11
	Tanggal_akhir	varchar	11

Sumber : [Perancangan]

Tabel demo_lowongan mempunyai sembilan *field*. *Field* id merupakan *primary key* dari tabel demo_lowongan. Tabel demo_lowongan berisi tentang lowongan kerja dari perusahaan yang membutuhkan tenaga kerja di *job placement center online* Politeknik Negeri Malang.

4.2.7 Data Object Description untuk Tabel Demo_Perusahaan

Tabel 4.52 : *Data Object Description* untuk Tabel demo_Perusahaan

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Id	int	11
	Nama	varchar	50
	Alamat	varchar	100
	Email	varchar	50
	Phone	varchar	30
	Photo	varchar	100

Sumber : [Perancangan]

Tabel perusahaan mempunyai enam *field*. *Field* id merupakan *primary key* dari table perusahaan. *Field* id tidak boleh kosong. Tabel perusahaan berisi data-data tentang profil perusahaan yang menjadi *client* Politeknik Negeri Malang.

4.2.8 Data Object Description untuk Tabel Demo_Peserta

Tabel 4.53 : *Data Object Description* untuk Tabel demo_Peserta

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Id	int	11
	Alumni_id	int	11
	Kategori_id	int	11
	Tanggal	date	
	Jam	time	
	Benar	smallint	6
	Salah	smallint	6
	Status	smallint	6

Sumber : [Perancangan]

Tabel demo_peserta mempunyai delapan *field*. *Field* id merupakan *primary key* dari tabel user. *Field* id tidak boleh kosong. Tabel demo_peserta yang berisi tentang alumni yang akan mengikuti *test online* di *job placement center online*.

4.2.9 Data Object Description untuk Tabel Demo_Soal

Tabel 4.54 : *Data Object Description* untuk Tabel demo_Soal

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Id	int	10
	Kategori_id	int	11
	Pertanyaan	text	
	Pilihan_a	varchar	50
	Pilihan_b	varchar	50
	Pilihan_c	varchar	50
	Pilihan_d	varchar	50
	Jawaban	varchar	1

Sumber : [Perancangan]

Tabel demo_soal mempunyai delapan *field*. *Field* id merupakan *primary key* dari tabel demo_soal. *Field* id tidak boleh kosong. Tabel demo_soal yang berisi tentang soal-soal yang disediakan oleh perusahaan yang menjadi *client job placement center online*.

4.2.10 Data Object Description untuk Tabel Demo_Soal_Kategori

Tabel 4.55 : *Data Object Description* untuk Tabel demo_Soal_Kategori

KEY	COLUMN NAME	DATA TYPE	SIZE
PK	Id	int	10
	perusahaan_id	int	10
	Nama	varchar	30
	Tanggal	date	
	Jam	Time	
	Durasi	int	11

Sumber : [Perancangan]

Tabel demo_Soal_Kategori mempunyai enam *field*. *Field* id merupakan *primary key* dari tabel demo_Soal_Kategori. *Field* id tidak boleh kosong. Tabel demo_Soal_Kategori yang berisi tentang kategori soal yang disediakan oleh perusahaan yang menjadi *client job placement center online*.

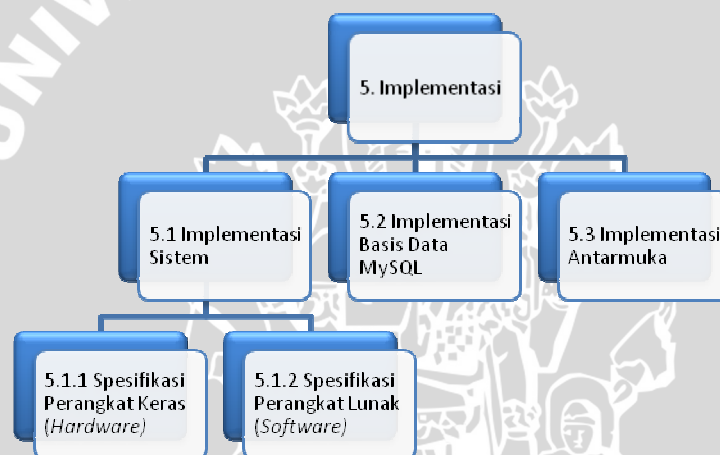


BAB V

IMPLEMENTASI

Bab ini membahas tentang implementasi perangkat lunak *job placement center 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 *job placement center online* yang nyata agar bisa berfungsi sesuai dengan kebutuhan. Sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Sistem *job placement center online* merupakan jenis aplikasi *client-server*. Dalam skripsi ini, baik *client* maupun *server* terdapat dalam satu komputer yang sama. Pengembangan sistem *job placement center online* menggunakan

sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Nama Komponen	Spesifikasi
Prosesor	AMD Athlon 1,7 GHz
Memori (RAM)	512 MB
Hardisk	80 GB
VGA Card	Ati Radeon 128 MB

Sumber : [Implementasi]

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan sistem *job placement center 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
IDE (<i>Integrated Development Environment</i>)	WebBuilder 2008

Sumber : [Implementasi]

5.2 Implementasi Basis Data MySQL

Implementasi perancangan basis data *test_online* dilakukan sesuai dengan rancangan tabel-tabel. Implementasi perancangan basis data *test_online* menggunakan *Data Definition Language* (DDL). DDL adalah struktur basis data yang menggambarkan desain basis data secara keseluruhan [FAT-02:15].

DDL yang digunakan dalam membentuk basis data *test_online* adalah sebagai berikut :

```
create database test_online;
```

```

CREATE TABLE `demo_admin` (
  `username` varchar(30) collate
latin1_general_ci NOT NULL,
  `nama` varchar(30) collate
latin1_general_ci NOT NULL,
  `alamat` varchar(50) collate
latin1_general_ci NOT NULL,
  `phone` varchar(20) collate
latin1_general_ci NOT NULL,
  `photo` varchar(100) collate
latin1_general_ci NOT NULL,
  `email` varchar(50) collate
latin1_general_ci NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci;

CREATE TABLE `demo_alumni` (
  `alumni_id` int(10) unsigned NOT
NULL auto_increment,
  `nim` varchar(20) collate
latin1_general_ci NOT NULL,
  `nama` varchar(50) collate
latin1_general_ci NOT NULL,
  `tanggal_lulus` date NOT NULL,
  `jenis_kelamin` enum('L','P')
collate latin1_general_ci NOT NULL
default 'L',
  `agama` varchar(20) collate
latin1_general_ci NOT NULL,
  `jurusan` varchar(30) collate
latin1_general_ci NOT NULL,
  `angkatan` varchar(5) collate
latin1_general_ci NOT NULL,
  `tempat_lahir` varchar(20)
collate latin1_general_ci NOT NULL,
  `tanggal_lahir` date NOT NULL,
  `alamat` varchar(100) collate
latin1_general_ci NOT NULL,
  PRIMARY KEY (`alumni_id`)
) ENGINE=MyISAM DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci;

CREATE TABLE `demo_perusahaan` (
  `id` int(11) NOT NULL
auto_increment,
  `nama` varchar(50) collate
latin1_general_ci NOT NULL,
  `alamat` varchar(100) collate
latin1_general_ci NOT NULL,
  `email` varchar(50) collate
latin1_general_ci NOT NULL,
  `phone` varchar(30) collate
latin1_general_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM
AUTO_INCREMENT=12 DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=12 ;

CREATE TABLE `demo_peserta` (
  `id` int(11) unsigned NOT NULL
auto_increment,
  `alumni_id` int(11) NOT NULL
default '0',
  `kategori_id` int(11) NOT NULL
default '0',
  `tanggal` date NOT NULL default
'0000-00-00',
  `jam` time NOT NULL default
'00:00:00',
  `benar` smallint(6) NOT NULL
default '0',
  `salah` smallint(6) NOT NULL
default '0',
  `status` smallint(6) NOT NULL
default '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `alumni_id`
(`alumni_id`,`kategori_id`)
) ENGINE=MyISAM DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=12 ;

```



```

latin1_general_ci NOT NULL,
`email` varchar(50) collate
latin1_general_ci NOT NULL,
`photo` varchar(100) collate
latin1_general_ci NOT NULL,
`kerja` varchar(30) collate
latin1_general_ci NOT NULL,
`phone` varchar(20) collate
latin1_general_ci NOT NULL,
PRIMARY KEY (`alumni_id`),
UNIQUE KEY `nim` (`nim`)
) ENGINE=MyISAM AUTO_INCREMENT=12
DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=12 ;

CREATE TABLE `demo_berita` (
`id` int(10) unsigned NOT NULL
auto_increment,
`judul` varchar(50) collate
latin1_general_ci NOT NULL,
`pengirim` varchar(20) collate
latin1_general_ci NOT NULL,
`waktu` timestamp NOT NULL
default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
`isi` text collate
latin1_general_ci NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM
AUTO_INCREMENT=5 DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=5 ;

CREATE TABLE `demo_cv` (
`id` int(10) unsigned NOT NULL
) ENGINE=MyISAM
AUTO_INCREMENT=4 DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=4 ;

CREATE TABLE `demo_soal` (
`id` int(10) unsigned NOT NULL
auto_increment,
`kategori_id` int(11) NOT NULL
default '0',
`pertanyaan` text collate
latin1_general_ci NOT NULL,
`pilihan_a` varchar(50) collate
latin1_general_ci NOT NULL,
`pilihan_b` varchar(50) collate
latin1_general_ci NOT NULL,
`pilihan_c` varchar(50) collate
latin1_general_ci NOT NULL,
`pilihan_d` varchar(50) collate
latin1_general_ci NOT NULL,
`jawaban` varchar(1) collate
latin1_general_ci NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=5
DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=5 ;

CREATE TABLE `demo_soal_kategori`
(
`id` int(10) unsigned NOT NULL
auto_increment,
`perusahaan_id` int(10) unsigned
NOT NULL,
`nama` varchar(30) collate
latin1_general_ci NOT NULL,
`tanggal` date NOT NULL,
) ENGINE=MyISAM
AUTO_INCREMENT=4 DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=4 ;

```

```

auto_increment,
`lowongan_id` int(11) NOT NULL,
`alumni_id` int(11) NOT NULL,
`waktu` timestamp NOT NULL
default CURRENT_TIMESTAMP,
PRIMARY KEY (`id`),
UNIQUE KEY `lowongan_id`
(`lowongan_id`,`alumni_id`)
) ENGINE=MyISAM AUTO_INCREMENT=5
DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=5 ;

CREATE TABLE `demo_login` (
`username` varchar(30) collate
latin1_general_ci NOT NULL,
`password` varchar(100) collate
latin1_general_ci NOT NULL,
`jenis` varchar(20) collate
latin1_general_ci NOT NULL,
`relasi` varchar(5) collate
latin1_general_ci NOT NULL,
PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT
CHARSET=latin1
COLLATE=latin1_general_ci;
`jam` time NOT NULL,
`durasi` int(11) unsigned NOT
NULL default '0',
PRIMARY KEY (`id`),
KEY `kode` (`perusahaan_id`)
) ENGINE=MyISAM AUTO_INCREMENT=5
DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci
AUTO_INCREMENT=5 ;

```


Basis data *job placement center* yang telah diimplementasikan pada basis data MySQL oleh phpMyAdmin ditunjukkan dalam Gambar 5.2.

Tabel	Aksi	Catatan	Jenis	Penyortiran	Ukuran	Kelebihan (Overhead)
demo_admin	        	1	MyISAM	latin1_general_ci	2,1 KB	-
demo_alumni	        	8	MyISAM	latin1_general_ci	3,5 KB	-
demo_berita	        	4	MyISAM	latin1_general_ci	2,4 KB	-
demo_cv	        	5	MyISAM	latin1_general_ci	3,1 KB	-
demo_login	        	12	MyISAM	latin1_general_ci	2,7 KB	-
demo_lowongan	        	6	MyISAM	latin1_general_ci	2,7 KB	-
demo_perusahaan	        	9	MyISAM	latin1_general_ci	2,5 KB	-
demo_peserta	        	1	MyISAM	latin1_general_ci	3,0 KB	-
demo_soal	        	3	MyISAM	latin1_general_ci	2,1 KB	-
demo_soal_kategori	        	4	MyISAM	latin1_general_ci	3,1 KB	-
tabel 10	Jumlah	53	MyISAM	latin1_general_ci	27,2 KB	0 Bytes

Gambar 5.2 Implementasi Basis Data *job placement center online* pada MySQL oleh phpMyAdmin

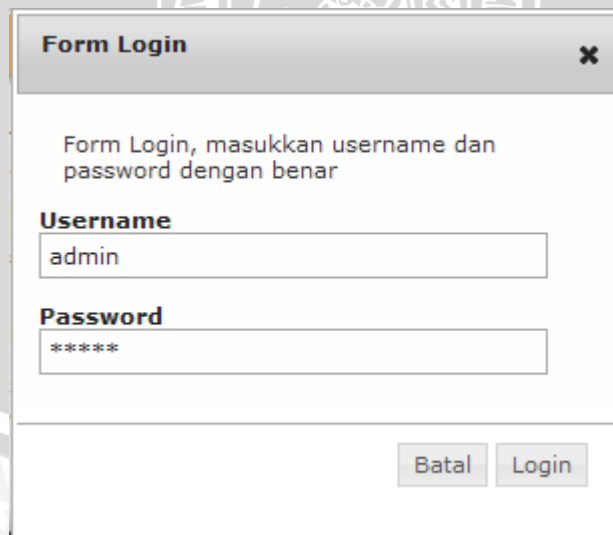
Sumber: Implementasi

5.3 Implementasi Antarmuka Kebutuhan Fungsional

Subbab 5.3 membahas semua implementasi antarmuka yang didapat dari daftar kebutuhan fungsional.

5.3.1 Implementasi Antarmuka Login

Gambar berikut ini merupakan adalah contoh login sebagai ‘administrator’.



Gambar 5.3 : Halaman Login

Sumber : [Implementasi]

Listing program yang digunakan dalam login adalah sebagai berikut :

```
function process()
```

```
* default value untuk return function
```

```
$code = "success";
```

```
$message = "";
```

```
# load table model
```

```
$this->load->model("login_model");
```

```
$this->login_model->_where = array(
```

```
    'username' => $_POST['username'],
```

```
    'password' => md5($_POST['password'])
```

```
# read table execute
```

```
$row = $this->login_model->read();
```

```
if(count($row) > 0)
```

```
# login sukses, setting session
```

```
$row = $row[0];
```

```
$session = array (
```

```
    'login_username' => $row->username,
```

```
    'login_password' => $row->password,
```

```
    'login_jenis' => $row->jenis,
```

```
    'login_relasi' => $row->relasi
```

```
$this->session->set_userdata($session);
```

```
($row->jenis == "perusahaan")
```

```
# load table model
```

```
$this->load->model("perusahaan_model", 'table_model');
```

```
$this->table_model->_where = array ('id' => $row
```

```
>relasi);
```

```
$row = $this->table_model->read();
```

```
$session = array ()
```

```
    else
```

```
# login gagal
```

```
# destroy session login
```

```
$this->session->unset_userdata('login_username');
```

```
$this->session->unset_userdata('login_password');
```

```
$this->session->unset_userdata('login_jenis');
```

```
# response code error
```

```
$code = "error";
```

```
$message = "Login gagal, silahkan mengulangi kembali";
```

```
echo '{"code":"' . $code . '", "message":"' . $message . '"}';
```

```
function logoff()
```

```
$this->session->unset_userdata('login_username');
#$this->session->unset_userdata('login_password');
#$this->session->unset_userdata('login_jenis');
#$this->session->unset_userdata('login_relasi');
#$this->session->unset_userdata('login_name');
$this->session->sess_destroy();
redirect_js(base_url());
```

5.3.2 Implementasi Antarmuka Home

Halaman home untuk web *job placement center online* berisi link-link untuk mengatur seluruh kebutuhan semua user *job placement center online* Politeknik Negeri Malang seperti ditunjukkan pada gambar 5.4 berikut:

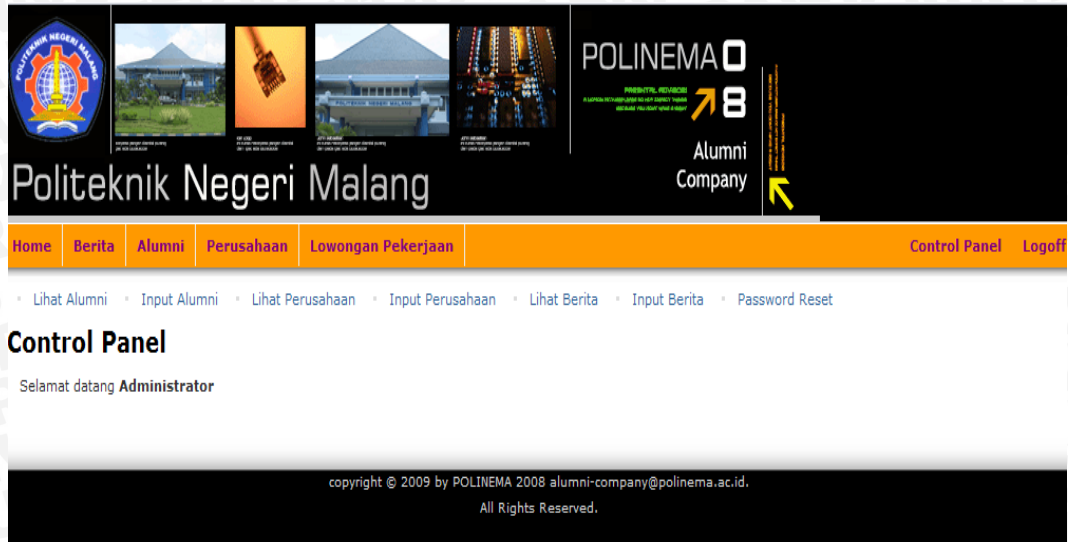


Gambar 5.4 : Halaman Home (Halaman Awal)

Sumber : [Implementasi]

5.3.3 Implementasi Antarmuka Administrator






Halaman admin berisi fasilitas admin yang terdaftar dalam *job placement center online*. User dibagi menjadi tiga macam, yaitu admin, alumni, dan perusahaan. Perbedaan ketiga fungsi petugas tersebut adalah pada hak aksesnya dalam program aplikasi. Halaman antarmuka administrator ditunjukkan seperti gambar 5.5 berikut:



Gambar 5.5 : Halaman Administrator
Sumber : [Implementasi]

5.3.4 Implementasi Antarmuka Input Alumni

Halaman input Alumni berisi form untuk menambahkan data alumni. Form input alumni terdiri dari biodata para alumni yang menjadi anggota *job placement center online* Politeknik Negeri Malang dan akan tersimpan didalam database. Form input alumni ditunjukkan seperti gambar 5.6 berikut:

No Induk	<input type="text"/>
Nama	<input type="text"/>
Jenis Kelamin	<input type="radio"/> Laki-Laki <input checked="" type="radio"/> Perempuan
Tempat Tanggal Lahir	<input type="text"/> <input type="text"/> 
Alamat	<input type="text"/>
Agama	Budha 
Alumnus	Politeknik Negeri Malang  Tambah
Tanggal Lulus	<input type="text"/> 
Jurusan	Elektro  Tambah
Angkatan	<input type="text"/>
IPK	<input type="text"/>
Lokasi Kerja	<input type="text"/>
Email	<input type="text"/>
Telepon	<input type="text"/>
Foto	<input type="text"/> Choose...
Username	<input type="text"/>
Password	<input type="text"/>
Simpan Data	

Gambar 5.6 : Legend Input Data Alumni

Sumber : [Implementasi]

Listing program yang digunakan dalam input alumni adalah sebagai berikut :

```
function input_process()
    $code = "success";
    $message = "Data berhasil disimpan";
    $this->load->helper(array('form', 'url'));
    $this->load->library('validation');
    # set rules validation
    $rules['nim'] = "trim|required";
    $rules['nama'] = "trim|required";
    $rules['email'] = "trim|valid_email";
    $rules['univ'] = "trim|required";
    $rules['jurusan'] = "trim|required";
    if($_POST['edit_mode'] == 0)
```

```

$rules['username'] = "trim|required";
$rules['password'] = "trim|required";
$this->validation->set_rules($rules);
# set alias name for field
$fields['nim']='No Induk';
$fields['nama']='Nama';
$fields['email']='Email';
$fields['univ']='Alumnus';
$fields['jurusan']='Jurusan';
if($_POST['edit_mode'] == 0)
$fields['username']='Username';
$fields['password']='Password';
$this->validation->set_fields($fields);
if ($this->validation->run() == FALSE)

    $code = "error";
    $message = $this->validation->_error_array[0];
    $this->print_json($code, $message);
    exit;

# load table model database
$this->load->model('login_model');
$this->load->model('alumni_model');
$data = array (
    'nim'=> $_POST['nim'],
    'nama'=>$_POST['nama'],
    'alamat'=>$_POST['alamat'],
    'tempat_lahir'=>$_POST['tempat_lahir'],
    'tanggal_lahir'=>$_POST['tanggal_lahir'],
    'jenis_kelamin'=>$_POST['jenis_kelamin'],
    'agama'=>$_POST['agama'],
    'email'=>$_POST['email'],
    'tanggal_lulus'=>$_POST['tanggal_lulus'],
    'jurusan'=>$_POST['jurusan'],
    'angkatan'=>$_POST['angkatan'],
    'ipk'=>$_POST['ipk'],
    'kerja'=>$_POST['kerja'],
    'phone'=>$_POST['phone'],
    'univ'=>$_POST['univ']
    if($_POST['edit_mode'] == 1)
    $this->alumni_model->_where = array('nim' =>
    $_POST['edit_id']);

```

5.3.5 Implementasi Antarmuka Lihat Alumni

Halaman Lihat alumni berisi daftar-daftar alumni yang menjadi anggota *job placement center*. user dibagi menjadi tiga macam, yaitu admin, alumni, dan perusahaan. Perbedaan ketiga fungsi petugas tersebut adalah pada hak aksesnya dalam program aplikasi. Form lihat alumni terdiri dari dua legend, yaitu legend lihat semua alumni dan legend lihat biodata alumni secara detail. Seperti ditunjukkan pada gambar 5.7 dan 5.8 berikut:

No Induk	Nama	Tempat Lahir	Tanggal Lahir	Jurusan	Angkatan	Tanggal Lulus
0430091234-99	Ratri Pawening		0 0 0	Elektro		0000-00-00
0430091235-99	Rukma Tri Kuncoro		0 0 0			0000-00-00
0430091236-99	Saiful Haqqi		0 0 0			0000-00-00
0430091237-99	Surya Fajariyanta		0 0 0			0000-00-00
0430091238-99	baru lagi		0 0 0			0000-00-00
0610632001	achmad mufti	Sumenep	4 Februari 1981	Informatika	2006	2009-07-15
0961632001	aisy	sumenep	23 Nopember 1987	B. Inggris	2009	2009-07-31
aku	baru lagi		0 0 0			0000-00-00
baru	oio k kabeh		0 0 0			0000-00-00

Gambar 5.7 : Halaman Lihat Alumni

Sumber : [Implementasi]



No Induk	0610632001
Nama	achmad mufti
TTL	Sumenep, 1981-02-04
Address: http://localhost/images/alumni/0610632001.jpg	
Alamat	Sumenep madura
Agama	Islam
Alumnus	unibraw
Angkatan	2006
Jurusan	Informatika
Tanggal Lulus	2009-07-15
IPK	3
Lokasi Kerja	malang
Email	mufti81@yahoo.com
Telepon	081555610991

Ubah Hapus

Gambar 5.8 : Halaman Alumni Secara Detail

Sumber : [Implementasi]

5.3.6 Implementasi Antarmuka Input Perusahaan

Halaman input perusahaan berisi form untuk menambahkan data perusahaan. Form input perusahaan terdiri dari biodata perusahaan yang menjadi *client job placement center* Politeknik Negeri Malang dan akan tersimpan didalam database. Form input perusahaan ditunjukkan pada gambar 5.9.

Nama	<input type="text" value="PT.Angkasa Raya"/>
Alamat	<input type="text" value="Jl.Ijen besar no.99 Malang"/>
Email	<input type="text" value="angkasa@gmail.co"/>
Telepon	<input type="text" value="0341478572"/>
Foto	<input type="text" value="K:\Gambar\LOGO\Pol"/> <input type="button" value="Choose..."/>
Username	<input type="text" value="angkasa"/>
Password	<input type="password" value="*****"/>
<input type="button" value="Simpan Data"/>	

Gambar 5.9 : Legend Input Perusahaan
Sumber : [Implementasi]

Listing program yang digunakan dalam input perusahaan adalah sebagai berikut :

```
function input_process()
    # directory tempat upload gambar
    $img_dir = "./images/perusahaan/";
    $code = "success";
    $message = "Data berhasil disimpan";
    $this->load->helper(array('form', 'url'));
    $this->load->library('validation');

    # set rules validation
    $rules['nama'] = "trim|required";
    $rules['email'] = "trim|valid_email";
    if($_POST['edit_mode'] == 0)

    $rules['username'] = "trim|required";
    $rules['password'] = "trim|required";
    $this->validation->set_rules($rules);

    # set alias name for field
    $fields['nama']='Nama';
    $fields['email']='Email';
    if($_POST['edit_mode'] == 0)

    $fields['username']='Username';
    $fields['password']='Password';

    $this->validation->set_fields($fields);

    if ($this->validation->run() == FALSE)
    $code = "error";
    $message = $this->validation->_error_array[0];
    exit;
```

```

# load table model database
$this->load->model('login_model');
$this->load->model('perusahaan_model', 'table_model');

$data = array (
    'nama'=>$_POST['nama'],
    'alamat'=>$_POST['alamat'],
    'email'=>$_POST['email'],
    'phone'=>$_POST['phone'] );

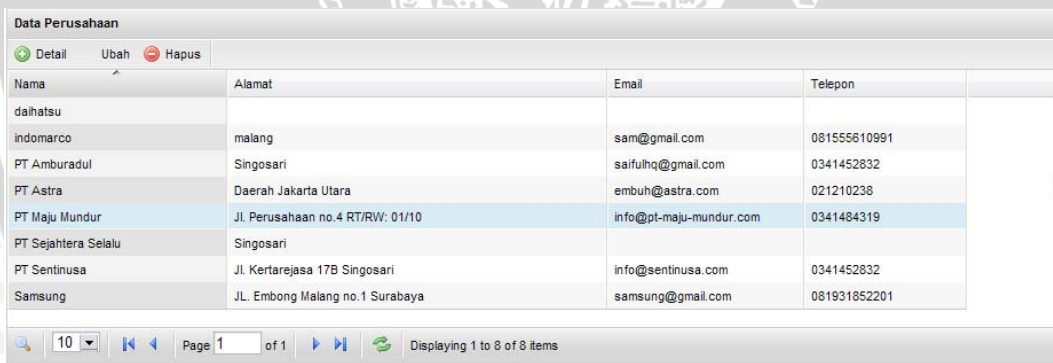
# nama file upload
$img_nama = "aa";

if($_POST['edit_mode'] == 1)
    $this->table_model->_where = array('id' =>
    $_POST['edit_id']);

```

5.3.7 Implementasi Antarmuka Lihat Perusahaan

Halaman Lihat perusahaan berisi daftar-daftar perusahaan yang menjadi *client job placement center*. user dibagi menjadi tiga macam, yaitu admin, alumni, dan perusahaan. Perbedaan ketiga fungsi petugas tersebut adalah pada hak aksesnya dalam program aplikasi. Form lihat perusahaan terdiri dari dua legend, yaitu legend lihat semua perusahaan dan legend lihat perusahaan secara detail seperti gambar 5.10 dan 5.11.



Nama	Alamat	Email	Telepon
daihatsu			
indomarco	malang	sam@gmail.com	081555610991
PT Amburadul	Singosari	saifulhq@gmail.com	0341452832
PT Astra	Daerah Jakarta Utara	embuh@astra.com	021210238
PT Maju Mundur	Jl. Perusahaan no.4 RT/RW: 01/10	info@pt-maju-mundur.com	0341484319
PT Sejahtera Selalu	Singosari		
PT Sentinusa	Jl. Kertarejasa 17B Singosari	info@sentinusa.com	0341452832
Samsung	JL. Embong Malang no.1 Surabaya	samsung@gmail.com	081931852201

Gambar 5.10 : Halaman Lihat Perusahaan

Sumber : [Implementasi]



Nama	Pertamina
Alamat	Jl.Untung Suropati no.9 Malang
Email	pertamina@gmail.com
Telepon	0341478571

Ubah Hapus

Gambar 5.11 : Halaman Perusahaan Secara Detail

Sumber : [Implementasi]

5.3.8 Implementasi Antarmuka Input Berita

Halaman input berita berisi form untuk menambahkan berita-berita seputar *job placement center* dan Politeknik Negeri Malang. Form input berita ditunjukkan seperti gambar 5.12.

Gambar 5.12 : Legend Input Berita

Sumber : [Implementasi]

Listing program yang digunakan dalam input berita adalah sebagai berikut :

```
function input()
{
    $data = array();
    $data = array();
    $data['edit'] = 0;
    $data['edit_id'] = 0;
    $data['row'] = array(
        'judul'=>"",
        'isi'=>"" );
    $this->my_layout->view("admin-berita-input", $data);}

function input_process() {
    $_file_name = "000";
    $code = "success";
    $message = "Data berhasil disimpan";
    $this->load->helper(array('form', 'url'));
    $this->load->library('validation');

    # set rules validation
    $rules['judul'] = "trim|required";
    $rules['isi'] = "trim|required";
    $this->validation->set_rules($rules);

    # set alias name for field
    $fields['judul']='Judul';
    $fields['isi']='Isi';
    $this->validation->set_fields($fields);

    if ($this->validation->run() == FALSE)
    $code = "error";
    $message = $this->validation->_error_array[0];
```



```

$this->print_json($code, $message);
exit; }
# load table model database
$this->load->model('berita_model', 'table_model');
$data = array(
    'judul' => $_POST['judul'],
    'isi' => $_POST['isi'],
    'pengirim' => $this->session->userdata('login_username'));
    if($_POST['edit_mode'] == 1) {
$this->table_model->_where = array('id' => $_POST['edit_id']);

```


5.3.9 Implementasi Antarmuka Lihat Berita

Halaman Lihat berita berisi berita-berita seputar *job placement center* dan Politeknik Negeri Malang. Form lihat berita terdiri dari dua legend, yaitu legend lihat semua jenis berita dan legend lihat berita secara detail seperti gambar 5.13 dan 5.14.

Data Berita	
+ Detail	Ubah - Hapus
Judul	Waktu
pemberitahuan	2009-07-25 08:04:59
woro-woro	2009-07-17 17:25:20
Ketiga	2009-07-10 15:12:50
Nomor 1	2009-07-10 15:08:01
pertama	2009-07-10 15:10:26

Gambar 5.13 : Halaman Lihat Berita

Sumber : [Implementasi]



pemberitahuan
Oleh admin, 2009-07-25 08:04:04

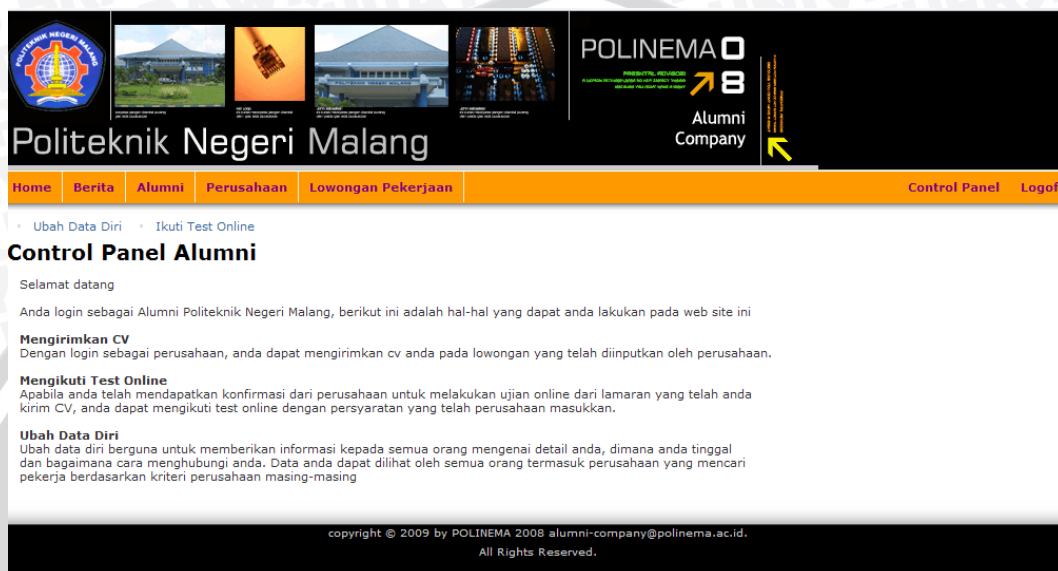
Kiss Mohon hadir pada rapat temu alumni Politeknik Negeri Malang, angkatan pertama sampai sekarang pada hari sabtu tanggal 22-07-2009 jam 8:00 wib. **Cool**

Gambar 5.14 : Halaman Berita Secara Detail

Sumber : [Implementasi]

5.4.10 Implementasi Antarmuka Alumni

Halaman alumni berisi fasilitas yang dapat dilakukan oleh user alumni yang terdiri dari : ubah data diri, lihat lowongan, kirim berkas lamaran dan ikuti *test online*. Halaman antarmuka alumni dalam *job placement center online* seperti gambar 5.15.



Gambar 5.15 : Halaman Alumni

Sumber : [Implementasi]

5.3.11 Implementasi Antarmuka Edit Data Alumni

Halaman edit data alumni berisi form untuk mengubah biodata alumni yang telah tersimpan dalam database. Form edit data alumni seperti gambar 5.16.

Home	Berita	Alumni	Perusahaan	Lowongan Pekerjaan
Ubah Data Diri Ikuti Test Online				
NIM	0610632001			
Nama	<input type="text" value="achmad mufti"/>			
Jenis Kelamin	<input checked="" type="radio"/> Laki-Laki <input type="radio"/> Perempuan			
Tempat Tanggal Lahir	<input type="text" value="Sumenep"/>	<input type="text" value="1981-02-04"/>		
Alamat	<input type="text" value="Sumenep madura"/>			
Agama	<input type="text" value="Islam"/>			
Tanggal Lulus	<input type="text" value="2009-07-15"/>			
Jurusan	<input type="text" value="Informatika"/>			
Angkatan	<input type="text" value="2006"/>			
IPK	<input type="text" value="3"/>			
Lokasi Kerja	<input type="text" value="malang"/>			
Email	<input type="text" value="mufti81@yahoo.com"/>			
Telepon	<input type="text" value="081555610991"/>			
Foto	<input type="text"/>	<input type="button" value="Choose..."/>		
Username	<input type="text" value="mufti"/>			
Password	<input type="text"/>	* Kosongi bila tidak diubah		
<input type="button" value="Simpan Data"/>				

Gambar 5.16 : Legend Detail Form Edit Data Alumni

Sumber : [Implementasi]

5.3.12 Implementasi Antarmuka Kirim Lamaran

Form kirim lamaran merupakan fasilitas dari user alumni untuk mengirim berkas lamaran perusahaan yang menyediakan lowongan pekerjaan sesuai dengan kebutuhan perusahaan. Berikut form untuk mengirim berkas lamaran seperti gambar 5.17.

The screenshot shows a web interface with a navigation bar (Home, Berita, Alumni, Perusahaan, Lowongan Pekerjaan) and a sidebar with links (Ubah Data Diri, Ikuti Test Online). The main content area is titled 'Detail Lowongan Pekerjaan' for 'indomarco'. It lists job details: 'Nama Perusahaan indomarco', 'Nama Lowongan indomarco perkasa', 'Deskripsi dibutuhkan segera', 'IPK minimal 2,50', 'Jurusan Manajemen Informatika', 'Posisi marketing', 'Umur 20-30 tahun', and 'Waktu 2009-07-01 s/d 2009-07-07'. A 'Kirim Lamaran' button is visible. A modal window titled 'Form Upload Lamaran' is open, containing instructions: 'File CV yang anda upload akan otomatis dilihat oleh perusahaan yang membuka lowongan. file format *.doc'. It features a text input field for 'CV', a 'Choose...' button, and 'Batal' and 'Kirim' buttons at the bottom.

Gambar 5.17 : Halaman Kirim Lamaran

Sumber : [Implementasi]

5.4.13 Implementasi Antarmuka Ikuti *Test Online*

Halaman *test online* berisi daftar alumni yang dapat mengikuti test disebuah perusahaan yang menjadi *client job placement center*. Alumni yang dapat melakukan test secara *online* adalah alumni yang telah mengirimkan berkas lamaran sesuai dengan kebutuhan perusahaan. Perusahaan berhak untuk menyeleksi para pelamar, apakah dia berhak atau tidak untuk mengikuti test secara online dan perusahaan berhak untuk menentukan materi soal yang diinginkan. Form *test online* seperti gambar 5.18

The screenshot shows a web interface with a navigation bar (Home, Berita, Alumni, Perusahaan, Lowongan Pekerjaan) and a sidebar with links (Ubah Data Diri, Ikuti Test Online). The main content area displays a table with the following data:

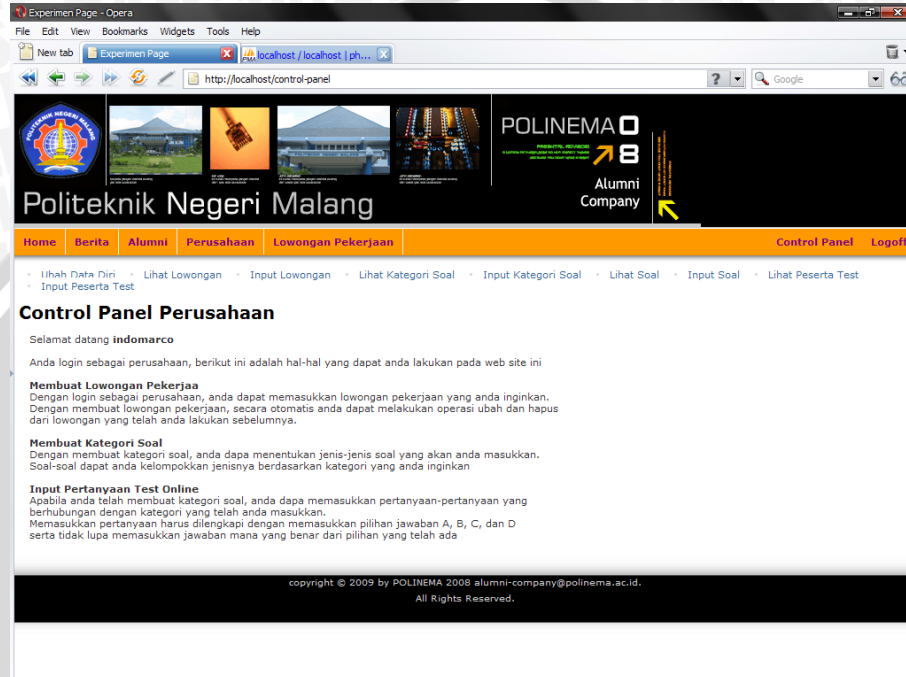
No	Kategori	Waktu Ujian	Jam Ujian	Lama Ujian	
1	psikotes online	2009-07-18	12:00:00	60 menit	Ikuti Test

Gambar 5.18 : Halaman Ikuti *Test Online*

Sumber : [Implementasi]

5.3.14 Implementasi Antarmuka Perusahaan

Halaman Perusahaan berisi fasilitas yang dapat dilakukan oleh user perusahaan yang terdiri dari : ubah data diri, lihat lowongan, input lowongan, lihat kategori soal, input kategori soal, lihat soal, input soal, lihat peserta test dan input peserta test. Seperti gambar 5.19 berikut ini:



Gambar 5.19 : Halaman Perusahaan
Sumber : [Implementasi]

5.3.15 Implementasi Antarmuka Edit Perusahaan

Halaman edit perusahaan berisi form untuk mengubah biodata perusahaan yang telah tersimpan dalam database. Form edit perusahaan seperti gambar 5.20 berikut:

Nama	<input type="text" value="indomarco"/>
Alamat	<input type="text" value="malang"/>
Email	<input type="text" value="sam@gmail.com"/>
Telepon	<input type="text" value="081555610991"/>
Foto	<input type="text"/> <input type="button" value="Choose..."/>
Username	<input type="text" value="indomarco"/>
Password	<input type="password"/> * kosongi bila tidak diubah
<input type="button" value="Simpan Data"/>	

Gambar 5.20 : Form Edit Perusahaan
Sumber : [Implementasi]



5.3.16 Implementasi Antarmuka Input Lowongan

Halaman input lowongan berisi tentang jenis lowongan yang dibutuhkan oleh sebuah perusahaan. Form input lowongan dilakukan oleh user perusahaan-perusahaan yang menyediakan lowongan pekerjaan. Form gambar input lowongan pekerjaan seperti gambar 5.21 berikut:

Nama	<input type="text"/>
Deskripsi	<input type="text"/>
Posisi	<input type="text"/>
Batas Umur	Mulai <input type="text" value="15"/> Sampai <input type="text" value="15"/>
ipk_min	<input type="text"/>
Jurusan	<input type="text" value="Teknik Elektronika"/>
Batas Waktu	Mulai <input type="text"/> Sampai <input type="text"/>
<input type="button" value="Simpan Data"/>	

Gambar 5.21 : Legend Detail pada Form Input Lowongan

Sumber : [Implementasi]

Listing program yang digunakan dalam input lowongan adalah sebagai berikut :

```
function input() {
    $data = array();
    $data['edit'] = 0;
    $data['edit_id'] = 0;
    $data['row'] = array(
        'nama'=>"",
        'deskripsi'=>"",
        'posisi'=>"",
        'batas_umur'=>"",
        'ipk_min'=>"",
        'jurusan'=>"",
        'tanggal_akhir'=>"",
        'tanggal_awal'=>"");
    $this->my_layout->view("pekerjaan-input", $data);
}

function input_process() {
    $code = "success";
    $message = "Data berhasil disimpan";
    $this->load->helper(array('form', 'url'));
    $this->load->library('validation');

    # set rules validation
    $rules['nama'] = "trim|required";
    $rules['deskripsi'] = "trim|required";
```



```

$rules['posisi'] = "trim|required";
$rules['batas_umur'] = "trim|required";
$rules['ipk_min'] = "trim|required";
$rules['jurusan'] = "trim|required";
$rules['tanggal_awal'] = "trim|required";
$rules['tanggal_akhir'] = "trim|required";
$this->validation->set_rules($rules);

# set alias name for field
$fields['nama']='Nama';
$fields['deskripsi']='Deskripsi';
$fields['posisi']='Posisi';
$fields['batas_umur']='Batas Umur';
$fields['ipk_min']='IPK Minimal';
$fields['jurusan']='Jurusan';
$fields['tanggal_awal']='Batas Waktu Awal';
$fields['tanggal_akhir']='Batas Waktu Akhir';
$this->validation->set_fields($fields);

if ($this->validation->run() == FALSE) {
    $code = "error";
    $message = $this->validation->_error_array[0];
    $this->print_json($code, $message);
    exit;
}

# load table model database
$this->load->model('lowongan_model', 'table_model');
$data = array (
    'nama' => $_POST['nama'],
    'id_perusahaan'=>$this->session->userdata['login_relatasi'],
    'deskripsi'=>$_POST['deskripsi'],
    'posisi'=>$_POST['posisi'],
    'batas_umur'=>$_POST['batas_umur'],
    'ipk_min'=>$_POST['ipk_min'],
    'jurusan'=>$_POST['jurusan'],
    'tanggal_awal'=>$_POST['tanggal_awal'],
    'tanggal_akhir'=>$_POST['tanggal_akhir']
);

```

5.3.17 Implementasi Antarmuka Lihat Lowongan

Halaman lihat lowongan berisi daftar lowongan yang diinputkan oleh perusahaan-perusahaan itu sendiri yang terdiri dari: detail, ubah, hapus dan lihat pelamar. Perusahaan dapat mengetahui jumlah pelamar dan melihat berkas lamaran yang dikirim oleh pelamar. Form lihat lowongan terdiri dari dua legend, yaitu legend lihat lowongan dan legend lihat jumlah pelamar. Seperti gambar 5.22 dan 5.23 berikut:

Data Lowongan yang Telah Diinputkan

Detail Ubah Hapus Lihat Pelamar

Nama Lowongan	Posisi	Batas Umur	Jurusan	IPK Minimal
pertamina	direktur utama	15-35	Teknik Elektronika	3.00
pertamina	supervisor	20-30	Manajemen Informatika	2,75
pertamina	skretaris	15-25	Administrasi Niaga	2,5

10 Page 1 of 1 Displaying 1 to 1 of 1 items

Gambar 5.22 : Halaman Lihat Lowongan

Sumber : [Implementasi]

Nama Lowongan	pertamina
Posisi	direktur utama
Batas Umur	15-35
IPK Minimal	3.00
Jurusan	Teknik Elektronika
Waktu	2009-07-01 s/d 2009-07-15

Jumlah Pelamar : 1

No	Nama	Jurusan	Angkatan	IPK	Tanggal Kelulusan	Download CV
1	achmad mufti	Informatika	2006	3	2009-07-15	Download

Gambar 5.23 : Halaman Lihat Pelamar

Sumber : [Implementasi]

5.3.18 Implementasi Antarmuka Input Kategori Soal

Halaman input kategori soal berisi tentang kategori soal yang dibutuhkan oleh sebuah perusahaan. Form input kategori soal dilakukan oleh user perusahaan-perusahaan itu sendiri dan disimpan dalam database. Seperti gambar 5.24 berikut.

Nama	<input type="text" value="Bahasa Indonesia"/>
Tanggal	<input type="text" value="2009-07-25"/>
Jam	<input type="text" value="12:00:00"/> * Format HH:MM:SS
Durasi	<input type="text" value="60"/> * angka dalam menit
<input type="button" value="Simpan Data"/>	

Gambar 5.24 : Legend Form Input Kategori Soal

Sumber : [Implementasi]

5.3.19 Implementasi Antarmuka Lihat Kategori Soal

Halaman lihat kategori soal berisi jenis-jenis kategori soal yang diinputkan oleh perusahaan itu sendiri yang terdiri dari: nama, tanggal, jam dan durasi. Setiap perusahaan dapat melihat kategori soal yang dimiliki oleh perusahaan itu sendiri. Seperti gambar 5.25 berikut.



Nama	Tanggal	Waktu	Durasi (menit)
Bahasa Indonesia	2009-07-25	12:00:00	60
psikotes online	2009-07-18	12:00:00	60
Soal kedua	2009-07-12	21:14:40	80
Soal ketiga	2009-07-17	01:02:03	50
Soal pertama	2009-07-17	12:00:00	60

Gambar 5.25 : Halaman Lihat Kategori Soal

Sumber : [Implementasi]

5.3.20 Implementasi Antarmuka Input Soal

Halaman input soal berisi tentang soal-soal yang akan diinputkan oleh perusahaan itu sendiri tanpa diketahui oleh user yang lain. Form input soal dilakukan oleh user perusahaan itu sendiri dan disimpan dalam database. Seperti gambar 5.26 berikut.



Kategori Soal	Bahasa Indonesia
Pertanyaan	kata-kata dibawah ini yang mana yang termasuk kata baku..?
Pilihan A	aman
Pilihan B	tertip
Pilihan C	pergi
Pilihan D	kamu
Jawaban	a

Simpan Data

Gambar 5.26 : Legend Form Input Soal

Sumber : [Implementasi]

Listing program yang digunakan dalam input lowongan adalah sebagai berikut :

```
function input() {
    $data = array();
    $data['edit'] = 0;
    $data['edit_id'] = 0;
    $data['row'] = array(
        'pertanyaan' => "",
        'pilihan_a' => "",
        'pilihan_b' => "",
        'pilihan_c' => "",
        'pilihan_d' => "",
        'jawaban' => ""
    );
    $this->load->model('kategori_soal_model', 'table_model');
    $this->table_model->order_by = array('nama asc');
    $this->table_model->result_array = TRUE;
    $data['kategori'] = $this->table_model->read();
    $this->my_layout->view("soal-input", $data);
}

function input_process() {
    $code = "success";
    $message = "Data berhasil disimpan";
    $this->load->helper(array('form', 'url'));
    $this->load->library('validation');
    # set rules validation
    $rules['pertanyaan'] = "trim|required";
    $rules['kategori_id'] = "trim|required";
    $rules['pilihan_a'] = "trim|required";
    $rules['pilihan_b'] = "trim|required";
    $rules['pilihan_c'] = "trim|required";
    $rules['pilihan_d'] = "trim|required";
    $rules['jawaban'] = "trim|required";
    $this->validation->set_rules($rules);
    # set alias name for field
    $fields['pertanyaan']='Pertanyaan';
    $fields['kategori_id']='Kategori Soal';
    $fields['pilihan_a']='Pilihan A';
    $fields['pilihan_b']='Pilihan B';
    $fields['pilihan_c']='Pilihan C';
    $fields['pilihan_d']='Pilihan D';
    $fields['jawaban']='Jawaban';
    $this->validation->set_fields($fields);
    if ($this->validation->run() == FALSE) {
        $code = "error";
        $message = $this->validation->error_array[0];
        $this->print_json($code, $message);
        exit;
    }

    # load table model database
    $this->load->model('soal_model', 'table_model');

    $data = array (
        'pertanyaan'=>$_POST['pertanyaan'],
        'pilihan_a'=>$_POST['pilihan_a'],
        'pilihan_b'=>$_POST['pilihan_b'],
        'pilihan_c'=>$_POST['pilihan_c'],
        'pilihan_d'=>$_POST['pilihan_d'],
        'jawaban'=>$_POST['jawaban'],
        'kategori_id'=>$_POST['kategori_id']
    );
}
```

5.3.21 Implementasi Antarmuka Lihat Soal

Halaman lihat soal berisi jenis-jenis kategori soal yang diinputkan oleh perusahaan itu sendiri yang terdiri dari: nama, tanggal, jam dan durasi. Perusahaan dapat melihat kategori soal yang dimiliki oleh perusahaan itu sendiri. Seperti gambar 5.27 berikut:

Kategori	Pertanyaan	Pilihan A	Pilihan B	Pilihan C	Pilihan D	Jawaban
Bahasa Indonesia	kata-kata dibawah ini yang termasuk kata baku adalah..?	aman	tertip	kamu	dia	a
Bahasa Indonesia	penulisan salam pembuka yang benar adalah..?	dengan hormat	Dengan hormat,	Dengan Hormat,	Dengan hormat.	b
Bahasa Indonesia	contoh penulisan kelompok kata yang benar adalah..?	dari pada	Menindaklanjuti	Menanda tangani	Bertandatangani	b

Gambar 5.27 : Halaman Lihat Soal

Sumber : [Implementasi]

5.3.22 Implementasi Antarmuka Input Peserta Test

Halaman input Peserta Test berisi tentang para alumni yang mengirim berkas lamaran pada salah satu perusahaan yang menyediakan lowongan pekerjaan dan telah memenuhi persyaratan yang telah ditentukan oleh perusahaan. Form input peserta test dilakukan oleh user perusahaan itu sendiri dan disimpan dalam database. Seperti gambar 5.28 berikut.

Kategori Soal		Bahasa Indonesia
Nama Lowongan		pertamina
Nama Peserta		
	NIM	Nama
<input type="checkbox"/>	aku	baru lagi
<input checked="" type="checkbox"/>	0610632001	achmad mufti
<input checked="" type="checkbox"/>	0961632001	aisy

Simpan Data

Gambar 5.28 : Legend Form Input Peserta Test

Sumber : [Implementasi]

Listing program yang digunakan dalam input lowongan adalah sebagai berikut :

```
function input() {
    $data = array();
    $data['edit'] = 0;
    $data['edit_id'] = 0;

    $this->load->model('kategori_soal_model');
    $this->kategori_soal_model->_order_by = array ('nama asc');
    $this->kategori_soal_model->_result_array = TRUE;
    $data['kategori'] = $this->kategori_soal_model->read();

    $this->load->model('lowongan_model');
    $_POST['qtype'] = "nama";
    $_POST['query'] = "";
    $_POST['offset'] = 0;
    $_POST['rp'] = 100;
    $data['lowongan'] = $this->lowongan_model->get_lowongan_perusahaan();

    $this->my_layout->view("peserta-input", $data); }

function input_process() {
    $code = "success";
    $message = "Data berhasil disimpan";

    # validasi peserta test
    if(empty($_POST['peserta'])) {
        $code = "error";
        $message = "Anda belum memilih peserta";
        $this->print_json($code, $message);
        exit; }
    # validasi kategori soal
    if(empty($_POST['kategori_id'])) {
        $code = "error";
        $message = "Anda belum memilih kategori soal";
        $this->print_json($code, $message);
        exit; }

    # load table model
    $this->load->model("peserta_model", "table_model");
    foreach ($_POST['peserta'] as $_row) {
        $data = array ( 'kategori_id' =>
            $_POST['kategori_id'], 'alumni_id' => $_row );
        $this->table_model->create($data); }

    $this->print_json($code, $message); }
```


5.3.23 Implementasi Antarmuka Lihat Peserta Test

Halaman lihat peserta test berisi alumni-alumni yang telah mengimkan berkas lamaran pada salah satu perusahaan, serta berisi jumlah nilai dari alumni yang telah melakukan test secara *online*. Perusahaan dapat melihat peserta test dan nilai yang diperoleh oleh. Seperti gambar 5.29 berikut:

Data Peserta Test						
Kategori	NIM	Nama	Tanggal	Jam	Benar	Salah
psikotes online	0610632001	achmad mufti	0000-00-00	00:00:00	0	0
Soal kedua	aku	baru lagi	2009-07-21	08:17:57	2	0

Gambar 5.29 : Halaman Lihat Peserta Test

Sumber : [Implementasi]

5.3.24 Implementasi Antarmuka Lihat Lowongan

Halaman lihat lowongan berisi konfirmasi tentang lowongan pekerjaan dari bermacam-macam perusahaan yang menjadi client *job placement center*. Seperti gambar 5.30 berikut:

The screenshot shows the website for Politeknik Negeri Malang Alumni Company. The main content area displays a grid of job listings. Each listing includes the following information:

- Pertamina:** IPK minimal 2,5; Jurusan Administrasi Niaga; Posisi sekretaris; Umur 15-25; Waktu 2009-07-20 s/d 2009-07-25.
- Pertamina:** IPK minimal 2,75; Jurusan Manajemen Informatika; Posisi supervisor; Umur 20-30; Waktu 2009-07-10 s/d 2009-07-20.
- Pertamina:** IPK minimal 3,00; Jurusan Teknik Elektronika; Posisi direktur utama; Umur 15-35; Waktu 2009-07-01 s/d 2009-07-15.
- indomarco:** IPK minimal 2,50; Jurusan Manajemen Informatika; Posisi marketing; Umur 20-30; Waktu 2009-07-01 s/d 2009-07-07.
- PT Maju Mundur:** IPK minimal 29,00; Jurusan Manajemen Informatika; Posisi test; Umur 15-15; Waktu 2009-07-07 s/d 2009-07-22.
- PT Maju Mundur:** IPK minimal 30,00; Jurusan Manajemen Informatika; Posisi Programmer; Umur 20-23; Waktu 2009-07-01 s/d 2009-07-31.
- PT Astra:** IPK minimal 27,35; Jurusan Manajemen; Posisi TS/HD; Umur 20-22; Waktu 2009-07-01 s/d 2009-07-30.
- PT Sejahtera Selalu:** IPK minimal 40,00; Jurusan Manajemen Informatika; Posisi programmer java; Umur 16-28; Waktu 2009-06-11 s/d 2009-06-27.
- PT Sejahtera Selalu:** IPK minimal 30,06; Jurusan Teknik Informatika; Posisi programmer; Umur 20-22; Waktu 2009-06-11 s/d 2009-06-27.

Gambar 5.30 : Konfirmasi Lihat Lowongan

Sumber : [Implementasi]

5.3.25 Implementasi Antarmuka *Detail* Lihat Lowongan

Halaman detail lihat lowongan berisi form yang sama dengan form tambah lowongan dan berfungsi untuk melihat instansi *client* yang telah tersimpan tanpa mengubahnya. Seperti gambar 5.31 berikut:

Gambar 5.31 : Halaman *Detail* Lihat Lowongan

Sumber : [Implementasi]

5.3.26 Implementasi Antarmuka Cari Perusahaan

Form pencarian yang terletak pada halaman perusahaan berfungsi untuk mencari data perusahaan berdasarkan nama, alamat, email. Seperti gambar 5.32 dan 5.33 berikut:

Gambar 5.32 : Halaman Pencarian Perusahaan jika yang dicari ada dalam Database

Sumber : [Implementasi]



Gambar 5.33 : Halaman Pencarian Perusahaan jika yang dicari tidak ada dalam Database

Sumber : [Implementasi]

5.3.27 Implementasi Antarmuka Detail Lihat perusahaan

Halaman detail lihat data perusahaan berisi form yang sama dengan form tambah perusahaan dan berfungsi untuk melihat instansi *client* yang telah tersimpan tanpa mengubahnya. Seperti gambar 5.34 berikut:

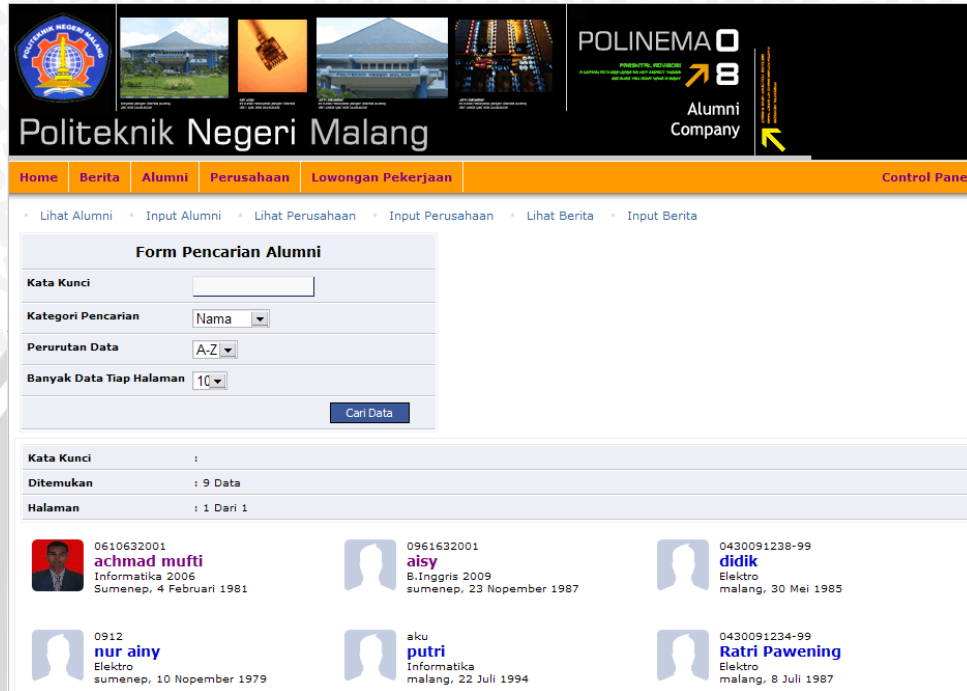


Gambar 5.34 : Halaman *Detail* Lihat Perusahaan

Sumber : [Implementasi]

5.3.28 Implementasi Antarmuka Cari Alumni

Form pencarian yang terletak pada halaman alumni berfungsi untuk mencari data alumni berdasarkan no.induk, nama, alamat, jurusan, angkatan dan email. Seperti gambar 5.35 dan 5.36 berikut:



Gambar 5.35 : Halaman Pencarian Alumni Jika yang Dicari ada dalam Database

Sumber : [Implementasi]



Gambar 5.36: Halaman Pencarian Alumni Jika yang Dicari tidak ada dalam Database

Sumber : [Implementasi]



5.3.29 Implementasi Antarmuka Detail Lihat Alumni

Halaman detail lihat data alumni berisi form yang sama dengan form tambah alumni dan berfungsi untuk melihat data alumni sebagai anggota *job placement center* yang telah tersimpan tanpa mengubahnya. Seperti gambar 5.37 berikut:



The screenshot shows the website interface for Politeknik Negeri Malang Alumni Company. At the top, there is a navigation bar with links: Home, Berita, Alumni, Perusahaan, and Lowongan Pekerjaan. Below the navigation bar, there is a profile card for an alumni. On the left of the profile card is a portrait photo of a man in a suit. On the right is a table containing the following information:

No Induk	0610632001
Nama	achmad mufti
TTL	Sumenep, 1981-02-04
Jenis Kelamin	Laki-laki
Alamat	Sumenep madura 69463
Agama	Islam
Alumnus	unibraw
Angkatan	2006
Jurusan	Informatika
Tanggal Lulus	2009-07-15
IPK	3
Lokasi Kerja	malang
Email	mufti81@yahoo.com
Telepon	081555610991

Gambar 5.37 : Halaman *Detail* Lihat Alumni

Sumber : [Implementasi]

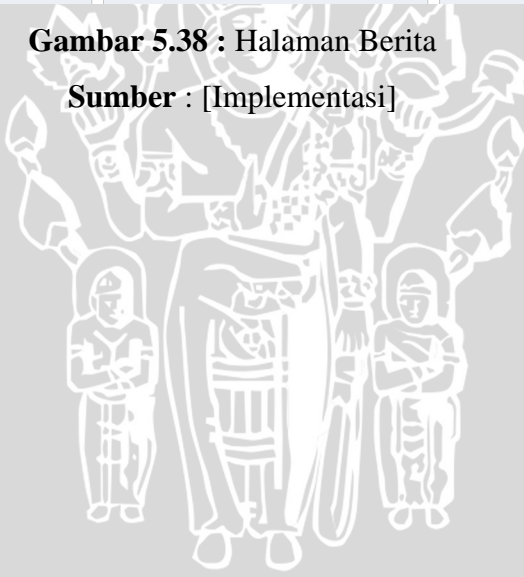
5.3.30 Implementasi Antarmuka Lihat Berita

Halaman lihat berita berisi berita-berita seputar *job placement center* Politeknik Negeri Malang. Seperti gambar 5.38 berikut:



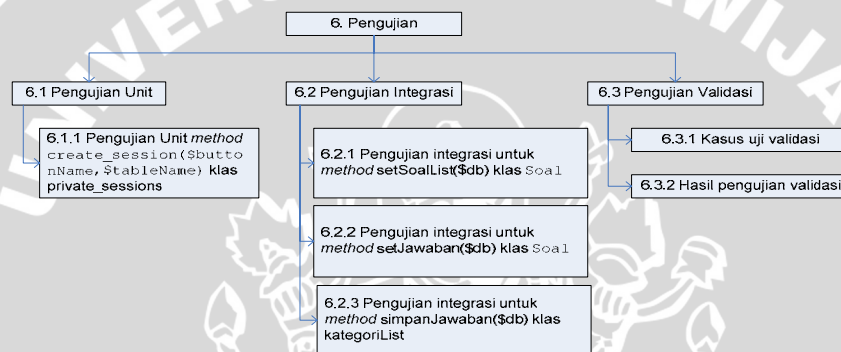
Gambar 5.38 : Halaman Berita

Sumber : [Implementasi]



BAB VI PENGUJIAN DAN ANALISIS

Bab ini membahas proses pengujian terhadap sistem *job placement center* 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*). Proses analisis dilakukan dengan membandingkan sistem aplikasi dengan teori yang ada sehingga didapatkan suatu kesimpulan. Gambar 6.1 menunjukkan langkah-langkah pengujian yang digunakan pada bab ini.



Gambar 6.1 : Diagram Pohon Pengujian
Sumber : Pengujian

6.1 Pengujian

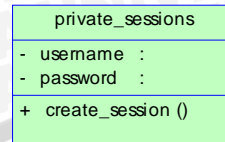
Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian unit, pengujian integrasi dan pengujian validasi.

6.1.1 Pengujian Unit

Sistem yang dibangun dengan paradigma pemrograman berorientasi objek menerapkan pengujian unit untuk suatu metode (operasi) dari suatu kelas. Pada pengujian unit sistem *job placement center 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]

6.1.2 Pengujian Integrasi

Pengujian integrasi diterapkan pada proses yang mengintegrasikan fungsionalitas dari beberapa kelas untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi yang dijadikan sebagai obyek uji adalah kelas-kelas yang menggabungkan kinerja dari kelas-kelas yang lain. Kelas-kelas yang mengintegrasikan beberapa kelas 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 *job placement center 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 `setInputSoal($db)` kelas soal

Method `setInputSoal($db)` ini dipergunakan untuk menentukan kategori soal yang harus dikerjakan oleh peserta. Gambar 6.4 memperlihatkan proses pemodelan dalam *flow graph* pada method `setInputSoal($db)`.

METHOD : `setInputSoal` PARAMETER `db`

DECLARATION :

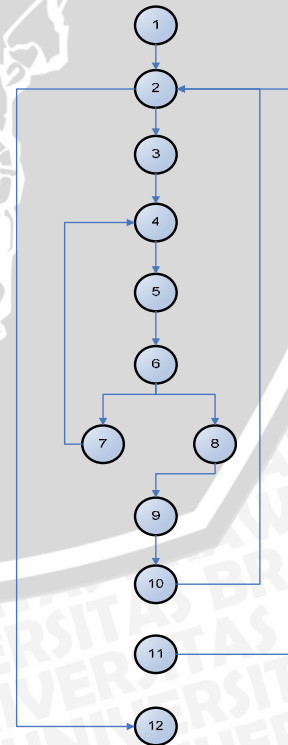
```
db
jumlahSoal
kategoriSoalYgDipilih
dataSoalYangTelahDirandom
controlLooping
kategoriYangDirandom
nilai
kategoriSoalYgDipilih
dataSoal
```

DESCRIPTION :

```
1 Get jumlahSoal
2 Query dataSoal limit jumlahSoal
3 kategoriSoalYgDipilih = array()
4 dataSoalYangTelahDirandom = array(array())
5 FOR i = 0 TO i < jumlahSoal
6   controlLooping = true
7   WHILE (controlLooping)
8     kategoriYangDirandom =
9     random(0, jumlahSoal-1)
10    nilai=in_array(kategoriYangDirandom,
11    kategoriSoalYgDipilih)
12    IF nilai = 1 THEN
13      controlLooping = true
14    ELSE
15      controlLooping = false
16    END IF
17    IF controlLooping = false
18      indexSoalYgDipilih[$i]=
19      kategoriYangDirandom
20      dataSoalYangTelahDirandom[$i]=
21      dataSoal[$kategoriYangDirandom]
22    END IF
23  END WHILE
24 END FOR
```

Node (N) = 12

Edge (E) = 13



Gambar 6.4 : Pemodelan operasi `setInputSoal($db)` ke dalam *flow graph*
Sumber : [Pengujian]

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method setInputSoal*. Jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$.

$$\begin{aligned} V(G) &= 13 - 12 + 2 \\ &= 3 \end{aligned}$$

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 setInputSoal*

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	jumlahSoal = 0	Tidak ada soal	Tidak ada soal
2	controlLooping bernilai false	Soal dimasukkan kedalam array dataSoal	Soal dimasukkan kedalam array dataSoal
3	controlLooping bernilai true	Soal tidak dimasukkan kedalam array dataSoal dan index soal diacak kembali	Soal tidak dimasukkan kedalam array dataSoal dan index soal diacak kembali

Sumber: [Pengujian]

b. Pengujian Integrasi untuk *Method setJawaban(\$db)* Klas soal

Method setJawaban(\$db) 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)*.

METHOD : setJawaban PARAMETER db

DECLARATION :

db

DESCRIPTION :

1 Query dataJawaban

Node (N) = 1

Edge (E) = 0

①

Gambar 6.5 : Pemodelan operasi *setJawaban(\$db)* ke dalam *flow graph*

Sumber : [Pengujian]

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method setJawaban*. Jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$.

$$\begin{aligned} V(G) &= 0 - 1 + 2 \\ &= 1 \end{aligned}$$

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]

c. Pengujian Integrasi untuk Method `simpanJawaban($db)` Kelas `KategoriList`

Method `simpanJawaban($db)` ini dipergunakan untuk menyimpan jawaban dari masing-masing soal yang telah dikerjakan oleh peserta. Gambar 6.6 memperlihatkan proses pemodelan dalam *flow graph* pada *method* `simpanJawaban($db)`.

METHOD : `simpanJawaban` PARAMETER `db`

DECLARATION :
`db`

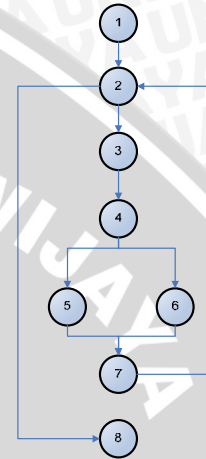
DESCRIPTION :

```

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

```

Node (N) = 8
Edge (E) = 9



Gambar 6.6 : Pemodelan operasi `simpanJawaban($db)` 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	<code>idSoal</code> = 0	Tidak ada soal yang terjawab, nilai 0	Tidak ada soal yang terjawab, nilai 0

Sumber : [Pengujian]

6.1.3 Pengujian Koneksi Basis Data dan Web Server

A. Tujuan

- Pengujian dilakukan untuk mengetahui koneksi *server* basis data MySQL dan *web server* Apache HTTP yang terletak di komputer *server* dalam sebuah jaringan LAN (*Local Area Network*).

B. Prosedur Pengujian

PC *Server*:

- *Window* Command Prompt dijalankan dengan memberikan perintah sebagai berikut:
Start | Run... | cmd
- *Server* basis data MySQL dan *web server* Apache diaktifkan melalui control panel XAMPP.
- Aplikasi dan koneksi yang sedang aktif dapat ditampilkan dengan memberikan perintah sebagai berikut:

```
C:\Documents and Settings\>netstat -an
```

PC *Client*:

- Aplikasi *job placement center* dijalankan dengan menuliskan alamat <http://localhost/> pada halaman *web browser*.
- Aplikasi dan koneksi yang sedang aktif dapat ditampilkan dengan memberikan perintah sebagai berikut:

```
C:\Documents and Settings\Asisten>netstat -an
```

C. Hasil Pengujian dan Analisis

- Perintah `netstat -an` pada komputer *server* yang digunakan sebelum ada koneksi dengan komputer *client* ditunjukkan dalam Gambar 6.7. Perintah tersebut digunakan untuk menampilkan koneksi yang sedang aktif. *Server* basis data MySQL memiliki kondisi (*state*) LISTENING pada alamat lokal 0.0.0.0:3306. *Server* basis data MySQL telah siap untuk menerima sebuah koneksi basis data pada *port* TCP 3306.

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1110	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1187	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1188	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4755	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4758	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4767	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1110	127.0.0.1:1187	ESTABLISHED
TCP	127.0.0.1:1110	127.0.0.1:1297	TIME_WAIT
TCP	127.0.0.1:1110	127.0.0.1:4755	ESTABLISHED
TCP	127.0.0.1:1187	127.0.0.1:1110	ESTABLISHED
TCP	127.0.0.1:4755	127.0.0.1:1110	ESTABLISHED
TCP	127.0.0.1:4766	0.0.0.0:0	LISTENING
TCP	127.0.0.1:4766	127.0.0.1:4767	ESTABLISHED
TCP	127.0.0.1:4767	127.0.0.1:4766	ESTABLISHED

Gambar 6.7: Perintah netstat -an pada komputer *server* sebelum ada koneksi sumber: *Pengujian*

- Perintah netstat -an pada komputer *client* yang digunakan sebelum ada koneksi dengan komputer *server* ditunjukkan dalam Gambar 6.8. Perintah tersebut digunakan untuk menampilkan koneksi yang sedang aktif.

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1031	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1110	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4674	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4676	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4717	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4718	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5000	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1110	127.0.0.1:4717	ESTABLISHED
TCP	127.0.0.1:1110	127.0.0.1:4719	TIME_WAIT
TCP	127.0.0.1:1110	127.0.0.1:4721	TIME_WAIT
TCP	127.0.0.1:1110	127.0.0.1:4723	TIME_WAIT

Gambar 6.8 : Perintah netstat -an pada komputer *client* sebelum ada koneksi sumber: *Pengujian*

- Perintah netstat -an pada komputer *client* yang digunakan setelah ada koneksi dengan komputer *server* ditunjukkan dalam Gambar 6.9. Koneksi terbentuk setelah komputer *client* membuka aplikasi *Job Placement Center*. Koneksi antara komputer *server* (alamat IP: 172.17.8.1) dengan komputer *client* (alamat IP: 172.17.8.10) ditunjukkan dengan adanya kondisi (*state*): ESTABLISHED. Kondisi (*state*): ESTABLISHED menunjukkan komputer *client* dapat terhubung dengan *server* basis data MySQL di komputer *server*.

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1031	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1110	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	0.0.0.0:4759	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5000	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1110	127.0.0.1:4754	TIME_WAIT
TCP	127.0.0.1:1110	127.0.0.1:4756	TIME_WAIT
TCP	127.0.0.1:1110	127.0.0.1:4758	ESTABLISHED
TCP	127.0.0.1:3001	0.0.0.0:0	LISTENING
TCP	127.0.0.1:3002	0.0.0.0:0	LISTENING
TCP	127.0.0.1:3003	0.0.0.0:0	LISTENING
TCP	127.0.0.1:4673	0.0.0.0:0	LISTENING
TCP	127.0.0.1:4673	127.0.0.1:4674	ESTABLISHED
TCP	127.0.0.1:4674	127.0.0.1:4673	ESTABLISHED
TCP	127.0.0.1:4675	0.0.0.0:0	LISTENING
TCP	127.0.0.1:4675	127.0.0.1:4676	ESTABLISHED
TCP	127.0.0.1:4676	127.0.0.1:4675	ESTABLISHED
TCP	127.0.0.1:4752	127.0.0.1:1110	CLOSE_WAIT
TCP	127.0.0.1:4758	127.0.0.1:1110	ESTABLISHED
TCP	172.17.68.248:139	0.0.0.0:0	LISTENING
TCP	172.17.68.248:3519	0.0.0.0:0	LISTENING
TCP	172.17.68.248:4759	172.17.68.243:80	ESTABLISHED
UDP	0.0.0.0:135	*:*	

Gambar 6.9 : Perintah netstat –an pada komputer *client* setelah ada koneksi sumber: *Pengujian*

- Perintah netstat –an pada komputer *server* yang digunakan setelah ada koneksi dengan komputer *client* ditunjukkan dalam Gambar 6.10. Koneksi terbentuk setelah komputer *client* membuka aplikasi *Job Placement Center*. Koneksi antara komputer *server* (alamat IP: 172.17.8.1) dengan komputer *client* (alamat IP: 172.17.8.10) ditunjukkan dengan adanya kondisi (*state*): ESTABLISHED. Kondisi (*state*): ESTABLISHED menunjukkan bahwa pertukaran data antara komputer *server* dan komputer *client* dapat dilakukan.

TCP	127.0.0.1:4766	0.0.0.0:0	LISTENING
TCP	127.0.0.1:4766	127.0.0.1:4767	ESTABLISHED
TCP	127.0.0.1:4767	127.0.0.1:4766	ESTABLISHED
TCP	172.17.68.243:80	172.17.68.248:4759	TIME_WAIT
TCP	172.17.68.243:80	172.17.68.248:4761	ESTABLISHED
TCP	172.17.68.243:80	172.17.68.248:4769	ESTABLISHED
TCP	172.17.68.243:80	172.17.68.248:4770	ESTABLISHED
TCP	172.17.68.243:80	172.17.68.248:4771	ESTABLISHED
TCP	172.17.68.243:80	172.17.68.248:4772	ESTABLISHED

Gambar 6.10 : Perintah netstat –an pada komputer *server* setelah ada koneksi sumber: [*Pengujian*]

D. Kesimpulan

- Aplikasi *Job Placement Center* dapat dijalankan pada jaringan komputer yang menggunakan protokol TCP/IP.
- Koneksi *server* basis data MySQL (*port* 3306) pada komputer *server* dapat dibangun menggunakan aplikasi *Job Placement Center* dari komputer *client*.

6.1.4 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.

6.1.4.1 Kasus Uji Validasi

Untuk mengetahui kesesuaian antara kebutuhan dengan kinerja sistem, pada setiap kebutuhan (*requirement*) dilakukan proses pengujian dengan kasus uji masing-masing.

6.1.4.1.1 Kasus Uji Login

Tabel 6.5a dan tabel 6.5b menunjukkan kasus uji *login*.

Tabel 6.5a : Kasus uji *login* (aliran utama)

Nama kasus uji	Kasus uji <i>login</i> (aliran utama)
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 pada pojok atas kanan (<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* (aliran alternatif 1)

Nama kasus uji	Kasus uji <i>login</i> (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 1 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa pasangan <i>username</i> dan <i>password</i> tidak cocok.
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> = "mufti", <i>password</i> = "mufti"). 3. Menekan tombol login.
Hasil yang diharapkan	Menampilkan pesan bahwa pasangan <i>username</i> dan <i>password</i> tidak cocok.

Sumber : Pengujian

6.1.4.1.2 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 tombol <i>logout</i>.
Hasil yang diharapkan	Pengguna berhasil <i>logout</i> dari sistem.

Sumber : Pengujian

6.1.4.1.3 Kasus Uji Menambah Data Alumni

Tabel 6.7a, tabel 6.7b dan tabel 6.7c menunjukkan kasus uji menambah data alumni.

Tabel 6.7a : Kasus uji menambah data alumni (aliran utama)

Nama kasus uji	Kasus uji menambah data alumni (aliran utama)
Obyek uji	Kebutuhan nomer 3 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah alumni. Data yang harus dicatat dari penambahan alumni adalah No induk, nama, alamat, jenis kelamin, tanggal lahir, tempat lahir, agama, alumnus, tanggal lulus, jurusan, angkatan, ipk, lokasi kerja, telepon, email, foto, username, password.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai admin. 2. Memilih menu input alumni. 3. Memasukkan data-data pada <i>form</i> (nama="coba", username = "coba", password = "coba"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Data alumni berhasil ditambah.

Sumber : Pengujian

Tabel 6.7b : Kasus uji menambah data alumni (aliran alternatif 1)

Nama kasus uji	Kasus uji menambah data alumni (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 3 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nama alumni atau <i>username</i> atau <i>password</i> tidak boleh kosong bila data <i>username</i> atau <i>password</i> kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu input alumni. 3. Memasukkan data-data pada <i>form</i> (nama="coba"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nama alumni atau <i>username</i> atau <i>password</i> tidak boleh kosong.

Sumber : Pengujian

Tabel 6.7c : Kasus uji menambah data alumni (aliran alternatif 2)

Nama kasus uji	Kasus uji menambah data alumni (aliran alternatif 2)
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. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu input alumni. 3. Memasukkan data-data pada <i>form</i> (nama="coba", <i>username</i>="coba" dan <i>password</i> = "coba", dimana telah ada petugas dengan <i>username</i> "coba"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa <i>username</i> tersebut telah ada di dalam sistem.

Sumber : Pengujian

6.1.4.1.4 Kasus Uji Melihat Daftar Alumni

Tabel 6.8 menunjukkan kasus uji melihat daftar alumni.

Tabel 6.8 : Kasus uji melihat daftar alumni (aliran utama)

Nama kasus uji	Kasus uji melihat daftar alumni (aliran utama)
Obyek uji	Kebutuhan nomer 4 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua alumni yang tercatat di dalam sistem. Data yang ditampilkan berupa No induk, nama, tempat lahir, tanggal lahir, agama, jurusan, angkatan, tanggal lulus.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat alumni.
Hasil yang diharapkan	Semua daftar alumni dapat ditampilkan.

Sumber : Pengujian

6.1.4.1.5 Kasus Uji Melihat Daftar Alumni Secara Detail

Tabel 6.9a dan 6.9b menunjukkan kasus uji melihat daftar alumni secara detail.

Tabel 6.9a : Kasus uji melihat daftar alumni secara detail (aliran utama)

Nama kasus uji	Kasus uji melihat daftar alumni secara detail (aliran utama)
Obyek uji	Kebutuhan nomer 5 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar alumni secara detail. Data yang ditampilkan berupa Foto, no induk, nama, tempat tanggal lahir, jenis kelamin, alamat, agama, alumnus, angkatan, jurusan, tanggal lulus, IPK, lokasi kerja, email, telepon.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat alumni. 3. Memilih alumni tertentu yang akan dilihat secara detail 4. Menekan tombol <i>detail</i>.
Hasil yang diharapkan	Data alumni dapat ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.9b : Kasus uji melihat daftar alumni secara detail (aliran alternatif 1)

Nama kasus uji	Kasus uji melihat daftar alumni (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 5 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat alumni. 3. Menekan tombol detail.
Hasil yang diharapkan	Data alumni tidak dapat ditampilkan secara detail.

Sumber : Pengujian

6.1.4.1.6 Kasus Uji Mengubah Data Alumni

Tabel 6.10a dan 6.10b menunjukkan kasus uji mengubah data alumni.

Tabel 6.10a : Kasus uji mengubah data alumni (aliran utama)

Nama kasus uji	Kasus uji mengubah data alumni (aliran utama)
Obyek uji	Kebutuhan nomer 6 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar ubah alumni. Data yang dapat diubah berupa foto, no induk, nama, tempat tanggal lahir, jenis kelamin, alamat, agama, alumnus, angkatan, jurusan, tanggal lulus, lokasi kerja, email, telepon.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat alumni. 3. Memilih alumni tertentu yang akan diubah 4. Menekan tombol ubah. 5. Menekan tombol simpan data
Hasil yang diharapkan	Data alumni dapat diedit dan ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.10b : Kasus uji mengubah data alumni (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah data alumni (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 6 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat alumni. 3. Menekan tombol ubah.
Hasil yang diharapkan	Data alumni tidak dapat diubah dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.7 Kasus Uji Menghapus Data Alumni

Tabel 6.11a dan 6.11b menunjukkan kasus uji menghapus data alumni.

Tabel 6.11a : Kasus uji menghapus data alumni (aliran utama)

Nama kasus uji	Kasus uji menghapus data alumni (aliran utama)
Obyek uji	Kebutuhan nomer 7 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus daftar alumni tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat alumni. 3. Memilih alumni tertentu yang akan diubah 4. Menekan tombol hapus. 5. Anda yakin menghapus tekan tombol 'OK'
Hasil yang diharapkan	Data alumni tertentu dapat dihapus.

Sumber : Pengujian

Tabel 6.11b : Kasus uji menghapus data alumni (aliran alternatif 1)

Nama kasus uji	Kasus uji menghapus data alumni (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 7 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat alumni. 3. Menekan tombol hapus.
Hasil yang diharapkan	Data alumni tidak dapat dihapus dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.8 Kasus Uji Menambah Data Perusahaan

Tabel 6.12a, tabel 6.12b dan tabel 6.12c menunjukkan kasus uji menambah data perusahaan.

Tabel 6.12a : Kasus uji menambah data perusahaan (aliran utama)

Nama kasus uji	Kasus uji menambah data perusahaan (aliran utama)
Obyek uji	Kebutuhan nomer 8 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah perusahaan sebagai <i>client</i> . Data yang harus dicatat dari penambahan perusahaan adalah nama, alamat, email, telepon, foto, username, password.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu input perusahaan. 3. Memasukkan data-data pada <i>form</i> (nama="coba", username = "coba", password = "coba"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Data perusahaan berhasil ditambah.

Sumber : Pengujian

Tabel 6.12b : Kasus uji menambah data alumni (aliran alternatif 1)

Nama kasus uji	Kasus uji menambah data perusahaan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 8 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nama perusahaan atau <i>username</i> atau <i>password</i> tidak boleh kosong bila data <i>username</i> atau <i>password</i> kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu input perusahaan. 3. Memasukkan data-data pada <i>form</i> (nama="coba"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nama perusahaan atau <i>username</i> atau <i>password</i> tidak boleh kosong.

Sumber : Pengujian

Tabel 6.12c : Kasus uji menambah data alumni (aliran alternatif 2)

Nama kasus uji	Kasus uji menambah data perusahaan (aliran alternatif 2)
Obyek uji	Kebutuhan nomer 8 (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. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu input perusahaan. 3. Memasukkan data-data pada <i>form</i> (nama="coba", <i>username</i>="coba" dan <i>password</i> = "coba", dimana telah ada perusahaan dengan <i>username</i> "coba"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa <i>username</i> tersebut telah ada di dalam sistem.

Sumber : Pengujian

6.1.4.1.9 Kasus Uji Melihat Daftar Perusahaan

Tabel 6.13 menunjukkan kasus uji melihat daftar perusahaan.

Tabel 6.13 : Kasus uji melihat daftar perusahaan (aliran utama)

Nama kasus uji	Kasus uji melihat daftar perusahaan (aliran utama)
Obyek uji	Kebutuhan nomer 9 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua perusahaan yang tercatat di dalam sistem. Data yang ditampilkan berupa nama, alamat, telepon, email.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat perusahaan.
Hasil yang diharapkan	Semua daftar perusahaan dapat ditampilkan.

Sumber : Pengujian

6.1.4.1.10 Kasus Uji Melihat Daftar Perusahaan Secara Detail

Tabel 6.14a dan 6.14b menunjukkan kasus uji melihat daftar perusahaan secara detail.

Tabel 6.14a : Kasus uji melihat daftar perusahaan secara detail (aliran utama)

Nama kasus uji	Kasus uji melihat daftar perusahaan secara detail (aliran utama)
Obyek uji	Kebutuhan nomer 10 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar perusahaan secara detail. Data yang ditampilkan berupa nama, alamat, telepon, email, foto.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat perusahaan. 3. Memilih perusahaan tertentu yang akan dilihat 4. Menekan tombol <i>detail</i>.
Hasil yang diharapkan	Data perusahaan dapat ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.14b : Kasus uji melihat daftar perusahaan secara detail (aliran alternatif 1)

Nama kasus uji	Kasus uji melihat daftar perusahaan secara detail (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 10 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat perusahaan. 3. Menekan tombol detail.
Hasil yang diharapkan	Profil perusahaan tidak dapat ditampilkan secara detail.

Sumber : Pengujian

6.1.4.1.11 Kasus Uji Mengubah Profil Perusahaan

Tabel 6.15a dan 6.15b menunjukkan kasus uji mengubah profil perusahaan.

Tabel 6.15a : Kasus uji mengubah profil perusahaan (aliran utama)

Nama kasus uji	Kasus uji mengubah profil perusahaan (aliran utama)
Obyek uji	Kebutuhan nomer 11 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar ubah perusahaan. Data yang dapat diubah berupa nama, alamat, telepon, email, foto.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat perusahaan. 3. Memilih perusahaan tertentu yang akan diubah 4. Menekan tombol ubah. 5. Menekan tombol simpan data
Hasil yang diharapkan	Profil perusahaan dapat diedit dan ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.15b : Kasus uji mengubah profil perusahaan (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah profil perusahaan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 11 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat perusahaan. 3. Menekan tombol ubah.
Hasil yang diharapkan	Profil perusahaan tidak dapat diubah dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.12 Kasus Uji Menghapus Profil Perusahaan

Tabel 6.16a dan 6.16b menunjukkan kasus uji menghapus profil perusahaan.

Tabel 6.16a : Kasus uji menghapus profil perusahaan (aliran utama)

Nama kasus uji	Kasus uji menghapus profil perusahaan (aliran utama)
Obyek uji	Kebutuhan nomer12 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus daftar perusahaan tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat perusahaan. 3. Memilih perusahaan tertentu yang akan dihapus. 4. Menekan tombol hapus. 5. Anda yakin menghapus tekan tombol 'OK'.
Hasil yang diharapkan	Profil perusahaan tertentu dapat dihapus.

Sumber : Pengujian

Tabel 6.16b : Kasus uji menghapus profil perusahaan (aliran alternatif 1)

Nama kasus uji	Kasus uji menghapus profil perusahaan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 12 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat perusahaan. 3. Menekan tombol hapus.
Hasil yang diharapkan	Profil perusahaan tidak dapat dihapus dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.13 Kasus Uji Menambah Data Berita

Tabel 6.17 menunjukkan kasus uji menambah data berita.

Tabel 6.17 : Kasus uji menambah data berita (aliran utama)

Nama kasus uji	Kasus uji menambah data berita (aliran utama)
Obyek uji	Kebutuhan nomer 13 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah berita. Data yang harus dicatat dari penambahan berita adalah judul, isi berita, foto.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu input berita. 3. Menekan tombol simpan data.
Hasil yang diharapkan	Data berita berhasil ditambah.

Sumber : Pengujian

6.1.4.1.14 Kasus Uji Melihat Daftar Berita

Tabel 6.18 menunjukkan kasus uji melihat daftar berita.

Tabel 6.18 : Kasus uji melihat daftar berita (aliran utama)

Nama kasus uji	Kasus uji melihat daftar berita (aliran utama)
Obyek uji	Kebutuhan nomer 14 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua berita yang tercatat di dalam sistem. Data yang ditampilkan berupa judul dan waktu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat berita.
Hasil yang diharapkan	Semua daftar berita dapat ditampilkan.

Sumber : Pengujian

6.1.4.1.15 Kasus Uji Melihat Daftar Berita Secara Detail

Tabel 6.19a dan 6.19b menunjukkan kasus uji melihat daftar berita secara detail.

Tabel 6.19a : Kasus uji melihat daftar berita secara detail (aliran utama)

Nama kasus uji	Kasus uji melihat daftar berita secara detail (aliran utama)
Obyek uji	Kebutuhan nomer 15 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar berita secara detail. Data yang ditampilkan berupa gambar, waktu dan isi berita tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat berita. 3. Memilih berita tertentu yang akan dilihat 4. Menekan tombol <i>detail</i>.
Hasil yang diharapkan	Data berita tertentu dapat ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.19b : Kasus uji melihat daftar berita secara detail (aliran alternatif 1)

Nama kasus uji	Kasus uji melihat daftar berita secara detail (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 15 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat berita. 3. Menekan tombol detail.
Hasil yang diharapkan	Data berita tidak dapat ditampilkan secara detail.

Sumber : Pengujian

6.1.4.1.16 Kasus Uji Mengubah Data Berita

Tabel 6.20a dan 6.20b menunjukkan kasus uji mengubah data berita.

Tabel 6.20a : Kasus uji mengubah data berita (aliran utama)

Nama kasus uji	Kasus uji mengubah data alumni (aliran utama)
Obyek uji	Kebutuhan nomer 16 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar ubah berita. Data yang dapat diubah berupa judul, isi berita dan foto.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat berita. 3. Memilih berita tertentu yang akan diubah 4. Menekan tombol ubah. 5. Menekan tombol simpan berita
Hasil yang diharapkan	Data berita dapat diubah dan ditampilkan.

Sumber : Pengujian

Tabel 6.20b : Kasus uji mengubah data berita(aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah data berita (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 16 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat berita. 3. Menekan tombol ubah.
Hasil yang diharapkan	Data berita tidak dapat diubah dan menampilkan pesan anda harus memilih data tertentu terlebih dahulu

Sumber : Pengujian

6.1.4.1.17 Kasus Uji Menghapus Berita

Tabel 6.21a dan 6.21b menunjukkan kasus uji menghapus berita.

Tabel 6.21a : Kasus uji menghapus berita (aliran utama)

Nama kasus uji	Kasus uji menghapus berita (aliran utama)
Obyek uji	Kebutuhan nomer 17 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus daftar berita tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat berita. 3. Memilih judul berita tertentu yang akan diubah 4. Menekan tombol hapus. 5. Anda yakin akan menghapus tekan tombol 'OK'
Hasil yang diharapkan	Daftar berita tertentu dapat dihapus.

Sumber : Pengujian

Tabel 6.21b : Kasus uji menghapus data berita (aliran alternatif 1)

Nama kasus uji	Kasus uji menghapus berita (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 17 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu lihat berita. 3. Menekan tombol hapus.
Hasil yang diharapkan	Daftar berita tidak dapat dihapus dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.18 Kasus Uji Mereset Password

Tabel 6.22 menunjukkan kasus uji menambah data berita.

Tabel 6.22 : Kasus uji mereset password (aliran utama)

Nama kasus uji	Kasus uji mereset <i>password</i> (aliran utama)
Obyek uji	Kebutuhan nomer 18 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk <i>reset password</i> . Data yang harus dicatat dari reset password adalah nama login dan <i>password</i> yang baru.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai administrator. 2. Memilih menu <i>password reset</i>. 3. Menekan tombol <i>replace</i>.
Hasil yang diharapkan	<i>Password</i> lama akan diganti dengan <i>password</i> yang baru.

Sumber : Pengujian

6.1.4.1.19 Kasus Uji Mengubah Profil Perusahaan

Tabel 6.23a dan tabel 6.23b menunjukkan kasus uji mengedit profil perusahaan.

Tabel 6.23a: Kasus uji mengubah profil perusahaan (aliran utama)

Nama kasus uji	Kasus uji mengubah data perusahaan (aliran utama)
Obyek uji	Kebutuhan nomer 19 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengubah profil perusahaan yang telah dimasukkan oleh administrator. profil yang dapat diedit adalah nama, alamat, email, telepon, foto, <i>password</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu ubah data diri. 3. Memasukkan data-data pada <i>form</i> yang akan diubah 4. Menekan tombol simpan data.
Hasil yang diharapkan	Profil perusahaan dapat diubah dan disimpan.

Sumber : Pengujian

Tabel 6.23b: Kasus uji mengubah profil perusahaan (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah profil perusahaan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 19 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu ubah data diri. 3. Menekan tombol ubah.
Hasil yang diharapkan	Profil perusahaan tidak dapat diubah dan menampilkan pesan anda harus memilih data tertentu terlebih dahulu

Sumber : Pengujian

6.1.4.1.20 Kasus Uji Menambah Data Lowongan

Tabel 6.24a, 6.24b dan 6.24c menunjukkan kasus uji menambah data lowongan.

Tabel 6.24a : Kasus uji menambah data lowongan (aliran utama)

Nama kasus uji	Kasus uji menambah data lowongan (aliran utama)
Obyek uji	Kebutuhan nomer 20 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah lowongan pekerjaan. Data yang harus dicatat dari penambahan lowongan pekerjaan adalah nama, deskripsi, posisi, batas umur, <i>ipk_min</i> , jurusan, batas waktu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input lowongan. 3. Mengisi data-data pada <i>form</i> (nama = "nama perusahaan"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Data lowongan perusahaan berhasil ditambah.

Sumber : Pengujian

Tabel 6.24b : Kasus uji menambah data lowongan (aliran alternatif 1)

Nama kasus uji	Kasus uji menambah data lowongan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 20 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nama perusahaan, deskripsi, posisi, batas umur, ipk_min, jurusan, batas waktu. tidak boleh kosong .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input lowongan. 3. Memasukkan data-data pada <i>form</i> (nama="nama perusahaan"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nama perusahaan, deskripsi, posisi, batas umur, ipk_min, jurusan, batas waktu tidak boleh kosong.

Sumber : Pengujian

Tabel 6.24c : Kasus uji menambah data lowongan (aliran alternatif 2)

Nama kasus uji	Kasus uji menambah data lowongan (aliran alternatif 2)
Obyek uji	Kebutuhan nomer 20 (aliran alternatif 2).
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan kesalahan bahwa data tersebut telah ada di dalam sistem bila data tersebut memang telah ada di dalam sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input lowongan. 3. Memasukkan data-data pada <i>form</i> (nama="nama perusahaan"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa data tersebut telah ada di dalam sistem.

Sumber : Pengujian

6.1.4.1.21 Kasus Uji Melihat Daftar Lowongan

Tabel 6.25 menunjukkan kasus uji melihat daftar lowongan.

Tabel 6.25 : Kasus uji melihat daftar lowongan (aliran utama)

Nama kasus uji	Kasus uji melihat daftar lowongan (aliran utama)
Obyek uji	Kebutuhan nomer 21 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan lowongan yang tercatat di dalam sistem. Data yang ditampilkan berupa nama lowongan, deskripsi, posisi, batas umur, ipk_min.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan.
Hasil yang diharapkan	Daftar lowongan dari perusahaan tersebut dapat ditampilkan.

Sumber : Pengujian

6.1.4.1.22 Kasus Uji Melihat Daftar Lowongan Secara Detail

Tabel 6.26a dan 6.26b menunjukkan kasus uji melihat daftar lowongan secara detail.

Tabel 6.26a : Kasus uji melihat daftar lowongan secara detail (aliran utama)

Nama kasus uji	Kasus uji melihat daftar lowongan secara detail (aliran utama)
Obyek uji	Kebutuhan nomer 22 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar lowongan secara detail. Data yang ditampilkan berupa foto, nama, alamat, email, telepon.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Memilih lowongan tertentu yang akan dilihat 4. Menekan tombol <i>detail</i>.
Hasil yang diharapkan	Data perusahaan dapat ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.26b : Kasus uji melihat daftar perusahaan secara detail (aliran alternatif 1)

Nama kasus uji	Kasus uji melihat daftar lowongan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 22 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Menekan tombol detail.
Hasil yang diharapkan	Profil perusahaan tidak dapat ditampilkan secara detail.

Sumber : Pengujian

6.1.4.1.23 Kasus Uji Mengubah Data Lowongan

Tabel 6.27a dan 6.27b menunjukkan kasus uji mengubah data lowongan.

Tabel 6.27a : Kasus uji mengubah data lowongan (aliran utama)

Nama kasus uji	Kasus uji mengubah data lowongan (aliran utama)
Obyek uji	Kebutuhan nomer 23 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar ubah lowongan. Data yang dapat diubah berupa nama perusahaan, deskripsi, posisi, batas umur, ipk_min, jurusan, batas waktu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Memilih lowongan tertentu yang akan diubah 4. Menekan tombol ubah. 5. Menekan tombol simpan data
Hasil yang diharapkan	lowongan perusahaan dapat diubah dan ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.27b : Kasus uji mengubah data lowongan (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah data lowongan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 23 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat perusahaan. 3. Menekan tombol ubah.
Hasil yang diharapkan	Lowongan perusahaan tidak dapat diubah dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.24 Kasus Uji Menghapus Data Lowongan

Tabel 6.28a dan 6.28b menunjukkan kasus uji menghapus data lowongan.

Tabel 6.28a : Kasus uji menghapus data lowongan (aliran utama)

Nama kasus uji	Kasus uji menghapus data lowongan (aliran utama)
Obyek uji	Kebutuhan nomer24 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus daftar lowongan tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Memilih lowongan tertentu yang akan dihapus. 4. Menekan tombol hapus. 5. Anda yakin menghapus tekan tombol 'OK'.
Hasil yang diharapkan	Data lowongan perusahaan tertentu dapat dihapus.

Sumber : Pengujian

Tabel 6.28b : Kasus uji menghapus data lowongan (aliran alternatif 1)

Nama kasus uji	Kasus uji menghapus data lowongan (aliran alternatif 1).
Obyek uji	Kebutuhan nomer 24 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Menekan tombol hapus.
Hasil yang diharapkan	Data lowongan perusahaan tidak dapat dihapus dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.25 Kasus Uji Melihat Daftar Alumni Yang Mengirim Lamaran

Tabel 6.29 menunjukkan kasus uji lihat daftar pelamar.

Tabel 6.29 : Kasus uji melihat pelamar (aliran utama)

Nama kasus uji	Kasus uji melihat daftar pelamar (aliran utama)
Obyek uji	Kebutuhan nomer 25 (aliran utama)

Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar pelamar . Data yang dapat ditampilkan berupa no, nama, jurusan, angkatan, ipk, tanggal kelulusan, download CV.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Menekan tombol lihat pelamar. 4. Menekan tombol download.
Hasil yang diharapkan	Perusahaan dapat melihat daftar alumni yang mengirim lamaran dan ditampilkan secara detail.

Sumber : Pengujian

6.1.4.1.26 Kasus Uji Men-Download Berkas Lamaran

Tabel 6.30a dan 6.30b menunjukkan kasus uji men-download berkas lamaran.

Tabel 6.30a : Kasus uji men-download berkas lamaran (aliran utama)

Nama kasus uji	Kasus uji men-download berkas lamaran (aliran utama)
Obyek uji	Kebutuhan nomer 26 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat men-download berkas lamaran. Data yang dapat didownload berupa file word.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Menekan tombol lihat pelamar. 4. Menekan tombol download.
Hasil yang diharapkan	Perusahaan dapat men-download berkas lamaran dan ditampilkan secara detail.

Tabel 6.30b : Kasus uji men-download berkas lamaran (aliran alternatif 1)

Nama kasus uji	Kasus uji men-download berkas lamaran (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 26 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem tidak dapat men-download berkas lamaran selain file word.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat lowongan. 3. Menekan tombol lihat lamaran 4. Menekan tombol download.
Hasil yang diharapkan	Berkas lamaran tidak dapat didownload

Sumber : Pengujian

6.1.4.1.27 Kasus Uji Menambah Data Kategori Soal

Tabel 6.31a dan 6.31b menunjukkan kasus uji menambah data kategori soal.

Tabel 6.31a : Kasus uji menambah data kategori soal (aliran utama)

Nama kasus uji	Kasus uji menambah data kategori soal (aliran utama)
Obyek uji	Kebutuhan nomer 27 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah kategori soal. Data yang harus dicatat dari penambahan kategori soal adalah nama, tanggal, jam dan durasi.

Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input kategori soal. 3. Mengisi data-data pada <i>form</i> (nama = “psikotest”). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Data kategori soal berhasil ditambah.

Sumber : Pengujian

Tabel 6.31b : Kasus uji menambah data kategori soal (aliran alternatif 1)

Nama kasus uji	Kasus uji menambah data kategori soal (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 27 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nama, tanggal, jam dan durasi. tidak boleh kosong.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input kategori soal. 3. Memasukkan data-data pada <i>form</i> (nama=”jenis soal”). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nama, tanggal, jam dan durasi tidak boleh kosong.

Sumber : Pengujian

6.1.4.1.28 Kasus Uji Melihat Daftar Kategori Soal

Tabel 6.32 menunjukkan kasus uji melihat daftar kategori soal.

Tabel 6.32 : Kasus uji melihat daftar kategori soal (aliran utama)

Nama kasus uji	Kasus uji melihat daftar kategori soal (aliran utama)
Obyek uji	Kebutuhan nomer 28 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan lowongan yang tercatat di dalam sistem. Data yang ditampilkan berupa nama, tanggal, waktu dan durasi.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat kategori soal.
Hasil yang diharapkan	Daftar kategori soal dari perusahaan tersebut dapat ditampilkan.

Sumber : Pengujian

6.1.4.1.29 Kasus Uji Mengubah Kategori Soal

Tabel 6.33a dan 6.33b menunjukkan kasus uji mengubah kategori soal.

Tabel 6.33a : Kasus uji mengubah kategori soal (aliran utama)

Nama kasus uji	Kasus uji mengubah kategori soal (aliran utama)
Obyek uji	Kebutuhan nomer 29 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar ubah lowongan. Data yang dapat diubah berupa nama, tanggal, jam, durasi.

Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat kategori soal. 3. Memilih kategori soal tertentu yang akan diubah 4. Menekan tombol ubah. 5. Menekan tombol simpan data
Hasil yang diharapkan	Kategori soal perusahaan dapat diubah dan ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.33b : Kasus uji mengubah kategori soal (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah kategori soal (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 29 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat kategori soal. 3. Menekan tombol ubah.
Hasil yang diharapkan	Kategori soal perusahaan tidak dapat diubah dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.30 Kasus Uji Menghapus Kategori Soal

Tabel 6.34a dan 6.34b menunjukkan kasus uji kategori soal.

Tabel 6.34a : Kasus uji menghapus kategori soal (aliran utama)

Nama kasus uji	Kasus uji menghapus kategori soal (aliran utama)
Obyek uji	Kebutuhan nomer 30 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus daftar kategori soal tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat kategori soal. 3. Memilih kategori soal tertentu yang akan dihapus. 4. Menekan tombol hapus. 5. Anda yakin menghapus tekan tombol 'OK'.
Hasil yang diharapkan	Data kategori soal perusahaan tertentu dapat dihapus.

Sumber : Pengujian

Tabel 6.34b : Kasus uji menghapus kategori soal (aliran alternatif 1)

Nama kasus uji	Kasus uji menghapus kategori soal (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 30 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat kategori soal. 3. Menekan tombol hapus.
Hasil yang diharapkan	Data kategori soal perusahaan tidak dapat dihapus dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.31 Kasus Uji Menambah Data Soal

Tabel 6.35a dan 6.35b menunjukkan kasus uji menambah data soal.

Tabel 6.35a : Kasus uji menambah data soal (aliran utama)

Nama kasus uji	Kasus uji menambah data soal (aliran utama)
Obyek uji	Kebutuhan nomer 31 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menginputkan soal perusahaan. Data yang harus dicatat dari penambahan soal adalah kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input soal. 3. Mengisi data-data pada <i>form</i> (kategori soal = "psikotest"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Data soal perusahaan berhasil ditambah.

Sumber : Pengujian

Tabel 6.35b : Kasus uji menambah data soal (aliran alternatif 1)

Nama kasus uji	Kasus uji menambah data soal (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 31 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban, tidak boleh kosong .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input soal. 3. Memasukkan data-data pada <i>form</i> (kategori soal="kategori soal"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D dan jawaban tidak boleh kosong.

Sumber : Pengujian

6.1.4.1.32 Kasus Uji Melihat Daftar Soal

Tabel 6.36a dan 6.36b menunjukkan kasus uji melihat daftar soal.

Tabel 6.36a : Kasus uji melihat daftar soal (aliran utama)

Nama kasus uji	Kasus uji melihat daftar soal (aliran utama)
Obyek uji	Kebutuhan nomer 32 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan data soal yang tercatat di dalam sistem. Data yang ditampilkan berupa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, dan jawaban.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat soal.
Hasil yang diharapkan	Daftar soal dari perusahaan tersebut dapat ditampilkan.

Sumber : Pengujian

Tabel 6.36b : Kasus uji melihat daftar soal (aliran alternatif 1)

Nama kasus uji	Kasus uji melihat daftar soal (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 32 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem menampilkan daftar soal milik perusahaan itu sendiri dan tidak bisa menampilkan soal dari perusahaan yang lain.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat soal. 3. Soal yang ditampilkan milik perusahaan itu sendiri
Hasil yang diharapkan	Sistem menampilkan daftar soal milik perusahaan masing-masing dan tidak bisa melihat daftar soal perusahaan yang lain.

Sumber : Pengujian

6.1.4.1.33 Kasus Uji Mengubah Daftar Soal

Tabel 6.37a dan 6.37b menunjukkan kasus uji mengubah daftar soal.

Tabel 6.37a : Kasus uji mengubah daftar soal (aliran utama)

Nama kasus uji	Kasus uji mengubah daftar soal (aliran utama)
Obyek uji	Kebutuhan nomer 33 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan daftar ubah soal. Data yang dapat diubah berupa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, dan jawaban.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat soal. 3. Memilih kategori soal tertentu yang akan diubah 4. Menekan tombol ubah. 5. Menekan tombol simpan data
Hasil yang diharapkan	Daftar soal perusahaan dapat diubah dan ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.37b : Kasus uji mengubah daftar soal (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah daftar soal (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 33 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat soal. 3. Menekan tombol ubah.
Hasil yang diharapkan	Daftar soal perusahaan tidak dapat diubah dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.34 Kasus Uji Menghapus Daftar Soal

Tabel 6.38a dan 6.38b menunjukkan kasus uji daftar soal.

Tabel 6.38a : Kasus uji menghapus daftar soal (aliran utama)

Nama kasus uji	Kasus uji menghapus daftar soal (aliran utama)
Obyek uji	Kebutuhan nomer 34 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus daftar soal tertentu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat soal. 3. Memilih kategori soal tertentu yang akan dihapus. 4. Menekan tombol hapus. 5. Anda yakin menghapus tekan tombol 'OK'.
Hasil yang diharapkan	Data soal perusahaan tertentu dapat dihapus.

Sumber : Pengujian

Tabel 6.38b : Kasus uji menghapus daftar soal (aliran alternatif 1)

Nama kasus uji	Kasus uji menghapus daftar soal (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 34 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat soal. 3. Menekan tombol hapus.
Hasil yang diharapkan	Data soal perusahaan tidak dapat dihapus dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.35 Kasus Uji Menambah Data Peserta Test

Tabel 6.39a, tabel 6.39b dan tabel 6.39c menunjukkan kasus uji menambah data peserta test.

Tabel 6.39a : Kasus uji menambah data peserta test (aliran utama)

Nama kasus uji	Kasus uji menambah data peserta test (aliran utama)
Obyek uji	Kebutuhan nomer 35 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menyediakan fasilitas untuk menambah peserta test. Data yang harus dicatat dari penambahan peserta test adalah kategori soal dan nama lowongan.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input peserta test. 3. Mengisi data-data pada <i>form</i> (kategori soal = "psikotest"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Data peserta test lowongan pekerjaan berhasil ditambah.

Sumber : Pengujian

Tabel 6.39b : Kasus uji menambah data peserta test (aliran alternatif 1)

Nama kasus uji	Kasus uji menambah data peserta test (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 35 (aliran alternatif 1).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa kategori soal dan nama lowongan. tidak boleh kosong .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu input peserta test. 3. Memasukkan data-data pada <i>form</i> (kategori soal="kategori soal"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa kategori soal dan nama soal tidak boleh kosong.

Sumber : Pengujian

Tabel 6.39c : Kasus uji menambah data peserta test (aliran alternatif 2)

Nama kasus uji	Kasus uji menambah data peserta test (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 35 (aliran alternatif 2).
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan tombol ikuti test pada user alumni.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Alumni telah mengirim berkas lamaran. 3. Memilih menu input peserta test. 4. Mengaktifkan peserta yang dapat ikut test 5. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem dapat menampilkan tombol ikuti test pada user alumni.

Sumber : Pengujian

6.1.4.1.36 Kasus Uji Melihat Daftar Peserta *Test Online*

Tabel 6.40 menunjukkan kasus uji melihat daftar peserta *test online*.

Tabel 6.40 : Kasus uji melihat daftar peserta test *online* (aliran utama)

Nama kasus uji	Kasus uji melihat daftar peserta test <i>online</i> (aliran utama)
Obyek uji	Kebutuhan nomer 36 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan data soal yang tercatat di dalam sistem. Data yang ditampilkan berupa kategori soal, nim, nama, tanggal, jam, benar, salah.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat peserta test.
Hasil yang diharapkan	Daftar peserta test dari lowongan perusahaan tersebut dapat ditampilkan.

Sumber : Pengujian

6.1.4.1.37 Kasus Uji Menghapus Daftar Peserta *Test Online*

Tabel 6.41a dan 6.41b menunjukkan kasus uji menghapus daftar peserta *test*.

Tabel 6.41a : Kasus uji menghapus daftar peserta test (aliran utama)

Nama kasus uji	Kasus uji menghapus daftar peserta test (aliran utama)
Obyek uji	Kebutuhan nomer 37 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menghapus daftar <i>peserta test online</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat peserta test. 3. Memilih peserta test yang akan dihapus 4. Menekan tombol hapus.
Hasil yang diharapkan	Data peserta test perusahaan dapat dihapus dan ditampilkan secara detail.

Sumber : Pengujian

Tabel 6.41b : Kasus uji menghapus daftar peserta test (aliran alternatif 1)

Nama kasus uji	Kasus uji menghapus daftar peserta test (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 37 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan pesan bahwa anda harus memilih data terlebih dahulu.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai perusahaan. 2. Memilih menu lihat peserta test. 3. Menekan tombol hapus.
Hasil yang diharapkan	Data peserta test tidak dapat dihapus dan menampilkan pesan anda harus memilih data terlebih dahulu

Sumber : Pengujian

6.1.4.1.38 Kasus Uji Mengubah Data Diri Alumni

Tabel 6.42a, 6.42b, 6.42c menunjukkan kasus uji mengubah data diri alumni.

Tabel 6.42a : Kasus uji mengubah data diri alumni (aliran utama)

Nama kasus uji	Kasus uji mengubah data diri alumni (aliran utama)
Obyek uji	Kebutuhan nomer 38 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mengubah data diri alumni yang telah dimasukkan oleh administrator. Data yang dapat diubah adalah nama, jenis kelamin, tempat tanggal lahir, alamat, agama, tanggal lulus, jurusan, angkatan, lokasi kerja, email, telepon, foto, <i>password</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai alumni. 2. Memilih menu ubah data diri. 3. Memasukkan data-data pada <i>form</i> yang akan diubah 4. Menekan tombol simpan data.
Hasil yang diharapkan	Data diri alumni dapat diubah dan ditampilkan.

Sumber : Pengujian

Tabel 6.42b : Kasus uji mengubah data diri alumni (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah data diri alumni (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 38 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan bahwa data nim, nama atau <i>username</i> dan ipk tidak dapat diubah.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai alumni. 2. Memilih menu ubah data diri. 3. Memasukkan data-data pada <i>form</i> kecuali nim dan <i>username</i> tidak dapat diubah. 4. Menekan tombol <i>simpan data</i>.
Hasil yang diharapkan	Sistem menampilkan nim dan nama user tidak aktif (tidak dapat diubah) <i>password</i> dikosongkan apabila tidak diubah.

Sumber : Pengujian

Tabel 6.42c : Kasus uji mengubah data diri alumni (aliran alternatif 1)

Nama kasus uji	Kasus uji mengubah data diri alumni (aliran alternatif 2)
Obyek uji	Kebutuhan nomer 38 (aliran alternatif 2).
Tujuan pengujian	Menguji bahwa sistem menampilkan pesan kesalahan bahwa nama, jenis kelamin, tanggal lahir. tidak boleh kosong .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai alumni. 2. Memilih menu ubah data diri. 3. Memasukkan data-data pada <i>form</i> (nama="achmad mufti"). 4. Menekan tombol simpan data.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa nama tidak boleh kosong.

Sumber : Pengujian

6.1.4.1.39 Kasus Uji Ikuti *Test Online*

Tabel 6.43a, 6.43b, 6.43c, dan 6.43d menunjukkan kasus uji ikuti *test online*.

Tabel 6.43a : Kasus uji ikuti *test online* (aliran utama)

Nama kasus uji	Kasus uji ikuti <i>test online</i> (aliran utama)
Obyek uji	Kebutuhan nomer 39 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan ikuti <i>test online</i> yang tercatat di dalam sistem. Data yang ditampilkan berupa kategori soal, waktu ujian, jam ujian, lama ujian.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai alumni. 2. Alumni harus mengirim berkas lamaran. 3. Perusahaan mengaktifkan alumni yang dapat mengikuti test. 4. Memilih menu ikuti <i>test online</i>. 5. Menekan tombol ikuti test.
Hasil yang diharapkan	Daftar soal test dari lowongan perusahaan tersebut dapat ditampilkan dan alumni dapat mengerjakan soal sesuai dengan durasi waktu.

Sumber : Pengujian

Tabel 6.43b : Kasus uji ikuti test online (aliran alternatif 1)

Nama kasus uji	Kasus uji ikuti <i>test online</i> (aliran alternatif 1).
Obyek uji	Kebutuhan nomer 39 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan tombol ikuti <i>test online</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai alumni. 2. Alumni harus mengirim berkas lamaran. 3. Perusahaan mengaktifkan alumni yang dapat mengikuti test. 4. Memilih menu ikuti <i>test online</i>.
Hasil yang diharapkan	Tombol ikuti test <i>online</i> dapat ditampilkan.

Sumber : Pengujian

Tabel 6.43c : Kasus uji ikuti test online (aliran alternatif 2)

Nama kasus uji	Kasus uji ikuti test <i>online</i> (aliran alternatif 1).
Obyek uji	Kebutuhan nomer 39 (aliran alternatif 2)
Tujuan pengujian	Menguji bahwa sistem tidak dapat menampilkan tombol ikuti <i>test online</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai alumni. 2. Alumni belum mengirim berkas lamaran. 3. Tombol ikuti test tidak tampil.
Hasil yang diharapkan	Tombol ikuti test <i>online</i> tidak dapat ditampilkan.

Sumber : Pengujian

Tabel 6.43d : Kasus uji ikuti test online (aliran alternatif 3)

Nama kasus uji	Kasus uji ikuti test <i>online</i> (aliran alternatif 1).
Obyek uji	Kebutuhan nomer 39 (aliran alternatif 3)
Tujuan pengujian	Menguji bahwa sistem tidak dapat menampilkan tombol ikuti test.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan pengguna telah login sebagai alumni. 2. Alumni telah mengirim berkas lamaran 3. Perusahaan belum mengaktifkan alumni yang dapat mengikuti test. 4. Tombol ikuti test tidak tampil.
Hasil yang diharapkan	Tombol ikuti test <i>online</i> tidak dapat ditampilkan.

Sumber : Pengujian

6.1.4.1.40 Kasus Uji Mencari Data Alumni

Tabel 6.44a, dan tabel 6.44b menunjukkan kasus uji mencari data alumni.

Tabel 6.44a : Kasus uji mencari data alumni (aliran utama)

Nama kasus uji	Kasus uji mencari data alumni (aliran utama)
Obyek uji	Kebutuhan nomer 40 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari suatu data alumni yang tercatat di dalam sistem. Pencarian dapat berdasarkan pada no induk, nama, alamat, jurusan, angkatan dan email. Metode pencarian dapat berupa seperti atau persis.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan semua user dapat menggunakan form pencarian data alumni. 2. Memilih menu alumni.

	<ol style="list-style-type: none"> 3. Memasukkan <i>keyword</i> pencarian (<i>keyword</i> = “mufti”). 4. Menekan tombol cari data
Hasil yang diharapkan	Data alumni tertentu dapat ditampilkan berdasarkan <i>keyword</i> .

Sumber : Pengujian

Tabel 6.44b : Kasus uji mencari data alumni (aliran alternatif 1)

Nama kasus uji	Kasus uji mencari data alumni (aliran alternatif 1).
Obyek uji	Kebutuhan nomer 40 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua daftar data alumni apabila <i>keyword</i> dikosongkan.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan semua user dapat menggunakan form pencarian data alumni. 2. Memilih menu alumni. 3. <i>Keyword</i> pencarian kosong. 4. Menekan tombol cari data.
Hasil yang diharapkan	Sistem menampilkan semua data alumni yang tersimpan apabila <i>keyword</i> kosong.

Sumber : Pengujian

6.1.4.1.41 Kasus Uji Mencari Data Perusahaan

Tabel 6.45a dan tabel 6.45b menunjukkan kasus uji mencari data perusahaan.

Tabel 6.45a : Kasus uji mencari data perusahaan (aliran utama)

Nama kasus uji	Kasus uji mencari data perusahaan (aliran utama)
Obyek uji	Kebutuhan nomer 41 (aliran utama)
Tujuan pengujian	Menguji bahwa sistem dapat mencari suatu data perusahaan yang tercatat di dalam sistem. Pencarian dapat berdasarkan pada nama, alamat dan email. Metode pencarian dapat berupa seperti atau persis.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan semua user dapat menggunakan form pencarian data perusahaan. 2. Memilih menu perusahaan. 3. Memasukkan <i>keyword</i> pencarian (<i>keyword</i> = “pertamina”). 4. Menekan tombol cari data
Hasil yang diharapkan	Data perusahaan tertentu dapat ditampilkan berdasarkan <i>keyword</i> .

Sumber : Pengujian

Tabel 6.45b : Kasus uji mencari data perusahaan(aliran alternatif 1)

Nama kasus uji	Kasus uji mencari data perusahaan (aliran alternatif 1)
Obyek uji	Kebutuhan nomer 41 (aliran alternatif 1)
Tujuan pengujian	Menguji bahwa sistem dapat menampilkan semua daftar data perusahaan apabila <i>keyword</i> dikosongkan.
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem telah berjalan dan semua user dapat menggunakan form pencarian data perusahaan. 2. Memilih menu perusahaan. 3. <i>Keyword</i> pencarian kosong. 4. Menekan tombol cari data.
Hasil yang diharapkan	Sistem menampilkan semua data perusahaan yang tersimpan apabila <i>keyword</i> kosong.

Sumber : Pengujian

6.1.4.2 Hasil Pengujian Validasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian pada sub pokok bahasan 6.1.4.1, didapatkan hasil seperti ditunjukkan pada Tabel 6.46 berikut:

Tabel 6.46 : Hasil Pengujian Validasi

No	Nama kasus uji	Hasil yang didapat	Status validitas
1	Kasus uji login	Pengguna berhasil login pada sistem.	Valid
2	Kasus uji <i>login</i> dimana pasangan username dan password tidak ada dalam sistem	Menampilkan pesan bahwa “Login gagal, silahkan mengulangi kembali”.	Valid
3	Kasus uji <i>logout</i>	Pengguna berhasil <i>logout</i> dari sistem.	Valid
4	Kasus uji menambah data alumni	Data alumni berhasil ditambah.	Valid
5	Kasus uji menambah data alumni dimana masukan nama alumni, username atau password kosong.	Sistem menampilkan pesan kesalahan bahwa nama alumni atau <i>username</i> atau <i>password</i> tidak boleh kosong.	Valid
6	Kasus uji menambah data alumni dimana masukan username telah ada dalam sistem	Sistem menampilkan pesan kesalahan bahwa <i>username</i> telah ada, silahkan menggunakan <i>username</i> yang lain.	Valid
7	Kasus uji melihat daftar alumni	Semua daftar alumni dapat ditampilkan	Valid
8	Kasus uji melihat daftar alumni secara detail	Data alumni dapat ditampilkan secara detail.	Valid
9	Kasus uji mengubah daftar alumni	Data alumni dapat diubah dan disimpan.	Valid
10	Kasus uji menghapus daftar alumni	Data alumni tertentu dapat dihapus.	Valid
11	Kasus uji menambah data perusahaan	Data perusahaan berhasil ditambah.	Valid
12	Kasus uji menambah data perusahaan dimana masukan nama perusahaan, username atau password kosong.	Sistem menampilkan pesan kesalahan bahwa nama perusahaan, <i>username</i> atau <i>password</i> tidak boleh kosong.	Valid
13	Kasus uji menambah data perusahaan dimana masukan username telah ada dalam sistem	Sistem menampilkan pesan kesalahan bahwa <i>username</i> telah ada, silahkan menggunakan <i>username</i> yang lain.	Valid
14	Kasus uji melihat daftar perusahaan	Semua daftar perusahaan yang menjadi <i>client</i> dapat ditampilkan.	Valid
15	Kasus uji melihat daftar perusahaan secara detail	Salah satu daftar perusahaan dapat ditampilkan secara detail.	Valid
16	Kasus uji mengubah profil perusahaan	Profil perusahaan dapat diubah dan disimpan.	Valid
17	Kasus uji menghapus profil perusahaan	Profil perusahaan dapat dihapus.	Valid
18	Kasus uji menambah daftar berita	Daftar berita berhasil ditambah.	Valid

19	Kasus uji melihat daftar berita	Semua daftar berita dapat ditampilkan.	Valid
20	Kasus uji melihat daftar berita secara detail	Salah satu daftar berita dapat ditampilkan secara detail.	Valid
21	Kasus uji mengubah daftar berita	Daftar berita dapat diubah dan disimpan.	Valid
22	Kasus uji menghapus berita	Daftar berita dapat dihapus.	Valid
23	Kasus uji reset password	Password lama akan diganti dengan password yang baru	Valid
24	Kasus uji mengubah profil perusahaan	Profil perusahaan dapat diubah dan disimpan.	Valid
25	Kasus uji menambah data lowongan	Data lowongan perusahaan berhasil ditambah	Valid
26	Kasus uji menambah data lowongan dimana masukan nama perusahaan, deskripsi, posisi, batas umur, ipk_min, jurusan, batas waktu kosong.	Sistem menampilkan pesan kesalahan bahwa nama perusahaan, deskripsi, posisi, batas umur, ipk_min, jurusan, batas waktu tidak boleh kosong.	Valid
27	Kasus uji menambah data lowongan perusahaan dimana masukan telah ada dalam sistem	Sistem menampilkan pesan kesalahan bahwa data tersebut telah ada dalam sistem.	Valid
28	Kasus uji melihat daftar lowongan	Semua daftar berita dapat ditampilkan.	Valid
29	Kasus uji melihat daftar lowongan secara detail	Data perusahaan dapat ditampilkan secara detail.	Valid
30	Kasus uji mengubah data lowongan	Daftar lowongan perusahaan dapat diubah dan disimpan.	Valid
31	Kasus uji menghapus data lowongan	Data lowongan perusahaan tertentu dapat dihapus.	Valid
32	Kasus uji melihat daftar alumni yang mengirim lamaran	Perusahaan dapat melihat daftar alumni yang mengirim lamaran dan ditampilkan secara detail	Valid
33	Kasus uji men-download berkas lamaran	Perusahaan dapat mendownload berkas lamaran dan ditampilkan.	Valid
34	Kasus uji men-download berkas lamaran selain file word	Berkas lamaran tidak dapat di download dan ditampilkan.	Valid
35	Kasus uji menambah data kategori soal	Data kategori soal berhasil ditambah.	Valid
36	Kasus uji menambah data kategori soal dimana masukan nama, tanggal, jam dan durasi kosong.	Sistem menampilkan pesan kesalahan nama, tanggal, jam dan durasi tidak boleh kosong.	Valid
37	Kasus uji melihat daftar kategori soal	Daftar kategori soal dari perusahaan tersebut dapat ditampilkan.	Valid
38	Kasus uji mengubah kategori soal	Kategori soal perusahaan dapat diubah dan ditampilkan	Valid

39	Kasus uji menghapus kategori soal	Data kategori soal perusahaan tertentu dapat dihapus.	Valid
40	Kasus uji menambah data soal	Data soal perusahaan berhasil ditambah.	Valid
41	Kasus uji menambah data soal dimana kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban kosong.	Sistem menampilkan pesan kesalahan bahwa kategori soal, pertanyaan, pilihan A, pilihan B, pilihan C, pilihan D, jawaban tidak boleh kosong.	Valid
42	Kasus uji melihat data soal	Daftar soal dari perusahaan tersebut dapat ditampilkan.	Valid
43	Kasus uji melihat data soal dari perusahaan yang lain.	Sistem menampilkan daftar soal milik perusahaan masing-masing dan tidak bisa melihat daftar soal perusahaan yang lain.	Valid
44	Kasus uji mengubah daftar soal	Daftar soal perusahaan dapat diubah dan ditampilkan secara detail.	Valid
45	Kasus uji menghapus daftar soal	Data soal perusahaan tertentu dapat dihapus.	Valid
46	Kasus uji menambah data peserta test	Data peserta test lowongan pekerjaan berhasil ditambah.	Valid
47	Kasus uji menambah data peserta test dimana kategori soal, nama lowongan kosong.	Sistem menampilkan pesan kesalahan bahwa kategori soal dan nama lowongan tidak boleh kosong.	Valid
48	Kasus uji untuk menampilkan tombol ikuti test.	Sistem menampilkan tombol ikuti test online pada user alumni yang telah terdaftar.	Valid
49	Kasus uji melihat daftar peserta <i>test online</i>	Daftar peserta test dari lowongan perusahaan tersebut dapat ditampilkan.	Valid
50	Kasus uji menghapus daftar peserta <i>test</i>	Data peserta test perusahaan dapat dihapus.	Valid
51	Kasus uji mengubah data diri alumni	Data diri alumni dapat diubah dan ditampilkan	Valid
52	Kasus uji mengubah data diri alumni dimana nim,username dan ipk diubah.	Sistem menampilkan nim dan username tidak aktif dan ipk tidak dapat diganti.	Valid
53	Kasus uji mengubah data diri alumni dimana nama, jenis kelamin, tanggal lahir kosong.	Sistem menampilkan pesan kesalahan bahwa nama, jenis kelamin, tanggal lahir tidak boleh kosong.	Valid
54	Kasus uji ikuti <i>test online</i>	Daftar soal dari salah satu perusahaan dapat ditampilkan dan alumni dapat mengerjakan sesuai dengan durasi waktu.	Valid
55	Kasus uji mencari data alumni	Data alumni berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid
56	Kasus uji mencari data perusahaan	Data perusahaan berhasil ditampilkan berdasarkan pada <i>keyword</i> tertentu.	Valid

Sumber : Pengujian

BAB VII PENUTUP

7.1 Kesimpulan

Dari hasil perancangan, implementasi dan pengujian yang telah dilakukan, dapat dibuat kesimpulan sebagai berikut:

1. Dari hasil pengujian koneksi basis data dan *web server*, diperoleh bahwa koneksi *server* basis data MySQL dan *web server* Apache HTTP yang terletak di komputer *server* dalam sebuah jaringan LAN (*Local Area Network*) dapat dilakukan sehingga hubungan *client server* dapat berjalan.
2. Dari hasil pengujian antarmuka sistem yang dilakukan, diperoleh hasil bahwa alur menu, tampilan dan proses yang terjadi dalam sistem telah sesuai dengan perancangan.
3. Sistem *job placement center online* ini berhasil dirancang secara *object-oriented* dan dimodelkan menjadi empat operasi dasar yang sederhana, yaitu *create, read, update, delete* (CRUD).
4. Hasil dari semua rangkaian pengujian yang dilakukan menunjukkan bahwa Sistem *job placement center* dapat berfungsi dengan baik dan sesuai dengan kebutuhan Politeknik Negeri Malang.
5. Pengujian menunjukkan bahwa sistem yang dibangun adalah valid sesuai dengan kebutuhan user.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan *job placement center online* berbasis *web* antara lain:

1. Fitur - fitur pendukung sistem *job placement center online* ditambahkan dan dibuat lebih bervariasi terutama untuk soal-soal yang berbentuk gambar.
2. Sistem dapat disempurnakan dengan melengkapi webcam yang dapat memantau peserta *test online* untuk menghindari kecurangan.
3. Sistem dapat diintegrasikan dengan program-program lain seperti *psikotest online, yahoo messenger* dan lain-lain.

DAFTAR PUSTAKA

- [ARB-04] Arbie. 2004. *“Manajemen Database dengan MySQL”*. Yogyakarta: Andi Offset.
- [DHA-02] Dharma, Budi Sutedjo. 2002. *“Perancangan dan Pembangunan Sistem Informasi”*. Yogyakarta: Andi Offset.
- [FAT-02] Fathansyah, Ir. 2002. *“Basis Data”*. Bandung: Informatika.
- [KAD-98] Kadir, Abdul. 1998. *“Konsep & Tuntutan Praktis Basis Data”*. Yogyakarta: Andi Offset.
- [KAD-03] Kadir, Abdul. 2003. *”Dasar Pemrograman Web Dinamis Menggunakan PHP”*. Andi: Yogyakarta.
- [POH-97] Pohan, Husni Iskandar. 1997. *“Pengantar Perancangan Sistem”*. Jakarta: Erlangga.
- [PRA-03] Prasetyo, Didik Dwi. 2003. *“Kolaborasi PHP dan MySQL Untuk Membuat Web Database Yang Interaktif”*. Jakarta: Elex Media Komputindo.
- [WID-03] Widigdo, Anon Kuncoro. 2003. *“Dasar Pemrograman PHP dan MySQL”*.
Akses dari : <http://ilmukomputer.com/umum/anon>
- [SYU-99] Syukur, Mark Ade. 1999. *“Aplikasi Web Dengan PHP”*. Jakarta : Fakultas Ilmu Komputer. Universitas Gunadarma. Akses dari: <http://ilmukomputer.com/umum/mark-php.php>
- [ONN-01] Purbo, Onno W. 2001. *“TCP/IP”*. PT. Elex Media Komputindo. Jakarta.
- [KRI-04] Kristianto, Budhi. 2005. *“Referensi Coding Pemrograman Web HTML, ASP, PHP Tingkat Dasar”*. Yogyakarta: PD. Anindya.
- [PEN-04] Penelitian, Divisi Pengembangan &. 2004. *”Aplikasi Program PHP & MySQL untuk membuat Website Interaktif”*. Yogyakarta: Andi.
- [SUT-07] Sutisna, Dadan. 2007. *“7 Langkah Mudah Menjadi Webmaster”*. Jakarta: Media Kita.

- [RAM-05] Ramadhan, Arif, S. Kom. & Hendra Saputra, S. Kom. 2005. "PHP 5 dan MySQL". Elex Media Komputindo. Kelompok Gramedia: Jakarta.
- [KRI-05] Kristianto, Budhi. 2005. "*Referensi Coding Pemrograman Web HTML, ASP, PHP Tingkat Dasar*". PD. Anindya. Yogyakarta.
- [SIM-06] Simarmata, Janner & Imam Paryudi. 2006. "Basis Data". Yogyakarta: Andi.

