



**APLIKASI UJIAN PENYARINGAN UNTUK SISWA SMU BERBASIS
WEB
MENGUNAKAN PHP DAN MYSQL**

Disusun Oleh:

ADHELIA RIZTIANA
01502-002

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2009**



**APLIKASI UJIAN PENYARINGAN UNTUK SISWA SMU BERBASIS
WEB
MENGUNAKAN PHP DAN MYSQL**

Laporan Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:

ADHELIA RIZTIANA
01502-002

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2009**

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

NIM : 01502-002
Nama : ADHELIA RIZTIANA
Judul Skripsi : APLIKASI UJIAN PENYARINGAN SISWA
SMU BERBASIS WEB MENGGUNAKAN PHP
DAN MYSQL

Menyatakan bahwa skripsi tersebut diatas adalah hasil karya saya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan skripsi saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 30 Maret 2009

(ADHELIA RIZTIANA)

LEMBAR PERSETUJUAN

NIM : 01502-002
Nama : ADHELIA RIZTIANA
Judul Skripsi : APLIKASI UJIAN PENYARINGAN CALON
SISWA SMU BERBASIS WEB
MENGUNAKAN PHP DAN MYSQL

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI

JAKARTA,.....

Ahmad Kodar, Drs., MT

Dosen Pembimbing I

Anis Cherid, MTI

Pembimbing II

Devi Fitriyah, S.Kom., MTI

Koord. Tugas Akhir Teknik Informatika

Abdusy Syarif, ST., MT

Kaprodi Teknik Informatika

KATA PENGANTAR

Puji syukur Alhamdulillah kehadiran Allah SWT yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Jurusan Teknik Informatika Universitas Mercu Buana. Penulis menyadari bahwa laporan tugas akhir ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa laporan tugas akhir ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Ahmad Kodar, Drs., MT selaku pembimbing I tugas akhir pada Jurusan Teknik Informatika Universitas Mercu Buana.
2. Bapak Anis cherid, MTI selaku pembimbing II tugas akhir pada Jurusan Teknik Informatika Universitas Mercu Buana.
3. Ibu Devi Fitriannah, S.Kom., MTI selaku koordinator tugas akhir pada Jurusan Teknik Informatika Universitas Mercu Buana.
4. Bapak Abdusy Syarif, ST., MT selaku Ketua Program Studi Teknik Informatika Universitas Mercu Buana.
5. Kedua orang tua bapak Noviar Rizky dan Ibu tercinta Martiana Mardewi yang telah mendukung penulis baik spirit maupun materi.
6. Adik-adik ku tercinta yang selalu memberikan spirit untuk terus menyelesaikan tugas akhir ini.

7. Irwan hermawan yang memberikan spirit untuk terus menyelesaikan tugas akhir ini.
8. Teman-teman angkatan 2002 yang telah membantu Tafthazani, Jheny, Lina, Iqar, Ika, Manto, Agus, Anto dan Asmi. Dan telah memberikan dukungan moral untuk terus menyelesaikan tugas akhir ini

Semoga Allah SWT membalas kebaikan dan selalu mencurahkan hidayah serta taufikNya, Amin.

Jakarta, Maret 2009

Penulis

ABSTRACT

On progress of the technology that really fast like now computer is not only for electronic sets of equipment that be able to store data. However it's constitute instrument for assist human live everyday. But ther still a lot of institute of education not yet using the benefit in maximum. We can see from this method to perform examination for new students that still manually, which is for every students we give paper of question and answered sheet that they have to answer with using pencil or pen to write.

This need more time, besides of that the students still have to wait for the result. Because the teacher have to checking the answer, and there still much more process after that. This is not ef fective and efficient and also may possible the mistake happen when the teacher checking the answer sheet.

Observe of those problem, the writer try to make a design for a testing application systems that automatic, flexible, efficient and effective. So this is can ease the institute of education in controling and organizing the system and to minimize any mistake that may happen.

Keywords: Web, PHP, MySQL, Dreamweaver, Online Test.

ABSTRAK

Pada perkembangan teknologi yang begitu pesat seperti sekarang komputer bukan lagi hanya perangkat elektronik yang dapat menyimpan data. Namun merupakan alat bantu untuk kehidupan manusia sehari-hari. Tetapi masih banyak lembaga-lembaga pendidikan yang belum memanfaatkannya secara maksimal dapat dilihat dari cara melaksanakan ujian penyaringan siswa baru yang masih menggunakan cara manual, yaitu untuk setiap siswa diberikan kertas soal dan lembaran jawaban yang harus diisi menggunakan alat tulis.

Ini membutuhkan waktu lebih lama, disamping itu calon siswa juga masih harus menunggu untuk dapat mengetahui hasil ujian karena para guru harus memeriksa jawaban untuk setiap siswa, dan masih banyak lagi proses-proses yang dilakukan. Hal ini dirasa penulis tidak efektif dan efisien serta memungkinkan terjadinya kesalahan pada saat pemeriksaan jawaban ujian.

Melihat permasalahan tersebut, penulis mencoba membuat perancangan aplikasi sistem pengujian yang *automatic*, fleksibel, efisien serta efektif sehingga dapat memudahkan lembaga-lembaga pendidikan dalam mengontrol dan mengatur sistem tersebut serta meminimalis terjadinya kesalahan.

Kata kunci: Web, PHP, MySQL, Dreamweaver, Ujian Online.

DAFTAR ISI

	Halaman
LEMBAR PERNYATAAN	i
LEMBAR PERSETUJUAN	ii
KATA PENGANTAR	iii
ABSTRACT	vi
ABSTRAKSI	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xvi
 BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Tujuan Dan Manfaat.....	3
1.3 Rumusan Masalah.....	3
1.4 Batasan Masalah.....	4
1.5 Metodologi.....	5
1.5.1 Metode Analisis.....	5
1.5.2 Metode Perancangan.....	6
1.6 Sistematika Penulisan.....	7
 BAB II LANDASAN TEORI	
2.1 Pengertian Sistem Informasi.....	9
2.1.1 Perancangan Sistem Informasi.....	10

2.2 Konsep Dasar Rekayasa Piranti Lunak.....	11
2.2.1 Pengertian Piranti Lunak.....	12
2.2.2 Pengertian Rekayasa Piranti Lunak.....	12
2.2.3 Proses Rekayasa Piranti Lunak SDLC.....	13
2.3 UML (<i>United Modelling Language</i>).....	15
2.3.1 Use Case Diagram.....	16
2.3.2 Colaboration Diagram.....	16
2.3.3 Activity Diagram.....	17
2.3.4 Class Diagram.....	19
2.4 Teori Pemodelan Data.....	22
2.4.1 <i>Entity Relationship Diagram</i> (ERD).....	22
2.4.2 Kardinalitas Relasi.....	23
2.5 Pengenalan Database.....	25
2.5.1 Proses Perancangan Basis Data.....	26
2.5.2 <i>Database Management System</i>	26
2.5.3 Normalisasi.....	28
2.5.4 Flowchart.....	30
2.5.5 <i>Computer Based Information System</i>	32
2.6 Web Server.....	34
2.6.1 Web Browser.....	35
2.6.2 <i>Hyper Text Transfer Protocol</i> (HTTP).....	36
2.7 PHP (<i>Personal Home Page</i>)	36
2.8 SQL Dan MySQL.....	39
2.9 Dreamweaver.....	44

BAB III ANALISIS SISTEM DAN PERANCANGAN

3.1 Identifikasi Permasalahan.....	47
3.2 Evaluasi dan Penyelesaian Masalah.....	49
3.3 Perancangan aplikasi.....	50
3.3.1 Use Case Diagram.....	50
3.3.2 Activity Diagram	51
3.3.3 Collaboration Diagram.....	67
3.4 Perancangan Basis Data Pada Aplikasi ujian penyaringan masuk.....	71
3.3.1 Pemodelan Class Diagram.....	69
3.4.2 Perancangan Basis Data Secara Konseptual.....	71
3.5 Perancangan Layar Aplikasi Ujian Berbasis Web.....	75
3.5.1 Perancangan Layar Halaman Administrator.....	75
3.5.2 Perancangan Layar Halaman Daftar Ulang Peserta.....	79
3.5.3 Perancangan Layar Halaman Login Peserta.....	80
3.5.4 Perancangan layar Melihat Hasil ujian Seluruh Peserta.....	84
3.6 Spesifikasi Sistem.....	85
3.6.1 Spesifikasi Aplikasi	85
3.6.2 Spesifikasi Pengguna.....	85
3.7 Lingkungan Operasi.....	86

BAB IV PENGUJIAN SISTEM DAN IMPLEMENTASI

4.1 Implementasi Aplikasi Ujian Penyaringan Siswa Baru Berbasis Web..	87
4.1.1 Spesifikasi Kebutuhan Sistem.....	88
4.2 Tampilan Aplikasi Ujian penyaringan berbasis web.....	88
4.2.1 Halaman pembuka.....	88
4.2.2 Tampilan untuk halaman daftar ulang peserta.....	90
4.2.3 Tampilan Halaman untuk Login peserta.....	93
4.2.4 Tampilan Halaman untuk Admin.....	105
4.3 Pengujian.....	131
4.3.1 Skenario Pengujian.....	132
4.3.2 Hasil Pengujian.....	136
4.3.3 Analisis Hasil Pengujian.....	141

BAB V PENUTUP

5.1 Kesimpulan.....	149
5.2 Saran-Saran.....	150

DAFTAR PUSTAKA.....	151
----------------------------	------------

LAMPIRAN KODE PROGRAM.....	152
-----------------------------------	------------

DAFTAR GAMBAR

Gambar 2.1 Simbol-simbol DFD	10
Gambar 2.2 Pendekatan Model Waterfall.....	13
Gambar 2.3 Notasi Use Case.....	16
Gambar 2.4 Contoh Collaboration Diagram.....	17
Gambar 2.5 Contoh Activity Diagram.....	18
Gambar 2.6 Notasi ERD.....	23
Gambar 2.7 Relasi satu ke satu.....	24
Gambar 2.8 Relasi satu ke banyak.....	24
Gambar 2.9 Relasi Banyak ke Banyak.....	25
Gambar 2.10 Notasi Dasar Flowchart.....	30
Gambar 2.11 Flowchart Mencari nilai genap.....	32
Gambar 2.12 Component Of Computer Based Information System.....	33
Gambar 2.13 Layout Eksekusi Script PHP.....	39
Gambar 2.14 Tampilan hasil membuat database.....	41
Gambar 2.15 Tampilan membuat tabel pada database.....	42
Gambar 2.16 Tampilan memasukkan nama field didalam database.....	42
Gambar 2.17 Tampilan Menghapus field pada tabel.....	43
Gambar 2.18 Tampilan menyisipkan satu field diantara field yang lain....	43
Gambar 2.19 Tampilan dreamweaver dengan mode tampilan code.....	45
Gambar 2.20 Tampilan dreamweaver dengan mode Design.....	45
Gambar 2.21 Tampilan dreamweaver dengan mode split.....	46
Gambar 3.1 Diagram Use Case.....	50

Gambar 3.2 Activity Diagram login peserta ujian.....	51
Gambar 3.3 Activity Diagram Daftar ulang peserta ujian.....	52
Gambar 3.4 Activity Diagram Melakukan ujian penyaringan.....	54
Gambar 3.5 Activity Diagram Melihat hasil ujian peserta.....	56
Gambar 3.6 Activity Diagram Melihat hasil ujian seluruh peserta.....	57
Gambar 3.7 Activity Diagram Untuk login administrator.....	58
Gambar 3.8 <i>Activity Diagram</i> Mengelola data soal ujian.....	60
Gambar 3.9 <i>Activity Diagram</i> Mengelola data peserta ujian.....	62
Gambar 3.10 <i>Activity Diagram</i> Mengelola data nilai ujian.....	64
Gambar 3.11 <i>Activity Diagram</i> Mengelola data admin.....	65
Gambar 3.12 Collaboration Diagram untuk user.....	67
Gambar 3.13 Collaboration Diagram untuk admin.....	68
Gambar 3.14 Class Diagram untuk peserta.....	70
Gambar 3.15 Class Diagram untuk admin.....	70
Gambar 3.16 Rancangan Layar Halaman Utama Program.....	75
Gambar 3.17 Rancangan Layar Login Administrator.....	75
Gambar 3.18 Rancangan Layar pembuka pada halaman administrator... 	76
Gambar 3.19 Rancangan Layar halaman data peserta.....	77
Gambar 3.20 Rancangan Layar halaman Modul soal.....	77
Gambar 3.21 Rancangan Layar halaman data soal pada admin.....	78
Gambar 3.22 Rancangan Layar data nilai pada halaman admin.....	78
Gambar 3.23 Rancangan Layar data admin pada halaman admin.....	79
Gambar 3.24 Rancangan Layar untuk halaman daftar ulang peserta.....	79
Gambar 3.25 Rancangan Layar untuk halaman login peserta.....	80

Gambar 3.26 Rancangan Layar halaman peserta.....	80
Gambar 3.27 Rancangan Layar modul soal ujian untuk peserta.....	81
Gambar 3.28 Rancangan Layar soal ujian untuk peserta.....	81
Gambar 3.29 Rancangan Layar soal ujian peserta jika sudah dijawab.....	82
Gambar 3.30 Rancangan Layar jika jawaban ingin diubah.....	82
Gambar 3.31 Rancangan Layar peserta jika ingin mengakhiri ujian.....	83
Gambar 3.32 Rancangan Layar peserta jika telah selesai ujian.....	83
Gambar 3.33 Rancangan Layar halaman hasil ujian peserta.....	84
Gambar 3.34 Rancangan Layar halaman hasil ujian seluruh peserta.....	84
Gambar 4.1 Tampilan Halaman Utama Program.....	90
Gambar 4.2 Tampilan Halaman Daftar Ulang Peserta.....	91
Gambar 4.3 Tampilan Halaman jika Daftar Ulang Peserta Berhasil.....	92
Gambar 4.4 Tampilan Halaman Jika Daftar Ulang Tidak Berhasil.....	93
Gambar 4.5 Tampilan Halaman Untuk Login Peserta	94
Gambar 4.6 Tampilan Halaman Kesalahan Login Peserta.....	95
Gambar 4.7 Tampilan Halaman setelah peserta Login.....	96
Gambar 4.8 Tampilan Halaman Modul soal.....	98
Gambar 4.9 Tampilan Halaman Soal Ujian pada Modul.....	98
Gambar 4.10 Tampilan Halaman Jika peserta menekan tombol jawab.....	98
Gambar 4.11 Tampilan Halaman soal jika sudah dijawab.....	99
Gambar 4.12 Tampilan Halaman Pesan Jika jawaban diubah.....	100
Gambar 4.13 Tampilan Halaman Jika peserta menekan tombol selesai....	101
Gambar 4.14 Tampilan Halaman Pesan untuk melihat hasil ujian.....	102
Gambar 4.15 Tampilan Halaman Untuk Hasil ujian peserta.....	104

Gambar 4.16 Tampilan Halaman hasil ujian seluruh peserta ujian.....	106
Gambar 4.17 Tampilan Halaman Untuk Login Administrator.....	106
Gambar 4.18 Tampilan Halaman Kesalahan Login Administrator.....	106
Gambar 4.19 Tampilan Halaman Pembuka Untuk Administrator.....	108
Gambar 4.20 Halaman Data Peserta pada Link administrator.....	109
Gambar 4.21 Halaman tambah Data Peserta pada Link administrator...	111
Gambar 4.22 Halaman jika Data Peserta berhasil ditambah.....	112
Gambar 4.23 Halaman ubah Data Peserta.....	113
Gambar 4.24 Halaman jika Data Peserta berhasil diubah.....	113
Gambar 4.25 Alert jika Data Peserta ingin dihapus.....	115
Gambar 4.26 Halaman jika Data Peserta berhasil dihapus.....	115
Gambar 4.27 Halaman Link Modul Data Soal pada administrator.....	116
Gambar 4.28 Halaman Modul Data soal Matematika.....	117
Gambar 4.29 Halaman Modul Data soal Bahasa Inggris.....	117
Gambar 4.30 Halaman Modul Data soal Bahasa Indonesia.....	117
Gambar 4.31 Halaman Modul Data soal Biologi.....	118
Gambar 4.32 Halaman Modul Data soal Fisika.....	118
Gambar 4.33 Halaman Modul Data soal Sejarah.....	118
Gambar 4.34 Halaman tambah Data Soal pada Link administrator	120
Gambar 4.35 Halaman jika tambah Data Soal berhasil.....	120
Gambar 4.36 Halaman ubah Data Soal.....	121
Gambar 4.37 Halaman hapus Data Soal.....	122
Gambar 4.38 Halaman jika Data Soal berhasil dihapus.....	123
Gambar 4.39 Halaman Data Nilai pada Link administrator.....	124

Gambar 4.40 Halaman Hapus Data Nilai pada Link administrator.....	125
Gambar 4.41 Halaman Pesan Jika data berhasil dihapus.....	125
Gambar 4.42 Halaman data admin pada link administrator.....	127
Gambar 4.43 Halaman tambah data admin pada link administrator.....	128
Gambar 4.44 Halaman ubah data admin pada link administrator.....	129
Gambar 4.45 Halaman kotak pesan untuk menghapus data.....	130
Gambar 4.46 Halaman kotak pesan jika data berhasil dihapus.....	130

DAFTAR TABEL

Tabel 2.1 Simbol-simbol pada Activity Diagram.....	18
Tabel 2.2 Notasi Pada Class Diagram.....	20
Tabel 3.1 Spesifikasi naratif untuk <i>Use Case</i> Login Peserta.....	52
Tabel 3.2 Spesifikasi naratif untuk <i>Use Case</i> Daftar ulang peserta ujian...	53
Tabel 3.3 Spesifikasi naratif <i>Use Case</i> Melakukan Ujian penyaringan.....	54
Tabel 3.4 Spesifikasi naratif <i>Use Case</i> Melihat hasil ujian peserta.....	57
Tabel 3.5 Spesifikasi naratif <i>Use Case</i> Melihat hasil ujian seluruh peserta..	58
Tabel 3.6 Spesifikasi naratif untuk <i>Use Case</i> Login Administrator.....	59
Tabel 3.7 Spesifikasi naratif <i>Use Case</i> Mengelola data modul soal.....	60
Tabel 3.8 Spesifikasi naratif <i>Use Case</i> Mengelola data peserta.....	63
Tabel 3.9 Spesifikasi naratif <i>Use Case</i> Mengelola Data Nilai.....	64
Tabel 3.10 Spesifikasi naratif <i>Use Case</i> Mengelola data admin.....	66
Tabel 3.11 Tabel admin.....	71
Tabel 3.12 Tabel dt_peserta.....	72
Tabel 3.13 Tabel soal.....	73
Tabel 3.14 Tabel jawab.....	74
Tabel 4.1 Skenario Pengujian.....	132
Tabel 4.2 Hasil Pengujian	136

BAB I

PENDAHULUAN

1.1 Latar Belakang

Berkembangnya teknologi informasi dan komputer yang begitu pesat pada masa sekarang ini membuat manusia dituntut untuk lebih kreatif dan maju. Saat ini informasi dan komputer menjadi kebutuhan yang penting, komputer bukan lagi hanya sebagai perangkat elektronik yang dapat menyimpan data namun merupakan alat bantu yang memberikan kemudahan bagi manusia dalam kehidupannya sehari-hari. Dengan perkembangan teknologi ini sangat memungkinkan terciptanya aplikasi-aplikasi yang dibangun untuk memudahkan berbagai macam proses baik yang bersifat komersil, pendidikan, transaksi dan masih banyak lagi. Lembaga pendidikan terdiri dari para siswa, guru, dan juga para karyawan hal ini menunjukkan sebuah kumpulan dari orang-orang dan sistem tersebut saling terkait dan saling ketergantungan satu sama lain yang diatur sedemikian rupa sehingga menjadi satu kesatuan dan mempunyai struktur serta tujuan yang jelas.

Oleh karena itu dengan adanya perkembangan teknologi ini telah merubah cara pandang suatu organisasi dalam menjalankan usaha dan tugas mereka, Informasi merupakan kebutuhan vital dalam peranan kegiatan dan pengambilan keputusan yang akan mempengaruhi perkembangan organisasi ataupun lembaga.

Dalam hal ini penulis mencoba membangun suatu aplikasi yang bersifat pendidikan yaitu sebuah sistem yang menangani proses penyaringan untuk calon siswa sekolah menengah umum yang ingin masuk ke sekolah yang mereka tuju, dimana dengan adanya proses penyaringan ini nantinya akan dapat diketahui kemampuan masing-masing calon siswa tersebut.

Topik ini diangkat oleh penulis dikarenakan masih banyaknya lembaga-lembaga pendidikan yang masih menggunakan sistem ujian penyaringan masuk secara manual. Hal ini dirasa penulis banyak memiliki kekurangan diantaranya keefisienan, keefektifan dan keamanan yang masih belum tercapai.

Ujian penyaringan ini digunakan sebagai salah satu instrument penting dalam penerimaan calon siswa baru selain nem. Karena itulah diharapkan dengan aplikasi ini penulis dapat membangun suatu aplikasi sistem pengujian yang otomatis, dan baik dalam hal sumber daya waktu dan tenaga yang digunakan, dengan kata lain komputerisasi ini dapat mempercepat, menghemat waktu serta menunjang kegiatan pada lembaga-lembaga pendidikan serta dalam mengontrol dan mengaturnya.

1.2 Tujuan Dan Manfaat

Adapun tujuan dari tugas akhir ini adalah:

1. Untuk merancang dan menghasilkan suatu aplikasi ujian penyaringan calon siswa baru yang memiliki keakuratan penilaian yang baik. .
2. Untuk merancang suatu aplikasi berbasis web untuk ujian penyaringan secara intranet yang sesuai dengan kebutuhan *user*.

Sedangkan manfaatnya adalah untuk :

1. Memberikan kemudahan bagi calon siswa yang ingin masuk ke sekolah yang dituju dalam melakukan ujian penyaringan.
2. Mempermudah pihak lembaga pendidikan dalam meningkatkan efisiensi kerja *staff* administrasi dalam pengolahan data, dan hasil kerja sistem khususnya dalam bidang ujian penyaringan masuk.

1.3 Rumusan Masalah

Berdasarkan latar belakang diatas maka perumusan masalahnya adalah penyelenggaraan ujian penyaringan calon siswa baru yang masih dilakukan secara manual sehingga membutuhkan waktu lebih lama dalam mempersiapkannya, oleh karena itu untuk lebih mempermudah dan mengefisiensikan waktu yang terbuang dengan sistem pengujian yang masih dilakukan secara manual, maka akan dibuat aplikasi sistem pengujian berbasis web dengan menggunakan PHP sebagai bahasa pemrogramannya dan MySQL untuk *databasenya*.

1.4 Batasan Masalah

Dengan tetap mempertahankan keutuhan dari pokok permasalahan yang telah dikemukakan diatas sehingga skripsi yang dikerjakan dapat dipelajari secara keseluruhan maka perlu dibatasi permasalahan yang ada dimana batasan masalah sistem tersebut adalah sebagai berikut:

1. Dalam aplikasi ini calon siswa baru harus memasukan kode peserta yang didapat setelah calon siswa melakukan registrasi diluar aplikasi ini.
2. Aplikasi ini tidak membahas tentang proses registrasi dan perancangan ini diimplementasikan dengan bahasa pemrograman PHP (Personal Home Page) dengan menggunakan database MySQL..
3. Analisis yang dibutuhkan untuk perancangan aplikasi adalah proses ujian penyaringan masuk yang masih berjalan manual atau belum menggunakan sistem komputerisasi
4. Aplikasi ini dibangun bersifat intranet dimana hanya dapat dilakukan di sekolah yang bersangkutan dan pada waktu yang sudah ditentukan.
5. Memiliki modul administrator yang digunakan untuk konfigurasi diantaranya proses pengaturan data calon siswa, soal-soal ujian, dan data hasil ujian.
6. Dalam aplikasi ini memiliki beberapa modul soal yang sudah ditentukan dan pertanyaannya hanya dalam bentuk pilihan ganda saja berikut ini adalah modul soal yang terdiri dari beberapa materi, yaitu:
 - a. Modul Matematika, yaitu soal-soal yang terdiri dari materi matematika.
 - b. Modul Bahasa Inggris, yaitu soal-soal yang terdiri dari materi bahasa inggris.

- c. Modul Bahasa Indonesia, yaitu soal-soal yang terdiri dari ilmu-ilmu bahasa Indonesia.
 - d. Modul Biologi, yaitu soal-soal yang terdiri dari ilmu-ilmu biologi.
 - e. Modul Fisika, yaitu soal-soal yang terdiri dari ilmu-ilmu fisika.
 - f. Modul Sejarah, yaitu soal-soal yang terdiri dari ilmu-ilmu sejarah.
7. Peserta hanya dapat melakukan satu kali ujian penyaringan saja.

1.5 Metodologi

Metodologi yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut :

1.5.1 Metode Analisis terbagi dalam beberapa tahap di bawah ini :

1. Studi Kepustakaan (*Library Research*)

Metode ini dilakukan dengan cara mencari pengetahuan mengenai strategi sistem informasi yaitu dengan membaca buku-buku, yang berkaitan dengan topik permasalahan sehingga dapat diperoleh informasi serta teori-teori yang mendukung dalam penyusunan tugas akhir ini.

2. Studi atau kajian mengenai aplikasi ujian penyaringan masuk siswa baru.

Yaitu teknik dalam pengumpulan data atau keterangan menggunakan penelitian mengenai ujian penyaringan masuk siswa baru. Mekanisme yang dibahas dalam perancangan ujian penyaringan adalah bagaimana membuat perangkat lunak sebagai antarmuka, serta pengujiannya yang menggunakan sistem operasi Windows.

1.5.2. Metode Perancangan

Dalam penyusunan tugas akhir ini, metode perancangan yang digunakan didalam membuat analisis aplikasi web ini adalah:

1. Pembuatan Aplikasi dengan menggunakan PHP.
2. Pembuatan database dengan menggunakan MySQL.
3. Rekayasa Perangkat Lunak SDLC (*System Development Life Cycle*) adalah fase pendekatan untuk analisa dan desain untuk suatu sistem yang dikembangkan melalui daur tertentu dari analisis dan aktifitas *user*. Model *Waterfall* merupakan salah satu dari model-model yang terdapat pada penerapan SDLC. Ruang lingkup aktifitasnya adalah sebagai berikut:

1. Rekayasa sistem dan pemodelan sistem atau informasi

Rekayasa dan sistem analisis termasuk pencarian data yang dibutuhkan pada sistem analisis yang dilakukan serta desain tingkat puncak.

2. Analisis kebutuhan perangkat lunak

Suatu proses pengumpulan yang mendefenisikan semua yang dibutuhkan aplikasi yang akan dibuat.

3. Desain

Proses yang berfokus pada tiga atribut dalam sebuah aplikasi antara lain struktur data, perancangan menu dan perancangan antar muka.

4. Kode

Desain yang ditetapkan untuk struktur data, perancangan menu dan perancangan antar muka diterjemahkan kedalam bentuk program, dengan bahasa pemrograman PHP.

5 Pengujian

Mengarah pada logika internal aplikasi yang meyakinkan bahwa semua pernyataan telah diuji, dan pada eksternal fungsional yang lebih mengarahkan pada penemuan kesalahan dan memastikan input yang dimasukan akan memberikan hasil yang diinginkan.

6. Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk menampung perubahan-perubahan didalam lingkungan luarnya.

1.6 Sistematika Penulisan

Penggambaran secara umum dan singkat mengenai bab-bab yang ada dalam tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, tujuan dan manfaat, metode penulisan, batasan masalah, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas mengenai landasan-landasan teori yang digunakan dalam pembangunan aplikasi.

BAB III ANALISA DAN PERANCANGAN SISTEM

Bab ini akan membahas mengenai analisa sistem, yang di gunakan sebagai dasar pembangunan aplikasi yang akan di bangun. Di samping itu juga pemodelan sistem yang menggambarkan muatan dan aliran informasinya.

BAB IV IMPLEMENTASI APLIKASI DAN PENGUJIAN

Bab ini menguraikan tentang pengujian sistem yang penulis buat, apakah sistem sudah berjalan dengan baik.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan terhadap uraian yang telah diberikan pada bab-bab sebelumnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem Informasi

Menurut (McLeod 2001:11) sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Suatu organisasi seperti perusahaan atau suatu bidang fungsional cocok dengan definisi ini. Organisasi terdiri dari sejumlah sumber daya seperti telah kita bahas sebelumnya, dan sumber daya tersebut bekerja menuju tercapainya suatu tujuan tertentu yang ditentukan oleh pemilik atau manajemen.

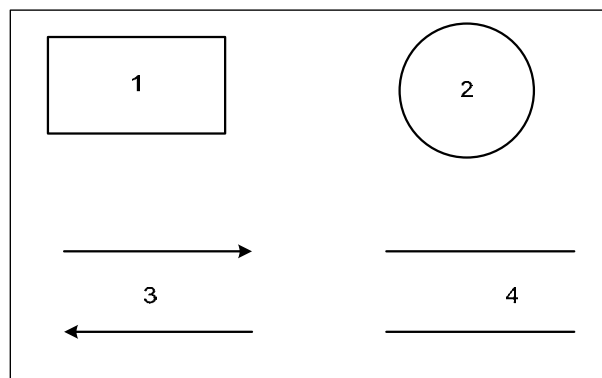
Menurut (McLeod 2000:15). data terdiri dari fakta-fakta dan angka-angka yang relatif tidak berarti bagi pemakai. Sebagai contoh, data dapat berupa jumlah jam kerja tiap pegawai dalam perusahaan. Informasi ialah data yang telah diproses, atau data yang memiliki arti.

2.1.1 Perancangan Sistem Informasi

Perancangan sistem menurut (Whitten 2004:472). merupakan suatu tugas yang berfokus pada spesifikasi khusus dari solusi berbasis komputer. Perancangan sistem disebut juga perancangan fisik. Analisis sistem menekankan pada masalah bisnis sementara perancangan sistem menekankan pada teknis atau implementasi dari sistem.

Perancangan sistem biasanya dilanjutkan dari model logis yang dikembangkan pada analisis model. Secara langsung, perancangan sistem model menjadi *blueprint* dari konstruksi dan implementasi sistem baru, Salah satu perancangan sistem yang menggunakan model adalah perancangan dengan DFD (*Data Flow Diagram*).

Menurut (Whitten 2004:344). *Data Flow Diagram* (DFD) ialah model proses yang digunakan untuk menggambarkan aliran dari data melalui sistem dan kerja atau proses yang ditampilkan oleh sistem. Beberapa keuntungan dari *data flow diagram* ialah hanya terdapat tiga buah simbol dan satu koneksi sehingga mudah untuk dimengerti. Simbol-simbol dan koneksi yang ada di DFD yaitu:



Gambar 2.1 (Whitten 2004:344) Simbol – simbol pada DFD

1. *External Entity* atau *Terminal*

Merupakan entitas yang berada di luar sistem, yang memberikan data kepada sistem (*source*) atau yang menerima informasi dari sistem (*sink*). *External Entity* tidak termasuk bagian dari sistem.

2. *Process*

Menggambarkan apa yang dilakukan oleh sistem. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa data masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering pula disebut *bubble*.

3. *Data Flow*

Menggambarkan aliran data dari suatu *entity* ke *entity* lainnya. Arah panah menggambarkan arah aliran data. Aliran data bisa terdapat di antara dua proses yang berurutan, dari *data store* ke proses dan sebaliknya, dari *source* ke proses, dan dari proses ke *sink*.

4. *Data Store*

Merupakan tempat menyimpan data. Proses dapat mengambil data dari atau memberikan data ke *data store*.

2.2 Konsep Dasar Rekayasa Piranti Lunak

Dalam merancang suatu aplikasi, perlu diadakan suatu pendekatan sistematis dalam rekayasa piranti lunak tersebut. Tujuan dari rekayasa piranti lunak adalah untuk menghasilkan piranti lunak yang ekonomis yang dapat diandalkan.

2.2.1 Pengertian Piranti Lunak

Menurut (Pressman 2001:3). piranti lunak adalah produk yang dirancang dan dibangun oleh software engineer. Produk itu meliputi program-program yang dijalankan dalam komputer, dokumen-dokumen baik dalam bentuk hard-copy maupun virtual dan data yang mengkombinasikan angka, teks serta meliputi representasi dari informasi gambar, video dan audio. Beberapa karakteristik piranti lunak dibanding dengan produk lain:

1. Piranti bersifat *intangible*, kita tidak dapat merasakan bentuk dari piranti lunak dan perancangan dari piranti lunak sulit untuk dibayangkan.
2. Mudah untuk mengembangkan suatu piranti lunak, akan tetapi sulit untuk mengembangkan suatu piranti lunak yang mudah dimengerti dan mudah dimodifikasi.
3. Piranti lunak tidak dapat rusak akan tetapi dapat ketinggalan zaman.

2.2.2 Pengertian Rekayasa Piranti Lunak

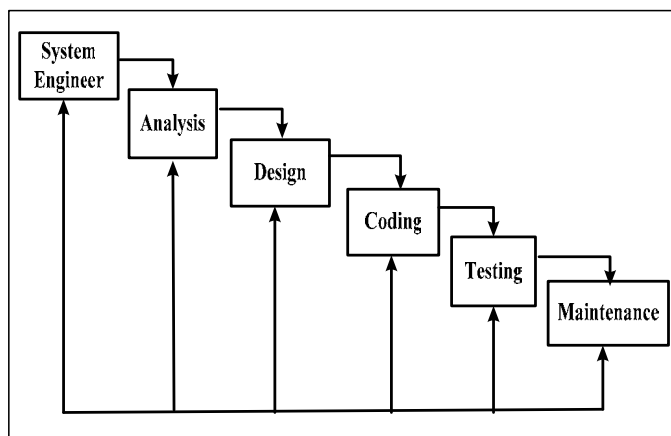
Tidak semua pengembangan piranti lunak disebut rekayasa piranti lunak, pengembangan suatu piranti lunak dapat disebut sebagai rekayasa piranti lunak apabila dalam serangkaian proses pengembangan tersebut digunakan pendekatan yang sistematis dengan memikirkan batasan-batasan seperti batasan biaya, waktu, dan batasan lainnya.

Berdasarkan pendapat (Lethbridge 2001:5). rekayasa piranti lunak merupakan suatu proses pemecahan masalah pelanggan dengan pengembangan sistematis dan evolusi dari sistem piranti lunak yang besar, berkualitas tinggi dan memenuhi batasan biaya, waktu serta batasan-batasan lainnya.

2.2.3 Proses Rekayasa Piranti Lunak *System Development Life Cycle (SDLC)*

SDLC adalah fase pendekatan untuk analisa dan desain untuk suatu sistem yang dikembangkan melalui daur tertentu dari analisis dan aktifitas *user* (Kenneth E. Kendall dan Julie E. Kendall). Model Waterfall merupakan salah satu dari model-model yang terdapat pada penerapan SDLC.

Waterfall disebut juga sebagai *classic life cycle* atau *linear sequential model*. Model ini memberikan sebuah pendekatan untuk pembangunan perangkat lunak yang sistematis dan berangkaian. Model *waterfall* pertama kali diajukan oleh (Winston Royce 1989:7).



Gambar 2.2 (Winston Royce 1989:7) Pendekatan Model *Waterfall*

Aktifitas-aktifitas dalam *waterfall* adalah sebagai berikut:

1. Tahapan Rekayasa Sistem pada tahapan ini menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak.
2. Tahapan Analisis pada tahap ini dilakukan pengumpulan perangkat lunak. Untuk memahami perangkat lunak yang dibangun, perekrayasa perangkat

lunak harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

3. Tahapan Perancangan pada tahapan perancangan perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut program yang berbeda; struktur data, arsitek perangkat lunak, representasi antarmuka, dan detail prosedural. Proses desain menerjemahkan kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.
4. Tahapan Pemograman pada desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Langkah pemograman melakukan tugas ini, jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.
5. Tahapan Pengujian sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika didalam perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan diluar fungsi yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.
6. Tahapan Pemeliharaan perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk menampung

perubahan-perubahan didalam lingkungan luarnya. Pemeliharaan perangkat lunak menerapkan lagi setiap tahap program sebelumnya dan tidak membuat yang baru lagi.

2.3 UML (*United Modelling Language*)

Menurut (Munawar 2005:17) UML adalah salah satu alat Bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek, dan UML merupakan kesatuan dari bahasa pemrograman yang dikembangkan oleh *Booch Object Modelling Technique (OMT)* dan *Object Oriented Software Engineering (OOSE)*, Booch sangat terkenal dengan nama metode *Design Object Oriented*, metode ini menjadikan proses analisis dan Desain kedalam 4 tahapan iterative yaitu :

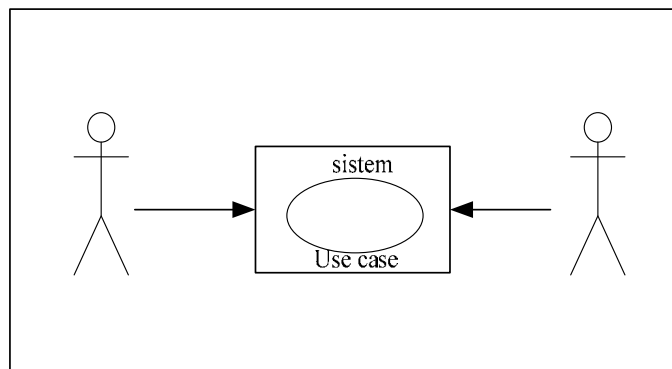
1. Identifikasi kelas – kelas
2. Obyek – obyek identifikasi semantic dari hubungan obyek dan kelas tersebut.
3. Perincian *interface*
4. Dan implementasi.

Karena UML bisa berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dari sistem dengan demikian semua anggota tim akan mempunyai gambaran yang sama tentang suatu sistem. Dan juga UML bisa berfungsi juga sebagai cetak biru karena, sangat lengkap dan detil dengan cetak biru ini maka akan bisa diketahui informasi detail tentang *coding program (Forward Engineering)* atau bahkan membaca program dan menginterpretasikannya kembali ke dalam diagram (*Reverse Engineering*), yang berguna pada situasi dimana *code*

program yang tidak terdokumentasi akan dimodifikasi atau dipelihara. Hal ini bisa terjadi ketika dokumentasi asli hilang atau bahkan belum dibuat sama sekali.

2.3.1 *Use case*

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna, *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Dengan demikian secara singkat bisa dikatakan *Use case* adalah serangkaian *scenario* yang digabungkan bersama – sama oleh tujuan umum pengguna. Notasi *Use case* dapat digambarkan pada gambar dibawah ini:



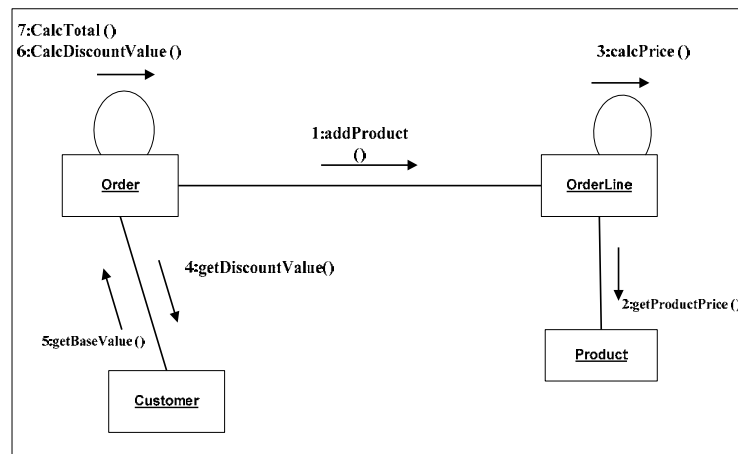
Gambar 2.3 (Munawar 2005:19) Notasi *Use Case*

2.3.2 *Colaboration Diagram*

Menurut (Munawar 2005:101) *Colaboration diagram* adalah perluasan dari obyek *diagram*. Atau obyek diagram menunjukkan obyek – obyek dan hubungannya satu dengan yang lain, untuk menunjukkan sebuah pesan buatlah 2 obyek arah panah menunjukkan obyek yang menerima pesan, label didekat panah

menunjukkan pesannya apa, tipikal *message* meminta kepada obyek yang menerimanya untuk menjalankan salah satu *operation*nya.

Sepasang tanda kurung digunakan untuk mengakhiri *message* jika ada *parameter* bisa diletakkan diantara tanda kurung. Antara *collaboration diagram* dengan *sequence diagram* bisa saling mengisi dengan demikian pada *collaboration diagram* kita bisa tambahkan nomor urut pada label sebuah *message* untuk menunjukkan urutan informasi. Titik dua (:) perlu digunakan untuk memisahkan nomor dengan *message*. Berikut ini contoh *Collaboration diagram* untuk *order*.



Gambar 2.4 (Munawar 2005:102) Contoh Collaboration Diagram

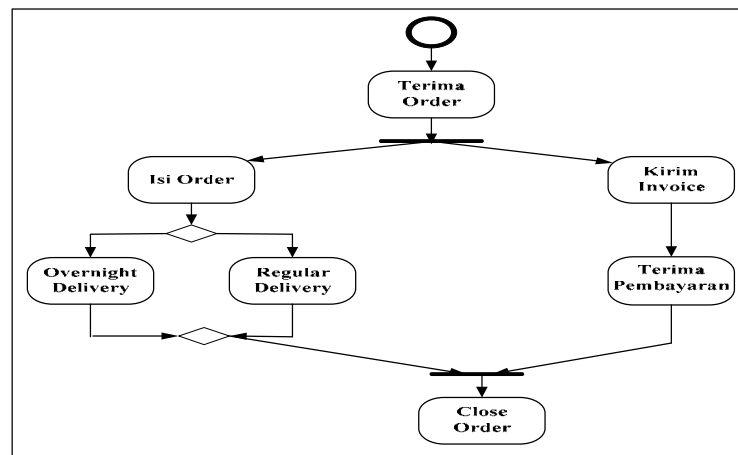
2.3.3 Activity Diagram

Menurut (Munawar 2005:109) *Activity diagram* adalah tehnik untuk mendeskripsikan logika *procedural*, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku

parallel sedangkan *flowchart* tidak bisa. Berikut ini adalah simbol – simbol yang sering digunakan dalam pembuatan *Activity Diagram*.

Tabel 2.1 (Munawar 2005:110) Simbol-simbol pada *Activity Diagram*

Simbol	Keterangan	Simbol	Keterangan
●	Titik Awal		Rake menunjukkan adanya dekomposisi
●	Titik Akhir		Tanda waktu
	Activity		Tanda pengiriman
◇	Pilihan untuk mengambil keputusan		Tanda penerimaan
—	Fork Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel		Aliran akhir (Flow Final)



Gambar 2.5 (Munawar 2005:111) Contoh *activity diagram* sederhana

Contoh sederhana *activity diagram* bisa dilihat pada Gambar 2.5 yang menjelaskan tentang saat proses penerimaan order, terlihat bahwa pengisian *order* dan pengiriman *invoice* terjadi secara paralel. Kondisi *paralel* jelas membutuhkan *sinkronisasi* dan *order* tidak akan ditutup sampai barang dikirim dan dibayar

untuk menunjukkan hal tersebut bisa digunakan *join* sebelum *action close order*, dengan *join* aliran keluar hanya akan dilakukan jika aliran kedatangan sampai ke *join* dengan demikian order hanya bisa ditutup jika pembayaran sudah dilakukan dan pengiriman sudah dilakukan.

Node pada *activity diagram* disebut dengan *action* bukan *activity*, *activity* menunjuk ke urutan *action* sehingga diagram tersebut menunjukkan *activity* yang membangun *action*. Perilaku bersyarat ditunjukkan dengan *decision* dan *merge decision* hanya mempunyai satu aliran masuk dan beberapa *guard* untuk aliran keluar, setiap aliran keluar mempunyai sebuah *guard* yaitu *Boolean* yang ditempatkan pada kurung kotak setiap kali mencapai *decision* hanya bisa mengambil satu keputusan sehingga *guard* harus *mutually exclusive* dan penggunaan *else* sebagai *guard* menunjukkan bahwa *guard* yang lain adalah salah.

2.3.4 Pemodelan Class Diagram

Diagram kelas ini digunakan untuk menampilkan kelas-kelas atau paket-paket didalam sebuah sistem dan relasi antara mereka, yang memberikan gambaran secara statis. Kelas adalah sebuah kategori yang membungkus informasi dan perilaku. Jadi diagram kelas merupakan alat perancangan terbaik untuk tim pengembang perangkat lunak. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum menuliskan kode program, membantu untuk memastikan bahwa sistem adalah rancangan terbaik (Sholiq, 2006).

Sebuah Class memiliki tiga area pokok :

1. Nama, merupakan nama dari sebuah kelas

2. Atribut, merupakan peroperti dari sebuah kelas. Atribut melambangkan batas nilai yang mungkin ada pada obyek dari class.
3. Operasi, adalah sesuatu yang bisa dilakukan oleh sebuah class atau yang dapat dilakukan oleh class lain terhadap sebuah class.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a. Private, tidak dapat dipanggil dari luar class yang bersangkutan.
- b. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak anak yang mewarisinya.
- c. Public, dapat dipanggil oleh siapa saja.
- d. Package, hanya dapat dipanggil oleh instance sebuah class pada paket yang sama.

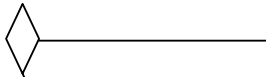
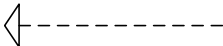
Berikut adalah notasi – notasi yang ada pada class diagram :


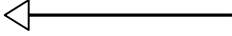
Tabel 2.2 Notasi pada Class Diagram

<p style="text-align: center;">Class</p>	<p>Class adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan properti/atribut class. Bagian akhir mendefinisikan method-method dari sebuah class.</p>	<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center; margin: 0;">Site config</p> <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> <p style="margin: 0;">+ SQLDNS:String +Adminemail: String</p> </div>
<p style="text-align: center;">Assotiation</p>	<p>Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa</p>	

	melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship (Contoh: One-to-one, one-to-many, many-to-many).	<u>1..n owned by 1</u>
--	--	------------------------

Lanjutan Tabel 3.2 Notasi pada Class Diagram

Composition	Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
Dependency	Dependency Kadangkala sebuah class menggunakan class yang lain hal ini disebut dependency umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah dependency dilambangkan sebagai sebuah panah bertitik-titik.	

<p>Aggregation</p>	<p>Aggregation mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah aggregation digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.</p>	
<p>Generalization</p>	<p>Generalization Sebuah relasi generalization sepadan dengan sebuah relasi inheritance pada konsep berorientasi obyek. Sebuah generalization dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “parent”-nya/induknya.</p>	

2.4 Teori Pemodelan Data

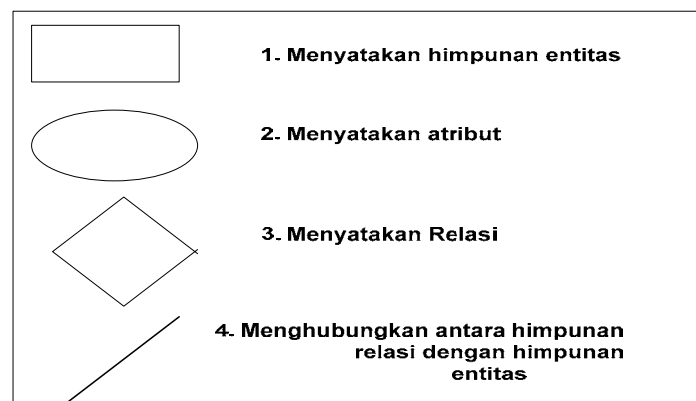
Menurut (Pressman 2002:354) pemodelan data menjawab serangkaian pertanyaan spesifik yang relevan dengan berbagai aplikasi pemrosesan data, Metode pemodelan data menggunakan ERD yang akan dijelaskan secara lengkap pada bagian ini memungkinkan perekayasa perangkat lunak mengidentifikasi objek data dan hubungannya dengan menggunakan notasi grafis.

Pada konteks analisis terstruktur, ERD menetapkan semua data yang dimasukan, disimpan, ditransnformasi, dan diproduksi pada suatu aplikasi. Model data terdiri dari tiga informasi yang saling tergantung yaitu:

1. Objek data adalah representasi dari hampir semua informasi gabungan yang harus dipahami oleh perangkat lunak.
2. Atribut menentukan properti suatu objek data dan mengambil salah satu dari tiga karakteristik yang berbeda.
3. Hubungan merupakan objek data disambungkan satu dengan yang lainnya dengan berbagai macam cara.

2.4.1 *Entity Relationship Diagram (ERD)*

Menurut (Peter Chen 77) ERD di gunakan untuk desain sistem *database* rasional. Tujuan utama ERD adalah untuk mewakili objek data dan hubungan antar mereka. Notasi-notasi dalam pembuatan *Entity Relationship Diagram* dapat di gambarkan pada Gambar 2.6.



Gambar 2.6 (Peter Chen 77) Notasi-notasi ERD

Keterangan dari tiap notasi adalah sebagai berikut:

1. Entitas

Entitas adalah objek yang mewakili sesuatu yang nyata. Contoh: Mahasiswa.

2. Atribut

Atribut merupakan Gambaran ciri-ciri entitas. Contoh: Nim, nama, alamat.

3. Relasi

Relasi adalah hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

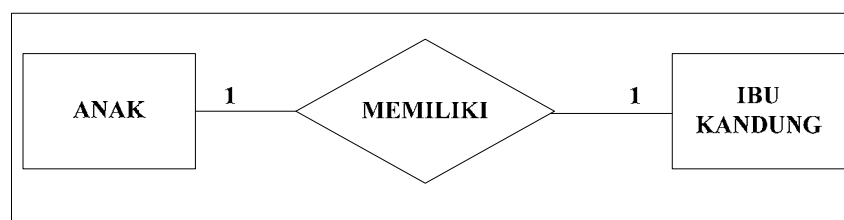
2.4.2 Kardinalitas Relasi

Selain menggunakan notasi, keterhubungan antar entitas juga menggunakan kardinalitas relasi, dimana definisi dari kardinalitas relasi tersebut menurut (Tillman 93) adalah merupakan spesifikasi dari sejumlah peristiwa dari satu objek yang dapat dihubungkan kesejumlah peristiwa dari objek yang lain.

Dan berikut ini adalah jenis-jenis dari kardinalitas relasi:

1. Relasi satu-ke-satu

Relasi ini menyatakan bahwa setiap entitas mempunyai hubungan paling banyak satu. Contoh: Satu Orang anak hanya dapat memiliki satu orang ibu kandung.



Gambar 2.7 Relasi satu ke satu.

2. Relasi satu-ke-banyak

Relasi ini menyatakan bahwa satu entitas mempunyai hubungan dengan banyak entitas, Contoh setiap dosen dapat mengajar lebih dari satu mata

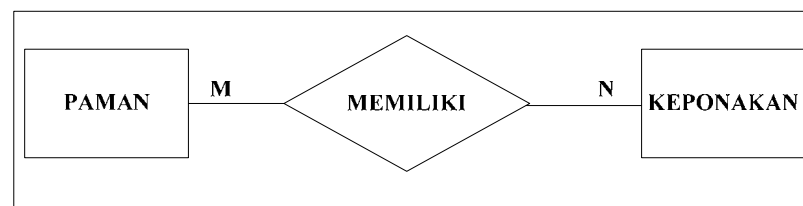
kuliah, sedang setiap mata kuliah diajar hanya oleh paling banyak satu orang dosen.



Gambar 2.8 Relasi satu ke banyak.

3. Relasi banyak-ke-banyak

Relasi ini menyatakan bahwa banyak entitas dapat berhubungan dengan banyak entitas. Contoh: Banyak Paman memiliki banyak keponakan.



Gambar 2.9 Relasi Banyak ke Banyak

2.5 Pengenalan *Database* (Basis Data)

Menurut (Kadir 2003:7) Data adalah nilai (*value*) yang turut merepresentasikan deskripsi dari suatu objek atau kejadian (*event*). Sedangkan Informasi merupakan hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu *event* (kejadian-kejadian) yang nyata yang digunakan untuk pengambilan keputusan.

Database adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dan dengan *software* untuk

melakukan manipulasi untuk kegunaan tertentu. *Database* diperlukan karena beberapa hal di antaranya:

1. Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
2. Menentukan kualitas informasi yang akurat, tepat pada waktunya dan relevan. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
3. Mengurangi duplikasi data (*data redudancy*).
4. Hubungan data dapat ditingkatkan (*data relatability*).
5. Mengurangi pemborosan tempat simpanan luar.

2.5.1 Proses Perancangan Basis Data

Menurut (Kadir 2003:39) Proses perancangan basis data, terlepas dari dari masalah yang ditangani dibagi menjadi tiga tahapan, yaitu:

1. Perancangan basis data secara konseptual.
2. Perancangan basis data secara logis.
3. Perancangan basis data secara fisis.

Perancangan basis data secara konseptual merupakan upaya untuk membuat model yang masih bersifat konsep. Perancangan basis data secara logis merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai, seperti model relasional, hirarki, atau jaringan. Namun sebagaimana halnya perancangan basis data secara konseptual, perancangan ini tidak tergantung pada DBMS (*DataBase Management System*) yang akan dipakai.

Itulah sebabnya perancangan basis data secara logis terkadang disebut pemetaan model data.

Perancangan basis data secara fisis merupakan tahapan untuk menuangkan perancangan basis data yang bersifat logis menjadi basis data fisis yang tersimpan pada media penyimpanan eksternal.

2.5.2 Database Management System

Menurut (Kadir 2003:17) *Database Management System* (DBMS) adalah sebuah perangkat lunak yang digunakan untuk memasukan, mengubah, menghapus, memanipulasi, dan memperoleh data atau informasi dengan praktis dan efisien.

Menurut (Ramakrishnan 2002:9), penggunaan *Database Management System* memiliki banyak keuntungan, antara lain :

1. Independensi data. Program aplikasi sebaiknya tidak menangani semua detail dari representasi dan penyimpanan data. DBMS menyediakan gambaran abstrak dari data dan menyembunyikan detail-detail tersebut.
2. Akses data yang efisien. DBMS menggunakan teknik-teknik khusus untuk menyimpan dan mengambil data secara efisien.
3. Integritas dan keamanan data. Jika data selalu diakses dengan menggunakan DBMS, integritas data akan selalu dipertahankan. DBMS juga dapat menerapkan kontrol akses yang menentukan bagian-bagian data mana yang dapat diakses oleh kelompok pengguna yang berbeda.
4. Administrasi data. Ketika data digunakan bersama-sama, administrasi data yang terpusat dapat memberikan peningkatan yang berarti. Representasi dari

data yang ada dapat diatur sedemikian rupa sehingga perulangan data dapat diminimalisasi dan efisiensi pengambilan data dapat ditingkatkan.

5. Akses bersamaan dan pemulihan kegagalan. DBMS dapat mengatur beberapa akses yang terjadi bersamaan sehingga penggunanya merasa bahwa data tersebut hanya diakses oleh satu orang saja. Lebih jauh lagi, DBMS melindungi penggunanya dari dampak kegagalan sistem.
6. Waktu pengembangan aplikasi yang lebih singkat. DBMS mendukung fungsi-fungsi penting yang dikenal secara umum oleh banyak aplikasi yang mengakses data dalam DBMS tersebut, sehingga aplikasi tersebut tidak perlu menangani detail fungsi tersebut yang akan mempercepat pengembangan aplikasi.

Sebuah *Database Management System* mempunyai tiga komponen (Laudon 2004:224) yaitu :

1. *Data Definition Language* (DDL)

Merupakan bahasa formal yang digunakan oleh para *programmer* untuk menentukan isi dan struktur dari *database*. DDL mendefinisikan setiap elemen data yang tampak dalam *database* sebelum elemen data tersebut diterjemahkan kedalam bentuk-bentuk yang dibutuhkan oleh program aplikasi.

2. *Data Manipulation Language* (DML)

Merupakan bahasa khusus yang dimiliki oleh sebuah DBMS yang digunakan bersamaan dengan bahasa pemrograman konvensional untuk memanipulasi data dalam *database*. Bahasa ini mengandung perintah-perintah yang memungkinkan pemakai untuk mengambil data dari *database* untuk memenuhi

informasi yang diminta. DML yang paling banyak digunakan pada saat ini adalah SQL atau *Structured Query Language*.

3. *Data Dictionary*

Data Dictionary adalah *file* manual atau otomatis yang menyimpan definisi dari elemen data dan karakteristik dari data seperti penggunaan, representasi fisik, kepemilikan, otorisasi dan keamanan.

2.5.3 Normalisasi

Menurut (Kadir 2003:65) Normalisasi merupakan sebuah teknik dalam logikal desain sebuah basis data (*database*), teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi).

Berikut bentuk-bentuk normalisasi :

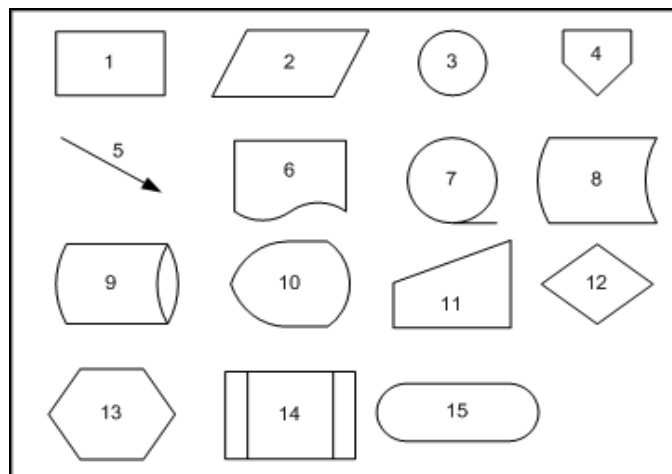
1. Normal Pertama (1st Normal Form) Memiliki ketentuan sebagai berikut:
 - a. Mendefinisikan atribut kunci.
 - b. Tidak adanya group berulang.
 - c. Semua atribut bukan kunci tergantung pada atribut kunci
2. Normalisasi Kedua (2nd Normal Form). Memiliki ketentuan sebagai berikut:
 - a. Sudah memenuhi dalam bentuk normal kesatu.
 - b. Sudah tidak ada ketergantungan parsial, dimana seluruh field hanya tergantung pada sebagian field kunci.
3. Normalisasi Ketiga (3rd Normal Form). Memiliki ketentuan sebagai berikut:
 - a. Sudah berada dalam bentuk normal kedua.
 - b. Tidak ada ketergantungan transitif (dimana field bukan kunci tergantung pada field bukan kunci lainnya).

Normal seharusnya berada dalam bentuk normal tertinggi dan bergerak dari bentuk normal satu dan seterusnya untuk setiap kali membatasi hanya satu jenis redundansi. Keseluruhannya cuma ada lima bentuk normal.

Tiga bentuk normal pertama menekankan redundansi yang muncul dari fungsi dependensi (*Function Dependencies*), sedangkan bentuk keempat dan kelima menekankan redundansi yang muncul dari kasus isi dependensi (*Multi Valued Dependencies*).

2.5.4 Flowchart

Menurut (Rademacher dan Gibson 1983:3) Flowchart merupakan metode yang menggambarkan dalam bentuk skematik aliran data dalam sistem. Notasi-notasi flowchart yang dipakai penulis dapat dilihat pada Gambar 2.10.



Gambar 2.10 (Rademacher dan Gibson 1983:3) Notasi-notasi dasar flowchart.

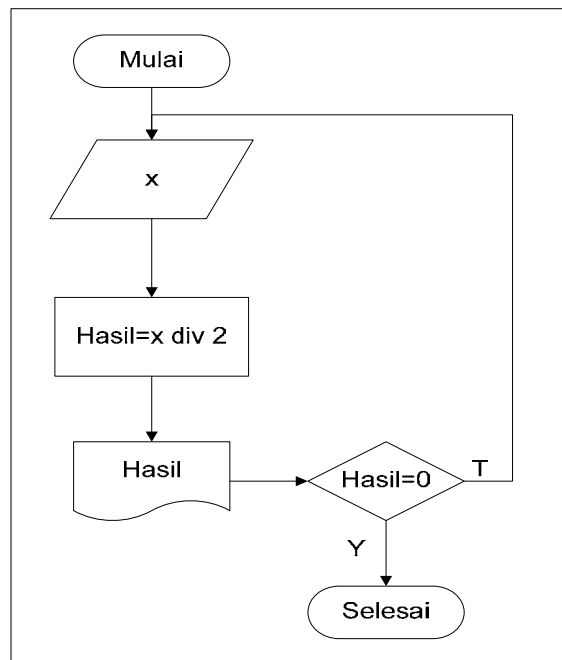
Berikut penjelasan dari notasi-notasi pada Gambar 2.10:

1. Simbol 1 adalah proses. Mewakili setiap proses yang menyebabkan perubahan pada nilai, bentuk, atau lokasi data.
2. Simbol 2 adalah fungsi masukan/keluaran. Mewakili fungsi dasar dari masukan data ke komputer atau mengeluarkan informasi.

3. Simbol 3 adalah On-page connector. Digunakan untuk mengidentifikasi titik umum dari jalur dimana hubungan garis tidak dapat digambar karena ruang yang terbatas pada halaman.
4. Simbol 4 adalah Off-page connector. Digunakan untuk menghubungkan aliran logis dari satu halaman ke halaman yang lain.
5. Simbol 5 adalah garis. Digunakan untuk menunjukkan jalur yang diikuti oleh data.
6. Simbol 6 adalah dokumen. Digunakan untuk menunjukkan perangkat keluaran.
7. Simbol 7 adalah magnetic tape. Digunakan sebagai perangkat masukan dan keluaran.
8. Simbol 8 adalah On-line storage. Mengindikasikan penggunaan kapasitas perangkat penyimpanan magnetic yang besar, seperti disk, drum, data sel, atau sistem penyimpanan massa.
9. Simbol 9 adalah Magnetic drum. Menunjukkan masukan, keluaran, atau penyimpanan data menggunakan drum.
10. Simbol 10 adalah tampilan. Menunjukkan informasi keluaran dari terminal CRT, plotter, dan sebagainya.
11. Simbol 11 adalah masukan manual. Menandakan penggunaan perangkat keyboard untuk memasukkan data ke komputer.
12. Simbol 12 adalah keputusan. Penanda titik percabangan dalam program.
13. Simbol 13 adalah persiapan. Menandakan modifikasi instruksi untuk merubah program.

14. Simbol 14 adalah Predifined process. Menunjukkan penamaan proses atau operasi atau sekumpulan langkah-langkah program dalam kumpulan flowchart.

Sebagai contoh, fungsi untuk mencari nilai genap pada hasil bagi nilai x dengan angka 2. Jika hasil mendapatkan nilai 0 maka nilai genap di dapatkan jika tidak maka akan kembali ke input data. Gambaran contoh ini dapat dilihat pada gambar 2.11.



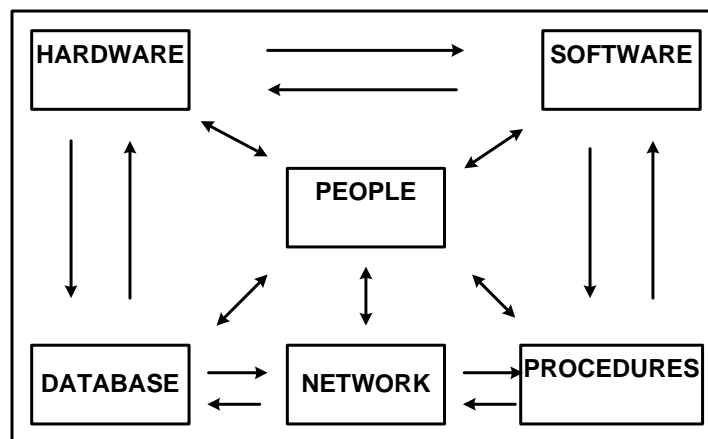
Gambar 2.11 Flowchart mencari nilai genap.

2.5.5 Computer Based Information System

Computer Based Information System adalah Sistem Informasi yang menggunakan komputer dan teknologi informasi untuk melaksanakan beberapa maupun seluruh tugas yang ada (Turban 2001:17).

Komponen dasar dari Sistem Informasi adalah sebagai berikut :

1. *Hardware* : Seperangkat alat-alat seperti *processor*, monitor, *keyboard*, dan *printer* yang menerima data dan informasi, kemudian memproses data dan informasi tersebut, dan menampilkannya kepada user.
2. *Software* : Seperangkat program komputer yang memungkinkan *hardware* dalam memproses data.
3. *Database* : Koleksi dari file-file yang saling berhubungan, *record-record*, yang menyimpan data dan memiliki asosiasi diantara data-data tersebut.
4. *Network* : Sistem yang saling berhubungan yang mengijinkan pembagian data antara komputer yang berbeda.
5. *Procedures* : Strategi, kebijakan, metode, dan aturan-aturan dalam menggunakan Sistem Informasi.
6. *People* : Elemen yang terpenting dalam Sistem Informasi, termasuk orang yang bekerja dengan Sistem Informasi atau menggunakan hasil dari sistem informasi tersebut.



Gambar 2.12 (Turban 2001:17) CBIS

Computer Based Information System (CBIS) memberikan banyak kontribusi yang sangat dibutuhkan bagi manajemen sebuah perusahaan dan saat ini sistem informasi merupakan isu yang paling penting dalam pengendalian manajemen. Hal ini disebabkan karena tujuan dari pengendalian manajemen adalah untuk membantu manajemen dalam mengkoordinasi subunit-subunit dari organisasi dan mengarahkan bagian-bagian tersebut untuk mencapai tujuan perusahaan.

Dua hal yang menjadi perhatian dari definisi diatas adalah mengkoordinasi dan mengarahkan. Tentu saja dalam dua proses tersebut diperlukan satu sistem agar proses koordinasi dan pengarahan dapat berjalan secara efektif sehingga tujuan perusahaan dapat tercapai.

Manfaat utama dari perkembangan sistem informasi bagi sistem pengendalian manajemen adalah :

1. Penghematan waktu (*time saving*).
2. Penghematan biaya (*cost saving*).
3. Peningkatan efektivitas (*effectiveness*).
4. Pengembangan teknologi (*technology development*).
5. Pengembangan personel akuntansi (*accounting staff development*).

Dengan berbagai manfaat dan kontribusi yang diberikan tersebut, diharapkan bahwa setiap perusahaan dapat bertahan dalam arena kompetisi yang semakin ketat.

2.6 *Web Server*

Web Server merupakan tempat bagi objek-objek web, yang setiap halamannya merupakan pengalamatan dari sebuah *URL*. *Web Server* juga implementasi sisi server dari HTTP (Kurose 2003:90). Fungsi dari *web server* (Kurose 2003:151) antara lain adalah sebagai berikut :

1. Menangani hanya satu HTTP request.
2. Mengakses dan memproses HTTP request.
3. Mengambil file yang di *request* dari sistem file server.
4. Membuat pesan *reponse* HTTP yang terdiri dari file yang di *request* yang diproses oleh *header lines*.
5. Mengirim *response* secara langsung kepada *client*.

Web Server memproses *request* untuk mengakses sebuah *web page*, *file*, atau sistem yang lain, dimana server berlokasi pada komputer yang sama, pada jaringan yang sama, pada organisasi yang sama, atau pada lokasi *remote* yang sama pada sebuah internet. Software *web server* juga seringkali disebut sebagai HTTPD, *Hyper Text Transfer Protocol Daemon* (Abrams 2001:42).

2.6.1 *Web Browser*

Menurut (Tanenbaum 1996:682) *Web Browser* adalah halaman-halaman yang dapat dilihat melalui sebuah program. Menurut Turban (2001:224) *Web Browser* adalah sebuah program yang dirancang untuk mengambil informasi-informasi yang dapat berupa GIF (*Graphics Interchange Format*), JPEG (*Join Photographic Expert Group*) sebagai gambar dan *Microsoft WAV* sebagai media

suaranya. Menurut (Govanus 1999:4) *Web Browser* adalah *client program* yang meminta layanan dari sebuah *Web Server*. *Web browser* merupakan sebuah *user agent* untuk *web*. *Web browser* menampilkan *web page* yang di *request* dan menyediakan sejumlah fitur-fitur navigasi dan konfigurasi. *Web browser* juga implementasi sisi *client* dari HTTP (Kurose 2003:89).

2.6.2 Hyper Text Transfer Protocol (HTTP)

HTTP (*Hyper Text Transfer Protocol*) adalah standar *web transfer protocol*. Setiap interaksi yang terjadi terdiri dari satu *ASCII request*, yang diikuti dengan *response* oleh RFC 822 MIME (Tanenbaum 1996:689-690). HTTP merupakan sebuah *protocol* yang digunakan untuk mentransfer *web pages* melalui internet (Connolly 2002:949).

HTTP mengalami perubahan secara konstan. Beberapa versi sedang digunakan sedangkan versi yang lain sedang dalam pengembangan. *Protocol* HTTP terdiri dari dua item yang berbeda yaitu :

1. Sekumpulan *request* dari *browser* untuk *server*
2. Sekumpulan *response* dari *server* untuk *browser*

2.7 PHP (Personal Home Page)

Menurut (Rafiza, p1) PHP dibuat pada tahun 1994 dan merupakan hasil kerja keras satu orang, Rasmus Lerdorf. Kemudian dilanjutkan oleh orang-orang lain dan telah melewati tiga kali penyusunan ulang secara besar-besaran untuk memberikan hasil produk yang matang seperti yang ada sekarang ini.

PHP adalah bahasa pemrograman untuk sisi *server* yang dirancang secara khusus untuk *web*. Dalam halaman HTML dapat dimasukkan kode-kode PHP yang akan dijalankan setiap kali halaman tersebut dieksekusi.

Kode-kode PHP akan diinterpretasikan pada *server web* dan menghasilkan HTML atau output lainnya yang akan dilihat oleh pengunjung *web*. PHP merupakan produk *open source*. PHP awalnya berarti *Personal Home Page*, tetapi kemudian diubah dengan penamaan konvensi rekursif GNU menjadi *PHP Hypertext Preprocessor*.

Beberapa pesaing dari PHP ialah Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) dan Allaire Cold Fusion. Dalam perbandingan dengan produk-produk tersebut, PHP memiliki beberapa kekuatan termasuk diantaranya ialah :

1. Performa yang tinggi

PHP sangat efisien. Menggunakan sebuah *server* yang tidak mahal, dapat melayani berjuta-juta permintaan setiap harinya.

2. Integrasi *Database*

PHP memiliki koneksi yang memungkinkan penggunaan berbagai sistem *database*. Selain dari koneksi ke MySQL, PHP juga dapat melakukan koneksi langsung ke *database* PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, InterBase dan Sybase. Menggunakan *Open Database Connectivity Standard* (ODBC), PHP dapat melakukan koneksi ke banyak *database* yang disediakan oleh driver ODBC, termasuk produk Microsoft.

3. Library yang built-in

Karena PHP dirancang untuk digunakan pada *web*, PHP memiliki banyak fungsi yang telah dibangun untuk mendukung berbagai tugas yang berguna pada *web*. Dengan PHP, kita dapat menampilkan gambar GIF, terkoneksi dengan layanan jaringan yang lain, mengirim *email*, dan membuat *file* PDF, semuanya hanya dengan beberapa baris kode.

4. Biaya yang rendah

PHP dapat diperoleh secara gratis. PHP dapat di-*download* kapanpun dari <http://www.php.net> tanpa biaya.

5. Mudah dipelajari dan digunakan

Sintaks dari PHP mengambil dasar bahasa pemrograman lain, terutama C dan Perl. Jika telah menguasai C atau Perl atau bahasa sejenis C seperti C++ atau Java, maka dapat dikatakan PHP hampir dapat secara langsung digunakan secara produktif.

6. Portabilitas

PHP dapat digunakan pada banyak sistem operasi. Kode PHP dapat ditulis pada sistem operasi UNIX yang gratis seperti Linux dan FreeBSD, sistem operasi UNIX yang komersial seperti Solaris dan IRIX atau berbagai versi dari Microsoft Windows.

7. Ketersediaan *Source Code*

Source code PHP dapat diakses. Tidak seperti produk komersial yang *source code*-nya tertutup, jika ada sesuatu yang hendak dimodifikasi atau ditambahkan pada PHP, dapat dilakukan secara langsung dan gratis.

Berikut ini adalah contoh script dari PHP.


```

<html>

<head> <title>Kode PHP</title> </head>

<body>

Ini Menggunakan Kode PHP <BR>

<?php

printf("Hari Ini Tgl. : %s ", Date ("d F Y"));

?>

</body>

</html>

```



Gambar 2.13 Layout eksekusi script PHP

2.8 SQL Dan MySQL

Menurut (Kadir 2003:101) SQL (dibaca "ess-que-el") merupakan kependekan dari *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah *Database*, sesuai dengan ANSI, SQL merupakan bahasa standar untuk sistem manajemen *database* relasional. Statemen SQL digunakan untuk melakukan tugas-tugas seperti melakukan *update* terhadap *database*, atau mengambil data dari sebuah database. Namun demikian, perintah

standar SQL seperti "*Select*", "*Insert*", "*Update*", "*Delete*", "*Create*", dan "*Drop*" dapat digunakan untuk melakukan hampir semua hal yang perlu dilakukan terhadap sebuah database. Tutorial ini akan mengenalkan kamu pada penggunaan dasar dari perintah tersebut.

Kamu bisa praktek dengan menginstal MySQL atau *database* lain di komputer kamu. (Sunday, 22 October 2006 - Contributed by Prothelord – Last. www.tutorSQL.com). Menurut (Welling and Thomson 2001:3), MySQL (dibaca *My-Ess-Que-Ell*) adalah *Relational Database Management System* (RDBMS) yang cepat dan kuat.

Suatu *database* memungkinkan kita untuk melakukan penyimpanan, pencarian, pengurutan dan pengambilan data secara efisien. *Server* MySQL mengendalikan akses ke data untuk menjamin bahwa beberapa *user* dapat bekerja bersama-sama, menyediakan akses yang cepat dan menjamin hanya *user* yang memiliki hak yang dapat mengakses.

Karena itu MySQL ialah *server multi-user, multi-threaded*. MySQL menggunakan SQL (*Structured Query Language*), standar bahasa *query database* di seluruh dunia. MySQL dipublikasikan sejak 1996, tetapi memiliki sejarah perkembangan dari tahun 1979.

MySQL memiliki banyak kekuatan, termasuk didalamnya performa yang tinggi, biaya yang rendah, mudah dikonfigurasi dan dipelajari, portabilitas dan *source code* yang tersedia secara umum. Dan berikut ini adalah beberapa kelebihan dari MySQL:

1. Performa yang tinggi

MySQL tidak dapat disangkal lagi cepat. Banyak dari hasil perbandingan menunjukkan bahwa MySQL lebih cepat dari pada para pesaing lainnya.

2. Biaya yang rendah

MySQL tersedia tanpa biaya apapun, dibawah lisensi *open source*, atau biaya yang rendah dibawah lisensi komersial jika diperlukan untuk aplikasi.

3. Mudah digunakan

Sebagian besar *database* modern menggunakan SQL. Jika telah menggunakan salah satu RDBMS, tidak akan mengalami kesulitan untuk beradaptasi dengan MySQL. MySQL juga lebih mudah diatur dari pada produk-produk lainnya.

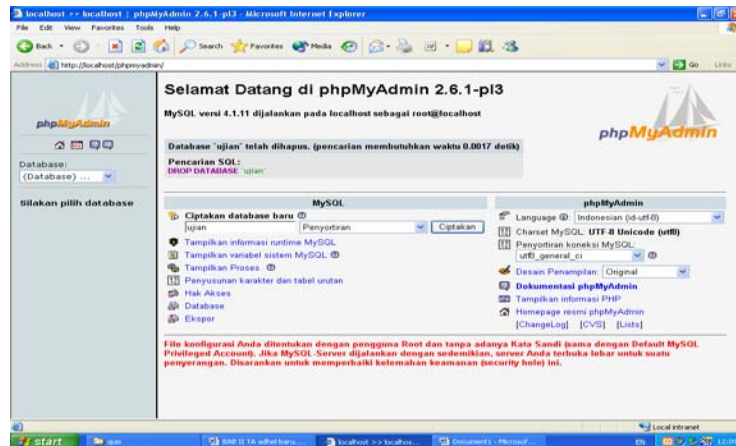
4. Portabilitas

MySQL dapat digunakan pada sistem UNIX yang berbeda-beda, juga pada Microsoft Windows.

5. Ketersediaan *Source Code*

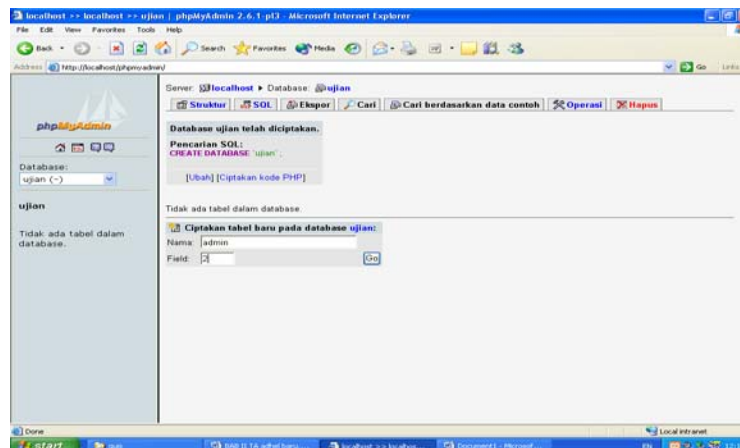
Seperti PHP, kode MySQL dapat diperoleh dan dimodifikasi.

Berikut beberapa contoh penggunaan MYSQL pada Gambar 2.14 adalah tampilan untuk membuat database dengan cara memasukan nama database yang ingin kita buat pada kolom "Ciptakan database baru" kemudian tekan tombol "Ciptakan" berikut adalah script SQLnya adalah: `CREATE DATABASE `ujian` ;`



Gambar 2.14 Tampilan hasil membuat *database*

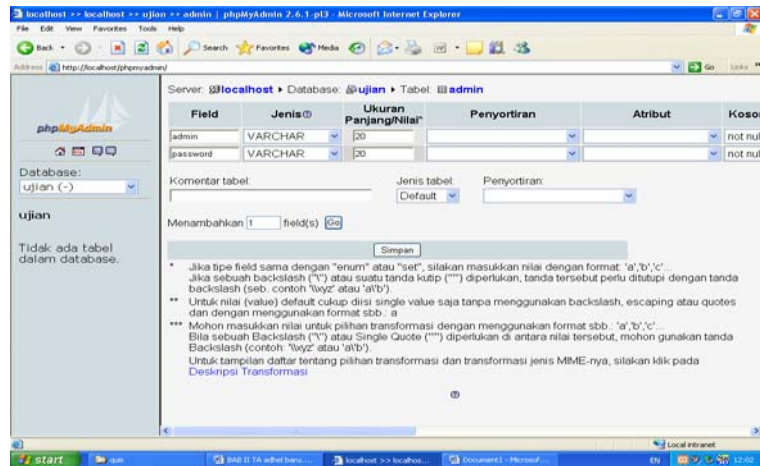
Pada gambar 2.15 adalah gambar proses membuat tabel pada database dengan cara memasukkan nama tabel yang ingin dibuat pada kolom nama dan isi pada kolom field berupa field yang ingin kita buat berikut ini adalah script SQLnya: `CREATE TABLE `admin`;`



Gambar 2.15 Tampilan membuat tabel pada database.

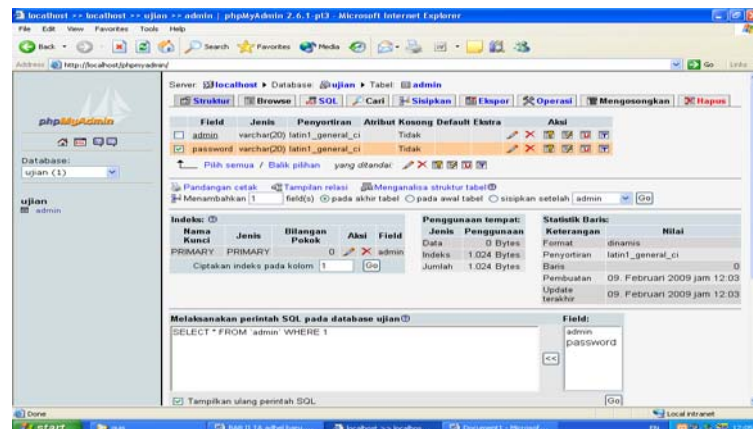
Gambar 2.16 adalah tampilan memasukkan nama field – field yang akan dibuat dan berikut ini adalah script SQLnya:

```
CREATE TABLE `admin` (
  `admin` VARCHAR( 20 ) NOT NULL ,
  `password` VARCHAR( 20 ) NOT NULL ,
  PRIMARY KEY ( `admin` ) );
```



Gambar 2.16 Tampilan memasukan nama field didalam database

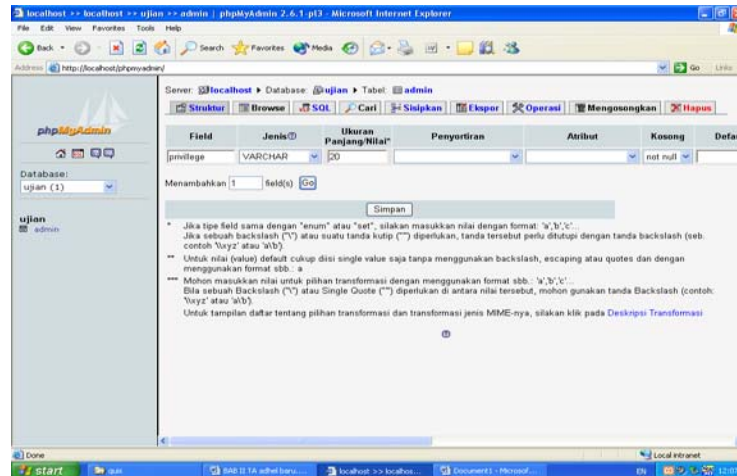
Gambar 2.17 adalah tampilan untuk menghapus field pada tabel yaitu dengan cara memberi tanda chek pada field yang ingin dihapus lalu tekan tanda silang dan berikut ini adalah script SQLnya: `ALTER TABLE `admin` DROP `password``



2.17 Tampilan menghapus field pada table

Gambar 2.18 adalah tampilan untuk menyisipkan satu field diantara field yang lainnya berikut ini adalah script SQLnya:

```
ALTER TABLE `Admin` ADD `privilege` VARCHAR( 8 ) NOT NULL AFTER  
`password`;
```



Gambar 2.18 Tampilan menyisipkan satu field diantara field lainnya.

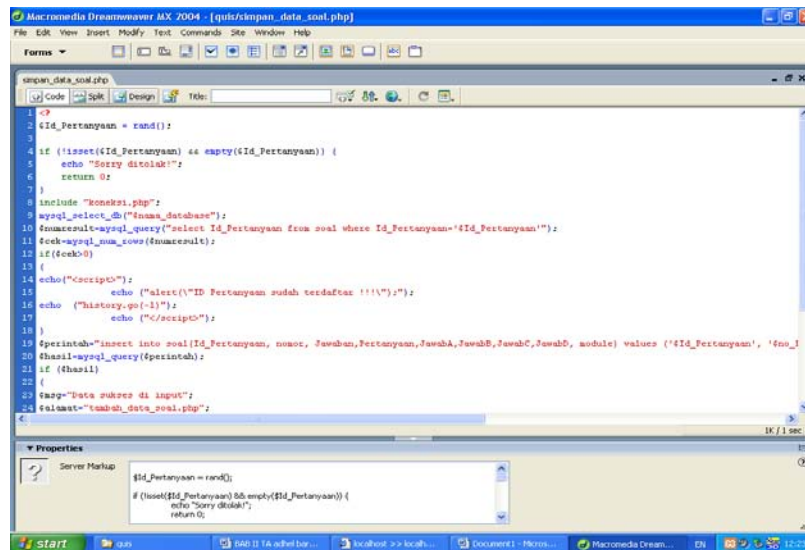
2.9 Dreamweaver

Menurut (Vita Prihatoni 2005:9) *Macromedia Dreamweaver* merupakan *software* yang dikenal sebagai *software web authoring tool* yaitu *software* untuk desain dan *layout* halaman web. *Macromedia dreamweaver* mulai menunjukkan daya tariknya sejak versi 3 lalu versi 4 dan saat ini memiliki versi terbaru yang disebut *Macromedia MX*.

Versi terbaru *dreamweaver* ini memiliki kemampuan bukan hanya sebagai *software* untuk desain web saja tetapi juga untuk *editing* kode serta pembuatan aplikasi web dengan menggunakan berbagai bahasa pemrograman web seperti: Jsp, PHP, ASP, dan Cold Fusion. Pada saat ini *Dreamweaver* merupakan *Software* utama yang digunakan oleh *web desaigner* maupun *web programmer* dalam membangun suatu situs web.

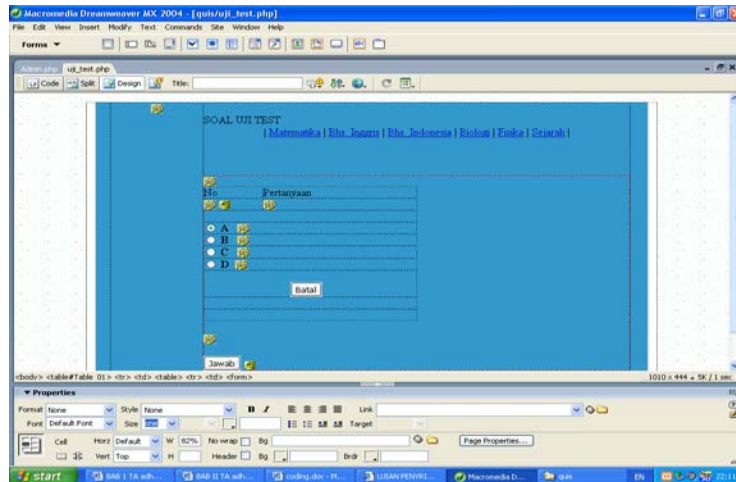
Hal ini disebabkan ruang kerja, fasilitas dan kemampuan dreamweaver yang mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun suatu situs web. Selain fasilitas untuk desain layout halaman web maupun aplikasi *database dreamweaver* juga dilengkapi dengan fasilitas untuk manajemen situs yang cukup lengkap. Berikut ini adalah *layout – layout* pada *dreamweaver* :

Gambar 2.19 merupakan contoh tampilan jendela dokumen *code* pada dreamweaver untuk menampilkan kode atau script HTML.



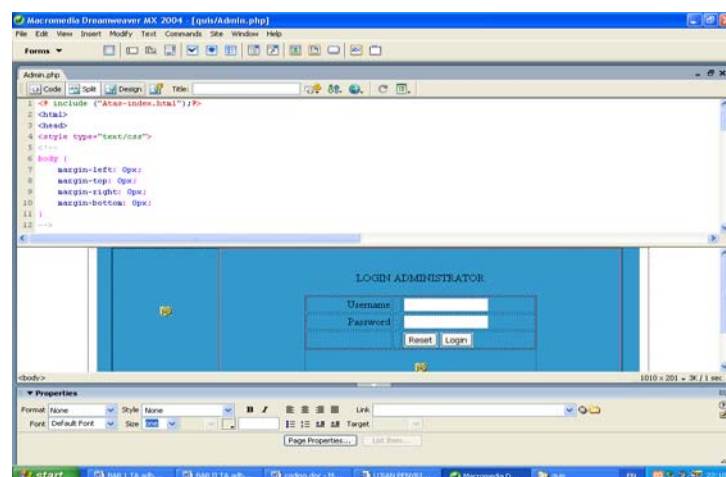
Gambar 2.19 Tampilan pada dreamweaver dengan mode tampilan Code.

Pada gambar 2.20 merupakan contoh jendela dokumen *design* pada dreamweaver dengan berbagai menu yang dapat dipergunakan untuk mendesain tampilan halaman – halaman pada web.



Gambar 2.20 Tampilan pada dreamweaver dengan mode Design.

Gambar 2.21 merupakan contoh jendela dokumen *split* pada dreamweaver. Mode tampilan ini dapat digunakan jika kita ingin menggunakan jendela dokumen desain untuk merancang tampilan halaman web, dan juga menggunakan jendela dokumen kode untuk melakukan kustomisasi dreamweaver jadi kita dapat melihat tampilan jendela dokumen desain dan jendela dokumen kode pada saat yang bersamaan.



Gambar 2.21 Tampilan pada dreamweaver dengan mode Split.

BAB III

ANALISIS SISTEM DAN PERANCANGAN

3.1 Identifikasi Permasalahan

Ujian penyaringan masuk untuk siswa sekolah menengah umum merupakan salah satu hal terpenting dalam penerimaan siswa baru, dimana dengan adanya ujian penyaringan ini pihak sekolah kurang lebihnya dapat mengetahui kemampuan dari calon-calon siswa yang mendaftar pada sekolah tersebut. Biasanya ujian penyaringan ini diadakan pada waktu yang telah disediakan oleh pihak sekolah tetapi proses pelaksanaannya masih berjalan secara manual yaitu dengan menggunakan kertas soal seperti ujian pada umumnya. Aplikasi sistem berbasis web ini mempunyai manfaat sebagai berikut:

1. Memberikan kemudahan bagi calon siswa yang ingin masuk ke sekolah yang dituju dalam melakukan ujian penyaringan.
2. Mempermudah pihak lembaga pendidikan dalam meningkatkan kinerja para staf guru dalam pengolahan data, dan hasil kerja sistem khususnya dalam bidang ujian penyaringan masuk.

Pada ujian penyaringan masuk yang masih secara manual pada saat ini masih terdapat banyak kekurangan seperti:

1. Banyak waktu yang terbuang karena mempersiapkan kertas soal dari mulai membuatnya sampai mencetak soal ujian tersebut.
2. Membutuhkan banyak waktu untuk memeriksa hasil ujian.
3. Dapat terjadi kesalahan penilaian pada ujian.

Dalam membuat aplikasi ujian penyaringan berbasis web ini penulis juga menganalisa apa saja yang dibutuhkan dalam pembuatan aplikasi tersebut, dan analisa yang dibutuhkan diantaranya sebagai berikut:

1. Dalam aplikasi ini dibutuhkan soal-soal yang dapat dikerjakan langsung pada sistem ini dan siswa juga dapat menjawabnya serta mengetahui nilai atau hasil yang didapatnya.
2. Dalam aplikasi ini dibutuhkan pengelompokan soal-soal kedalam suatu modul yang bertujuan agar soal dapat dimasukan berdasarkan modulnya sehingga siswa akan lebih mudah untuk memilih materi soal yang akan dikerjakan lebih dulu.
3. Dalam aplikasi ini bagi peserta ujian yang ingin mengikuti ujian penyaringan harus mendaftar ulang terlebih dahulu untuk dapat masuk kehalaman peserta.

3.2 Evaluasi dan Penyelesaian Masalah

Dari permasalahan tersebut didapatkan solusi untuk menyelesaikannya. Penjabaran penyelesaian dijabarkan di bawah ini:

1. Ujian dapat di lakukan pada media komputerisasi dengan menggunakan web, sehingga untuk calon-calon siswa yang akan mengikuti ujian penyaringan tidak perlu membawa alat-alat tulis.
2. Karena media yang digunakan adalah komputer maka lebih efektif dalam memaksimalkan waktu dalam pelaksanaan ujian tersebut.
3. Hak akses calon siswa pada aplikasi ini di batasi dengan memberikan *username* dan *password* untuk dapat ikut dalam ujian penyaringan tersebut. Jika *username* dan *password* yang dimasukan sama dengan data pada *database* maka calon siswa itu memiliki akses untuk ikut dalam ujian penyaringan.

Dalam melakukan pengujian harus ada kebutuhan sistem yang didasarkan pada sudut pandang pemakai, yaitu:

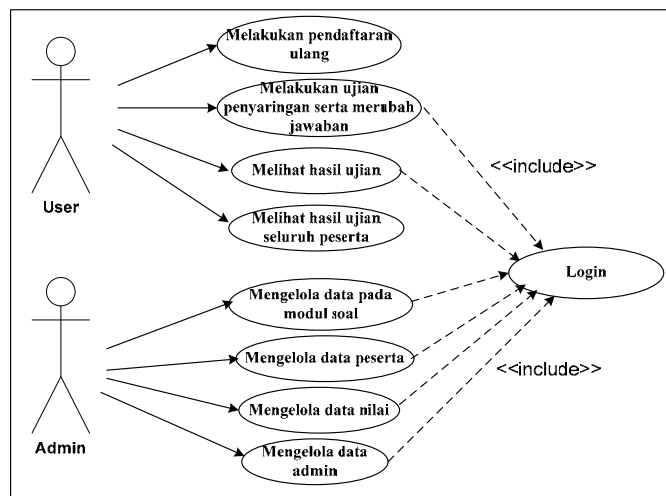
1. Terdapat bantuan penggunaan.
2. Nama link sesuai dengan proses yang dikerjakan.
3. Hasil dari pengiriman informasi dapat dilihat langsung oleh user.
4. Tampilan setiap objek pada setiap halaman harus sesuai dengan fungsi halamannya.

3.3 Perancangan Aplikasi

Berdasarkan hasil analisa, penulis membuat sebuah aplikasi menggunakan php dan mysql untuk membuat aplikasi tersebut. Agar aplikasi ini mudah dipahami penulis membuat Diagram *Use Case* untuk menunjukkan fungsionalitas suatu sistem. *Collaboration* Diagram untuk menggambarkan obyek – obyek dan hubungannya satu dengan yang lain. Diagram aktifitas untuk mengurutkan aktifitas dalam suatu proses.

3.3.1 Use Case Diagram

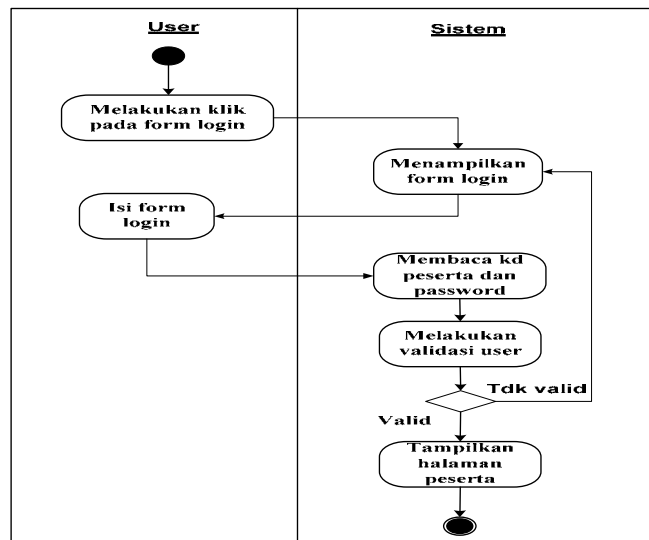
Use case diagram mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar, dan berikut ini adalah use case diagram pada aplikasi ini yang dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram *Use Case*

3.3.2 Pemodelan dengan Activity Diagram Dan Spesifikasi Naratif.

Activity diagram adalah tehnik untuk mendeskripsikan logika *procedural*, proses bisnis dan aliran kerja dalam banyak kasus. Activity Diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku *parallel* sedangkan *flowchart* tidak bisa. Berikut ini adalah *activity* diagram pada aplikasi ini yang dapat dilihat pada gambar berikut ini:

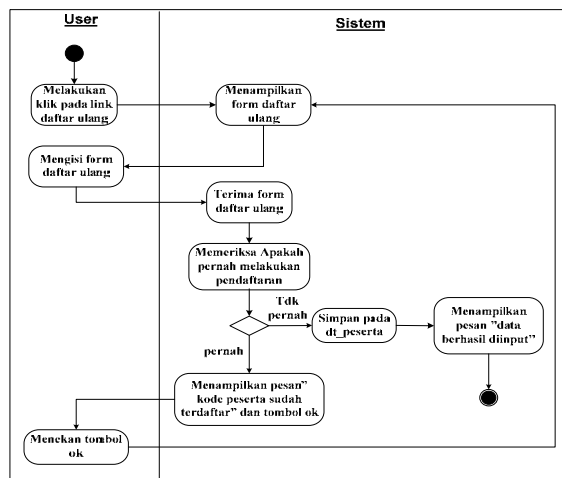


Gambar 3.2 Activity Diagram Untuk login peserta

Tabel 3.1 Spesifikasi naratif untuk Login Peserta.

Use Case	:	Login Peserta
Deskripsi Singkat	:	Use Case ini memungkinkan user untuk dapat mengakses halaman peserta.
Actor	:	peserta ujian
Pre Condition	:	Tidak ada.

Aliran Utama	:	<ol style="list-style-type: none"> 1. Peserta melakukan klik terhadap link login peserta pada halaman menu utama. 2. Kemudian sistem akan menampilkan form login yang terdiri dari field kd_peserta dan password. 3. Peserta mengisi field kd_peserta dan password tersebut. 4. Setelah itu sistem akan melakukan validasi apakah benar kd_peserta dan pasword yang dimasukan valid datanya, jika ya maka sistem akan menampilkan halaman peserta. Jika tidak maka sistem akan menampilkan form login kembali.
Post Condition	:	Halaman peserta ditampilkan.

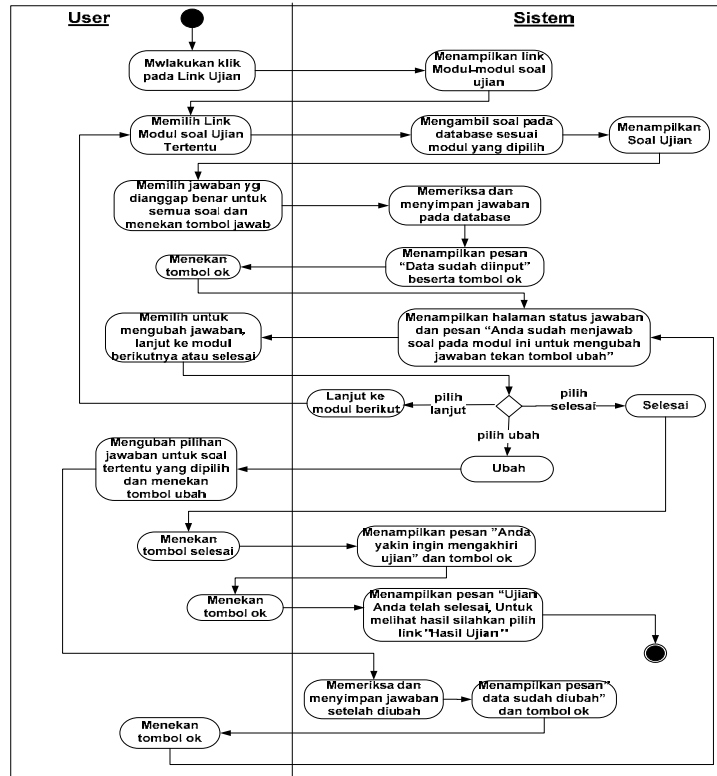


Gambar 3.3 Activity Diagram Untuk Daftar ulang peserta ujian

Tabel 3.2 Spesifikasi naratif untuk Use Case Daftar ulang peserta ujian.

Use Case	:	Daftar ulang peserta ujian
Deskripsi Singkat	:	Use Case ini memungkinkan user untuk melakukan pendaftaran ulang.
Actor	:	peserta ujian

<i>Pre Condition</i>	:	Tidak ada.
Aliran Utama	:	<ol style="list-style-type: none"> 1. Peserta melakukan klik terhadap link Daftar pada halaman menu utama. 2. Kemudian sistem akan menampilkan form daftar yang terdiri dari field kd_peserta, Nama, Username, Password, J_kelamin, dan Email. 3. Peserta mengisi form daftar ulang dengan lengkap data yang diminta. 4. Setelah itu sistem akan menerima form daftar ulang dan
Aliran Utama	:	<p>memeriksa pada <i>database</i> apakah sudah pernah melakukan pendaftaran ulang sebelumnya, jika ya sudah pernah melakukan pendaftaran ulang maka sistem akan menampilkan pesan "Kode peserta sudah terdaftar" dan juga tombol ok kemudian peserta menekan tombol ok dan sistem akan menampilkan kembali form daftar ulang untuk diisi kembali dengan data yang benar. Jika tidak pernah melakukan daftar ulang maka sistem akan menyimpan pada <i>database</i> dan menampilkan pesan "Data Berhasil Diinput".</p>
Post Condition	:	Data peserta tersimpan pada <i>database</i> dt_peserta.



Gambar 3.4 Activity Diagram Untuk Melakukan Ujian penyingaran.

Tabel 3.3 Spesifikasi naratif untuk Use Case Melakukan Ujian penyingaran.

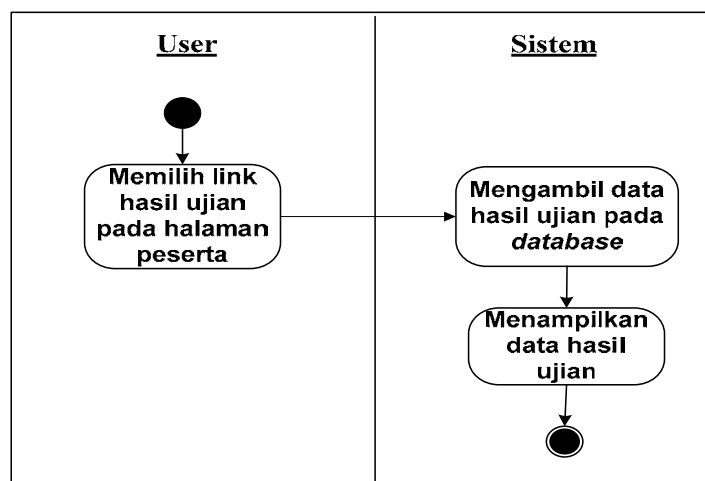
Use Case	:	Melakukan Ujian penyingaran
Deskripsi Singkat	:	Use Case ini memungkinkan user untuk melakukan ujian penyingaran pada halaman peserta.
Actor	:	peserta ujian
Pre Condition	:	Peserta sudah login
Aliran Utama	:	<ol style="list-style-type: none"> 1. Peserta melakukan klik terhadap link ujian yang ada pada halaman peserta. 2. Kemudian sistem akan menampilkan modul soal yang dapat dipilih peserta.

(Lanjutan Tabel 3.3)

	<ol style="list-style-type: none">3. Peserta melakukan klik terhadap salah satu link modul soal ujian tertentu.4. Lalu sistem akan mengambil soal pada database sesuai modul soal ujian yang dipilih .5. dan sistem akan menampilkan semua soal ujian pada peserta sesuai dengan modul yang dipilih peserta.6. Peserta memilih satu pilihan jawaban yang dianggap benar yaitu A, B, C, atau D, untuk semua soal. Jika sudah mengisi jawaban pada modul soal tersebut maka peserta harus menekan tombol jawab.7. Dan sistem akan memeriksa dan menyimpan jawaban pada <i>database</i> setelah itu sistem akan menampilkan pesan bahwa "Data sudah diinput" beserta tombol ok. lalu peserta melakukan klik terhadap tombol ok.8. Kemudian sistem akan menampilkan kembali halaman soal dengan status jawaban yang sudah diisi dan juga pesan "Anda sudah menjawab soal pada modul ini untuk mengubah jawaban tekan tombol ubah" pada halaman bagian bawah.9. Peserta dapat memilih pilihan untuk mengubah jawaban, lanjut ke modul berikutnya atau selesai.10. Jika peserta memilih lanjut ke modul berikut maka akan kembali kelangkah yang sama seperti aliran utama pada no 3.
--	--

(Lanjutan Tabel 3.3)

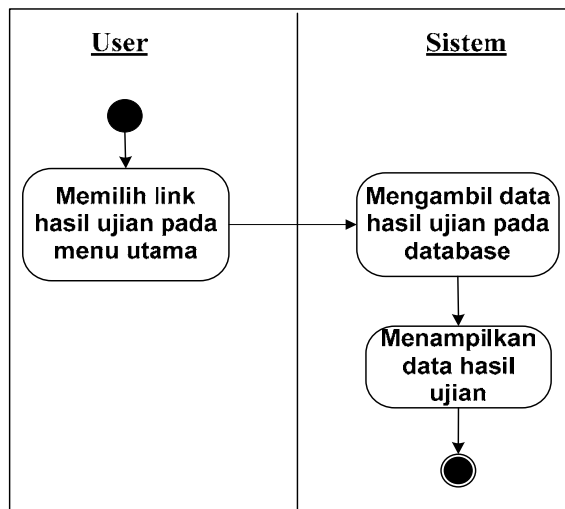
		<p>11. Jika peserta memilih mengubah jawaban, peserta dapat memilih ulang pilihan jawaban untuk soal tertentu dan menekan tombol ubah.</p> <p>12. Kemudian sistem akan memeriksa dan menyimpan jawaban kembali setelah diubah. Dan menampilkan pesan "Data sudah diubah" beserta tombol ok.</p> <p>13. Peserta menekan tombol ok. dan kembali kelangkah 8.</p> <p>Jika peserta memilih tombol selesai maka sistem akan menampilkan pesan "Anda telah mengakhiri ujian" beserta tombol ok. Kemudian peserta menekan tombol ok tersebut, lalu sistem akan menampilkan pesan "Ujian Anda telah selesai, untuk melihat hasil silahkan pilih link "Hasil Ujian".</p>
<i>Post Condition</i>	:	Jawaban peserta tersimpan pada <i>database</i> .



Gambar 3.5 Activity Diagram melihat data hasil ujian peserta

Tabel 3.4 Spesifikasi naratif untuk *Use Case* melihat hasil ujian peserta.

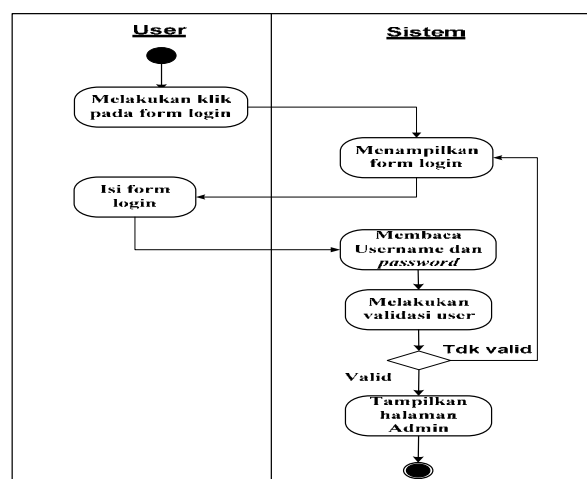
Use Case	:	Melihat hasil ujian peserta
Deskripsi Singkat	:	<i>Use Case</i> ini memungkinkan user untuk melihat hasil ujian dengan cara mengklik <i>link</i> hasil ujian pada halaman ujian.
Actor	:	Peserta ujian
<i>Pre Condition</i>	:	Seluruh modul soal telah dikerjakan dan tombol selesai telah ditekan.
Aliran Utama	:	<ol style="list-style-type: none"> 1. Peserta Memilih link hasil ujian. 2. Sistem Mengambil data hasil hasil ujian pada <i>database</i>. 3. Dan kemudian sistem menampilkan halaman hasil ujian peserta.
<i>Post Condition</i>	:	Peserta dapat mengetahui hasil ujiannya.



Gambar 3.6 *Activity Diagram* Melihat hasil ujian seluruh peserta

Tabel 3.5 Spesifikasi naratif untuk *Use Case* Melihat hasil ujian seluruh peserta.

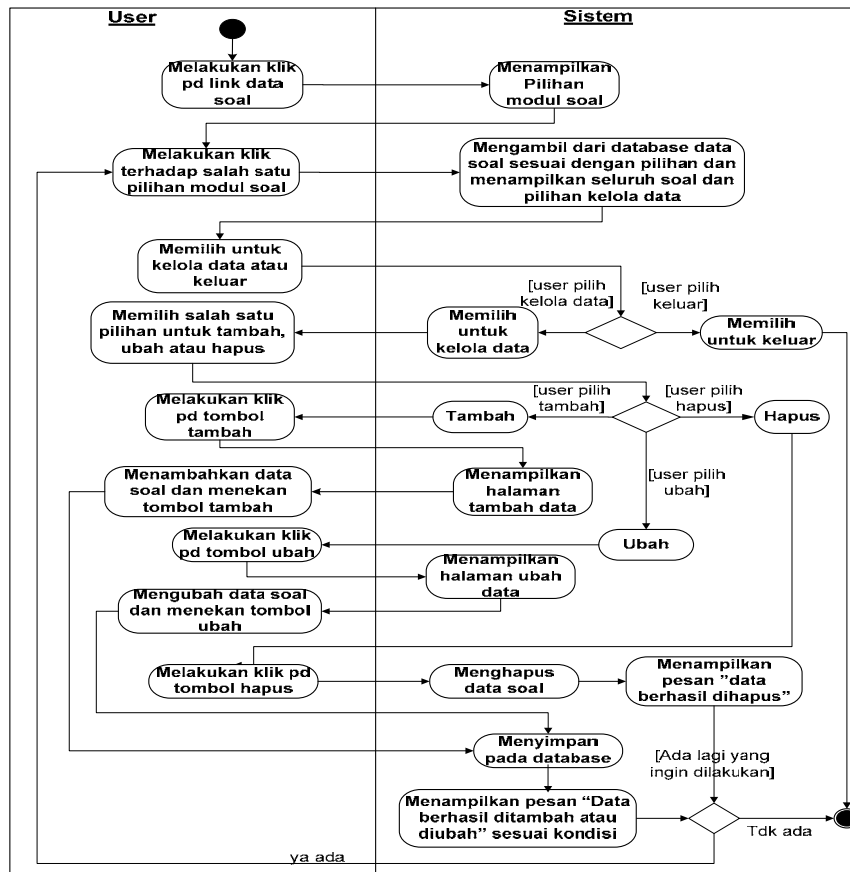
Use Case	:	Melihat hasil ujian seluruh peserta.
Deskripsi Singkat	:	<i>Use Case</i> ini memungkinkan <i>user</i> untuk melihat hasil ujian seluruh peserta dengan cara mengklik link hasil ujian pada halaman menu utama.
Actor	:	peserta ujian
<i>Pre Condition</i>	:	Semua soal telah dikerjakan dan tombol selesai telah ditekan
Aliran Utama	:	<ol style="list-style-type: none"> 1. Peserta melakukan klik pada link hasil ujian pada halaman hasil ujian pada menu utama. 2. Sistem mengambil data hasil ujian seluruh peserta pada <i>database</i>. 3. Setelah itu sistem menampilkan halaman hasil ujian seluruh peserta yang telah mengikuti ujian.
<i>Post Condition</i>	:	Hasil ujian peserta diketahui.



Gambar 3.7 Activity Diagram Untuk login Administrator

Tabel 3.6 Spesifikasi naratif untuk *Use Case* Login Administrator.

Use Case	:	Login Administrator
Deskripsi Singkat	:	<i>Use Case</i> ini memungkinkan user untuk mengakses halaman administrator.
Actor	:	Administrator
<i>Pre Condition</i>	:	Tidak ada
Aliran Utama	:	<ol style="list-style-type: none">9. Administrator melakukan klik terhadap login admin pada menu utama.10. Kemudian sistem akan menampilkan form login yang terdiri dari <i>field</i> username dan password.11. Setelah itu admin mengisi form login tersebut.12. Dan sistem akan melakukan validasi apakah benar username dan password yang dimasukan terdaftar pada <i>database</i> sebagai administrator, jika ya valid maka sistem akan menampilkan halaman administrator tetapi jika tidak maka sistem akan menampilkan kembali form <i>login</i> admin.
<i>Post Condition</i>	:	Halaman admin ditampilkan.



Gambar 3.8 Activity Diagram Untuk Mengelola data soal

Tabel 3.7 Spesifikasi naratif untuk Use Case Mengelola Data Modul Soal.

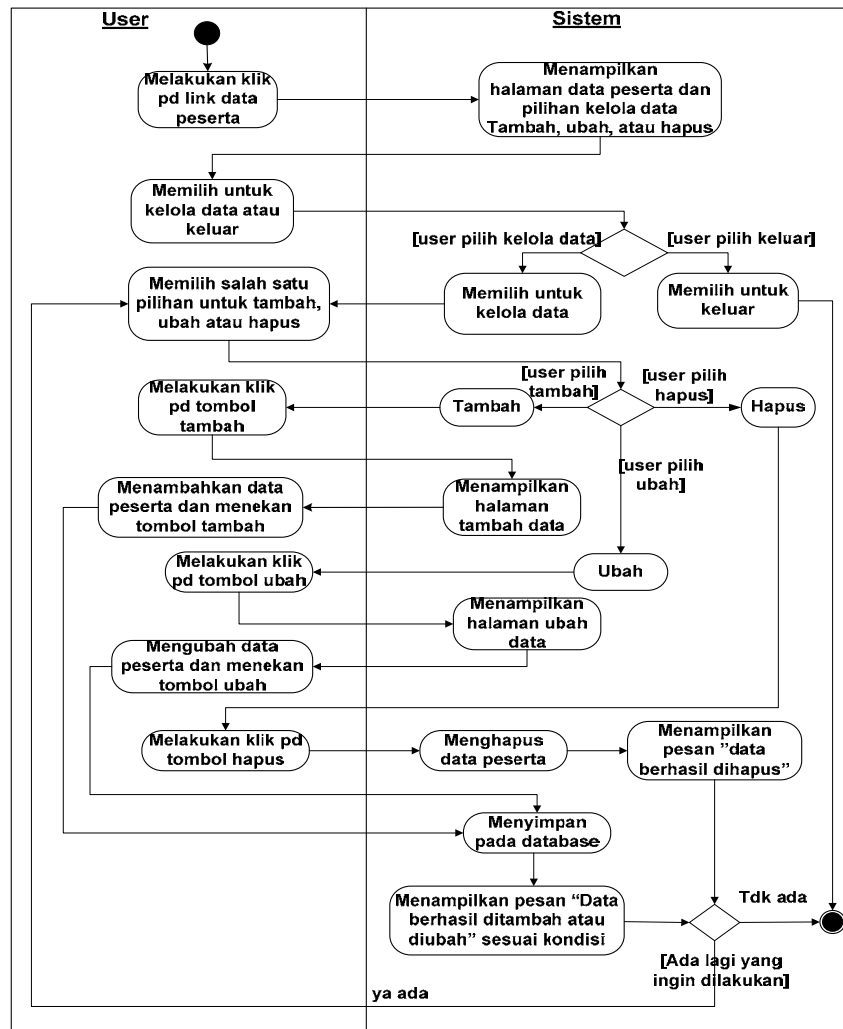
Use Case	:	Mengelola Data Modul Soal.
Deskripsi Singkat	:	Use Case ini memungkinkan admin untuk mengelola data pada halaman Modul soal.
Actor	:	Administrator
Pre Condition	:	Admin harus melakukan usecase Login terlebih dahulu
Aliran Utama	:	<ol style="list-style-type: none"> Admin melakukan klik terhadap link Data soal. Kemudian sistem akan menampilkan halaman modul soal yang dapat dikelola datanya.

(Lanjutan Tabel 3.7)

Aliran Utama	:	<ol style="list-style-type: none">3. Admin melakukan klik terhadap salah satu pilihan modul soal tertentu yang akan dikelola.4. Dan sistem mengambil data soal pada database dan menampilkan seluruh soal. Dan juga menampilkan pilihan kelola data untuk tambah, ubah, hapus.5. Admin dapat memilih pilihan kelola data (tambah, ubah, hapus) atau keluar dari halaman modul soal dan selesai pada state akhir. Jika admin melakukan klik pada tombol tambah maka sistem akan menampilkan halaman untuk tambah data soal dan admin dapat menambahkan soal, setelah itu admin menekan tombol tambah dan sistem akan menambahkan data dan menyimpannya pada database kemudian menampilkan pesan "data berhasil ditambah".6. Jika admin melakukan klik pada tombol ubah maka sistem akan menampilkan halaman untuk ubah data soal, dan admin dapat mengubah data soal, setelah itu admin menekan tombol ubah. dan sistem akan mengubah data dan menyimpannya pada database kemudian menampilkan pesan "data berhasil diubah".7. Jika admin ingin menghapus data maka admin harus memilih data tertentu yang ingin dihapus kemudian menekan tombol tombol hapus maka sistem akan menghapus data kemudian menampilkan pesan " Data berhasil dihapus".
--------------	---	---

(Lanjutan Tabel 3.7)

	<p>jika ada hal lain yang ingin dilakukan lagi oleh admin maka admin dapat memilih kembali pilihan modul soal atau sama seperti langkah 3 jika tidak ada yang ingin dilakukan maka akan menuju ke state akhir.</p>
<i>Post Condition</i>	Data pada modul soal yang telah dikelola disimpan pada <i>database</i>



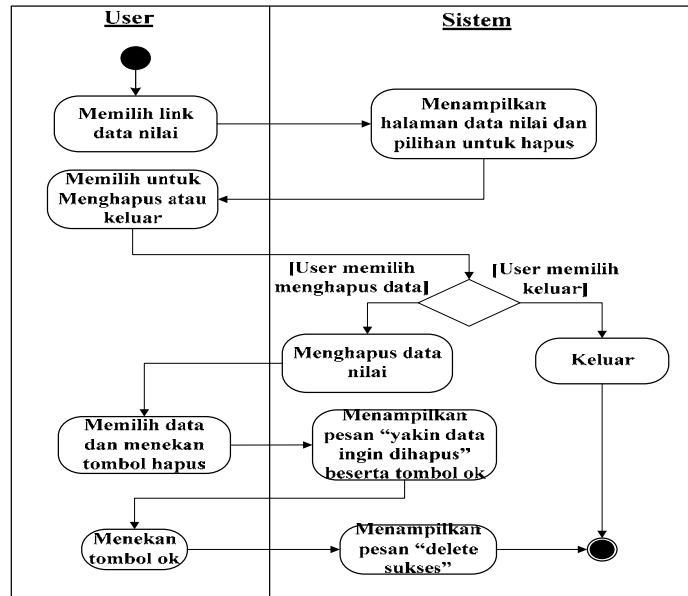
Gambar 3.9 Activity Diagram Untuk mengelola data peserta

Tabel 3.8 Spesifikasi naratif untuk *Use Case* mengelola data peserta.

Use Case	:	Mengelola data peserta.
Deskripsi Singkat	:	<i>Use Case</i> ini memungkinkan Admin untuk mengelola data peserta pada halaman administrator setelah login terlebih dahulu.
Actor	:	Administrator
<i>Pre Condition</i>	:	Admin harus melakukan usecase Login terlebih dahulu
Aliran Utama	:	<ol style="list-style-type: none"> 1. Admin melakukan klik terhadap link Data peserta. 2. Kemudian sistem akan menampilkan halaman data peserta dan pilihan kelola data. 3. Admin dapat memilih pilihan apakah itu kelola data atau keluar dari halaman data peserta tersebut. Jika admin memilih keluar maka akan selesai pada state akhir. 4. Tetapi jika admin memilih kelola data maka admin dapat memilih salah satu pilihan kelola data yang ada yaitu Tambah, Ubah atau hapus jika salah satu tombol pilihan ditekan maka sistem akan menampilkan halaman kelola data yang dipilih admin. 5. Setelah admin mengelola data pada halaman Data peserta, yaitu menambahkan, mengubah atau menghapus maka sistem akan menyimpannya pada <i>database</i> dt_peserta dan sistem akan menampilkan pesan "Data berhasil ditambah" untuk tambah data dan "Data berhasil diubah" untuk ubah data dan "Data berhasil dihapus" untuk hapus data.

(Lanjutan Tabel 3. 8)

	6. Jika ada hal lain yang ingin dilakukan admin maka kembali ke alur utama no 3.
<i>Post Action</i>	Data peserta yang telah dikelola disimpan pada <i>database</i>



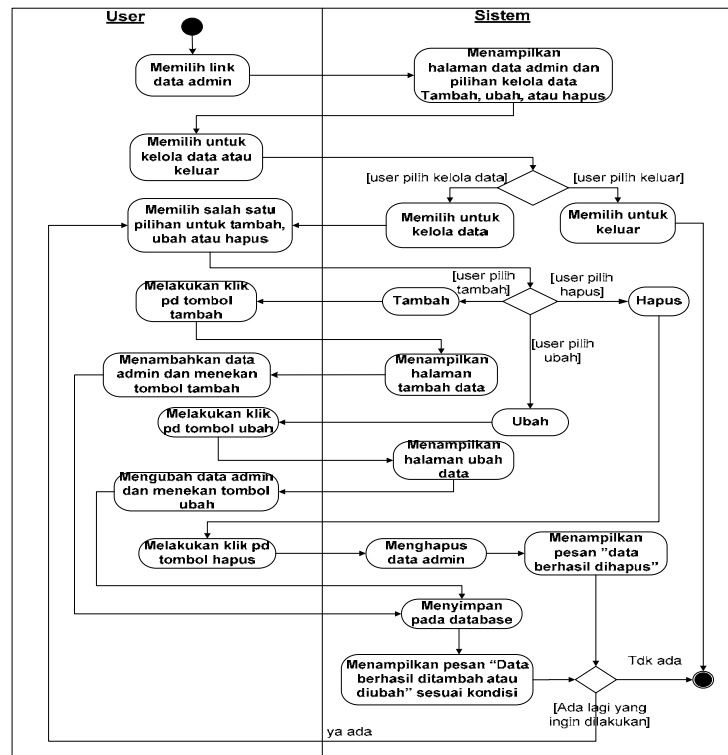
Gambar 3.10 Activity Diagram Untuk Mengelola Data Nilai

Tabel 3.9 Spesifikasi naratif untuk Use Case Mengelola Data Nilai.

Use Case	:	Mengelola Data Nilai.
Deskripsi Singkat	:	Use Case ini memungkinkan Administrator untuk mengelola data nilai peserta ujian dengan cara mengakses halaman administrator setelah login.
Actor	:	Administrator
<i>Pre Condition</i>	:	Admin harus melakukan usecase Login terlebih dahulu
Aliran Utama	:	1. Admin melakukan klik terhadap link Data nilai pada halaman administrator.

(Lanjutan Tabel 3.9)

	<ol style="list-style-type: none">2. Kemudian sistem akan menampilkan halaman data nilai dan pilihan untuk hapus data.3. Admin memilih untuk hapus data atau keluar dari halaman data nilai. Jika admin memilih untuk menghapus data nilai maka admin menekan tombol hapus yang berada disamping kanan data nilai peserta.4. Kemudian sistem akan menampilkan halaman kotak pesan "Yakin data ingin dihapus" dan juga tombol ok.5. Setelah itu admin menekan tombol ok dan sistem akan menampilkan pesan "Data berhasil dihapus" dan selesai.
<i>Post Action</i>	Data nilai yang telah dikelola disimpan pada <i>database</i>



Gambar 3.11 Activity Diagram Untuk mengelola data admin

Tabel 3.10 Spesifikasi naratif untuk *Use Case* mengelola Data Admin.

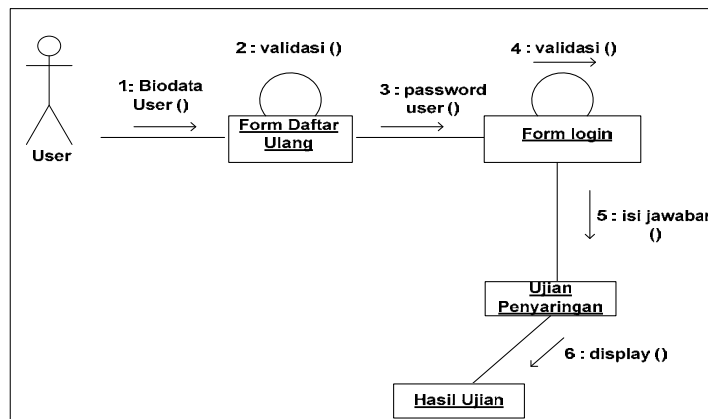
Use Case	:	mengelola data admin.
Deskripsi Singkat	:	<i>Use Case</i> ini memungkinkan Administrator untuk mengelola data administrator dengan cara login terlebih dahulu.
Actor	:	Administrator
<i>Pre Condition</i>	:	Admin harus melakukan usecase Login terlebih dahulu
Aliran Utama	:	<ol style="list-style-type: none"> 1. Admin melakukan klik terhadap link Data admin. 2. Kemudian sistem akan menampilkan halaman data admin dan pilihan kelola data. 3. Admin memilih pilihan apakah itu kelola data atau keluar dari halaman data admin tersebut. Jika admin memilih keluar maka akan selesai pada state akhir. 4. Tetapi jika admin memilih kelola data maka admin dapat memilih salah satu pilihan kelola data yang ada yaitu Tambah, Ubah atau hapus jika salah satu tombol pilihan ditekan maka sistem akan menampilkan halaman kelola data yang dipilih admin. 5. Setelah admin mengelola data pada halaman Data admin, yaitu menambahkan, mengubah atau menghapus maka sistem akan menyimpannya pada <i>database</i> admin dan sistem akan menampilkan pesan "Data berhasil ditambah" untuk tambah data dan "Data berhasil diubah" untuk ubah data dan "Data berhasil dihapus" untuk hapus data. 6. Jika ada hal lain yang ingin dilakukan admin maka

(Lanjutan Tabel 3.10)

		langkah-langkah yang akan dilalui admin persis seperti aliran utama pada no 3.
Post Action		Data admin yang telah dikelola disimpan pada database

3.3.3 Collaboration Diagram

Colaboration diagram adalah perluasan dari obyek *diagram*. Atau obyek diagram menunjukkan obyek-obyek dan hubungannya satu dengan yang lain. Berikut ini adalah *collaboration diagram* pada aplikasi ini yang dapat dilihat pada gambar dibawah ini:

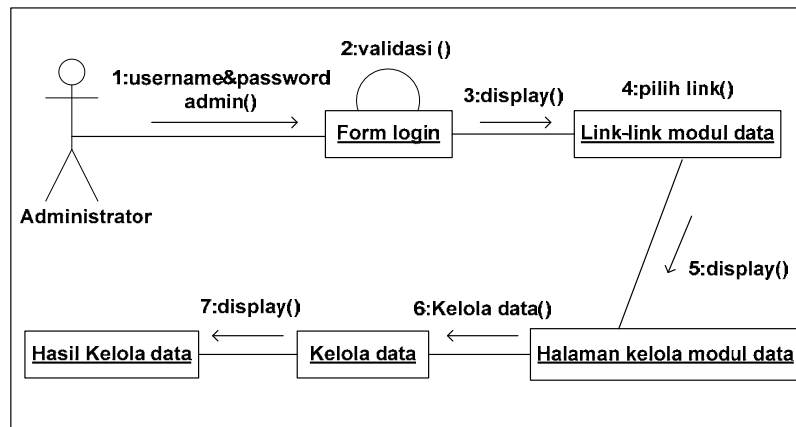


Gambar 3.12 *Colaboration Diagram* Untuk user

Penjelasannya sebagai berikut :

User atau calon siswa mengisi biodata pada *form* daftar ulang setelah itu akan divalidasi jika data yang diisi sudah lengkap maka peserta ujian sudah mempunyai kode peserta dan *password* untuk dapat masuk ke halaman ujian penyaringan dengan cara memasukan kode peserta dan password tersebut pada

form *login*, jika data valid maka akan ditampilkan halaman ujian penyaringan. Pada halaman ini peserta ujian melakukan proses penginputan jawaban dan selanjutnya data yang sudah diisi lengkap akan disimpan pada database dan kemudian peserta ujian dapat melihat hasil ujiannya secara langsung.



Gambar 3.13 Collaboration Diagram Untuk administrator

Penjelasannya sebagai berikut :

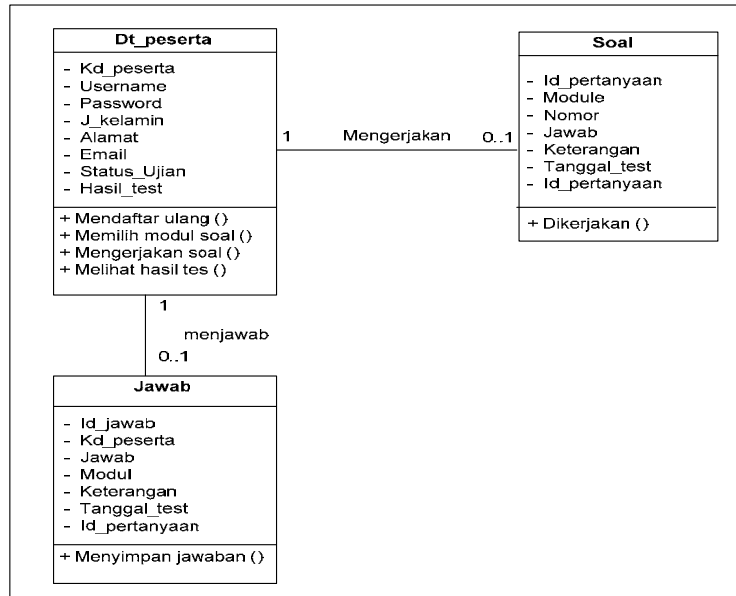
User dengan hak akses administrator mengisi *username* dan *password* pada *form login* kemudian akan divalidasi. Jika *username* dan *password* yang diisi pada *form login* tersebut valid maka akan ditampilkan link-link modul data yang dapat dipilih salah satu oleh admin untuk dikelola. Jika admin memilih salah satu link modul data pada halaman administrator maka akan ditampilkan halaman data tersebut beserta aktivitas kelola data. Pada form tersebut dapat dilakukan proses kelola data dan selanjutnya data akan disimpan pada *database* dan admin akan dapat melihat hasil data yang sudah dikelola.

3.4 Perancangan Basis Data Pada Aplikasi ujian penyaringan masuk

Pada bagian ini akan dijelaskan tentang perancangan aplikasi ujian penyaringan masuk berbasis web yang meliputi perancangan aplikasi, perancangan basisdata yang dijelaskan dalam pemodelan diagram kelas, dan perancangan tampilan interface. Tujuan dari rancangan basis data adalah diperolehnya basis data yang efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah untuk memanipulasi data serta bebas dari redundansi data.

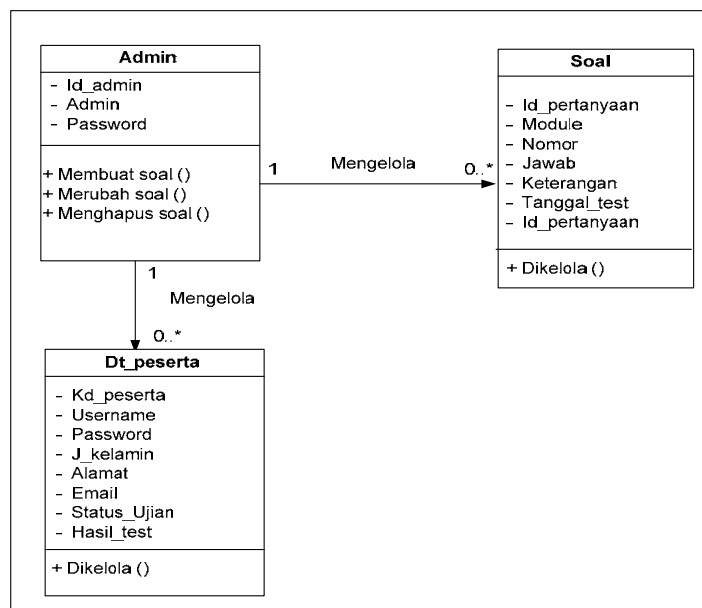
3.4.1 Pemodelan *Class Diagram*

Diagram kelas ini digunakan untuk menampilkan kelas-kelas atau paket-paket didalam sebuah sistem dan relasi antara mereka, yang memberikan gambaran secara statis. Kelas adalah sebuah kategori yang membungkus informasi dan perilaku. Jadi diagram kelas merupakan alat perancangan terbaik untuk tim pengembang perangkat lunak. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum menuliskan kode program, membantu untuk memastikan bahwa sistem adalah rancangan terbaik (Sholih, 2006). Dalam bagian ini akan dijelaskan perancangan sistem aplikasi ujian penyaringan dengan diagram kelas yang terlihat pada gambar 3.14



Gambar 3.14 *Class diagram* untuk peserta

Pada Gambar 3.14 dijelaskan bahwa peserta dapat mengerjakan soal, serta siswa dapat memilih modul soal dan melihat hasil tes, hal ini terlihat pada diagram kelas yang ditunjukkan oleh tabel data yang saling berhubungan antara peserta dengan tabel jawab dan soal.



Gambar 3.15 *Class diagram* untuk admin

Pada diagram kelas admin dijelaskan bahwa admin dapat mengelola banyak soal, selain itu admin juga dapat mengelola data siswa, hal ini terlihat pada pada tabel guru yang saling berhubungan dengan tabel soal, dan data siswa.

3.4.2 Perancangan basis data secara konseptual

Spesifikasi basis data yang akan memberikan penjelasan secara detail tentang masing-masing basis data yang digunakan dalam sistem ini dimana pada aplikasi ini menggunakan 1 buah *database* dan 5 tabel. Berikut penjabaran dari masing-masing database dan tabel:

1. Tabel admin.

Entitas admin memiliki 3 buah *attribute* yang di gunakan yaitu : *id_admin*, *admin* dan *password*.

Nama file : Admin

Isi : Digunakan untuk menyimpan data kode akses administrator.

Primary Key : *Id_admin*

Tabel 3.11 Tabel admin.

Field	Jenis	Ukuran	Keterangan
<u>Id_admin</u>	integer	11	Identitas unik administrator
admin	Char	20	Nama administrator
password	Char	10	Kode password administrator

Tabel ini dibuat dengan definisi sebagai berikut:

```
CREATE TABLE admin (
  id_admin int(11) NOT NULL default '0',
  Admin char(20) NOT NULL default '',
  `Password` char(10) default NULL,
  PRIMARY KEY (id_admin)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

2. Tabel dt_peserta.

Pada Entitas data peserta terdapat 9 buah attribute yaitu: kd_peserta, nama, username, password, j_kelamin, alamat, email, status_ujian, hasil_tes.

Nama file : Data peserta

Isi : Di gunakan untuk menyimpan biodata dan kode akses peserta ujian.

Primary Key : Kd_peserta.

Tabel 3.12 Tabel dt_peserta.

Field	Jenis	Ukuran	Keterangan
<u>Kd_peserta</u>	Integer	10	Kode ujian untuk peserta
name	Varchar	100	Nama lengkap peserta ujian
Username	Varchar	20	Username peserta ujian
Password	Varchar	10	Kode untuk validasi peserta ujian
J_kelamin	Varchar	10	Jenis kelamin peserta ujian
alamat	varchar	200	Alamat peserta ujian
Email	Varchar	100	Email peserta ujian
Hasil_ujian	Integer	10	Untuk menyimpan nilai ujian
Status_ujian	Varchar	50	Untuk menyimpan status selesai atau belumnya peserta mengikuti ujian

Tabel ini dibuat dengan definisi sebagai berikut:

```
CREATE TABLE dt_peserta (  
  Kd_Peserta varchar(10) NOT NULL default '',  
  name varchar(100) NOT NULL default '',  
  Username varchar(20) default NULL,  
  `Password` varchar(10) default NULL,  
  J_Kelamin varchar(10) default NULL,  
  Alamat varchar(200) default NULL,  
  email varchar(100) default NULL,  
  Hasil_Ujian int(10) NOT NULL default '0',  
  status_ujian varchar(50) NOT NULL default '',  
  PRIMARY KEY (Kd_Peserta)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

3. Tabel soal

Pada Entitas soal memiliki 9 buah atribut yaitu : id_pertanyaan, nomor, modul, pertanyaan, jawabA, jawabB, jawabC, jawab D, jawaban.

Nama file : Soal

Isi : Digunakan untuk menyimpan data soal dan jawabannya.

Primary key : id_pertanyaan

Tabel 3.13 Tabel soal.

Field	Jenis	Ukuran	Keterangan
<u>Id_pertanyaan</u>	Char	10	Identitas soal
nomor	Int	11	Untuk nomor soal
Modul	Varchar	100	Modul-modul soal
Pertanyaan	Varchar	255	Soal ujian
jawabA	Varchar	255	Pilihan jawaban A
JawabB	Varchar	255	Pilihan jawaban B
JawabC	Varchar	255	Pilihan jawaban C
JawabD	Varchar	255	Pilihan jawaban D
Jawaban	Varchar	200	Jawaban yang benar

Tabel ini dibuat dengan definisi sebagai berikut:

```
CREATE TABLE soal (  
  Id_Pertanyaan char(10) NOT NULL default '',  
  nomor int(11) NOT NULL default '0',  
  module varchar(100) NOT NULL default '',  
  Pertanyaan varchar(255) default NULL,  
  JawabA varchar(255) default NULL,  
  JawabB varchar(255) default NULL,  
  JawabC varchar(255) default NULL,  
  JawabD varchar(255) default NULL,  
  Jawaban varchar(200) default NULL,  
  PRIMARY KEY (Id_Pertanyaan)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

4. Tabel jawab.

Pada Entitas jawab memiliki 7 buah atribut yaitu : id_jawab, kd_peserta, id_pertanyaan, jawab, module, keterangan, tanggal test.

Nama file : Jawab

Isi : Digunakan untuk menyimpan data hasil jawaban peserta ujian.

Primary key : Id_jawab.

Tabel 3.13 Tabel jawab.

Field	Jenis	Ukuran	Keterangan
<u>Id_jawab</u>	integer	10	Identitas untuk jawaban
Kd_peserta	varchar	10	Kode ujian peserta (<i>Primay key</i>)
Id_pertanyaan	varchar	10	Kode soal ujian
Jawab	varchar	10	Jawaban soal ujian
module	varchar	100	Ujian
keterangan	varchar	100	Keterangan ujian
Tanggal test	date		Tanggal ujian dilaksanakan

Tabel ini dibuat dengan definisi sebagai berikut:

```
CREATE TABLE jawab (  
  Id_jawab int (10) Not Null default',  
  Kd_Peserta varchar(10) NOT NULL default '',  
  Id_Pertanyaan varchar(10) default NULL,  
  jawab varchar(10) default NULL,  
  module varchar(50) NOT NULL default '',  
  keterangan varchar(100) default NULL,  
  tgl_test date default NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

3.5 Perancangan Aplikasi Ujian penyaringan berbasis web

Rancangan layar dari ini dibuat untuk menampilkan informasi dan memudahkan dalam pencarian untuk melakukan perubahan-perubahan yang diperlukan. Ada empat macam rancangan layar dalam pembuatan aplikasi ini, yaitu: administrator, peserta, daftar ulang dan hasil ujian seluruh peserta:

The screenshot shows the main page of the application. At the top, the title 'APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB' is centered. Below the title is a navigation bar with links for 'Home', 'Admin', 'Peserta', and 'Hasil Ujian'. The main content area features a 'Menu' box on the left with options: 'Login Peserta', 'Login Admin', 'Daftar', and 'Hasil Ujian'. To the right of the menu, the text reads: 'SELAMAT DATANG' followed by a paragraph: 'Perancangan Aplikasi Ujian ini dibuat khusus bagi calon siswa Sekolah Menengah Umum yang ingin masuk kesekolah yang dituju. Aplikasi ini sangat memudahkan calon siswa Sekolah Menengah Umum sehingga tidak perlu perlengkapan yang berarti seperti alat tulis dan sebagainya. Aplikasi ini memiliki tingkat kekuratan yang tinggi dalam hal penilaian dan juga hasil ujian dapat diketahui secara langsung'.

Gambar 3.16 Rancangan Layar Halaman Utama Program

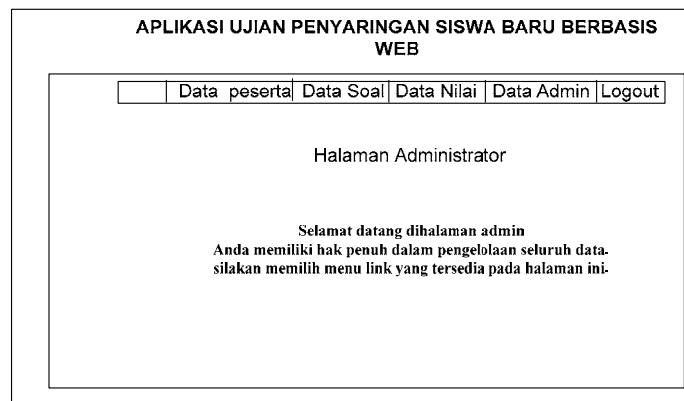
Gambar 3.16 adalah halaman pembuka yang akan muncul setelah *user* mengakses aplikasi ini.

3.5.1 Perancangan layar Halaman Administrator

The screenshot shows the 'Login Administrator' page. At the top, the title 'APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB' is centered. Below the title is a navigation bar with links for 'Home', 'Admin', 'Peserta', and 'Hasil Ujian'. The main content area features a 'Login Administrator' section. On the left, there is a 'Menu' box with options: 'Login admin', 'Login peserta', 'Daftar', and 'Hasil Ujian'. To the right of the menu, the text reads: 'Login Administrator' followed by two input fields: 'Kd peserta' and 'Password'. Below the input fields are two buttons: 'Reset' and 'Login'.

Gambar 3.17 Rancangan layar Login Administrator

Pada gambar 3.17 adalah rancangan layar *login* administrator yang digunakan oleh *user* yang mempunyai hak akses sebagai administrator saja. Setelah *user* menginput *username* dan *password* kemudian sistem akan melakukan validasi pada *database* apakah *user* tersebut administrator atau bukan. Jika *user* adalah administrator maka *user* berhak mengakses halaman ini. Tapi jika *user* bukan administrator maka dia tidak berhak untuk mengakses halaman administrator ini lalu sistem akan menampilkan kembali halaman login untuk dapat diisi kembali dengan *username* dan *password* nya dengan benar.



Gambar 3.18 Rancangan layar pembuka pada halaman administrator

Gambar 3.18 adalah rancangan layar halaman pembuka yang akan muncul setelah administrator login dan memasukan *username* serta *password*nya. Setelah halaman pembuka muncul administrator akan dihadapkan pada rancangan layar untuk link-link yang bisa dipilih untuk kelola data. Dimana administrator dapat memilih halaman yang akan dikelola datanya dan *link-link* perancangan layarnya yaitu: perancangan layar halaman data peserta, rancangan layar halaman data soal, rancangan layar data nilai dan perancangan layar data administrator, Lihat gambar-gambar berikut ini:

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Data Peserta

Kode peserta	Kode peserta	User name	Password	Jenis Kelamin	Alamat	Email	Action		
							Tambah	Ubah	Hapus

Gambar 3.19 Rancangan layar halaman data peserta

Gambar 3.19 adalah perancangan layar halaman data peserta pada halaman tersebut admin akan dapat mengelola data peserta seperti menambah, mengubah, dan menghapus data jika diperlukan.

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Matematika|Bhs Ing|Bhs Ind|Biologi|Fisika|Sejarah

Gambar 3.20 Rancangan layar halaman modul soal

Gambar 3.20 adalah rancangan layar modul soal, pada halaman tersebut admin akan dapat merubah, menambah serta menghapus soal-soal ujian sesuai dengan kepentingannya.



Gambar 3.21 Rancangan layar halaman data soal pada halaman admin

Gambar 3.21 adalah perancangan layar halaman soal yang dapat dilihat setelah user admin melakukan klik pada salah satu link modul soal, dan pada halaman ini admin akan dapat mengelola soal-soal ujian seperti menambah soal, mengubah soal, dan juga menghapus soal jika diperlukan.



Gambar 3.22 Rancangan layar data nilai pada halaman admin

Gambar 3.22 adalah perancangan layar halaman data nilai peserta. Pada halaman ini admin akan dapat menghapus data nilai jika memang diperlukan.

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Home | Admin | Peserta | Hasil Test

Data Admin

Admin	Password	Action		
		Tambah	Ubah	Hapus

Gambar 3.23 Rancangan layar data admin pada halaman administrator

Gambar 3.23 adalah perancangan layar halaman data administrator. Pada halaman ini admin akan dapat mengelola data administrator, jika ada *account* admin yang ingin ditambahkan diubah, dan dihapus jika diperlukan maka akan dapat dilakukan dihalaman ini.

3.5.2 Perancangan layar untuk halaman Daftar ulang peserta

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Home | Admin | Peserta | Hasil ujian

Formulir Pendaftaran Ujian

Login admin

Login peserta

Daftar

Hasil ujian

Kd peserta

Username

Password

Jenis kelamin Laki-laki Perempuan

Alamat

Email

Gambar 3.24 Rancangan layar untuk halaman daftar ulang peserta

Gambar 3.24 adalah perancangan layar halaman daftar ulang yang digunakan oleh peserta ujian untuk mendaftarkan ulang dan membuat *username* serta *password* nya sendiri untuk dapat *login* ke halaman ujian.

3.5.3 Perancangan layar untuk halaman Login peserta

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Home | Admin | Peserta | Hasil ujian

Login Peserta

Login admin
Login peserta
Daftar
Hasil ujian

Kd peserta

Password

Gambar 3.25 Rancangan layar untuk halaman Login peserta

Gambar 3.25 adalah rancangan layar *login* peserta yang digunakan oleh user yang mempunyai hak sebagai peserta ujian. Jika *user* adalah peserta ujian maka dia berhak mengakses halaman ini.

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Ujian | Logout

Halaman Peserta

Selamat datang di halaman peserta,
Terima kasih anda telah mendaftar dan login ke halaman ini.
Bagi peserta yang ingin mengikuti test kami ucapkan semoga berhasil dan memperoleh hasil nilai yang memuaskan.
silakan memilih menu link yang tersedia pada halaman ini.

Gambar 3.26 Rancangan layar halaman peserta

Gambar 3.26 ini adalah halaman pembuka yang akan muncul setelah peserta ujian login dan memasukan kd_peserta serta *passwordnya*. Setelah halaman pembuka muncul peserta ujian akan dihadapkan pada rancangan layar untuk *link* menuju halaman ujian, dan untuk juga link logout untuk keluar dari halaman peserta.

Jika peserta memilih *link* ujian maka akan dapat terlihat modul – modul soal yang diujikan seperti yang terlihat pada gambar 3.27 adalah rancangan layar untuk link modul soal ujian yang dapat dipilih tersebut dapat dilihat sebagai berikut.

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Ujian Logout

SOAL UJIAN

Matematika|Bhs Ing|Bhs Ind|Biologi|Fisika|Sejarah

Gambar 3.27 Rancangan layar modul soal ujian untuk peserta

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Ujian Logout

Soal Ujian

No	Pertanyaan
1	Siapa nama saya ?
	<input type="radio"/> A. Adhelia
	<input type="radio"/> B. Ade
	<input type="radio"/> C. Lia
	<input type="radio"/> D. Del

Jawaban sebelumnya:

Jawab Ubah Selesai

Gambar 3.28 Rancangan layar soal ujian untuk peserta

Gambar 3.28 tersebut adalah rancangan layar untuk soal ujian peserta dimana jika peserta memilih link modul untuk soal tertentu maka akan ditampilkan halaman soal yang harus dijawab setelah selesai menjawab soal pada modul tertentu maka peserta dapat pindah ke link modul soal selanjutnya yang harus dijawab dan gambar 3.29 berikut adalah gambar rancangan layar soal ujian jika sudah dijawab.

The screenshot shows a web application titled "APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB". At the top, there are two buttons: "Ujian" and "Logout". Below this is a section titled "Soal Ujian". It contains a table with two columns: "No" and "Pertanyaan". The first row shows the number "1" and the question "Siapa nama saya?". Below the question are four radio button options: "A. Adhelia" (which is selected), "B. Ade", "C. Lia", and "D. Del". Underneath the options, it says "Jawaban sebelumnya: A". At the bottom of the question area, there are three buttons: "Jawab", "Ubah", and "Selesai".

Gambar 3.29 Rancangan layar soal ujian untuk peserta jika sudah dijawab

The screenshot shows a simple dialog box with a title "Data Berhasil Dirubah" and a single "Ok" button centered below it.

Gambar 3.30 Rancangan layar jika jawaban ingin dirubah

Gambar 3.30 tersebut adalah rancangan layar jika peserta ingin merubah maka peserta harus memilih jawaban pada radio button setelah itu menekan

tombol ubah setelah itu akan tampil kotak pesan “data berhasil dirubah”, lalu peserta harus menekan tombol ok agar kembali kehalaman modul.

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Ujian Logout

Soal Ujian

No 1 Pertanyaan
Siapa nama saya ?

- A. Adhelia
- B. Ade
- C. Lia
- D. Del

Jawaban sebelumnya:

Jawab Ubah Selesai

Apakah anda yakin ingin mengakhiri ujian?

OK Cancel

Gambar 3.31 Rancangan layar peserta jika ingin mengakhiri ujian

Gambar 3.31 tersebut adalah gambar jika peserta ingin mengakhiri ujian dan menekan tombol selesai maka akan keluar pesan tersebut kemudian peserta harus menekan tombol ok jika benar ingin mengakhiri ujian.

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Hasil ujian Logout

Ujian Anda telah selesai, Untuk melihat hasil
silahkan pilih link "Hasil Ujian"

Gambar 3.32 Rancangan layar peserta jika telah selesai ujian

Gambar 3.32 adalah gambar rancangan layar peserta jika telah selesai ujian dan pesan untuk melihat hasil ujian.

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Home | Admin | Peserta | Hasil ujian

Hasil ujian Peserta

Menu

Hasil ujian
Logout

Kode peserta :
Soal Benar :
Nama peserta :
Soal salah :
Nilai test :
Tidak dijawab :
Keterangan :
Tanggal Test :

No soal	Jawaban anda	Kunci jawaban	Keterangan

Gambar 3.33 Rancangan layar halaman hasil ujian peserta

Gambar 3.33 tersebut adalah gambar rancangan layar untuk halaman hasil ujian peserta yang dimana pada halaman ini dapat dilihat hasil ujian yang telah diikuti oleh peserta.

3.5.4 Perancangan layar Halaman Melihat Hasil ujian Seluruh Peserta

APLIKASI UJIAN PENYARINGAN SISWA BARU BERBASIS WEB

Home | Admin | Peserta | Hasil ujian

Hasil ujian seluruh peserta

Kd peserta

Login admin
Login peserta
Daftar
Hasil ujian

Kode peserta	Nama peserta	Hasil ujian	Keterangan

Gambar 3.34 Rancangan layar untuk melihat hasil ujian seluruh peserta

Gambar 3.34 adalah perancangan layar halaman yang dapat digunakan oleh peserta ujian untuk dapat melihat hasil ujian seluruh peserta.

3.6 Spesifikasi Sistem

Dalam spesifikasi sistem ini terbagi ke dalam 2 kategori yaitu spesifikasi aplikasi dan spesifikasi pengguna.

3.6.1 Spesifikasi Aplikasi

Aplikasi *web* yang akan dibangun memiliki kemampuan sebagai berikut:

1. Dapat menampilkan sebuah ujian penyaringan siswa baru secara online intranet.
2. Pada aplikasi ujian penyaringan ini hasil ujian dan status kelulusan dapat diketahui secara langsung.
3. Aplikasi ini bersifat CMS (Content Management System) maksudnya soal-soal dan jawaban dapat dirubah atau ditambah secara dinamis pada modul administrator.

3.6.2 Spesifikasi Pengguna

Aplikasi ini ditujukan untuk digunakan oleh calon siswa SMU yang ingin masuk kesekolah yang dituju secara *online* intranet, menggunakan web browser.

3.7 Lingkungan Operasi

Untuk membangun aplikasi web sesuai dengan spesifikasi kebutuhan, dibutuhkan lingkungan operasi sebagai berikut:

1. Sistem operasi Linux.

Sistem operasi ini dipilih karena bersifat *free* (bebas) dan *open source*.

2. Sistem operasi Windows

Sistem ini dipilih karena penggunaannya yang bersahabat walaupun sifatnya yang tidak *free* (bebas).

3. Web Server.

Web server digunakan untuk memberikan layanan *web* sehingga dapat diakses oleh pengguna baik yang berada pada jaringan lokal maupun pada jaringan *internet*. *Web server* yang digunakan adalah Apache.

4. *Interpreter* PHP

PHP adalah bahasa pemrograman *server side* yang bekerja pada sisi server yang berfungsi untuk menangani *request* dari pengguna. Perangkat lunak ini digunakan karena bersifat gratis dan *open source*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Aplikasi Ujian Penyaringan Siswa Baru Berbasis Web

Pembuatan program aplikasi sistem ujian penyaringan berbasis web ini ditujukan untuk calon siswa dalam melakukan ujian. Sistem penyaringan ujian yang sebelumnya dilakukan bersifat manual, sehingga kurang efisien baik dari segi waktu maupun tenaga. Aplikasi ujian ini dapat diakses oleh peserta ujian hanya pada waktu dan tempat yang sudah ditetapkan karena sistem ini berbasis web dan dalam lingkup intranet saja. Disamping itu peserta ujian dapat mengetahui hasil ujian pada saat itu juga tanpa menunggu. Aplikasi ini juga membatasi peserta sehingga hanya bisa mengikuti ujian sebanyak satu kali saja.

Setelah melakukan analisis dan perancangan terhadap aplikasi ujian penyaringan ini tahapan selanjutnya adalah implementasi dan pengujian. Pada tahapan pengujian terdapat dua cakupan spesifikasi kebutuhan sistem yang meliputi perangkat keras dan perangkat lunak, implementasi pada aplikasi ini meliputi proses pengkodean basis data dan tampilan antar muka.

4.1.1 Spesifikasi Kebutuhan Sistem

Dalam aplikasi ini spesifikasi kebutuhan sistem menggunakan dua buah perangkat yaitu perangkat keras dan perangkat lunak. Berikut ini adalah kebutuhan sistem ketika sistem ini dibuat dan diuji.

1. Perangkat Keras (hardware)
 - a. Processor Intel Pentium R 2.26 Ghz.
 - b. RAM 512MB VGA ATI RADEON 7000 Series 64Mb.
 - c. Hardisk 40 Gb.
2. Perangkat Lunak (Software)
 - a. operating system windows XP Profesional.
 - b. XAMPP 1.6.8 PHP Script language version 5.1.6.
 - c. My SQL Database version 5.0.24a.

4.2 Tampilan Aplikasi Ujian penyaringan berbasis web

Untuk menjalankan program ini, yang dilakukan adalah membuka *web browser*, kemudian ketikkan nama host yang menjadi *server* pada bagian *address/location*. Bila *host* yang digunakan sama dengan *host* yang dipakai saat ini, nama *host* dapat diisi dengan localhost.

4.2.1 Halaman pembuka

Gambar 4.1 merupakan gambar halaman tampilan utama program yang berisi halaman pembuka untuk user yang mengakses aplikasi ini, dan berisi menu-menu yaitu login untuk admin, login untuk peserta, daftar ulang untuk peserta,

dan juga hasil ujian. Berikut ini adalah penggalan *coding* dan tampilan halaman pembuka pada aplikasi ini.

Penggalan skrip halaman pembuka

```
</style></head><BODY text=#000000 bgproperties="fixed">
<table width="794" height="260" border="0" align="center"
cellpadding="0" cellspacing="0" id="Table_01">
<tr>
<td width="34" background="image/index_04.gif">&nbsp;</td>
<td width="722" align="left" valign="top"
background="image/index_05.gif"><table width="100%" border="0"
cellpadding="0" cellspacing="0" style="BORDER-RIGHT: #003366 1px
solid; BORDER-TOP: #003366 1px solid; BORDER-LEFT: #003366 1px
solid; BORDER-BOTTOM: #003366 1px solid" >
<tr>
<td width="21%" height="218" align="center" valign="top"><p>
<? include("kiri-Index.html");?>
</p>
<p>&nbsp;</p></td>
<td width="79%" align="center" valign="top" class="isi">
<p><br>
</p>
<p><span class="judul"><strong>SELAMAT DATANG</strong></span></p>
<p>&nbsp;</p>
<p align="justify">Aplikasi Ujian ini dibuat khusus untuk calon
siswa Sekolah Menengah Umum yang ingin masuk kesekolah yang dituju.
Aplikasi ini memudahkan calon siswa Sekolah Menengah Umum sehingga
tidak perlu perlengkapan yang berarti seperti alat tulis dan
sebagainya.
Aplikasi ini memiliki tingkat kecuratan dalam hal penilaian dan juga
hasil ujian dapat diketahui secara langsung. <br>
</p>
<p>&nbsp;</p>
<p></font> </p></td>
</tr>
</table></td>
<td width="38" background="image/index_06.gif">&nbsp;</td>
</tr>
</table>
<? include "footer.php"; ?>
</body>
</html>
```


dilihat pada Gambar 4.3. Jika data tidak berhasil diinput maka akan muncul pesan bahwa kode peserta sudah terdaftar halaman ini dapat dilihat pada gambar 4.4.

Penggalan skrip halaman Daftar Ulang Peserta jika berhasil

```
<?
if (!isset($Kd_Peserta) && empty($Kd_Peserta)) {
    echo "Sorry ditolak!";
    return 0; }
include "koneksi.php";
mysql_select_db("$nama_database");
$numresult=mysql_query("select Kd_Peserta from dt_peserta where
Kd_Peserta='$Kd_Peserta'");
$cek=mysql_num_rows($numresult);
if($cek>0) {
echo("<script>");
    echo ("alert(\"Kode Peserta sudah terdaftar !!!\")");
echo ("history.go(-1)");
    echo ("</script>"); }
$perintah="insert into
dt_peserta(Kd_Peserta,Username,Password,J_Kelamin,Alamat,email)
values
('$Kd_Peserta','$Username','$Password','$J_Kelamin','$Alamat','$Semai
l')";
$hasil=mysql_query($perintah);
if ($hasil) {
$msg="Data sukses di input";
$alamat="daftar.php";
header("Location:$alamat?msg=$msg");
exit();
```



Gambar 4.3 Tampilan Halaman jika Daftar Ulang Peserta Berhasil



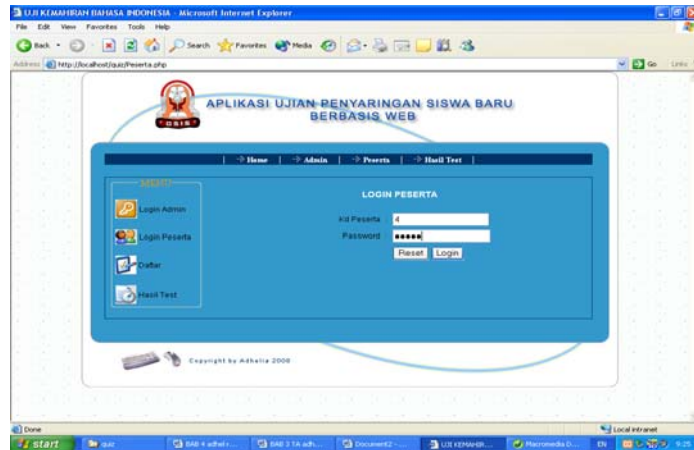
Gambar 4.4 Tampilan Halaman jika Daftar Ulang tidak berhasil

4.2.3 Tampilan Halaman untuk Login peserta

Layar login peserta digunakan oleh user yang mempunyai hak akses sebagai peserta ujian. Setelah user menginput kode peserta dan password kemudian akan divalidasi pada *database* apakah user tersebut benar terdaftar sebagai peserta ujian atau bukan. Validasi dilakukan dengan mencocokkan kode peserta dan password dengan yang ada dalam *database*. Jika user adalah peserta ujian maka dia berhak mengakses halaman ini dan berikut ini adalah penggalan koding dan tampilan halamannya yang dapat dilihat pada gambar 4.5.

Penggalan coding untuk login peserta

```
<form method="post" action="peserta_login.php">
<table border="0" cellpadding="2" cellspacing="2" style="border-
collapse: collapse" bordercolor="#111111" width="106%"
id="AutoNumber2" class="isi">
<tr>
<td width="40%" align="right">Kd Peserta </td>
<td width="3%" align="center">:</td>
<td width="57%" align="center"><input name="Kd_Peserta" size="20"
style="float: left">
</p>
</td></tr><tr>
<td width="40%" align="right">Password </td>
<td width="3%" align="center">:</td>
<td width="57%" align="center"><input type="password"
name="Password" size="20" style="float: left"></td>
</tr><tr><td align="center"><p>
<p align="center"></td><td align="center">&nbsp;</td>
<td align="left"><input name="reset" type="reset" value="Reset">
<input name="Submit" type="Submit" value="Login"></td>
</tr></table></form>
```



Gambar 4.5 Tampilan Halaman Untuk Login Peserta

Jika terjadi kesalahan pada saat peserta login maka akan keluar pesan data tidak valid kemudian peserta harus login kembali dengan kode peserta dan password yang benar. Pesan tersebut dapat dilihat pada gambar 4.6 dan berikut adalah penggalan kodingnya.

Penggalan Koding Kesalahan Login peserta.

```

<?php
$Kd_Peserta=strtolower($Kd_Peserta);
>Password=strtolower($Password);
if (($Kd_Peserta=='') or (ereg(";", $Kd_Peserta)) or
(ereg("", $Kd_Peserta)) {
$msg="Harap diisi dengan benar";
$alamat="peserta.php";
header("Location:$alamat?msg=$msg");
exit();
} elseif (($Password=='') or (ereg(";", $Password)) or
(ereg("", $Password))) {
$msg="Harap diisi dengan benar";
$alamat="peserta.php";
header("Location:$alamat?msg=$msg");
exit();
}
include "koneksi.php";
mysql_select_db("$nama_database");
$numresult=mysql_query("select Kd_Peserta,Password from dt_peserta
where Kd_Peserta='$Kd_Peserta' and Password='$Password'");
$cek=mysql_num_rows($numresult);
if ($cek==0) {
$msg="Data tidak valid";
$alamat="peserta.php";
header("Location:$alamat?msg=$msg");
exit();
} else {

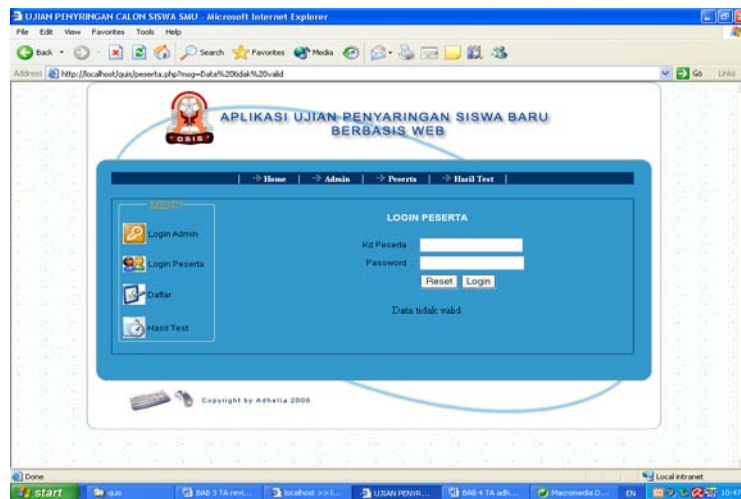
```



```

$hasil=mysql_query("select Kd_Peserta,Password from dt_peserta where
Kd_Peserta='$Kd_Peserta' and Password='$Password'");
$baris=mysql_fetch_row($hasil);
$Kd_Peserta=$baris[0];
$Password=$baris[1];
session_start();
session_register("Kd_Peserta","Password");
}
session_start();
?>

```



Gambar 4.6 Tampilan Halaman Kesalahan Login Peserta

Gambar 4.7 adalah halaman pembuka yang akan muncul setelah peserta ujian login dan memasukkan kode peserta serta *password*nya. Setelah halaman pembuka muncul, peserta ujian akan dihadapkan pada link untuk menuju ke halaman ujian dan *logout*. Setelah peserta selesai ujian, maka akan muncul link hasil ujian yang dapat diakses oleh peserta. Berikut ini adalah penggalan koding dan tampilan halamannya.

Penggalan coding Halaman Untuk Setelah Peserta Login

```

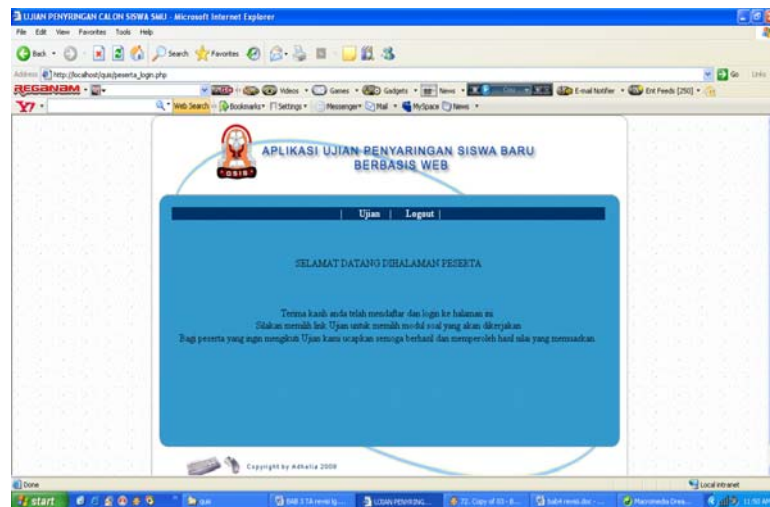
<?php
$Kd_Peserta=strtolower($Kd_Peserta);
$Password=strtolower($Password);
if (($Kd_Peserta=='') or (ereg(";", $Kd_Peserta)) or
(ereg("'", $Kd_Peserta))) {
$msg="Harap diisi dengan benar";
$alamat="peserta.php";

```

```

header("Location:$alamat?msg=$msg" );
exit();
} elseif (($Password=='') or (ereg(";", $Password)) or
(ereg("'", $Password))) {
$msg="Harap diisi dengan benar";
$alamat="peserta.php";
header("Location:$alamat?msg=$msg" );
exit(); }
include "koneksi.php";
mysql_select_db("$nama_database");
$numresult=mysql_query("select Kd_Peserta,Password from dt_peserta
where Kd_Peserta='$Kd_Peserta' and Password='$Password'");
$cek=mysql_num_rows($numresult);
if ($cek==0) {
$msg="Data tidak valid";
$alamat="peserta.php";
header("Location:$alamat?msg=$msg" );
exit(); } else {
$hasil=mysql_query("select Kd_Peserta,Password from dt_peserta where
Kd_Peserta='$Kd_Peserta' and Password='$Password'");
$baris=mysql_fetch_row($hasil);
$Kd_Peserta=$baris[0];
$Password=$baris[1];
session_start();
session_register("Kd_Peserta", "Password"); }
session_start();
?>

```



Gambar 4.7 Tampilan Halaman setelah peserta Login

Gambar 4.8 adalah tampilan halaman ujian untuk peserta. Pada halaman tersebut terdapat modul-modul soal seperti matematika, bahasa indonesia, bahasa inggris, biologi, fisika, dan sejarah yang dapat dipilih peserta. Gambar 4.9 adalah tampilan halaman soal ujian untuk peserta, yaitu pada saat peserta tersebut

memilih salah satu modul maka akan keluar halaman soal ujian yang harus dijawab. Jika peserta sudah menjawab semua soal pada modul tersebut maka peserta harus menekan tombol jawab agar jawaban dapat tersimpan pada *database*.

Setelah itu akan keluar halaman kotak pesan “Data berhasil diinput” dan tombol OK yang harus ditekan oleh peserta agar dapat kembali ke halaman soal ujian yang sudah ada jawabannya dan juga pada halaman soal bagian bawah akan keluar pesan berjalan “anda sudah selesai mengerjakan soal pada modul ini dan untuk mengubah jawaban tekan tombol ubah”. Berikut ini adalah penggalan koding dan tampilan gambarnya yang dapat dilihat pada gambar 4.8, 4.9, 4.10, 4.11.

Penggalan coding Halaman Uji Test Peserta

```
?>
<center>
<span class="judul"><br>
SOAL UJIAN</span><br>
</p> <div align="center">|
<a href="<? echo "$PHP_SELF?soal=matematika&&peserta=$Kd_Peserta";
?>">Matematika</a> |
<a href="<? Echo "$PHP_SELF?soal=bhs_inggris&&peserta=$Kd_Peserta";
?>">Bhs.Inggris</a> |
<a href="<? Echo "$PHP_SELF?soal=bhs_indonesia&&peserta=$Kd_Peserta";
?>">Bhs. Indonesia</a> |
<a href="<? echo "$PHP_SELF?soal=biologi&&peserta=$Kd_Peserta";
?>">Biologi</a> |
<a href="<? echo "$PHP_SELF?soal=fisika&&peserta=$Kd_Peserta";
?>">Fisika</a> |
<a href="<? echo "$PHP_SELF?soal=sejarah&&peserta=$Kd_Peserta";
?>">Sejarah</a> | </div> <br>
<br>
<center>
<form action="simpan_test.php">
<?
include "koneksi.php";
$querys=mysql_db_query($nama_database,"select * from soal where
module=' $soal' ");
$b=1;
while ($row=mysql_fetch_array ($querys)) {
    //$jmls = $row[Id_Pertanyaan];
    $Id_Pertanyaan[$b]=$row[0];
    $Pertanyaan[$b]=$row[4];
    $JawabA[$b]=$row[5];
    $JawabB[$b]=$row[6];
    $JawabC[$b]=$row[7];
```

```
$JawabD[$b]=$row[8];  
$no_Pertanyaan[$b]=$row[1];  
$module[$b]=$row[3];
```

?>



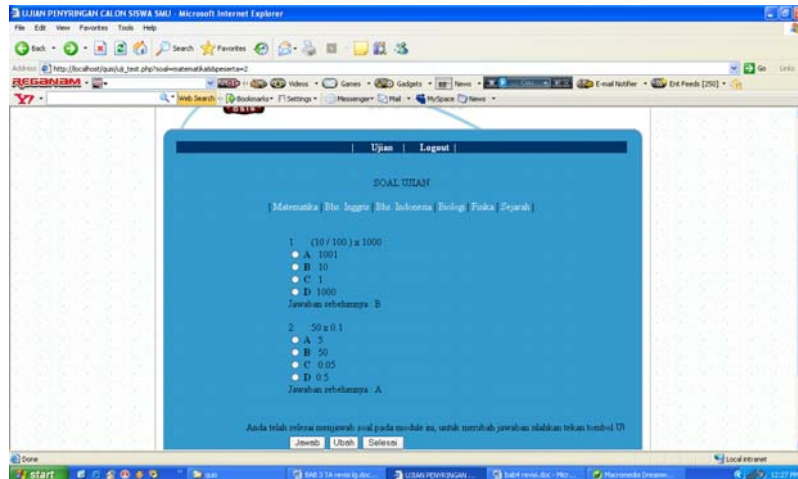
Gambar 4.8 Tampilan Halaman Modul soal



Gambar 4.9 Tampilan Halaman soal ujian pada modul



Gambar 4.10 Tampilan Halaman jika peserta menekan tombol jawab



Gambar 4.11 Tampilan Halaman soal jika sudah dijawab

Jika peserta ingin mengubah jawaban sebelumnya maka peserta harus memilih jawaban lagi pada radio button dan menekan tombol ubah kemudian akan ada kotak pesan “Data berhasil diubah” dan tombol OK yang harus ditekan oleh peserta agar kembali ke halaman soal yang jawabannya telah berubah. Tampilan halaman tersebut dapat dilihat pada gambar 4.12 berikut ini.

Penggalan coding Halaman Ubah jawaban ujian

```

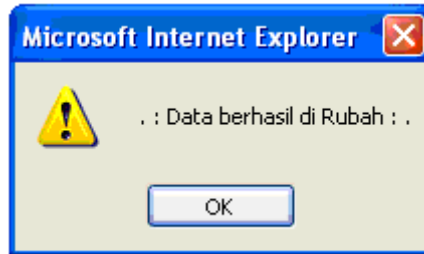
if($Submit == "Ubah" )
{
$dates = date("Y/m/d");
for($x=1; $x<$jml; $x++)
{
$hasil="select      Id_Pertanyaan,Jawaban      from      soal      where
Id_Pertanyaan=' $Id_Pertanyaans[$x] ' ";
$result=mysql_query($hasil);
$row=mysql_fetch_row($result);
$Id_Pertanyaan[$x]=$row[0];
$Jawaban[$x]=$row[1];
if ($Jawaban[$x]==$jawab[$x]) {
$keterangans[$x]=" benar ";
}
if ($Jawaban[$x]!=$jawab[$x]) {
$keterangans[$x]=" salah ";
}
$perintah="update jawab set
jawab      = '$jawab[$x]',
module = '$modules',
keterangan = '$keterangans[$x]',
tgl_test      = '$dates'
where
Kd_Peserta = '$Kd_Peserta' and

```

```

    Id_Pertanyaan = '$Id_Pertanyaans[$x]' ";
    $hasil=mysql_query($perintah);
    echo $perintah;
}

```



Gambar 4.12 Tampilan Halaman pesan jika jawaban diubah

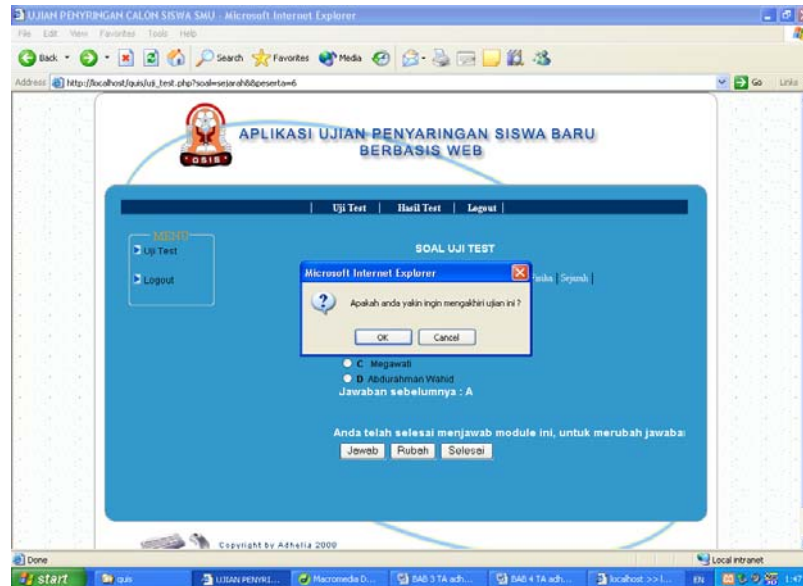
Selanjutnya peserta dapat pindah ke modul soal yang lain yang harus dikerjakan sampai selesai. Setelah peserta menjawab semua soal pada modul maka peserta harus menekan tombol selesai sehingga akan muncul kotak pesan untuk mengakhiri ujian dan keluar dari halaman ujian untuk melihat hasilnya. Tampilan halaman tersebut dapat dilihat pada gambar 4.13.

Penggalan coding Halaman pesan untuk mengakhiri ujian

```

    <?
    if($_GET['soal'] !="")
    {
    ?>
    <table width="50%" border="0" align="center" cellpadding="0"
    cellspacing="0">
    <tr>
    <td align="left"><input name="Submit" type="Submit" id="Submit"
    value="Jawab">
    <input name="Submit" type="submit" value="Ubah">
    <label>
    <input name="Submit" type="Submit" id="Submit" value="Selesai"
    onClick="return confirm('Apakah anda yakin ingin mengakhiri ujian
    ini ? ');">
    </label>
    <input name="jml" type="hidden" value="<? echo $b; ?>"></td>
    </tr>
    </table>
    <?
    }
    ?>

```



Gambar 4.13 Tampilan Halaman jika peserta menekan tombol selesai

Pada Gambar 4.14 adalah tampilan halaman pesan untuk peserta yaitu jika dia ingin melihat hasil ujian yang telah diikutinya. Untuk melakukannya peserta harus memilih link hasil ujian. Setelah peserta mengikuti ujian penyangannya maka dia tidak dapat mengakses lagi halaman ujian karena halaman tersebut telah diblok untuk para peserta yang telah selesai ujian. Yang dapat diakses hanya halaman hasil ujian saja, berikut adalah penggalan kodingnya dan tampilan halamannya.

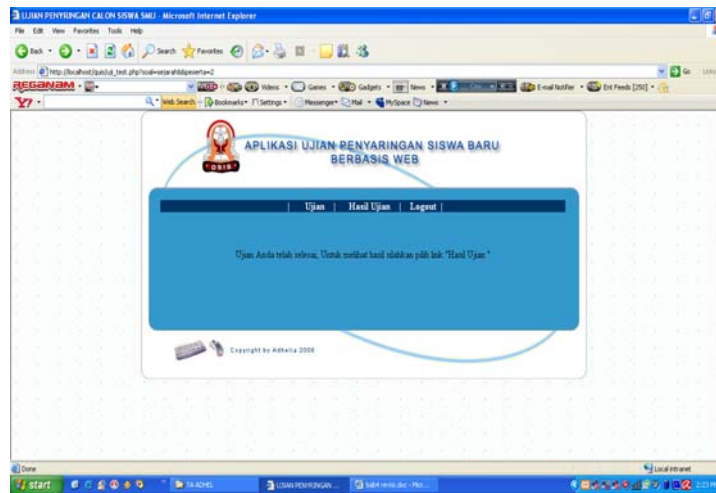
Penggalan coding Halaman pesan melihat hasil ujian

```
<? include ("Atas-Peserta.html");?>
<html>
<head>
</head><BODY text=#000000 leftMargin=0 bgproperties="fixed"
background="image/Bgrnd.gif">
<table width="794" height="144" border="0" align="center"
cellpadding="0" cellspacing="0" id="Table_01"><tr>
<td width="34" height="144"
background="image/index_04.gif"&nbsp;  ></td>
<td width="722" align="left" valign="top" bgcolor="#3399CC"><table
border="0" cellpadding="0" cellspacing="0" bordercolor="#111111"
width="100%"><tr>
```

```

<td width="18%" align="center" valign="top">
<? include("kiri-Peserta.php");?></td>
<td width="82%" valign="top">
<p>
<?
if($kd_jenissoal !=" ")
{
?>
</p>
<p>&nbsp;   </p>
<p align="center"><span class="judul">Ujian Anda telah selesai,
Untuk melihat hasil silahkan pilih link "Hasil Test"
</span>
<?
}
if($kd_jenissoal ==" ")
{
?>

```



Gambar 4.14 Tampilan Halaman pesan untuk peserta melihat Hasil Ujian

Pada gambar 4.15 adalah tampilan halaman untuk melihat hasil ujian peserta yang dapat diketahui secara langsung setelah peserta tersebut mengikuti ujian penyingiran. Berikut ini adalah penggalan koding dan tampilannya.

Penggalan coding Halaman pesan untuk melihat hasil ujian peserta

```

<?
include "koneksi.php";
mysql_select_db("$nama_database");
$query=mysql_db_query($nama_database,"select count(keterangan) from
jawab where Kd_Peserta='$Kd_Peserta' and keterangan='benar'");
while ($row=mysql_fetch_array ($query)) {
$jum=$row[0];
}
echo"<br>";

```



```

$nilai=$jum*2;
include "koneksi.php";
mysql_select_db("$nama_database");
include "koneksi.php";
mysql_select_db("$nama_database");
$query=mysql_db_query($nama_database,"select count(keterangan) from
jawab where Kd_Peserta='$Kd_Peserta' and keterangan='benar'");
while ($row=mysql_fetch_array ($query)) {
$jum=$row[0];
}
$query=mysql_db_query($nama_database,"select count(keterangan) from
jawab where Kd_Peserta='$Kd_Peserta' and keterangan='salah'");
while ($row=mysql_fetch_array ($query)) {
$salah=$row[0];
}
include "koneksi.php";
?>

```



Gambar 4.15 Tampilan Halaman Hasil ujian peserta

Pada gambar 4.16 adalah tampilan halaman untuk melihat hasil ujian seluruh peserta yang dapat diketahui secara langsung setelah peserta mengikuti ujian penaringan dengan cara mengklik link hasil ujian pada halaman menu utama, berikut ini adalah penggalan kodingnya dan tampilan gambarnya.

Penggalan coding Halaman Hasil ujian seluruh peserta ujian

```

<?
include "koneksi.php";
if (empty($awal))
$awal=0;

```

```

$query=mysql_db_query($nama_database,"select
dt_hasil.Kd_Peserta,dt_peserta.Username,dt_hasil.Hasil_ujian from
dt_hasil,dt_peserta where dt_peserta.Kd_Peserta =
dt_Hasil.Kd_Peserta limit $awal,5");
?
<span class="judul" align="center">HASIL UJIAN SELURUH
PESERTA</span>
<form method="post" action="cari.php">
<table class="isi"><tr><td><b>Kode Peserta : </td>
<td><input name="Kd_Peserta" type="text">
<input name="submit" type="submit" value="Cari"></td>
</tr></table>
</form>
<table border="1" cellpadding="2" cellspacing="2" style="border-
collapse: collapse" bordercolor="#111111" width="90%"
id="AutoNumber2" class="isi">
<tr bgcolor="#FFFFFF" class="isi">
<td width="30%" align="center"><b>Kd Peserta</td>
<td width="30%" align="center"><b>Nama Peserta</td>
<td width="20%" align="center"><b>Hasil Test</td>
<td width="20%" align="center"><b>Keterangan</td>
</tr>
<?
while ($row=mysql_fetch_array ($query)) {
if ($row[Hasil_Ujian]>=60) {
$keterangan="Lulus";} else { $keterangan="Tidak Lulus";}
$keterangan=$keterangan;
?>
<tr><td width="30%" align="center" bgcolor="#99CCFF"><? echo
"$row[Kd_Peserta]" ?></td><td width="30%" align="center"
bgcolor="#99CCFF"><? echo "$row[Username]" ?></td>
<td width="20%" align="center" bgcolor="#99CCFF"><? echo
"$row[Hasil_ujian]" ?></td>
<td width="20%" align="center" bgcolor="#99CCFF"><? echo
"$keterangan" ?></td>
</tr> <? } ?>
</table><?

```



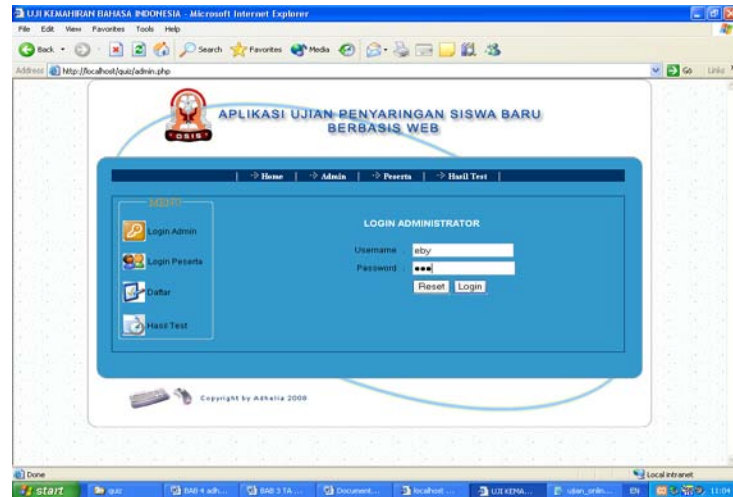
Gambar 4.16 Tampilan Halaman Untuk Hasil Ujian seluruh peserta

4.2.4 Tampilan Halaman untuk Admin

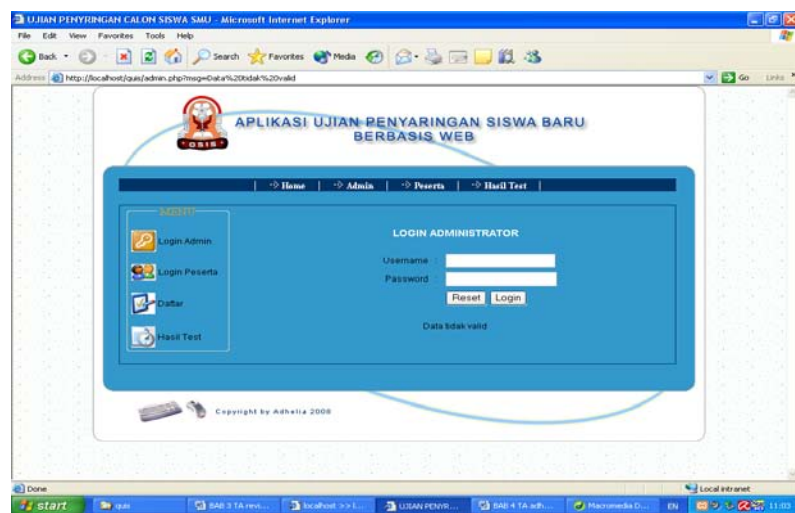
Gambar 4.17 adalah tampilan halaman untuk login administrator yang dapat digunakan oleh user yang mempunyai hak akses sebagai administrator saja. Setelah user mengisi username dan password kemudian akan divalidasi ke *database* apakah user tersebut administrator atau bukan dengan mencocokkan username dan password dengan yang ada dalam *database*. Jika user adalah administrator maka user berhak mengakses halaman ini. Jika terjadi kesalahan pada saat admin login maka akan keluar pesan data tidak valid kemudian peserta harus login kembali dengan *username* dan *password* yang benar. Pesan tersebut dapat dilihat pada Gambar 4.18 dan berikut ini adalah penggalan koding dan tampilan halamannya.

Penggalan coding Untuk Login dan kesalahan login Administrator

```
<?php
$Admin=strtolower($Admin);
>Password=strtolower($Password);
if (($Admin=='') or (ereg(";", $Admin)) or (ereg("'", $Admin))) {
$msg="Harap diisi dengan benar";
$alamat="admin.php";
header("Location:$alamat?msg=$msg");
exit();
} elseif (($Password=='') or (ereg(";", $Password)) or
(ereg("'", $Password))) {
$msg="Harap diisi dengan benar";
$alamat="admin.php";
header("Location:$alamat?msg=$msg");
exit();
}
include "koneksi.php";
mysql_select_db("$nama_database");
$numresult=mysql_query("select * from admin where Admin='$Admin' and
Password='$Password'");
$cek=mysql_num_rows($numresult);
if ($cek==0) {
$msg="Data tidak valid";
$alamat="admin.php";
header("Location:$alamat?msg=$msg");
exit();
} else {
$hasil=mysql_query("select Admin,Password from admin where
Admin='$Admin' and Password='$Password'");
$baris=mysql_fetch_row($hasil);
$Admin=$baris[0];
>Password=$baris[1];
```



Gambar 4.17 Tampilan Halaman Untuk Login Administrator



Gambar 4.18 Tampilan Halaman kesalahan Login Administrator

Pada gambar 4.19 adalah tampilan halaman pembuka yang akan muncul setelah administrator login dan memasukkan *username* serta *password*nya. Setelah halaman pembuka muncul, administrator akan dihadapkan pada link-link yang dapat dipilih untuk mengelola data. Administrator dapat memilih halaman sebagai

berikut: halaman data peserta, halaman data soal, data nilai dan data administrator untuk dikelola datanya. Berikut ini adalah penggalan koding dan tampilan gambarnya.

Penggalan Koding Untuk Halaman Pembuka Untuk Administrator

```
<?
  include ("Atas-admin.html");?>
<html>
<head>
</head><BODY text=#000000 leftMargin=0 bgproperties="fixed"
background="image/Bgrnd.gif">
<table width="794" height="144" border="0" align="center"
cellpadding="0" cellspacing="0" id="Table_01">
<tr>
<td width="34" height="144"
background="image/index_04.gif">&nbsp;&nbsp;&nbsp;</td>
<td width="722" align="left" valign="top" bgcolor="#3399CC"><table
width="100%" height="139" border="0" cellpadding="0" cellspacing="0"
style="BORDER-RIGHT: #003366 1px solid; BORDER-TOP: #003366 1px
solid; BORDER-LEFT: #003366 1px solid; BORDER-BOTTOM: #003366 1px
solid">
<tr>
<td align="center" valign="top"> <p><br>
<br>
<center>
<span class="judul">HALAMAN ADMINISTRATOR</span>
</p>
<br>
Selamat datang dihalaman admin,<br>
Anda memiliki hak penuh dalam pengelolaan seluruh data. <br>
silakan memilih menu link yang tersedia pada halaman ini.
</p>
</p>
<p>&nbsp;&nbsp;&nbsp;</p></td>
</tr>
</table></td>
  <td width="38" background="image/index_06.gif">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
<? include "footer.php"; ?>
</body>
</html>
```



Gambar 4.19 Tampilan Halaman Pembuka Untuk Administrator

Gambar 4.20 adalah tampilan halaman data peserta pada link administrator yang dapat digunakan untuk melihat daftar peserta yang mengikuti ujian penyaringan siswa baru. Pada halaman ini administrator dapat menambah, mengubah, serta menghapus data peserta. Berbagai aktifitas tersebut dapat dilihat pada gambar 4.21, 4.22, 4.23, 4.24, 4.25, 4.26. Berikut ini adalah penggalan kodingnya dan tampilan gambarnya.

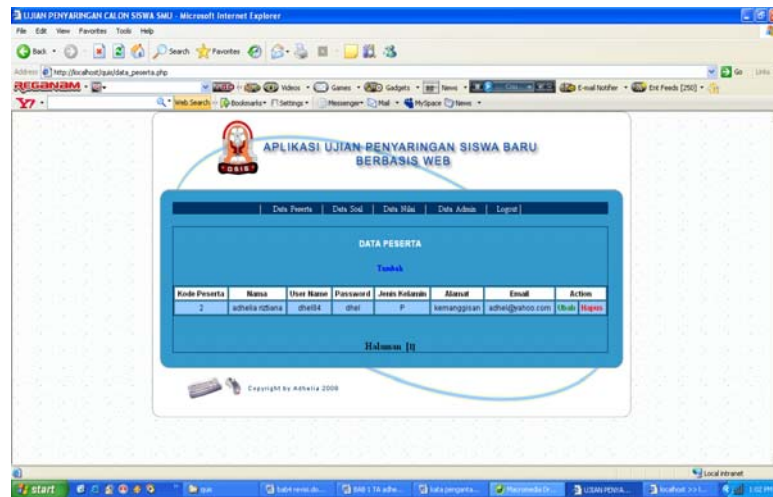
Penggalan Halaman Data Peserta pada Link administrator

```
<?
include "koneksi.php";
if (empty($awal))
$awal=0;
$query=mysql_db_query($nama_database,"select * from dt_peserta order
by Kd_Peserta limit $awal,5");
?>
<?      echo      "<A      href=\"tambah_data_peserta.php\"><font
color=blue><b>Tambah</a>" ?>      </p>
<table border="1" cellpadding="2" cellspacing="2" style="border-
collapse: collapse" bordercolor="#111111" width="100%"
id="AutoNumber2">
<tr class="isi">
<td align="center" bgcolor="#FFFFFF"><b>Kode Peserta</b></td>
<td align="center" bgcolor="#FFFFFF"><b>Nama</b></td>
<td align="center" bgcolor="#FFFFFF"><b>User Name</b></td>
<td align="center" bgcolor="#FFFFFF"><b>Password</b></td>
<td align="center" bgcolor="#FFFFFF"><b>Jenis Kelamin</b></td>
<td align="center" bgcolor="#FFFFFF"><b>Alamat</b></td>
```

```

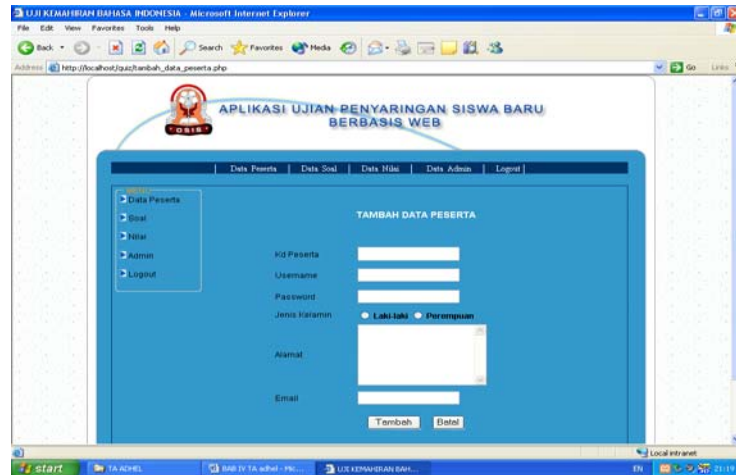
<td align="center" bgcolor="#FFFFFF"><b>Email</b></td>
<td colspan="2" align="center" bgcolor="#FFFFFF"><b>Action</b></td>
</tr>
<?
while ($row=mysql_fetch_array ($query)) {
?>
<tr class="isi">
<td align="center" bgcolor="#99CCFF"><? echo "$row[Kd_Peserta]"
?></td>
<td align="center" bgcolor="#99CCFF"><? echo "$row[name]" ?></td>
<td align="center" bgcolor="#99CCFF"><? echo "$row[Username]"
?></td>
<td align="center" bgcolor="#99CCFF"><? echo "$row[Password]" ?>
</td>
<td align="center" bgcolor="#99CCFF"><? echo "$row[J_Kelamin]" ?>
</td>
<td align="center" bgcolor="#99CCFF"><? echo "$row[Alamat]" ?></td>
<td align="center" bgcolor="#99CCFF"><? echo "$row[email]" ?></td>
<td align="center" bgcolor="#99CCFF"><? echo "
<a href=\"ubah_data_peserta.php?Kd_Peserta=$row[Kd_Peserta]\"><font
color=green><b>Ubah</a>" ?></td>
<td align="center" bgcolor="#99CCFF" onClick="return hapus()" > <?
echo"<a href=\"hapus_data_peserta.php?Kd_Peserta=$row[Kd_Peserta]\">
<font color=red><b>Hapus</a>" ?>
</td>
</tr>
<?
}
?>
</table>
<?

```



Gambar 4.20 Halaman Data Peserta pada Link administrator

Gambar 4.21 adalah tampilan untuk halaman tambah data peserta pada link administrator yang nantinya akan digunakan jika ada penambahan peserta



Gambar 4.21 Halaman tambah Data Peserta pada Link administrator

Gambar 4.22 adalah tampilan halaman jika data peserta ujian berhasil ditambahkan maka akan muncul pesan seperti yang terlihat pada gambar 4.23 dan berikut ini adalah penggalan kodingnya.

Penggalan Koding Halaman jika Data Peserta berhasil ditambah

```

<?
if (!isset($Kd_Peserta) && empty($Kd_Peserta)) {
echo "Sorry ditolak!";
return 0;}
include "koneksi.php";
mysql_select_db("$nama_database");
$numresult=mysql_query("select Kd_Peserta from dt_peserta where
Kd_Peserta='$Kd_Peserta'");
$cek=mysql_num_rows($numresult);
if($cek>0){
echo("<script>");
echo ("alert(\"Kode peserta sudah terdaftar !!!\")");
echo ("history.go(-1)");
echo("</script>"); }
$perintah="insert into
dt_peserta(Kd_Peserta,Username,Password,J_Kelamin,Alamat,email)
values
('$Kd_Peserta','$Username','$Password','$J_Kelamin','$Alamat','$emai
l')";
$hasil=mysql_query($perintah);
if ($hasil){
$msg="Data sukses di input";
$alamat="tambah_data_peserta.php";
header("Location:$alamat?msg=$msg");
exit();
} else {echo("<script>");
echo ("alert(\"Data gagal diinput,Ulangi !!!\")");
echo ("history.go(-1)");
echo("</script>"); }?>

```



Gambar 4.22 Halaman jika Data Peserta berhasil ditambah

Gambar 4.23 adalah tampilan halaman untuk mengubah data peserta pada link administrator yang nantinya akan digunakan jika ada data peserta yang salah dan harus diubah. Gambar 4.24 adalah tampilan halaman pesan yang akan muncul jika data peserta ujian berhasil diubah. Berikut ini adalah penggalan kodingnya serta tampilan halamannya.

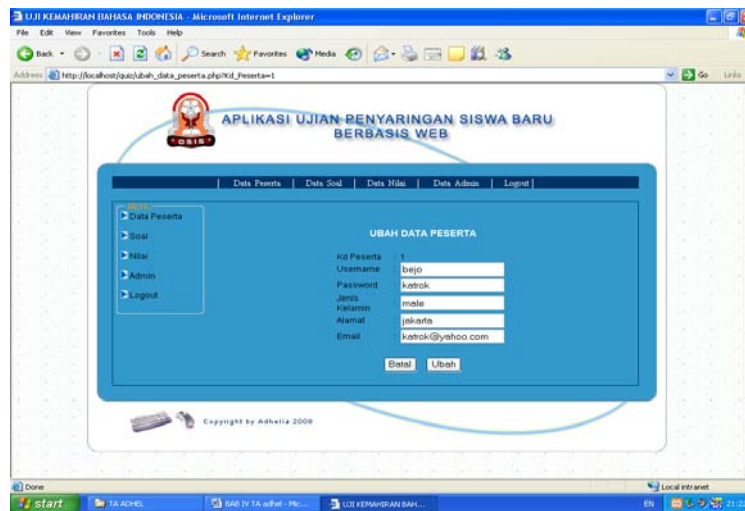
Penggalan Koding Halaman ubah Data Peserta

```
<form action="ubah1_data_peserta.php" method="post">
<table border="0" cellpadding="0" cellspacing="0" style="border-
collapse: collapse" bordercolor="#111111" width="40%"
id="AutoNumber1" class="isi">
<tr>
<td width="33%">Kd Peserta</td>
<td width="4%">:</td>
<td width="63%"><? echo "$Kd_Peserta"; ?></td>
</tr>
<tr>
<td width="33%">Username</td>
<td width="4%">:</td>
<td width="63%"><? echo "<input type='text' name='Username'
value='\"$Username\" size='20'>"; ?></td>
</tr>
<tr>
<td width="33%">Password</td>
<td width="4%">:</td>
<td width="63%"><? echo "<input type='text' name='Password'
value='\"$Password\" size='20'>"; ?></td>
</tr>
<tr>
<td width="33%">Jenis Kelamin</td>
```

```

<td width="4%">:</td>
<td width="63%"><? echo"<input type=\"text\" name=\"J_Kelamin\"
value=\"\$J_Kelamin\" size=\"20\">"; ?></td>
</tr>
<tr>
<td width="33%">Alamat </td>
<td width="4%">:</td>
<td width="63%"><? echo"<input type=\"text\" name=\"Alamat\"
value=\"\$Alamat\" size=\"20\">"; ?></td>
</tr>
<tr>
<td width="33%">Email</td>
<td width="4%">:</td>
<td width="63%"><? echo"<input type=\"text\" name=\"email\"
value=\"\$email\" size=\"20\">"; ?></td>
</tr>
<tr>
<?

```



Gambar 4.23 Halaman ubah Data Peserta



Gambar 4.24 Halaman jika Data Peserta berhasil diubah

Gambar 4.25 adalah tampilan halaman untuk menghapus data peserta pada link administrator yang nantinya dapat digunakan oleh admin untuk menghapus data. Jika ada peserta yang ingin membatalkan mengikuti ujian penyaringan maka akan muncul kotak pesan seperti pada gambar untuk memastikan apakah admin benar-benar akan menghapus data yang dimaksud.

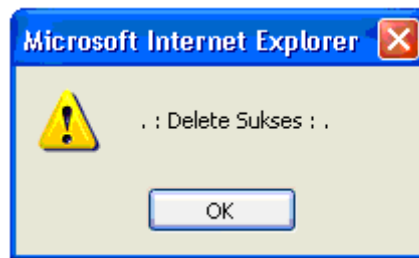
Gambar 4.26 adalah tampilan halaman pesan yang akan muncul jika data peserta ujian berhasil dihapus berikut ini adalah penggalan kodingnya dan tampilan halamannya.

Penggalan Koding Alert jika Data Peserta ingin dihapus

```
<?
session_start();
?>
<?
if (!isset($Kd_Peserta) && empty($Kd_Peserta)){
echo "Sorry ditolak!";
return 0;
}
include "koneksi.php";
mysql_select_db("$nama_database");
$perintah = mysql_query("delete from dt_peserta where
Kd_Peserta='$Kd_Peserta'");
if(!$perintah)
{
echo "<script>";
echo "alert(\". : Delete Gagal : .\");";
echo "</script>";
include "data_peserta.php";
}
else{
echo "<script>";
echo "alert(\". : Delete Sukses : .\");";
echo "</script>";
$perintah = mysql_query("delete from dt_hasil where
Kd_Peserta='$Kd_Peserta'");
$perintah = mysql_query("delete from jawab where
Kd_Peserta='$Kd_Peserta'");
include "data_peserta.php";
}
?>
```



Gambar 4.25 Alert jika Data Peserta ingin dihapus



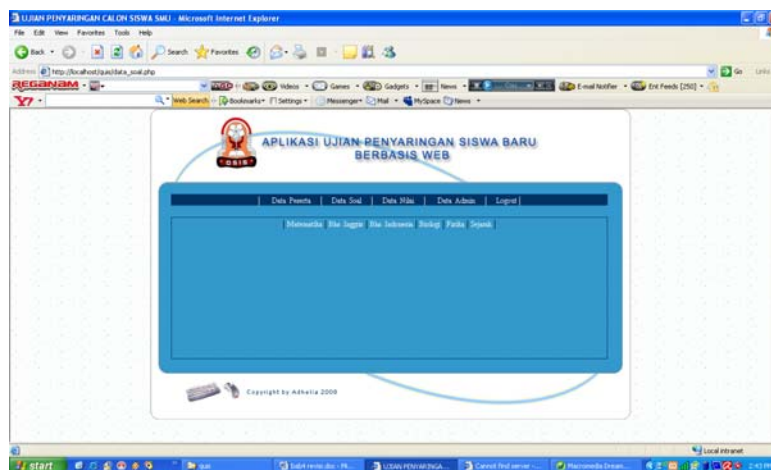
Gambar 4.26 Halaman jika Data Peserta berhasil dihapus

Gambar 4.27 adalah tampilan halaman link modul data soal pada halaman administrator yang dapat digunakan untuk melihat data soal ujian penyaringan siswa baru. Pada halaman ini terdapat link modul-modul soal matematika, bahasa indonesia, bahasa inggris, biologi, fisika dan sejarah yang dapat ditambah, diubah, serta dihapus data soalnya berikut ini adalah penggalan kodingnya dan tampilan halaman modul-modul soal yang dapat dilihat pada gambar 4.27, 4.28, 4.29, 4.30, 4.31, 4.32, 4.33.

Penggalan Koding Halaman Modul-modul Data Soal pada Link administrator

```
<?php
session_start();
session_register("Kd_Peserta","Password");
?>
<? include ("Atas-Peserta.html");?>
<html>
<head>
</head><BODY text=#000000leftMargin=0 bgproperties="fixed"
background="image/Bgrnd.gif">
<table width="794" height="144" border="0" align="center"
cellpadding="0" cellspacing="0" id="Table_01"><tr>
<td width="34" height="144"
background="image/index_04.gif">&nbsp;   </td>
<td width="722" align="left" valign="top" bgcolor="#3399CC"><table
border="0" cellpadding="0" cellspacing="0" bordercolor="#111111"
width="100%">
<tr>
<td width="18%" align="center" valign="top">
<? include("kiri-Peserta.html");?> </td>
<td width="82%" valign="top"><center>
<span class="judul"><br>
SOAL UJIAN</span> <br>
<div align="center">| <a href="<? echo "$PHP_SELF?soal=matematika";
?>">Matematika</a> | <a href="<? echo "$PHP_SELF?soal=bhs_inggris";
?>">Bhs. Inggris</a> | <a href="<? Echo
"$PHP_SELF?soal=bhs_indonesia";
?>">Bhs. Indonesia</a> | <a href="<? echo "$PHP_SELF?soal=biologi";
?>">Biologi</a> | <a href="<? echo "$PHP_SELF?soal=fisika";
?>">Fisika</a> | <a href="<? echo "$PHP_SELF?soal=sejarah";
?>">Sejarah</a> | </div>
<br>
<br>
<center>
<form action="simpan_test.php">
<?

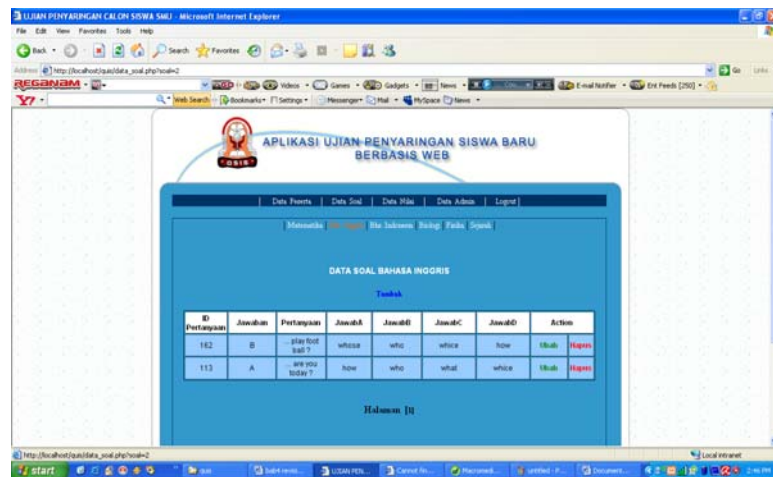
```



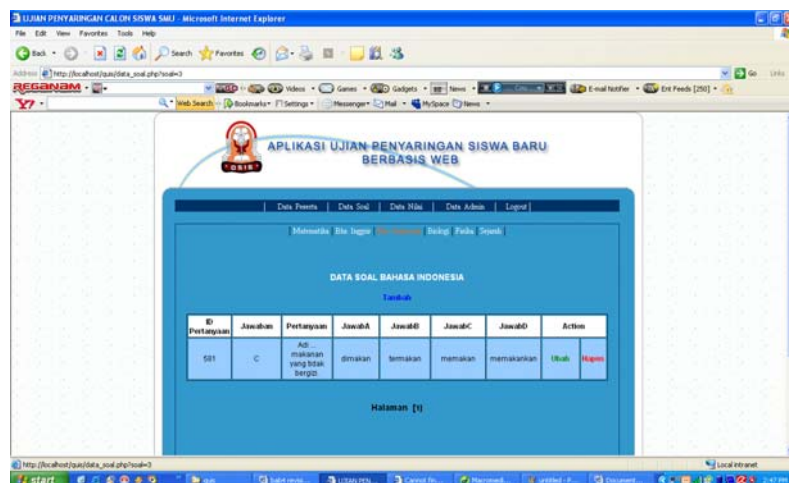
Gambar 4.27 Halaman Link Modul Data Soal pada administrator



Gambar 4.28 Halaman Modul Data soal Matematika



Gambar 4.29 Halaman Modul Data soal Bahasa Inggris



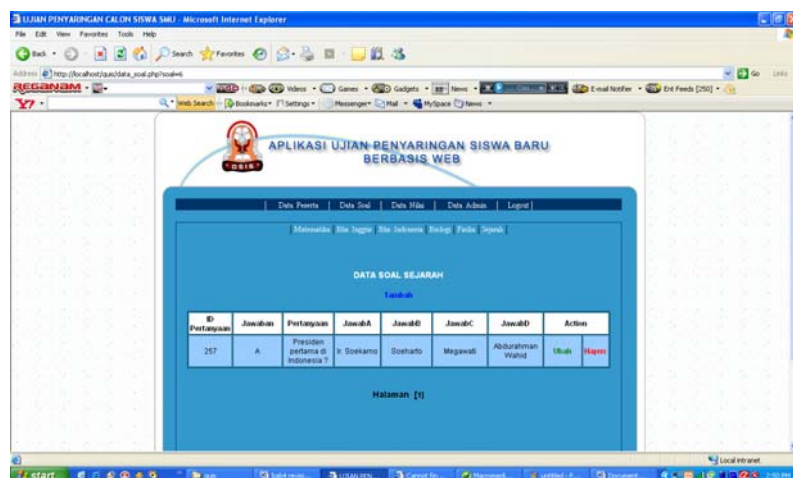
Gambar 4.30 Halaman Modul Data soal Bahasa Indonesia



Gambar 4.31 Halaman Modul Data soal Biologi



Gambar 4.32 Halaman Modul Data soal Fisika



Gambar 4.33 Halaman Modul Data soal Sejarah

Gambar 4.34 adalah tampilan halaman yang akan muncul jika link tambah data soal pada halaman administrator ditekan. Halaman ini akan digunakan jika ada penambahan soal ujian. Pada halaman ini hanya administrator saja yang mempunyai hak akses untuk dapat masuk serta menambahkan data soal ujiannya. Gambar 4.35 adalah tampilan halaman pesan yang akan muncul jika data soal ujian berhasil ditambahkan. Berikut ini adalah penggalan koding dan tampilan halamannya.

Penggalan Koding Halaman tambah Data Soal pada Link administrator

```

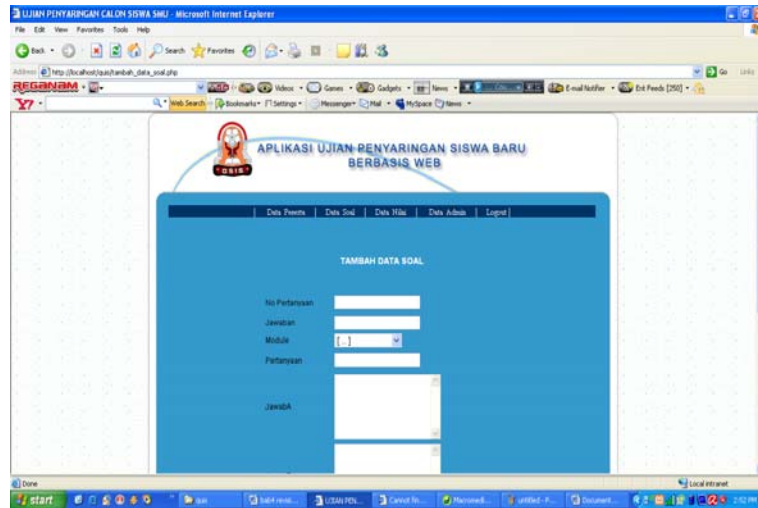
<form method="post" action="simpan_data_soal.php">
<table border="0" width="400" cellpadding="0" cellspacing="0"
height="86">
<tr>
<td width="400" height="19"><b> <? echo ("<center><b><font
color=#FF0000 size=4>$msg</b></center><br>"); ?>
<center>
<b>
<span class="judul">TAMBAH DATA SOAL</span></td>
<br></tr><tr>
<td width="245" height="1"><table border="0" width="400"
height="118" class="isi">
<tr><br><br>
<td width="112" height="31">Id Pertanyaan</td>
<td width="272" height="31"><input type="text" name="Id_Pertanyaan"
size="20"></td>
</tr>
<tr>
<td width="112" height="33">Jawaban</td>
<td width="272" height="33"><input type="text" name="Jawaban"
size="20"></td>
</tr>
<tr>
<td width="112" height="33">Pertanyaan</td>
<td width="272" height="33"><input type="text" name="Pertanyaan"
size="20"></td>
</tr>
<tr>
<td width="112" height="1">JawabA</td>
<td width="272" height="1"><textarea rows="6" name="JawabA"
cols="20"></textarea></td>
</tr>
<tr>
<td width="112" height="1">JawabB</td>
<td width="272" height="1"><textarea rows="6" name="JawabB"
cols="20"></textarea></td>
</tr>
<tr>
<td width="112" height="1">JawabC</td>
<td width="272" height="1"><textarea rows="6" name="JawabC"
cols="20"></textarea></td>
</tr>
<tr>

```

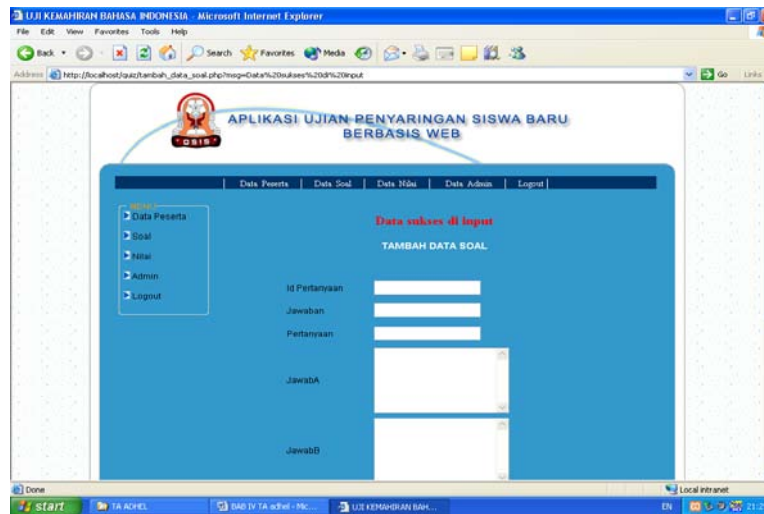
```

<td width="112" height="1">JawabD</td>
<td width="272" height="1"><textarea rows="6" name="JawabD"
cols="20"></textarea></td>
</tr>
<tr>

```



Gambar 4.34 Halaman tambah Data Soal pada Link administrator

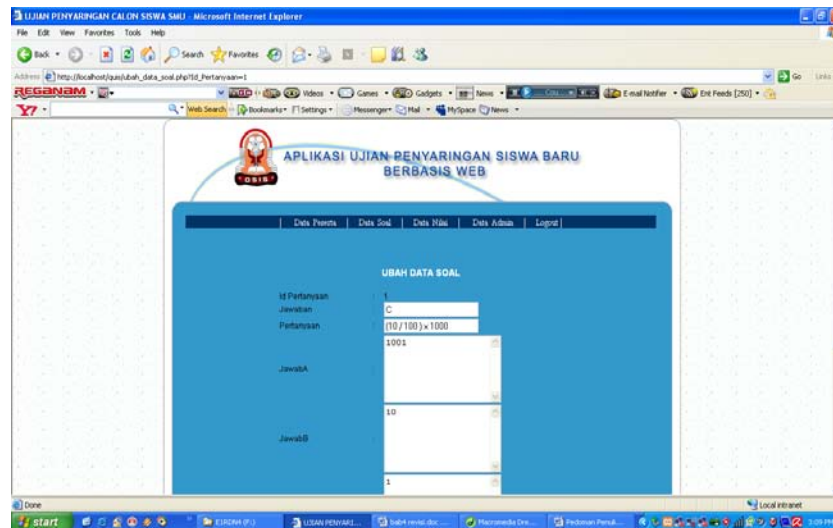


Gambar 4.35 Halaman jika tambah Data Soal berhasil

Gambar 4.36 adalah tampilan halaman yang akan muncul jika administrator menekan link untuk mengubah data soal. Halaman ini akan digunakan jika ada data soal yang akan diperbaharui, salah dan harus diubah.

Penggalan Koding Halaman ubah Data Soal

```
<?
include "koneksi.php";
mysql_select_db("$nama_database");
$hasil=mysql_query("select * from soal where
Id_Pertanyaan='$Id_Pertanyaan'");
$baris=mysql_fetch_row($hasil);
$Id_Pertanyaan=$baris[0];
$Jawaban=$baris[1];
$Pertanyaan=$baris[2];
$JawabA=$baris[3];
$JawabB=$baris[4];
$JawabC=$baris[5];
$JawabD=$baris[6];
?>
<div align="center">
<center>
<form action="ubah1_data_soal.php" method="post">
<table border="0" cellpadding="0" cellspacing="0" style="border-
collapse: collapse" bordercolor="#111111" width="60%"
id="AutoNumber1" class="isi">
<tr>
<td width="33%">Id Pertanyaan</td>
<td width="4%">:</td>
<td width="63%"><? echo "$Id_Pertanyaan"; ?></td>
</tr><tr>
<td width="33%">Jawaban</td>
<td width="4%">:</td>
<td width="63%"><? echo "<input type='text' name='Jawaban'
value='\"$Jawaban\" size='20'>"; ?></td>
</tr>
<tr>
```



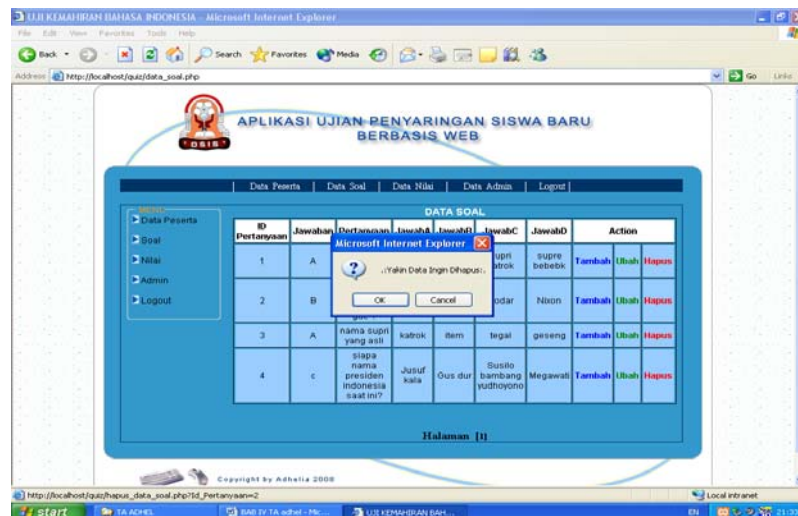
Gambar 4.36 Halaman ubah Data Soal

Gambar 4.37 adalah tampilan halaman yang akan muncul pada saat tombol untuk menghapus data soal ditekan. Jika ada data soal yang ingin dihapus maka akan muncul pesan seperti yang terlihat pada gambar untuk memastikan apakah admin benar-benar akan menghapus data soal yang dimaksud.

Gambar 4.38 adalah tampilan halaman pesan yang akan muncul jika data soal ujian berhasil dihapus. Berikut ini adalah penggalan kodingnya dan tampilan halamannya.

Penggalan Koding Halaman hapus Data Soal

```
<?
session_start();
?>
<?
if (!isset($Id_Pertanyaan) && empty($Id_Pertanyaan)){
echo "Sorry ditolak!";
return 0;}
include "koneksi.php";
mysql_select_db("$nama_database");
$perintah = mysql_query("delete from soal where
Id_Pertanyaan='$Id_Pertanyaan'");
if (!$perintah){
echo "<script>";
echo "alert(\". : Delete Gagal : .\");";
echo "</script>";
include "data_soal.php";
}else{
include "data_soal.php"; }
```



Gambar 4.37 Halaman hapus Data Soal



Gambar 4.38 Halaman jika Data Soal berhasil dihapus

Gambar 4.39 adalah tampilan link data nilai pada halaman administrator yang dapat digunakan untuk melihat hasil ujian peserta yang mengikuti ujian penyaringan siswa baru. Halaman ini juga dapat digunakan untuk melihat status peserta yang sedang ujian apakah dia sudah selesai mengikuti ujian atau belum. Pada halaman ini administrator juga dapat menghapus data nilai. Halaman tersebut dapat dilihat pada Gambar 4.40 dan halaman pesan setelah data dihapus dapat dilihat pada Gambar 4.41.

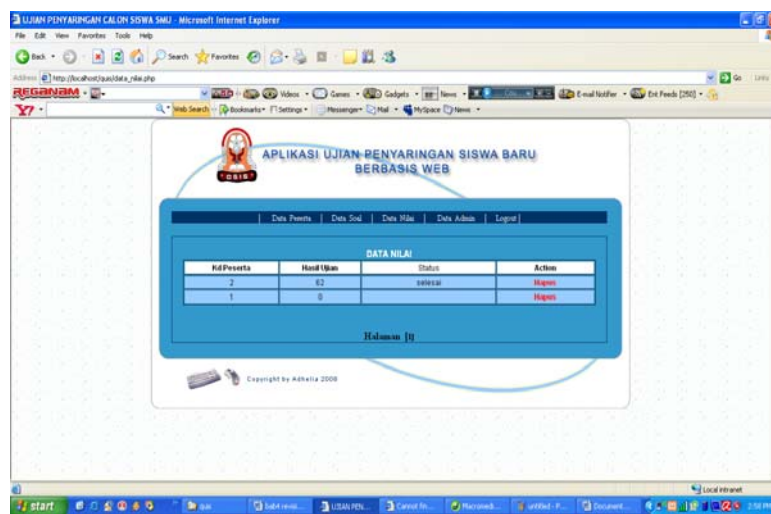
Penggalan Koding Halaman Data Nilai pada Link administrator

```
<?
include "koneksi.php";
if (empty($awal))
$awal=0;
$query=mysql_db_query($nama_database,"select * from dt_hasil limit
$awal,5");
?>
<table width="95%" border="1" align="center" cellpadding="2"
cellspacing="2" bordercolor="#111111" id="AutoNumber2"
style="border-collapse: collapse" class="isi">
<tr><td width="11%" align="center" bgcolor="#FFFFFF"><b>Kd
Peserta</td>
<td width="13%" align="center" bgcolor="#FFFFFF"><b>Hasil Test</td>
<td width="24%" colspan="3" align="center"
bgcolor="#FFFFFF"><b>Action</td>
</tr>
<?
while ($row=mysql_fetch_array ($query)) {
?>
<tr>
<td width="11%" align="center" bgcolor="#99CCFF"><? echo
"$row[Kd_Peserta]" ?></td>
<td width="13%" align="center" bgcolor="#99CCFF"><? echo
"$row[Hasil_Test]" ?></td>
```

```

<td width="10%" align="center" bgcolor="#99CCFF"><? echo "<A
href=\"tambah_data_nilai.php\"><font color=blue><b>Tambah</a>"
?></td>
<td width="7%" align="center" bgcolor="#99CCFF"><? echo "<a
href=\"ubah_data_nilai.php?Kd_Peserta=$row[Kd_Peserta]\"><font
color=green><b>Ubah</a>" ?></td>
<td width="7%" align="center" bgcolor="#99CCFF" onClick="return
hapus()"><? echo "<a
href=\"hapus_data_nilai.php?Kd_Peserta=$row[Kd_Peserta]\"><font
color=red><b>Hapus</a>" ?></td>
</tr>
<?
}
?>
</table>

```



Gambar 4.39 Halaman Data Nilai pada Link administrator

Penggalan Koding Halaman Hapus Data Nilai pada Link administrator

```

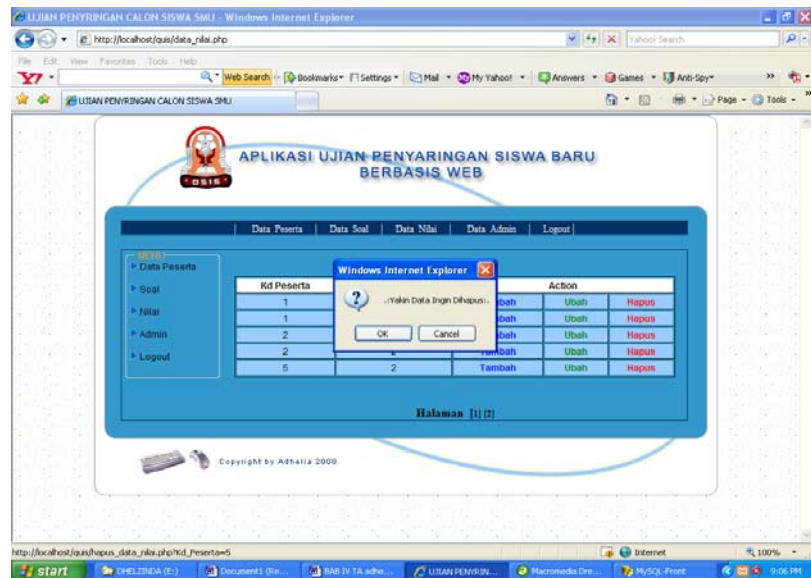
<?
session_start();
?>
<?
if (!isset($Kd_Peserta) && empty($Kd_Peserta)){
echo "Sorry ditolak!";
return 0;
}
include "koneksi.php";
mysql_select_db("$nama_database");
$perintah = mysql_query("delete from dt_hasil where
Kd_Peserta='$Kd_Peserta'");
if (!$perintah)
{
echo "<script>";
echo "alert(\". : Delete Gagal : .\");";
echo "</script>";
include "data_nilai.php";
}

```

```

else
{
echo "<script>";
echo "alert(\". : Delete Sukses : .\");";
echo "</script>";
include "data_nilai.php";
}
?>

```



Gambar 4.40 Halaman Hapus Data Nilai pada Link administrator.



Gambar 4.41 Halaman Pesan jika Data Nilai berhasil di Hapus

Gambar 4.42 adalah tampilan link data admin pada halaman administrator yang dapat digunakan untuk melihat nama administrator yang terdaftar dan mempunyai hak untuk mengakses halaman administrator ini. Berikut adalah penggalan koding serta gambar tampilan halamannya yang dapat dilihat sebagai berikut.

Penggalan Koding Halaman Data Admin pada Link administrator

```

<?php
session_start();
?>
<? include ("Atas-admin.html");?>
<html>
<head>
</head><BODY text=#000000 leftMargin=0 bgproperties="fixed"
background="image/Bgrnd.gif">
<table width="794" height="144" border="0" align="center"
cellpadding="0" cellspacing="0" id="Table_01">
<tr>
<td width="34" height="144"
background="image/index_04.gif">&nbsp;  </td>
<td width="722" align="left" valign="top" bgcolor="#3399CC">
<table width="100%" border="0" cellpadding="0" cellspacing="0"
style="BORDER-RIGHT: #003366 1px solid; BORDER-TOP: #003366 1px
solid; BORDER-LEFT: #003366 1px solid; BORDER-BOTTOM: #003366 1px
solid">
<tr>
<td align="center" valign="top"><center><b>
<span class="judul">
<p><br>
DATA ADMIN</p>
<p></font> </h3>
<script>
function hapus()
{
return confirm(".:Yakin Data Ingin Dihapus.:");
}
</script>
<?
include "koneksi.php";
if (empty($awal))
$awal=0;
$query=mysql_db_query($nama_database,"select * from admin limit
$awal,5");
?>
<? echo "<A href=\"tambah_data_admin.php\"><font
color=blue><b>Tambah</a>" ?> </p>
<table border="1" cellpadding="2" cellspacing="2" style="border-
collapse: collapse" bordercolor="#111111" width="90%"
id="AutoNumber2" class="isi">
<tr> <br>
<td width="24%" align="center" bgcolor="#FFFFFF">
<b>Admin</td>
<td width="46%" align="center" bgcolor="#FFFFFF">
<b>Password</td>
<td colspan="2" align="center" bgcolor="#FFFFFF">
<b>Action</td>
</tr> <?

```




Gambar 4.42 Halaman data admin pada link administrator

Gambar 4.43 adalah tampilan halaman yang akan muncul jika link tambah data admin pada halaman administrator ditekan. Halaman ini akan digunakan jika ada penambahan *account* administrator. Pada halaman ini hanya administrator saja yang mempunyai hak akses untuk dapat masuk serta menambahkan data admin

Penggalan Koding Halaman tambah Data Admin pada Link administrator

```
<?php
session_start();
?>
<? include ("Atas-admin.html");?>
<html>
<head>
</head><BODY text=#000000 leftMargin=0 bgproperties="fixed"
background="image/Bgrnd.gif">
<table width="794" border="0" align="center" cellpadding="0"
cellspacing="0" id="Table_01">
<tr>
<td width="34" height="144"
background="image/index_04.gif">&nbsp;  </td>
<td width="722" align="left" valign="top" bgcolor="#3399CC"><table
border="0" cellpadding="0" cellspacing="0" bordercolor="#111111"
width="100%">
<tr>
<td align="center" valign="top">
<div align="center">
<center>
<form method="post" action="simpan_data_admin.php">
<table border="0" width="400" cellpadding="0" cellspacing="0"
height="86">
<tr>
```

```

<td width="400" height="19"><b> <? echo ("<center><b><font
color=#FF0000 size=4>$msg</font></b></center><br>"); ?>
<center>
<span class="judul">TAMBAH DATA ADMIN</span></td>
<br>
</tr>

```



Gambar 4.43 Halaman tambah data admin pada link administrator

Gambar 4.44 adalah tampilan halaman yang akan muncul jika administrator menekan link untuk mengubah data admin. Halaman ini akan digunakan jika ada data admin yang akan diperbaharui password atau usernamena.

Penggalan Koding Halaman Ubah Data Admin

```

<?
session_start();
include "koneksi.php";
mysql_select_db("$nama_database");
$query="update admin set Admin='$Admins' ,Password='$Passwords' where
id_admin='$id_adm' ";
$hasil=mysql_query($query);
//echo $query;
if($hasil)
{
echo "<script>";
echo "alert(\". : Pengubahan Sukses : .\");";
echo "</script>";
include "data_admin.php";
}
if(!$hasil)
{
echo "<script>";
echo "alert(\". : Pengubahan Gagal : .\");";
echo ("history.go(-1)");
echo "</script>";
}
?>

```



Gambar 4.44 Halaman ubah data admin pada link administrator

Gambar 4.45 adalah tampilan halaman yang akan muncul pada saat tombol untuk menghapus data admin ditekan. Jika ada data admin yang ingin dihapus maka akan muncul pesan seperti yang terlihat pada gambar untuk memastikan apakah admin benar-benar akan menghapus data yang dimaksud.

Gambar 4.46 adalah tampilan halaman pesan yang akan muncul jika data admin berhasil dihapus. Berikut ini adalah penggalan kodingnya dan tampilan halamannya.

Penggalan Koding Halaman hapus Data Soal

```
<?
session_start();
include "koneksi.php";
mysql_select_db("$nama_database");
$perintah = mysql_query("delete from admin where
Password='$Password'");
if(!$perintah)
{
    echo "<script>";
    echo "alert(\". : Delete Gagal : .\");";
    echo "</script>";
    include "data_admin.php";
}
else
{
    echo "<script>";
    echo "alert(\". : Delete Sukses : .\");";
    echo "</script>";
include "data_admin.php"; } ?>
```



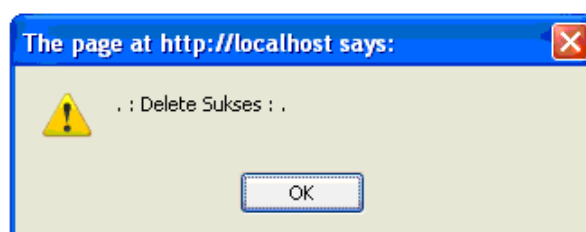
Gambar 4.45 Halaman kotak pesan untuk menghapus data

Penggalan Koding Halaman hapus Data Soal jika berhasil

```

<?
session_start();
include "koneksi.php";
mysql_select_db("$nama_database");
$perintah = mysql_query("delete from admin where
Password='$Password' ");
if(!$perintah)
{
    echo "<script>";
    echo "alert(\". : Delete Gagal : .\");";
    echo "</script>";
    include "data_admin.php";
}
else
{
    echo "<script>";
    echo "alert(\". : Delete Sukses : .\");";
    echo "</script>";
    include "data_admin.php";
}
?>

```



Gambar 4.46 Halaman kotak pesan jika data berhasil dihapus

4.3 Pengujian

Pengujian program aplikasi ini dilakukan dengan metode Black box testing. Metode black box ini merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode black box testing ini adalah untuk menemukan kesalahan fungsi pada program.

Pengujian dengan metode black box testing dilakukan dengan cara memberikan sejumlah masukan (*input*) pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi menghasilkan keluaran (*output*) yang diinginkan dan sesuai dengan fungsi dari program aplikasi tersebut.

Bila dari input yang diberikan proses menghasilkan output yang sesuai dengan kebutuhan fungsionalnya, maka program aplikasi yang bersangkutan telah benar, tetapi bila output yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi tersebut.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan dilakukan secara berulang-ulang. Jika dalam pengujian ditemukan kesalahan, maka akan dilakukan penelusuran dan perbaikan (*debugging*) untuk memperbaiki kesalahan yang terjadi.

Jika telah selesai melakukan perbaikan maka akan dilakukan pengujian kembali. Pengujian dan perbaikan dilakukan secara terus menerus hingga diperoleh hasil yang terbaik.

4.3.1 Skenario Pengujian

Skenario pengujian diperlukan untuk mengetahui apakah menu-menu yang ada dapat berfungsi dengan baik atau tidak. Skenario pengujian dilakukan dengan memilih dari setiap pilihan yang ada pada tampilan aplikasi pendaftaran ini dan tentunya mengantisipasi pendaftaran apabila calon siswa melakukan pendaftaran lebih dari 1kali. Skenario dilakukan satu persatu dalam kondisi yang mungkin terjadi. Skenario tersebut dirangkum dalam Tabel 4.1 skenario pengujian.

Tabel 4.1 Skenario Pengujian

No.	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil yang Di harapkan
1	Menu Daftar Ulang	Normal	Memasukan biodata peserta seperti: Kd_peserta: Username: Password: Jenis kelamin: Alamat: Email:	Menampilkan pesan “Data berhasil diinput”
		Data Salah	Memasukkan biodata peserta salah seperti Username menggunakan koma	Menampilkan pesan kesalahan. “Maaf data yang anda masukkan salah”.
2	Login User	Normal	Memasukan Kd_peserta dan password, seperti adhelia dengan password dhelzz.	Menampilkan halaman peserta

(Lanjutan Tabel 4.1)

		Data Salah	User memasukkan kd_peserta dan password yang tidak sesuai dengan basis data	Menampilkan pesan kesalahan. "data tidak valid".
3	Login admin	Normal	Memasukkan Username dan Password seperti dewi dengan pasword dewi	Menampilkan halaman admin.
		Data salah	Memasukan Username dan Password yang tidak ada didalam basis data seperti dewi dengan Password "dewiq"	Menampilkan halaman login admin dengan pesan "Data tidak valid"
4	Menu halaman Ujian	Normal	Memilih link ujian pada halaman peserta	Menampilkan halaman Modul soal.
5	Halaman soal ujian	Normal	Memasukkan jawaban Dengan menekan Tombol jawab.	Menyimpan jawaban pada database.
		Data salah	Memilih jawaban tapi tidak menekan tombol jawab	Jawaban tidak tersimpan pada database
6	Halaman admin	Normal	Memilih salah satu link halaman data yang akan dilihat datanya	Menampilkan halaman data yang dipilih admin.
7	Halaman Data Peserta	Normal	Mengelola data peserta seperti menambah, jika data telah ditambah maka menekan tombol tambah, mengubah jika data telah	Diklarifikasi bahwa data tersebut telah dikelola dan disimpan pada

(Lanjutan Tabel 4.1)

			diubah maka menekan tombol ubah atau menghapus jika data akan dihapus maka tekan tombol ok pada kotak pesan	database
8	Halaman Data Soal	Data salah	Mengelola data peserta seperti menambah, jika data telah ditambah maka menekan tombol batal, mengubah jika data telah diubah maka menekan tombol batal atau menghapus jika data akan dihapus maka menekan tombol cancel pada kotak pesan	Data tersebut tidak tersimpan pada database
		Normal	Mengelola data Soal seperti menambah, jika data telah ditambah maka menekan tombol tambah, mengubah jika data telah diubah maka menekan tombol ubah atau menghapus jika data akan dihapus maka tekan tombol ok pada kotak pesan	Diklarifikasi bahwa data tersebut telah dikelola dan disimpan pada database..
9.	Halaman Data Nilai	Data salah	Mengelola data soal seperti menambah, jika data telah ditambah maka menekan tombol batal, mengubah jika data telah diubah maka menekan tombol batal atau menghapus jika data akan dihapus maka menekan	Data tersebut tidak tersimpan pada database

(Lanjutan Tabel 4.1)

			tombol cancel pada kotak pesan	
.		Normal	Mengelola data Nilai seperti menambah, jika data telah ditambah maka menekan tombol tambah, mengubah jika data telah diubah maka menekan tombol ubah atau menghapus jika data akan dihapus maka tekan tombol ok pada kotak pesan	Diklarifikasi bahwa data tersebut telah dikelola dan disimpan pada database.
		Data salah	Mengelola data Nilai seperti menambah, jika data telah ditambah maka menekan tombol batal, mengubah jika data telah diubah maka menekan tombol batal atau menghapus jika data akan dihapus maka menekan tombol cancel pada kotak pesan	Data tersebut tidak tersimpan pada database
10.	Halaman Data Admin	Normal	Mengelola data admin seperti menambah, jika data telah ditambah maka menekan tombol tambah, mengubah jika data telah diubah maka menekan tombol ubah atau menghapus jika data akan dihapus maka tekan tombol ok pada kotak pesan	Diklarifikasi bahwa data tersebut telah dikelola dan disimpan pada database..

(Lanjutan Tabel 4.1)

		Data salah	Mengelola data admin seperti menambah, jika data telah ditambah maka menekan tombol batal, mengubah jika data telah diubah maka menekan tombol batal atau menghapus jika data akan dihapus maka menekan tombol cancel pada kotak pesan	Data tersebut tidak tersimpan pada database
--	--	------------	--	---

4.3.2 Hasil Pengujian

Dari skenario pengujian terakhir yang dilakukan dapat diperoleh hasil pengujian. Dari hasil pengujian pada menu setiap pilihan yang ada pada aplikasi ini yaitu tampilan login user, dan menu update data pada aplikasi program ini dan terangkum dalam tabel hasil pengujian. Lihat tabel 4.2 tabel hasil pengujian.

Tabel 4.2 Hasil Pengujian

No.	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil yang Di harapkan	Hasil Uji
1	Menu Daftar Ulang	Normal	Memasukan biodata peserta seperti: Kd_peserta: Username: Password: Jenis kelamin: Alamat: Email:	Menampilkan pesan "Data berhasil diinput"	Sesuai

(Lanjutan Tabel 4.2)

		Data Salah	Memasukkan biodata peserta salah seperti Username menggunakan koma	Menampilkan pesan kesalahan. "Maaf data yang anda masukkan salah".	Sesuai
2	Login User	Normal	Memasukan Kd_peserta dan password, seperti adhelia dengan password dhelzz.	Menampilkan halaman peserta	Sesuai
		Data Salah	User memasukkan kd_peserta dan password yang tidak sesuai dengan basis data	Menampilkan pesan kesalahan. "data tidak valid".	Sesuai
3	Login admin	Normal	Memasukkan Username dan Password seperti dewi dengan pasword dewi	Menampilkan halaman admin.	Sesuai
		Data salah	Memasukan Username dan Password yang tidak ada didalam basis data seperti dewi dengan Password "dewiq"	Menampilkan halaman login admin dengan pesan "Data tidak valid"	Sesuai
4	Menu halaman Ujian	Normal	Memilih link ujian pada halaman peserta	Menampilkan halaman Modul soal.	Sesuai

(Lanjutan Tabel 4.2)

5	Halaman soal ujian	Normal	Memasukkan jawaban Dengan menekan Tombol jawab.	Menyimpan jawaban pada database.	Sesuai
		Data salah	Memilih jawaban tapi tidak menekan tombol jawab	Jawaban tidak tersimpan pada datase	Sesuai
6	Halaman admin	Normal	Memilih salah satu link halaman data yang akan dilihat datanya	Menampilkan halaman data yang dipilih admin.	Sesuai
7	Halaman Data Peserta	Normal	Mengelola data peserta seperti menambah, jika data telah ditambah maka menekan tombol tambah, mengubah jika data telah diubah maka tekan tombol ubah atau menghapus jika data akan dihapus maka tekan tombol ok pada kotak pesan	Mengklarifikasi bahwa data tersebut telah dikelola dan disimpan pada database..	Sesuai
		Data salah	Mengelola data peserta seperti menambah, jika data telah ditambah maka menekan tombol	Data tersebut tidak tersimpan pada database	Sesuai

(Lanjutan Tabel 4.2)

			batal, mengubah jika data telah diubah maka menekan tombol batal atau menghapus jika data akan dihapus maka menekan tombol cancel pada kotak pesan		
8	Halaman Data Soal	Normal	Mengelola data Soal seperti menambah, jika data telah ditambah maka menekan tombol tambah, mengubah jika data telah diubah maka menekan tombol ubah atau menghapus jika data akan dihapus maka tekan tombol ok pada kotak pesan	Mengklarifikasi bahwa data tersebut telah dikelola dan disimpan pada database..	Sesuai
		Data salah	Mengelola data soal seperti menambah, jika data telah ditambah maka menekan tombol batal, mengubah jika data telah	Data tersebut tidak tersimpan pada database	Sesuai

(Lanjutan Tabel 4.2)

			diubah maka menekan tombol batal atau menghapus jika data akan dihapus maka menekan tombol cancel pada kotak pesan		
9	Halaman Data Nilai	Normal	Mengelola data Nilai seperti atau menghapus jika data akan dihapus maka tekan tombol ok pada kotak pesan	Mengklarifikasi bahwa data tersebut telah dihapus.	Sesuai
		Data salah	Mengelola data Nilai seperti menghapus jika data akan dihapus maka menekan tombol cancel pada kotak pesan	Data tersebut tidak dihapus	Sesuai
10.	Halaman Data Admin	Normal	Mengelola data admin seperti menambah, jika data telah ditambah maka menekan tombol tambah, mengubah jika data telah diubah maka menekan tombol ubah atau menghapus jika	Mengklarifikasi bahwa data tersebut telah dikelola dan disimpan pada database.	Sesuai

(Lanjutan Tabel 4.2)

			data akan dihapus maka tekan tombol ok pada kotak pesan		
		Data salah	Mengelola data admin seperti menambah, jika data telah ditambah maka menekan tombol batal, mengubah jika data telah diubah maka menekan tombol batal atau menghapus jika data akan dihapus maka menekan tombol cancel pada kotak pesan	Data tersebut tidak tersimpan pada database	Sesuai

4.3.3 Analisis Hasil Pengujian

Pengujian perangkat lunak telah dilakukan. Analisis dari hasil pengujian perangkat lunak yaitu :

Pada menu daftar ulang, kasus uji yang dijalankan yaitu memasukan data peserta, seperti Username, Nama, Password, Kd_Peserta, Alamat, Email. dengan kelompok uji normal yang artinya semua data yang diperlukan harus dimasukan, yang akan menghasilkan tampilan halaman login user. Sedangkan jika kelompok uji data tidak normal, akan menghasilkan tampilan pesan kesalahan. Modul ini

telah dilakukan pengujian dan hasilnya diterima atau dapat dijalankan sesuai dengan rancangan yang telah dibangun.



Gambar 4.4 Tampilan Halaman jika Daftar Ulang tidak berhasil

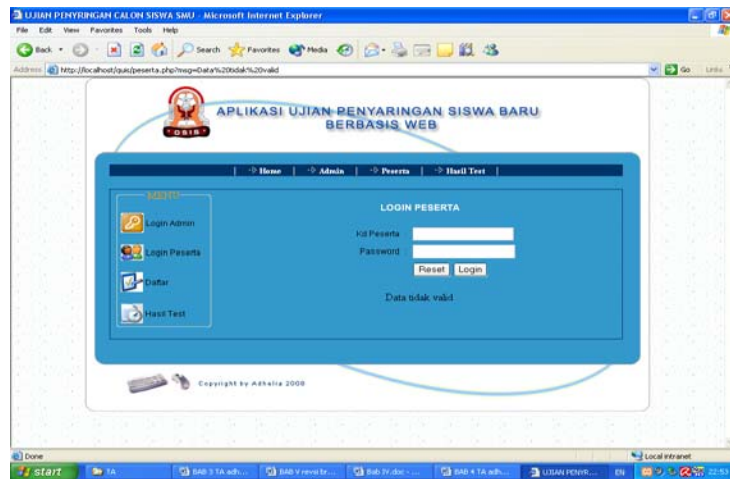
Pada menu login, kasus uji yang di masukan yaitu memilih login user dan atau memilih login admin, dengan kelompok uji normal yang artinya dijalankan dengan semestinya, yang akan menghasilkan tampilan halaman login user dan atau halaman login admin. Modul ini telah dilakukan pengujian dan hasilnya diterima atau dapat dijalankan sesuai dengan rancangan yang telah dibangun. Analisis pada menu login juga sama dengan analisis pada deskripsi fungsional halaman utama, dan halaman admin.

Pada login user, kasus uji yang dijalankan yaitu memasukan kd_peserta dan *username*, seperti nama user adhelia dengan *password* adhel dengan kelompok uji normal yang artinya memasukan data yang benar, yang akan menghasilkan tampilan halaman utama.

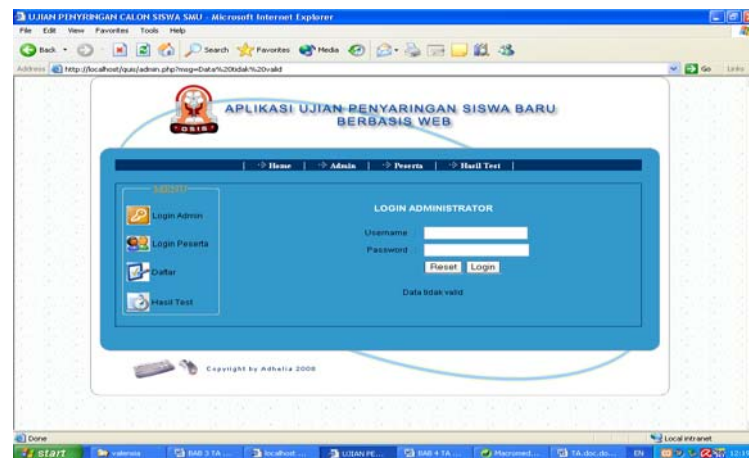
Sedangkan pada login admin dijalankan dengan kelompok uji data tidak normal yang artinya memasukan data yang salah, sehingga menghasilkan tampilan pesan kesalahan pada halaman login admin. Modul ini telah dilakukan pengujian dan hasilnya diterima atau dapat dijalankan sesuai dengan rancangan yang telah dibangun.



Gambar 4.5 Tampilan halaman Login peserta

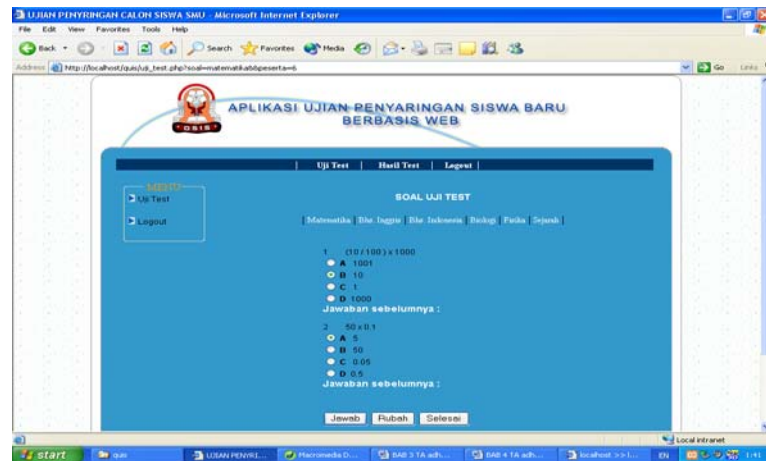


Gambar 4.6 Tampilan halaman Kesalahan Login peserta



Gambar 4.18 menunjukkan tampilan data salah pada menu login admin

Pada halaman soal, kasus uji yang dijalankan yaitu setelah peserta menjawab semua soal pada modul maka harus menekan tombol jawab agar sistem dapat menyimpan jawaban pada *database*. Jika sudah dilakukan maka akan keluar pesan bahwa data berhasil diinput, dan kelompok uji data tidak normal yaitu setelah peserta menjawab semua soal pada modul dia tidak menekan tombol jawab maka data jawaban tidak disimpan pada *database*. Modul ini telah dilakukan pengujian dan hasilnya diterima atau dapat dijalankan sesuai dengan rancangan yang telah dibangun.



Gambar 4.9 Tampilan Halaman soal ujian pada modul



Gambar 4.10 Tampilan Halaman jika peserta menekan tombol jawab

4.4 Pengujian White Box

Dikenal juga dengan nama glass box, structural, clear box dan open box testing. Merupakan tehnik testing perangkat lunak yang harus mengetahui secara detail tentang perangkat lunak yang akan diuji. Uji coba white box adalah metode perancangan test case yang menggunakan struktural kontrol dari perancangan prosedural untuk mendapatkan test case. Dengan menggunakan metode white box, analisis sistem akan dapat diperoleh tset case yang:

1. Menjamin seluruh independent path didalam modul akan dikerjakan sekurang-kurangnya satu kali.
2. Mengerjakan seluruh keputusan logikal.
3. Mengerjakan seluruh loop yang sesuai dengan batasannya.

4.4.1 Pengujian white box pada proses login

Berikut ini adalah pengujian white box pada halaman login peserta.

1. Session start();
2.

```
<form method="post" action="peserta_login.php">
<table border="0" cellpadding="2" cellspacing="2" style="border-
collapse: collapse" bordercolor="#111111"
width="106%" id="AutoNumber2" class="isi"><tr>
<td width="40%" align="right">Kd Peserta </td>
<td width="3%" align="center">:</td>
<td width="57%" align="center"><input name="Kd_Peserta"
size="20" style="float: left"></p></td>
</tr>
<tr>
<td width="40%" align="right">Password </td>
<td width="3%" align="center">:</td>
<td width="57%" align="center"><input type="password"
name="Password" size="20" style="float: left"></td> </tr>
<tr>
<td align="center"><p>
<p align="center"></td>
<td align="center">&nbsp;</td>
<td align="left"><input name="reset" type="reset" value="Reset">
<input name="Submit" type="Submit" value="Login"></td></tr>
</table>
</form>
```

```

3. if (($Kd_Peserta=='' ) or (ereg(";", $Kd_Peserta)) or
    (ereg("'", $Kd_Peserta))) {
    $msg="Harap diisi dengan benar";
    $alamat="peserta.php";
4. header("Location:$alamat?msg=$msg");
   exit();
5. } elseif (($Password=='' ) or (ereg(";", $Password)) or
    (ereg("'", $Password))) {
    $msg="Harap diisi dengan benar";
    $alamat="peserta.php";
6. header("Location:$alamat?msg=$msg");
   exit();
   }
7. include "koneksi.php";
   mysql_select_db("$nama_database");
8. $numresult=mysql_query("select Kd_Peserta,Password from
   dt_peserta where Kd_Peserta='$Kd_Peserta' and
   Password='$Password'");
   $cek=mysql_num_rows($numresult);
9. if ($cek==0) {
   $msg="Data tidak valid";
   $alamat="peserta.php";
   header("Location:$alamat?msg=$msg");
   exit();
   }
10. else {
   $hasil=mysql_query("select Kd_Peserta,Password from dt_peserta
   where Kd_Peserta='$Kd_Peserta' and Password='$Password'");
   $baris=mysql_fetch_row($hasil);
   $Kd_Peserta=$baris[0];
   $Password=$baris[1];

```

Jalur logika (path) yang terjadi pada halaman login terlihat seperti terlihat

dibawah ini:

Path 1: 1-2-3-10

Path2 : 1-2-3-4-10

Path3: 1-2-3-4-5-6-7-8

Path4: 1-2-3-4-5-6-10

Path5: 1-2-3-4-5-6-7-10

Path6: 1-2-3-4-5-6-7-9-10

4.4.2 Pengujian white box pada proses mengerjakan soal

Pengujian white box pada proses mengerjakan soal yang diawali dengan menampilkan soal yang ada pada bsisdata. Berikut ini pengujian nya.

- ```
$querys=mysql_db_query($nama_database,"select * from soal where
module='$soal'");
$b=1;
while ($row=mysql_fetch_array ($querys)) {
//$jmls = $row[Id_Pertanyaan];
$Id_Pertanyaan[$b]=$row[0];
$Pertanyaan[$b]=$row[3];
$JawabA[$b]=$row[4];
$JawabB[$b]=$row[5];
$JawabC[$b]=$row[6];
$JawabD[$b]=$row[7];
$no_Pertanyaan[$b]=$row[1];
$module[$b]=$row[2];
?>
```
- ```
<table width="50%" border="0" align="center" cellpadding="0"
cellspacing="0" bordercolor="#111111" class="isi"
id="AutoNumber1" style="border-collapse: collapse">
<tr> <td width="10%" valign="top">
<?
echo "$no_Pertanyaan[$b]";
?>
<input type="hidden" name="Id_Pertanyaans[<? echo $b; ?>]"
value="<? echo $Id_Pertanyaan[$b]; ?>" size="20">
<input type="hidden" name="modules" value="<? echo $module[$b];
?>" size="20"></td>
```
- ```
<td width="90%" valign="top"><? echo "$Pertanyaan[$b]" ?></td>
</tr>
<tr valign="top">
<td colspan="2">
<?
?>
```
- ```
$query3="select * from jawab where Kd_Peserta='$peserta' and
Id_Pertanyaan='$Id_Pertanyaan[$b]' and module='$soal'";
result3=mysql_db_query($nama_database,$query3);
//echo $query3;
```
- ```
while ($row=mysql_fetch_array ($result3)) {
$Kd_Peserta3=$row[0];
$Id_Pertanyaan3=$row[1];
$jawab=$row[2];
$keterangan3=$row[3];
$stgl_test3=$row[4];
//echo $jawab;
6. if($jawab == "A") { $checked1="checked"; }
//if($jawab != "A") { $checked=""; }
7. if($jawab == "B") { $checked2="checked"; }
//if($jawab != "B") { $checked=""; }
8. if($jawab == "C") { $checked3="checked"; }
//if($jawab != "C") { $checked=""; }
9. if($jawab == "D") { $checked4="checked"; }
//if($jawab != "D") { $checked=""; }
}
?>
```

```
10. <td colspan="2"><? echo "Jawaban
sebelumnya : ".$jawab.""; ?></td>
```

Jalur logika (path) yang terjadi pada proses melihat nilai dapat dilihat dari daftar path yang ada dibawah ini:

Path 1: 1-2-3-4-5-6

Path 2 : 1-2-3-7

#### **4.4.3 Analisis Hasil Pengujian**

Dari hasil pengujian diatas dapat dianalisa:

1. Aplikasi ini dapat terkoneksi dengan basis data tdk ada kesalahan hal ini terlihat pada halaman aplikasi yang mengambil data dari basis data, data yang ada didalam basis data dapat ditampilkan sesuai perintah program.
2. Pada saat proses ujian yang ada didalam basis data dapat ditampilkan sesuai dengan yang diinginkan, dan sistem dapat memproses nilai siswa dengan benar.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Demikian pembahasan mengenai pembangunan aplikasi Ujian penyaringan siswa baru untuk sekolah menengah umum berbasis web beserta alternatif masalah yang diusulkan oleh penulis, yang secara umum dapat disimpulkan sebagai berikut :

1. Dari hasil pengujian didapatkan, jika peserta memasukkan kode peserta yang sudah terdaftar maka peserta tidak bisa melakukan daftar ulang. Dan pada hasil ujian lainnya didapatkan bahwa apabila peserta sudah menjawab soal pada modul dan menekan tombol selesai maka dia tidak bisa melihat halaman soal lagi, karena halaman tersebut diblok untuk peserta yang telah mengikuti ujian, yang dapat peserta akses setelah mengikuti ujian hanya halaman hasil ujian saja.

2. Halaman pada perancangan aplikasi ini sudah dapat berjalan sesuai dengan fungsinya, contohnya : Halaman modul soal menampilkan soal-soal yang akan dikerjakan oleh peserta ujian.
3. Pada rancangan situs ujian berbasis web ini mampu menyaring calon siswa baru sesuai dengan standarisasi nilai yang ditetapkan dari pihak sekolah yang terkait.
4. Pemanfaatan aplikasi ujian berbasis web ini mampu mengurangi tingkat kesalahan yang terjadi pada proses penilaian.

## **5.2 Saran**

Saran yang diajukan untuk mengembangkan aplikasi ini dimasa mendatang antara lain:

1. Pada aplikasi ini perlu diupayakan untuk lebih meminimalisasi tingkat kecurangan ujian oleh calon siswa maka harus dibuatkan bank data soal.
2. Perlu ditambahkan modul cara penggunaan aplikasi bagi peserta ujian dan administrator.
3. Pengawas ujian gunakan sistem admin pada aplikasi untuk melakukan pengaturan dan memantau jalannya ujian.



## DAFTAR PUSTAKA

Pressman, R.S. *Rekayasa Perangkat Lunak Pendekatan Praktisi II*. Yogyakarta: Andi, 2004.

Munawar. 2005. *Pemodelan Visual dengan UML* Yogyakarta: Graha Ilmu, 2005

Kadir, Abdul. *Konsep & Tuntunan Praktis Basis Data* EDISI II. Yogyakarta: Penerbit Andi, 2003.

Whitten, J. L. Bentley, L.D. dan Dittman, K.C. *Systems Analysis and Design Methods* Sixth Edition. McGraw-Hill, 2004.

Raghu Ramakrishnan dan Johannes Gehrke. *Database Management Systems*, McGraw Hill, 2002.

Rademacher, R.A. dan Gibson, H.L. *An Introduction to Computers and Information Systems*, South Western Publishing co, 1983.

Turban, Efraim dan Aronson, Jay E. *Decision Support Systems and Intelligence System* Sixth Edition, Prentice-Hall, Inc, Upper Saddle River, New Jersey, 2001.

Vita Prihatoni Purnomo. *Macromedia Dreamweaver MX*. Jakarta : Penerbit PT. Dian Rakyat, 2005.

Connolly Thomas M dan Begg Carolyn E, *Database Systems: A Practical Approach to Design, Implementation, and Management* 4th Edition, Addison Wesley, 2005.

Welling, Luke, dan Thomson, Laura. *PHP and Mysql Web Development*. PDF [www.php.net/manual/en/2004](http://www.php.net/manual/en/2004)