

**LATIHAN PSIKOTES DENGAN MENGGUNAKAN PHP DAN
MYSQL**

Laporan Tugas Akhir

**Diajukan untuk melengkapi persyaratan
mencapai gelar sarjana strata satu (S1)
Program Studi Teknik Informatika**

oleh

ZOGY PANJAITAN

NIM 01503-073



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MERCU BUANA

JAKARTA

2009

ABSTRACT

A company which requiring officer, will do officer selection by using psikotest. All applicant having an interest in the company will visit upon the company to follow psikotest although with pass through over which enough far.

In line with its fast progress in technology and science, hence the computer role as a means of assist a lot of felt benefit. Science applying and technological become consideration in develop a application web, to get maximal result.

To overcome the problem, required a application of psikotes in order to can be accessed anywhere place all applicant reside in, so that all direct applicant can follow psikotest.

Without need come direct a place of company which is gone to and can cost effective to all applicant because do not expense for transport car.

Applicant also can train how big and ability knowledge in psikotes melakukan, so that applicant can know do him accepted in company or not.

Keyword: Teknologi, Application Web, Psikotest.

ABSTRAK

Suatu perusahaan yang sedang membutuhkan pegawai, akan melakukan seleksi pegawai dengan menggunakan psikotes. Para pelamar yang berminat pada perusahaan tersebut akan mendatangi perusahaan tersebut untuk mengikuti psikotes walaupun dengan menempuh jarak yang cukup jauh.

Sejalan dengan pesatnya kemajuan dalam ilmu pengetahuan dan teknologi, maka peranan komputer sebagai alat bantu banyak sekali dirasakan manfaatnya. Penerapan ilmu dan teknologi menjadi bahan pertimbangan dalam membangun sebuah aplikasi web, untuk mendapatkan hasil yang maksimal.

Untuk mengatasi masalah tersebut, dibutuhkan suatu aplikasi psikotes agar dapat diakses dimanapun tempat para pelamar berada, sehingga para pelamar langsung bisa mengikuti psikotes.

Tanpa perlu datang langsung ketempat perusahaan yang dituju dan dapat menghemat biaya bagi para pelamar karena tidak mengeluarkan biaya untuk transport mobil.

Para pelamar juga dapat melatih seberapa besar kemampuan dan pengetahuan diri dalam melakukan psikotes, sehingga para pelamar dapat mengetahui apakah dia diterima dalam perusahaan atau tidak.

Kata Kunci: Teknologi, Aplikasi Web, Psikotes.

LEMBAR PENGESAHAN

Nama : Zogy Panjaitan
NIM : 01503-073
Fakultas : Ilmu Komputer
Program Studi : Teknik Informatika

Judul Tugas Akhir : **Latihan Psikotes Dengan Menggunakan PHP dan MYSQL**

Yang bertanda tangan dibawah ini, menyatakan bahwa Tugas Akhir dari mahasiswa tersebut di atas, telah disahkan dalam sidang Tugas Akhir Program Studi Teknik Informatika.

Jakarta,..... Januari 2009

Menyetujui,

Menyetujui,

(Ir. Nixon Erzed. MT)
Pembimbing I Tugas Akhir

(Sarwati Rahayu, ST., MMSI)
Pembimbing II Tugas Akhir

Mengesahkan,

Mengetahui,

(Abdusy Syarif, ST., MT)
Ketua Program Studi
Teknik Informatika

(Devi Fitriyah, ST., MTi)
Koordinator Tugas Akhir

KATA PENGANTAR

Puji dan Syukur kepada Tuhan Yesus Kristus.

Maksud dan tujuan dari penyusunan tugas akhir atau skripsi dengan mengambil judul **"LATIHAN PSIKOTES DENGAN MENGGUNAKAN PHP DAN MYSQL"** adalah untuk melengkapi salah satu syarat dalam kelulusan untuk jenjang Srata 1 untuk program studi Teknik Informatika.

Dalam penyusunan Tugas Akhir ini, penulis berusaha semaksimal mungkin untuk mengatasi kesulitan dan hambatan yang ada mengingat keterbatasan kemampuan penulis. Namun berkat bantuan dari beberapa pihak kesulitan-kesulitan tersebut dapat diatasi, sehingga memungkinkan penulis untuk menyelesaikan Tugas Akhir. Atas segala bantuannya, penulis tidak lupa menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Orang Tua & Kakak – kakak yang aku Cintai, yang telah memberikan dorongan moril dan materil kepada penulis.
2. Bapak Abdusy Syarif, ST, MT selaku Ketua Program Studi Teknik Informatika.
3. Bapak Ir. Nixon Erzed, MT selaku Pembimbing I Tugas Akhir yang telah banyak membantu, membimbing, mengarahkan, dan memberikan saran–saran kepada penulis.
4. Ibu Sarwati Rahayu, ST., MMSI sebagai Pembimbing II Tugas Akhir. Tugas Akhir, yang telah banyak membantu, membimbing, mengarahkan, dan memberikan saran-saran kepada penulis.
5. Devi Fitriannah, ST., MTi., sebagai Koordinator Tugas Akhir.

6. Dosen-dosen Universitas Mercu Buana, khususnya dosen program studi teknik informatika yang telah mendidik dan mengarahkan kepada penulis
7. Moch. Jatra yang telah memberikan dorongan moril dan membantu hingga Tugas Akhir ini selesai dibuat.
8. M. Khairudin yang telah memberikan dorongan moril hingga Tugas Akhir ini selesai dibuat.
9. Rekan–rekan mahasiswa Teknik Informatika khususnya angkatan 2003 yang tidak dapat disebutkan satu persatu, yang telah memberikan bantuan secara langsung maupun tidak langsung selama penulisan Tugas Akhir ini.

Penulis juga menyadari bahwa dalam penulisan Tugas Akhir ini masih banyak kesalahan dan kekurangan, maka kritik dan saran yang membangun sangat penulis harapkan. Semoga Tugas Akhir ini dapat memberikan manfaat yang sangat berguna untuk rekan–rekan mahasiswa/i.

Jakarta, Januari 2009

Penulis

(Zogy Panjaitan)

DAFTAR ISI

Abstract	i
Abstrak	ii
Lembar Pengesahan	iii
Kata Pengantar	iv
Daftar Isi	vi
Daftar Gambar	x
Daftar Tabel	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penulisan	2
1.3 Batasan Masalah	2
1.4 Metodologi	2
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI	7
2.1 Psikotes	7
2.2 Pengertian Sistem	10
2.3 Pengertian Informasi	11
2.4 Pengertian Sistem Informasi	12
2.5 Metodologi Rekayasa Perangkat Lunak (RPL)	13
2.6 <i>Unified Modelling Language</i> (UML)	15
2.6.1 Bekerja Dengan Model & Diagram	15

2.7	Perancangan Data Base	30
2.7.1	Proses Perancangan Data Base	30
2.7.2	Perancangan Data Base Secara Logika	36
2.7.2.1	Model Data Relasional	37
2.7.2.2	Batasan Integritas	37
2.7.2.3	Mentranformasi Diagram E-R Ke Relasi	38
2.7.2.4	Normalisasi	38
2.7.3	Perancangan Data Base Secara Fisik	41
2.7.4	Kamus Data	43
2.7.5	Isi Kamus Data	45
2.7.6	<i>Entity Relationship</i> Diagram (ERD)	46
2.8	Web	49
2.9	Konsep Dasar PHP	49
2.9.1	Aplikasi PHP dengan MySQL	50
2.10	Dreamweaver	55
BAB III ANALISIS DAN PERANCANGAN SISTEM		56
3.1	Analisis	57
3.1.1	Proses yang terdapat dalam latihan psikotes	61
3.1.2	Keluaran Sistem	61
3.1.3	Diagam konteks	61
3.2	Rancangan Sistem	62
3.2.1	<i>Rancangan Unified Modeling Language (UML) ...</i>	63
3.2.1.1	Diagram Use Case	63
3.2.2	<i>Entity Relationship diagram (ERD)</i>	68

3.2.3	Struktur Tabel	70
3.2.4	Diagram Aktifitas	72
3.2.5	Diagram Sequence	76
3.2.6	Rancangan Interface	78
BAB IV IMPLEMENTASI DAN PENGUJIAN		88
4.1	Implementasi	88
4.1.1	Implementasi Basis Data	88
4.1.1.1	Implementasi Basis Data Tb_admin	88
4.1.1.2	Implementasi Basis Data Tb_soal	89
4.1.1.3	Implementasi Basis Data Tb_peserta	90
4.1.1.4	Implementasi Basis Data Tb_hasil	91
4.1.2	Implementasi Perancangan Layar	92
4.1.2.1	Implementasi Perancangan Layar Register ...	92
4.1.2.2	Implementasi Perancangan Login Peserta	94
4.1.2.3	Implementasi Perancangan Layar Soal	96
4.1.2.4	Implementasi Perancangan Layar Hasil	97
4.1.2.5	Implementasi Perancangan Login Admin	99
4.1.2.6	Implementasi Perancangan Layar Admin	100
4.2	Pengujian	102
4.2.1	Lingkungan Pengujian	102
4.2.2	Skenario Pengujian	103
4.2.3	Dokumen Hasil Pengujian	104
4.2.4	Analisis Hasil Pengujian	105

BAB V KESIMPULAN DAN SARAN	112
5.1 Kesimpulan	112
5.2 Saran	113
DAFTAR PUSTAKA	114
LAMPIRAN	L- 1 -

DAFTAR GAMBAR

Gambar 2.1 Skema Model Air Terjun	13
Gambar 2.2 Notasi <i>activity</i> diagram	17
Gambar 2.3 Class Diagram	20
Gambar 2.4 Communication diagram contoh Kasus	21
Gambar 2.5 Diagram Komponen	21
Gambar 2.6 Composite Diagram	22
Gambar 2.7 Deployment Diagram gallery VCD	23
Gambar 2.8 Interaction Diagram VCD Gallery	24
Gambar 2.9 Objek Diagram	25
Gambar 2.10 Contoh Package diagram	26
Gambar 2.11 Notasi <i>sequence</i> diagram	27
Gambar 2.12 Simbol Statechart Diagram	28
Gambar 2.13 Timming Diagram	28
Gambar 2.14 Notasi <i>use case</i> diagram	29
Gambar 2.15 Notasi Kamus Data	44
Gambar 2.16 Penggambaran ERD	47
Gambar 2.17 Interface PHP & MySQL	54
Gambar 2.18 Interface Dreamweaver 8	55
Gambar 3.1 Diagram Konteks	62
Gambar 3.2 Diagram use case aplikasi psikotes	63
Gambar 3.3 Relasi antar admin dengan soal psikotes	68
Gambar 3.4 Relasi antar Soal psikotes, Peserta dan Hasil	68

Gambar 3.5 Model data fisik aplikasi psikotes	69
Gambar 3.6 Diagram aktifitas register	72
Gambar 3.7 Diagram aktifitas login	73
Gambar 3.8 Diagram aktifitas tes psikotes	74
Gambar 3.9 Diagram aktifitas hasil psikotes	75
Gambar 3.10 Diagram aktifitas admin soal psikotes	75
Gambar 3.11 Diagram sequence aplikasi psikotes	76
Gambar 3.12 Diagram sequence <i>update</i> soal	77
Gambar 3.13 Rancangan Halaman Utama	78
Gambar 3.14 Rancangan Halaman Register Bagi Para Peserta	79
Gambar 3.15 Rancangan halaman <i>log in</i>	80
Gambar 3.16 Rancangan Halaman Salah	80
Gambar 3.17 Rancangan Halaman Soal	81
Gambar 3.18 Rancangan Halaman Selesai	81
Gambar 3.19 Rancangan Halaman Hasil	82
Gambar 3.20 Perancangan Halaman <i>Login</i> Admin	82
Gambar 3.21 Rancangan Halaman Utama	83
Gambar 3.22 Rancangan Halaman Input Soal	84
Gambar 3.23 Rancangan Data Peserta	84
Gambar 3.24 Rancangan Halaman Data Nilai	85
Gambar 3.25 Rancangan Halaman Data Soal	86
Gambar 3.26 Rancangan Halaman Edit Data Soal	86
Gambar 3.27 Rancangan Halaman Rubah Username Admin	87
Gambar 4.1 Implementasi Pembuatan Basis Data Tabel Admin	89
Gambar 4.2 Implementasi Pembuatan Basis Data Tabel Soal	90

Gambar 4.3 Implementasi Pembuatan Basis Data Tb_peserta	91
Gambar 4.4 Implementasi Pembuatan Basis Data Tb_hasil	91
Gambar 4.5 Implementasi Perancangan Layar Register	94
Gambar 4.6 Implementasi Perancangan Layar Login Peserta	95
Gambar 4.7 Implementasi Perancangan Layar Soal	97
Gambar 4.8 Implemenasi Perancangan Layar Hasil	98
Gambar 4.9 Implementasi Perancangan Layar Login Admin	100
Gambar 4.10 Implementasi Perancangan Layar Admin	101
Gambar 4.11 Hasil Pengujian Halaman Register	105
Gambar 4.12 Hasil Pengujian Halaman Login Peserta	106
Gambar 4.13 Hasil Pengujian Halaman Soal	107
Gambar 4.14 Hasil Pengujian Halaman Login Admin	108
Gambar 4.15 Hasil Pengujian Halaman Input Soal	109
Gambar 4.16 Hasil Pengujian Halaman Data Soal	110
Gambar 4.17 Hasil Pengujian Halaman Data peserta	110
Gambar 4.18 Hasil Pengujian Proses Logout	111

DAFTAR TABEL

Tabel 2.1 Jenis diagram resmi UML	16
Tabel 3.1 Perancangan tabel peserta	70
Tabel 3.2 Perancangan tabel soal	70
Tabel 3.3 Perancangan tabel hasil	71
Tabel 3.4 Perancangan tabel admin	71
Tabel 4.1 Skenario Pengujian Perangkat lunak	104

BAB I

PENDAHULUAN

1.1 Latar Belakang

Definisi Tes Psikologi atau lebih dikenal sebagai Psikotes adalah tes untuk mengukur aspek individu secara psikis. Tes dapat berbentuk tertulis, visual, atau evaluasi secara verbal yang teradministrasi untuk mengukur fungsi kognitif dan emosional, tes dapat diaplikasikan kepada anak-anak maupun dewasa.

Tujuan Tes Psikologi digunakan untuk mengukur berbagai kemungkinan atas bermacam kemampuan secara mental dan apa-apa yang mendukungnya, termasuk prestasi dan kemampuan, kepribadian, intelegensi, atau bahkan fungsi neurologis.

Aplikasi Tes Psikologi dapat dilakukan pada bermacam setting termasuk rekrutmen dalam perusahaan, mengetahui minat dan bakat anak / siswa, tujuan klinis, perkembangan anak, atau kustomisasi design dan modul dalam pelatihan / training.

Latihan psikotes merupakan hal yang penting untuk mengavaluasi atau menilai bakat, kemampuan, minat, sifat-sifat, dan kemampuan yang dimiliki seseorang. Dalam kenyataannya latihan psikotes sering dianggap sangat sulit dan sering dikeluhkan oleh sebagian besar peserta latihan psikotes, terutama bagi mereka yang baru pertama kali mengikutinya.

Seseorang yang berpendidikan tinggi, tidak dijamin dapat menyelesaikan latihan psikotes dengan baik. Hal ini disebabkan karena latihan psikotes bukan semata-mata mengukur tingkat kecerdasan atau intelektual seseorang. Banyak faktor

yang mempengaruhi keberhasilan atau kegagalan seseorang dalam menyelesaikan tes psikotes ini. Salah satunya adalah faktor mental, misalkan tegang dalam menghadapi soal, dan hal lain yang dapat mengganggu konsentrasi.

Anggapan bahwa tes psikotes merupakan hal yang sulit tidaklah benar, karena soal-soal tes psikotes dibuat mudah dan sederhana. Persoalannya adalah para peserta tes kurang memahami maksud dari soal-soal yang diberikan.

Untuk dapat memahami soal-soal latihan psikotes para peserta disarankan lebih banyak berlatih, oleh karena itu penulis ingin mencoba membuat latihan psikotes dengan menggunakan PHP dan MYSQL, yang merupakan pengembangan dari aplikasi yang sudah ada sebelumnya.

1.2 Tujuan Penulisan

Tujuan dari penulisan tugas akhir ini adalah :

1. Membangun sebuah aplikasi sebagai alat bantu dalam menghadapi soal-soal dalam bentuk pilihan ganda untuk peserta psikotes.
2. Menawarkan fitur yang lebih baik seperti tampilan lebih mudah digunakan.
3. Memberikan nilai yang berbentuk grafik dalam latihan psikotes.

1.3 Batasan Masalah

1. Membahas mengenai rancangan web site Latihan Psikotes.
2. Dalam membangun aplikasi, penulis menggunakan Phptriad digunakan untuk menjalankan aplikasi, PhpMyAdmin sebagai basis data, Macromedia Dreamweaver 8 sebagai *user interface*, dan adobe photoshop 7.0, swish max, dan flash mx 2004 sebagai aplikasi tambahan.

3. Aplikasi yang dibangun hanya memuat tentang soal-soal psikotes, dan hasil nilai dalam bentuk grafik.

1.4 Metodologi Yang Digunakan

Dalam melakukan perancangan aplikasi ini metodologi perangkat lunak yang digunakan oleh penulis adalah metode *Waterfall* (metode air terjun), metodenya sebagai berikut :

1. Rekayasa Sistem

Tahapan Rekayasa sistem. Tahapan ini menekankan pada masalah pengumpulan kebutuhan pengguna pada tingkatan sistem dengan menentukan konsep sistem beserta antar muka yang menghubungkannya dengan lingkungan sekitar. Hasilnya berupa spesifikasi sistem.

2. Analisis

Tahapan Analisis. Pada tahapan ini dilakukan pengumpulan kebutuhan elemen-elemen di tingkat rekayasa sistem. Dengan analisis harus dapat ditentukan domain-domain data atau informasi, fungsi, proses, atau prosedur yang diperlukan beserta unjuk kerjanya, dan antar muka. Hasilnya berupa spesifikasi kebutuhan rekayasa sistem.

3. Desain

Tahapan Perancangan. Pada tahap perancangan, kebutuhan-kebutuhan rekayasa sistem, yang dihasilkan pada tahap analisis, di transformasikan ke

dalam bentuk arsitektur rekayasa sistem yang memiliki karakteristik mudah di mengerti dan tidak sulit untuk di terapkan.

4. Kode

Tahapan Pemograman. Tahapan ini melakukan penerapan hasil rancangan ke dalam baris-baris kode program yang dapat di mengerti oleh komputer.

5. Pengujian

Tahapan Pengujian. Setelah rekayasa sistem selesai di terapkan, pengujian dapat segera di mulai. Pengujian terlebih dahulu di lakukan pada setiap modul, jika tidak ada masalah, modul tersebut akan di integrasikan hingga membentuk rekayasa sistem secara utuh. Kemudian di lakukan pengujian di tingkat rekayasa sistem yang memfokuskan pada masalah-masalah logika internal, fungsi internal, potensi masalah yang mungkin terjadi masalah yang mungkin terjadi dan pemeriksaan hasil apakah sudah sesuai dengan permintaan.

6. Pemeliharaan

Sebuah rekayasa sistem akan mengalami perubahan setelah di sampaikan kepada user (perkecualian yang mungkin adalah rekayasa sistem yang di letakan). Dalam penelitian Tugas Akhir ini tahap pemeliharaan tidak dilakukan.

1.5 Sistematika Penulisan

Agar penyusunan lebih sempurna, maka laporan tugas akhir ini disusun dalam beberapa bab yaitu ;

Bab I : PENDAHULUAN

Bab ini akan membahas mengenai latar belakang, tujuan penulis, ruang lingkup, batasan masalah, metodologi dan sistematika penulisan.

Bab II : LANDASAN TEORI

Bab ini membahas mengenai teori-teori pendukung yang bersangkutan dengan isi laporan ini, meliputi System Development Life Cycle (SDLC), Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), basis data, State Transaction Diagram (STD), konsep kerja PHP dan MYSQL.

Bab III : ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai hal-hal analisis masalah, analisis kebutuhan sistem, dan perancangan sistem.

Bab IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai hal-hal yang berhubungan dengan kode program, dan pengujian aplikasi.

Bab V : KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang merupakan uraian-uraian yang diambil dari bab sebelumnya, dan beberapa saran untuk pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Psikotes

Psikotes dilakukan untuk mengetahui kematangan sosial dan gangguan emosi. Sedangkan alat tes yang lain yaitu tes proyektif yang memiliki beberapa jenis tes yaitu (<http://www.ditplb.or.id/profile.php?id=47>) :

- a. ***Tes Rorchach*** ; Tes ini memberikan gambaran mengenai keseluruhan kepribadian, kelainan dan perlunya psikoterapi. Gambaran ini ditafsirkan dari reaksi anak terhadap gambar-gambar yang terbuat dari tetesan tinta.
- b. ***Thematic Apperception Test (TAT)*** ; Tes ini memperlihatkan berbagai situasi-emosi dalam bentuk gambar-gambar. Gambaran kepribadian nampak dari tafsiran anak mengenai situasi emosi tersebut untuk itu disediakan skala khusus.
- c. ***Tes Gambar Orang*** ; Dalam tes ini persoalan-persoalan emosi nampak dari gambar yang harus dibuat oleh anak. Gambarnya ialah seorang laki-laki dan seorang perempuan.
- d. ***Dispert Fable Tes*** ; Tes ini memberikan gambaran mengenai: iri hati, rasa dosa, rasa cemas, tanggapan terhadap diri sendiri, ketergantungan kepada orang tua, dan sebagainya.

e. Tes Individual dan Klasikal ;

- Perbedaannya adalah pada jumlah individu yang dites.
- Contoh tes individual: TAT, Ro, WB, WAIS, WISC, dsb.
- Contoh tes klasikal: IST, EPPS, RMIB, TKD, CFIT, dsb.
- Tes individual biasanya digunakan untuk asesmen individual mendalam, misal: klien klinis, pasien rumah sakit.
- Tes klasikal biasanya digunakan untuk seleksi karyawan, seleksi siswa, untuk tujuan riset, screening, dsb.

f. Tes Performance dan Verbal ;

- Yang membedakan adalah materi tes yang digunakan serta aktivitas yang dilakukan berhubungan dengan tes (cara pengerjaan tes).
- Tes Verbal misal: paper & pencil test, kuesioner, visual tes, pilihan ganda, dsb.
- Tes Performance berkaitan dengan aktivitas motorik. Misal: DAP, HTP, Baum, Wartegg, sub tes melengkapi gambar, menata balok dalam tes IQ, dsb.

g. Tes Terstruktur dan Tidak Terstruktur ;

- Perbedaannya terletak pada luas respon dan kepastian tugas dari tes.
- Tes tidak terstruktur memberikan kebebasan testee dan kepastian tugas dari tes, misal: soal essay, tes projektif (TAT, Ro, Hand Test, dsb). Lebih sulit diskor dan diinterpretasi.
- Tes terstruktur biasa disebut juga tes objektif, misal: tes benar-salah, tes pilihan ganda, tes IQ, dsb.

h. Tes Performance Kepribadian ;

- Testee menunjukkan penampilan kepribadiannya, misal: tes proyeksi (TAT, Ro, Hand Test, Grafis, dsb).

Yang berhak melakukan psikotes dan mengumumkannya adalah psikolog, psikiater, dan counselor, atau orang lain di bawah bimbingannya. Tenaga-tenaga ini ada yang membuka praktek sendiri, ada pula yang tidak membuka praktek sendiri tetapi bekerja di Fakultas Psikologi, Fakultas Kedokteran, Lembaga Kesehatan Jiwa, Balai Bimbingan dan Penyuluhan, Biro Konsultasi Psikologi, dan sebagainya.

Biasanya terdapat 3 aspek pokok yang diungkap dalam psikotes. Ketiga aspek tersebut merupakan satu kesatuan integritas yang tidak bisa dipisahkan secara segmentatif. Hasil dari ketiga aspek ini, nantinya akan menentukan "kualitas" seseorang (<http://www.ssantsons.com/Artikel%20Psikotes.html>).

1. **Pertama**, aspek kecerdasan umum atau intelegensi yang untuk mendeteksinya, dibutuhkan sebuah alat tes yang memancing kemampuan intelegensi umum dan kemampuan khusus. Alat tes yang biasa digunakan bisa berupa tes verbal, non-verbal dan *performance*.
2. **Aspek kedua**, karakteristik/perilaku kerja. Hal ini meliputi berbagai unsur: kecepatan, ketelitian, perencanaan dan semacamnya, biasanya disesuaikan dengan kebutuhan khusus pekerjaan.
3. **Aspek ketiga** adalah aspek kepribadian. Hal ini biasanya mencerminkan sisi-sisi unik seseorang. Untuk menggali aspek ini, dibutuhkan ketajaman dan kepekaan psikolog. Untuk menghindari hal-hal yang subyektif seperti marah atau tersinggung, dibutuhkan pengalaman yang memadai

2.2 Pengertian Sistem

Sebuah sistem terdiri dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud (Davis, 1984). Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu (Mcleod, 2001).

Menurut pengertian para pakar di atas bahwa dalam sistem terdapat beberapa unsur atau elemen yang saling terkait dan bekerjasama menjadi satu kesatuan untuk mencapai suatu tujuan tertentu. Elemen-elemen tersebut terdiri dari manusia, mesin, prosedur, dokumen, data atau elemen lain yang terorganisir. Elemen sistem di samping berhubungan satu sama lain, juga berhubungan dengan lingkungannya untuk mencapai tujuan yang telah ditentukan sebelumnya. Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, adapun karakteristik sistem tersebut yaitu:

a. Komponen sistem (*components*)

Komponen sistem adalah bagian sistem yang saling berinteraksi dan membentuk satu kesatuan. Komponen atau elemen sistem dapat berupa subsistem atau beberapa bagian sistem.

b. Batas sistem (*boundary*)

Batas sistem ialah daerah yang membatasi antara suatu sistem dengan lingkungannya atau dengan sistem lainnya. Batas sistem inilah yang membuat sistem dipandang sebagai satu kesatuan.

c. Lingkungan luar sistem (*environments*)

Lingkungan luar sistem segala sesuatu yang berada di luar sistem yang mempengaruhi sistem. Lingkungan luar sistem dapat bersifat menguntungkan sistem atau merugikan sistem.

d. Penghubung sistem (*interface*)

Penghubung sistem merupakan media penghubung merupakan media penghubung antara satu subsistem lainnya. Penghubungan inilah yang menyebabkan beberapa subsistem berintegrasi dan membentuk satu kesatuan.

e. Masukan sistem (*input*)

Masukan sistem adalah sesuatu yang dimasukkan kedalam sistem yang berasal dari lingkungan.

f. Keluaran sistem (*output*)

Keluaran sistem suatu hasil dari proses pengolahan sistem yang dikeluarkan ke lingkungan.

g. Pengolah sistem (*proses*)

Pengolah sistem ialah bagian dari sistem yang mengubah masukan (*input*) menjadi keluaran (*output*).

h. Sasaran sistem (*objectives*) atau Tujuan (*goal*)

Sasaran sistem adalah sesuatu yang menyebabkan mengapa sistem itu dibuat atau ada. Suatu system dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.3 Pengertian Informasi

Informasi adalah data yang sudah diolah, dibentuk atau dimanipulasi sesuai dengan keperluan tertentu atau hasil dari pengolahan data yang secara prinsip memiliki nilai atau value yang lebih dibandingkan data mentah. Informasi dapat juga dianggap suatu data untuk diolah lagi dan menjadikan informasi sesuai dengan keperluan unit kerja tertentu.

Informasi dapat juga dibuat untuk keperluan manajemen sesuai dengan unit kerjanya pada tingkatnya masing-masing. Informasi memiliki tingkat kualitas. Yang ditentukan beberapa hal antara lain:

- a) *Akurat*, informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan dan harus jelas penyampaian maksudnya.
- b) *Tepat pada waktunya*, informasi yang datang tidak boleh terlambat pada penerima.
- c) *Relevan*, informasi harus mempunyai manfaat bagi pemakainya.
- d) *Lengkap*, informasi berisi informasi yang diperlukan.
- e) *Jelas*, isi informasi sesuai dengan keperluan pemakai.

2.4 Pengertian Sistem Informasi

Pengertian Sistem Informasi menurut Henry C. Lucas.

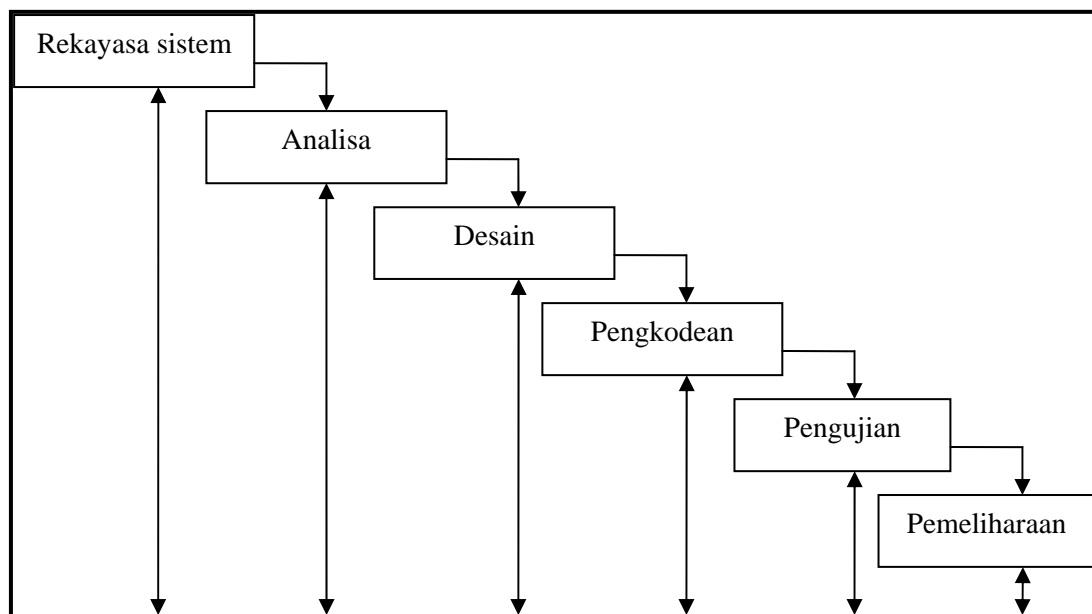
“Sistem Informasi adalah kegiatan dari suatu prosedur-prosedur yang diorganisasikan bilamana dieksekusi dan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam organisasi”.

Pengertian Sistem Informasi menurut Jhon F. Nash dan Martin B. Robert.

“Sistem Informasi adalah suatu kombinasi dari orang-orang, fasilitas teknologi, media, prosedur-prosedur dan pengendalian ditujukan untuk mendapatkan jalur komunikasi yang penting, memproses tipe transaksi yang rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar untuk pengambilan keputusannya yang cerdas”.

2.5 Metodologi Rekayasa Perangkat Lunak

Rekayasa perangkat lunak yang digunakan dalam analisis dan perancangan basis data ialah Model Waterfall. Model ini pertama kali diperkenalkan oleh Winston Royce pada tahun 1970. Model waterfall merupakan model klasik yang sederhana dengan aliran sistem linear. Keluaran dari setiap tahap merupakan masukan bagi tahap selanjutnya. Setiap tahap dalam model ini selalu dilakukan verifikasi untuk menekan kesalahan sekecil mungkin. Model Waterfall memiliki beberapa tahapan seperti yang terdapat pada Gambar dibawah.



Gambar 2.1 Skema model Air terjun

Tahapan-tahapan dalam Model Waterfall adalah sebagai berikut (Pressman, 1997:25):

1. Tahapan Rekayasa Sistem. Tahapan ini sangat menekan pada masalah pengumpulan kebutuhan pengguna pada tingkatan sistem dengan menentukan konsep sistem beserta antarmuka yang menghubungkannya dengan lingkungan sekitar. Hasilnya berupa spesifikasi sistem.

2. Tahapan Analisis. Pada tahap ini dilakukan pengumpulan kebutuhan elemen-elemen di tingkat perangkat lunak. Dengan analisis harus dapat ditentukan domain-domain data atau informasi, fungsi, proses, atau prosedur yang diperlukan beserta unjuk kerjanya, dan antarmuka. Hasilnya berupa spesifikasi kebutuhan perangkat lunak.
3. Tahapan Perancangan. Pada tahap perancangan, kebutuhan-kebutuhan perangkat lunak, yang dihasilkan pada tahap analisis, ditransformasikan ke dalam bentuk arsitektur perangkat lunak yang memiliki karakteristik mudah dimengerti dan tidak sulit untuk diterapkan.
4. Tahapan Pemrograman. Tahap ini melakukan penerapan hasil rancangan ke dalam baris-baris kode program yang dapat dimengerti oleh Komputer.
5. Tahapan Pengujian. Setelah perangkat lunak selesai diterapkan, pengujian dapat segera dimulai. Pengujian terlebih dahulu dilakukan pada setiap modul. Jika tidak ada masalah, modul tersebut akan diintegrasikan hingga membentuk perangkat lunak secara utuh. Kemudian dilakukan pengujian di tingkat perangkat lunak yang memfokuskan pada masalah-masalah logika internal, fungsi internal, potensi masalah yang mungkin terjadi dan pemeriksaan hasil apakah sudah sesuai dengan permintaan.

Tahapan Pengoperasian dan Pemeliharaan. Dalam masa operasional sehari-hari, suatu perangkat lunak mungkin saja mengalami kesalahan atau kegagalan dalam menjalankan fungsi-fungsinya. Atau, pemilik bias saja meminta peningkatan kemampuan perangkat lunak pada pengembangnya. Dengan demikian, kedua factor ini menyebabkan perlunya perangkat lunak dipelihara dari waktu-kewaktu. Biasanya ini merupakan tahapan yang panjang dari perangkat lunak.

2.6 Unified Modelling Language (UML)

UML adalah salah satu alat bantu yang handal di dunia pengembangan system yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembangan system untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan kelas (*class*) dan operasi dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi obyek seperti C++, Java, C#, atau VB.NET.

2.6.1 Bekerja Dengan Model & Diagram

Di proyek pengembangan sistem apapun, fokus utama dalam analisis dan perancangan adalah model. Hal ini berlaku umum tidak hanya untuk perangkat lunak. Dengan model dapat dipresentasikan sesuatu karena :

- Model mudah dan cepat dibuat.
- Model bisa digunakan sebagai simulasi untuk mempelajari lebih tentang sesuatu.
- Model bisa dikembangkan sejalan dengan pemahaman kita tentang sesuatu.
- Kita bisa memberikan penjelasan lebih rinci tentang sesuatu dengan model.
- Model bisa mewakili sesuatu yang nyata maupun yang tidak nyata.

Ada alat bantu yang sering dipakai oleh sistem analis dan perancang. Alat bantu tersebut adalah diagram. Diagram tersebut digunakan untuk :

- Mengkomunikasikan ide.
- Melahirkan ide-ide baru dan peluang-peluang baru.
- Menguji dan membuat prediksi.
- Memahami struktur dan relasi-relasinya.

Tabel 2.1 Jenis diagram resmi UML

No.	Diagram	Kegunaan
1.	<i>Activity</i>	<i>Behavior prosedural</i> dan <i>parallel</i>
2.	<i>Class</i>	<i>Class, fitur</i> , dan hubungan-hubungan
3.	<i>Communication</i>	Interaksi antar objek; penekanan pada <i>jalur</i>
4.	<i>Component</i>	Struktur dan koneksi komponen
5.	<i>Composite structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i>
6.	<i>Deployment</i>	Pemindahan artefak ke node
7.	<i>Interaction overview</i>	Campuran <i>sequence</i> dan <i>activity</i> diagram
8.	<i>Object</i>	Contoh konfigurasi dari contoh-contoh
9.	<i>Package</i>	Struktur hirarki <i>compile-time</i>
10.	<i>Sequence</i>	Interaksi antar objek; penekanan pada <i>sequence</i>
11.	<i>State machine</i>	Bagaimana <i>even</i> mengubah objek selama aktif
12.	<i>Timing</i>	Interaksi antar objek; penekanan pada <i>timing</i>
13.	<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sebuah sistem

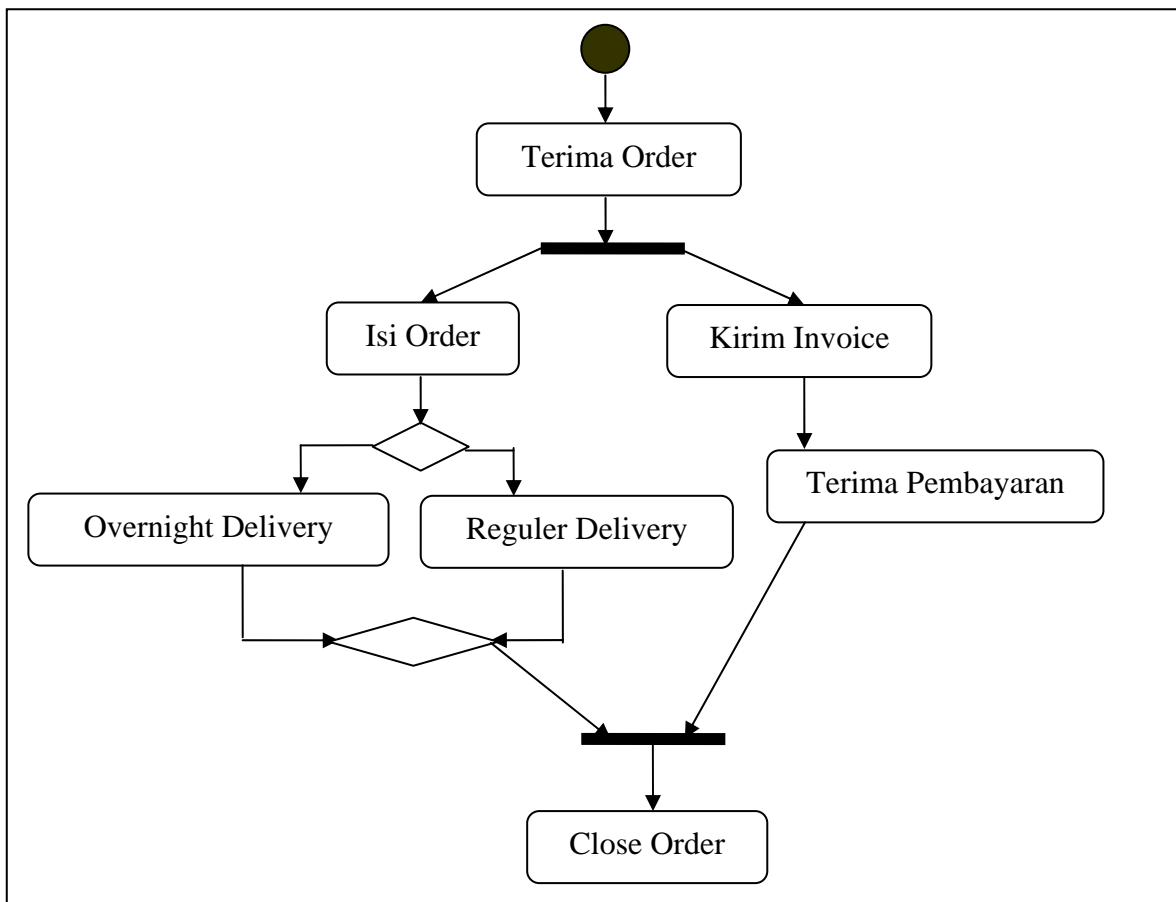
1. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram mempunyai

peran seperti flowchart, akan tetapi perbedaannya dengan flowchart activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa.

Diagram aktifitas menggambarkan berbagai alur aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir. Diagram aktifitas juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Diagram aktifitas merupakan diagram state khusus, dimana sebagian besar keadaan (state) adalah aksi (action) dan sebagian besar transisi dipicu oleh selesainya sebelumnya (internal processing).



Gambar 2.2 Notasi *activity* diagram

secara paralel. Intinya tidak jadi masalah mengenai mana yang terlebih dahulu harus diselesaikan. Kondisinya paralel jelas membutuhkan sinkronisasi. Pada kasus di atas,

order tidak akan di tutup sampai barang dikirim dan dibayar. Untuk menunjukan 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 di tutup jika pembayaran sudah di lakukan dan pengiriman sudah dilakukan.

2. Diagram Kelas (class diagram)

diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar mereka. Ia memberikan gambaran sistem secara statis. Biasanya, dibuat beberapa diagram kelas untuk satu system. Satu diagram kelas menampilkan *subset* dari kelas-kelas dan relasinya. Yang lainnya, mungkin menampilkan kelas-kelas termasuk atribut dan operasi dalam kelas-kelas. Dan yang lainnya lagi mungkin menampilkan paket-paket kelas dan relasi antar paket-paket. Dapat dibuat diagram sesuai yang diinginkan untuk mendapatkan gambaran lengkap terhadap system yang dibangun. Tidak ada aturan yang mengharuskan berapa banyaknya diagram kelas yang harus dibuat.

Diagram kelas adalah alat perancangan terbaik untuk tim pengembang perangkat lunak diagram tersebut membantu pengembang mendapatkan struktur system sebelum menuliskan kode program, membantu untuk memastikan bahwa system adalah rancangan terbaik.

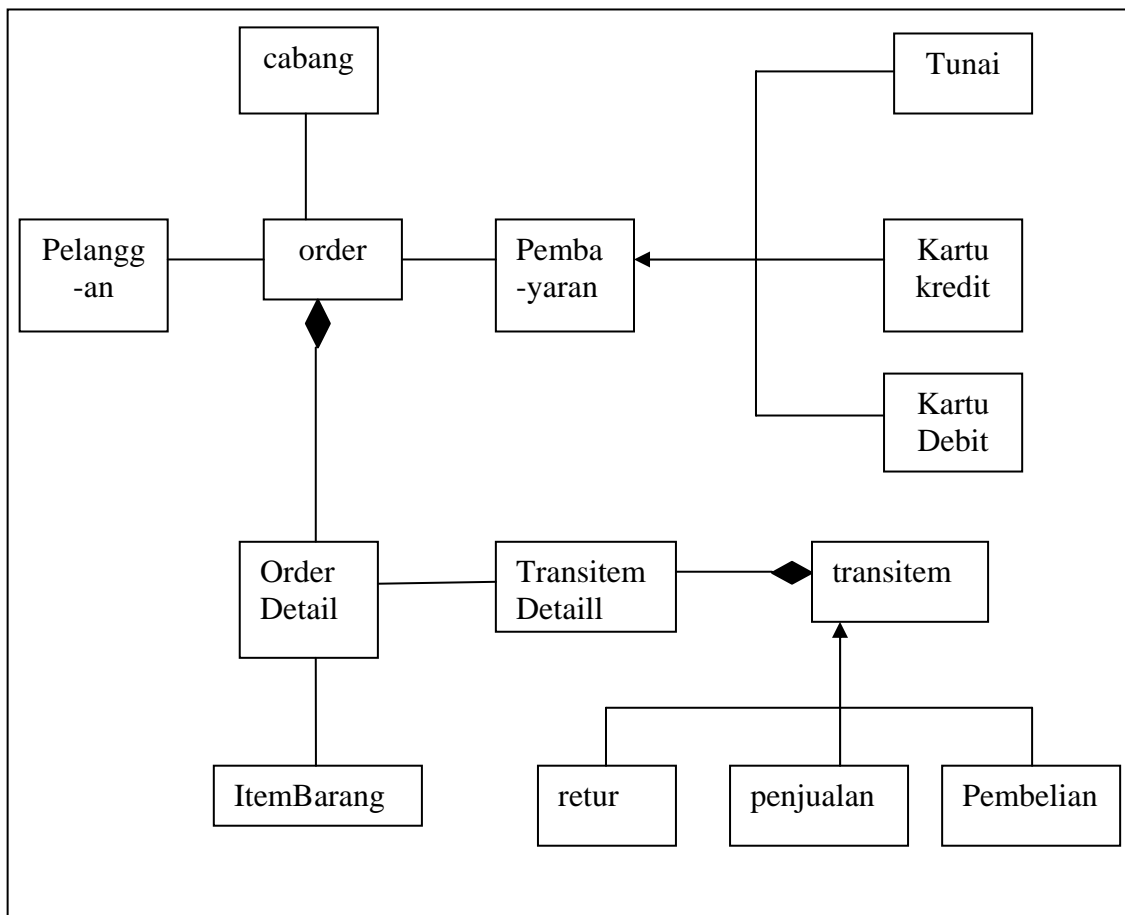
Contoh Kasus :

VCD galeri adalah adalah sebuah galeri yang bergerak dibidang retail dan menjual produk-produk hiburan VCD, DVD, CD Playstation, accessories dan sebagainya. Pada saat ini galeri tersebut ingin meningkatkan efisiensi kerja dari

pegawainya dan meningkatkan penjualan dengan cara mengotomisasi system penjualan.

Berikut adalah system yang bersifat manual yang sedang berjalan digaleri tersebut. Proses penjualan dimulai saat seorang pelanggan menanyakan tentang suatu produk kepada suatu petugas toko. Oleh petugas toko, permintaan tersebut akan ditulis dalam bentuk bon. Selanjutnya atas petunjuk bon tersebut, petugas toko akan mengecek dan mengambil barang yang dipesan. Jika barang tidak ada atau jumlah permintaan tidak sesuai maka transaksi akan dikonfirmasi untuk tetap dilakukan atau tidak. Jika transaksi tetap dilakukan dengan cara mengambil barang jenis lain atau mengubah jumlah barang yang dipesan, maka petugas akan melakukan perhitungan pembayaran. Pembayaran bisa dilakukan dengan cash, kartu kredit atau dengan kartu debit. Jika pembayaran dilakukan dengan tunai, maka transaksi akan dicatat dalam buku penjualan, namun jika dilakukan dengan kartu kredit atau kartu debit maka akan dilakukan otorisasi terlebih dahulu dengan mesin otorisasi yang disediakan oleh bank. Setelah semua proses selesai, barulah proses penjualan dicatat dibuku penjualan.

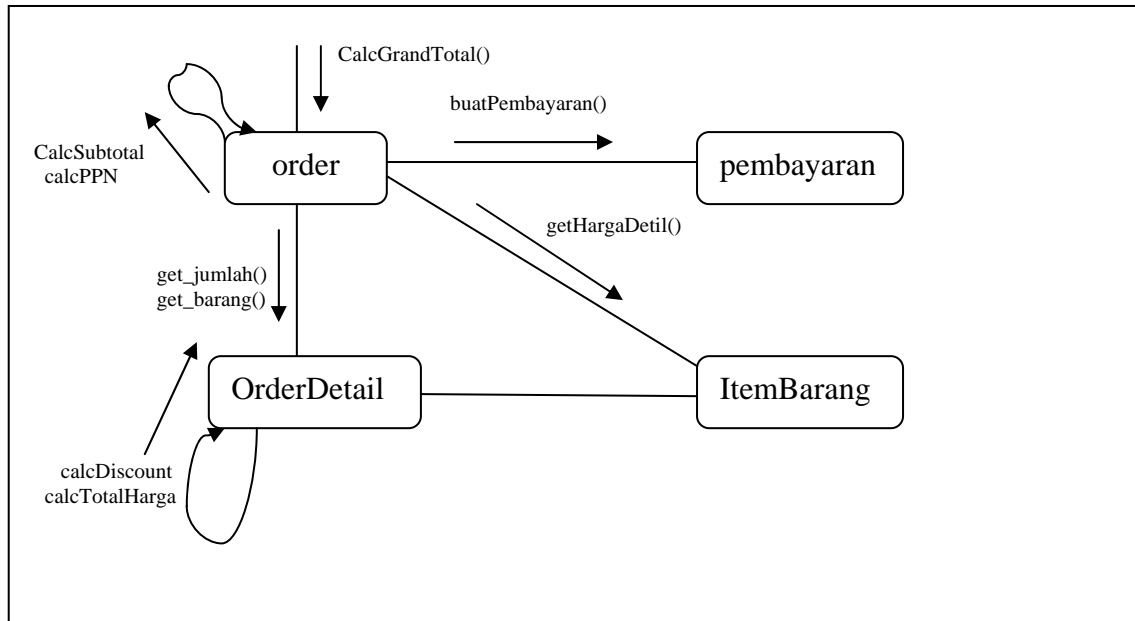
Untuk memberikan pelayanan yang lebih baik kepada pelanggan, pelanggan diperbolehkan mengembalikan (return) barang yang tidak sesuai dengan spesifikasi dengan barang yang sama atau dengan barang yang lain yang senilai. Petugas akan mencatat transaksi ini dibuku return. Untuk menghitung stok, petugas stok akan membaca buku inventori barang sebagai acuan untuk menghitung stok awal. Selanjutnya berdasarkan dokumen penjualan dan dokumen return akan bias dihitung stok akhir.



Gambar 2.3 Class Diagram

3. Communication Diagram

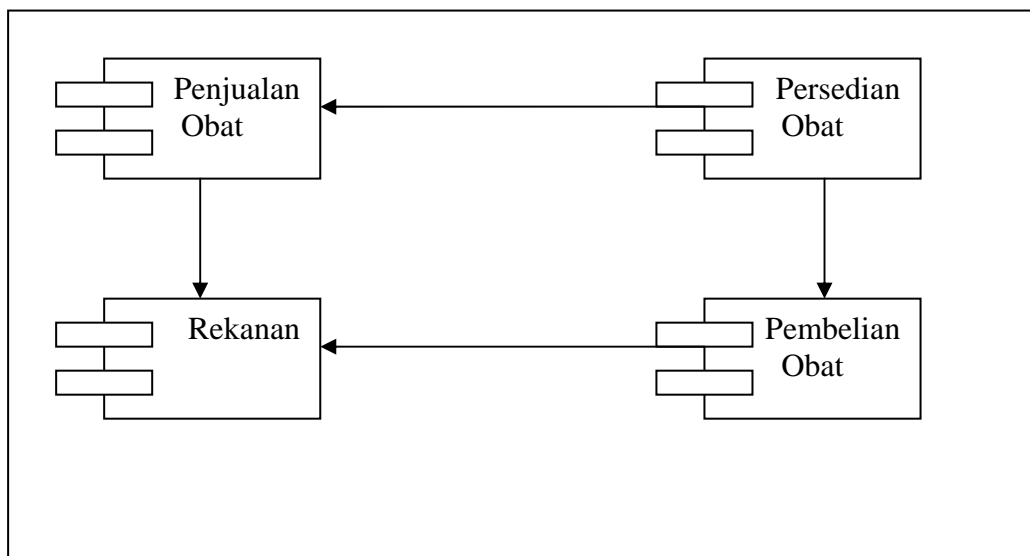
Communication diagram adalah sejenis dengan diagram interaksi, yang lebih menekankan kepada link data diantara bermacam-macam participant pada interaksi tersebut. Pada communication bebas ditempatkan participant, boleh boleh menggambarkan link untuk menunjukkan bagaimana participant berhubungan serta boleh juga memberikan penomoran untuk menunjukkan urutan message.



Gambar 2.4 Communication diagram contoh Kasus

4. Diagram Komponen (component diagram)

Komponen adalah modul fisik dari kode. Komponen bisa mencatumkan pustaka kode program dan berkas-berkas runtime sekaligus. Diagram komponen adalah diagram UML yang menampilkan komponen dalam sistem dan hubungan antar mereka.



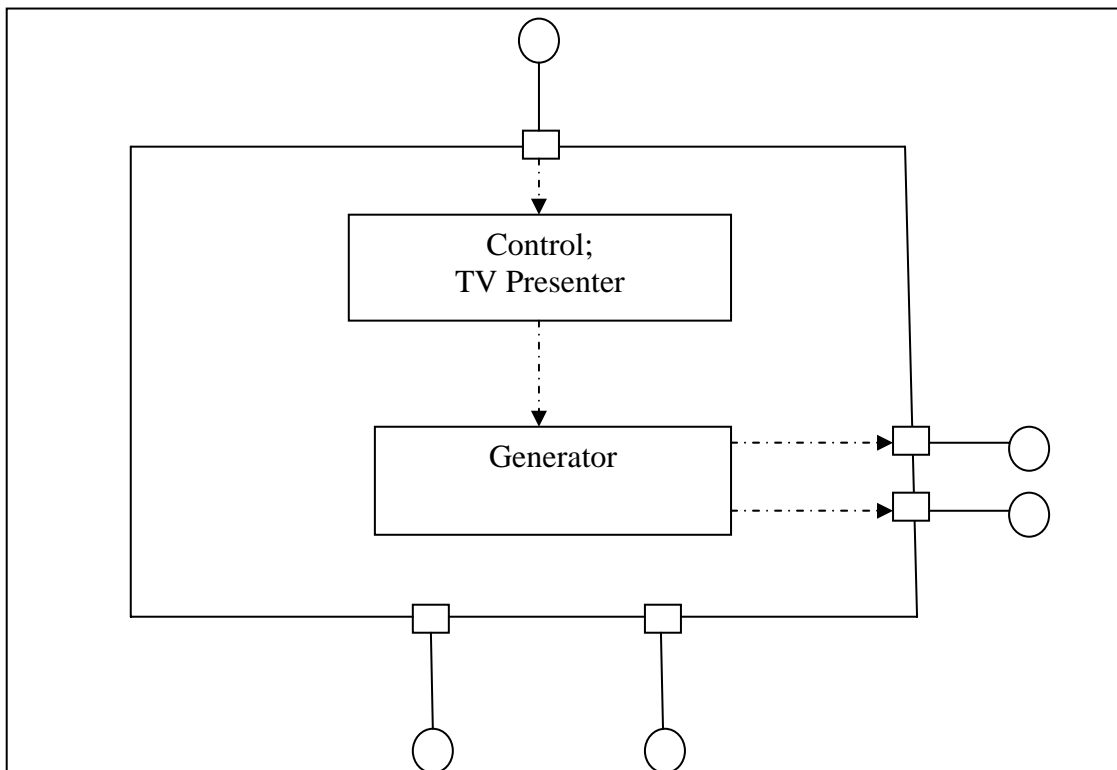
Gambar 2.5 Diagram Komponen

5. Composite Structure Diagram

Diagram composite structure adalah diagram untuk menunjukkan dekomposisi secara hirarkis sebuah class ke sebuah struktur internal. Hal ini memungkinkan untuk memecah objek yang kompleks menjadi bagian – bagian kecil.

Perbedaan antara package dengan composite adalah package digunakan untuk pengelompokkan saat kompilasi (*compile-time*), sedangkan composite structure digunakan untuk pengelompokkan pada saat dijalankan (*runtime*).

Gambar 2.6 akan menunjukkan class didekomposisi secara internal menjadi dua bagian di mana bagian-bagian ini mendukung dan membutuhkan interface yang berbeda. Setiap bagian diberi nama form *name:class*, dengan kedua elemen tersebut optional secara individunya. Bagian-bagian tersebut bukan instance spesifikasi, oleh karena itu harus dicetak tebal bukan digaris bawah.

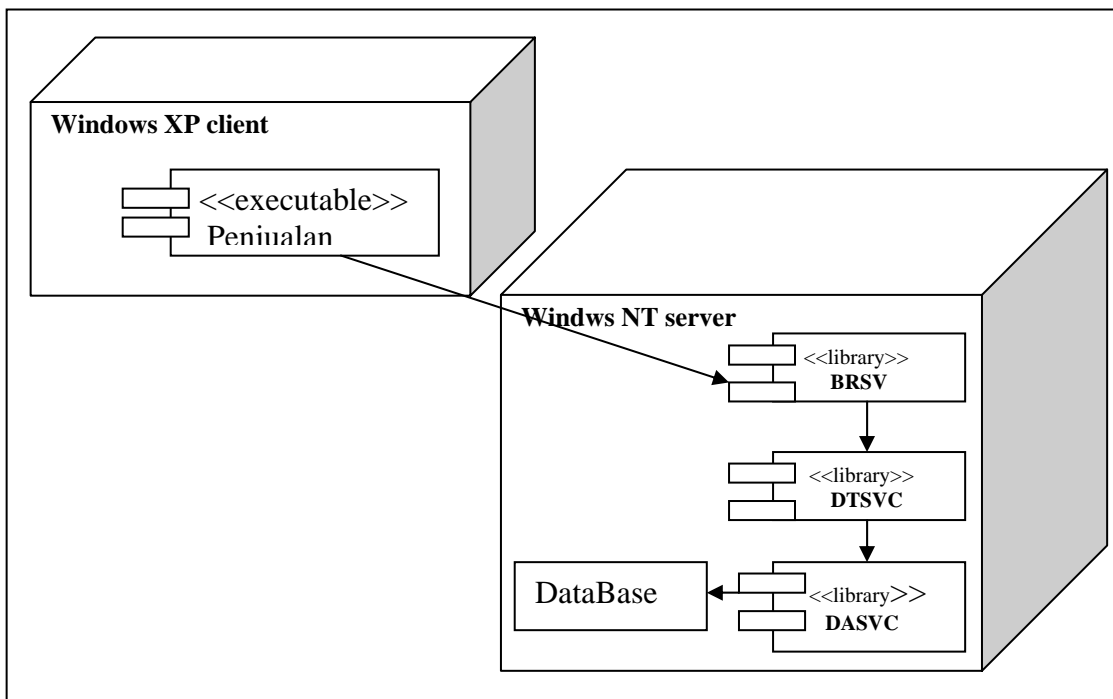


Gambar 2.6 Composite Diagram

6. Diagram Deployment

Deployment adalah segala hal yang berkaitan dengan penyebaran fisik aplikasi. Hal ini termasuk persoalan *layout* jaringan dan lokasi komponen-komponen dalam jaringan. Juga mempertimbangkan persoalan-persoalan jaringan seperti berapa banyak *bandwith* yang dibutuhkan dalam jaringan, seberapa banyak pemakai yang dapat menggunakan jaringan secara bersamaan, apa yang dilakukan jika server dalam keadaan down dan juga pada saat on.

Diagram deployment berisikan prosesor-prosesor, peralatan-peralatan, proses-proses, dan hubungan antar prosesor atau antar peralatan.

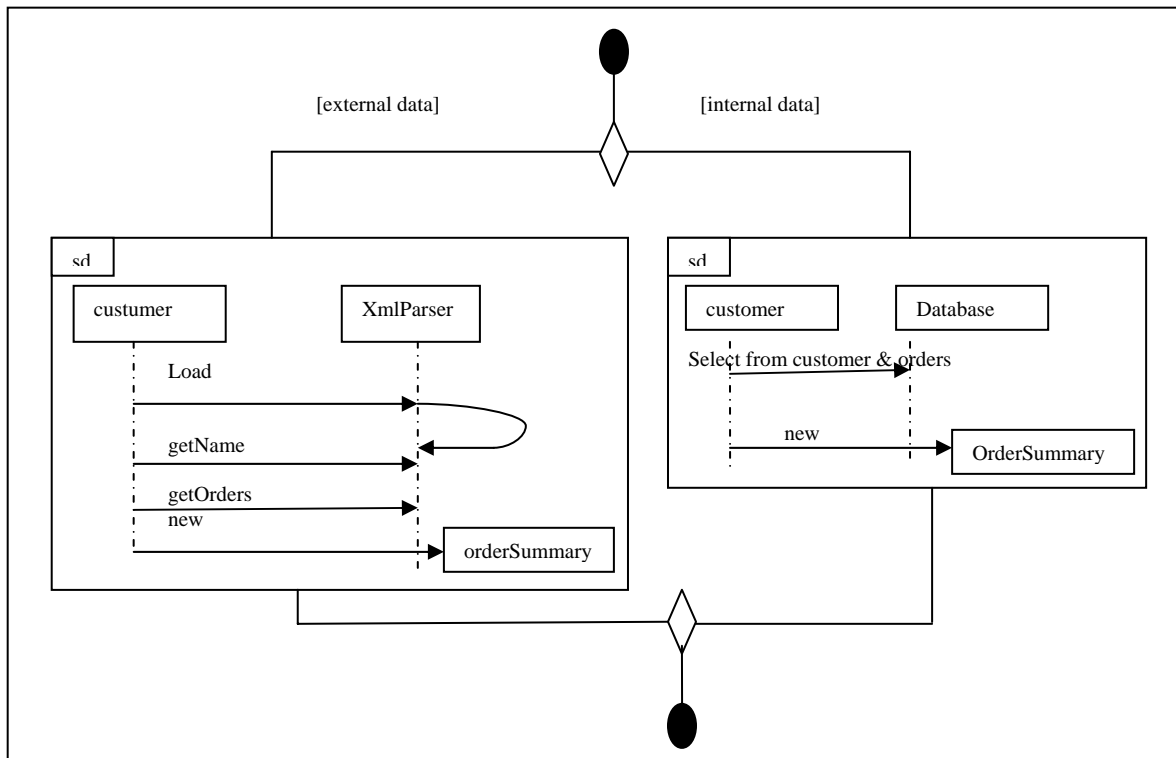


Gambar 2.7 Deployment Diagram gallery VCD

7. Interaction Overview Diagram

Interaction Overview diagram adalah pecangkokkan secara bersama antara activity diagram dengan sequence diagram. Interaction overview diagram bisa dianggap sebagai activity diagram di mana semua aktivitas diganti dengan sedikit sequence diagram, atau bisa juga dianggap sebagai sequence diagram yang dirincikan

dengan notasi activity diagram yang digunakan untuk menunjukkan aliran pengawasan. Dengan kata lain interaction diagram adalah gabungan diantara keduanya yang membuat campuran sedikit ganjil.



Gambar 2.8 Interaction Diagram VCD Gallery

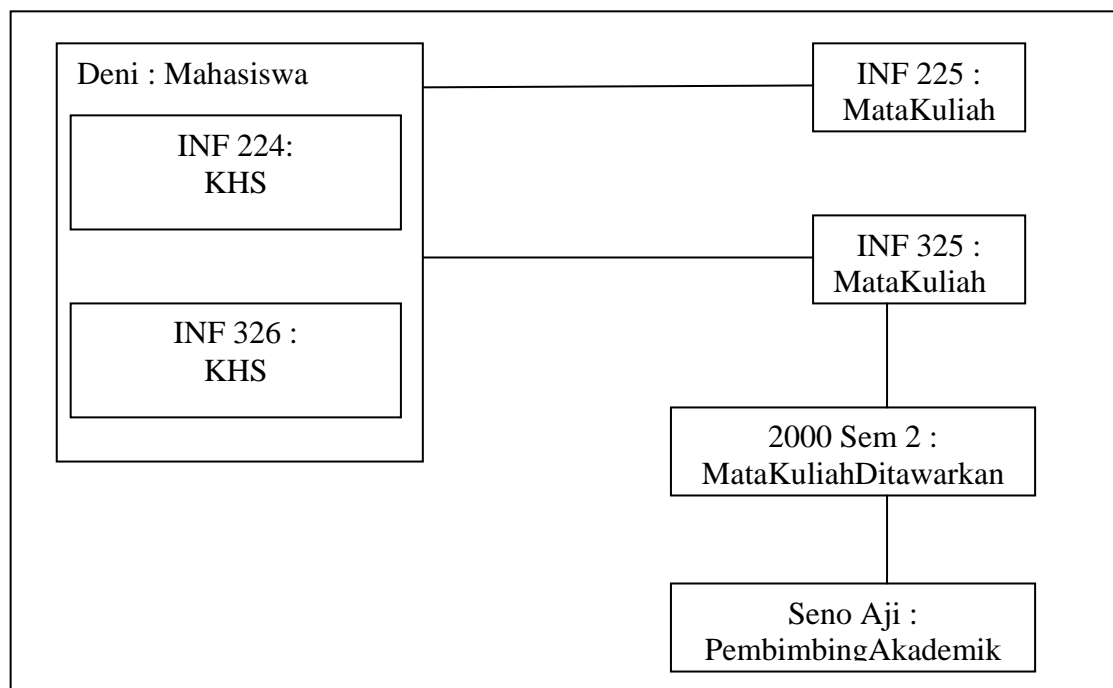
8. Object Diagram

Object diagram adalah objek-objek secara ringkas disebuah sistem pada suatu waktu. Object bisa digunakan untuk menunjukkan contoh konfigurasi dari objek-objek. Secara umum object diagram mengandung objek dan link atau bisa juga mengandung package atau subsistem.

Object diagram sangat berdaya guna dalam menunjukkan contoh-contoh objek yang saling terhubung satu sama lain. Dalam banyak kasus struktur yang tepat bisa digambarkan secara tepat dengan class diagram, akan tetapi struktur tersebut mungkin

masih susah untuk dimengerti. Pada situasi seperti ini pembuatan contoh dengan objek diagram akan sangat membantu sekali.

Gambar di bawah ini menunjukkan bahwa mahasiswa bernama deni sudah mengambil mata kuliah statistic dan matematika diskrit. Saat ini deni sedang mangambil matakuliah kalkulus dan pemrograman. Sedangkan Seno Aji adalah pembimbing pada semester tersebut.

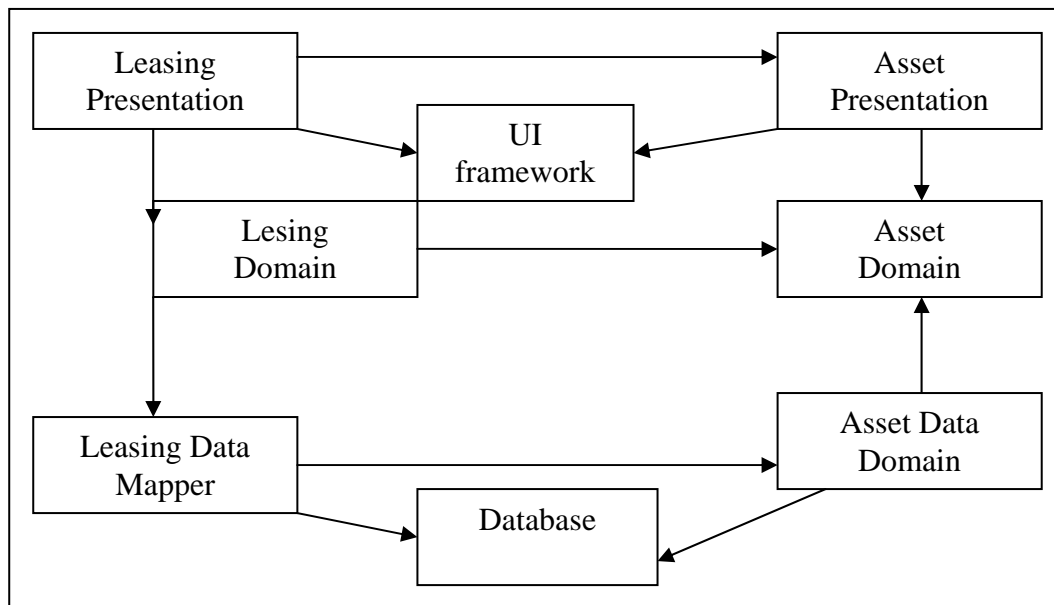


Gambar 2.9 Objek Diagram

9. Package Diagram

Package adalah pengelompokkan konstruksi ke level yang lebih tinggi. Sebuah package bisa menjadi anggota package yang lain. Bila dibuat hierarki, package yang paling tinggi akan mengandung package-package yang lain. Demikian seterusnya hingga yang paling bawah adalah class.

Manfaat utama penggunaan package adalah penerapannya pada sistem skala besar untuk mendapatkan gambaran saling ketergantungan diantara component-component utama pada sistem.

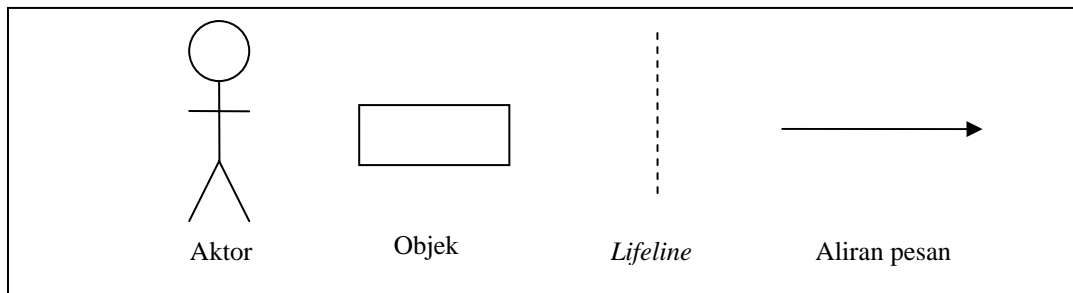


Gambar 2.10 Contoh Package diagram

10. Diagram Sequensial (Sequence Diagram)

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan pesan yang diletakkan diantara objek-objek ini dalam use case. . Masing-masing obyek, termasuk aktor, memiliki garis hidup (lifeline) vertical. Pesan digambarkan sebagai garis berpanah dari satu obyek ke obyek lainnya. Notasi-notasi yang digunakan dalam pemodelan Diagram Sequensial adalah:

1. Aktor.
2. Obyek, hal yang dikenai perbuatan oleh Aktor.
3. Garis hidup, menggambarkan bahwa hadirnya obyek terhadap waktu.
4. Aliran pesan, menggambarkan komunikasi antar objek.



Gambar 2.11 Notasi *sequence* diagram

11. State Machine Diagram

Objek pada sistem mengubah state-nya untuk merespon event dan waktu. State diagram menangkap perubahan state tersebut. State diagram fokus pada perubahan state hanya pada satu objek. Segi empat yang pinggirnya oval bisa mewakili state, sedangkan garis tanda dengan tanda panah mewakili transition.

Kadangkala sebuah state terdiri dari sub state. Sub state bisa menjadi seugensial (terjadinya satu sub state setelah sub state yang lain) dan bisa menjadi concurrent (terjadi pada saat yang bersamaan). State yang mempunyai sub state disebut composite state mengungat sub state-nya saat transition objek keluar dari composite state.

Sangat penting untuk membuat state diagram, karena dapat membantu analis, desaigner dan developer dalam memahami perilaku objek yang ada pada sistem. Developer secra spesifik harus mengetahui apa yang seharusnya dipunyai oleh objek-objek, karena mengimplementasika perilaku tersebut ke dalam software. Hal tersebut masih belum cukup, karena developer harus membuat objek tersebut melakukan sesuatu.

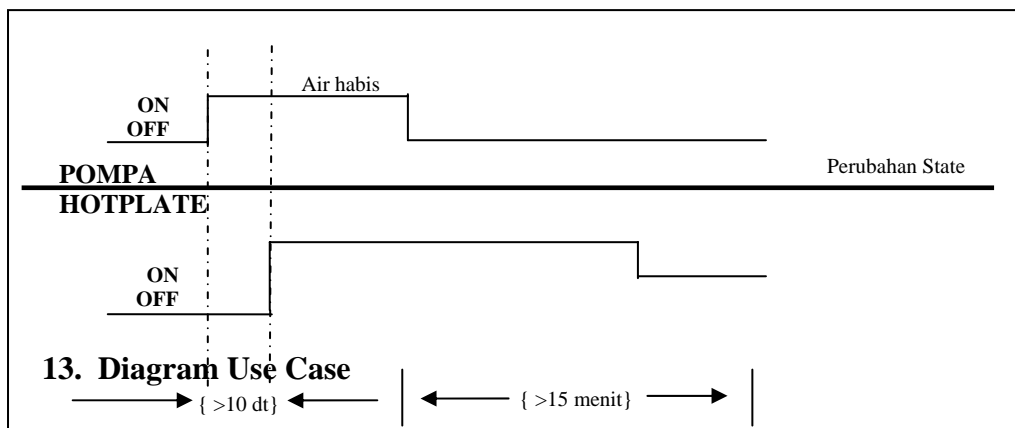


Gambar 2.12 Simbol Statechart Diagram

12. Timing Diagram

Timing diagram sudah dipakai sejak lama di bidang elektronik. Timing diagram adalah bentuk lain dari interaction daigram, di mana focus utamanya lebih kewaktu. Timing diagram ini bisa dipakai untuk objek tunggal maupun sekelompok objek.

Contoh pada mesin pembuat kopi terdapat dua bagian yang penting yaitu pompa dan hotplate. Katakanlah ada sebuah skenario sederhana untuk mesin pembuat kopi ini di mana ada sebuah aturan bahwa jeda antara pompa aktif (on) dengan hotplate aktif (on) adalah 10 detik. Ketika persediaan air habis, pompa akan dalam posisi off dan hotplate tidak bisa diaktifkan selama 15 menit.



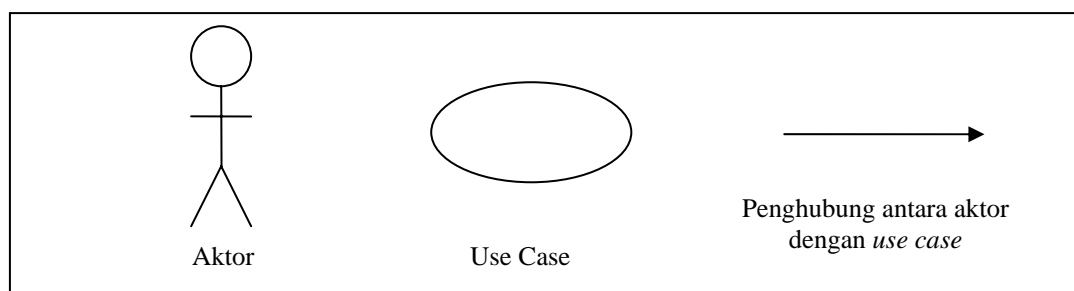
Gambar 2.13 Timing Diagram

Use case adalah abstraksi dari interaksi antara sistem dan actor. Use case merupakan alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang

suatu sistem dan sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna susah kali menjawabnya tatkala dimintai tentang sesuatu.

Ide dasarnya adalah bagaimana melibatkan penggunaan system di fase-fase analisis dan perancangan sistem. Dengan semikian diharapkan akan dibangun sesuatu sistem yang bisa membantu pengguna. Notasi-notasi yang digunakan dalam pemodelan use case dapat dilihat pada Gambar 2.14 adalah:

1. Aktor, merupakan subjek yang menjalankan sistem.
2. Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.
3. Penghubung antara aktor dengan use case (*generalization*), menggambarkan hubungan khusus dalam obyek anak (*child*) yang menggantikan obyek induk (*parent*).



Gambar 2.14 Notasi *use case* diagram

Sebagai contoh, saat pelanggan menelepon restoran untuk melakukan booking, dia akan berbicara pada karyawan restoran untuk mencatat booking tersebut ke system. Untuk melakukan hal tersebut, karyawan akan menjalankan peran sebagai resepsionis meskipun pekerjaan tersebut mungkin bukan pekerjaan formalnya. Pada situasi ini,

karyawan merupakan instance dari aktor resepsionis dan interaksi diantara karyawan dengan sistem adalah instance dari use case.

2.7 Perancangan Database

Database merupakan komponen dasar dari sebuah sistem informasi dan pengembangan serta penggunaannya sebaiknya dipandang dari perspektif kebutuhan organisasi yang lebih besar. Oleh karena itu siklus hidup sebuah system informasi organisasi berhubungan dengan siklus hidup sistem database yang mendukungnya.

2.7.1 Proses Perancangan Database

6 Fase proses perancangan database :

1. Pengumpulan data dan analisis
2. Perancangan database secara konseptual
3. Pemilihan DBMS
4. Perancangan database secara logika (data model mapping)
5. Perancangan database secara fisik
6. Implementasi Sistem database.

Fase 1 : Pengumpulan data dan analisa

Proses identifikasi dan analisa kebutuhan-kebutuhan data disebut pengumpulan data dan analisa. Untuk menentukan kebutuhan-kebutuhan suatu sistem database, pertama-tama harus mengenal bagian-bagian lain dari sistem informasi yang akan berinteraksi dengan sistem database, termasuk para pemakai yang ada dan para pemakai yang baru serta aplikasi-aplikasinya. Kebutuhan-kebutuhan dari para pemakai dan aplikasi-aplikasi inilah yang kemudian dikumpulkan dan dianalisa.

Aktifitas-aktifitas pengumpulan data dan analisa :

1. Menentukan kelompok pemakai dan bidang-bidang aplikasinya

Menentukan aplikasi utama dan kelompok user yang akan menggunakan database. Individu utama pada tiap-tiap kelompok pemakai dan bidang aplikasi yang telah dipilih merupakan peserta utama pada langkah-langkah berikutnya dari pengumpulan dan spesifikasi data.

2. Peninjauan dokumentasi yang ada

Dokumen yang ada yang berhubungan dengan aplikasi-aplikasi dipelajari dan dianalisa. Dokumen-dokumen lainnya (seperti : kebijaksanaan-kebijaksanaan, form, report, dan bagan organisasi) diuji dan ditinjau kembali untuk menguji apakah dokumen-dokumen tsb berpengaruh terhadap kumpulan data dan proses spesifikasi.

3. Analisa lingkungan operasi dan pemrosesan data

Informasi yang sekarang dan yang akan datang dipelajari. Termasuk juga analisa jenis-jenis transaksi dan frekuensi-frekuensi transaksinya dan juga arus informasi dalam sistem. Input-output data untuk transaksi-transaksi tsb diperinci.

4. Daftar pertanyaan dan wawancara

Tuliskan tanggapan -tanggapan dari pertanyaan-pertanyaan yang telah dikumpulkan dari para pemakai database yang berpotensi. Ketua kelompok (individu utama) dapat diwawancarai sehingga input yang banyak dapat diterima dari mereka dengan memperhatikan informasi yang berharga dan mengadakan prioritas.

Fase 2 : Perancangan database secara konseptual

Tujuan dari fase ini adalah menghasilkan conceptual schema untuk database yang tergantung pada sebuah DBMS yang spesifik. Sering menggunakan sebuah

high-level data model seperti ER/EER model selama fase ini. Dalam conceptual schema, kita harus memerinci aplikasi-aplikasi database yang diketahui dan transaksi transaksi yang mungkin.

Fase perancangan database secara konseptual mempunyai 2 aktifitas paralel :

1. Perancangan skema konseptual :

Menguji kebutuhan-kebutuhan data dari suatu database yang merupakan hasil dari fase 1, dan menghasilkan sebuah conceptual database schema pada DBMS independent model data tingkat tinggi seperti EER (enhanced entity relationship) model. Skema ini dapat dihasilkan dengan menggabungkan bermacam-macam kebutuhan user dan secara langsung membuat skema database atau dengan merancang skema-skema yang terpisah dari kebutuhan tiap-tiap user dan kemudian menggabungkan skema-skema tsb. Model data yang digunakan pada perancangan skema konseptual adalah DBMS-independent, dan langkah selanjutnya adalah memilih sebuah DBMS untuk melaksanakan rancangan tsb.

2. Perancangan transaksi :

Menguji aplikasi-aplikasi database dimana kebutuhan-kebutuhannya telah dianalisa pada fase 1, dan menghasilkan perincian transaksi-transaksi ini. Kegunaan fase ini yang diproses secara paralel bersama fase perancangan skema konseptual adalah untuk merancang karakteristik dari transaksi-transaksi database yang telah diketahui pada suatu DBMS-independent. Transaksi-transaksi ini akan digunakan untuk memproses dan memanipulasi database suatu saat dimana database tsb dilaksanakan.

Fase 3 : Pemilihan DBMS

Pemilihan database di tentukan oleh beberapa faktor, diantaranya : faktor teknik, ekonomi, dan politik organisasi. Contoh faktor teknik : keberadaan DBMS dalam menjalankan tugasnya seperti jenis-jenis DBMS (relational, network, hierarchical, dll), struktur penyimpanan, dan jalur akses yang mendukung DBMS, pemakai, dll.

Faktor-faktor ekonomi dan organisasi yang mempengaruhi satu sama lain dalam pemilihan DBMS :

1. Struktur data

Jika data yang disimpan dalam database mengikuti struktur hirarki, maka suatu jenis hirarki dari DBMS harus dipikirkan.

2. Personal yang telah terbiasa dengan suatu sistem

Jika staf programmer dalam suatu organisasi sudah terbiasa dengan suatu DBMS, hal ini dapat mengurangi biaya latihan dan waktu belajar.

3. Tersedianya layanan penjual

Keberadaan fasilitas pelayanan penjual sangat dibutuhkan untuk membantu memecahkan beberapa masalah sistem.

Fase 4 : Perancangan database secara logika (pemetaan model data)

Fase selanjutnya dari perancangan database adalah membuat sebuah skema konseptual dan skema eksternal pada model data dari DBMS yang terpilih. Fase ini dilakukan oleh pemetaan skema konseptual dan skema eksternal yang dihasilkan pada fase 2. Pada fase ini, skema konseptual ditransformasikan dari model data tingkat

tinggi yang digunakan pada fase 2 ke dalam model data dari DBMS yang dipilih pada fase 3. Pemetaannya dapat diproses dalam 2 tingkat :

1. Pemetaan system-independent :

Pemetaan ke dalam model data DBMS dengan tidak mempertimbangkan karakteristik atau hal-hal yang khusus yang berlaku pada implementasi DBMS dari model data tsb.

2. Penyesuaian skema ke DBMS yang spesifik :

Mengatur skema yang dihasilkan pada langkah 1 untuk disesuaikan pada implementasi yang khusus di masa yang akan datang dari suatu model data yang digunakan pada DBMS yang dipilih. Hasil dari fase ini memakai perintah-perintah DDL dalam bahasa DBMS yang dipilih yang menentukan tingkat skema konseptual dan eksternal dari sistem database. Tetapi dalam beberapa hal, perintah-perintah DDL memasukkan parameter-parameter rancangan fisik sehingga DDL yang lengkap harus menunggu sampai fase perancangan database secara fisik telah lengkap.

Fase ini dapat dimulai setelah pemilihan sebuah implementasi model data sambil menunggu DBMS yang spesifik yang akan dipilih. Contoh: jika memutuskan untuk menggunakan beberapa relational DBMS tetapi belum memutuskan suatu relasi yang utama. Rancangan dari skema eksternal untuk aplikasi-aplikasi yang spesifik seringkali sudah selesai selama proses ini.

Fase 5 : Perancangan database secara fisik

Perancangan database secara fisik merupakan proses pemilihan struktur-struktur penyimpanan dan jalur-jalur akses pada file-file database untuk mencapai penampilan yang terbaik pada bermacam-macam aplikasi. Selama fase ini, dirancang spesifikasi-spesifikasi untuk database yang disimpan yang berhubungan dengan

struktur-struktur penyimpanan fisik, penempatan record dan jalur akses. Berhubungan dengan internal schema (pada istilah 3 level arsitektur DBMS). Beberapa petunjuk dalam pemilihan perancangan database secara fisik :

1. Response time :

Waktu yang telah berlalu dari suatu transaksi database yang diajukan untuk menjalankan suatu tanggapan. Pengaruh utama pada response time adalah di bawah pengawasan DBMS yaitu : waktu akses database untuk data item yang ditunjuk oleh suatu transaksi. Response time juga dipengaruhi oleh beberapa factor yang tidak berada di bawah pengawasan DBMS, seperti penjadwalan sistem operasi atau penundaan komunikasi.

2. Space utility :

Jumlah ruang penyimpanan yang digunakan oleh file-file database dan strukturstruktur jalur akses.

3. Transaction throughput :

Rata-rata jumlah transaksi yang dapat diproses per menit oleh sistem database, dan merupakan parameter kritis dari sistem transaksi (misal : digunakan pada pemesanan tempat di pesawat, bank, dll). Hasil dari fase ini adalah penentuan awal dari struktur penyimpanan dan jalur akses untuk file-file database.

Fase 6 : Implementasi sistem database

Setelah perancangan secara logika dan secara fisik lengkap, kita dapat melaksanakan sistem database. Perintah-perintah dalam DDL dan SDL(storage definition language) dari DBMS yang dipilih, dihimpun dan digunakan untuk membuat skema database dan file-file database (yang kosong). Sekarang database tsb dimuat (disatukan) dengan datanya. Jika data harus dirubah dari sistem komputer

sebelumnya, perubahan-perubahan yang rutin mungkin diperlukan untuk format ulang datanya yang kemudian dimasukkan ke database yang baru. Transaksi-transaksi database sekarang harus dilaksanakan oleh para programmer aplikasi. Spesifikasi secara konseptual diuji dan dihubungkan dengan kode program dengan perintah-perintah dari embedded DML yang telah ditulis dan diuji. Suatu saat transaksi-transaksi tersebut telah siap dan data telah dimasukkan ke dalam database, maka fase perancangan dan implementasi telah selesai, dan kemudian fase operasional dari sistem database dimulai.

2.7.2 Perancangan DataBase Secara Logika

Sasaran perancangan basis data adalah untuk menterjemahkan perancangan konseptual ke perancangan basis data logika yang dapat langsung diimplementasikan pada sistem basis data yang dipilih. Basis data yang dihasilkan harus memenuhi kebutuhan pengguna dalam hal berbagi data, fleksibilitas, serta kemudahan akses.

2.7.2.1 Model Data Relasional

Model data relasional menggambarkan data dalam bentuk tabel-tabel,. Model data relasional berbasis pada teori matematika (Kalkulus relasional), yang merupakan dasar teori yang sangat kuat. Meski demikian, kita hanya perlu konsep yang sederhana untuk mendeskripsikan model relasional, yang relatif sederhana untuk dipahami dan digunakan oleh mereka yang tidak akrab terhadap teori yang mendasarinya, yang lebih bermanfaat saat kita melakukan pengembangan sistem informasi. Model data relasional mengandung 3 komponen inti sebagai berikut :

1. *Struktur Data*. Data-data dalam diorganisasi dalam bentuk tabel dengan baris-baris dan kolom-kolom.
2. *Manipulasi Data*. Organisasi yang sangat berdaya-guna (menggunakan *SQL/Structure Query Language*) digunakan untuk memanipulasi data-data disimpan direlasi-relasi.
3. *Integritas data*. Fasilitas-fasilitas untuk menspesifikasi aturan bisnis yang memelihara integritas data saat mereka memanipulasi.

2.7.2.2 Batasan Integritas

Model data relasional mencakup beberapa batasan yang berguna untuk memelihara keakuratan serta integritas data pada basis data. Batasan integritas yang utama adalah batasan ranah nilai (domain constraint), integritas entitas, integritas referensial, serta batasan operasional.

2.7.2.3 Mentransformasi Diagram E-R Ke Relasi

Transformasi (pemetaan) diagram E-R relasi merupakan proses yang mekanis; dalam arti ia memiliki keteraturan-keteraturan (aturan-aturan) tertentu. Pada faktanya, banyak kakas ('tool') CASE yang dapat melakukannya secara otomatis. Bagaimanapun juga, kita perlu tahu langkah-langkahnya karena alasan-alasan sebagai berikut:

1. CASE tidak dapat memodelkan data-data yang kompleks seperti relasi ternary dan relasi supertype/subtype. Dalam situasi seperti ini., kita perlu melakukan transformasi secara manual.

2. Dalam beberapa kasus, ada beberapa alternatif transformasi untuk diagram E-R yang sama. Dalam hal ini, kita perlu melakukan pilihan-pilihan untuk mendapatkan relasi-relasi yang optimal secara manual.
3. Kadang kita perlu melakukan uji kualitas pada hasil-hasil yang kita peroleh dari langkah transformasi yang dilakukan oleh kakas CASE.

2.7.2.4 Normalisasi

Secara umum, sasaran perancangan basis data adalah menghasilkan himpunan skema relasi yang mengizinkan pengguna untuk menyimpan informasi-informasi tanpa redundansi yang dikehendaki (meminimalisasi redundansi dan meningkatkan derajat konsistensi data dengan pemeliharaan integritas data) serta mengizinkan pengguna untuk mencari informasi yang dikehendaki dengan cara mudah.

Salah satu pendekatan yang dapat digunakan adalah merancang relasi-relasi menjadi bentuk normal (normal form). Untuk menentukan skema suatu relasi ada dalam bentuk yang dikehendaki, kita perlu tambahan informasi dari kondisi organisasi/perusahaan sebenarnya yang kita modelkan. Normalisasi dapat dipahami sebagai tahapan-tahapan yang masing-masing berhubungan dengan bentuk normal.

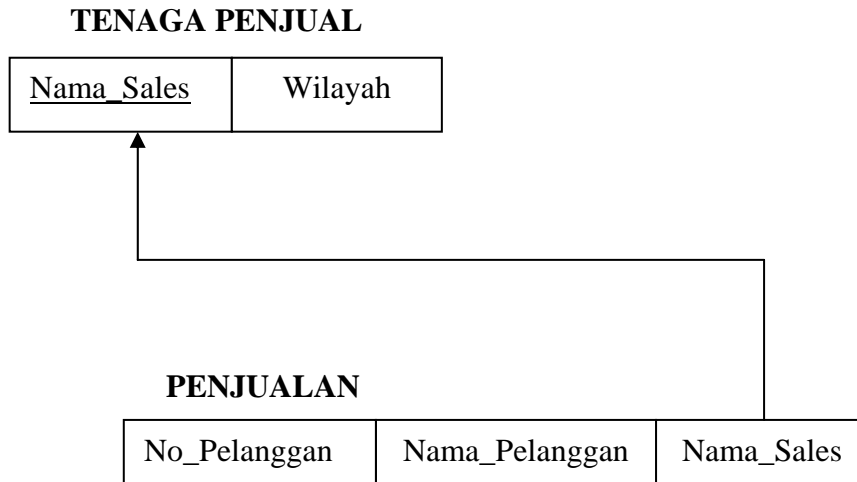
Bentuk Normalisasi Kesatu

Bentuk normal yang pertama adalah suatu bentuk relasi di mana atribut bernilai banyak (multivalued atribut) telah dihilangkan sehingga akan dijumpai nilai tunggal (mungkin saja nilai null) pada perpotongan setiap baris dan kolom pada table.

Bentuk Normalisasi Kedua (2NF)

Suatu relasi adalah dalam bentuk *normal kedua (2NF)* jika dia berada dalam bentuk normal pertama dan setiap atribut bukan kunci bergantung penuh pada kunci

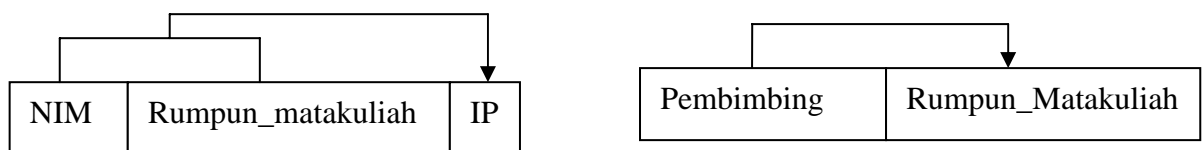
PENJUALAN (No_Pelanggan, Nama_Pelanggan, Nama_Sales, Wilayah)



Gambar Normal Ketiga (3NF)

Boyce-Codd Normal Form (BCNF)

Ketika relasi memiliki lebih dari 1 kunci kandidat (candidate key), anomali-anomali yang tersisa dari hasil penyempurnaan kebergantungan fungsional di atas telah dihilangkan. Dapat dilihat dari contoh di bawah ini :



Gambar 2 Relasi BCNF

Bentuk Normalisasi Keempat (4NF)

Relasi adalah dalam bentuk *normal keempat (4NF)* yang terbentuk jika suatu relasi ada dalam bentuk BCNF dan tidak mengandung *multivalued dependencies*.

BUKU_TEKS

MataKuliah	Buku Teks
Manajemen	Drucker
Manajemen	Peters
Manajemen Keuangan	Jones
Manajemen Keuangan	Chang

Gambar Normal Keempat (4NF)

2.7.3 Perancangan Database secara Fisik

Maksud dari perancangan basis data secara fisik adalah adalah menterjemahkan deskripsi logis data ke spesifikasi teknik untuk menyimpan (data store) serta memanggil data (data retrieval). Sasarannya adalah menciptakan perancangan untuk penyimpanan data yang menyediakan kinerja yang baik dan memastikan integritas, keamanan, serta kemampuan untuk dipulihkan (jika terjadi hal-hal yang merusak basis data).

Perancangan basis data dan berkas-berkas secara fisik menghendaki informasi-informasi yang dikumpulkan dan dihasilkan dari tahap-tahap sebelumnya. Informasi-informasi yang dibutuhkan untuk perancangan basis data dan berkas-berkas secara fisik itu mencakup (a.1) :

1. Relasi yang sudah ternormalisasi, mencakup juga perhitungan kasar volume data yang akan disimpan.
2. Definisi dan deskripsi masing-masing atribut pada setiap entitas yang ada.
3. Deskripsi tentang kapan dan di mana data digunakan, meliputi langkah-langkah pemasukkan data, pemanggilan, penghapusan, serta pembaharuan (termasuk frekuensinya).

4. Harapan dan kebutuhan tentang waktu basis data menanggapi permintaan data tertentu dari basis data, penyalinan (*back-up*), pemulihan (*recovery*), serta pemeliharaan integritas data.
5. Deskripsi teknologi (sistem basis data-DBMS/ *DataBase Management System*) yang digunakan untuk mengimplemetasikan basis data.

Perancangan basis data secara fisik membutuhkan beberapa pilihan kritis yang akan berimbas pada integritas dan kinerja dari aplikasi. Kunci-kunci untuk melakukan pilihan-pilihan mencakup :

- Pemilihan format penyimpanan (dinamakan *type data*) untuk tiap atribut dari tiap model data logika. Format dipilih untuk meminimisasi dan mengoptimalkan penggunaan ruang fisik dan untuk memaksimalkan integritas data.
- Pengelompokkan atribut-atribut dari model data dan logika ke *rekaman fisik*. Pembaca, di bab VI ini (*kosep pengembangan sistemn basis data*), akan menemukan bahwa, meskipun kolom-kolom pada tabel relasional merupakan definisi yang alami untuk isi rekaman fisik, ia tidak selalu merupakan pengelompokkan atribut yang dikehendaki dalam kaitannya dengan kinerja sistem basis data.
- Perancangan rekaman-rekaman pada memori sekunder (terutama hardisk) sehingga rekaman-rekaman secara individual maupun kelompok rekaman (kita namakan *pengorganisasian berkas*) dapat disimpan, dipanggil kembali, serta diperbaharui dengan cepat,. Pertimbangan-pertimbangan harus dilakukan juga untuk melindungi data serta melakukan pemulihan saat terjadi kesalahan-kesalahan serta perusakkan-perusakkan pada basis data.

- Memilih struktur (dinamakan *indeks* dan *arsitektur basis data*) untuk menyimpan dan menghubungkan berkas-berkas (file) sehingga pemanggilan data berlangsung dengan cara efisien.
- Menyiapkan strategi-strategi untuk menangani query pada basis data dengan tujuan mengoptimasi kinerja basis data dalam menangani query. Struktur basis data yang efisien hanya bermanfaat jika sistem basis data yang menangani query dirancang secara cerdas.

2.7.4 Kamus Data

Kamus data atau *data dictionary* atau disebut juga dengan istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat pada tahap perancangan sistem.

Pada tahap analisis, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang input, merancang laporan-laporan dan database. Kamus data dibuat berdasarkan arus data yang ada di DAD. Arus data di DAD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja. Notasi-notasi yang digunakan dalam kamus data adalah

Notasi	Arti
=	Terbentuk dari atau terdiri dari atau sama dengan
+	AND
[]	Salah satu dari (memilih salah satu dari elemen-elemen data di dalam kurung bracket ini)
	Sama dengan simbol []
M{ }M	Iterasi (elemen data di dalam kurung brace beriterasi mulai minimum N kali dan maksimum M kali)
()	Optional (elemen data di dalam kurung perenthesis sifatnya optional, dapat ada dan dapat tidak ada)
*	Keterangan setelah tanda ini adalah komentar

Gambar 2.15 Notasi Kamus Data

Berikut adalah contoh umum dari kamus data

1. Mahasiswa = {NIM, Nama_Mhs, Alamat, Tpt_lhr, Tgl_lhr}
2. Dosen = {*NID + Nama_Dos + Keahlian + Alamat}
3. mengajar = {*Kode_MK + **NID + Waktu + Ruang}

2.7.5 Isi Kamus Data

Kamus data harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini, maka kamus data harus memuat hal-hal berikut ini:

1. Nama arus data

Karena kamus data dibuat berdasarkan arus data yang mengalir di DAD, maka nama arus data juga harus dicatat di kamus data, sehingga mereka yang membaca DAD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di DAD dapat langsung mencarinya dengan mudah di kamus data

2. Alias

Alias/nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

3. Bentuk data

Bentuk dari data ini perlu dicatat di kamus data, karena dapat digunakan untuk mengelompokkan kamus data kedalam kegunaannya sewaktu perancangan sistem.

4. Arus data

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju. Keterangan arus data ini perlu dicatat di kamus data agar memudahkan mencari arus data di DAD.

5. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

6. Periode

Periode ini menunjukkan kapan arus data ini.

7. Volume

Volume yang perlu dicatat di kamus data adalah tentang volume rata-rata dan volume puncak arus data.

8. Struktur data

Struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja.

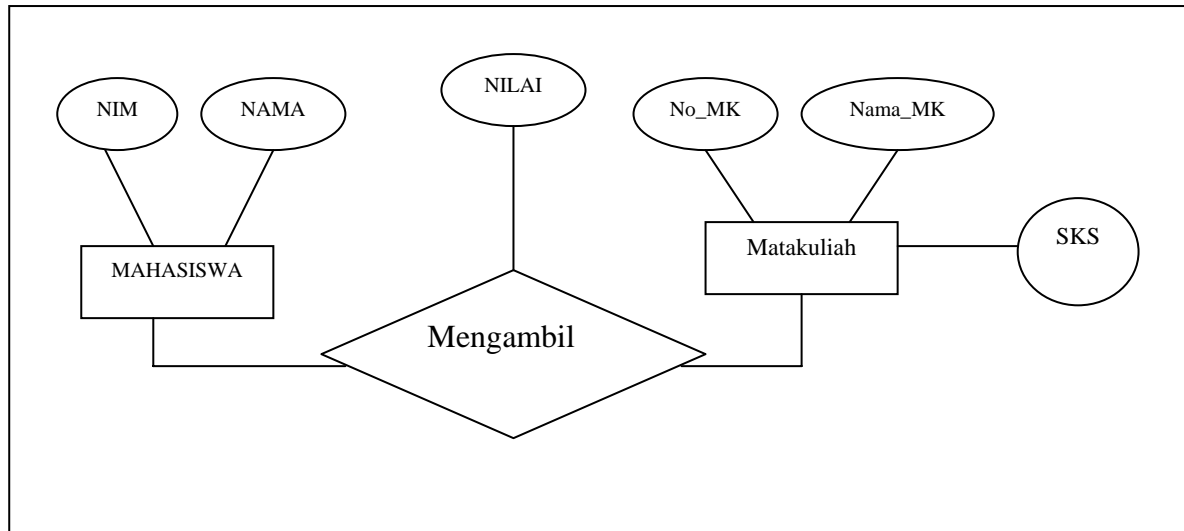
2.7.6 ERD (*Entity Relationship Diagram*)

Model data diagram hubungan entitas (ERD) dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi objek-objek dasar yang dinamakan entitas (entity) serta hubungan (relationship) antara entitas-entitas itu. Entitas adalah 'sesuatu' atau 'objek' pada dunia nyata yang dapat dibedakan satu terhadap lainnya, yang bermanfaat bagi aplikasi yang sedang kita kembangkan. Sebagai contoh, setiap orang adalah entitas dan rekening bank dapat dipertimbangkan sebagai sebuah entitas.

Entitas dalam basis data dideskripsikan berdasarkan atribut-nya. Sebagai contoh, nomor rekening membedakan suatu rekening adalah milik seseorang yang menyimpan uangnya dengan rekening milik orang lain di suatu bank tertentu dan nomor-nomor rekening tersebut merupakan atribut dari entitas yang bersangkutan. Dalam hal ini, nomor rekening secara unik membedakan sebuah rekening dengan rekening yang lainnya. Beberapa rekening mungkin memiliki saldo yang sama, tetapi mereka pasti memiliki nomor rekening yang berbeda.

Relationship adalah hubungan antara beberapa entitas. Sebagai contoh, mahasiswa memiliki orang tua; *memiliki* menjelaskan hubungan tertentu antara

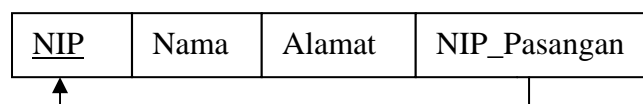
mahasiswa dengan orang tuanya. Dalam hal ini, himpunan semua entitas dengan tipe yang sama dan himpunan semua hubungan antarentitas dirujuk sebagai himpunan entitas (entity set) dan himpunan relasi (relation set).



Gambar 2.16 Penggambaran ERD

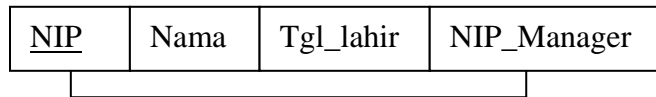
Hubungan Satu Ke Satu

Hubungan ‘satu-ke-satu’ untuk relasi berderajat satu 1 (unary relationship) adalah kasus khusus dari hubungan ‘satu-ke-banyak’. Gambar di bawah ini menunjukkan relasi karyawan **menikahi** karyawan (Asumsi yang diambil adalah: baik karyawan pria maupun karyawan wanita bekerja pada perusahaan yang sama).



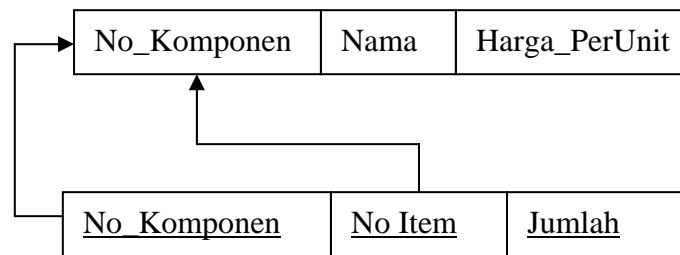
Gambar Pemetaan Relasi ‘Satu-ke-Satu’ Pada Unary Relationship

Hubungan Satu Ke Banyak



Gambar Pemetaan Relasi 'Satu-ke-Banyak' Pada Unary Relationship

Hubungan Banyak Ke Banyak



Gambar Pemetaan Relasi 'Banyak-ke-Banyak' Pada Unary Relationship

2.8 Web

World wide web atau yang biasa disebut dengan istilah website atau site mengacu pada sekumpulan file dalam web yang dapat diakses dengan browser. Web dibuat pada awal tahun 1990. Sedangkan pengertian situs web itu sendiri adalah merupakan sebuah kumpulan dari semua halaman web yang terangkum dalam satu kesatuan di internet. Sebuah situs web terdiri dari beberapa halaman yang kesemuanya saling dihubungkan dengan Hyperlink.

Situs web pribadi seluk beluk seseorang secara privasi dan mendetail, dengan tujuan yang beraneka ragam. Seperti curriculum vitae, mempromosikan ketrampilan khusus di suatu bidang tertentu dalam rangka mencari pekerjaan dll.

Sementara situs web profesional ditujukan bagi dunia kerja, baik dalam lingkup profit maupun non profit. Isinya dapat berupa segala hal yang berkaitan

dengan suatu perusahaan tertentu. Bisa berupa sejarah berdirinya, informasi produk, jadwal kegiatan dsb.

2.9 Konsep Dasar PHP

Untuk membuat website yang dinamis dan mudah diupdate setiap saat dari browser, dibutuhkan sebuah program yang mampu mengolah data dari komputer client atau dari komputer server itu sendiri sehingga mudah dan nyaman disajikan dibrowser. Salah satu program yang dapat dijalankan diserver dan cukup handal adalah PHP. PHP adalah salah satu pemrograman yang berjalan dalam sebuah web server dan berfungsi sebagai pengolah data pada sebuah server.

Dengan menggunakan program PHP, sebuah website akan lebih interkatif dan dinamis. Saat ini banyak website yang menggunakan program PHP sebagai dasr pengolahan data. Beberapa keunggulan yang dimiliki program PHP adalah :

- PHP memiliki tingkat akses yang lebih cepat.
- PHP memiliki tingkat lifecycle yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
- PHP memiliki tingkat keamanan yang tinggi.
- PHP mampu berjalan dibeberapa server yang ada, misalnya Apache, Microsoft IIS, PWS, AOLserver, phttpd, fhttpd, dan Xitami.
- PHP mampu berjalan di LINUX sebagai platform sistem operasi utama bagi PHP, namun juga dapat berjalan di FreeBSD, Unix, Solaris, Windows, dan yang lain.
- PHP mendukung akses ke beberapa database yang sudah ada, baik yang bersifat free/gratis ataupun komersial. Database itu antara lainMySQL, PosgreSQL, mSQL, Informix, dan MicrosoftSQL server.
- PHP bersifat gratis/free.

2.9.1 Aplikasi PHP dengan MySQL

PHP dan MySQL merupakan kombinasi pasangan piranti yang banyak dan umum digunakan dalam pembangunan dan pengembangan situs web yang menggunakan basis data. PHP dan MySQL dapat dikatakan merupakan pasangan perangkat lunak yang serasi untuk pengembangan aplikasi web basis data. Dalam melakukan pemrograman pada situs web dinamis, minimal dibutuhkan PHP, Apache sebagai server web, dan MySQL sebagai basis data. Konsep kerja PHP adalah ketika seseorang mengetikkan alamat di browser web, maka browser akan mengirimkan perintah tersebut ke server. Jika yang diminta adalah berkas (*file*) yang mengandung program *server-side* maka server web akan menjalankan terlebih dahulu program tersebut dan mengirimkan hasilnya ke browser. Selain itu, jika yang diminta adalah file HTML maka browser web akan langsung mengirimkan ke browser apa adanya. Server web secara default akan memberikan dokumen HTML pada setiap permintaan yang ditujukan kepadanya. Dokumen HTML dalam pengertian umum merupakan dokumen yang berisi teks yang diformat dengan tag-tag HTML yang dibuat dengan menggunakan tesk editor atau editor HTML, informasi yang disajikan pada setiap dokumen bersifat statis, jika akan ada perubahan maka dokument tersebut harus diedit dengan menggunakan editor HTML kemudian dikirimkan (*upload*) kembali kedalam server. MySQL dapat digunakan sebagai sumber basis data untuk informasi yang akan ditampilkan pada suatu halaman web. Komponen pemformatan informasi dengan menggunakan HTML dan menggunakan informasi yang akan disajikan dapat dipisahkan sehingga apabila terjadi perubahan pada data atau informasi, maka perubahan dapat dilakukan pada bagian data atau informasi saja, jika terjadi perubahan pada bagian pemformatan, maka perubahan dapat dilokalisasi pada bagian pemformatan ini.

Dalam situs web dinamis data-data disimpan pada basis data dan diakses dari basis data, dalam hal ini menggunakan MySQL sebagai basis data. Secara umum, akses ke basis data melalui tiga tahapan yaitu:

1. Koneksi ke basis data.
2. Query/permintaan data (operasi manipulasi data).
3. Pemutusan koneksi.

Untuk melakukan hubungan antara kode program PHP dengan MySQL dapat dilakukan dengan sintaks di bawah ini:

```
mysql_connect("server_name", "user_name", "password")  
mysql_db_query("database_name", "query")
```

Pada baris pertama PHP akan melakukan koneksi ke basis data berdasarkan nama server yang digunakan (*server_name*), pengguna yang diberi hak untuk mengakses basis data (*user_name*), serta kata kunci yang digunakan (*password*).

Sedangkan baris kedua digunakan untuk melakukan manipulasi terhadap data dengan memilih basis data (*database_name*) dan manipulasi yang diinginkan (*query*).

Beberapa operasi manipulasi data didalam basis data yang dapat dilakukan adalah sebagai berikut:

1. Memasukan data.

Sintaks:

```
INSERT INTO nama_tabel (field1, field2...) VALUES ('data1', 'data2', ...)
```

Contoh:

```
mysql_db_query(web, "INSERT INTO karyawan(nik,nama) VALUES ('$nik','$nama')");
```

Pada contoh diatas PHP memberikan perintah untuk memasukkan variable , \$nik, \$nama ke dalam field nim dan nama yang ada pada tabel karyawan.

2. Menampilkan data.

Sintaks:

```
SELECT field1, field2, ... FROM nama_tabel WHERE syarat1, syarat2, ...  
ORDER BY nama_field
```

Contoh:

```
$query01 = MYSQL_DB_QUERY (data, "SELECT id,nik,nama,alamat FROM  
karyawan WHERE id = '$id' ORDER BY id DESC");
```

Pada contoh diatas PHP memberikan perintah untuk mencari isi field nik,nama,alamat dari tabel karyawan dengan kondisi field id sesuai dengan variable \$id dan diurutkan secara urutan data terkecil (urutan angka) dan dari A-Z (*descending*) pada basis data data.

3. Mengubah data.

Sintaks:

```
UPDATE nama_tabel SET field1='nilai_baru', field2='nilai_baru', ...  
  
WHERE kondisi 1, kondisi 2, ... .
```

Contoh:

```
Mysql_db_query (data, "UPDATE karyawan SET nik = '$nik', nama =  
'$nama', alamat='$jurusan' WHERE id = '$id'");
```

Pada contoh diatas PHP memberikan perintah untuk merubah isi tabel karyawan dengan field `nik`, `nama`, `alamat` dirubah dengan variable `$id` pada basis data `data`.

4. Menghapus data.

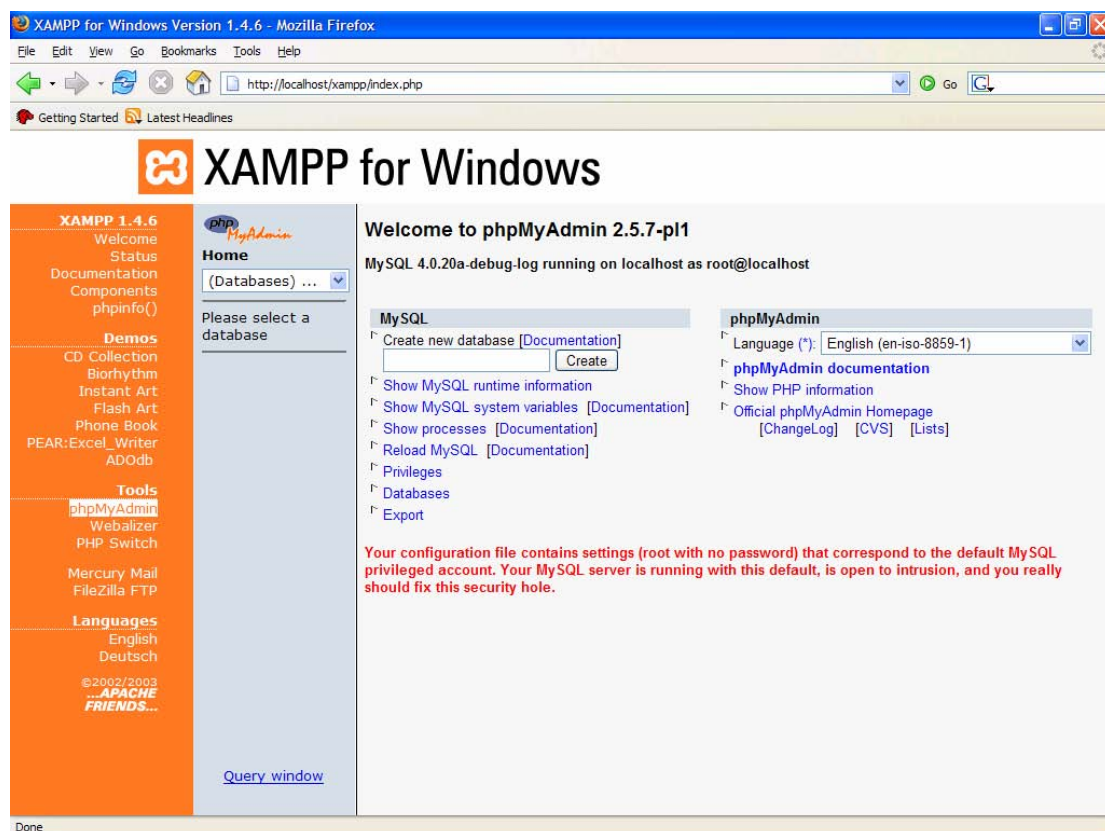
Sintaks:

```
DELETE FROM nama_tabel WHERE syarat1, syarat2, ... .
```

Contoh:

```
mysql_db_query (data, "DELETE FROM karyawan WHERE nomor = '$id'");
```

Pada contoh tersebut PHP menghapus data adalah mencari data dengan nomor yang dideklarasikan oleh variabel `$id`, kemudian dihapus oleh bentuk penulisan MySQL yang berfungsi menghapus daftar data yang dipilih.

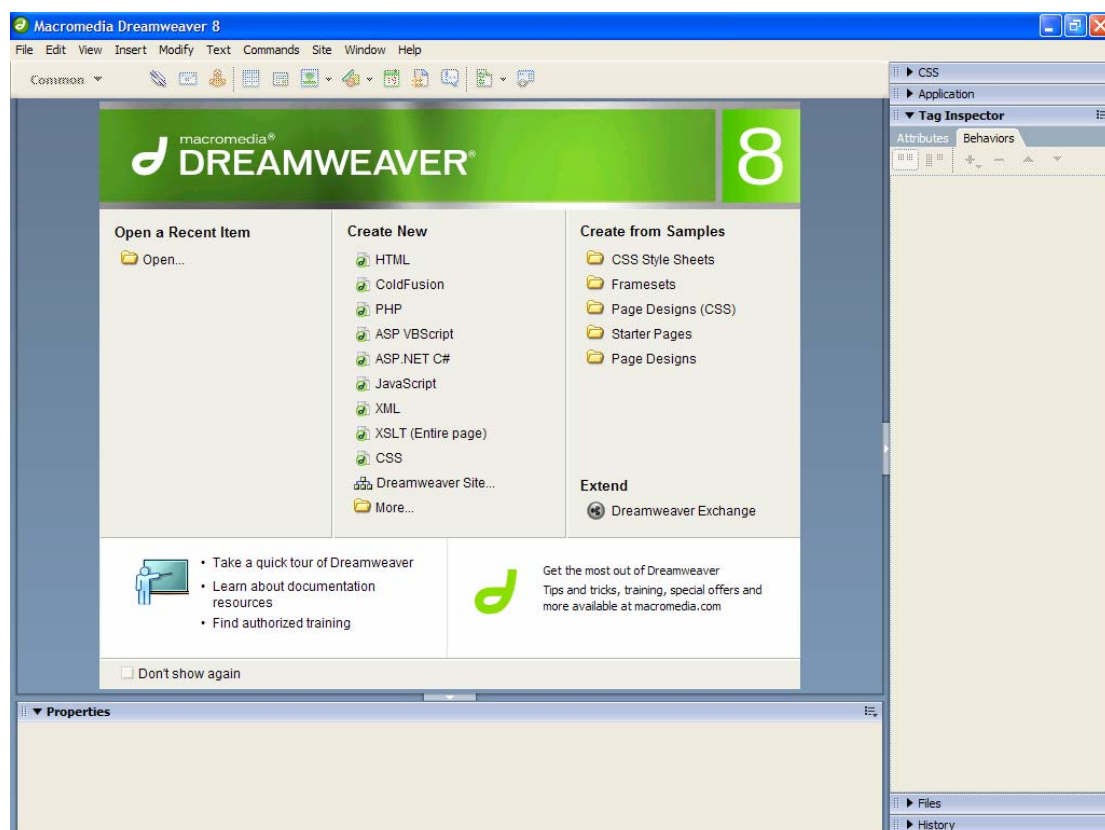


Gambar 2.17 Interface PHP & MySQL

2.10 Dreamweaver

Adalah sebuah program web editor yang digunakan untuk membuat dan mendesain web. Dreamweaver mempunyai keahlian dalam membuat dan mendesain web tanpa harus menuliskan tag-tag HTML satu persatu. Dreamweaver menggunakan metode klik dan drag yang dapat mempermudah kita dalam membuat website dengan cepat, mudah, menarik dan interaktif.

Dreamweaver juga mempunyai kemampuan untuk mendukung program *Serverside* dan *Client side*. *Server side* digunakan untuk memproses data yang berhubungan dengan server, misalnya pengolahan database. *Client side* merupakan bahasa pemrograman tambahan sekaligus sebaga pelengkap dari bahasa pemrograman lainnya.



Gambar 2.18 Interface Dreamweaver 8

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Latihan psikotes ini nantinya diharapkan akan dapat menampilkan informasi yang akan dijadikan bahan untuk menilai atau mengetahui kemampuan yang dimiliki oleh seseorang.

Adapun informasi-informasi yang akan disampaikan oleh latihan psikotes ini adalah:

1. Tes verbal merupakan tes yang digunakan untuk mengukur kemampuan bahasa yang dimiliki oleh seseorang, baik secara lisan maupun tulisan, serta untuk mengukur sejauh mana seseorang memahami hal-hal yang dibicarakan. Kunci utama dalam menyelesaikan tes ini adalah daya ingat yang tinggi dan kemampuan perbendaharaan kata. Beberapa contoh soal :

1. INOVASI

Jawaban :

- a. penemuan
- b. serangan
- c. evaluasi
- d. situasi

2. Buku-Baca Radio-....

Jawaban :

- a. dengar
- b. lihat
- c. melihat
- d. kedengaran

Untuk penilaian benar atau salah jawaban peserta di cocokkan berdasarkan Data Soal yang ada pada Data Base administrator, 1 jawaban soal yang benar di beri nilai 1 %, dan hasil nilai dapat dilihat dalam bentuk grafik.

2. Peserta yang ingin melakukan tes psikotes di haruskan melakukan *Register* terlebih dahulu untuk bisa *log in* dan mengerjakan soal-soal psikotes.
3. Informasi hasil tes merupakan menu yang dapat diakses oleh peserta untuk mengetahui hasil tes yang telah dicapai oleh seseorang.
4. Menu admin merupakan menu untuk mengolah semua data yang akan disampaikan.

3.1 Analisis

Pemahaman tentang sistem yang akan dirancang sangat diperlukan sebelum sebuah perangkat lunak dibangun, Pembangunan perangkat lunak dimulai dari tahap analisis yang dilanjutkan pada tahap perancangan.

Sistem pada latihan psikotes ini menggunakan sistem penyeleksian atau ukur kemampuan, latihan psikotes ini menyediakan sarana soal-soal psikotes yang dapat digunakan untuk menyeleksi calon pegawai yang dibutuhkan oleh perusahaan. Para calon pegawai yang akan mengikuti psikotes ini diharuskan untuk mengisi formulir pendaftaran yang telah disediakan, setelah calon pegawai mengisi formulir dan mengembalikan formulir tersebut. Selanjutnya para calon pegawai mendapatkan lembar soal dan lembar jawaban, calon pegawai harus mengerjakan soal-soal psikotes sampai batas waktu yang telah ditentukan oleh lembaga atau perusahaan yang bersangkutan. Setelah waktu yang ditentukan telah habis para calon pegawai mengembalikan lembar jawaban kepada pengawas tes, yang nantinya lembar jawaban tersebut akan diberikan kembali kepada pihak yang bersangkutan. Pihak yang

bersangkutan akan membandingkan jawaban calon pegawai dengan jawaban yang benar. Hasil dari proses membandingkan tersebut adalah nilai dan tingkat kemampuan dari calon pegawai.

Permasalahan yang dihadapi lembaga atau pihak yang bersangkutan :

1. Pembuatan soal psikotes yang membutuhkan biaya untuk memfoto kopi soal psikotes dan lembar jawabannya.
2. Penilaian tes psikotes calon pegawai masih membutuhkan waktu yang cukup lama, karena pihak yang bersangkutan harus memasukan lembar jawaban peserta ke dalam program penilaian terlebih dahulu untuk mendapatkan hasil dan tingkat kemampuan dari peserta.

Dari permasalahan yang dihadapi oleh lembaga atau perusahaan, dapat diselesaikan dengan membuat sebuah aplikasi psikotes. Dimana latihan psikotes ini memiliki fasilitas antara lain :

1. Pembuatan soal langsung pada aplikasi komputer.
admin memasukan soal-soal psikotes beserta jawabannya ke aplikasi, kemudian aplikasi akan menyimpan masukan dari admin ke dalam basis data. Soal-soal yang telah dimasukan admin akan ditampilkan pada halaman soal peserta psikotes..
2. Merubah dan menghapus soal
admin dapat merubah soal psikotes kemudian soal-soal yang telah dirubah admin akan disimpan kedalam basis data. Selain merubah soal, admin dapat menghapus soal secara satu per satu maupun secara keseluruhan.

3. Penentuan nilai dan tingkat kemampuan dapat dilakukan dengan cepat.

Penilaian akan dilakukan ketika peserta psikotes selesai mengerjakan soal-soal psikotes. Aplikasi ini akan membandingkan jawaban yang dimasukan oleh peserta psikotes dengan jawaban yang benar. Hasilnya berupa nilai dan tingkat kemampuan, hasil tersebut dalam bentuk laporan, berisi nama calon mahasiswa.

4. Laporan hasil dan tingkat kemampuan.

Selain dapat dilihat oleh admin, laporan hasil dan tingkat kemampuan dapat juga dilihat oleh peserta. Laporan hasil dan tingkat kemampuan akan tampil secara menyeluruh, artinya hasil dan tingkat kemampuan peserta akan di tampilkan dan diurutkan berdasarkan hasil.

5. Daftar calon peserta psikotes.

Para calon peserta psikotes memasukan *username* dan *password* untuk dapat masuk ke halaman soal.

Adapun data-data yang dibutuhkan untuk membuat aplikasi psikotes antara lain :

1. Data peserta memiliki 6 buah *atribut*, antara lain :

- a. Uname memiliki type data *varchar* dengan lebar *field* maksimum 25. *uname(username)* harus disesuaikan dengan lebar *field*.
- b. Pswd memiliki *type* data *varchar* dengan lebar *field* maksimum 15, *pswd(password)* harus disesuaikan dengan lebar *field*.
- c. Nama memiliki *type* data *varchar* dengan lebar *field* maksimum 30. Apabila terdapat nama yang melebihi lebar *field* maka nama akan disingkat dan disesuaikan dengan lebar *field*.

- d. Jkel memiliki *type* data *varchar* dengan lebar *field* maksimum 8, *atribut* ini berisi jenis kelamin dari setiap peserta.
 - e. Alamat memiliki *type* data *varchar* dengan lebar *field* 100, lebar maksimumnya sengaja ditetapkan lebih besar dari atribut lainnya karena data alamat umumnya lebih panjang.
 - f. E-mail memiliki *type* data *varchar* dengan lebar *field* maksimum 50, angka 50 ini merupakan perkiraan bahwa untuk alamat email tidak akan melebihi 50.
2. Data admin memiliki 3 buah *atribut*, antara lain:
- a. *Username* memiliki *type* data *varchar* dengan lebar *field* 15.
 - b. *Pswd* memiliki *type* data *varchar* dengan lebar *field* maksimum 15.
 - c. Nama admin memiliki *type* data *varchar* dengan lebar *field* maksimum 25.
3. Data hasil memiliki 3 buah *atribut*, antara lain :
- a. *Uname* memiliki *type* data *varchar* dengan lebar *field* maksimum 25, *uname* ini berasal dari tabel peserta.
 - b. *Nama* memiliki *type* data *varchar* dengan lebar *field* maksimum 30, *nama* ini berasal dari tabel peserta.
 - c. *Nilai* memiliki *type* data *smallint* nilai yang digunakan dalam aplikasi ini berkisar 0-300, jika memakai *type* data *int* maka *range* yang disediakan oleh *type* data *int* banyak yang tidak terpakai, mengakibatkan ruang *memory* menjadi kosong.

3.1.1 Proses yang terdapat dalam latihan psikotes

Pada latihan psikotes ini memiliki 4 proses antara lain :

- a. Menyimpan soal.
- b. Membandingkan jawaban peserta dengan jawaban yang benar.

- c. Memvalidasi *username* dan *password*.
- d. Membuat laporan hasil psikotes peserta.

3.1.2 Keluaran Sistem

Dalam aplikasi psikotes memiliki 2 data antara lain :

- a. Informasi *dependent* (informasi yang dihasilkan pada proses).

Contohnya :

- 1) Data nilai peserta psikotes.

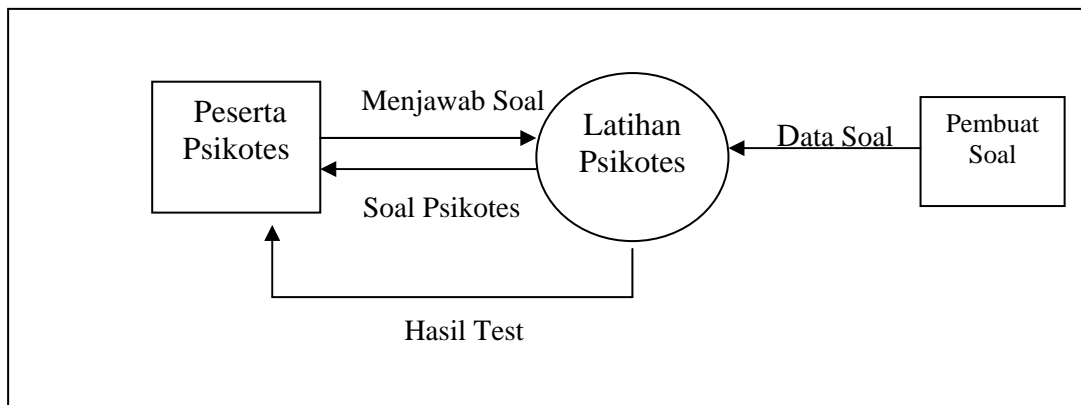
- b. Data *independent* (data yang tidak tergantung pada proses), sebagai arsip.

Contohnya :

- 1) Data soal.
- 2) Data peserta.

3.1.3. Diagram Konteks

Diagram tingkat konteks yang terdapat pada aplikasi psikotes memiliki 2 entitas yaitu: peserta dan admin. Diagram tingkat konteks ini memiliki satu proses yaitu proses aplikasi psikotes. Diagram ini biasanya juga dipakai untuk mendokumentasi proses yang akan memperlihatkan sistem dari sebuah proses dengan tujuan memberikan pandangan umum dari proses tersebut. **Gambar 3.1.** memperlihatkan aliran data diagram tingkat nol untuk aplikasi psikotes.



Gambar 3.1 Diagram Konteks.

Dalam **Gambar 3.1.** diagram konteks diatas memberikan gambaran bahwa sistem Latihan Psikotes ini berinteraksi dengan dua terminator yaitu : Peserta, dan Admin. Tanda panah diatas menyatakan masukan dan juga keluaran dari sistem tersebut. Sebagai contoh peserta menjawab semua soal-soal yang ada, kemudian peserta dapat melihat hasil yang diperolehnya. Admin selain dapat melihat hasil psikotes, admin memiliki peranan yang sangat penting yaitu memasukan data-data yang dibutuhkan oleh aplikasi ini, yaitu: data soal dan data jawaban.

3.2 Rancangan Sistem

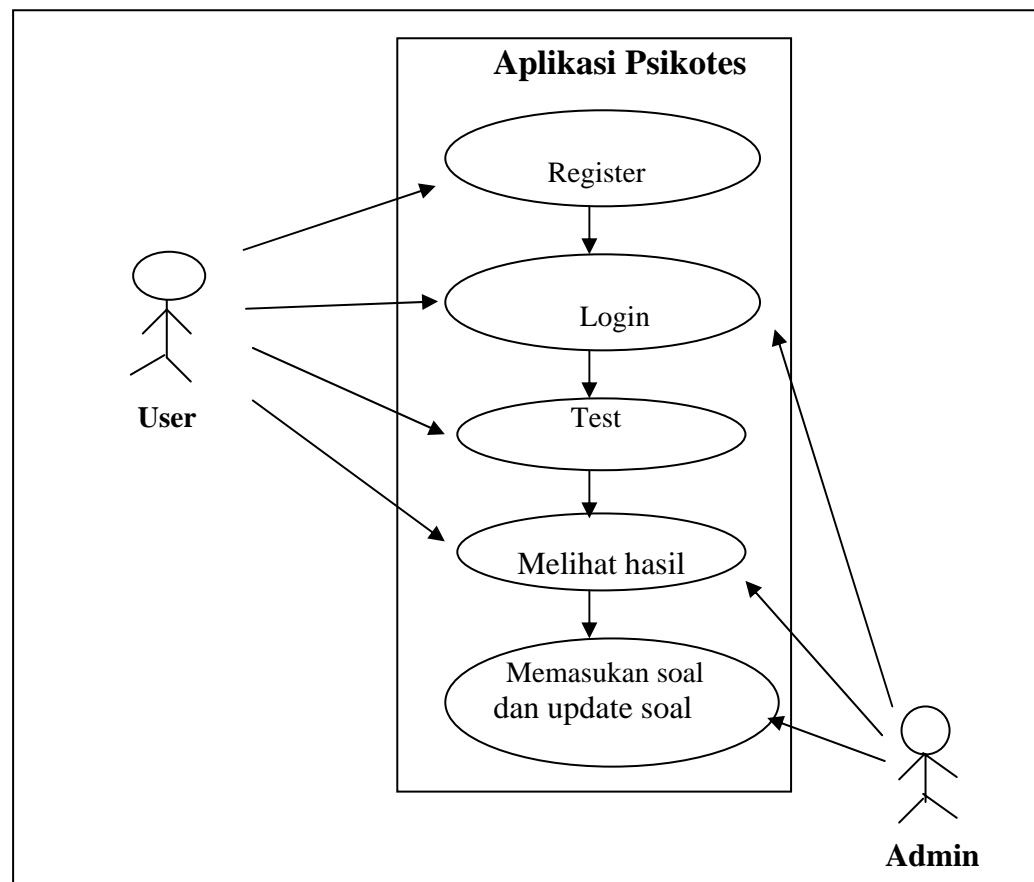
Berdasarkan hasil analisis, penulis akan membuat sebuah latihan psikotes dengan menggunakan php dan mysql. Secara garis besar latihan psikotes ini memiliki fasilitas yang telah dijelaskan pada sub bab analisis tepatnya pejelasan pemecahan masalah.

3.2.1 Rancangan Unified Modelling Language (UML)

UML dapat memenuhi kebutuhan peserta dengan memodelkan sistem yang dibangun dengan menggunakan beberapa diagram, yaitu diagram use case, diagram aktifitas, dan diagram sequence.

3.2.1.1 Diagram Use Case

Gambar 3.2. menunjukkan diagram use case aplikasi psikotes dengan menggunakan php dan mysql.



Gambar 3.2 Diagram use case aplikasi psikotes.

A. Sisi Peserta

Berikut ini akan menjelaskan **Gambar 3.2** dari sisi diagram use case yang dilakukan peserta :

1. Use case melakukan Register ;

Nama Use Case	Register
Deskripsi Singkat	peserta harus melakukan register pada saat menggunakan sistem, aktifitas yang dilakukan peserta pada saat register telah diatur oleh sistem.
Aktor	Peserta
Pra Kondisi	Peserta melakukan register
Tindakan Utama	<ol style="list-style-type: none">Peserta melakukan link register.Sistem akan menampilkan halaman register.Peserta diminta memasukan data oleh sistem.Sistem akan mengecek data pesertaSistem akan menyimpan data peserta kedalam basis data.
Tindakan alternatif	<ul style="list-style-type: none">Pada langkah d, sistem akan mengecek apakah semua data telah diisi, jika belum peserta akan kembali kelangkah c.Pada langkah e, jika terdapat data yang sama atau ganda didalam basis data maka peserta akan kembali kelangkah c.
Pasca Kondisi	Jika peserta telah berhasil melakukan register, maka dapat <i>Log In</i> .

2. Use case melakukan *Log In* ;

Nama Use Case	<i>Log In</i>
Deskripsi Singkat	peserta harus melakukan login pada saat menggunakan sistem, sistem akan mengatur aktifitas yang dilakukan peserta pada saat login.
Aktor	Peserta
Pra Kondisi	Peserta melakukan <i>Log In</i>

Tindakan Utama	<ul style="list-style-type: none"> a. Peserta memilih link login. b. Sistem akan menampilkan halaman login. c. Peserta diminta untuk mengisi username dan password oleh sistem. d. Sistem akan mengecek data peserta. e. Sistem akan menampilkan halaman soal psikotes.
Tindakan alternatif	Pada langkah d, sistem akan mengecek username dan password dari peserta, jika username dan password peserta tidak terdaftar didalam basis data, peserta akan kembali kelangkah d.
Pasca Kondisi	Jika peserta telah berhasil <i>Log In</i> , maka dapat mengerjakan soal psikotes.

3. Use case melakukan *Test* ;

Nama Use Case	<i>Test</i>
Deskripsi Singkat	peserta akan masuk kehalaman soal psikotes.
Aktor	Peserta
Pra Kondisi	-
Tindakan Utama	<ul style="list-style-type: none"> a. Peserta akan menghadapi soal psikotes. b. Sistem akan menampilkan soal psikotes yang terdapat didalam basis data. c. Peserta akan menjawab soal dengan memilih jawaban yang benar. d. Sistem akan mengecek jawaban peserta dengan jawaban yang benar di dalam basis data.
Tindakan alternatif	-
Pasca Kondisi	Jika peserta telah mengerjakan soal psikotes, maka dapat melihat hasil test.

4. Use case melihat Hasil ;

Nama Use Case	Hasil
Deskripsi Singkat	peserta akan melihat hasil yang diperoleh setelah melakukan psikotes.
Aktor	Peserta
Pra Kondisi	-

Tindakan Utama	<ul style="list-style-type: none"> a. Peserta memilih link hasil. b. Sistem akan memanggil data hasil psikotes peserta yang terdapat didalam basis data. c. Sistem akan menampilkan hasil psikotes peserta.
Tindakan alternatif	peserta yang tidak ingin melihat hasil psikotesnya dapat keluar dari halaman hasil dengan memilih link logout.
Pasca Kondisi	Setelah melihat hasil test, maka peserta dapat log out

B. Sisi Admin

Berikut ini akan menjelaskan Gambar 3.6 dari sisi diagram use case yang dilakukan admin:

1. Use case melakukan *Log In*;

Nama Use Case	<i>Log In</i>
Deskripsi Singkat	Admin dapat melakukan <i>log in</i> untuk mengakses sistem pada aplikasi psikotes
Aktor	Admin
Pra Kondisi	-
Tindakan Utama	<ul style="list-style-type: none"> a. Admin memilih link login. b. Sistem akan menampilkan halaman login. c. Admin diminta untuk mengisi username dan password oleh sistem. d. Sistem akan mengecek data Admin. e. Sistem akan menampilkan halaman Admin.
Tindakan alternatif	Pada langkah d, sistem akan mengecek username dan password dari Admin, jika username dan password Admin tidak terdaftar didalam basis data, Admin akan kembali kelangkah d.
Pasca Kondisi	Jika Admin telah berhasil <i>Log In</i> , maka dapat mengakses sistem pada aplikasi psikotes.

2. Use case melihat Hasil ;

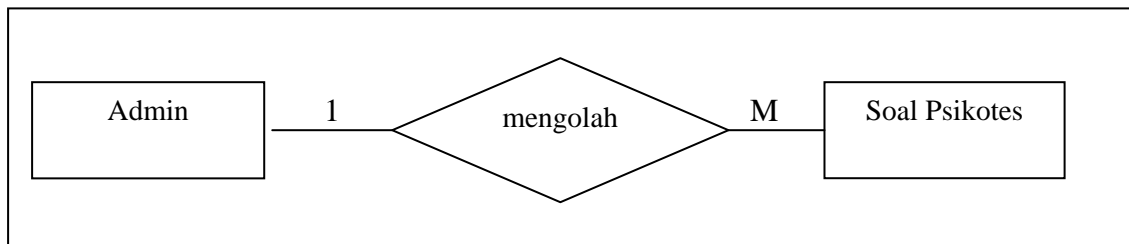
Nama Use Case	Melihat hasil
Deskripsi Singkat	Admin dapat melihat hasil tes dari peserta
Aktor	Admin
Pra Kondisi	-
Tindakan Utama	a. Admin memilih link hasil. b. Sistem akan memanggil data hasil psikotes peserta yang terdapat didalam basis data. c. Sistem akan menampilkan hasil psikotes peserta.
Tindakan alternatif	-
Pasca Kondisi	Setelah melihat hasil test, maka Admin dapat log out

3. Use case melakukan *Update* soal ;

Nama Use Case	<i>Update</i> soal
Deskripsi Singkat	Admin dapat memasukan soal dan meng- <i>update</i> soal.
Aktor	Admin
Pra Kondisi	admin memilih link soal untuk dapat meng- <i>update</i> jenis soal, daftar soal, soal, jumlah soal dan jawaban soal.
Tindakan Utama	a. Admin memilih link soal b. Sistem menampilkan halaman soal c. Admin dapat memilih link jenis soal, daftar soal, soal, jumlah soal dan jawaban soal. d. Sistem menampilkan halaman yang dipilih oleh admin.
Tindakan alternatif	-
Pasca Kondisi	Setelah admin update soal lalu di simpan ke data base.

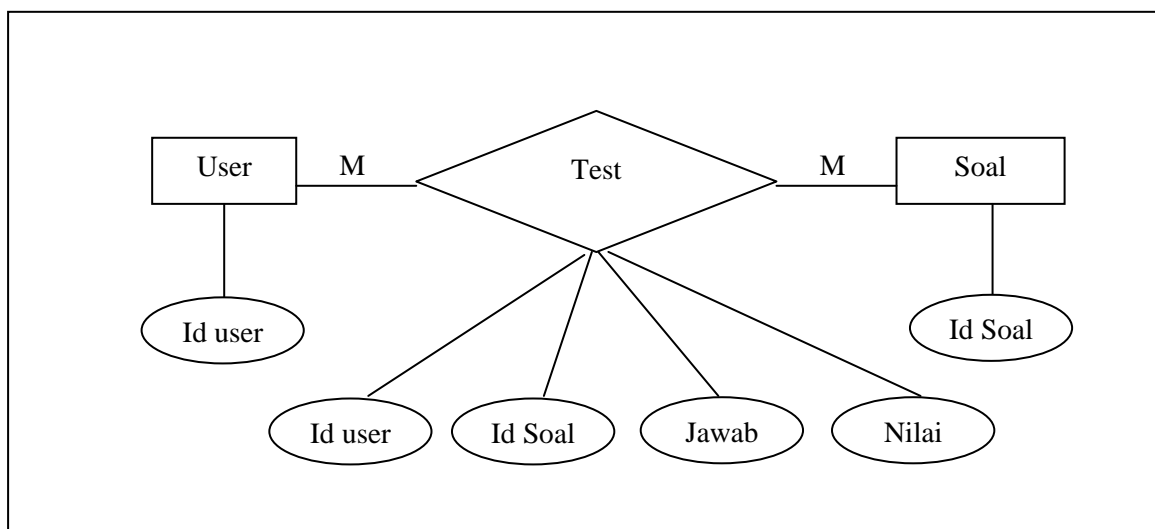
3.2.2 Entity Relationship Diagram (ERD)

Pada perancangan DFD terdapat entitas yang saling tergantung dengan entitas yang lain, pada **Gambar 3.3** menggambarkan diagram hubungan antar entitas (*Entity Relationship Diagram/ERD*). Digambarkan relasi antara admin dengan soal, admin menambahkan data soal, mengedit data soal, maupun menghapus data soal. Hanya admin yang dapat melakukan semua itu. Admin dapat melakukan hal itu kepada semua data soal yang ada.



Gambar 3.3 Relasi antar admin dengan soal psikotes.

Berikutnya akan dijabarkan relasi yang terjadi pada beberapa entitas, seperti yang telah diketahui bahwa ada beberapa entitas yang saling berhubungan, antara lain entitas Admin, Soal psikotes, Peserta, dan Hasil. Untuk lebih jelasnya perhatikan **Gambar 3.4**.



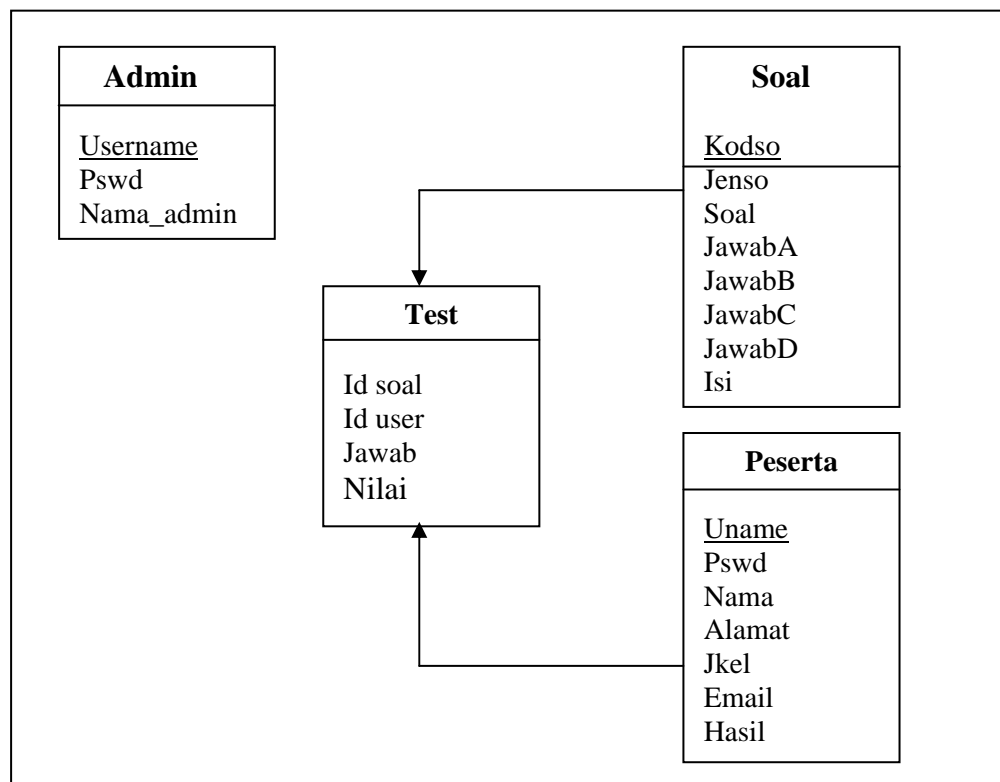
Gambar 3.4 Relasi antar Soal psikotes, Peserta dan Hasil.

Pada **Gambar 3.4** menggambarkan hubungan antar entitas. Digambarkan relasi antar admin, soal, peserta, dan hasil. Admin dapat membuat, menambah, dan menghapus soal. Setiap peserta dapat mengerjakan soal psikotes, tetapi tidak dapat merubah atau membuat soal. Setiap peserta akan mendapatkan hasil yang diperolehnya setelah mengerjakan soal psikotes.

Kamus data berdasarkan ERD aplikasi psikotes pada gambar 3.5 adalah sebagai berikut :

1. Entitas admin = {username + pswd + nama_admin }
2. Entitas soal = {kode_soal + jenis soal + soal + jawabA + jawabB + jawabC + jawabD + isi }
3. Entitas peserta = {uname + pswd + nama + alamat + jkel + email + nilai + hasil }
4. Entitas test = {idsoal+iduser+jawab+nilai }

Berdasarkan ERD diatas dapat dikembangkan menjadi model data, model data fisik tersebut dapat dilihat pada **Gambar 3.5**.



Gambar 3.5 Model data fisik aplikasi psikotes.

3.2.3 Struktur Tabel

Berdasarkan ERD yang telah dibangun pada tahap perancangan aplikasi menghasilkan perancangan tabel sebagai berikut :

1. Tabel Peserta ;

Tabel peserta, tabel yang berisikan biodata para peserta setelah mendaftarkan diri pada aplikasi psikotes pada aplikasi psikotes.

Tabel 3.1 Perancangan tabel peserta

Nama field	Type	Length	Key	keterangan
uname	varchar	25	Primary key	username
pswd	varchar	15		password
nama	varchar	30		Nama peserta
jkel	varchar	8		Jenis kelamin
alamat	varchar	100		
email	varchar	50		
Nilai	smallint	4		

2. Tabel Soal ;

Tabel soal, tabel yang digunakan oleh admin untuk memasukan soal psikotes

Tabel 3.2 perancangan tabel soal

Nama field	Type	Length	Key	keterangan
kodso	varchar	6	Primary key	Kode soal
jenso	varchar	10		Jenis soal
No_soal	Tinyint	4		Nomor soal
soal	Text			soal
jawabA	varchar	40		Jawaban a
jawabB	varchar	40		Jawaban b
jawabC	varchar	40		Jawaban c
jawabD	varchar	40		Jawaban d
Jawaban benar	varchar	1		Jawaban

3. Tabel Hasil ;

Tabel hasil, tabel yang berisikan hasil tes dari peserta

Tabel 3.3 Perancangan tabel hasil

Nama field	Type	Length	Key	keterangan
NomorRec			Primary Key	AutoNumber
uname	varchar	25	Primary key	Username
Kodso	varchar	6	Forend Key	Kode Soal
Jawab	varchar	1		

4. Tabel Admin ;

Tabel admin, tabel yang melakukan *log in*

Tabel 3.4 Perancangan tabel admin

Nama field	Type	Length	Key	keterangan
Username	varchar	25	Primary Key	username
Pswd	varchar	15		password
Nama_admin	varchar	20		Namaadmin

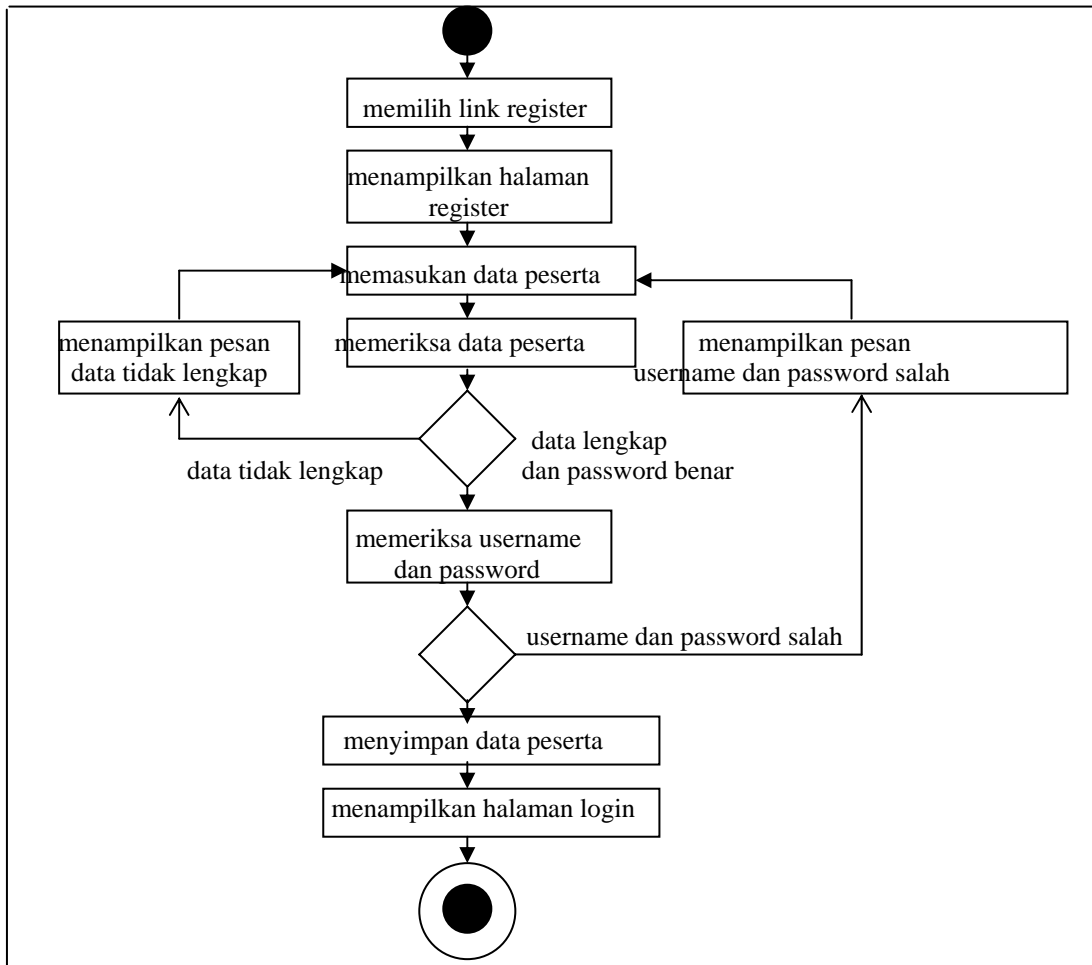
Pada rancangan tabel diatas, terdapat kolom yang menggunakan tipe data *smallint*, seperti nilai. Karena tabel tersebut memiliki perkiraan laju tidak terlalu besar, tipe data *tinyint* lebih menghemat memori karena jumlah bit lebih kecil dari pada tipe data integer. Dengan alasan tersebut tipe data *tinyint* sesuai digunakan dalam aplikasi ini.

Sedangkan tipe data *varchar* digunakan pada beberapa kolom, seperti *uname*, *password*, *nama*, *jkkel*, *alamat*, *email*, *jawabA*, *jawabB*, *jawabC*, *jawabD*, dan *isi*. Karena tipe data *varchar* memiliki penyimpanan memori bergantung pada berapa karakter yang dipakai, maka akan lebih menghemat penggunaan memori.

3.2.4 Diagram Aktifitas

Diagram aktifitas berfungsi untuk memodelkan aliran kendali dari suatu operasi yang menekankan aliran kontrol dari suatu operasi. **Gambar 3.6** menunjukkan diagram aktifitas registrasi.

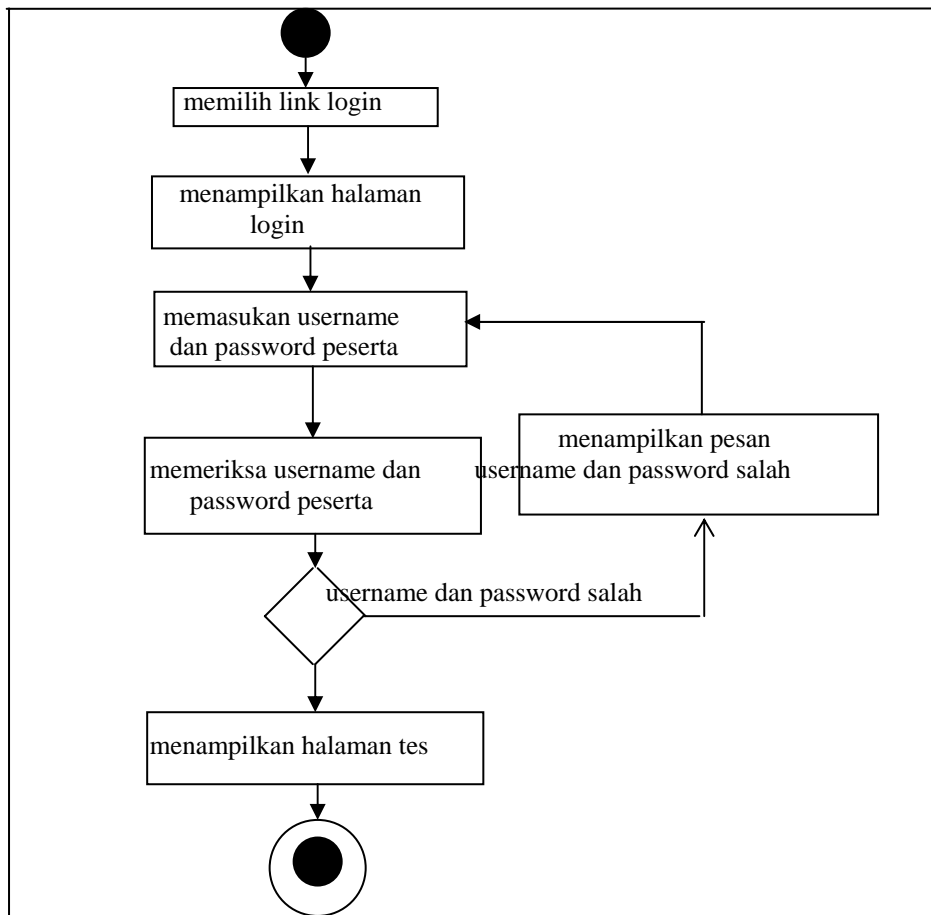
a. Diagram aktifitas registrasi



Gambar 3.6. Diagram aktifitas register

Pada **Gambar 3.6.** diagram register dimulai dari peserta memilih link register dan sistem akan menampilkan halaman register. Kemudian peserta memasukkan data-data peserta yang diminta oleh sistem, sistem akan memeriksa semua data-data yang dimasukkan oleh peserta. Jika data-data peserta tidak lengkap maka sistem akan menampilkan pesan salah dan peserta diminta untuk kembali mengisi data-data. Tetapi jika data-data peserta lengkap maka aliran akan memeriksa username dan password peserta. Jika username dan password peserta telah terdapat pada basis data, maka sistem akan menampilkan pesan username dan password salah. Setelah username dan password peserta benar maka sistem akan menampilkan halaman login.

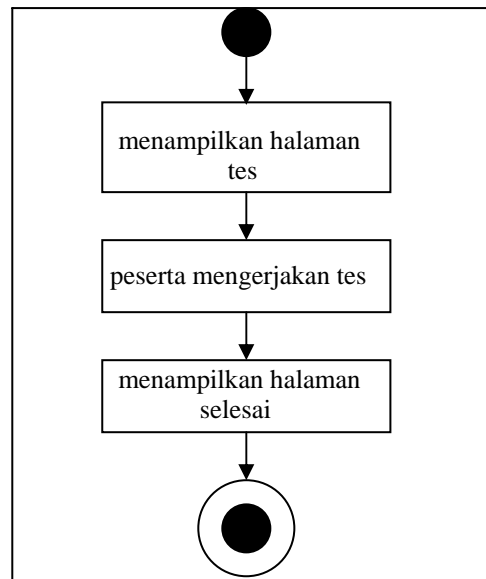
b. Diagram aktifitas login



Gambar 3.7. Diagram aktifitas login.

Pada **Gambar 3.7.** diagram *login* dimulai dari peserta memilih *link login* dan sistem menampilkan halaman *login*, kemudian peserta diminta untuk mengisi *username* dan *password* peserta, kemudian sistem akan memeriksa *username* dan *password* peserta. Jika salah peserta akan diminta oleh sistem untuk kembali mengisi *username* dan *password* dengan benar. Setelah *username* dan *password* benar maka sistem akan menampilkan halaman tes.

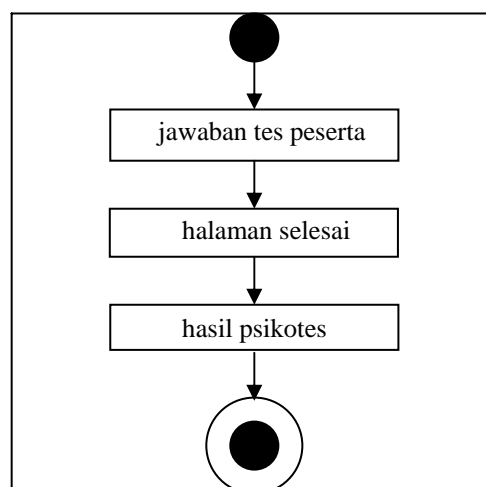
c. Diagram aktifitas tes psikotes



Gambar 3.8. Diagram aktifitas tes psikotes.

Setelah peserta login, sistem akan menampilkan halaman tes, kemudian peserta akan mengerjakan tes dengan memilih jawaban yang benar, setelah peserta selesai mengerjakan tes, sistem akan menampilkan halaman selesai.

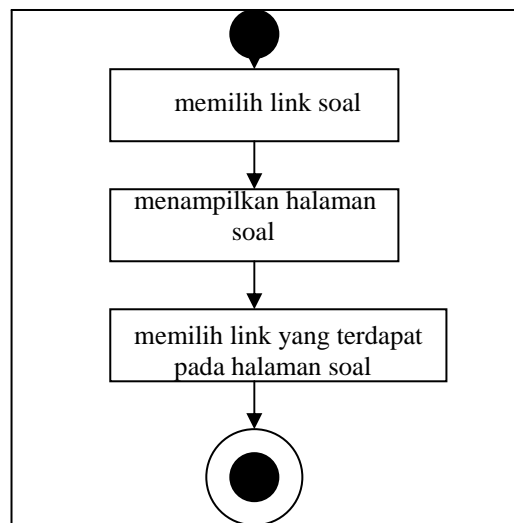
d. Diagram aktifitas hasil psikotes



Gambar 3.9. Diagram aktifitas hasil nsikotes.

Setelah peserta selesai mengerjakan tes, maka sistem akan menampilkan halaman selesai, kemudian pada halaman selesai terdapat *link* untuk melihat hasil dari psikotes peserta, kemudian sistem akan menampilkan hasil psikotes yang diperoleh oleh peserta.

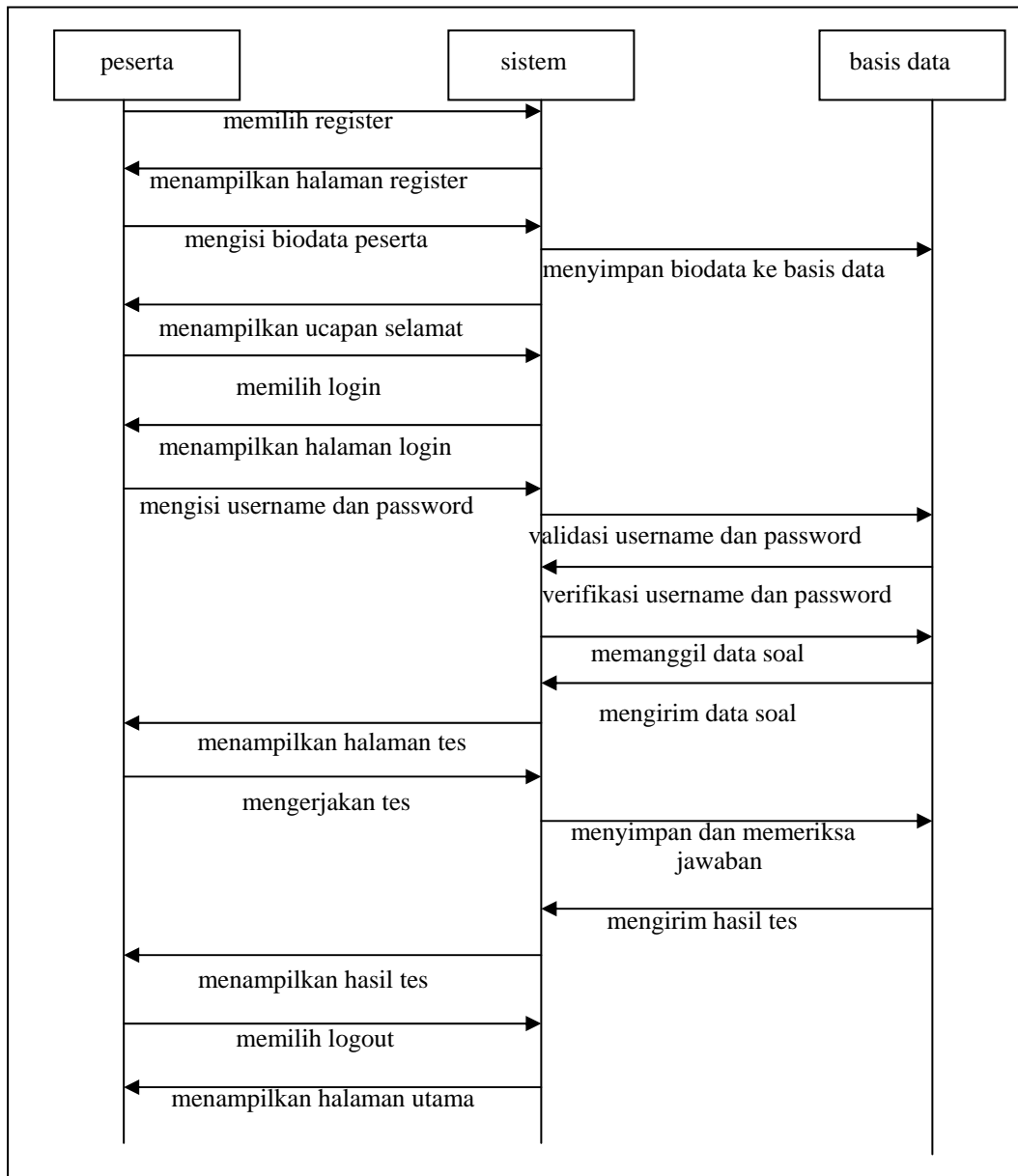
e. Diagram aktifitas admin soal psikotes



Gambar 3.10. Diagram aktifitas admin soal psikotes

Admin memilih link data soal, dan sistem akan menampilkan halaman data soal, pada halaman data soal terdapat beberapa *link* yang dapat dipilih oleh admin, diantaranya: kode soal, edit soal, dan hapus soal

3.2.5 Diagram Sequence

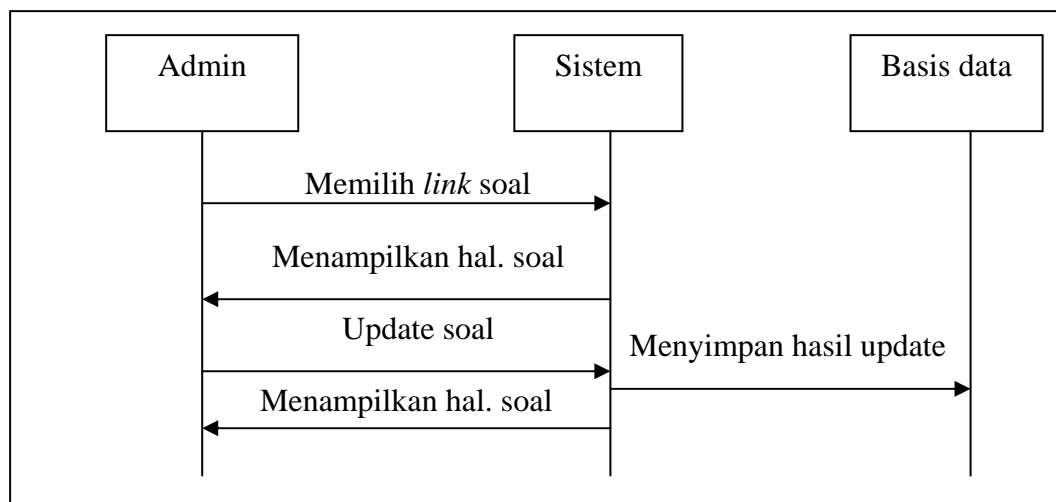


Gambar 3.11. Diagram sequence aplikasi psikotes.

Peserta memilih register, kemudian sistem akan menampilkan halaman register, peserta mengisi biodata yang diminta oleh sistem. Sistem akan menyimpan biodata peserta ke dalam basis data, setelah biodata tersimpan sistem akan menampilkan ucapan selamat. Setelah biodata peserta tersimpan, peserta memilih login, kemudian sistem akan menampilkan halaman login. Pada halaman login ini

peserta wajib untuk mengisi *username* dan *password*, sistem akan melakukan *validasi username* dan *password* yang dimasukan oleh peserta, apakah *username* dan *password* tersebut berada didalam basis data atau tidak. Kemudian basis data akan melakukan *verifikasi username* dan *password* tersebut, jika sesuai sistem akan memanggil data soal didalam basis data, dan basis data akan mengirim data soal ke sistem, kemudian sistem akan menampilkan halaman tes. Peserta mengerjakan tes, dan sistem akan menyimpan dan memeriksa jawaban dari peserta didalam basis data, kemudian basis data akan mengirim hasil tes ke sistem, dan sistem akan menampilkan hasil tes.

Setelah hasil tes diketahui peserta memilih logout, dengan tujuan orang yang tidak bertanggung jawab tidak dapat masuk kehalaman hasil, dan untuk menjaga kerahasiaan hasil yang diperoleh peserta. Kemudian sistem akan menampilkan halaman utama.



Gambar 3.12. Diagram sequence *update soal*.

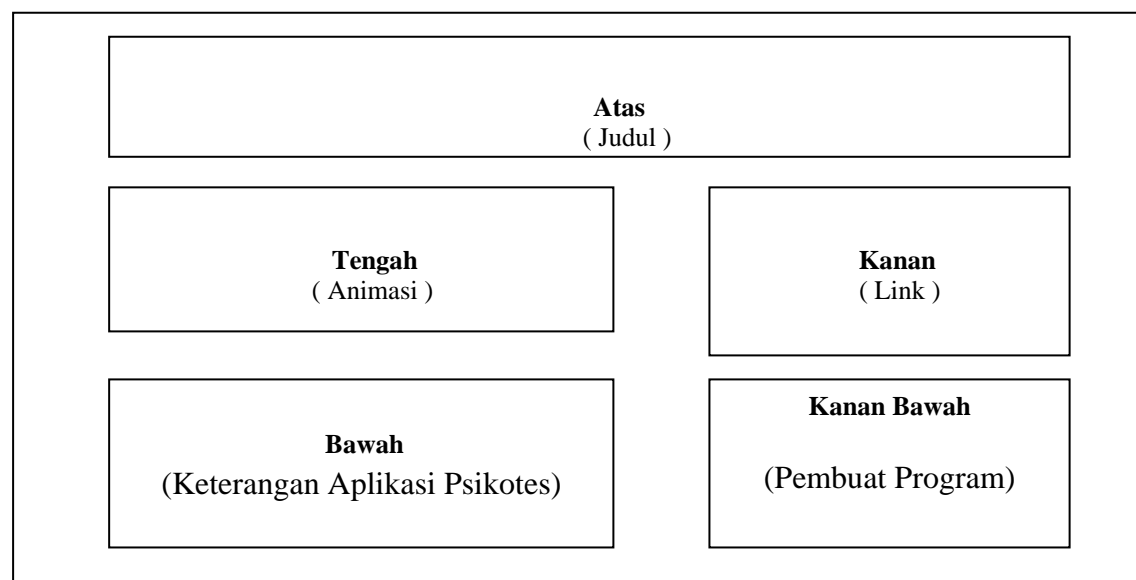
Pada **Gambar 3.12.** dapat dijelaskan bahwa hanya admin yang dapat melakukan *update soal*, pada tahap pertama admin terlebih dahulu memilih *link soal*, kemudian sistem akan menampilkan halaman soal yang dituju. Kemudian admin dapat meng-

update (jenis soal, daftar soal, soal, jumlah soal dan jawaban soal), lalu hasil *update*-an akan tersimpan ke dalam basis data. Dan sistem akan menampilkan kembali halaman yang telah di pilih oleh admin.

3.2.6 Rancangan *Interface*

Perancangan ini merupakan perancangan antar muka tahapan terakhir didalam rancangan aplikasi psikotes ini. Interaksi manusia komputer (IMK) dibutuhkan didalam rancangan ini agar dibuat *user friendly* dengan pengguna aplikasi, untuk itu dibutuhkan rancangan antar muka yang baik dengan tujuan agar mempermudah bagi para pengguna aplikasi ini.

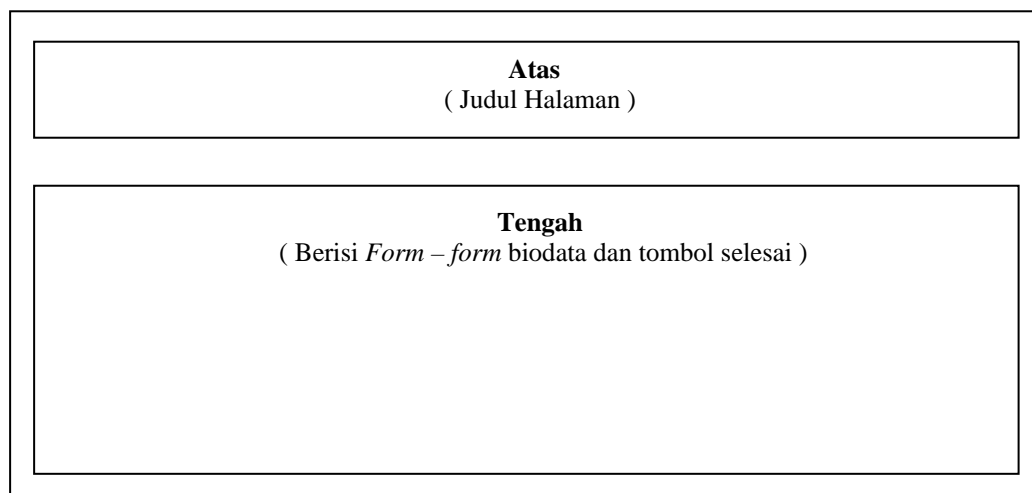
Pada subbab ini akan dijelaskan mengenai tampilan layar utama pada setiap halaman yang ada pada aplikasi psikotes ini, pada setiap halaman terdapat tabel–tabel yang berfungsi agar tampilan aplikasi ini rapih dan mudah dimengerti bagi para pengguna aplikasi ini. Berikut ini adalah rancangan halaman utama untuk para pengguna aplikasi.



Gambar 3.13 Rancangan Halaman Utama.

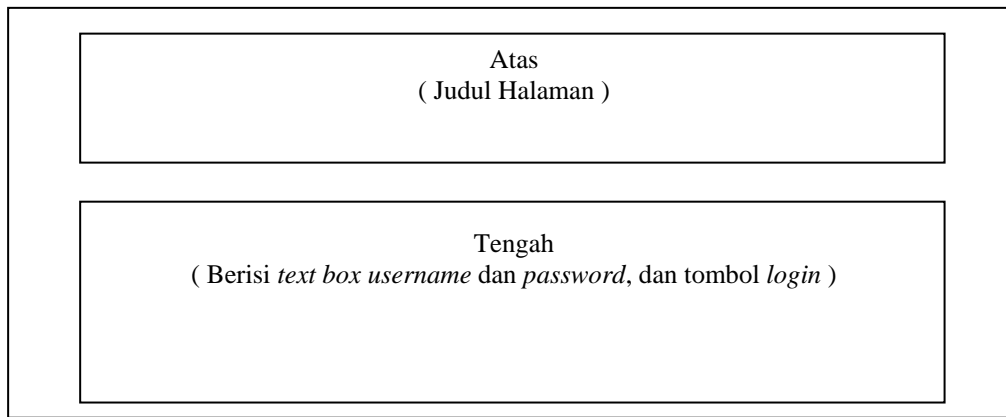
Tabel kanan pada halaman ini digunakan para peserta psikotes untuk masuk kehalaman daftar dan Login agar dapat mengikuti psikotes, sedangkan di bawah judul terdapat animasi agar layar utama ini lebih menarik, tabel bawah merupakan keterangan yang bersangkutan dengan aplikasi psikotes, seperti petunjuk pengisian pada halaman tes, pada tabel kanan bawah terdapat nama dan tahun pembuatan aplikasi psikotes.

Apabila para peserta tes psikotes ingin mengikuti tes, maka para peserta tes wajib untuk daftar terlebih dahulu. Pada halaman ini para peserta tes wajib mengisi semua form biodata yang telah tersedia. Berikut ini merupakan rancangan halaman daftar bagi para peserta tes.



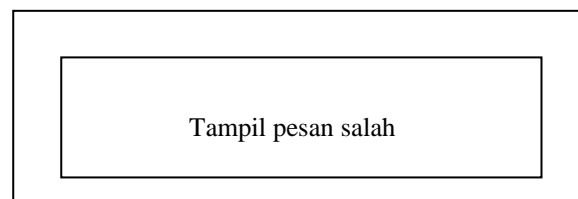
Gambar 3.14 Rancangan Halaman Register Bagi Para Peserta.

Setelah peserta registrasi peserta akan masuk kehalaman login, dimana pada halaman login peserta diminta untuk memasukan *username* dan *password* yang telah dibuat pada halaman register. Untuk lebih jelasnya perhatikan **Gambar 3.15**.



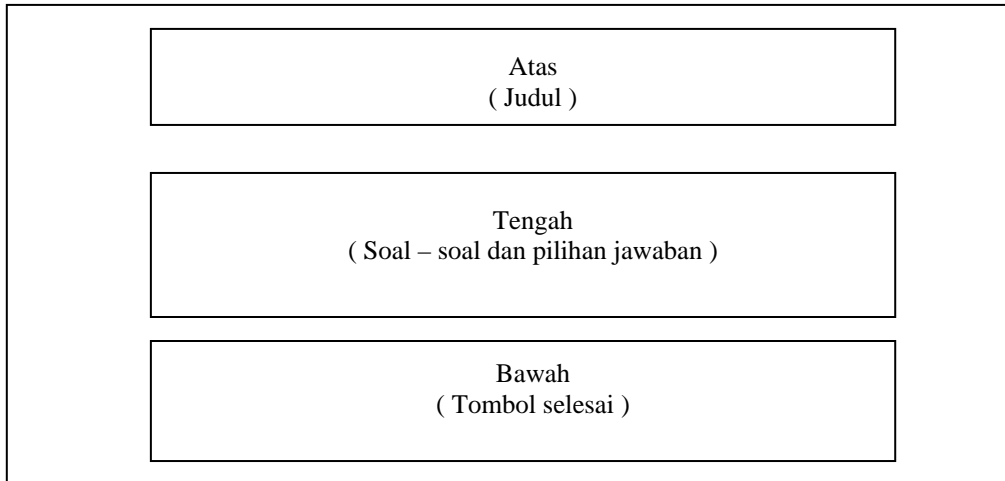
Gambar 3.15. Rancangan halaman *log in*

Tengah terdapat *text box* yang wajib diisi oleh peserta tes psikotes setelah mengisi *text box* peserta menekan tombol *login*, Jika peserta salah memasukan *username* dan *password* akan tampil pesan salah seperti **Gambar 3.24**



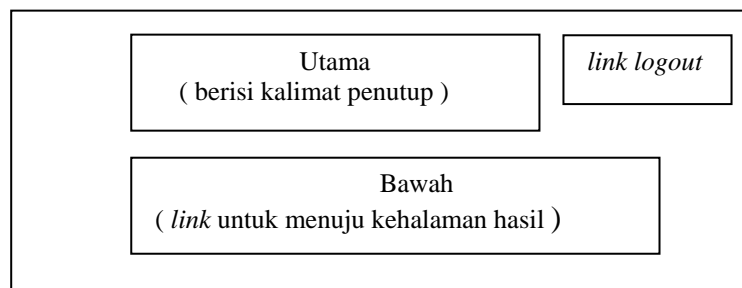
Gambar 3.16 Rancangan Halaman Salah.

Peserta akan masuk kahalaman soal jika peserta mengisi *username* dan *password* dengan benar. Pada halaman soal, peserta akan diberi soal sebanyak 30 soal, soal tersebut berupa pilihan ganda, dimana peserta diminta untuk menjawab dengan memilih jawaban yang benar. Setelah peserta selesai menjawab semua soal-soal yang diberikan, peserta mengklik tombol selesai yang terdapat pada halaman soal yang terakhir. Berikut ini merupakan rancangan halaman soal.



Gambar 3.17 Rancangan Halaman Soal.

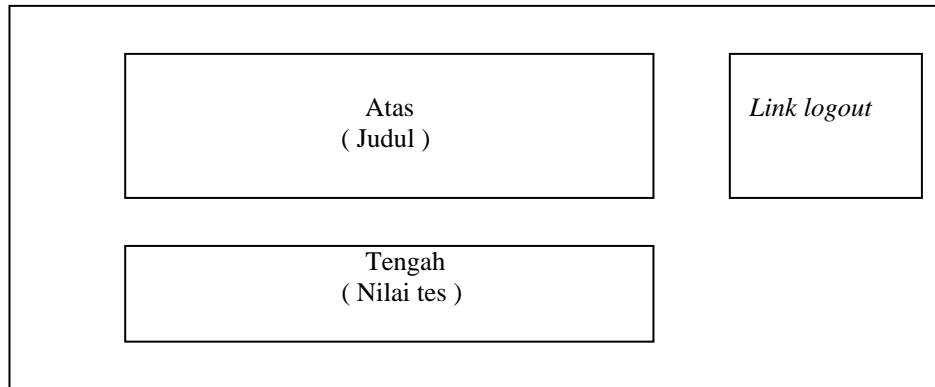
Setelah peserta mengklik tombol selesai peserta akan masuk kehalaman selesai, dimana halaman selesai berisi kalimat penutup dan terdapat *link* untuk menuju kehalaman hasil. Berikut ini merupakan rancangan halaman selesai.



Gambar 3.18 Rancangan Halaman Selesai.

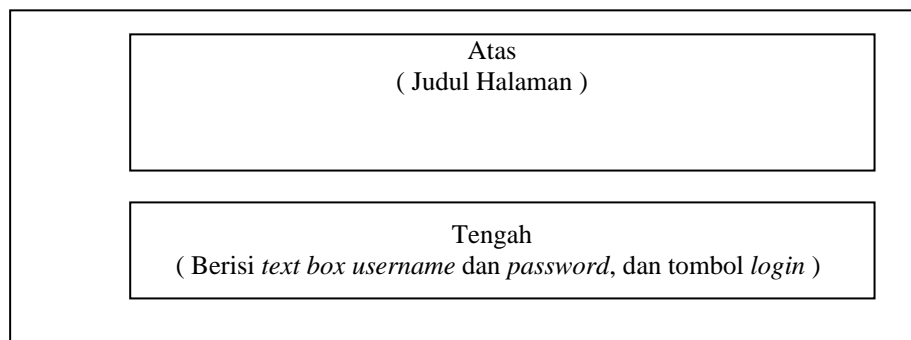
Pada halaman selesai ini terdapat *link logout* dan *link* hasil, *link logout* dipergunakan jika peserta ingin keluar menuju kehalaman utama dan tidak ingin melihat nilai yang diperolehnya. *Link* hasil dipergunakan peserta untuk melihat nilai yang diperolehnya setelah mengikuti tes psikotes.

Setelah peserta mengklik *link* hasil peserta akan masuk kehalaman nilai, halaman nilai ini berisi nama peserta, nilai peserta, dan peringkat tes (*grade*). Untuk lebih jelasnya perhatikan Gambar 3.27.



Gambar 3.19 Rancangan Halaman Hasil.

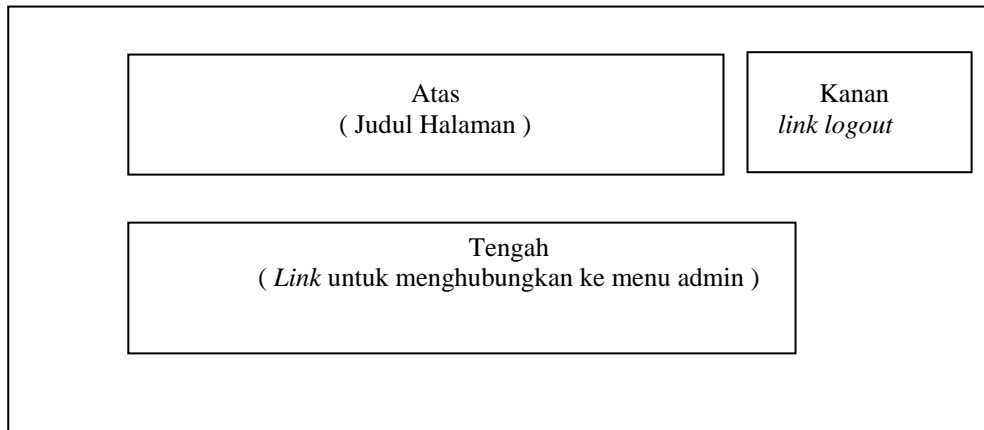
Pada aplikasi psikotes ini penggunanya tidak hanya untuk peserta, melainkan untuk admin juga. Selanjutnya rancangan untuk halaman admin, admin wajib memasukan alamat URL sebelum masuk kehalaman *login*. Perancangan halaman *login* admin dapat dilihat pada Gambar 3.28.



Gambar 3.20 Perancangan Halaman *Login* Admin.

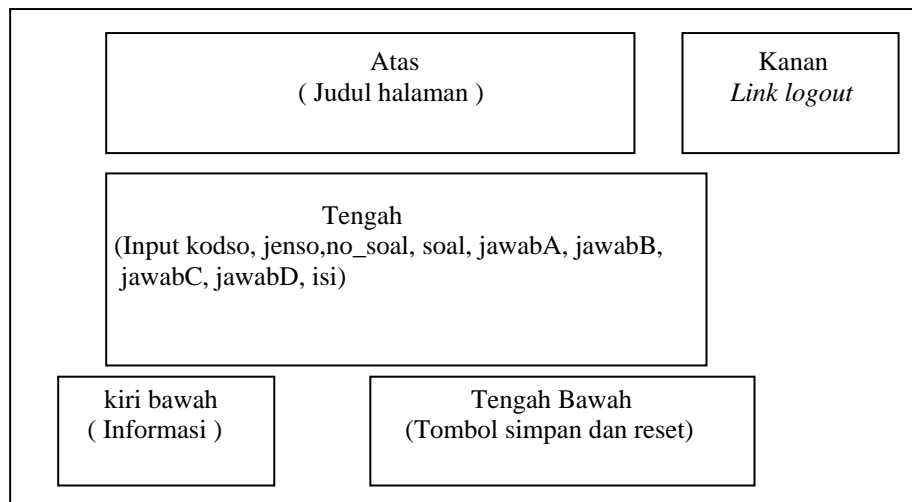
Pada rancangan ini admin wajib mengisi *text box* yang telah tersedia, jika admin salah memasukan *username* dan *password*, maka akan tampil pesan salah seperti yang terlihat pada **Gambar 3.20**. Admin akan masuk kehalaman admin jika

memasukan *username* dan *password* dengan benar. Atas pada rancangan halaman admin terdapat judul, kanan terdapat *link logout* yang berfungsi untuk keluar dari halaman admin, sedangkan ditengah terdapat *link* untuk menghubungkan ke menu admin. Untuk lebih jelasnya perhatikan **Gambar 3.21** berikut ini.



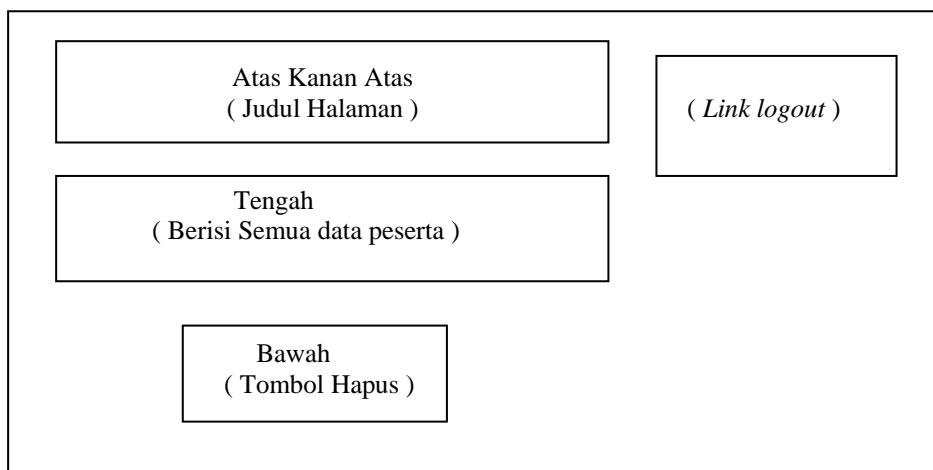
Gambar 3.21 Rancangan Halaman Utama

Perancangan halaman input soal digunakan oleh admin untuk memasukan soal dan juga merubah soal, perancangan ini terdiri dari beberapa sisi, yaitu: Atas digunakan sebagai judul halaman. Tengah digunakan sebagai input kodso, jenso, no_soal, soal, jawabA, jawabB, jawabC, jawabD, isi tombol simpan dan *reset*. Kanan atas digunakan sebagai *link logout* untuk keluar dari halaman admin. Kanan bawah digunakan sebagai *link* untuk kembali kehalaman admin. Apabila admin salah mengisi data soal akan tampil pesan salah seperti yang terlihat pada **Gambar 3.21**. admin diminta untuk kembali mengisi data soal dengan benar. Bawah berfungsi jika admin berhasil mengisi data soal dengan benar akan tampil informasi data berhasil disimpan. Untuk lebih jelasnya perhatikan **Gambar 3.22**.



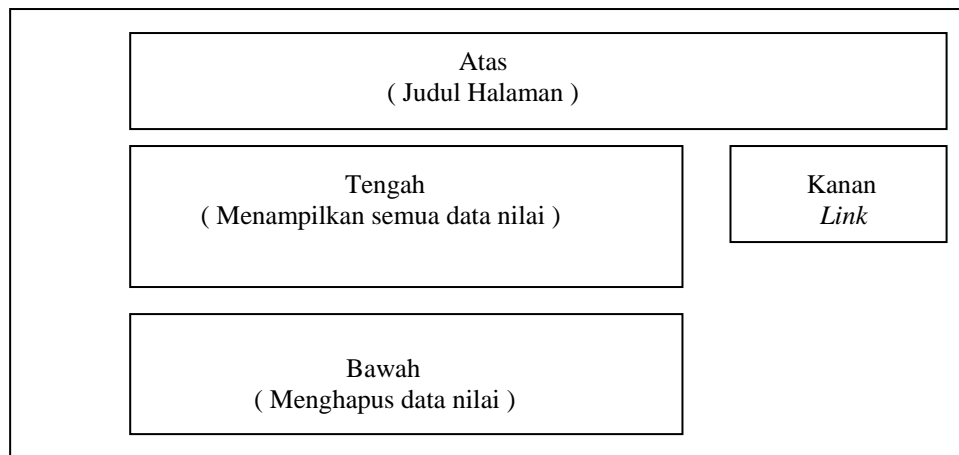
Gambar 3.22 Rancangan Halaman Input Soal.

Pada **Gambar 3.23** rancangan halaman data peserta terdiri dari beberapa sisi, yaitu : Atas digunakan sebagai judul halaman, Kanan Atas adalah *link logout* yang dipergunakan oleh admin untuk keluar dari halaman admin, Kanan Bawah adalah *link* untuk kembali kehalaman admin, Tengah berisi semua data peserta, Bawah berisi tombol hapus yang akan digunakan oleh admin jika admin ingin menghapus data peserta.



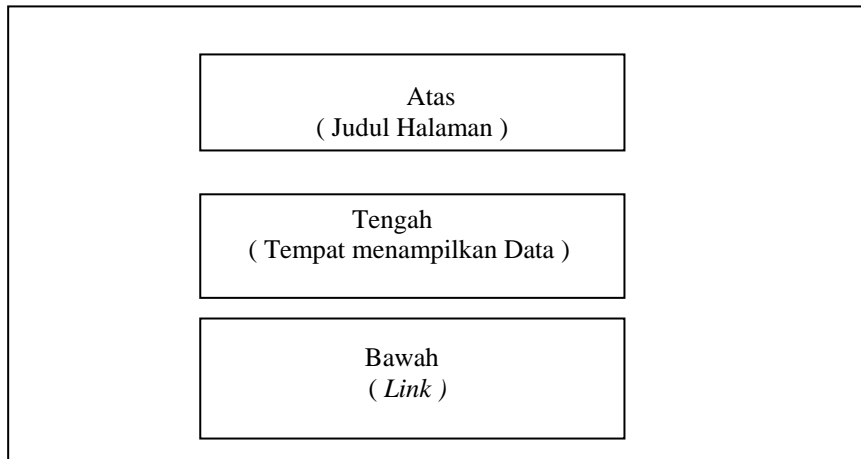
Gambar 3.23 Rancangan Data Peserta.

Perancangan data nilai terdiri dari beberapa sisi, yaitu : Atas yang berisi judul halaman, Kanan terdapat *link* untuk kembali kehalaman admin, Kiri terdapat *form* pencarian yang berguna untuk mencari data nilai peserta tes, Tengah akan menampilkan semua data nilai peserta tes, Bawah akan digunakan oleh admin untuk menghapus data nilai peserta tes. Pada rancangan ini admin tidak dapat *mengedit* data nilai peserta, admin hanya dapat melihat semua data nilai peserta saja. Untuk lebih jelasnya perhatikan **Gambar 3.24**.



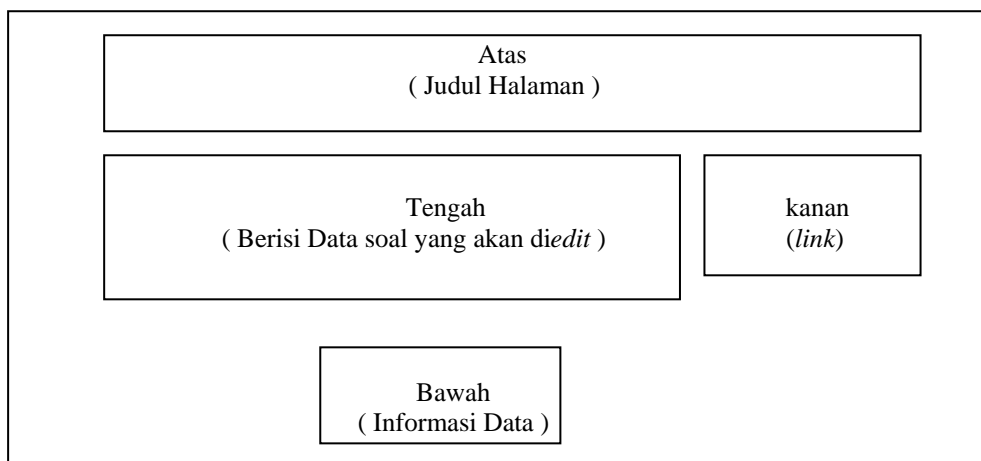
Gambar 3.24 Rancangan Halaman Data Nilai.

Gambar 3.25 merupakan rancangan halaman data soal yang terdiri dari, Atas merupakan judul halaman, Tengah berisi satu *record* data soal dan ditampilkan berdasarkan data soal yang akan dilihat oleh admin, Bawah berisi *link* untuk kembali kehalaman admin dan *link* untuk masuk kehalaman edit data soal.



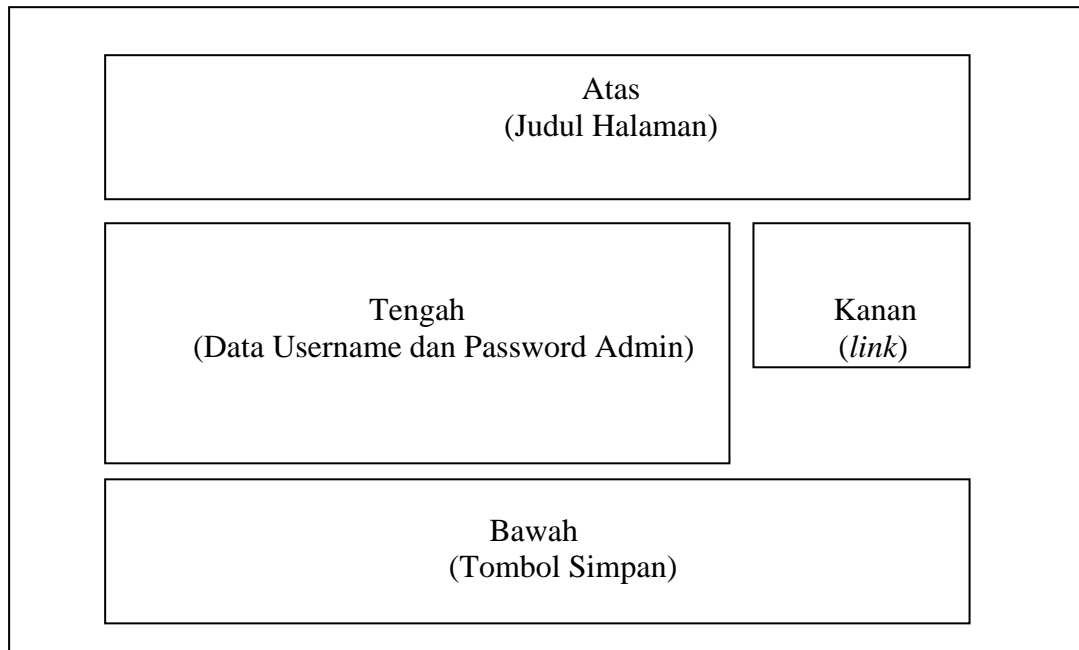
Gambar 3.25 Rancangan Halaman Data Soal.

Rancangan halaman edit data soal terdiri dari, Atas merupakan judul halaman, Kanan adalah *link* untuk kembali kehalaman admin dan *link* untuk kembali kehalaman data soal, Tengah berisi data soal yang akan *diedit*, apabila admin salah mengedit data soal akan tampil pesan salah seperti yang terlihat pada Gambar 3.24. admin diminta untuk kembali *mengedit* data soal dengan benar. Bawah berfungsi jika admin berhasil mengisi data soal dengan benar akan tampil informasi data berhasil *diedit*. Untuk lebih jelasnya perhatikan **Gambar 3.26**.



Gambar 3.26 Rancangan Halaman Edit Data Soal.

Gambar 3.27 merupakan rancangan halaman *ubah password admin*, yang terdiri dari, Atas merupakan judul halaman, Kanan merupakan *link* untuk kembali kehalaman admin, Tengah merupakan data *username* dan *password* admin yang akan dirubah, Bawah berisi tombol *simpan* yang akan digunakan oleh admin untuk menyimpan *username* dan *password* baru.



Gambar 3.27 Rancangan Halaman *Rubah Username Admin*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

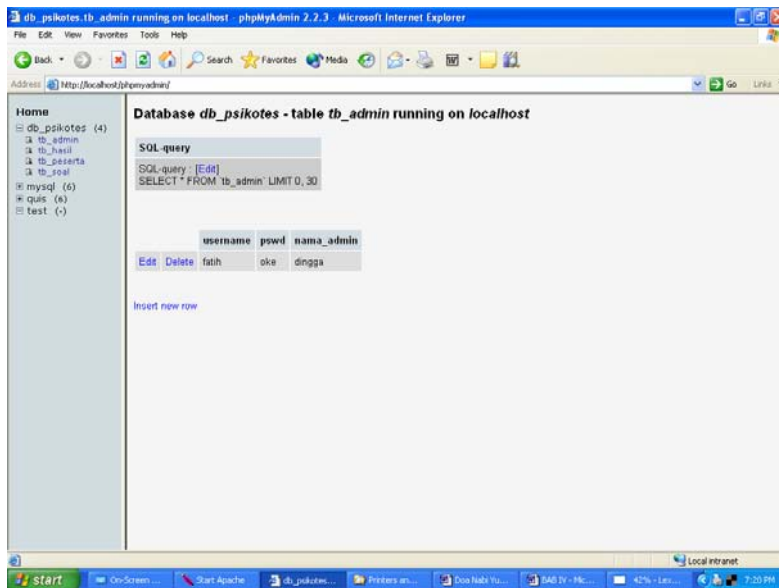
Setelah tahap analisis dan perancangan pada bab III, maka tahap selanjutnya yaitu tahap implementasi. Pada bab ini akan dibahas mengenai implementasi pembuatan basis data dan implementasi perancangan layar.

4.1.1 Implementasi Basis Data

Pada implementasi pembuatan basis data ini perangkat lunak yang digunakan yaitu phptriad. Pada implementasi ini hanya menampilkan basis data dari tabel-tabel yang fungsional saja, seperti: *tb_admin*, *tb_soal*, *tb_peserta*, dan *tb_hasil*. Masing-masing tabel tersebut telah dijelaskan pada tahap perancangan pada bab III.

4.1.1.1 Implementasi Basis Data *Tb_admin*

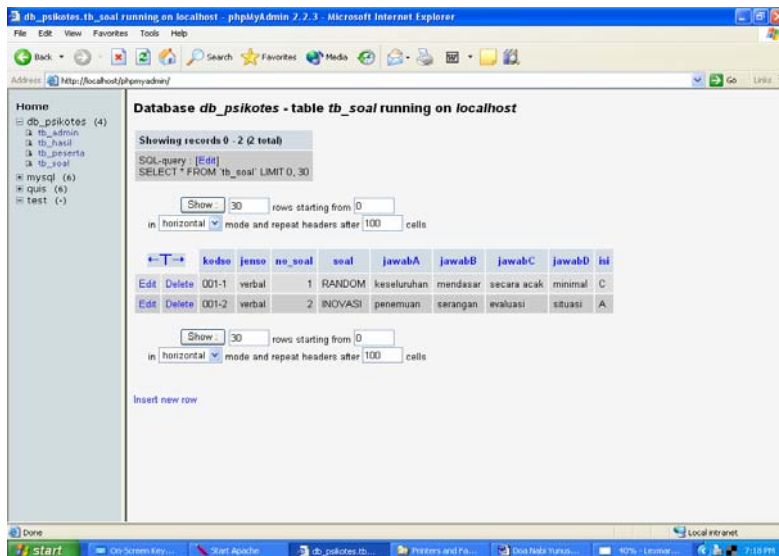
Tb_admin terdiri dari tiga kolom data, kolom pertama berisi data *username* dengan lebar maksimum 15 karakter, kolom kedua berisi data *password* dengan lebar maksimum 15 karakter, dan kolom ketiga berisi data nama admin dengan lebar maksimum 25 karakter. **Gambar 4.1** merupakan implementasi basis data *tb_admin*.



Gambar 4.1 Implementasi Pembuatan Basis Data Tabel Admin.

4.1.1.2 Implementasi Basis Data Tb_soal

Tb_soal terdiri dari sembilan kolom data, kolom pertama berisi data angka kode soal dengan lebar maksimum 6 digit, kolom kedua berisi data jenis soal dengan lebar maksimum 10 karakter, kolom ketiga berisi data angka nomor soal dengan lebar maksimum 4 digit, kolom keempat berisi data soal, kolom kelima berisi data jawaban A dengan lebar maksimum 60 karakter, kolom keenam berisi data jawaban B dengan lebar maksimum 60 karakter, kolom ketujuh berisi data jawaban C dengan lebar maksimum 60 karakter, kolom kedelapan berisi data jawaban D dengan lebar maksimum 60 karakter, dan kolom kesembilan berisi data isi jawaban dengan lebar maksimum 2 karakter. **Gambar 4.2** merupakan implementasi basis data tb_soal.

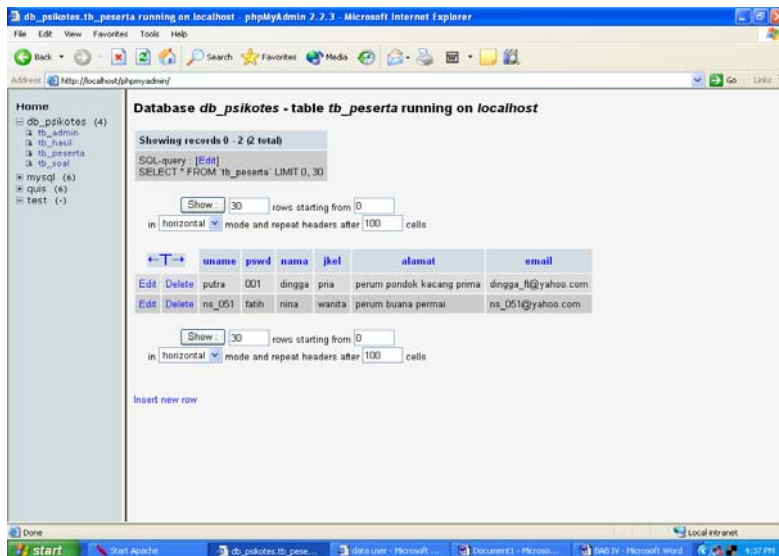


Gambar 4.2 Implementasi Pembuatan Basis Data Tabel Soal.

4.1.1.3 Implementasi Basis Data Tb_peserta

Tb_peserta terdiri dari enam kolom data, kolom pertama berisi data *username* dengan lebar maksimum 25 karakter, kolom kedua berisi data *password* dengan lebar maksimum 15 karakter, kolom ketiga berisi data nama dengan lebar maksimum 30 karakter, kolom keempat berisi data jenis kelamin dengan lebar maksimum 7 karakter, kolom kelima berisi data alamat dengan lebar maksimum 100 karakter, dan kolom keenam berisi data e-mail dengan lebar maksimum 50 karakter.

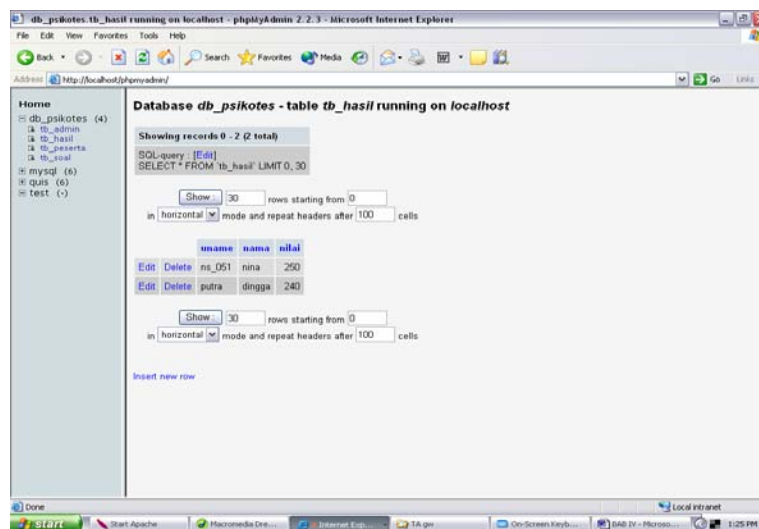
Gambar 4.3 merupakan implementasi basis data tb_peserta.



Gambar 4.3 Implementasi Pembuatan Basis Data Tb_peserta.

4.1.1.4 Implementasi Basis Data Tb_hasil

Tb_hasil terdiri dari tiga kolom data, kolom pertama berisi data *username* dengan lebar maksimum 25 karakter, kolom kedua berisi data nama dengan lebar maksimum 30 karakter, dan kolom ketiga berisi data angka nilai dengan lebar maksimum 4 digit. Gambar 4.4 merupakan implementasi basis data tb_hasil.



Gambar 4.4 Implementasi Pembuatan Basis Data Tb_hasil.

4.1.2 Implementasi Perancangan Layar

Pada implementasi perancangan layar akan menampilkan rancangan layar dan potongan program, seperti: menu register, menu login peserta, menu soal, menu hasil, menu login admin, dan menu admin. Masing-masing perancangan layar tersebut telah dijelaskan pada bab III.

4.1.2.1 Implementasi Perancangan Layar Register

Implementasi perancangan layar *register* akan menampilkan halaman register yang berisi form biodata untuk para peserta psikotes. **Gambar 4.5** merupakan implementasi perancangan layar *register*.

Kode 4.1. Proses register

```
1  <?
2  if (!$uname||!$pswd||!$nama||!$jkel||!$alamat)
3  {
4  echo("semua form harus diisi!!!<br>");
5  echo("<input type=button value=back onclick=\"self.history.back();\">");
6  }
7  else
8  {
9  if(ereg("^.+@.+\\..+$",$email))
10 {
11 $conn = mysql_connect("localhost","","") or die
    ("koneksi gagal :".mysql_error());
12 mysql_select_db("db_psikotes",$conn);
13 $sql = "select * from tb_peserta";
14 $qry=mysql_query($sql,$conn)or die
    ("Query salah, karena:".mysql_error());
15 $input="insert intotb_peserta      (uname,pswd,nama,jkel,alamat,email)
    VALUES ('$uname','$pswd','$nama','$jkel','$alamat','$email')";
16 $qry = mysql_query($input,$conn) ;
17 echo"data anda telah tersimpan<br>";
18 echo"<A HREF='login.php'> klik </A> untuk login";
19 }
20 else
21 {
22 echo"penulisan email tidak benar <br>";
23 echo("<input type=button value=back onclick=\"self.history.back();\">");
24 }
25 }
26 ?>
```

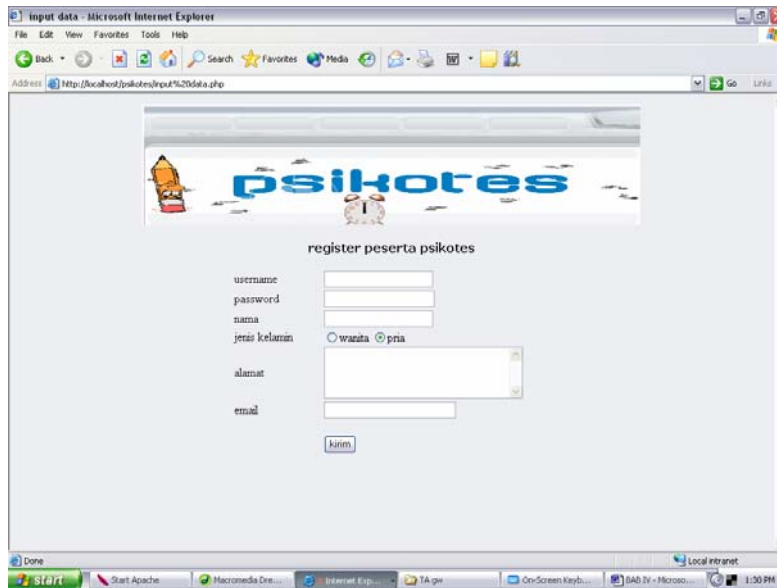
Kode program dari nomor **2** sampai nomor **6** akan menjelaskan : Sistem akan melakukan pengecekan terhadap data *uname*, *pswd*, *nama*, *jkel*, dan *alamat* telah terisi atau tidak. Jika tidak, maka sistem akan menampilkan pesan kesalahan dan sistem akan meminta anda untuk kembali mengisi form *register*.

Kode program nomor **9** akan menjelaskan : Jika data *uname*, *pswd*, *nama*, *jkel*, dan *alamat* telah terisi. Maka sistem akan melakukan pengecekan terhadap penulisan e-mail

Kode program dari nomor **11** sampai nomor **14** akan menjelaskan: Jika penulisan e-mail telah benar. Sistem akan melakukan koneksi ke *server* basis data *mysql*. Jika koneksi tidak berhasil, maka sistem akan menampilkan pesan kesalahan. Setelah koneksi berjalan baik, sistem akan mengaktifkan basis data, dalam aplikasi *psikotes* ini *db_psikotes* merupakan basis data yang digunakan dan *tb_peserta* merupakan tabel yang digunakan dalam halaman *register*.

Kode program dari nomor **15** sampai nomor **18** akan menjelaskan: Setelah basis data *db_psikotes* aktif, sistem akan menyimpan data yang dimasukan oleh peserta ke dalam tabel *tb_peserta*. Apabila data telah tersimpan, maka sistem akan menampilkan pesan bahwa data telah tersimpan, dan sistem akan meminta anda untuk *login*.

Kode program dari nomor **22** sampai nomor **23** akan menjelaskan: Jika penulisan e-mail salah, maka sistem akan menampilkan pesan kesalahan, dan sistem akan meminta anda untuk kembali mengisi *form register*. Kode 4.1. akan menggambarkan proses yang terjadi setelah peserta melakukan *register*.



Gambar 4.5 Implementasi Perancangan Layar Register.

4.1.2.2 Implementasi Perancangan Layar Login Peserta

Implementasi perancangan layar login peserta dibuat dengan tujuan untuk membatasi siapa saja yang berhak mengakses aplikasi psikotes ini, sistem akan memberi ijin kepada siapapun untuk mengakses aplikasi psikotes ini jika sudah register terlebih dahulu. **Gambar 4.6** merupakan implementasi perancangan layar login peserta.

Kode 4.2. Proses login peserta

```

1  <?
2  $data = "select * from tb_peserta where uname = '$uname' and
   pswd='$pswd'";
3  $conne = mysql_connect("localhost","","")or die ("koneksi tidak berhasil
   = ".mysql_error());
4  $db = mysql_select_db("db_psikotes",$conne);
5  $load = mysql_query($data);
6  $cek = mysql_fetch_array($load);
7  if ($cek[uname]==$uname and $cek[pswd]==$pswd)
8  {
9  echo"<font size=6>selamat saudara/i $uname, anda berhasil login.<br> ";
10 echo"<a href='soal.php'>soal</a>";
11 }
12 else
13 {
14 echo"<center>username anda tidak terdaftar<center><br>";
15 echo"silakan anda login kembali <a href='login.php'>login</a>";
16 }

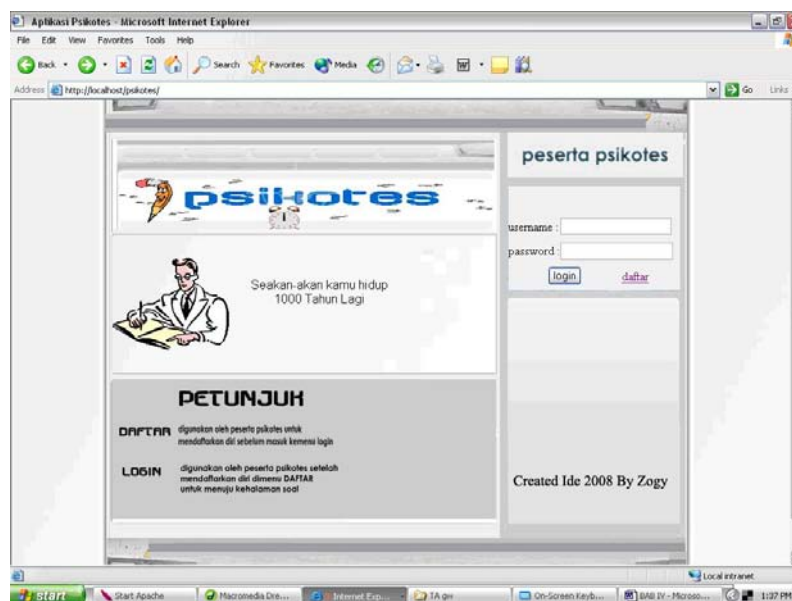
```

Kode program nomor 2 akan menjelaskan: Sistem akan menampilkan data *username* dan *password* yang terdapat pada tabel *tb_peserta*.

Kode program dari nomor 3 sampai nomor 5 akan menjelaskan: Sistem akan melakukan koneksi ke *server* basis data *mysql*. Jika koneksi tidak berhasil, maka sistem akan menampilkan pesan kesalahan. Setelah koneksi berjalan baik, sistem akan mengaktifkan basis data, dalam aplikasi psikotes ini *db_psikotes* merupakan basis data yang digunakan.

Kode program nomor 6 akan menjelaskan: Sistem akan mengambil data secara baris perbaris dari tabel *tb_peserta* dalam bentuk array.

Kode program dari nomor 7 sampai nomor 15 akan menjelaskan: Sistem akan melakukan pemeriksaan terhadap *username* dan *password* yang terdapat dalam basis data dengan *username* dan *password* yang dimasukan oleh peserta. Jika benar, maka sistem akan meminta anda untuk masuk ke halaman soal, jika salah sistem akan menampilkan pesan kesalahan dan anda diminta untuk kembali ke halaman *login*. Kode program 4.2. akan menggambarkan proses yang terjadi setelah peserta melakukan login.



Gambar 4.6 Implementasi Perancangan Layar Login Peserta.

4.1.2.3 Implementasi Perancangan Layar Soal

Implementasi perancangan layar soal akan menampilkan halaman soal yang berisi 30 soal, peserta menjawab soal pskotes dengan memilih jawaban yang benar.

Gambar 4.7 merupakan implementasi perancangan layar soal.

Kode 4.3. Proses soal

```
1  <?
2  $query= mysql_query ("select * from tb_soal");
3  while($stampil=mysql_fetch_array($query))
4  {
5  $no=$stampil["no_soal"];
6  $soal=$stampil["soal"];
7  $a=$stampil["jawabA"];
8  $b=$stampil["jawabB"];
9  $c=$stampil["jawabC"];
10 $d=$stampil["jawabD"];
11 ?>
```

Kode program nomor **2** akan menjelaskan: sistem akan menampilkan data yang terdapat pada tabel tb_soal.

Kode program dari nomor **3** sampai nomor 10 akan menjelaskan: sistem akan mengambil data dari basis data dengan tabel tb_soal, seperti no_soal, soal, jawabA, jawabB, jawabC dan jawabD secara baris perbaris dalam bentuk array untuk dideklarasikan kedalam variabel yang baru. Kode program 4.3. merupakan gambaran yang terjadi pada saat peserta mengerjakan soal psikotes.



Gambar 4.7 Implementasi Perancangan Layar Soal.

4.1.2.4 Implementasi Perancangan Layar Hasil

Implementasi perancangan layar hasil digunakan oleh peserta untuk melihat hasil yang diperoleh setelah mengikuti psikotes. **Gambar 4.8** merupakan Implementasi perancangan layar hasil.

Kode 4.5. Proses hasil psikotes

```

1  <?
2  $que="select * from tb_hasil where uname='$uname'";
3  $has=mysql_query($que);
4  while($roww=mysql_fetch_array($has)){
5  if($roww['nilai'] >= 270 && $roww['nilai'] <= 300){
6  $jmlpoll=jmlpoll+$roww['nilai'];
7  }else if ($roww['nilai'] <= 240 && $roww['nilai'] <= 270 ){
8  $jmlpoll=26+$roww['nilai'];
9  }else if ($roww['nilai'] <= 210 && $roww['nilai'] <= 240 ){
10 $jmlpoll=60+$roww['nilai'];
11 }else if ($roww['nilai'] <= 180 && $roww['nilai'] <= 210 ){
12 $jmlpoll=120+$roww['nilai'];
13 }else if ($roww['nilai'] <= 150 && $roww['nilai'] <= 180 ){
14 $jmlpoll=180+$roww['nilai'];
15 }else if ($roww['nilai'] <= 120 && $roww['nilai'] <= 150 ){
16 $jmlpoll=250+$roww['nilai'];
17 }else if($roww['nilai'] <= 120 ){
18 $jmlpoll=300+$roww['nilai'];

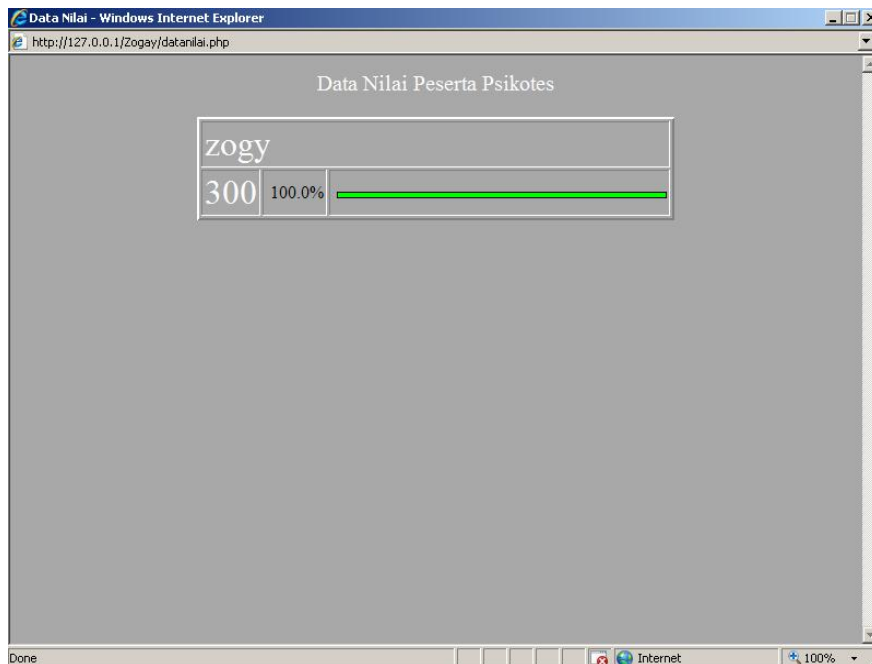
```

```

19 }}
20 $jmlpol=$jmlpoll;
21 $quer="select * from tb_hasil where uname='$uname'";
22 $hasi=mysql_query($quer);
23 $i=1;
24 while($ro=mysql_fetch_array($hasi)){
25 $h_jwb_a=sprintf("%01.1f",(($ro['nilai']/$jmlpol)*100))."%";
26 $g_jwb_a=$h_jwb_a*3;
27 $sen="&nbsp;".$h_jwb_a;
28 $gbr="&nbsp;<img src=\"images/blank.png\"
  class=\"polls_color_\".$i++.\"\" height=\"4\" width=\"$g_jwb_a\"
  alt=\"\"/>";
29 ?>

```

Kode program nomor 1 sampai dengan 4 akan menjelaskan: sistem pemanggilan data base. Kode program dari nomor 5 sampai nomor 19 akan menjelaskan: sistem yang membagi golongan nilai dalam bentuk persen (%). Kode program nomor 21 sampai dengan 24 menjelaskan pemanggilan data base. Kode program nomor 25 menjelaskan tentang sistem yang menghasilkan nilai dalam bentuk persen (%). Kode program nomor 26 menjelaskan panjang grafik nilai. Kode program nomor 28 menjelaskan sistem *image* dari grafik. **Kode 4.5.** merupakan gambaran yang terjadi pada saat peserta ingin menampilkan halaman hasil.



Gambar 4.8 Implemenasi Perancangan Layar Hasil.

4.1.2.5 Implementasi Perancangan Layar Login Admin

Implementasi perancangan layar login admin digunakan oleh admin. Untuk dapat masuk ke halaman admin, admin diwajibkan untuk melakukan login terlebih dahulu. **Gambar 4.9** merupakan Implementasi perancangan layar login admin.

Kode 4.6. Proses login admin

```
1 <?
2 if ($row[username]==$username and $row[pswd]==$pswd )
3 {
4 session_start();
5 session_register("username");
6 session_register("pswd");
7 $username=$row[username];
8 $pswd=$row[pswd];
9 if ($row[username]==$username)
10 {header("Location:admin.php");}
11 }
12 else
13 {
14 echo"anda gagal login";
15 echo"<a href='logad.php'>login</a>";
16 }
17 ?>
```

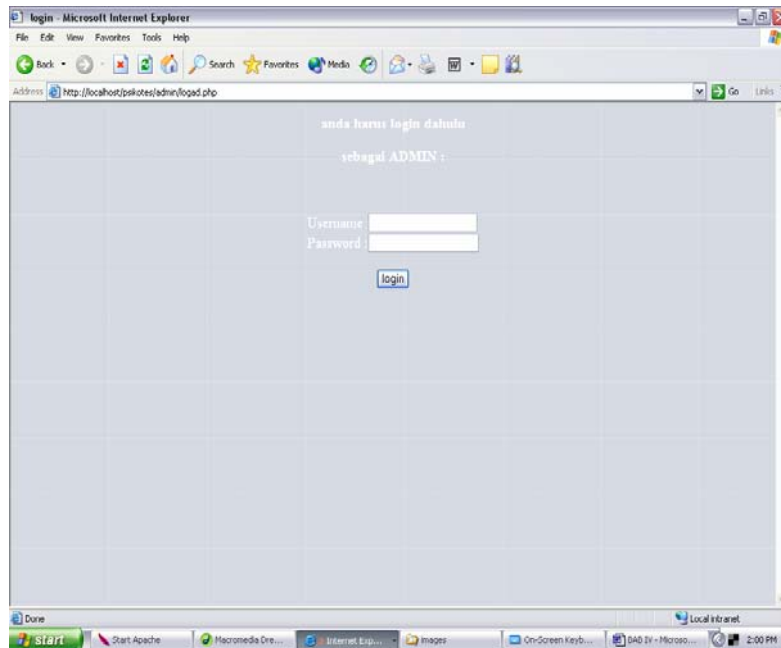
Kode program nomor **2** akan menjelaskan: Sistem akan melakukan pemeriksaan terhadap *username* dan *password* yang dimasukan oleh admin apakah sesuai dengan yang ada pada basis data atau tidak.

Kode program nomor **4** akan menjelaskan: Sistem akan menciptakan sesi. Sedangkan kode program nomor **5** sampai nomor **6** akan menjelaskan: Sistem akan mendaftarkan nama variabel *username* dan *password* ke dalam sesi.

Kode program nomor **9** sampai nomor **10** akan menjelaskan: sistem akan melakukan pemeriksaan terhadap *username* apakah sesuai atau tidak dengan yang terdapat di dalam basis data. Jika *username* sesuai, maka sistem akan mengirim ke alamat admin.php

Kode program nomor **12** sampai nomor **16** akan menjelaskan: jika *username* dan *password* admin salah, maka sistem akan menampilkan pesan kesalahan dan

sistem akan meminta anda untuk kembali ke halaman login. Kode program 4.6. merupakan potongan program perancangan layar admin



Gambar 4.9 Implementasi Perancangan Layar Login Admin.

4.1.2.6 Implementasi Perancangan Layar Admin

Implementasi perancangan layar admin digunakan oleh admin, admin akan masuk ke halaman admin setelah admin berhasil login. **Gambar 4.10** merupakan Implementasi perancangan layar admin.

Kode 4.7. Proses halaman admin

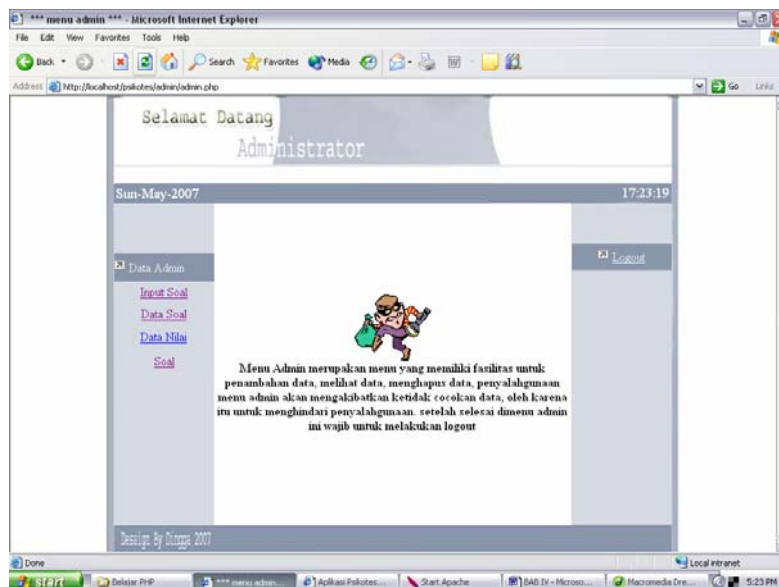
```
1 <?
2 session_start();
3 include('koneksi.inc');
4 if(!empty($username) and
!empty($pswd))
5 {
6 ?>
```

Kode program nomor 2 akan menjelaskan: digunakan oleh sistem untuk memulai sesi.

Kode program nomor 3 merupakan file untuk melakukan koneksi ke *server* basis data *mysql*.

kode program nomor 4 akan menjelaskan: sistem akan melakukan pemeriksaan terhadap *username* dan *password* admin.

Pada halaman ini terdapat *link* yang digunakan admin untuk menampilkan data soal, memasukan soal, menampilkan data nilai, mengedit data soal, dan menghapus data soal. Kode program 4.7. merupakan potongan program implementasi perancangan layar admin.



Gambar 4.10 Implementasi Perancangan Layar Admin.

4.2 Pengujian

Setelah tahap implementasi, tahap selanjutnya adalah tahap pengujian. Pada tahap ini akan dibahas mengenai lingkungan pengujian, skenario pengujian, dokumen pengujian dan hasil pengujian, dan analisis pengujian.

4.2.1 Lingkungan Pengujian

Lingkungan pengujian pada aplikasi ini meliputi perangkat keras dan perangkat lunak, spesifikasi masing-masing lingkungan tersebut adalah sebagai berikut:

1. Spesifikasi Perangkat Lunak, meliputi:

- a. Sistem operasi yang digunakan pada aplikasi psikotes ini adalah Microsoft Windows XP.
- b. Macromedia Dreamweaver 8 digunakan untuk membuat program aplikasi psikotes ini.
- c. Adobe Photoshop 7.0, digunakan untuk mendesain tampilan pada aplikasi psikotes ini.
- d. Swish digunakan untuk membuat animasi pada aplikasi psikotes ini.
- e. Phptriad digunakan untuk menjalankan aplikasi psikotes ini
- f. PhpMyAdmin digunakan untuk membuat basis data.
- g. Apache digunakan sebagai localhost.
- h. Microsoft Internet Explorer 4.0 digunakan sebagai *browser* internet.

2. Spesifikasi Perangkat Keras, meliputi

- a. Komputer dengan prosesor AMD Sempron 2.2 GHz.
- b. Memori 256 MB.
- c. VGA 64 MB.

- d. Monitor 15'
- e. Keyboard dan Mouse

4.2.2 Skenario Pengujian

Skenario pengujian meliputi pengujian perangkat lunak terhadap fungsionalitas aplikasi yang dibangun berfungsi untuk melakukan pengujian proses yang terjadi pada perangkat lunak yang dibangun dengan proses yang terjadi didalam sistem dan menguji kelayakan aplikasi dalam sebuah situs yang akan digunakan.

Metode yang digunakan dalam aplikasi psikotes ini adalah *black-box testing*. Metode ini sesuai dengan aplikasi yang akan dibangun, karena metode *black-box testing* melakukan pengujian dengan cara memberikan sejumlah masukan pada program aplikasi yang akan diproses sesuai dengan kebutuhan fungsionalnya untuk menghasilkan keluaran yang diinginkan.

Apabila keluaran proses tidak menghasilkan sesuai yang diinginkan, maka program aplikasi masih terdapat kesalahan-kesalahan yang harus diperbaiki pada program aplikasi tersebut. Apabila keluaran proses menghasilkan kebutuhan fungsionalnya, maka program aplikasi tersebut benar.

Web browser Internet Explorer digunakan untuk melakukan pengujian pada program aplikasi psikotes, dengan cara mengetikkan alamat URL sebagai berikut: <http://localhost/psikotes/index.php>. *Black-box testing* melakukan pengujian pada program aplikasi psikotes dimulai dari menu register sampai semua menu yang terdapat pada program aplikasi psikotes.

Tabel 4.1 Skenario Pengujian Perangkat Lunak

No	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil Yang Diharapkan
1	Menu register	Data salah	Tidak memasukan semua biodata peserta	Menampilkan pesan kesalahan
2	Halaman login peserta	Normal	Memasukan <i>username</i> dan <i>password</i> peserta	Menampilkan halaman proses soal
3	Halaman soal	Normal	Memilih jawaban	Menampilkan halaman

				selesai
4	Halaman hasil	Normal	Memilih <i>link</i> hasil	Menampilkan halaman hasil
5	Login admin	Data salah	Tidak Memasukan <i>username</i> dan <i>password</i>	Menampilkan pesan kesalahan
6	Halaman input soal	Data salah	Tidak memasukan soal	Menampilkan pesan kesalahan
7	Halaman data soal	Normal	Memilih <i>link</i> data soal	Menampilkan halaman data soal
8	Halaman hasil	Normal	Memilih <i>link</i> hasil	Menampilkan halaman hasil
9	Halaman data peserta	Normal	Memilih <i>link</i> data peserta	Menampilkan halaman data peserta
10	Halaman logout	Normal	Memilih <i>link</i> <i>logout</i>	Keluar dari halaman admin

4.2.3 Dokumen Hasil Pengujian

Dokumen hasil pengujian merupakan dokumen yang berisi berdasarkan nomor urut pada skenario pengujian perangkat lunak. **Tabel 4.2** adalah tabel dokumen hasil pengujian.

Tabel 4.2 Dokumen Hasil Pengujian Perangkat Lunak

No	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil Yang Diharapkan	Hasil Uji
1	Halaman register	Data salah	Tidak memasukan semua biodata peserta	Menampilkan pesan kesalahan	Diterima
2	Halaman login peserta	Normal	Memasukan <i>username=d&n</i> dan <i>password=fatih</i>	Menampilkan halaman proses soal	Diterima

Tabel 4.2 (Lanjutan) Dokumen Hasil Pengujian Perangkat Lunak

3	Halaman soal	Normal	Memilih jawaban A, B, C dan D	Menampilkan halaman selesai	Diterima
4	Halaman hasil	Normal	Memilih <i>link</i> hasil	Menampilkan halaman hasil	Diterima
5	Halaman login admin	Data salah	Tidak Memasukan <i>username</i> dan <i>password</i>	Menampilkan pesan kesalahan	Diterima
6	Halaman input soal	Data salah	Tidak memasukan soal	Menampilkan pesan kesalahan	Diterima
7	Halaman data soal	Normal	Memilih <i>link</i> data soal	Menampilkan halaman data soal	Diterima

8	Halaman data hasil	Normal	Memilih <i>link</i> hasil	Menampilkan halaman hasil	Diterima
9	Halaman data peserta	Normal	Memilih <i>link</i> data peserta	Menampilkan halaman data peserta	Diterima
10	Halaman logout	Normal	Memilih <i>link</i> <i>logout</i> 1	Keluar dari halaman admin	Diterima

4.2.4 Analisis Hasil Pengujian

Setelah pengujian perangkat lunak telah dilakukan, selanjutnya melakukan analisis dari hasil pengujian perangkat lunak.

Gambar 4.11 merupakan gambar yang menunjukkan hasil pengujian halaman register.



Gambar 4.11 Hasil Pengujian Halaman Register.

Pada halaman register, pengujian dilakukan dengan kelompok uji salah yang berarti kelompok uji memasukkan data salah. Pada **Gambar 4.11** kelompok uji tidak memasukkan semua biodata peserta, sistem akan menampilkan pesan kesalahan, dan anda diminta oleh sistem untuk kembali ke halaman register. Pengujian pada halaman register ini menghasilkan rancangan yang sesuai diinginkan.

Gambar 4.12 merupakan gambar yang menunjukkan proses yang terjadi setelah peserta melakukan *login*.



Gambar 4.12 Hasil Pengujian Halaman Login Peserta.

Pada halaman login peserta, pengujian dilakukan dengan kelompok uji normal yang berarti kelompok uji memasukkan data benar. Kelompok uji melakukan login dengan *username*=d&n dan *password*=fatih, sistem akan memproses *username* dan *password* tersebut. Setelah *username* dan *password* diproses, sistem akan menampilkan halaman proses soal, seperti yang terlihat pada Gambar 4.12.

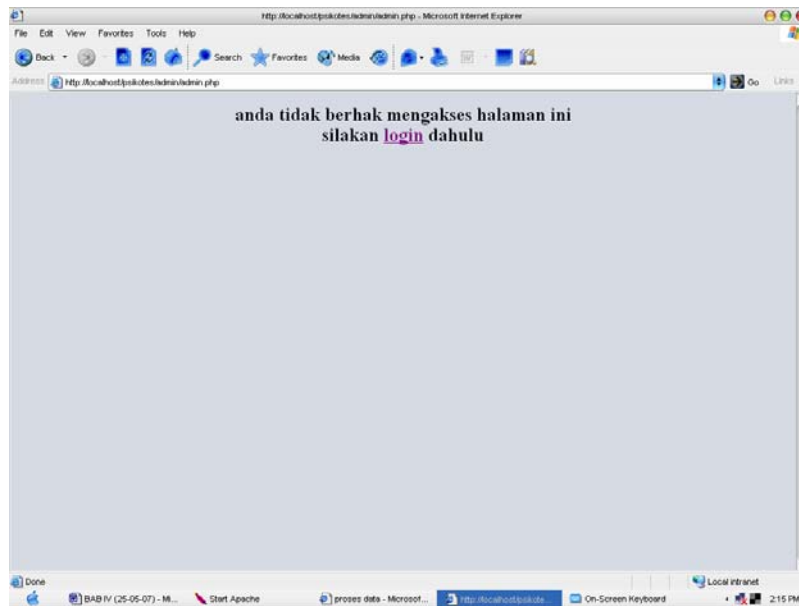
Gambar 4.13 merupakan gambar yang menunjukkan hasil pengujian pada halaman soal.



Gambar 4.13 Hasil Pengujian Halaman Soal.

Pada halaman soal, pengujian dilakukan dengan cara memilih jawaban yang dianggap benar dengan kelompok uji benar. Pada halaman soal peserta akan diberi waktu oleh sistem untuk mengerjakan soal psikotes. Apabila peserta tidak dapat mengerjakan sesuai dengan waktu yang telah ditentukan, maka secara otomatis halaman soal akan tertutup. Setelah peserta selesai mengerjakan tes psikotes, sistem akan menampilkan halaman selesai. Pada halaman selesai terdapat *link* hasil yang berarti peserta memilih *link* hasil untuk mengetahui hasil psikotes yang diperoleh.

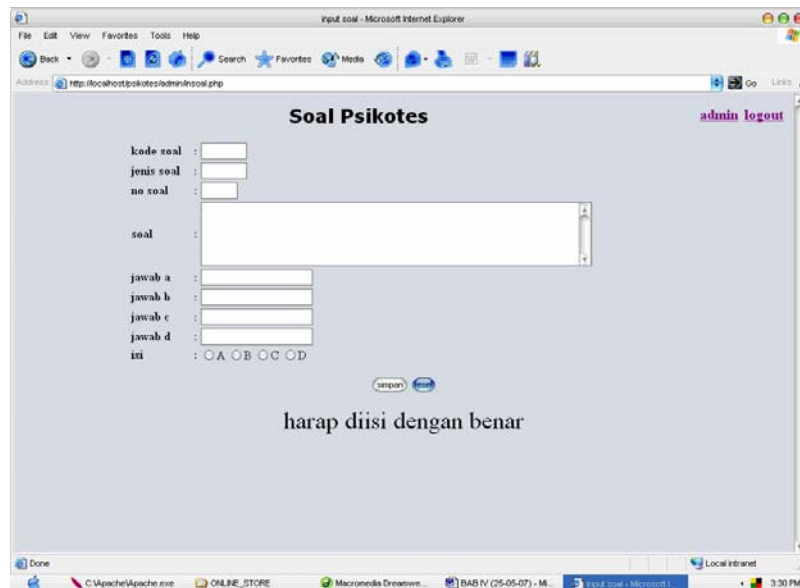
Gambar 4.14 merupakan gambar yang menunjukkan hasil pengujian pada halaman login admin.



Gambar 4.14 Hasil Pengujian Halaman Login Admin.

Pada halaman login admin, pengujian dilakukan dengan kelompok uji data salah yang berarti kelompok uji memasukan data salah. Pada Gambar 4.15 kelompok uji tidak memasukan *username* dan *password* admin, sistem akan menampilkan pesan kesalahan, dan anda diminta oleh sistem untuk login terlebih dahulu. Sebelum dapat masuk kehalaman admin, Admin wajib untuk melakukan *login*. Rancangan ini berguna untuk membatasi siapa saja yang dapat mengakses halaman admin. Pada halaman admin terdapat *link* yang berguna untuk merubah atau menghapus data yang ada didalam basis data, diantaranya: *link* input soal, *link* data soal, *link* data hasil, *link* data peserta, dan *link* logout.

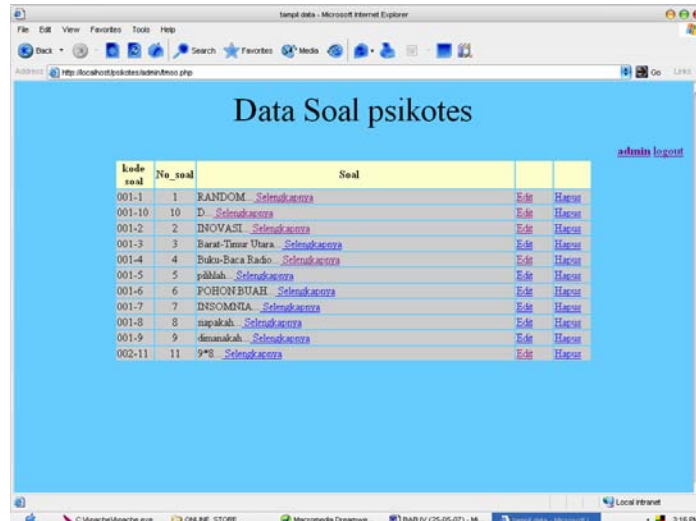
Gambar 4.15 merupakan gambar yang menunjukkan hasil pengujian pada halaman input soal.



Gambar 4.15 Hasil Pengujian Halaman Input Soal.

Pada halaman input soal, pengujian dilakukan dengan kelompok uji data salah yang berarti kelompok uji memasukkan data salah. Pada **Gambar 4.16** kelompok uji tidak memasukkan data soal, sistem akan menampilkan pesan kesalahan. Pada halaman ini terdapat dua buah tombol. Tombol pertama berguna untuk menyimpan soal ke dalam basis data jika data soal yang dimasukkan telah benar, tombol kedua berguna untuk mengembalikan nilai ke kondisi awal.

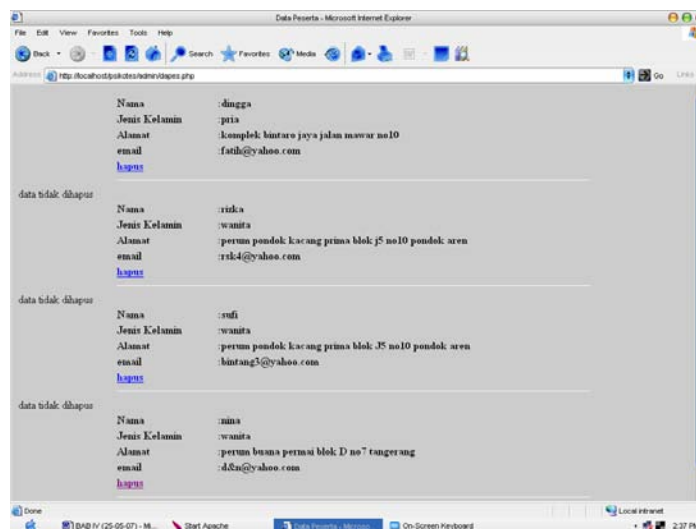
Gambar 4.16 merupakan gambar yang menunjukkan hasil pengujian pada halaman data soal.



Gambar 4.16 Hasil Pengujian Halaman Data Soal.

Pada halaman data soal, pengujian dilakukan dengan menampilkan data soal secara keseluruhan dengan kelompok uji benar. Sistem akan menampilkan soal yang terdapat didalam basis data berdasarkan kode soal. Soal yang terlihat pada Gambar 4.16 ditampilkan secara terpotong, untuk melihat soal secara utuh kelompok uji memilih *link* selengkapnya. Pada halaman ini juga terdapat *link* untuk menghapus soal dan merubah soal.

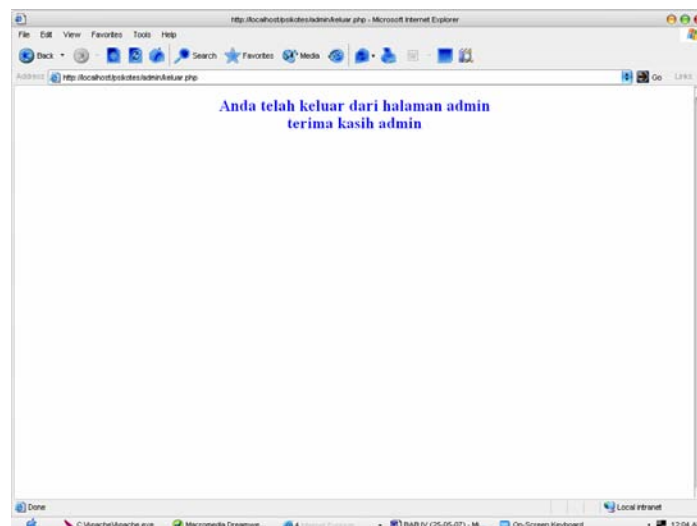
Gambar 4.17 merupakan gambar yang menunjukkan hasil pengujian pada halaman data peserta.



Gambar 4.17 Hasil Pengujian Halaman Data peserta.

Pada halaman data peserta, pengujian dilakukan dengan kelompok uji benar yang berarti dijalankan dengan semestinya. Pada halaman ini digunakan oleh admin untuk melihat peserta yang terdaftar dan juga digunakan oleh admin untuk menghapus data setiap peserta. Untuk menghapus data peserta, admin memilih *link* hapus, setelah admin memilih *link* hapus sistem akan memproses dengan menghapus data peserta yang terdapat didalam basis data berdasarkan nama peserta.

Gambar 4.18 merupakan gambar yang menunjukkan hasil pengujian pada halaman logout



Gambar 4.18 Hasil Pengujian Proses Logout.

Untuk menghindari penyalahgunaan. setelah selesai pada halaman admin ini wajib untuk melakukan logout

Pengujian yang dilakukan dari halaman register hingga halaman admin menghasilkan proses keluaran sesuai dengan yang dirancang. Secara fungsionalitas program, program aplikasi psikotes ini tidak terdapat kesalahan dan hasil program sesuai dengan hasil yang telah diproses didalam sistem.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Ada beberapa hal yang dapat disimpulkan dari uraian pada bab I sampai dengan bab IV. Kesimpulan yang dapat diambil antara lain:

1. Aplikasi psikotes yang dibangun telah memenuhi tujuan keberhasilan, yang dalam hal ini psikotes menjadi lebih mudah diakses dengan teknologi internet.
2. Sistem dapat menekan kebutuhan ruang penyimpanan dengan menjadikan prioritas yang merupakan hasil relasi antara tabel peserta dan tabel soal hanya bersifat temporer file. Tetapi akibatnya data test yang pernah dilakukan peserta tidak dapat ditelusuri.

5.2 Saran

Ada beberapa saran yang dapat penulis sampaikan pada aplikasi psikotes ini, yaitu:

1. Agar informasi yang ditampilkan lebih menarik, aplikasi ini dapat ditambahkan fasilitas-fasilitas online seperti: buku tamu, dan pooling.
2. Jangan terlalu banyak menggunakan gambar dengan ukuran file yang sangat besar, karena dapat memperlambat proses pemanggilan web tersebut.
3. Selalu gunakan *Cascading Style Sheet*(CSS) untuk pengaturan format halaman Web, dan *minimalisasi filesize*.
4. Aplikasi ini merupakan langkah awal dalam rangka mengembangkan teknologi komunikasi internet, penulis mengharapkan aplikasi ini dapat dikembangkan agar nantinya menjadi lebih sempurna.

DAFTAR PUSAKA

1. Fathansyah, Ir, Basis Data, Informatika Bandung, 2001
2. Lani Sidharta, PHP dan *MySQL*, 1997
3. Sampurna, *Membuat Home Page dengan HTML*, Elex Media Komputindo, 1998.
4. Sutarman, SKom, *Membangun Aplikasi Web dengan PHP dan MySQL*, Graha Ilmu, 2003.
5. <http://www.master.web.id> ; tentang Master Web tanggal 2004
6. http://id.wikipedia.org/wiki/Kategori:Situs_web ; dasar – dasar PSIKOTES
7. <http://ikc.cbn.net.id/umum/kemas-cms.php> tentang PHP 2003
8. <http://ikc.cbn.net.id/umum/yanti-uml.php>; tentang UML juli 2003
9. <http://www.php.net/usage.php> ; tentang PHP apache april 2007
10. <http://www.ditplb.or.id/profile.php?id=47> teknik mengenal PSIKOTES tanggal 3 febuari 2009
11. <http://www.ssantsons.com/Artikel%20Psikotes.html>
tentang pengertian PSIKOTES atau test PSIKOLOGI tanggal 3 febuari 2009
12. <http://ggkarir.com/karir.php?karir=contoh-psikotes-1>
tentang Latihan Psikotes - **Analogi Verbal (Korelasi Ma** tanggal) 3 febuari 2009

LAMPIRAN

index.html

```
<HTML>
<HEAD>
<TITLE>Aplikasi Psikotes</TITLE>
<script language=javascript>
<!-- Beginning of JavaScript -
//untuk menampilkan waktu saat ini
function clockon(){ thistime= new Date()
var hours=thistime.getHours()
var minutes=thistime.getMinutes()
var seconds=thistime.getSeconds()
if(eval(hours)<10){hours="0"+hours}if(eval(minutes)<10)
{minutes="0"+minutes}
if(seconds<10) {seconds="0"+seconds}
thistime=hours+":"+minutes+":"+seconds
if(document.all) {
bgclocknoshade.innerHTML=thistime
}
if(document.layers) {
document.bgclockshade.document.write('<div
id="bgclocknoshade"
align="center" style="font-size:24px;color:#A8A8A8">'+thistime+'</div>')
document.close()
}
var timer=setTimeout("clockon()",200)
}

// - End of JavaScript - -->
</script>
<STYLE>
<style type="text/css">
<!--
a:link {
color: #993300;
text-decoration: none;
}
a:hover {
color: #00FFCC;
text-decoration: none;
}
a:visited {
text-decoration: none;
}
a:active {
text-decoration: none;
}
}
body {
background-image:url(images/bgindex.jpg);
}
-->
</style>
</HEAD>
```



```

<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0 MARGINHEIGHT=0
onLoad="clockon()">
<!-- ImageReady Slices (main.psd) -->
<TABLE WIDTH=624 BORDER=0 CELLPADDING=0 CELLSPACING=0 align="center">
<TR>
<TD ROWSPAN=16>
<IMG SRC="images/utama_01.jpg" WIDTH=9 HEIGHT=546 ALT=""></TD>
<TD COLSPAN=14>
<IMG SRC="images/utama_02.jpg" WIDTH=767 HEIGHT=42 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=42 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=7>
<IMG SRC="images/utama_03.jpg" WIDTH=508 HEIGHT=8 ALT=""></TD>
<TD ROWSPAN=15>
<IMG SRC="images/utama_04.jpg" WIDTH=10 HEIGHT=504 ALT=""></TD>
<TD COLSPAN=4 ROWSPAN=2>
<IMG SRC="images/utama_05.jpg" WIDTH=227 HEIGHT=51 ALT=""></TD>
<TD ROWSPAN=15>
<IMG SRC="images/utama_06.jpg" WIDTH=4 HEIGHT=504 ALT=""></TD>
<TD ROWSPAN=15>
<IMG SRC="images/utama_07.jpg" WIDTH=8 HEIGHT=504 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=8 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=14>
<IMG SRC="images/utama_08.jpg" WIDTH=8 HEIGHT=496 ALT=""></TD>
<TD COLSPAN=4 ROWSPAN=4><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.ca
b#version=6,0,29,0" width="495" height="117">
<param name="movie" value="swish%20template.swf">
<param name="quality" value="high">
<embed src="swish%20template.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash"width="495"height="117"> </embed>
</object></TD>
<TD COLSPAN=2 ROWSPAN=10>
<IMG SRC="images/utama_10.jpg" WIDTH=5 HEIGHT=309 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=43 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=4>
<IMG SRC="images/utama_11.jpg" WIDTH=227 HEIGHT=18 ALT=""></TD> <TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=18 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=3 bgcolor="#EDEF2"> <div id="bgclocknoshade" align="center"
style="font-size:24px; color:#A8A8A8"></div>
</TD>
<TD ROWSPAN=12>
<IMG SRC="images/utama_13.jpg" WIDTH=6 HEIGHT=435 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=36 ALT=""></TD>
</TR>
<form method="post" action="proses.php">
<TR>
<TD ROWSPAN=3 bgcolor="#EDEF2">username :</TD>
<TD COLSPAN=2 ROWSPAN=3 bgcolor="#EDEF2"><input type="text"
name="uname"></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=24 ALT=""></TD>
</TR>

```

```

<TR>
<TD COLSPAN=4>
<IMG SRC="images/utama_16.jpg" WIDTH=495 HEIGHT=6 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=6 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=4>
</TD>
<TD ROWSPAN=4><script type="text/javascript">

/*****
* Pausing updown message scroller- © Dynamic Drive DHTML code library
(www.dynamicdrive.com)
* This notice MUST stay intact for legal use
* Visit Dynamic Drive at http://www.dynamicdrive.com/ for full source code
*****/
//untuk menampilkan font scroll
//configure the below five variables to change the style of the scroller
var scrollerdelay='4000' //delay between msg scrolls. 4000=4 seconds.
var scrollerwidth='182px'
var scrollerheight='70px'
var scrollerbgcolor=''
//set below to '' if you don't wish to use a background image
var scrollerbackground='scrollerback.gif'
//configure the below variable to change the contents of the scroller
var messages=new Array()
messages[0]="<font face='Arial'><center><a><br>Tuntutlah Ilmu Walaupun
Sampai Ke Negeri Cina</a></font>"
messages[1]="<font face='Arial'><center><a>Seakan-akan kamu hidup 1000 Tahun
Lagi<br></a></font>"
messages[2]="<font face='Arial'><center><a><br>Jangan mencontek <br>diwaktu
tes<br></a></font>"
messages[3]="<font face='Arial'><center><a><br>Maju terus jangan mudah
menyerah yah<br></a></font>"
/////////Do not edit pass this line/////////
var ie=document.all
var dom=document.getElementById

if (messages.length>3)
i=3
else
i=0
function move(whichdiv){
tdiv=eval(whichdiv)
if (parseInt(tdiv.style.top)>0&&parseInt(tdiv.style.top)<=5){
tdiv.style.top=0+"px"
setTimeout("move(tdiv)",scrollerdelay)
setTimeout("move2(second2_obj)",scrollerdelay)
return
}
if parseInt(tdiv.style.top)>=tdiv.offsetHeight*-1){
tdiv.style.top=parseInt(tdiv.style.top)-5+"px"
setTimeout("move(tdiv)",50)
}
else{
tdiv.style.top=parseInt(scrollerheight)+"px"
tdiv.innerHTML=messages[i]
if (i==messages.length-1)
i=0
else
i++
}
}
function move2(whichdiv){
tdiv2=eval(whichdiv)
if (parseInt(tdiv2.style.top)>0&&parseInt(tdiv2.style.top)<=5){
tdiv2.style.top=0+"px"

```

```

setTimeout("move2(tdiv2)",scrollerdelay)
setTimeout("move(first2_obj)",scrollerdelay)
return
}
if (parseInt(tdiv2.style.top)>=tdiv2.offsetHeight*-1){
tdiv2.style.top=parseInt(tdiv2.style.top)-5+"px"
setTimeout("move2(second2_obj)",50)
}
else{
tdiv2.style.top=parseInt(scrollerheight)+"px"
tdiv2.innerHTML=messages[i]
if (i==messages.length-1)
i=0
else
i++
}
}
function startscroll(){
first2_obj=ie? first2 : document.getElementById("first2")
second2_obj=ie? second2 : document.getElementById("second2")
move(first2_obj)
second2_obj.style.top=scrollerheight
second2_obj.style.visibility='visible'
}

if (ie||dom){
document.writeln('<div id="main2"
style="position:relative;width:'+scrollerwidth+'height:'+scrollerheight+';o
verflow:hidden;background-color:'+scrollerbgcolor+' ;background-
image:url('+scrollerbackground+')">')
document.writeln('<div
style="position:absolute;width:'+scrollerwidth+'height:'+scrollerheight+';c
lip:rect(0 '+scrollerwidth+' '+scrollerheight+' 0);left:0px;top:0px">')
document.writeln('<div id="first2"
style="position:absolute;width:'+scrollerwidth+';left:0px;top:1px;">')
document.write(messages[0])
document.writeln('</div>')
document.writeln('<div id="second2"
style="position:absolute;width:'+scrollerwidth+';left:0px;top:0px;visibility
:hidden">')
document.write(messages[dyndetermine=(messages.length==1)? 0 : 1])
document.writeln('</div>')
document.writeln('</div>')
document.writeln('</div>')
}
if (window.addEventListener)
window.addEventListener("load", startscroll, false)
else if (window.attachEvent)
window.attachEvent("onload", startscroll)
else if (ie||dom)
window.onload=startscroll
</script></TD>
<TD COLSPAN=2 ROWSPAN=5>
<IMG SRC="images/utama_19.jpg" WIDTH=2 HEIGHT=182 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=3 ALT=""></TD>
</TR>
<TR>
<TD bgcolor="#EDEF2">password :</TD>
<TD COLSPAN=2 bgcolor="#EDEF2"><input type="password" name="pswd"></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=30 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=2 bgcolor="#EDEF2"><align="center">
<input type="submit" value="login"></p></TD>
<TD bgcolor="#EDEF2"><a href="register.php">daftar</a> </TD>
<TD>

```

```

<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=35 ALT=""></TD>
</TR></form>
<TR>
<TD COLSPAN=3 ROWSPAN=3>
<IMG SRC="images/utama_24.jpg" WIDTH=221 HEIGHT=172 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=106 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=2>
<IMG SRC="images/utama_25.jpg" WIDTH=493 HEIGHT=8 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=8 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=3 ROWSPAN=3><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.ca
b#version=6,0,29,0" width="494" height="180">
<param name="movie" value="petunjuk.swf">
<param name="quality" value="high">
<embed src="petunjuk.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-
flash"width="494"height="180"></embed></object></TD>
<TD COLSPAN=2 ROWSPAN=2>
<IMG SRC="images/utama_27.jpg" WIDTH=2 HEIGHT=184 ALT=""></TD>
<TD ROWSPAN=4>
<IMG SRC="images/utama_28.jpg" WIDTH=4 HEIGHT=187 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=58 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=3 ROWSPAN=3 bgcolor="#E1E2E4"><p><font size="4"> Created Ide
2007 By Dingga</font> </p>
<p>& Nina Susanti </p></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=126 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=2 ROWSPAN=2>
<IMG SRC="images/utama_30.jpg" WIDTH=2 HEIGHT=3 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=2 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=3>
<IMG SRC="images/utama_31.jpg" WIDTH=494 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=15>
<IMG SRC="images/utama_32.jpg" WIDTH=777 HEIGHT=84 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=84 ALT=""></TD>
</TR>
<TR>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=9 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=8 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=176 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=317 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>

```

```

<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=4 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=10 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=67 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=79 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=75 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=6 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=4 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="images/spacer.gif" WIDTH=8 HEIGHT=1 ALT=""></TD>
</TR>
</TABLE>
<!-- End ImageReady Slices -->
</BODY>
</HTML>

```

register.php

```

<html>
<head>
<title>input data</title>
<style type="text/css">
<!--
.style10 {color: #EDEEF2}
.style11 {color:#DBDBDB}
-->
</style>
</head>
<body bgcolor="#EDEEF2">
<div align="center">
<p><strong><font face="Verdana, Arial, Helvetica, sans-serif">
<object
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.ca
b#version=6,0,29,0" width="654" height="157">
<param name="movie" value="swish%20template.swf">
<param name="quality" value="high">
<embed
src="swish%20template.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash" width="654" height="157"></embed>
</object>
</font></strong></p>
<p><strong><font face="Verdana, Arial, Helvetica, sans serif">register peserta
psikotes</font></strong></p>
</div>
<form method="post" action="data user.php">
<table width="700" border="0" cellspacing="0" bordercolor="#6C7782">
<tr>
<td width="18">&nbsp;</td>
<td width="18">&nbsp;</td>
<td width="18">&nbsp;</td>
<td width="18">&nbsp;</td>
<td width="202">&nbsp;</td>
<td width="116">username</td>
<td width="296"><input type="text" name="uname" maxlength="25"></td>
</tr>
<tr>

```

```

<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>password</td>
<td><input type="password" name="pswd"smmaxlength="25"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>nama</td>
<td><input type="text" name="nama"maxlength="30"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>jenis kelamin</td>
<td><input type="radio" value="wanita"checked name="jkel">wanita<input
type="radio" value="pria" checked name="jkel">pria</td>
</tr><tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>alamat </td>
<td><textarea name="alamat" rows="4"cols="30"></textarea
></td></tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>email</td>
<td><input type="text" name="email" size="25" maxlength="30"></td>
</tr>
<tr>
<td align="center">
<td align="center">
<td align="center">
<td align="center">
<td align="center">
<td align="center">
<td><br><input type="submit" value="kirim"></td>
</tr><tr>
<td>
<td>
<td>
<td>
<td>
</div>
</table>
</form>
</body>
</html>

```

[login.php](#)

```

<html>
<head>

```

```

<title>login</title>
</head>
<body>
<div align="center">
<strong><font size="+1">anda harus login dahulu: <BR></font></strong>
<PRE>
<FORM NAME="frmLogIn" METHOD="post" ACTION="cekpass.php">
Username      : <INPUT TYPE="text" NAME="uname">
Password      : <INPUT TYPE="password" NAME="pswd">
<INPUT TYPE="submit" VALUE="submit">
</FORM>
</PRE>
untuk peserta baru silakan
<A HREF="register.php"> klik </A> untuk daftar dahulu
</div>
</body>
</html>

```

data.user.php

```

<html>
<head>
<title>data user</title>
<style type="text/css">
a:link {
color: #993300;
text-decoration: none;
}
a:hover {
color: #00FFCC;
text-decoration: none;
}
a:visited {
text-decoration: none;
}
a:active {
text-decoration: none;
}
</style>
</head>
<body>
<?
//$uname = $_POST['uname'];
//$stop = 0;
//$sql = "select uname from tb_peserta"; ?>
<font size="+6"><p align="center">
<?
if (!$uname||!$pswd||!$nama||!$jkel||!$alamat)
{
echo("semua form harus diisi!!!<br>");
echo("<input type=button value=back onclick=\"self.history.back();\>");
}
else
{
if(ereg("^.+@.+\\..+$",$email))
{
$conn=mysql_connect("localhost","","") or die("koneksi gagal :");
mysql_error();
mysql_select_db("db_psikotes",$conn);
$input = "insert into tb_peserta (uname,pswd,nama,jkel,alamat,email)
VALUES (' $uname', '$pswd', '$nama', '$jkel', '$alamat', '$email')";
$qry= mysql_query($input,$conn) ;
echo"data anda telah tersimpan<br>";
echo"<A HREF='login.php'> klik </A> untuk login";
}
else
{
echo"penulisan email tidak benar <br>";
echo("<input type=button value=back onclick=\"self.history.back();\>");
}
}
}

```

```

}
}
?>
</p></font>
</body>
</html>

```

proses.php

```

<?
session_start();
include('koneksi.inc');
if(!empty($uname)and!empty($pswd))
{
?>
<html>
<head>
<title>proses data</title>
<script>
function winopen(){
win=window.open("soal", "", "fullscreen,scrollbars")
startTime()
}
function startTime(){
var time= new Date();
hours= time.getHours();
mins= time.getMinutes();
secs= time.getSeconds();
closeTime=hours*3600+mins*60+secs;
closeTime+=60; // Amount of time that the window stays open in seconds
Timer();
}
function Timer(){
var time= new Date();
hours= time.getHours();
mins= time.getMinutes();
secs= time.getSeconds();   curTime=hours*3600+mins*60+secs
if (curTime>=closeTime){
if (win.closed == false){
win.close()}}
else{           window.setTimeout("Timer()",1000)}
}
</script>
</head>
<body bgcolor="#EDEF2">
<?
$uname=$_POST['uname'];
$pswd=$_POST['pswd'];
$data="select * from tb_peserta where uname='$uname' and pswd='$pswd'";
$conne=mysql_connect("localhost","","")or die ("koneksi tidak berhasil =
".mysql_error());
$db=mysql_select_db("db_psikotes",$conne);
$load=mysql_query($data);
$ccek=mysql_fetch_array($load);
if ($ccek[uname]==$uname and $ccek[pswd]==$pswd)
{
?>
<p align="center"><strong>Selamat <? echo"$uname" ?> anda berhasil login<br>
pada halaman soal nanti, <? echo"$uname" ?> akan menghadapi soal psikotes
sebanyak 30 soal pilihan ganda,masing-masing soal akan terdiri dari soal 10
tes verbal, soal 10 tes numerik, dan 10 soal tes gambar.<br>
<?echo"$uname"?>akan diberi waktu 30 menit untuk menyelesaikan soal psikotes
tersebut. Apabila dalam waktu 30 menit <? echo"$uname" ?>
tidak dapat menyelesaikannya, maka halaman soal psikotes akan tertutup
secara otomatis.

```



```

bila setuju anda klik <a href="soal.php"onClick="winopen();return
false;">SOAL</a><br>
bila anda tidak setuju anda klik<font size="+3"><a
href="logout.php">batal</a></font></strong>
<?
} }
else
{
echo"<center>username anda tidak terdaftar<center><br>";
echo"silakan anda login kembali <a href='login.php'>login</a>";
}
?>
</body>
</html>

```

soal.php

```

<?
session_start();
include('koneksi.inc');
?>
<html>
<head>
<SCRIPT LANGUAGE="JavaScript">
<!-- Begin
var up,down;
var min1,sec1;
var cmin1,csec1,cmin2,csec2;
function Minutes(data) {
for(var i=0;i<data.length;i++) if(data.substring(i,i+1)==":") break;
return(data.substring(0,i)); }
function Seconds(data) {
for(var i=0;i<data.length;i++) if(data.substring(i,i+1)==":") break;
return(data.substring(i+1,data.length)); }
function Display(min,sec) {
var disp;
if(min<=9) disp=" 0";
else disp=" ";
disp+=min+": ";
if(sec<=9) disp+="0"+sec;
else disp+=sec;
return(disp); }
function Up(){cmin1=0;csec1=0;
min1=0+Minutes(document.sw.beg1.value);sec1=0+Seconds(document.sw.beg1.value
);UpRepeat(); }
function UpRepeat() {csec1++;
if(csec1==60){ sec1=0; cmin1++; }
document.sw.disp1.value=Display(cmin1,csec1);
if((cmin1==min1)&&(csec1==sec1)) alert("Timer-CountUp Stopped");
else up=setTimeout("UpRepeat()",1000); }function Down() {
cmin2=1*Minutes(document.sw.beg2.value);csec2=0+Seconds(document.sw.beg2.val
ue);DownRepeat(); }
function DownRepeat() {
csec2--;if(csec2==--1) { csec2=59; cmin2--; }
document.sw.disp2.value=Display(cmin2,csec2);
if((cmin2==0)&&(csec2==0)) alert("Waktu Anda Sudah Habis");
else down=setTimeout("DownRepeat()",1000); }function startTime(){
var time= new Date();
hours= time.getHours();
mins= time.getMinutes();
secs= time.getSeconds();
closeTime=hours*3600+mins*60+secs;closeTime+=120; // This number is how
long the window stays open
Timer();
}
function Timer(){
var time= new Date(); hours= time.getHours();
mins= time.getMinutes();

```

```

secs= time.getSeconds();
curTime=hours*3600+mins*60+secs
if (curTime>=closeTime){
win.closeed();}
else{          window.setTimeout("Timer()",1000)}
}
function winopen()
{
win=window.open("selesai"," ", "700")
startTime()
}
// End -->
</SCRIPT>
<title>=== Soal Psikotes ===</title>
</HEAD>
</head>
<body onLoad="startTime();Down();"><div align="center"><p><strong>==== SOAL
PSIKOTES =====</strong></p>
<p><?echo"<span          class=style1><b>(Tahun          :          ",date          ("Y")
?)</b></span></p></div>
<CENTER>
<FORM NAME="sw">
<TR ALIGN="center">
<TD><INPUT type="hidden" NAME="beg2"SIZE="7"VALUE="2:00"></TD></TR>
<TR ALIGN="center">
<TD COLSPAN="2"><TD><INPUT TYPE="text" NAME="disp2" SIZE="5" style="text-
align:center;"></TD>
</TR>
</FORM>
</CENTER>          <form method="post" action="selesai.php"><?
$query= mysql_query ("select * from tb_soal order by no_soal");
while($stampil=mysql_fetch_array($query)){ $no=$stampil["no_soal"];
$soal=$stampil["soal"];
$a=$stampil["jawabA"];
$b=$stampil["jawabB"];
$c=$stampil["jawabC"];
$d=$stampil["jawabD"];
?><table width="700" align="center">
<tr>
<td width="68">&nbsp;</td>
<td colspan="2"><? echo "$no. " ?><? echo "$soal" ?></td>
</tr>
<tr><td>&nbsp;</td>
<td colspan="2">A.<?echo"$a"?> </td>
</tr>
<tr>
<td>&nbsp;</td>
<td colspan="2">B. <? echo "$b"?></td>
</tr>
<tr>
<td>&nbsp;</td>
<td colspan="2">C.<? echo "$c"?></td>
</tr>
<tr>
<td>&nbsp;</td>
<td colspan="2">D. <? echo "$d"?></td>
</tr>
<tr>
<td idth="68" valign="middle">Jawaban: <br></td>
<td width="620" valign="middle">
<input type="radio" value='A' name="isian<? echo $no; ?>">A. <? echo "$a";
?>
<input type="radio" value="B" name="isian<? echo $no; ?>">B. <? echo "$b";
?>
<input type="radio" value="C" name="isian<? echo $no; ?>">C. <? echo "$c";
?>
<input type="radio" value="D" name="isian<? echo $no; ?>">D. <? echo "$d";
?>

```

```

</td>
</tr>
<tr>
<td colspan="2"><hrsize="1"></td>
</tr>
</table>
<?
}
?>
<br><div align="center"><input type="submit" value="selesai"
onClick="winopen();return false;"></div>
</form>
</div>
</body>
</html>

```

selesai.php

```

<?
include('koneksi.inc');
?>
<html>
<head>
<title>Untitled Document</title>
<style type="text/css">
a:link {
color: #993300;
text-decoration: none;
}
a:hover {
color: #00FFCC;
text-decoration: none;
}
a:visited {
text-decoration: none;
}
a:active {
text-decoration: none;
}
</style>
</head>
<body>
<?
if($submit)
{
$hsl=0;
$nl=mysql_query("select * tb_soal order by no_soal asc");
if(!mysql_num_rows($nl))
{ echo"nilai kosong";}
else
{ while($nilai=mysql_fetch_array ($nl))
{ $no=$nilai["no_soal"];
$isian=$nilai["isi"];
switch($isian[$no])
{
case 'A':{$isian="a";break;}
case 'B':{$isian="b";break;}
case 'C':{$isian="c";break;}
case 'D':{$isian="d";break;}
}
if($isian==$isian[$no])
{
$hsl=$hsl*10;
}
else
{
$hsl=$hsl+0;

```

```

}
}
}
?>
<?
if($hsl >= 270 && $hsl <= 300)
{ $n="Istimewa";
$ok="update tb_peserta set nilai=$hsl, kecerdasan=$n where uname=$uname";
$qry=mysql_query($ok);
}
elseif ($hsl <= 240 && $hsl < 270 )
{
$n="Cerdas";
$qry=mysql_query("update tb_peserta set nilai='$hsl', kecerdasan='$n' where
uname='$uname'");
}
elseif ($hsl <= 210 && $hsl < 240 )
{
$n="Rata-rata cerdas";
$qry=mysql_query("update tb_peserta set nilai='$hsl', kecerdasan='$n' where
uname='$uname'");
}
elseif ($hsl <= 180 && $hsl < 210 )
{ $n="sedang";
$qry=mysql_query("update tb_peserta set nilai='$hsl', kecerdasan='$n' where
uname='$uname'");
}
elseif ($hsl <= 150 && $hsl < 180 )
{ $n="Rata-rata lambat";
$qry=mysql_query("update tb_peserta set nilai='$hsl', kecerdasan='$n' where
uname='$uname'");
}
elseif ($hsl <= 120 && $hsl < 150 )
{ $n="lambat";
$qry=mysql_query("update tb_peserta set nilai='$hsl', kecerdasan='$n' where
uname='$uname'");
}
elseif ($hsl < 120 )
{ $n="sangat lambat";
$qry=mysql_query("update tb_peserta set nilai='$hsl', kecerdasan='$n' where
uname='$uname'");
}
}??>
<p align="center">&nbsp;&nbsp;&nbsp;</p>
<p align="center">&nbsp;&nbsp;&nbsp;</p>
<p align="center">Terima kasih <? echo "<b>$uname </b>"; ?> semoga nilai
anda baik </p>
<p align="center">Sampai bertemu lagi </p>
<p align="center"><a href="datanilai.php"
onClick="window.close">Selesai...</a></p>
</body>
</html>

```

datanilai.php

```

<?
session_start();
include('koneksi.inc');
if(!empty($uname)and!empty($pswd))
{
?>
<html>
<head>
<title>Data Nilai</title>
<script type = "text/javascript">
<!--

```

```

        var link = document.createElement('link');
        link.setAttribute('href', 'images/poll_bars.css');
        link.setAttribute('rel', 'stylesheet');
        link.setAttribute('type', 'text/css');
        var head = document.getElementsByTagName('head').item(0);
        head.appendChild(link);
        //-->
    </script>
<style type="text/css">
<!--
.style3 {font-size: 24px}
-->
</style>
</head>
<body bgcolor="#A8A8A8">
<p align="center" style="font-size:20px; color:#FFFFFF ">Data Nilai Peserta
Psikotes</p>
<?
$que="select * from tb_hasil where uname='$uname'";
$has=mysql_query($que);
while($roww=mysql_fetch_array($has)){
if($roww['nilai'] >= 270 && $roww['nilai'] <= 300){
    $jmlpoll=jmlpoll+$roww['nilai'];
}
else if ($roww['nilai'] <= 240 && $roww['nilai'] <= 270 ){
    $jmlpoll=26+$roww['nilai'];
}
else if ($roww['nilai'] <= 210 && $roww['nilai'] <= 240 ){
    $jmlpoll=60+$roww['nilai'];
}
else if ($roww['nilai'] <= 180 && $roww['nilai'] <= 210 ){
    $jmlpoll=120+$roww['nilai'];
}
else if ($roww['nilai'] <= 150 && $roww['nilai'] <= 180 ){
    $jmlpoll=180+$roww['nilai'];
}
else if ($roww['nilai'] <= 120 && $roww['nilai'] <= 150 ){
    $jmlpoll=250+$roww['nilai'];
}
else if($roww['nilai'] <= 120 ){
    $jmlpoll=300+$roww['nilai'];
}
}
}
$jmlpol=$jmlpoll;
$quer="select * from tb_hasil where uname='$uname'";
$hasi=mysql_query($quer);
$i=1;
while($ro=mysql_fetch_array($hasi)){
    $jwb="&nbsp;". $ro['judul'];
    $hsl="&nbsp;<strong>". $ro['nilai']. "</strong>";
    $h_jwb_a=sprintf("%01.1f", (($ro['nilai']/$jmlpol)*100)). "%";
    $g_jwb_a=$h_jwb_a*3;
    $sen="&nbsp;". $h_jwb_a;
    $gbr="&nbsp;<img
                                                src=\"images/blank.png\"
class=\"polls_color_\".$i++."\" height=\"4\" width=\"\"$g_jwb_a\" alt=\"\"/>";
if($i%2 != 0)$class = "class=\"sectiontableentry1\"";
if($i%2 == 0)$class = "class=\"sectiontableentry2\"";
?>
<table border="2" cellspacing="2" cellpadding="2" align="center"
width="40%">
<?
$query=mysql_query("select * from tb_hasil where uname='$uname'") or
die(mysql_error());
while($data=mysql_fetch_array($query))
{
$username=$data['uname'];
$nama=$data['nama'];
$nilai=$data['nilai'];

```

```

}
?>
<tr>
<td colspan="3"><font color="#FFFFFF" size="6"><?
echo"$uname";?></font></td>
</tr>
<tr>
<td width="10%"><font color="#FFFFFF" size="6"><?
echo"$nilai";?></font></td>
<td width="13%"><? echo $sen;?></td>
<td width="77%"><? echo $gbr;?></td>
</tr>
</table>
<? }?>
</body>
</html>
<?
}
else
{
echo"<center>username anda tidak terdaftar<center><br>";
echo"silakan anda login kembali <a href='login.php'>login</a>";
}
?>

```

Admin.php

```

<?
session_start();
include('koneksi.inc');
if(!empty($username)and !empty($pswd))
{
?>
<HTML>
<HEAD>
<TITLE>admin</TITLE>
<script language=javascript>
<!-- Beginning of JavaScript -
//untuk menampilkan waktu saat ini
function clockon() {
thistime= new Date()
var hours=thistime.getHours()
var minutes=thistime.getMinutes()
var seconds=thistime.getSeconds()
if(eval(hours)<10) {hours="0"+hours}
if(eval(minutes)<10) {minutes="0"+minutes}
if(seconds<10) {seconds="0"+seconds}thistime hours+":"+minutes+":"+seconds
if(document.all) {
bgclocknoshade.innerHTML=thistime
}
if(document.layers) {
document.bgclockshade.document.write('<div id="bgclocknoshade"
align="center" style="font-size:50px;color:#A8A8A8">'+thistime+'</div>')
document.close()
}
var timer=setTimeout("clockon()",200)
}// - End of JavaScript - -->
</script>
<style type="text/css">
<!--
a:link {
color: #993300;
text-decoration: none;
}
a:hover {
color: #00FFCC;
text-decoration: none;
}
}

```

```

a:visited {
text-decoration: none;
}
a:active {
text-decoration: none;
}.style1 {
color: #CCCCCC;
font-weight: bold;
}-->
</style>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0 MARGINHEIGHT=0
onLoad="clockon()"><div align="center"><TABLE WIDTH=750 BORDER=0
CELLPADDING=0 CELLSPACING=0><TR>
<TD ROWSPAN=31>
<IMG SRC="../images/admin_01.gif" WIDTH=9 HEIGHT=598 ALT=""></TD>
<TD COLSPAN=14>
<IMG SRC="../images/admin_02.gif" WIDTH=731 HEIGHT=97 ALT=""></TD>
<TD COLSPAN=2 ROWSPAN=5> <IMG SRC="../images/admin_03.gif" WIDTH=2
HEIGHT=123 ALT=""></TD>
<TD ROWSPAN=32>
<IMG SRC="../images/admin_04.gif" WIDTH=8 HEIGHT=599 ALT=""></TD>
<TD>
<IMG SRC="../images/spacer.gif" WIDTH=1 HEIGHT=97 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=14>
<IMG SRC="../images/admin_05.gif" WIDTH=731 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR><TD COLSPAN=9>
<IMG SRC="../images/admin_06.gif" WIDTH=532 HEIGHT=1 ALT=""></TD>
<TD COLSPAN=4 ROWSPAN=2>
<IMG SRC="../images/admin_07.gif" WIDTH=198 HEIGHT=24 ALT=""></TD>
<TD ROWSPAN=3>
<IMG SRC="../images/admin_08.gif" WIDTH=1 HEIGHT=25 ALT=""></TD>
<TD>
<IMG SRC="../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=2>
<IMG SRC="../images/admin_09.gif" WIDTH=1 HEIGHT=24 ALT=""></TD>
<TD COLSPAN=7>
<IMG SRC="../images/admin_10.gif" WIDTH=215 HEIGHT=23 ALT=""></TD>
<TD ROWSPAN=3 bgcolor="#8898AE"><div align="center"style="font-size:18px;
color:#FFFFFF " ><strong><? echo date("d-M-Y");?></strong></div></TD>
<TD>
<IMG SRC="../images/spacer.gif" WIDTH=1 HEIGHT=23 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=7>
<IMG SRC="../images/admin_12.gif" WIDTH=215 HEIGHT=1 ALT=""></TD>
<TD COLSPAN=4>
<IMG SRC="../images/admin_13.gif" WIDTH=198 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=4 ROWSPAN=3>
<IMG SRC="../images/admin_14.gif" WIDTH=130 HEIGHT=116 ALT=""></TD>
<TD COLSPAN=4>
<IMG SRC="../images/admin_15.gif" WIDTH=86 HEIGHT=2 ALT=""></TD>
<TD COLSPAN=2>
<IMG SRC="../images/admin_16.gif" WIDTH=68 HEIGHT=2 ALT=""></TD>
<TD COLSPAN=5 ROWSPAN=2>
<IMG SRC="../images/admin_17.gif" WIDTH=133 HEIGHT=108 ALT=""></TD>
<TD>

```

```

<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=2 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=3>
<IMG SRC="../../../images/admin_18.gif" WIDTH=1 HEIGHT=147 ALT=""></TD>
<TD COLSPAN=5 ROWSPAN=3 bgcolor="#FFFFFF"><strong><div id="bgclocknoshade"
align="center" style="font-size:50px; color:#D6DAE3"></div></strong></TD>
<TD ROWSPAN=13>
<IMG SRC="../../../images/admin_20.gif" WIDTH=1 HEIGHT=258 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=106 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=5 ROWSPAN=2 bgcolor="#8997AB"><center><a href="logout.php"><font
color="#FFFFFF"><strong>logout</strong></font></a></center> </TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=8 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=4 bgcolor="#8997AB"><center><font color="#FFFFFF"><strong>data
admin</strong></font></center></TD><TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=33 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=2>
<IMG SRC="../../../images/admin_23.gif" WIDTH=1 HEIGHT=36 ALT=""></TD>
<TD COLSPAN=5 ROWSPAN=2 bgcolor="#D6DAE3"><center><font
color="#FFFFFF"><strong><a href="insoal.php">input
soal</a></strong></font></center></TD>
<TD COLSPAN=4>
<IMG SRC="../../../images/admin_25.gif" WIDTH=467 HEIGHT=20 ALT=""></TD>
<TD COLSPAN=4 ROWSPAN=3 bgcolor="#D6DAE3"><center><strong><a
href="ganbio.php">rubah username</a></strong></font></center> </TD>
<TD ROWSPAN=4>
<IMG SRC="../../../images/admin_27.gif" WIDTH=1 HEIGHT=39 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=20 ALT=""></TD> </TR>
<TR>
<TD COLSPAN=4 ROWSPAN=17><div align="center"></div>
<div align="center"><strong><font color="#000000" size="3">Menu Admin
merupakan menu yang memiliki fasilitas untuk penambahan data, melihat data,
menghapus data, penyalahgunaan menu admin akan mengakibatkan ketidak cocokan
data, oleh karena itu untuk menghindari penyalahgunaan. setelah selesai
dimenu admin ini wajib untuk melakukan logout</font></strong></div></TD>
<TD><IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=16 ALT=""></TD>
</TR><TR>
<TD COLSPAN=6 ROWSPAN=3 bgcolor="#D6DAE3"><center class="style1">
<a href="tmso.php">data soal</a>
</center></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=2 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=4>
<IMG SRC="../../../images/admin_30.gif" WIDTH=132 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=5 ROWSPAN=3>
<IMG SRC="../../../images/admin_31.gif" WIDTH=133 HEIGHT=34 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=32 ALT=""></TD>
</TR>
<TR>

```



```

<TD COLSPAN=6>
<IMG SRC="../../../images/admin_32.gif" WIDTH=132 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=2 ROWSPAN=10>
<IMG SRC="../../../images/admin_33.gif" WIDTH=2 HEIGHT=115 ALT=""></TD>
<TD COLSPAN=3 ROWSPAN=2 bgcolor="#D6DAE3"><center><font
color="#FFFFFF"><strong><a href="danil.php">data
nilai</a></strong></font></center></TD>
<TD ROWSPAN=8>
<IMG SRC="../../../images/admin_35.gif" WIDTH=1 HEIGHT=78 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR><TD COLSPAN=4 ROWSPAN=3>
<IMG SRC="../../../images/admin_36.gif" WIDTH=132 HEIGHT=38 ALT=""></TD>
<TD ROWSPAN=17>
<IMG SRC="../../../images/admin_37.gif" WIDTH=1 HEIGHT=255 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=36 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=3>
<IMG SRC="../../../images/admin_38.gif" WIDTH=129 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=3 ROWSPAN=4
bgcolor="#D6DAE3"><center><strong><a href="dapes.php">data
peserta</a></strong></font></center></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=5>
<IMG SRC="../../../images/admin_40.gif" WIDTH=133 HEIGHT=35 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=35 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=5>
<IMG SRC="../../../images/admin_41.gif" WIDTH=133 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=6>
<IMG SRC="../../../images/admin_42.gif" WIDTH=1 HEIGHT=128 ALT=""></TD>
<TD COLSPAN=4 ROWSPAN=3>
<IMG SRC="../../../images/admin_43.gif" WIDTH=132 HEIGHT=30 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=2 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=3>
<IMG SRC="../../../images/admin_44.gif" WIDTH=129 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=2>
<IMG SRC="../../../images/admin_45.gif" WIDTH=1 HEIGHT=37 ALT=""></TD>
<TD COLSPAN=3 ROWSPAN=2>
<IMG SRC="../../../images/admin_46.gif" WIDTH=129 HEIGHT=37 ALT=""></TD>
<TD>

```

```

<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=27 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=2>
<IMG SRC="../../../images/admin_47.gif" WIDTH=1 HEIGHT=97 ALT=""></TD>
<TD COLSPAN=3 ROWSPAN=2>
<IMG SRC="../../../images/admin_48.gif" WIDTH=131 HEIGHT=97 ALT=""></TD>
<TD><IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=10 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=5>
<IMG SRC="../../../images/admin_49.gif" WIDTH=1 HEIGHT=125 ALT=""></TD>
<TD COLSPAN=5 ROWSPAN=3>
<IMG SRC="../../../images/admin_50.gif" WIDTH=131 HEIGHT=89 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=87 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=2 ROWSPAN=4>
<IMG SRC="../../../images/admin_51.gif" WIDTH=130 HEIGHT=38 ALT=""></TD>
<TD COLSPAN=2 ROWSPAN=2>
<IMG SRC="../../../images/admin_52.gif" WIDTH=2 HEIGHT=2 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=3>
<IMG SRC="../../../images/admin_53.gif" WIDTH=1 HEIGHT=37 ALT=""></TD>
<TD COLSPAN=4 ROWSPAN=2>
<IMG SRC="../../../images/admin_54.gif" WIDTH=467 HEIGHT=36 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=2>
<IMG SRC="../../../images/admin_55.gif" WIDTH=1 HEIGHT=36 ALT=""></TD>
<TD COLSPAN=4 ROWSPAN=2>
<IMG SRC="../../../images/admin_56.gif" WIDTH=130 HEIGHT=36 ALT=""></TD>
<TD COLSPAN=2 ROWSPAN=2>
<IMG SRC="../../../images/admin_57.gif" WIDTH=2 HEIGHT=36 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=35 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=4>
<IMG SRC="../../../images/admin_58.gif" WIDTH=467 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD COLSPAN=14 ROWSPAN=3>
<IMG SRC="../../../images/admin_59.gif" WIDTH=731 HEIGHT=16 ALT=""></TD>
<TD ROWSPAN=3>
<IMG SRC="../../../images/admin_60.gif" WIDTH=1 HEIGHT=16 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=14 ALT=""></TD>
</TR>
<TR>
<TD ROWSPAN=2>
<IMG SRC="../../../images/admin_61.gif" WIDTH=9 HEIGHT=2 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
</TR>
<TR>
<TD>
<IMG SRC="../../../images/admin_62.gif" WIDTH=8 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>

```

```

</TR>
<TR>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=9 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=127 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=83 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=316 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=67 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=129 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=1 HEIGHT=1 ALT=""></TD>
<TD>
<IMG SRC="../../../images/spacer.gif" WIDTH=8 HEIGHT=1 ALT=""></TD>
<TD></TD>
</TR>
</TABLE>
</div>
<?
}
else
{
?>
<body bgcolor="#D6DAE3"><div align="center"><font size="+2"><strong>anda
tidak berhak mengakses halaman ini<br>
silakan <a href="logad.php">login</a> dahulu</strong></font></div></body>
<?
}
?>
</BODY>
</HTML>

```

cekloginadmin.php

```

<?
if (submit)
{
include('koneksi.inc');
$data="select * from tb_admin where username='$username' and pswd='$pswd'";
$hasil=mysql_query($data);
$row=mysql_fetch_array($hasil);
if ($row[username]==$username and $row[pswd]==$pswd )
{
session_start();

```

```

session_register("username");
session_register("pswd");
$username=$row[username];
$pswd=$row[pswd];
if ($row[username]==$username)
{header("Location:admin.php");}
}
else
{
echo"anda gagal login";
echo"<ahref='logad.php'>login</a>";
}
}
?>

```

dapes.php

```

<?
session_start();
include('koneksi.inc');
if(!empty($username)and !empty($pswd))
{
?>
<html>
<head>
<title>Data Peserta</title>
</head>
<body bgcolor="#CCCCCC">
<table align="right">
<tr>
<td><strong><a href="admin.php">admin</a></strong> </td>
<td></td>
<td><strong><a href="logout.php"> logout</a></strong></td>
</tr>
</table>
<br>
<br>
<?
$query=mysql_query("select * from tb_peserta");
while($stampil=mysql_fetch_array($query))
{
$username=$stampil["uname"];
$nama=$stampil["nama"];
$jkel=$stampil["jkel"];
$alamat=$stampil["alamat"];
$email=$stampil["email"];
?>
<table width="700" align="center">
<tdwidth="15%"><strong>Nama</strong></td><td width="50%"><strong>:<?
echo"$nama" ?></strong></td>
</tr>
<tr><td width="15%"><strong>Jenis
Kelamin</strong></td><td><strong>:<?echo"$jkel" ?></strong></td>
</tr>
<tr><tdwidth="15%"><strong>Alamat</stro></td> <td><strong>:<? echo"$alamat"
?></strong></td>
</tr>
<tr><tdwidth="15%"><strong>email</strong></td><td><strong>:<?echo"$email" ?><
/strong></td>
</tr><tr><td><strong><?echo"<a
href=$PHP_SELF?delete=oke&nama=$nama>hapus</a>" ?></strong></td>
</tr>
<tr align="center">
<td colspan="2"><hralign="center" size="1" color="#FFFFFF"></td>
</tr>
</table>
<?
if($delete=="oke")

```

```

{
$data=mysql_query("delete from tb_peserta where uname='$uname'");
if($data)
echo"data telah dihapus<a href=$PHP_SELF>refresh</a>";
}
else
{
echo"data tidak dihapus";
}
?>
<?
}
}
else
{
?>
<body bgcolor="#D6DAE3"><div align="center"><font size="+2"><strong>anda
tidak berhak mengakses halaman ini<br>
silakan <a href="logad.php">login</a> dahulu</strong></font></div></body>
<?
}
?>
</body>
</html>

```

insoal.php

```

<?
session_start();
include('koneksi.inc');
if(!empty($username)and !empty($pswd))
{
?>
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resizedif
(init==true) with (navigator){if
((appName=="Netscape")&&(parseInt(appVersion)==4)) {
document.MM_pgW=innerWidth; document.MM_pgH=innerHeight;
onresize=MM_reloadPage; }}elseif(innerWidth!=document.MM_pgW ||
innerHeight!=document.MM_pgH) location.reload();
}
MM_reloadPage(true);
//-->
</script>
<!-- Mulai tinymce -->
<script type="text/javascript"
src="tinymce/jscripts/tiny_mce/tiny_mce_src.js"></script>
<script type="text/javascript">
tinyMCE.init({
theme : "advanced",
language : "en",
mode : "specific_textareas",//document_base_url :
"http://localhost/joomla/",
relative_urls : false,
remove_script_host : false,
theme_advanced_toolbar_location : "top",directionality: "ltr",
force_br_newlines : "false",
force_p_newlines : "true",
cleanup : truecleanup_on_startup : false, plugins:"advlink,
advimage,,preview, insertdatetime, emotions, table, style",
theme_advanced_buttons2_add : ", preview, insertdate, inserttime, emotions",
theme_advanced_buttons3_add : ", tablecontrols, fullscreen, styleprops",
plugin_insertdate_dateFormat : "%Y-%m-%d",
plugin_insertdate_timeFormat : "%H:%M:%S",
extended_valid_elements : "a[name|href|target|title|onclick],
img[class|src|border=0|alt|title|hspace|vspace|width|height|align|onmouseove
r|onmouseout|name], , hr[class|width|size|noshade]",

```

```

fullscreen_settings : {
theme_advanced_path_location : "top"}
});
function winopen(){
//win = window.open("soal.php", "", "fullscreen,scrollbars")
win=window.open("upload.php", "newWin", "toolbar=no,location=no,close=no,minim
ize=no,directories=no,status=no,menubar=no,scrollbars=yes,resizale=no,copyhi
story=no,width=600,height=300")
//startTime()
}
</script>
<!-- Selesai tinymce -->
<html>
<head>
<title>input soal</title>
<style type="text/css">
<!--
a:link {
color: #993300;
text-decoration: none;
}
a:hover {
color: #00FFCC;
text-decoration: none;
}a:visited {
text-decoration: none;
}a:active {
text-decoration: none;
}
-->
</style>
</head>
<body bgcolor="#D6DAE3" >
<table align="right">
<tr>
<td><a
href="admin.php"><font
size="+1"><strong>admin</strong></font></a></td>
<td></td>
<td><a
href="logout.php"><font
size="+1"><strong>logout</strong></font></a></td>
</tr>
</table>
<div align="center">
<p><strong><font face="Verdana, Arial, Helvetica, sans-serif"
size="+2"><strong>Soal Psikotes</strong></font></strong></p>
<form action="" method="post" enctype="multipart/form-data">
<table width="700" border="0" cellspacing="0">
<tr><td><strong>kode soal</strong></td><td><strong>:</strong></td>
<td><input type="text" name="kodso" size="6" maxlength="6"></td>
</tr>
<tr>
<td><strong>jenis soal</strong> </td>
<td><strong> : </strong></td>
<td><input type="text" name="jenso" size="6" maxlength="10"></td>
</tr>
<tr>
<td><strong>no soal</strong> </td>
<td><strong> :</strong> </td>
<td><input type="text" name="no_soal" size="4" maxlength="4"></td>
</tr>
<tr>
<td><strong>soal</strong> </td>
<td><strong> :</strong> </td>
<td><textarea
name="soal"
mce_editable="true"></textarea></td>
</tr>
<tr>
<td>&nbsp;</td>

```

```

<td>&nbsp;</td><td><a href="#" onClick="winopen();return false;">Lihat
Gambar</a> </td>
</tr>
<tr>
<td><strong>jawab a</strong> </td>
<td><strong> : </strong></td>
<td><input type="text" name="jawabA" maxlength="60"></td>
</tr>
<tr>
<td><strong>jawabb </strong></td>
<td><strong> :</strong> </td>
<td><input type="text" name="jawabB" maxlength="60"></td>
</tr>
<tr>
<td><strong>jawab c</strong> </td>
<td><strong> :</strong></td>
<td><input type="text" name="jawabC" maxlength="60"></td>
</tr>
<tr>
<td><strong>jawabd </strong></td>
<td><strong> :</strong> </td>
<td><input type="text" name="jawabD" maxlength="60"></td>
</tr>
<tr>
<td><strong> isi</strong></td>
<td><strong> : </strong></td>
<td><input type="radio" value="A" name="isi">A
<input type="radio" value="B" name="isi">B
<input type="radio" value="C" name="isi" >C
<input type="radio" value="D" name="isi" >D </td>
</tr>
<tr>
<td>
<p align="center">
<input type="submit" value="simpan" name="submit">
<input type="reset" value="reset">
</p></td>
</tr>
</table>
</form>
<?
if($submit)
{
if(($kodso<>"")and($jenso<>"")and($no_soal<31)and($soal<>"")and($jawabA<>"")
and($jawabB<>"")and($jawabC<>"")and($jawabD<>"")and($isi<>""))
{
$kodso=strip_tags($kodso);
$jenso=strip_tags($jenso);
$no_soal=strip_tags($no_soal);
//$soal=strip_tags($soal);
$A=strip_tags($jawabA);
$B=strip_tags($jawabB);
$C=strip_tags($jawabC);
$D=strip_tags($jawabD);
switch ($isi)
{
case 'A':{$isi="A";break;}
case 'B':{$isi="B";break;}
case 'C':{$isi="C";break;}
case 'D':{$isi="D";break;}
}
if(!move_uploaded_file($foto,"../admin/gambarsoal/$foto_name"))
{
$nilai=mysql_query("insert into
tb_soal(kodso,jenso,no_soal,soal,jawabA,jawabB,jawabC,jawabD,isi)
('$kodso','$jenso','$no_soal','$soal','$A','$B','$C','$D','$isi')");
}
}
else

```

```

{
$nilai=mysql_query("insert                                into
tb_soal(kodso,jenso,no_soal,soal,jawabA,jawabB,jawabC,jawabD,isi)      values
('$kodso','$jenso','$no_soal','$soal','$A','$B','$C','$D','$isi)");
}if($nilai)
{
echo"data berhasil diisi, ingin mengisi kembali <a href='insoal.php'> input
soal</a><br>";echo"<a href='admin.php'>selesai</a><br>";
}
else
{
echo"<font size=+3>harap diisi dengan benar </font><br>";
}
}?><?
}
else
{
?>
<center>Anda tidak berhak mengakses halaman ini.</center>
<center>Silahkan Anda <a class='login' href='logad.php'>LOGIN</a> sebagai
Administrator</center>
<?
}
?>
</div>
</body>
</html>

```

logad.php

```

<html>
<head>
<title>login</title>
<style type="text/css">
<!--
body {
background-image:url(../images/admin_12.gif);}
.style1 {
font-size: 18px;
font-weight: bold;
}
.style2 {font-size: 18px}
-->
</style>
</head>
<body >
<align="center"><font color="#FFFFFF"><span class="style1">anda harus login
dahulu </span></font></p>
<align="center"><font color="#FFFFFF"><span class="style1">sebagai ADMIN :
</span></font><font color="#FFFFFF"><span class="style1"></span>
</font></p>
<div align="center">
<FORMNAME="frmLogIn" METHOD="post" ACTION="cekloginadmin.php">
<table border="0" cellspacing="0" cellpadding="0">
<tr>
<td><font color="#FFFFFF"><span class="style2">Username</span>: </td>
<td> <INPUT TYPE="text" NAME="username"></td>
</tr>
<p>&nbsp;</p>
<tr>
<td><font color="#FFFFFF"><span class="style2">Password : </font></td>
<td><INPUT TYPE="password" NAME="pswd"></td>
</tr>
</table>
<br>
<INPUT TYPE="submit" ALUE="login">
</FORM>
</div>

```



```

</PRE>
</body>
</html>

```

tms0.php

```

<?
session_start();
include('koneksi.inc');
if(!empty($username)and !empty($pswd))
{
?>
<html>
<head>
<title>tampil data</title>
</head>
<body bgcolor="#66CCFF">
<align="center" class="style1"><font size="+4">Data Soal
psikotes</font></p>
<div align="right">
<table>
<tr>
<td><a href="admin.php"><font
size="+1"><strong>admin</strong></font></a></td>
<td width="45"><font size="+1"><a href="logout.php">logout</a></font></td>
</tr>
</table>
</div>
<div align="center">
<table align="center"width="700" border="0"> <tr>
<tdwidth="50" bgcolor="#FFFFCC"><div align="center"><strong>kode
soal</strong></div></td>
<td width="37" bgcolor="#FFFFCC"><div
align="center"><strong>No_soal</strong></div></td>
<td width="446" bgcolor="#FFFFCC"><div
align="center"><strong>Soal</strong></div>
<tdwidth="50" bgcolor="#FFFFCC"><div align="center"><strong></strong></div>
<tdwidth="50" bgcolor="#FFFFCC"><div align="center"><strong></strong></div>
</tr>
<?$oke=mysql_query("select * from tb_soal order by no_soal");
if (!mysql_num_rows($oke))
{echo "data kosong" ;
}
else
{
while($stampil = mysql_fetch_array ($oke))
{$kodso=$stampil["kodso"];$nom=$stampil["no_soal"];
$soal=$stampil["soal"];
echo "<tr bgcolor=#" .($kode % 2? "F0F0F0" : "cccccc").">";
?>
<td><? echo "$kodso" ?></td>
<td><center><?echo "$no_soal";?></center></td>
<?$kata=strtok($soal," ");
for ($i=1; $i<=14; $i++)
{ }?><td><? echo "$kata...<a href=selengkapnya.php?kodso=$kodso>
Selengkapnya </a> "?></td>d<? echo "<ahref=edit_soal.php?nom=$nom>Edit</a>"
?></td>
<td><?echo"<a href=$PHP_SELF?hapus=siip&kodso=$kodso>Hapus</a>" ?></td>
</tr>
<?
} //while
} //if
?>
<?if($hapus=="siip")

```

```

{
$hasil=mysql_query("delete from tb_soalwhere no_soal='$no_soal'");
if($hasil) echo>Data telah dihapus silahkan <a href=$PHP_SELF>Refresh</a>;
}
?>
<?
}
else
{
?>
<center>Anda tidak berhak mengakses halaman ini.</center>
<center>Silahkan Anda <a class="login" href="logad.php">LOGIN</a> sebagai
administrator</center>
<?
}
?>
</table>
</body>
</html>

```

edit soal.php

```

<?
session_start();
include('koneksi.inc');
if(!empty($username)and !empty($pswd))
{
?>
<html>
<head>
<title>edit soal</title>
</head>
<body>
<?
$perintah=mysql_query("select* from tb_soal order by kodso");
$row=mysql_fetch_array($perintah);
$kodso=$row["kodso"];
$no=$row["no_soal"];
$soal=$row["soal"];
$A=$row["jawabA"];
$B=$row["jawabB"];
$C=$row["jawabC"];
$D=$row["jawabD"];
$isian=$row["isi"];
?>
<align="center"><strong><font color="#CCCCCC" size="+3">Edit
Soal</font></strong></p>
<form method="post" action="">
<table align="right">
<tr>
<td align="right" width="381"><strong><font color="#CCCCCC"size="+2"><a
href="logout.php">logout</a></font></strong></td>
</tr>
</table>
<br>
<br>
<table border="1">
<tr>
<td width="67">kode soal </td>
<td width="3">:</td>
<td width="396"><? echo "$kodso"; ?></td>
</tr>
<tr>
<td>nomor soal</td>
<td>:</td>
<td><? echo "$no"; ?></td>
</tr>
<tr>

```

```

<td>soal</td>
<td>:</td>
<td><textarea name="soal2" rows="6" cols="60"><? echo "$soal";
?></textarea></td>
</tr>
<tr>
<td>A</td>
<td>:</td>
<td><input type="text" name="a1" value="<? echo "$A"; ?>" size="66"></td>
</tr>
<tr>
<td>B</td>
<td>:</td>
<td><input type="text" name="b1" value="<? echo "$B"; ?>" size="66"></td>
</tr>
<tr>
<td>C</td>
<td>:</td>
<td><input type="text" name="c1" value="<? echo "$C"; ?>" size="66"> </td>
</tr>
<tr>
<td>D</td>
<td>:</td>
<td><input type="text" name="d1" value="<? echo "$D"; ?>" size="66"></td>
</tr>
<td>isi</td>
<td>:</td>
<td><input type="radio" name="isian <? echo "$no" ?>">A
<input type="radio" name="isian<? echo "$no" ?>">B
<input type="radio" name="isian<? echo "$no" ?>">C
<input type="radio" name="isian<? echo "$no" ?>">D</td>
</tr>
</table>
<align="center"><input type="submit" name=" kirim" value="edit">
<a href="tms0.php">Kembali</a></p>
</form>
<?
if ($ kirim)
{
switch ($isian)
{
case '1':{$isian="A";break;}
case '2':{$isian="B";break;}
case '3':{$isian="C";break;}
case '4':{$isian="D";break;}
}
$edit="update b_soal SET soal='$soal2',jawabA='$a1',
jawabB='$b1',jawabC='$c1', jawabD='$d1', isi='$isian' where no_soal=$nom";
if(!empty($soal2)and !empty ($a1) and !empty ($b1) and !empty ($c1) and
!empty ($d1) and !empty ($isian))
{
mysql_query($edit);
echo"soal berhasil di update<br>";echo"kembali
kehalaman<a href='admin.php'>admin</a>";
}else
{echo "maaf anda harap mengisi dengan benar yah !!! <br>";
echo "<a href=edit_soal.php?kodso=$kodso>Ulang</a>";
}
}
else
{
?>
<p align="center">anda tidak berhak mengakses halaman ini <br>
silakan anda <a href="logad.php">login</a> sebagai admin</p>
<?
}
?>

```

```
</body>
</html>
```

selengkapnya.php

```
<?
session_start();
include('koneksi.inc');
if(!empty($username)and !empty($pswd))
{
?>
<html>
<head>
<title>tampil data soal</title>
</head>
<body bgcolor="#D6DAE3">
<?
$data=mysql_query("select * from tb_soal where kodso='$kodso'");
while($stampil=mysql_fetch_array($data))
{
$kodso=$stampil["kodso"];
$no_soal=$stampil["no_soal"];
$soal=$stampil["soal"];
$jawabA=$stampil["jawabA"];
$jawabB=$stampil["jawabB"];
$jawabC=$stampil["jawabC"];
$jawabD=$stampil["jawabD"];
$isi=$stampil["isi"];
?>
<tablealign="center" cellpadding="0" cellspacing="2" width="700">
<tr><td width="135"><strong>kode soal</strong></td>
<td width="12">: </td>
<td width="543"><strong><?echo "$kodso"?></strong></td>
</tr><tr><td><strong>nomor soal</strong></td>
<td>:</td>
<td><strong><?echo "$no_soal"?></strong></td>
</tr>
<tr>
<td><strong>soal</strong></td>
<td>:</td>
<td><strong><?echo "$soal"?></strong></td>
</tr>
<tr>
<td><strong>A</strong></td>
<td>: </td>
<td><strong><?echo "$jawabA"?></strong></td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>:</td>
<td><strong><?echo "$jawabB"?></strong></td>
</tr>
<tr>
<td><strong>C</strong></td>
<td>:</td>
<td><strong><?echo "$jawabC"?></strong></td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>:</td>
<td><strong><?echo "$jawabD"?></strong></td></tr>
<tr><td><strong>jawaban</strong>
<td>:</td><td><strong><?echo "$isi"?></strong></td></tr>
<TR>
<td></td>
<td> </td>
<TD><A href="tms0.php"><font color="#FFFFFF"
size="+1"><strong>kembali</strong></font></A></TD>
```

```

</TR></table>
<table width="700">
<tr align="center">
<td width="129"></td>
<td align="center" width="559"><hr align="center" size="2" noshade
color="#FFFFFF"></td>
</tr></table>
<?
}
?>
<?
}
else
{
?>
<align="center">anda tidak berhak mengakses halaman ini<br>
<a href="admin.php">login</a> terlebih dahulu sebagai admin</p>
<?
}
?>
</body>
</html>

```

logout.php

```

<?
session_start("admin");
unset($_SESSION['admin']);
session_destroy();
header("Location:keluar.php");
?>

```

upload.php

```

<?
session_start();
include("koneksi.inc");
if(!empty($username) and !empty($pswd))
{
?>
<head>
<title>Upload Gambar</title>
</head>
<body>
<?
switch($pilih)
{default:
if($submit)
{
if(!$foto)
{
"Pilih gambar";
} else {
$upload=@move_uploaded_file($foto,"gambarsoal/$foto_name");
if(!$upload){
echo "Upload Gambar Gagal";
} else {
echo "Upload Gambar Berhasil";
}
}
}
?><br />
<form action="" method="post" enctype="multipart/form-data">
<table>
<tr>
<td><input type="file" name="foto" />&nbsp;&nbsp;&nbsp;<input type="submit"
name="submit" value="Upload Gambar"></td></tr>
</table>

```

```

</form><hr size="1"><a href="?pilih=lihat">Lihat Gambar</a>
<li><?
break;
case "lihat":
echo "<a href=\"?\">Kembali</a><hr size=\"1\">";
//echo "Lihat";
$dir="./gambarsoal/";
$dh=opendir($dir) or die("salah");
echo "<ul type=\"square\">";
while( ($f = readdir($dh)) != false )
{
if(is_file($dir.$f))
{
echo "<li><img width=\"150\" height=\"200\" src=\"gambarsoal/$f\"><br>
htmlcode:<input name=\"htmlcode\" type=\"text\"
value=\"$url/admin/gambarsoal/$f\" size=\"75\" /></li>";
}
}
echo "</ul>";
closedir($dh);
break;
}
?>
<?
}
else
{
?>
<strong>anda tidak berhak mengakses halaman ini <br>silakan nda a
ref="admin.php">login dahulu</a><br>
<?
}
?>
</body>
</html>

```