

**PEMBUATAN *PROTOTYPE* APLIKASI PENDATAAN ALUMNI
PROGRAM STUDI TEKNIK INFORMATIKA UMB BERBASIS *WEB***

Laporan Tugas Akhir

Diajukan untuk melengkapi Salah Satu Persyaratan
Menyelesaikan Program Studi Stara Satu (S1)



Di Susun Oleh :

SANUSI

01501-085

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2008**

Abstract

Some needs grows as the times grows its follows by the growing of computer technology today. Someone needs support the growing of technology specially computer. Today, the favorite technology is computer. Other than it can solve various Human needs, computer technology also help human. On doing. Their job computer also can entertain the human himself.

Wich finally in the making of alumni web application than exist in mercu buana university especially in. Teknik Informatika major, to be able to see how much students alumny that has graduated, until it can make an information searching when ever it needed.

The writer able to made “Pembuatan Prototype aplikasi Pendataan data alumni program studi teknik informatika UMB berbasis web”. Using PHP and MySQL. In wich has the purpose to search informatika from the alumny that graduated.

Keyword : Technology, PHP and MySQL.

LEMBAR PENGESAHAN

Nama : Sanusi
NIM : 01501-085
Fakultas : Ilmu Komputer
Program Studi : Teknik Informatika
Judul Tugas Akhir : **Pembuatan *Prototipe* Aplikasi Pendataan Alumni Program Studi Teknik Informatika UMB Berbasis Web.**

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, Agustus 2008

Menyetujui,

Abdusy Syarif, ST, MT
Pembimbing Tugas Akhir I

Mengetahui,

Rapelino Ferdiansyah, ST
Pembimbing Tugas Akhir II

Mengesahkan,

Raka Yusuf, ST, MTI
Koordinator Tugas Akhir

Abdusy Syarif, ST, MT
Ketua Program Studi Teknik Informatika

ABSTRAK

Semakin berkembangnya zaman, semakin berkembangnya kebutuhan seseorang, semakin berkembang pula teknologi komputerisasi pada saat ini. Kebutuhan seseorang sangatlah mendukung dalam perkembangan teknologi, khususnya komputer. Saat ini teknologi yang paling digandrungi adalah teknologi komputerisasi. Selain dapat memecahkan berbagai kebutuhan manusia, teknologi komputerisasi juga dapat membantu manusia dalam pekerjaannya, komputer juga dapat menghibur manusia itu sendiri.

Yang mana akhirnya dalam pembuatan Aplikasi web Alumni yang ada di universitas mercubuana khususnya buat jurusan Teknik Informatika, untuk dapat melihat beberapa banyak Mahasiswa/I alumni yang sudah lulus, sehingga dapat memudahkan pencarian informasi sewaktu-waktu yang dibutuhkan.

Sehingga penulis dapat membuat "Pembuatan Prototipe Aplikasi pendataan data alumni program studi teknik informatika umb berbasis web", dengan menggunakan PHP dan MySQL. Yang bertujuan untuk mencari informasi dari para alumni yang sudah pada lulus.

Kata kuncinya : PHP dan MySQL

DAFTAR ISI

	Halaman
Halaman Pengesahan	i
Lembar Pernyataan	ii
Abstrak	iii
Abstract	iv
Kata Pengantar	v
Daftar Isi	vii
Daftar Gambar	xii
Daftar Tabel	xiv

BAB I : PENDAHULUAN

1.1 Latar Belakang	1
1.2 Tujuan Penulisan	2
1.3 Metode Penelitian	2
1.4 Rumusan Masalah	4
1.5 Batasan Masalah	4
1.6 Sistematika Penulisan	4

BAB II : LANDASAN TEORI

2.1 Sistem Development Life Cycle (SDLC)	7
2.2 Sistem	11

2.3 Informasi	11
2.3.1 Sistem Informasi	11
2.3.2 Definisi Sistem Informasi	12
2.4 Komponen Fisik Sistem Informasi	13
2.5 Konsep-konsep Sistem Informasi	14
2.6 Unified Modeling Language (UML).....	16
2.7 Use Case Diagram	17
2.8 Basis data (<i>database</i>)	19
2.8.1 Key	20
2.8.2 Normalisasi	21
2.8.3 <i>DataFlow Diagram</i> (DFD)	23
2.8.4 <i>Entity Relationship Diagram</i> (ERD)	26
2.8.5 <i>Data Dictionary</i> (DD)	28
2.8.6 <i>State Transition Diagram</i> (STD)	29
2.8.7 Diagram Alur (<i>Flowchart</i>)	29
2.9 Personal Home Page (PHP).....	31
2.9.1 Konsep kerja PHP dan MySQL	32
2.10 MySQL.....	33
2.10.1 Tipe data pada MySQL.....	35
2.9 Pendataan Alumni	35

BAB III : ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis	36
---------------------	----

3.2. Perancangan Sistem	37
3.2.1. Data Flow Diagram (DFD)	38
3.2.1.1. Rancangan Diagram Konteks	38
3.2.1.2. Diagram Level Satu	39
3.2.2. Entity Relationship Diagram (ERD)	41
3.2.3. Rancangan Unified Modelling Language (UML).....	42
3.2.3.1 Diagram Use Case	42
3.2.3.2 Diagram Aktifitas	47
3.2.3.3 Diagram Sequence	51
3.3. Struktur Tabel	52
3.4. Perancangan Antarmuka (interface).....	55
3.4.1. Rancangan Antarmuka Menu Utama Pendataan Alumni	55
3.4.2. Rancangan Antarmuka Menu Input Data	56
3.4.3. Rancangan Antarmuka Menu Data Alumni.....	57
3.4.4. Rancangan Antarmuka Menu Konfirmasi	58
3.4.5 Rancangan Antarmuka Menu Bagian Administrator	59
3.4.6. Rancangan Antarmuka Menu Setelah Admin berhasil Login.....	60
3.4.7. Rancangan Antarmuka Cek Data	62
3.4.8. Rancangan Antarmuka Tambah Data	62
3.4.9. Rancangan Antarmuka Data Alumni	64
3.4.10. Rancangan Antarmuka Konfirmasi	65

BAB IV : IMPLEMENTASI DAN PENGUJIAN

4.1.Implementasi	66
4.1.1. Implementasi Pembuatan Basis data.....	66
4.1.1.1. Implementasi Pembuatan Basis Data Tb_admin	67
4.1.1.2.Implementasi Pembuatan Basis Data Tb_Pendataan alumni.....	68
4.1.1.3. Implementasi Pembuatan basis data Tb_Konfirmasi.....	68
4.1.2. Implementasi Perancangan Layar Menu Utama.....	69
4.1.2.1. Implementasi Perancangan Layar Input Data	70
4.1.2.2. Implementasi Perancangan Layar Data Alumni	76
4.1.2.3. Implementasi Perancangan Layar Konfirmasi	78
4.1.2.4. Implementasi Perancangan Layar Login Admin.....	80
4.1.2.5. Implementasi Perancangan Layar Admin	80
4.1.2.5.1. Implementasi Perancangan Layar Cek Data.....	81
4.1.2.5.2. Implementasi Perancangan Layar Tambah Data..	82
4.1.2.5.3. Implementasi Perancangan Layar Data Alumni...	83
4.1.2.5.4. Implementasi Perancangan Layar Konfirmasi.....	84
4.2. Pengujian	85
4.2.1 Lingkungan Pengujian	86
4.2.2 Skenario Pengujian	87
4.2.3 Dokumen Hasil Pengujian.....	88
4.2.4 Analisa Hasil Pengujian	89

BAB V : PENUTUP

5.1. Kesimpulan94

5.2. Saran.....95

DAFTAR PUSTAKA.....96

LAMPIRAN (CODING)97

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Skema Model <i>Waterfall</i>	9
Gambar 2.2 Konsep sistem informasi	14
Gambar 2.3 Notasi <i>use case diagram</i>	19
Gambar 2.4 Hubungan ERD 1-1	27
Gambar 2.5 Hubungan ERD 1-many	27
Gambar 2.6 Hubungan ERD many-many	27
Gambar 2.7 Konsep Kerja PHP dan MySQL	33
Gambar 3.1 Diagram Konteks	38
Gambar 3.2 Diagram tingkat satu	40
Gambar 3.3 Admin, Alumni dan Konfirmasi	41
Gambar 3.4 Model data fisik aplikasi Pendataan Alumni	42
Gambar 3.5 Diagram use case Aplikasi Pendataan Alumni	43
Gambar 3.6 Diagram aktifitas login Admin	48
Gambar 3.7 Diagram aktifitas input data	49
Gambar 3.8 Diagram aktifitas admin	50
Gambar 3.9 Diagram sequence untuk aplikasi pendataan alumni	51
Gambar 3.10 Rancangan AntarMuka Menu Utama Pendataan Alumni	55
Gambar 3.11 Rancangan menu Input Data	57
Gambar 3.12 Rancangan Menu Data Alumni.....	58
Gambar 3.13 Rancangan AntarMuka Menu Konfirmasi	59
Gambar 3.14 Rancangan AntarMuka halaman login	60

Gambar 3.15	Rancangan AntarMuka Setelah Admin Login	61
Gambar 3.16	Rancangan AntarMuka Cek Data	62
Gambar 3.17	Rancangan AntarMuka Tambah Data	63
Gambar 3.18	Rancangan AntarMuka Data Alumni	64
Gambar 3.19	Rancangan AntarMuka Konfirmasi	65
Gambar 4.1	Implementasi Pembuatan Basis data Tabel Admin	67
Gambar 4.2	Implementasi Pembuatan Basis data Tabel Pendataan Alumni	68
Gambar 4.3	Implementasi Pembuatan Basis data Tabel Konfirmasi	69
Gambar 4.4	Perancangan Layar Menu Utama	69
Gambar 4.5	Perancangan Layar Input Data	71
Gambar 4.6	Perancangan Layar Data Alumni	76
Gambar 4.7	Perancangan Layar Konfirmasi	78
Gambar 4.8	Perancangan Layar Login Admin	80
Gambar 4.9	Perancangan Layar Admin	81
Gambar 4.10	Perancangan Layar Cek Data	82
Gambar 4.11	Perancangan Layar Tambah Data.....	83
Gambar 4.12	Perancangan Layar Data Alumni.....	83
Gambar 4.13	Perancangan Layar Konfirmasi	85
Gambar 4.14	Hasil Pengujian Halaman Input Data	89
Gambar 4.15	Hasil Pengujian Halaman Data Alumni Detail.....	90
Gambar 4.16	Hasil Pengujian Halaman cek_data	91

DAFTAR TABEL

	Halaman
Tabel 2.1 Jenis Diagram Resmi UML.....	17
Tabel 2.2 contoh normal kesatu (1NF)	22
Tabel 2.3 contoh normal kedua (2NF)	22
Tabel 2.4 contoh normal ketiga (3NF)	22
Tabel 2.5 Simbol DFD	23
Tabel 2.6 Simbol ERD	26
Tabel 2.7 Notasi standar kamus data	28
Tabel 2.8 simbol Flowchart	30
Tabel 3.1 Tabel admin	52
Tabel 3.2 Tabel Pendataan Alumni	53
Tabel 3.3 Tabel Konfirmasi	54
Tabel 4.1 Skenario Pengujian Perangkat Lunak	87
Tabel 4.2 Dokumen Hasil Perangkat Lunak	88

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi dengan menggunakan komputer sebagai basis utama nampak sudah tidak dapat dilepaskan lagi dari kehidupan kita. Komputer sebagai alat bantu yang dapat memudahkan kinerja kita serta menjadi sarana peningkatan efisiensi kerja yang amat baik.

Dalam dunia informasi saat ini begitu cepat mengalami perkembangan bahkan menjadi peran penting diberbagai bidang, sehingga tak heran dilingkungan Universitas Mercu Buana saat ini sangat membutuhkan banyak informasi-informasi terutama dalam pendataan terhadap Mahasiswa/I yang sudah pada lulus.

Dalam perkembangannya maka dibutuhkan aplikasi Pendataan Alumni yang berfungsi untuk mengisi data Mahasiswa/I yang sudah lulus sehingga lebih mudah mencari data - datanya, dan disajikan berupa dalam bentuk web.

Dengan adanya aplikasi Pendataan Alumni yang terdapat di Universitas Mercu Buana Khususnya di Program Studi Teknik Informatika yang diharapkan dapat membantu pencarian data – data Alumni dengan cepat dan mudah secara efektif artinya cepat tepat dan akurat.

1.2 Tujuan Penulis

1. Merancang sebuah sistem pendataan alumni berbasis web yang dapat digunakan untuk Alumni sebagai referensi dalam mengetahui berapa banyak Alumni yang sudah pada lulus.
2. Untuk memudahkan pencarian data – data Alumni Universitas Mercu Buana khususnya di Program Studi Teknik Informatika.

1.3 Metodologi Penelitian

Dalam penulisan tugas akhir ini, penulis melakukan penelitian dengan menggunakan metode penelitian sebagai berikut :

1. Tahapan Rekayasa Sistem

Tahapan ini Sangat menekankan pada masalah pengumpulan kebutuhan pengguna pada tingkat sistem dengan mendefinisikan konsep sistem beserta antar muka (*interface*) yang menghubungkannya dengan lingkungan sekitarnya.

2. Analisis

Melakukan pengumpulan data yang akan digunakan untuk membangun aplikasi Pelacak Alumni di Universitas Mercu Buana khususnya di Program Studi Teknik Informatika. data tersebut akan disimpan pada basis data yang kemudian akan menghasilkan web basis data.

3. Desain

Penulis melakukan perancangan arsitektur aplikasi, desain antar muka, dan pemodelan interaksi antar pengguna dengan aplikasi Pelacak Alumni Teknik Informatika berbasis web.

4. Coding

Penulis melakukan pemrograman aplikasi berdasarkan desain yang dirancang.

5. Pengujian

Penulisan melakukan pengujian terhadap seluruh program yang dibuat, pengujian dilakukan untuk mengetahui apakah aplikasi yang telah penulis kembangkan telah dapat berjalan dengan baik dan membandingkan apakah keluaran yang dihasilkan oleh aplikasi telah sesuai dengan keluaran yang diharapkan pada tahap analisa.

1.4 Rumusan Masalah

Adapun rumusan masalah yang penulis lakukan adalah:

1. Belum adanya sistem Aplikasi Pendataan Alumni untuk jurusan Teknik Informatika.
2. Datanya tidak valid.
3. Kesulitan pencarian data terhadap alumni.
4. Data base penyedia gambar atau Foto.

1.5 Batasan Masalah

Pada penulisan ini, penulis akan membahas sebagai berikut:

1. Penulis hanya melakukan pembuatan aplikasi untuk mahasiswa/I teknik Informatika yang sudah lulus, dengan dibuatkannya sebuah content management sistem untuk kebutuhan penambahan, perubahan dan penghapusan data.
2. Perancangan aplikasi sistem ini menggunakan bahasa pemrograman PHP. Dengan menggunakan database MySQL. Berbasis web application.
3. Laporan ini akan membahas mengenai sistem pelacakan yang meliputi pencarian data mahasiswa/I teknik informatika yang sudah lulus.

1.6 Sistematika Penulisan

Agar penyusunan lebih sempurna, maka laporan tugas akhir disusun dalam beberapa bab. Adapun judul dari bab-bab tersebut adalah:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang, Tujuan Penulis, Metode Penelitian, Batasan Masalah, Sistematika Penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang teori-teori dasar dari pemrograman web menggunakan bahasa pemrograman PHP dan MySQL sebagai data basenya.

BAB III PERANCANGAN DAN ANALISIS

Bab ini akan membahas tentang cara dan langkah-langkah pembuatan aplikasi web dengan menggunakan program PHP dan MySQL.

BAB IV IMPLEMENTASI DAN PENGUJIAN

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

BAB V**PENUTUP**

Pada bab ini akan membahas mengenai kesimpulan yang merupakan uraian – uraian yang diambil dari bab sebelumnya, dan saran untuk pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 *System Development Life Cycle (SDLC)*

Menurut Kenneth E. Kendall, Julie E. Kendall [P.6, 1992] *System Development Life Cycle (SDLC)* adalah fase pendekatan untuk analisa dan desain untuk suatu sistem yang dikembangkan melalui daur tertentu dari analisis dan aktifitas *user*. SDLC dibagi dalam 7 fase meski setiap fase dijelaskan terpisah namun merupakan satu kesatuan dan beberapa aktifitas dapat dilakukan bersamaan dan berulang-ulang. Maka lebih mudah menggunakan SDLC ini sebagai suatu fase, bukan suatu tingkat pengembangan. Pada SDLC ini memiliki 6 langkah yaitu:

1. *System Initiation and Feasibility Study*

Melakukan dokumentasi atas sistem yang ada saat ini, tujuan dan manfaat pembuatan sistem baru atau pengembangan dari sistem yang telah ada, identifikasi untuk alternatif solusi sistem dan penentuan solusi secara pasti.

2. *System Analisis*

Membuat sketsa awal untuk perencanaan proyek dan dokumentasi untuk sistem baru.

3. *System design*

Menghasilkan spesifikasi file database, spesifikasi input/output, spesifikasi modul program dan rencana proyek yang lebih rinci.

4. *Programming*

Membuat dan menghasilkan program yang dibutuhkan oleh sistem baru

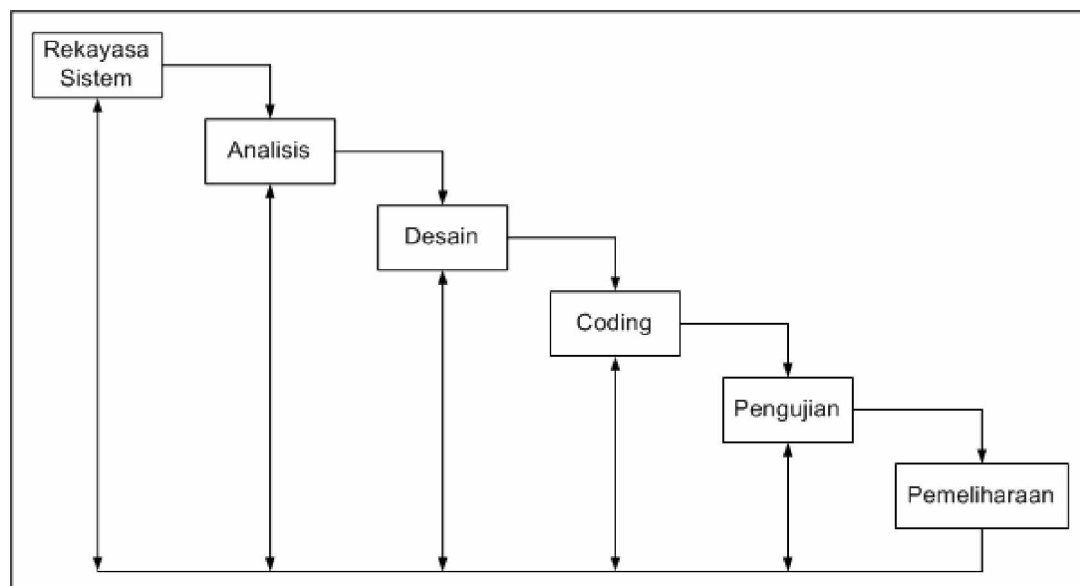
5. *Implementation*

User melakukan testing terhadap program yang telah dibuat, file database *conversion*, *user training*, dan melakukan pengenalan terhadap sistem baru.

6. *Post-implementation evaluation*

Melakukan peninjauan ulang apakah sistem ini memenuhi kebutuhan *user*.

Model *waterfall* merupakan salah satu dari model-model yang terdapat pada penerapan SDLC. Roger S. Pressman dalam bukunya yang berjudul *Rekayasa Perangkat Lunak (Pendekatan Praktisi)* membagi model *waterfall* ke dalam beberapa tahap, yaitu: tahap rekayasa sistem, analisis kebutuhan perangkat lunak, desain, generasi kode, pengujian, dan pemeliharaan yang dapat digambarkan pada Gambar 2.1.



Gambar 2.1. Skema model *Waterfall*

Tahapan-tahapan dalam *waterfall* adalah sebagai berikut

1. Tahapan Rekayasa Sistem. Tahapan ini menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak.
2. Tahapan Analisis. Pada tahap ini dilakukan pengumpulan perangkat lunak. Untuk memahami perangkat lunak yang dibangun, perekayasa perangkat lunak harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.
3. Tahapan Desain. Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut program yang berbeda; struktur data, arsitek perangkat lunak, representasi antarmuka, dan detail prosedural. Proses desain menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak

yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan mejadi bagian dari konfigurasi perangkat lunak.

4. Tahapan *Coding*. Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.
5. Tahapan Pengujian. Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional – yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.
6. Tahapan Pemeliharaan. Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya. Pemerlihaaran perangkat lunak menerapkan lagi setiap tahap program sebelumnya dan tidak membuat yang baru lagi.

2.2 Sistem

Sistem adalah sekumpulan unsur / elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk suatu tujuan. Menurut *Jerry Fitz Gerald (1981, h5)*, (penulis buku kamus komputer dan teknologi informasi) sebagai berikut :

- Sistem harus dibentuk untuk menyelesaikan tujuan.
- Elemen sistem harus mempunyai rencana yang ditetapkan.
- Adanya hubungan diantara elemen sistem.
- Tujuan organisasi lebih penting dari pada tujuan elemen.

2.3 Informasi

Informasi data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat atau prospek keputusan.

2.3.1 Sistem Informasi

Sistem Informasi adalah aplikasi komputer untuk mendukung operasi dari suatu organisasi: operasi, instalasi, dan perawatan komputer, perangkat lunak, dan data. Sistem Informasi Manajemen adalah kunci dari bidang yang menekankan finansial dan personal manajemen. Sistem Informasi Penjualan adalah suatu sistem informasi yang mengorganisasikan serangkaian prosedur dan metode yang dirancang untuk menghasilkan, menganalisa, menyebarkan dan memperoleh informasi guna mendukung pengambilan keputusan pengertian lain Sistem Informasi adalah sekumpulan *hardware*,

software, brainware, prosedur dan atau aturan yang diorganisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan, Sistem Informasi adalah satu kesatuan data olahan yang terintegrasi dan saling melengkapi yang menghasilkan *output* baik dalam bentuk gambar, suara maupun tulisan.

2.3.2 Definisi Sistem Informasi

Sebuah sistem terintegrasi, sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dan fungsi pengambilan keputusan dalam suatu organisasi. Sistem ini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual, model manajemen dan pengambilan keputusan dan basis data.

Dari definisi di atas terdapat beberapa kata kunci:

1. Berbasis komputer dan Sistem Manusia / Mesin.

Berbasis komputer: perancang harus memahami pengetahuan komputer dan pemrosesan informasi.

Sistem manusia mesin: ada interaksi antara manusia sebagai pengelola dan mesin sebagai alat untuk memroses informasi. Ada proses manual yang harus dilakukan manusia dan ada proses yang terotomasi oleh mesin. Oleh karena itu diperlukan suatu prosedur/manual sistem.

2. Sistem basis data terintegrasi

Adanya penggunaan basis data secara bersama-sama (*sharing*) dalam sebuah data base manajemen system.

3. Mendukung Operasi Informasi yang diolah dan di hasilkan digunakan untuk mendukung operasi organisasi.
4. Pemanfaatan model manajemen dan pengambilan keputusan untuk dapat mengolah data menjadi informasi yang bermanfaat digunakan model-model manajemen dan pengambilan keputusan.

2.4 Komponen Fisik Sistem Informasi

Komponen - komponen Fisik Sistem Informasi adalah sebagai berikut :

1. Perangkat keras Komputer: CPU, *Storage*, Perangkat *Input / Output*, Terminal untuk interaksi, Media Komunikasi data.
2. Perangkat lunak komputer: perangkat lunak sistem (sistem operasi dan utilitinya), perangkat lunak umum aplikasi (bahasa pemrograman), perangkat lunak aplikasi (aplikasi akuntansi dll).
3. Basis data: penyimpanan data pada media penyimpan komputer.
4. Prosedur: langkah-langkah penggunaan sistem
5. Personil untuk pengelolaan operasi (SDM), meliputi:
 - *Clerical personnel* (untuk menangani transaksi dan pemrosesan data dan melakukan *inquiry* = operator).

- *First level manager*: untuk mengelola pemrosesan data didukung dengan perencanaan, penjadwalan, identifikasi situasi *out of control* dan pengambilan keputusan level menengah ke bawah.
- *Staff specialist*: digunakan untuk analisis untuk perencanaan dan pelaporan.
- *Management*: untuk pembuatan laporan berkala, permintaan khusus, analisis khusus, laporan khusus, pendukung identifikasi masalah dan peluang, pendukung analisis pengambilan keputusan level atas.

2.5 Konsep-konsep Sistem Informasi

Untuk memahami sistem informasi diperlukan pemahaman tentang konsep-konsep: Informasi, Manusia sebagai pemroses informasi, konsep sistem, organisasi dan manajemen, konsep pengambilan keputusan, konsep nilai suatu informasi.

Informasi: data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat atau prospek keputusan. Jadi ada suatu proses transformasi data menjadi suatu informasi



Gambar 2.2 Konsep sistem informasi

Data merupakan *raw material* untuk suatu informasi. Perbedaan informasi dan data sangat relatif tergantung pada nilai gunanya bagi manajemen yang memerlukan. Suatu

informasi bagi level manajemen tertentu bisa menjadi data bagi manajemen level di atasnya, atau sebaliknya.

Representasi informasi: pelambangan informasi, misalnya: representasi biner.

Kualitas informasi: bias terhadap *error*, karena: kesalahan cara pengukuran dan pengumpulan, kegagalan mengikuti prosedur pemrosesan, kehilangan atau data tidak terproses, kesalahan perekaman atau koreksi data, kesalahan prosedur pemrosesan ketidak berfungsian sistem.

Umur informasi: kapan atau sampai kapan sebuah informasi memiliki nilai/arti bagi penggunanya. Ada *condition information* (mengacu pada titik waktu tertentu) dan *operating information* (menyatakan suatu perubahan pada suatu range waktu).

HUBUNGAN PENGELOLA DENGAN SISTEM INFORMASI

Salah satu komponen dari sistem informasi adalah personel sebagai pengelola informasi. Oleh karena itu hubungan antara sistem informasi dengan pengelolanya sangat erat. Sistem informasi yang dibutuhkan sangat tergantung dari kebutuhan pengelolanya.

Pengelola sistem informasi terorganisasi dalam suatu struktur manajemen Oleh karena itu bentuk / jenis sistem informasi yang diperlukan sesuai dengan level manajemennya.

Manajemen Level Atas: untuk perencanaan strategis, kebijakan dan pengambilan keputusan.

Manajemen Level Menengah: untuk perencanaan taktis dan pengambilan keputusan.

Manajemen Level Bawah: untuk perencanaan dan pengawasan operasi dan pengambilan keputusan.

Operator: untuk pemrosesan transaksi dan merespon permintaan.

2.6 Unified Modeling Language (UML)

Unified Modelling Language (selanjutnya disebut UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Fowler, 2005:1). Selain itu UML juga dapat diartikan sebagai sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (Dharwiyanti dan Wahono, 2003:2). UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *modeling* aplikasi prosedural dalam VB atau C.

Seperti bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram

piranti lunak. Setiap bentuk memiliki makna tertentu, dan *syntax* UML mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. UML terdiri atas 13 jenis diagram resmi seperti tertulis dalam Tabel 2.1.

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 jalur
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 artifak 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

Diagram UML yang akan dibahas pada laporan kerja praktek ini adalah *use case* diagram.

2.7 Use Case Diagram

Use Case diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* diagram mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem

tersebut digunakan (Fowler, 2005:141). *Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana (Dharwiyanti dan Wahono, 2003:4).

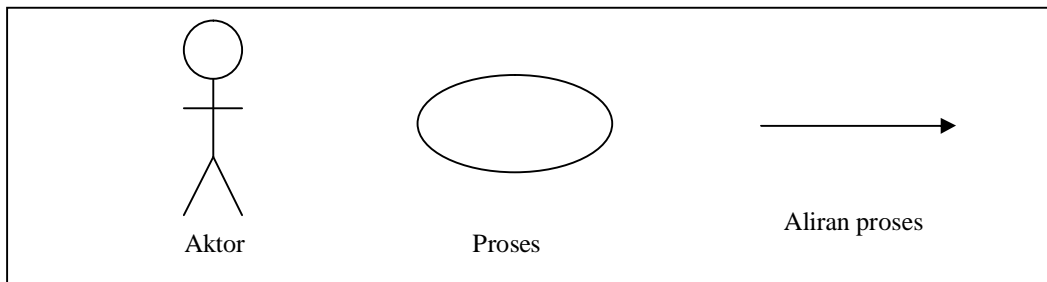
Sebuah *use case* diagram merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu bila kita sedang menyusun persyaratan (*requirement*) sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* diagram dapat meng-*include* fungsionalitas *use case* diagram lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* diagram yang di-*include* akan dipanggil setiap kali *use case* diagram yang meng-*include* dieksekusi secara normal. Sebuah *use case* diagram dapat di-*include* oleh lebih dari satu *use case* diagram lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang umum (*common*).

Sebuah *use case* diagram juga dapat meng-*extend* *use case* diagram lain dengan aktivitasnya (*behaviour*) sendiri. Sementara hubungan generalisasi antar *use case* diagram menunjukkan bahwa *use case* diagram yang satu merupakan spesialisasi dari yang lain.

Notasi-notasi yang digunakan dalam pemodelan *use case* diagram (lihat Gambar 2.3):

1. Aktor, merupakan sebuah peran yang dimainkan seorang pengguna dalam kaitannya dengan sistem,
2. Proses,
3. Aliran proses.



Gambar 2.3 Notasi *use case* diagram

2.8 Basis data (*database*)

Fathansyah, Ir (1999:2) mengatakan Basis data adalah komputerisasi sistem penyimpanan data, yang bertujuan memelihara informasi dan mengusahakan agar informasi tersedia pada saat dibutuhkan. Basis data memiliki 3 level arsitektur umum, antara lain :

1. Eksternal level

Level yang terdekat dengan *user*, dalam arti bagaimana cara dapat disajikan ataupun dilihat oleh *user*.

2. Internal level

Level yang terdekat dengan *physical storage* yaitu bagaimana pengorganisasian data yang tersimpan dalam *secondary storage volume*. Pemakai melihat data sebagai gabungan dari struktur dan datanya.

3. Konseptual level

Level yang menghubungkan dengan antara kedua level diatas dan memberikan gambaran menyeluruh dari basis data secara logikal.

Ada dua hal yang terpenting dalam membuat basis data yang baik, dua hal tersebut antara lain :

2.8.1 Key

Fathansyah, Ir (1999:39) mengatakan *Key* adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam tabel secara unik. Ada beberapa macam kunci (*key function*) yang digunakan untuk proses pencarian, penyaringan, hapus dan lain sebagainya yang biasa digunakan didalam proses pengolahan database, kunci tersebut antara lain :

1. *Candidate key* (kunci calon)

Field-field yang menjadi determinan yang dapat dijadikan identitas dari suatu record.

2. *Primary key* (kunci primer)

Adalah *candidat key* yang menjadi record, karena dapat mengidentifikasi record secara *unique*.

3. *Superkey*

Superkey adalah kumpulan *atribut* yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.

4. *Foreign key* (kunci tamu)

Adalah *non key* atribut pada sebuah relasi yang juga menjadi *primary key* atribut pada relasi lainnya. *Foreign key* biasanya digunakan sebagai penghubung antara record-record dari kedua relasi.

2.8.2 Normalisasi

Fathansyah, Ir (1999:46) mengatakan Normalisasi adalah suatu teknik yang menstrukturkan data dalam cara tertentu untuk membantu mengurangi atau mencegah timbulnya suatu masalah yang berhubungan dengan pengolahan data dalam basis data.

Proses normalisasi merupakan proses pengelompokan data elemen menjadi tabel yang menunjukkan *entity* dan relasinya, sebuah tabel dapat dikategorikan baik atau normal, jika telah memenuhi 3 kriteria berikut :

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data.
3. Tidak melanggar *Boyce-Code Normal Form (BCNF)* artinya jika semua ketergantungan fungsional dengan notasi $X \rightarrow Y$, maka X haruslah *primary key* pada tabel tersebut.

Pada proses normalisasi perlu dikenal definisi dari tahapan-tahapan normalisasi seperti berikut ini :

1. Bentuk normal kesatu (1NF/*First Normal Form*)

Setiap data dibentuk dalam *record* demi *record*, tidak ada set atribut yang berulang atau atribut dengan domain nilai yang sama.

Tabel 2.2. Contoh normal kesatu (1NF)

<u>No-pem</u>	<u>No-bar</u>	Jum
P01	B01	1000
P01	B02	1500
P02	B02	3000

2. Bentuk normal kedua (2NF/*Second Normal Form*)

Memenuhi syarat, dimana bentuk data telah memenuhi kriteria bentuk normal kesatu, dan atribut yang bukan *key* sudah tergantung penuh terhadap keynya serta memiliki ketergantungan transitif.

Tabel 2.3. Contoh normal kedua (2NF)

<u>No-pem</u>	Kode-kota	Kota
P01	1	Jakarta
P02	2	Bandung
P03	3	Surabaya

3. Bentuk normal ketiga (3NF/*Third Normal Form*)

Bentuk normal ini merupakan kriteria alternatif, jika kriteria *Boyce-Codd Normal form* (BCNF) yang ketat tidak dapat terpenuhi. Sebuah tabel dikatakan berada dalam bentuk normal ketiga jika atribut yang bukan *key* tidak tergantung transitif terhadap *key*.

Tabel 2.4. Contoh normal ketiga (3NF)

<u>No-pem</u>	<u>No-bar</u>	Jum
P01	B01	1000

P01	B02	1500
P02	B02	3000

Sedangkan *Boyce-Codd Normal Form* (BCNF) adalah Sebuah tabel dikatakan BCNF jika semua ketergantungan fungsional dinotasikan seperti $X \rightarrow A$, maka X harus merupakan *super key* pada tabel tersebut.


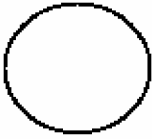
Contoh : $nim \rightarrow nama_mhs \ almt_mhs \ tgl_lahir$



2.8.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Sistem pengolahan data DFD ini memiliki simbol-simbol. Berikut ini adalah simbol-simbol DFD :

Tabel 2.5. Simbol DFD

Simbol	Keterangan simbol
	Terminator/Entitas luar Entitas diluar sistem yang berkomunikasi/berhubungan langsung Dengan system
	Proses Menggambarkan transformasi input menjadi output

	<p>Data Store</p> <p>Digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda bersifat jamak.</p>
	<p>Alur Data</p> <p>Digunakan untuk menerangkan perpindahan data/paket data dari satu bagian ke bagian lainnya.</p>

Tidak ada aturan baku untuk menggambarkan DFD, tapi dari berbagai referensi yg ada, secara garis besar :

1. Buat diagram konteks atau diagram nol.

Diagram konteks, merepresentasikan seluruh elemen sistem sebagai sebuah *bubble* tunggal dengan data *input* dan *output* yang ditunjukkan oleh anak panah yang masuk dan keluar secara berurutan. Cara untuk membuat diagram konteks adalah sebagai berikut :

- a. Tentukan nama sistemnya.
- b. Tentukan batasan sistemnya.
- c. Tentukan terminator apa saja yang ada dalam sistem.
- d. Tentukan apa yang diterima/diberikan terminator dari/pada sistem.

2. Buat diagram *level* Satu

Diagram ini digunakan untuk menggambarkan arus data secara lebih lengkap dari tahapan-tahapan proses pada diagram konteks, biasanya pada *level* satu dapat berisi lima atau enam *bubble* dengan anak panah yang saling menghubungkan dan setiap proses pada *level* ini merupakan subfungsi dari seluruh sistem yang digambarkan di dalam diagram konteks. Cara untuk menggambarkan diagram *level* satu adalah sebagai berikut :

- a. Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di *level* konteks.
- b. Tentukan apa yang diberikan/diterima masing-masing sub-proses pada/dari sistem dan perhatikan konsep keseimbangan.
- c. Apabila diperlukan, munculkan data store (transaksi) sebagai sumber maupun tujuan alur data.

1. Hindari perpotongan arus data.

2. Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya. Contoh : 1.1, 1.2, 2.1



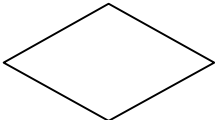

3. DFD level dua dan tiga

Diagram ini merupakan dekomposisi dari level sebelumnya. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program. Aturan yang digunakan sama dengan level satu.

2.8.4 Entity Relationship Diagram (ERD)

Fathansyah, Ir (1999:70) mengatakan *Entity Relationship Diagram* (ERD) digunakan untuk menggambarkan hubungan antar penyimpanan/data store yang terdapat pada DFD. ERD menggunakan sejumlah notasi/symbol untuk menggambarkan struktur dan hubungan antar data. ERD menggunakan 4 macam notasi/symbol antara lain seperti pada Tabel 2.6 :

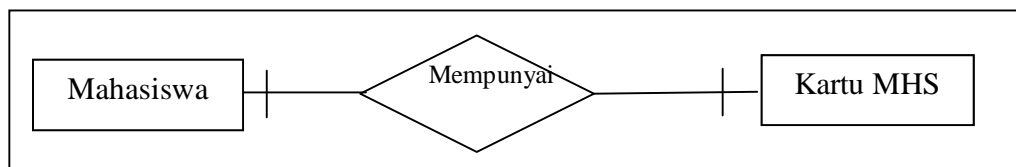
Tabel 2.6 Simbol ERD

Simbol	Keterangan symbol
	<i>Entity</i> Suatu objek yang dapat dibedakan dari objek lainnya dengan jumlah atribut-atribut tertentu.
	<i>Atribut</i> Merupakan elemen dari entity yang berfungsi mendeskripsikan karakter dari entity.
	<i>Relasi</i> Adalah suatu hubungan diantara beberapa entity.
	<i>Line</i> Menghubungkan atribut-atribut ke himpunan <i>entity</i> dan himpunan <i>entity</i> ke relasi.

Dalam ERD terdapat beberapa hubungan dalam *entity*, contohnya sebagai berikut:

1. Hubungan satu ke satu

Misalnya setiap mahasiswa harus memiliki kartu identitas mahasiswa.

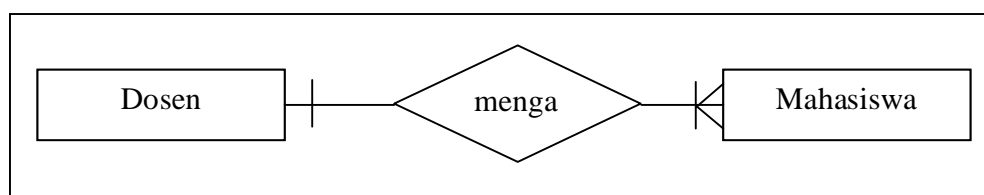


Gambar 2.4 Hubungan ERD 1-1

2. Hubungan satu ke banyak atau banyak ke satu

Misal terdapat hubungan antara seorang dosen dengan banyak mahasiswa.

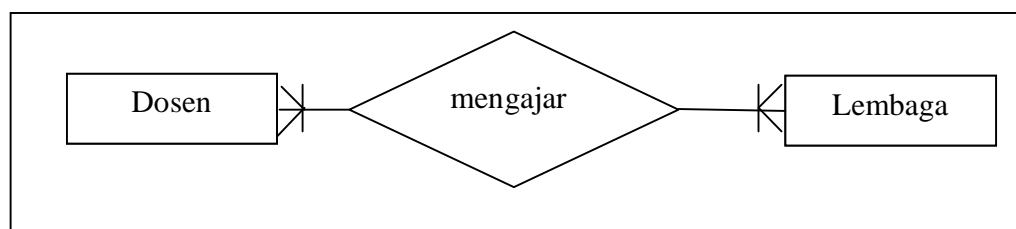
Hubungan ini dapat dibaca 1 k M atau sebaliknya.



Gambar 2.5 Hubungan ERD 1-many

3. Hubungan banyak ke banyak

Misal terdapat hubungan antara beberapa dosen dengan beberapa lembaga pendidikan. Hubungan ini dapat dibaca N ke M atau sebaliknya.



Gambar 2.6 Hubungan ERD many-many

2.8.5 Data Dictionary (DD)

Kamus data disebut juga dengan istilah sistem data dictionary :adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat berdasarkan arus data yang ada di DFD. Arus yang ada di DFD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja dan data harus mencerminkan keterangan yang jelas tentang data yang dicatat,data yang dicatatnya adalah sebagai berikut :

1. “Nama arus data” dibuat karena kamus data dibuat berdasarkan arus data yang mengalir di DFD, maka nama arus data harus dicatat di kamus data.
2. “Bentuk data” dari data ini berupa formulir, variable, paramenter, dan field dapat digunakan untuk perancangan basis data.
3. “Arus data” berfungsi untuk menunjukkan dari mana data mengalir dan kemana data menuju.
4. “Periode” digunakan untuk mengidentifikasi kapan input data harus dimasukan kedalam sistem.
5. “Struktur data” menunjukan arus data terdiri atas bermacam item data.

Notasi standar yang digunakan pada kamus data seperti pada tabel berikut ini:

Tabel 2.7. Notasi standar kamus data

Simbol	Keterangan
@	Field kunci (pengenal) dalam suatu file atau data store.
{ }	Iterasi atau pengulangan dari komponen yang ada didalamnya.
+	Dan

* *	Komentar
=	Terdiri dari atau sama dengan

2.8.6 *State Transition Diagram (STD)*

State Transition Diagram (STD) adalah suatu yang menggambarkan urutan dan variasi layer yang muncul pada saat penggunaan system menggunakan terminal tersebut. *Whitten, Bentley dan Dittman* [3] STD merupakan sebuah sarana atau alat untuk mengetahui perilaku dari system dari waktu ke waktu. Diagram ini menunjukkan bagaimana proses-proses dalam sebuah sistem berhubungan satu dengan yang lainnya dalam sebuah kurun waktu tertentu. Sebuah STD secara umum terdiri dari atas 3 komponen, yaitu : *States, Events, Transitions*. Sebuah states menyatakan kondisi dari sistem disuatu waktu (biasanya dilambangkan dengan sebuah kotak), *transitions* dilambangkan dengan panah yang menunjukkan arah jalannya sistem, sedangkan events menjelaskan peristiwa apa yang terjadi pada perpindahan status sistem, biasanya diletakkan disebelah garis *transitions*.

Secara umum dapat digunakan 2 buah cara untuk menggambarkan STD :

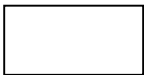

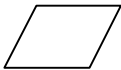
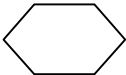




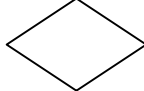


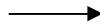
1. Penelusuran jalannya sistem awal sistem berjalan.
2. Penulusuran dari hasil akhir keadaan sistem yang ingin dicapai, lalu meningkat keadaan di atasnya sampai pada keadaan awal sistem.

2.8.7 *Diagram Alur (Flowchart)*

Pemakaian komputer dewasa ini sedemikian pesatnya sejalan denagn kemajuan teknologi komputer itu sendiri. Kebanyakan komputer digunakan untuk memproses dan

menyimpan data serta informasi. Untuk melakukan tugas-tugas tersebut komputer harus deprogram. Menurut Suryadi H.S, Agus Sumin [4] *Flowchart* adalah urutan-urutan intruksi program yang digambarkan dengan simbol-simbol diagram, simbol-simbol terdapat dilihat pada tabel 2.8 :

Tabel2.8. simbol *Flowchart*

Simbol :					
1	2	3	4	5	6
					
7	8	9	10	11	12
					
Keterangan symbol :					
1. Penugasan / perhitungan / proses					
2. Operasi secara manual / bukan dengan computer					
3. Masukan serta keluaran					
4. Persiapan / pemberian harga awal					
5. Masukan secara manual					
6. <i>Pita punched</i>					
7. Mencetak hasil					
8. Tempat penyimpanan <i>online</i>					

9. Kotak keputusan
10. *Stop / end*
11. Simbul penghubung
12. *Flow*

2.9 Personal Home Page (PHP)

Kadir, Abdul (2002:1) mengatakan PHP (secara resmi "PHP: *Hypertext Preprocessor*") merupakan bahasa *server side* HTML-embedded scripting, yaitu segala proses pemograman dilakukan di *server* sebelum dikirim ke komputer *client* sehingga komputer *client* hanya menerima keluaran dalam bentuk HTML.

PHP adalah salah satu bahasa *Server-side* yang didesain khusus untuk aplikasi *web*. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *Server side*, maka bahasa PHP akan dieksekusi di *server*, sehingga yang dikirimkan ke *browser* adalah "hasil jadi" dalam bentuk HTML, dan kode PHP tidak akan terlihat.

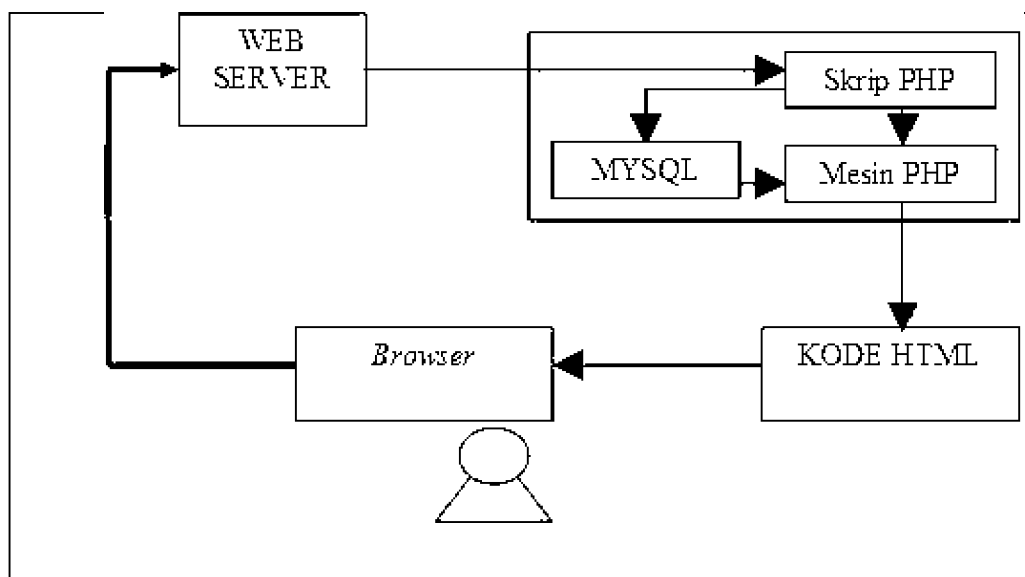
PHP dibuat pertama kali oleh seorang yang bernama Rasmus Lerdorf, pada awalnya PHP dibuat untuk menghitung jumlah pengunjung pada situsnya. Diawal Januari 2001, PHP telah dipakai lebih dari 5 juta domain diseluruh dunia, dan akan terus bertambah karena kemudahan aplikasi PHP ini dibandingkan dengan bahasa *Server side* yang lain, menurut hasil survei www.netcraft.co.uk, PHP terus meningkat penggunaannya dan telah digunakan pada jutaan domain dan jutaan alamat IP.

Menurut hasil survei *www.netcraft.co.uk*, PHP terus meningkat penggunaannya dan telah digunakan pada jutaan domain dan jutaan alamat IP, PHP telah digunakan oleh berbagai situs populer baik luar negeri maupun situs dalam negeri. Termasuk situs *benpinter.net* sendiri juga dibuat dengan PHP..

Pada kebanyakan tingkat dasar, PHP dapat melakukan apapun selain pemograman CGI yang dapat lakukan, seperti mengumpulkan data form, membangkitkan isi halaman dinamis, atau mengirim dan menerima *cookies*.

2.9.1 Konsep kerja PHP dan MYSQL

Kadir, Abdul (2002:6) Model kerja PHP diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) dikenal juga dengan sebutan alamat internet, browser mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Informasinya berupa : "nama *browser*, versinya, dan sistem operasinya.." Selanjutnya, *web server* akan memilih apakah yang diminta klien adalah kode HTML atau PHP. Apabila klien ingin menampilkan kode HTML maka web server akan melakukan proses penerjemahan kode HTML dan menampilkannya ke layar *client*, sebaliknya ketika berkas PHP yang diminta didapatkan oleh *web server*, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke *web server*. Selanjutnya, *web server* menyampaikan ke *client*. Untuk lebih jelasnya perhatikan Gambar 2.6 berikut ini.



Gambar 2.7 Konsep Kerja PHP dan MYSQL

2.10 MYSQL

Sutarman, S.Kom (2003:171) mengatakan Basis data telah menjadi bagian integral pada hampir setiap kehidupan manusia. Bank, universitas dan perpustakaan adalah tiga contoh organisasi yang tergantung kepada suatu sistem basis data. Di internet, *search engine* dan *online shopping* tidak bisa tidak harus menggunakan basis data.

Pada edisi pertamanya di tahun 2000, *Linux Journal* memberi penghargaan untuk MySQL sebagai basis data favorit pilihan pembaca. Ini bukti bahwa MySQL banyak diminati dan digunakan oleh sebagian besar pengguna sistem operasi Linux, baik untuk kebutuhan pengembangan web atau yang lainnya. MySQL adalah basis data *server* yang *multi user* dan *multithreaded* SQL tercepat. Tujuan utama MySQL dibuat adalah kecepatan, kehandalan dan mudah digunakan. MySQL pada mulanya dikembangkan

karena kebutuhan akan server SQL yang dapat menangani basis data yang sangat besar dengan kecepatan yang lebih tinggi dari yang ditawarkan oleh pembuat perangkat lunak basis data lainnya dengan hardware yang tidak mahal.

Kemampuan dan kehandalan MySQL diantaranya adalah:

1. Dapat menangani *user* secara simultan tanpa dibatasi oleh jumlah *user*.
2. Dapat menangani sampai lebih dari 50 juta record.
3. Eksekusi perintah yang sangat cepat, mungkin tercepat dari yang ada.
4. Sangat mudah dan efisien adalah sangat akurat dan cepat dalam mengatur hak setiap *user* (*user privileges*).
5. Multiplatform, bisa dipasang pada berbagai sistem operasi seperti Linux, Solaris, Sun OS, AIX, Windows 95/98/NT, FreeBSD dan lain-lain.

Namun yang paling menarik, MySQL adalah *free software*, gratis dibawah lisensi GPL. *Server* atau situs *web* yang telah memakai MySQL antara lain *Silicon Graphic* (www.sgi.com), *Siemens* (www.siemens.com), dan di Indonesia digunakan oleh *web hosting* Indoglobal dan Cakraweb.

2.10.1 Tipe data pada MySQL

MySQL memiliki tipe data yang cukup banyak yang dapat dikategorikan dalam tiga kelompok yaitu:

1. *Tipe numerik* : *TinyInt, SmallInt, MediumInt, Integer, BigInteger, Float, Double, Real, Decimal*.
2. *Tipe tanggal dan waktu* : *DateTime, Date, TimeStamp, Time, Year*.

3. *Tipe string* : *Char, Varchar, TinyBlob, TinyText, Blob, Text, MediumBlob, MediumText, LongBlob, LongText.*

2.11 Pendataan Data Alumni

Pendataan data alumni adalah sebuah form yang digunakan untuk mengisi data para alumni yang semakin lama alumni semakin banyak, sehingga penulis akan membuat suatu *web* yang berjudul “pembuatan prototipe aplikasi pendataan data alumni program studi teknik informatika di universitas mercu buana berbasis *web*”.

Adapun fungsi dari pembuatan *web* tersebut untuk mengetahui atau melihat beberapa banyak Mahasiswa/I yang telah lulus pada program studi S1 jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Mercu Buana dari awal berdirinya hingga sekarang.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

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 pendataan alumni ini harus menyediakan sarana untuk mengisi form alumni yang dapat digunakan oleh alumni. Para alumni yang belum mengisi data – data diharuskan untuk mengisi form pendataan alumni yang telah disediakan, setelah para alumni mengisi form pendataan alumni dan mengembalikan form tersebut.

Permasalahan yang dihadapi lembaga atau pihak yang bersangkutan :

1. Pembuatan form pendataan alumni yang membutuhkan biaya untuk memperbanyak form pendataan alumni.
2. Pengembalian form pendataan alumni masih membutuhkan waktu yang cukup lama, karena pihak yang bersangkutan harus menunggu alumni menyerahkan kembali.

Dari permasalahan yang dihadapi oleh lembaga atau perusahaan, dapat diselesaikan dengan membuat sebuah Aplikasi Pendataan Alumni. Dimana form pendataan alumni ini memiliki fasilitas antara lain :

1. Pembuatan form langsung pada aplikasi komputer.

Admin memasukan data – data ke aplikasi, kemudian aplikasi akan menyimpan masukan dari admin ke dalam basis data. Data-data yang telah dimasukan admin akan ditampilkan pada web pendataan alumni.

2. Merubah data alumni

Admin dapat merubah form pendataan alumni kemudian data – data yang telah dirubah admin akan disimpan kedalam basis data.

3.2 Perancangan Sistem

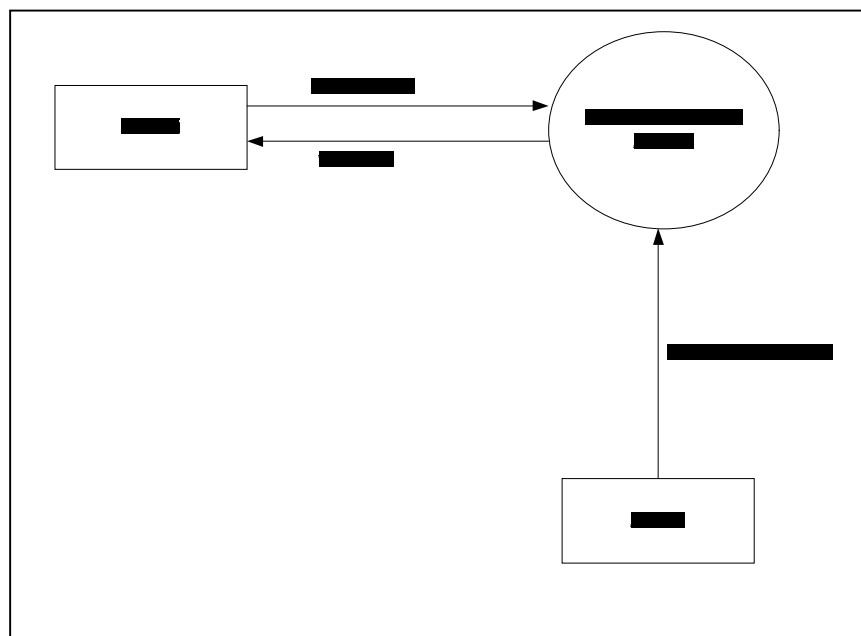
Berdasarkan hasil analisis, penulis akan membuat sebuah aplikasi pendataan alumni dengan menggunakan php dan mysql. Secara garis besar pendataan alumni ini memiliki fasilitas yang telah dijelaskan pada sub bab analisis tepatnya penjelasan pemecahan masalah.

3.2.1 Data Flow Diagram (DFD)

DFD pada "Aplikasi pendataan alumni" memiliki beberapa tingkat yaitu diagram konteks, nol, dan rinci. Diagram tingkat rinci akan menjelaskan semua proses yang terdapat pada diagram tingkat nol.

3.2.1.1 Rancangan Diagram Konteks

Diagram tingkat konteks yang terdapat pada aplikasi pendataan alumni memiliki 2 entitas yaitu: alumni dan admin. Diagram tingkat konteks ini memiliki satu proses yaitu proses aplikasi pendataan alumni. 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 pendataan alumni.



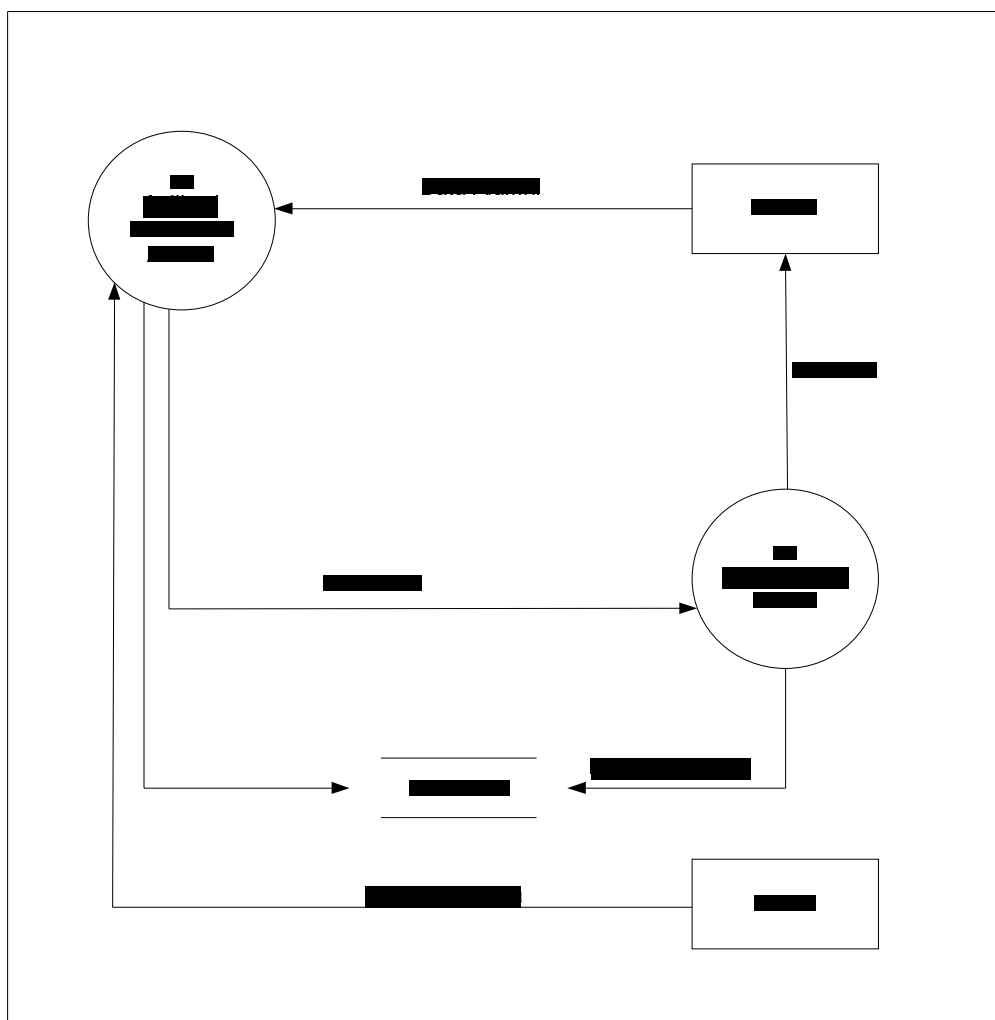
Gambar 3.1 Diagram Konteks.

Dalam gambar diagram konteks diatas memberikan gambaran bahwa sistem pendataan alumni ini berinteraksi dengan dua terminator yaitu : Alumni, dan Admin. Tanda panah diatas menyatakan masukan dan juga keluaran dari sistem tersebut. Sebagai contoh alumni mengisi data – data nya, kemudian alumni dapat melihat hasil data yang diperolehnya. Admin selain dapat melihat data alumni, admin memiliki peranan yang sangat penting yaitu memasukan data-data yang dibutuhkan oleh aplikasi tersebut.

3.2.1.2 Diagram Level Satu

Diagram tingkat satu pada pendataan alumni merupakan penjabaran dari diagram konteks. Pada diagram konteks terdapat satu proses yaitu proses aplikasi pendataan alumni. Sedangkan pada diagram level satu terdapat dua proses subsistem dan satu proses pembuatan laporan untuk lebih jelasnya perhatikan Gambar 3.2. Dua proses subsistem ini memiliki proses yang berbeda. Alumni akan memasukan datanya,

kemudian data alumni akan dimasukkan ke dalam basis data.



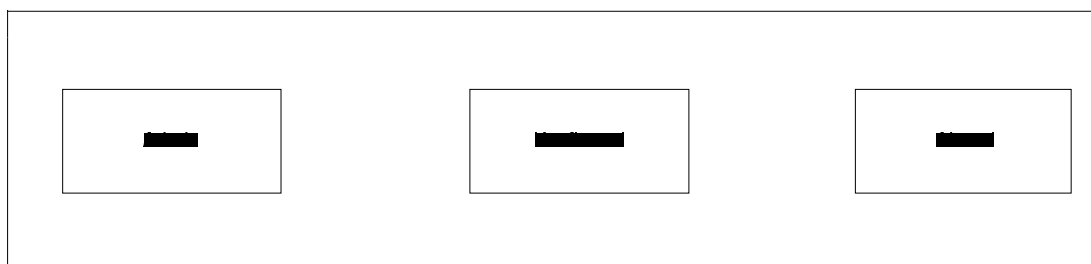
Gambar 3.2 Diagram tingkat satu.

Pada gambar 3.2 menggambarkan dua proses subsistem, masing-masingnya memiliki proses-proses yang lain. alumni masuk ke halaman form alumni, alumni akan mengisi datanya, kemudian alumni akan, dari proses tersebut akan menghasilkan hasil pendataan alumni dimana hasil input data akan dimasukkan ke basis data .

Gambar 3.2 memperlihatkan laporan data alumni, admin dapat mengontrol aplikasi pendataan alumni dan admin dapat mengatur basis data, Data alumni akan digunakan untuk proses penyimpanan data.

3.2.2 *Entity Relationship Diagram (ERD)*

Pada perancangan DFD tidak ada entitas yang saling berhubungan dengan entitas yang lain, pada Gambar 3.3 menggambarkan diagram tidak ada hubungan antar entitas (*Entity Relationship Diagram/ERD*). Digambarkan antara admin dengan data pendataan alumni, admin mengelolah data alumni, mengedit data alumni. Hanya admin yang dapat melakukan semua itu. Admin dapat melakukan hal itu kepada semua data pendataan alumni yang ada.



Gambar 3.3 Admin, Alumni dan Konfirmasi

Berikutnya akan dijabarkan yang terjadi pada beberapa entitas, seperti yang telah diketahui bahwa tidak ada beberapa entitas tidak saling berhubungan, antara lain entitas admin, alumni dan konfirmasi.

Berdasarkan ERD diatas dapat dikembangkan menjadi model data, model data fisik tersebut dapat dilihat pada Gambar 3.4



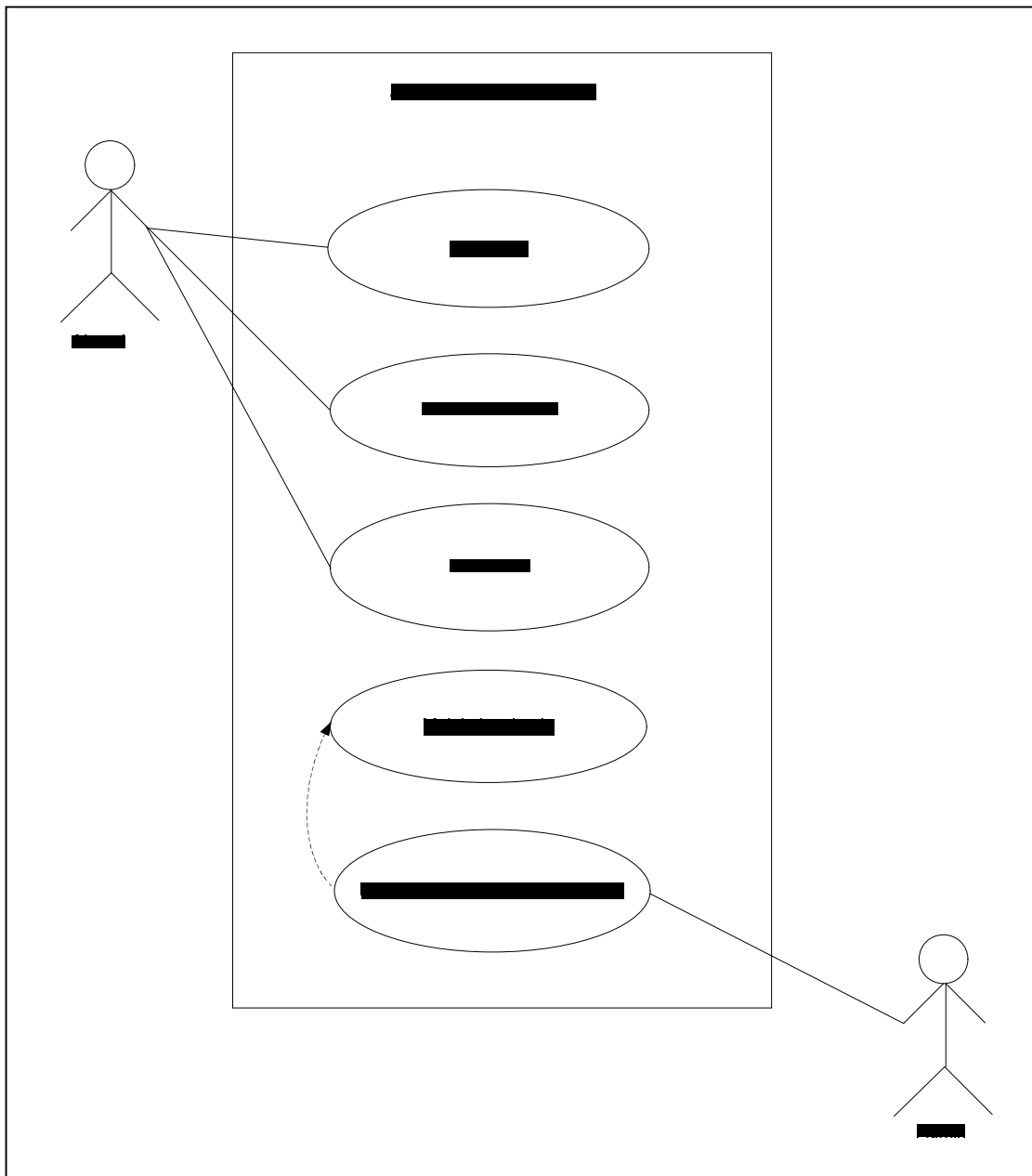
Gambar 3.4 Model data fisik aplikasi Pendaftaran Alumni.

3.2.3 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.

3.2.3.1 Diagram Use Case

Gambar 3.5 menunjukkan diagram use case aplikasi pendaftaran alumni dengan menggunakan php dan mysql.



Gambar 3.5 Diagram use case Aplikasi Pendataan Alumni

A. Sisi Alumni

Berikut ini akan menjelaskan Gambar 3.5 dari sisi diagram use case yang dilakukan Alumni :

1. Use case melakukan Input data

Nama : Input data

Aktor : Alumni

Deskripsi : Alumni akan masuk kehalaman Pendataan Alumni.

Kecuali : -

Proses :

- a. Alumni akan menghadapi form Pendataan Alumni.
- b. sistem akan menampilkan form pendataan alumni yang terdapat didalam basis data.
- c. sistem akan mengecek data alumni dengan benar didalam basis data.

2. Use case melakukan Lihat data alumni

Nama : Data Alumni

Aktor : Alumni

Deskripsi : Alumni dapat melihat data pada saat data tersebut di simpan di sistem, sistem akan mengatur aktifitas yang dilakukan alumni pada saat Lihat data.

Kecuali : Alumni yang tidak mengisi data dan di simpan maka tidak dapat melihat data.

Proses :

- a. Alumni memilih link lihat data.
- b. sistem akan menampilkan halaman Lihat data.
- c. sistem akan mengecek data alumni.

d. sistem akan menampilkan halaman lihat data.

Kesalahan Alumni :

1. pada langkah d, sistem akan lihat data dari alumni, jika alumni tidak mengisi data maka tidak terdaftar di dalam basis data, alumni akan kembali kelangkah d.

3. Use case Konfirmasi

Nama : Konfirmasi

Aktor : Alumni

Deskripsi : Alumni dapat melakukan konfirmasi data pada saat data tersebut ada kesalahan dan di simpan di sistem, sistem akan mengatur aktifitas yang dilakukan alumni pada saat konfirmasi.

Kecuali : Alumni yang tidak mengisi data dan di simpan maka tidak dapat konfirmasi.

Proses :

- a. Alumni memilih link konfirmasi.
- b. sistem akan menampilkan halaman konfirmasi.
- c. sistem akan mengecek konfirmasi.
- d. sistem akan menampilkan halaman lihat data.

Kesalahan Alumni :

2. pada langkah d, sistem akan mengkonfirmasi ke admin, jika alumni tidak mengisi form konfirmasi maka tidak terdaftar didalam basis data, alumni akan kembali kelangkah d.

B. Sisi Admin

Berikut ini akan menjelaskan Gambar 3.5 dari sisi diagram use case yang dilakukan admin:

1. Use case melakukan login

Nama : Login

Aktor : Admin

Deskripsi : Admin harus melakukan login pada saat menggunakan sistem, sistem akan mengatur aktifitas yang dilakukan Admin pada saat login.

Kecuali : Admin yang tidak melakukan login tidak dapat menggunakan sistem.

Proses :

- 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 pendataan alumni.

Kesalahan Alumni :

3. pada langkah d, sistem akan mengecek username dan password dari alumni, jika username dan password alumni tidak terdaftar didalam basis data, alumni akan kembali kelangkah d.

2. Use case Mengelolah pendataan alumni

Nama : admin

Aktor : admin

Deskripsi : admin memilih link data alumni untuk dapat mengatur data alumni.

Kecuali : -

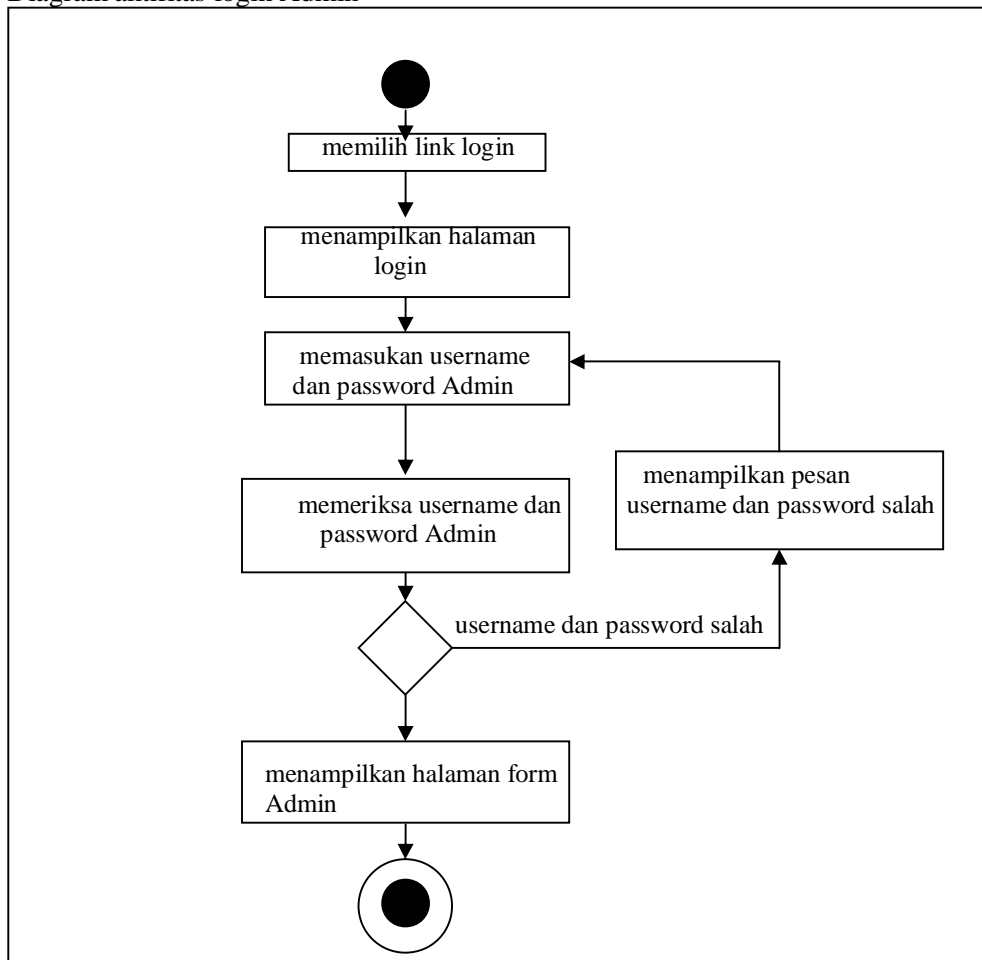
Proses :

- a. admin memilih link data alumni
- b. sistem menampilkan data alumni
- c. admin dapat memilih link data alumni.
- d. sistem menampilkan halaman yang dipilih oleh admin.

3.2.3.2 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 Login Admin.

a. Diagram aktifitas login Admin

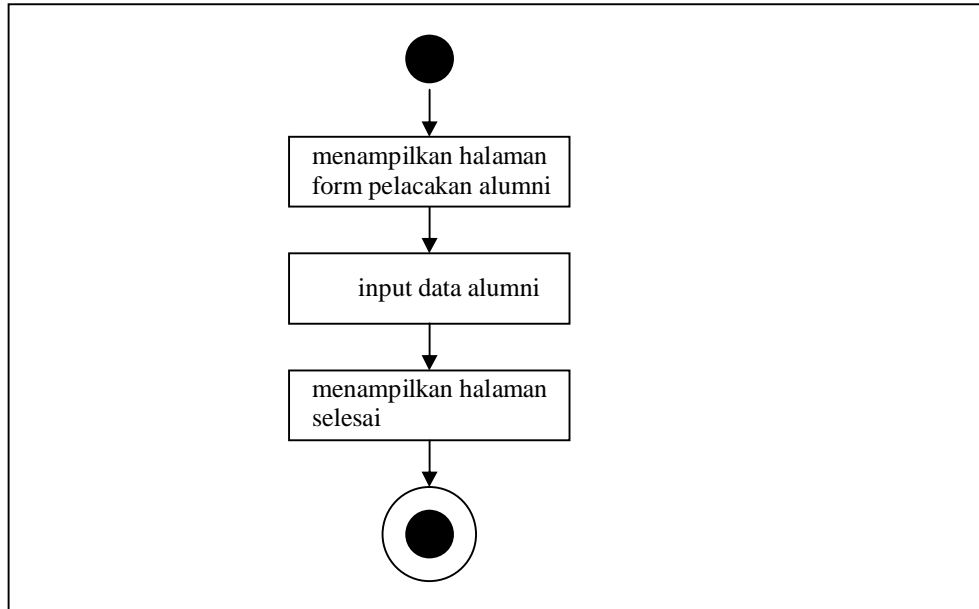


Gambar 3.6 Diagram aktifitas login Admin.

Pada Gambar 3.6 diagram *login* dimulai dari Admin memilih *link login* dan sistem menampilkan halaman *login*, kemudian Admin diminta untuk mengisi *username* dan *password* Admin, kemudian sistem akan memeriksa *username* dan *password* Admin. Jika salah Admin akan diminta oleh sistem untuk kembali mengisi *username* dan *password* dengan benar. Setelah *username* dan *password* benar maka sistem akan menampilkan halaman form Admin.

b. Diagram aktifitas input data

Gambar 3.7 menunjukan diagram aktifitas input data

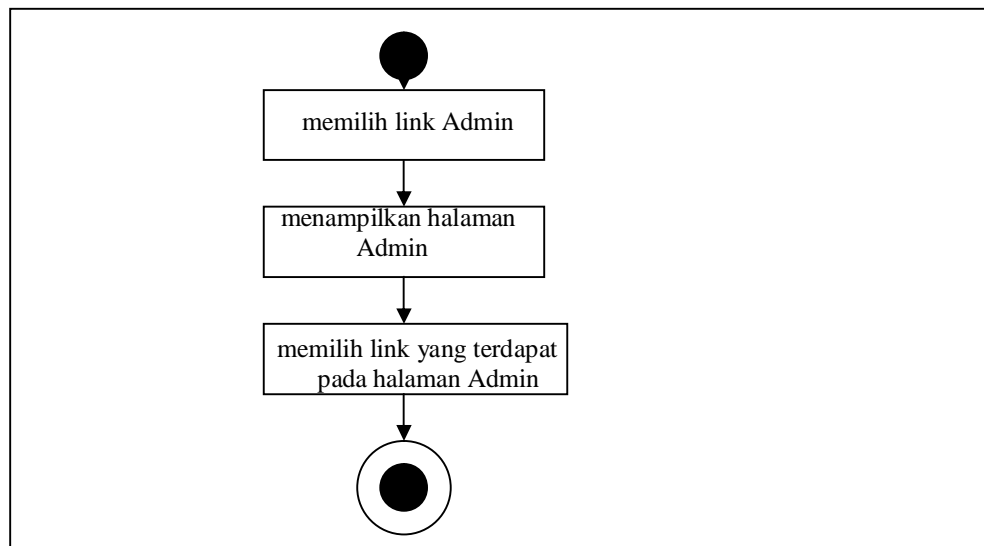


Gambar 3.7 Diagram aktifitas input data.

Setelah alumni login, sistem akan menampilkan halaman form pelacakan alumni, kemudian alumni akan mengisi form pelacakan alumni dengan lengkap, setelah alumni selesai mengisi form pendataan alumni, sistem akan menampilkan selesai.

c. Diagram aktifitas admin

Pada Gambar 3.8 menunjukan diagram aktifitas admin

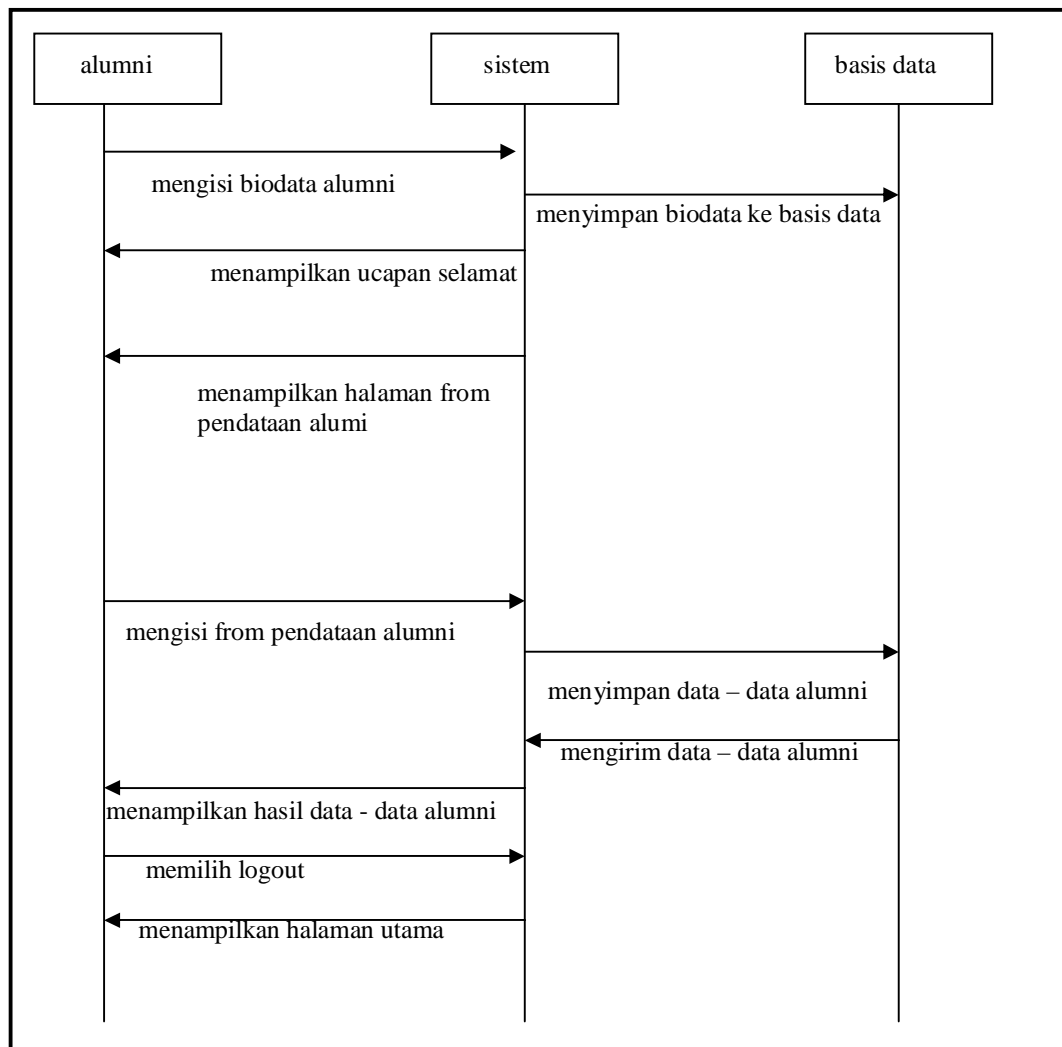


Gambar 3.8 Diagram aktifitas admin.

Admin memilih link alumni, dan sistem akan menampilkan halaman data alumni, pada halaman data alumni terdapat beberapa *link* yang dapat dipilih oleh admin, diantaranya: lihat data alumni.

3.2.3.3 Diagram Sequence

Gambar 3.10 menunjukkan diagram sequence dari aplikasi pendataan alumni



Gambar 3.9 Diagram sequence untuk aplikasi pendataan alumni.

Sistem akan menyimpan biodata alumni ke dalam basis data, setelah biodata tersimpan sistem akan menampilkan ucapan selamat.

Setelah biodata alumni tersimpan, alumni mengisi form pendataan alumni, dan sistem akan menyimpan dan memeriksa data – data alumni didalam basis data,

kemudian basis data akan mengirim data – data alumni ke sistem, dan sistem akan menampilkan hasil data – data alumni.

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

3.3 Struktur Tabel

Secara detail struktur tabel yang akan digunakan pada aplikasi pendataan alumni adalah sebagai berikut:

1. Tabel Admin

Tabel ini nantinya akan digunakan oleh administrator untuk menyimpan data *username* dan *password* yang akan digunakan untuk proses *login*.

Tabel 3.1 Tabel admin

Nama field	Type	Length	Key	Keterangan
<u>Id</u>	Integer	11	Primary key	Id
Username	Varchar	30		Username
Paswword	varchar	10		Paswword

2. Tabel pendataan Alumni

Tabel pendataan alumni, tabel yang digunakan oleh admin untuk memasukan form pelacakan alumni.

Tabel 3.2 Tabel pendataan alumni

Nama field	Type	Length	Key	Keterangan
<u>Id</u>	Integer	11	Primary key	Nim alumni
Nama	Varchar	30		Nama alumni
Nim	Varchar	15		Lulus alumni
Angkatan	Tinyint	4		
Lulus	Date			Tgl, bln, tahun lulus
Jkel	char	1		P=perempuan L=laki-laki
Email	Varchar	20		
Foto	Varchar	50		Foto alumni/path fotoalumni
Nama_instansi	Varchar	40		
Alamat_instansi	Varchar	100		
Mulai_Bekerja	Tinyint	4		
Jabatan	Varchar	15		
Status	Smallint			1 = ya 0 = tdk alumni

Pada rancangan tabel diatas, terdapat kolom yang menggunakan tipe data *tinyint* dan *smallint* karena lebih menghemat memori karena jumlah bit lebih kecil dari pada tipe data *integer*. Dengan alasan tersebut tipe data *tinyint* dan *smallint* sesuai digunakan dalam aplikasi ini.

Sedangkan tipe data *varchar* digunakan pada beberapa kolom, seperti Id , nama, email, nim, nama, nama_instansi, alamat_instansi, jabatan. Karena tipe data *varchar* memiliki penyimpanan memori bergantung pada berapa karakter yang dipakai, maka akan lebih menghemat penggunaan memori.

3. Tabel Konfirmasi

Tabel konfirmasi, tabel yang digunakan oleh admin untuk melihat konfirmasi dari alumni pendataan alumni.

Tabel 3.3 Tabel Konfirmasi

Nama field	Type	Length	Key	Keterangan
Konfirmasi <u>Id</u>	integer	10	Primary key	Konfirmasi <u>Id</u>
Nama	Varchar	30		Nama alumni
Nim	Varchar	15		Lulus alumni
Konfirmasi	Text			

Pada rancangan table diatas, terdapat kolom yang menggunakan tipe data *varchar* digunakan pada beberapa kolom, seperti , nim, nama. Karena tipe data *varchar* memiliki penyimpanan memori bergantung pada berapa karakter yang dipakai, maka akan lebih menghemat penggunaan memori. Sedangkan tipe data *text* digunakan pada kolom konfirmasi, karena tipe data *text* memiliki lebih banyak karakter.

3.4 Perancangan AntarMuka (*Interface*)

Perancangan antarmuka mendeskripsikan bagaimana suatu perangkat lunak dapat berkomunikasi dengan dirinya sendiri dan sistem yang berinteroperasi dengannya, serta dengan para *user*. Untuk merancang suatu antarmuka diperlukan data-data dan diagram - diagram sebagai informasi aliran data dan perilaku spesifik dari sistem.

Pada aplikasi pendataan alumni ini perancangan terbagi menjadi dua bagian, yaitu pada bagian administrator yang dapat mengelola aplikasi (cek data, tambah data, Data Alumni, Konfirmasi dan Logout) data-data yang tersimpan pada basis data.

3.4.1 Rancangan AntarMuka Menu Utama Pendataan Alumni

Rancangan antarmuka yang diberikan disini hanya rancangan antarmuka halaman utama menu pendataan alumni.



Gambar 3.10 Rancangan AntarMuka Menu Utama Pendataan Alumni.

Gambar 3.10 merupakan rancangan antarmuka halaman utama menu pendataan alumni. Terdapat empat buah *link* menu yang dapat alumni akses, yaitu menu input data yang akan menampilkan *link* dari halaman input data, menu data alumni yang akan menampilkan *link* dari halaman informasi data alumni, menu konfirmasi yang akan menampilkan *link* dari halaman form konfirmasi, dan menu Login admin yang akan menampilkan *link* dari halaman form login Admin. Dimana pada halaman Admin ini dapat melihat data-data alumni yang sudah masuk.

3.4.2 Rancangan Antarmuka menu input data

Rancangan antarmuka yang di berikan disini hanya rancangan antarmuka halaman input data.

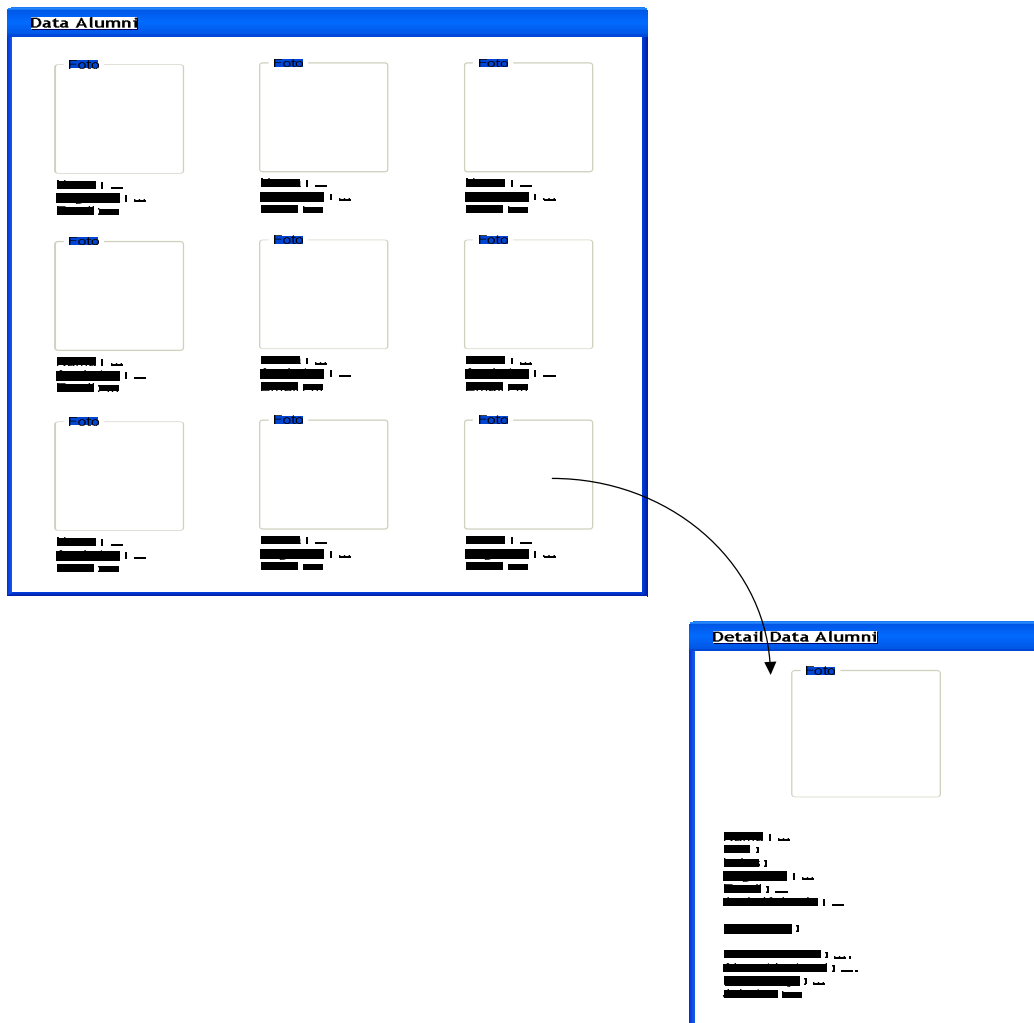
The image shows a web form titled "Input Data" with a blue header. Inside the form, there are two main sections. The first section, "Form Pengisian Data Alumni", contains several input fields: a text box, another text box, a dropdown menu, three dropdown menus, a text box, another dropdown menu, and a text box with a button next to it. The second section, "Pekerjaan", contains a text box, a list box with scrollbars and arrow buttons, a dropdown menu, and another text box. At the bottom right of the form, there are two buttons.

Gambar 3.11 Rancangan menu Input Data.

Gambar 3.11 ini merupakan menu input data yang di isi oleh alumni, data tersebut harus di isi semua lalu disimpan.

3.4.3 Rancangan Antarmuka menu Data Alumni

Rancangan antarmuka yang di berikan disini hanya rancangan antarmuka halaman data alumni.

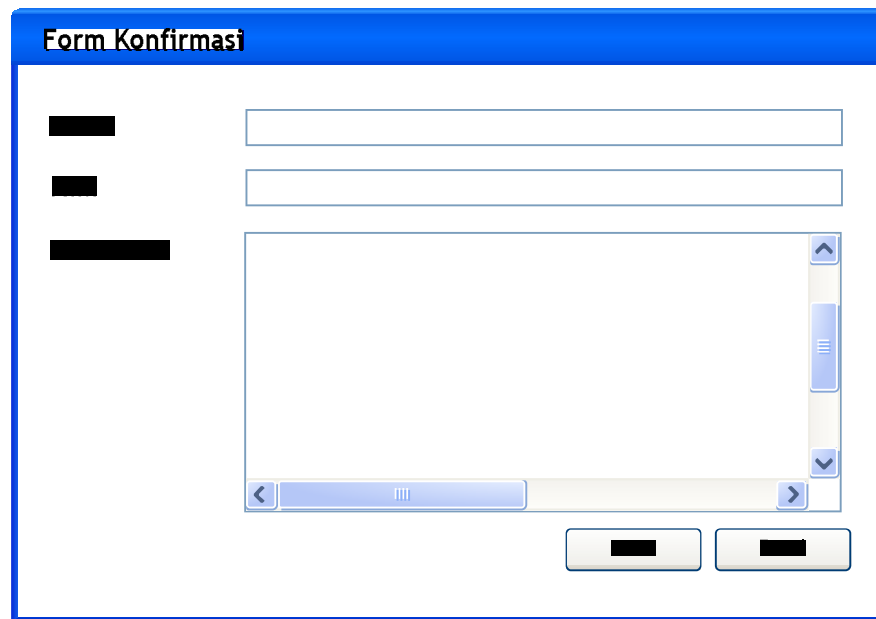


Gambar 3.12 Rancangan Menu Data Alumni.

Gambar 3.12 ini merupakan data alumni yang sudah di isi oleh alumni, sehingga admin lebih jelas siapa saja alumni yang sudah mengisi form pendataan alumni tersebut.

3.4.4 Rancangan Antarmuka menu Konfirmasi

Rancangan antarmuka yang di berikan disini hanya rancangan antarmuka halaman konfirmasi kepada admin.

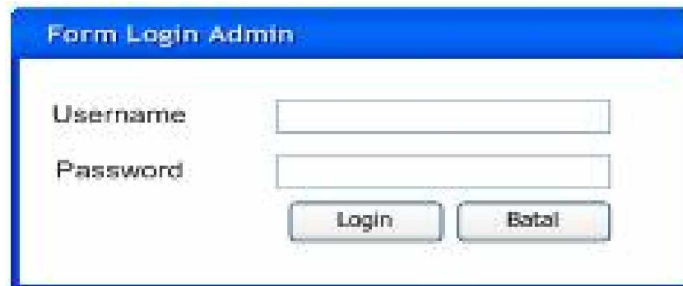


Gambar 3.13 Rancangan Antarmuka Menu Konfirmasi.

Gambar 3.13 ini merupakan menu konfirmasi apabila ada kesalahan redaksi di aplikasi, menu ini berfungsi untuk mengkonfirmasi data-data yang benar ke admin.

3.4.5 Rancangan Antarmuka Bagian Administrator

Rancangan antarmuka yang diberikan di sini hanya rancangan antarmuka bagian *login* dan rancangan antarmuka bagian menu utama administrator saja. Untuk antarmuka halaman-halaman yang lain akan dijelaskan lebih detail pada bab implementasi dan pengujian.

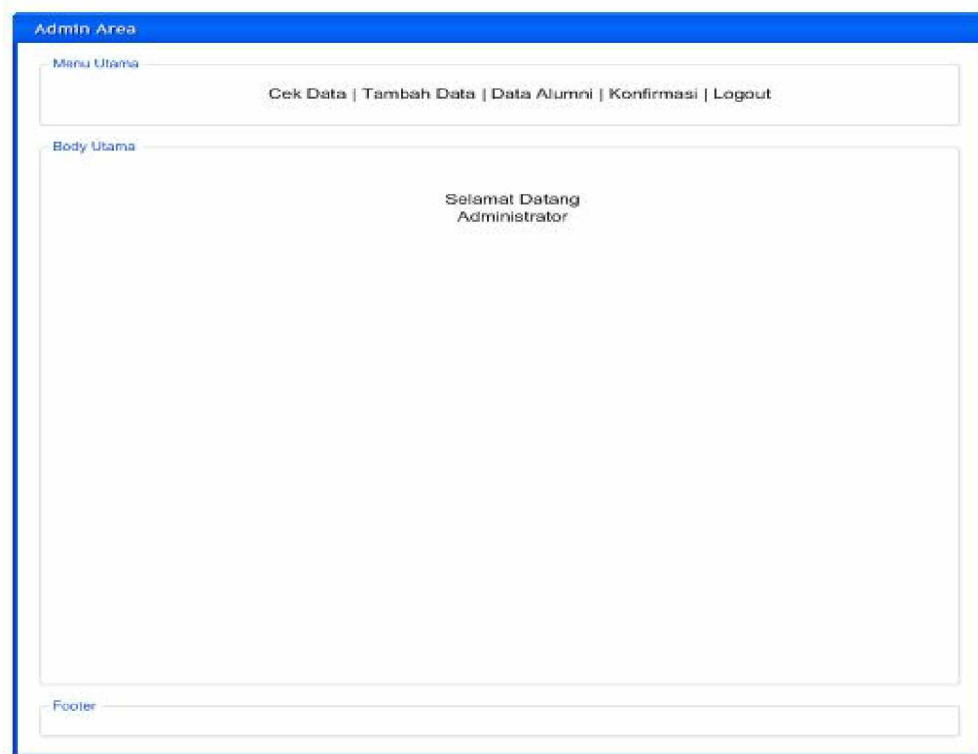


Gambar 3.14 Rancangan Antarmuka halaman login.

Pada gambar 3.14 merupakan rancangan antarmuka halaman *login* dan menu Admin. Dimana ketika pertama administrator mengakses aplikasi pendataan alumni maka akan tampil halaman *login* ini. Pada halaman *login* administrator harus memasukkan *username* dan *password* yang tersimpan pada basis data untuk dapat mengakses ke halaman utama menu administrator.

3.4.6 Rancangan Antarmuka setelah Admin berhasil login

Rancangan antarmuka yang diberikan disini hanya rancangan antarmuka halaman utama menu Admin.

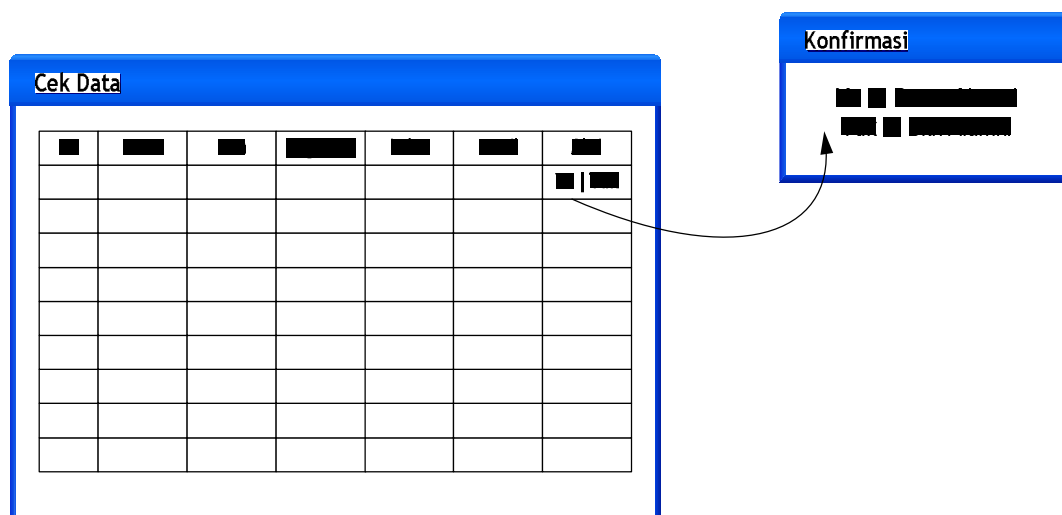


Gambar 3.15 Rancangan AntarMuka Setelah Admin Login.

Gambar 3.15 merupakan rancangan antarmuka halaman setelah admin login. Terdapat lima buah *link* menu yang dapat admin akses, yaitu menu cek data yang akan menampilkan *link* dari halaman cek data, menu tambah data yang akan menampilkan *link* dari halaman tambah data, menu data alumni yang akan menampilkan *link* dari halaman data alumni, menu konfirmasi yang akan menampilkan *link* dari halaman konfirmasi admin yang diberikan oleh alumni, dan menu logout admin akan keluar dari aplikasi pendataan alumni.

3.4.7 Rancangan Antarmuka Cek Data

Rancangan antarmuka yang diberikan disini hanya rancangan antarmuka halaman menu cek data.



Gambar 3.16 Rancangan Antarmuka Cek Data.

Gambar 3.16 Rancangan antarmuka cek data ini merupakan halaman dimana admin dapat mengecek apakah benar bahwa nama, nim, angkatan, lulus, email, termasuk alumni jurusan teknik informatika, jika Ya → Benar alumni dan jika Tdk → Bukan alumni.

3.4.8 Rancangan Antarmuka Tambah Data

Rancangan antarmuka yang diberikan disini hanya rancangan antarmuka halaman menu tambah data.

Input Data

Form Pengisian Data Alumni

Nama

NIM

Angkatan

Lulus Tgl Bulan Tahun

Email

Jenis Kelamin

Upload Foto

Pekerjaan

Nama Instansi

Alamat Instansi

Mulai Bekerja

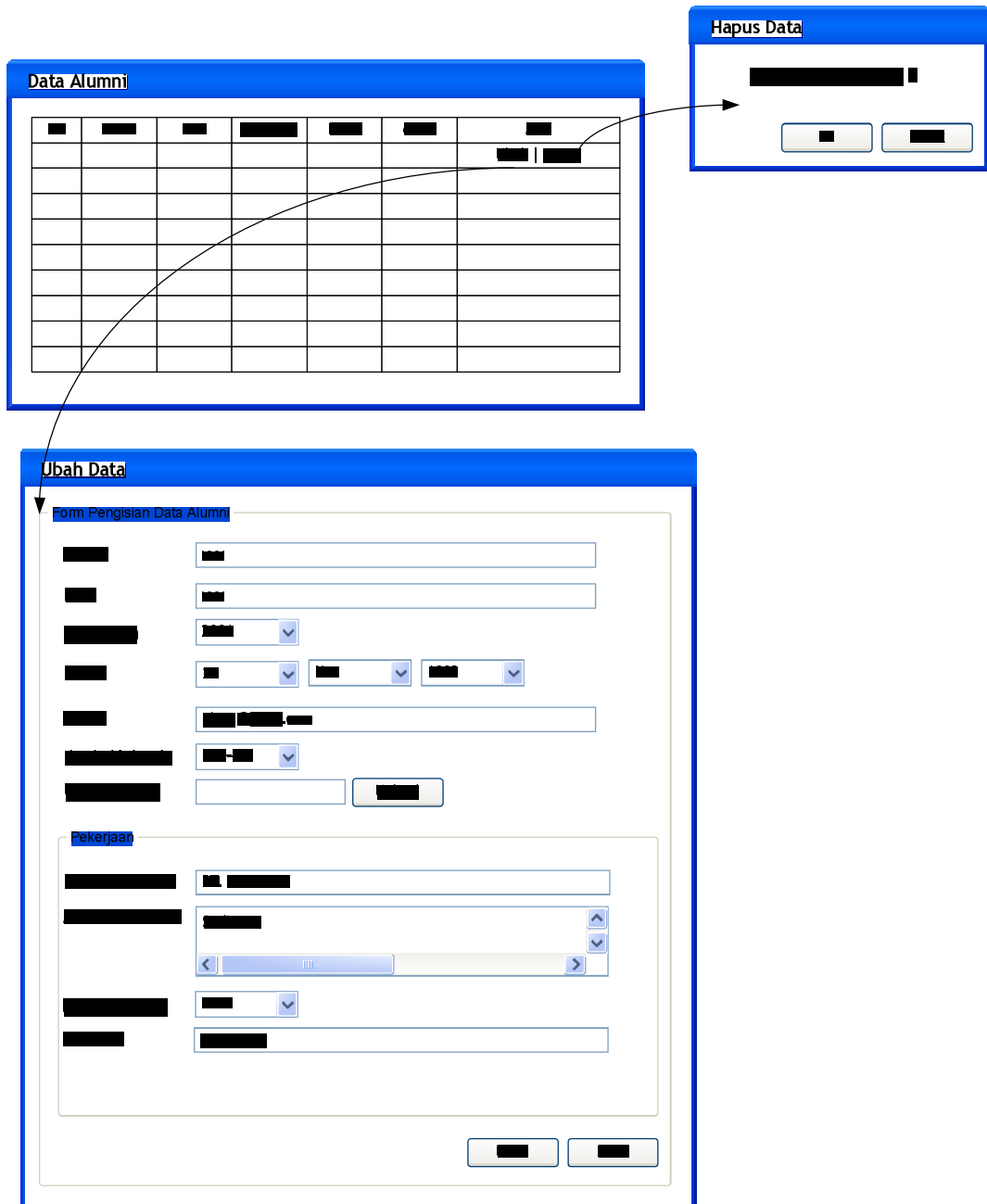
Jabatan

Gambar 3.17 Rancangan Antarmuka Tambah Data.

Gambar 3.17 Rancangan antarmuka tambah data ini merupakan tambah data apabila alumni, dimana alumni dapat menambah data – data nya yang tidak lengkap.

3.4.9 Rancangan Antarmuka Data Alumni

Rancangan antarmuka yang diberikan disini hanya rancangan antarmuka halaman menu data alumni.

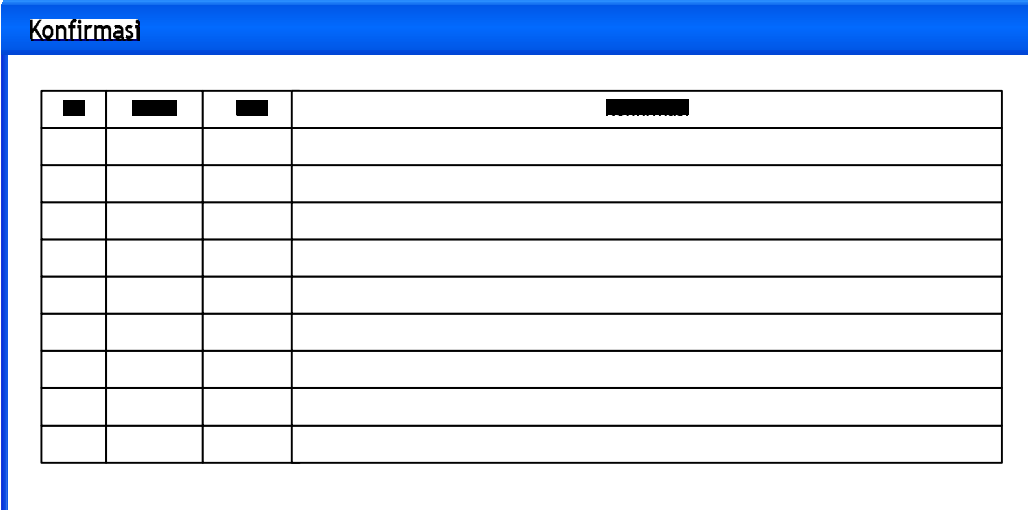


Gambar 3.18 Rancangan Antarmuka Data Alumni.

Gambar 3.18 Rancangan antarmuka data alumni ini admin dapat melihat siapa - siapa alumni yang sudah mengisi form tersebut, sehingga kalau terjadi perubahan pada data alumni, admin merubah data alumni yang di konfirmasikan oleh alumni.

3.4.10 Rancangan AntarMuka Konfirmasi

Rancangan antarmuka yang diberikan disini hanya rancangan antarmuka halaman menu Konfirmasi.



■	■	■	■

Gambar 3.19 Rancangan AntarMuka Konfirmasi.

Gambar 3.19 ini merupakan menu konfirmasi apabila ada kesalahan redaksi di aplikasi, menu ini berfungsi untuk mengkonfirmasikan data-data yang benar ke admin.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

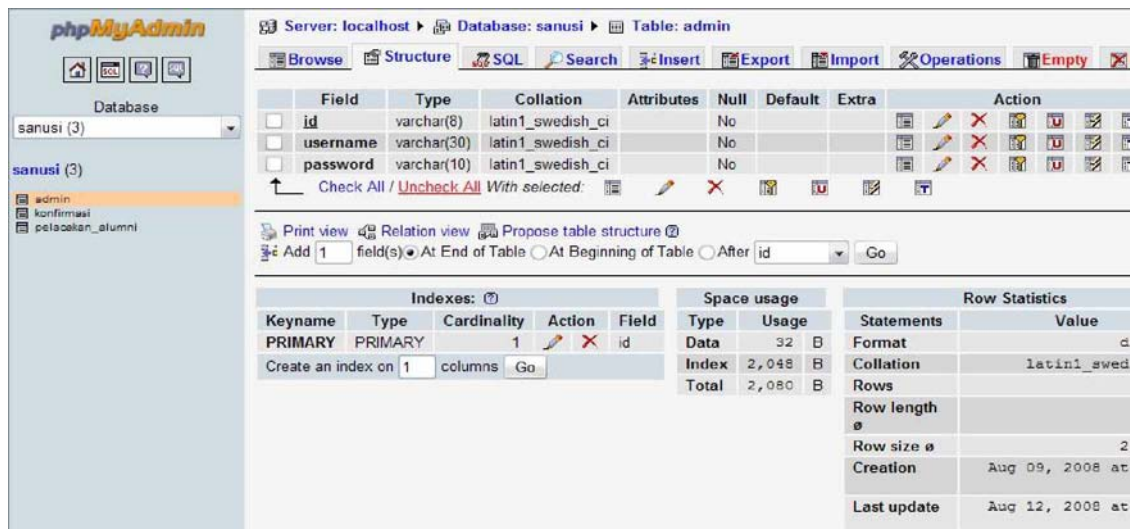
Setelah tahap analisis dan perancangan pada bab III, maka tahap selanjutnya yaitu tahap implementasi. Pada bab ini akan dibahas mengenai implementasi pembuatan basis data dan implementasi perancangan layar.

4.1.1 Implementasi Pembuatan Basis Data

Pada implementasi pembuatan basis data ini perangkat lunak yang digunakan yaitu xampp. Pada implementasi ini hanya menampilkan basis data dari tabel-tabel yang fungsional saja, seperti: tb_admin, tb_pelacakan alumni, tb_konfirmasi dan. Masing-masing tabel tersebut telah dijelaskan pada tahap perancangan pada bab III.

4.1.1.1 Implementasi Pembuatan Basis Data Tb_admin

Tb_admin terdiri dari dua kolom data, kolom pertama berisi data *username* dengan lebar maksimum 30 karakter, kolom kedua berisi data *password* dengan lebar maksimum 10 karakter. Gambar 4.1 merupakan implementasi basis data tb_admin.

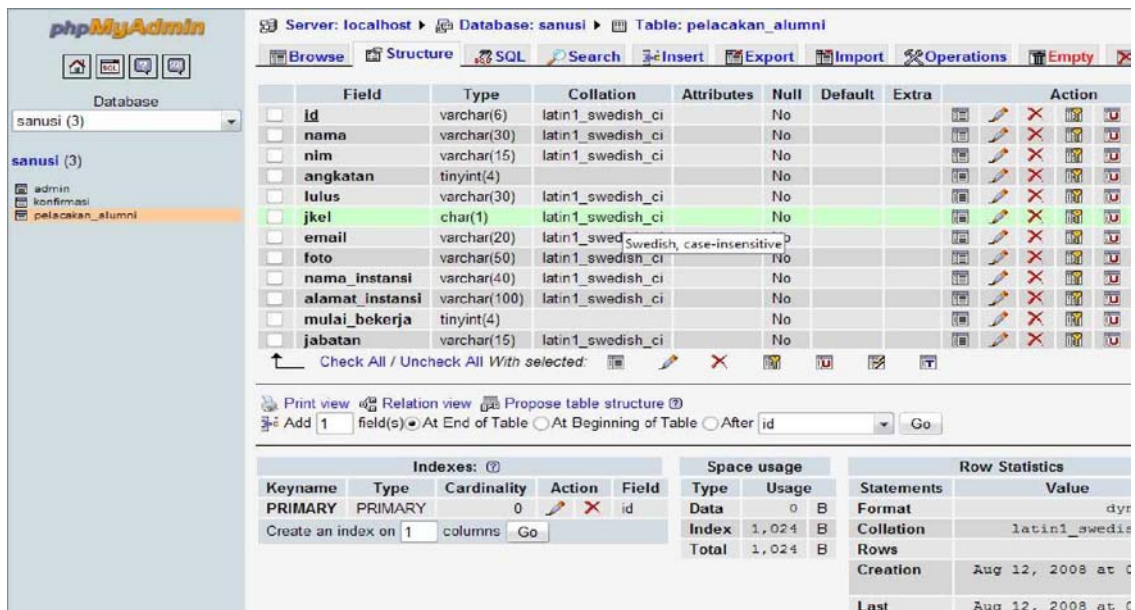


Gambar 4.1 Implementasi Pembuatan Basis Data Tabel Admin.

4.1.1.2 Implementasi Pembuatan Basis Data Tb_Pendataan Alumni

Tb_pendataan alumni terdiri dari sebelas kolom data, kolom pertama berisi nama dengan lebar maksimum 6 karakter, kolom kedua berisi nim dengan lebar maksimum 30 karakter, kolom ketiga berisi angkatan dengan lebar maksimum 4 karakter, kolom keempat berisi lulus, kolom kelima berisi jkel dengan lebar maksimum 1 karakter, kolom keenam berisi email dengan lebar maksimum 20 karakter, kolom ketujuh berisi foto dengan lebar maksimum 50 karakter, kolom kedelapan berisi nama_instansi dengan lebar maksimum 40 karakter, kolom kesembilan berisi alamat_instansi dengan lebar maksimum 100 karakter,

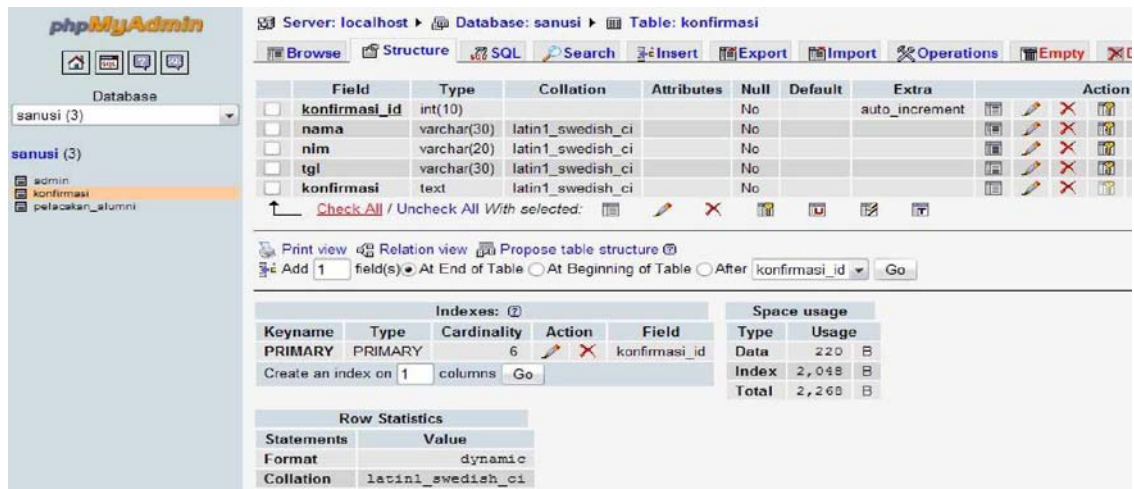
kolom kesepuluh berisi mulai_bekerja dengan lebar maksimum 4 karakter, dan kolom kesebelas berisi jabatan dengan lebar maksimum 15 karakter . Gambar 4.2 merupakan implementasi basis data tb_pendataan alumni.



Gambar 4.2 Implementasi Pembuatan Basis Data Tabel Pendataan Alumni

4.1.1.3 Implementasi pembuatan basis data Tb_Konfirmasi

Tb_konfirmasi alumni terdiri dari lima kolom data, kolom pertama berisi konfirmasi Id nama dengan lebar maksimum 10 karakter, kolom kedua berisi nama dengan lebar maksimum 30 karekter, kolom ketiga berisi nim dengan lebar maksimum 20 digit, kolom keempat berisi tgl dengan lebar maksimal 30 karakter , kolom kelima berisi konfirmasi. Gambar 4.3 merupakan implementasi basis data tb_konfirmasi.



Gambar 4.3 Implementasi Pembuatan Basis Data Tabel konfirmasi

4.1.2 Implementasi Perancangan Layar Menu Utama

Pada implementasi perancangan layar akan menampilkan rancangan layar dan potongan program, seperti: home, menu input data, menu data alumni, konfirmasi, menu login admin. Masing-masing perancangan layar tersebut telah dijelaskan pada bab III.



Gambar 4.4 Perancangan Layar Menu Utama

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>
  <head>
    <title>Aplikasi Pendataan Data Alumni</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-15" />
    <link rel="stylesheet" href="styles.css" />
```

```

</head>
<body>
<div id="conteneur">
  <div id="header">Aplikasi Pendataan Data Alumni <br>Program Studi Teknik
Informatika Univ. Mercu Buana</div>
  <?include "menu.html"?>
  <div id="centre">
    <?
      $mode = $_GET['mode'];

      switch($mode){
        case 'index' :
          include "index.php";
          break;
        case 'input' :
          include "input.php";
          break;
        case 'alumni' :
          include "alumni.php";
          break;
        case 'konfirmasi' :
          include "konfirmasi.php";
          break;
        case 'login' :
          include "login.php";
          break;
        default : include "awal.php";
      }
    ?>
  </div>
  <div id="pied">Sanusi | 01501-085</div>
</div>
</body>
</html>

```

4.1.2.1 Implementasi Perancangan Layar Input Data

Implementasi perancangan layar input data akan menampilkan halaman input data yang berisi form biodata untuk para alumni. Gambar 4.4 merupakan implementasi perancangan layar input data.

Gambar 4.5 Perancangan Layar Input Data

```

<?php
include "koneksi.php";
@session_start();
if(!empty($_SESSION['namauser']) and !empty($_SESSION['passuser']))
{
include "menu_admin.html";
}
else
{
//include "menu.html";
}

if ($_POST['simpan'])
{
$nim = $_POST['nim'];
$nama = $_POST['nama'];
$angkatan = $_POST['angkatan'];
$jeniskelamin = $_POST['jeniskelamin'];
$email = $_POST['email'];
$instansi = $_POST['instansi'];
$alamatinstansi = $_POST['alamatinstansi'];
$tgl_kerja = $_POST['tgl_kerja'];
$jabatan = $_POST['jabatan'];

if ($_FILES['upfile']['error'] == 0) {
$_IMG_DIR = "./image/foto/";
}
}

```

```

$_FILE_SIZE = 80000;

$sqlNextVal = " SELECT AUTO_INCREMENT AS last_id
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = '$dbName.'" AND TABLE_NAME='pendataan_alumni' ";

$rsNextval = mysql_query($sqlNextVal);
$data=mysql_fetch_array($rsNextval);
$id_next= $data['last_id'];

$type_file = $_FILES["upfile"]["type"];

if ($type_file == "image/gif"){
    $nama_file = $id_next.".gif";
} else if ($type_file == "image/jpeg") {
    $nama_file = $id_next.".jpg";
}

$error_file = $_FILES["upfile"]["error"];
$size_file = $_FILES["upfile"]["size"];
$temp_file = $_FILES["upfile"]["tmp_name"];
$path_file = $_IMG_DIR;
$file_image = $path_file.$nama_file;

$path_file_best = $_IMG_DIR_BEST;
$file_image_best = $path_file_best."not_available.gif";

if (($type_file == "image/gif") || ($type_file == "image/jpeg") && ($size_file < $_FILE_SIZE))
{
    if ($error_file > 0)
    {
        die ("Return Code: " . $error_file . "<br />");
    }
    else
    {
        if (file_exists($path_file . $nama_file))
        {
            die ($nama_file . " already exists. ");
        }
        else
        {
            move_uploaded_file($_FILES["upfile"]["tmp_name"],
            $path_file . $nama_file);
        }
    }
}
else
{
    if ($type_file == ""){
        //Image Kosong
        $file_image = $path_file."no_picture.gif";
    }else {
        print "

```

```

        <script>
            alert ('Invalid File, Only JPG or GIF or size image to big , max 75 Kb ');
            history.back(-1);
        </script>
        ";
        exit;
    }
}

$lulus="$_POST[tgl]-$_POST[bulan]-$_POST[tahun]";

$perintah="insert into pendataan_alumni
(nim,nama,angkatan,lulus,jkel,email,foto,nama_instansi,alamat_instansi,mulai_bekerja,jabatan) values
('$nim','$nama','$angkatan','$lulus','$jeniskelamin','$email','$file_image','$instansi','$alamatinstansi','$tgl_ke
rja','$jabatan)";

$hasil=mysql_query($perintah);

if ($hasil==1)
    {
        echo "Pengisian data berhasil!<br> <meta http-equiv=refresh Content='2 url=?mode=index'>";
    }
else
    {
        echo "Pengisian data gagal!<br> <meta http-equiv=refresh Content='2 url=?mode=index'>";
    }
}
}
?>

<?php
$mySelf=$_SERVER['PHP_SELF'];
?>
<br>
<form action="<? $mySelf ?>" method="post" name="input" enctype="multipart/form-data">
<fieldset><legend>Form Pengisian Data Alumni</legend>

<table>
    <tr>
        <td>Nama</td><td>:</td><td><input type="text" name="nama" value="<? $nama ?>"></td>
    </tr>

    <tr>
        <td>NIM</td><td>:</td><td><input type="text" name="nim" value="<? $nim ?>"></td>
    </tr>

    <tr>
        <td>Angkatan</td><td>:</td><td><input type="text" name="angkatan" value="<? $angkatan
?>"></td>

```

```

</tr>

<tr>
<td>Lulus</td><td>:</td><td>
<SELECT NAME="tgl">
  <OPTION VALUE="0" SELECTED >--Tgl--
  <?php
//Bentuk pilihan tanggal 0 sampai dengan 31
for($i = 1; $i < 32; $i++)
{
  if ($tgl == $i)
    $sel = "SELECTED";
  else
    $sel = "";

  print("<OPTION VALUE=\"\$i\" $sel>$i");
}
?>
  </OPTION>
  </SELECT> <SELECT NAME="bulan">
  <OPTION VALUE="0" SELECTED >--Bulan--
  <?php
for($i=1; $i<13; $i++)
{
  $sel[$i] = "";

  if ($i == $bulan)
    $sel[$i] = "SELECTED";
}

print("<OPTION VALUE=\"Januari\" $sel[1]>Januari ");
print("<OPTION VALUE=\"Pebruari\" $sel[2]>Pebruari ");
print("<OPTION VALUE=\"Maret\" $sel[3]>Maret ");
print("<OPTION VALUE=\"April\" $sel[4]>April ");
print("<OPTION VALUE=\"Mei\" $sel[5]>Mei ");
print("<OPTION VALUE=\"Juni\" $sel[6]>Juni ");
print("<OPTION VALUE=\"Juli\" $sel[7]>Juli ");
print("<OPTION VALUE=\"Agustus\" $sel[8]>Agustus ");
print("<OPTION VALUE=\"September\" $sel[9]>September");
print("<OPTION VALUE=\"Oktober\" $sel[10]>Oktober ");
print("<OPTION VALUE=\"November\" $sel[11]>November");
print("<OPTION VALUE=\"Desember\" $sel[12]>Desember");
?>
  </OPTION>
  </SELECT> <SELECT NAME="tahun">
  <OPTION VALUE="0" SELECTED >--Tahun--
  <?php
// Bentuk pilihan tahun dari 1900
// sampai sekarang

$sekarang = (integer) date("Y");
for($i = 1980; $i <= $sekarang; $i++)
{

```

```

if ($i == $tahun)
    $sel = "SELECTED";
else
    $sel = "";

print("<OPTION VALUE=\"\$i\" $sel>");
}
?>
    </OPTION>
</SELECT>

</td>
</tr>

<tr>
<td>E-mail</td><td>:</td><td><input type=text name="email"></td>
</tr>

<tr>
<td>Jenis Kelamin</td><td>:</td><td>
<?php
$checkedP = "";
$checkedW = "";
if ($jeniskelamin == "Wanita")
    $checkedW = "CHECKED";
else
    $checkedP = "CHECKED";

print("<INPUT TYPE=\"RADIO\" NAME=\"jeniskelamin\"");
print(" VALUE=\"P\" $checkedP >Pria");
print("<INPUT TYPE=\"RADIO\" NAME=\"jeniskelamin\"");
print(" VALUE=\"W\" $checkedW >Wanita");
?>
</td>
</tr>

<tr>
<td>Upload Foto</td><td>:</td><td><input type="file" size="30" name="upfile"></td>
</tr>
</table>

<fieldset><legend>Pekerjajaan</legend>
<table>
<tr>
<td>Nama Instansi</td><td>:</td><td><input type=text name="instansi"></td>
</tr>

<tr>
<td>Alamat Instansi</td><td>:</td><td><textarea name="alamatinstansi"></textarea></td>
</tr>

<tr>
<td>Mulai Bekerja</td><td>:</td><td><input type=text name="tgl_kerja"></td>

```

```

</tr>

<tr>
<td>Jabatan</td><td>:</td><td><input type="text" name="jabatan"></td>
</tr>
</table>

<input type="submit" name="simpan" value="Simpan">
<input type="reset" name="reset" value="Batal">
</fieldset>
</form>

```

4.1.2.2 Implementasi Perancangan Layar Data Alumni

Implementasi perancangan layar data alumni dibuat dengan tujuan untuk melihat siapa saja alumni yang sudah isi biodata atau mengakses aplikasi pelacakan alumni ini, sistem akan memberi ijin kepada siapapun untuk mengakses aplikasi pelacakan alumni ini jika alumni sudah isi biodata terlebih dahulu. Gambar 4.6 merupakan implementasi perancangan layar data alumni.



Gambar 4.6 Perancangan Layar Data Alumni

```

<script language='javascript'>
function tanya(x){

    if (x == 'ubah') {text='Apakah Data ingin anda Update ?'
}else {text='Apakah Data Ingin Anda Hapus ?';}

    if (confirm (text)){
        return true;
    }else{
        return false;
    }
}

```



```

    }
</script>

<?
include "koneksi.php";
@session_start();
if(!empty($_SESSION['namauser']) and !empty($_SESSION['passuser']))
{
include "menu_admin.html";
echo "<fieldset><legend>Data Alumni</legend>
<table border=1 align=center width=100%>
<tr>
    <td align='center'><b>No</b></td>
    <td align='center'><b>Nama</b></td>
    <td align='center'><b>Nim</b></td>
    <td align='center'><b>Angkatan</b></td>
    <td align='center'><b>Lulus</b></td>
    <td align='center'><b>Email</b></td>
    <td align='center'><b>Aksi</b></td>
</tr>";
$konfir="select * from pelacakan_alumni WHERE status=1 order by nama
ASC";
$konfir1=mysql_query($konfir);
$i=0;
while ($konfir3=mysql_fetch_array($konfir1))
{
    $i++;
    echo "<tr>
        <td align='center'>$i</td>
        <td>$konfir3[nama]</td>
        <td>$konfir3[nim]</td>
        <td>$konfir3[angkatan]</td>
        <td>$konfir3[lulus]</td>
        <td>$konfir3[email]</td>
        <td align='center'>
            <a          onclick=\"return          tanya('ubah')\"
href='query_aksi.php?id=$konfir3[id]'>Ubah</a> |
            <a          onclick=\"return          tanya('hapus')\"
href='?aksi=hapus&id=$konfir3[id]'>Hapus</a>
        </td>
    </tr>";
}
echo "</table></fieldset>";

//Aksi alumni atau bukan
$aksi = $_GET['aksi'];
$id = $_GET['id'];

if(!empty($aksi) && !empty($id)){
    if($aksi == 'hapus'){

        $sql ="DELETE FROM pendataan_alumni WHERE id=$id";
$result = mysql_query($sql);

```

```

        echo "<h3>Anda Berhasil Menghapus Data dari database</h3>";
<meta http-equiv=refresh Content='2 url=data_alumni.php'>";
    }

}

}
else
{
    echo("Maaf untuk mengakses halaman ini anda harus login terlebih
dahulu.");
    echo("<a href='login.php'>Login</a>");
}
?>

```

4.1.2.3 Implementasi Perancangan Layar Konfirmasi

Implementasi perancangan layar konfirmasi dibuat dengan tujuan untuk mengkonfirmasi ke admin jika terjadi kesalahan pengisian biodata alumni, sehingga admin dapat membenarkan. Gambar 4.7 merupakan implementasi perancangan layar konfirmasi.

Gambar 4.7 Perancangan Layar Konfirmasi

```

<?php
include "koneksi.php";
//include "menu.html";

if ($_POST['simpan'])
{
    $tanggal=date("d-F-Y");
    $perintah="insert into konfirmasi(nama,nim,konfirmasi,tgl) values
('$_POST[nama]','$_POST[nim]','$_POST[konfirmasi]','$tanggal)";
    if (!empty($_POST[nama]) and !empty($_POST[nim]) and !empty($_POST[konfirmasi]))
    {

```

```

$hasil=mysql_query($perintah);
echo "Selamat data anda berhasil di simpan.....<meta http-equiv=refresh Content='2
url=?mode=konfirmasi>";

}
else
{
echo("Salah satu atau lebih data tidak anda isi!!");
echo("<br>Proses Input Konfirmasi baru gagal!!");
echo("<br><br><a href=konfirmasi.php>Ulangi</a>");
}
}
else
{

$mySelf=$_SERVER['PHP_SELF'];

?>
<br>
<form action="<? $mySelf ?>" method="post" name="input">
<table align=center>
<tr>
<td>Nama</td><td>:</td><td><input type=text name=nama></td>
</tr>
<tr>
<td>Nim</td><td>:</td><td><input type=text name=nim></td>
</tr>
<tr>
<td>Konfirmasi</td><td>:</td><td><textarea name=konfirmasi></textarea></td>
</tr>
<td colspan="3" align="right">
<input type=submit name=simpan value=Simpan>
<input type=reset name=reset value=Batal>
</td>
</table>
</form>

<?php
}
?>

```

4.1.2.4 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 4.8 merupakan Implementasi perancangan layar login admin.

Gambar 4.8 Perancangan Layar Login Admin

```

<?php
include "koneksi.php";
?>
<FORM method="post" action="login_msk.php">

<table align=center>
  <tr>
    <td>Username </td><td></td><td><INPUT  name=username type="text" class=textfieldlogin
id="username2"></td>
  </tr>
  <tr>
    <td>Password </td><td></td><td><INPUT  class=textfieldlogin  name=password
type=password></td>
  </tr>
  <tr>
    <td><td>
      <td align="right">
<input class=button type="submit" name="submit" value="Login">
<input class=button type="reset" name="reset" value="Batal">
      </td>
    </td>
  </tr>
</table>
</FORM>

```

4.1.2.5 Implementasi Perancangan Layar Admin

Implementasi perancangan layar admin digunakan oleh admin, admin akan masuk ke halaman admin setelah admin berhasil login. Gambar 4.9 merupakan Implementasi perancangan layar admin.

Selamat Datang
Administrator
Silahkan Klik di bawah ini untuk masuk
Masuk

Gambar 4.9 Perancangan Layar Admin

```

<?php
include "koneksi.php";

$username = $_POST['username'];
$password = $_POST['password'];

$cek="select * from admin where username='$username' AND password='$password'";
$cek1=mysql_query($cek);
$cek3=mysql_fetch_array($cek1);

session_start();

    $_SESSION['namauser']=$cek3['username'];
    $_SESSION['passuser']=$cek3['password'];

if ($password==$cek3[password] and $username==$cek3[username])
{
    echo "<center><br><br>Selamat Datang <br><strong>Administrator</strong><br>";
    echo "Silahkan Klik di bawah ini untuk masuk<br>";
    echo("<a href='admin.php'>Masuk</a></center>");
}
else
{
echo("<br><br><center>Login gagal, karena username dan password tidak benar.<br>");
echo("Atau anda belum terdaftar. Silahkan hubungi administrator anda.<br><br>");
echo("<a href='login.php'>Login ulang</a><br></center>");
}
?>

```

4.1.2.5.1 Implementasi Perancangan Layar Cek Data

Implementasi perancangan layar cek data digunakan oleh admin, dimana admin dapat mengecek apakah benar bahwa nama, nim, angkatan, lulus, email, termasuk alumni jurusan teknik informatika, jika Ya → Benar alumni dan jika Tdk → Bukan alumni

Cek Data						
No	Nama	Nim	Angkatan	Lulus	Email	Aksi
1	Ssanusi	1234	127	1-Januari-1980	sanusi@yahoo.com	Ya Tidak
2	Ssanusi	1234	127	1-Januari-1980	sanusi@yahoo.com	Ya Tidak
3	Ssanusi	1234	127	1-Januari-1980	sanusi@yahoo.com	Ya Tidak
4	ssss	123123	123	2-Februari-1981	sanusi@yahoo.com	Ya Tidak
5	ssss	123123	123	2-Februari-1981	sanusi@yahoo.com	Ya Tidak

Gambar 4.10 Perancangan Layar Cek Data

```

<?
include "koneksi.php";
session_start();
if(!empty($_SESSION['namauser']) and !empty($_SESSION['passuser']))
{
include "menu_admin.html";
echo "<fieldset><legend>Cek Data</legend>";
<table border=1 align=center
width=100%><tr><td>No</td><td>Nama</td><td>Nim</td><td>Angkatan</td>
<td>Lulus</td><td>Email</td><td>Aksi</td></tr>";
$konfir="select * from pelacakan_alumni order by nama ASC";
$konfir1=mysql_query($konfir);
$i=0;
while ($konfir3=mysql_fetch_array($konfir1))
{
    $i++;
    echo
    "<tr><td>$i</td><td>$konfir3[nama]</td><td>$konfir3[nim]</td><td>$konfir3[angkatan]</td>
    <td>$konfir3[lulus]</td><td>$konfir3[email]</td><td><a href=#>Ya</a> | <a href=#>Tidak</a>
    </td></tr>";
}
echo "</table></fieldset>";

}
else
{
    echo("Maaf untuk mengakses halaman ini anda harus login terlebih dahulu.");
    echo("<a href='login.php'>Login</a>");
}
?>

```

4.1.2.5.2 Implementasi Perancangan Layar Tambah Data

Implementasi perancangan layar tambah data digunakan oleh admin, dimana admin dapat menambahkan data alumni jika alumni mengkonfirmasi yang diberikan oleh alumni.

Gambar 4.11 Perancangan Layar Tambah Data

4.1.2.5.3 Implementasi Perancangan Layar Data Alumni

Implementasi perancangan layar data alumni digunakan oleh admin, dimana admin dapat melihat siapa-siapa alumni yang sudah mengisi form Pendataan alumni.

No	Nama	Nim	Angkatan	Lulus	Email	Aksi
1	Ssanusi	1234	127	1-Januari-1980	sanusi@yahoo.com	Ubah Hapus
2	Ssanusi	1234	127	1-Januari-1980	sanusi@yahoo.com	Ubah Hapus
3	Ssanusi	1234	127	1-Januari-1980	sanusi@yahoo.com	Ubah Hapus
4	ssss	123123	123	2-Februari-1981	sanusi@yahoo.com	Ubah Hapus
5	ssss	123123	123	2-Februari-1981	sanusi@yahoo.com	Ubah Hapus

Gambar 4.12 Perancangan Layar Data Alumni

```

<?
include "koneksi.php";
session_start();
if(!empty($_SESSION['namauser']) and !empty($_SESSION['passuser']))
{
include "menu_admin.html";

$mySelf=$_SERVER['PHP_SELF'];
echo "<fieldset><legend>Data Alumni</legend>
<table border=1 align=center
width=100%><tr><td>No</td><td>Nama</td><td>Nim</td><td>Angkatan</td>
<td>Lulus</td><td>Email</td><td>Aksi</td></tr>";
$konfir="select * from pelacakan_alumni order by nama ASC";
$konfir1=mysql_query($konfir);
$i=0;
while ($konfir3=mysql_fetch_array($konfir1))
{
    $i++;
    echo
" <tr><td>$i</td><td>$konfir3[nama]</td><td>$konfir3[nim]</td><td>$konfir3[angkatan]</td>
    <td>$konfir3[lulus]</td><td>$konfir3[email]</td>
    <td><a href=query_aksi.php?&ubah=ubah'>Ubah</a> | <a href=$mySelf.'&hapus=hapus'>Hapus</a>
</td></tr>";
}
echo "</table></fieldset>";
}

else
{
    echo("Maaf untuk mengakses halaman ini anda harus login terlebih dahulu.");
    echo("<a href='login.php'>Login</a>");
}
?>

```

4.1.2.5.4 Implementasi Perancangan Layar Konfirmasi

Implementasi perancangan layar konfirmasi digunakan oleh admin, apabila ada kesalahan redaksi di aplikasi, layar konfirmasi ini berfungsi untuk mengkonfirmasi data-data yang benar ke admin.

No	Nama	Nim	Konfirmasi
1	Eff	00000	00000
2	Eff	00000	00000
3	Eff	00000	00000
4	Eff	00000	00000
5	Eff	00000	00000
6	Nama Konfirmasi	NIM Konfirmasi	Konfirmasi
7	Eff	00000	00000

Gambar 4.13 Perancangan Layar Konfirmasi

```

<?
include "koneksi.php";
session_start();
if(!empty($_SESSION['namauser']) and !empty($_SESSION['passuser']))
{
include "menu_admin.html";
echo "<fieldset><legend>Konfirmasi</legend><table border=1 align=center
width=100%><tr><td>No</td><td>Nama</td><td>Nim</td><td>Konfirmasi</td></tr>";
$konfir="select * from konfirmasi order by nama ASC";
$konfir1=mysql_query($konfir);
$i=0;
while ($konfir3=mysql_fetch_array($konfir1))
{
    $i++;
    echo
    "<tr><td>$i</td><td>$konfir3[nama]</td><td>$konfir3[nim]</td><td>$konfir3[konfirmasi]</td></tr>";
}
echo "</table></fieldset>";

}
else
{
    echo("Maaf untuk mengakses halaman ini anda harus login terlebih dahulu.");
    echo("<a href='login.php'>Login</a>");
}
?>

```

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 pendataan alumni ini adalah Microsoft Windows XP.
- b. Macromedia Dreamweaver MX 2004 digunakan untuk membuat program aplikasi pendataan alumni ini.
- c. Adobe Photoshop 7.0, digunakan untuk mendesain tampilan pada aplikasi pendataan alumni ini.
- d. Xampp digunakan untuk menjalankan aplikasi pendataan alumni ini
- e. Phpmyadmin digunakan untuk membuat basis data.
- f. *Browser* internet, seperti Internet Explorer, Mozilla atau Opera.

2. Spesifikasi Perangkat Keras, meliputi

- a. Komputer dengan prosesor AMD Sempron 2.2 GHz.
- b. Memori 256 MB.
- c. VGA 64 MB.
- d. Monitor 17'
- 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 pendataan alumni 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, Mozilla atau Opera digunakan untuk melakukan pengujian pada program aplikasi pelacakan alumni, dengan cara mengetikkan alamat URL sebagai berikut: <http://localhost/sanusi2/index.php>. *Black-box testing* melakukan pengujian pada program aplikasi pelacakan alumni dimulai dari menu utama sampai semua menu yang terdapat pada program aplikasi pelacakan alumni.

Tabel 4.1 Skenario Pengujian Perangkat lunak

No	Deskripsi Fungisonal	Kelompok Uji	Kasus Uji	Hasil Yang Diharapkan
1	Halaman input data alumni	Normal	Mengisi data	Menampilkan halaman selesai
2	Halaman data alumni	Normal	Melihat	Menampilkan halaman lihat

3	Konfirmasi	Normal	Mengisi data	Menampilkan halaman konfirmasi
4	Halaman login admin	Normal	Mengisi data	Menampilkan halaman admin
5	Login admin	Data salah	Tidak Memasukan <i>username</i> dan <i>password</i>	Menampilkan pesan kesalahan
6	Halaman cek data alumni	Normal	Memilih <i>link</i> cek data	Menampilkan cek data alumni
7	Halaman tambah data	Normal	Memilih <i>link</i> l tambah data	Menampilkan halaman tambah data alumni
8	Halaman data alumni	Normal	Memilih <i>link</i> data alumni	Menampilkan halaman data alumni
9	Halaman konfirmasi	Normal	Memilih link konfirmasi	Menampilkan konfirmasi
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 input data alumni	Normal	Memilih <i>link</i> input data alumni	Menampilkan halaman input data alumni	Diterima
2	Halaman data alumni	Normal	Memilih <i>link</i> data alumni	Menampilkan halaman data alumni	Diterima
3	Konfirmasi	Normal	Memilih <i>link</i> konfirmasi	Menampilkan konfirmasi	Diterima
4	Halaman login admin	Normal	Memasukan <i>username=admin</i> dan <i>password=admin</i>	Menampilkan halaman pendataan alumni	Diterima
5	Halaman login admin	Data salah	Tidak Memasukan <i>username</i> dan <i>password</i>	Menampilkan pesan kesalahan	Diterima
6	Halaman cek_data alumni	Normal	Memilih <i>link</i> cek_data alumni	Menampilkan halaman hasil cek_data alumni	Diterima

7	Halaman tambah alumni	Normal	Memilih <i>link</i> data alumni	Menampilkan halaman data alumni	Diterima
8	Halaman data alumni	Normal	Memilih <i>link</i> data alumni	Menampilkan halaman hasil alumni	Diterima
9	Halaman konfirmasi	Normal	Memilih <i>link</i> konfirmasi	Menampilkan halaman hasil konfirmasi	Diterima
10	Halaman logout	Normal	Memilih <i>link</i> <i>logout</i>	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 4.14 merupakan gambar yang menunjukkan hasil pengujian halaman input data.

The screenshot shows a web application titled 'Aplikasi Pendataan Alumni' for 'Program Studi Teknik Informatika Univ. Mercu Buana'. The navigation menu includes 'Home', 'Input Data', 'Data Alumni', 'Konfirmasi', and 'Login Admin'. The main content area is titled 'Form Pengisian Data Alumni' and contains the following fields:

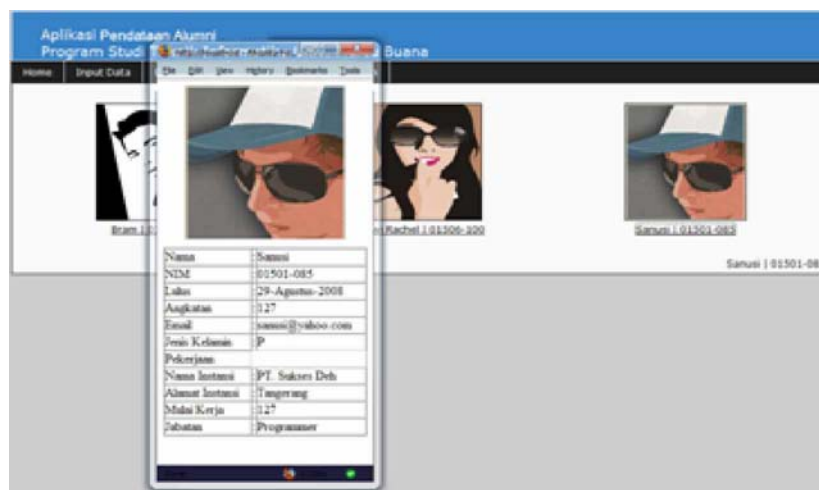
- Nama :
- NIM :
- Angkatan :
- Lulus :
- Email :
- Jenis Kelamin : Pria Wanita
- Upload Foto :
- Pekerjaan:
 - Nama Instansi :
 - Alamat Instansi :
 - Mulai Bekerja :
 - Jabatan :
- Buttons:

Footer: Sanusi | 01501-085

Gambar 4.14 Hasil Pengujian Halaman Input Data.

Pada halaman input data, pengujian dilakukan dengan kelompok uji salah yang berarti kelompok uji memasukan data salah. Pada Gambar 4.14 kelompok uji tidak memasukan semua biodata alumni, sistem akan menampilkan pesan kesalahan, dan anda diminta oleh sistem untuk kembali ke halaman input data. Pengujian pada halaman ini menghasilkan rancangan yang sesuai diinginkan.

Gambar 4.15 merupakan gambar yang menunjukkan proses yang terjadi setelah alumni melihat halaman data alumni.



Gambar 4.15 Hasil Pengujian Halaman Data Alumni Detail

Pada halaman data alumni, pengujian dilakukan dengan kelompok uji salah yang berarti kelompok uji tidak dapat melihat data alumni. Pada Gambar 4.15 kelompok uji tidak melihat semua biodata alumni, sistem akan menampilkan pesan kesalahan, dan anda diminta oleh sistem untuk kembali ke halaman input data. Pengujian pada halaman ini menghasilkan rancangan yang sesuai diinginkan.

Gambar 4.9 merupakan gambar yang menunjukkan proses yang terjadi setelah admin melakukan *login*.

Pada halaman login admin, pengujian dilakukan dengan kelompok uji normal yang berarti kelompok uji memasukkan data benar. Kelompok uji melakukan login dengan *username=admin* dan *password=admin*, sistem akan memproses *username* dan *password* tersebut. Setelah *username* dan *password* diproses, sistem akan menampilkan halaman admin, seperti yang terlihat pada Gambar 4.9

Gambar 4.16 merupakan gambar yang menunjukkan hasil pengujian pada halaman *cek_data*.



Gambar 4.16 Hasil Pengujian Halaman *cek_data*.

Pada halaman *cek_data*, pengujian dilakukan dengan cara melihat data yang di pilih. Pada halaman *cek_data* akan diberi waktu oleh sistem untuk melihat hasil *cek_data*. Apabila alumni tidak mengisi data sesuai dengan yang telah ditentukan, maka secara otomatis halaman akan tertutup. Setelah alumni selesai mengisi form aplikasi alumni, sistem akan menampilkan halaman selesai. Pada halaman selesai terdapat *link* hasil yang berarti admin memilih *link* *cek_data* untuk mengetahui hasil aplikasi alumni yang diperoleh.

Gambar 4.11 merupakan gambar yang menunjukkan hasil pengujian pada halaman Tambah data.

Pada halaman tambah data, pengujian dilakukan dengan kelompok uji data salah yang berarti kelompok uji tambah data salah. Pada Gambar 4.11 kelompok uji tidak memasukan data alumni, sistem akan menampilkan pesan kesalahan. Pada halaman ini terdapat dua buah tombol. Tombol pertama berguna untuk menyimpan data ke dalam basis data jika data alumni yang dimasukan telah benar, tombol kedua berguna untuk batal.

Gambar 4.12 merupakan gambar yang menunjukkan hasil pengujian pada halaman data alumni.

Pada halaman data alumni, pengujian dilakukan dengan menampilkan data alumni secara keseluruhan dengan kelompok uji benar. Sistem akan menampilkan data alumni yang terdapat didalam basis data berdasarkan kode data alumni. Data yang terlihat pada Gambar 4.12 ditampilkan secara terpotong, untuk melihat data alumni secara utuh kelompok uji memilih *link* selengkapnya. Pada halaman ini juga terdapat *link* untuk menghapus data alumni dan merubah data alumni.

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

Pada halaman konfirmasi, pengujian dilakukan dengan kelompok uji benar yang berarti dijalankan dengan semestinya. Pada halaman ini digunakan oleh admin untuk melihat alumni yang mengkonfirmasi datanya yang terdaftar dan juga digunakan oleh admin untuk menghapus data setiap alumni. Untuk menghapus data alumni, admin

memilih *link* hapus, setelah admin memilih *link* hapus sistem akan memproses dengan menghapus data alumni yang terdapat didalam basis data berdasarkan nama alumni.

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

Pengujian yang dilakukan dari halaman alumni hingga halaman admin menghasilkan proses keluaran sesuai dengan yang dirancang. Secara fungsionalitas program, program aplikasi pendataan alumni ini tidak terdapat kesalahan dan hasil program sesuai dengan hasil yang telah diproses didalam sistem. Hasil analisis yang diperoleh menghasilkan rancangan aplikasi pendataan alumni yang telah dibangun secara keseluruhan berfungsi baik.

BAB V

PENUTUP

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. Setelah dilakukan beberapa kali pengujian, dapat diketahui bahwa aplikasi pendataan alumni ini menghasilkan proses, yaitu: Input data, Lihat data alumni, memvalidasi proses *username* dan *password*, dan memudahkan pendataan alumni.
2. Dalam membangun aplikasi pendataan alumni ini menggunakan sebuah basis data dan tiga buah tabel yang terdiri dari *tb_admin*, *tb_pendataan alumni* dan *tb_konfirmasi*. Semua tabel berguna untuk menyimpan semua data yang diperlukan. Diantara tabel-tabel tersebut terdapat tabel utama, seperti: *tb_pendataan*

- alumni. Tabel utama tersebut mempunyai fungsi utama dalam aplikasi pendataan alumni ini, yaitu menampilkan halaman form pendataan alumni dan memproses isi data alumni untuk ditampilkan pada halaman data alumni.
3. Aplikasi pendataan alumni yang dibangun telah memenuhi tujuan keberhasilan, yang dalam hal ini alumni lebih mudah mengakses data alumni dengan teknologi berbasis *web*.

5.2 Saran

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

1. Agar informasi yang ditampilkan lebih menarik, aplikasi ini dapat ditambahkan fasilitas-fasilitas seperti: buku tamu, pooling, dan pencarian.
2. Jangan terlalu banyak menggunakan gambar dengan ukuran file yang sangat besar, karena dapat memperlambat proses pemanggilan web tersebut.
3. Jangan 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 PUSTAKA

1. Sutarman, SKom, *Membangun Aplikasi Web dengan PHP dan MySQL*, Graha Ilmu, 2003.
2. Sampurna, *Membuat Home Page dengan HTML*, Elex Media Komputindo, 1998.
3. Edi Purwono., *Apa yang harus dipahami oleh Sistem*, Andi Yogyakarta, 2002
4. Fowler, Martin. 2005. *UML Distilled edisi 3, Panduan singkat bahasa pemodelan Objek standar*. Yogyakarta : Andi.
5. Fathansyah, Ir, *Basis Data*, Informatika Bandung, 2001
6. Lani Sidharta, *PHP dan MySQL*, 1997
7. <http://www.master.web.id>
8. http://id.wikipedia.org/wiki/Kategori:Situs_web
9. <http://cybertech.cbn.net.id/detil.asp?kategori=Web>
10. <http://ikc.cbn.net.id/umum/kemas-cms.php>
11. <http://www.ilmukomputer.com>
12. <http://www.php.net/usage.php>
13. <http://www.elisa.ugm.ac./imk.php>

LAMPIRAN (CODING)

Menu Utama

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

```
<html>
  <head>
    <title>Aplikasi Pendataan Data
    Alumni</title>
    <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-15" />
    <link href="styles.css" rel="stylesheet" />
  </head>
  <body>
    <div id="conteneur">
      <div id="header">Aplikasi
      Pendataan Data Alumni <br>Program Studi Teknik
      Informatika Univ. Mercu Buana</div>
      <?include "menu.html";?>
      <div id="centre">
        <?
          $mode = $_GET['mode'];

          switch($mode){
            case 'index':

              include "index.php";

            case 'input':

              include "input.php";

            case 'alumni':

              include "alumni.php";

            case 'konfirmasi':

              include "konfirmasi.php";

            case 'login':

              include "login.php";

            default:

              include "awal.php";
          }
        <?>
      </div>
    </div>
  </body>
</html>
```

Input Data

```
<?php
```

```
include "koneksi.php";
@session_start();
if(!empty($_SESSION['namauser']) and !empty
($_SESSION['passuser']))
{
  include "menu_admin.html";
}
else
{
  //include "menu.html";
}

if ($_POST['simpan'])
{
  $nim = $_POST['nim'];
  $nama = $_POST['nama'];
  $angkatan = $_POST['angkatan'];
  $jeniskelamin = $_POST['jeniskelamin'];
  $email = $_POST['email'];
  $instansi = $_POST['instansi'];
  $alamatinstansi = $_POST['alamatinstansi'];
  $tgl_kerja = $_POST['tgl_kerja'];
  $jabatan = $_POST['jabatan'];

  if ($_FILES['upfile']['error'] == 0) {
    $IMG_DIR = "/image/foto/";
    $FILE_SIZE = 80000;

    $sqlNextVal = "SELECT
    AUTO_INCREMENT AS last_id
    FROM
    INFORMATION_SCHEMA.TABLES
    WHERE TABLE_SCHEMA = '". $dbName.'"
    AND TABLE_NAME='pendataan_alumni' ";

    $rsNextval = mysql_query($sqlNextVal);
    $data=mysql_fetch_array($rsNextval);
    $id_next = $data['last_id'];

    $type_file = $_FILES["upfile"]["type"];

    if ($type_file == "image/gif"){
      $nama_file = $id_next.".gif";
    } else if ($type_file == "image/jpeg") {
      $nama_file = $id_next.".jpg";
    }

    $error_file = $_FILES["upfile"]["error"];
    $size_file = $_FILES["upfile"]["size"];
    $temp_file = $_FILES["upfile"]["tmp_name"];
    $path_file = $IMG_DIR;
    $file_image = $path_file.$nama_file;

    $path_file_best = $IMG_DIR_BEST;
    $file_image_best = $path_file_best.$nama_file;
    if ($path_file_best == "not_available.gif");

    if (($type_file == "image/gif") || ($type_file ==
    "image/jpeg") && ($size_file < $FILE_SIZE))
    {
      if ($error_file > 0)
      {
        die ("Return Code: " . $error_file .
        "<br />");
      }
    }
  }
}
```

```

else
{
if (file_exists($path_file .
$nama_file))
{
die ($nama_file . " already exists. ");
}
else
{
move_uploaded_file($_FILES["upfile"]["tmp_name"],
$path_file . $nama_file);
}
}
else
{
if ($type_file == ""){
//Image Kosong
$file_image =
$path_file."no_picture.gif";
}else {
print "
<script>
alert ('Invalid File, Only
JPG or GIF or size image to big , max 75 Kb ');
history.back(-1);
</script>
";
exit;
}
}

$lulus="$_POST[tgl]-$_POST[bulan]-$_POST[tahun]";

$perintah="insert into pendataan_alumni
(nim,nama,angkatan,lulus,jkel,email,foto,nama_instansi,ala
mat_instansi,mulai_bekerja,jabatan) values
('$nim','$nama','$angkatan','$lulus','$jenis_kelamin','$email','$
file_image','$instansi','$salamat_instansi','$tgl_kerja','$jabatan')
";

$hasil=mysql_query($perintah);

if ($hasil==1)
{
echo "Pengisian data berhasil!<br> <meta http-
equiv=refresh Content=2 url=?mode=index>";
}
else
{
echo "Pengisian data gagal!<br> <meta http-
equiv=refresh Content=2 url=?mode=index>";
}
}
}
?>

<?php
$mySelf=$_SERVER['PHP_SELF'];
?>

```

```

<br>
<form action="<? $mySelf ?>" method="post"
name="input" enctype="multipart/form-data">
<fieldset><legend>Form Pengisian Data Alumni</legend>

<table>
<tr>
<td>Nama</td><td></td><td><input
type="text" name="nama" value="<? $nama ?>"></td>
</tr>
<tr>
<td>NIM</td><td></td><td><input type="text"
name="nim" value="<? $nim ?>"></td>
</tr>
<tr>
<td>Angkatan</td><td></td><td><input
type="text" name="angkatan" value="<? $angkatan
?>"></td>
</tr>
<tr>
<td>Lulus</td><td></td><td>
<SELECT NAME="tgl">
<OPTION VALUE="0" SELECTED >--Tgl-
-
<?php
//Bentuk pilihan tanggal 0 sampai dengan 31
for($i = 1; $i < 32; $i++)
{
if ($tgl == $i)
$sel = "SELECTED";
else
$sel = "";

print("<OPTION VALUE='\$i\' $sel>$i");
}
?>
</OPTION>
</SELECT> <SELECT NAME="bulan">
<OPTION VALUE="0" SELECTED >--
Bulan--
<?php
for($i=1; $i<13; $i++)
{
$sel[$i] = "";

if ($i == $bulan)
$sel[$i] = "SELECTED";
}

print("<OPTION VALUE='\$i\' $sel[1]>Januari\
$sel[1]>Januari ");
print("<OPTION VALUE='\$i\' $sel[2]>Pebruari\
$sel[2]>Pebruari ");
print("<OPTION VALUE='\$i\' $sel[3]>Maret
");
print("<OPTION VALUE='\$i\' $sel[4]>April
");
print("<OPTION VALUE='\$i\' $sel[5]>Mei
");
print("<OPTION VALUE='\$i\' $sel[6]>Juni
");
print("<OPTION VALUE='\$i\' $sel[7]>Juli
");

```

```

        print("<OPTION          VALUE=\"Agustus\"
$sel[8]>Agustus ");
        print("<OPTION          VALUE=\"September\"
$sel[9]>September");
        print("<OPTION          VALUE=\"Oktober\"
$sel[10]>Oktober ");
        print("<OPTION          VALUE=\"November\"
$sel[11]>November");
        print("<OPTION          VALUE=\"Desember\"
$sel[12]>Desember");
        ?>
        </OPTION>
        </SELECT> <SELECT NAME="tahun">
        <OPTION VALUE="0" SELECTED >--
Tahun--
        <?php
// Bentuk pilihan tahun dari 1900
// sampai sekarang

$sekarang = (integer) date("Y");
for($i = 1980; $i <= $sekarang; $i++)
{
    if ($i == $tahun)
        $sel = "SELECTED";
    else
        $sel = "";

    print("<OPTION VALUE=\"\$i\" $sel>\$i");
}
?>
        </OPTION>
        </SELECT>

        </td>
        </tr>

        <tr>
        <td>E-mail</td><td></td><td><input type=text
name="email"></td>
        </tr>

        <tr>
        <td>Jenis Kelamin</td><td></td><td>
        <?php
$checkedP = "";
$checkedW = "";
if ($jeniskelamin == "Wanita")
    $checkedW = "CHECKED";
else
    $checkedP = "CHECKED";

print("<INPUT          TYPE=\"RADIO\"
NAME=\"jeniskelamin\"");
print(" VALUE=\"P\" $checkedP >Pria");
print("<INPUT          TYPE=\"RADIO\"
NAME=\"jeniskelamin\"");
print(" VALUE=\"W\" $checkedW >Wanita");
?>
        </td>
        </tr>

        <tr>
        <td>Upload Foto</td><td></td><td><input
type="file" size="30" name="upfile"></td>
        </tr>
        </table>

```

```

        <fieldset><legend>Pekerjaaan</legend>
        <table>
        <tr>
        <td>Nama Instansi</td><td></td><td><input
type=text name="instansi"></td>
        </tr>
        <tr>
        <td>Alamat
Instansi</td><td></td><td><textarea
name="alamatinstansi"></td>
        </tr>
        <tr>
        <td>Mulai Bekerja</td><td></td><td><input
type=text name="tgl_kerja"></td>
        </tr>
        <tr>
        <td>Jabatan</td><td></td><td><input
type=text name="jabatan"></td>
        </tr>
        </table>

        <input type=submit name=simpan
value=Simpan>
        <input type=reset name=reset value=Batal>
        </fieldset>
        </form>

```

Data Alumni

```

<script language=' javascript'>
    function tanya(x){

        if (x == 'ubah')
        {text='Apakah Data ingin anda Update ?'
        }else {text='Apakah Data
        Ingin Anda Hapus ?';}

        if (confirm (text)){
            return true;
        }else{
            return false;
        }
    }
</script>

<?
include "koneksi.php";
@session_start();
if(!empty ($_SESSION['namauser']) and
!empty ($_SESSION['passuser']))
{
include "menu_admin.html";
echo " <fieldset><legend>Data
Alumni</legend>
<table border=1 align=center width=100%>
<tr>
        <td align='center'><b>No</b></td>
        <td align='center'><b>Nama</b></td>
        <td align='center'><b>Nim</b></td>

```



```

        <td
align='center'><b>Angkatan</b></td>
        <td
align='center'><b>Lulus</b></td>
        <td
align='center'><b>Email</b></td>
        <td
align='center'><b>Aksi</b></td>
</tr>;
$konfir="select * from pelacakan_alumni
WHERE status=1 order by nama ASC";
$konfir1=mysql_query($konfir);
$i=0;
while
($konfir3=mysql_fetch_array($konfir1))
{
    $i++;
    echo "<tr>
        <td
align='center'>$i</td>
        <td>$konfir3[nama]</td>
        <td>$konfir3[nim]</td>

        <td>$konfir3[angkatan]</td>
        <td>$konfir3[lulus]</td>
        <td>$konfir3[email]</td>
        <td align='center'>
            <a
onclick=\"return tanya('ubah')\"
href='query_aksi.php?id=$konfir3[id]'\>Ubah</a> |
            <a
onclick=\"return tanya('hapus')\"
href='?aksi=hapus&id=$konfir3[id]'\>Hapus
</a>
        </td>
    </tr>";
}
echo "</table></fieldset>";

//Aksi alumni atau bukan
$aksi = $_GET['aksi'];
$id = $_GET['id'];

if(!empty($aksi) && !empty($id)){
    if($aksi == 'hapus'){

        $sql = "DELETE FROM
pendataan_alumni WHERE id=$id";
        $result =
mysql_query($sql);
        echo "<h3>Anda Berhasil
Menghapus Data dari database</h3> <meta
http-equiv=refresh Content='2
url=data_alumni.php'>";
    }
}
}
else
{
    echo("Maaf untuk mengakses
halaman ini anda harus login terlebih
dahulu.");
}

```

```

        echo( "<a
href='login.php'>Login</a>" );
    }
    ?>

```

Konfirmasi

```

<?php
include "koneksi.php";
//include "menu.html";

if ($_POST['simpan'])
{
    $tanggal=date("d-F-Y");
    $perintah="insert into konfirmasi(nama,nim,konfirmasi,tgl)
values
($_POST[nama],'$_POST[nim]','$_POST[konfirmasi]','$tan
ggal)";
    if (!empty($_POST[nama]) and !empty($_POST[nim]) and
!empty($_POST[konfirmasi]))
    {
        $hasil=mysql_query($perintah);
        echo "Selamat data anda berhasil di simpan....<meta http-
equiv=refresh Content='2 url=?mode=konfirmasi'>";
    }
    else
    {
        echo("Salah satu atau lebih data tidak anda isi!!");
        echo("<br>Proses Input Konfirmasi baru gagal!!");
        echo("<br><br><a href=konfirmasi.php>Ulangi</a>");
    }
}
else
{
    $mySelf=$_SERVER['PHP_SELF'];

    ?>
    <br>
    <form action="<? $mySelf ?>" method="post"
name="input">
    <table align=center>
    <tr>
    <td>Nama</td><td>:</td><td><input
type=text
name=nama</td>
    </tr>
    <tr>
    <td>Nim</td><td>:</td><td><input
type=text
name=nim</td>
    </tr>
    <tr>
    <td>Konfirmasi</td><td>:</td><td><textarea
name=konfirmasi</td>
    </tr>
    <td colspan="3" align="right">
    <input type=submit name=simpan value=Simpan>
    <input type=reset name=reset value=Batal>
    </td>
    </table>
    </form>

    <?php
    }
    ?>

```

Login Admin

```
<?php
include "koneksi.php";
?>
<FORM method="post" action="login_msk.php">

<table align=center>
  <tr>
    <td>Username </td><td></td><td><INPUT
name=username type="text" class=textfieldlogin
id="username2"></td>
    </tr>
    <tr><td>Password </td><td></td><td><INPUT
class=textfieldlogin name=password type=password></td>
    </tr>
    <tr>
      <td><td>
        <td align="right">
<input class=button type="submit" name="submit"
value="Login">
<input class=button type="reset" name="reset"
value="Batal">
        </td>
      </tr>
</table>
</FORM>
```

Lavar Admin

```
<?php
include "koneksi.php";

$username =$_POST['username'];
$password =$_POST['password'];

$cek="select * from admin where username='$username'
AND password='$password'";
$cek1=mysql_query($cek);
$cek3=mysql_fetch_array($cek1);

session_start();

$_SESSION['namauser']=$cek3['username'];
$_SESSION['passuser']=$cek3['password'];

if ($password==$cek3[password] and
$username==$cek3[username])
{
  echo "<center><br><br>Selamat Datang
<br><strong>Administrator</strong><br>";
  echo "Silahkan Klik di bawah ini untuk
masuk<br>";
  echo("<a
href='admin.php'>Masuk</a></center>");
}
else
{
  echo("<br><br><center>Login gagal, karena username dan
password tidak benar.<br>");
  echo("Atau anda belum terdaftar. Silahkan hubungi
administrator anda.<br><br>");
}
```

```
echo("<a href='login.php'>Login
ulang</a><br></center>");
}
```

```
?>
```

Cek Data

```
<?
include "koneksi.php";
session_start();
if(!empty($_SESSION['namauser']) and !empty
($_SESSION['passuser']))
{
  include "menu_admin.html";
  echo "<fieldset><legend>Cek Data</legend>
<table border=1 align=center
width=100%><tr><td>No</td><td>Nama</td><td>Nim</t
d><td>Angkatan</td>
<td>Lulus</td><td>Email</td><td>Aksi</td></tr>";
  $konfir="select * from pelacakan_alumni order by nama
ASC";
  $konfir1=mysql_query($konfir);
  $i=0;
  while ($konfir3=mysql_fetch_array($konfir1))
  {
    $i++;
    echo
"<tr><td>$i</td><td>$konfir3[nama]</td><td>$konfir3[ni
m]</td><td>$konfir3[angkatan]</td>
    <td>$konfir3[lulus]</td><td>$konfir3[email]</t
d><td><a href=#>Ya</a> | <a href=#>Tidak</a>
</td></tr>";
  }
  echo "</table></fieldset>";
}
else
{
  echo("Maaf untuk mengakses halaman ini anda
harus login terlebih dahulu.");
  echo("<a href='login.php'>Login</a>");
}
?>
```

Data Alumni admin

```
<?
include "koneksi.php";
session_start();
if(!empty($_SESSION['namauser']) and !empty
($_SESSION['passuser']))
{
  include "menu_admin.html";

  $mySelf=$_SERVER['PHP_SELF'];
  echo "<fieldset><legend>Data Alumni</legend>
<table border=1 align=center
width=100%><tr><td>No</td><td>Nama</td><td>Nim</t
d><td>Angkatan</td>
<td>Lulus</td><td>Email</td><td>Aksi</td></tr>";
  $konfir="select * from pelacakan_alumni order by nama
ASC";
```

```

$konfir1=mysql_query($konfir);
$i=0;
while ($konfir3=mysql_fetch_array($konfir1))
{
    $i++;
    echo
    "<tr><td>$i</td><td>$konfir3[nama]</td><td>$konfir3[ni
    m]</td><td>$konfir3[angkatan]</td>
    <td>$konfir3[lulus]</td><td>$konfir3[email]</t
    d>
    <td><a href=query_aksi.php?&ubah=ubah>Ubah</a> |
    <a href=$mySelf.'&hapus=hapus'>Hapus</a> </td></tr>";
}
echo "</table></fieldset>";

}

else
{
    echo("Maaf untuk mengakses halaman ini anda
    harus login terlebih dahulu.");
    echo("<a href='login.php'>Login</a>");
}
?>

```

Konfirmasi Admin

```

<?
include "koneksi.php";
session_start();
if(!empty($_SESSION['namauser']) and !empty
($_SESSION['passuser']))
{
include "menu_admin.html";
echo "<fieldset><legend>Konfirmasi</legend><table
border=1 align=center
width=100%><tr><td>No</td><td>Nama</td><td>Nim</t
d><td>Konfirmasi</td></tr>";
$konfir="select * from konfirmasi order by nama ASC";
$konfir1=mysql_query($konfir);
$i=0;
while ($konfir3=mysql_fetch_array($konfir1))
{
    $i++;
    echo
    "<tr><td>$i</td><td>$konfir3[nama]</td><td>$konfir3[ni
    m]</td><td>$konfir3[konfirmasi]</td></tr>";
}
echo "</table></fieldset>";

}

else
{
    echo("Maaf untuk mengakses halaman ini anda
    harus login terlebih dahulu.");
    echo("<a href='login.php'>Login</a>");
}
?>

```

RIWAYAT HIDUP



Sanusi
Jl.Abadi RT. 02 RW 03 Kebon Besar
Batuceper Tangerang 15122
Telp. 021-92739345, 08129824682

Identitas Diri

Nama	: Sanusi	Tinggi	: 168 cm
Tempat / Tgl Lahir	: Tangerang, 29 Oktober 1982	Berat	: 59 kg
Kewarganegaraan	: Indonesia	Sex	: Laki-Laki
Alamat	: Jl.Abadi RT. 02 RW. 03 Kebon Besar, Kel. Kebon Besar, Batuceper Tangerang 15122	Status	: Belum menikah
E-mail	: nanu_01085@yahoo.com	Telp.	: (021) 92739345
		HP	: 0812-9824682

Latar Belakang Pendidikan

Tahun	Status	Jurusan	Universitas/Sekolah
2001 – 2008	S1	T. Informatika	Universitas Mercu Buana
1997 - 2000	Berijazah	-	SMK YUPPENTEK 1, Tangerang
1994 - 1997	Berijazah	-	SLTP YP-KARYA, Tangerang
1988 - 1994	Berijazah	-	SDN Empang Bahagia 1, Tangerang