PENERAPAN TEKNOLOGI AJAX UNTUK MEMBANGUN WEBSITE STUDI KASUS PADA YAYASAN MANARUL HUDA

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika



Disusun oleh:

Nama : Acep Irham Gufroni
NIM : 03 523 167

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PENGESAHAN PEMBIMBING

PENERAPAN TEKNOLOGI AJAX UNTUK MEMBANGUN WEBSITE STUDI KASUS PADA YAYASAN MANARUL HUDA

TUGAS AKHIR

Disusun Oleh :

Nama : Acep Irham Gufroni
NIM : 03 523 167

Yogyakarta, 12 Desember 2007

Pembimbing I

Taufiq Hidayat, ST., MCS.

Pembimbing II

Hendrik, ST.

LEMBAR PENGESAHAN PENGUJI

PENERAPAN TEKNOLOGI AJAX UNTUK MEMBANGUN WEBSITE STUDI KASUS PADA YAYASAN MANARUL HUDA

TUGAS AKHIR

Disusun Oleh :

Nama : Acep Irham Gufroni
NIM : 03 523 167

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 2 Januari 2008

Tim Penguji:

Taufiq Hidayat, ST., MCS. _____
Ketua

Hendrik, ST. _____
Anggota I

Syarif Hidayat, S.Kom. _____
Anggota II

Mengetahui
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

ISLAM
UNIVERSITAS
INDONESIA

ISLAM
UNIVERSITAS
INDONESIA

KATA PENGANTAR

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

Assalamu'alaikum Wr. Wb.

Alhamdulillah segala puji bagi Alloh SWT. Atas segala rahmat, hidayah dan inayah-Nya, sehingga penulisan laporan tugas akhir yang berjudul ***Penerapan Teknologi AJAX Untuk Membangun Website Studi Kasus Pada Yayasan Manarul Huda*** dapat penulis selesaikan dengan baik. Laporan tugas akhir ini disusun sebagai salah satu syarat guna memperoleh gelar sarjana Teknik Informatika pada Universitas Islam Indonesia. Dan juga sebagai sarana untuk mempraktekkan secara langsung ilmu dan teori yang telah diperoleh selama menjalani masa studi di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Penyusunan laporan ini tidak lepas dari bimbingan, dukungan dan bantuan baik materil maupun spirituil dari berbagai pihak. Oleh karena itu dalam kesempatan ini dengan segala kerendahan hati, penulis bermaksud menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

a. Alloh SWT, Tuhan bagi seluruh alam yang melimpahkan rahmat dan karunia-Nya sehingga penulis diberi kesehatan dan kemudahan selama pembuatan tugas akhir ini.
b. APA dan MAMAH beserta keluarga yang telah memberikan seluruh do'a dan restu, serta dorongan baik spirituil maupun materil sehingga penulis dapat menyelesaikan studi dengan baik .
c. Bapak Rektor dan seluruh jajaran Rektoriat Universitas Islam Indonesia.
d. Bapak Fathul Wahid, ST., M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
e. Bapak Yudi Prayudi, S.Si., M.Kom, selaku Ketua Jurusan Teknik Informatika. Terima kasih atas kemudahan dan dukungan yang telah diberikan.

f. Bapak Taufiq Hidayat, ST., MCS. selaku dosen payung dalam pembuatan tugas akhir ini.

g. Bapak Hendrik, ST. selaku pembimbing yang telah memberikan pengarahan, bimbingan, serta masukan selama pelaksanaan tugas akhir dan penulisan laporan.

h. Dosen-dosen Jurusan Teknik Informatika. Terima kasih atas semua ilmu pengetahuan dan motivasi serta bantuannya.

i. Teman-teman ICON'03 yang tidak bisa disebutkan satu per satu. Terima kasih untuk semuanya, pokonya *one for all, all for one*..

j. Keluarga besar Yayasan Manarul Huda yang telah memberikan kesempatan dan kerjasama yang sangat baik dalam pembuatan tugas akhir ini.

k. Seseorang yang selalu menjadi bintang yang menemani hari-hariku, menjadi cahaya setiap langkahku (...).

Penulis menyadari bahwa dalam penyusunan laporan tugas akhir ini masih banyak terdapat kekeliruan dan kekurangan, untuk itu penulis menyampaikan permohonan maaf sebesar-besarnya serta sangat diharapkan kritik dan saran yang sifatnya membangun untuk penyempurnaan di masa mendatang. Akhir kata semoga laporan ini dapat bermanfaat bagi penulis dan semua pembaca.

Wassalamu'alaikum Wr.Wb.

Yogyakarta, Desember 2007

Acep Irham Gufroni

SARI

Pada saat ini kebutuhan *website* tidak lagi hanya menjadi alat penyedia informasi yang statis, tetapi dalam perkembangannya *website* dituntut untuk menjadi lebih dinamis. AJAX (*Asynchronous Javascript And XML*) memberikan solusi untuk menjadikan *website* lebih responsif dan dinamis. Dengan penerapan teknologi AJAX, interaksi *website* dengan *server* dapat dilakukan secara *background* tanpa perlu me-*refresh* ulang halaman secara keseluruhan, sehingga *website* akan lebih responsif karena setiap aksi dari pengguna akan langsung direspon oleh *server*.

Website Yayasan Manarul Huda merupakan *website* yang dibangun dengan penerapan teknologi AJAX, penerapan teknologi AJAX yang digunakan terdapat pada proses *load* halaman, proses *login*, proses validasi masukan data, dan proses pencarian. Adapun perangkat lunak yang digunakan dalam membangun sistem ini adalah sistem operasi Windows, *Bundle* Apache2Triad 1.5.4 (*MySQL 5.0.8, PHP5.1.2, Apache 2.2.0*), dan Macromedia Dreamweaver.

Dengan penerapan teknologi AJAX, *website* Yayasan Manarul Huda ini memiliki keunggulan di dalam proses *login*, proses validasi masukan data, dan proses pencarian, dimana setelah mengisikan masukan data pada proses-proses tersebut, hasilnya akan langsung ditampilkan pada halaman *website* tanpa harus menekan tombol terlebih dahulu.

Kata-kunci: *Website*, Teknologi AJAX, Yayasan Manarul Huda

TAKARIR

Asynchronous	properti pada metode javascript yang digunakan dalam proses koneksi
Data Flow Diagram (DFD)	diagram yang menunjukkan aliran data atau informasi dan transformasi data yang bergerak dari pemasukan data hingga ke keluaran
edit	proses mengubah isi dari sebuah data
event-handler	penanganan pada javascript yang merespon aksi dari pengguna
form	sebuah tampilan jendela dalam sebuah aplikasi
function	fungsi pada javascript yang dibuat untuk menangani proses tertentu
hardware	perangkat keras dari sebuah komputer
input	masukan pada sistem yang umumnya berupa data yang diturunkan dari kebutuhan informasi
output	keluaran, hasil dari suatu proses baik berupa data maupun berbentuk informasi yang telah diolah
script	kode program yang ditulis dalam bahasa pemrograman
software	perangkat lunak atau program komputer
tag	kode program pada halaman HTML
user	orang yang menggunakan sistem aplikasi komputer

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PENGESAHAN PEMBIMBING ii
LEMBAR PENGESAHAN PENGUJI iii
HALAMAN PERSEMBAHAN ivi
MOTTO..... v
KATA PENGANTAR vi
SARI..... viii
TAKARIR ix
DAFTAR ISI..... x
DAFTAR GAMBAR xv
DAFTAR TABEL xix
DAFTAR LISTING xx

BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 3
1.6 Metodologi Penelitian 3
1.6.1 Metode Pengumpulan Data 3
1.6.2 Metode Pengembangan Sistem 4
1.7 Sistematika Penulisan 5

BAB II LANDASAN TEORI 7
2.1 Yayasan Manarul Huda 7
2.2 World Wide Web (WWW) 8
2.2.1 Cara Kerja World Wide Web 8
2.3 HyperText Transfer Protocol (HTTP) 10

2.4 *Client-side Scripting* dan *Server-side Scripting* 12
2.5 HTML (Hypertext Markup Language) 13
2.6 AJAX (Asynchronous Javascript and XML) 14
2.6.1 Javascript 17
2.6.2 Extensible Markup Language (XML) 18
2.6.3 XMLHttpRequest 20
2.7 Perpaduan Antara AJAX dengan PHP 23

BAB III METODOLOGI 26
3.1 Analisis Kebutuhan Perangkat Lunak 26
3.1.1 Metode Analisis 26
3.2 Hasil Analisis 26
3.2.1 Analisis Kebutuhan Input 27
3.2.2 Analisis Kebutuhan Proses 28
3.2.3 Analisis Kebutuhan Output 29
3.2.4 Kebutuhan Antar Muka 29
3.2.5 Analisis Kebutuhan Perangkat Lunak 30
3.2.6 Kebutuhan Perangkat Keras 30
3.3 Analisis dan Perancangan Sistem 31
3.3.1 Metode Perancangan 31
3.3.2 Hasil Perancangan 31
3.3.2.1 Perancangan Data Flow Diagram 31
3.3.2.1.1 Diagram Konteks 31
3.3.2.1.2 DFD level 1 33
3.3.2.1.3 DFD Level 2 Proses Manipulasi Data Anggota 34
3.3.2.1.4 DFD Level 2 Proses Manipulasi Data Forum 35
3.3.2.1.5 DFD Level 2 Proses Pencarian 35
3.3.2.2 Perancangan Penerapan Teknologi AJAX 36
3.3.2.2.1 Proses *Load* Halaman 37
3.3.2.2.2 Proses Validasi 38

3.3.2.2.3 Proses *Login* 39
3.3.2.2.4 Proses Pencarian 40
3.3.2.3 Perancangan Tabel Basis Data 41
3.3.2.4 Relasi Antar Tabel 47
3.3.2.5 Perancangan Antar Muka 48
3.3.2.5.1 Perancangan Halaman Pengunjung 48
3.3.2.5.2 Rancangan Halaman Anggota 52
3.3.2.5.3 Rancangan Halaman *Administrator* 54
3.3.2.5.4 Antarmuka Halaman *Login* 60

BAB IV HASIL DAN PEMBAHASAN 62
4.1 Implementasi Perangkat Lunak 62
4.1.1 Batasan Implementasi 62
4.1.2 Implementasi Antar Muka 62
4.1.2.1 Halaman Pengunjung 62
4.1.2.1.1 Halaman Beranda (Halaman Depan) 63
4.1.2.1.2 Halaman Berita 63
4.1.2.1.3 Halaman Pengumuman 64
4.1.2.1.4 Halaman Forum 65
4.1.2.1.5 Halaman Tambah Topik 66
4.1.2.1.6 Halaman Tambah Komentar 67
4.1.2.1.7 Halaman Buku Tamu 68
4.1.2.1.8 Halaman Tambah Buku Tamu 69
4.1.2.1.9 Halaman Pencarian 70
4.1.2.1.10 Halaman Yayasan 71
4.1.2.1.11 Halaman Pendidikan Salafiyah 72
4.1.2.1.12 Halaman Madrasah Tsanawiyah 73

4.1.2.1.13 Halaman Madrasah Aliyah 74
4.1.2.1.14 Halaman Publikasi 75
4.1.2.1.15 Halaman Galeri 76
4.1.2.2 Halaman *Login* 77
4.1.2.3 Halaman Anggota 78
4.1.2.3.1 Halaman Status 78
4.1.2.3.2 Halaman Akademik 79
4.1.2.3.3 Halaman Tagihan 80
4.1.2.3.4 Data Pribadi 81
4.1.2.3.5 Pengaturan 82
4.1.2.3.6 Pencarian Anggota 83
4.1.2.4 Halaman *Administrator* 84
4.1.2.4.1 Halaman Beranda Admin 84
4.1.2.4.2 Halaman Berita Admin 85
4.1.2.4.3 Halaman Pengumuman Admin 86
4.1.2.4.4 Halaman Forum Admin 86
4.1.2.4.5 Halaman Buku Tamu Admin 87
4.1.2.4.6 Halaman Publikasi Admin 88
4.1.2.4.7 Halaman Galeri Admin 88
4.1.2.4.8 Halaman Data Induk 89
4.1.2.4.9 Halaman Tambah Anggota 90
4.1.2.4.10 Halaman Lihat Kelas 91
4.1.2.4.11 Halaman Lihat Jadwal 91
4.1.2.4.12 Halaman Tagihan 92
4.1.2.4.13 Halaman Akademik 93
4.1.2.4.14 Halaman Pencarian 93

4.1.2.4.15 Halaman Ganti *Username* 94
4.1.2.4.16 Halaman Ganti *Password* Admin 95
4.2 Pengujian Program 95
4.2.1 Analisis Kinerja Sistem. 96
4.2.1.1 Penanganan Kesalahan 96
4.2.1.2 Pengujian dan Analisis 99
4.3 Perbandingan *Website* AJAX dengan *Website* Tradisional 106
4.3.1 Perbandingan Proses *Load* Halaman 106
4.3.2 Perbandingan Proses Validasi Masukan Data 109
4.3.3 Perbandingan Proses *Login* 115
4.3.4 Perbandingan Proses Pencarian 120

BAB V SIMPULAN DAN SARAN 125
5.1 Simpulan 125
5.2 Saran 125

DAFTAR PUSTAKA 126

DAFTAR GAMBAR

Gambar 2.1	Request dan Response Pada World Wide Web	8
Gambar 2.2	*Server-side Scripting* dan *Client-side Scripting*, (a) *Server-side Scripting* dengan PHP. (b) *Client-side Scripting* dengan Javascript.	13
Gambar 2.3	Perbedaan Arsitektur Traditional Web Application Dengan AJAX Web Application Model.	16
Gambar 2.4	Perbedaan Interaksi Aplikasi Web Tradisional Dengan Aplikasi Web Dengan AJAX	17
Gambar 3.1	Diagram Konteks *Website* Yayasan Manarul Huda	32
Gambar 3.2	DFD level 1	33
Gambar 3.3	DFD Level 2 Proses Manipulasi Data Anggota	34
Gambar 3.4	DFD Level 2 Proses Manipulasi Data Forum	35
Gambar 3.5	DFD Level 2 Proses Pencarian	36
Gambar 3.6	*Flowchart* Proses Validasi Dengan Penerapan Teknologi AJAX	38
Gambar 3.7	Flowchart Proses *Login* Dengan Penerapan Teknologi AJAX	39
Gambar 3.8	Flowchart Pencarian Dengan Penerapan Teknologi AJAX	40
Gambar 3.9	Relasi Antar Tabel	48
Gambar 3.10	Rancangan Halaman Pengunjung	49
Gambar 3.11	Rancangan Halaman Tambah Topik	50
Gambar 3.12	Rancangan Halaman Tambah Komentar	50
Gambar 3.13	Rancangan Halaman Tambah Buku Tamu	51
Gambar 3.14	Rancangan Halaman Pencarian	51
Gambar 3.15	Rancangan Halaman Anggota	52
Gambar 3.16	Rancangan Halaman Pengaturan	53
Gambar 3.17	Rancangan Halaman Pencarian Anggota	53
Gambar 3.18	Rancangan Halaman Admin	54
Gambar 3.19	Rancangan Halaman Tambah Anggota	55
Gambar 3.20	Rancangan Halaman Tambah Kelas	56
Gambar 3.21	Rancangan Halaman Tambah Tagihan	56
Gambar 3.22	Rancangan Halaman Tambah Jadwal Pelajaran	57
Gambar 3.23	Rancangan Halaman Tambah Berita	58

Gambar 3.24 Rancangan Halaman Pengumuman 58
Gambar 3.25 Rancangan Halaman Tambah Publikasi 59
Gambar 3.26 Rancangan Halaman Tambah Galeri 59
Gambar 3.27 Rancangan Halaman Pencarian 60
Gambar 3.28 Rancangan Antarmuka Halaman *Login* 61
Gambar 4.1 Halaman Beranda (Halaman Depan). 63
Gambar 4.2 Halaman Berita 64
Gambar 4.3 Halaman Pengumuman 65
Gambar 4.4 Halaman Forum 66
Gambar 4.5 Halaman Tambah Topik 67
Gambar 4.6 Halaman Tambah Komentar 68
Gambar 4.7 Halaman Buku Tamu 69
Gambar 4.8 Halaman Tambah Buku Tamu 70
Gambar 4.9 Halaman Pencarian 71
Gambar 4.10 Halaman Yayasan 72
Gambar 4.11 Halaman Pendidikan Salafiyah 73
Gambar 4.12 Halaman Madrasah Tsanawiyah 74
Gambar 4.13 Halaman Madrasah Aliyah 75
Gambar 4.14 Halaman Publikasi 76
Gambar 4.15 Halaman Galeri 77
Gambar 4.16 Halaman *Login* 78
Gambar 4.17 Halaman Status 79
Gambar 4.18 Halaman Akademik 80
Gambar 4.19 Halaman Tagihan 81
Gambar 4.20 Halaman Data Pribadi 82
Gambar 4.21 Halaman Pengaturan 83
Gambar 4.22 Halaman Pencarian Anggota 84
Gambar 4.23 Halaman Beranda Admin 85
Gambar 4.24 Halaman Berita Admin 85
Gambar 4.25 Halaman Pengumuman Admin 86
Gambar 4.26 Halaman Forum Admin 87
Gambar 4.27 Halaman Buku Tamu Admin 87

Gambar 4.28 Halaman Publikasi Admin 88
Gambar 4.29 Halaman Galeri Admin 89
Gambar 4.30 Halaman Data Induk 90
Gambar 4.31 Halaman Tambah Anggota 90
Gambar 4.32 Halaman Lihat Kelas 91
Gambar 4.33 Halaman Lihat Jadwal 92
Gambar 4.34 Halaman Tagihan 92
Gambar 4.35 Halaman Akademik 93
Gambar 4.36 Halaman Pencarian 94
Gambar 4.37 Halaman Ganti *Username* 94
Gambar 4.38 Halaman Ganti *Password* Admin 95
Gambar 4.39 Tampilan Pesan Kesalahan Jika Nama Tidak Diisi 96
Gambar 4.40 Tampilan Pesan Kesalahan Jika No.Induk Diisi String. 97
Gambar 4.41 Pesan Kesalahan *Username* Telah Ada 98
Gambar 4.42 Pesan Kesalahan Data Tidak Ditemukan 98
Gambar 4.43 Message Box Kesalahan Hak Akses 99
Gambar 4.44 Antar Muka Masukan *Login* Username dan *Password* Cocok. 100
Gambar 4.45 Antar Muka Masukan *Login Username* Dan *Password* Tidak Cocok. 100
Gambar 4.46 Antar Muka Masukan *Login Password* Kosong 101
Gambar 4.47 Antar Muka Hasil Masukan Tambah Anggota 102
Gambar 4.48 Antar Muka Hasil Masukan Kelas 102
Gambar 4.49 Antar Muka Hasil Masukan Data Topik Forum 103
Gambar 4.50 Antarmuka Hasil Masukan Data Komentar Forum 104
Gambar 4.51 Antar Muka Masukan Berita 105
Gambar 4.52 Antar Muka Masukan Galeri 105
Gambar 4.53 Halaman Index.php Yang Telah Diisikan Forum.php ke dalam tag id "*content_hal_depan*" 107
Gambar 4.54 Tampilan Masukan Data Tambah Anggota *Username* Telah Ada 114
Gambar 4.55 Tampilan Masukan Tambah Anggota Sesuai 115
Gambar 4.56 Tampilan *Login Password* Harus Diisi 119
Gambar 4.57 Tampilan *Login* Masukan Data Salah 119
Gambar 4.58 Tampilan Validasi *Login* Masukan *Username* dan *Password* cocok 120

Gambar 4.59 TampilanPencarian Data Tidak Ditemukan 123
Gambar 4.60 Pencarian Data Ditemukan Dengan Kata Kunci "r" 124

DAFTAR TABEL

Tabel 2.1 Status Kode Respon Dari *Server.* 10
Tabel 3.1 Tabel User 41
Tabel 3.2 Tabel Data_anggota 41
Tabel 3.3 Tabel Kelas 42
Tabel 3.4 Tabel Data Kelas 42
Tabel 3.5 Tabel Kehadiran 43
Tabel 3.6 Tabel Jadwal 43
Tabel 3.7 Tabel Tagihan 44
Tabel 3.8 Tabel Berita 44
Tabel 3.9 Tabel Pengumuman 44
Tabel 3.10 Tabel Publikasi 45
Tabel 3.11 Tabel Galeri 45
Tabel 3.12 Tabel Forum 45
Tabel 3.13 Tabel Komentar 46
Tabel 3.14 Tabel Bk_tamu 46
Tabel 3.15 Tabel Snapshot 47

DAFTAR LISTING

Listing 2.1	Permintaan Pengguna (*Client Request)*	12
Listing 2.2	Jawaban *Server* (*Server Response*).	12
Listing 2.3	Contoh *Tag* HTML sederhana.	14
Listing 2.4	Contoh Javascript Pada Dokumen HTML	18
Listing 2.5	Bentuk XML Sederhana.	19
Listing 2.6	Contoh Pembuatan Obyek XMLHttpRequest.	21
Listing 2.7	Contoh Pengambilan Dokumen XML Dengan XMLHttpRequest.	24
Listing 2.8	Contoh Peng-*update*-an Suatu Tag Id HTML	24
Listing 4.1	Tag Event OnClick=" tampil_halaman_ajax()"	106
Listing 4.2	Script Function tampil_halaman_ajax()	108
Listing 4.3	Tag Event onkeyup="validasi()" Pada Proses Tambah Anggota	109
Listing 4.4	Script Function validasi() Pada Proses Tambah Anggota	110
Listing 4.5	Modul PHP Validasi Tambah Anggota	113
Listing 4.6	Tag OnKeyUp password_cek()	116
Listing 4.7	Function password_cek() dan user_cek()	117
Listing 4.8	Modul PHP *Login*	118
Listing 4.9	Tag OnKeyUp cari_anggota()	121
Listing 4.10	Function cari_anggota()	121
Listing 4.11	Modul PHP Pencarian Anggota	122

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada awalnya halaman web adalah statis, pengguna meminta halaman web, dan kemudian *server* mengirimkannya, tidak ada yang berubah. *Website* hanya sebuah salinan teks elektronik yang dapat didistribusikan melalui internet. Solusi awal untuk membuat *website* lebih dinamis yaitu dengan CGI (*common gateway interface*), CGI memperbolehkan *programmer* untuk membuat program yang akan dieksekusi oleh *server* ketika ada permintaan dari pengguna.

Pada saat yang hampir bersamaan, Netscape membuat bahasa *scripting* yang sekarang disebut *Javascript*. Tujuan awal dari pembuatan javascript adalah untuk memodifikasi *tag* HTML secara langsung dan berjalan di sisi *browser*, sehingga menyediakan interaksi yang lebih baik bagi pengguna.

Pada saat ini kebutuhan *website* tidak lagi menjadi alat penyedia informasi yang statis, dalam perkembangannya *website* dituntut agar lebih dinamis. AJAX (*Asynchronous Javascript And XML*) memberikan solusi agar *website* lebih responsif dan dinamis. Dengan penerapan teknologi AJAX, interaksi *website* dengan *server* dilakukan secara *background* tanpa perlu me-*refresh* ulang halaman secara keseluruhan, sehingga menjadikan *website* akan lebih responsif karena setiap aksi dari pengguna akan langsung direspon pada saat itu juga.

Untuk meningkatkan pelayanan dan mutu kepada anggota pada khususnya dan masyarakat pada umumnya, Yayasan Manarul Huda akan membangun sebuah *website* yang menjadi portal resmi dari yayasan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas dapat dirumuskan permasalahan yang akan diselesaikan adalah bagaimana membangun sebuah *website* yang lebih responsif dan dinamis dengan menerapkan keunggulan dari teknologi AJAX, dimana *website* ini akan digunakan sebagai portal resmi dari Yayasan Manarul huda.

1.3 Batasan Masalah

Batasan-batasan masalah dalam pembuatan tugas akhir ini adalah:

1. *Website* ini dibangun dengan penerapan teknologi AJAX.
2. Sistem keamanan hanya dibatasi pada autentifikasi *username* dan *password.*
3. *Website* ini dibuat untuk diaplikasikan sebagai portal resmi dari Yayasan Manarul Huda.
4. Dalam implementasi dan pembahasannya, tidak semua bagian dalam *website* ini dibangun dengan penerapan teknologi AJAX.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membuat sebuah *website* yang lebih responsif dengan penerapan teknologi AJAX, pada *client-side* akan dihasilkan *website* yang lebih memudahkan pengguna, karena pengguna tidak perlu menunggu halaman untuk di-*refresh* terlebih dahulu. Sedangkan pada *server-side*, *website* ini lebih dinamis dikarenakan interaksi dengan *server* dilakukan secara *background* sehingga setiap aksi dari pengguna akan langsung direspon oleh *server* pada saat itu juga.

1.5 Manfaat Penelitian

Manfaat yang ingin dicapai dalam pembuatan tugas akhir ini adalah:

1. Bagi pengguna, khususnya bagi pihak Yayasan Manarul Huda, dengan tugas akhir ini dapat dihasilkan portal resmi yayasan yang responsif dan dinamis, serta dibangun berdasarkan aturan-aturan dari Yayasan Manarul Huda.

2. Bagi penulis, dapat memahami pembuatan *website* pada umumnya, dan penggunaan teknologi AJAX pada khususnya, yang mana dengan teknologi ini, dapat dihasilkan *website* yang lebih unggul, baik dalam interaksi dengan pengguna, maupun dari sisi komunikasi dengan *server*.

3. Bagi kalangan akademis, tugas akhir ini di harapkan dapat memberikan wawasan tentang pembangunan *website* khususnya dengan penerapan teknologi AJAX.

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data

Metode pengumpulan data adalah metode yang digunakan untuk mengumpulkan data yang diperlukan dalam penelitian, yaitu melalui studi literatur dengan mempelajari materi yang diperoleh yang berkaitan dengan topik tugas akhir melalui media seperti buku, artikel, tulisan-tulisan pada situs internet maupun media informasi lainnya, serta berkonsultasi dengan narasumber yang lebih mendalami topik yang berkaitan.

1.6.2 Metode Pengembangan Sistem

Metode pengembangan sistem disusun berdasarkan hasil dari data yang sudah diperoleh. Metode ini meliputi:

1. Analisis

Pada tahap analisis, yang dilakukan adalah mengidentifikasikan kebutuhan-kebutuhan yang diperlukan dan mengumpulkan data yang sesuai dengan kebutuhan sistem yang akan dibangun.

2. Perancangan (*design*)

Tahap perancangan ini meliputi perancangan proses, perancangan basisdata, struktur *website* dan perancangan antarmuka.

3. Pembuatan Program (*coding*)

Hal yang dilakukan pada tahap ini adalah menerjemahkan rancangan ke dalam bahasa pemrograman tertentu. Penulisan kode program sesuai dengan *design* yang dibuat, sehingga dapat menghasilkan aplikasi yang bermanfaat bagi pengguna.

4. Pengujian (*testing*)

Dalam tahap ini, hal yang dilakukan adalah melakukan pengujian terhadap aplikasi yang telah dibuat. Aplikasi akan diuji apakah telah sesuai dengan apa yang diinginkan, dan memperbaiki *bug* yan ditemukan. Tahap ini dilakukan untuk menjamin kualitas dari perangkat lunak yang dibuat.

5. Implementasi

Dalam tahap ini hal yang dilakukan adalah melakukan peng-*install*-an sistem pada instansi yang menggunakan perangkat lunak yang telah dikembangkan.

1.7 Sistematika Penulisan

Untuk memudahkan penulisan tugas akhir agar lebih terperinci, maka dibuat sistematika penulisan sebagai berikut:

Bab I Pendahuluan

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

Bab II Landasan Teori

Berisi latar belakang dan tujuan pembangunan *website* Yayasan Manarul Huda, serta berisi uraian tentang konsep teknologi AJAX dan bagian-bagian yang termasuk di dalamnya, yaitu: Javascript, XML dan obyek XMLHttpRequest.

Bab III Metodologi

Bab ini berisi tentang metodologi dan perancangan perangkat lunak. Pada bagian metodologi memuat uraian tentang analisis kebutuhan perangkat lunak, metode analisis kebutuhan perangkat lunak yang berupa analisis kebutuhan proses, analisis kebutuhan masukan, analisis kebutuhan keluaran, kebutuhan perangkat lunak, kebutuhan perangkat keras dan kebutuhan antar muka. Pada bagian perancangan perangkat lunak membahas tentang metode perancangan yang digunakan, hasil perancangan berupa perancangan diagram arus data, flowchart, perancangan penerapan teknologi AJAX, dan perancangan tabel basis data.

Bab IV Hasil dan Pembahasan

Bab ini berisi hasil dan pembahasan yang memuat uraian tentang implementasi perangkat lunak, pengujian dan analisis kinerja perangkat lunak yang meliputi penanganan kesalahan dan pembahasan pengujian pada proses-proses yang ada, serta memuat perbandingan *website* tradisional dengan *website* AJAX, dilihat dari proses-proses yang mengimplementasikan penerapan teknologi AJAX tersebut.

Bab V Penutup

Bagian ini memuat kesimpulan dari proses pengembangan perangkat lunak, baik pada tahap analisis kebutuhan perangkat lunak, perancangan implementasi dan terutama pada analisis kinerja perangkat lunak. Selain itu pada bagian ini juga memuat tentang saran yang perlu diperhatikan untuk pengembangan perangkat lunak tersebut.

BAB II

LANDASAN TEORI

2.1 Yayasan Manarul Huda

Yayasan Manarul Huda merupakan yayasan yang bergerak di bidang pendidikan Islam, bertempat di Tasikmalaya Jawa Barat. didirikan pada tanggal 7 Oktober 1977 oleh K.H. Endin Saepudin. Pada awal terbentuknya, yayasan ini merupakan sebuah Pondok Pesantren yang mengkhususkan pada pendidikan *salafiyah*. Untuk meningkatkan pelayanan kepada masyarakat, setelah hampir 20 tahun berdiri, tepatnya pada tanggal 7 Februari 1997, Pondok Pesantren Manarul Huda resmi menjadi Yayasan Pondok Pesantren Manarul Huda. Didaftarkan dengan akta notaris Heri Hendriyana S. H. No. 03 – 1997.

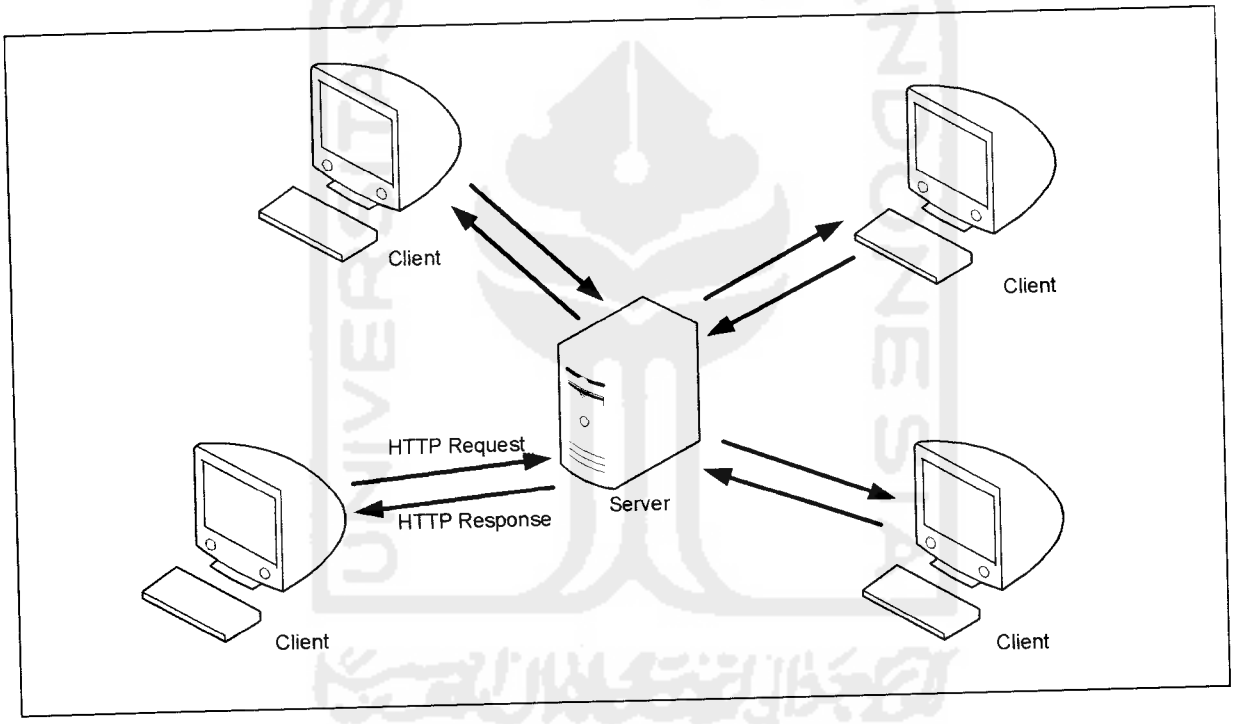
Setelah resmi menjadi yayasan, langkah pertama dari Yayasan Manarul Huda adalah dengan mendirikan lembaga pendidikan formal tingkat menengah MTs (*Madrasah Tsanawiyah*) pada tahun 1997. Kemudian selanjutnya pada tahun 2002 Yayasan Manarul Huda mendirikan lembaga pendidikan formal tingkat atas MA (*Madrasah Aliyah*). Selain lembaga-lembaga pendidikan tersebut, pendidikan *salafiyah* tetap mendapatkan porsi utama, karena tujuan awal pendirian yayasan ini adalah sebagai Pondok Pesantren.

Setelah hampir satu dekade menjadi sebuah yayasan, untuk meningkatkan pelayanan kepada anggota dan menunjang operasional dari yayasan, Yayasan Manarul Huda akan membangun sebuah portal resmi yayasan, yang diharapkan memberikan kontribusi dalam pencapaian visi dan misi yayasan, serta menjadikan Yayasan Manarul Huda lebih baik.

2.2 World Wide Web (WWW)

2.2.1 Cara Kerja World Wide Web

World wide web beroperasi menggunakan prinsip jaringan *client/server*. Setelah menerima *request* yang dikirim oleh pengguna, *server* merespon dengan memberikan yang diminta oleh pengguna dalam format yang dapat di mengerti dan ditampilkan oleh *browser* pengguna. Ilustrasi hubungan ini dapat dilihat dari gambar 2.1 berikut.



Gambar 2.1 Request dan Response Pada World Wide Web [BAL06].

a. Request

Setelah membuka koneksi ke *server* yang dimaksud, HTTP *client* melewatkan suatu *request* dengan format sebagai berikut:

- Baris pembuka

- *Header* (tidak harus)
- Sebuah baris kosong
- Pesan (tidak harus)

Baris pembuka biasanya dibagi atas tiga bagian; *method* yang digunakan, nama *path* yang diminta dan versi HTTP yang sedang digunakan, Contohnya;

```
Get /irham/testpage.html HTTP/1.0.
```

Pada baris ini pengguna memberitahukan *server* bahwa pengguna sedang mengirim suatu HTTP *Request* dengan tipe GET, meminta */irham/testpage.html* dan *request* dikirim dengan menggunakan HTTP versi 1.0.

Baris *header* digunakan untuk mengirim informasi mengenai *request* atau mengenai data yang sedang dikirim di bagian pesan. Pasangan parameter dan nilai dikirim per baris, parameter dan nilai dipisahkan oleh tanda titik dua (:). Contohnya;

```
User-Agent: [nama program pengirim request].
```

Untuk IE versi 5.5 baris *header*-nya adalah:

```
User-Agent: Mozilla/4.0 {compatible;MSIE 5.5; Windows NT 5.0}.
```

Request header yang umum lainnya adalah *accept*, yang menyatakan informasi-informasi yang mampu diterima sebagai respon dari *server*. Contohnya;

```
Accept: text/plain, text/html.
```

Header accept di atas menyatakan bahwa pengguna mampu menerima suatu teks biasa atau HTML sebagai *response* dari *server*. Dengan kata lain, pengguna tidak bisa menerima selain berkas teks biasa atau berkas HTML.

b. Response

Untuk menjawab permintaan dari pengguna, *server* memunculkan suatu respon. Baris pertama dari respon disebut sebagai baris status. Pada baris tersebut *server* menampilkan versi HTTP, memberikan sebuah kode status respon dan sebuah pesan pendek yang dikenal sebagai *reason phrase*. Contohnya;

```
HTTP/1.0 200 OK.
```

Status kode respon dari *server* ada bermacam-macam, beberapa kode yang umum diantaranya dirangkum dalam tabel 2.1 dibawah ini.

Kode Status	**Arti**
200 - OK	*Request* berhasil
204 - No Content	Tidak ada data dalam dokumen
401 - Not Authorized	*Request* memerluksn autentifikasi dari pengguna
403 - Forbidden	*Server* menolak *request*
404 - Not Found	*Resource* yang diminta tidak ada di *server*
408 - Request Timeout	*Client* gagal mengirim *request* dalam waktu yang sudah ditentukan oleh *server*
500 - Server Error	Ada kesalahan pada konfigurasi *server* atau sejenisnya

Tabel 2.1 Status Kode Respon Dari *Server.*

2.3 HyperText Transfer Protocol (HTTP)

HyperText transfer Protocol (HTTP) merupakan protokol pada lapis aplikasi yang paling popular saat ini. HTTP adalah sebuah metode untuk megirimkan atau menyampaikan pesan pada *world wide web*. Awalnya digunakan sebagai cara untuk mempublikasikan atau mengambil halaman HTML [ANO01].

Pengembangan HTTP dikoordinasikan oleh dua lembaga internasional, yaitu *World Wide Web Consortium* (W3C) dan *Internet Engineering Task Force* (IETF). Dua lembaga tersebut bertugas untuk menertibkan dokumen *request from comment* (RFC) seputar standarisasi web. HTTP/1.1, yang didefinisikan pada RFC 2616, merupakan versi HTTP yang paling banyak digunakan saat ini.

HTTP adalah protokol *request/response* (permintaan/jawaban) antara pengguna dan *server*. Pengguna, yang merupakan asal dari permintaan (semisal *browser*) disebut sebagai *user-agent*. *Server* tujuan, yang menyimpan atau membuat *resource* semisal berkas HTML dan gambar, disebut sebagai *server* asal. Di antara *user-agent* dan *server* asal mungkin terdapat beberapa perantara, semisal *proxy* dan *gateway*.

Pengguna HTTP memulai *request* dengan membuat koneksi TCP pada *port* tertentu ke *remote host* (biasanya pada *port* 80). *Server* HTTP yang mendengarkan pada *port* tersebut menunggu pengguna mengirimkan pesan permintaan.

Setelah pesan permintaan diterima, *server* menukarkan dengan sebuah status, misal "`HTTP/1.1 200 OK`", dan pesan itu sendiri, bagian yang mungkin merupakan berkas yang diminta, pesan kesalahan atau informasi lainnya.

Resource yang akan diakses melalui HTTP diidentifikasikan oleh *Uniform Resource Identifier* (URI) dan skema *http:* atau *https:*.

Di bawah ini adalah contoh komunikasi antara pengguna dan *server* yang terjadi pada protokol HTTP. *Server* berjalan pada www, contoh .com, *port* 80.

Permintaan pengguna (permintaan dari pengguna selalu diakhiri dengan baris kosong) ditunjukan pada Listing 2.1 .

```
GET /index.html HTTP/1.1
Host: www.example.com
```

Listing 2.1 Permintaan Pengguna (*Client Request)* [ANO01].

Jawaban *server* dari permintaan pengguna ditunjukkan pada listing 2.2.

```
HTTP/1.1 200 OK
Date: Mon, 23 August 2001 22:38:34 GMT
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1h6-3elcb03b"
Accept-Ranges: bytes
```

Listing 2.2 Jawaban *Server* (*Server Response*) [ANO01].

2.4 *Client-side Scripting* dan *Server-side Scripting*

Sebuah halaman *website* ditampilkan dalam sebuah *web browser*, halaman ini merupakan hasil dari penerjemahan sebuah "kode" yang diterima oleh *web browser*. Pada umumnya kode tersebut bernama HTML (*Hypertext Markup Language*), namun tidak hanya itu, masih banyak yang telah dapat di-*support* oleh *web browser*, salah satunya adalah Javascript [JON06].

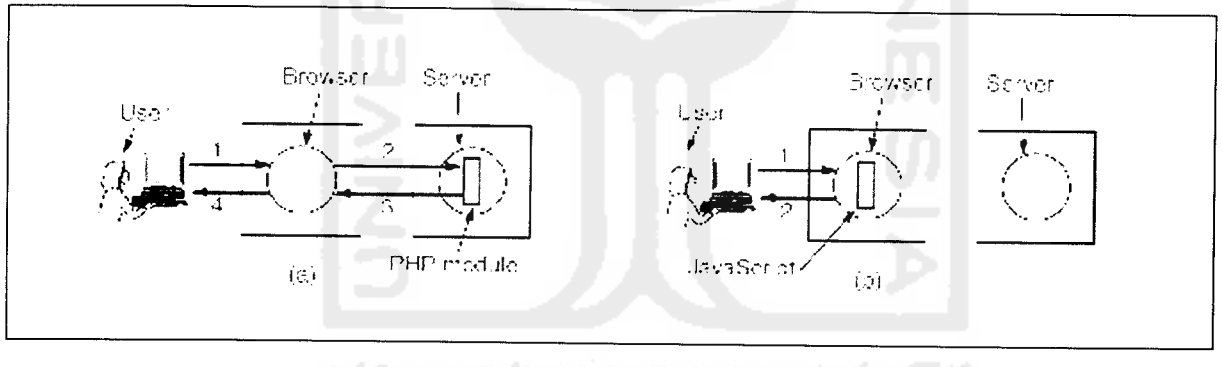
"Kode" yang diterima dan kemudian dapat diterjemahkan dan dieksekusi oleh *web browser*, dapat disebut sebagai *client-side scripting*. Tempat di mana halaman web tersebut berada, dapat disebut sebagai *web server*.

Ketika pengunjung mengetik sebuah alamat di *web browser*, permintaan akan sebuah halaman telah terkirim, dan kemudian *web server* mengirim halaman tersebut ke pengunjung

yang berupa "kode yang berupa *client-side scripting*" yang kemudian akan tampil pada *web browser.*

Kemudian untuk memanipulasi "kode yang akan dikirim ke *web browser*", dapat menggunakan *server-side scripting* yang dapat diterjemahkan oleh *web server.* Ketika sebuah halaman di-*request* maka *server-side scripting* pada *web browser*, akan meng-*generate* sebuah "kode" dan kemudian diterima oleh *web browser*, *web browser* mengenali "kode" tersebut sebagai *client-side scripting* yang kemudian ia terjemahkan untuk ditampilkan pada layar *web browser.*

Maka, *client-side scripting* merupakan program yang dieksekusi oleh *web browser*, dan *server-side scripting* merupakan program yang dijalankan pada *web server*, biasanya program tersebut dapat mengakses *database* yang tersimpan di *web server.* Ilustrasi *client-side scripting* dan *server-side scripting* dapat dilihat pada gambar 2.4 dibawah ini.



Gambar 2.2 *Server-side Scripting* dan *Client-side Scripting*, (a) *Server-side Scripting* dengan PHP. (b) *Client-side Scripting* dengan Javascript.

2.5 HTML (Hypertext Markup Language)

Hyper Text Markup Language merupakan kepanjangan dari HTML yang merupakan salah satu dari *Client-side scripting*, file HTML adalah sebuah *text file* yang berisi *tags* (< >),

tag-tag tersebut memberitahukan *web browser* bagaimana cara menampilkan halaman tersebut, contoh *tag* HTML sederhana terlihat pada listing 2.3.

```
<html>
<head>
<title>test page</title>
<body>
     test
</body>
</head>
</html>
```

Listing 2.3 Contoh *Tag* HTML sederhana.

2.6 AJAX (Asynchronous Javascript and XML)

AJAX adalah singkatan dari "*Asynchronous Javascript and XML*", dan terdiri dari serangkaian teknologi dengan berbagai kemampuan, yaitu : Javascript, XML dan sebuah *method* komunikasi asinkron antara pengguna dan *server*.

Istilah AJAX dicetuskan oleh Garret (2005) untuk menyebut suatu teknik pada aplikasi web yang memanfaatkan alat bantu yang pada umumnya sudah tertanam di dalam *web browser* untuk memungkinkan terjadinya pemutakhiran (*update*) potongan data secara *realtime* dan interaksi antara aplikasi dan pengguna. Pada teknologi AJAX, javascript menangkap isyarat, gerak serta aksi. Javascript menggunakan jalur komunikasi pada *server* (obyek javascript dengan nama XMLHttpRequest) untuk memanggil *method* yang tersimpan pada *server* dan menggunakan XML sebagai mekanisme pengiriman data [GAR05].

Meskipun istilah AJAX baru dicetuskan pada tahun 2005, namun bukan berarti teknik serupa belum pernah digunakan sebelumnya. Pada saat mengembangkan teknologi *Remote Scripting*, Microsoft telah menggunakan teknik serupa untuk melakukan pemuatan isi sebuah

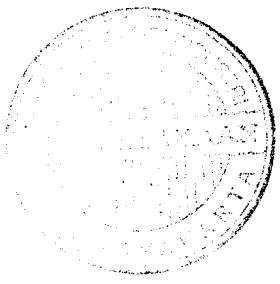
halaman web secara asinkron tanpa melakukan pemuatan ulang secara penuh [GAR05]. Jauh sebelum adanya *Remote Scripting*, ada berbagai macam teknik semisal elemen IFRAME di Internet Explorer , IE 3 (diperkenalkan pada tahun 1996) dan elemen LAYER di Netscape 4 (diperkenalkan pada tahun 1997). Kedua elemen tersebut mempunyai atribut *src* yang memungkinkan untuk memuat URL eksternal manapun. Dengan memuat halaman yang berisi javascript yang memodifikasi halaman induk, efek Serupa AJAX dapat dicapai.

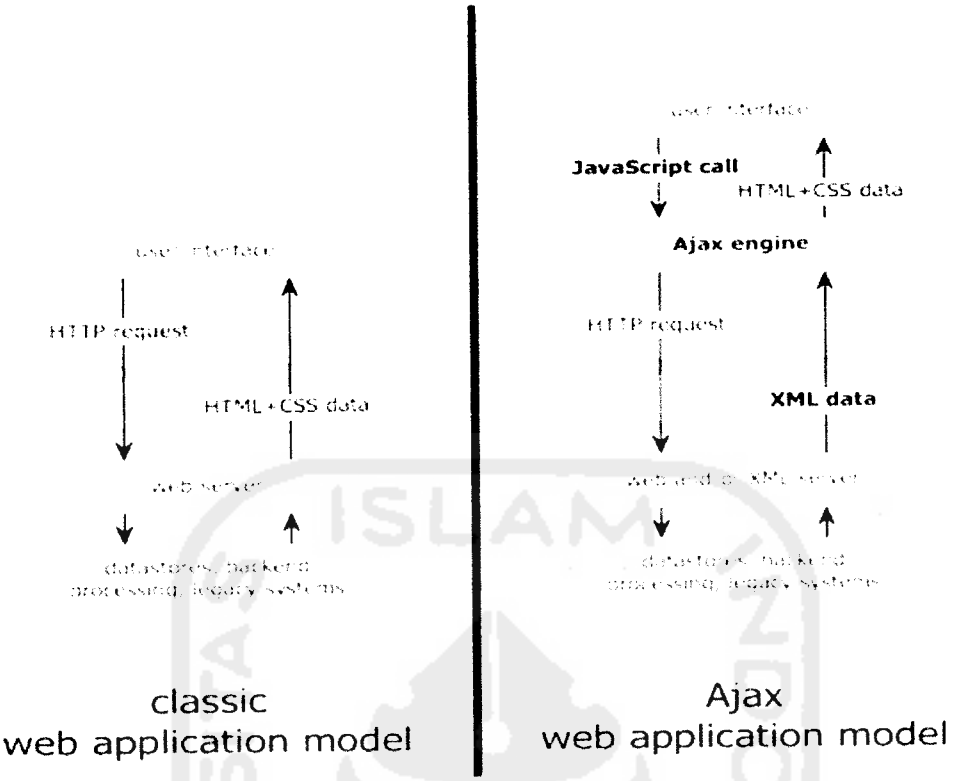
Remote scripting dari Microsoft (diperkenalkan pada tahun 1998) berperan sebagai pengganti yang lebih baik dibandingkan dengan solusi-solusi sebelumnya. Pada *remote scripting*, *client-side* menggunakan javascript untuk berkomunikasi dengan sebuah *applet java* yang bertugas mengambil data. Teknik ini bisa berjalan pada IE 4 dan Netscape Navigator 4 dan versi setelahnya. Pada IE 5, Microsoft menciptakan objek XMLHttpRequest, dan yang pertama kali memanfaatkan teknik ini adalah *Microsoft Outlook Web Access* yang tersedia pada *Microsoft Exchange Server 2000.*

Karena XMLHttpRequest sudah diterapkan pada banyak *browser*, teknik-teknik alternatif menjadi jarang digunakan. Meskipun begitu, teknik-teknik alternatif tersebut masih tetap digunakan untuk menjaga kompabilitas dengan situs web dan aplikasi lama.

Adanya *layer* tambahan pada aplikasi web dengan teknologi AJAX menjadikan arsitekturnya sedikit berbeda jika dibandingkan dengan aplikasi web tradisional. *Layer* tambahan ini biasa disebut AJAX-*engine*, dan berfungsi sebagai perantara antara pengguna dengan *server* [GAR05].

Gambar 2.3 dibawah menunjukan perbedaan arsitektur web tradisional dengan arsitektur web yang menggunakan teknologi AJAX [GAR05].

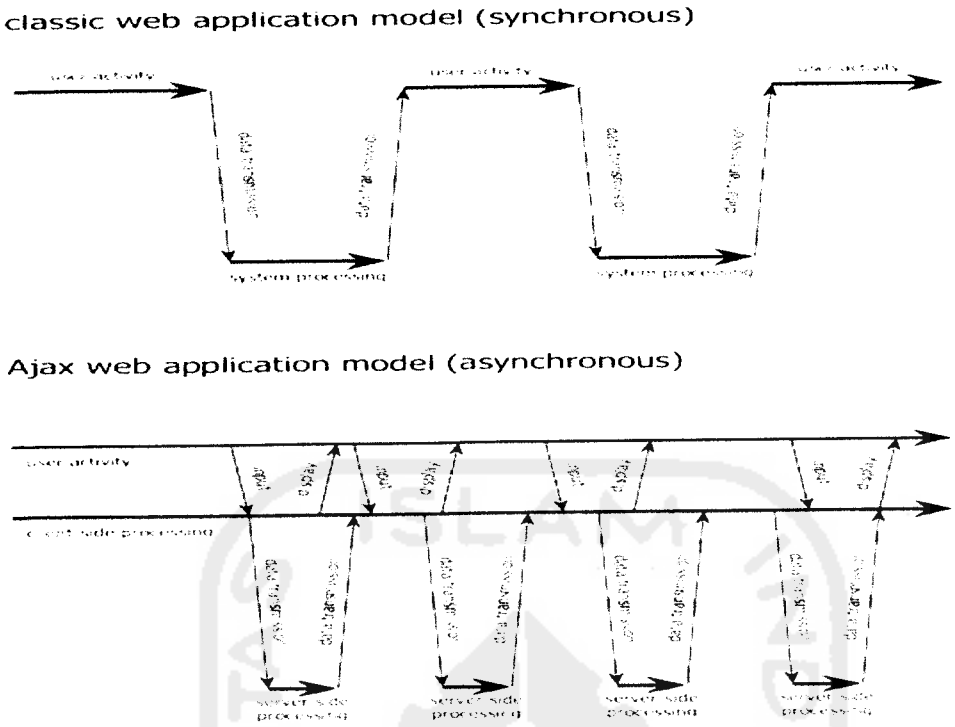




Gambar 2.3 Perbedaan Arsitektur Traditional Web Application Dengan AJAX Web Application Model.

AJAX *engine* bertanggung jawab untuk mengelola tampilan (halaman web) dan berkomunikasi dengan *server* atas nama pengguna. Hal ini memungkinkan terjadinya interaksi antara pengguna dengan aplikasi secara asinkron, terpisah dari aktivitas komunikasi dengan *server*. Oleh karena itu, tidak ada lagi halaman kosong atau ikon jam pasir selama pengguna menunggu *server* menyelesaikan pengolahan data.

Gambar 2.4 menunjukan perbedaan interaksi antara aplikasi web tradisional dengan aplikasi web dengan teknologi AJAX [GAR05].



Gambar 2.4 Perbedaan Interaksi Aplikasi Web Tradisional Dengan Aplikasi Web Dengan AJAX.

2.6.1 Javascript

Javascript adalah bahasa *scripting* berbasis *prototype* dengan aturan penulisan yang mirip dengan bahasa C [ANO01]. Javascript muncul pada tahun 1995 yang dibangun oleh Netscape, pada awalnya pembuatan javascript bertujuan untuk membantu para *web developer* untuk secara dinamis dapat merubah *tag* HTML dalam halaman web agar lebih memberikan interaksi yang lebih baik dengan *pengguna*. Javascript bergantung pada lingkungan induk di mana javascript tersebut dipasang. Teknologi web, khususnya *browser*, adalah salah satu di antara banyak lingkungan induk dimana javascript digunakan.

Salah satu kegunaan utama javascript pada *browser* adalah untuk menulis fungsi-fungsi yang membantu HTML berinteraksi dengan *Document Object Model* (DOM) di dalam

halaman tersebut untuk melakukan tugas yang tidak mungkin dilakukan oleh HTML sendiri. Fungsi yang ditulis dalam javascript ditempelkan langsung di dalam dokumen HTML atau dipanggil dari *file* javascript yang terpisah dari dokumen HTML. Salah satu contoh kegunaan javascript adalah untuk memeriksa masukan dari pengguna sebelum dikirim ke *server*.

Contoh lain kegunaan javascript adalah untuk melakukan permintaan lain halaman web ke *web server* atau layanan pada web service setelah halaman web dimuat. Informasi yang didapat dari permintaan tersebut kemudian digabungkan pada halaman web yang sudah ada sehingga dapat ditampilkan. Teknik ini merupakan dasar dari AJAX. Berikut contoh javascript pada dokumen HTML pada listing 2.4.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
/>
<title>Test Page..</title>
<script language="javascript">
function gosearchAction(){

  document.form1.action="domain";
  document.form1.navigation.value="golookup";
  document.form1.goaction.value="true";
  document.form1.submit();
  return true;
 }
</script>
</head>
<body> This is Test Page..
</body>
</html>
```

Listing 2.4 Contoh Javascript Pada Dokumen HTML

2.6.2 Extensible Markup Language (XML)

Extensible Markup Language, disingkat dengan XML, menggambarkan kelas pada data obyek yang disebut XML-*documents*. XML adalah suatu profil aplikasi dari SGML

(*Standard Generalized Markup Language*), yang dalam pengembangannya, XML-*documents* harus dapat disesuaikan dengan SGML-*documents*. (catatan : HTML 4 juga merupakan salah satu aplikasi yang telah menyesuaikan SGML). [W3C04].

XML muncul bukan untuk menggantikan HTML, XML dan HTML didesain dengan tujuan yang berbeda. XML didesain untuk menjelaskan data yang ada dan berfokus pada ini datanya dan ini strukturnya. Sedangkan HTML didesain untuk bagaimana menampilkan sebuah informasi. XML tidak didesain untuk mengerjakan sesuatu, XML didesain untuk membuat struknya sendiri, menyimpan informasi dan kemudian mengirimnya. Untuk melihat bagaimana bentuk XML, dapat dilihat pada listing 2.5.

```
<note>
      <to>Aragorn</to>
      <from>Jani</from>
      <heading>Reminder</heading>
      <body>Don't forget me this weekend</body>
</note>
```

Listing 2.5 Bentuk XML Sederhana.

Listing 2.5 diatas merupakan contoh sebuah catatan kecil yang disimpan dalam XML, dalam catatan tersebut berisi informasi, walaupun secara sekilas terdapat *tag heading* dan *body*, tetapi XML tidak berbuat apa-apa, gambar diatas hanya sebuah teks biasa yang disimpan dalam XML, harus ada sebuah perangkat lunak yang dapat mengirim, menerima dan membacanya.

2.6.3 XMLHttpRequest

Keajaiban AJAX terjadi karena teknologi javascript, yaitu obyek XMLHttpRequest. Walaupun obyek ini belum dispesifikasikan secara formal pada teknologi javascript, tetapi *browser* terbaru yang diluncurkan telah men-*support*-nya.

XMLHttpRequest adalah sebuah obyek javascript yang digunakan untuk mengakses *web server* secara asinkron, jadi pengguna dapat terus melakukan aktifitasnya di sebuah halaman web tanpa me-*refresh* halaman tersebut, karena proses *request*-nya dilakukan secara *background*. Pada proses *request* ini, tentunya ada data yang akan dilewatkan untuk diproses di sisi *server*. Metode *request* yang dapat digunakan adalah "GET" atau "POST". Tentunya akan sangat mudah bagi *server* untuk membaca data yang di kirimkan oleh pengguna dengan menggunakan metode tersebut.

Pada implementasinya, untuk membuat sebuah obyek XMLHttpRequest cukup dengan meng-*instance class* XMLHttpRequest(), kecuali pada *browser* Internet Explorer, IE, Microsoft mengimplentasikannya pada sebuah *ActiveX Control*, sehingga harus menggunakan *ActiveX* tersebut untuk mengimplementasikan AJAX pada IE. Contoh pembuatan obyek XMLHttpRequest terlihat pada listing 2.6. [ANO07].

```
var xmlHttp=null;
try
 {
 // Firefox, Opera 8.0+, Safari
 xmlHttp=new XMLHttpRequest();
 }
catch (e)
 {
 //Internet Explorer
 try
  {
  xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
  }
 catch (e)
  {
  xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
 }

      xmlHttp.onreadystatechange=function(){

            loadpage(xmlHttp,containerid)
      }
      xmlHttp.open('GET',url,true)
      xmlHttp.send(null)
}
```

Listing 2.6 Contoh Pembuatan Obyek XMLHttpRequest.

Setelah obyek XMLHttpRequest dibuat, dapat dilakukan HTTP *request* mengunakan javascript dengan memanggil fungsi *open()* yang merupakan member dari *class* XMLHttpRequest. Fungsi ini mempunyai 3 parameter diantaranya:

- ***Type Request*** : Sebuah string yang menunjukkan dengan tipe *request* apa koneksi yang akan dilakukan.
- ***URL*** dimana kita akan mengirim *request* tersebut.
- ***Async*** : Sebuah *value* bertipe boolean yang menunjukkan apakah *request* akan diminta secara sinkron atau asinkron (bila diset "*true*" request secara *asynchronous*, "*false*" request secara *synchronous*).

Parameter terakhir *Async* sangat penting karena ia mengontrol javascript tentang bagaimana cara menangani *request*. Ketika ia di set *True* maka *request* akan dilakukan secara asinkon, artinya javascript akan tetap menjalankan kode salanjutnya tanpa menunggu respon dari *server*, tetapi jika di set *False* maka *browser* akan menunggu sampai ada respon sebelum melanjutkan eksekusi kode. Ini berarti jika respon membutuhkan waktu yang banyak, maka pengguna tidak dapat berinteraksi dengan *browser* sampai *request* selesai.

Selanjutnya setelah obyek XMLHttpRequest terbentuk, kita perlu mendefinisikan sebuah *eventhandler onreadystatechange*. Obyek XMLHttpRequest mempunyai sebuah properti dengan nama *readyState* yang berisi informasi tentang status respon dari *server*. Terdapat 5 jenis status respon dari *server*, yaitu:

- 0 (*Uninitialized*) : Obyek XMLHttpRequest sudah dibuat tetapi fungsi *open* belum dipanggil.
- 1 (*Loading*) : Fungsi *open* telah dipanggil tetapi *request* belum dikirim.
- 2 (*Loaded*) : *Request* telah dikirim.
- 3 (*Interactive*) : Sebagian respon telah diterima.
- 4 (*Complete*) : Semua data telah diterima.

Setiap ada perubahan *value* pada properti *readystate* maka *eventhandler onreadystatechange* akan di panggil. Adapun metode pada obyek XMLHttpRequest adalah sperti tabel 2.2 dibawah ini.

Method/Property	**Keterangan**
`Abort()`	Membatalkan *request* yang telah dikirim
`GetAllResponseHeaders ()`	Mendapatkan *header* respon sebagai string
`Open ( "method", "URL","asyncflag")`	Inisialiasi parameter *request*
`Send (content)`	Melakukan HTTP Request

`SetRequestHeader("Label", "value")`	Set pasangan label dan *value*-nya pada *header request*
`OnReadyStateChange`	Digunakan untuk mengeset fungsi *callback* yang meng-*handle* perubahan *request state*
`ReadyState`	Memberikan informasi status *request*
`ResponseText`	Mengembalikan respon *server* dalam format string
`ResponseXML`	Mengembalikan respon *server* dalam format XML
`Status`	Mengembalikan kode status *request*
`StatusText`	Mengembalikan pesan status *request*

Tabel 2.2 Metode Pada XMLHttpRequest [ANO07].

2.7 Perpaduan Antara AJAX dengan PHP

Menurut Rendell (yang dirangkum oleh Schwein), AJAX adalah sebuah solusi yang berjalan pada *browser*, yang mana javascript dipadukan dengan XML untuk menjaga *browser* tetap *update* ketika ada perubahan data (seperti *database*) atau ketika suatu bagian dari halaman tersebut perlu untuk dilakukan *refresh*. Kita bisa menggunakan AJAX sebagai *Front-end* dengan PHP sebagai *server-side scripting*, dengan digunakannya teknologi javascript, sebuah *client-side scripting* dapat secara langsung memanggil *server* untuk melakukan *query* ke dokumen XML maupun memanggil PHP sebagai *server-side scripting* untuk melakukan sesuatu (misal *query* ke *database*), yang kemudian data akan direspon oleh AJAX sebagai teks, HTML atau dokumen XML, yang kemudian dapat digunakan oleh javascript untuk meng-*update* atau memodifikasi DOM dari halaman HTML [MUR05].

Contoh penggunaan XMLHttpRequest untuk berkomunikasi dengan *server-side scripting* (dalam hal ini, PHP) dapat dilihat pada Listing 2.7.

```
<script language="javascript">
      Var page_request=false
      If (window.XMLHttpRequest)
            Page_request=new XMLHttpRequest()
      else if (window.ActiveObject)
            page_request=new ActiveObject("Microsoft.XMLHTTP")
      else
            return false

      page_request.onreadystatechange=function(){
            loadpage(page_request,containerid)
      }
      Page_request.open('GET',"testpage.php?get=xml",true)
      Page_request.send(null)
</script>
```

Listing 2.7 Contoh Pengambilan Dokumen XML Dengan XMLHttpRequest.

Pada listing 2.7 diatas, dihasilkan *variable XML document* yang menyimpan dokumen XML yang di-*generate* oleh *testpage.php?get=xml* dengan method "*GET*", sehingga akan ditampilkan halaman *testpage.php* pada *browser*. Kemudian bagaimana javascript meng-*update* halaman yang aktif dengan data hasil dari XMLHttpRequest, dapat dilihat pada listing 2.8.

```
Function loadpage(page_request, containerid){
      If (page_request.readyState == 4 &&page_request.status == 200)
      document.getElementById(containerid).innerHTML=page_request.responseText
}
```

Listing 2.8 Contoh Peng-*update*-an Suatu Tag Id HTML

Pada listing 2.8 diatas terdapat *function* bernama *loadpage* yang mempunyai parameter "*page_request*" dan "*containerid*", parameter "*page_request*" merujuk pada obyek XMLHttpRequest yang datanya akan direspon, kemudian parameter "*containerid*" merujuk pada alamat berdasarkan "*id*" yang berada pada halaman yang aktif tersebut.

BAB III
METODOLOGI

3.1 Analisis Kebutuhan Perangkat Lunak

3.1.1 Metode Analisis

Metode analisis berfungsi untuk menganalisis kebutuhan perangkat lunak pada sistem yang akan dibuat. Metode yang digunakan dalam pembuatan *website* ini adalah metode analisis berarah aliran data. yaitu dengan cara mengumpulkan data-data yang dibutuhkan dan mempelajari bahan yang ada untuk membangun sistem yang diinginkan

3.2 Hasil Analisis

Berdasarkan hasil analisis kebutuhan, maka diperoleh gambaran tentang *website* yang akan dibuat. Secara garis besar gambaran *website* yang akan dibuat adalah *website* yang dapat digunakan untuk memasukan data yang kemudian diolah menjadi infomasi. Infomasi hasil pengolahan data digunakan oleh para pengunjung dan anggota untuk mengetahui berita dan informasi tentang Yayasan Manarul Huda. Data tersebut dapat dimanipulasi, ditambah ataupun dikurangi kapan saja oleh orang yang diberi otorisasi untuk melakukan hal tersebut atau sering disebut *administrator*. Sehingga pada intinya, *website* ini berfungsi sebagai media penyampaian informasi tentang Yayasan Manarul Huda, dimana dalam beberapa bagiannya menerapkan teknologi AJAX.

Beberapa kendala dalam membangun *website* Yayasan Manarul Huda dengan penerapan teknologi AJAX antara lain :

1. Javascript merupakan bahasa pemrograman utama di dalam penerapan teknologi AJAX, sehingga diperlukan pemahaman tentang Javascript secara lebih mendalam.
2. Masih sedikitnya sumber dan literatur tentang teknologi AJAX dikarenakan penerapan teknologi AJAX yang "relatif" masih baru.
3. Web *browser* versi lama tidak mendukung teknologi AJAX.

3.2.1 Analisis Kebutuhan Input

Input atau masukan dari *website* Yayasan Manarul Huda ini, terdiri dari 3 jenis masukan, yaitu admin, anggota dan pengunjung.

a) Masukan Admin

Input admin adalah suatu masukan yang diberikan oleh admin berupa data masukan tentang anggota, data tentang admin, berita dan pengumuman tentang Yayasan Manarul Huda, Galeri dan publikasi, dan Topik dan komentar forum.

1. **Masukan data anggota**
   Masukan data tentang anggota yang terdaftar, berupa identitas diri dan status di yayasan.
2. **Masukan data admin**
   Masukan data identitas administrator, berupa *username* dan *password* .
3. **Masukan berita**
   Masukan informasi mengenai berita, agenda dan hal-hal yang terkait dengan Yayasan Manarul Huda.
4. **Masukan pengumuman**
   Masukan data pengumuman dari yayasan bagi anggota dan pengunjung.
5. **Masukan galeri**
   Masukan data dokumentasi dari yayasan yang berupa gambar atau foto.
6. **Masukan publikasi**
   Masukan tentang materi yang berkaitan dengan yayasan yang bisa di *download* oleh pengunjung.

7. Masukan forum

Masukan untuk forum yang dibuat, berupa topik dan komentar.

b) Masukan anggota

Input anggota adalah masukan yang diberikan oleh seorang anggota yang berupa *update* data anggota, juga *kata kunci* dari pencarian anggota.

c) Masukan pengunjung

Input pengunjung adalah masukan yang diberikan oleh seorang pengunjung yang berupa isian buku tamu, topik dan komentar forum, dan kata kunci pada pencarian.

3.2.2 Analisis Kebutuhan Proses

Kebutuhan proses dalam *website* Yayasan Manarul Huda adalah sebagai berikut:

1. Proses *login* dan logout *administrator*
   Pada proses *login* ini, diperlukan *username administrator* dan *password administrator* untuk *login* ke dalam *website.*
2. Proses *update* data administrator
   Pada proses ini, *administrator* dapat mengganti *username* dan *password.*
3. Proses *login* dan logout anggota
   Pada proses *login* ini, diperlukan *username* anggota dan *password* anggota untuk *login* ke dalam *website.*
4. Proses pengolahan data anggota
   Pada proses ini, administrator dapat mengubah data anggota.
5. Proses pencarian anggota
   Pada proses ini, *administrator* dan anggota dapat melakukan pencarian anggota dengan kata kunci nama anggota.
6. Proses pencarian berita
   Pada proses ini, pengunjung dapat melakukan pencarian berita dengan kata kunci judul berita.

7. Proses pengolahan galeri

   Pada proses ini, *administrator* dapat melakukan penambahan dan penghapusan galeri.

8. Proses pengolahan publikasi

   Pada proses ini, *administrator* dapat melakukan penambahan dan penghapusan publikasi.

9. Proses pengolahan buku tamu

   Pada proses ini, pengunjung dapat melakukan pengisian buku tamu.

10. Proses pengolahan forum

    Pada proses ini, pengunjung dapat melakukan pengisian topik dan komentar.

11. Proses pengolahan pengumuman

    Pada proses ini, *administrator* dapat melakukan penambahan dan penghapusan pengumuman.

3.2.3 Analisis Kebutuhan Output

Data keluaran yang diperoleh dari proses pada *Website* Yayasan Manarul Huda adalah data anggota, pengumuman dan berita, hasil pencarian, serta galeri dan publikasi.

3.2.4 Kebutuhan Antar Muka

Perangkat antar muka dengan menggunakan Macromedia Dreamweaver 8 merupakan pilihan yang tepat untuk mengimplementasikan penerapan teknologi AJAX yang lebih menekankan pada *client-side scripting.*

Rancangan *website* yang dibangun diharapkan dapat membantu pengguna pada umumnya dan anggota dari Yayasan Manarul Huda pada khususnya dalam mendapatkan informasi yang tepat mengenai Yayasan Manarul Huda.

3.2.5 Analisis Kebutuhan Perangkat Lunak

Perangkat Lunak yang digunakan untuk pembangunan *website* Yayasan Manarul Huda ini adalah :

1. Macromedia Dreamweaver 8
2. Macromedia Fireworks 8
3. Microsoft Windows XP
4. Mozilla Firefox 2.0.4
5. Firebug 1.0.5
6. *Bundle* Apache2Triad 1.5.4 (*MySQL 5.0.8, PHP5.1.2, Apache 2.2.0*)

3.2.6 Kebutuhan Perangkat Keras

Program ini dapat berjalan dengan baik, apabila memenuhi standar minimal dari perangkat keras (*hardware*) yang dimiliki. Spesifikasi minimal dari perangkat keras yang dapat digunakan untuk menjalankan program ini, yaitu :

1. Satu unit komputer dengan spesifikasi minimum prosessor Pentium III.
2. Memori (RAM) 256 Mb.
3. Harddisk 2Gb.
4. Monitor VGA atau SVGA.
5. Mouse.
6. Keyboard.

3.3 Analisis dan Perancangan Sistem

3.3.1 Metode Perancangan

Metode yang digunakan dalam perancangan perangkat lunak *Website* Yayasan Manarul Huda adalah metode perancangan berarah alir data dengan menggunakan alat-alat pengembangan sistem *Diagram Context* dan *Data Flow Diagram (DFD)*.

Untuk perancangan penerapan teknologi AJAX yang digunakan, akan digambarkan dengan *flowchart*.

3.3.2 Hasil Perancangan

Berdasarkan analisis yang telah dilakukan maka dapat diketahui apa saja yang manjadi masukan sistem, keluaran sistem, metode yang digunakan sistem, serta antar muka sistem yang dibuat, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan.

Perancangan sistem ini akan dibagi menjadi beberapa subsistem yaitu :

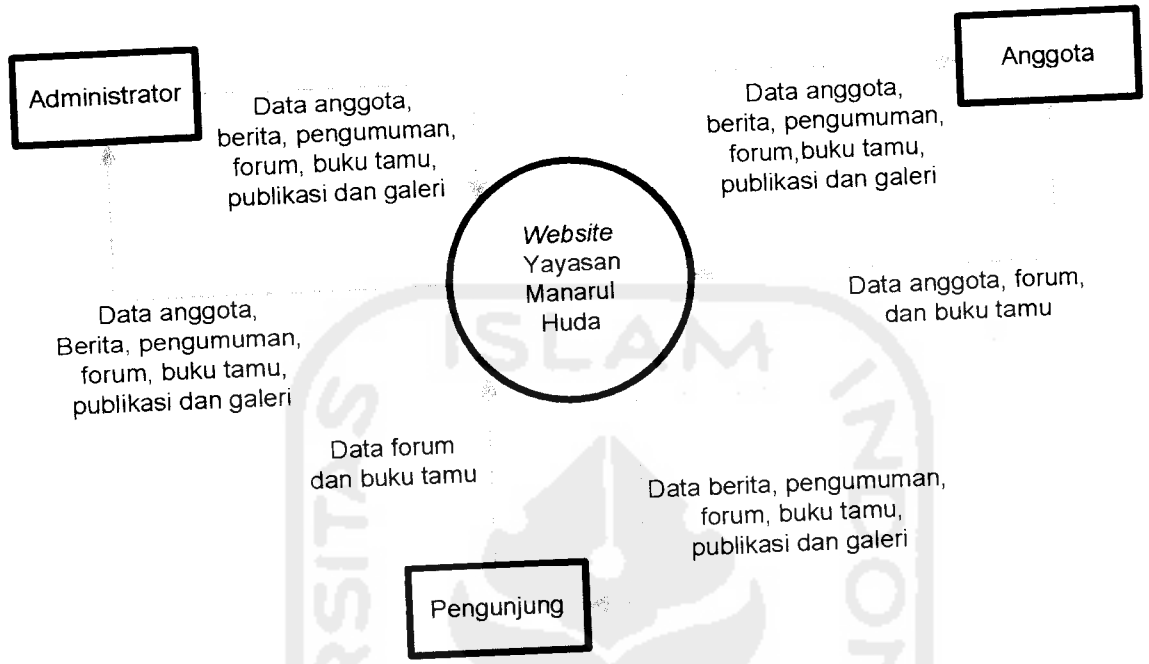
1. Perancangan *Data Flow Diagram*
2. Perancangan Penerapan Teknologi AJAX
3. Perancangan Tabel Basis Data
4. Perancangan Antar Muka

3.3.2.1 Perancangan Data Flow Diagram

3.3.2.1.1 Diagram Konteks

Desain ini dimulai dari bentuk yang paling umum yaitu diagram konteks, kemudian diagram konteks ini akan diturunkan sampai bentuk yang paling detail. Dalam pembuatan diagram konteks perlu menganalisis sistem yang akan dibangun, apa saja yang dibutuhkan,

sumber data dan tujuan ataupun hasil akhir yang diinginkan. Dari hasil analisis tersebut, diperoleh diagram konteks sebagaimana ditunjukkan pada Gambar 3.1

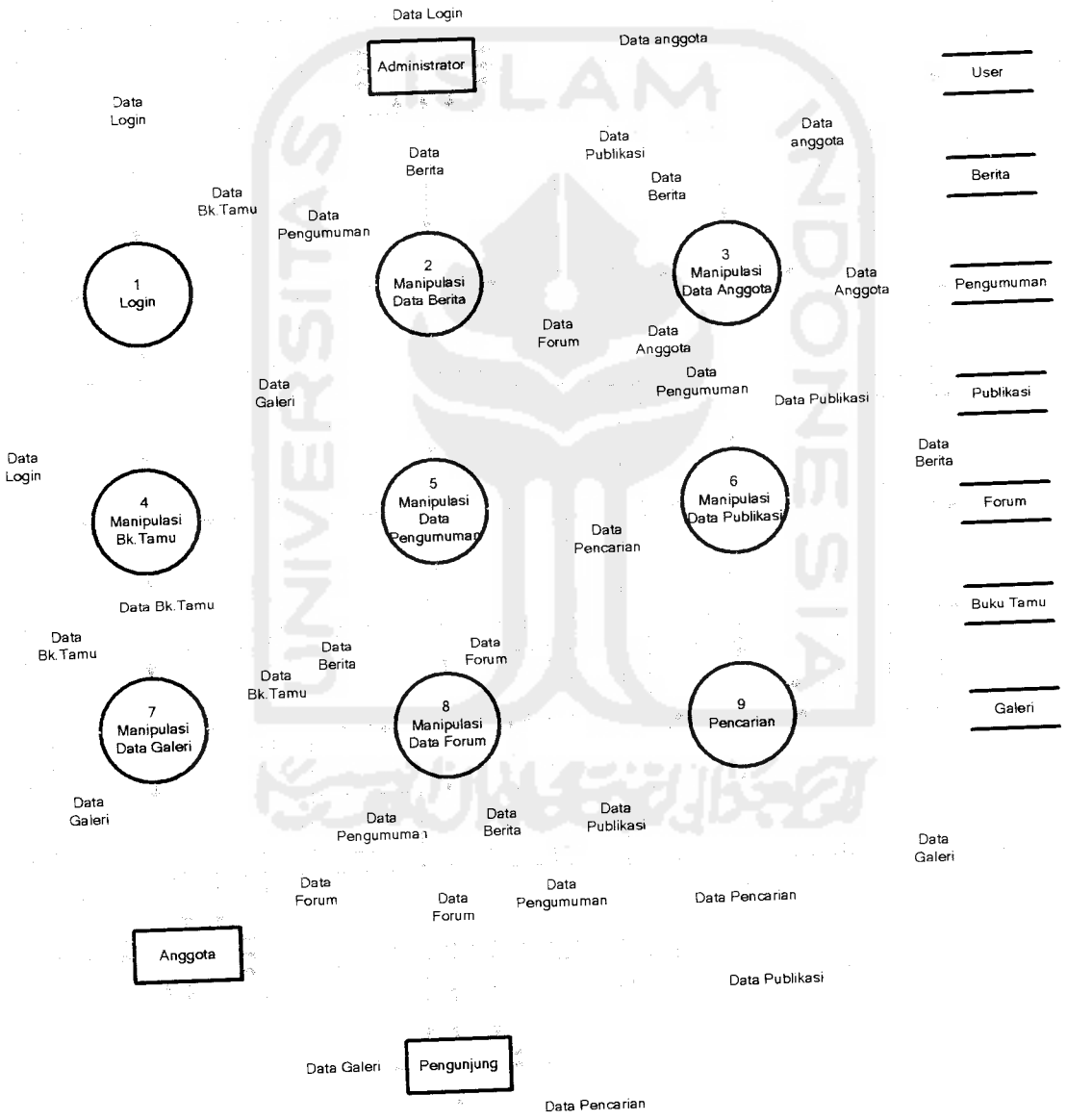


Gambar 3.1 Diagram Konteks Website Yayasan Manarul Huda

Context diagram atau dapat dikatakan sebagai DFD level 0 merupakan bentuk sistem yang paling global. Pada DFD level 0 diatas, Aliran data bersumber dari *administrator*, yang memasukan data dan memprosesnya sehingga kemudian data yang diolah tersebut disajikan ke dalam *website*, misalnya: publikasi, berita, forum, dsb. Sehingga isi dari masukan *administrator* dapat dilihat baik oleh anggota maupun pengunjung, kecuali data anggota yang hanya dapat dilihat oleh *administrator* dan anggota saja.

3.3.2.1.2 DFD level 1

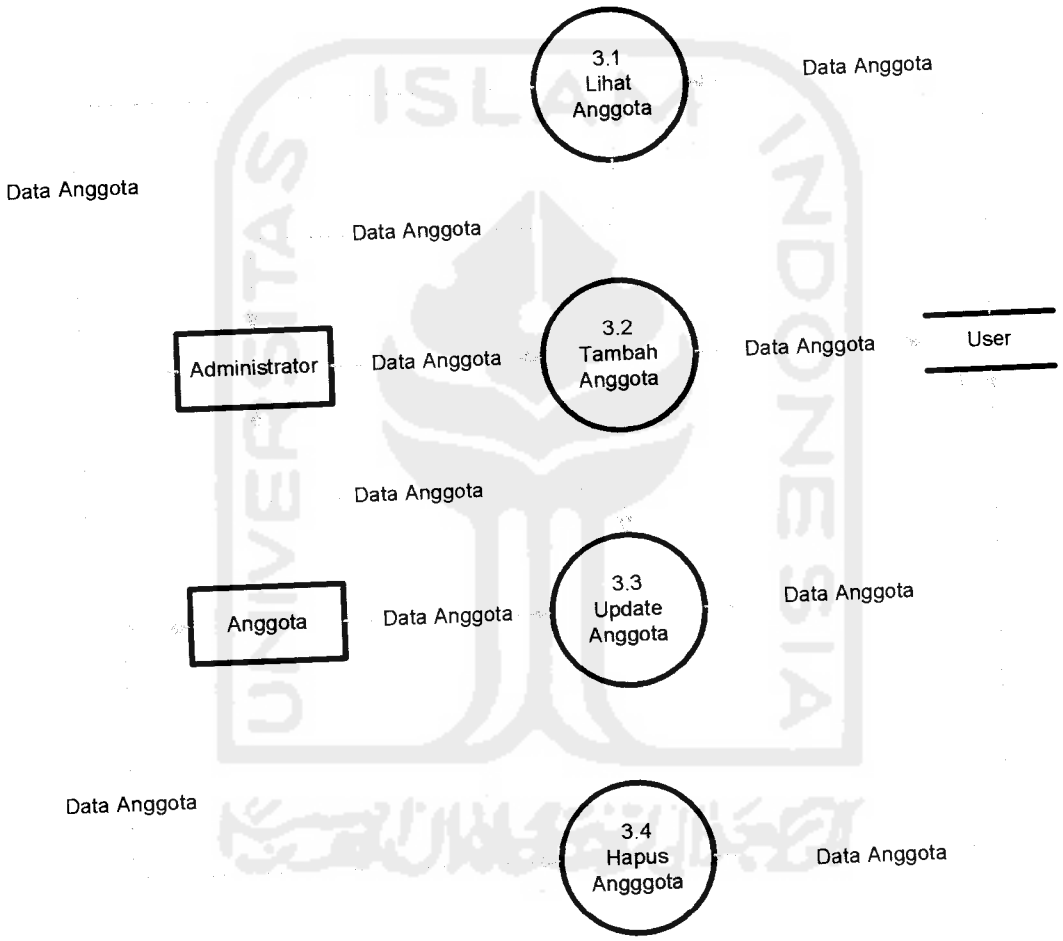
DFD level 1 terdiri dari 9 proses, yaitu: proses *login*, proses manipulasi data berita, proses manipulasi data anggota, proses manipulasi data buku tamu, proses manipulasi data pengumuman, proses manipulasi data publikasi, proses manipulasi data galeri, proses manipulasi data forum dan proses pencarian. DFD level 1 ditunjukkan pada Gambar 3.2 dibawah ini.



Gambar 3.2 DFD level 1

3.3.2.1.3 DFD Level 2 Proses Manipulasi Data Anggota

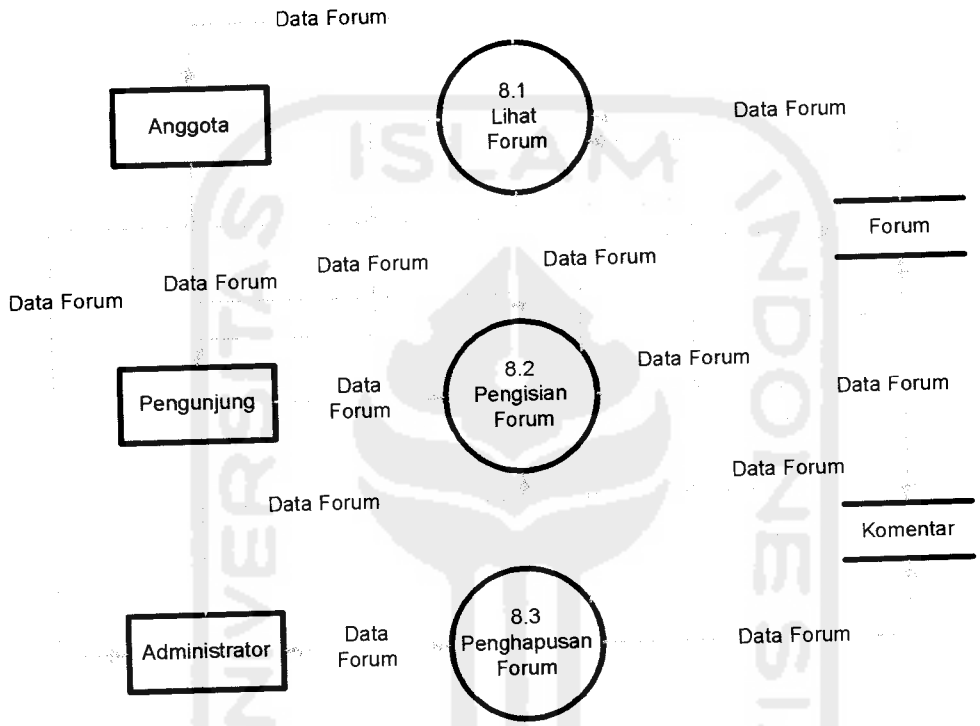
Pada DFD level 2 proses manipulasi data anggota ini, terdapat 4 proses, yakni: lihat anggota, tambah anggota, *update* anggota dan hapus anggota. DFD Level 2 proses manipulasi data anggota dapat dilihat pada gambar 3.3 dibawah ini.



Gambar 3.3 DFD Level 2 Proses Manipulasi Data Anggota

3.3.2.1.4 DFD Level 2 Proses Manipulasi Data Forum

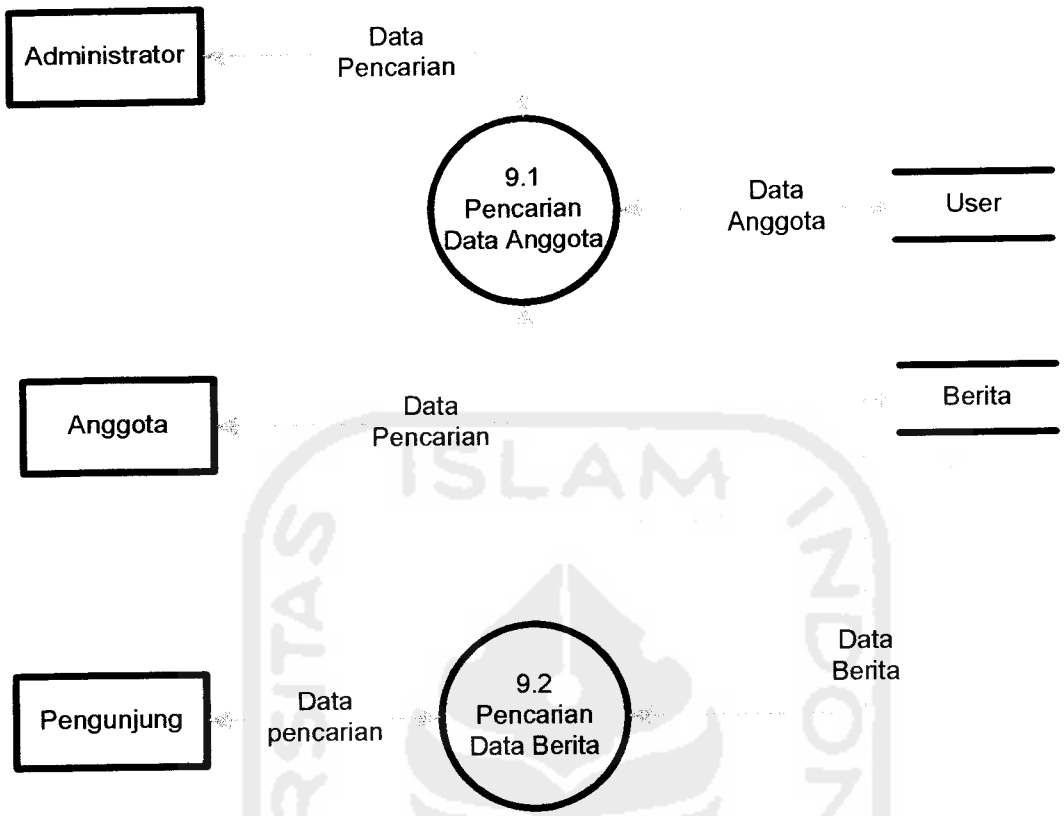
Pada DFD level 2 proses manipulasi data forum, terdapat 3 proses, yakni: proses lihat forum, proses pengisian forum, dan proses penghapusan forum. DFD Level 2 proses manipulasi data forum dapat dilihat pada gambar 3.4 dibawah ini.



Gambar 3.4 DFD Level 2 Proses Manipulasi Data Forum

3.3.2.1.5 DFD Level 2 Proses Pencarian

Pada DFD level 2 proses pencarian ini, terdapat 2 proses, yakni: Pencarian data anggota, dan pencarian berita. DFD Level 2 proses pencarian dapat dilihat pada gambar 3.5 dibawah ini.



Gambar 3.5 DFD Level 2 Proses Pencarian

3.3.2.2 Perancangan Penerapan Teknologi AJAX

Uraian perancangan penerapan teknologi AJAX yang digunakan pada *website* ini adalah sebagai berikut:

1. Javascript *event handler* memicu XMLHttpRequest (AJAX), pemicu ini bisa dibuat dari Javascript *function*, event-event seperti *OnMouseOver*, *OnClick*, *OnFocuss* dan *OnKeyUp*.

2. Terbentuknya obyek XMLHttpRequest oleh event-event tersebut diatas, menggunakan *method* open( ) dan send( ). Obyek XMLHttpRequest akan melakukan *request* ke *server*.
3. *Request* ke *server* pun terjalin, hal ini akan memanggil modul yang telah disiapkan dengan teknologi *server-side* (dalam kasus ini menggunakan PHP).
4. Modul dalam PHP melakukan tugasnya sesuai dengan modul apa yang di *request*. Dapat berupa koneksi ke *database* ataupun lainnya.
5. Setelah proses dari modul selesai, kemudian respon akan dikembalikan ke pengguna, dapat berupa data *string* ataupun lainnya, respon ini diterima kembali oleh obyek XMLHttpRequest (AJAX).
6. Respon dikembalikan oleh AJAX untuk melakukan peng-*update*-an DOM dari halaman *website* yang sedang aktif.

Dengan uraian seperti disebutkan diatas, maka implementasi teknologi AJAX pada *website* ini diterapkan pada proses :

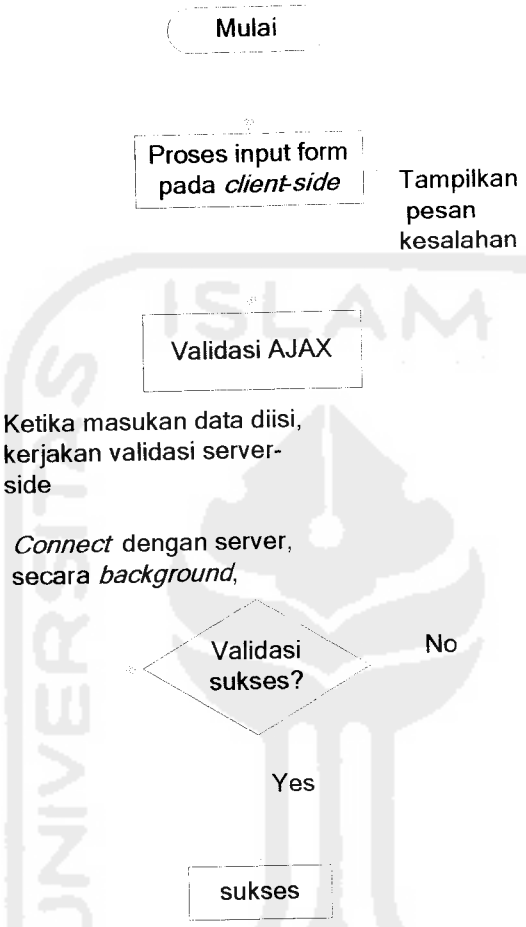
1. Proses *Load* Halaman
2. Proses Validasi
3. Proses *Login*, dan
4. Proses Pencarian

3.3.2.2.1 Proses *Load* Halaman

Untuk menampilkan halaman pada *website*, digunakan function tampil_halaman_ajax(). Fungsi ini me-*load* halaman yang dipanggil ke dalam tag id yang dimaksud dengan menggunakan obyek XmlHttpRequest yang telah dibentuk sebelumnya.

3.3.2.2.2 Proses Validasi

Gambar 3.6 berikut adalah proses validasi input masukan AJAX dalam bentuk *Flowchart*.

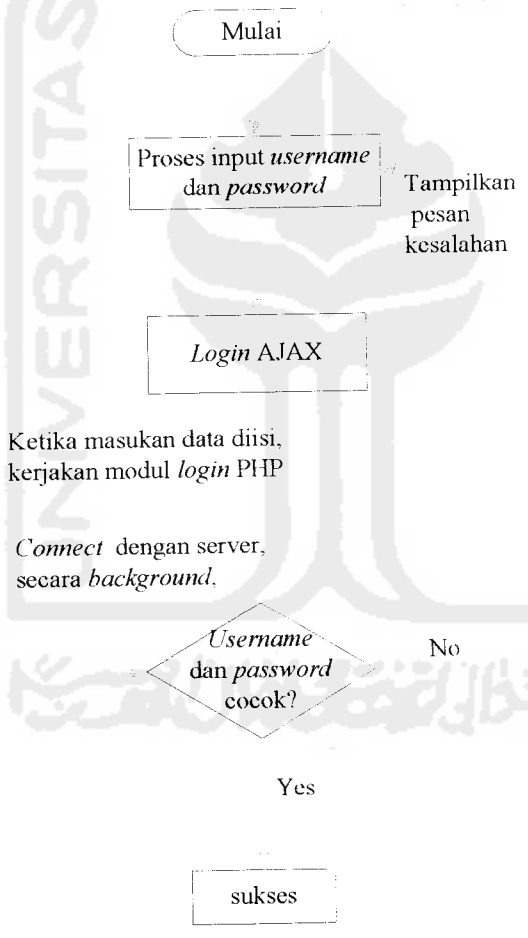


Gambar 3.6 *Flowchart* Proses Validasi Dengan Penerapan Teknologi AJAX

Pada proses validasi ini, modul PHP yang telah disiapkan sebelumnya dipanggil menggunakan obyek XMLHttpRequest dengan function validasi (). *Event* yang digunakan adalah OnKeyUp, sehingga setiap setelah pengguna memasukan input, maka setiap itu juga AJAX memproses input yang dimasukan pengguna sesuai dengan modul PHP yang dipanggil. Modul PHP yang dipanggil dapat berupa koneksi dengan *database*.

3.3.2.2.3 Proses *Login*

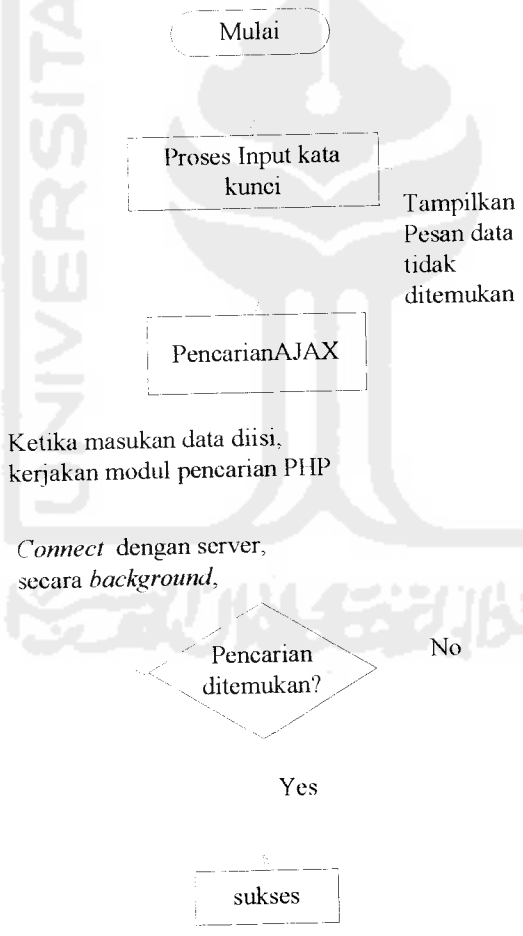
Dengan langkah yang hampir sama dengan proses validasi, proses *login* dengan AJAX menggunakan function login () yang dipicu oleh *event* OnKeyUp pada *username* dan *password*. Setelah pengguna memasukan *username* atau *password*, obyek XMLHttpRequest memanggil modul PHP yang berfungsi mengecek data masukan *username* dan *password* dengan data *username* dan *password* pada *database*. Gambar 3.7 berikut ini adalah *flowchart* proses *login* dengan penerapan teknologi AJAX.



Gambar 3.7 Flowchart Proses *Login* Dengan Penerapan Teknologi AJAX

3.3.2.2.4 Proses Pencarian

Keunggulan penerapan teknologi AJAX pada pencarian (*live search)* adalah hasil pencarian ditampilkan langsung setelah memasukan kata kunci. Pada AJAX pencarian ini, modul PHP yang berfungsi melakukan pencarian dipanggil oleh Obyek XMLHttpRequest pada function pencarian (), hasil dari modul PHP tersebut langsung ditampilkan dalam *tag* id yang telah ditentukan sebelumnya dengan pemicu *event* OnKeyUp pada *textfield* "kata kunci". *Flowchart* pencarian dengan penerapan teknologi AJAX dapat dilihat pada gambar 3.8 dibawah ini.



Gambar 3.8 Flowchart Pencarian Dengan Penerapan Teknologi AJAX

3.3.2.3 Perancangan Tabel Basis Data

Pada penelitian ini akan digunakan basis data relasional. Ada beberapa tabel yang digunakan dalam penelitian ini, tabel-tabel tersebut antara lain:

1) Tabel User

Tabel User digunakan untuk menyimpan data *login* admin dan anggota. Struktur tabel user dapat di lihat pada tabel 3.1

Tabel 3.1 Tabel User

Nama Field	Tipe Data	Panjang	Keterangan
Niy	Varcha	7	No. Induk (PK)
Username	Varchar	20	Username
Password	Varchar	20	Password
Status	Varchar	3	Status anggota

2) Tabel Data_anggota

Table data anggota digunakan untuk menyimpan detail data anggota. Struktur tabel data anggota dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel Data_anggota

Nama Field	Tipe Data	Panjang	Keterangan
Niy	Varchar	7	Nomor induk yayasan (PK)
Nm_lgkp	Varchar	45	Nama lengkap
Gender	Char	1	Jenis kelamin
Tmpt_lhr	Varchar	20	Tempat lahir
Tgl_lhr	Date		Tanggal lahir
Almt	Varchar	45	Alamat lengkap
Telp	Varchar	15	Nomor telepon
Foto	Varchar	45	Foto anggota
Nm_ayah	Varchar	45	Nama lengkap ayah
Nm_ibu	Varchar	45	Nama lengkap ibu
Kerja_ayah	Varchar	20	Pekerjaan ayah

Kerja_ibu	Varchar	20	Pekerjaan ibu
Pnddkn_ayah	Varchar	20	Pendidikan terakhir ayah
Pnddkn_ibu	Varchar	20	Pendidikan terakhir ibu
Pndptn	Varchar	20	Pendapatan per bulan
Anak	Integer	2	Jumlah anak
Anak ke	Integer	2	Urutan Anak
Tgl_update	datetime		Tanggal update data

3) Tabel Kelas

Tabel kelas digunakan untuk menyimpan data anggota pada setiap kelas. Struktur tabel kelas dapat di lihat pada tabel 3.3

Tabel 3.3 Tabel Kelas

Nama Field	Tipe Data	Panjang	Keterangan
Id_kelas	Integer	4	Id kelas (PK)
Id_data_kls	Integer	4	Id data kelas
Niy	Varchar	7	No. Induk

4) Tabel Data_kelas

Tabel data_kelas digunakan untuk menyimpan data kelas. Struktur tabel kelas dapat diihat pada tabel 3.4

Tabel 3.4 Tabel Data Kelas

Nama Field	Tipe Data	Panjang	Keterangan
Id_data_kls	Integer	4	Id data kelas (PK)
Nm_kls	Varchar	20	Nama kelas
Th_kelas	Varchar	9	Tahun kelas
Wali_kls	Varchar	45	Nama wali kelas

5) Tabel Kehadiran

Tabel kehadiran digunakan untuk menyimpan data kehadiran anggota. Struktur tabel kehadiran dapat di lihat pada tabel 3.5

Tabel 3.5 Tabel Kehadiran

Nama Field	Tipe Data	Panjang	Keterangan
Id_kehadiran	Integer	4	Id kehadiran (PK)
Niy	Varchar	7	Nomor induk yayasan anggota
Hadir	Integer	3	hadir
Sakit	Integer	3	Sakit
Izin	Integer	3	Izin
Tanpa_ket	Integer	3	Tanpa keterangan

6) Tabel Jadwal

Tabel jadwal digunakan untuk menyimpan jadwal pelajaran anggota dengan status 'MTs' dan 'MA'. Struktur tabel jadwal dapat di lihat pada tabel 3.6

Tabel 3.6 Tabel Jadwal

Nama Field	Tipe Data	Panjang	Keterangan
Id_jadwal	Integer	4	Id jadwal pelajaran (PK)
Hari	Varchar	6	Hari
Jam_ke	Integer	1	Waktu
Id_pelajaran	Integer	4	Id pelajaran
Id_kls	Integer	4	Id kelas
Pengampu	Varchar	35	Pengampu pelajaran
Th_jadwal	Varchar	9	Tahun jadwal

7) Tabel Tagihan

Tabel tagihan digunakan untuk menyimpan data tagihan dari anggota. Struktur tabel tagihan dapat di lihat pada tabel 3.7.

Tabel 3.7 Tabel Tagihan

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id tagihan (PK)
Status	Varchar	3	No. induk yayasan
Keterangan	Text		Keterangan tagihan
Jumlah	Varchar	25	Jumlah tagihan
Waktu	Varchar	30	Waktu pembayaran

8) Tabel Berita

Tabel berita digunakan untuk menyediakan berita mengenai Yayasan Manarul Huda. Struktur berita dapat di lihat pada tabel 3.8

Tabel 3.8 Tabel Berita

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id berita (PK)
Judul	Varchar	45	Judul berita
Ringkasan	Text		Ringkasan berita
Selengkapnya	Text		Berita selengkapnya
Tanggal	Datetime		Tanggal dimuat
Foto	Varchar	45	Gambar berita

9) Tabel Pengumuman

Tabel pengumuman digunakan untuk menyimpan informasi yang disampaikan oleh yayasan. Struktur tabel pengumuman dapat di lihat pada tabel 3.9.

Tabel 3.9 Tabel Pengumuman

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id pengumuman (PK)
Judul	Text		Isi
Tanggal	Datetime		Tanggal

10) Tabel Publikasi

Tabel publikasi digunakan untuk menyimpan publikasi dari yayasan, yang berupa dokumen yang dapat di-*download* oleh pengunjung. Struktur tabel publikasi dapat di lihat pada tabel 3.10

Tabel 3.10 Tabel Publikasi

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id publikasi (PK)
Keterangan	Text		Keterangan
File	Varcharr	45	File publikasi
Tanggal	Datetime		Tanggal input

11) Tabel Galeri

Tabel galeri digunakan untuk menyimpan data dokumentasi yayasan berupa gambar atau foto. Struktur tabel galeri dapat di lihat pada tabel 3.11

Tabel 3.11 Tabel Galeri

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id galeri (PK)
Keterangan	Text		Keterangan
Foto	Varchar	20	File galeri
Tanggal	Datetime		Tanggal input

12) Tabel Forum

Tabel Forum digunakan untuk menyimpan data topik dari forum. Struktur tabel forum dapat di lihat pada tabel 3.12

Tabel 3.12 Tabel Forum

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id forum (PK)
Isi_forum	Text		Topik forum

Tanggal	Date		Tanggal topik
Nama	Varchar	45	Nama
Email	Varchar	30	Email
Baru	Integer	1	Topik baru
Tampil	Integer	1	Ditampilkan atau tidak

13) Tabel Komentar

Tabel komentar digunakan untuk menyimpan data komentar dari topik forum yang dibuat. Struktur tabel komentar dapat di lihat pada tabel 3.13

Tabel 3.13 Tabel Komentar

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id komentar (PK)
Nama	Varchar	45	Nama
Email	Varchar	30	Email
Isi	Text		Isi komentar
Tanggal	Datetime		Tanggal komentar
Id_forum	Integer	4	Id forum
Baru	Integer	1	Komentar baru
Tampil	Integer	1	Ditampilkan atau tidak

14) Tabel Bk_tamu

Tabel bk_tamu digunakan untuk menyimpan data dari buku tamu. Struktur tabel bk_tamu dapat di lihat pada tabel 3.14

Tabel 3.14 Tabel Bk_tamu

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id buku tamu (PK)
Nama	Varchar	45	Nama
Pesan	Text		Isi buku tamu

Email	Varchar	30	Email
Tanggal	Datetime		Tanggal buku tamu
Baru	Integer	1	Pesan baru
Tampil	Integer	1	Ditampilkan atau tidak

Tabel 3.14 Tabel Bk_tamu

15) Table Snapshot

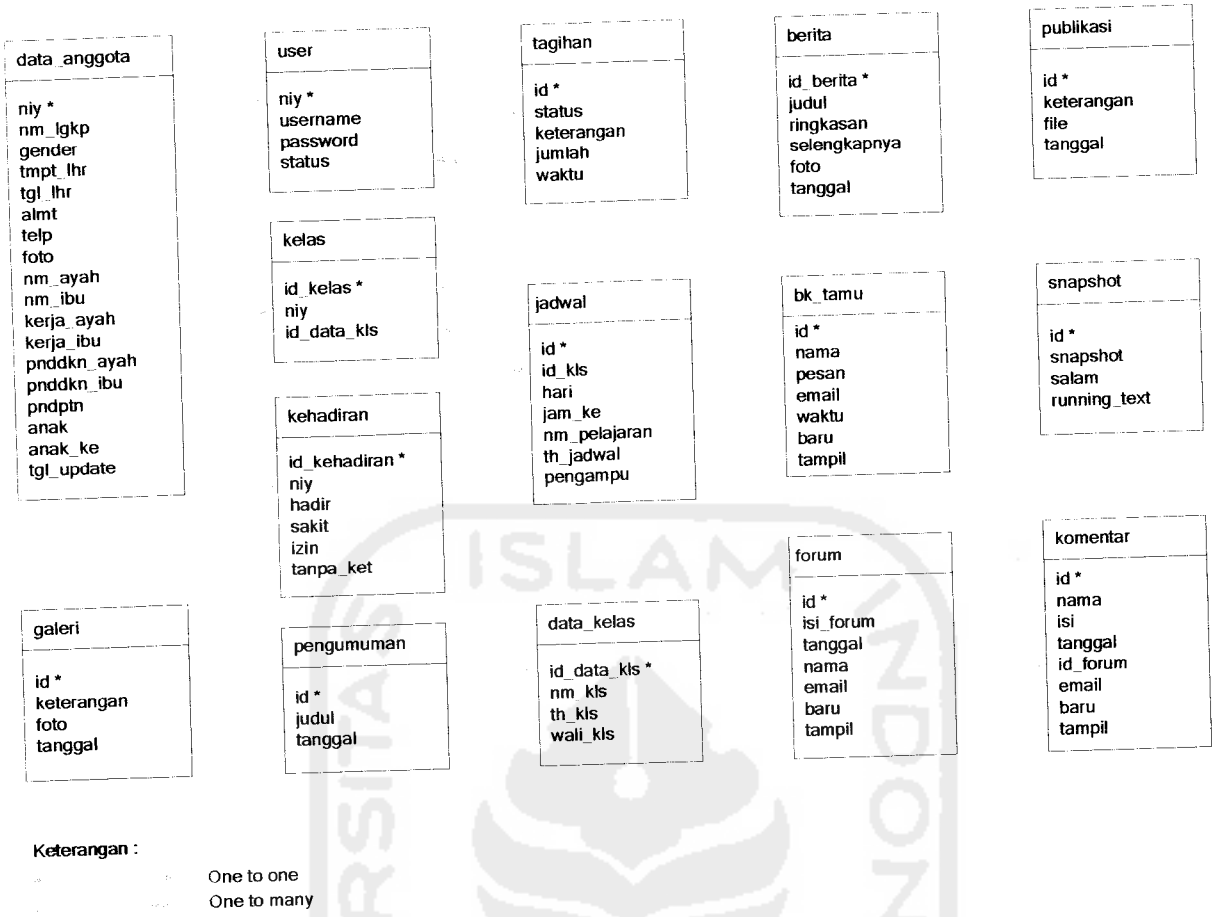
Tabel snapshot digunakan untuk menyimpan pesan admin pada halaman *website*.. Struktur tabel snapshot dapat di lihat pada tabel 3.15

Tabel 3.15 Tabel Snapshot

Nama Field	Tipe Data	Panjang	Keterangan
Id	Integer	4	Id snapshot (PK)
Snapshot	Text		Isi dari snapshot
Salam	Text		Isi dari salam
Running_text	Text		Isi dari running text

3.3.2.4 Relasi Antar Tabel

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya. Data tersebut disimpan dalam sebuah *file* atau tabel yang memiliki keterangan tertentu dan apabila direlasikan akan menghasilkan informasi yang kompleks. Relasi antar tabel di dalam *website* ini dapat dilihat pada gambar 3.9.



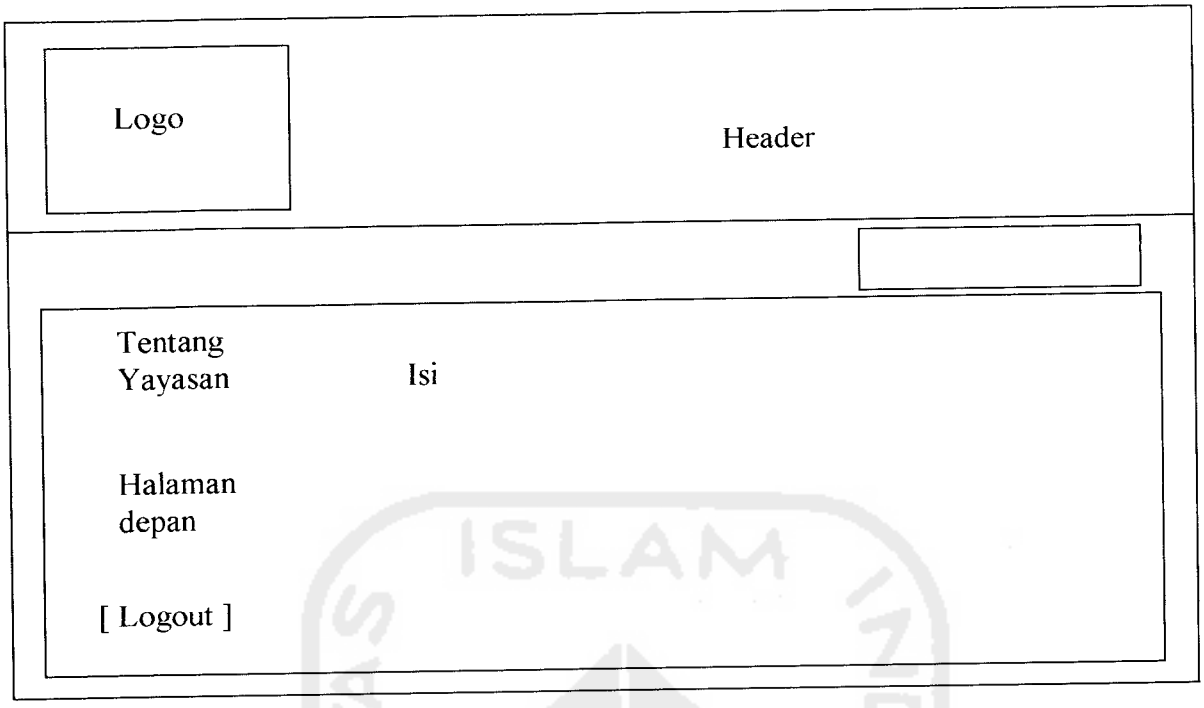
Gambar 3.9 Relasi Antar Tabel

3.3.2.5 Perancangan Antar Muka

Rancangan antar muka dari *Website* Yayasan Manarul Huda ini menggunakan Macromedia Fireworks 8 untuk desain tampilan, dan Macromedia Dreamweaver 8 sebagai *script* editornya.

3.3.2.5.1 Perancangan Halaman Pengunjung

Halaman pengunjung merupakan halaman yang akan dilihat pertama kali saat membuka URL, untuk mengakses halaman ini tidak diharuskan *login* terlebih dahulu. Tampilan halaman untuk pengunjung dapat dilihat melalui gambar 3.10 dibawah ini.

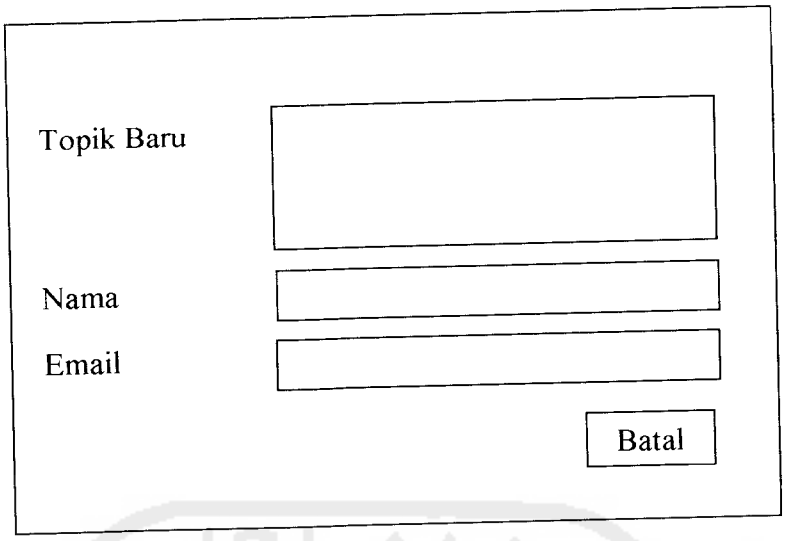


Gambar 3.10 Rancangan Halaman Pengunjung

Pada halaman pengunjung ini terdapat beberapa halaman input data, diantaranya adalah sebagai berikut:

1. **Rancangan Halaman Tambah Topik**

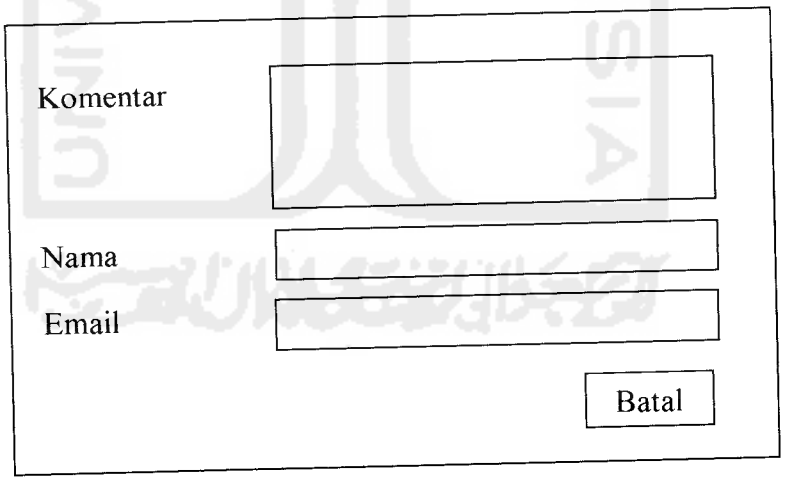
   Pengunjung dapat menambah topik pada forum yayasan. Rancangan halaman tambah topik dapat dilihat pada gambar 3.11 berikut ini.



Gambar 3.11 Rancangan Halaman Tambah Topik

2. Rancangan Halaman Tambah Komentar

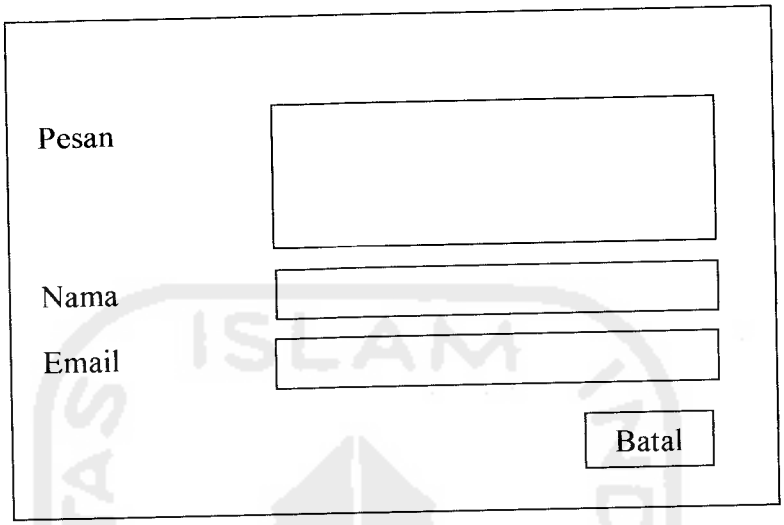
Pada halaman ini pengunjung dapat menambah komentar dari topik yang dibuat pada forum yayasan. Rancangan Halaman Tambah Komentar dapat dilihat pada gambar 3.12.



Gambar 3.12 Rancangan Halaman Tambah Komentar

3. Rancangan Halaman Tambah Buku Tamu

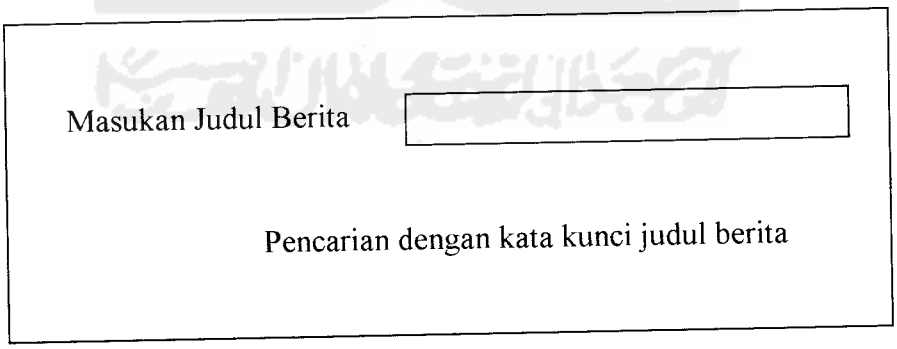
Pada halaman ini pengunjung dapat meninggalkan pesan di buku tamu. Rancangan Halaman Tambah Komentar dapat dilihat pada gambar 3.13 berikut ini.



Gambar 3.13 Rancangan Halaman Tambah Buku Tamu.

4. Rancangan Halaman Pencarian

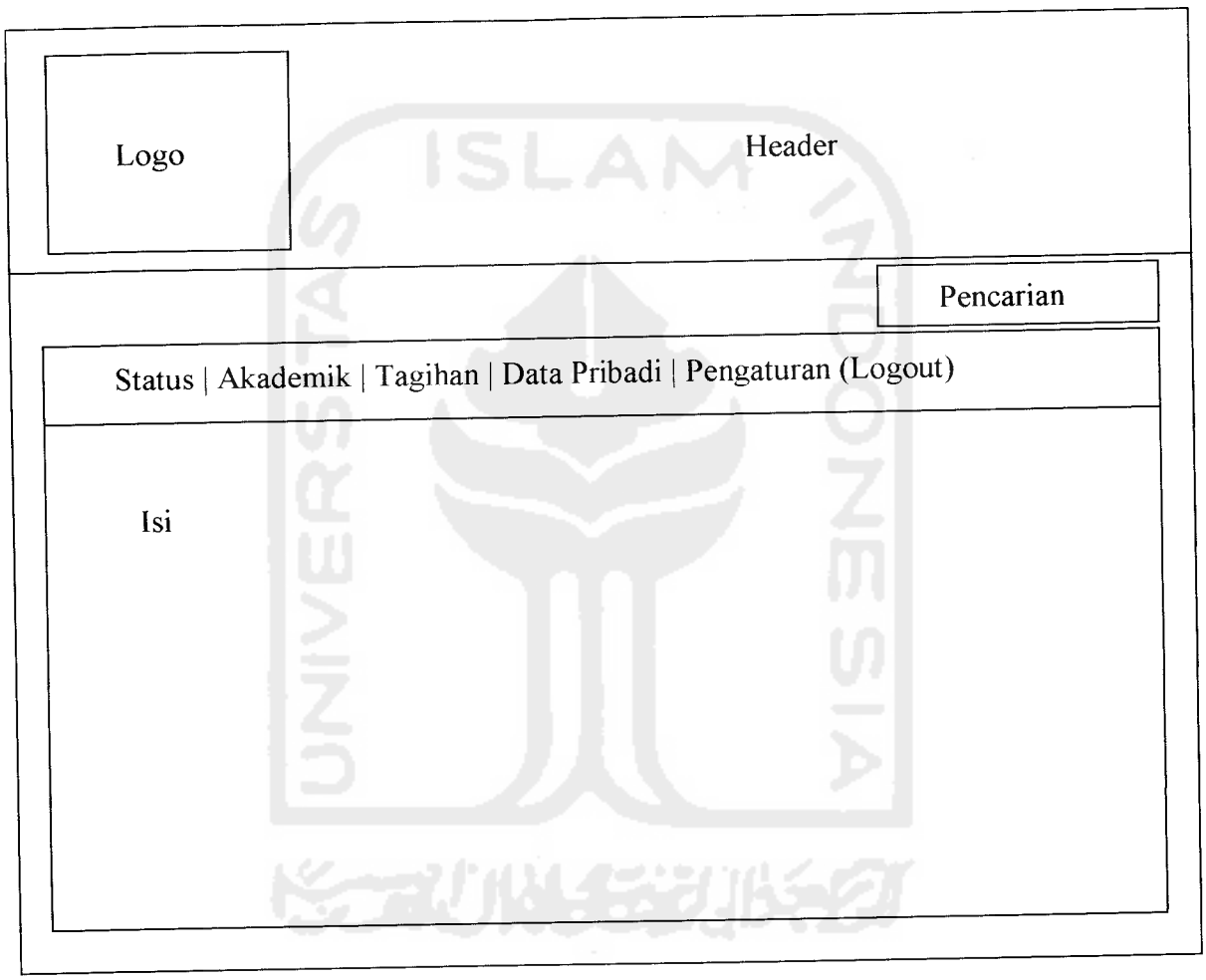
Pengunjung dapat melakukan pencarian berita yang tersedia pada halaman ini, dengan input kata kunci dari judul berita. Halman pencarian dapat dilihat pada gambar 3.14 berikut ini.



Gambar 3.14 Rancangan Halaman Pencarian

3.3.2.5.2 Rancangan Halaman Anggota

Halaman anggota ditujukan bagi anggota yang terdaftar. Untuk mengaksesnya diharuskan *login* terlebih dahulu. Pada halaman ini terdiri dari menu Status, Akademik, Tagihan, Data Pribadi dan Pengaturan. Selain itu juga terdapat link untuk melakukan pencarian anggota. Rancangan Halaman anggota dapat dilhat pada gambar 3.15 berikut ini.

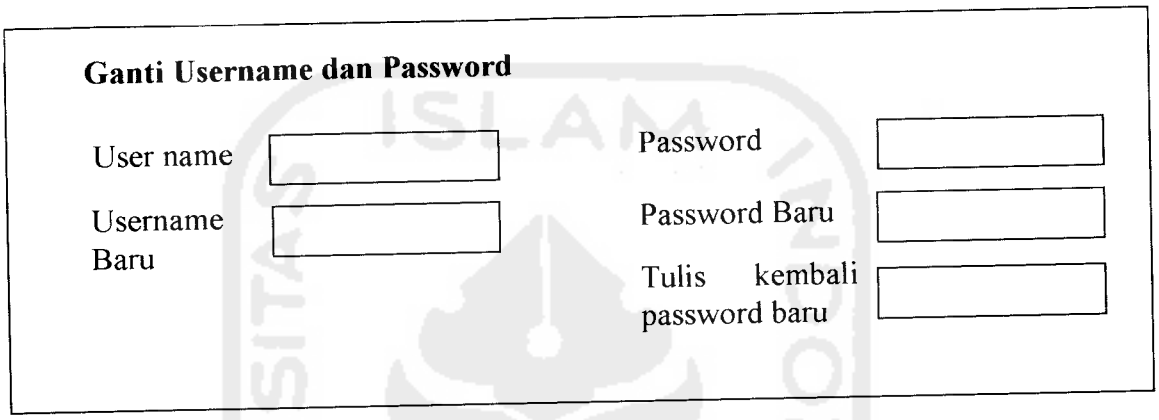


Gambar 3.15 Rancangan Halaman Anggota

Pada halaman anggota ini, terdapat beberapa halaman input data, diantaranya adalah sebagai berikut:

1. Rancangan Halaman Pengaturan

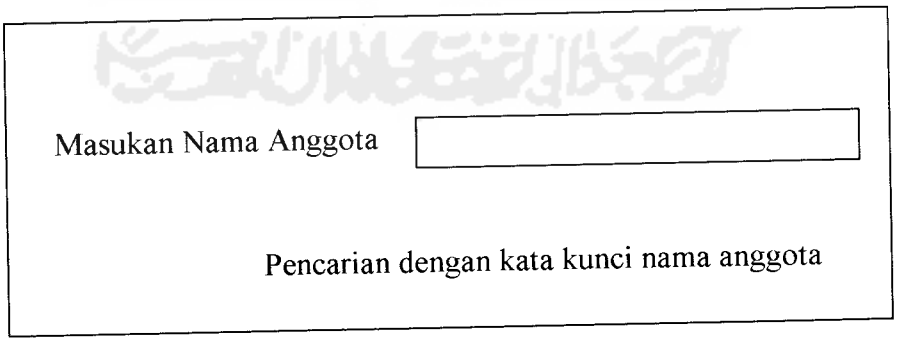
Anggota dapat mengubah *username* dan *password* pada halaman pengaturan ini. Rancangan halaman pengaturan anggota dapat dilihat pada gambar 3.16.



Gambar 3.16 Rancangan Halaman Pengaturan

2. Rancangan Halaman Pencarian Anggota

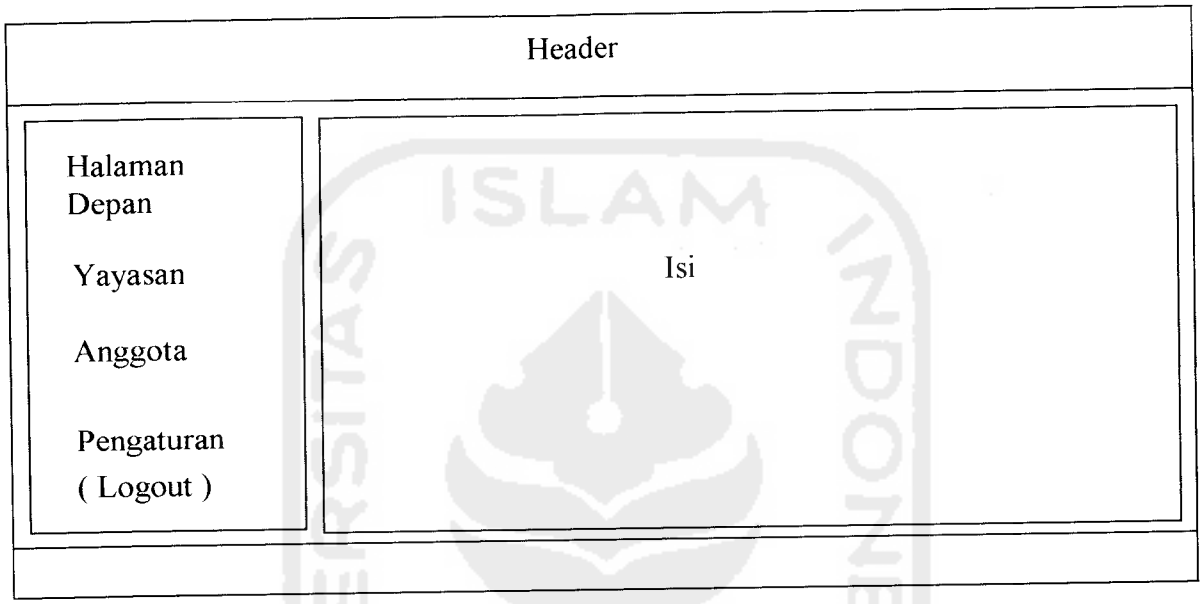
Pada halaman ini, anggota dapat melakukan pencarian anggota lain. Rancangan halaman pencaarian anggota dapat dilihat pada gambar 3.17.



Gambar 3.17 Rancangan Halaman Pencarian Anggota

3.3.2.5.3 Rancangan Halaman *Administrator*

Halaman *administrator* merupakan halaman dimana admin dapat melakukan penambahan, peng-*update*-an dan penghapusan data pada *website*. Untuk mengakses halaman ini, admin diharuskan *login* terlebih dahulu di halaman *login*. Diperlukan *username* admin dan *password* admin. Rancangan halaman *administrator* dapat dilihat pada gambar 3.18.

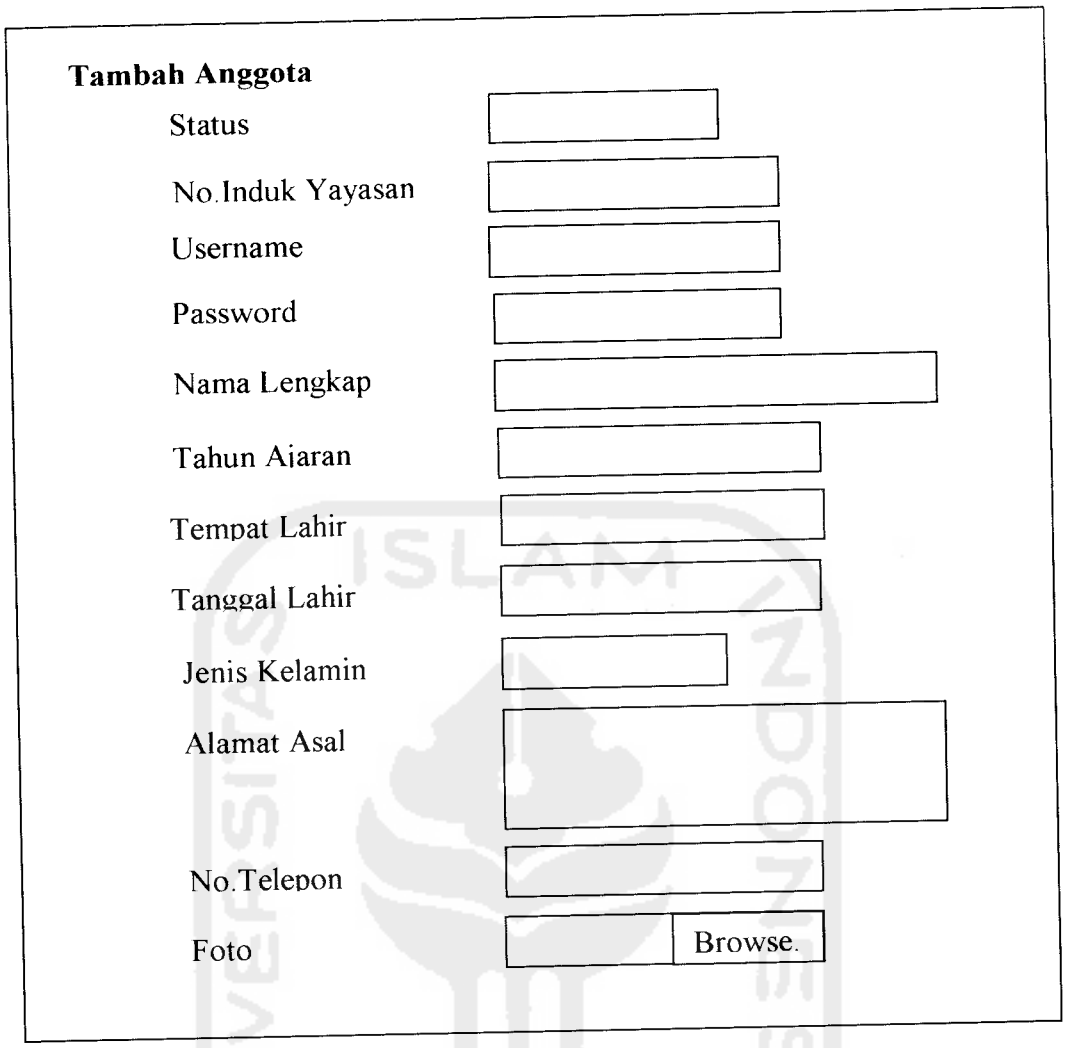


Gambar 3.18 Rancangan Halaman Admin

Halaman *administrator* terdiri dari beberapa halaman masukan data, diantaranya adalah sebagai berikut:

1. **Rancangan Halaman Tambah Anggota**

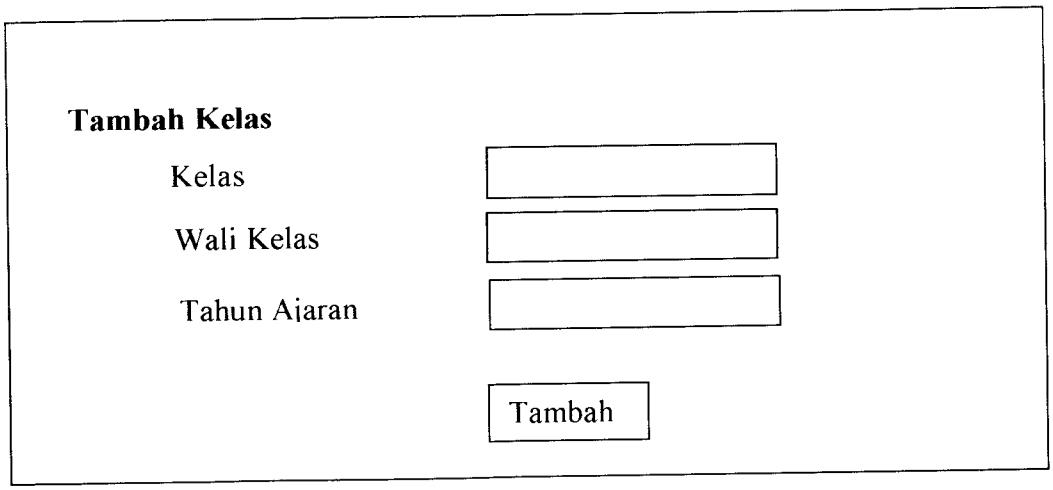
   Pada halaman ini, admin dapat menambah anggota yayasan. Halaman ini dapat dilihat pada gambar 3.19.



Gambar 3.19 Rancangan Halaman Tambah Anggota.

2. Rancangan Halaman Tambah Kelas

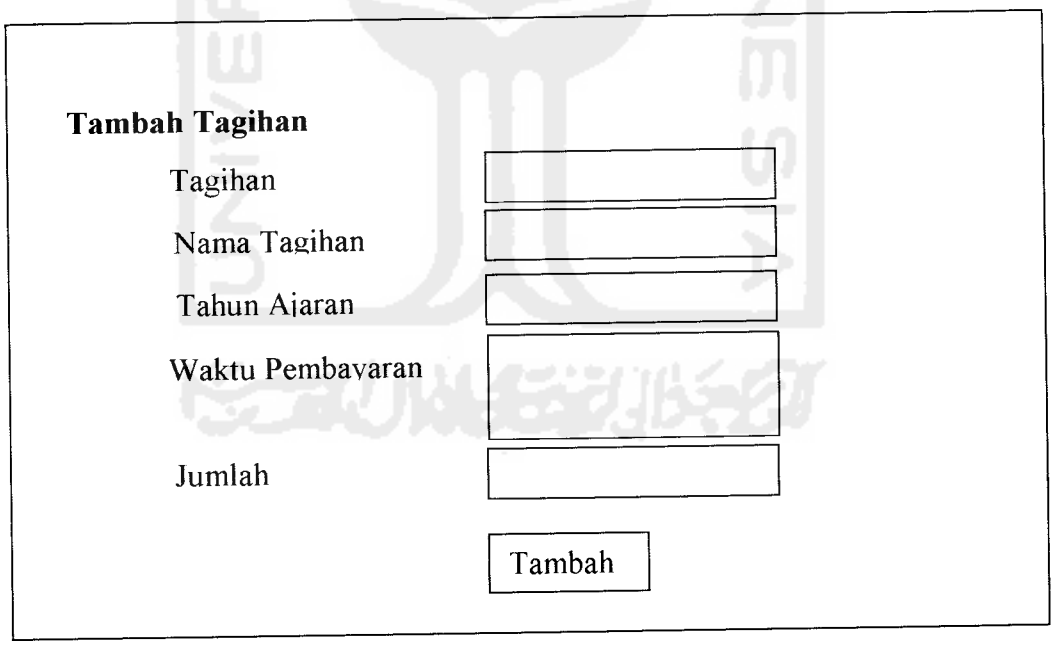
Pada halaman ini, admin dapat menambah kelas dari MTs dan MA. Rancangan halaman tambah kelas dapat dilihat pada gambar 3.20.



Gambar 3.20 Rancangan Halaman Tambah Kelas

3. Rancangan Halaman Tambah Tagihan

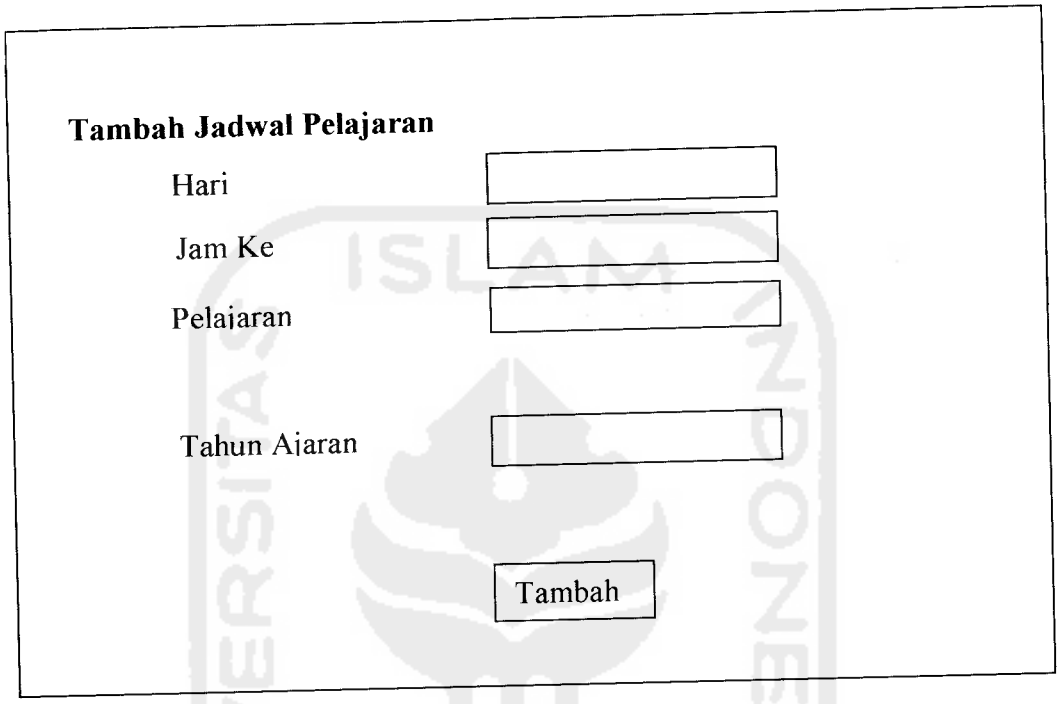
Pada halaman ini admin dapat menambah tagihan anggota. Rancangan halaman tambah tagihan anggota dapat dilihat pada gambar 3.21.



Gambar 3.21 Rancangan Halaman Tambah Tagihan

4. Rancangan Halaman Tambah Jadwal Pelajaran

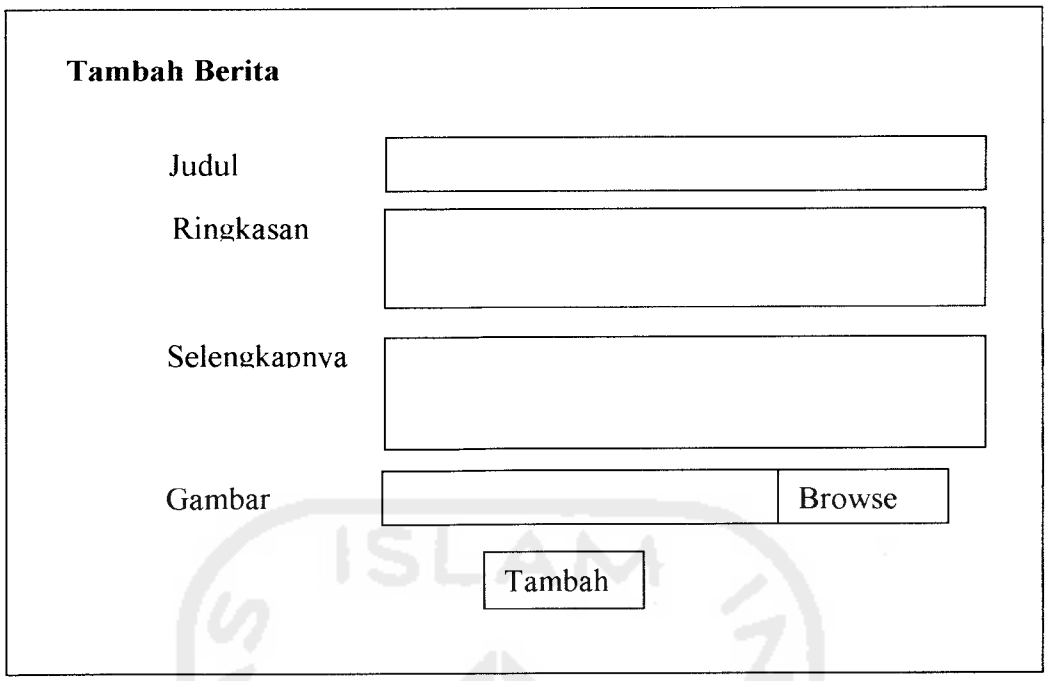
Pada halaman ini admin dapat menambah jadwal pelajaran. Rancangan halaman jadwal pelajaran dapat dilihat pada gambar 3.22.



Gambar 3.22 Rancangan Halaman Tambah Jadwal Pelajaran

5. Rancangan Halaman Tambah Berita

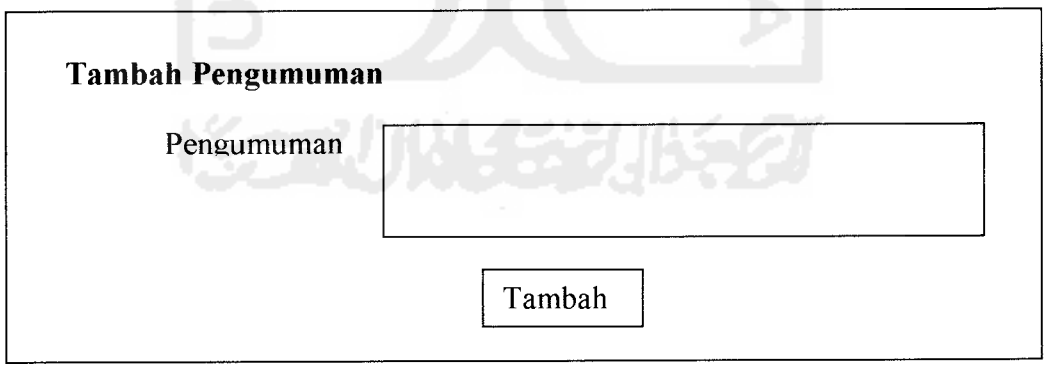
Pada halaman ini admin dapat menambah berita. Rancangan halaman tambah berita dapat dilihat pada gambar 3.23.



Gambar 3.23 Rancangan Halaman Tambah Berita

6. Rancangan Halaman Tambah Pengumuman

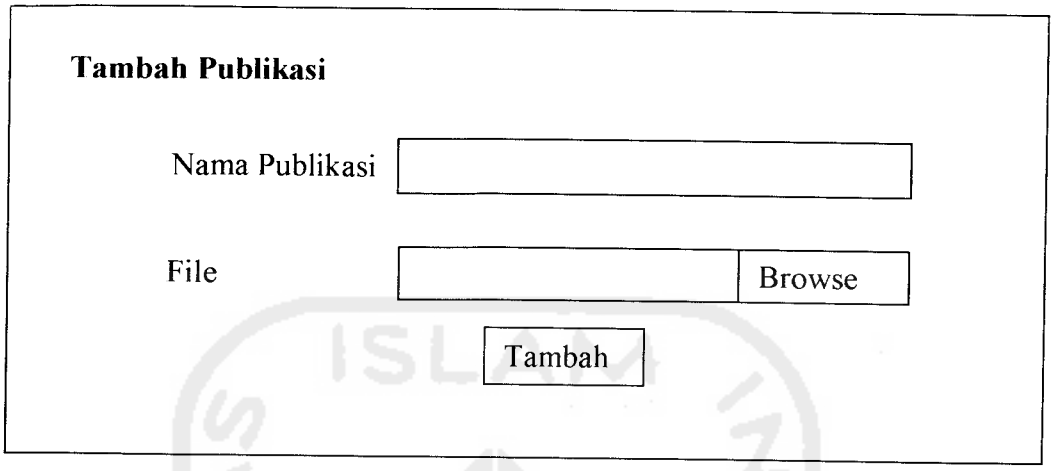
Pada halaman ini admin dapat menambah pengumuman yayasan. Rancangan halaman tambah pengumuman dapat dilihat pada gambar 3.24.



Gambar 3.24 Rancangan Halaman Pengumuman

7. Rancangan Halaman Publikasi

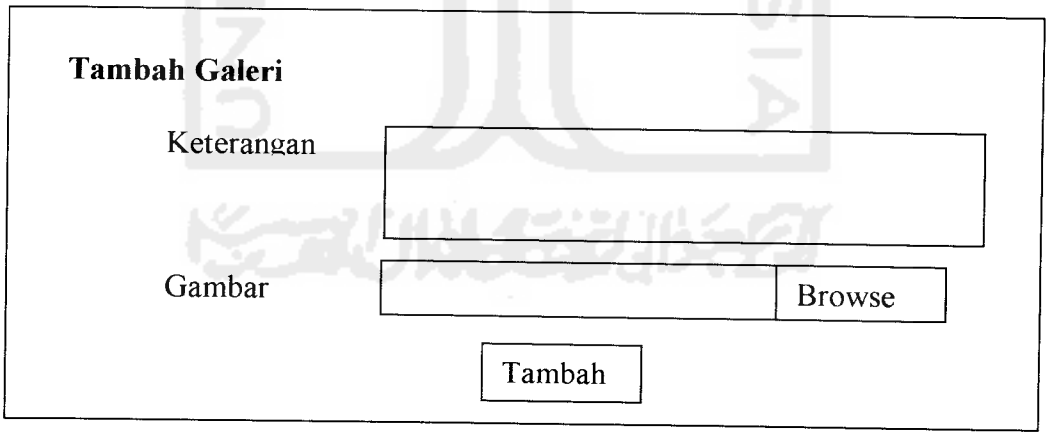
Pada halaman ini admin dapat menambah publikasi yayasan. Rancangan halaman tambah publikasi dapat dilihat pada gambar 3.25.



Gambar 3.25 Rancangan Halaman Tambah Publikasi

8. Rancangan Halaman Tambah Galeri

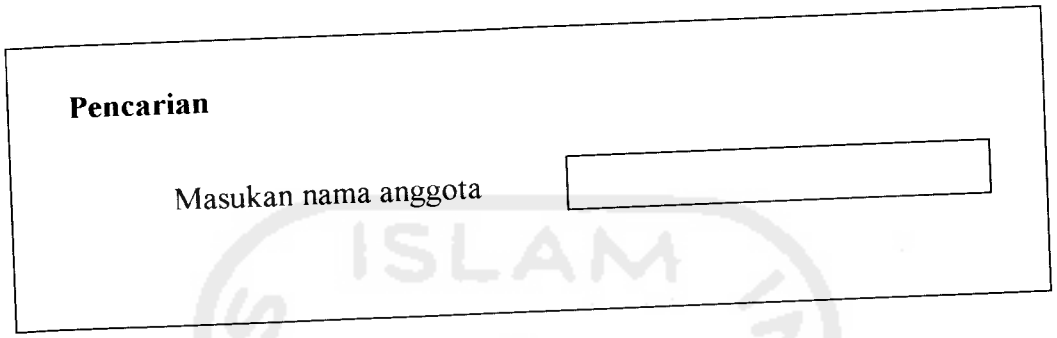
Pada halaman ini admin dapat menambah galeri yayasan. Rancangan halaman tambah galeri dapat dilihat pada gambar 3.26.



Gambar 3.26 Rancangan Halaman Tambah Galeri

9. Rancangan Halaman Pencarian

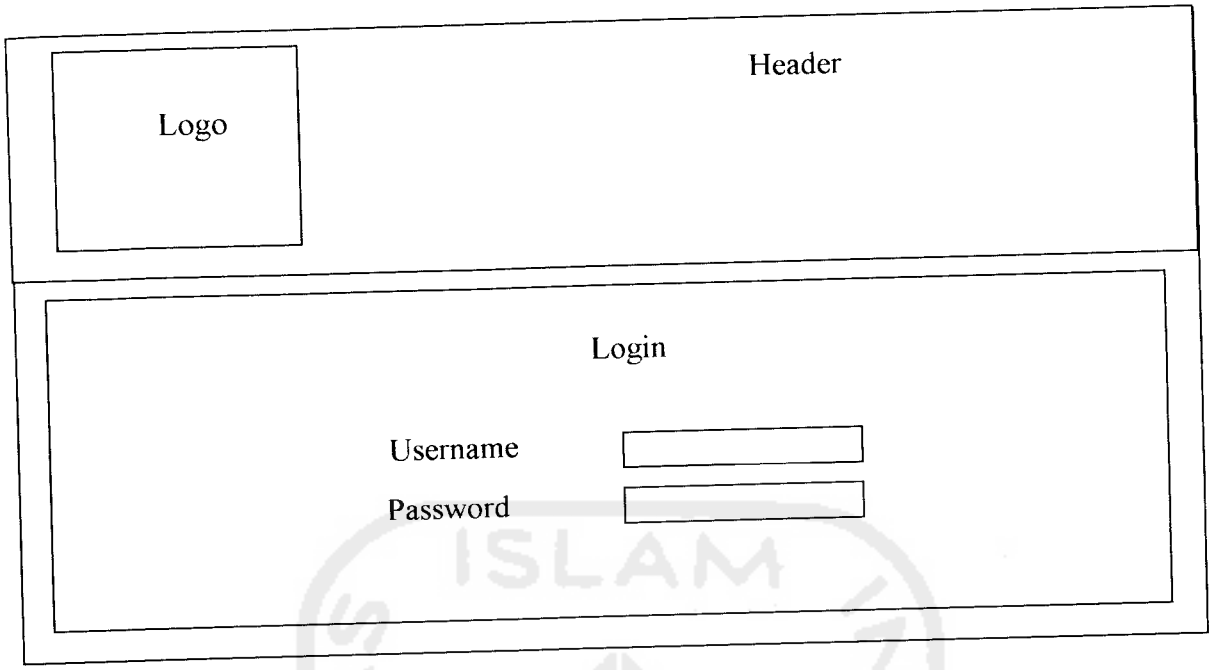
Pada Halaman ini admin dapat melakukan pencarian anggota. Rancangan halaman pencarian dapat dilihat pada gambar 3.27.



Gambar 3.27 Rancangan Halaman Pencarian

3.3.2.5.4 Antarmuka Halaman *Login*

Untuk dapat mengakses halaman *administrator* dan halaman anggota, admin dan anggota diharuskan *login* terlebih dahulu. UNtuk melakukan *login*, diperlukan *username* admin dengan *password* admin, dan *username* anggota dengan *password* anggota. Rancangan halaman *login* dapat dilihat pada Gambar 3.28.



Gambar 3.28 Rancangan Antarmuka Halaman *Login*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Perangkat Lunak

Tahap implementasi merupakan tahap dimana *website* telah dirancang untuk dapat dioperasikan pada keadaan yang sebenarnya, sehingga pada tahap ini dapat diketahui apakah sistem telah sesuai dengan tujuan yang diharapkan atau tidak.

4.1.1 Batasan Implementasi

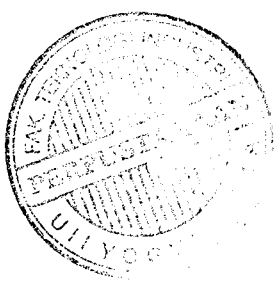
Batasan implementasi dari *Website* Yayasan Manarul Huda ini adalah implementasi proses pemasukan data baru, peng-*update*-an data, penghapusan data, proses menampilkan data, dan proses pencarian.

4.1.2 Implementasi Antar Muka

Halaman pada *Website* Yayasan Manarul Huda ini dibagi atas tiga kategori, yaitu; halaman pengunjung, halaman anggota dan halaman a*dministrator*.

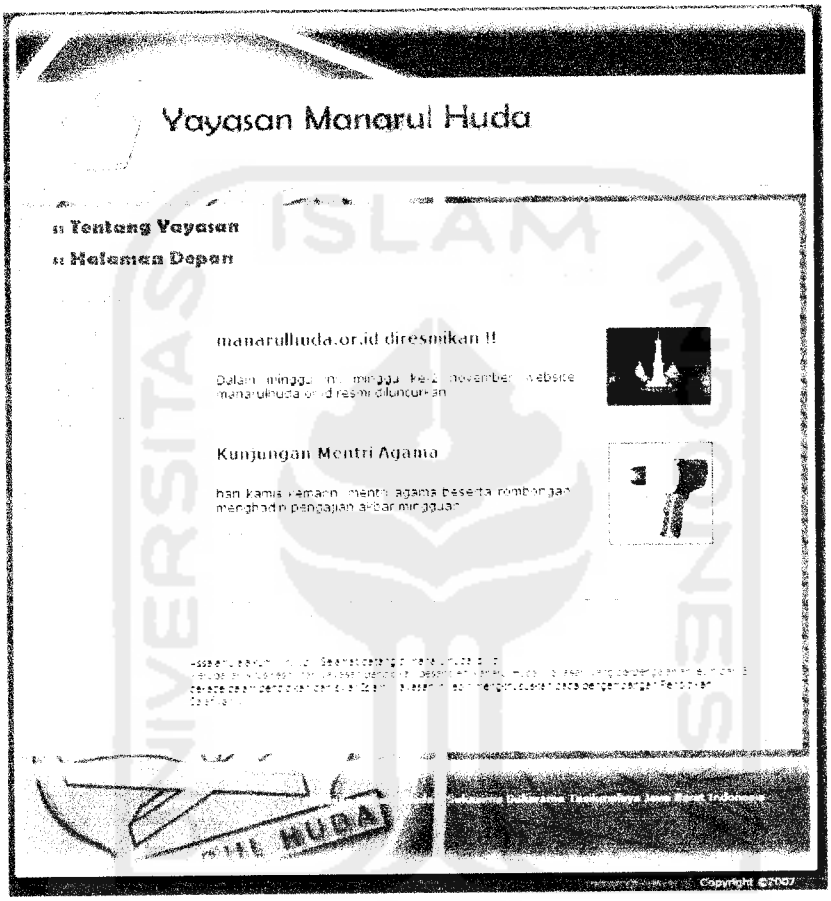
4.1.2.1 Halaman Pengunjung

Halaman pengunjung ini dapat dilihat oleh siapa saja dan tidak diharuskan untuk *login* terlebih dahulu.



4.1.2.1.1 Halaman Beranda (Halaman Depan)

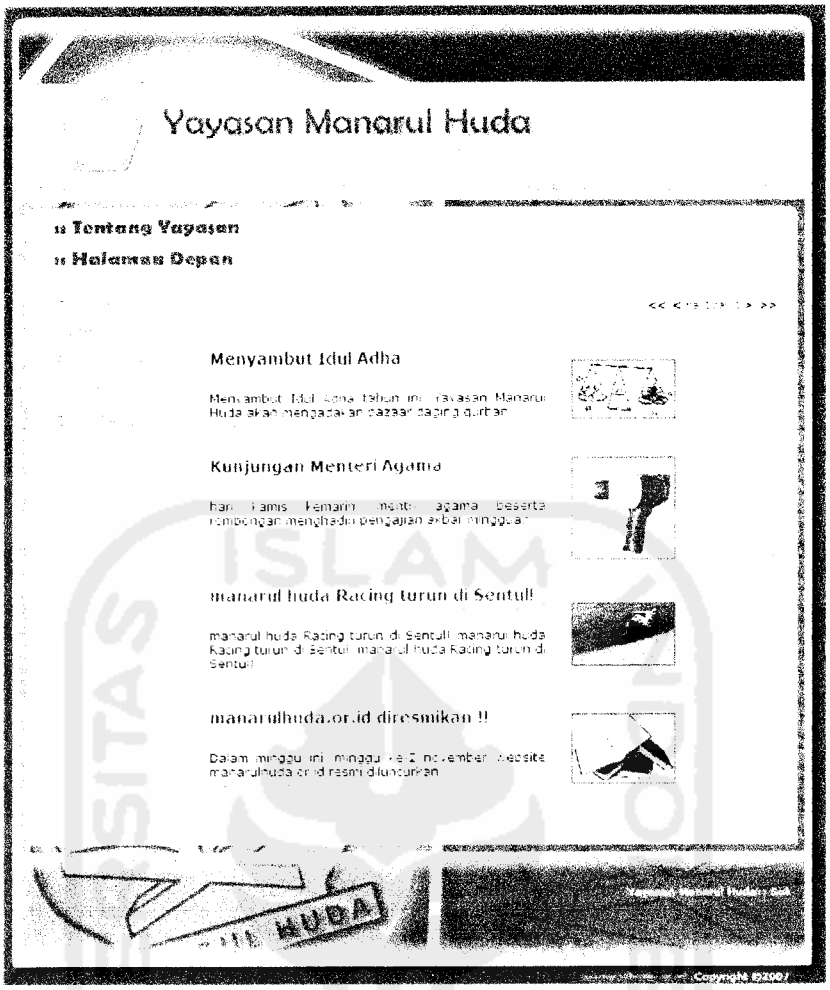
Halaman ini merupakan halaman depan (*Home*) dari *Website* Yayasan Manarul Huda berisi berita dan informasi tentang yayasan. Halaman beranda dapat dilihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Halaman Beranda (Halaman Depan).

4.1.2.1.2 Halaman Berita

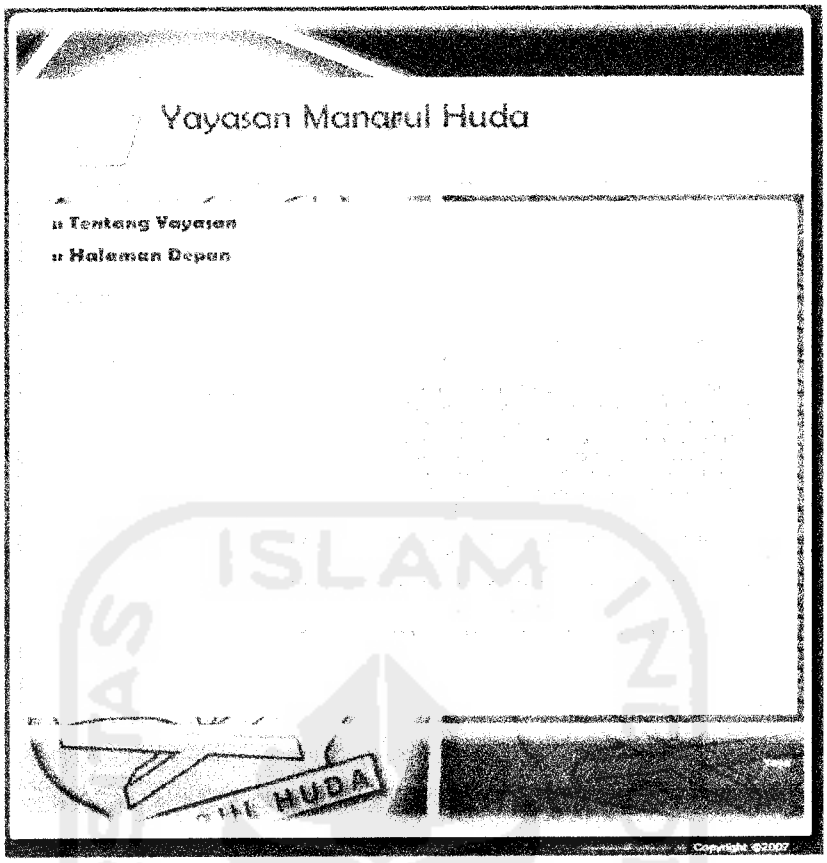
Halaman ini berisi tentang berita mengenai Yayasan Manarul Huda. Ditampilkan judul, ringkasan dan gambar dari berita yang bersangkutan. Lebih lengkapnya dapat dilihat pada gambar 4.2



Gambar 4.2 Halaman Berita

4.1.2.1.3 Halaman Pengumuman

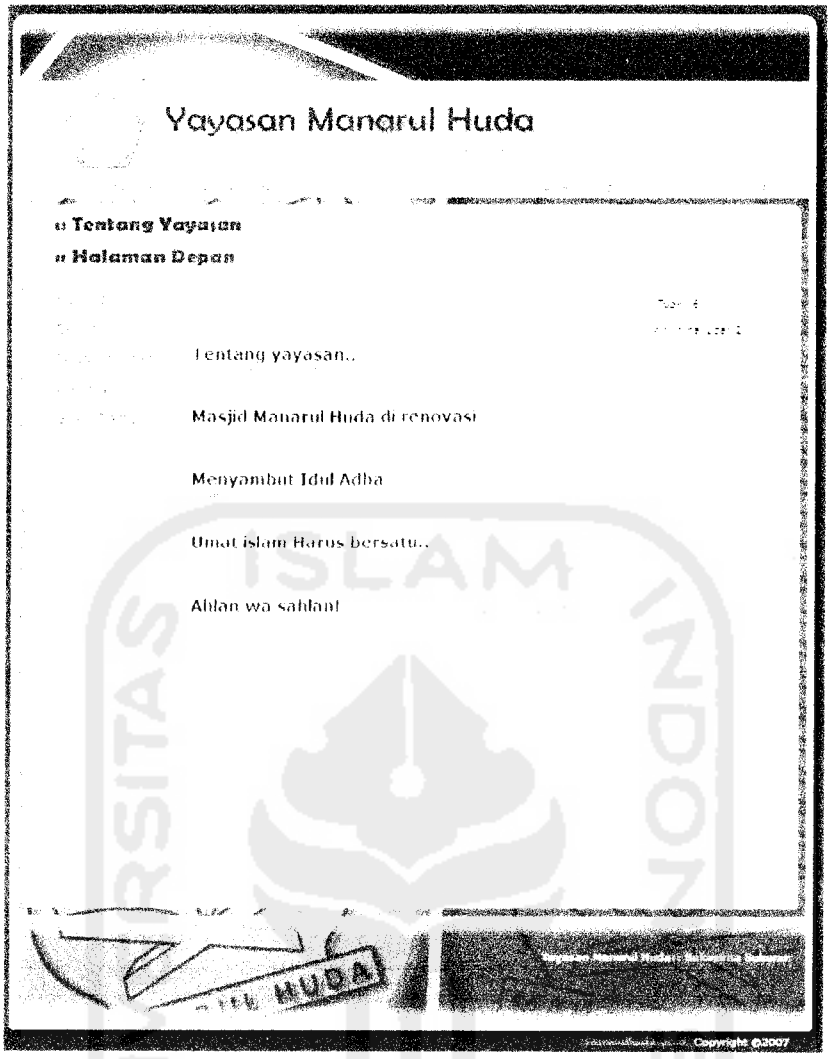
Pada halaman ini terdapat pengumuman dari yayasan untuk para anggota dan pengunjung, diurutkan berdasarkan tanggal dan waktu *posting*, untuk lebih jelasnya, halaman pengumuman dapat dilihat pada gambar 4.3 dibawah ini.



Gambar 4.3 Halaman Pengumuman

4.1.2.1.4 Halaman Forum

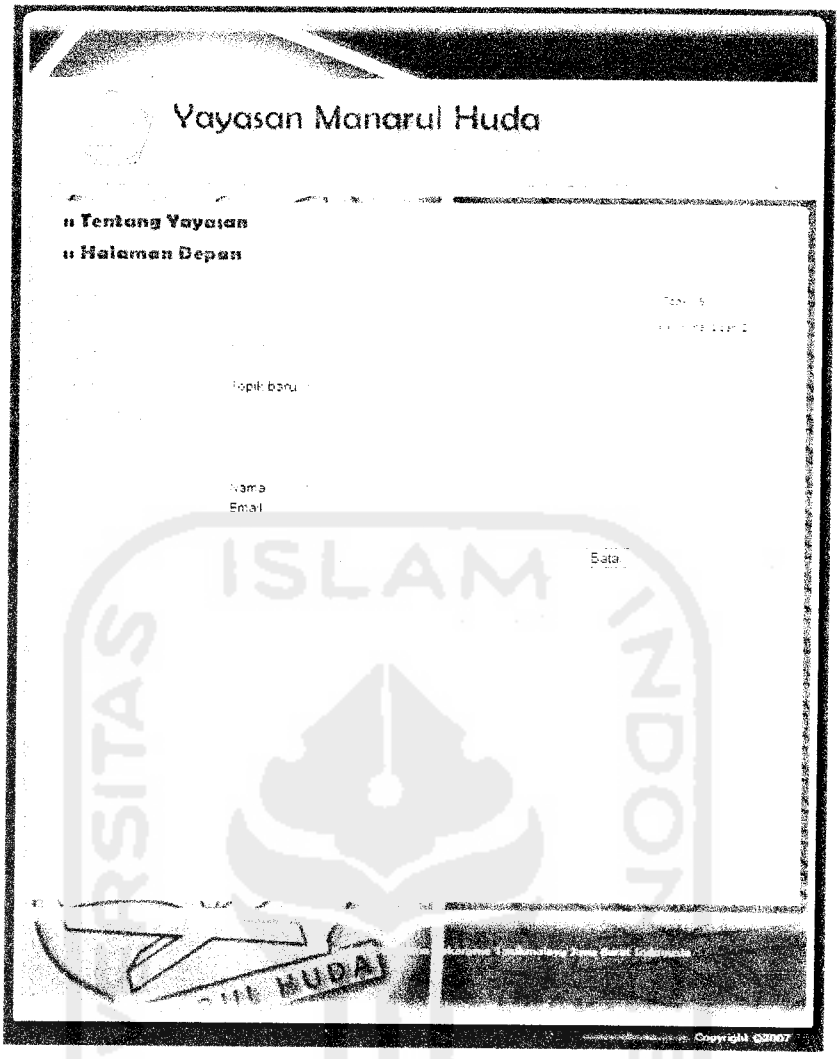
Halaman ini berisi topik–topik pada forum yayasan. Halaman forum dapat dilihat pada gambar 4.4 dibawah ini.



Gambar 4.4 Halaman Forum

4.1.2.1.5 Halaman Tambah Topik

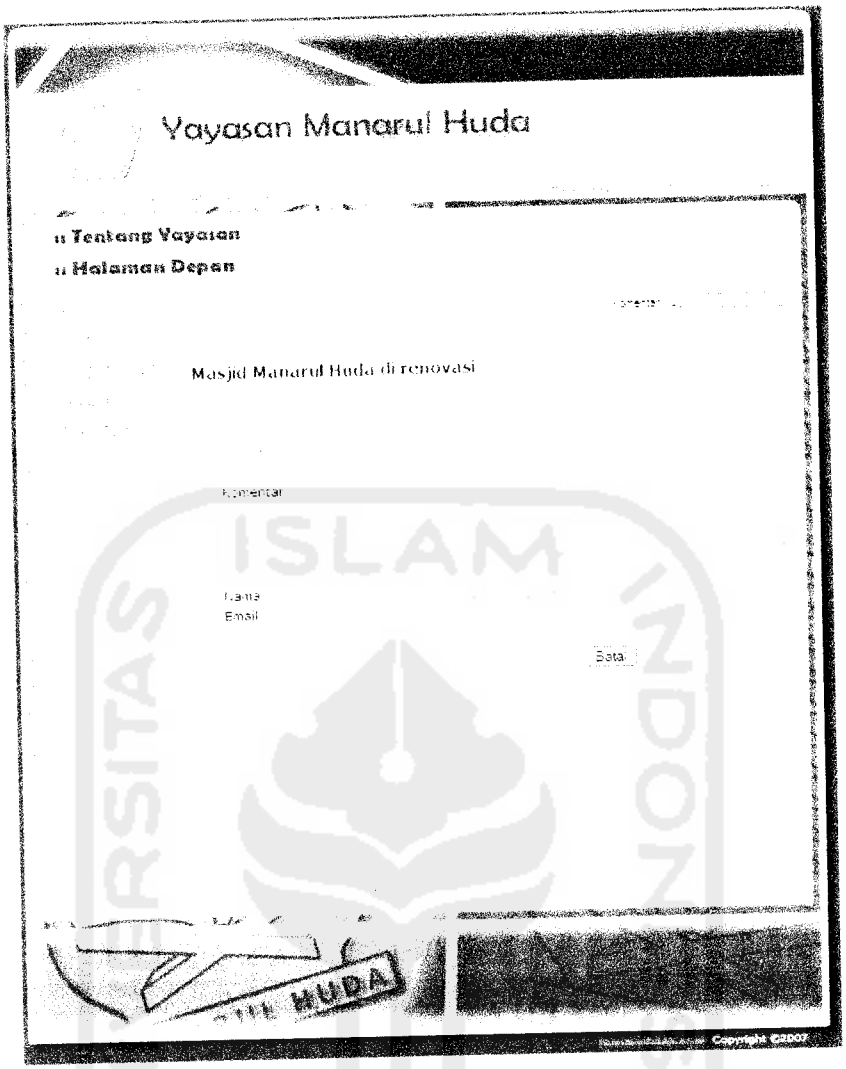
Pada pada halaman tambah topik ini, pengunjung dapat menambahkan topik baru pada forum. Data yang dimasukan yaitu; topik baru, nama dan email. Halaman tambah topik dapat dilihat pada gambar 4.5 dibawah ini.



Gambar 4.5 Halaman Tambah Topik

4.1.2.1.6 Halaman Tambah Komentar

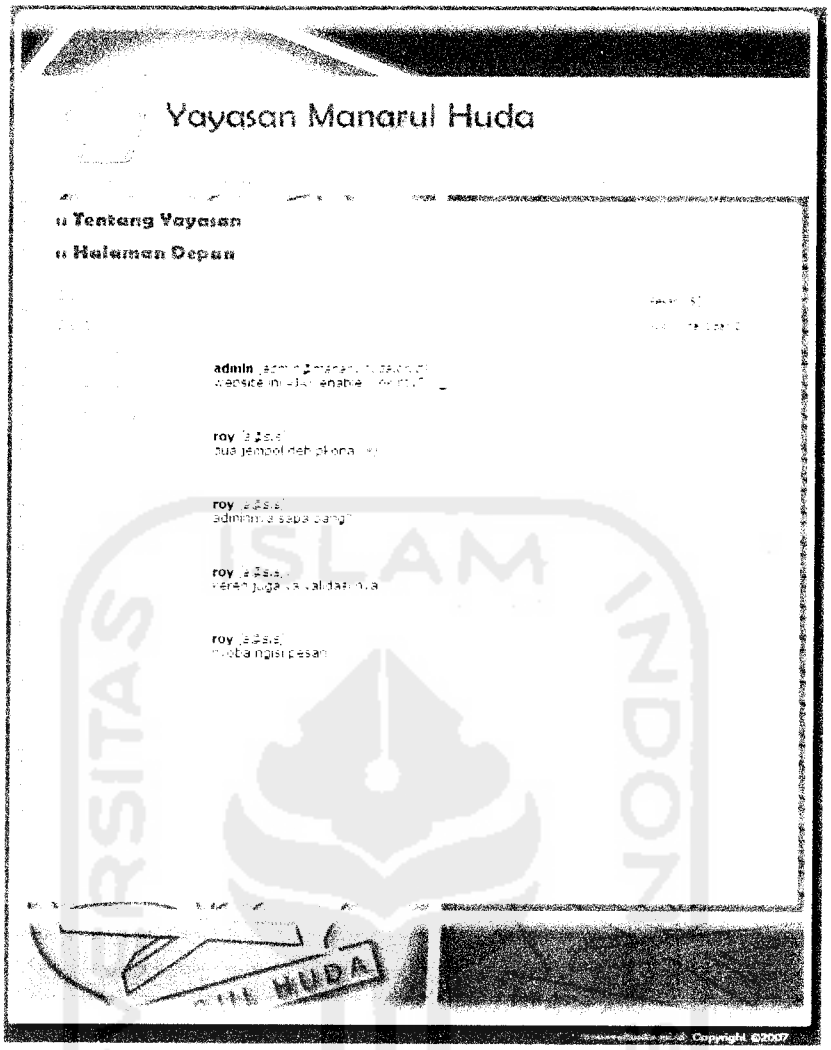
Halaman ini digunakan untuk memberikan komentar ke dalam topik yang sudah dibuat. Input data yang dimasukan yaitu; komentar, nama dan email. Halaman tambah komentar dapat dilihat pada gambar 4.6 berikut ini.



Gambar 4.6 Halaman Tambah Komentar

4.1.2.1.7 Halaman Buku Tamu

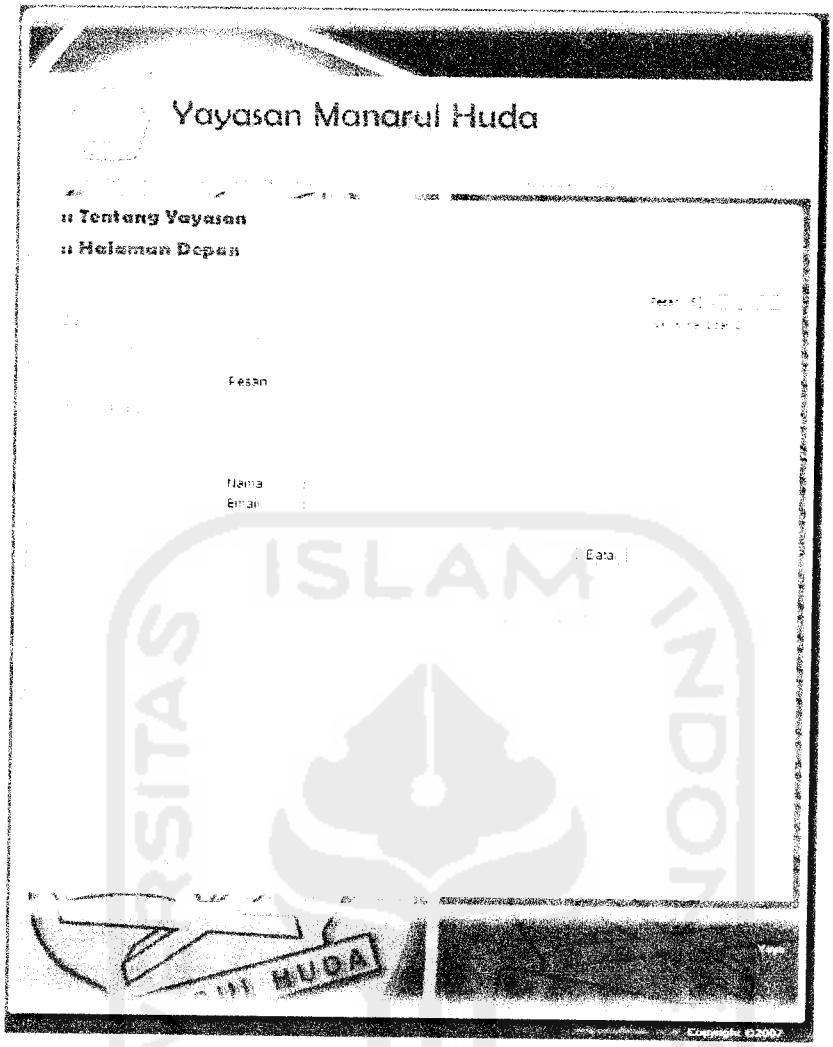
Pada halaman ini berisi pesan-pesan pada buku tamu yayasan yang telah dimasukan oleh pengunjung. Halaman buku tamu dapat dilihat pada gambar 4.7 berikut ini..



Gambar 4.7 Halaman Buku Tamu

4.1.2.1.8 Halaman Tambah Buku Tamu

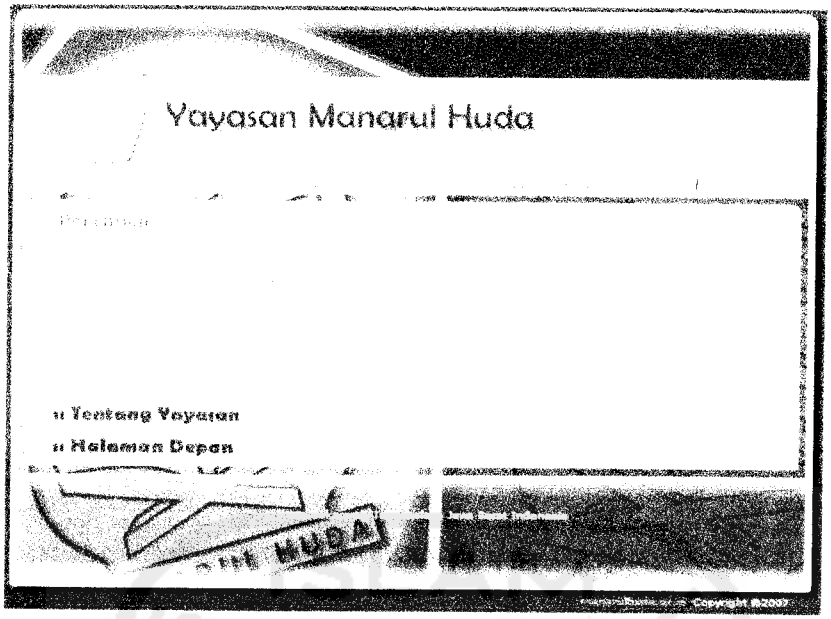
Halaman ini berisi *form* tambah buku tamu yang bisa digunakan oleh pengunjung. Input data yang dimasukan adalah pesan, nama dan email. Halaman tambah forum dapat dilihat pada gambar 4.8 berikut ini.



Gambar 4.8 Halaman Tambah Buku Tamu

4.1.2.1.9 Halaman Pencarian

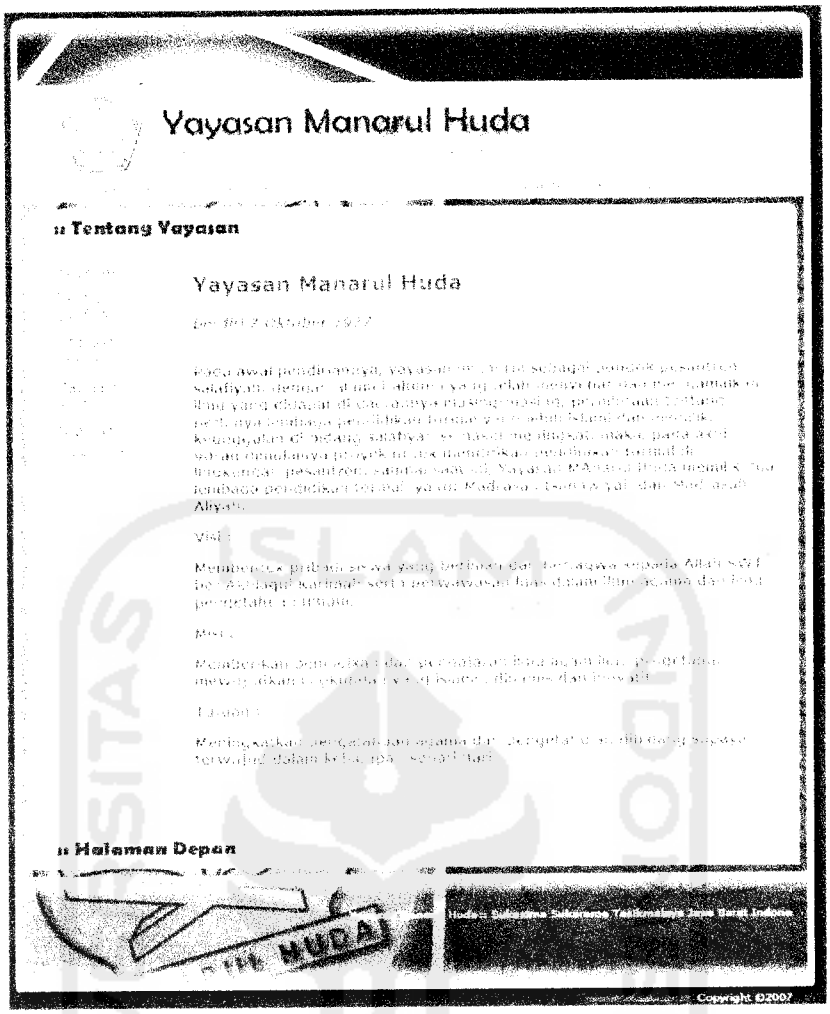
Pada halaman pencarian ini terdapat *form* isian kata kunci judul berita, sehingga pengunjung dapat melakukan pencarian dengan memasukan judul dari berita yang dicari. Halaman pencarian dapat dilihat pada gambar 4.9 dibawah ini.



Gambar 4.9 Halaman Pencarian

4.1.2.1.10 Halaman Yayasan

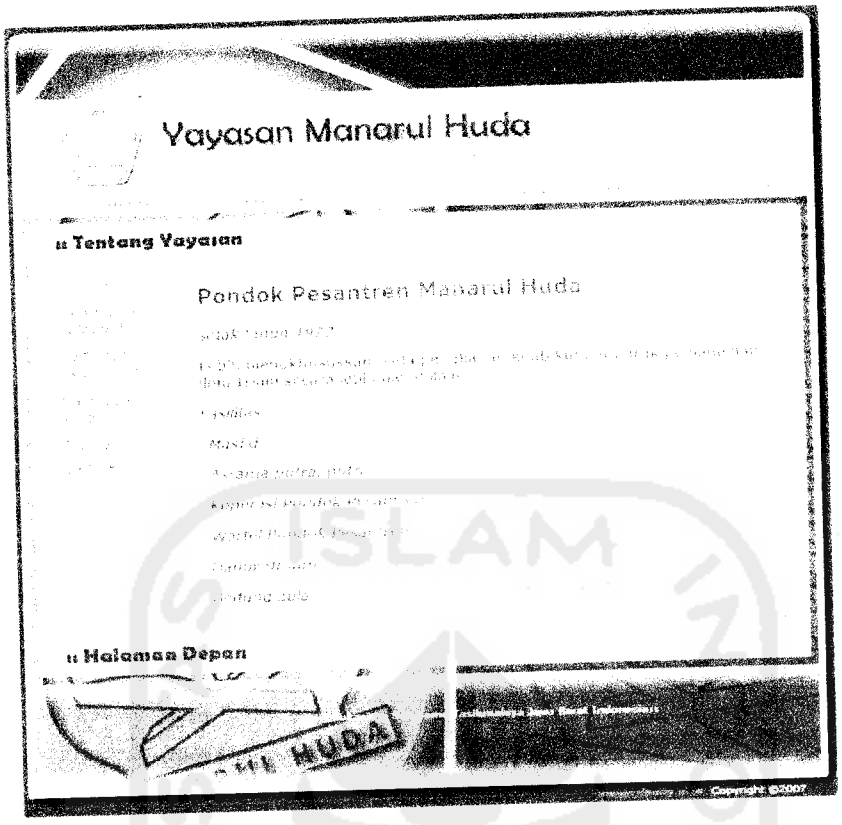
Halaman ini bersifat umum yang dapat diakses oleh setiap pengunjung, Halaman ini menjelaskan latar belakang dan tujuan dari yayasan Manarul Huda. Halaman Yayasan dapat dilihat pada gambar 4.10 berikut ini.



Gambar 4.10 Halaman Yayasan

4.1.2.1.11 Halaman Pendidikan Salafiyah

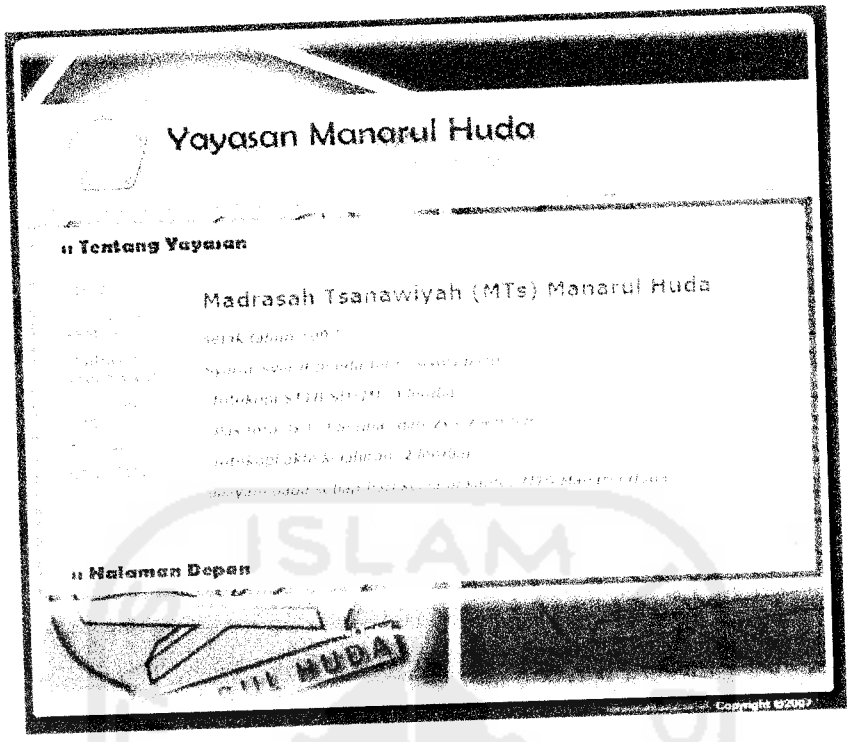
Halaman ini berisi tentang informasi penyelenggaraan pendidikan salafiyah yang diadakan oleh Yayasan Manarul Huda. Pendidikan Salafiyah adalah pendidikan Islam yang berorientasi pada pemahaman kitab-kitab ilmu Islam, atau sering disebut dengan "*kitab kuning*". Halaman pendidikan salafiyah ditunjukkan pada gambar 4.11 dibawah ini.



Gambar 4.11 Halaman Pendidikan Salafiyah

4.1.2.1.12 Halaman Madrasah Tsanawiyah

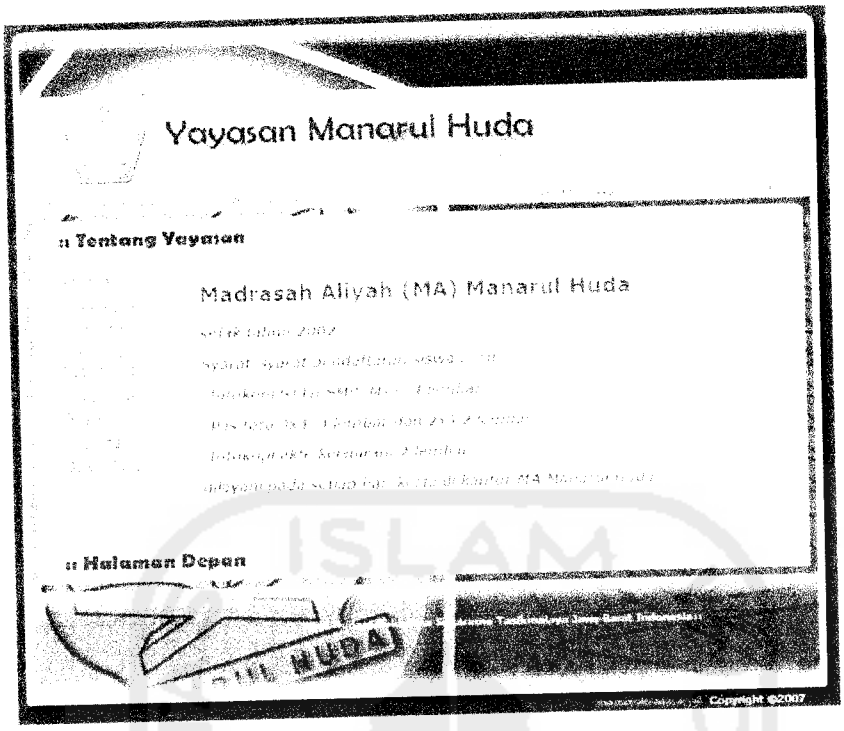
Halaman ini berisi tentang Madrasah Tsanawiyah (MTs) Manarul Huda. MTs adalah sekolah pendidikan formal tingkat menengah setingkat SMP. Penyelenggaraan pendidikannya diatur oleh Depertemen Agama RI dan statusnya telah diakui. Berikut tampilan halaman madrasah tsanawiyah pada gambar 4.12.



Gambar 4.12 Halaman Madrasah Tsanawiyah

4.1.2.1.13 Halaman Madrasah Aliyah

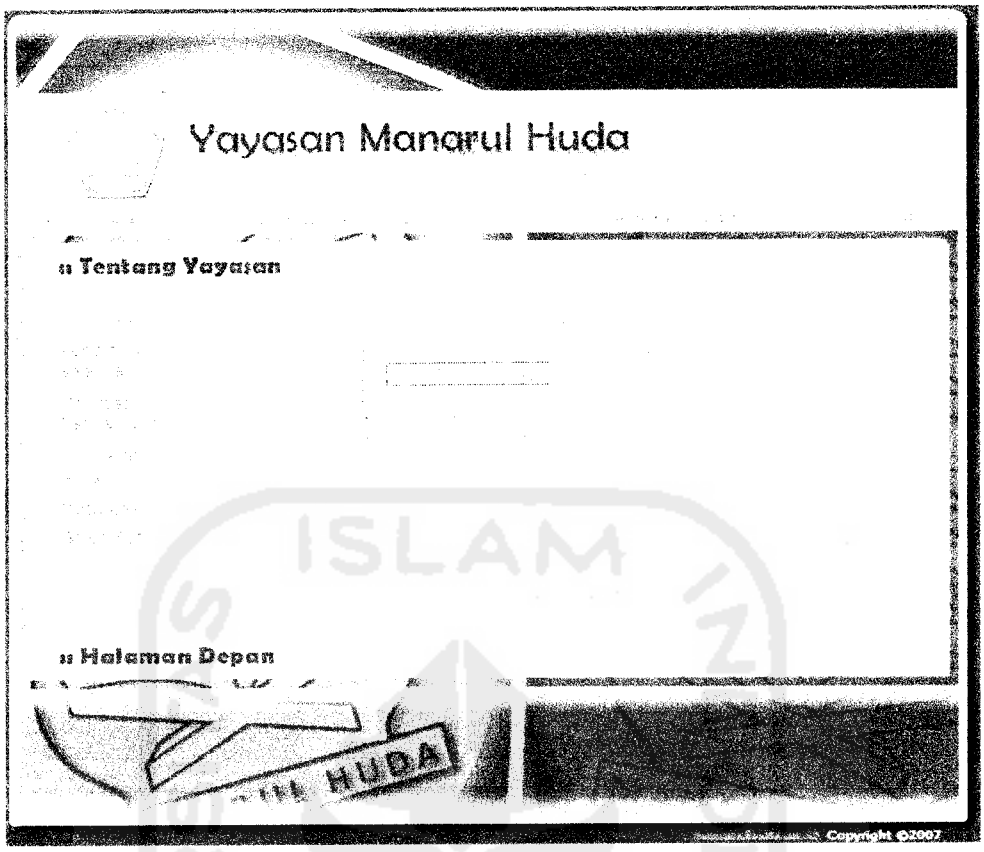
Halaman ini berisi informasi tentang Madrasah Aliyah (MA) Manarul Huda. MA adalah jenjang pendidikan tingkat atas setingkat SMU. Merupakan jenjang lanjutan setelah MTs. Halaman madrasah aliyah ditunjukkan pada gambar 4.13 dibawah ini.



Gambar 4.13 Halaman Madrasah Aliyah

4.1.2.1.14 Halaman Publikasi

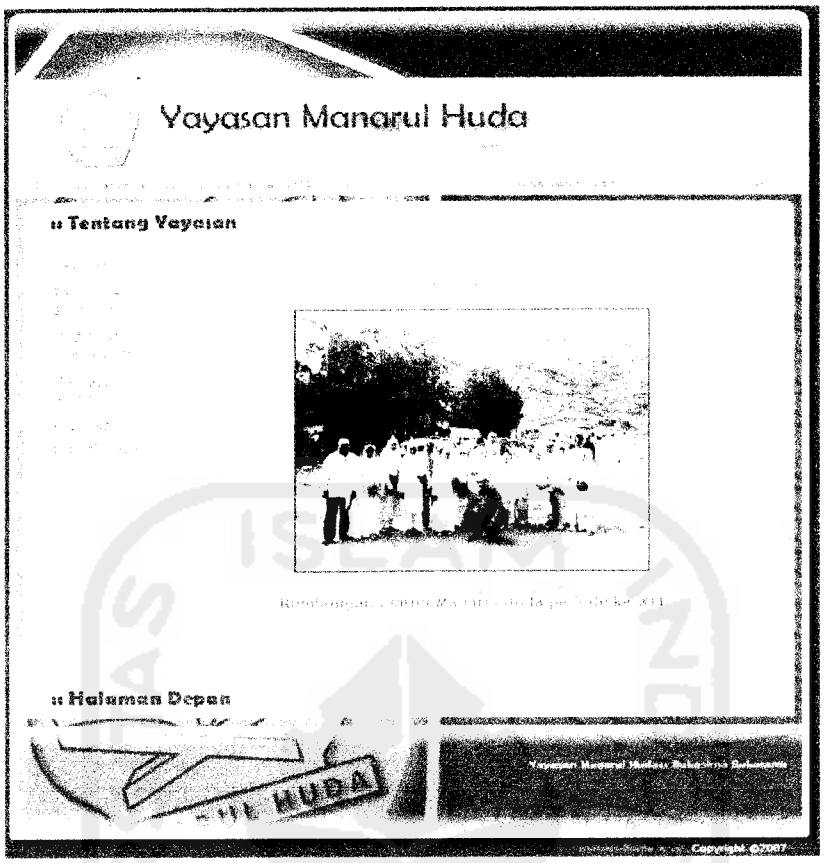
Halaman ini berisi publikasi dari Yayasan Manarul Huda yang terdiri dari dokumen-dokumen yang dapat di-*download* oleh pengunjung. Halaman publikasi ditunjukkan pada gambar 4.14 dibawah ini.



Gambar 4.14 Halaman Publikasi

4.1.2.1.15 Halaman Galeri

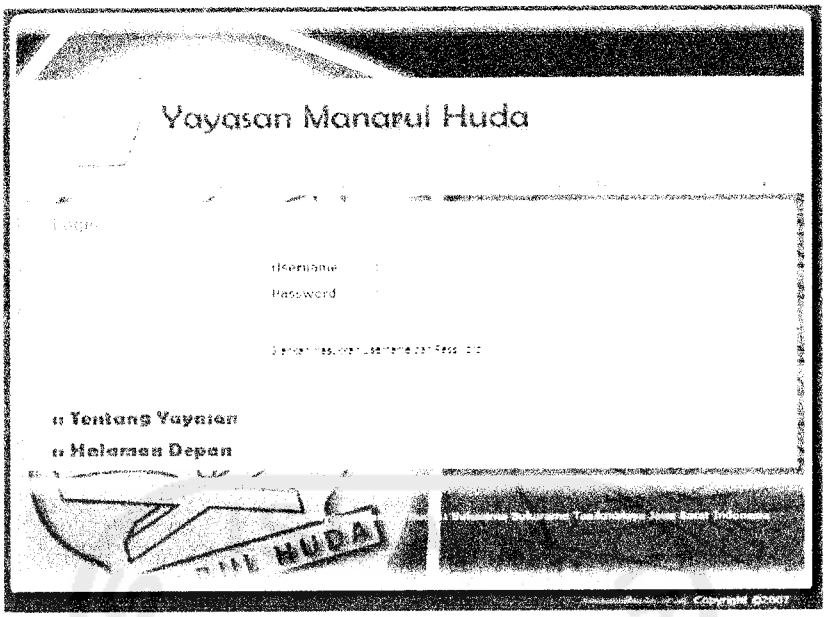
Halaman ini berisi gambar dan foto dokumentasi dari Yayasan Manarul Huda. Halaman galeri dapat dilihat pada gambar 4.15 dibawah ini.



Gambar 4.15 Halaman Galeri

4.1.2.2 Halaman *Login*

Halaman ini digunakan oleh admin untuk masuk ke halaman *administrator* dan oleh anggota untuk masuk ke halaman anggota.. Diperlukan *username* admin dengan *password* admin dan *username* anggota dengan *password* anggota. Halaman *login* dapat dilihat pada gambar 4.16 dibawah ini.



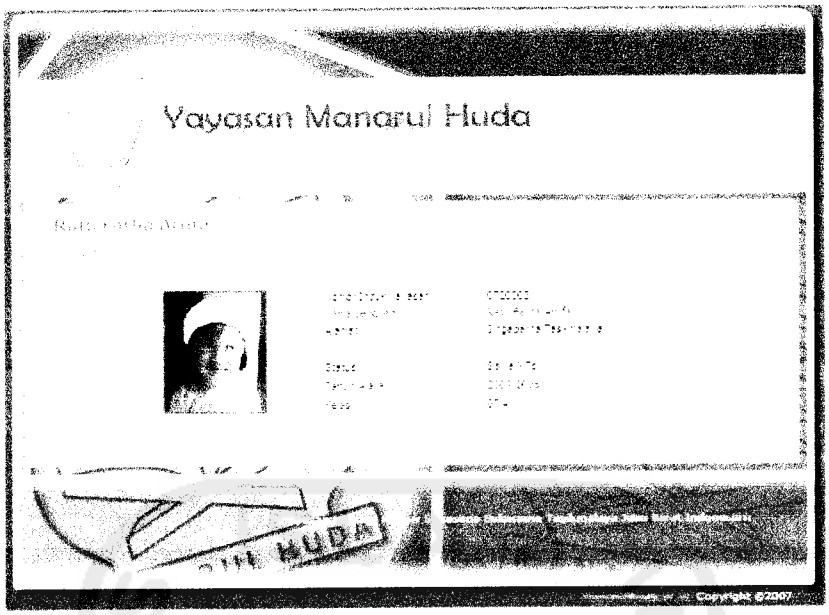
Gambar 4.16 Halaman *Login*

4.1.2.3 Halaman Anggota

Halaman ini bersifat khusus, dan hanya bisa diakses oleh anggota yang terdaftar. Terdapat beberapa halaman yang dapat diakses setiap anggota, diantaranya:

4.1.2.3.1 Halaman Status Anggota

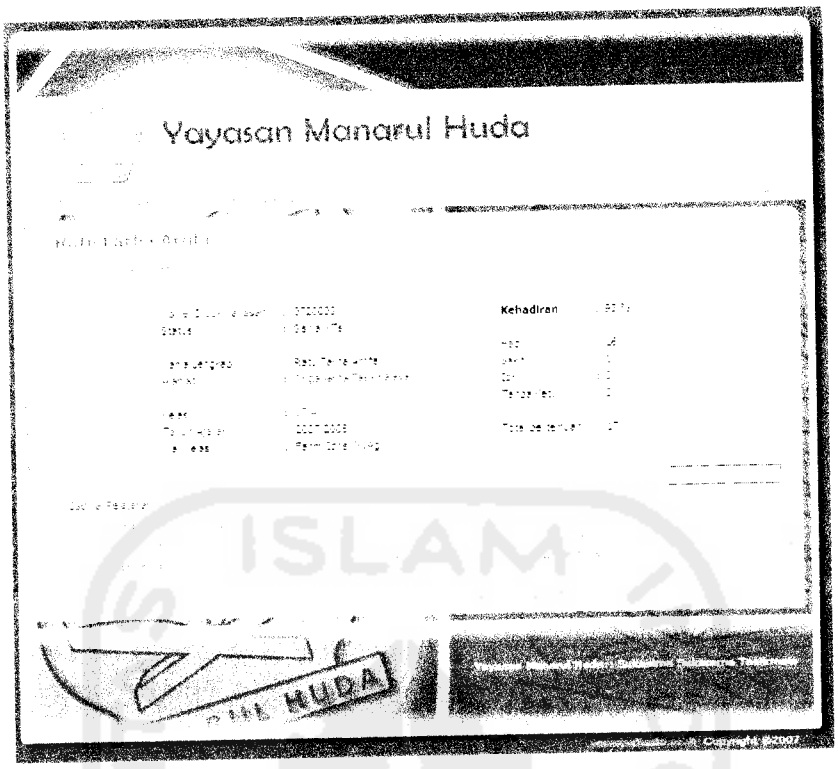
Halaman ini berisi tentang status anggota di yayasan, menampilkan data No. Induk Yayasan, Nama, Alamat, dan Status. Halaman status dapat dilihat pada gambar 4.17.



Gambar 4.17 Halaman Status

4.1.2.3.2 Halaman Akademik

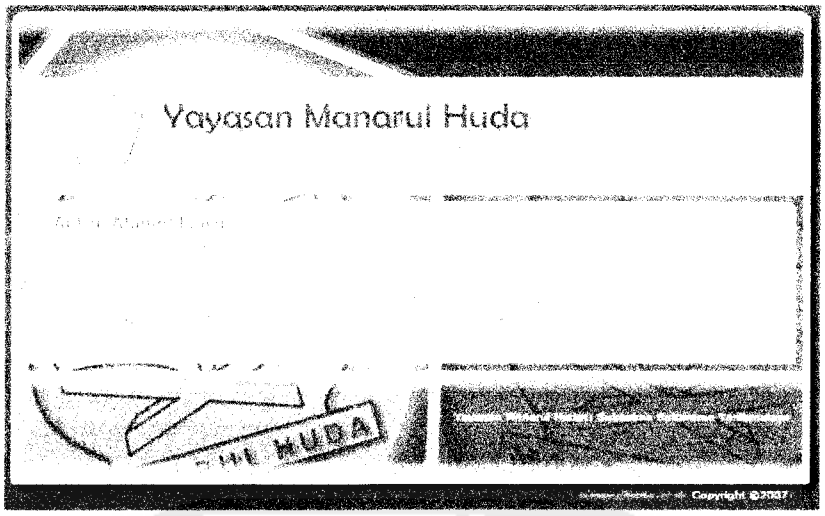
Halaman ini berisi tentang informasi akademik anggota yang berupa kehadiran dan jadwal pelajaran, Halaman akademik dapat dilihat pada gambar 4.18 dibawah ini.



Gambar 4.18 Halaman Akademik

4.1.2.3.3 Halaman Tagihan

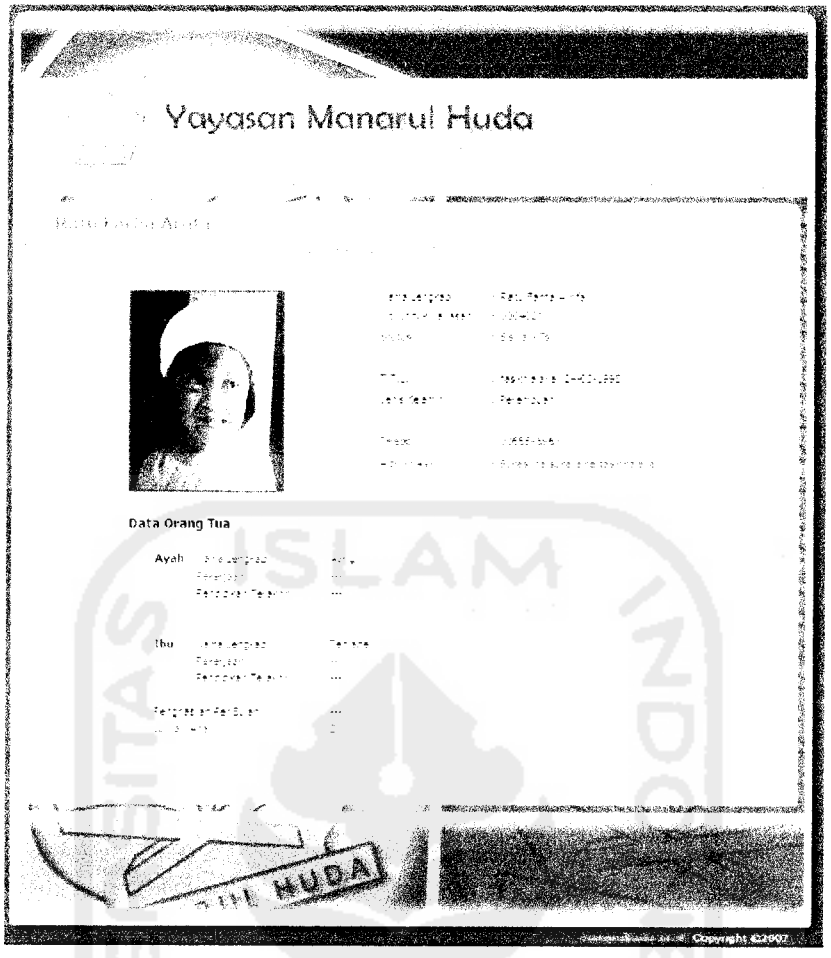
Halaman ini berisi informasi tagihan yang harus dibayar oleh setiap anggota. Halaman tagihan anggota ditunjukkan pada gambar 4.19 dibawah ini.



Gambar 4.19 Halaman Tagihan

4.1.2.3.4 Data Pribadi

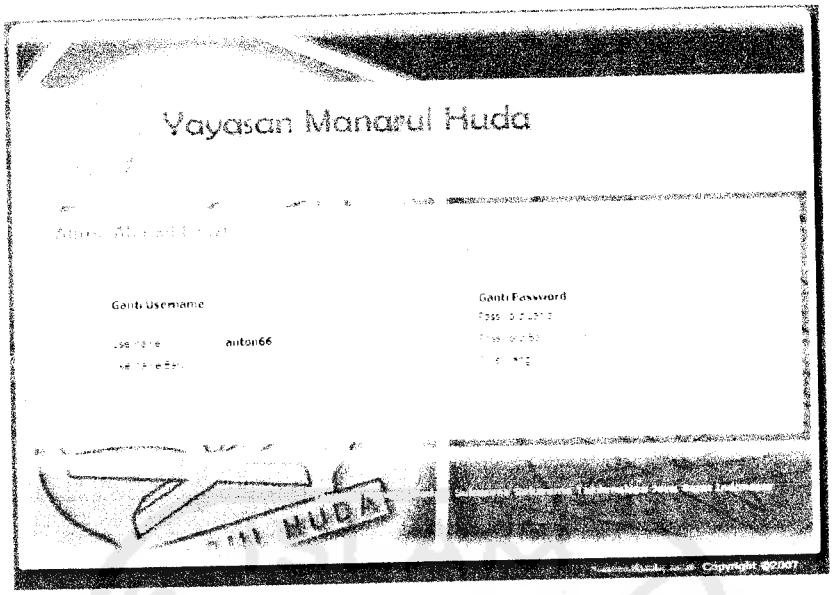
Berisi data pribadi dari anggota yang bersangkutan. Halaman data pribadi dapat dilihat pada gambar 4.20 dibawah ini.



Gambar 4.20 Halaman Data Pribadi

4.1.2.3.5 Pengaturan

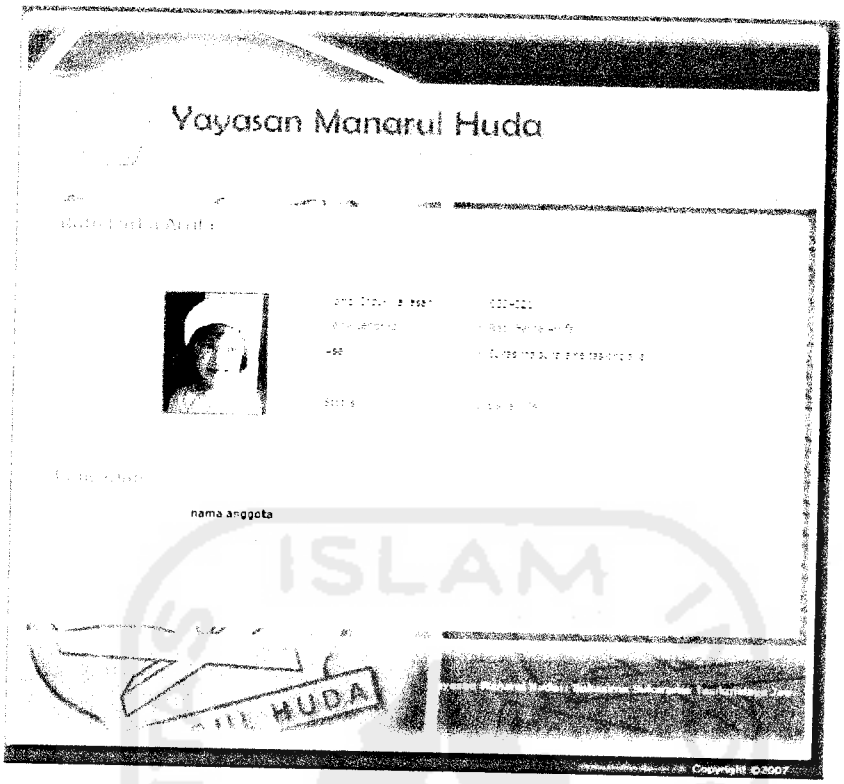
Berisi pengaturan data *login* dari anggota, anggota dapat mengubah *username* dan *password*. Halaman pengaturan dapat dilihat pada gambar 4.21.



Gambar 4.21 Halaman Pengaturan

4.1.2.3.6 Pencarian Anggota

Pada halaman ini nggota dapat mencari data anggota lain dengan memasukan kata kunci nama anggota. Data yang dihasilkan berupa nama, foto anggota, alamat anggota dan status anggota. Halaman pencarian anggota dapat dilihat pada gambar 4.22 dibawah ini.



4.1.

Gambar 4.22 Halaman Pencarian Anggota

daf

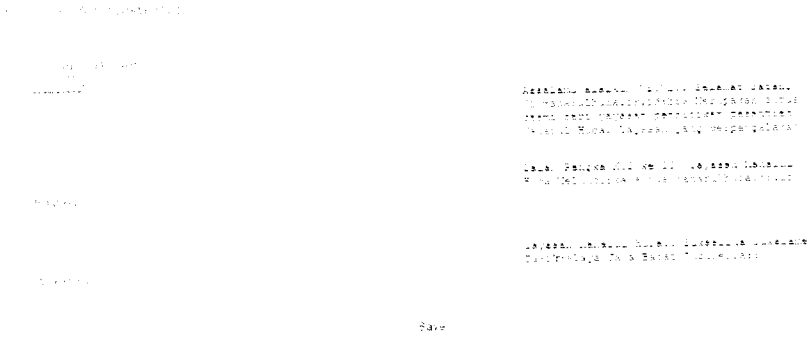
ber

4.1.2.4 Halaman *Administrator*

Halaman ini bersifat khusus, dan hanya bisa diakses oleh admin yang mempunyai hak akses dalam menambah, menghapus, ataupun meng-*update* data. Admin diharuskan *login* terlebih dahulu untuk mengakses halaman ini, diperlukan *username* admin dan *password admin*.

4.1.2.4.1 Halaman Beranda Admin

Halaman ini berisi tentang pengaturan dari halaman depan *website*, terdiri dari input salam, snapshot dan *running text*. Halaman Beranda admin dapat dilihat pada gambar 4.23.



Gambar 4.23 Halaman Beranda Admin

4.1.2.4.2 Halaman Berita Admin

Halaman ini berisi pengaturan berita yayasan pada *website*, pada halaman ini admin dapat melihat berita, menambah berita, mengubah berita dan menghapus berita. Halaman berita admin dapat dilihat pada gambar 4.24

Gambar 4.24 Halaman Berita Admin

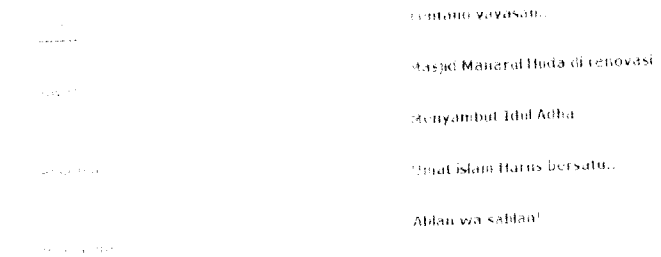
4.1.2.4.3 Halaman Pengumuman Admin

Halaman ini berisi pengaturan pengumuman yayasan pada *website*, admin dapat melihat, menambah dan menghapus pengumuman. Halaman pengumuman admin dapat dilihat pada gambar 4.25.

Gambar 4.25 Halaman Pengumuman Admin

4.1.2.4.4 Halaman Forum Admin

Halaman ini berfungsi sebagai halaman pengaturan forum. Admin dapat melihat, menambah, dan menghapus topik dan komentar. Halaman forum admin dapat dilihat pada gambar 4.26.



Gambar 4.26 Halaman Forum Admin

4.1.2.4.5 Halaman Buku Tamu Admin

Halaman ini berisi pengaturan buku tamu yayasan. Admin dapat melihat, menambah dan menghapus data buku tamu. Halaman buku tamu admin dapat dilihat pada gambar 4.27.

Gambar 4.27 Halaman Buku Tamu Admin

4.1.2.4.6 Halaman Publikasi Admin

Halaman ini berisi pengaturan data publikasi yayasan. Admin dapat menambah dan menghapus data publikasi. Halaman publikasi admin dapat dilihat pada gambar 4.28.

Gambar 4.28 Halaman Publikasi Admin

4.1.2.4.7 Halaman Galeri Admin

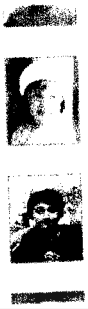
Halaman ini berisi pengaturan dari galeri yayasan. Admin dapat melihat, menambah dan menghapus galeri. Halaman galeri admin dapat dilihat pada gambar 4.48 dibawah ini.



Gambar 4.29 Halaman Galeri Admin

4.1.2.4.8 Halaman Data Induk

Halaman ini menjadi inti dari pengaturan anggota yayasan, pada halaman ini admin dapat melihat daftar anggota secara keseluruhan, menambah anggota, mengedit anggota, menghapus anggota, melihat data pribadi, mengedit data pribadi, melihat dan mengedit akademik anggota, menambah dan menghapus kelas, menambah dan menghapus jadwal pelajaran serta menambah dan menghapus tagihan. Halaman data induk dapat dilihat pada gambar 4.30.



Gambar 4.30 Halaman Data Induk

4.1.2.4.9 Halaman Tambah Anggota

Pada halaman ini admin dapat menambah anggota yayasan. Halaman tambah anggota dapat dilihat pada gambar 4.31.

Gambar 4.31 Halaman Tambah Anggota

4.1.2.4.10 Halaman Lihat Kelas

Pada halaman ini, amin dapat melihat, menambah, mengedit dan menghapus kelas dari Madrasah Tsanawiyah dan Madrasah aliyah. Halaman lihat kelas dapat dilihat pada gambar 4.32.

Gambar 4.32 Halaman Lihat Kelas

4.1.2.4.11 Halaman Lihat Jadwal

Pada halaman ini admin dapat melihat, menambah dan menghapus jadwal untuk setiap kelas pada Madrasah Tsanawiyah dan Madrasah Aliyah. Halaman lihat jadwal dapat dilihat pada gambar 4.33.

Gambar 4.33 Halaman Lihat Jadwal

4.1.2.4.12 Halaman Tagihan

Pada halaman ini admin dapat menambah dan menghapus tagihan bagi Santri. Madrasah Tsanawiyah dan Madrasah Aliyah. Halaman tagihan dapat dilihat pada gambar 4.34.

Gambar 4.34 Halaman Tagihan

4.1.2.4.13 Halaman Akademik

Pada halaman ini, admin dapat melihat data akademik anggota pada Madrasah Tsanawiyah dan Madrasah Aliyah, admin dapat melakukan *update* kehadiran dan *update* kelas anggota. Halaman akademik dapat dilihat pada gambar 4.35.

Gambar 4.35 Halaman Akademik

4.1.2.4.14 Halaman Pencarian

Pada halaman ini admin dapat melakukan pencarian anggota pada. Pencarian dilakukan dengan kata kunci nama anggota. Halaman pencarian dapat dilihat pada gambar 4.36.

Gambar 4.36 Halaman Pencarian

4.1.2.4.15 Halaman Ganti *Username*

Pada halaman ini, admin dapat mengganti *username* yang digunakan untuk *Login*. Halaman ganti *username* dapat dilihat pada gambar 4.37.

Gambar 4.37 Halaman Ganti *Username*

4.1.2.4.16 Halaman Ganti *Password* Admin

Pada halaman ini, admin dapat mengganti *password* yang digunakan untuk *Login*. Halaman ganti *password* dapat dilihat pada gambar 4.38.

Gambar 4.38 Halaman Ganti *Password* Admin

4.2 Pengujian Program

Pada tahap analisis kinerja perangkat lunak dijelaskan tentang pengujian *website* dengan penerapan teknologi AJAX. Pengujian dilakukan dengan kompleks dan diharapkan dapat diketahui kekurangan-kekurangan dari sistem untuk kemudian diperbaiki sehingga kesalahan dari sistem dapat diminimalisasi atau bahkan dihilangkan. Pengujian sistem ini dilakukan untuk mendapatkan hasil yang akurat. Pengujian *website* ini dilakukan dengan mengisi form masukan yang telah ditampilkan pada bagian implementasi sistem.

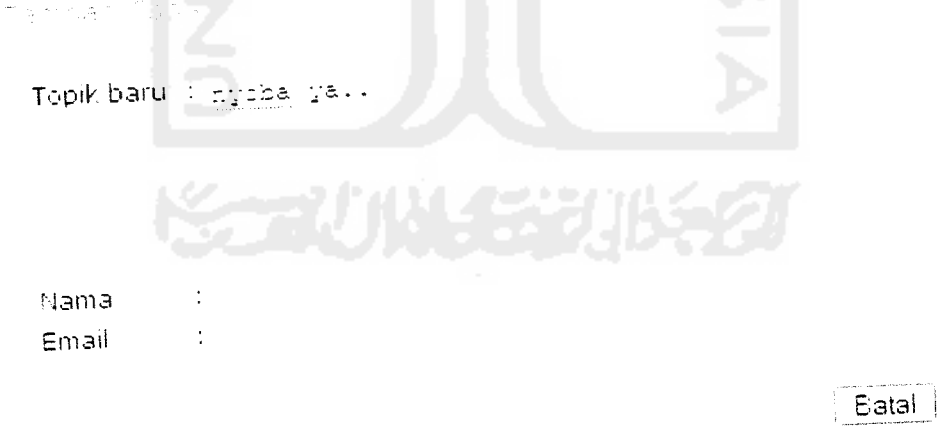
4.2.1 Analisis Kinerja Sistem.

4.2.1.1 Penanganan Kesalahan

Perangkat lunak ini dibuat dengan memperhatikan sisi komunikatif, artinya mudah dimengerti oleh pengguna. Jika terdapat kesalahan-kesalahan pada pemasukan data atau pilihan pada beberapa proses yang akan dilakukan maka sistem akan memberikan tanggapan (*feed back*) kepada pengguna baik berupa jendela dialog (*message box*) ataupun berupa pesan kesalahan (*error message*). Ada beberapa tipe dari penanganan kesalahan, antara lain :

1. Penanganan Kesalahan Masukan Data

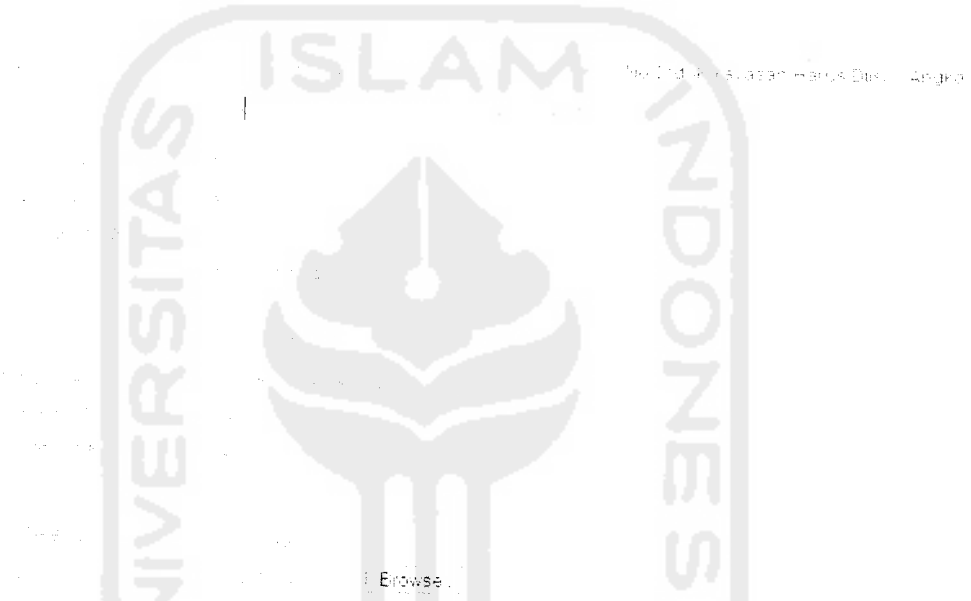
Penanganan kesalahan masukan data ini dilakukan untuk menangkap *error* yang terjadi ketika salah satu *field* pada *form* masukan kosong. Penanganan kesalahan masukan data ini terdapat pada proses tambah topik forum. Jika nama dikosongkan maka akan muncul pesan kesalahan seperti pada gambar 4.39.



Gambar 4.39 Tampilan Pesan Kesalahan Jika Nama Tidak Diisi

2. Penanganan Kesalahan Input Tipe Data

Penanganan kesalahan input tipe data ini dilakukan untuk menangkap *error* yang terjadi ketika data yang dimasukan tidak sesuai dengan tipe data yang seharusnya. Contoh penanganan kesalahan input data terdapat pada input tambah anggota, jika no.induk yayasan diisi dengan *string*, maka akan muncul pesan kesalahan seperti pada gambar 4.40.

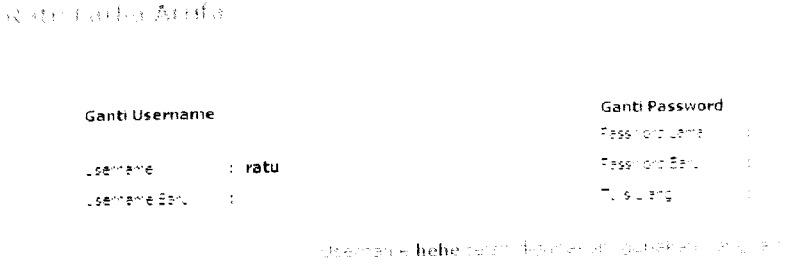


Gambar 4.40 Tampilan Pesan Kesalahan Jika No.Induk Diisi String.

3. Penanganan Kesalahan Input Data Telah Ada

Penanganan kesalahan input data telah ada dilakukan untuk menangkap *error* yang terjadi ketika data yang dimasukan telah ada atau telah digunakan pada *database*, sehingga menghindarkan dari adanya *redundancy* data. Contoh penanganan kesalahan input data telah ada terdapat pada input ganti *username* anggota, jika anggota

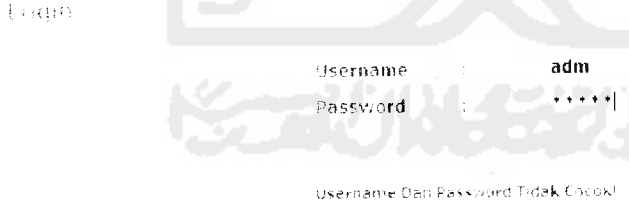
memasukan *username* baru yang telah digunakan, maka akan muncul pesan kesalahan sepert pada gambar 4.41.



Gambar 4.41 Pesan Kesalahan *Username* Telah Ada

4. Penanganan Kesalahan Input Data Tidak Ditemukan

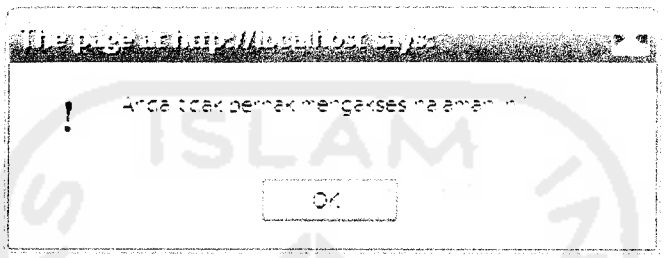
Penanganan kesalahan input data tidak ditemukan dilakukan untuk menangkap *error* yang terjadi ketika data yang dimasukan tidak ditemukan. Contoh penanganan kesalahan input data tidak ditemukan terdapat pada form *login*, jika kombinasi username dan *password* tidak ditemukan pada *database* , maka akan muncul pesan kesalahan sepert pada gambar 4.42.



Gambar 4.42 Pesan Kesalahan Data Tidak Ditemukan

5. Penanganan Kesalahan Hak Akses

Penanganan kesalahan hak akses dilakukan untuk menangkap *error* yang terjadi ketika pengunjung akan mengakses halaman khusus bagi admin dan anggota. Ketika kesalahan hak akses ini terjadi, maka akan muncul *message box* seperti pada gambar 4.43.



Gambar 4.43 Message Box Kesalahan Hak Akses

4.2.1.2 Pengujian dan Analisis

Pada tahap pengujian dan analisis sistem ini, dilakukan pembandingan antara kebenaran masukan serta kesesuaian program dengan kebutuhan sistem.

1. Masukan Proses *Login*

Pada form input *login,* diberikan contoh masukan seperti dibawah ini untuk menguji *output* yang dihasilkan.

Username : adm

Password : adm

Tampilan *login* dengan *password="adm"* dan *username="adm"* dapat dilihat pada gambar 4.44 berikut.



Gambar 4.44 Antar Muka Masukan *Login* Username dan *Password* Cocok.

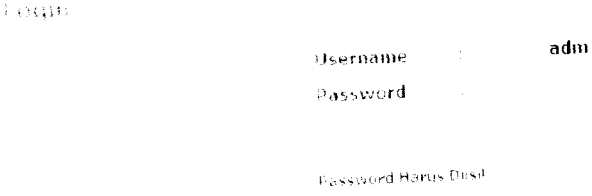
Pada gambar 4.44 diatas dapat di lihat bahwa apabila *username* dan *password* cocok, maka akan ditampilkan tombol untuk masuk keahalaman selanjutnya.

Ketika masukan *username* dan *password* tidak cocok, maka tampilannya adalah seperti pada gambar 4.45 sebagai berikut:



Gambar 4.45 Antar Muka Masukan *Login Username* Dan *Password* Tidak Cocok.

Ketika masukan *password* tidak disi (kosong), maka tampilannya adalah seperti pada gambar 4.46 berikut ini.



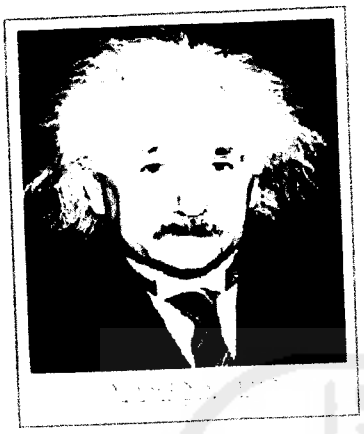
Gambar 4.46 Antar Muka Masukan *Login Password* Kosong

2. Masukan Data Anggota

Pada form tambah anggota, diberikan contoh masukan sebagai berikut;

No.Induk Yayasan	: 0506029
Username	: 0506029
Password	: 0506029
Nama Lengkap	: Albert Einstein
Status	: Siswa MA
Tempat Lahir	: Tasikmalaya
Tanggal Lahir	: 01-01-1980
Jenis Kelamin	: Laki-laki
Alamat Asal	: Tasikmalaya
No. Telepon	: 0123456
Foto	: Einstein3.jpeg

Hasil dari masukan data tersebut dapat dilihat pada gambar 4.47.



Gambar 4.47 Antar Muka Hasil Masukan Tambah Anggota

3. Masukan Data Kelas

Pada form input tambah kelas, diberikan contoh sebagai berikut

Kelas : 07 C

Wali Kelas : Suherman, Drs.

Hasil dari masukan tambah kelas diatas dapat dilihat pada gambar 4.48.



Gambar 4.48 Antar Muka Hasil Masukan Kelas

4. Masukan Data Topik

Pada form masukan data topik forum, diberikan contoh masukan sebagai berikut:

Topik Baru : Tentang yayasan..

Nama : Administrator

Email : admin@manarulhuda.or.id

Hasil dari masukan diatas dapat dilihat pada gambar 4.49.



Gambar 4.49 Antar Muka Hasil Masukan Data Topik Forum

5. Masukan Data Komentar

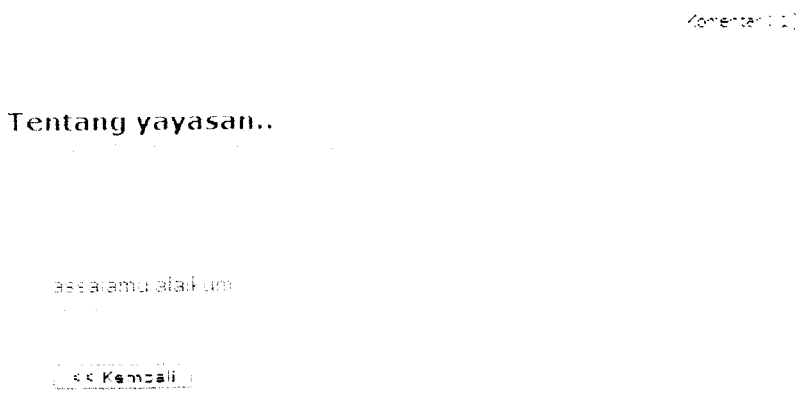
Pada form masukan data komentar forum, diberikan contoh masukan sebagai berikut:

Pesan : assalamu'alaikum..

Nama : Ikhwan

Email : email@email.com

Hasil dari masukan data komentar ini, dapat dilihat pada gambar 4.50.



Gambar 4.50 Antarmuka Hasil Masukan Data Komentar Forum

6. Masukan Data Berita

Pada form masukan data berita, diberikan contoh masukan data sebagai berikut;

Judul : Menyambut Idul Adha

Ringkasan : Menyambut Idul Adha tahun ini, Yayasan Manarul Huda akan mengadakan bazaar daging qurban..

Selengkapnya : Menyambut Idul Adha tahun ini, Yayasan Manarul Huda akan mengadakan bazaar daging qurban.. dengan persediaan hewan qurban sebanyak 5 ekor sapi dan 10 ekor domba.

Hasil dari data berita diatas dapat dilihat pada gambar 4.51

Menyambut Idul Adha

Gambar 4.51 Antar Muka Masukan Berita

7. Masukan Galeri

Pada form masukan data galeri ini, diberikan contoh masukan data sebagai berikut:

Keterangan : Rombongan Umroh Manarul Huda periode ke- XII

Gambar : umroh1.jpeg

Hasil dari masukan data galeri diatas, dapat dilihat pada gambar 4.52



Gambar 4.52 Antar Muka Masukan Galeri

4.3 Perbandingan *Website* AJAX dengan *Website* Tradisional

Implementasi penerapan teknologi AJAX pada *website* ini digunakan pada 4 proses, yaitu; proses *load* halaman, proses validasi masukan data, proses *login* dan proses pencarian.

4.3.1 Perbandingan Proses *Load* Halaman

Pada *website* dengan penerapan teknologi AJAX, halaman yag dipanggil dimasukan (di-*load*) ke dalam sebuah *tag* id tertentu, sehingga proses *loading* akan jauh lebih cepat, karena hanya me-*load tag* id yang ditentukan saja, bukan me-*load* halaman secara keseluruhan.

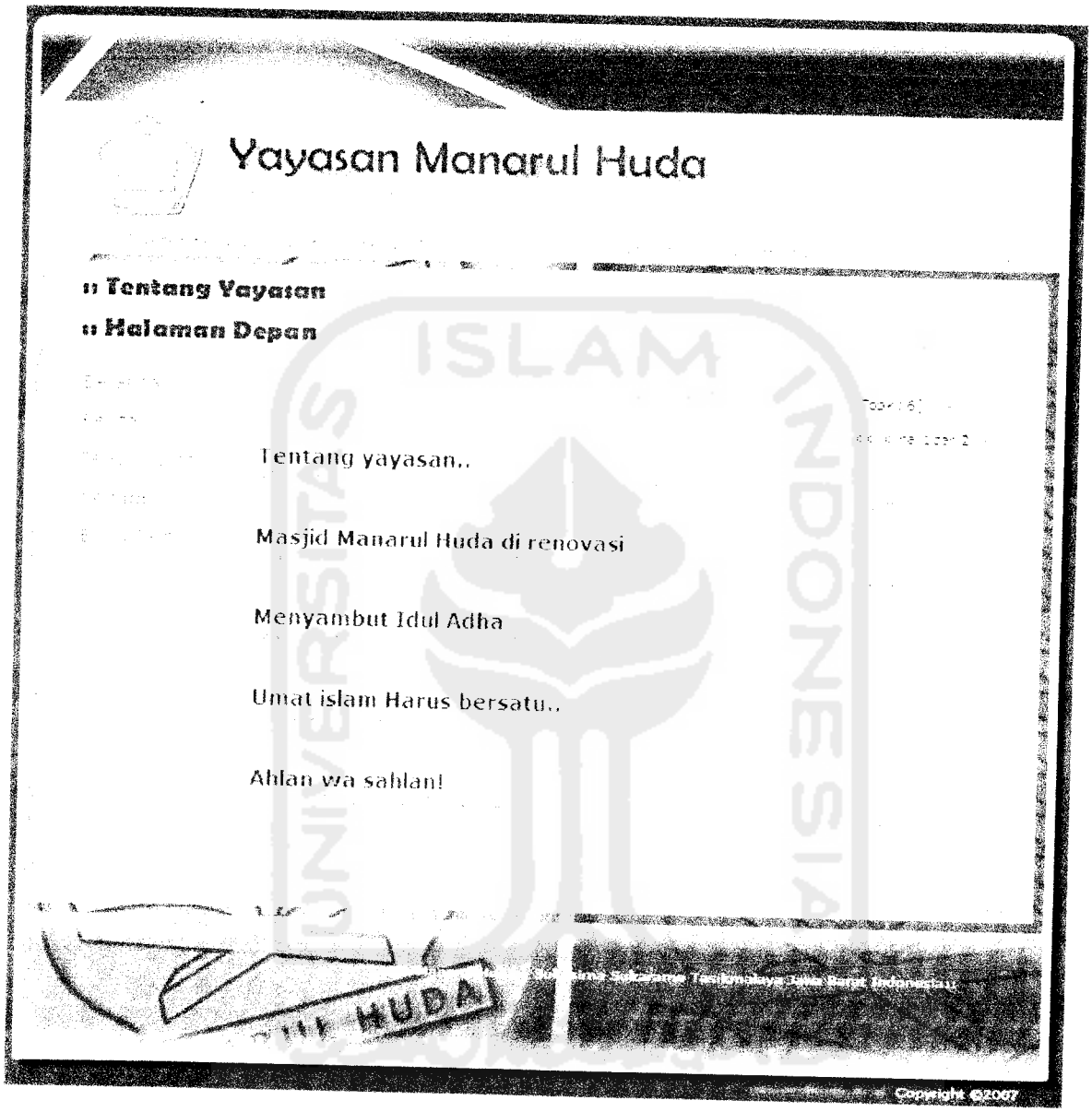
Di dalam *website* ini, hampir semua halaman di-*load* menggunakan penerapan teknologi AJAX. Halaman yang dipanggil akan diisikan (di-*load*) ke dalam tag id "content_hal_depan" yang terdapat pada index.php. pemanggilan halaman menggunakan obyek XMLHttpRequest yang terdapat pada *function* tampil_halaman_ajax() dengan dipicu oleh event Onclick, Onload dan OnKeyUp. Pemanggilan halaman dilakukan dengan menentukan halaman yang akan di-*load* dan menentukan *tag* id yang akan menampilkan halaman tersebut. Contoh *Tag event* yang memicu *function* tampil_halaman_ajax() dapat dilihat pada listing 4.1 dibawah ini.

```
<a href="javascript:;"
onclick="tampil_halaman_ajax('forum.php','content_hal_depan');">
Forum</a>
```

Listing 4.1 Tag Event OnClick=" tampil_halaman_ajax()"

Pada listing 4.1, ketika kita meng-klik link "Forum", *function* tampil_halaman_ajax() akan me-*request* halaman forum.php dengan menggunakan obyek XMLHttpRequest, dan menampilkannya ke dalam tag id "content_hal_depan" yang terdapat pada index.php.

Halaman index.php dimana tag id "content_hal_depan"-nya telah berisi forum.php dapat dilihat pada gambar 4.53.



Gambar 4.53 Tampilan Halaman Index.php Yang Telah Diisikan Forum.php ke dalam tag id "*content_hal_depan*"

Function tampil_halaman_ajax() selengkapnya dapat dilihat pada listing 4.2 berikut.

```
function tampil_halaman_ajax(url,containerid){
var xmlHttp=null;
try
 {
 // Firefox, Opera 8.0+, Safari
 xmlHttp=new XMLHttpRequest();
 }
catch (e)
 {
 //Internet Explorer
 try
  {
  xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
  }
 catch (e)
  {
  xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
 }
      xmlHttp.onreadystatechange=function(){

            loadpage(xmlHttp,containerid)
      }
      xmlHttp.open('GET',url,true)
      xmlHttp.send(null)
}
function loadpage(xmlHttp,containerid){
      if(xmlHttp.readyState==1)
      document.getElementById(containerid).innerHTML = '
      <div class=style3><center><br><br><br><br>
      <img src="images/loading.gif"><br>loading..<br><br><br></center>
      </div>';
      else if(xmlHttp.readyState==4)
      document.getElementById(containerid).innerHTML=xmlHttp.responseText
}
```

Listing 4.2 Script Function tampil_halaman_ajax()

Pada listing 4.2 diatas, *function* tampil_halaman_ajax() menggunakan obyek XMLHttpRequest untuk me-*request* URL yang dimaksud dan hasil dari *request* ini akan direspon oleh *function* loadpage() dengan mengisikan halaman yang di-*load* ke dalam "*containerid*", *container* id berisi *tag* id yang ditentukan.

4.3.2 Perbandingan Proses Validasi Masukan Data

Validasi masukan data dengan penerapan teknologi AJAX digunakan pada proses input data anggota. Pada proses validasi ini, setelah data dimasukan, function validasi() yang dipicu oleh event OnKeyUp pada setiap *textfield*, dengan menggunakan obyek XMLHttpRequest, akan memanggil modul PHP yang bertugas mengecek data masukan dengan data pada *database*. kemudian hasil dari modul PHP ini akan ditampilkan ke dalam tag id tertentu pada halaman yang ditentukan. Pada proses pengisian data, setiap kita mengetikan karakter per karakter masukan data, maka setiap itu pula lah EventOnKeyUp bekerja.

Event OnKeyUp yang memicu function validasi() pada halaman tambah anggota dapat dilihat pada listing 4.3 dibawah ini.

```
<input name="username" type="text" class="style3" id="username"
onkeyup="validasi()" size="30"/>
```

Listing 4.3 Tag Event onkeyup="validasi()" Pada Proses Tambah Anggota

Function validasi () yang dipicu oleh event OnKeyUp diatas, dapat dilihat pada listing 4.4 dibawah ini.

```
function validasi()
{
      xmlHttp=GetXmlHttpObject()
      if (xmlHttp==null)
            {
                  alert ("Browser tidak support HTTP Request")
                  return
            }
      var url="../admin/function.php"
      tgl= encodeURIComponent(document.getElementById("tgl").value);
      bln= encodeURIComponent(document.getElementById("bln").value);
      thn= encodeURIComponent(document.getElementById("thn").value);
      thn_ajaran=
encodeURIComponent(document.getElementById("thn_ajaran").value);
      username=
encodeURIComponent(document.getElementById("username").value);
      niy= encodeURIComponent(document.getElementById("niy").value);
      tmpt_lhr=
encodeURIComponent(document.getElementById("tmpt_lhr").value);
      password=
encodeURIComponent(document.getElementById("password").value);
      nm_lgkp=
encodeURIComponent(document.getElementById("nm_lgkp").value);
      gender=
encodeURIComponent(document.getElementById("gender").value);
      status=
encodeURIComponent(document.getElementById("status").value);
      almt= encodeURIComponent(document.getElementById("almt").value);
      url=url+"?tipe=1&password="+password+"&nm_lgkp="+nm_lgkp+"&gender=
"+gender+"&almt="+almt+"&tmpt_lhr="+tmpt_lhr+"&status="+status+"&niy="+n
iy+"&username="+username+"&thn_ajaran="+thn_ajaran
      xmlHttp.onreadystatechange=validasi_view
      xmlHttp.open("GET",url,true)
      xmlHttp.send(null)
}

function validasi_view()
{
      if (xmlHttp.readyState==1)
            {
                  document.getElementById("validasi").innerHTML = '<div
align=right><b>loading..</b></div>';
            }
      else if (xmlHttp.readyState==4 )
            {

      document.getElementById("validasi").innerHTML=xmlHttp.responseText
            }
}
```

Listing 4.4 Script Function validasi() Pada Proses Tambah Anggota

Function validasi() diatas akan menggunakan obyek XMLHttpRequest yang telah dibentuk sebelumnya untuk memanggil modul PHP function.php. Modul PHP ini berfungsi untuk mengecek jika masukan data kosong atau tipenya tidak benar. Dan mengecek data yang dimasukan dengan data yang telah terdapat pada *database*. Kemudian hasil dari modul PHP ini akan diisikan ke dalam tag id "validasi" oleh function validasi_view().

Modul PHP yang digunakan dalam proses tambah anggota, selengkapnya dapat dilihat pada listing 4.5 dibawah ini.

```
<?php
include "../koneksi.php";
if(($status == 1)){
        echo"Status Harus Diisi<br /><br />";
}if(empty($niy)){
        echo"No.Induk Yayasan Harus Diisi!<br /><br />";
}if(!empty($niy)){
                if(ereg("([0-9]+[0-9]+[0-9]+[0-9]+[0-9]+[0-9]+[0-9])", $niy)){
                        $sql="SELECT niy FROM user WHERE niy = '$niy'";
                        $result = mysql_query($sql);
                        if($row = mysql_fetch_array($result)) {
                                echo "No.Induk Yayasan Telah Ada,
                                        Gunakan Yang Lain!<br /><br />";
                                }
                }else{
                        echo"No.Induk Yayasan Harus Diisi 7 Angka!<br /><br />";
                        }
}if(empty($username))  {
        echo"Username Harus Diisi!<br /><br />";
}if(!empty($username)) {
                        $sql="SELECT username FROM user WHERE username = '$username'";
                        $result = mysql_query($sql);
                        if($row = mysql_fetch_array($result)) {
                                echo "Username Telah Ada, Gunakan Yang Lain!<br /><br />";
                }
}if(empty($password))  {
        echo"Password Harus Diisi!<br /><br />";
}if(!empty($password)) {
                if ((!ereg ("[0-9]+[a-z]", $password))
                        and (!ereg ("[a-z]+[0-9]", $password))){
                        echo"Password Harus Terdiri Dari Kombinasi Angka Dan Huruf
                        <br /><br />";
                        }
}if(empty($nm_lgkp)){
        echo"Nama Lengkap Harus Diisi!<br /><br />";
}if($thn_ajaran==1){
        echo"Tahun Ajaran Harus Diisi!<br /><br />";
}if(empty($tmpt_lhr))  {
        echo"Tempat Lahir Harus Diisi!<br /><br />";
}if (($tgl==x) or ($bln==x) or ($thn==x))    {
        echo"Tanggal Lahir Harus Diisi!<br /><br />";
}if ($gender == 1){
        echo "Jenis Kelamin Harus Diisi!<br /><br />";
}if (empty($almt)){
        echo"Alamat Harus Diisi!<br /><br />";
}
//submit
if (!($status == 1)){
if (!empty($niy)){
                if(ereg("([0-9]+[0-9]+[0-9]+[0-9]+[0-9]+[0-9]+[0-9])", $niy)){
                        include "../koneksi.php";
                        $sql="SELECT niy FROM user WHERE niy = '$niy'";
                        $result = mysql_query($sql);
                        if($row = mysql_fetch_array($result)) {
                }else{
```

(bersambung ke halaman selanjutnya)

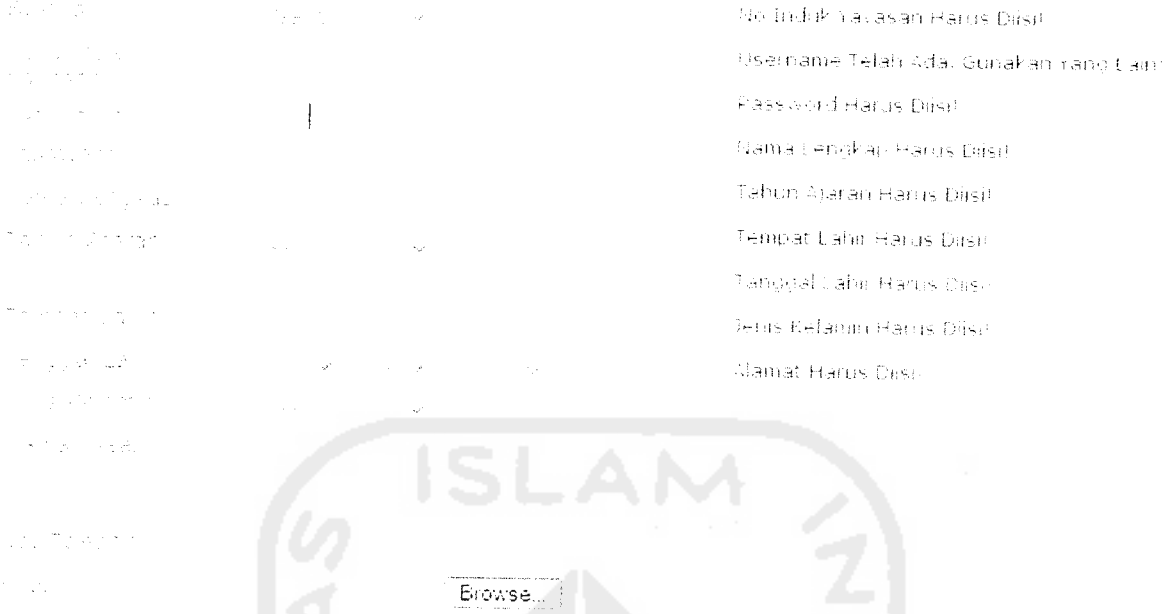
```
        if (!empty($username)) {
                $sql2="SELECT username FROM user WHERE username = '$username'";
                $result2 = mysql_query($sql2);
                        if($row2 = mysql_fetch_array($result2)){
                        }else{
                        if (!empty($password)) {
                        if (!empty($nm_lgkp)) {
                        if (!($thn_ajaran=-1)) {
                        if (!empty($tmpt_lhr)) {
                        if (($tgl==x) or ($bln==x) or ($thn--x)){
                        }else{
                        if ($gender == 1){
                        }else{
                        if (empty($almt)){
                        }else{
                                echo"<div align=left >
                                <br /><br /><br /><br /><b>Simpan Anggota >>
                                <input name=\"button\" type=\"submit\"  value=\"Save\">
                                </div>";
}}}}}}}}}}}}}

?>
```

Listing 4.5 Modul PHP Validasi Tambah Anggota

Modul PHP pada listing 4.5 diatas berfungsi untuk melakukan validasi masukan data yang diisikan, hasil dari modul PHP ini akan menampilkan masukan data mana saja yang isinya kosong atau tidak sesuai, validasi akan ditampilkan per setiap masukan data yang diisikan, tidak menunggu sampai dengan semua masukan data diisikan.

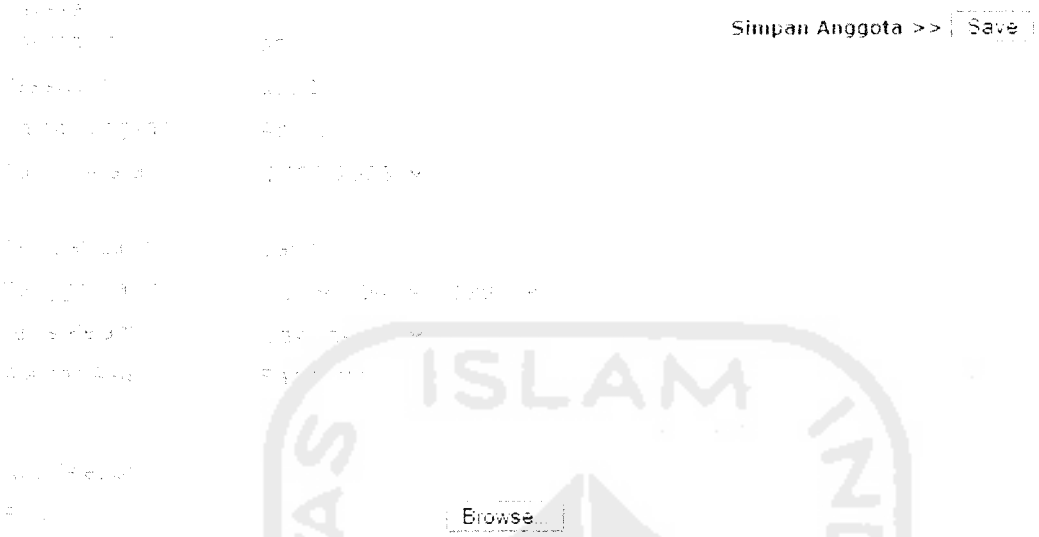
Kita akan mencoba proses tambah anggota dengan mengisikan masukan data status="santri" dan username="roni", tampilannya dapat dilihat pada gambar 4.54 berikut.



Gambar 4.54 Tampilan Masukan Data Tambah Anggota *Username* Telah Ada.

Pada gambar 4.54 diatas dapat kita lihat tampilan validasi bahwa *username* "roni" telah digunakan, dan masukan data no.induk yayasan, *password*, nama lengkap, tahun ajaran, tempat lahir, tanggal lahir, jenis kelamin dan alamat harus diisi (kosong).

Tampilan validasi tambah anggota dengan masukan data sesuai dan telah diisi semua, dapat dilihat pada gambar 4.55 berikut.



Gambar 4.55 Tampilan Masukan Tambah Anggota Sesuai

Pada gambar 4.55 diatas, dapat dilihat apabila semua masukan data diisi dan sesuai, maka akan ditampilkan tombol untuk memproses tambah anggota.

4.3.3 Perbandingan Proses *Login*

Proses *login* dengan penerapan teknologi AJAX akan memudahkan pengguna, karena kesesuaian *username* dan *password* yang dimasukan akan langsung ditampilkan tanpa harus menekan tombol lebih dahulu.

Untuk masuk ke halaman anggota dan halaman admin, diperlukan l*ogin* terlebih dahulu. Pada proses ini, AJAX digunakan untuk validasi form masukan *username* dan

password. Apabila semua input data sesuai, AJAX akan memanggil modul PHP yang berfungsi untuk melakukan pengecekkan terhadap data di dalam *database*.

Pada proses *login* ini, Ketika pengguna baik admin ataupun anggota mengisi masukan data, *event* OnKeyUp pada akan memicu function_password_cek(). Hal ini dapat kita lihat pada listing 4.6 dibawah ini.

```
<input name="password" type="password" class="berita_isi" id="password"
size="14" onkeyup="password_cek()"/>
```

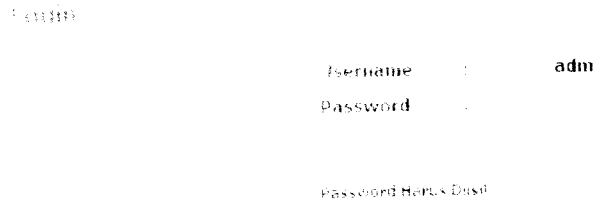
Listing 4.6 Tag OnKeyUp password_cek()

Function password_cek() pada listing 4.6 diatas menggunakan obyek XMLHttpRequest yang telah dibentuk sebelumnya untuk memanggil modul PHP yang berfungsi mengecek masukan data *username* dan *password* dengan data *username* dan *password* pada *database*. Hasil dari pengecekkan *database* ini akan di respon oleh obyek XMLHttpRequest pada *function* password_v() yang kemudian akan ditampilkan ke dalam *tag* id "*content2*", seperti yang terlihat pada listing 4.7 dibawah ini.

```
<?php
session_start();
include "../koneksi.php";
if($tipe==2){
if(empty($username) and !empty($password))    {
                echo "<font style=color:#FF0000; ><b>Username Harus Diisi!
                        </font></div><br />";
}if(!empty($username) and empty($password))   {
                echo "<font style=color:#FF0000; ><b>Password Harus Diisi!
                        </font></div><br />";
}if(empty($username) and empty($password))    {
                        echo "<font style=color:#FF0000; >
                        <b>Username Dan Password Harus Diisi!</font></div><br />";
}if(!empty($username) and !empty($password)) {
$sql="  SELECT *
        FROM user
        WHERE password = '$_GET[password]'
        AND username = '$_GET[username]'";
$result = mysql_query($sql);
$row = mysql_fetch_array($result);
if ($row[username]==$_GET[username] and $row[password]==$_GET[password]){
        if ($row[status]=='adm'){
                session_register("admin","pass_admin");
                $admin=$row[username];
                $pass_admin=$row[password];
                echo "<font style=color:#009900; >    <b>Username Dan Password Cocok..
                </b></font><div align=right><input type=\"submit\"
                onclick=\"javascript:location.href='admin/admin.php';\"
                value=\"Masuk &gt;&gt;\"/></div>";
        }else{
                session_register("anggota","pass_anggota");
                $anggota=$row[niy];
                $pass_anggota=$row[password];
                echo "<font style=color:#009900; ><b>Username Dan Password Cocok..
                </b></font><div align=right>  <input name=\"Button\"  type=\"button\"
                onclick=\"javascript:location.href='anggota/halaman_anggota.php
                ?n=$row[niy]&login=2';\" value=\"Masuk &gt;&gt;\"/></div>";
                }
        }else{
                echo "<font style=color:#FF0000; ><b>Username Dan Password Tidak Cocok!
                </b></font>";
        }
        }
}
?>
```

Listing 4.8 Modul PHP *Login*

Modul PHP pada listing 4.8 diatas berfungsi untuk memproses proses *login* admin maupun anggota. Kita akan mencoba melakukan *login* dengan masukan data *username* ="adm" dan *password* dikosongkan. Tampilannya seperti pada gambar 4.56 berikut.



Gambar 4.56 Tampilan *Login Password* Harus Diisi

Pada gambar 4.56 diatas dapat kita lihat bahwa apabila *password* dikosongkan, maka akan ditampilkan validasi " *Password* Harus Diisi!".

Tampilan *login* dengan masukan data salah, dimana menggunakan *username*="adm" dan *password*="admin", dapat kita lihat pada gambar 4.57 berikut.



Gambar 4.57 Tampilan *Login* Masukan Data Salah

Pada gambar 4.57 diatas dapat kita lihat bahwa jika masukan data yang diisikan pada proses *login* salah, maka akan ditampilkan validasi "*Username* dan *Password* Tidak Cocok!".

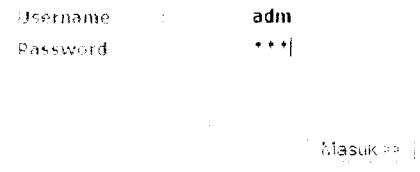
Tampilan *login* dengan masukan data benar, dimana menggunakan *username*="adm" dan *password*="adm" dapat dilihat pada gambar 4.58 berikut.

```
{
xmlHttp=GetXmlHttpObject()
if (xmlHttp==null)
 {
 alert ("Browser tidak support HTTP Request")
 return
 }
var url="login/cek_login.php"

username= encodeURIComponent(document.getElementById("username").value);
password= encodeURIComponent(document.getElementById("password").value);
url=url+"?tipe=2&username="+username+"&password="+password
xmlHttp.onreadystatechange=password_v
xmlHttp.open("GET",url,true)
xmlHttp.send(null)
}
function password_v()
{
if (xmlHttp.readyState==1)
        {
        document.getElementById("content2").innerHTML = '
        <div align=right><b>loading..</b></div>';
        }
else if (xmlHttp.readyState==4 )
 {
 document.getElementById("content2").innerHTML=xmlHttp.responseText
 }
}
```

Listing 4.7 Function password_cek() dan user_cek()

Modul PHP yang digunakan pada proses *login* dapat dilihat pada listing 4.8 berikut.



Gambar 4.58 Tampilan Validasi *Login* Masukan *Username* dan *Password* cocok

Pada gambar 4.58 diatas dapat kita lihat bahwa apabila *username* dan *password* yang diisikan benar, maka akan ditampilkan tombol untuk masuk ke dalam halaman admin ataupun halaman anggota.

4.3.4 Perbandingan Proses Pencarian

Dengan penerapan teknologi AJAX, pengguna akan dipermudah dalam melakukan proses pencarian, dengan adanya event handler onKeyUp yang memicu obyek XMLHttpRequest yang mampu berinteraksi dengan *server* secara *background*, pengguna akan dipermudah melakukan pencarian harus tanpa meng-klik tombol "*submit*" seperti pada proses pencarian tradisional.

Proses pencarian yang terdapat pada *website* ini terdiri atas tiga macam, yaitu proses pencarian berita pengunjung, proses pencarian anggota, dan proses pencarian admin. Ketiga proses pencarian ini pada dasarnya sama, hanya saja modul PHP yang dipanggilnya berbeda. Pada proses pencarian ini ketika kata kunci dimasukan, *event* OnKeyUp pada *textfield* kata kunci memicu obyek XMLHttpRequest yang terdapat pada *function* cari() untuk memanggil modul PHP.

Tag yang memicu *function* cari_anggota () dapat dilihat pada listing 4.9 berikut.

```
<input    name="kt_kunci"    type="text"    class="style5"    id="kt_kunci"
onkeyup="cari_anggota()" />
```

Listing 4.9 Tag OnKeyUp cari_anggota()

Function cari_anggota() yang digunakan pada pencarian anggota, yang dipicu oleh event OnKeyUp pada listing 4.9 diatas dapat dilihat pada listing 4.11 dibawah ini.

```
function cari_anggota()
{
xmlHttp=GetXmlHttpObject()
if (xmlHttp==null)
 {
 alert ("Browser tidak support HTTP Request")
 return
 }
var url="../cari_function.php"
kt_kunci= encodeURIComponent(document.getElementById("kt_kunci").value);
url=url+"?tipe=3&kt_kunci="+kt_kunci
xmlHttp.onreadystatechange=stateChanged
xmlHttp.open("GET",url,true)
xmlHttp.send(null)
}

function stateChanged()
{
if (xmlHttp.readyState==1)
      {
      document.getElementById("hasil_cari").innerHTML                    =
'<center><br><br><br><br><img
src="../images/loading.gif"><br>loading..<br><br><br></center>';
      }
else if (xmlHttp.readyState==4 )
 {
 document.getElementById("hasil_cari").innerHTML=xmlHttp.responseText
 }
}
```

Listing 4.10 Function cari_anggota()

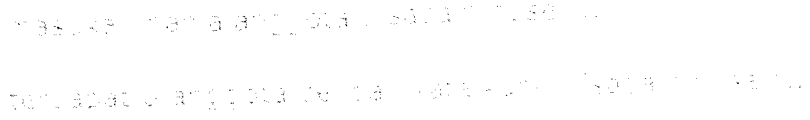
Pada listing 4.10 diatas dapat dilihat bahwa *function* cari_anggota() menggunakan obyek XMLHttpRequest untuk memanggil cari_function.php, kemudian hasil dari modul PHP tersebut akan direspon oleh obyek XMLHttpRequest pada *function* stateChanged() dan

ditampilkan pada tag id "hasil_cari". Modul PHP yang digunakan pada proses ini dapat dilihat pada listing 4.11.

```
<?
if (!empty($kt_kunci))
      {
                        include_once "koneksi.php";
      $query_cari="      SELECT *
                              FROM user, data_anggota
                              WHERE user.status !='adm'
                              AND user.niy=data_anggota.niy
                              AND       data_anggota.nm_lgkp       like
'%$kt_kunci%' order by data_anggota.niy";
    $hasil_cari=mysql_query($query_cari);
      $no=1;
      $jumlah = mysql_num_rows($hasil_cari);
      echo   "   <br>terdapat   $jumlah   anggota   dengan   kata   kunci:
\"$kt_kunci\"<br>" ;
      echo "<br><table border='0'>";
      if (!empty($jumlah))
      {
      ?>
<?
      while ($row=mysql_fetch_array($hasil_cari))
                  {
echo "<tr>";?>
<? echo"<td align=\"left\" valign=\"top\">";?><?=$no++;?><? echo". ";?>
<?    echo"<td    align=\"left\"    valign=\"top\">";?><?    echo"<img
src=\"../foto/$row[foto]\"   width=\"50\"   height=\"60\"   border=\"2\"
class=\"style5\" />";?><? echo"";?>
<? echo"<td align=\"left\" valign=\"top\"> No. Induk <br><br> Nama <br>
Status <br></td>";?>
<?
 echo "<td align=\"left\" valign=\"top\">: <br><br>:<br>:<br></td>";
 echo  "<td  align=\"left\"  valign=\"top\">$row[niy]<br><br>$row[nm_lgkp]
<br>   ";?><?   if   ($row[status]=="san")   {   echo"Santri";}   else
if($row[status]=="mts")  {echo"Siswa  MTs";}else  if($row[status]=="ma")
{echo"Siswa MA";}?><? echo"</td>";
 echo "</tr>";
                  }//akhir tabel pencarian
echo "</table><br>";
      }
      }
      else
      {
      echo "<br>kata kunci harus diisi";
      }
?>
```

Listing 4.11 Modul PHP Pencarian Anggota

Kita akan coba mengisikan kata kunci="sadam husain", dimana anggota dengan kata kunci "sadam husain" tidak terdapat pada *database*, maka tampilannya adalah seperti pada gambar 4.59 berikut ini.



Gambar 4.59 TampilanPencarian Data Tidak Ditemukan

Pada gambar 4.59 diatas, dapat kita lihat bahwa apabila data tidak ditemukan, maka akan ditampilkan "terdapat 0 anggota dengan kata kunci: "sadam husain.."".

Ketika kita memasukan data pencarian yang terdapat pada *database*, dimana menggunakan kata kunci= "r", maka tampilannya seperti pada gambar 4.45 berikut ini.



Nama : Ratu Farha Arrifa



Nama : Gufroni

Gambar 4.60 Tampilan Pencarian Data Ditemukan Dengan Kata Kunci "r"

Pada gambar 4.60 diatas, dapat kita lihat bahwa anggota dengan nama yang mengandung huruf "r" terdapat 2 anggota.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan maka penulis dapat mengambil beberapa kesimpulan, yaitu :

1. Penerapan teknologi AJAX pada *website* Yayasan Manarul Huda ini digunakan pada 4 proses, yaitu: proses *load* halaman, proses *login,* proses validasi masukan data, dan proses pencarian.
2. Dengan penerapan teknologi AJAX, *website* Yayasan Manarul Huda ini memiliki keunggulan di dalam proses *login*, proses validasi masukan data, dan proses pencarian, dimana setelah kita mengisikan masukan data, maka hasil dari proses-proses tersebut akan langsung ditampilkan pada halaman *website* tanpa harus menekan tombol terlebih dahulu.

5.2 Saran

Secara umum program yang dibuat masih belum sempurna, karena itu beberapa saran yang ingin disampaikan, yaitu :

1. Diharapkan pada pengembangan sistem selanjutnya, penerapan teknologi AJAX tidak hanya diimplementasikan pada beberapa bagian *website*, tetapi juga diimplementasikan pada seluruh bagian pada *website* yang dibuat.
2. Diharapkan untuk lebih meningkatkan pemahaman tentang Javascript, karena Javascript merupakan bahasa pemrograman utama dalam implementasi penerapan teknologi AJAX.

DAFTAR PUSTAKA

[ANO01] Anonymous, 2001. *Ajax The XMLHttpRequest Object* http://www.w3schools.com/ajax/ajax_xmlhttp_request.asp. Diakses tanggal 26 Juli 2007.

[ANO05] Anonymous, 2005. *AJAX (Programming)* http://en.wikipedia.org/wiki/ajax_(programming) . Diakses pada tanggal 28 Juni 2007

[ANO07] Anonymous, 2007. *XMLHttpRequest Object* http://www.depelover.com/tech/article_php/875008.html. Diakses pada tanggal 11 September 2007.

[ANO99] Anonymous, 1999. *Web In The Beginning The History* http://en.wikipedia.org/wiki/web/767_a.html. Diakses pada tanggal 12 Agustus 2007.

[BAL06] Ballard, P. 2006. *Teach Yourself AJAX in 10 Minutes*. Sams

[GAR05] Garret, J.J. 2005. *Ajax : A New Approach to Web Applications*, California. http://adaptivepath.com/publications/essay/archives/000385.php. Diakses pada tanggal 18 Juli 2007

[JON06] Jones, b. l. 2006 *Client-side Versus Server-side Coding – Part 1* http://www.depelover.com/tech/article_php/923111.html. Diakses Tanggal 20 Juli 2007.

[MUR05] Murray, G. 2005 *Asynchronous Javascript Technology and XML (AJAX) With Java 2 Platform, Enterprise Edition, Sun Java's Article.* http://java.sun.com/developer/technicalArtilces/J2EE/AJAX/ diakses pada tanggal 24 Juli 2007.

[OET03] Oetomo, B. SD. 2002. *Perencanaan dan Pembangunan Sistem Informasi*, ANDI. Yogyakarta

[W3C04] XML Core Working Group, 2004. *Extensible Markup Language (XML) 1.0 (Third edition) – W3C Recommendation* http://www.w3.org/TR_2004_REC-xml-20040204. Diakses pada tanggal 29 Juli 2007.