

**SISTEM INFORMASI PENERIMAAN MAHASISWA BARU
BERBASIS WEB DAN WAP
(STUDI KASUS DI STMIK AMIKOM YOGYAKARTA)**



SKRIPSI

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri Sunan Kalijaga
Untuk Memenuhi Sebagian Syarat Memperoleh Gelar Sarjana
Strata Satu Teknik Informatika

Disusun oleh:

Sidiq Wahyu Surya Wijaya
NIM. 06650036

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA
2010**



Universitas Islam Negeri Sunan Kalijaga

FM-UINSK-BM-05-07/R0

PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/1684/2010

Skripsi/Tugas Akhir dengan judul : Sistem Informasi Penerimaan Mahasiswa Baru
Berbasis WEB dan WAP (Studi Kasus di STMIK AMIKOM
Yogyakarta)

Yang dipersiapkan dan disusun oleh

Nama : Sidiq Wahyu Surya Wijaya

NIM : 06650036

Telah dimunaqasyahkan pada : 30 Agustus 2010

Nilai Munaqasyah : A-

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Agus Mulyanto, S.Si., M.Kom
NIP.19710823 199903 1 003

Penguji I

Shofwatul 'Uyun, S.T., M.Kom
NIP.19820511 200604 2 002

Penguji II

Agung Fatwanto, S.Si., M.Kom
NIP. 19770103 200501 1 003

SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 3 September 2010

UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan

Dra. Maizer Said Nahdi, M.Si
NIP. 19550427 198403 2 001



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir

Lamp :

Kepada

Yth. Dekan Fakultas Sains & Teknologi

UIN Sunan Kalijaga Yogyakarta

DI Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan **seperlunya**, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Sidiq Wahyu Surya Wijaya
NIM : 06650036
Judul Skripsi : Perancangan Sistem Informasi Penerimaan Mahasiswa Baru Berbasis
WEB dan WAP (Studi Kasus di STMIK AMIKOM Yogyakarta)

sudah dapat diajukan kembali kepada Fakultas Sains & Teknologi Jurusan/ Program Studi Teknik **Informatika** UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana **Strata Satu** .

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera **dimunaqsyahkan**. Atas perhatiannya kami ucapkan terima kasih.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 2 Agustus 2010
Pembimbing I

Agus Mulyanto, M.Kom
NIP: 19710823 199903 1 003



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir
Lamp :

Kepada
Yth. Dekan Fakultas Sains & Teknologi
UIN Sunan Kalijaga Yogyakarta
Di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Sidiq Wahyu Surya Wijaya
NIM : 06650036
Judul Skripsi : Perancangan Sistem Informasi Penerimaan Mahasiswa Baru Berbasis
WEB dan WAP (Studi Kasus di STMIK AMIKOM Yogyakarta)

sudah dapat diajukan kembali kepada Fakultas Sains & Teknologi Jurusan/ Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu .

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Yogyakarta, 2 Agustus 2010

Pembimbing II


M. Mustakim, M.T

NIP. 19790331 200501 1 004

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Sidiq Wahyu Surya Wijaya
NIM : 06650036
Program Studi : Teknik Informatika
Fakultas : Sains Dan Teknologi

Menyatakan bahwa skripsi dengan judul “**Sistem Informasi Penerimaan Mahasiswa Baru Berbasis WEB dan WAP (Studi Kasus di STMIK AMIKOM Yogyakarta)**” tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 02 Agustus 2010

Mahasiswa



Sidiq Wahyu Surya Wijaya

NIM. 06650036

KATA PENGANTAR

Alhamdulillahirabbi'alamin. Puji syukur bagi Allah SWT yang telah melimpahkan rahmat-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi dengan judul Sistem Informasi Penerimaan Mahasiswa Baru Berbasis WEB dan WAP (Studi Kasus di STMIK AMIKOM Yogyakarta). Sebuah sistem yang mampu mengintegrasikan antara sains dan teknologi informasi serta memanfaatkan perkembangan teknologi informasi, khususnya *handphone*, sebagai media pendaftaran mahasiswa baru alternatif.

Selanjutnya penulis mengucapkan terima kasih kepada:

1. Ibu Maizer Said Nahdi, M.Si, selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga;
2. Bapak Agus Mulyanto, M.Kom, selaku Ketua Program Studi dan Dosen Pembimbing I yang dengan kesabarannya telah membimbing selama penyusunan skripsi ini;
3. Bapak Mustakim, M.T, selaku Dosen Pembimbing II yang dengan sangat teliti memberikan koreksi terhadap penulisan skripsi ini;
4. Para Dosen Program Studi Teknik Informatika yang telah memberi bekal ilmu pengetahuan kepada penulis, semoga ilmunya menjadi amal *jariyah* di dunia hingga akhirat;

5. Kedua orang tuaku yang selalu mendo'akanku dan selalu menasehatiku untuk senantiasa menjadi anak yang berbakti dan berguna bagi orang lain dan senantiasa bersyukur atas semua nikmat yang diberikan Allah SWT.
6. Calon isteriku yang telah memberikan dukungan dan motivasi dalam segala hal.
7. Alex dan Imam yang telah membantu menyelesaikan kesulitan-kesulitan dalam mempelajari PHP.
8. Serta teman-teman program studi teknik informatika, khususnya angkatan 2006 yang telah banyak memberi dukungan serta meminjamkan fasilitas dalam pengembangan sistem dan penyusunan skripsi ini.

Penulis merasa masih banyak sekali kekurangan dan kelemahan dalam penelitian ini, oleh karena itu segala kritik dan saran senantiasa penulis harapkan dari para pembaca. Akhir kata, semoga penelitian ini dapat menjadi panduan serta referensi yang sangat berguna bagi pembaca dan dapat dimanfaatkan sebaik-baiknya.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 2 Agustus 2010

Penulis

HALAMAN PERSEMBAHAN

Kupersembahkan untuk:

- ❖ *Sembah sujudku kepada Allah Subhanahu wa Ta'ala yang senantiasa melimpahkan rahmat dan hidayah-Nya.*
- ❖ *Sholawat dan Salam kepada junjungan nabi besar Muhammad Shollallahu 'alaihi wa Sallam.*
- ❖ *Ayah dan Ibuku tercinta yang tak pernah henti-hentinya berjuang demi aku.*
- ❖ *Kakak-adikku tersayang yang selalu memberikan dorongan semangat dan inspirasi.*
- ❖ *Teman-temanku yang selalu memberikan do'a dan dukungan : Imam, Alex.*
- ❖ *Seluruh teman-teman belajar dan maen futsal : Rifqi, Sigit, Nurdin, Irvan, Fathan, Dika, Qori, Sunu, Fikri, Aslam, Uxad.*
- ❖ *Seluruh teman-teman Teknik Informatika : Doyok, Mas Anthok, Agonk, Azizi, Ali, Wahid, Feby, Ayu, Tamy, Didik, Arfan, Ayun, Ryan, dan yang tidak bisa kusebutkan satu persatu.*
- ❖ *Terima kasih buat temen-teman seperjuanganku atas segala masukan dan dukungannya.*

HALAMAN MOTTO

Hidup ini penuh perjuangan, tidak ada kata "tidak bisa";

Ikhtiar, berdo'a, kemudian bertawakkal, itulah kunci kesuksesan;

Ingat 3 hal yang tidak akan berhenti aliran pahalanya sampai kapan pun, salah satunya adalah memberikan ilmu yang bermanfaat bagi orang lain.



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR ISI

Halaman Judul.....	i
Halaman Pengesahan	ii
Surat Persetujuan Skripsi/Tugas Akhir	iii
Pernyataan Keaslian Skripsi.....	v
Kata Pengantar	vi
Halaman Persembahan.....	viii
Motto.....	ix
Daftar Isi	x
Daftar Tabel	xvi
Daftar Gambar.....	xviii
Daftar Lampiran.....	xxi
Intisari	xxii
Abstract.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5

1.6 Keaslian Penelitian.....	6
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	7
2.1 Tinjauan Pustaka.....	7
2.2 Landasan Teori.....	8
2.2.1 Konsep Sistem Informasi.....	8
2.2.1.1 Sistem.....	8
2.2.1.2 Informasi.....	9
2.2.1.3 Sistem Informasi.....	10
2.2.2 Web.....	10
2.2.2.1 Konsep Aplikasi Web <i>Database</i>	11
2.2.2.2 Konsep RDBMS.....	14
2.2.3 <i>Wireless Application Protocol (WAP)</i>	14
2.2.3.1 Konsep Aplikasi Berbasis WAP.....	16
2.2.4 Web dan WAP <i>Scripting</i> dengan PHP dan WML.....	18
2.2.4.1 PHP.....	19
2.2.4.2 WML.....	20
2.2.4.3 <i>Database Server</i> dengan MySQL.....	21
2.2.5 Pemodelan Data.....	22
2.2.5.1 Diagram Alir Data.....	22
2.2.5.2 <i>Entity Relationship Diagram</i>	24
2.2.5.3 Normalisasi.....	26

BAB III METODE PENELITIAN	28
3.1 Metode Penelitian	27
3.2 Perangkat Keras (<i>Hardware</i>) yang Dibutuhkan.....	30
3.3 Perangkat Lunak (<i>Software</i>) yang Dibutuhkan	31
BAB IV ANALISA DAN PERANCANGAN	32
4.1 Analisis.....	32
4.1.1 Analisa Alur Kerja Pendaftaran Mahasiswa Baru	32
4.1.2 Analisa Alur Kerja Sistem yang Dibangun.....	34
4.2 Desain.....	36
4.2.1 Arsitektur Sistem.....	36
4.2.2 Desain DFD (<i>Data Flow Diagram</i>)	38
4.2.2.1 DFD Level 0 (<i>Diagram Konteks</i>).....	38
4.2.2.2 DFD Level 1.....	39
4.2.2.3 DFD Level 2 Proses 1 (Pembuatan Master Data)	40
4.2.2.4 DFD Level 2 Proses 2 (Pendaftaran)	42
4.2.2.5 DFD Level 2 Proses 3 (Seleksi Tes)	44
4.2.2.6 DFD Level 3 Proses Konfirmasi Pembayaran	45
4.2.2.7 DFD Level 3 Proses Data Pendaftar	46
4.2.3 Desain ERD (<i>Entity Relationship Diagram</i>).....	48
4.2.4 Desain Tabel.....	48
4.2.4.1 Analisa Desain Tabel	62

4.2.4.2 Relasi Antar Tabel.....	64
4.2.5 Desain Layout dan Navigasi	66
4.2.5.1 Desain Layout pada Modul Administrator.....	66
4.2.5.2 Desain Layout pada Modul User Berbasis Web	67
4.2.5.3 Desain Layout pada Modul User Berbasis WAP.....	67
4.2.6 Desain Halaman Utama.....	68
4.2.6.1 Desain Halaman Utama pada Modul Administrator...	68
4.2.6.2 Desain Halaman Utama Web dan WAP untuk User...	69
4.2.7 Desain Antar Muka	73
4.2.7.1 Desain Masukan Login	73
4.2.7.2 Desain Masukan Admin.....	73
4.2.7.3 Desain Masukan Pendaftaran.....	74
4.2.7.4 Desain Masukan Konfirmasi.....	75
BAB V HASIL DAN PEMBAHASAN.....	77
5.1 Pengkodean	77
5.1.1 Implementasi Sistem	77
5.1.2 Implementasi Basis Data.....	78
5.1.3 Implementasi Koneksi MySQL dengan PHP.....	82
5.1.4 Implementasi WinWAP <i>Browser</i>	84
5.1.5 Implementasi Konfigurasi File WML.....	85

5.1.6 Implementasi Modul Administrador	85
5.1.6.1 Implementasi Halaman Utama Modul Administrator.	87
5.1.6.2 Implementasi Halaman Manajemen Admin.....	90
5.1.6.3 Implementasi Proses Pendaftaran	94
5.1.6.4 Implementasi Proses Seleksi Tes	103
5.1.7 Implementasi Modul User Berbasis Web	108
5.1.7.1 Implementasi Halaman Utama.....	109
5.1.7.2 Implementasi Proses Pendaftaran	111
5.1.7.3 Implementasi Proses Seleksi Tes	116
5.1.7.4 Implementasi Proses Konfirmasi Pembayaran.....	119
5.1.8 Implementasi Modul User Berbasis WAP	122
5.1.8.1 Implementasi Halaman Utama.....	123
5.1.8.2 Implementasi Proses Pendaftaran	125
5.1.8.3 Implementasi Proses Seleksi Tes	131
5.1.8.4 Implementasi Proses Konfirmasi Pembayaran.....	133
5.2 Pengujian.....	135
5.2.1 Metode <i>Black Box Testing</i>	136
5.2.2 Metode <i>Alfa Testing</i>	138
BAB VI PENUTUP.....	141
6.1 Kesimpulan	141
6.2 Saran.....	141

DAFTAR PUSTAKA	143
LAMPIRAN.....	145



DAFTAR TABEL

Tabel 4.1 Desain Tabel Admin	49
Tabel 4.2 Desain Tabel Pendaftaran	50
Tabel 4.3 Desain Tabel Jenis	51
Tabel 4.4 Pembuatan Master Data di Tabel Jenis	51
Tabel 4.5 Desain Tabel Agama	52
Tabel 4.6 Pembuatan Master Data di Tabel Agama	52
Tabel 4.7 Desain Tabel Jurusan Smu	52
Tabel 4.8 Pembuatan Master Data di Tabel Jurusan Smu	53
Tabel 4.9 Desain Tabel Kabupaten	53
Tabel 4.10 Desain Tabel Propinsi	54
Tabel 4.11 Desain Tabel Warga	54
Tabel 4.12 Pembuatan Master Data di Tabel Warga	54
Tabel 4.13 Desain Tabel Jurusan	55
Tabel 4.14 Pembuatan Master Data di Tabel Jurusan	55
Tabel 4.15 Desain Tabel Pilihan jurusan	56
Tabel 4.16 Desain Tabel Info	56
Tabel 4.17 Pembuatan Master Data di Tabel Info	56
Tabel 4.18 Desain Tabel Tempat	57
Tabel 4.19 Pembuatan Master Data di Tabel Tempat	57
Tabel 4.20 Desain Tabel Syarat	58

Tabel 4.21 Pembuatan Master Data di Tabel Syarat.....	58
Tabel 4.22 Desain Tabel Gelombang.....	58
Tabel 4.23 Pembuatan Master Data di Tabel Gelombang	59
Tabel 4.24 Desain Tabel Beasiswa	59
Tabel 4.25 Pembuatan Master Data di Tabel Beasiswa.....	60
Tabel 4.26 Desain Tabel Status.....	60
Tabel 4.27 Pembuatan Master Data di Tabel Status	61
Tabel 4.28 Desain Tabel Status_admin.....	61
Tabel 4.29 Pembuatan Master Data di Tabel Status_admin	61
Tabel 4.30 Desain Tabel pmb_config	62
Tabel 4.31 Desain Tabel Pendaftaran Terbaru.....	63
Tabel 4.32 Contoh Bentuk tidak Normal	64
Tabel 5.1 Skenario Pengujian	137
Tabel 5.2 Daftar Penguji pada Pengujian <i>Alfa Test</i>	138
Tabel 5.3 Hasil Pengujian Fungsional Sistem	139
Tabel 5.4 Hasil Pengujian Interface dan Pengaksesan.....	139

DAFTAR GAMBAR

Gambar 2.1 Model Komunikasi antara Web <i>Browser</i> dan Web <i>Server</i>	12
Gambar 2.2 WinWAP <i>Browser</i>	17
Gambar 2.3 Hirarki dalam WML.....	21
Gambar 2.4 Simbol Entity Luar pada Teknik Yourdon and De Macro	23
Gambar 2.5 Simbol Alir Data dalam Teknik Yourdon and De Macro	23
Gambar 2.6 Simbol Proses pada Teknik Yourdon and De Macro.....	24
Gambar 2.7 Simbol Berkas di dalam Teknik Yourdon and De Macro.....	24
Gambar 3.1 SDLC <i>waterfall</i> menurut Roger Pressman.....	29
Gambar 4.1 Arsitektur Sistem Informasi PMB.....	37
Gambar 4.2 DFD Level 0 (<i>Diagram Konteks</i>).....	39
Gambar 4.3 DFD Level 1.....	40
Gambar 4.4 DFD Level 2 Proses 1 (Pembuatan Master Data)	41
Gambar 4.5 DFD Level 2 Proses 2 (Pendaftaran).....	43
Gambar 4.6 DFD Level 2 Proses 3 (Seleksi Tes)	44
Gambar 4.7 DFD Level 3 Proses Konfirmasi Pembayaran	45
Gambar 4.8 DFD Level 3 Proses Data Pendaftar	47
Gambar 4.9 ERD Sistem Informasi PMB	48
Gambar 4.10 Relasi Antar Tabel.....	65
Gambar 4.11 Desain Layout dan Navigasi Modul Administrator	66
Gambar 4.12 Desain Layout dan Navigasi Modul User Berbasis Web.....	67

Gambar 4.13 Desain Layout dan Navigasi Modul User Berbasis WAP	68
Gambar 4.14 Struktur Halaman Admin.	69
Gambar 4.15 Struktur Halaman User.....	72
Gambar 4.16 Desain Masukan Login.....	73
Gambar 4.17 Desain Masukan Admin.....	74
Gambar 4.18 Desain Masukan Pendaftaran.....	75
Gambar 4.19 Desain Masukan Konfirmasi.....	76
Gambar 5.1 Apache dan MySQL dalam keadaan aktif.....	78
Gambar 5.2 Aplikasi WinWAP <i>Browser</i>	84
Gambar 5.3 Susunan File PHP pada Modul Administrator	86
Gambar 5.4 Tampilan Form Login untuk Admin	87
Gambar 5.5 Pesan Javascript ketika Username dan Password Salah.....	88
Gambar 5.6 Pesan Javascript ketika Inputan Captcha Security Salah	88
Gambar 5.7 Tampilan Halaman main.php setelah Admin Berhasil Login	89
Gambar 5.8 Tampilan Halaman list.php	91
Gambar 5.9 Form untuk Mengubah Data Admin	93
Gambar 5.10 Tampilan Halaman Data Pendaftar untuk Admin	96
Gambar 5.11 Tampilan Halaman Tambah Pendaftaran.....	97
Gambar 5.12 Tampilan Pesan Kesalahan Menggunakan JQuery	98
Gambar 5.13 Tampilan Halaman Cek Pendaftaran.....	100
Gambar 5.14 Tampilan Halaman Tes Tertulis.....	104
Gambar 5.15 Tampilan Form Ubah Status Tes Tertulis	105

Gambar 5.16 Tampilan Form Pilih Jurusan	105
Gambar 5.17 Tampilan Halaman Tes Wawancara	107
Gambar 5.18 Tampilan Form Konfigurasi PMB	107
Gambar 5.19 Susunan File PHP pada Modul User Berbasis Web.....	109
Gambar 5.20 Tampilan Halaman Utama Berbasis Web	110
Gambar 5.21 Tampilan Halaman Pendaftaran Berbasis Web.....	112
Gambar 5.22 Tampilan Halaman Data Pendaftar Berbasis Web.....	115
Gambar 5.23 Tampilan Halaman Kelulusan Berbasis Web.....	117
Gambar 5.24 Tampilan Halaman Kelulusan Tes Tertulis.....	119
Gambar 5.25 Tampilan Halaman Kelulusan Tes Wawancara	119
Gambar 5.26 Tampilan Form Konfirmasi.....	121
Gambar 5.27 Susunan File PHP pada Modul User Berbasis WAP	123
Gambar 5.28 Tampilan Halaman Utama Berbasis WAP.....	124
Gambar 5.29 Tampilan Halaman Pendaftaran Berbasis WAP	126
Gambar 5.30 Tampilan Halaman Data Pendaftar Berbasis WAP.....	127
Gambar 5.31 Tampilan Halaman Data Pendaftar Gelombang Khusus.....	128
Gambar 5.32 Tampilan Halaman Data Detail Pendaftar Berbasis WAP	131
Gambar 5.33 Tampilan Halaman Kelulusan Berbasis WAP	132
Gambar 5.34 Tampilan Hasil Tes Tertulis Berbasis WAP	133
Gambar 5.35 Tampilan Hasil Tes Wawancara Berbasis WAP.....	133
Gambar 5.36 Tampilan Form Konfirmasi Berbasis WAP	134

DAFTAR LAMPIRAN

Lampiran A Program Web Admin.....	145
Lampiran B Program Manajemen Admin.....	147
Lampiran C Program Manajemen Pendaftaran.....	149
Lampiran D Program Manajemen Seleksi Tes	163
Lampiran E Program Proses Pendaftaran Berbasis Web.....	169
Lampiran F Program Proses Seleksi Tes Berbasis Web.....	177
Lampiran G Program Proses Pendaftaran Berbasis WAP	182
Lampiran H Program Proses Seleksi Tes Berbasis WAP	192
<i>Curriculum Vitae</i>	203



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

INTISARI

Sistem informasi penerimaan mahasiswa baru berbasis web sudah diterapkan oleh beberapa universitas di Indonesia, salah satunya di STMIK AMIKOM Yogyakarta. Namun permasalahannya adalah calon mahasiswa masih banyak yang merasa kesulitan dengan sistem tersebut karena tidak adanya fitur konfirmasi pembayaran pendaftaran *online*. Kemudian permasalahan berikutnya adalah tidak semua calon mahasiswa mempunyai komputer terkoneksi dengan internet, sehingga perlu diterapkan sistem berbasis WAP agar calon mahasiswa juga bisa mengakses sistem melalui *handphone*.

Metode penelitian yang digunakan dalam penelitian ini adalah pengembangan sistem, yaitu eksperimen untuk merancang sistem informasi penerimaan mahasiswa baru berbasis web yang dapat diakses melalui *web browser* dan juga berbasis WAP yang dapat diakses melalui *handphone*. Sistem informasi penerimaan mahasiswa baru ini dibangun dengan bahasa pemrograman PHP, tag HTML, dan tag WML, serta memanfaatkan *database* MySQL sebagai *database server*.

Hasil dari penelitian ini adalah sistem informasi penerimaan mahasiswa baru berbasis web sudah memiliki kemampuan memberikan kemudahan bagi calon mahasiswa baru untuk memperoleh semua informasi tentang penerimaan mahasiswa baru dan melakukan proses pendaftaran serta konfirmasi pembayaran pendaftaran *online*. Bagi calon mahasiswa yang tidak mempunyai komputer terkoneksi internet, masih tetap bisa mendapatkan kemudahan mengakses sistem ini dengan menggunakan *handphone* yang dilengkapi dengan fasilitas WAP.

Kata Kunci: penerimaan mahasiswa baru, pendaftaran *online*, PHP, WML, MySQL, web, WAP.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

ABSTRACT

Web based admission of new student information system already applied by some universities in Indonesia, and STMIK AMIKOM Yogyakarta also applied it. But the problem is still a lot of prospective students who find it difficult with such systems lack features online registration payment confirmation. Then the next problem is not all candidates have a computer that connected to the internet, so it needs to WAP-based system applied to the prospective student can also access the system via mobile phone.

The research method in this research used the development of the system, namely designing experiments to web based admission of new student information system that can be accessed by web browsers and WAP-based that accessed by mobile phone. Admission of new student information system was built with the PHP programming language, HTML and WML tags, and MySQL as the database server.

The results of this study is web based admission of new student information system already has the ability to make it easier for prospective new students to obtain all the information about student admission and registration process as well as online registration payment confirmation. For prospective students who do not have internet connected computer, can access the system by using mobile phones equipped with WAP easily.

Keywords: admission of new student, online registration, PHP, WML, MySQL, web, WAP.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB I

PENDAHULUAN

1.1 Latar Belakang

World Wide Web atau sering disebut dengan WWW merupakan salah satu fasilitas internet yang sangat populer. Saat ini teknologi Web sudah digunakan oleh ribuan perusahaan atau pengusaha di dunia sebagai media promosi dan menawarkan produknya dalam bentuk informasi *online* melalui internet. Dengan internet penyampaian informasi menjadi lebih praktis, efektif dan efisien karena pengunjung website dapat dengan mudah mengetahui semua informasi tentang perusahaan atau bisnis dengan jelas dan akurat.

Sementara itu di zaman ini, perkembangan teknologi yang semakin canggih juga bisa memberikan ide untuk mencari alternatif dalam mengakses internet tanpa harus melalui komputer. Dari ide tersebut akhirnya muncul berbagai macam produk elektronik yang bisa digunakan untuk akses internet. Salah satunya adalah teknologi WAP atau *Wireless Application Protocol* yang ada pada *handphone*. WAP merupakan suatu protokol aplikasi yang memungkinkan *handphone* untuk dapat mengakses internet. Hadirnya teknologi WAP ini merupakan langkah maju di dunia komunikasi *seluler*. Dengan adanya WAP ,maka berbagai informasi tentang

perusahaan ataupun bisnis dapat di akses setiap saat hanya dengan menggunakan *handphone*.

Mengapa *handphone* yang dipilih, hal ini disebabkan karena *handphone* merupakan alat komunikasi *seluler* yang mudah dibawa kemana saja dan hampir semua kalangan masyarakat memilikinya. Dengan demikian diharapkan agar informasi tentang perusahaan atau bisnis mudah diakses oleh siapa saja yang memiliki *handphone* kapan saja dan dimana saja. Tentunya *handphone* tersebut juga harus mempunyai fasilitas WAP dan pulsa yang cukup untuk akses internet (Nurdiansyah, 2009).

Berangkat dari permasalahan tersebut, maka perlu dirancang sebuah sistem informasi penerimaan mahasiswa baru berbasis Web dan WAP. Sistem ini dibuat dengan tujuan memberikan kemudahan kepada calon mahasiswa baru dan orangtua dalam mengakses informasi Penerimaan Mahasiswa Baru (PMB). Informasi PMB merupakan salah satu informasi yang penting untuk diketahui mahasiswa ataupun orang tua mahasiswa. Misalnya informasi tentang PMB, informasi beasiswa, pilihan program studi dan biayanya, jadwal tes atau ujian PMB, pencarian data pendaftar, pengumuman hasil kelulusan tes, sekretariat PMB, dan melakukan pendaftaran serta konfirmasi pembayaran pendaftaran *online*.

Dengan penerapan teknologi WAP ini, nantinya calon mahasiswa baru bisa mendaftar atau mengisi formulir pendaftaran *online* cukup melalui *handphone*-nya

masing-masing, tanpa harus datang jauh-jauh untuk pengambilan formulir di kampus. Meskipun sistem informasi PMB di STMIK AMIKOM berbasis Web sudah dijalankan dan dapat diakses melalui internet sehingga calon mahasiswa baru bisa mendaftar dan mengisi formulir pendaftaran melalui Web, namun tentu saja akan lebih lengkap jika disertai teknologi WAP ini. Selain itu permasalahan lain yang dihadapi adalah tidak adanya fitur konfirmasi pembayaran *online* di sistem yang sebelumnya sehingga mahasiswa yang berada di luar kota tidak bisa melakukan konfirmasi pembayaran pendaftaran *online*. Padahal di zaman sekarang, mahasiswa ingin yang serba mudah dan cepat. Kemudian tidak semua calon mahasiswa baru mempunyai laptop yang terkoneksi dengan internet, sehingga membutuhkan teknologi WAP untuk membangun sistem yang bisa diakses melalui *handphone*.

Berdasarkan kenyataan tersebut, maka sistem informasi penerimaan mahasiswa baru berbasis Web tersebut perlu dikembangkan lagi. Kemudian perlu juga dibangun sistem berbasis WAP yang baru.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang sebuah sistem informasi pendaftaran mahasiswa baru berbasis Web yang disertai menu pendaftaran dan konfirmasi pembayaran pendaftaran secara *online*?
2. Bagaimana merancang sistem informasi pendaftaran mahasiswa baru berbasis WAP yang bisa diakses melalui *handphone*?

1.3 Batasan Masalah

Sistem yang dibuat dalam penyusunan skripsi ini adalah suatu sistem yang dapat memberikan informasi penerimaan mahasiswa baru dan pendaftaran *online* dengan dibatasi pada beberapa hal sebagai berikut:

1. Menggunakan bahasa pemrograman PHP dan WML untuk membangun sistem informasi PMB dan MySQL *sebagai database*.
2. Menggunakan WinWAP *Browser* untuk testing suatu halaman aplikasi WAP.
3. Tidak membahas jaringan GPRS (*General Pocket Radio System*).
4. *Handphone* yang digunakan untuk menguji sistem ini adalah Nokia seri 2730 classic atau 6303 classic atau E63 dengan menggunakan *browser* Opera Mini versi 4.1.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka tujuan penelitian ini adalah merancang suatu sistem yang berfungsi untuk memberikan informasi penerimaan mahasiswa baru dan melakukan pendaftaran serta konfirmasi pembayaran pendaftaran *online* melalui Web dan WAP, agar memudahkan proses pendaftaran calon mahasiswa baru, baik itu yang mempunyai laptop terkoneksi dengan internet maupun yang tidak mempunyai.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberi manfaat sebagai berikut:

1. Memberikan suatu sumbangan ide kepada pihak kampus untuk memudahkan mahasiswa ataupun orang tua mahasiswa dalam hal mengakses informasi penerimaan mahasiswa baru serta melakukan pendaftaran serta konfirmasi pembayaran pendaftaran *online* melalui Web dan WAP, dimana saja dan kapan saja.
2. Dapat dikembangkan untuk penelitian lebih lanjut dan memicu dikembangkannya sistem informasi penerimaan mahasiswa baru yang lebih kompleks lagi.

1.6 Keaslian Penelitian

Penelitian yang berhubungan dengan masalah sistem informasi berbasis Web ataupun WAP sudah pernah dilakukan, namun untuk sistem informasi penerimaan mahasiswa baru berbasis integrasi Web dan WAP belum pernah dilakukan khususnya di STMIK AMIKOM Yogyakarta. Keunggulan penelitian ini dibandingkan dengan penelitian sebelumnya adalah penerapan teknologi WAP ke dalam sistem informasi penerimaan mahasiswa baru yang sebenarnya cukup kompleks. Sementara itu *handphone* mempunyai keterbatasan dalam mengakses informasi dan melakukan proses-proses yang terkait dengan sistem tersebut.

BAB V

HASIL DAN PEMBAHASAN

5.1 Pengkodean

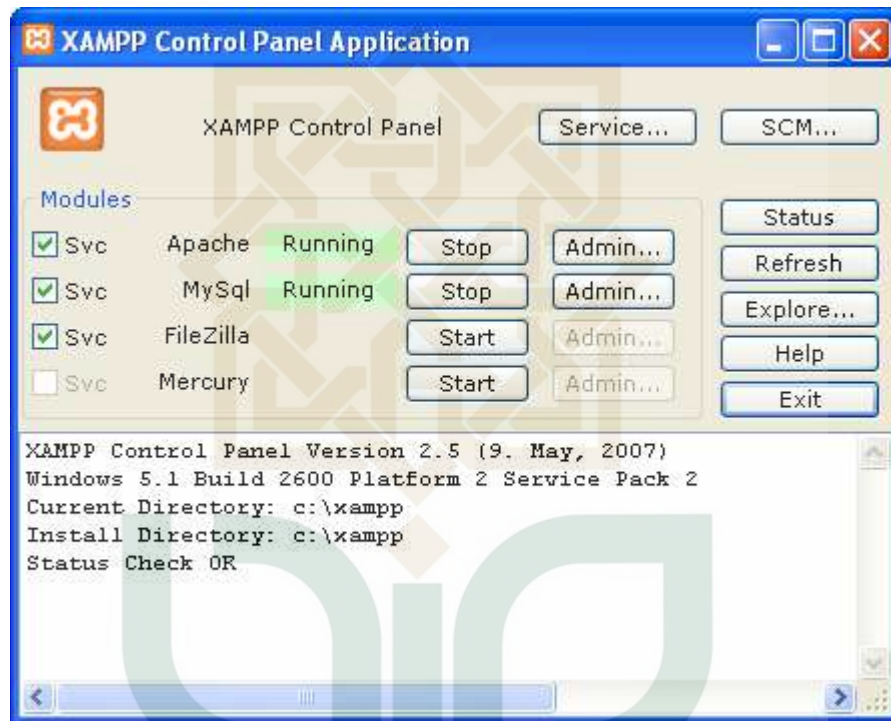
Di tahap pengkodean ini terdapat beberapa langkah untuk mengimplementasikan sistem, yaitu sebagai berikut:

5.1.1 Implementasi Sistem

Sistem ini dibuat menggunakan 2 teknologi yaitu web dan WAP. Untuk sistem berbasis web dibangun menggunakan bahasa pemrograman PHP dan database MySQL. Sistem berbasis web dapat diakses melalui *web browser*. Sedangkan untuk sistem berbasis WAP menggunakan bahasa pemrograman WML dan *database* MySQL, serta diakses melalui *handphone*. Sistem ini seperti yang sudah dibahas sebelumnya, mempunyai beberapa fitur dan modul terkait dengan proses pendaftaran mahasiswa baru.

Untuk menjalankan aplikasi pada bagian-bagian modul yang diinginkan, digunakan sebuah *web browser* yang dapat mengakses halaman web. Dalam penelitian ini, *web browser* yang digunakan adalah Mozilla Firefox versi 3.0.6. Dalam penelitian ini juga digunakan aplikasi XAMPP yang di dalamnya telah terintegrasi Apache *web server* , PHP, dan MySQL. Setelah instalasi XAMPP maka Apache, MySQL dan PHP yang terintegrasi di dalamnya juga terinstal. Untuk mengetahui apakah Apache dan MySQL sudah aktif maka dapat di buka melalui

XAMPP Control Panel, biasanya sudah ada di *desktop* atau *toolbars* setelah terinstal. Apabila Apache dan MySQL yang telah terinstal sudah aktif maka akan terlihat seperti pada gambar 5.1.



Gambar 5.1 Apache dan MySQL dalam keadaan aktif.

5.1.2 Implementasi Basis Data

Seperti yang telah dijelaskan sebelumnya, bahwa ada 17 tabel yang digunakan dalam merancang sistem informasi penerimaan mahasiswa baru ini. Masing-masing tabel tersebut diimplementasikan dengan SQL sebagai berikut:

```
CREATE TABLE `admin` (  
  `ad_id` int(11) NOT NULL auto_increment,  
  `ad_nama` varchar(20) NOT NULL,  
  `ad_alamat` varchar(100) NOT NULL,  
  `ad_tlp` varchar(20) NOT NULL,  
  `username` varchar(40) NOT NULL,  
  `password` varchar(40) NOT NULL,  
  `sa_id` int(11) NOT NULL,  
  PRIMARY KEY (`ad_id`)  
);  
  
CREATE TABLE `agama` (  
  `a_id` int(11) NOT NULL auto_increment,  
  `a_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`a_id`)  
);  
  
CREATE TABLE `beasiswa` (  
  `b_id` int(11) NOT NULL auto_increment,  
  `b_nama` bigint(20) NOT NULL,  
  `b_max` float NOT NULL,  
  `b_min` float NOT NULL,  
  PRIMARY KEY (`b_id`)  
);  
  
CREATE TABLE `gelombang` (  
  `g_id` int(11) NOT NULL auto_increment,  
  `g_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`g_id`)  
);  
  
CREATE TABLE `info` (  
  `i_id` int(11) NOT NULL auto_increment,  
  `i_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`i_id`)  
);  
  
CREATE TABLE `jenis` (  
  `j_id` int(11) NOT NULL auto_increment,  
  `j_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`j_id`)  
);  
  
CREATE TABLE `jurusan` (  
  `ju_id` int(11) NOT NULL auto_increment,  
  `ju_nama` varchar(100) NOT NULL,  
  PRIMARY KEY (`ju_id`)  
);
```

```
CREATE TABLE `jurusan_smu` (
  `js_id` int(11) NOT NULL auto_increment,
  `js_nama` varchar(20) NOT NULL,
  PRIMARY KEY (`js_id`)
);
```

```
CREATE TABLE `kabupaten` (
  `k_id` int(11) NOT NULL auto_increment,
  `pr_id` int(11) NOT NULL,
  `k_nama` varchar(20) NOT NULL,
  PRIMARY KEY (`k_id`)
);
```

```
CREATE TABLE `pendaftaran` (
  `p_id` int(11) NOT NULL auto_increment,
  `p_tgl` datetime NOT NULL,
  `p_nama` varchar(100) NOT NULL,
  `p_no_id` varchar(100) NOT NULL,
  `j_id` int(11) NOT NULL,
  `a_id` int(11) NOT NULL,
  `p_asal` varchar(100) NOT NULL,
  `js_id` int(11) NOT NULL,
  `p_nem` decimal(3,2) NOT NULL,
  `p_alamat` text NOT NULL,
  `w_id` int(11) NOT NULL,
  `k_id` int(11) NOT NULL,
  `p_kodepos` varchar(6) NOT NULL,
  `p_tlp` varchar(30) NOT NULL,
  `p_email` varchar(30) NOT NULL,
  `ju_id` int(11) NOT NULL,
  `ju_id2` int(11) NOT NULL,
  `ju_fix` int(11) NOT NULL,
  `i_id` int(11) NOT NULL,
  `t_id` int(11) NOT NULL default '1',
  `s_id` int(11) NOT NULL default '1',
  `g_id` int(11) NOT NULL default '1',
  `b_id` int(11) NOT NULL,
  `st_id` int(11) NOT NULL default '1',
  `ko_tgl` varchar(10) NOT NULL,
  `ko_jam` varchar(10) NOT NULL,
  `ko_no_rek` varchar(40) NOT NULL,
  PRIMARY KEY (`p_id`)
);
```

```
CREATE TABLE `pmb_config` (
  `g_id` int(11) NOT NULL,
  `pm_tahun` varchar(10) NOT NULL
);
```

```
CREATE TABLE `propinsi` (  
  `pr_id` int(11) NOT NULL auto_increment,  
  `pr_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`pr_id`)  
);  
  
CREATE TABLE `status` (  
  `st_id` int(11) NOT NULL auto_increment,  
  `st_nama` varchar(40) NOT NULL,  
  PRIMARY KEY (`st_id`)  
);  
  
CREATE TABLE `status_admin` (  
  `sa_id` int(11) NOT NULL auto_increment,  
  `sa_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`sa_id`)  
);  
  
CREATE TABLE `syarat` (  
  `s_id` int(11) NOT NULL auto_increment,  
  `s_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`s_id`)  
);  
  
CREATE TABLE `tempat` (  
  `t_id` int(11) NOT NULL auto_increment,  
  `t_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`t_id`)  
);  
  
CREATE TABLE `warga` (  
  `w_id` int(11) NOT NULL auto_increment,  
  `w_nama` varchar(20) NOT NULL,  
  PRIMARY KEY (`w_id`)  
);
```

5.1.3 Implementasi Koneksi MySQL dengan PHP

Agar *database* yang telah dimplementasikan di atas dapat diakses melalui *web browser* maupun *WAP browser*, maka diperlukan *script* untuk mengkoneksikan MySQL dengan PHP.

Untuk melakukan koneksi dengan *database* dibutuhkan file `config.inc.php` dan `class.mysql.php` yang berada pada direktori `“./program/inc”`. Di dalam file `config.php` terdapat inisialisasi dari parameter-parameter yang dibutuhkan. Kemudian di dalam file `class.mysql` terdapat fungsi dan prosedur yang berkaitan dengan *query* SQL. Adapun kedua file yang dibuat tersebut adalah sebagai berikut:

file config.inc.php
<pre> <?php \$host = 'localhost'; \$user = 'root'; \$pass = ''; \$database = 'amikom'; \$timezone = -5; \$home = "pendaftaran.php"; \$my_email = "mobildakwah@gmail.com"; \$order = 'DESC'; \$page_rows = '6'; \$table = 'pendaftaran'; \$sign = "pendaftaran.php"; \$post = "post.php"; \$show = "result.wml"; \$show_inc = "result.php"; \$version = "v1.5"; ?> </pre>

file class.mysql.php

```

<?php
class Mysql{
    var $db;
    var $host;
    var $user;
    var $password;
    var $database;
    var $query;
    var $result;
    var $row;
    var $dataSet;
    var $numRows;

    function
Mysql($host="localhost", $user="root", $password="", $database="amikom"
){
        $this->host=$host;
        $this->user=$user;
        $this->password=$password;
        $this->database=$database;
    }

    function connect(){
        $this->db=mysql_connect($this->host,$this-
>user,$this->password);
        mysql_select_db($this->database,$this->db);
    }

    function execute($query){
        $this->query=$query;
        if($this->result=mysql_query($query,$this->db) ){
            return true;
        } else {
            return false;
        }
    }

    function getDataSet(){
        $dataSet=array();
        $i=0;
        while($r=mysql_fetch_row($this->result)){
            $field=0;
            for($field=0;$field<mysql_num_fields($this-
>result);$field++){
                $dataSet[$i][$field]=$r[$field];
            }
            $i++;
        }
        return $dataSet;
    }
}

```

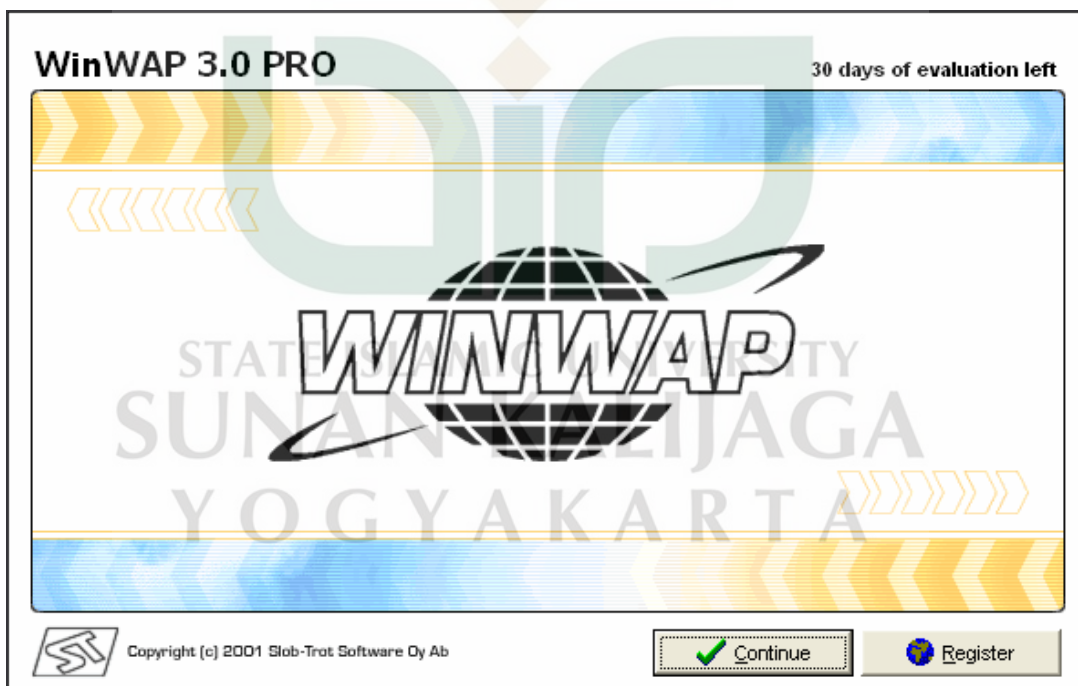
```

function getNumRows() {
    $this->numRows=mysql_num_rows($this->result);
    return $this->numRows;
}
function closeConnection(){
    mysql_close($this->db);}
}
?>

```

5.1.4 Implementasi WinWAP Browser

Untuk menjalankan aplikasi WAP, digunakan sebuah WAP *browser* yang dapat mengakses halaman tersebut. Dalam penelitian ini, WAP *browser* yang digunakan adalah WinWAP Browser versi 3.0 . Setelah instalasi WinWAP sukses, maka aplikasi WinWAP tersebut jika dijalankan akan muncul tampilan default seperti gambar 5.2.



Gambar 5.2 Aplikasi WinWAP browser

5.1.5 Implementasi Konfigurasi File WML

Agar Apache dapat mendukung file berekstensi wml, maka diperlukan konfigurasi khusus, yaitu pada file httpd.conf harus diberikan definisi file yang nantinya akan diakses pada *web server* Apache. File httpd.conf tersebut terdapat pada folder C:\xampp\apache\conf. Kemudian lakukan perubahan pada IfModule dir_module (baris 237) menjadi seperti berikut:

```
<IfModule dir_module>
    DirectoryIndex index.html
    DirectoryIndex index.php
    DirectoryIndex default.php
    DirectoryIndex main.php
    DirectoryIndex index.wml
</IfModule>
```

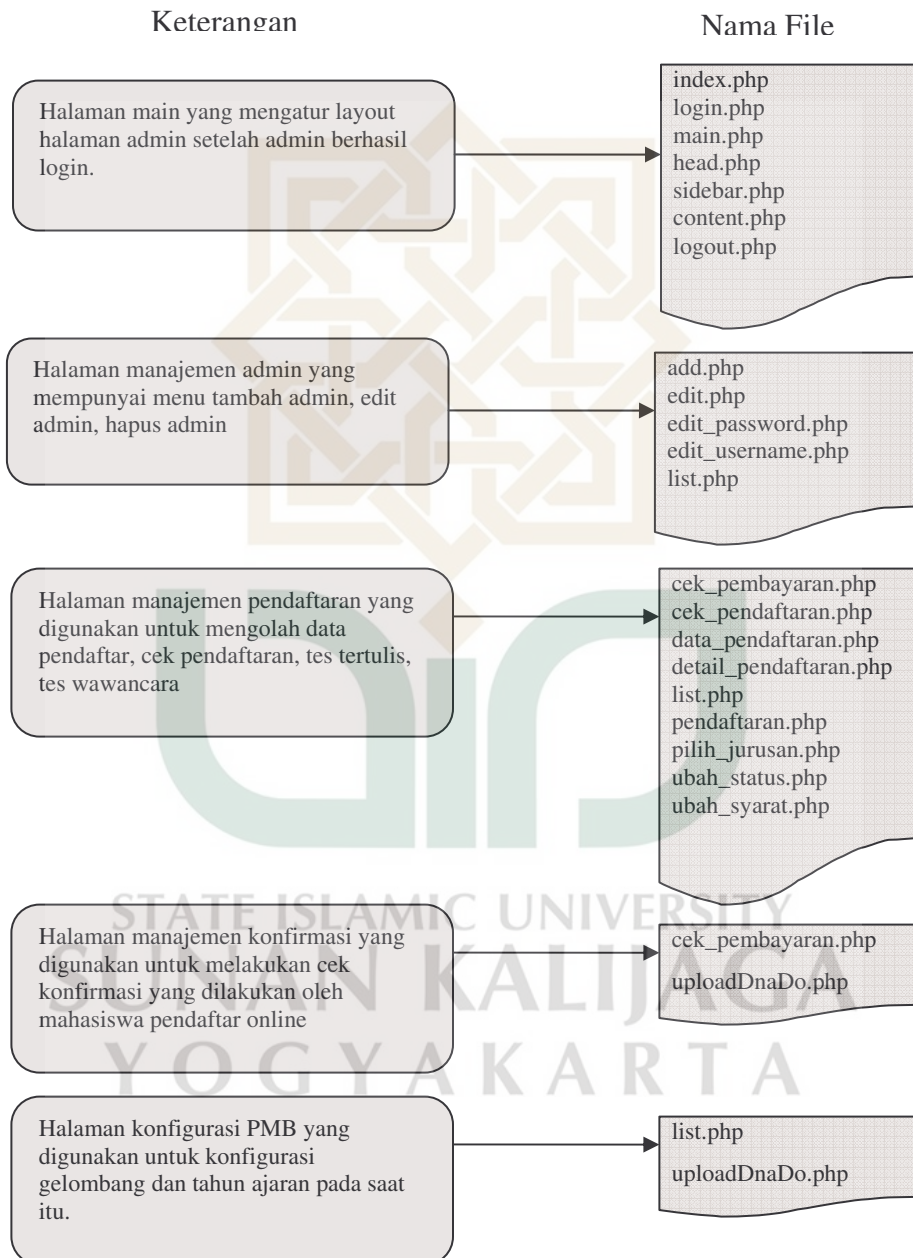
Lalu deskripsi AddType nya juga diubah menjadi seperti berikut:

```
<IfModule mime_module>
    AddType application/x-httpd-wap .wml
    AddType text/vnd.wap.wml .wml
    AddType image/vnd.wap.wbmp .wbmp
    AddType text/vnd.wap.wmlscript .wmlsc
    AddType text/vnd.wap.wmlscript .wmls
    AddType Application/vnd.wap.wmlc .wmlc
    AddType Application/vnd.wap .wmlscript
    AddType text/vnd.wap .wmlscript
```

5.1.6 Implementasi Modul Administrator

Modul administrator diakses melalui web digunakan untuk mengelola data yang berada pada sistem ini, seperti data admin, data pendaftaran, data konfirmasi,

data pmb_config, dan sebagainya. Susunan file-file PHP pada modul administrator dapat dilihat pada gambar 5.3.



Gambar 5.3 Susunan File PHP pada Modul Administrator

5.1.6.1 Implementasi Halaman Utama Modul Administrator

Halaman utama modul administrator diimplementasikan oleh file `index.php`. Halaman ini *include* file `head.php` untuk menampilkan header dan file `login.php` untuk menampilkan form login admin. Untuk menjalankan halaman utama modul administrator dapat dijalankan dengan mengetikkan <http://localhost/program/admin> atau <http://localhost/program/admin/index.php> pada *address bar* sehingga muncul halaman utama modul administrator yang intinya menampilkan form login seperti gambar 5.4.

The image shows a web form titled "login area" on a purple background. It contains two input fields labeled "username" and "password". Below the password field is a CAPTCHA image with the text "qy75vc". A "Login" button is located at the bottom right of the form. The background of the image has a watermark of the State Islamic University Sunan Kalijaga logo and name.

Gambar 5.4 Tampilan Form Login untuk Admin

Form login yang berada pada halaman `index` tersebut diimplementasikan oleh file `login.php`. Di form tersebut terdapat *captcha security images* yang diproses menggunakan file `captchasecurityimages.php`. Kemudian ketika user menekan tombol login, maka aksinya akan ditangani oleh file `login.php` itu.

Apabila form username, password, *captcha security images* belum terisi atau salah, maka *browser* akan menampilkan pesan *java script* bahwa username, password, *captcha security images* masih salah. Untuk melakukan aksi tersebut digunakan *script* berikut:

```
if(count($cekLoginAdmin)==0)
{
    echo"<Script>alert('username dan password belum
benar');window.history.go(-1)</script>";
}
else ...
{
    echo"<Script>alert('masukkan security code dengan
benar');window.history.go(-1)</script>";
}
...
```

Kode program diatas akan menampilkan pesan kesalahan seperti pada gambar 5.5 dan gambar 5.6.



Gambar 5.5 Pesan Javascript ketika Username dan Password Salah



Gambar 5.6 Pesan Javascript ketika Inputan Captcha Security Salah

Apabila username dan password yang dimasukkan benar, maka *script* akan membuat sesi baru menggunakan perintah `session_start()`; kemudian mengisikan data tersebut ke variabel sesi seperti pada potongan *script* berikut:

```
$_SESSION['loginadmin']=$cekLoginAdmin[0][0];
$_SESSION['status']=$cekLoginAdmin[0][1];
$_SESSION['namaadmin']=$cekLoginAdmin[0][4];
$_SESSION['adminstatus']=$cekLoginAdmin[0][5];
```

Maksud dari penggunaan sesi tersebut adalah agar admin bisa bebas mengakses menu-menu admin selama admin belum logout dari halaman admin. Kemudian setelah itu *browser* akan menampilkan halaman utama admin yang berisi menu-menu yang dapat diakses seperti pada gambar 5.7. Kode sumber (*source code*) halaman utama modul administrator dapat dilihat pada lampiran A.



Gambar 5.7 Tampilan Halaman main.php setelah Admin Berhasil Login

5.1.6.2 Implementasi Halaman Manajemen Admin

Halaman manajemen admin diimplementasikan oleh file list.php. Halaman ini digunakan untuk mengelola data admin. Halaman ini hanya dapat diakses oleh super admin dan admin seperti yang sudah dijelaskan sebelumnya di Bab 3. Halaman ini terdapat tiga *link* untuk mengelola data admin, yaitu tambah admin, edit admin, dan hapus admin. Kode Sumber (*source code*) proses manajemen admin dapat dilihat pada lampiran B.

Halaman list.php adalah halaman yang digunakan untuk menampilkan data admin dalam bentuk tabel. Pertama-tama, select data dari tabel admin dan tabel status_admin dengan perintah berikut:

```
$mysql->execute("select a.ad_id, a.ad_nama, a.ad_alamat,
a.ad_tlp, a.username, b.sa_nama, a.sa_id from admin a,
status_admin b where a.sa_id=b.sa_id order by a.sa_id ")
```

Setelah data ditemukan, maka data disimpan ke dalam suatu *array* menggunakan perintah berikut:

```
$getdataA = $mysql->getDataSet();
```

Script `getDataSet()` tersebut akan memanggil fungsi `getDataSet` yang ada di file `class.mysl.php`. Setelah itu, hasil datanya akan ditampilkan oleh *web browser*, seperti contoh pada potongan *script* berikut:

```
<?=$getdataA[$i][2];?>
```

Script tersebut bertugas untuk menampilkan data di kolom alamat. *Script* mengambil data yang ada di field 2 yaitu a.ad_alamat (berdasarkan *query* select sebelumnya), kemudian menampilkan dalam bentuk array [\$i] .Kalau datanya ada 10 berarti nilai \$i ini diisi dari 1-10 sehingga bisa menampilkan 10 data di kolom alamat tersebut. Untuk menampilkan data di kolom yang lainnya pun logikanya juga sama seperti ini. Sehingga halaman list.php akan ditampilkan oleh *web browser* seperti pada gambar 5.8



.: Halaman Konfigurasi .:

List Admin

Selamat datang "amikom", anda login sebagai "Super Admin"
 >> Berikut ini adalah List Admin yang ada di web ini.

No	Nama	Alamat	Telp	Username	Status	Action
1	amikom	tes	089876656333	webmaster	Super Admin	
2	admin	disini	123456	pengusaha	Admin	 

Gambar 5.8 Tampilan Halaman list.php

Ketika admin memilih menu tambah admin, maka halaman akan menuju ke add.php sesuai *script* berikut ini:

```
<a href="main.php?page=5" style="text-decoration:none" >
<?php if($_SESSION['status']==1){ ?> <input name="button"
type="button" value="Tambah Admin"/><?php } ?>
</a>
```

Maksud dari main.php?page=5 adalah add.php. Kemudian di *script* tersebut ada perintah if, berguna untuk melakukan cek status admin yang login. Jika yang login adalah super admin, maka diperbolehkan mengakses menu tambah admin, namun jika

yang login hanya admin biasa, maka tidak diijinkan untuk mengakses menu tambah admin (menu tambah admin tidak akan muncul). Form inputan untuk menambah admin sama dengan yang sudah dijelaskan sebelumnya di tahap desain.

Data yang dimasukkan melalui form tersebut akan di eksekusi di halaman `add.php` itu sendiri. Pertama-tama data dikirim ke masing-masing variabel menggunakan perintah `$_POST`, misalkan `$nama = $_POST['nama'];` untuk mengirimkan data nama ke variable `$nama`, dan seterusnya. Setelah itu, data admin tersebut dimasukkan ke database menggunakan *query* insert terhadap tabel admin dengan perintah berikut:

```
$mysql->execute("INSERT INTO admin (ad_nama, ad_alamat, ad_tlp, username, password, sa_id) values ('$nama', '$alamat', '$no_tlp', '$username', md5('$password'), '$id_status')")
```

Jika *query* sukses dijalankan, maka halaman akan otomatis *refresh* menuju ke halaman `list.php`. Tentunya sekarang halaman tersebut sudah menampilkan data admin yang baru saja ditambahkan.

Begitu juga saat melakukan proses edit, form untuk mengubah data hampir sama dengan form untuk tambah data. Perbedaannya, di form edit ini fieldnya tentu sudah ada datanya, yaitu data admin lama yang akan diedit. Kemudian form tersebut akan di eksekusi di halaman `edit.php` itu sendiri. Form untuk melakukan proses edit admin dapat dilihat pada gambar 5.9.

:: Halaman Konfigurasi ::

Admin Konfigurasi

Selamat datang "amikom" , anda login sebagai "Super Admin"
 >> ini form untuk merubah Admin.

Nama :

Alamat :

No Telp :

Status :

Username : webmaster >>> [change](#)

Password : ***** >>> [change](#)

Gambar 5.9 Form untuk Mengubah Data Admin

Setelah tombol ubah di klik, maka *script* akan menjalankan *query* update data seperti pada potongan *script* berikut ini:

```
$mysql->execute("Update admin set ad_nama='$nama',
ad_alamat='$alamat', ad_tlp='$no_tlp', sa_id='$id_status' where
ad_id=$a_id")
```

Di tampilan form edit tersebut, terlihat juga ada *link* untuk ubah username ([edit_username.php](#)) dan ubah password ([edit_password.php](#)). Lalu juga ada *drop down* list untuk memilih status admin. Pilihan status admin ini ada 2 yaitu super admin dan admin. Untuk yang login sebagai super admin bisa bebas memilih pilihan

statusnya, sedangkan yang login sebagai admin biasa tidak bisa menaikkan sendiri pangkatnya menjadi super admin tanpa seijin pengelola sistem. Karena jika yang login admin biasa, secara otomatis program edit.php ini akan *men-disabled* (menonaktifkan) pilihan status super admin.

Untuk proses delete, yang mempunyai hak hanyalah super admin. Admin biasa tidak diijinkan untuk menghapus anggota admin. Karena jika yang login admin biasa, secara otomatis action *link* delete juga tidak akan muncul. Seperti yang dijelaskan pada potongan *script* berikut:

```
<?php if(($_SESSION['status']==1) && ($getdataA[$i][6]>1)){ ?>
</a></div>
<?php } ?>
```

Maksud dari *script* tersebut adalah jika nilai variabel session status=1 yaitu super admin dan data status admin yang lain adalah admin biasa, maka tampilkan *link* delete di kolom action admin biasa tersebut. Jadi, di kolom super admin nya pun nanti tidak ada *link* delete data dirinya sendiri, karena super admin juga tidak diijinkan melakukan hal tersebut tanpa sepengetahuan pengelola sistem. Kemudian untuk *query* delete nya adalah sebagai berikut:

```
$a_id= (int) addslashes($_GET['a_id']);
mysql->execute("delete from admin where ad_id=$a_id");
```

5.1.6.3 Implementasi Proses Pendaftaran

Halaman untuk mengelola proses pendaftaran ini diimplementasikan oleh 9 file program seperti yang sudah dibahas sebelumnya di susunan file PHP. Halaman

ini terdiri dari 4 menu yaitu data pendaftar untuk mengelola data pendaftaran, cek pendaftaran untuk mengubah status syarat pendaftaran, tes tertulis untuk menampilkan data mahasiswa yang sedang atau sudah tes tertulis, tes wawancara untuk mengubah besarnya beasiswa dan mengubah pilihan jurusan yang diterima. Kode Sumber (*source code*) proses manajemen pendaftaran dapat dilihat pada lampiran C.

1. Menu data pendaftar

















Halaman `data_pendaftaran.php` adalah halaman yang dituju ketika admin mengakses menu data pendaftar yang ada di menu admin. Tugas `data_pendaftaran.php` ini adalah menampilkan seluruh data pendaftaran yang sudah masuk ke sistem. Kemudian ada juga pilihan untuk menampilkan data pendaftar sesuai gelombang. Di tampilan data pendaftarnya juga disertai *link* untuk edit data pendaftar dan juga hapus data pendaftar. Tampilan dari halaman `data_pendaftaran.php` seperti gambar 5.10.

:: Halaman Konfigurasi ::

Data Pendaftar

Selamat datang "amikom", anda login sebagai "Super Admin"
 >> Berikut ini adalah List Admin yang ada di web ini.

- Semua Gelombang -

No	Nama	NEM	Pilihan 1	Pilihan 2	Gelombang	Action
1	surya	8.78	D3 Manajemen Informatika	S1 Teknik Informatika	Gelombang Khusus	 
2	anton.	6.50	D3 Teknik Informatika	S1 Teknik Informatika	Gelombang Khusus	 
3	lintang	6.50	S1 Sistem Informasi	S1 Teknik Informatika	Gelombang I	 
4	akulah	6.25	D3 Manajemen Informatika	S1 Sistem Informasi	Gelombang I	 
5	alex	6.75	S1 Sistem Informasi	S1 Teknik Informatika	Gelombang I	 
6	aryanto	8.50	S1 Sistem Informasi	S1 Teknik Informatika	Gelombang I	 
7	wijaya	6.00	D3 Manajemen Informatika	D3 Teknik Informatika	Gelombang I	 
8	sulistyo	7.00	S1 Sistem Informasi	D3 Teknik Informatika	Gelombang I	 

Gambar 5.10 Tampilan Halaman Data Pendaftar untuk Admin

Untuk menu tambah pendaftaran merupakan sebuah *link* yang mengarah ke halaman pendaftaran.php yang berupa form untuk memasukan data pendaftaran *offline* ke sistem. Menu tersebut dibuat dengan *script* berikut:

```
<a href="main.php?page=14" style="text-decoration:none" ><input type="submit" name="Submit" value="Tambah Pendaftaran" /></a>
```

Form untuk memasukan data pendaftaran dapat dilihat pada gambar 5.11.

:: Halaman Konfigurasi ::

Pendaftaran Calon Mahasiswa STMIK "AMIKOM" Yogyakarta
Tahun Akademik: 2010/2011

Nama Lengkap * :

Nomor Identitas * :
(KTP/SIM/Paspor)

Jenis Kelamin * : Pria Wanita

Agama * : --Pilih Agama--

Asal Sekolah * :

Jurusan SMU * : Pilih Salah Satu

Rata-rata NEM * : ,

Alamat * :

Kewarganegaraan * : WNI WNA

Propinsi * : -Pilih data-

Kabupaten* :

Kode Pos :

Telepon :

E-mail :

Pilihan 1* : Pilih Salah Satu

Gambar 5.11 Tampilan Halaman Tambah Pendaftaran

Ketika data telah diisikan dan tombol kirim ditekan, maka form tersebut akan melakukan aksi ke halaman pendaftaran.php. Apabila form tersebut tidak diisi, maka akan muncul sebuah pesan kesalahan yang dibuat menggunakan JQuery. File program JQuery ini harus diletakkan dalam folder sendiri yaitu jquery. Program yang digunakan untuk validasi pendaftaran ini ada 2 yaitu jquery-1.2.3.pack.js dan

jquery.validate.pack.js. Contoh tampilan kesalahan yang muncul adalah seperti gambar 5.12.

Gambar 5.12 Tampilan Pesan Kesalahan Menggunakan JQuery

Setelah form tersebut diisi lengkap dan user klik *button* daftar, maka akan muncul pesan bahwa data bisa disimpan, setelah itu otomatis halaman akan *refresh* ke halaman `data_pendaftaran.php`. Tentunya sekarang halaman tersebut juga sudah menampilkan data pendaftaran yang baru saja ditambahkan. *Query* yang digunakan untuk insert data pendaftaran adalah sebagai berikut:

```
$mysql->execute("INSERT INTO pendaftaran (p_tgl, p_nama,
p_no_id, j_id, a_id, p_asal, js_id, p_nem, p_alamat, w_id,
p_kodepos, p_tlp, p_email, ju_id, ju_id2, i_id, t_id, b_id,
g_id, k_id) values (NOW(), '$nama', '$no_id', '$jk', '$agama',
'$sekolah', '$jurusan', '$nem1', '$salamat', '$swarga',
'$kodepos', '$telepon', '$email', '$pilihan1', '$pilihan2',
'$info', '$tempat', '$beasiswa', '$gelombang', '$k_id')")
```

Sedangkan untuk menu edit dan delete pada data pendaftaran ini sebenarnya logikanya sama dengan menu edit dan delete pada data admin yang sebelumnya. Yang perlu diperhatikan adalah ketika *cursor* mouse diarahkan ke *link* edit atau hapus. Jika *cursor* mouse diarahkan ke *link* edit, maka di *toolbars* sebelah kiri bawah pasti akan muncul URL berikut ini :

http://localhost/program/admin/main.php?page=12&a_id=32

URL tersebut didapatkan dari *script* berikut:

```
<a href="main.php?page=12&a_id=<?=$getdataA[$i][0];?>"></a>
```

Dari *script* diatas, bisa disimpulkan bahwa *link* tersebut membawa parameter *a_id* dengan nilai 32. Parameter *a_id* yang dibawa ini digunakan untuk select data pendaftaran yang akan diedit (di file *detail_pendaftaran.php*). Sehingga data yang ditampilkan di form edit hanyalah data yang *a_id* nya sama dengan parameter *a_id* yang dibawa. Untuk form editnya sama dengan form ketika menambah pendaftaran. Perbedaannya, di form edit ini ditampilkan data pendaftaran yang akan diedit. Jadi, fieldnya tidak kosong lagi.

Sedangkan jika *cursor* mouse diarahkan ke *link* hapus, URL yang dituju adalah http://localhost/program/admin/main.php?page=11&delete=1&a_id=33

URL tersebut didapatkan dari *script* berikut:

```
<a
href="main.php?page=11&delete=1&a_id=<?=$getdataA[$i][0];?>"></a>
```

Dari *script* diatas, bisa disimpulkan juga bahwa *link* tersebut membawa parameter *delete* dengan nilai 1 dan parameter *a_id* dengan nilai 33. Parameter *delete=1* yang dibawa ini digunakan untuk menjalankan *query* delete data pendaftaran (di file *data_pendaftaran.php*). Kemudian parameter *a_id* digunakan untuk menentukan data yang akan di delete. *Query* delete data pendaftarannya adalah sebagai berikut:

```
if($_GET['delete']==1){
    $a_id= (int) addslashes($_GET['a_id']);
    $mysql->execute("delete from pendaftaran where p_id=$a_id");}
```

2. Menu Cek Pendaftaran

Halaman `cek_pendaftaran.php` adalah halaman yang dituju ketika admin mengakses menu cek pendaftaran yang ada di menu admin. File `cek_pendaftaran.php` ini digunakan untuk menampilkan seluruh data pendaftaran yang sudah masuk ke sistem, namun disertai *link* di kolom syarat. *Link* ini menuju ke halaman `ubah_syarat.php` yang bertugas untuk menampilkan data pendaftaran secara detail dan *drop down* list untuk mengubah syarat pendaftaran mahasiswa tersebut. Tampilan dari halaman `cek_pendaftaran.php` seperti gambar 5.13.

:: Halaman Konfigurasi ::						
Seleksi Syarat Pendaftaran						
Selamat datang "amikom", anda login sebagai "Super Admin"						
>> Berikut ini adalah List Pendaftaran yang ada di sistem ini.						
No	Nama	NEM	Pilihan 1	Pilihan 2	Syarat	Gelombang
1	akulah	6.25	D3 Manajemen Informatika	S1 Sistem Informasi	Dalam Proses	Gelombang I
2	alex	6.75	S1 Sistem Informasi	S1 Teknik Informatika	Dalam Proses	Gelombang I
3	sulistyo	7.00	S1 Sistem Informasi	D3 Teknik Informatika	Dalam Proses	Gelombang I

Gambar 5.13 Tampilan Halaman Cek Pendaftaran

Ketika link di kolom syarat tersebut di klik, maka akan menuju halaman `ubah_syarat` dengan membawa parameter sesuai *script* berikut:

```
<a href="main.php?page=8&id=?=$getdataA[$i][0];?>">
<? =$getdataA[$i][8];?></a>
```

Script `<?=$getdataA[$i][0];?>` bertugas untuk membawa parameter id dengan nilai yang sesuai dengan nilai field `p_id` di dalam *array*. Parameter id ini kemudian

diproses di halaman `ubah_syarat.php` untuk menampilkan data pendaftaran secara detail. Setelah itu admin bisa mengubah syarat pendaftaran mahasiswa di field syarat. Sedangkan *script* `<? = $getdataA[$i][8];?>` bertugas untuk menampilkan data syarat dalam bentuk *array* saat berada di halaman `cek_pendaftaran.php`.

Untuk field syarat memiliki nilai default =1 yang berarti dalam proses. Admin bisa mengubah syarat pendaftaran ini setelah admin melakukan cek semua persyaratan pendaftaran dan pembayaran pendaftaran. Jika mahasiswa mendaftar secara *offline*, maka saat menginputkan data pendaftaran ke sistem, admin pun bisa langsung mengubah syarat tersebut menjadi lengkap setelah admin memastikan bahwa syarat pendaftaran dari mahasiswa tersebut sudah lengkap dan juga sudah melakukan pembayaran uang pendaftaran. Untuk menampilkan *drop down* list ubah syarat menggunakan *option value* terdapat pada *script* berikut:

```
<select name="syarat" >
  <option value="<?=$dataadmin[0][26];?>">-
  <?=$dataadmin[0][18];?>-</option>
  <?
  //list menu
  for ($i=0; $i<count($datastatus); $i++){
    $status=$datastatus[$i][0];
    ?>
  <option value="<?=$datastatus[$i][0]?>">
  <?=$datastatus[$i][1]?></option>
  <?
  }
  ?>
</select>
```

Di *script* diatas terdapat perintah *looping for* untuk menampilkan pilihan syarat yang diambil dari tabel syarat. Setelah admin melakukan perubahan dan menekan tombol

ubah, maka akan dieksekusi di halaman `ubah_syarat.php` itu sendiri. Seperti yang terdapat pada potongan *script* berikut:

```
if(isset($_POST['edit'])){
    $syarat = $_POST['syarat'];

    if($mysql->execute("Update pendaftaran set s_id='$syarat'
    where p_id=$id")){
        echo"<meta http-equiv='refresh'
        content='1;URL=main.php?page=7'>";
    }
}
```

Script tersebut menjelaskan bahwa jika tombol edit di klik maka kirimkan nilai syarat ke dalam variabel `$syarat`, kemudian jalankan *query* update data. Setelah update data berhasil, maka otomatis halaman akan *refresh* menuju halaman `cek_pendaftaran.php` dengan tampilan data yang terbaru.

Ketika admin mengubah syarat pendaftaran menjadi lengkap dan menekan tombol ubah, maka saat *refresh* halaman, data mahasiswa yang baru saja diubah syaratnya menjadi lengkap tidak akan ditampilkan lagi di menu cek pendaftaran tersebut (`cek_pendaftaran.php`). Namun data mahasiswa tersebut akan ditampilkan di menu tes tertulis (`list.php`). Hal ini bisa terjadi karena di file `cek_pendaftaran.php` hanya akan melakukan select data dimana `a.s_id = 1 or a.s_id=3` yang berarti bahwa status pendaftarannya dalam proses atau tidak lengkap.

Ada satu hal lagi yang penting yaitu field pembayaran pendaftaran. Jika data pendaftaran mahasiswa tersebut diinputkan admin secara manual, maka di field tersebut akan muncul data bahwa pembayaran *offline*. Namun, jika data pendaftaran dilakukan secara *online* oleh mahasiswa, maka di field tersebut akan muncul *link* untuk cek konfirmasi. *Link* cek konfirmasi ini menuju ke halaman

cek_pembayaran.php. Namun admin hanyalah sebatas melakukan cek konfirmasi yang sudah masuk ke sistem saja. Setelah cek ke sistem, admin harus melakukan cek lewat *e-banking* untuk memastikan apakah pembayaran tersebut benar-benar masuk. Jika benar sudah masuk, maka admin bisa mengubah syarat pendaftaran mahasiswa tersebut menjadi lengkap sehingga bisa melanjutkan ke tahap berikutnya yaitu tes tertulis atau tes wawancara.

5.1.6.4 Implementasi Proses Seleksi Tes

Proses seleksi tes di modul administrator ini merupakan proses untuk mengelola atau manajemen hasil seleksi tes. Proses ini terdiri dari menu tes tertulis dan tes wawancara. Halaman ini diimplementasikan oleh file list.php. Kode Sumber (*source code*) proses seleksi tes dapat dilihat pada lampiran D.

1. Menu Tes Tertulis

Seperti yang sudah dijelaskan sebelumnya, bahwa mahasiswa yang memiliki nilai NEM < 7 dan status pendaftarannya sudah lengkap, maka datanya akan ditampilkan di menu ini (list.php). Di halaman ini juga disertai *link* untuk ubah status tes. *Link* tersebut menuju ke halaman ubah_status.php yang bertugas untuk menampilkan form ubah status tes mahasiswa tersebut. Tampilan dari halaman list.php adalah seperti gambar 5.14.

:: Halaman Konfigurasi ::								
List Pendaftaran [Tes Tertulis]								
Selamat datang "amikom", anda login sebagai "Super Admin"								
>> Berikut ini adalah List Pendaftaran yang ada di web ini.								
No	Nama	NEM	Beasiswa	Pilihan 1	Pilihan 2	Jurusan Diterima	Status Tes	Gelombang
1	anton	6.50	700000	D3 Teknik Informatika	S1 Teknik Informatika	S1 Teknik Informatika	Lulus Tes Tertulis	Gelombang Khusus
2	lintang	6.50	700000	S1 Sistem Informasi	S1 Teknik Informatika	S1 Sistem Informasi	Lulus Tes Tertulis	Gelombang I
3	sidiq	6.00	---	D3 Manajemen Informatika	D3 Teknik Informatika	---	Tes Tertulis	Gelombang I

Gambar 5.14 Tampilan Halaman Tes Tertulis

Ketika *link* di kolom status tes tersebut di klik, maka akan menuju halaman `ubah_status` dengan membawa parameter sesuai *script* berikut:

```
<a href="main.php?page=9&id=?=$getdataA[$i][0];?>&t=?=$t;?>">
<?=$getdataA[$i][9];?></a>
```

Script `<?=$getdataA[$i][0];?>` bertugas untuk membawa parameter `id` dengan nilai yang sesuai dengan nilai field `p_id` di dalam *array*. Parameter `id` ini kemudian diproses di halaman `ubah_status.php`. Kemudian `t=?=$t;?>` bertugas untuk membawa parameter `t` dengan nilai yang didapatkan dari *script* `$t=(int)addslashes($_GET['t']);`. Sedangkan *script* `<?=$getdataA[$i][9];?>` bertugas untuk menampilkan data status dalam bentuk *array* saat berada di halaman `list.php`. Tampilan form untuk mengubah status tes mahasiswa tersebut dapat dilihat pada gambar 5.15.

Gambar 5.15 Tampilan Form Ubah Status Tes Tertulis

Setelah admin mengubah status menjadi lulus tes tertulis dan menekan tombol ubah, maka otomatis halaman akan *refresh* menuju halaman pilih_jurusan.php. Halaman pilih_jurusan.php ini merupakan form untuk mengubah besarnya beasiswa dan mengubah pilihan jurusan diterima. Setelah admin memilih besar beasiswa dan memilih jurusan dan menekan tombol ubah, maka halaman akan *refresh* menuju halaman list.php sebelumnya, tentunya dengan data yang sudah diperbarui. Tampilan halaman pilih jurusan adalah seperti gambar 5.16.

Gambar 5.16 Tampilan Form Pilih Jurusan

2. Menu Tes Wawancara

Seperti yang sudah dijelaskan sebelumnya, bahwa mahasiswa yang memiliki nilai NEM > 7 dan status pendaftarannya sudah lengkap, maka datanya akan ditampilkan di menu ini (list.php) dengan membawa parameter $t=2$. Tampilan pertama di halaman ini sama seperti dengan halaman tes tertulis sebelumnya. Setelah admin mengubah status tes menjadi lulus tes wawancara, maka halaman akan *refresh* menuju halaman pilih_jurusan.php, namun tanpa ada pilihan untuk mengubah beasiswa. Karena jika NEM > 7 beasiswa secara otomatis akan diisi oleh sistem ini sesuai dengan data yang ada di database menggunakan *script* menentukan besarnya beasiswa.

Setelah admin mengubah beasiswa dan pilihan jurusan diterima dan menekan tombol ubah, maka otomatis halaman akan *refresh* menuju halaman list.php. Field data jurusan diterima yang tadinya kosong, sekarang tentu sudah ada datanya yang diambil dari ju_fix dalam tabel pendaftaran. *Query* update data dari form tersebut adalah sebagai berikut:

```
$mysql->execute("Update pendaftaran set ju_fix='$jurusan',  
b_id='$beasiswa' where p_id=$id")
```

Tampilan halaman tes wawancara adalah seperti gambar 5.17.

.. Halaman Konfigurasi ..

List Pendaftaran [Tes Wawancara]

Selamat datang "amikom", anda login sebagai "Super Admin"
 >> Berikut ini adalah List Pendaftaran yang ada di web ini.

No	Nama	NEM	Beasiswa	Pilihan 1	Pilihan 2	Jurusan Diterima	Status Tes	Gelombang
1	surya	8.78	1400000	D3 Manajemen Informatika	S1 Teknik Informatika	D3 Manajemen Informatika	Lulus Tes Wawancara	Gelombang Khusus
2	aryanto	8.50	1400000	S1 Sistem Informasi	S1 Teknik Informatika	---	Tes Wawancara	Gelombang I

Gambar 5.17 Tampilan Halaman Tes Wawancara


3. Menu konfigurasi PMB

Halaman manajemen pmb_config (konfigurasi PMB) ini diimplementasikan oleh file list.php. Halaman ini berupa form untuk mengubah gelombang PMB dan tahun ajarannya. Tampilan dari halaman list.php seperti gambar 5.18.

.. Halaman Konfigurasi ..

Konfigurasi PMB

Selamat datang "amikom", anda login sebagai "Super Admin"
 >> Berikut ini adalah informasi mengenai PMB yang ada di web ini.

Gelombang PMB : 

Tahun :

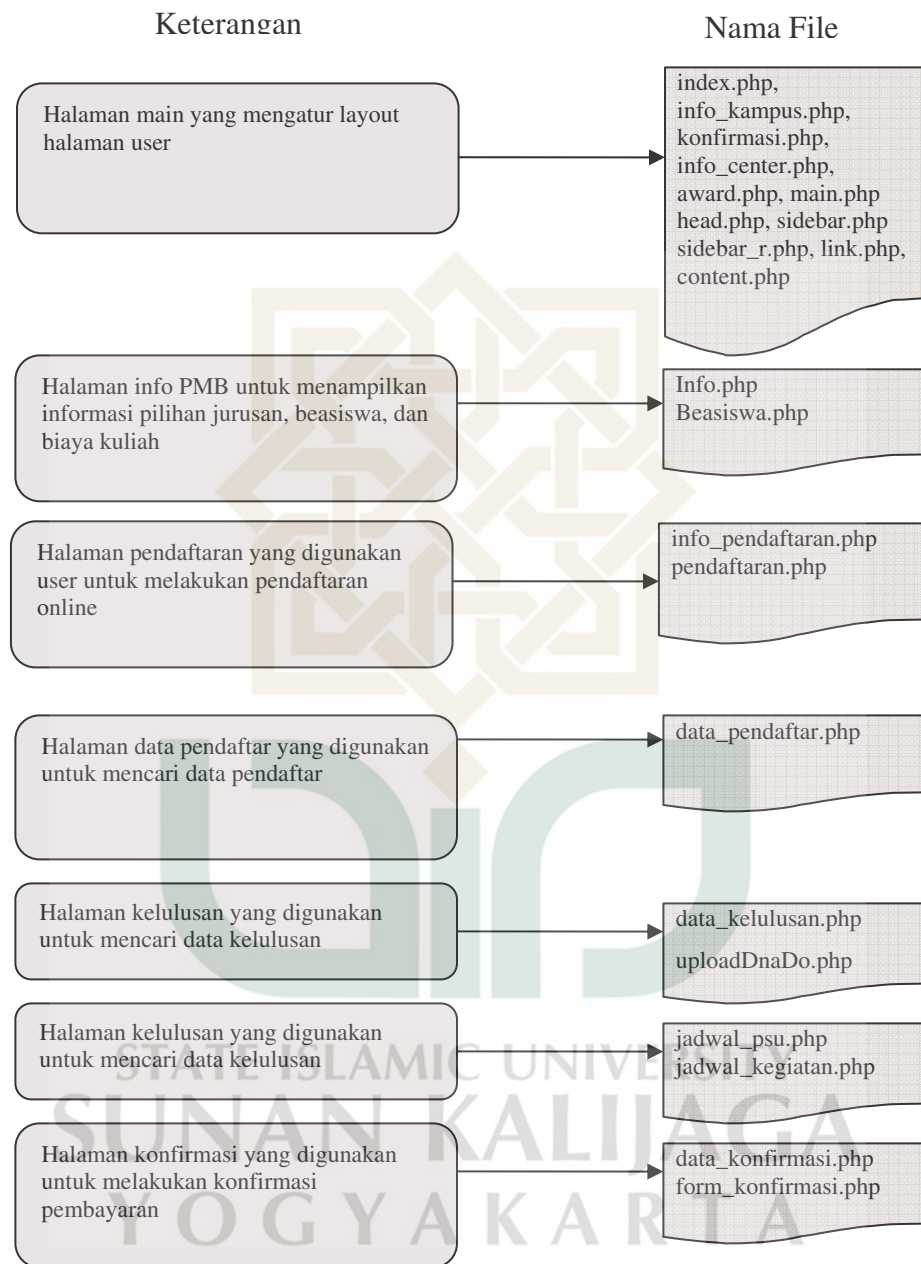
Gambar 5.18 Tampilan Form Konfigurasi PMB

Setelah admin mengubah gelombang PMB dan menekan tombol ubah, maka *query* update data akan dijalankan dan otomatis halaman akan *refresh* menuju halaman list.php (di folder admin) seperti dijelaskan pada potongan script berikut:

```
if($mysql->execute("Update pmb_config set g_id='$gelombang',  
pm_tahun='$tahun'")){  
echo"<meta http-equiv='refresh'  
content='1;URL=main.php?page=1'>";  
}
```

5.1.7 Implementasi Modul User Berbasis Web

Modul user yang diakses melalui web digunakan untuk menjalankan proses-proses yang dilakukan oleh user terhadap sistem. Susunan file-file PHP pada modul user berbasis web dapat dilihat pada gambar 5.19.



Gambar 5.19 Susunan File PHP pada Modul User Berbasis Web

5.1.7.1 Implementasi Halaman Utama

Halaman utama modul user berbasis web ini diimplementasikan oleh file index.php. Halaman ini meng-include file head.php untuk menampilkan header,

sidebar untuk menampilkan menu di layout sebelah kiri, content.php untuk menampilkan konten di layout tengah, dan sidebar_r.php untuk menampilkan link di layout sebelah kanan. Untuk menjalankan halaman utama modul administrator dapat dijalankan dengan mengetikkan <http://localhost/program/index.php> pada *address bar* sehingga muncul halaman utama seperti gambar 5.20.



Gambar 5.20 Tampilan Halaman Utama Berbasis Web

Halaman utama tersebut juga disertai *style* CSS untuk memperindah tampilan website. *Style* CSS tersebut berada di file tersendiri yaitu default.css dan dipanggil menggunakan perintah berikut:

```
<link href="default.css" rel="stylesheet" type="text/css"
media="screen">
```


Sedangkan untuk mengatur layout halaman nya menjadi 4 area adalah menggunakan div id dan dibungkus menggunakan wrapper seperti dijelaskan pada potongan *script* berikut:

```
<div id="wrapper">
<? include('head.php'); ?>
<div id="page">
<div id="sidebar">
<? include('sidebar.php');?>
</div>
<div id="content">
<? include('content.php');?>
</div>
<div id="sidebar_r">
<? include('sidebar_r.php');?>
</div>
```

Untuk halaman yang akan dibahas adalah halaman yang terkait dengan proses-prosesnya saja, seperti yang sudah digambarkan dalam DFD sebelumnya, yaitu proses pendaftaran dan konfirmasi pembayaran.

5.1.7.2 Implementasi Proses Pendaftaran

Proses pendaftaran ini merupakan proses inti dari sistem penerimaan mahasiswa baru. Proses ini terdiri dari 2 menu yaitu pendaftaran, dan data pendaftaran. Halaman ini diimplementasikan oleh file pendaftaran.php, dan data_pendaftar.php. Kode Sumber (*source code*) proses pendaftaran dapat dilihat pada lampiran E.

1. Menu Pendaftaran

Menu ini digunakan user untuk melakukan pendaftaran *online* melalui WEB. Tampilan dari halaman pendaftaran.php sebenarnya sama dengan file pendaftaran yang ada di modul admin sebelumnya. Perbedaannya hanyalah di field tempat pendaftaran. Kalau pendaftaran dilakukan langsung oleh user maka secara otomatis tempat pendaftaran akan terisi "Via Online". Tampilannya seperti gambar 5.21.

The screenshot shows a registration form with the following fields and options:

- Pilihan 1 ***: Pilih Salah Satu (dropdown menu)
- Pilihan 2**: Pilih Salah Satu (dropdown menu)
- Informasi Pertama Kali dari**:
 - Internet
 - TV / Radio
 - Lainnya
 - Brosur / Koran
 - Teman / Saudara / Guru
- Tempat Pendaftaran**: Via Online

Below the form, there are two buttons: **Daftar** and **Batal**. A note section contains the following text:

- Catatan :
- Form bertanda * harus diisi.
- Tag HTML tidak diijinkan.

Gambar 5.21 Tampilan Halaman Pendaftaran Berbasis Web

Apabila form tersebut tidak diisi, maka akan muncul sebuah pesan kesalahan yang dibuat menggunakan JQuery seperti yang sudah dibahas sebelumnya. Contoh tampilan kesalahannya pun sudah dibahas sebelumnya di modul admin.

Setelah form tersebut diisi lengkap dan user menekan tombol daftar, maka akan muncul pesan bahwa data bisa disimpan. Prosesnya pertama-tama data dikirim ke masing-masing variabel menggunakan perintah `$_POST`, seperti pada potongan *script* berikut:

```

if(isset($_POST['daftar'])) {
    $gelombang=$dataPMB[0][0];
    $nama = $_POST['nama'];
    $k_id = $_POST['k_id'];
    $no_id = $_POST['no_id'];
    ...
}

```

Setelah itu, data pendaftaran tersebut dimasukkan ke database menggunakan *query* insert terhadap tabel pendaftaran seperti yang sudah dibahas sebelumnya.

Sedangkan untuk menentukan besarnya beasiswanya berdasarkan NEM mahasiswa tersebut adalah menggunakan *script* berikut ini:

```

if ($nem1 < 7) {
    $status=1; }
else {$status=3;}

```

Script tersebut berguna untuk set status mahasiswa pendaftar yang memiliki NEM < 7 menjadi tes tertulis, sedangkan yang memiliki NEM >= 7 statusnya diisi menjadi tes wawancara. Kemudian *script* berikut ini:

```

if($mysql->execute("select b_id, b_nama, b_max, b_min from
beasiswa"))
{
    $dataB = $mysql->getDataSet();
}
for($i=0;$i<count($dataB);$i++){
    $max=$dataB[$i][2];
    $min=$dataB[$i][3];
    if(($nem1>=$min) && ($nem1<=$max)){
        $beasiswa=$dataB[$i][0];
    }
}

```

Script diatas berfungsi untuk menentukan besarnya beasiswa. Maksud dari *script* tersebut adalah.pertama jalankan *query* select data dari tabel beasiswa, kemudian menyimpannya dalam sebuah *array*. Lalu ada script looping for dari $\$i=0$, selama $\$i$ masih lebih kecil dari jumlah data yang ada, tambahkan terus $\$i$ nya, dan lakukan perintah selanjutnya. Ambil data *array* di field 3 yaitu *b_max*, masukkan ke dalam variabel $\$max$, dan juga data *array* di field 3 yaitu *b_min*, kemudian masukkan ke dalam variabel $\$min$. Jika $\$nem1$ (NEM yang dimasukkan mahasiswa) $\geq \$min$ dan $\leq \$max$, maka ambil data *array* di field 0 (yaitu *b_id*) yang sesuai dengan i yang sudah ditemukan.

2. Menu Data Pendaftar

Halaman data pendaftar ini berguna untuk menampilkan data mahasiswa yang sudah melakukan pendaftaran. Data ini akan ditampilkan dalam bentuk tabel yang datanya diambil dari tabel pendaftaran yang sudah ada dalam *database*. Data mahasiswa ini akan diurutkan sesuai nomor pendaftarannya, bukan sesuai urutan abjad namanya.

Di halaman ini juga ada *link* gelombang I, gelombang II, dan seterusnya. *Link* ini berguna untuk menampilkan data pendaftar sesuai gelombang masing-masing. Mahasiswa nantinya juga bisa melakukan pencarian data pendaftar menggunakan menu pencarian yang terletak di atas tabel. Pencarian dilakukan berdasarkan nama atau nomor pendaftaran. Tampilan halaman data pendaftarnya seperti gambar 5.22.

Data Pendaftar STMIK AMIKOM Yogyakarta Tahun Akademik 2010/2011								
Gelombang Khusus Gelombang I Gelombang II Gelombang III								
Pencarian				<input type="text"/>	Cari...			
No.	No. Daftar	Nama	NEM	Gelombang	Pil-1	Pil-2	Syarat	Tgl.Daftar
1.	32	surya	8.78	Gelombang Khusus	D3 Manajemen Informatika	S1 Teknik Informatika	Lengkap	2010-07-03 13:42:45
2.	34	anton	6.50	Gelombang Khusus	D3 Teknik Informatika	S1 Teknik Informatika	Lengkap	2010-07-04 14:45:12
3.	53	lintang	6.50	Gelombang I	S1 Sistem Informasi	S1 Teknik Informatika	Lengkap	2010-07-19 21:49:56

Gambar 5.22 Tampilan Halaman Data Pendaftar Berbasis Web

Ketika *link* gelombang di halaman tersebut di klik, maka akan menuju halaman `data_pendaftar.php` itu sendiri, namun dengan membawa parameter sesuai *script* berikut:

```
<a href="main.php?page=5&gel=1">Gelombang Khusus</a>
```

Script tersebut membawa parameter `gel=1`. Parameter ini kemudian ditangkap menggunakan menggunakan *script* `$gel = $_GET['gel'];`. Sehingga halaman tersebut akan menampilkan data mahasiswa pendaftar gelombang khusus saja.

Sedangkan untuk form pencarian data pendaftar, input name dari textfield tersebut terdapat pada potongan *script* berikut:

```
<td><strong>Pencarian</strong></td>
<td><input name="cari_nama" size="40" type="text" />
```

Dari *script* tersebut, diketahui bahwa setiap inputan yang dilakukan di form tersebut akan dibaca sebagai `cari_nama`. Untuk memasukkan ke dalam variabel digunakan *script* berikut:

```
$cari_nama = $_POST['cari_nama'];
```

Setelah dimasukkan ke dalam variabel, maka *script* akan menjalankan *query* untuk mencari data yang dimaksud. *Query* tersebut adalah sebagai berikut:

```
$mysql->execute("SELECT a.p_id, a.p_tgl, a.p_nama, a.p_nem,
a.ju_id, a.ju_id2, j.s_nama, k.g_nama FROM pendaftaran a, syarat
j, gelombang k where a.s_id=j.s_id and a.g_id=k.g_id and
((a.p_nama LIKE '%$cari_nama%') or a.p_id='$cari_nama') order by
a.p_id")
```

Dari perintah tersebut digunakan kondisi `a.p_nama LIKE '%$cari_nama%' or a.p_id='$cari_nama'`. Hal ini berarti bahwa apabila terdapat kata yang disimpan dalam variabel `$cari_nama` berada dalam field `p_nama`(nama mahasiswa) atau `p_id`(nomor pendaftaran), maka data tersebut akan diambil. Karakter `'%'` dimaksudkan untuk mengabaikan huruf kapital, sehingga huruf kapital dianggap sebagai huruf biasa.

5.1.7.3 Implementasi Proses Seleksi Tes

Proses seleksi tes di modul user berbasis web ini merupakan proses untuk menampilkan hasil seleksi tes yang sebelumnya sudah dikelola oleh admin. Proses ini terdiri dari menu kelulusan. Halaman ini diimplementasikan oleh file `data_kelulusan.php`. Kode Sumber (*source code*) proses seleksi tes dapat dilihat pada lampiran F.

1. Menu Kelulusan

Menu kelulusan ini sebenarnya konsepnya sama dengan menu data pendaftar sebelumnya. Yang berbeda hanyalah terletak pada field yang ditampilkan saja. Di

menu kelulusan ini, data yang ditampilkan hanya 4 field saja dan ditampilkan dalam bentuk tabel yang datanya diambil dari tabel pendaftaran.

Di halaman ini juga ada *link* hasil tes tertulis dan hasil tes wawancara. *Link* hasil tes tertulis ini berguna untuk menampilkan data mahasiswa yang memiliki $NEM < 7$ dan sudah lulus tes tertulis. Sedangkan *link* hasil tes wawancara ini berguna untuk menampilkan data mahasiswa yang memiliki $NEM \geq 7$ dan sudah lulus tes wawancara. Jadi, mahasiswa yang memiliki $NEM < 7$ tidak perlu ikut tes wawancara, sedangkan mahasiswa yang memiliki $NEM \geq 7$ tidak perlu ikut tes tertulis. Mahasiswa nantinya juga bisa melakukan pencarian data kelulusan menggunakan menu pencarian yang terletak di atas tabel. Pencarian dilakukan berdasarkan nama atau nomor pendaftaran. Tampilan halaman kelulusannya seperti gambar 5.23.



Gambar 5.23 Tampilan Halaman Kelulusan Berbasis Web

Ketika link hasil test tertulis di halaman tersebut di klik, maka akan program akan mengeksekusi halamannya sendiri, namun dengan membawa parameter sesuai *script* berikut:

```
<a href="main.php?page=6&tes=1">[Hasil Test Tertulis]</a>
```


Script tersebut membawa parameter tes=1. Parameter ini kemudian ditangkap menggunakan menggunakan *script* `$tes = $_GET['tes'];`. Kemudian ada perintah if tes=1, maka *query* yang akan dijalankan adalah *query* berikut:

```

if($tes==1){
    if($mysql->execute("SELECT a.p_id, a.p_nama, a.p_no_id,
a.ju_fix, a.ju_id2, l.b_nama, m.st_nama FROM pendaftaran a,
gelombang k, beasiswa l, status m where a.g_id=k.g_id and
a.b_id=l.b_id and a.st_id=m.st_id and a.b_id=l.b_id and
((a.p_nama LIKE '%$cari_nama%') or a.p_id='$cari_nama') and
(a.st_id=1 or a.st_id=2 or a.st_id=3) and s_id=2 order by
a.p_id")){
        $data = $mysql->getDataSet();}
    } else if($tes==2){
        if($mysql->execute("SELECT a.p_id, a.p_nama, a.p_no_id,
a.ju_fix, a.ju_id2, l.b_nama, m.st_nama FROM pendaftaran a,
gelombang k, beasiswa l, status m where a.g_id=k.g_id and
a.b_id=l.b_id and a.st_id=m.st_id and a.b_id=l.b_id and
((a.p_nama LIKE '%$cari_nama%') or a.p_id='$cari_nama') and
(a.st_id=4 or a.st_id=5 or a.st_id=6) and s_id=2 order by
a.p_id")){
            $data = $mysql->getDataSet();}
        }
    }

```

Query tersebut bertugas melakukan select data yang nama mahasiswa atau nomor pendaftarannya sama dengan karakter yang diinput dan status tes nya tes tertulis atau tidak lulus tes tertulis atau lulus tes tertulis. Sedangkan jika tes=2 maka akan menampilkan data mahasiswa yang status tesnya tes wawancara atau tidak lulus tes wawancara atau lulus tes wawancara. Contoh tampilan hasil pencarian kelulusan setelah *link* tes tertulis di klik adalah seperti gambar 5.24 dan untuk *link* tes wawancara seperti gambar 5.25.

Hasil Test Tertulis dan Wawancara
Tahun Akademik 2010/2011
 [Hasil Test Tertulis] - [Hasil Test Wawancara]

Pencarian Data Kelulusan (Masukkan nama atau nomor pendaftaran)

Pencarian Cari...

No.	No. Daftar	Nama	Diterima di	Beasiswa	Keterangan
1.	34	anton	S1 Teknik Informatika	700000	Lulus Tes Tertulis
2.	53	lintang	S1 Sistem Informasi	700000	Lulus Tes Tertulis
3.	61	sidiq	---	0	Tes Tertulis

Gambar 5.24 Tampilan Halaman Kelulusan Tes Tertulis

Hasil Test Tertulis dan Wawancara
Tahun Akademik 2010/2011
 [Hasil Test Tertulis] - [Hasil Test Wawancara]

Pencarian Data Kelulusan (Masukkan nama atau nomor pendaftaran)

Pencarian Cari...

No.	No. Daftar	Nama	Diterima di	Beasiswa	Keterangan
1.	32	surya	D3 Manajemen Informatika	1400000	Lulus Tes Wawancara
2.	58	aryanto	---	1400000	Tes Wawancara

Gambar 5.25 Tampilan Halaman Kelulusan Tes Wawancara

Untuk status tes nya ada 6 yaitu tes tertulis, tidak lulus tes tertulis, lulus tes tertulis, tes wawancara, dan tidak lulus tes wawancara, serta lulus tes wawancara. Dalam tampilan hasil tes diatas, aryanto belum dinyatakan diterima di jurusan apapun (masih kosong) karena aryanto memang sedang tes wawancara atau seharusnya sudah tes wawancara namun tidak hadir.

5.1.7.4 Implementasi Proses Konfirmasi Pembayaran

Proses ini diimplementasikan oleh program tambah konfirmasi (form_konfirmasi.php). Proses konfirmasi ini berguna untuk melakukan konfirmasi pembayaran setelah mahasiswa melakukan pendaftaran *online* dan transfer uang

pendaftaran. Transfer uang pendaftaran bisa dilakukan lewat ATM atau *teller* ke nomor rekening panitia pelaksana PMB. Kode Sumber (*source code*) proses konfirmasi pembayaran dapat dilihat pada lampiran G.

Komponen form konfirmasi pembayaran tersebut terdiri dari tanggal transfer, jam transfer, nomor rekening yang digunakan untuk transfer. Komponen form ini harus diisi secara lengkap oleh mahasiswa. Tampilan form konfirmasinya seperti gambar 5.26.

Jika data di bawah ini benar data Anda, silahkan lanjutkan dengan mengisi form konfirmasi

Form Konfirmasi

No Pendaftaran	:	47
Nama	:	aryanto
Nem	:	8.00
Gelombang	:	Gelombang I
Pilihan 1	:	51 Sistem Informasi
Pilihan 2	:	51 Teknik Informatika
Tanggal	:	<input type="text"/> * (dd/mm/yyyy) cth.(20/02/2010)
Jam	:	<input type="text"/> cth. 14:00
No Rekening	:	<input type="text"/>

Gambar 5.26 Tampilan Form Konfirmasi

Setelah form tersebut diisi lengkap dan user menekan tombol konfirmasi, maka akan muncul pesan bahwa konfirmasi telah diterima. Prosesnya pertama-tama data dikirim ke masing-masing variabel menggunakan perintah `$_POST`, seperti pada potongan *script* berikut:

```

if(isset($_POST['konfirmasi'])){
    $tgl = $_POST['tgl'];
    $jam = $_POST['jam'];
    $rek = $_POST['rek'];
    ...
}

```

Setelah itu, data konfirmasi tersebut dimasukkan ke tabel pendaftaran dalam *database*

menggunakan *query* update terhadap tabel pendaftaran seperti berikut:

```

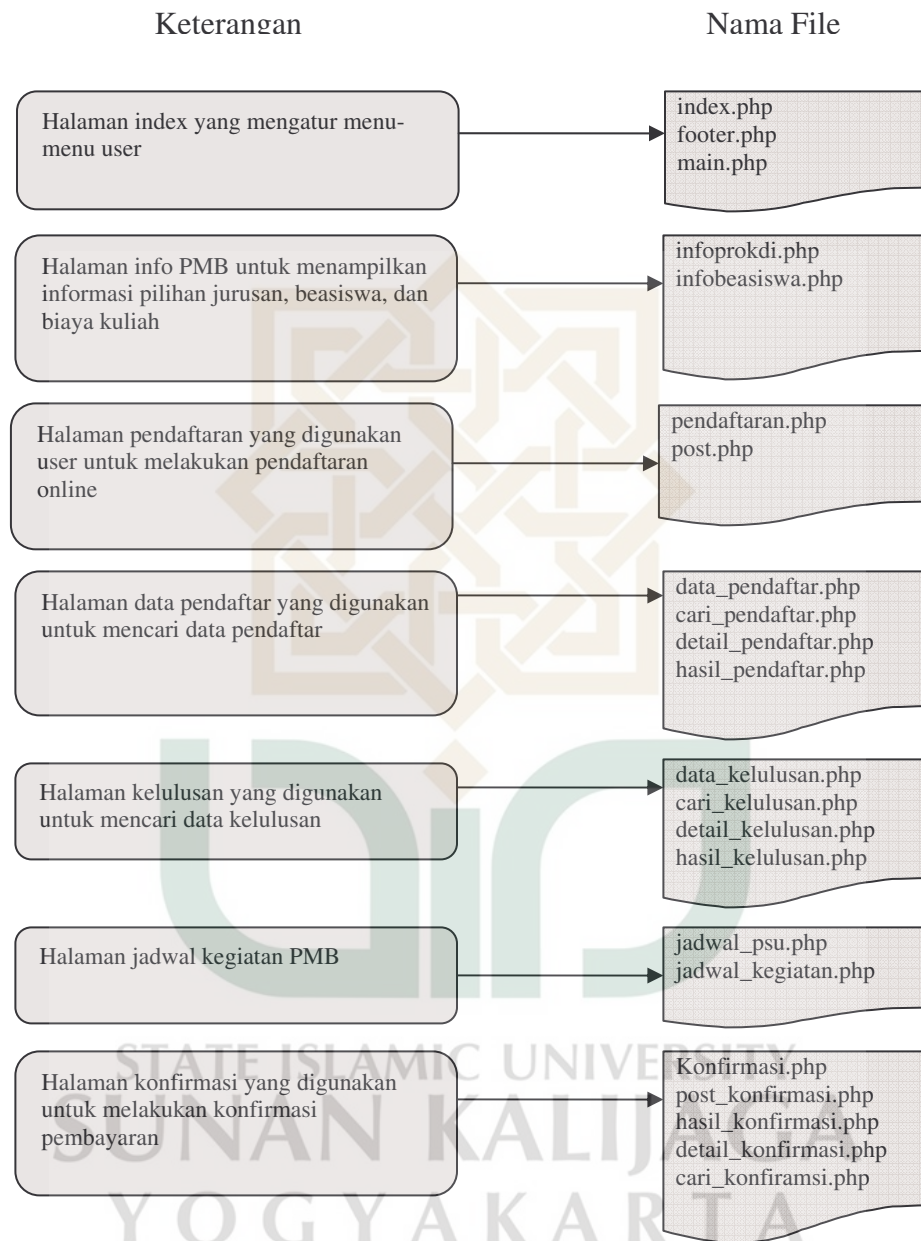
if($mysql->execute("UPDATE pendaftaran set ko_tgl='$tgl',
ko_jam='$jam', ko_no_rek='$rek' where p_id='$id'"))
{
    echo"<center>Konfirmasi anda telah kami terima, pantau terus
    untuk informasi pendaftaran di web ini.";
    echo"<meta http-equiv='refresh'
    content='2;URL=main.php?page=10'>";
    exit;
}

```

Maksud dari *query* update tersebut adalah untuk mengubah nilai field `ko_tgl`, `ko_jam`, dan `ko_no_rek` yang ada di tabel pendaftaran. Misal data tanggal dimasukkan ke dalam variabel `$tgl`, maka kemudian nilai dalam variabel tersebut dimasukkan ke dalam field `ko_tgl` yang sesuai dengan `p_id` mahasiswa yang melakukan konfirmasi. Begitu juga untuk field yang lainnya. Jika konfirmasi berhasil maka akan muncul pesan sukses dan secara otomatis halaman akan *refresh* ke halaman `data_konfirmasi.php`. Sedangkan untuk pengecekan konfirmasinya dilakukan oleh admin seperti yang sudah dijelaskan sebelumnya di menu cek pendaftaran halaman admin.

5.1.8 Implementasi Modul User Berbasis WAP

Modul user berbasis WAP diakses menggunakan *handphone* (dalam penelitian ini menggunakan *WinWAP simulator*). Proses-proses yang bisa dilakukan oleh user terhadap sistem berbasis WAP ini sebenarnya sama dengan ketika user mengakses melalui web. Perbedaannya hanyalah pada bahasa yang digunakan untuk pemrogramannya. Susunan file-file PHP pada modul user berbasis WAP dapat dilihat pada gambar 5.27.



Gambar 5.27 Susunan File PHP pada Modul User Berbasis WAP

5.1.8.1 Implementasi Halaman Utama

Halaman utama modul user berbasis WAP ini diimplementasikan oleh file `index.php` yang juga *include* `footer.php`. Halaman ini akan menampilkan menu-menu

yang bisa diakses oleh user. Untuk menjalankan halaman utama ini dapat dijalankan dengan mengetikkan <http://localhost/program/wap/index.php> pada *address bar* WinWAP *Browser* sehingga muncul halaman utama seperti gambar 5.28.



Gambar 5.28 Tampilan Halaman Utama Berbasis WAP

Setiap menu yang ada di halaman ini menuju ke halaman main.php semuanya.

Yang berbeda hanyalah pada parameter *page* yang dibawa oleh setiap menu. Di halaman index.php dan main.php ini juga terdapat *script* berikut:


```

<?php
header("Content-type: text/vnd.wap.wml\n");
echo("<?xml version=\"1.0\"?>\n");
echo("<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\"
\"http://www.wapforum.org/DTD/wml_1.1.xml\">\n");
?>

```

Script tersebut berguna untuk mengenalkan file WML dan menunjukkan bahwa dokumen WML merupakan bagian dari dokumen XML. Jika *script* tersebut tidak ada maka program tidak akan bisa dijalankan di WinWAP *Browser*.

Untuk halaman yang akan dibahas adalah halaman yang terkait dengan proses-prosesnya saja, seperti yang sudah digambarkan dalam DFD sebelumnya, yaitu proses manajemen data pendaftaran dan manajemen konfirmasi.

Hal yang penting lagi adalah tentang koneksi ke *database*. Untuk sistem berbasis WAP ini koneksinya dibuat secara langsung di file program masing-masing. *Script* koneksinya adalah seperti berikut:

```

<?php
$db = mysql_connect($host,$user,$pass);
mysql_select_db($database,$db) or die ("Query 1 failed"); ?>

```

Variabel yang ada di *script* tersebut sudah di definisikan di file terpisah yaitu `config.inc.php` yang juga sudah *include* di halaman `main.php`.

5.1.8.2 Implementasi Proses Pendaftaran

Penjelasan proses ini sudah pernah dibahas sebelumnya. Kode Sumber (*source code*) proses pendaftaran dapat dilihat pada lampiran G.

1. Menu Pendaftaran

Menu ini digunakan user untuk melakukan pendaftaran online melalui *handphone*. Tampilan dari halaman pendaftaran ini dibagi menjadi 2 bagian yaitu pendaftaran.php dan pendaftaran2.php. Di halaman pendaftaran.php, mahasiswa diminta untuk memilih kota asal, setelah itu klik next, sehingga akan muncul form pendaftaran secara lengkap. Tampilannya seperti gambar 5.29.

Pendaftaran Calon Mahasiswa STMIK "AMIKOM"
Yogyakarta
Tahun Akademik: 2010/2011

Propinsi * :
- Pilih Propinsi -

Catatan :
▪ Pilih Propinsi Anda Berada.

[next](#)

[home](#) | [pendaftaran](#) | [data pendaftar](#) | [data kelulusan](#)
| [konfirmasi pembayaran](#)

Kampus Terpadu AMIKOM
Jl. Ring Road Utara, Condong Catur, Sleman,
Yogyakarta - Indonesia
Telp: (0274) 884201 - 207, Fax: (0274) 884208
E-Mail: amikom@amikom.ac.id
Kodepos: 55283

Nama Lengkap * :

Nomor Identitas (KTP/SIM/Paspor) * :

Jenis Kelamin * :
- Pilih Menu -

Agama * :
- Pilih Agama -

Asal Sekolah * :

Jurusan SMU * :
- Jurusan SMU -

Rata-rata NEM * :
 [contoh 8.55]

Alamat :

Kewarganegaraan * :
- Pilih -

Gambar 5.29 Tampilan Halaman Pendaftaran Berbasis WAP

Apabila form tersebut tidak diisi, maka akan muncul sebuah pesan kesalahan bahwa data harus diisi. Karena WAP tidak support java *script*, sehingga tidak bisa menggunakan *alert*, maka pesan kesalahannya pun cukup sederhana. Setelah form tersebut diisi lengkap dan user menekan tombol daftar, maka halaman yang dituju adalah post.php. dan akan muncul pesan bahwa data bisa disimpan. Prosesnya

pertama-tama data dikirim ke masing-masing variabel menggunakan perintah `$_POST`, seperti pada potongan *script* berikut:

```
if (isset($_GET['type']) == "post")
{
    $nama = $_POST['nama'];
    $id = $_POST['id'];
```

Setelah itu, data pendaftaran tersebut dimasukkan ke database menggunakan *query insert* terhadap tabel pendaftaran seperti berikut:

```
$sql="INSERT INTO $table (p_nama, p_no_id, j_id, a_id, p_asal,
js_id, p_nem, p_alamat, w_id, p_id, k_id, p_kodepos, p_tlp,
p_email, ju_id1, ju_id2, i_id, p_tgl) VALUES ('$nama', '$id',
'$jenis', '$agama', '$asal', '$jurusan', '$nem', '$alamat',
'$warga', '$propinsi', '$kabupaten', '$kode', '$tlp', '$email',
'$pill1', '$pill2', '$info', NOW());"
```

Script SQL diatas sebenarnya sama fungsinya dengan yang sudah dibahas sebelumnya di modul berbasis web.

2. Menu Data Pendaftar

Penjelasan tentang menu ini sudah pernah dibahas sebelumnya di modul user berbasis web. Tampilan halaman data pendaftar via WAP adalah seperti gambar 5.30.

Data Pendaftar Calon Mahasiswa STMIK "AMIKOM" Yogyakarta
Tahun Akademik: 2010/2011

[Gelombang Khusus](#) | [Gelombang I](#) | [Gelombang II](#)
[Gelombang III](#)

Pencarian Data Kelulusan
(Masukkan nama atau nomor pendaftaran)

Pencarian [Cari](#)

[home](#) | [pendaftaran](#) | [data pendaftar](#) | [data kelulusan](#) | [konfirmasi pembayaran](#)

Kampus Terpadu AMIKOM
Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta - Indonesia
Telp: (0274) 884201 - 207, Fax: (0274) 884208
E-Mail: amikom@amikom.ac.id
Kodepos: 55283

Gambar 5.30 Tampilan Halaman Data Pendaftar Berbasis WAP

Ketika *link* gelombang di halaman tersebut di klik, maka akan menuju halaman `data_pendaftar.php` itu sendiri, namun dengan membawa parameter sesuai *script* berikut:

```
<a href="main.php?page=6&id=1">Gelombang Khusus</a>
```

Script tersebut membawa parameter `gel=1`. Parameter ini kemudian ditangkap menggunakan menggunakan *script* `$gel = $_GET['gel'];` .Sehingga halaman tersebut akan menampilkan data mahasiswa pendaftar gelombang khusus saja. Kunci *query* agar bisa menampilkan data pendaftar sesuai gelombang yang dipilih adalah menggunakan kondisi berikut:

```
...where a.g_id=k.g_id ... and a.g_id=$id order by a.p_id ...
```

Tampilan data pendaftar gelombang khusus adalah seperti gambar 5.31 berikut.

Data Pendaftar Calon Mahasiswa STMIK "AMIKOM" Yogyakarta Tahun Akademik: 2010/2011		
No Pendaftaran	Nama	Asal Sekolah
32	surya	SMK Negeri 1 Madiun
34	anton	SMA 2 Yogyakarta
home pendaftaran data pendaftar data kelulusan konfirmasi pembayaran		

Gambar 5.31 Tampilan Halaman Data Pendaftar Gelombang Khusus

Sedangkan untuk form pencarian data pendaftar, input name dari textfield tersebut terdapat pada potongan *script* berikut:

```
<input name="nama" /> <anchor>Cari<go  
href="main.php?page=7&id=0" method="post">" ;?>  
<postfield name="nama" value="$(nama)" />
```

```
</go></anchor>
```

Dari *script* tersebut, diketahui bahwa setiap inputan yang dilakukan di form tersebut akan dibaca sebagai nama. Perbedaannya dari yang form di halaman web adalah bahwa untuk mengeksekusi form disini menggunakan perintah `<anchor>Cari<go href="main.php?page=7&id=0"` . Untuk memasukkan ke dalam variabel digunakan *script* berikut (yang ada di file post.php):

```
$cari_nama = $_POST['nama'];
```

Setelah dimasukkan ke dalam variabel, maka *script* akan menjalankan *query* untuk mencari data yang dimaksud. *Query* tersebut adalah sebagai berikut:

```
if(!($id==0)){
    $data = mysql_query("SELECT a.p_id, a.p_nama, a.p_asal, a.g_id FROM
    pendaftaran a, gelombang k where a.g_id=k.g_id and ((a.p_nama LIKE
    '%$cari_nama%') or a.p_id='$cari_nama') and a.g_id=$id order by
    a.p_id") or die(mysql_error());
    $rows = mysql_num_rows($data);
} else {
    $data = mysql_query("SELECT a.p_id, a.p_nama, a.p_asal, a.g_id FROM
    pendaftaran a, gelombang k where a.g_id=k.g_id and ((a.p_nama LIKE
    '%$cari_nama%') or a.p_id='$cari_nama') order by a.p_id") or
    die(mysql_error());
    $rows = mysql_num_rows($data);
}
```

Query tersebut berguna untuk mencari data sesuai inputan mahasiswa. Jadi di dalam *query* ini ada 2 pengkondisian. Jika yang diinputkan oleh mahasiswa adalah nomor pendaftaran, maka *query* akan mencari data yang nomor pendaftarannya sama dengan yang diinputkan mahasiswa ke dalam form sebelumnya.

Atau, jika yang diinputkan oleh mahasiswa berupa nama, maka *query* akan mencari data dalam tabel yang namanya sama dengan nama inputan mahasiswa sewaktu berada dalam halaman form pencarian.

Query tersebut pada intinya bertugas untuk melakukan select data nomor pendaftaran, nama mahasiswa, dan asal sekolah yang ada dalam tabel pendaftaran dan tabel gelombang. Data yang ditampilkan tentunya yang nama mahasiswa atau nomor pendaftarannya sama dengan yang di inputkan oleh mahasiswa ke dalam form pencarian. Hal ini terdapat pada potongan *script* berikut:

```
where ... ((a.p_nama LIKE '%$cari_nama%' ) or a.p_id='$cari_nama')
and ...
```

\$cari_nama ini merupakan variable yang isinya nilai inputan mahasiswa sewaktu berada di form pencarian.

Sedangkan untuk menampilkan detail pendaftarannya (*detail_pendaftar.php*), *query* select nya pun sama dengan yang sudah dibahas sebelumnya. Data detail pendaftaran ini muncul setelah user melakukan klik *link* nama mahasiswa yang sebelumnya ditampilkan. Yang perlu diperhatikan adalah *script* untuk menampilkan data ke dalam tabelnya, yaitu terdapat pada potongan *script* berikut:

```
while ($line = mysql_fetch_object($data))
{
    print "<table columns=\"2\">";
    print "<tr>";
    print "<td>No Pendaftaran</td>";
    //print "<td>:</td>";
    print "<td>";
    print "$line->p_id";
    print "</td></tr>";
}
```

```

print "<tr>";
print "<td>Tanggal Pendaftaran</td>";
//print "<td>:</td>";
print "<td>";
print "$line->p_tgl";
print "</td></tr>";
print "<tr>";
print "<td>Nama</td>";
//print "<td>:</td>";
print "<td>";
print "$line->p_nama";
print "</td></tr>";
...

```

Jadi, untuk menampilkan datanya disini menggunakan perintah `$line->`. Ketika *query* dijalankan dan ketemu hasilnya, maka baris dari data yang ditemukan tersebut dijadikan objek dan dimasukkan ke variabel baru yaitu `$line`. Tampilan data detail pendaftarannya adalah seperti gambar 5.32.

Data Pendaftar Calon Mahasiswa STMIK "AMIKOM" Yogyakarta Tahun Akademik: 2010/2011	
No Pendaftaran	34
Tanggal Pendaftaran	2010-07-04 14:45:12
Nama	anton
Pilihan-1	D3 Teknik Informatika
Pilihan-2	S1 Teknik Informatika
Syarat	Lengkap
Gelombang	Gelombang Khusus

Gambar 5.32 Tampilan Halaman Data Detail Pendaftar Berbasis WAP

5.1.8.3 Implementasi Proses Seleksi Tes

Penjelasan tentang proses ini sudah dijelaskan sebelumnya di modul user berbasis web. Kode Sumber (*source code*) proses seleksi tes dapat dilihat pada lampiran H.

1. Menu Data Kelulusan

Penjelasan tentang menu ini sudah pernah dibahas sebelumnya di modul user berbasis WEB. Tampilan halaman data kelulusan berbasis WAP adalah seperti gambar 5.33.

Pencarian Data Kelulusan
(Masukkan nama atau nomor pendaftaran)

Pencarian [Cari](#)

[home](#) | [pendaftaran](#) | [data.pendaftar](#) | [data.kelulusan](#) | [konfirmasi](#)
[pembayaran](#)

Kampus Terpadu AMIKOM
 Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta - Indonesia
 Telp: (0274) 884201 - 207, Fax: (0274) 884208
 E-Mail: amikom@amikom.ac.id
 Kodepos: 55283

Gambar 5.33 Tampilan Halaman Kelulusan Berbasis WAP

Jika dilihat, memang sama dengan form pencarian data pendaftar sebelumnya. Hasil pencarian dan *query* untuk select datanya pun juga hampir sama. Perbedaannya adalah field yang ditampilkan, di menu ini, field yang ditampilkan adalah nomor pendaftaran, nama mahasiswa, dan status tes. Gambar 5.34 dan 5.35 berikut merupakan tampilan hasil tes tertulis dan hasil tes wawancara.

Hasil Tes Tertulis dan Wawancara		
Calon Mahasiswa Baru STMIK AMIKOM Yogyakarta		
Tahun Akademik : 2010/2011		
No Pendaftaran	Nama	Status
34	anton	Lulus Tes Tertulis
53	lintang	Lulus Tes Tertulis
61	wijaya	Tes Tertulis
home pendaftaran data pendaftar data kelulusan konfirmasi pembayaran		

Gambar 5.34 Tampilan Hasil Tes Tertulis Berbasis WAP

Hasil Tes Tertulis dan Wawancara		
Calon Mahasiswa Baru STMIK AMIKOM Yogyakarta		
Tahun Akademik : 2010/2011		
No Pendaftaran	Nama	Status
32	surya	Lulus Tes Wawancara
58	aryanto	Tes Wawancara
home pendaftaran data pendaftar data kelulusan konfirmasi pembayaran		

Gambar 5.35 Tampilan Hasil Tes Wawancara Berbasis WAP

Script untuk menampilkan datanya pun sama dengan yang sudah dibahas sebelumnya. Kemudian data nama mahasiswa tersebut sekaligus sebagai *link* yang digunakan untuk menampilkan data mahasiswa secara detail (`detail_kelulusan.php`). Program `detail_kelulusan.php` ini logikanya sama dengan program `detail_pendaftar.php`

5.1.8.4 Implementasi Proses Konfirmasi Pembayaran

Penjelasan tentang proses ini sudah dijelaskan sebelumnya di modul berbasis web. Kode Sumber (*source code*) proses konfirmasi pembayaran dapat dilihat pada

lampiran J. Tampilan halaman utama konfirmasi pembayaran pun juga sama seperti saat akan melakukan pencarian pendaftar. Sedangkan tampilan form konfirmasi nya (konfirmasi.php) adalah seperti gambar 5.36.

Halaman Konfirmasi
Tolong masukkan form yang ada.

No Pendaftaran	34
Nama	anton
Tanggal Transfer :	<input type="text"/>
Jam Transfer :	<input type="text"/>
No Rekening :	<input type="text"/>
Konfirmasi!	

Gambar 5.36 Tampilan Form Konfirmasi Berbasis WAP

Apabila form tersebut tidak diisi, maka akan muncul sebuah pesan kesalahan bahwa data harus diisi. Setelah form tersebut diisi lengkap dan user melakukan klik konfirmasi, maka halaman yang dituju adalah post_konfirmasi.php muncul pesan bahwa data bisa disimpan. Prosesnya pertama-tama data dikirim ke halaman post_konfirmasi.php menggunakan perintah postfield, seperti pada potongan *script* berikut:

```

<anchor>Konfirmasi !
<go href="main.php?page=17&id=<?=$cari_id;?>&type=post "
method="post">" ;?>
    <postfield name="tgl" value="$ (tgl) " />
    <postfield name="jam" value="$ (jam) " />
    <postfield name="no_rek" value="$ (no_rek) " />
</go></anchor>

```

Kemudian data yang dipost-kan tersebut ditangkap menggunakan menggunakan *script* `$id = $_GET['id'];` atau `$tgl = $_POST['tgl'];`, sehingga kemudian data konfirmasi tersebut bisa dimasukkan ke tabel pendaftaran menggunakan *query update* berikut:

```

$sql= "UPDATE pendaftaran set ko_tgl='$tgl', ko_jam='$jam',
ko_no_rek='$no_rek' where p_id='$id'";

```

5.2 Pengujian

Pengujian sistem dalam penelitian ini dilakukan dengan menggunakan metode *black box testing* dan *alfa testing*. Perangkat keras dan perangkat lunak yang digunakan dalam pengujian antara lain:

1. Laptop Axioo Neon dengan prosesor Intel Dual Core 1,8 Hz dan RAM 1 GB yang terkoneksi dengan internet.
2. Handphone Nokia 2730 classic atau 6303 classic atau E63 yang memiliki fasilitas WAP dengan mini browser Opera versi 4.1.
3. Mozila Firefox versi 3.6.
4. WinWAP browser versi 3.0.

5.2.1 Metode *Black Box Testing*

Merupakan sebuah metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional aplikasi saat dioperasikan, apakah input diterima dengan benar dan output yang dihasilkan telah sesuai dengan yang diharapkan.

1. Skenario pengujian.

Menguji masing-masing modul dalam sistem. Daftar , metode uji dan kriteria evaluasi hasil sistem dinyatakan dalam Tabel 5.1.

Tabel 5.1 Skenario Pengujian

No	Nama Pengujian	Kriteria Evaluasi Hasil
1.	Pengujian Terhadap Halaman Utama dan Proses Login Admin	Sistem menampilkan pesan kesalahan jika username atau password tidak terdaftar dalam database. Begitu juga jika captcha security images yang diinputkan salah.
2.	Authentifikasi administrator	Sistem memeriksa proses login administrator. Jika berhasil sistem akan membuat session untuk administrator pada saat proses login, dan mematikan session tersebut saat administrator sudah logout, sehingga sistem tidak bisa dibuka lagi oleh siapapun, kecuali harus login lagi.
3.	Pengujian Terhadap Proses Pendaftaran	Sistem menampilkan pesan kesalahan jika masih ada form yang belum diisi. Setelah mendaftar maka sistem secara otomatis akan mengeset gelombang pendaftaran sesuai dengan yang sudah dikonfigurasi. Jika diakses menggunakan handphone, maka proses pendaftaran akan dibagi menjadi 2 halaman karena keterbatasan handphone dalam mengakses sistem.
4.	Pengujian menampilkan data pendaftaran dan pencarian data pendaftaran	Sistem menampilkan data pendaftaran sesuai dengan gelombang yang dipilih oleh user. Namun jika diakses lewat handphone, tampilannya memang sangat terbatas. Ketika user melakukan pencarian, maka data yang ditampilkan hanya yang sesuai dengan yang dicari saja.
5.	Pengujian Terhadap Proses Seleksi Tes	Sistem secara otomatis akan mengubah status calon mahasiswa dengan NEM < 7 menjadi tes tertulis. Sedangkan mahasiswa dengan NEM > 7 diubah statusnya menjadi tes wawancara. Sistem juga menyediakan fitur bagi admin untuk memilih pilihan jurusan diterima dari pilihan mahasiswa tersebut jika dinyatakan lulus tes.
6.	Pengujian menampilkan data kelulusan dan pencarian data kelulusan	Sistem menampilkan data kelulusan sesuai dengan menu tes yang dipilih oleh user. Ketika user melakukan pencarian, maka data yang ditampilkan hanya yang sesuai dengan yang dicari saja.
7.	Pengujian terhadap proses konfirmasi pembayaran	Sistem menyediakan form konfirmasi pembayaran bagi mahasiswa yang memang tidak sempat membayar secara offline. Sistem akan menampilkan pesan kesuksesan jika konfirmasi berhasil dilakukan. Jika mahasiswa sudah melakukan konfirmasi, maka admin yang akan melakukan cek kebenaran pembayaran tersebut via e-banking.
8.	Pengujian konfigurasi PMB	Sistem menyediakan form bagi admin untuk melakukan konfigurasi gelombang PMB yaitu memilih gelombang saat itu. Sehingga jika ada mahasiswa yang mendaftar, secara otomatis sistem akan mengelompokkan mahasiswa tersebut ke dalam gelombang saat itu.

5.2.2 Metode Alfa Testing

Pengujian *alfa testing* dilakukan oleh user dengan mengakses sistem baik melalui *handphone* maupun komputer. Kemudian user mengisi form kuisioner. Pengujian dilakukan terhadap fungsional sistem serta interface dan pengaksesan. Daftar penguji pada pengujian alfa test dapat dilihat pada tabel 5.2.

Tabel 5.2 Daftar Penguji pada Pengujian *Alfa Test*

No	Nama	Jabatan	Instansi
1	Dian Sigit Prastowo	Mahasiswa	Jurusan Teknik Informatika, UIN Sunan Kalijaga
2	Endah Fatma	Mahasiswi	Jurusan Teknik Informatika, STMIK AMIKOM
3	Febriani Sekti Saputri	Mahasiswi	Jurusan Ilmu Pemerintahan, UMY
4	Muhammad Alex	Mahasiswa	Jurusan Teknik Informatika, UIN Sunan Kalijaga
5	Uhlul	Mahasiswi	Jurusan Sistem Informasi, STMIK AMIKOM
6	Lintang Waskita	Pelajar	SMA N 1 Depok, Yogyakarta
7	Natasha Amelia	Pelajar	SMA N 1 Prambanan
8	Harno	Pelajar	SMK N 2 Sleman
9	Lucky Priyanka	Pelajar	SMA Muhammadiyah Kolombo
10	Gatot Setiawan	Pelajar	SMK Nasional, Berbah
11	Piyan Rudianto	Mahasiswa	Jurusan Pendidikan Kimia, UIN Sunan Kalijaga
12	Syanti Hidyasiwi	Mahasiswi	Jurusan Teknik Informatika, UIN Sunan Kalijaga
13	Susilo Wibowo	Pengusaha	Tiketing Online , Dirgantara Jaya
14	Muhammad Aziz	Pengusaha	Event Organizer
15	Arif	Karyawan	Swalayan

Hasil pengujian fungsional sistem dapat dilihat pada tabel 5.3, sedangkan hasil pengujian interface dan pengaksesan dapat dilihat pada tabel 5.4.

Tabel 5.3 Hasil Pengujian Fungsional Sistem

No	Pernyataan	SS	S	TS	STS
1	Sistem menampilkan data pendaftar sesuai gelombang dan bisa melakukan pencarian data pendaftar berdasarkan nomor atau nama pendaftaran	8	7	-	-
2	Pada form input data pendaftaran, saat tombol daftar di klik, sistem akan menampilkan pesan berhasil jika semua form diisi dengan lengkap, sedangkan jika form ada yang kosong maka sistem akan menampilkan pesan kesalahan	10	5	-	-
3	Sistem menampilkan data kelulusan berdasarkan jalur tes dan bisa melakukan pencarian data kelulusan berdasarkan nomor atau nama pendaftaran	6	9	-	-
4	Pada menu konfirmasi pembayaran, user memasukkan nomor atau nama pendaftaran. Sistem akan menampilkan pesan kesalahan jika nomor atau nama tersebut tidak terdaftar	7	8	-	-
5	Sistem memberi kemudahan dalam melakukan konfirmasi pembayaran online karena user cukup menginputkan data tanggal, jam transfer dan no rekeningnya ke dalam form konfirmasi	12	3	-	-
TOTAL		43	32	0	0

Tabel 5.4 Hasil Pengujian Interface dan Pengaksesan

No	Pernyataan	SS	S	TS	STS
1	Form-form inputan dalam sistem memberi kemudahan sehingga memudahkan dalam penggunaan aplikasi.	8	5	2	-
2	Konten yang disediakan cukup sederhana, sehingga memudahkan dalam penggunaan sistem	2	12	1	-
3	Sistem memiliki navigasi yang mudah.	5	8	2	-
4	Waktu loading relatif cepat.	3	7	5	-
5	Menu dan navigasi pada sistem sudah berfungsi	7	8	-	-
TOTAL		25	40	10	0

Dari tabel 5.3 dan 5.4 didapatkan hasil data yang menunjukkan bahwa 45,3 % menyatakan sangat setuju; 48 % menyatakan setuju; 6,7 % menyatakan tidak setuju dan 0 % menyatakan sangat tidak setuju. Sebagian banyak user menyatakan setuju dengan aplikasi yang telah dibuat, dan hanya sedikit user yang merasa kesulitan dengan aplikasi tersebut.

Dari hasil pengujian, baik pada metode *black box testing* ataupun *alfa testing*, dapat disimpulkan bahwa sistem informasi penerimaan mahasiswa baru yang telah dibuat layak untuk digunakan. Akan tetapi perlu adanya pengembangan sistem yang lebih lanjut untuk mendapatkan hasil yang maksimal.



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan kegiatan yang telah dilakukan oleh penulis selama perancangan sampai implementasi sistem informasi penerimaan mahasiswa baru ini, maka dapat diambil beberapa kesimpulan berikut:

1. Telah dirancang sistem informasi penerimaan mahasiswa baru berbasis WEB dan WAP menggunakan bahasa pemrograman PHP dan WML, serta MySQL sebagai *database*.
2. Sistem informasi penerimaan mahasiswa baru berbasis WEB dan WAP sudah memberikan kemudahan kepada calon mahasiswa dalam mengakses informasi dan melakukan proses pendaftaran serta konfirmasi pembayaran pendaftaran *online*. Hal ini terbukti dengan hasil akhir dari kuisioner yang dilakukan saat tahap pengujian.

6.2. Saran

Sistem informasi penerimaan mahasiswa baru ini tidak terlepas dari kekurangan dan kelemahan, terutama dalam tampilan sistem berbasis WAP yang cukup sederhana serta konten seadanya yang disebabkan keterbatasan yang dimiliki

penulis. Oleh karena itu, untuk pengembangan sistem informasi penerimaan mahasiswa baru yang lebih baik, penulis menyarankan beberapa hal, antara lain:

1. Penambahan beberapa fasilitas yang lebih kompleks di dalam sistem, misalnya tes atau ujian *online*.
2. Sistem bisa terintegrasi dengan bank, sehingga admin tidak perlu melakukan cek manual saat ada mahasiswa yang melakukan pembayaran pendaftaran *online*.
3. Sistem bisa diakses dari semua *handphone* yang berfasilitas WAP, tanpa dibatasi mini browsernya harus Opera Mini versi 4.1.

Akhirnya dengan segala keterbatasan sistem yang dibuat penulis ini, penulis tetap berharap bahwa sistem ini akan memberikan gagasan baru bagi pembaca untuk mengembangkan lebih lanjut. Selain itu semoga sistem ini dapat digunakan sebagai langkah awal dalam pemanfaatan teknologi informasi, khususnya teknologi internet dan *handphone* sebagai media pendaftaran mahasiswa baru alternatif.

DAFTAR PUSTAKA

- Arief, M. Rudyanto. 2006. "Pemrograman Basis Data Menggunakan Transact-SQL dengan Microsoft SQL Server 2000". Penerbit Andi. Yogyakarta.
- Arifin, Agus. 2009. "Analisa dan Perancangan Sistem Informasi Berbasis WEB Penjualan Online di PT. Intan Pariwara". Skripsi, STMIK AMIKOM, Yogyakarta.
- Davis, Gordon B., 1991. "Kerangka Dasar Sistem Informasi Manajemen Bagian 1". PT Pustaka Binamas Pressindo, Jakarta.
- Jogiyanto HM. 1999. "Analisis dan Desain Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis". Andi Offset, Yogyakarta.
- Kristanto, Abraham. 2005. "Aplikasi Sistem Registrasi KRS pada handphone dengan menggunakan J2ME". Skripsi, STMIK AKAKOM, Yogyakarta.
- Kristanto, Andri. 2003. "Perancangan Sistem Informasi dan Aplikasinya". Gava Media, Yogyakarta.
- Kusrini. 2007. "Strategi Perancangan dan Pengelolaan Basis Data". Penerbit Andi Yogyakarta.
- Mahyuzir, T.,D., 1991. "Pengantar Analisis dan Perancangan Perangkat Lunak". Elex Media Komputindo, Jakarta.
- Mulyanto, Agus. 2005. "Sistem Informasi Akademik Berbasis Web Pada Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga". Tesis, UGM, Yogyakarta.
- Mulyanto, Agus. 2009. "Sistem Informasi Konsep & Aplikasi". Pustaka Pelajar, Yogyakarta.
- Nugroho, Bunafit. 2005. "Pengembangan Program WAP dengan WML dan PHP", Gava Media, Yogyakarta.
- Nurdiansyah, Didin. 2009. "Rancangan Sistem Berbasis WAP Pemesanan Tiket Online Kereta Eksekutif".
http://didinnurdiansyah.blog.upi.edu/files/2009/06/sesudah_revisi.pdf diakses tanggal 21 April 2010.

- Pasumbung, Eka Linda. 2007. "Analisis dan Perancangan Aplikasi Berbasis Web Pada Rumah Sakit Mata dr.YAP Yogyakarta". Skripsi, STMIK AMIKOM, Yogyakarta.
- Prasetyawan, Daru. 2009. "Aplikasi *Wireless Application Protocol* Untuk Portal Mobile Dakwah". Skripsi, UIN Sunan Kalijaga, Yogyakarta.
- Sariana, Krisna. 2010. "Perancangan Sistem Informasi Berbasis WEB pada kantor BPN Sleman". Skripsi, STMIK AMIKOM, Yogyakarta.
- Simarmata, Janner. 2006. "Pemrograman WAP dengan menggunakan WML". Penerbit Andi. Yogyakarta.
- Simarmata, Janner. 2006. "Aplikasi Mobile Commerce Menggunakan PHP dan MySQL". Penerbit Andi. Yogyakarta.
- Sunarfrihantono, Bimo. 2002. "PHP & MySQL Untuk Web". Penerbit Andi. Yogyakarta.
- Sutejo, Bambang. 2007. "Konsep Dasar RDBMS". http://www.klik-oracle.web.id/download-materi/Konsep_Dasar.pdf diakses tanggal 28 Juli 2010.
- Utomo, Prasetyo Ambang, S.T. 2006. "Membangun Aplikasi WAP Portal untuk Instansi/Lembaga" Penerbit Andi. Yogyakarta.
- Wahyuni, Sri Ngudi. 2009. "Wireless Application Protocol". <http://p3m.amikom.ac.id/p3m/dasi/maret07/06%20-%20STMIK%20AMIKOM%20Yogyakarta%20Wireless%20Application%20Protocol.pdf> diakses tanggal 20 April 2010.