

**IMPLEMENTASI SOA (*SERVICE ORIENTED ARCHITECTURE*)
DALAM SISTEM INTEGRASI *WEB SERVICES* PADA
APLIKASI *SMART MAHAD AL-JAMIAH*
UIN MAULANA MALIK IBRAHIM
MALANG**

SKRIPSI

Oleh :
MUHAMMAD NAFIAN WILDANA
NIM 11650086



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

HALAMAN PENGANTAR

**IMPLEMENTASI SOA (*SERVICE ORIENTED ARCHITECTURE*)
DALAM SISTEM INTEGRASI *WEB SERVICES* PADA
APLIKASI *SMART MAHAD AL-JAMIAH*
UIN MAULANA MALIK IBRAHIM
MALANG**

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komunikasi (S.Kom)**

Oleh :

M. Nafian Wildana

NIM.11650086

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

HALAMAN PERSETUJUAN

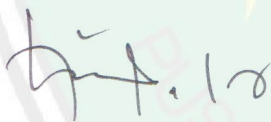
IMPLEMENTASI SOA (*SERVICE ORIENTED ARCHITECTURE*) DALAM SISTEM INTEGRASI *WEB SERVICES* PADA APLIKASI *SMART MAHAD AL-JAMIAH* UIN MAULANA MALIK IBRAHIM MALANG

SKRIPSI

Oleh :
M. NAFIAN WILDANA
NIM.11650086

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 21 Mei 2018

Dosen Pembimbing I



Linda Salma Angreani, M.T
NIP. 19770803 200912 2 005

Dosen Pembimbing II



Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Sanyo Crysdian

NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

**IMPLEMENTASI SOA (*SERVICE ORIENTED ARCHITECTURE*)
DALAM SISTEM INTEGRASI *WEB SERVICES* PADA
APLIKASI *SMART MAHAD AL-JAMIAH*
UIN MAULANA MALIK IBRAHIM
MALANG
SKRIPSI**

Oleh :

**M. NAFIAN WILDANA
NIM.11650086**

Telah dipertahankan di depan dewan penguji skripsi
dan dinyatakan diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana komputer (S.Kom)
Tanggal : Juli 2018

Susunan Dewan Penguji

Penguji Utama

: Ajib Hanani, M.T

NIDT. 19840731 20160801 1 076

Tanda Tangan

()

Ketua Penguji

: A'la Syauqi, M.Kom

NIP. 19771201 200801 1 007

()

Sekretaris Penguji

: Linda Salma Angreani, M.T

NIP. 19770803 200912 2 005

()

Anggota Penguji

: Syahiduz Zaman, M.Kom

NIP. 19700502 2005 01 1 005

()

Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika

Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdian

NIP. 19740424 200901 1 008

SURAT PERNYATAAN
ORISINALITAS PENELITIAN

Saya yang bertanda tangan di bawah ini:

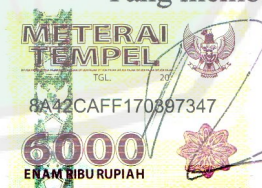
Nama : M. Nafian Wildana
NIM : 11650086
Fakultas/ Jurusan : Sains dan Teknologi / Teknik Informatika
Angkatan tahun/semester : **IMPLEMENTASI SOA (*SERVICE ORIENTED ARCHITECTURE*) DALAM SISTEM INTEGRASI *WEB SERVICES* PADA APLIKASI *SMART MAHADAL-JAMIAH* UIN MAULANA MALIK IBRAHIM MALANG**

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur penjiplakan, maka saya bersedia untuk mempertanggungjawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 21 Mei 2018

Yang membuat pernyataan



M. Nafian Wildana

NIM. 11650086

HALAMAN PERSEMBAHAN

Dengan rasa syukur seraya mengharap ridho Ilahi kupersembahkan karya ini kepada :

Mukhammad Gufron dan Umi Kultsum selaku orang tua penulis

Yang selalu memberikan kasih sayang, Doa, perhatian, support yang tiada hentinya sehingga skripsi ini dibuat sampai detik ini.

Tanpa Doa dan dukungan luar biasa dari beliau mungkin saya tidak akan pernah bisa masuk pada kampus yang sangat luar biasa ini. Semoga Allah SWT selalu menjaga dalam setiap langkah Orang tua kami.

Terimakasih kepada yang selalu setia mendukung dan mendampingi sampai saat ini istri tercinta saya

Lailatul Lutfiyah

Terimakasih untuk Bapak/Ibu Dosen **Bu Linda** sebagai dosen pembimbing I dan **Bapak Syahid** sebagai dosen Pembimbing II yang selalu sabar membimbing saya dalam menyelesaikan tugas akhir ini. **Pak Faisal** sebagai wali dosen yang senantiasa mengawasi perkembangan perkuliahan saya selama beberapa tahun ini. Terimakasih juga kepada ibu bapak dosen pengajar, uztadz uztadzah yang telah memberikan ilmu dengan keikhlasan, semoga ilmu yang telah beliau beri ini dapat bermanfaat bagi nusa, bangsa dan agama. Kepada teman seperjuangan dan seataap:

Hapid (TI11), Al(TI11), Alif (TI11), Emil Enan (TI11), Ambon (TI11), Iwan (TI11), Dani (TI11), Asroni (TI11).

yang bersama-sama saling menyemangati satu sama lain dan saling mengingatkan jika lalai.

Kepada para **sahabat TI** angkatan 2011, yang selalu ada untuk membantu sesama. Dan kepada teman-temanku semua yang tidak bisa kusebutkan satu persatu yang selalu membantuku dan menyemangatiku di saat aku susah dan terpuruk.

Semoga Allah SWT melindungi, menyayangi dan menempatkan mereka semuanya pada surganya kelak dan melimpahkan rezeki kepada mereka semua...

MOTTO

“JUST DO IT”

(NIKE)



KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT tuhan semesta alam, karena atas segala rahmat dan karunia-Nya kepada penulis sehingga penulis mampu menyelesaikan skripsi dengan judul “IMPLEMENTASI SOA (*SERVICE ORIENTED ARCHITECTURE*) DALAM SISTEM INTEGRASI *WEB SERVICES* PADA APLIKASI *SMART MAHADAL-JAMIAH* UIN MAULANA MALIK IBRAHIM MALANG” dengan baik dan lancar. Shalawat dan salam selalu tercurah kepada tauladan terbaik kita Nabi Agung Muhammad SAW yang telah membimbing umatnya dari zaman kegelapan dan kebodohan menuju cahaya islam yang terang *rahmatan lil alamiin* ini.

Dalam penyelesaian skripsi ini, banyak pihak yang telah memberikan bantuan baik secara moril, nasihat dan semangat maupun materiil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terimakasih yang sedalam-dalamnya kepada :

1. Bapak Prof. Dr. Abdul Haris, M.Ag., selaku rektor UIN Maulana Malik Ibrahim Malang beserta seluruh staf. Dharma Bakti Bapak dan Ibu sekalian terhadap Universitas Islam Negeri Maulana Malik Ibrahim Malang turut membesarkan dan mencerdaskan penulis.
2. Ibu Dr. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh staf. Bapak dan ibu sekalian sangat berjasa memupuk dan menumbuhkan semangat untuk maju kepada penulis.

3. Bapak Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang sudah memberi banyak pengetahuan, inspirasi dan pengalaman yang berharga.
4. Ibu Linda Salma Angreani, M.T selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
5. Bapak Syahiduz Zaman, M.Kom, selaku dosen pembimbing II yang juga senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
6. Ayah, Ibu, dan Adik serta keluarga besar saya tercinta yang selalu memberi dukungan yang tak terhingga serta do'a yang senantiasa mengiringi setiap langkah penulis.
7. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
8. Lailatul Lutfiyah selaku istri tercinta penulis yang selalu memberikan semangat dan dukugan penuh yang tak terhingga serta do'a untuk menyelesaikan penelitian ini.
9. Teman – teman seperjuangan Teknik Informatika 2011.
10. Para peneliti yang telah ikut dalam mengembangkan sistem integrasi *web service* dan aplikasi Android yang menjadi acuan penulis dalam pembuatan skripsi ini. Serta semua pihak yang telah membantu yang tidak bisa disebutkan satu-satu. Terimakasih banyak.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan saran yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya tulis ini bisa bermanfaat dan menginspirasi bagi kita semua. Amin.

Wassalamualaikum Wr. Wb.

Malang, Juli 2018

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
SURAT PERNYATAAN	v
HALAMAN PERSEMBAHAN	vi
MOTTO	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xv
ABSTRAK.....	xix
ABSTRACT	xx
الملخص.....	xxi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	5
1.3. Tujuan Penelitian.....	5
1.4. Batasan Masalah	5
1.5. Manfaat Penelitian	6
BAB II STUDI PUSTAKA	7
2.1. Tinjauan Studi	7
2.2. Tinjauan Pustaka	8
2.2.1. <i>Service Oriented Architecture (SOA)</i>	8
2.2.2. Jenis Integrasi	11
2.2.3. Jenis Arsitektur Integrasi.....	12
2.2.4. <i>Web Services</i>	17
1. SOAP (Simple Object Access Protocol)	18
2. WSDL (Web Services Description Language).....	19
3. <i>UDDI Registry (Universal Description, Discovery and Integration)</i> ..	20
2.2.5. Skala Likert.....	21
BAB III METODE PENELITIAN.....	24
3.1. Analisis Kebutuhan Sistem	24
3.2. Perancangan Sistem.....	25
3.2.1. Model Layer Aplikasi.....	25
3.2.2. Aplikasi Smart Mahad.....	25
3.2.3. Layer Integrasi	26

3.2.4. Layer Database	26
3.2.5. Layer Infrastruktur Jaringan	26
3.2.6. Arsitektur Sistem.....	27
1. UDDI (<i>Universal Discovery Description Integration</i>) / <i>Service Registry</i>	27
2. <i>Service Orchestration</i>	28
3. <i>Smart-Mahad Web Service</i>	28
3.2.7. <i>Sequence Diagram</i>	29
1. <i>Sequence Diagram</i> Untuk Mahasantri	29
2. <i>Sequence diagram</i> untuk staff <i>mahad/admin</i>	30
3.2.8. <i>Activity Diagram</i>	31
1. <i>Diagram activity login</i>	31
2. <i>Diagram activity news feed</i>	31
3. <i>Diagram activity report</i>	32
3.2.9. Use Case Diagram.....	33
3.2.10. Perancangan Basis Data	34
1. Conceptual Data Model.....	35
2. Physical Data Model (PDM)	37
3.2.11. Struktur Basis Data	38
1. Tabel admin.....	38
2. Tabel user	38
3. Tabel pinlog.....	38
4. Tabel fakultas	39
5. Tabel jurusan	39
6. Tabel mahasantri.....	39
7. Tabel musyrif.....	40
8. Tabel murobbi.....	41
9. Tabel pengasuh	41
10. Tabel mabna	41
11. Tabel absensi	42
12. Tabel jadwal mahad.....	42
13. Tabel penjadwalan regular	43
14. Tabel kelas PPBA	43
15. Tabel Mahasantri regular.....	44
16. Tabel Mahasantri PPBA.....	44
17. Tabel Track Posisi.....	45
3.2.12. Perancangan antarmuka.....	45
BAB IV UJI COBA DAN PEMBAHASAN	47
4.1. Kebutuhan Perangkat Keras	47
4.2. Kebutuhan Perangkat Lunak	47
4.3. Implementasi Sistem.....	48
4.3.1. Implementasi Basis Data	48
4.3.2. Implementasi <i>Web Service</i>	55
1. Mahasantri <i>Service</i>	56
2. Musyrif <i>Service</i>	64
3. Pengasuh <i>Service</i>	80
4. Global <i>Service</i>	90

5. News Service	92
4.3.3. Implementasi Mobile Apps Android.....	93
4.4. Pengujian Sistem	104
4.4.1. Metode Pengujian	104
1. Pengujian Alpha.....	104
a. Pengujian data pada mahasantri <i>service</i>	105
b. Pengujian data pada <i>musyrif service</i>	107
c. Pengujian data pada pengasuh <i>service</i>	107
d. Pengujian data pada <i>global service</i>	107
e. Pengujian data pada <i>news service</i>	107
2. Pengujian <i>Blacbox</i>	107
4.5. Integrasi Sistem Mahad Dalam Pandangan Islam	123
BAB V KESIMPULAN DAN SARAN	125
5.1. Kesimpulan	125
5.2. Saran	126
DAFTAR PUSTAKA.....	127



DAFTAR GAMBAR

Gambar 2.1 <i>Spaghety Integration Dilema</i> (Siriwardena, 2012).....	13
Gambar 2.3 Central Message Broker yang memiliki arsitektur hub and spoke (Juric, 2007).....	14
Gambar 2.4 Service Oriented Architecture (Erl, 2005)	15
Gambar 2.5 Arsitektur ESB (IBM developer networks)	16
Gambar 2.6 Arsitektur Web Service (Juric, 2007)	18
Gambar 2.7 Arsitektur UDDI.....	20
Gambar 3.1 Model layer aplikasi untuk sistem integrasi <i>smart-mahad</i>	25
Gambar 3.2 Model SOA Aplikasi Smart Mahad.....	27
Gambar 3.3 Ilustrasi message routing (EAI Patterns)	28
Gambar 3.4 <i>Sequence</i> diagram aktifitas mahasantri.....	30
Gambar 3.5 <i>Sequence</i> diagram <i>staff mahad</i>	30
Gambar 3.6 Diagram <i>activity login</i>	31
Gambar 3.7 Diagram <i>activity news feed</i>	32
Gambar 3.8 Diagram activity reporting	32
Gambar 3.9 Use Case Diagram SIAMA.....	33
Gambar 3.10 Rancangan <i>mockup</i> halaman <i>login</i>	46
Gambar 3.11 Rancangan mockup halaman <i>container home sliding menu</i> dan <i>menu content</i>	46
Gambar 4.1 Tampilan halaman <i>login</i> SIAMA	94
Gambar 4.2 Tampilan <i>news updated</i> dan menu utama SIAMA	95
Gambar 4.3 Tampilan halaman <i>profile</i> dan presensi mahasantri	97
Gambar 4.4 Tampilan halaman fasilitas dan registrasi <i>musyrif</i>	97
Gambar 4.5 Tampilan depan dan menu presensi <i>musyrif</i> divisi bahasa	100
Gambar 4.6 Tampilan halaman depan <i>musyrif</i> divisi inventaris	101
Gambar 4.7 Tampilan halaman depan <i>musyrif</i> divisi kesantrian	101
Gambar 4.8 Tampilan halaman depan pengasuh dan data mahasantri.....	103
Gambar 4.9 Tampilan halaman report presensi mahsantri berdasar	104
Gambar 4.10 Tampilan software Insomnia untuk <i>testing web service</i>	105

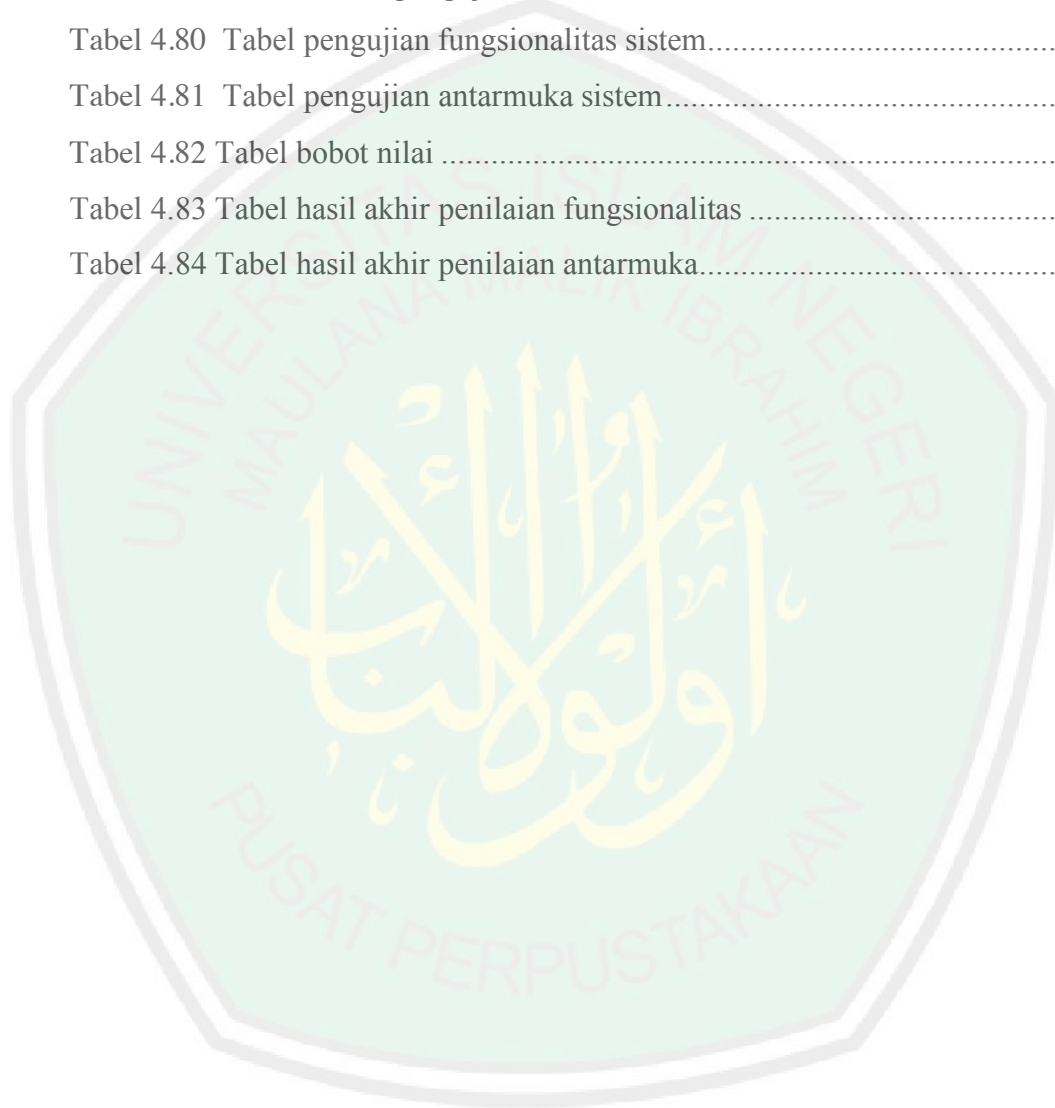
DAFTAR TABEL

Tabel 3.1 Tabel <i>Web Service</i> sistem integrasi SIAMA	28
Tabel 3.2 Tabel <i>actor use case</i> SIAMA.....	33
Tabel 3.3 Definisi <i>activity</i>	34
Tabel 3.4 Keterangan CDM	35
Tabel 3.5 Tabel detail admin.....	38
Tabel 3.6 Tabel detail user	38
Tabel 3.7 Tabel pinlog	38
Tabel 3.8 Tabel fakultas.....	39
Tabel 3.9 Tabel jurusan.....	39
Tabel 3.10 Tabel mahasiswa	39
Tabel 3.11 Tabel musyrif.....	40
Tabel 3.12 Tabel murobbi.....	41
Tabel 3.13 Tabel pengasuh.....	41
Tabel 3.14 Tabel mabna.....	41
Tabel 3.15 Tabel absensi.....	42
Tabel 3.16 Tabel mahad.....	42
Tabel 3.17 Tabel penjadwalan reguler.....	43
Tabel 3.18 Tabel kelas PPBA.....	43
Tabel 3.19 Tabel mahasiswa reguler	44
Tabel 3.20 Tabel mahasiswa PPBA.....	44
Tabel 3.21 Tabel trackposisi	45
Tabel 4.1 <i>Script</i> SQL absensi	48
Tabel 4.2 <i>Script</i> SQL admin.....	49
Tabel 4.3 <i>Script</i> SQL user.....	49
Tabel 4.4 <i>Script</i> SQL pinlog	50
Tabel 4.5 <i>Script</i> SQL fakultas	50
Tabel 4.6 <i>Script</i> SQL jurusan.....	50
Tabel 4.7 <i>Script</i> SQL mahasiswa	51
Tabel 4.8 <i>Script</i> SQL <i>musyrif</i>	52
Tabel 4.9 <i>Script</i> SQL <i>murabbi</i>	53

Tabel 4.10 <i>Script SQL pengasuh</i>	53
Tabel 4.11 <i>Script SQL mabna</i>	53
Tabel 4.12 <i>Script SQL jadwalmahad</i>	54
Tabel 4.13 <i>Script SQL penjadwalan reguler</i>	54
Tabel 4.14 <i>Tabel Bahasa pemrograman</i>	55
Tabel 4.15 <i>Operasi web service mahasantri</i>	55
Tabel 4.16 <i>Script PHP untuk method profil()</i>	56
Tabel 4.17 <i>Script PHP untuk method absensi()</i>	57
Tabel 4.18 <i>Script PHP untuk method iqob()</i>	58
Tabel 4.19 <i>Script PHP untuk method tashih()</i>	59
Tabel 4.20 <i>Script PHP untuk method khs()</i>	59
Tabel 4.21 <i>Script PHP untuk method fasilitas()</i>	60
Tabel 4.22 <i>Script PHP untuk method kondisikamar()</i>	61
Tabel 4.23 <i>Script PHP untuk method infopendaftaran()</i>	62
Tabel 4.24 <i>Script PHP untuk method regmusyrif()</i>	62
Tabel 4.25 <i>Script PHP untuk method pengumuman()</i>	63
Tabel 4.26 <i>Script PHP untuk method sendlocation()</i>	64
Tabel 4.27 <i>Operasi web services mahad activity Musyrif</i>	64
Tabel 4.28 <i>Script PHP untuk method presensi()</i>	66
Tabel 4.29 <i>Script PHP untuk method updatepresensi()</i>	67
Tabel 4.30 <i>Script PHP untuk method kelasbahasa()</i>	68
Tabel 4.31 <i>Script PHP untuk method nilaitest()</i>	68
Tabel 4.32 <i>Script PHP untuk method nilaiutsuas()</i>	69
Tabel 4.33 <i>Script PHP untuk method mahasantripandai()</i>	70
Tabel 4.34 <i>Script PHP untuk method kelasquran()</i>	71
Tabel 4.35 <i>Script PHP untuk method ijinpulang()</i>	72
Tabel 4.36 <i>Script PHP untuk method pelanggaranSantri()</i>	72
Tabel 4.37 <i>Script PHP untuk method notifikasiiqob()</i>	73
Tabel 4.38 <i>Script PHP untuk method kelasafkar()</i>	74
Tabel 4.39 <i>Script PHP untuk method listsantri()</i>	75
Tabel 4.40 <i>Script PHP untuk method detailfasiltas()</i>	75
Tabel 4.41 <i>Script PHP untuk method detailkondisikamar()</i>	76

Tabel 4.42 Script PHP untuk <i>method invenmenunggu()</i>	77
Tabel 4.43 Script PHP untuk <i>method invenditolak()</i>	78
Tabel 4.44 Script PHP untuk <i>method invendiproses()</i>	78
Tabel 4.45 Script PHP untuk <i>method inventidakdirespons()</i>	79
Tabel 4.46 Operasi <i>web services mahad activity</i> Pengasuh	79
Tabel 4.47 Script PHP untuk <i>method listsantrimabna()</i>	80
Tabel 4.48 Script PHP untuk <i>method listiqob()</i>	81
Tabel 4.49 Script PHP untuk <i>method listpresensi()</i>	82
Tabel 4.50 Script PHP untuk <i>method listpenjadwalanmualim()</i>	83
Tabel 4.51 Script PHP untuk <i>method listbulanmualim()</i>	84
Tabel 4.52 Script PHP untuk <i>method listbulanmualim()</i>	84
Tabel 4.53 Script PHP untuk <i>method listkehadiranbaik()</i>	85
Tabel 4.54 Script PHP untuk <i>method listkehadiranburuk()</i>	86
Tabel 4.55 Script PHP untuk <i>method mhspandai()</i>	87
Tabel 4.56 Script PHP untuk <i>method penerimaanmusyrif()</i>	88
Tabel 4.57 Script PHP untuk <i>method penempatanmusyrif()</i>	89
Tabel 4.58 Script PHP untuk <i>method listsetoran()</i>	89
Tabel 4.59 Script PHP untuk <i>method listmabna()</i>	90
Tabel 4.60 Operasi <i>web services mahad activity</i> Global	90
Tabel 4.61 Script PHP untuk <i>method userlogin()</i>	90
Tabel 4.62 Operasi <i>web services news feed</i>	91
Tabel 4.63 Script Javascript untuk <i>method getUrlFeed()</i>	92
Tabel 4.64 Script Java untuk <i>method LoginService()</i>	93
Tabel 4.65 Script Java untuk <i>method LoginService()</i>	94
Tabel 4.66 Script Java untuk <i>method MahasantriService()</i>	96
Tabel 4.67 Script Java untuk <i>method MusyrifService()</i>	98
Tabel 4.68 Script Java untuk <i>method PengasuhService()</i>	102
Tabel 4.69 Operasi <i>web services mahad testing</i> mahasantri data normal.....	106
Tabel 4.70 Operasi <i>web services mahad testing</i> mahasantri data salah.....	107
Tabel 4.71 Operasi <i>web services mahad testing</i> musyrif data benar	108
Tabel 4.72 Operasi <i>web services mahad testing</i> Musyrif data salah	110
Tabel 4.73 Operasi <i>web services mahad testing</i> Pengasuh data normal.....	112

Tabel 4.74 Operasi <i>web services mahad testing</i> Pengasuh data salah.....	113
Tabel 4.75 Operasi <i>web services mahad testing global service</i> data normal.....	114
Tabel 4.76 Operasi <i>web services mahad testing global service</i> data salah.....	114
Tabel 4.77 Operasi <i>web services news feed testing</i> data benar	115
Tabel 4.78 Operasi <i>web services news feed testing data salah</i>	115
Tabel 4.79 Tabel kriteria pengujian sistem	116
Tabel 4.80 Tabel pengujian fungsionalitas sistem.....	117
Tabel 4.81 Tabel pengujian antarmuka sistem.....	118
Tabel 4.82 Tabel bobot nilai	118
Tabel 4.83 Tabel hasil akhir penilaian fungsionalitas	119
Tabel 4.84 Tabel hasil akhir penilaian antarmuka.....	121



ABSTRAK

Wildana, Muhammad Nafian. 2018. **Implementasi SOA (*Service Oriented Architecture*) Dalam Sistem Integrasi *Web Services* Pada Aplikasi *Smart Mahadal-Jamiah* UIN Maulana Malik Ibrahim Malang**. Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Linda Salma Angreani, M.T (II) Syahiduz Zaman, M.Kom

Kata Kunci : *SOA, Integrasi Web Service, Pusat Mahad Al-Jami'ah*

Universitas Islam Negeri Maulana Malik Ibrahim Malang sebagai perguruan tinggi Islam memiliki program wajib bagi mahasiswa baru mengikuti *mahad* selama satu tahun, kegiatan ini dikelola oleh Pusat Mahad *Al-Jami'ah*. Pada Pusat Mahad *Al-Jami'ah* telah terdapat beberapa layanan berbasis IT yaitu mahad *service*, mahasantri *tracking service* dan *news service*. Guna pengembangan integrasi antar layanan maka diimplementasikan SOA (*Service Oriented Architecture*) dan pembangunan aplikasi *Smart mahad* berbasis *mobile*. Berdasarkan hasil pengujian menunjukkan bahwa SOA dapat diimplementasikan dengan baik dimana *service* dapat diintegrasikan setelah dilakukan modularisasi *service*, kemudian data *request web service* digunakan oleh aplikasi *mobile*. Hasil pengujian *acceptance test* yang dilakukan pada *user* menunjukkan bahwa 75% fungsionalitas berjalan baik dan 84% memiliki tampilan yang mudah dipahami.

ABSTRACT

Wildana, Muhammad Nafian. 2018. *Implementation of SOA (Service Oriented Architecture) In Web Services Integration System On Smart Mahad Al-Jamiah Application at UIN Maulana Malik Ibrahim Malang*. Theses. Department of Informatic Engineering, Faculty of Science and Technology. State Islamic University of Maulana Malik Ibrahim Malang.

Advisors: (I) Linda Salma Angreani, M.T (II) Syahiduz Zaman, M.Kom

Maulana Malik Ibrahim Islamic State University of Malang as an Islamic college has a mandatory program for new students to attend *mahad* for one year, this activity is managed by Mahad Al-Jami'ah Center. In Mahad Al-Jami'ah Center there have been several IT-based services namely mahad service, mahasantri tracking service and news service. To develop integration between that services, SOA (Service Oriented Architecture) implemented and build an app smart mahad mobile application. Based on the test results show that SOA can be implemented well where the service can be integrated after modularization service, then data request web service used by mobile application. Acceptance test results performed on the user shows that 75% of functionality is running well and 84% have a view that is easy to understand.

Key words : SOA, Integrated Web Service, Center Mahad Al-Jami'ah

الملخص

ولدان، محمد نافعن، ٢٠١٨، تطبيق البنية الخدمية الخدمية في نظام تكامل خدمات الويب على تطبيق "مهاده الجامي" الذكي في جامعة الملك عبدالله مولانا مالك إبراهيم مالانج. الرسائل العلمية. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا. جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانج تحت المشرف : (١) ليندا سلمى الماجستير (٢) شحذامن الماجستير

Mahad Al-Jami'ah ، تكامل خدمة الويب ، مركز ، SOA : الكلمة الأساسية

مولانا مالك إبراهيم جامعة ولاية الإسلامية الإسلامية في مالانج ككلية إسلامية لديها برنامج إلزامي للطلاب الجدد لحضور مهاده لمدة عام ، يدير هذا النشاط مركز "مركز الجامع". في مركز "مركز الجامع" كانت هناك mahasanthri ، وخدمة تعقب mahad العديد من الخدمات القائمة على تكنولوجيا المعلومات ، وهي خدمة خدمة المنحى المعماري) وبناء التطبيق (SOA وخدمة الأخبار. لتطوير التكامل بين تلك الخدمات ، نفذت يمكن تنفيذها بشكل جيد SOA استنادا إلى نتائج الاختبار تبين أن mahad. المحمول تطبيقات الهاتف الذكي حيث يمكن دمج الخدمة بعد خدمة النمذجة ، ثم خدمة الويب طلب البيانات المستخدمة من قبل تطبيقات الهاتف المتحرك. تظهر نتائج اختبار القبول على المستخدم أن 75٪ من الوظائف تعمل بشكل جيد وأن 84٪ لديهم طريقة عرض يسهل فهمها

BAB I

PENDAHULUAN

Pada bab ini akan membahas mengenai tentang dasar dan tujuan penelitian ini dilakukan mulai dari latar belakang hingga manfaat penelitian ini dibuat.

1.1. Latar Belakang

Pemanfaatan teknologi informasi dan sistem informasi saat ini telah mencapai perkembangan yang sangat pesat mengikuti kebutuhan akses informasi pada segala bidang, salah satunya adalah pemanfaatan teknologi informasi pada banyak perusahaan maupun organisasi besar dengan tujuan utama untuk menerapkan sistem bisnis yang berlaku di instansinya masing-masing. Kedua sistem yang berbeda ini dikolaborasikan bersama dengan aturan yang saling menopang satu sama lain antar sistem. Dikarenakan dalam suatu organisasi modern, pemanfaatan teknologi informasi dan sistem informasi tidak hanya sebatas pada sarana pendukung atau alat saja, namun juga sebagai bagian utama dari organisasi itu sendiri dalam mencapai tujuannya.

Namun perkembangan arus teknologi yang pesat masih tidak dapat diimbangi oleh perkembangan sistem bisnis, karena itu terdapat kesenjangan antara sistem bisnis dan teknologi informasi yang mendukungnya. (Utomo, 2012).

Akibat dari masalah tersebut, contoh permasalahan yang sering dihadapi adalah suatu aplikasi yang dimaksudkan untuk mendukung tugas-tugas suatu

sistem bisnis akan berjalan dengan tidak optimal. Permasalahan ini dapat ditemukan di organisasi-organisasi maupun perusahaan yang membutuhkan interoperabilitas yang tinggi antar sistem. Masalah yang paling sering dihadapi suatu organisasi besar adalah keberagaman atau heterogenitas dari aplikasi dan infrastruktur IT yang sudah dibangun dalam sebuah organisasi tersebut, contoh yang sering ditemui adalah jika suatu organisasi atau perusahaan memiliki departemen yang terlepas dari proses bisnis utama dan dibangun dengan infrastruktur IT yang berbeda dari sistem utama. Masalah lain adalah aplikasi harus tetap berjalan pada saat sistem sedang dilakukan perbaikan atau penambahan fitur tertentu. Cara untuk menjaga integritas antar sistem agar tetap stabil dan tidak terganggu adalah sebuah jembatan atau penyelarasan antar sistem sebagai jalan tengah kedua lapisan sistem informasi dan sistem bisnis tersebut (Juric, 2007).

Untuk melakukan pengelolaan sebagai media yang akan menjembatani kedua sistem berbeda tersebut telah dikembangkan beberapa metode untuk solusi permasalahan tersebut, antara lain dikenalkan sebuah arsitektur *service* yang dapat melakukan pendekatan dan penanganan terhadap kedua sistem bisnis dan sistem informasi yang disebut dengan SOA (*Service Oriented Architectures*). SOA adalah sebuah bentuk teknologi arsitektur yang mengikuti prinsip-prinsip berorientasi layanan (Erl, 2005). Menurut Utomo, implementasi SOA dalam suatu organisasi atau perusahaan mampu menjawab masalah keberagaman penggunaan infrastruktur IT dan ragam proses bisnis

antar organisasi dengan tujuan peningkatan tujuan bisnis dan penggunaan sumber daya luar organisasi atau perusahaan (Utomo, 2012).

Salah satu contoh konsep integrasi yang akan diimplementasikan dalam penelitian ini adalah mengambil konsep layanan pada penggunaan *Web Service* atau dapat disebut WS pada jaringan sistem informasi aktifitas Mahad Al-Jamiah UIN Maulana Malik Ibrahim Malang, yang akan diterapkan pada aplikasi *mobile Smart-Mahad* sebagai sarana media informasi yang lebih cepat dan efektif dalam penunjang aktifitas di dalam Mahad.

Perlu diketahui, mahad Al-Jamiah adalah salah satu unit dalam lembaga kampus UIN Maulana Malik Ibrahim Malang yang memiliki pola pesantren di dalamnya, dikarenakan pesantren adalah sebagai salah satu sistem pendidikan nasional yang berperan penting dalam membangun manusia, khususnya mahasiswa yang mendapat gelar mahasantri yang terdapat di Mahad Al-Jamiah. Dikarenakan mahasiswa adalah komunitas lahirnya ilmuan (ulama') yang mampu menjelaskan ilmu pengetahuannya, ini sesuai dengan Surat At-Taubah ayat 122 dengan lafadz sebagai berikut:

وَمَا كَانَ الْمُؤْمِنُونَ لِيَنْفِرُوا كَافَّةً ۚ فَلَوْلَا نَفَرَ مِنْ كُلِّ فِرْقَةٍ مِنْهُمْ
طَائِفَةٌ لِيَتَفَقَّهُوا فِي الدِّينِ وَلِيُنذِرُوا قَوْمَهُمْ إِذَا رَجَعُوا إِلَيْهِمْ لَعَلَّهُمْ
يَحْذَرُونَ

Yang artinya :

“Tidak sepatutnya bagi mukminin itu pergi semuanya (ke medan perang). Mengapa tidak pergi dari tiap-tiap golongan di antara mereka beberapa orang untuk memperdalam pengetahuan mereka tentang agama dan untuk

memberi peringatan kepada kaumnya apabila mereka telah kembali kepadanya, supaya mereka itu dapat menjaga dirinya” (At Taubah : 122).

Maka dari itu mahasantri khususnya yang ada di Mahad Al-Jamiah UIN Maulana Malik Ibrahim Malang adalah calon-calon ilmuwan yang harus ditunjang dengan fasilitas belajar mengajar yang lebih efektif dengan ditunjang oleh aplikasi *mobile Smart Mahad* yang akan terintegrasi pada layanan Sistem Informasi Aktifitas Mahad UIN Maliki Malang (SIAMA).

Penggunaan konsep layanan integrasi pada sistem informasi aktifitas mahad ini dimaksudkan untuk mempersiapkan model layanan yang lebih besar untuk pengembangan sistem informasi dan sistem bisnis kedepan, dikarenakan perkembangan proses bisnis dan layanan dalam Mahad pusat Al-Jamiah yang kian hari semakin kompleks dan membutuhkan layanan data dari *eksternal* Mahad yang artinya membutuhkan integrasi layanan yang berbeda, juga mengingat perkembangan kampus UIN Maulana Malik Ibrahim Malang yang juga semakin mengembangkan fasilitas dan infrastrukturnya pada pemanfaatan teknologi dan informasi.

Maka dari itu penelitian ini akan berfokus pada mengimplementasikan SOA sebagai metode pengembangan *web service* yang digunakan sebagai sarana pertukaran data antara platform *mobile* dan *web* layanan serta pembuatan aplikasi *Smart-Mahad* yang berbasis *mobile* pada platform android. Dengan adanya aplikasi *mobile* penunjang untuk para staff serta jajarannya *musyirif* dan mahasantri ini diharapkan mampu untuk mempercepat proses belajar para mahasantri, mempercepat arus informasi layanan mahad

serta koordinasi para pengasuh, *murobbi/murobbiah*, staff, dan *musyrif/musyrifah* di Mahad Al-Jamiah UIN Maulana Malik Ibrahim Malang.

1.2. Rumusan Masalah

1. Apakah SOA dapat diterapkan dalam pengembangan aplikasi dan sistem informasi mahad Al-Jamiah UIN Malang?
2. Bagaimana merancang *web service* di mahad dengan mengimplementasikan SOA dalam membagi setiap layanan *service*?
3. Bagaimana membangun aplikasi mobile *Smart-Mahad*?

1.3. Tujuan Penelitian

1. Membuktikan bahwa metode SOA dapat digunakan dalam pengembangan aplikasi dan sistem informasi mahad Al-Jamiah UIN Malang.
2. Merancang *web service* dan mengimplementasikan SOA untuk membagi setiap layanan *service*.
3. Membangun aplikasi mobile *Smart-Mahad* pada platform android.

1.4. Batasan Masalah

Batasan masalah dalam penelitian ini meliputi:

1. Implementasi SOA dimodelkan pada layanan *service mahad*, yaitu untuk layanan *sistem reporting* mahasantri.
2. User dari aplikasi *Smart-Mahad* dikhususkan melakukan *service* untuk para mahasantri, *musyrif* dan pengasuh untuk melihat perkembangan mahasantri selama di Mahad UIN Al-Jamiah.

1.5. Manfaat Penelitian

1. Bagi Mahad Al-Jamiah UIN Maulana Malik Ibrahim Malang, mempersiapkan dan mempermudah untuk integrasi layanan lain yang akan masuk dalam sistem informasi aktifitas Mahad.
2. Bagi pengasuh dan *musyrif*, untuk mempermudah mengelola dan memberikan informasi secara langsung dalam lingkungan *mahad* Al-Jamiah secara *mobile*.
3. Bagi para mahasantri, untuk mempermudah akses dalam melihat info terbaru di *mahad*.

BAB II

STUDI PUSTAKA

Pada bab ini akan dijelaskan lebih rinci tentang dasar teori dan tinjauan studi yang terkait terhadap penelitian yang sedang dilakukan.

2.1. Tinjauan Studi

Penelitian yang dilakukan oleh Oktalifryan, dalam penelitiannya mengenai perancangan aplikasi rawat jalan dengan melakukan pengembangan terintegrasi dengan menggunakan teknologi ESB sebagai *middleware* untuk jembatan integrasi antara modul IRJ (Instalasi Rawat Jalan) dengan sistem yang terdapat di rumah sakit. Dengan kemampuan ESB, maka modul IRJ yang dikembangkan mampu berintegrasi dengan sistem lain walaupun dikembangkan dengan service dan standarisasi yang berbeda (Oktalifryan, 2011).

Sedangkan penelitian yang dilakukan oleh Putra mengenai kriptografi untuk pengamanan sistem rekam medis pada data sensitif di rumah sakit memiliki kesamaan terhadap penelitian yang dilakukan oleh Oktalifian dalam menerapkan integrasi data antara aplikasi rekam medis yang dibangun dengan sistem yang berbeda menggunakan mediasi ESB, dalam penelitiannya dilakukan sebuah pengamanan dengan metode kriptografi pada data-data sensitif rekam medis. (Putra, 2011)

Penelitian selanjutnya yang dilakukan oleh Utomo, dilakukan implementasi ESB untuk melakukan integrasi antar *web service e-commerce* yang berasal dari *service* eBay, Yahoo, Amazon, dan PayPal untuk

membangun aplikasi *e-Shop* dalam membangun *e-commerce* yang terintegrasi. Dalam penelitiannya, penggunaan ESB mampu mengintegrasikan service antar sistem yang berbeda dengan perannya untuk *routing*, transformasi pesan dan protokol (Utomo, 2012).

Penelitian lain mengenai SOA dilakukan oleh Khoirul mengenai sistem informasi apotek yang terintegrasi dengan konsep *collaborative E-commerce*. Pada penelitiannya, Khoirul membangun sebuah aplikasi yang mengintegrasikan sistem antar apotek untuk melakukan order obat. Aplikasi yang tersedia dalam platform *web*, *mobile* (J2ME), dan *desktop*. Namun sistem yang dibangun masih menggunakan tipe integrasi dengan *service point to point*, yang artinya tidak menggunakan ESB sebagai media *routing* dan transformasi data, sehingga akan memiliki kelemahan jika menggunakan standar proses bisnis yang lebih rumit (Khoirul, 2012).

Maka dari itu, setelah mengetahui penelitian terkait mengenai implementasi SOA pada setiap penelitian sebelumnya, dalam penelitian ini peneliti akan menerapkan kontribusi penelitian berupa agregasi pada setiap *service*, *routing* pesan, dan transformasi pesan dari setiap *service* yang diintegrasikan pada layanan SIAMA.

2.2. Tinjauan Pustaka

2.2.1. Service Oriented Architecture (SOA)

Menurut Erl, SOA merupakan arsitektur perangkat lunak yang dibangun menggunakan prinsip-prinsip perancangan berorientasi *service*, sedangkan orientasi *service* merupakan konsep dalam rekayasa perangkat lunak yang merepresentasikan pendekatan berbeda untuk memisahkan

kepentingan (Erl, 2005). Hal ini berarti bahwa fungsionalitas sistem dipecah ke dalam unit logic yang lebih kecil yang dinamakan dengan *service*. *Service-service* ini lepas satu sama lain, tetapi mempunyai kemampuan untuk berinteraksi satu sama lain melalui mekanisme komunikasi tertentu. Karena itu, Erl (2005) mendefinisikan komponen SOA sebagai *service*, *description*, dan *messages*. *Service* berkomunikasi dengan yang lain melalui *message* yang memungkinkan interaksi antara *services*, yang ditetapkan oleh *description*. Dua *service* berkomunikasi satu sama lain yang diacu sebagai *service requestor* dan *service provider*. *Service requestor* adalah *service* yang memanggil *service* lain, sedangkan yang dipanggil disebut *service provider*.

Service sendiri dapat dipandang sebagai enkapsulasi logik dari satu atau sekumpulan aktivitas tertentu. Otomatisasi proses bisnis merupakan sekumpulan aktivitas yang disusun dalam langkah-langkah sebagai implementasi proses bisnis. Setelah seluruh permasalahan dapat dibagi dalam beberapa *service*, solusi dari permasalahan tersebut harus bisa diselesaikan dengan memungkinkan seluruh *service* berpartisipasi dalam sebuah orkestrasi. Untuk itu ada beberapa permasalahan yang harus dimiliki oleh *service* yaitu bagaimana *service* saling berhubungan, bagaimana *service* berkomunikasi, bagaimana *service* didesain, dan bagaimana pesan antar *service* didefinisikan (Erl, 2005).

Pembagian berdasarkan *service* ini sesungguhnya bukan sesuatu yang baru, karena telah banyak diterapkan. Namun hal baru dari pendekatan *service oriented* ini terkait sifat-sifat yang dimilikinya (Erl, 2005), yaitu :

1. *Loosly coupled*, yaitu setiap *service* berdiri sendiri secara independen dan tidak bergantung dengan *service* lain untuk berjalan. Ketergantungan diminimalisir sehingga hanya butuh mekanisme komunikasi satu sama lain.
2. *Service contract*, yaitu setiap *service* memiliki kesepakatan mengenai cara untuk berkomunikasi.
3. *Autonomy*, yaitu *service* memiliki hak penuh terhadap semua logik yang dienkapsulasinya.
4. *Abstraction*, yaitu *service* tidak memperlihatkan bagaimana logik yang diimplementasikan di dalamnya.
5. *Reusability*, yaitu logik dibagi menjadi sekumpulan *service* yang dapat digunakan lagi tanpa membuat ulang.
6. *Statelessness*, yaitu *service* tidak memiliki status tertentu terkait dengan aktivitas yang dilakukannya.
7. *Discoverability*, yaitu *service* didesain secara deskriptif sehingga bisa ditemukan dan diakses melalui mekanisme pencarian tertentu.

SOA terdiri atas sekumpulan *service*. Namun sekumpulan *service* tidak cukup untuk membentuk sebuah arsitektur ini. Menurut (Erl, 2005), SOA terdiri atas empat komponen, yaitu:

1. *Message*, yaitu data yang dibutuhkan untuk menyelesaikan sebagian atau sebuah unit kerja, yang dipertukarkan antara satu *service* dengan *service* yang lain.
2. *Operation*, yaitu fungsi-fungsi yang dimiliki oleh sebuah *service* untuk memproses *message* hingga menghasilkan sesuatu. Fungsi-fungsi inilah yang nantinya akan saling berinteraksi untuk menyelesaikan sebuah unit kerja.

3. *Service*, mempresentasikan sekumpulan *operation* yang berhubungan untuk menyelesaikan sekumpulan unit kerja yang berhubungan.
4. *Process*, merupakan *business rule* yang menentukan operasi mana yang digunakan untuk mencapai tujuan tertentu.

2.2.2. Jenis Integrasi

Menurut Hohpe, *Enterprise Integration* adalah membuat beberapa aplikasi dan platform yang berbeda dapat berjalan bersama dan saling terhubung walaupun dengan lokasi yang berbeda, sehingga dapat menghasilkan kesatuan fungsionalitas (Hohpe, 2003). Juric membagi konsep integrasi menjadi beberapa tipe menurut jenisnya, yaitu : 1) *Data level integration*, 2) *Application integration*, 3) *Businnes process integration*, 4) *Presentation integration* (Juric, 2007).

Pada penelitian ini akan diterapkan jenis integrasi berupa *application integration*, dimana menurut Juric menjelaskan bahwa *application integration* lebih berfokus pada pembagian fungsi-fungsi dan logika bisnis, tidak hanya pada pembagian data seperti pada *data level integration* (Juric, 2012). Dalam integrasi berbasis aplikasi sudah diterapkan penggunaan API untuk komunikasi antar aplikasi. Pada integrasi ini sudah menerapkan fungsi penggunaan ESB untuk proses integrasi, yang akan dibahas lebih lanjut pada bab selanjutnya. Tujuan dari integrasi ini lebih pada pengaksesan fungsionalitas antar aplikasi dan mengatasi perbedaan teknologi yang digunakan, dan digunakan *service* antar aplikasi yang digunakan untuk mengakses setiap antar muka atau *Application Program Interface* (API).

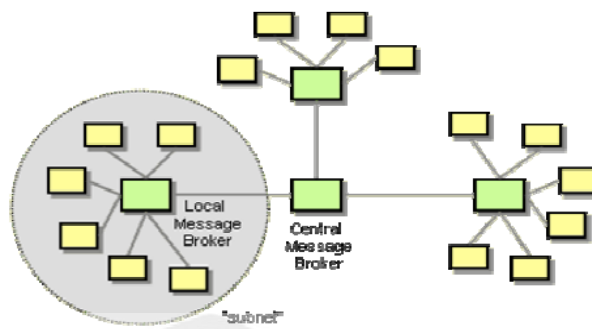
2.2.3. Jenis Arsitektur Integrasi

Menurut Utomo, terdapat beberapa arsitektur integrasi, yaitu 1.) *point to point*, 2.) *hub-and-spoke*, dan 3.) penerapan SOA (Utomo, 2012). Pada sistem integrasi yang banyak diimplementasikan sebelumnya masih banyak melakukan sistem integrasi dengan metode titik per titik (*point to point*), penerapan *point to point* sudah dianggap tidak relevan lagi saat ini, dikarenakan penggunaan metode tersebut akan membuat semakin rumitnya komunikasi antar aplikasi yang berjalan dan komunikasi antar protokol. Ini dikarenakan komunikasi akan berlaku aplikasi per aplikasi dalam satu antarmuka, jika aplikasi masih dalam skala kecil dan tidak memiliki interoperabilitas yang tinggi maka belum menjadi masalah yang besar, namun masalah akan muncul jika banyak aplikasi yang terhubung antar sistem. Menurut Utomo, model integrasi ini memiliki kelemahan pada sulitnya untuk memperluas integrasi dan sulitnya pemeliharaan karena seperti yang sudah dijelaskan sebelumnya, terdapat kompleksitas dari permodelan ini (Utomo, 2012). Untuk jumlah antarmuka yang dihasilkan jika menerapkan permodelan ini adalah $N(N-1)/2$, artinya jika akan melakukan integrasi pada 10 aplikasi, maka memerlukan 45 antarmuka, dan untuk integrasi 200 aplikasi, maka akan membutuhkan 19900 antarmuka. Dapat dibayangkan sulitnya melakukan perawatan maupun pengembangan dan modifikasi aplikasi jika menerapkan integrasi berbasis *point to point*. Gambaran konsep integrasi ini menurut Siriwardena adalah seperti gambar 2.1 berikut yang dapat disebut *Spaghety Integration Dilema*:



Gambar 2.1 *Spaghety Integration Dilema* (Siriwardena, 2012)

Model integrasi yang kedua adalah *hub and spoke*, jika di analogikan tentang dunia komunikasi dan jaringan komputer, untuk menghubungkan setiap komputer, dapat digunakan sebuah *hub*, dimana tugas sebuah *hub* adalah pemisah atau pembagi dalam sebuah jaringan. Fungsi ini sama dengan model integrasi *hub and spoke*, dimana pada model integrasi ini memiliki kemiripan dengan model integrasi *point to point*, namun terdapat *hub* atau penghubung diantara tiap aplikasi. Seluruh proses *routing* dan transformasi data dilakukan pada *hub* ini. Dikarenakan proses lalu lintas dilakukan pada *hub* ini, maka tentunya akan mengurangi jumlah antarmuka koneksi yang diperlukan pada tiap aplikasi. Maka dari itu *hub* menurut Hohpe disebut juga sebagai *central message-broker*, karena mengirimkan setiap pesan pada setiap anggota dari tiap aplikasi yang terintegrasi pada *hub* tersebut sesuai dengan alamat tujuannya masing-masing (Hohpe, 2003). Kelebihan lainnya adalah kekacauan jika menggunakan metode *point to point* dapat dihilangkan, seperti hubungan komunikasi yang bertumpu pada setiap aplikasi yang dapat membuat rumitnya pengaturan integrasi tidak akan ditemukan lagi jika menggunakan *hub*. Berikut gambaran dari integrasi *hub and spoke* pada gambar 2.2 berikut :



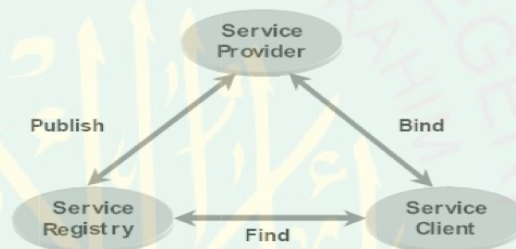
Gambar 2.2 *Central Message Broker* yang memiliki arsitektur *hub and spoke* (Juric, 2007)

Meskipun cara ini dianggap sudah mampu mengurangi kesulitan dan kompleksitas dari metode *point to point*, namun masih terdapat kelemahan pada metode ini. Menurut Utomo, kelemahan dari metode ini berada pada sifat dari *hub* itu sendiri yang terpusat, jika *hub* mengalami kegagalan dalam sistem, maka semua proses integrasi antar aplikasi akan mengalami kegagalan. Selain itu kunci dari teknologi ini juga masih dimiliki oleh *vendor* penyedia layanan integrasi (Utomo, 2012).

Pada model integrasi selanjutnya diperkenalkan konsep integrasi SOA, konsep ini hadir dikarenakan untuk melengkapi metode integrasi yang sudah ada sebelumnya, yaitu konsep *point to point* dan *hub and spoke*. Menurut Erl, SOA adalah sebuah bentuk teknologi arsitektur yang mengikuti prinsip-prinsip berorientasi layanan (Erl, 2005). Konsep berorientasi pada layanan ini dilakukan dengan pendekatan membagi setiap proses dan masalah menjadi sekumpulan layanan, setelah seluruh layanan tersebut dapat dibagi menjadi per bagian, seluruh layanan tersebut harus mampu saling berpartisipasi untuk melakukan integrasi dalam sebuah orkestrasi. Ini dilakukan untuk menjawab beberapa permasalahan mengenai kesesuaian teknologi, dan perbedaan penerapan sistem bisnis yang berbeda antar

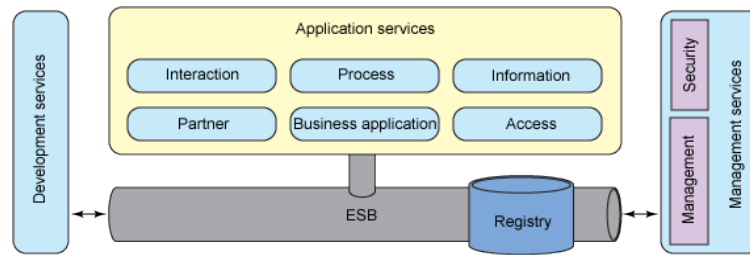
organisasi, yang mana pada setiap organisasi memiliki perbedaan dengan bermacam-macam *platform* sistem dan arsitektur yang berbeda dalam penerapan teknologi. Sehingga jika suatu organisasi maupun perusahaan akan melakukan pertukaran data untuk saling melengkapi pada tujuan bisnisnya, ada jembatan di antara keduanya yang ditawarkan oleh konsep SOA ini. Integrasi yang ditawarkan dalam penelitian Utomo adalah penerapan ESB sebagai pilar dari teknologi SOA.

Arsitektur dasar SOA menurut Erl secara umum dapat digambarkan pada gambar 2.3 berikut :



Gambar 2.3 Service Oriented Architecture (Erl, 2005)

Service client akan berperan sebagai media penyedia layanan yang akan digunakan sebagai data *response* dari *client request*. Layanan ini berperan sebagai penyedia layanan yang sudah didaftarkan deskripsi layanannya di dalam pendaftaran layanan (*service registry*). *Service registry* akan meminta *service* pada penyedia layanan (*service provider*), sehingga layanan akan terikat satu sama lain.



Gambar 2.4 Arsitektur ESB (IBM developer networks)

Dikarenakan konsep yang ditawarkan oleh SOA adalah integrasi berbasis pada layanan, sehingga pada metode ini perubahan-perubahan proses bisnis yang mungkin terjadi pada perusahaan dan organisasi dapat dirubah dengan mudah dikarenakan setiap proses bisnis akan dipecah menjadi bagian kecil menjadi sebuah layanan, seperti yang sudah dibahas pada bagian sebelumnya pada *application integration* dan *business process integration*. WS atau *web service* yang digunakan untuk melakukan pertukaran data juga memiliki kelemahan dalam menangani ketidaksesuaian protokol dan ketidaksesuaian data.

Maka dari itu, penelitian yang dilakukan oleh Utomo menggunakan ESB untuk orkestrasi *web service*, dimana fungsi ESB mampu melakukan *routing*, tranformasi protokol maupun transformasi pesan dalam penelitiannya. Dengan penggunaan ESB dalam metode integrasinya, mampu mengatasi masalah kompleksitas infrastruktur IT dari berbagai aplikasi yang heterogen yang dibangun dari macam-macam *platform* yang berbeda dan protokol yang berbeda juga (Utomo, 2012). Pada Gambar 2.4 adalah ilustrasi ESB sebagai *middleware*.

2.2.4. *Web Services*

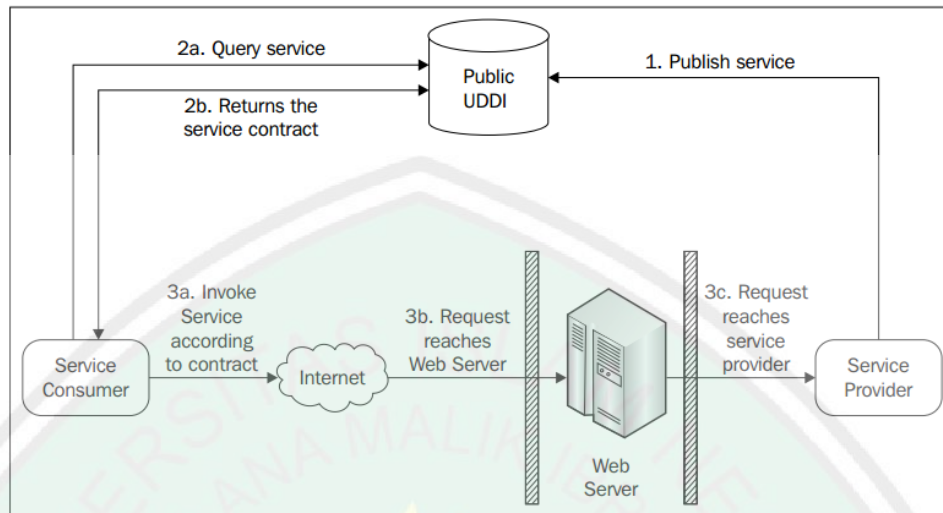
Web services adalah sebuah teknologi yang memungkinkan untuk membuat aplikasi yang independen terhadap *platform* pengembangan. *Web services* dapat dikembangkan dengan beragam bahasa dan beragam *platform* yang mengikuti standar teknologi. Sebuah *Web services* merupakan suatu aplikasi yang membuka kode fungsionalitas aplikasi ke beberapa aplikasi (Arora dan Kishore, 2002).

Web service adalah sepotong logika bisnis, yang dapat diakses melalui internet dengan standar protocol seperti HTTP atau SMTP. *Web service* merupakan sebuah teknologi yang dapat digunakan untuk mengimplementasikan layanan (Salter-Jennings, 2008). Menurut Hwang et al (2008), WS telah menjadi *standard* untuk melakukan ekspose fungsi dari aplikasi bisnis, WS menjadi kerangka untuk pengembangan aplikasi menggunakan SOA. WS secara efektif mampu mengintegrasikan beberapa WS kedalam sebuah komposit atau aplikasi ESB.

WS telah banyak digunakan untuk membangun aplikasi berbasis SOA. Hal ini yang menyebabkan WS identik dengan SOA. Atas dasar ini Erl (2005) menyebutkan sebagai *Contemporary SOA*. *Contemporary SOA* merupakan SOA yang menggunakan WS dan XML dalam implementasinya. Untuk selanjutnya, yang disebut dengan SOA adalah *Contemporary SOA*.

Menurut W3C *Web service* merupakan aplikasi perangkat lunak yang diidentifikasi melalui URI yang antarmuka dan binding nya mampu diidentifikasi, dideksripsikan dan ditemukan melalui XML dan mendukung

interaksi langsung dengan perangkat lunak lain menggunakan pesan berbasis XML melalui protocol berbasis internet atau HTTP.



Gambar 2.5 Arsitektur Web Service (Juric, 2007)

Komponen *Web Service* terdiri dari 3 yaitu SOAP, WSDL, dan UDDI :

1. SOAP (Simple Object Access Protocol)

Dalam pertukaran pesan dalam WS diperlukan suatu standard format pesan antara peminta layanan (*requestor*) dengan penyedia layanan (*provider*). SOAP adalah format pesan yang digunakan untuk komunikasi tersebut. SOAP mendefinisikan format pembungkusan pesan dalam WS. Dalam SOAP dengan XML mengandung tiga elemen yakni *Envelope*, *Header* dan *Body*.

Elemen *Envelope* sebagai surat yang ditulis dalam dokumen XML yang berisi *Header (optional)* dan *Body (required)* sebagai *SOAP Message*. Elemen Header merupakan fitur tambahan pada SOAP, biasanya pada elemen header ini disematkan sebuah kunci/sandi atau *token*. Elemen body berisikan deskripsi pesan yang akan dikirim dalam bentuk *request message*

dan akan dikembalikan dalam bentuk *response message*. Komunikasi ini berjalan dalam protocol HTTP pada jaringan computer.

2. WSDL (Web Services Description Language)

WSDL adalah sebuah format XML untuk mendeskripsikan *service* yang berada pada jaringan computer sebagai sebuah *endpoint* yang di dalamnya terdapat dokumen, fungsi atau prosedur sebuah proses aplikasi. Dengan mengacu pada dokumen WSDL aplikasi lain akan bisa mengetahui operasi-operasi apa saja yang bisa dijalankan dan data apa saja yang bisa didapatkan.

Dalam definisi tersebut termasuk juga deskripsi layanan dan fungsi-fungsi yang disediakan oleh WS.

a. Types

Elemen ini mendefinisikan tipe data yang terdapat dalam pesan yang dipertukarkan sebagai bagian dari layanan. Type yang diacu dalam elemen dokumen pesan WSDL didefinisikan dalam elemen type dokumen WSDL.

b. Message

Elemen ini mendefinisikan pesan atau data dari layanan yang akan dikomunikasikan. Dokumen WSDL mempunyai elemen pesan yang tiap pesannya dipertukarkan.

c. Port Type

Elemen ini menetapkan secara abstrak sekumpulan operasi yang merupakan bagian dari *service*. Dalam dokumen WSDL mempunyai satu atau lebih definisi *Port Type*.

d. Binding

Elemen ini menjelaskan spesifikasi protokol dan format pesan untuk jenis *port type* tertentu

e. Port

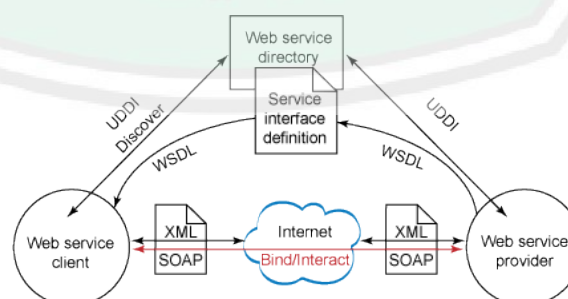
Elemen ini menjelaskan sebuah *endpoint* didefinisikan sebagai sebuah kombinasi dari sebuah binding dan alamat jaringan

f. Service

Elemen ini merupakan koleksi dari serangkaian *endpoint*.

3. UDDI Registry (Universal Description, Discovery and Integration)

Menurut W3C UDDI adalah kerangka platform independen yang berfungsi untuk mendeskripsikan, menemukan serta mengintegrasikan *service* bisnis. UDDI adalah spesifikasi standar untuk pendaftaran suatu WS. UDDI mendefinisikan metadata dan protocol yang digunakan untuk mengetahui dan mengubah informasi dari suatu WS. Langkah pertama dalam menemukan WS adalah dengan meminta alamat tempat penyimpanan dari WS yang akan dipakai yang biasa disebut dengan direktori. Setelah menemukan direktori, peminta WS dapat mengirimkan permintaan lagi untuk mendapatkan informasi detail tentang WS.



Gambar 2.6 Arsitektur UDDI

Tempat untuk menyimpan UDDI dibagi kedalam tiga cara, yakni:

1. *Public UDDI*

Dianggap sebagai *resource* bagi WS berbasis internet. Salah satu contohnya adalah *UDDI Business Registry* (UBR) yang dimiliki oleh grup seperti IBM, Microsoft, SAP

2. *Intra Enterprise UDDI*

Merupakan tempat penyimpanan yang *private* yang menyediakan layanan bagi kalangan internal suatu perusahaan.

3. *Inter Enterprise UDDI*

Merupakan repository yang dapat diakses oleh perusahaan yang merupakan partner dalam suatu bisnis.

Seperti yang telah dijelaskan terdahulu, penemuan layanan memainkan peranan yang penting dalam SOA. Banyak cara yang dapat dipakai untuk mekanisme ini, tapi dalam teknologi WS, UDDI menyediakan standard yang baik dan fleksibel untuk penemuan WS.

2.2.5. Skala Likert

Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Empat skala pilihan juga kadang digunakan untuk kuesioner skala Likert yang memaksa orang memilih salah satu kutub karena pilihan “netral” tak tersedia. Dalam membuat skala Likert, ada beberapa langkah prosedur yang harus dilakukan peneliti (Nazir, 2005), antara lain:

1. Peneliti mengumpulkan item-item yang cukup banyak, memiliki relevansi dengan masalah yang sedang diteliti, dan terdiri dari item yang cukup jelas

disukai dan tidak disukai. Kemudian item-item itu dicoba kepada sekelompok responden yang cukup representatif dari populasi yang ingin diteliti.

2. Responden di atas diminta untuk mengecek tiap item, apakah ia menyenangkan (+) atau tidak menyukainya (-). Respons tersebut dikumpulkan dan jawaban yang memberikan indikasi menyenangkan diberi skor tertinggi. Tidak ada masalah untuk memberikan angka 5 untuk yang tertinggi dan skor 1 untuk yang terendah atau sebaliknya. Yang penting adalah konsistensi dari arah sikap yang diperlihatkan. Demikian juga apakah jawaban “setuju” atau “tidak setuju” disebut yang disenangi, tergantung dari isi pertanyaan dan isi dari item-item yang disusun.

3. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti:

a. Pertanyaan Positif (+)

Skor 1. Sangat (tidak setuju/buruk/kurang sekali)

Skor 2. Tidak (setuju/baik/) atau kurang

Skor 3. Netral / Cukup

Skor 4. (Setuju/Baik/suka)

Skor 5. Sangat (setuju/Baik/Suka)

b. Pertanyaan Negatif (-)

Skor 1. Sangat (setuju/Baik/Suka)

Skor 2. (Setuju/Baik/suka)

Skor 3. Netral / Cukup

Skor 4. Tidak (setuju/baik/) atau kurang

Skor 5. Sangat (tidak setuju/buruk/kurang sekali)

4. Total skor dari masing-masing individu adalah penjumlahan dari skor masing-masing item dari individu tersebut.
5. Respon dianalisis untuk mengetahui item-item mana yang sangat nyata batasan antara skor tinggi dan skor rendah dalam skala total. Misalnya, responden pada upper 25% dan lower 25% dianalisis untuk melihat sampai berapa jauh tiap item dalam kelompok ini berbeda. Item-item yang tidak menunjukkan beda yang nyata, apakah masuk dalam skoring tinggi atau rendah juga dibuang untuk mempertahankan konsistensi internal dari pertanyaan.

BAB III

METODE PENELITIAN

Pada bab ini akan dipaparkan mengenai perancangan sistem yang akan dibangun untuk membuat *back end web service* beserta tampilan aplikasi *mobile* yang akan dibangun sesuai dengan teori yang sudah dideskripsikan pada bab sebelumnya.

3.1. Analisis Kebutuhan Sistem

Pada fase analisis kebutuhan sistem akan dilakukan analisis terkait seluruh proses bisnis yang terkait pada sistem informasi aktifitas mahad Al-Jamiah UIN Malang. Analisa pada fase ini berkaitan dengan fungsionalitas sistem yang nantinya akan dijabarkan dengan diagram UML (*Unified Modeling Language*).

Penjelasan fungsionalitas sistem SIAMA akan berfokus pada beberapa proses alur sistem saja, yaitu proses bisnis untuk aktifitas mahasantri, *musyrif*, dan pengasuh. Ini dikarenakan banyaknya proses bisnis yang ada di dalam SIAMA yang sudah berjalan dan akan lebih diutamakan untuk proses yang memiliki aktifitas tertinggi dan membutuhkan mobilitas yang dapat diakses melalui fasilitas *mobile*. Berikut adalah daftar proses bisnis yang terdapat pada SIAMA (Putra, 2015) :

1. Proses manajemen seluruh user yang terlibat dalam sistem informasi aktifitas mahad al-jamiah.
2. Proses manajemen data *mahad* pada bagian ksantrian.

3. Proses manajemen data *musyrif* pada bagian kesantrian.
4. Manajemen data santri, manajemen pendaftaran *musyrif*, dan manajemen fasilitas yang didapatkan santri.
5. Manajemen pelaporan pengasuh, dimaksudkan untuk melihat data laporan *musyrif* dan penempatan *musyrif*.

3.2. Perancangan Sistem

Pada tahap perancangan sistem, rancangan berdasarkan analisis kebutuhan yang telah dipaparkan sebelumnya.

3.2.1. Model Layer Aplikasi

Pada aplikasi berikut terdapat beberapa layer aplikasi yang saling terhubung, secara permodelan maka akan digambarkan seperti berikut :



Gambar 3.1 Model layer aplikasi untuk sistem integrasi *smart-mahad*

3.2.2. Aplikasi Smart Mahad

Pada layer ini merepresentasikan interaksi *user* dan aplikasi, dimana terdapat *platform mobile* dan portal web SIAMA. Pada layer ini akan dilakukan proses penyajian informasi dari sumber data yang didapat dari

layer integrasi yang sudah dijabarkan di bab sebelumnya. Untuk *platform mobile* yang akan dibangun menggunakan android dengan dukungan API minimal level 17 (*Jelly Bean*). Aplikasi yang akan dibuat akan melakukan penyajian informasi *reporting* data mahasantri.

3.2.3. Layer Integrasi

Pada tahap ini menggambarkan departemen yang melakukan pertukaran dan integrasi data. Dimana kemungkinan akan ditemui perbedaan bahasa pemrograman, protokol, dan database yang berbeda antar departemen. Maka dari itu pada layer ini akan diimplementasikan SOA.

3.2.4. Layer Database

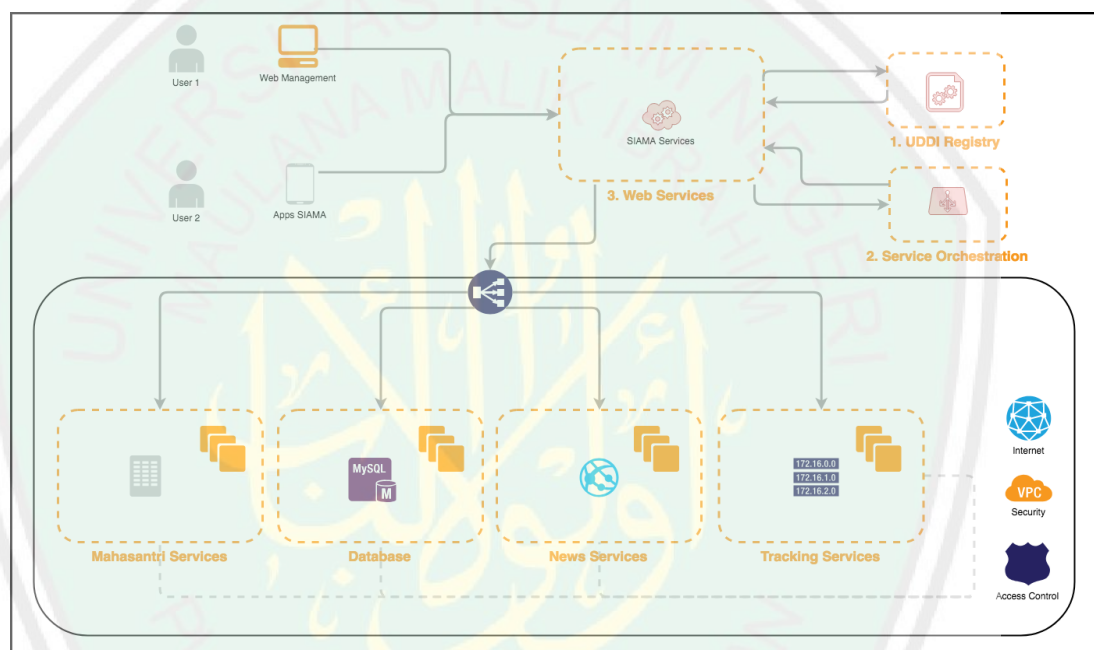
Layer *database* merepresentasikan seluruh data yang disimpan dalam proses integrasi pada layer sebelumnya, dikarenakan perbedaan database pada setiap *service* yang berbeda pada layer sebelumnya, sehingga data yang diperoleh perlu disimpan terlebih dahulu untuk dikonsumsi oleh *web service* sebagai *end point* dari aplikasi *smart mahad*.

3.2.5. Layer Infrastruktur Jaringan

Pada layer ini terdapat komponen jaringan yang digunakan dan infrastruktur penggunaan protokol yang digunakan, sistem operasi yang digunakan, format XML dan JSON yang akan digunakan sesuai dengan standar yang telah didefinisikan, dan perangkat keras untuk mendukung layer-layer sebelumnya.

3.2.6. Arsitektur Sistem

Seperti yang sudah dipaparkan pada bab sebelumnya mengenai dasar arsitektur mengenai SOA dari definisi dari penelitian yang dilakukan Erl, maka dapat dibuat sebuah gambaran arsitektur SOA *web service* untuk layanan platform *mobile* yang akan digunakan di Mahad, gambaran tersebut dapat dilihat pada Gambar 3.2 berikut :



Gambar 3.2 Model SOA Aplikasi Smart Mahad

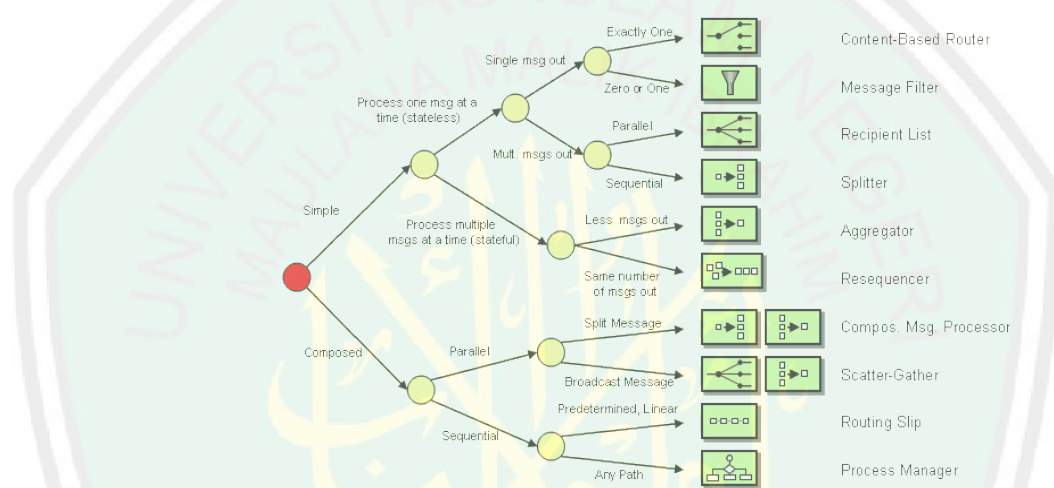
Mengacu pada gambar 3.2, berikut paparan lebih lanjut mengenai *service registry*, *service orchestration*, dan *web service* yang akan di implementasikan.

1. UDDI (Universal Discovery Description Integration) / Service Registry

Seperti yang sudah dijelaskan pada bab sebelumnya mengenai UDDI. Maka setiap *service* yang akan masuk ke dalam sistem aplikasi *Smart-Mahad* akan didaftarkan pada UDDI terlebih dahulu.

2. Service Orchestration

Orkestrasi layanan ini dapat dilakukan dengan bantuan sebuah *message routing*. *Message routing* bertanggung jawab untuk mengubah dan memperkaya pesan yang diperlukan oleh layanan dan berkomunikasi dengan berbagai layanan sesuai dengan logika bisnis yang diterapkan. Proses *message routing* dapat dilihat pada gambar 3.3 dibawah ini.



Gambar 3.3 Ilustrasi message routing (EAI Patterns)

3. Smart-Mahad Web Service

Service yang berlaku pada penelitian ini adalah mahad *activity service*, *mahasantri tracking service*, dan *news service* dari web resmi mahad UIN Malang. Berikut keterangan dari setiap *service* tersebut :

Tabel 3.1 Tabel *web service* sistem integrasi SIAMA

No	Nama <i>service</i>	Departemen	Keterangan
1.	<i>Mahad activity service</i>	Mahad	Mengambil dan mengirim data pada aplikasi sistem informasi aktifitas di Mahad Al-Jamiah.

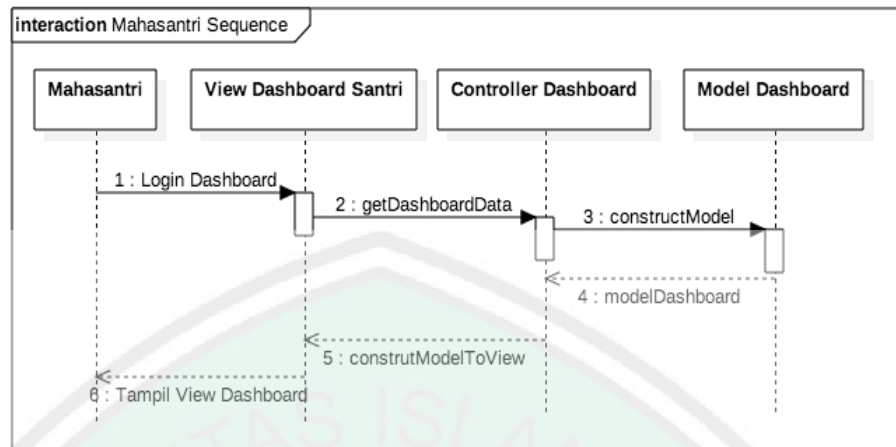
2.	Mahasantri <i>tracking service</i>	Mahad	Mengirim data mengenai <i>tracking</i> mahasantri yang diperlukan service lain dari aplikasi <i>tracking</i> dari penelitian tim Mahad.
3.	Mahad <i>news service</i>	Mahad	Mengambil data <i>newsfeed</i> pada web portal mahad untuk ditampilkan pada aplikasi.

3.2.7. Sequence Diagram

Diagram *sequence* digunakan untuk melihat proses pengiriman dan penerimaan data yang terjadi pada sistem, berikut *sequence* diagram yang terjadi pada SIAMA :

1. Sequence Diagram Untuk Mahasantri

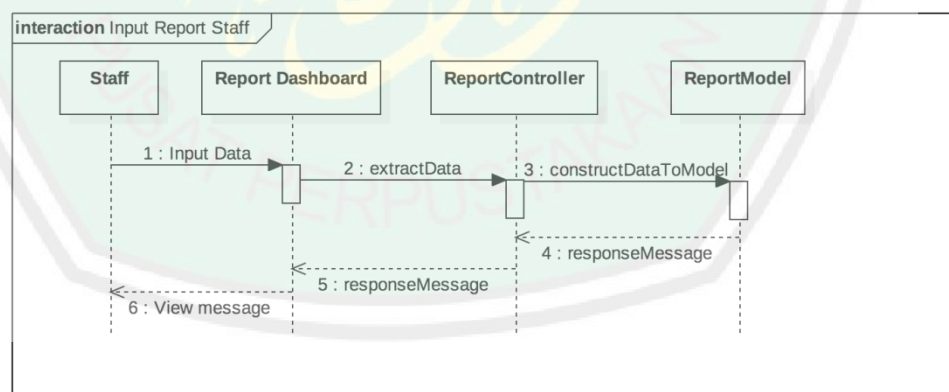
Pada gambar 3.4 adalah gambaran umum untuk proses dan gambaran umum mahasantri untuk melihat halaman utama pada aplikasi, proses ini mewakili secara keseluruhan dalam penyajian data untuk tipe user mahasantri, yaitu *interactor view* akan melakukan request ke *controller*, dimana fungsi pemanggilan *controller* dicontohkan sebagai *getDashboardData()*, lalu *controller* akan melakukan proses *request data* pada *web service* dan mendapatkan kembalian data sesuai model data yang ditentukan, pada proses ini data data akan di olah pada *method constructModel()*, setelah itu hasil permodelan data akan dikembalikan ke *controller* untuk disajikan ke dalam *view* dengan *method constructModelToView()* sehingga *view* akan menyajikan data yang telah dimodelkan ke dalam bentuk *friendly user interface*.



Gambar 3.4 *Sequence diagram* aktifitas mahasantri

2. *Sequence diagram* untuk staff mahad/admin

Pada gambar 3.5 adalah proses input data *reporting* dalam sistem dari admin ataupun staff pada mahad untuk dengan memanggil fungsi *extractData()*, setelah itu data akan di modelkan dengan memanggil fungsi *constructDataToModel()* pada *ReportService* dan untuk melakukan penyimpanan pada database.



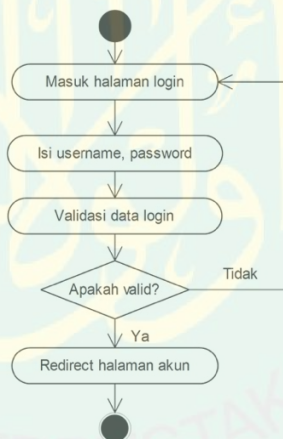
Gambar 3.5 *Sequence diagram* staff mahad

3.2.8. Activity Diagram

Activity diagram digunakan untuk menggambarkan aktivitas yang terjadi pada sebuah sistem. Berikut adalah diagram yang digunakan pada sistem aplikasi mahad Al-Jamiah.

1. Diagram activity login

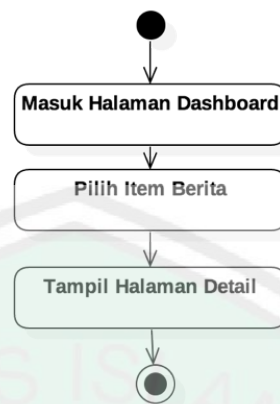
Gambar 3.6 menunjukkan proses *login*. *User* harus memasukkan data *credential* yang diperlukan. Setelah itu akan dilakukan pengecekan *login* dalam database. Apabila *login valid* maka *user* akan masuk ke halaman *dashboard* utama aplikasi. Jika login tidak valid, maka user akan diberikan informasi login gagal dan user akan tetap berada di halaman login.



Gambar 3.6 Diagram activity login

2. Diagram activity news feed

Gambar 3.7 menunjukkan proses melihat *news feed update* pada mahad yang diambil melalui *scrapping* halaman web mahad. Setiap *user* yang masuk pertama kali pada aplikasi, halaman utama yang akan ditampilkan adalah tampilan *news feed*.



Gambar 3.7 Diagram *activity news feed*

3. Diagram *activity report*

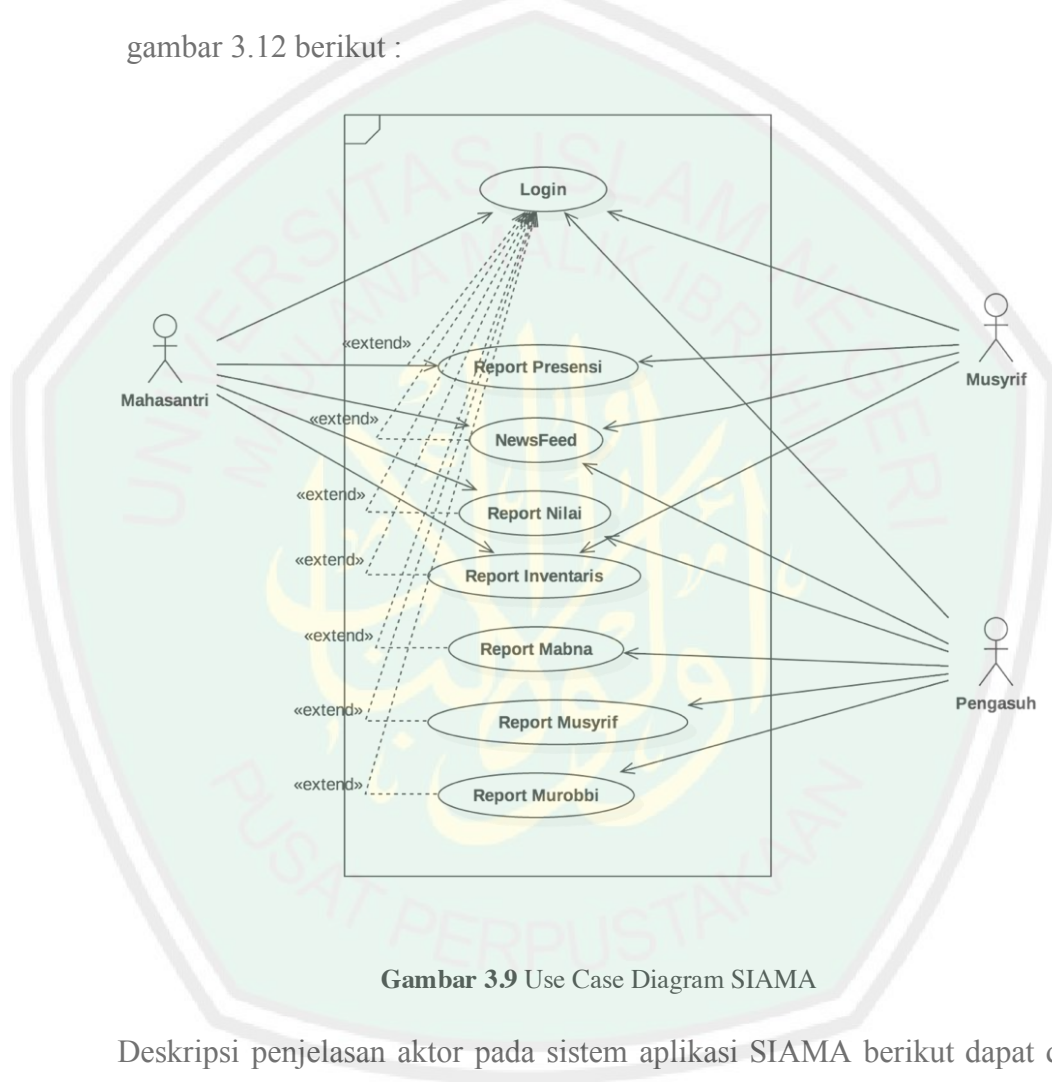
Gambar 3.8 menunjukkan proses reporting data mahasiswa. *User* akan masuk ke halaman *dashboard* aplikasi, lalu pilih menu yang terdapat pada aplikasi dengan berbagai macam pilihan filter sesuai dengan pilihan yang dipilih untuk disajikan kepada user.



Gambar 3.8 Diagram *activity reporting*

3.2.9. Use Case Diagram

Diagram *use case* menggambarkan interaksi antara *stakeholder* dengan *sistem* yang dijalankan berdasarkan proses bisnis yang telah dijelaskan pada analisis sistem diatas. Seperti yang akan dijelaskan pada gambar 3.12 berikut :



Gambar 3.9 Use Case Diagram SIAMA

Deskripsi penjelasan aktor pada sistem aplikasi SIAMA berikut dapat dilihat pada tabel 3.2. di bawah ini.

Tabel 3.2 Tabel *actor use case* SIAMA

No	Aktor	Deskripsi
----	-------	-----------

1	Mahasantri	User yang mengunjungi untuk melihat data reporting nilai, presensi, laporan inventaris kamar santri dan bagian kemusyrifan.
2	Musyrif	User yang melihat reporting mahasantri dan menkoordinir inventaris pada mabna.
3	Pengasuh	User yang dapat melihat seluruh reporting yang terdapat dalam sistem.

Selanjutnya, pendefinisian *activity* dijelaskan secara detail pada tabel 3.3.

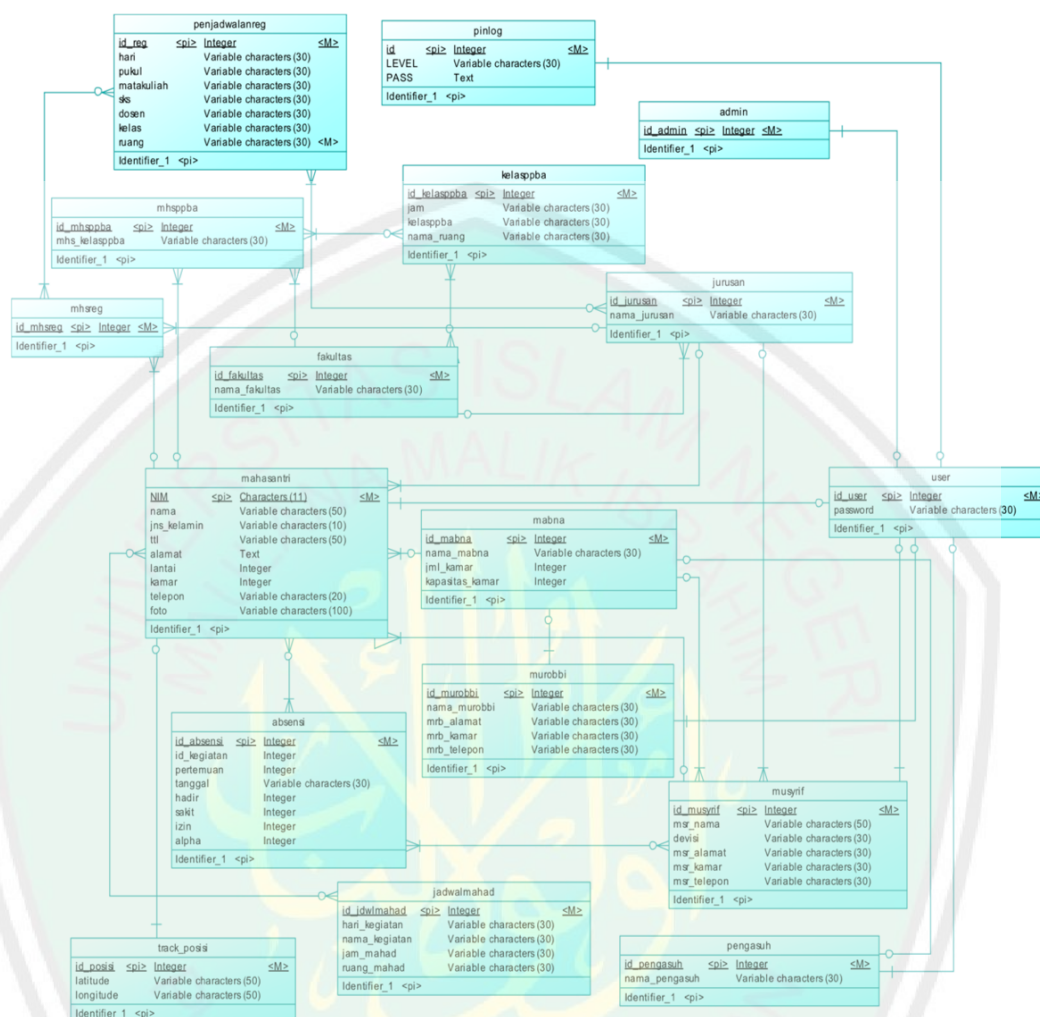
Tabel 3.3 Definisi *activity*

No	Nama Activity	Deskripsi
1	Melihat <i>news feed</i>	Menampilkan halaman <i>news feed</i> berita, atau <i>update</i> terbaru seputar mahad.
2	Login	Menampilkan halaman dan form untuk <i>login user</i>
3	Melihat <i>report</i> nilai ujian	Melihat <i>report</i> penilaian, baik nilai ujian UTS maupun UAS.
4	Melihat <i>report</i> inventaris	Menampilkan halaman <i>report</i> inventaris barang-barang yang berkaitan dengan mabna dan kamar santri.
6	Melihat <i>report</i> mabna	Menampilkan halaman <i>report</i> mabna yang berkaitan dengan presensi kehadiran, beserta mahsantri berprestasi.
7	Melihat <i>report</i> musyrif	Menampilkan halaman <i>report musyrif</i> berupa pengumuman penerimaan <i>musyrif</i> dan hasil ujian penerimaan <i>musyrif</i> .
8	Melihat <i>reportmurabbi</i>	Menampilkan halaman <i>reportmurabbi</i> berupa pembagian jadwal dan jurnal-jurnal <i>murabbi</i> .

3.2.10. Perancangan Basis Data

Pada bagian ini akan ditampilkan perancangan basis data untuk aplikasi *activity* mahad berupa *Conceptual Data Model* (CDM) maupun *Physical Data Model* (PDM).

1. Conceptual Data Model



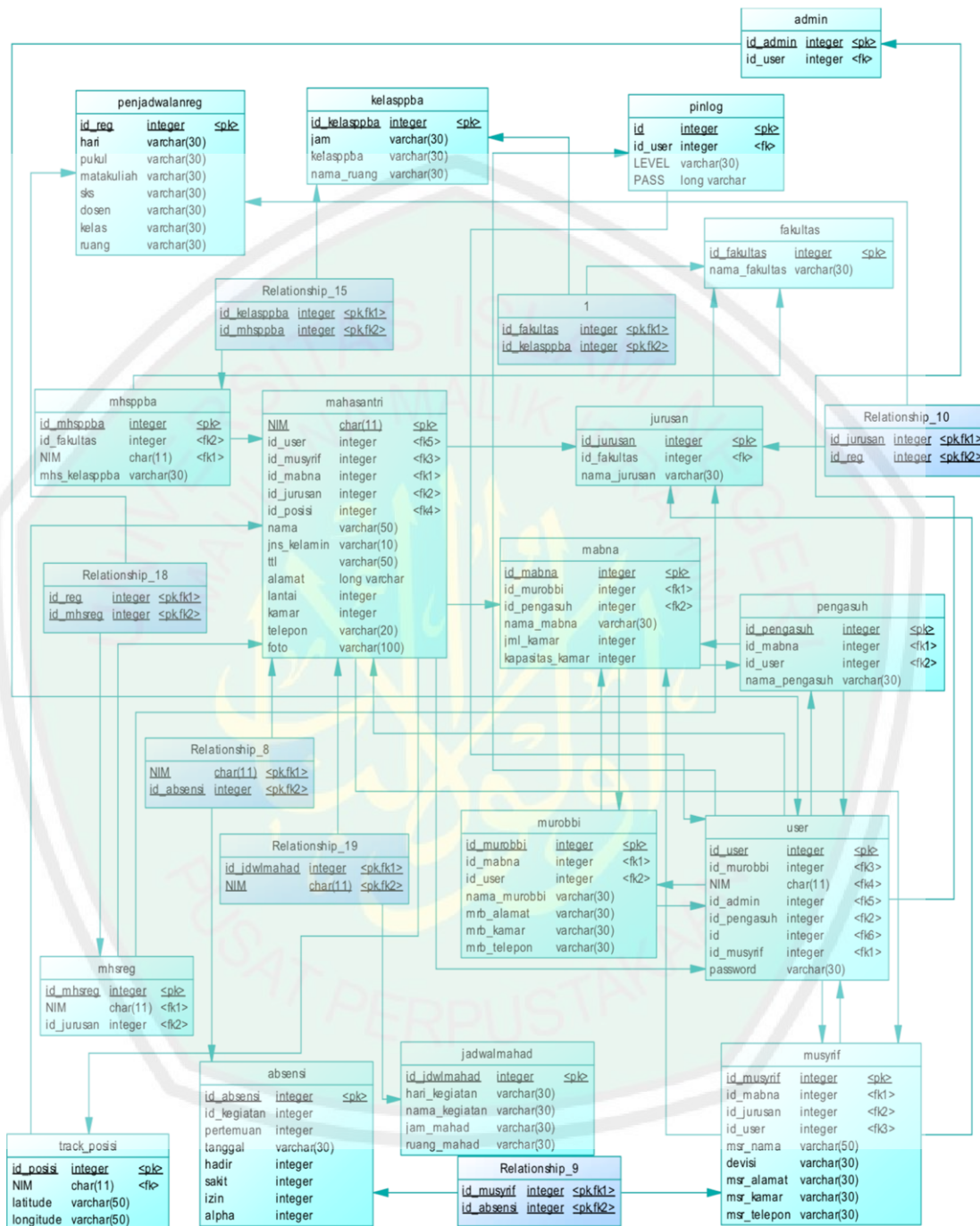
Gambar 3.13 Conceptual Data Model SIAMA

Tabel 3.4 Keterangan CDM

No	Nama Tabel	Keterangan
1	Admin	Tabel berisi data user admin
2	User	Tabel berisi data user
3	Pinlog	Tabel berisi data login user
4	fakultas	Tabel berisi data fakultas

No	Nama Tabel	Keterangan
5	jurusan	Tabel berisi data jurusan
6	mahasantri	Tabel berisi data detail mahasantri
7	musyrif	Tabel berisi data musyrif/musyrifah
8	murobbi	Tabel berisi data Murobbi/murobbiyah
9	pengasuh	Tabel berisi data pengasuh
10	Mabna	Tabel berisi data mabna
11	Absensi	Tabel berisi data absensi mahasantri
12	Jadwalmahad	Tabel berisi data jadwal kegiatan ma'had
13	penjadwalanreg	Tabel berisi data penjadwalan kuliah regular semua jurusan
14	kelasppba	Tabel berisi data penjadwalan PPBA
15	Mhsreg	Tabel berisi data jadwal regular mahasantri
16	Mhspgba	Tabel berisi data jadwal PPBA mahasantri
17	track_posisi	Tabel berisi data lokasi mahasantri

2. Physical Data Model (PDM)



Gambar 3.14. Physical Data Model SIAMA

3.2.11. Struktur Basis Data

Dalam perancangan basis data aplikasi mahad ini, *database* disimpan dengan nama *siama_mahad*. Berikut adalah daftar tabel pada aplikasi ini beserta detail *field* tabelnya :

1. Tabel admin

Tabel 3.5 Tabel detail admin

No	Field	Tipe data	Key
1	id_admin	Int	Primary key
2	id_user	Int	

2. Tabel user

Tabel 3.6 Tabel detail user

No	Field	Tipe data	Key
1	id_user	Int	Primary key
2	pass	Text	
3	level	Varchar(30)	

3. Tabel pinlog

Tabel 3.7 Tabel pinlog

No	Field	Tipe data	Key
1	id_user	Int	Foreign key
2	pass	Text	
3	level	Varchar(30)	

4. Tabel fakultas

Tabel 3.8 Tabel fakultas

No	Field	Tipe data	Key
1	id_fakultas	Int	Primary key
2	nama_fakultas	Varchar(30)	

5. Tabel jurusan

Tabel 3.9 Tabel jurusan

No	Field	Tipe data	Key
1	id_jurusan	Int	Primary key
2	id_fakultas	Int	Foreign key
3	nama_jurusan	Varchar(30)	

6. Tabel mahasantri

Tabel 3.10 Tabel mahasantri

No	Field	Tipe data	Key
1	nim	Int	Primary key
2	id_mabna	Int	Foreign key
3	id_user	Varchar(30)	Foreign key
4	id_jurusan	Int	Foreign key
5	nama	Varchar(50)	
6	jenis_kelamin	Varchar(10)	
7	ttl	Varchar(50)	

No	Field	Tipe data	Key
8	alamat	Text	
9	lantai	Int	
10	telepon	Varchar(20)	
11	kamar	Int	
12	id_musyrif	Int	Foreign key
13	foto	Varchar(20)	

7. Tabel musyrif

Tabel 3.11 Tabel musyrif

No	Field	Tipe data	Key
1	id_musyrif	Int	Primary key
2	id_mabna	Int	Foreign key
3	id_user	Int	Foreign key
4	id_jurusan	Int	Foreign key
5	nama	Varchar(50)	
6	jenis_kelamin	Varchar(10)	
7	alamat	Text	
8	telepon	Varchar(20)	
9	kamar	Int	
10	divisi	Int	

8. Tabel murobbi

Tabel 3.12 Tabel murobbi

No	Field	Tipe data	Key
1	id_murobbi	Int	Primary key
2	id_mabna	Int	Foreign key
3	id_user	Int	Foreign key
4	nama	Varchar(50)	
5	alamat	Text	
6	telepon	Varchar(20)	
7	kamar	Int	

9. Tabel pengasuh

Tabel 3.13 Tabel pengasuh

No	Field	Tipe data	Key
1	id_pengasuh	Int	Primary key
2	id_user	Int	Foreign key
3	nama	Varchar(50)	

10. Tabel mabna

Tabel 3.14 Tabel mabna

No	Field	Tipe data	Key
1	id_mabna	Int	Primary key
2	nama_mabna	Varchar(50)	

No	Field	Tipe data	Key
3	jumlah_kamar	Int	
4	kapasitas_kamar	Int	

11. Tabel absensi

Tabel 3.15 Tabel absensi

No	Field	Tipe data	Key
1	id_absensi	Int	Primary key
2	nim	Varchar(30)	Foreign key
3	id_kegiatan	Int	Foreign key
4	id_musyrif	Int	Foreign key
5	pertemuan	Int	
6	tanggal	Varchar(30)	
7	hadir	Int	
8	sakit	Int	
9	izin	Int	
10	alpha	Int	

12. Tabel jadwal mahad

Tabel 3.16 Tabel mahad

No	Field	Tipe data	Key
1	id_jadwalmahad	Int	Primary key
2	hari_kegiatan	Varchar(30)	

No	Field	Tipe data	Key
3	jam	Varchar(30)	
4	nama_kegiatan	Varchar(30)	
5	ruang	Varchar(30)	

13. Tabel penjadwalan regular

Tabel 3.17 Tabel penjadwalan regular

No	Field	Tipe data	Key
1	id_reg	Int	Primary key
2	id_jurusan	Varchar(30)	Foreign key
3	Hari	Varchar(30)	
4	Pukul	Varchar(30)	
5	Matakuliah	Varchar(30)	
6	Dosen	Varchar(30)	
7	Sks	Varchar(30)	
8	Kelas	Varchar(30)	
9	Ruang	Varchar(30)	

14. Tabel kelas PPBA

Tabel 3.18 Tabel kelas PPBA

No	Field	Tipe data	Key
1	id_kelasppba	Int	Primary key
2	id_fakultas	Int	Foreign key

No	Field	Tipe data	Key
3	Jam	Varchar(30)	
4	Kelas ppba	Varchar(30)	
5	Nama_ruang	Varchar(30)	

15. Tabel Mahasantri regular

Tabel 3.19 Tabel mahasantri regular

No	Field	Tipe data	Key
1	id_mhsreg	Int	Primary key
2	NIM	Char(11)	Foreign key
3	Id_reg	Int	Foreign key
4	Id_jurusan	Int	Foreign key

16. Tabel Mahasantri PPBA

Tabel 3.20 Tabel mahasantri PPBA

No	Field	Tipe data	Key
1	id_mhsppba	Int	Primary key
2	NIM	Char(11)	Foreign key
3	Id_fakultas	Int	Foreign key
4	kelasppba	Varchar(30)	

17. Tabel Track Posisi

Tabel 3.21 Tabel trackposisi

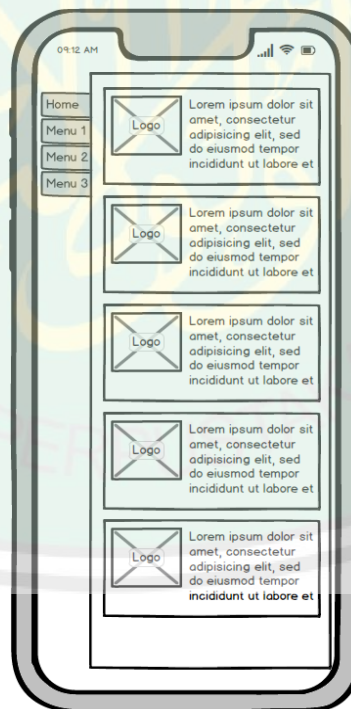
No	Field	Tipe data	Key
1	id_posisi	Int	Primary key
2	NIM	Char(11)	Foreign key
3	Latitude	Varchar(50)	
4	Logintude	Varchar(50)	

3.2.12. Perancangan antarmuka

Pada bagian ini akan ditampilkan rancangan antarmuka (*interface*) aplikasi SIAMA *mobile application*. Layout untuk halaman depan adalah tampilan login dengan beberapa komponen yaitu 2 *textbox* dengan *property* username dan password beserta satu *dropdown* untuk pilihan level user. Berikut rancangan *mockup* tampilannya :



Gambar 3.10 Rancangan *mockup* halaman *login*



Gambar 3.11 Rancangan *mockup* halaman *container home sliding menu* dan *menu content*

BAB IV

UJI COBA DAN PEMBAHASAN

Pada bab ini akan dipaparkan implementasi sistem yang sudah di analisis dan dirancang pada bab sebelumnya, yaitu mengimplementasikan *back end web service* beserta tampilan aplikasi *mobile* yang dibangun.

4.1. Kebutuhan Perangkat Keras

Dalam proses pengembangan sistem *mahad* ini menggunakan komputer dengan spesifikasi perangkat keras sebagai berikut. Untuk kebutuhan pengembangan *software*, spesifikasi perangkat yang digunakan adalah sebagai berikut :

1. CPU Processor Intel® Core™ i5 1.6GHz
2. SSD Storage 256GB
3. RAM Memory 8 GB

Untuk kebutuhan Virtual Private Server (VPS) yang digunakan, spesifikasi yang digunakan adalah sebagai berikut :

1. CPU Processor AMD Opteron 2 Cores
2. HDD Storage 10GB
3. RAM Memory 2 GB

4.2. Kebutuhan Perangkat Lunak

Dalam proses pengembangan sistem *smart mahad* ini menggunakan komputer dengan daftar perangkat lunak yang digunakan sebagai berikut :

1. Sistem Operasi macOS Sierra 10.12.5
2. Emulator Android, AVD SDK 26 Nougat
3. Bahasa Pemrograman Java, PHP, Javascript dan scripting XML
4. IDE Android Studio 2.0 dan Sublime text editor
5. MAMP 3.5.2
6. Slim Framework 2.0
7. StarUML 5
8. Node.js

4.3. Implementasi Sistem

4.3.1. Implementasi Basis Data

Dalam implementasi basis data aplikasi ini, sesuai dengan penjelasan pada bab perancangan, database disimpan dengan nama *siama_mahad*. Masing masing table yang sudah dirancang diimplementasikan dengan SQL dalam DBMS MySQL. Berikut merupakan *script* yang digunakan untuk membangun tabel dan database seperti yang dijelaskan diatas :

Tabel 4.1 Script SQL absensi

```
--
-- Table structure for table `absensi`
--
CREATE TABLE IF NOT EXISTS `absensi` (
  `ID_ABSENSI` int(11) NOT NULL AUTO_INCREMENT,
  `NIM` char(10) DEFAULT NULL,
```

```

`IDTAHUN` int(11) DEFAULT NULL,
`ID_KEGIATAN` int(11) DEFAULT NULL,
`PERTEMUAN` int(11) DEFAULT NULL,
`HADIR` int(11) DEFAULT NULL,
`SAKIT` int(11) DEFAULT NULL,
`IZIN` int(11) DEFAULT NULL,
`ALPHA` int(11) DEFAULT NULL,
`ID_MUSYRIF` char(11) NOT NULL,
`ID_MUSOHIH` int(13) NOT NULL,
`TGL_ABSEN` date NOT NULL,
`TGL_BUAT` datetime NOT NULL,
`TGL_UPDATE` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
`HAID` tinyint(1) NOT NULL,
PRIMARY KEY (`ID_ABSENSI`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2465 ;

```

Tabel 4.2 Script SQL admin

```

--
-- Table structure for table `admin`
--
CREATE TABLE IF NOT EXISTS `admin` (
  `ID_ADMIN` int(11) NOT NULL,
  `ID_USER` varchar(30) NOT NULL,
  `AKTIF` tinyint(1) NOT NULL,
  PRIMARY KEY (`ID_ADMIN`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel 4.3 Script SQL user

```

--
-- Table structure for table `user`
--
CREATE TABLE IF NOT EXISTS `user` (
  `PASS` text,

```

```

`LEVEL` int(11) DEFAULT NULL,
`ID_USER` varchar(30) NOT NULL,
PRIMARY KEY (`ID_USER`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel 4.4 Script SQL pinlog

```

--
-- Table structure for table `pinlog`
--
CREATE TABLE IF NOT EXISTS `pinlog` (
  `ID_USER` varchar(30) NOT NULL,
  `LEVEL` int(11) NOT NULL,
  `PASS` varchar(20) NOT NULL,
  UNIQUE KEY `pinlog_key` (`ID_USER`,`LEVEL`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel 4.5 Script SQL fakultas

```

--
-- Table structure for table `fakultas`
--
CREATE TABLE IF NOT EXISTS `fakultas` (
  `ID_FAKULTAS` int(11) NOT NULL AUTO_INCREMENT,
  `NAMA_FAKULTAS` varchar(30) NOT NULL,
  PRIMARY KEY (`ID_FAKULTAS`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

```

Tabel 4.6 Script SQL jurusan

```

--
-- Table structure for table `jurusan`
--
CREATE TABLE IF NOT EXISTS `jurusan` (
  `ID_JURUSAN` int(11) NOT NULL AUTO_INCREMENT,
  `ID_FAKULTAS` int(11) NOT NULL,

```

```

`NAMA_JURUSAN` varchar(30) NOT NULL,
PRIMARY KEY (`ID_JURUSAN`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=68 ;

```

Tabel 4.7 Script SQL mahasantri

```

--
-- Table structure for table `mahasantri`
--
CREATE TABLE IF NOT EXISTS `mahasantri` (
  `NIM` char(10) NOT NULL,
  `ID_MABNA` int(11) DEFAULT NULL,
  `ID_USER` varchar(30) NOT NULL,
  `ID_HASIL` char(20) DEFAULT NULL,
  `ID_JURUSAN` int(11) NOT NULL,
  `NAMA` varchar(30) DEFAULT NULL,
  `JENIS_KELAMIN` varchar(20) NOT NULL,
  `TEMPAT_LAHIR` varchar(25) NOT NULL,
  `TANGGAL_LAHIR` date NOT NULL,
  `ALAMAT` text,
  `PROVINSI` int(5) NOT NULL,
  `KOTA` varchar(30) NOT NULL,
  `LANTAI` int(5) NOT NULL,
  `KAMAR` int(11) DEFAULT NULL,
  `TELEPON` varchar(14) NOT NULL,
  `EMAIL` varchar(30) NOT NULL,
  `PRESTASI` text NOT NULL,
  `ID_PMB` int(4) NOT NULL,
  `RIWAYAT_PENDIDIKAN` varchar(50) DEFAULT NULL,
  `NAMA_PESANTREN` varchar(25) NOT NULL,
  `KOTA_PESANTREN` varchar(30) NOT NULL,
  `LAMA_PESANTREN` varchar(15) NOT NULL,
  `LULUS_SEKOLAH` int(5) NOT NULL,
  `IDTAHUN` int(11) NOT NULL,

```

```

`LULUS` tinyint(1) NOT NULL,
`KELUAR` tinyint(1) NOT NULL,
`KET_KELUAR` text NOT NULL,
`AKTIF` tinyint(1) NOT NULL,
`ID_MUSYRIF` char(11) NOT NULL,
`FOTO` varchar(100) NOT NULL,
PRIMARY KEY (`NIM`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel 4.8 Script SQL *musyrif*

```

--
-- Table structure for table `musyrif`
--
CREATE TABLE IF NOT EXISTS `musyrif` (
  `ID_MUSYRIF` char(11) NOT NULL,
  `IDTAHUN` int(11) NOT NULL,
  `NAMA` varchar(35) NOT NULL,
  `JENIS_KELAMIN` varchar(20) NOT NULL,
  `ALAMAT` text NOT NULL,
  `KAMAR` int(11) NOT NULL,
  `TELP` varchar(15) NOT NULL,
  `AYAH` varchar(30) NOT NULL,
  `IBU` varchar(30) NOT NULL,
  `TELP_ORTU` varchar(15) NOT NULL,
  `JALUR_PMB` varchar(20) NOT NULL,
  `RIWAYAT_PENDIDIKAN` varchar(40) NOT NULL,
  `LULUS` tinyint(1) NOT NULL,
  `KELUAR` tinyint(1) NOT NULL,
  `KET_KELUAR` text NOT NULL,
  `ID_MABNA` int(11) NOT NULL,
  `ID_USER` varchar(30) NOT NULL,
  `ID_JURUSAN` int(11) NOT NULL,
  `DIVISI` varchar(30) NOT NULL,

```

```

`AKTIF` tinyint(1) NOT NULL,
PRIMARY KEY (`ID_MUSYRIF`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel 4.9 Script SQL murabbi

```

--
-- Table structure for table `murabi`
--
CREATE TABLE IF NOT EXISTS `murabi` (
  `ID_MURABI` int(11) NOT NULL,
  `NAMA` varchar(35) NOT NULL,
  `ALAMAT` text NOT NULL,
  `KAMAR` int(11) NOT NULL,
  `TELP` varchar(30) NOT NULL,
  `ID_MABNA` int(11) NOT NULL,
  `ID_USER` varchar(30) NOT NULL,
  `AKTIF` tinyint(1) NOT NULL,
  PRIMARY KEY (`ID_MURABI`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel 4.10 Script SQL pengasuh

```

--
-- Table structure for table `pengasuh`
--
CREATE TABLE IF NOT EXISTS `pengasuh` (
  `ID_PENGASUH` int(11) NOT NULL,
  `NAMA` varchar(35) NOT NULL,
  `ID_USER` varchar(30) NOT NULL,
  `AKTIF` tinyint(1) NOT NULL,
  PRIMARY KEY (`ID_PENGASUH`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel 4.11 Script SQL mabna

```
--
-- Table structure for table `mabna`
--
CREATE TABLE IF NOT EXISTS `mabna` (
  `ID_MABNA` int(11) NOT NULL AUTO_INCREMENT,
  `NAMA_MABNA` varchar(10) DEFAULT NULL,
  `JML_KAMAR` int(11) DEFAULT NULL,
  `KAPASITAS_KAMAR` int(11) DEFAULT NULL,
  `KD_RUANG` varchar(20) CHARACTER SET latin1 COLLATE latin1_general_ci NOT NULL,
  `KD_LOKASI` varchar(20) CHARACTER SET latin1 COLLATE latin1_general_ci NOT NULL,
  PRIMARY KEY (`ID_MABNA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;
```

Tabel 4.12 Script SQL jadwalmahad

```
--
-- Table structure for table `jadwalmahad`
--
CREATE TABLE IF NOT EXISTS `jadwalmahad` (
  `id_jadwalmahad` int(11) NOT NULL AUTO_INCREMENT,
  `hari_kegiatan` varchar(20) NOT NULL,
  `jam_awal` varchar(20) NOT NULL,
  `jam_akhir` varchar(20) NOT NULL,
  `nama_kegiatan` varchar(500) NOT NULL,
  `ruang` varchar(50) NOT NULL,
  PRIMARY KEY (`id_jadwalmahad`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=71 ;
```

Tabel 4.13 Script SQL penjadwalan regular

```
CREATE TABLE IF NOT EXISTS `penjadwalanreg` (
  `id_reg` int(11) NOT NULL AUTO_INCREMENT,
  `hari` varchar(10) NOT NULL,
```

```

`pukul` varchar(20) NOT NULL,
`kode` varchar(20) NOT NULL,
`matakuliah` varchar(50) NOT NULL,
`dosen` varchar(100) NOT NULL,
`ruang` varchar(10) NOT NULL,
`kelas` varchar(20) NOT NULL,
`id_jurusan` varchar(10) NOT NULL,
`hasil_kode` varchar(10) NOT NULL,
`jam_awal` varchar(5) NOT NULL,
`jam_akhir` varchar(5) NOT NULL,
PRIMARY KEY (`id_reg`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1664 ;

```

4.3.2. Implementasi *Web Service*

Dalam proses implementasi pembangunan *web service*, peneliti mengimplementasikan bahasa PHP (dengan *SLIM framework 2.0*) dan bahasa *javascript* dengan *engine node.js*.

Tabel 4.14 Tabel Bahasa pemrograman

No	Bahasa Pemrograman	Service
1	PHP (<i>SLIM framework</i>)	Mahasantri <i>service</i>
2	Javascript (<i>Node.js</i>)	<i>NewsFeed service</i>

Detail *web service* yang digunakan beserta operasinya akan dijelaskan sebagai berikut.

1. Mahasantri *Service*

Tabel 4.15 Operasi *web service* mahasantri

	Nama Operasi	Parameter <i>Input</i>	<i>Output</i>
--	--------------	------------------------	---------------

No			
1	profil()	String idUser	Data profil mahasantri
2	absensi()	String idUser	Data absensi mahasantri
3	iqob()	String idUser, String detail, String idKegiatan	Data iqob mahasantri
4	tashih()	String idUser	Data <i>report</i> tashih mahasantri
5	khs()	String idUser	Data khs mahasantri
6	fasilitas()	String idUser	Data fasilitas kamar mahasantri
7	kondisikamar()	String idUser	Data kondisi kamar mahasantri
8	infopendaftaran()	-	Data info pendafraran musyrif
9	regmusyrif()	String idUser	Data info status registrasi musyrif
10	pengumuman()	String idUser	Data <i>report</i> nilai akhir mahasantri
11	sendlocation()	String idUser, double lat, double long	Status <i>location</i>

Penjelasan *method service* untuk mahasantri adalah sebagai berikut :

- a. *profil()* adalah *method* sekaligus *path url* untuk menampilkan data profil mahasantri dengan parameter *input* berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method profil()*.

Tabel 4.16 Script PHP untuk *method profil()*

```

1. static function profil($id)
2. {
3.     $sql = "SELECT * FROM mahasantri where NIM=mahasantri.NIM and ID_USER=$i
         d";
4.
5.     $datajur = "SELECT mahasantri.*,jurusan.NAMA_JURUSAN from mahasantri,jur
         usan where nim=$id and mahasantri.ID_JURUSAN=jurusan.ID_JURUSAN";
6.
7.     $datamabna = "SELECT mahasantri.*,mabna.NAMA_MABNA from mahasantri,mabna
         where nim=$id and mahasantri.ID_MABNA=mabna.ID_MABNA";

```

```

8.
9. $dataortu = "SELECT * from data_ortu where NIM=$id";
10.     try {
11.         $db = getDB();
12.         $stmt = $db->query($sql);
13.         $stmt_jur = $db->query($datajur);
14.         $stmt_mabna = $db->query($datamabna);
15.         $stmt_dataortu = $db->query($dataortu);
16.
17.         $datasantri = $stmt->fetch(PDO::FETCH_BOTH);
18.         $datajurusan = $stmt_jur->fetch(PDO::FETCH_BOTH);
19.         $datamabna = $stmt_mabna->fetch(PDO::FETCH_BOTH);
20.         $dataortu = $stmt_dataortu->fetch(PDO::FETCH_BOTH);
21.         $db = null;
22.
23.     echo '{
24.         "profil": ' . json_encode($arrRtn, JSON_PRETTY_PRINT) . ',
25.         "mabna": ' . json_encode($arrMabnaRtn, JSON_PRETTY_PRINT) . ',
26.         "data_ortu": ' . json_encode($arrOrtuRtn, JSON_PRETTY_PRINT) . ' }';
27.
28. } catch (PDOException $e) {
29.     echo '{"error":{"text":"' . $e->getMessage() . '}}';
30. }
31. }

```

- b. *absensi()* adalah *method* sekaligus *path url* untuk menampilkan data absensi mahasantri dengan parameter *input* berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method absensi()*.

Tabel 4.17 Script PHP untuk *method absensi()*

```

1. static function absensi($id)
2. {
3.     try {
4.         for ($i = 1; $i <= 6; $i++) {
5. $sholat1 = "SELECT MAX(`PERTEMUAN`) as PERTEMUAN from absensi where id_k
   kegiatan = $i and IDTAHUN=(select IDTAHUN from tahunakademik where AKTIF=
   '1' order by IDTAHUN desc limit 1)";
6.
7. $sholat2 = "SELECT sum(Hadir) as HADIR from absensi where id_kegiatan =
   $i and IDTAHUN=(select IDTAHUN from tahunakademik where AKTIF='1' order
   by IDTAHUN desc limit 1)and NIM = $id";
8.
9. $sholat3 = "SELECT sum(SAKIT)+sum(IZIN)+sum(ALPHA) as ALPHA from absensi
   where id_kegiatan = $i and IDTAHUN=(select IDTAHUN from tahunakademik w
   here AKTIF='1' order by IDTAHUN desc limit 1) and nim = $id";
10.
11. $sholat4 = "SELECT(sum(Hadir) /count(*))*100 as HADIR from absensi where
   id_kegiatan = $i and IDTAHUN=(select IDTAHUN from tahunakademik where A
   KTIF='1' order by IDTAHUN desc limit 1) and nim = $id";
12. $sholat5 = "SELECT(sum(SAKIT)+sum(IZIN)+sum(ALPHA)) /count(*)*100 as ALP
   HA from absensi where id_kegiatan = $i and IDTAHUN=(select IDTAHUN from

```

```

    tahunakademik where AKTIF='1' order by IDTAHUN desc limit 1) and nim = $
    id";
13.
14.         $db = getDB();
15.         $stmt = $db->query($sholat1);
16.         $stmt2 = $db->query($sholat2);
17.         $stmt3 = $db->query($sholat3);
18.         $stmt4 = $db->query($sholat4);
19.         $stmt5 = $db->query($sholat5);
20.         $db = null;
21.
22.         $datasholat1 = $stmt->fetch(PDO::FETCH_BOTH);
23.         $datasholat2 = $stmt2->fetch(PDO::FETCH_BOTH);
24.         $datasholat3 = $stmt3->fetch(PDO::FETCH_BOTH);
25.         $datasholat4 = $stmt4->fetch(PDO::FETCH_BOTH);
26.         $datasholat5 = $stmt5->fetch(PDO::FETCH_BOTH);
27.
28.         if ($i == 1) {
29. $str1 = '{"sholat": ' . json_encode($arrRtn, JSON_PRETTY_PRINT) . '}' ;
30.         } else if ($i == 3) {
31. $str2 = '{"quran": ' . json_encode($arrRtn, JSON_PRETTY_PRINT) . '}' ;
32.         } else if ($i == 4) {
33. $str3 = '{"afkar": ' . json_encode($arrRtn, JSON_PRETTY_PRINT) . '}' ;
34.         } else if ($i == 5) {
35. $str4 = '{"arab": ' . json_encode($arrRtn, JSON_PRETTY_PRINT) . '}' ;
36.         } else if ($i == 6) {
37. $str5 = '{"inggris": ' . json_encode($arrRtn, JSON_PRETTY_PRINT) . '}' ;
38.         }
39.         }
40. echo "[" . $str1 . "," . $str2 . "," . $str3 . "," . $str4 . "," . $str5
    . "]" ;
41.     } catch (PDOException $e) {
42.         echo '{"error":{"text":"' . $e->getMessage() . '}}';
43.     }
44. }

```

- c. *iqob()* adalah *method* sekaligus *path url* untuk menampilkan data *iqob* mahasantri dengan parameter input berupa NIM mahasantri, detail *iqob* maupun id kegiatan. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method iqob()*.

Tabel 4.18 Script PHP untuk *method iqob()*

```

1. static function iqob($id, $detail = '', $id_kegiatan = null)
2. {
3.     $row = array();
4.     if ($detail == 'detail' AND isset($id_kegiatan)) {
5. $sql = "SELECT pertemuan, sakit, alpha, izin FROM `absensi` WHERE NIM=?
    and IDTAHUN=(SELECT IDTAHUN from tahunakademik where AKTIF=1 order by ID
    TAHUN desc limit 1) and ID_KEGIATAN=? and (SAKIT=1 or IZIN=1 or ALPHA=1)
    ";

```

```

6.         $db = getDB();
7.         $stmt = $db->prepare($sql);
8.         $stmt->execute(array($id, $id_kegiatan));
9.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
10.        } else {
11. $sql = "select k.ID_KEGIATAN id_kegiatan, k.KATEGORI kegiatan, i.TINGKAT
_HUKUMAN_IQOB tingkat, i.ALFA alfa_kumulatif, i.KUMPUL kumpul from iqob
i, kegiatan k where i.ALFA>2 and i.IQOB>0 and i.NIM=? AND i.ID_KEGIATAN=
k.ID_KEGIATAN";
12.         $db = getDB();
13.         $stmt = $db->prepare($sql);
14.         $stmt->execute(array($id));
15.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
16.        }
17.        echo json_encode($row, JSON_PRETTY_PRINT);
18.    }

```

d. *tashih()* adalah *method* sekaligus *path url* untuk menampilkan data iqob mahasantri dengan parameter input berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method tashih()*.

Tabel 4.19 Script PHP untuk *method tashih()*

```

1. static function tashih($id)
2. {
3.     $row = array();
4.     $sql = "SELECT @ranking:=@ranking+1 AS pertemuan, surat_ke, awal_ayat, a
_khir_ayat, JUZ_AKHIR juz_akhir FROM setoran_tashih ,(SELECT @ranking:=0)
_r where NIM=? and IDTAHUN=(select IDTAHUN from tahunakademik where AKTI
F='1' order by IDTAHUN desc limit 1) and JENIS=1 ORDER BY JUZ_AKHIR ASC"
;
5.         $db = getDB();
6.         $stmt = $db->prepare($sql);
7.         $stmt->execute(array($id));
8.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
9.         echo json_encode($row, JSON_PRETTY_PRINT);
10.    }

```

e. *khs()* adalah *method* sekaligus *path url* untuk menampilkan data khs mahasantri dengan parameter input berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method khs()*.

Tabel 4.20 Script PHP untuk *method khs()*

```

1. static function khs($id)
2. {
3. $row = array();
4. $sql = "SELECT @ranking:=@ranking+1 AS pertemuan, kegiatan.kategori, khs
        .nilai, khs.angka FROM khs,kegiatan,(SELECT @ranking:=0) r WHERE khs.ID_
        KEGIATAN = kegiatan.ID_KEGIATAN AND khs.IDTAHUN=(select IDTAHUN from tah
        unakademik where AKTIF=1 order by IDTAHUN desc limit 1) AND khs.NIM = ?"
        ;
5. $db = getDB();
6. $stmt = $db->prepare($sql);
7. $stmt->execute(array($id));
8. $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
9. echo json_encode($row, JSON_PRETTY_PRINT);
10. }

```

f. *fasilitas()* adalah *method* sekaligus *path url* untuk menampilkan data fasilitas mahasantri yang di dapat di *mahad* dengan parameter *input* berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method fasilitas()*.

Tabel 4.21 Script PHP untuk *method fasilitas()*

```

1. static function fasilitas($id) {
2. $row = array();
3. $sql = "SELECT m.nim, m.id_mabna, mab.nama_mabna, m.kamar,id_barang, nam
        a_barang, m.idtahun, f.status FROM mahasantri m, mabna mab, fasilitasmhs
        f WHERE m.ID_USER=? AND m.IDTAHUN=(SELECT IDTAHUN FROM `tahunakademik`
        ORDER BY IDTAHUN DESC LIMIT 1) AND m.ID_MABNA=mab.ID_MABNA";
4. $db = getDB();
5. $stmt = $db->prepare($sql);
6. $stmt->execute(array($id));
7. while ($row0 = $stmt->fetch(PDO::FETCH_ASSOC)) {
8. $stmt_ket = $db->prepare
9. ("SELECT keterangan FROM `cekfasilitas` WHERE `NIM` = ? AND `IDMABNA` =
        ? AND `KAMAR` = ? AND `IDTAHUN` = ? AND `ID_BARANG` = ?");
10. $stmt_ket->execute
11. (array($row0['nim'], $row0['id_mabna'], $row0['kamar'], $row0['idtahun']
        , $row0['id_barang']));
12. $row_ket = $stmt_ket->fetch(PDO::FETCH_ASSOC);
13. if ($row_ket) {
14. $row0['keterangan'] = $row_ket['keterangan'];
15. } else {
16. $row0['keterangan'] = "3";
17. }
18. $row[] = $row0;
19. }
20. $status['id_mabna'] = $row[0]['id_mabna'];
21. $status['mabna'] = $row[0]['nama_mabna'];
22. $status['kamar'] = $row[0]['kamar'];
23. foreach ($row as $key => $value) {

```

```

24.         unset($row[$key]['idtahun']);
25.         unset($row[$key]['id_mabna']);
26.         unset($row[$key]['nama_mabna']);
27.         unset($row[$key]['nim']);
28.     }
29.     $status['barang'] = $row;
30.     echo json_encode($status, JSON_PRETTY_PRINT);
31. }

```

g. *kondisikamar()* adalah *method* sekaligus *path url* untuk menampilkan data kondisi kamar mahasantri yang di dapat di *mahad* dengan parameter *input* berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method kondisikamar()*.

Tabel 4.22 Script PHP untuk *method kondisikamar()*

```

1. static function kondisikamar($id)
2. {
3.     $row = array();
4.     $sql = "SELECT m.nim, m.id_mabna, mab.nama_mabna, m.kamar,id_barang, nam
a_barang, m.idtahun, f.status FROM mahasantri m, mabna mab, fasilitasmhs
f WHERE m.ID_USER=? AND m.IDTAHUN=(SELECT IDTAHUN FROM `tahunakademik`
ORDER BY IDTAHUN DESC LIMIT 1) AND m.ID_MABNA=mab.ID_MABNA";
5.     $db = getDB();
6.     $stmt = $db->prepare($sql);
7.     $stmt->execute(array($id));
8.     while ($row0 = $stmt->fetch(PDO::FETCH_ASSOC)) {
9.         $stmt_ket = $db->prepare
10. ("SELECT keterangan FROM `laporkondisikamar` WHERE `NIM` = ? AND `IDMABN
A` = ? AND `KAMAR` = ? AND `IDTAHUN` = ? AND `ID_BARANG` = ?");
11. $stmt_ket->execute
12. (array($row0['nim'], $row0['id_mabna'], $row0['kamar'], $row0['idtahun']
, $row0['id_barang']));
13.     $row_ket = $stmt_ket->fetch(PDO::FETCH_ASSOC);
14.     if ($row_ket) {
15.         $row0['keterangan'] = $row_ket['keterangan'];
16.     } else {
17.         $row0['keterangan'] = "";
18.     }
19.     $row[] = $row0;
20. }
21. $status['id_mabna'] = $row[0]['id_mabna'];
22. $status['mabna'] = $row[0]['nama_mabna'];
23. $status['kamar'] = $row[0]['kamar'];
24. foreach ($row as $key => $value) {
25.     unset($row[$key]['idtahun']);
26.     unset($row[$key]['id_mabna']);
27.     unset($row[$key]['nama_mabna']);
28.     unset($row[$key]['nim']);
29. }
30. $status['barang'] = $row;
31.

```

```

32.         echo json_encode($status, JSON_PRETTY_PRINT);
33.     }

```

h. *infopendaftaran()* adalah *method* sekaligus *path url* untuk menampilkan data informasi pendaftaran *musyrif* di *mahad*. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method infopendaftaran()*.

Tabel 4.23 Script PHP untuk *method infopendaftaran()*

```

1. static function infopendaftaran()
2. {
3.     $row = array();
4.     $sql = "SELECT * FROM m_informasi ORDER BY id DESC";
5.     $db = getDB();
6.     $stmt = $db->prepare($sql);
7.     $stmt->execute();
8.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
9.     echo json_encode($row, JSON_PRETTY_PRINT);
10. }

```

i. *regmusyrif()* adalah *method* sekaligus *path url* untuk menampilkan data informasi update registrasi *musyrif* di *mahad* dengan parameter *input* berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method regmusyrif()*.

Tabel 4.24 Script PHP untuk *method regmusyrif()*

```

1. static function regmusyrif($id)
2. {
3.     $row = array();
4.     $sql = "select m.nim,m.nama,m.jenis_kelamin, m.tempat_lahir, m.alamat, m
        .telepon,f.nama_fakultas, j.nama_jurusan,m.riwayat_pendidikan, m.lulus_s
        ekolah, m.lama_pesantren, m.id_pesantren FROM mahasantri m,jurusan j,fak
        ultas f where m.NIM=? AND m.ID_JURUSAN=j.ID_JURUSAN and j.ID_FAKULTAS=f.
        ID_FAKULTAS";
5.     $db = getDB();
6.     $stmt = $db->prepare($sql);
7.     $stmt->execute(array($id));
8.     $row = $stmt->fetch(PDO::FETCH_ASSOC);
9.
10. $stmt = $db->prepare
11. ("SELECT `nm_pesantren` FROM `pesantren` WHERE `id_pesantren`=?");
12.     $stmt->execute(array($row['id_pesantren']));
13.     $row_pesantren = $stmt->fetch(PDO::FETCH_ASSOC);

```

```

14.         $row['nm_pesantren'] = $row_pesantren['nm_pesantren'];
15.
16. $stmt = $db->prepare
17. ("SELECT nama_ayah, pekerjaan_ayah, nama_ibu, pekerjaan_ibu FROM `data_o
    rtu` WHERE NIM=? LIMIT 1");
18.         $stmt->execute(array($id));
19.         $row_oru = $stmt->fetch(PDO::FETCH_ASSOC);
20.         $row['nama_ayah'] = $row_oru['nama_ayah'];
21.         $row['pekerjaan_ayah'] = $row_oru['pekerjaan_ayah'];
22.         $row['nama_ibu'] = $row_oru['nama_ibu'];
23.         $row['pekerjaan_ibu'] = $row_oru['pekerjaan_ibu'];
24.         echo json_encode($row, JSON_PRETTY_PRINT);
25.     }

```

- j. *pengumuman()* adalah *method* sekaligus *path url* untuk menampilkan data informasi *update* berupa nilai interview, ujian, dll untuk seleksi *musyrif*, parameter yang di *input* berupa NIM mahasantri. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method* *pengumuman()*.

Tabel 4.25 Script PHP untuk *method* *pengumuman()*

```

1. static function pengumuman($id)
2. {
3.     $row = array();
4.     $db = getDB();
5.
6.     $stmt = $db->prepare("select hasil from m_administrasi where nim=?");
7.     $stmt->execute(array($id));
8.     $row['administrasi'] = $stmt->fetch(PDO::FETCH_ASSOC);
9.
10.    $stmt = $db->prepare
11.    ("select nilai_interview, nilai_arab, nilai_inggris, nilai_alquran, nila
        i_akhir, keterangan from m_nilaites where jalur_seleksi='musyrif/ah' and
        `nim`=?");
12.
13.    $stmt->execute(array($id));
14.    $row['musyrif'] = $stmt->fetch(PDO::FETCH_ASSOC);
15.
16.    $stmt = $db->prepare
17.    ("select nilai_hafalan, nilai_interview, nilai_kepribadian, nilai_akhir,
        keterangan from m_nilaites where jalur_seleksi='htq' and `nim`=?");
18.    $stmt->execute(array($id));
19.    $row['htq'] = $stmt->fetch(PDO::FETCH_ASSOC);
20.
21.    echo json_encode($row, JSON_PRETTY_PRINT);
22. }

```

k. *sendlocation()* adalah *method* sekaligus *path url* untuk mengupdate lokasi *user* untuk lokasi mahasantri dengan parameter *latitude* dan *longitude*. *Method* ini menggunakan POST, berikut potongan *source code* untuk *method sendlocation()*.

Tabel 4.26 Script PHP untuk *method sendlocation()*

```

1. static function sendlocation($id, $lat, $longi)
2. {
3.     $row = array();
4.     $row['mh_nim'] = $id;
5.     $row['mh_lat'] = $lat;
6.     $row['mh_long'] = $longi;
7.     echo json_encode($row, JSON_PRETTY_PRINT);
8.     $payload = file_get_contents('http://api.someservice.com/SomeMethod?
    param=value');
9.
10.    $url = 'http://server.com/path';
11.    $data = array('key1' => 'value1', 'key2' => 'value2');
12.
13.    // use key 'http' even if you send the request to https://...
14.    $options = array(
15.        'http' => array(
16.            'header' => "Content-type: application/x-www-form-
    urlencoded\r\n",
17.            'method' => 'POST',
18.            'content' => http_build_query($data),
19.        ),
20.    );
21.    $context = stream_context_create($options);
22.    $result = file_get_contents($url, false, $context);
23. }

```

2. Musyrif Service

Tabel 4.27 Operasi *web services* mahad *activity* Musyrif

No	Nama Operasi	Parameter Input	Output
1	presensi()	String idMusyrif, Integer pertemuanKe, integer kamarMinimum, integer kamarMaximum, String idKegiatan	Data <i>report</i> presensi mahasantri

2	updatepresensi()	String idAbsensi, Integer jumlahKehadiran, Integer jumlahSakit, Integer jumlahAlpha, Integer jumlahIzin	Status <i>update</i> presensi mahasiswa
3	kelasbahasa()	String idMusrif	Data presensi kelas Bahasa mahasiswa
4	nilaites()	String idMusrif, String idKelasTest	Data hasil nilai test mahasiswa
5	nilautsuas()	String idMusrif, String idKegiatan	Data hasil nilai uas mahasiswa
6	mahasantripandai()	String idKegiatan, String jenisTest	Data mahasiswa berprestasi
7	kelasquran()	String idMusrif	Data presensi kelas Quran mahasiswa
8	ijinpulang()	String idMusrif	Data mahasiswa ijin pulang
9	pelanggaransantri()	String idMusrif	Data mahasiswa dengan pelanggaran
10	notifikasiqob()	String idMusrif	Data iqob mahasiswa
11	kelasafkar()	String idMusrif	Data kelas Afkar mahasiswa
12	listsantri()	String idMusrif	Data list profil mahasiswa
13	detailfasiltas()	String nimSantri	Data fasilitas kamar mahasiswa
14	detailkondisikamar()	String idMusrif, String nimSantri	Data kondisi kamar mahasiswa
15	invenmenunggu()	String idMusrif	Data <i>report</i> inventaris
16	invenditolak()	String idMusrif	Data <i>report</i> inventaris yang

			ditolak
17	invendiproses()	String idMusyrif	Data <i>report</i> inventaris yang diproses
18	inventidakdirespons()	String idMusyrif	Data <i>report</i> inventaris yang tidak direspons

Penjelasan mengenai method pada musyrif service dapat dilihat pada penjelasan berikut :

a. *presensi()* adalah *method* sekaligus *path url* untuk menampilkan data informasi *update* nilai presensi mahasantri di *mahad*. Parameter untuk *method* ini adalah Id musyrif, jumlah pertemuan, kamar *minimum* dan *maximum* untuk *paging* data, dan id kegiatan. Method ini menggunakan GET, berikut potongan *source code* untuk *method presensi()*.

Tabel 4.28 Script PHP untuk *method presensi()*

```

1. static function presensi($idmusyrif, $pertemuan, $kamarmin, $kamarmax, $
   id_keg)
2. {
3.     $sql = "SELECT id_absensi, nim as mh_nim, nama as mh_nama, idtahun,
4.             kategori, pertemuan,
5.             hadir as mh_hadir, sakit as mh_sakit,
6.             izin as mh_izin, alpha as mh_alpha,
7.             id_musyrif, id_musohih, nama_musyrif,
8.             tgl_absen, kamar as mh_kamar_mabna FROM
9.             (SELECT a.id_absensi, a.nim nim1, b.nama,
10.                  a.idtahun, d.kategori,
11.                  a.pertemuan, a.hadir,
12.                  a.sakit, a.izin,
13.                  a.alpha, a.id_musyrif, e.nama as nama_musyrif,
14.                  a.id_musohih, a.tgl_absen,
15.                  b.kamar FROM absensi a, mahasantri b, kegiatan d
16.                  , musyrif e
17.                  WHERE a.IDTAHUN =
18.                      (SELECT IDTAHUN FROM tahunakademik WHERE AKT
19.                       IF=1 ORDER BY IDTAHUN DESC LIMIT 1)
20.                      AND PERTEMUAN=?

```

```

19.         AND a.ID_KEGIATAN=?
20.         AND a.nim = b.nim
21.         AND a.id_musyrif = e.id_musyrif
22.         AND a.id_kegiatan = d.id_kegiatan) t1,
23.         (SELECT c.nim FROM mahasantri c, musyrif d W
    HERE c.ID_MABNA = d.ID_MABNA
24.         AND d.ID_USER = ?
25.         AND c.IDTAHUN =
26.         (SELECT IDTAHUN FROM tahunakademik W
    HERE AKTIF =1 ORDER BY IDTAHUN DESC LIMIT 1)
27.         AND c.KAMAR >= ?
28.         AND c.KAMAR <= ?
29.         ORDER BY c.KAMAR ASC) t2
30.         WHERE nim1=nim";
31.
32.         $db = getDB();
33.         $stmt = $db->prepare($sql);
34.         $stmt->execute(array
35. ($pertemuan, $id_keg, $idmusyrif, $kamarmax, $kamarmax));
36.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
37.         $obj = (object)array('pertemuan' => $pertemuan, 'kelas' => $row[
    0]['kategori'], 'presensi' => $row);
38.         echo json_encode($obj, JSON_PRETTY_PRINT);
39.
40.     }

```

b. *updatepresensi()* adalah *method* sekaligus *path url* untuk update presensi mahasantri. Parameter yang digunakan adalah NIM mahasantri. *Method* ini menggunakan POST, berikut potongan *source code* untuk *method updatepresensi()*.

Tabel 4.29 Script PHP untuk *method updatepresensi()*

```

1. static function updatepresensi($idabsensi, $hadir, $sakit, $izin, $alpha
    )
2. {
3.     $sql = "UPDATE absensi set hadir=?, sakit=?, izin=?, alpha=? where i
    d_absensi=?";
4.     $db = getDB();
5.     $stmt = $db->prepare($sql);
6.     $stmt->execute(array($hadir, $sakit, $izin, $alpha, $idabsensi));
7.     if ($stmt) {
8.         echo "success";
9.     } else {
10.        echo $stmt->errorCode();
11.    }
12. }

```

- c. *kelasbahasa()* adalah *method* sekaligus *path url* untuk memperoleh data nilai kelas bahasa di *mahad*. Parameter yang digunakan adalah id musyrif. Method ini menggunakan GET, berikut potongan *source code* untuk *method kelasbahasa()*.

Tabel 4.30 Script PHP untuk *method kelasbahasa()*

```

1. static function kelasbahasa($idmusyrif)
2. {
3.     $sql = "SELECT * FROM
4.         (SELECT
5.             tingkat.id_tingkat, tingkat.id_mabna,
6.             tingkat.nama_tingkat, tingkat.tingkat_ke,
7.             kelas_tes.id_kelas_tes as id_kelas_tes,
8.             tingkat.idtahun, kelas_tes.nama_kelas, mabna.nama_mabna
9.         FROM tingkat, kelas_tes, mabna
10.        WHERE kelas_tes.ID_KELAS_TES = tingkat.ID_KELAS_TES
11.        AND tingkat.ID_MABNA = (SELECT ID_MABNA FROM musyrif WHERE
12.        id_musyrif=? LIMIT 1)
13.        AND tingkat.ID_MABNA = mabna.id_mabna
14.        AND tingkat.IDTAHUN = (SELECT IDTAHUN from tahunakademik
15.        where AKTIF=1 order by IDTAHUN desc limit 1) ) AS F1
16.        WHERE id_kelas_tes= 1 or id_kelas_tes= 2";
17.
18.     $db = getDB();
19.     $stmt = $db->prepare($sql);
20.     $stmt->execute(array($idmusyrif));
21.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
22.     $obj = (object)array('nama_kelas' => $row[0]['nama_kelas'], 'nama_mabna' => $row[0]['nama_mabna'], 'daftar_kelas' => $row);
23.     echo json_encode($obj, JSON_PRETTY_PRINT);
24. }

```

- d. *nilaitest()* adalah *method* sekaligus *path url* untuk memperoleh data nilai test di *mahad*. Parameter yang digunakan adalah id musyrif dan id kelas test yang diselenggarakan. Method ini menggunakan GET, berikut potongan *source code* untuk *method nilaitest()*.

Tabel 4.31 Script PHP untuk *method nilaitest()*

```

1. static function nilaites($idmusyrif, $id_kelas_tes)
2. {
3.     $sql = "SELECT
4.         mahasantri.nim as mh_nim, mahasantri.nama as mh_nama,
5.         mahasantri.kamar as mh_kamar_mabna, nilai_tes.nilai_tes

```

```

        AS tes_awal,
6.         nilai_tes.nilai_uts AS monitoring, musyrif.nama AS penil
ai,
7.         nilai_tes.id_nilai_tes, kelas_tes.nama_kelas
8.         FROM mahasantri, musyrif, nilai_tes , kelas_tes
9.         WHERE mahasantri.ID_MABNA=musyrif.ID_MABNA
10.        and mahasantri.IDTAHUN=(SELECT IDTAHUN FROM tahunakademi
k WHERE AKTIF=1 ORDER BY IDTAHUN DESC LIMIT 1)
11.        and musyrif.ID_USER= ?
12.        and nilai_tes.nim = mahasantri.nim
13.        and nilai_tes.id_musyrif = musyrif.ID_USER
14.        and nilai_tes.jenis_tes= ?
15.        and kelas_tes.id_kelas_tes = nilai_tes.jenis_tes
16.        order by mahasantri.nim ASC";
17.        $db = getDB();
18.        $stmt = $db->prepare($sql);
19.        $stmt->execute(array($idmusyrif, $id_kelas_tes));
20.        $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
21.        $obj = (object)array('id_kelas' => $id_kelas_tes, 'nama_kelas' =
> $row[0]['nama_kelas'], 'list_nilai' => $row);
22.        echo json_encode($obj, JSON_PRETTY_PRINT);
23.
24.    }

```

e. *nilaiutsuas()* adalah *method* sekaligus *path url* untuk memperoleh data nilai UAS maupun UTS di *mahad*. Parameter yang digunakan adalah id musyrif dan id kegiatan uts dan id kegiatan UAS maupun UTS yang diselenggarakan. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method nilaitest()*.

Tabel 4.32 Script PHP untuk *method nilaiutsuas()*

```

1. static function nilautsuas($idmusyrif, $id_kegiatan)
2.     {
3.
4.     $sql = "SELECTmahasantri.nim as mh_nim, mahasantri.nama as mh_nama,
5.         mahasantri.kamar as mh_kamar_mabna, utsuas.uts AS nilai_ut
s , utsuas.uas AS nilai_uas, musyrif.nama as penilai, utsuas.id_utsuas, ke
giatan.kategori
6.         FROM mahasantri, musyrif, utsuas , kegiatan
7.         WHERE mahasantri.ID_MABNA=musyrif.ID_MABNA
8.         and mahasantri.IDTAHUN=(SELECT IDTAHUN FROM tahunakademik
WHERE AKTIF=1 ORDER BY IDTAHUN DESC LIMIT 1)
9.         and musyrif.ID_USER= ?
10.        and mahasantri.nim = utsuas.nim
11.        and utsuas.id_musyrif = musyrif.ID_USER
12.        and utsuas.id_kegiatan = ?
13.        and kegiatan.id_kegiatan = ?
14.        order by mahasantri.nim ASC";
15.

```

```

16.     $db = getDB();
17.     $stmt = $db->prepare($sql);
18.     $stmt->execute(array($idmusyraf, $id_kegiatan, $id_kegiatan));
19.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
20.     $obj = (object)array
21. ('id_kegiatan' => $id_kegiatan, 'nama_kelas' => $row[0]['kategori'], 'list
    _nilai' => $row);
22.     echo json_encode($obj, JSON_PRETTY_PRINT);
23. }

```

f. *mahasantripandai()* adalah *method* sekaligus *path url* untuk memperoleh data mahasantri berprestasi di *mahad* berdasarkan nilai UAS dan UTS yang berlangsung. Parameter yang digunakan untuk *method* ini adalah id jenis test dan id kegiatan di *mahad*. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method mahasantripandai()*.

Tabel 4.33 Script PHP untuk *method mahasantripandai()*

```

1. static function mahasantripandai($id_kegiatan, $jenis_tes)
2. {
3.     $sql = "SELECT
4.         mahasantripandai.id_pandai, mahasantripandai.nim as mh_nim
5.     ,
6.         mahasantri.nama as mh_nama, mahasantri.kamar as mh_kamar_m
7.     abna,
8.         mahasantripandai.nilai_arab as nilai_jenis, nilai_tes.nila
9.     i_tes,
10.         nilai_tes.nilai_uts, (
11.         SELECT (
12.             SUM( hadir ) / COUNT( * )
13.         ) * 100
14.         FROM `absensi`
15.         WHERE `ID_KEGIATAN` = ?
16.         AND NIM = mahasantri.NIM
17.         ) AS hasil, (
18.         SELECT (
19.             mahasantripandai.NILAI_ARAB
20.         ) + ( nilai_tes.NILAI_TES ) + ( nilai_tes.NILAI_UTS ) + (
21.         hasil )
22.         ) AS hasil_total
23.     FROM mahasantripandai, mahasantri, nilai_tes
24.     WHERE mahasantri.NIM = mahasantripandai.NIM
25.     AND mahasantripandai.NIM = nilai_tes.NIM
26.     AND mahasantri.NIM = nilai_tes.NIM
27.     AND mahasantripandai.JENIS_TES = ?
28.     AND nilai_tes.JENIS_TES = ?
29.     AND mahasantripandai.IDTAHUN = (SELECT IDTAHUN FROM tahuna

```

```

    kademik WHERE AKTIF=1 ORDER BY IDTAHUN DESC LIMIT 1)
26.         ORDER BY hasil DESC
27.         LIMIT 0 , 30";
28.
29.         $db = getDB();
30.         $stmt = $db->prepare($sql);
31.         $stmt->execute(array($id_kegiatan, $jenis_tes, $jenis_tes));
32.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
33.         echo json_encode($row, JSON_PRETTY_PRINT);
34.     }

```

g. *kelasquran()* adalah *method* sekaligus *path url* untuk memperoleh data mahasantri pada kelas AL-Quran. Parameter yang digunakan untuk *method* ini adalah *id musyrif*. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method kelasquran()*.

Tabel 4.34 Script PHP untuk *method kelasquran()*

```

1.     static function kelasquran($idmusyrif)
2.     {
3.         $sql = "SELECT * FROM
4.             (SELECT
5.                 tingkat.id_tingkat, tingkat.id_mabna,
6.                 tingkat.nama_tingkat, tingkat.tingkat_ke,
7.                 kelas_tes.id_kelas_tes as id_kelas_tes,
8.                 tingkat.idtahun, kelas_tes.nama_kelas, mabna.nama_mabna
9.             FROM tingkat, kelas_tes, mabna
10.            WHERE kelas_tes.ID_KELAS_TES = tingkat.ID_KELAS_TES
11.            AND tingkat.ID_MABNA = (SELECT ID_MABNA FROM musyrif WHE
RE id_musyrif=? LIMIT 1)
12.            AND tingkat.ID_MABNA = mabna.id_mabna
13.            AND tingkat.IDTAHUN = (SELECT IDTAHUN from tahunakademik
where AKTIF=1 order by IDTAHUN desc limit 1) ) AS F1
14.            WHERE id_kelas_tes= 3";
15.
16.         $db = getDB();
17.         $stmt = $db->prepare($sql);
18.         $stmt->execute(array($idmusyrif));
19.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
20.         $obj = (object)array('nama_kelas' => $row[0]['nama_kelas'], 'nama_mabna' => $row[0]['nama_mabna'], 'daftar_kelas' => $row);
21.         echo json_encode($obj, JSON_PRETTY_PRINT);
22.     }

```

h. *ijin pulang()* adalah *method* sekaligus *path url* untuk memperoleh data mahasantri yang *ijin* untuk meninggalkan *mahad*. Parameter

yang digunakan untuk method ini adalah id musyrif. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method* *ijinpulang()*.

Tabel 4.35 Script PHP untuk *method* *ijinpulang()*

```

1. static function ijinpulang($idmusyrif)
2. {
3.     $sql = "SELECT
4.         ijin_pulang_baru.id_ijin_pulang, mahasantri.nim AS mh_ni
5.     m,
6.         mahasantri.nama AS mh_nama, mahasantri.kamar as mh_kamar
7.     _mabna, ijin_pulang_baru.tgl_pulang,
8.         ijin_pulang_baru.batas_kembali, ijin_pulang_baru.tgl_kem
9.     bali,
10.        ijin_pulang_baru.keperluan, musyrif.nama AS nama_musyrif
11.
12.        FROM ijin_pulang_baru, mahasantri, musyrif
13.        WHERE ijin_pulang_baru.NIM = mahasantri.NIM
14.        AND ijin_pulang_baru.ID_MUSYRIF = musyrif.ID_MUSYRIF
15.        AND mahasantri.ID_MABNA = (SELECT ID_MABNA FROM musyrif
16.        WHERE id_musyrif=? LIMIT 1)
17.        AND ijin_pulang_baru.IDTAHUN = (
18.        SELECT IDTAHUN
19.        FROM tahunakademik
20.        WHERE AKTIF =1
21.        ORDER BY IDTAHUN DESC
22.        LIMIT 1 )";
23.     $db = getDB();
24.     $stmt = $db->prepare($sql);
25.     $stmt->execute(array($idmusyrif));
26.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
27.     echo json_encode($row, JSON_PRETTY_PRINT);
28. }
```

i. *pelanggaransantri()* adalah *method* sekaligus *path url* untuk memperoleh data mahasantri yang melakukan pelanggaran di *mahad*. Parameter yang digunakan adalah id musyrif. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method* *pelanggaransantri()*.

Tabel 4.36 Script PHP untuk *method* *pelanggaransantri()*

```

1. static function pelanggaransantri($idmusyrif)
2. {
3.     $sql = "SELECT
```

```

4.      pelanggaran.id_pelanggaran, mahasantri.nim as mh_nim,
5.      mahasantri.nama AS mh_nama, mahasantri.kamar as mh_kamar
6.      _mabna, pelanggaran.nama_pelanggaran,
7.      pelanggaran.tanggal, pelanggaran.tanggal_tindak,
8.      musyrif.nama AS nama_musyrif, pelanggaran.keterangan
9.      FROM pelanggaran, mahasantri, musyrif
10.     WHERE pelanggaran.NIM = mahasantri.NIM
11.     AND pelanggaran.ID_MUSYRIF = musyrif.ID_MUSYRIF
12.     AND mahasantri.ID_MABNA = (
13.     SELECT ID_MABNA
14.     FROM musyrif
15.     WHERE id_musyrif = ?
16.     LIMIT 1 )
17.     AND pelanggaran.IDTAHUN";
18.     $db = getDB();
19.     $stmt = $db->prepare($sql);
20.     $stmt->execute(array($idmusyrif));
21.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
22.     echo json_encode($row, JSON_PRETTY_PRINT);
23. }

```

- j. *notifikasiiqob()* adalah *method* sekaligus *path url* untuk memperoleh data mahasantri yang terkena iqob mahad. Parameter yang digunakan untuk method ini adalah id musyrif. Method ini menggunakan GET, berikut potongan source code untuk method *notifikasiiqob()*.

Tabel 4.37 Script PHP untuk *method notifikasiiqob()*

```

1. static function notifikasiiqob($idmusyrif)
2. {
3.     $sql = "SELECT
4.         mahasantri.nim as mh_nim, mahasantri.nama as mh_nama,
5.         kegiatan.kategori, mahasantri.kamar as mh_kamar_mabna,
6.
7.         iqob.tingkat_hukuman_iqob, iqob.alfa as alfa_kumulatif
8.     FROM iqob, mahasantri, kegiatan
9.     WHERE mahasantri.ID_MABNA = (
10.     SELECT ID_MABNA
11.     FROM musyrif
12.     WHERE id_musyrif = ?
13.     LIMIT 1 )
14.     AND mahasantri.NIM = iqob.NIM
15.     AND ALFA >2
16.     AND IQOB >0
17.     AND iqob.IDTAHUN = (
18.     SELECT IDTAHUN
19.     FROM tahunakademik
20.     WHERE AKTIF =1 )
21.     AND kegiatan.id_kegiatan = iqob.id_kegiatan";

```

```

21.
22.     $db = getDB();
23.     $stmt = $db->prepare($sql);
24.     $stmt->execute(array($idmusyrif));
25.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
26.     echo json_encode($row, JSON_PRETTY_PRINT);
27. }

```

k. *kelasafkar()* adalah *method* sekaligus *path url* untuk memperoleh data mahasantri yang mengikuti kelas afkar. Parameter yang digunakan untuk *method* ini adalah *id musyrif*. *Method* ini menggunakan GET, berikut potongan source code untuk *method kelasafkar()*.

Tabel 4.38 Script PHP untuk *method kelasafkar()*

```

1. static function kelasafkar($idmusyrif)
2. {
3.     $sql = "SELECT * FROM
4.         (SELECT
5.             tingkat.id_tingkat, tingkat.id_mabna,
6.             tingkat.nama_tingkat, tingkat.tingkat_ke,
7.             kelas_tes.id_kelas_tes as id_kelas_tes,
8.             tingkat.idtahun, kelas_tes.nama_kelas, mabna.nama_mabna
9.
10.            FROM tingkat, kelas_tes, mabna
11.            WHERE kelas_tes.ID_KELAS_TES = tingkat.ID_KELAS_TES
12.            AND tingkat.ID_MABNA = (SELECT ID_MABNA FROM musyrif WHE
13.            RE id_musyrif=? LIMIT 1)
14.            AND tingkat.ID_MABNA = mabna.id_mabna
15.            AND tingkat.IDTAHUN = (SELECT IDTAHUN from tahunakademik
16.            where AKTIF=1 order by IDTAHUN desc limit 1) ) AS F1
17.            WHERE id_kelas_tes= 4";
18.
19.     $db = getDB();
20.     $stmt = $db->prepare($sql);
21.     $stmt->execute(array($idmusyrif));
22.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
23.     $obj = (object)array('nama_kelas' => $row[0]['nama_kelas'], 'nam
24.     a_mabna' => $row[0]['nama_mabna'], 'daftar_kelas' => $row);
25.     echo json_encode($obj, JSON_PRETTY_PRINT);
26. }

```

l. *listsantri()* adalah *method* sekaligus *path url* untuk memperoleh data mahasantri berupa info kamar maupun profil. Parameter yang

digunakan untuk method ini adalah id musyrif. Method ini menggunakan GET, berikut potongan source code untuk method *listsantri()*.

Tabel 4.39 Script PHP untuk method *listsantri()*

```

1. static function listsantri($idmusyrif)
2. {
3.     $sql = "SELECT mahasantri.nim as mh_nim, mahasantri.nama as mh_nama,
4.             mabna.nama_mabna as mh_mabna, mahasantri.kamar a
5.             s mh_kamar_mabna
6.             FROM mahasantri, mabna
7.             WHERE mahasantri.ID_MABNA = mabna.ID_MABNA
8.             AND mabna.ID_MABNA = (
9.             SELECT ID_MABNA
10.            FROM musyrif
11.            WHERE id_musyrif = ?
12.            LIMIT 1 )
13.            ORDER BY mahasantri.kamar";
14.     $db = getDB();
15.     $stmt = $db->prepare($sql);
16.     $stmt->execute(array($idmusyrif));
17.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
18.     echo json_encode($row, JSON_PRETTY_PRINT);
19. }
```

m. *detailfasiltas()* adalah method sekaligus path url untuk memperoleh data fasilitas yang diterima mahasantri berupa info kamar. Parameter yang digunakan untuk method ini adalah id mahasantri. Method ini menggunakan GET, berikut potongan source code untuk method *detailfasiltas()*.

Tabel 4.40 Script PHP untuk method *detailfasiltas()*

```

1. static function detailfasiltas($nim_santri)
2. {
3.     $sql = "SELECT m.nim as mh_nim, mab.nama_mabna as mh_mabna, m.na
4.             ma as mh_nama,
5.             m.kamar as mh_kamar_mabna, id_barang, nama_baran
6.             g, f.status
7.             FROM mahasantri m, mabna mab, fasilitasmhs f
8.             WHERE m.ID_USER = ?
9.             AND m.IDTAHUN = (
10.            SELECT IDTAHUN
11.            FROM `tahunakademik`
12.            ORDER BY IDTAHUN DESC
```

```

11.         LIMIT 1 )
12.         AND m.ID_MABNA = mab.ID_MABNA";
13.     $db = getDB();
14.     $stmt = $db->prepare($sql);
15.     $stmt->execute(array($nim_santri));
16.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
17.     $obj = (object)array('mh_nim' => $row[0]['mh_nim'], 'mh_mabna' =
> $row[0]['mh_mabna'], 'mh_nama' => $row[0]['mh_nama'], 'mh_kamar_mabna'
=> $row[0]['mh_kamar_mabna'], 'daftar_fasilitas' => $row);
18.     echo json_encode($obj, JSON_PRETTY_PRINT);
19. }

```

n. *detailkondisikamar()* adalah *method* sekaligus *path url* untuk memperoleh data kondisi kamar yang diterima mahasantri berupa info kamar. Parameter yang digunakan untuk *method* ini adalah id mahasantri. *Method* ini menggunakan GET, berikut potongan source code untuk *method detailkondisikamar()*.

Tabel 4.41 Script PHP untuk *method detailkondisikamar()*

```

1. static function detailkondisikamar($nim_santri, $idmusyrif)
2. {
3.     $sql = "SELECT fasilitas.mhs.nama_barang, laporkondisikamar.status, keter
    angan
4.         FROM laporkondisikamar, fasilitas.mhs
5.         WHERE `NIM` = ?
6.         AND `IDMABNA` = (
7.             SELECT ID_MABNA
8.             FROM musyrif
9.             WHERE id_musyrif = ?
10.            LIMIT 1 )
11.        AND `IDTAHUN` = (
12.            SELECT IDTAHUN
13.            FROM tahun.akademik
14.            WHERE AKTIF =1
15.            AND fasilitas.mhs.id_barang = laporkondisikamar.id_barang

16.        ORDER BY IDTAHUN DESC
17.        LIMIT 1 ) ";
18.     $db = getDB();
19.     $stmt = $db->prepare($sql);
20.     $stmt->execute(array($nim_santri, $idmusyrif));
21.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
22.     echo json_encode($row, JSON_PRETTY_PRINT);
23. }

```

- o. *invenmenunggu()* adalah *method* sekaligus *path url* untuk memperoleh data inventaris yang masih menunggu pelaporan. Parameter yang digunakan untuk method ini adalah id musyrif. Method ini menggunakan GET, berikut potongan source code untuk method *invenmenunggu()*.

Tabel 4.42 Script PHP untuk method *invenmenunggu()*

```

1. static function invenmenunggu($idmusyrif)
2. {
3.     $sql = "SELECT a . * , e.ur_sskel
4.         FROM inven_rusak a, t_dir b, musyrif c, mabna d, t_sskel
5.         e
6.         WHERE c.ID_MABNA = d.ID_MABNA
7.         AND d.kd_lokasi = b.kd_lokasi
8.         AND d.kd_ruang = b.kd_ruang
9.         AND b.no_aset = a.urutan
10.        AND b.kd_brg = a.kode_barang
11.        AND a.kode_barang = e.kd_brg
12.        AND c.ID_USER = ?
13.        AND (
14.            a.tanggal > ?
15.            AND a.tanggal < ?
16.        )
17.        AND a.hapus = '0'
18.        AND a.kirim = '0'
19.        AND a.proses = '2'
20.        ORDER BY tanggal DESC ";
21.     $dateLastMonth = date("Y-m-d", strtotime("-30 days"));
22.     $dateNow = date("Y-m-d", strtotime("+1 days"));
23.     $db = getDB();
24.     $stmt = $db->prepare($sql);
25.     $stmt->execute(array($idmusyrif, $dateLastMonth, $dateNow));
26.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
27.     echo json_encode($row, JSON_PRETTY_PRINT);
28. }
```

- p. *invenditolak()* adalah *method* sekaligus *path url* untuk memperoleh data inventaris yang ditolak dalam pelaporan. Parameter yang digunakan untuk method ini adalah id musyrif. Method ini menggunakan GET, berikut potongan source code untuk method *invenditolak()*.

Tabel 4.43 Script PHP untuk *method invenditolak()*

```

1. static function invenditolak($idmusyrif)
2. {
3.     $sql = "SELECT a . * , e.ur_sskel FROM inven_rusak a, t_dir b, musyrif c,
        mabna d, t_sskel e
4.         WHERE c.ID_MABNA = d.ID_MABNA
5.         AND d.kd_lokasi = b.kd_lokasi
6.         AND d.kd_ruang = b.kd_ruang
7.         AND b.no_aset = a.urutan
8.         AND b.kd_brg = a.kode_barang
9.         AND a.kode_barang = e.kd_brg
10.        AND c.ID_USER = ?
11.        AND a.proses = '0'
12.        AND a.hapus = '0'
13.        AND a.kirim = '0'
14.        ORDER BY tanggal DESC
15.        LIMIT 0 , 10";
16.     $db = getDB();
17.     $stmt = $db->prepare($sql);
18.     $stmt->execute(array($idmusyrif));
19.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
20.     echo json_encode($row, JSON_PRETTY_PRINT);
21. }

```

q. *invendiproses()* adalah *method* sekaligus *path url* untuk memperoleh data inventaris yang diproses dalam pelaporan. Parameter yang digunakan untuk *method* ini adalah id musyrif. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method invendiproses()*.

Tabel 4.44 Script PHP untuk *method invendiproses()*

```

1. static function invendiproses($idmusyrif)
2. {
3.     $sql = "SELECT a . * , e.ur_sskel
4.         FROM inven_rusak a, t_dir b, musyrif c, mabna d, t_sskel
        e
5.         WHERE c.ID_MABNA = d.ID_MABNA
6.         AND d.kd_lokasi = b.kd_lokasi
7.         AND d.kd_ruang = b.kd_ruang
8.         AND b.no_aset = a.urutan
9.         AND b.kd_brg = a.kode_barang
10.        AND a.kode_barang = e.kd_brg
11.        AND c.ID_USER = ?
12.        AND a.proses = '1'
13.        AND a.hapus = '0'
14.        AND a.kirim = '0'
15.        ORDER BY tanggal DESC
16.        LIMIT 0 , 10";

```

```

17.     $db = getDB();
18.     $stmt = $db->prepare($sql);
19.     $stmt->execute(array($idmusyrif));
20.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
21.     echo json_encode($row, JSON_PRETTY_PRINT);
22. }

```

r. *inventidakdirespon()* adalah *method* sekaligus *path url* untuk memperoleh data inventaris yang tidak direspon dalam pelaporan. Parameter yang digunakan untuk *method* ini adalah id musyrif. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method inventidakdirespon()*.

Tabel 4.45 Script PHP untuk *method inventidakdirespon()*

```

1. static function inventidakdirespon($idmusyrif)
2. {
3.     $sql = "SELECT a . * , e.ur_sskel
4.     FROM inven_rusak a, t_dir b, musyrif c, mabna d, t_sskel
5.     e
6.     WHERE c.ID_MABNA = d.ID_MABNA
7.     AND d.kd_lokasi = b.kd_lokasi
8.     AND d.kd_ruang = b.kd_ruang
9.     AND b.no_aset = a.urutan
10.    AND b.kd_brg = a.kode_barang
11.    AND a.kode_barang = e.kd_brg
12.    AND c.ID_USER = ?
13.    AND (
14.        a.tanggal < ?
15.    )
16.    AND a.hapus = '0'
17.    AND a.kirim = '0'
18.    AND a.proses = '2'
19.    ORDER BY tanggal DESC
20.    LIMIT 0 , 10";
21.    $dateLastMonth = date("Y-m-d", strtotime("-30 days"));
22.    $db = getDB();
23.    $stmt = $db->prepare($sql);
24.    $stmt->execute(array($idmusyrif, $dateLastMonth));
25.    $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
26.    echo json_encode($row, JSON_PRETTY_PRINT);
27. }

```

3. Pengasuh Service

Tabel 4.46 Operasi *web services* mahad activity Pengasuh

Nama Operasi	Parameter Input	Output
listsantrimabna()	-	Data mahasantri semua mabna
listiqob()	-	Data mahasantri terkena iqob
listpresensi()	String idMabna	Data presensi mahasantri berdasar mabna
listpenjadwalanmualim()	String idMabna	Data penjadwalan mualim berdasar mabna
listbulanmualim()	-	Data jadwal mualim
listjurnalmualim()	String idJurnal, String idBulan	Data jurnal mualim
listkehadiranbaik()	-	Data kehadiran baik mahasantri
listkehadiranburuk()	-	Data kehadiran buruk mahasantri
mhspandai()	String jenisTest	Data mahasantri berprestasi
penerimaanmusyrif()	-	Data status penerimaan musyrif
penempatanmusyrif()	-	Data status penempatan musyrif
listsetoran()	-	Data status setoran mahasantri
listmabna()	-	Data mabna

Penjelasan mengenai method pengasuh service dapat dilihat pada penjelasan berikut :

- listsantrimabna()* adalah *method* sekaligus *path url* untuk menampilkan data list profil mahasantri pada mabna. Method ini menggunakan GET, berikut potongan *source code* untuk *method listsantrimabna()*.

Tabel 4.47 Script PHP untuk *method listsantrimabna()*

```

1. static function listsantrimabna()
2. {
3.     $sql = "SELECT
4.         mahasantri.nim as mh_nim, mahasantri.nama as mh_nama,
5.         mabna.nama_mabna as mh_mabna, mahasantri.kamar as mh_kam
6.     FROM mahasantri, mabna
7.     WHERE mahasantri.ID_MABNA = mabna.ID_MABNA
8.     ORDER BY mahasantri.ID_MABNA";
9.     $db = getDB();
10.    $stmt = $db->prepare($sql);
11.    $stmt->execute(array());
12.    $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
13.    echo json_encode($row, JSON_PRETTY_PRINT);
14. }

```

- b. *listiqob()* adalah *method* sekaligus *path url* untuk menampilkan data list iqob mahasantri pada mabna. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method listiqob()*.

Tabel 4.48 Script PHP untuk *method listiqob()*

```

1. static function listiqob()
2. {
3.     $sql = "SELECT id_iqob, mahasantri.nim AS mh_nim, mahasantri.nam
4.     a AS mh_nama,
5.     mabna.nama_mabna AS mh_mabna, mahasantri.kamar AS mh_kam
6.     ar_mabna, alfa as alfa_kumulatif,
7.     tingkat_hukuman_iqob, kegiatan.kategori, kumpul as statu
8.     s
9.     FROM iqob, mahasantri, mabna, kegiatan
10.    WHERE ALFA >2
11.    AND IQOB >0
12.    AND mahasantri.nim = iqob.nim
13.    AND mabna.ID_MABNA = mahasantri.ID_MABNA
14.    AND kegiatan.id_kegiatan = iqob.id_kegiatan
15.    AND iqob.IDTAHUN = (
16.        SELECT IDTAHUN
17.        FROM tahunakademik
18.        WHERE AKTIF =1 )";
19.    $db = getDB();
20.    $stmt = $db->prepare($sql);
21.    $stmt->execute(array());
22.    $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
23.    echo json_encode($row, JSON_PRETTY_PRINT);
24. }

```

- c. *listpresensi()* adalah *method* sekaligus *path url* untuk menampilkan data list presensi mahasantri pada mabna. Parameter untuk *method* ini adalah id mabna. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method listpresensi()*.

Tabel 4.49 Script PHP untuk *method listpresensi()*

```

1. static function listpresensi($idmabna)
2. {
3.     $sql = "SELECT nim as mh_nim, nama as mh_nama, kamar as mh_kamar_mabna,
4.         nama_mabna as mh_mabna,
5.         IFNULL(sholat, 0) sholat,
6.         IFNULL(quran, 0) quran,
7.         IFNULL(afkar, 0) afkar,
8.         IFNULL(arab, 0) arab,
9.         IFNULL(inggris, 0) inggris,
10.        IFNULL(tashih, 0) tashih
11.        FROM (SELECT mahasantri.nim, mahasantri.nama, mahasantri
12.            .kamar, mabna.nama_mabna,
13.            SELECT FLOOR( SUM( Hadir ) / COUNT( * ) ) *100 AS jumlah
14.            hadir
15.            FROM `absensi`
16.            WHERE NIM = mahasantri.nim
17.            AND IDTAHUN = (
18.                SELECT IDTAHUN
19.                FROM tahunakademik
20.                WHERE AKTIF = '1'
21.                ORDER BY IDTAHUN DESC
22.                LIMIT 1 )
23.            AND ID_KEGIATAN = 6
24.            ) AS inggris,
25.        (
26.            SELECT JUZ_AKHIR AS jumlahhadir
27.            FROM `setoran_tashih`
28.            WHERE NIM = mahasantri.nim
29.            AND IDTAHUN = (
30.                SELECT IDTAHUN
31.                FROM tahunakademik
32.                WHERE AKTIF = '1'
33.                ORDER BY IDTAHUN DESC
34.                LIMIT 1 )
35.            ORDER BY jumlahhadir DESC
36.            LIMIT 1
37.        ) as tashih
38.        FROM mahasantri, mabna
39.        WHERE mahasantri.ID_MABNA = mabna.ID_MABNA
40.        AND mabna.ID_MABNA = ?
41.        AND mahasantri.IDTAHUN = (
42.            SELECT IDTAHUN
43.            FROM tahunakademik
44.            WHERE AKTIF =1
45.            ORDER BY IDTAHUN DESC

```

```

46.             LIMIT 1 )
47.             ORDER BY KAMAR ASC
48.             LIMIT 0 , 30) as tabelakhir";
49.     $db = getDB();
50.     $stmt = $db->prepare($sql);
51.     $stmt->execute(array($idmabna));
52.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
53.     echo json_encode($row, JSON_PRETTY_PRINT);
54. }

```

- d. *listpenjadwalanmualim()* adalah *method* sekaligus *path url* untuk menampilkan data list penjadwalan *mualim* pada *mabna*. Parameter untuk *method* ini adalah *id mabna*. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method* *listpenjadwalanmualim()*.

Tabel 4.50 Script PHP untuk *method* *listpenjadwalanmualim()*

```

1. static function listpenjadwalanmualim($idmabna)
2. {
3.     $sql = "SELECT
4.         id_jurnal, nama_jurnal,
5.         kelas, mualim, tempat,
6.         hari, nama_mabna as mh_mabna , semester
7.         FROM jurnal_absensi, mabna, tahunakademik
8.         WHERE tahunakademik.IDTAHUN = jurnal_absensi.IDTAHUN
9.         AND jurnal_absensi.ID_MABNA = mabna.ID_MABNA
10.        AND jurnal_absensi.JENIS =2
11.        AND mabna.ID_MABNA = ?
12.        LIMIT 0 , 30";
13.     $db = getDB();
14.     $stmt = $db->prepare($sql);
15.     $stmt->execute(array($idmabna));
16.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
17.     echo json_encode($row, JSON_PRETTY_PRINT);
18. }

```

- e. *listbulanmualim()* adalah *method* sekaligus *path url* untuk menampilkan data *list* jadwal bulanan *mualim* pada *mabna*. Parameter untuk *method* ini adalah *id mabna*. *Method* ini

menggunakan GET, berikut potongan *source code* untuk method *listbulanmualim()*.

Tabel 4.51 Script PHP untuk *method listbulanmualim()*.

```
1. static function listbulanmualim()
2. {
3.     $sql = "SELECT tgl id_bulan, MONTHNAME(STR_TO_DATE(tgl, '%m')) bulan
4.         FROM (SELECT distinct(MONTH(tanggal)) as tgl FROM jurnal_data) 1
5.         istbulan";
6.     $db = getDB();
7.     $db->query("SET lc_time_names = 'id_ID'");
8.     $stmt = $db->prepare($sql);
9.     $stmt->execute(array());
10.    $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
11.    echo json_encode($row, JSON_PRETTY_PRINT);
12. }
```

f. *listjurnalmualim()* adalah *method* sekaligus *path url* untuk menampilkan data list jurnal bulanan *mualim* pada *mabna*. Parameter untuk *method* ini adalah *id mabna*. *Method* ini menggunakan GET, berikut potongan *source code* untuk method *listjurnalmualim()*.

Tabel 4.52 Script PHP untuk *method listbulanmualim()*

```
1. static function listjurnalmualim($idjurnal, $idbulan)
2. {
3.     $sql = "SELECT id_jurnal_data as id_jurnal, waktu, tanggal,
4.         materi, jumlah_mhs_hadir as mh_hadir,
5.         jurnal_absensi.mualim as nama_mualim, musyrif.n
6.         ama AS nama_musyrif
7.         FROM jurnal_data , jurnal_absensi, musyrif
8.         WHERE jurnal_data.ID_JURNAL = jurnal_absensi.ID_JURNAL
9.         AND jurnal_data.ID_JURNAL= ?
10.        AND musyrif.ID_MUSYRIF=jurnal_data.ID_MUSYRIF
11.        AND MONTH (jurnal_data.TANGGAL) = ?";
12.    $db = getDB();
13.    $stmt = $db->prepare($sql);
14.    $stmt->execute(array($idjurnal, $idbulan));
15.    $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
16.    echo json_encode($row, JSON_PRETTY_PRINT);
17. }
```

g. *listkehadiranbaik()* adalah *method* sekaligus *path url* untuk menampilkan data list kehadiran terbaik di *mabna*. Method ini menggunakan GET, berikut potongan *source code* untuk method *listkehadiranbaik()*.

Tabel 4.53 Script PHP untuk *method listkehadiranbaik()*

```

1. static function listkehadiranbaik()
2. {
3.     $sql = "SELECT mahasantri.NIM as mh_nim,
4.             mahasantri.NAMA as mh_nama,
5.             mahasantri.KAMAR as mh_kamar_mabna,
6.             mabna.NAMA_MABNA as mh_mabna,
7.             IFNULL((SELECT
8.                 (SELECT
9.                     (SELECT FLOOR(sum(HADIR) /count(*))*100 AS jumla
10.                hhadir
11.                FROM `absensi`
12.                WHERE NIM=mahasantri.NIM
13.                AND IDTAHUN=
14.                (SELECT IDTAHUN
15.                 FROM tahunakademik
16.                 WHERE AKTIF='1'
17.                 ORDER BY IDTAHUN DESC LIMIT 1)
18.                AND ID_KEGIATAN=1))+
19.            dir
20.            (SELECT floor(sum(HADIR) /count(*))*100 AS jumlahha
21.            FROM `absensi`
22.            WHERE NIM=mahasantri.NIM
23.            AND IDTAHUN=
24.            (SELECT IDTAHUN
25.             FROM tahunakademik
26.             WHERE AKTIF='1'
27.             ORDER BY IDTAHUN DESC LIMIT 1)
28.            AND ID_KEGIATAN=6))/5, 0) AS jumlah
29.            FROM mahasantri,
30.            mabna
31.            WHERE mahasantri.ID_MABNA = mabna.ID_MABNA
32.            AND mahasantri.IDTAHUN=
33.            (SELECT IDTAHUN
34.             FROM tahunakademik
35.             WHERE AKTIF=1
36.             ORDER BY IDTAHUN DESC LIMIT 1)
37.            ORDER BY jumlah DESC LIMIT 10";
38.     $db = getDB();
39.     $stmt = $db->prepare($sql);
40.     $stmt->execute(array());
41.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
42.     echo json_encode($row, JSON_PRETTY_PRINT);
43. }
```

- h. *listkehadiranburuk()* adalah *method* sekaligus *path url* untuk menampilkan data list kehadiran yang tidak memenuhi standar kriteria di *mabna*. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method listkehadiranburuk ()*.

Tabel 4.54 Script PHP untuk *method listkehadiranburuk()*

```

1.
2. static function listkehadiranburuk()
3. {
4.     $sql = "SELECT mahasantri.NIM as mh_nim,
5.             mahasantri.NAMA as mh_nama,
6.             mahasantri.KAMAR as mh_kamar,
7.             mabna.NAMA_MABNA as mh_mabna,
8.             IFNULL((SELECT
9.                 (SELECT
10.                    (SELECT FLOOR(sum(HADIR) /count(*))*100 AS jumlahhad
11.                FROM `absensi`
12.                WHERE NIM=mahasantri.NIM
13.                AND IDTAHUN=
14.                    (SELECT IDTAHUN
15.                     FROM tahunakademik
16.                     WHERE AKTIF='1'
17.                     ORDER BY IDTAHUN DESC LIMIT 1)
18.                AND ID_KEGIATAN=1))+
19.
20. +
21.             (SELECT floor(sum(HADIR) /count(*))*100 AS jumlahhadir
22.             FROM `absensi`
23.             WHERE NIM=mahasantri.NIM
24.             AND IDTAHUN=
25.                 (SELECT IDTAHUN
26.                  FROM tahunakademik
27.                  WHERE AKTIF='1'
28.                  ORDER BY IDTAHUN DESC LIMIT 1)
29.             AND ID_KEGIATAN=5))+
30.             (SELECT floor(sum(HADIR) /count(*))*100 AS jumlahhadir
31.             FROM `absensi`
32.             WHERE NIM=mahasantri.NIM
33.             AND IDTAHUN=
34.                 (SELECT IDTAHUN
35.                  FROM tahunakademik
36.                  WHERE AKTIF='1'
37.                  ORDER BY IDTAHUN DESC LIMIT 1)
38.             AND ID_KEGIATAN=6))/5, 0) AS jumlah
39.         FROM mahasantri,
40.             mabna
41.         WHERE mahasantri.ID_MABNA = mabna.ID_MABNA
42.             AND mahasantri.IDTAHUN=
43.                 (SELECT IDTAHUN
44.                  FROM tahunakademik
45.                  WHERE AKTIF=1

```

```

46.         ORDER BY IDTAHUN DESC LIMIT 1)
47.         ORDER BY jumlah ASC LIMIT 10";
48.
49.         $db = getDB();
50.         $stmt = $db->prepare($sql);
51.         $stmt->execute(array());
52.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
53.         echo json_encode($row, JSON_PRETTY_PRINT);
54.     }
55.

```

- i. *mhspondai()* adalah *method* sekaligus *path url* untuk menampilkan data list mahasiswa berprestasi di mabna. Method ini mempunyai parameter jenis test yang dilaksanakan. Method ini menggunakan GET, berikut potongan *source code* untuk method *mhspondai()*.

Tabel 4.55 Script PHP untuk method *mhspondai()*

```

1. static function mhspondai($jenistes)
2. {
3.     $sql = "SELECT mabna.NAMA_MABNA as mh_mabna,
4.             mahasantripandai.ID_PANDAI as id_pandai,
5.             mahasantripandai.NIM as mh_nim,
6.             mahasantri.NAMA as mh_nama,
7.             mahasantri.KAMAR as mh_kamar_mabna,
8.             mahasantripandai.NILAI_ARAB as nilai_jenis,
9.             nilai_tes.NILAI_TES as nilai_tes,
10.            nilai_tes.NILAI_UTS as nilai_uts,
11.
12.            (SELECT(sum(hadir)/count(*)*100
13.                FROM `absensi`
14.                WHERE `ID_KEGIATAN`=6
15.                AND NIM=mahasantri.NIM) AS hasil,
16.
17.            (SELECT(mahasantripandai.NILAI_ARAB
18.                +(nilai_tes.NILAI_TES)+(nilai_tes.NILAI_UTS)
19.                +(hasil))
20.            AS hasil_total
21.        FROM mahasantripandai,
22.            mahasantri,
23.            nilai_tes,
24.            mabna
25.        WHERE mahasantri.NIM = mahasantripandai.NIM
26.            AND mahasantripandai.NIM = nilai_tes.NIM
27.            AND mahasantri.NIM = nilai_tes.NIM
28.            AND mahasantri.ID_MABNA = mabna.ID_MABNA
29.            AND mahasantripandai.JENIS_TES = ?
30.            AND nilai_tes.JENIS_TES = ?
31.            AND mahasantripandai.IDTAHUN =
32.            (SELECT IDTAHUN
33.                FROM tahunakademik
34.                WHERE AKTIF=1

```

```

35.         ORDER BY IDTAHUN DESC LIMIT 1)
36.         ORDER BY hasil_total DESC";
37.
38.         $db = getDB();
39.         $stmt = $db->prepare($sql);
40.         $stmt->execute(array($jenistes, $jenistes));
41.         $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
42.         echo json_encode($row, JSON_PRETTY_PRINT);
43.     }

```

j. *penerimaanmusyrif()* adalah *method* sekaligus *path url* untuk menampilkan data *list* mahasantri yang melakukan pendaftaran *musyrif*. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method penerimaanmusyrif()*.

Tabel 4.56 Script PHP untuk *method penerimaanmusyrif()*

```

1. static function penerimaanmusyrif()
2. {
3.     $sql = "SELECT a.nim as mh_nim, b.nama_mhs as mh_nama, b.jalur_s
   eleksi
4.         FROM m_nilaites a, m_daftarmusyrif b
5.         WHERE b.IDTAHUN = (
6.
7.             SELECT IDTAHUN
8.             FROM `tahunakademik`
9.             ORDER BY IDTAHUN DESC
10.            LIMIT 1
11.        )
12.        AND a.id_pendaftaran = b.id_pendaftaran
13.        AND keterangan = 'Lulus'
14.        AND a.jalur_seleksi = b.jalur_seleksi
15.        ORDER BY nilai_akhir DESC ";
16.     $db = getDB();
17.     $stmt = $db->prepare($sql);
18.     $stmt->execute(array());
19.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
20.     echo json_encode($row, JSON_PRETTY_PRINT);
21. }

```

k. *penempatanmusyrif()* adalah *method* sekaligus *path url* untuk menampilkan data *list musyrif* yang ditempatkan pada suatu *mabna*.

Method ini menggunakan GET, berikut potongan *source code* untuk *method penempatanmusyrif()*.

Tabel 4.57 Script PHP untuk *method penempatanmusyrif()*

```

1. static function penempatanmusyrif()
2. {
3.     $sql = "SELECT id_musyrif,
4.             nama AS nama_musyrif,
5.             mabna.nama_mabna
6.             FROM musyrif,
7.             mabna
8.             WHERE IDTAHUN=
9.             (SELECT IDTAHUN
10.            FROM tahunakademik
11.            WHERE AKTIF= 1
12.            ORDER BY IDTAHUN DESC LIMIT 1)
13.            AND mabna.ID_MABNA = musyrif.ID_MABNA";
14.     $db = getDB();
15.     $stmt = $db->prepare($sql);
16.     $stmt->execute(array());
17.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
18.     echo json_encode($row, JSON_PRETTY_PRINT);
19. }
```

- l. *listsetoran()* adalah *method* sekaligus *path url* untuk menampilkan data list setoran AlQuran mahasantri pada tiap *mabna*. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method listsetoran()*.

Tabel 4.58 Script PHP untuk *method listsetoran()*

```

1. static function listsetoran()
2. {
3.     $sql = "SELECT musyrif.id_musyrif,
4.             musyrif.nama,
5.             jumlah_juz
6.             FROM setoran,
7.             musyrif
8.             WHERE musyrif.id_musyrif = setoran.id_musyrif";
9.     $db = getDB();
10.    $stmt = $db->prepare($sql);
11.    $stmt->execute(array());
12.    $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
13.    echo json_encode($row, JSON_PRETTY_PRINT);
14. }
```

m. *listmabna()* adalah *method* sekaligus *path url* untuk menampilkan data list mabna. *Method* ini menggunakan GET, berikut potongan *source code* untuk *method listmabna()*.

Tabel 4.59 Script PHP untuk *method listmabna()*

```

1. static function listmabna()
2. {
3.     $sql = "SELECT id_mabna, nama_mabna as mh_mabna from mabna ORDER
        BY id_mabna";
4.     $db = getDB();
5.     $stmt = $db->prepare($sql);
6.     $stmt->execute(array());
7.     $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
8.     echo json_encode($row, JSON_PRETTY_PRINT);
9. }

```

4. Global Service

Tabel 4.60 Operasi *web services* mahad activity Global

No	Nama Operasi	Parameter Input	Output
1	login()	String username, String password, String level	Status login dan token

a. *userlogin()* adalah *method* sekaligus *path url* untuk proses login user. Parameter untuk *method* ini adalah *username*, *password*, dan level *user*. *Method* ini menggunakan POST, berikut potongan *source code* untuk *method userlogin()*.

Tabel 4.61 Script PHP untuk *method userlogin()*

```

1.
2. static function userlogin()
3. {
4.     $app = Slim::getInstance();
5.     $paramValueUser = $app->request->params('username');
6.     $paramValuePass = $app->request->params('password');
7.     $paramValueLevel = $app->request->params('level');
8.     $encrypt_pass = md5($paramValuePass);
9.     $cek1 = "SELECT $paramValueLevel.* FROM $paramValueLevel,user wh
        ere user.ID_USER='$paramValueUser' and $paramValueLevel.ID_USER=user.ID_

```

```

USER and user.PASS='$encrypt_pass';
10.     $cek = "SELECT $paramValueLevel.* FROM $paramValueLevel,user where user.ID_USER='$paramValueUser' and $paramValueLevel.ID_USER=USER";
11.     try {
12.         $db = getDB();
13.         $result_pass = $db->query($cek1);
14.         $row_pass = $result_pass->fetch(PDO::FETCH_NUM);
15.         $result_username = $db->query($cek);
16.         $row_username = $result_username->fetch(PDO::FETCH_NUM);
17.         if ($paramValueUser == "" && $paramValuePass == "") {
18.             $messageError = "User dan Password harus diisi";
19.             $app->contentType('application/json');
20.             echo '{"status": "error" , "message" : ' . json_encode($messageError, JSON_PRETTY_PRINT) . '}' ;
21.         } else if ($row_username == 0) {
22.             $messageError = "Username yang Anda isi salah";
23.             $app->contentType('application/json');
24.             echo '{"status": "error" , "message" : ' . json_encode($messageError, JSON_PRETTY_PRINT) . '}' ;
25.         } else if ($row_pass == 0) {
26.             $messageError = "Password yang Anda isi salah...!!!";
27.             $app->contentType('application/json');
28.             echo '{"status": "error" , "message" : ' . json_encode($messageError, JSON_PRETTY_PRINT) . '}' ;
29.         } else {
30.             $query_ceklimit = $db->query($cek . " limit 1");
31.             $LIHAT = $query_ceklimit->fetch(PDO::FETCH_ASSOC);
32.             if ($LIHAT['AKTIF'] == '1') {
33.                 $arrRtn['user'] = $paramValueUser;
34.                 $arrRtn['status'] = "success";
35.                 if ($paramValueLevel == "musyrif") {
36.                     $arrRtn['jabatan'] = $LIHAT['DIVISI'];
37.                 } else if ($paramValueLevel == "pengasuh") {
38.                     $arrRtn['jabatan'] = 'Pengasuh';
39.                 } else if ($paramValueLevel == "mahsantri") {
40.                     $arrRtn['jabatan'] = 'Mahasantri';
41.                 } else {
42.                     $arrRtn['jabatan'] = '0';
43.                 }
44.                 $app->contentType('application/json');
45.                 echo json_encode($arrRtn);
46.             } else {
47.                 $messageError = "Akun Anda Tidak Aktif";
48.                 $app->contentType('application/json');
49.                 echo '{"status": "error" , "message" : ' . json_encode($messageError, JSON_PRETTY_PRINT) . '}' ;
50.             }
51.         }
52.     } catch (PDOException $e) {
53.         echo '{"error":{"text":"' . $e->getMessage() . '}}';
54.     }
55. }
56. }
57.

```

5. News Service

Tabel 4.62 Operasi web services news feed

NO	Nama Operasi	Parameter Input	Output
1	getUrlFeed()	String port, String urlFeed	Data json <i>newsfeed</i>

Lalu penjelasan mengenai news service adalah sebagai berikut :

- a. *getUrlFeed()* adalah *method* untuk melakukan *scrapping* data pada website resmi mahad dan mendapatkan XML data untuk selanjutnya melakukan transformasi data ke dalam bentuk REST JSON, berikut potongan *source code* untuk method *getUrlFeed()*.

Tabel 4.63 Script Javascript untuk method *getUrlFeed()*

```

1. var express = require('express');
2. var app = express();
3. var Feed = require('rss-to-json');
4.
5. app.use(function(req, res, next) {
6.   res.header("Access-Control-Allow-Origin", "*");
7.   res.header("Access-Control-Allow-Headers", "Origin, X-Requested-
   With, Content-Type, Accept");
8.   next();
9. });
10.
11. app.set('port', (process.env.PORT || 5000));
12.
13. app.get('/', function(req, res, next) {
14.   if (req.query.feedURL) {
15.     Feed.load(req.query.feedURL, function(err, rss){
16.       if (err) {
17.         res.send({ 'error': 'An error has occurred' });
18.       } else {
19.         res.send(rss);
20.       }
21.     });
22.   } else {
23.     res.status(400).send({ 'error': 'feedURL is required' });
24.   }
25. });
26.
27. app.listen(app.get('port'), function() {
28.   console.log('Node app is running on port', app.get('port'));
29. });

```

4.3.3. Implementasi Mobile Apps Android

Pada tahapan ini, setelah *web service* sudah siap dan telah diimplementasikan, maka selanjutnya adalah mengimplementasikan aplikasi android untuk mengkonsumsi *web service* yang sudah ada, berikut adalah beberapa *method* dan tangkapan layar untuk aplikasi android *smart mahad*.

- a. *LoginService()* adalah *Class* service HTTP URL dengan *framework* Spring android untuk melakukan proses *login* dan autentifikasi. *Class* ini melakukan *request* sekaligus melakukan konversi JSON file ke bentuk string agar dapat melakukan *mapping* dengan GSON sehingga akan membentuk *class data object* atau dapat disebut *Plain Old Data Object* (POJO) *class* di java, berikut potongan *source code* untuk class *LoginService()* :

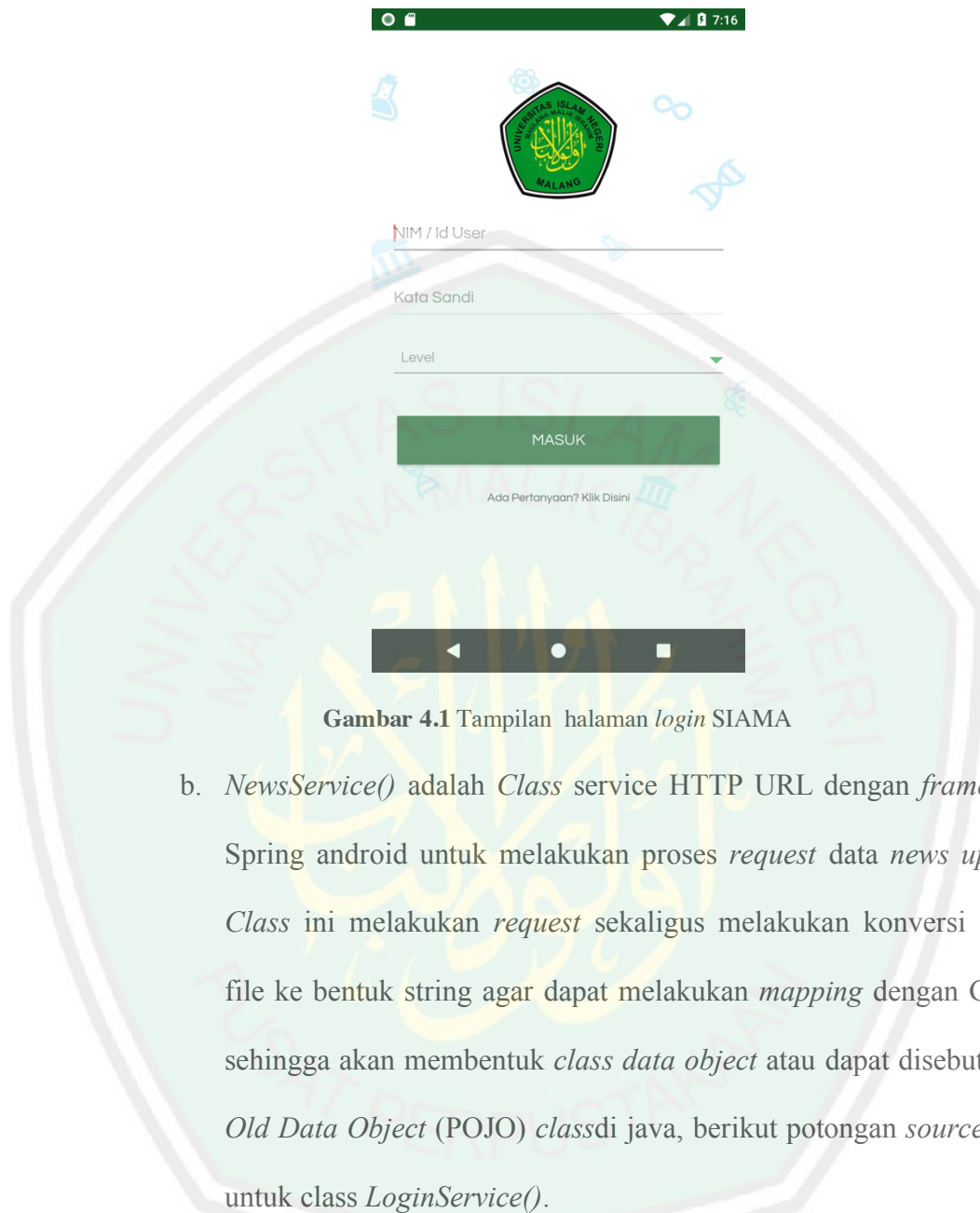
Tabel 4.64 Script Java untuk *method LoginService()*

```

1. package net.mnafian.smartmahad.RestService;
2. /**
3.  * Created by mnafian on 22/10/15.
4.  */
5. @Rest(
6.     rootUrl = Constan.URL_ROOT,
7.     converters = {FormHttpMessageConverter.class, StringHttpMessageC
8.     onverter.class},
9.     interceptors = MyAuthInterceptor.class
10. )
11. @Accept(MediaType.APPLICATION_JSON)
12. public interface LoginService {
13.     @Post(Constan.POST_URL_LOGIN)
14.     String getDataLogin(MultiValueMap formData);
15. }

```

Implementasi tampilan untuk halaman login pada aplikasi adalah sebagai berikut :



Gambar 4.1 Tampilan halaman *login* SIAMA

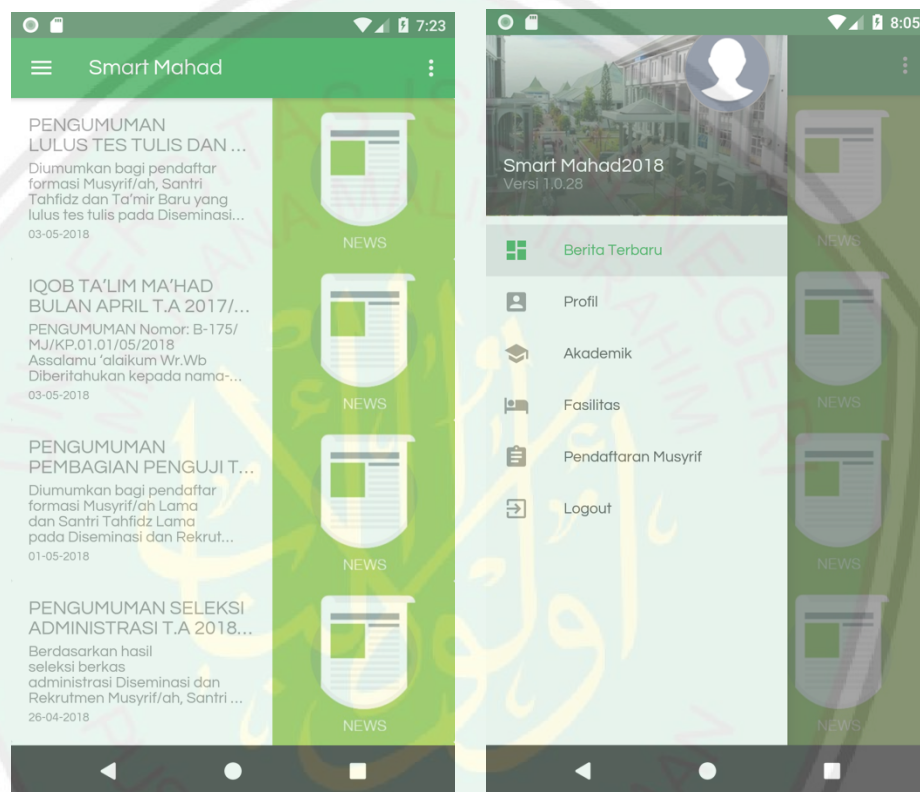
- b. *NewsService()* adalah *Class* service HTTP URL dengan *framework* Spring android untuk melakukan proses *request* data *news update*. *Class* ini melakukan *request* sekaligus melakukan konversi JSON file ke bentuk string agar dapat melakukan *mapping* dengan GSON sehingga akan membentuk *class data object* atau dapat disebut *Plan Old Data Object* (POJO) *class* di java, berikut potongan *source code* untuk class *LoginService()*.

Tabel 4.65 Script Java untuk method *LoginService()*

1.	<code>package net.mnafian.smartmahad.RestService;</code>
2.	<code>@Rest(</code>
3.	<code> rootUrl = Constan.URL_ROOT,</code>
4.	<code> converters = {StringHttpMessageConverter.class}</code>
5.	<code>)</code>
6.	<code>@Accept(MediaType.APPLICATION_JSON)</code>
7.	<code>public interface NewsService {</code>
8.	<code> @Get(Constan.URL_NEWS_FEED)</code>
9.	<code> String getDataNewsFeed();</code>

```
10. }
```

Implementasi tampilan untuk halaman *news* pada aplikasi adalah sebagai berikut :



Gambar 4.2 Tampilan *news update* dan menu utama SIAMA

- c. *MahasantriService()* adalah *Class service* HTTP URL dengan *framework* Spring android untuk melakukan semua proses *request* terhadap *web service* mahasantri yang telah disebutkan pada halaman sebelumnya. *Class* ini melakukan request sekaligus melakukan konversi JSON file ke bentuk string agar dapat dilakukan *mapping* dengan GSON sehingga akan membentuk *classdata object*

atau dapat disebut *Plan Old Data Object* (POJO) class di java, berikut potongan *source code* untuk class *MahasantriService()*.

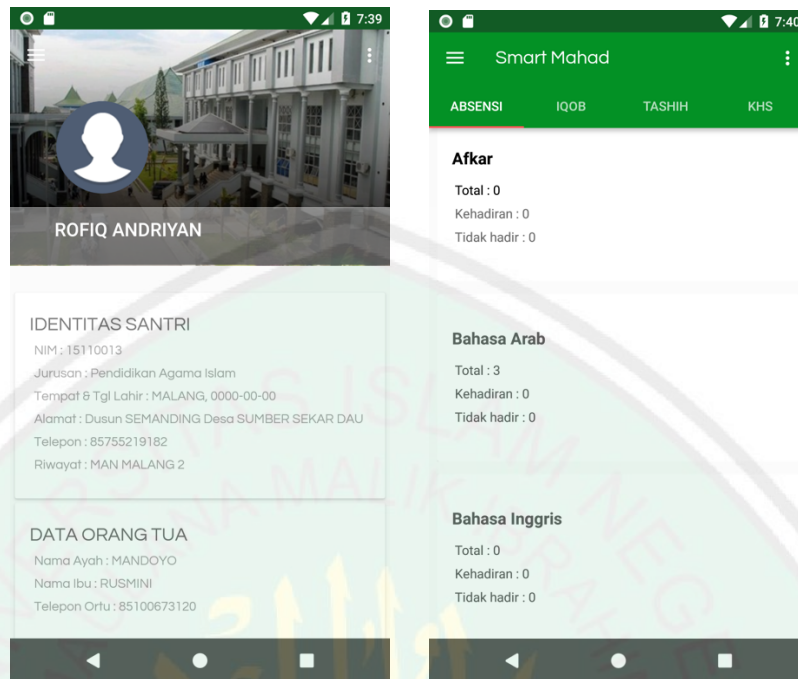
Tabel 4.66 Script Java untuk method *MahasantriService()*

```

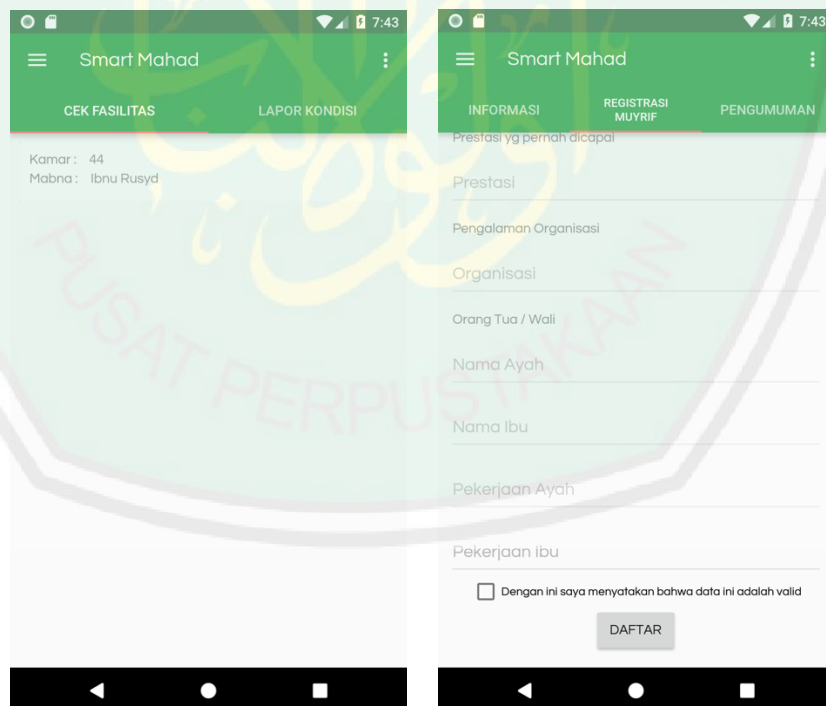
1. package net.mnafian.smartmahad.RestService;
2. @Rest(
3.     rootUrl = Constan.URL_ROOT,
4.     converters = {StringHttpMessageConverter.class},
5.     interceptors = MyAuthInterceptor.class
6. )
7. @Accept(MediaType.APPLICATION_JSON)
8. public interface MahasantriService {
9.
10.     @Get(Constan.GET_URL_PROFILE)
11.     String getDataProfile(String nim);
12.
13.     @Get(Constan.GET_URL_ABSENSI)
14.     String getDataAbsensi(String nim);
15.
16.     @Get(Constan.GET_URL_IQOB)
17.     String getDataIqob(String nim);
18.
19.     @Get(Constan.GET_URL_IQOB_DETAIL)
20.     String getDetailIqob(String nim, String id);
21.
22.     @Get(Constan.GET_URL_TASHIH)
23.     String getDataTashih(String nim);
24.
25.     @Get(Constan.GET_URL_KHS)
26.     String getDataKHS(String nim);
27.
28.     @Get(Constan.GET_URL_CEKFASILITAS)
29.     String getDataCekFasilitas(String nim);
30.
31.     @Get(Constan.GET_URL_LAPOR_KONDISI)
32.     String getLaporKondisi(String nim);
33.
34.     @Get(Constan.GET_URL_INFORMASI_PENDAFTARAN)
35.     String getInformasiPendaftaran();
36.
37.     @Get(Constan.GET_URL_REGISTRASIMUSYRIF)
38.     String getRegisMusyrif(String nim);
39.
40.     @Get(Constan.GET_URL_PENGUMUMAN)
41.     String getPengumuman(String nim);
42.
43.     @Get(Constan.SEND_LOCATION)
44.     String sendToServer(String nim, String lat, String longi);
45. }

```

Implementasi tampilan untuk halaman mahasantri pada aplikasi adalah sebagai berikut :



Gambar 4.3 Tampilan halaman *profile* dan presensi mahasiswa



Gambar 4.4 Tampilan halaman fasilitas dan registrasi *musyirif*

- d. *MusyrifService()* adalah *Class* service HTTP URL dengan *framework* Spring android untuk melakukan semua proses *request* terhadap *web service* musyrif yang telah disebutkan pada halaman sebelumnya. *Class* ini melakukan *request* sekaligus melakukan konversi JSON file ke bentuk string agar dapat dilakukan *mapping* dengan GSON sehingga akan membentuk *classdata object* atau dapat disebut *Plan Old Data Object* (POJO) *class* di java, berikut potongan *source code* untuk class *MusyrifService()*.

Tabel 4.67 Script Java untuk method *MusyrifService()*

```

1. @Rest(rootUrl = Constan.URL_ROOT, converters = {FormHttpMessageConverter
   .class, StringHttpMessageConverter.class}, interceptors = MyAuthIntercep
   tor.class)
2. @Accept(MediaType.APPLICATION_JSON)
3. public interface MusyrifService {
4.
5.     @Get(Constan.GET_URL_PRESENSI)
6.     String getDataPresensi(String idmusyrif, String pertemuan, String ka
   marmin, String kamarmax, String idkegiatan);
7.
8.     @Get(Constan.GET_URL_DAFTAR KELAS_QURAN)
9.     String getDataKelasQuran(String idmusyrif);
10.
11.    @Get(Constan.GET_URL_DAFTAR KELAS_BAHASA)
12.    String getDataKelasBahasa(String idmusyrif);
13.
14.    @Get(Constan.GET_URL_DAFTAR KELAS_AFKAR)
15.    String getDataKelasAfkari(String idmusyrif);
16.
17.    @Get(Constan.GET_URL_NILAITES)
18.    String getNilaiTes(String idmusyrif, String idtes);
19.
20.    @Get(Constan.GET_URL_NILAIUTSUAS)
21.    String getNilaiUtsUas(String idmusyrif, String idUtsUas);
22.
23.    @Get(Constan.GET_URL_MHSPANDAI)
24.    String getMahasiswaPandai(String idKeg, String idJenisTes);
25.
26.    @Get(Constan.GET_URL_KEAMANAN_IJIN)
27.    String getMahasiswaIjin(String idmusyrif);
28.
29.    @Get(Constan.GET_URL_KEAMANAN_PELANGGARAN)
30.    String getMahasiswaPelanggaran(String idmusyrif);
31.
32.    @Get(Constan.GET_URL_KEAMANAN_NOTIF_IQOB)
33.    String getMahasiswaNotifIqob(String idmusyrif);
34.

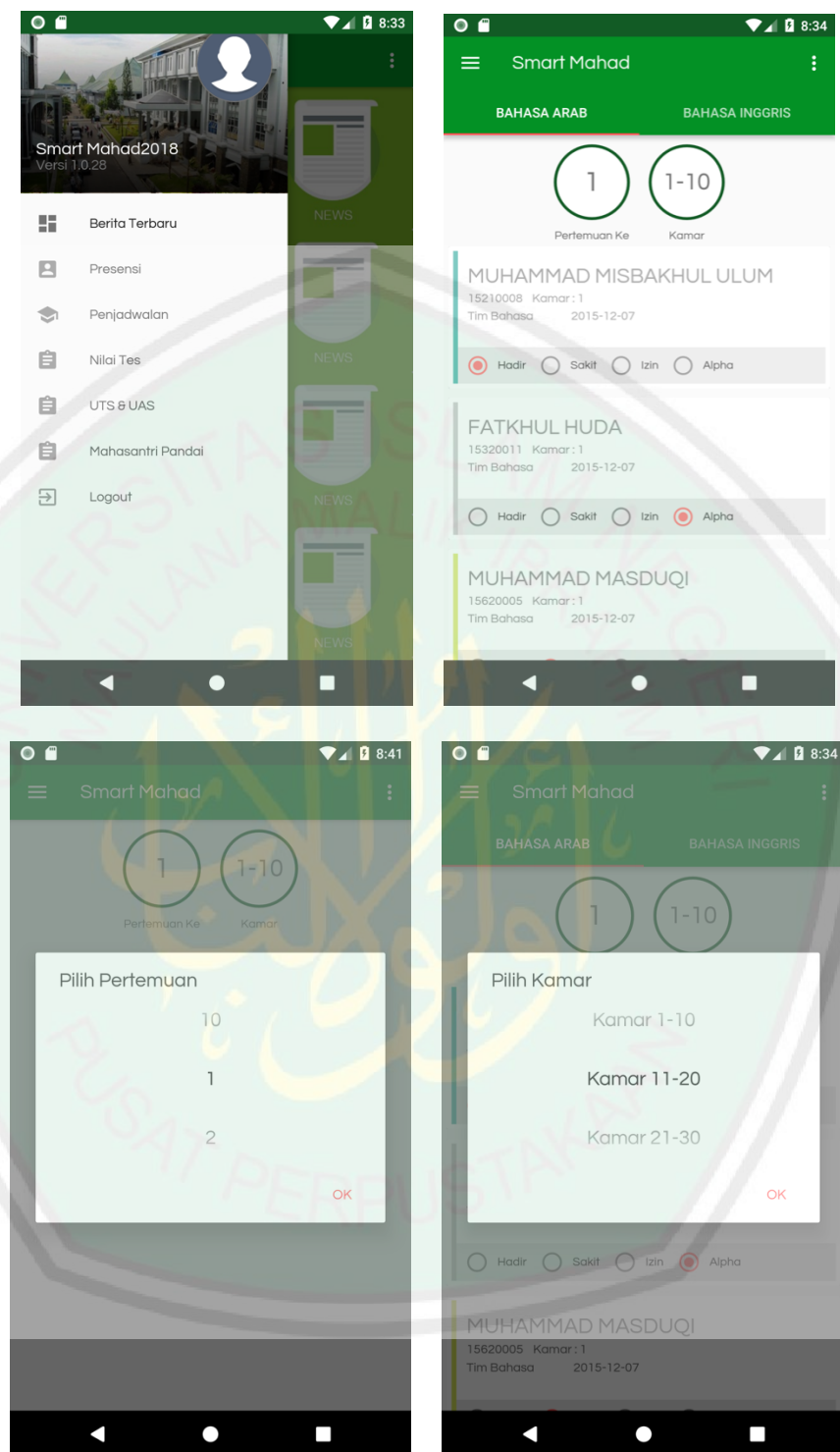
```

```

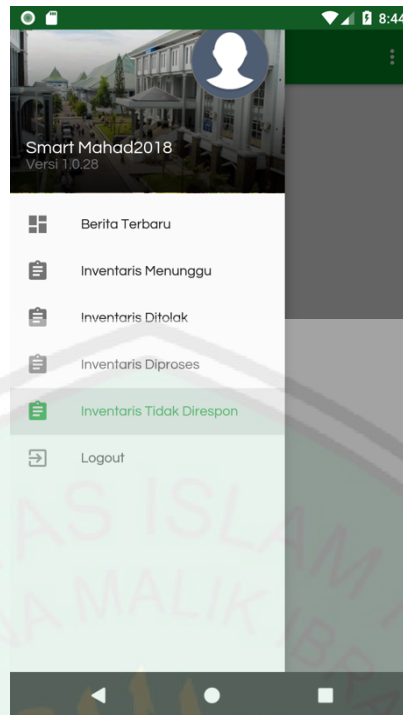
35. @Get(Constan.GET_URL_KESANTRIAN_LIST_SANTRI)
36. String getListSantri(String idmusyrif);
37.
38. @Get(Constan.GET_URL_KESANTRIAN_LIST_FASILITAS)
39. String getListFasilitas(String nimSantri);
40.
41. @Get(Constan.GET_URL_KESANTRIAN_LIST_KONDISI)
42. String getListKondisi(String nimSantri, String idmusyrif);
43.
44. @Get(Constan.GET_URL_INVENTARIS_MENUNGGU)
45. String getInvenMenunggu(String idmusyrif);
46.
47. @Get(Constan.GET_URL_INVENTARIS_DITOLAK)
48. String getInvenDitolak(String idmusyrif);
49.
50. @Get(Constan.GET_URL_INVENTARIS_DIPROSES)
51. String getInvenDiproses(String idmusyrif);
52.
53. @Get(Constan.GET_URL_INVENTARIS_TIDAK_RIESPON)
54. String getInvenTidakDirespon(String idmusyrif);
55.
56. @Get(Constan.GET_URL_UPDATE_PRESENSI)
57. String updateDataPresensi(int idkegiatan, int hadir, int sakit, int
    ijin, int alpha);
58. }

```

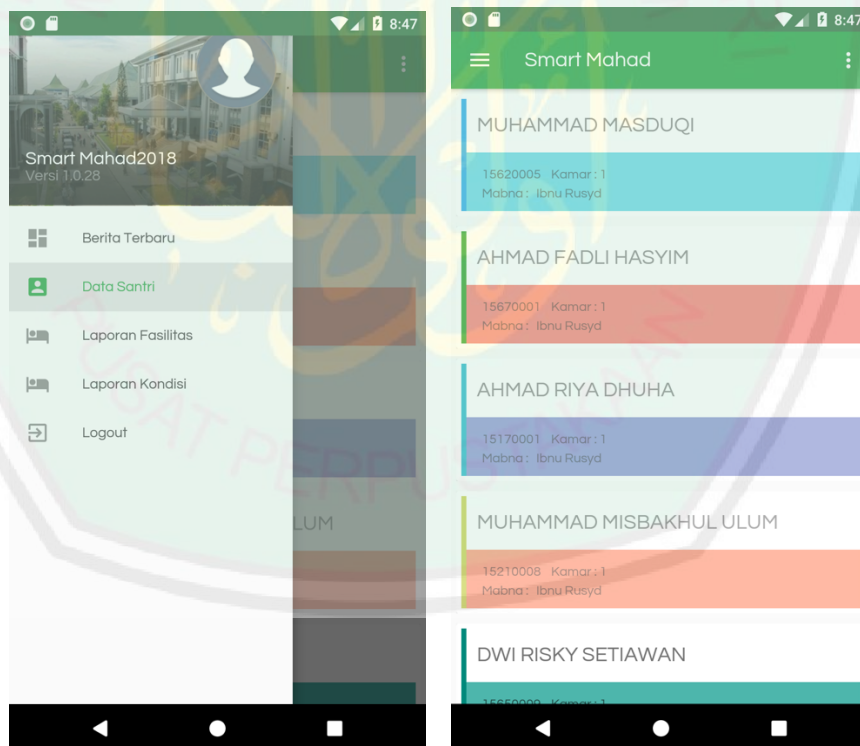
Implementasi tampilan untuk halaman musyrif pada aplikasi adalah sebagai berikut :



Gambar 4.5 Tampilan depan dan menu presensi *musyrif* divisi bahasa



Gambar 4.6 Tampilan halaman depan *musyrif* divisi inventaris



Gambar 4.7 Tampilan halaman depan *musyrif* divisi kesantrian

- e. *PengasuhService()* adalah *Class service* HTTP URL dengan *framework* Spring android untuk melakukan semua proses *request*

terhadap *web service* pengasuh yang telah disebutkan pada halaman sebelumnya. *Class* ini melakukan *request* sekaligus melakukan konversi JSON file ke bentuk string agar dapat dilakukan *mapping* dengan GSON sehingga akan membentuk *class data object* atau dapat disebut *Plan Old Data Object* (POJO) *class* di java, berikut potongan *source code* untuk *class PengasuhService()*.

Tabel 4.68 Script Java untuk *method PengasuhService()*

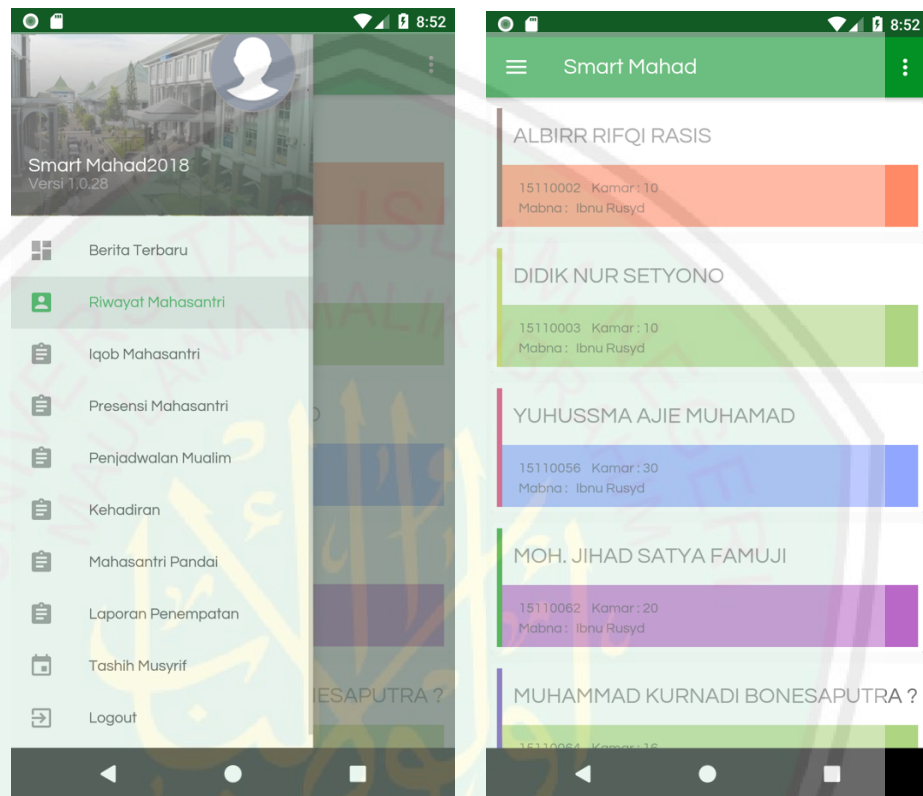
```

1. @Rest(rootUrl = Constan.URL_ROOT, converters = {FormHttpMessageConverter
   .class, StringHttpMessageConverter.class}, interceptors = MyAuthIntercep
   tor.class)
2. @Accept(MediaType.APPLICATION_JSON)
3. public interface PengasuhService {
4.
5.     @Get(Constan.GET_URL_PENGASUH_LIST_SANTRI)
6.     String getListMahasantri();
7.
8.     @Get(Constan.GET_URL_PENGASUH_LIST_IQOB)
9.     String getListIqobSantri();
10.
11.    @Get(Constan.GET_URL_PENGASUH_LIST_PRESENSI)
12.    String getListPresensiSantri(String idmabna);
13.
14.    @Get(Constan.GET_URL_PENGASUH_LIST_PENJADWALAN_MUALIM)
15.    String getListJadwalMualim(String idmabna);
16.
17.    @Get(Constan.GET_URL_PENGASUH_LIST_BULAN)
18.    String getListBulan();
19.
20.    @Get(Constan.GET_URL_PENGASUH_LIST_JURNAL_MUALIM)
21.    String getListJurnal(String idjurnal, String idbulan);
22.
23.    @Get(Constan.GET_URL_PENGASUH_LIST_KEHADIRAN_BAIK)
24.    String getListKehadiranBaik();
25.
26.    @Get(Constan.GET_URL_PENGASUH_LIST_KEHADIRAN_BURUK)
27.    String getListKehadiranBuruk();
28.
29.    @Get(Constan.GET_URL_PENGASUH_LIST_JENIS_TES)
30.    String getListMhsPandai(String jenistes);
31.
32.    @Get(Constan.GET_URL_PENGASUH_LIST_PENERIMAAN_MUSYRIF)
33.    String getPenerimaanMusyrif();
34.
35.    @Get(Constan.GET_URL_PENGASUH_LIST_PENEMPATAN_MUSYRIF)
36.    String getPenempatanMusyrif();
37.
38.    @Get(Constan.GET_URL_PENGASUH_LIST_SETORAN)
39.    String getSetoran();
40.
41.    @Get(Constan.GET_URL_PENGASUH_LIST_MABNA)
42.    String getListMabna();
43.

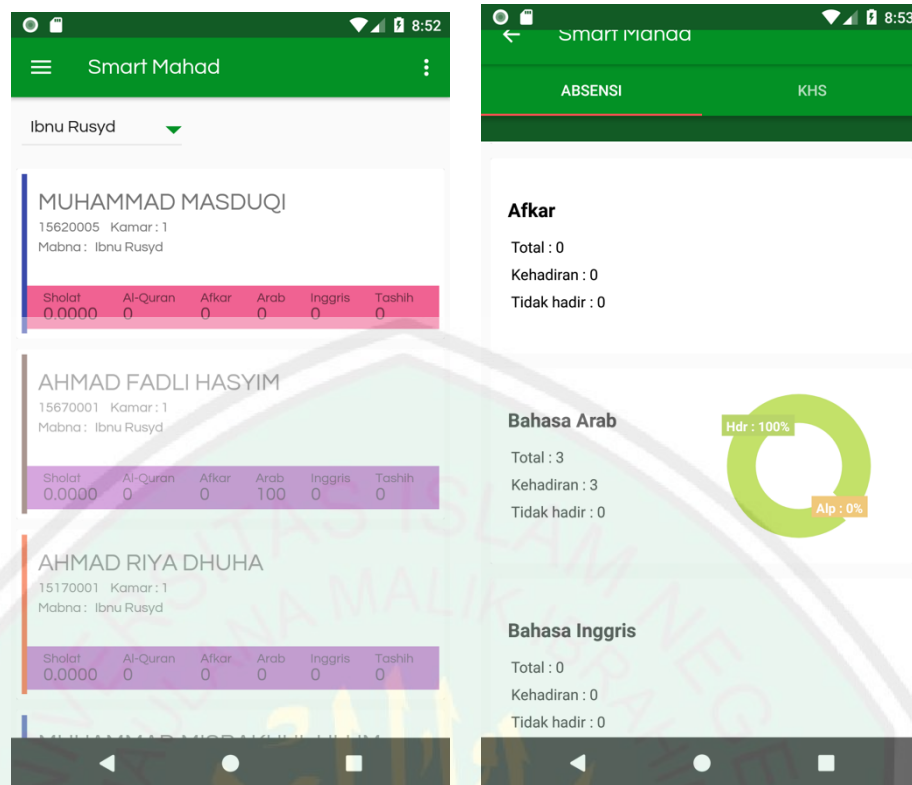
```

44. }

Implementasi tampilan untuk halaman *musyrif* pada aplikasi adalah sebagai berikut :



Gambar 4.8 Tampilan halaman depan pengasuh dan data mahasantri



Gambar 4.9 Tampilan halaman report presensi mahasiswa berdasarkan mahad

4.4. Pengujian Sistem

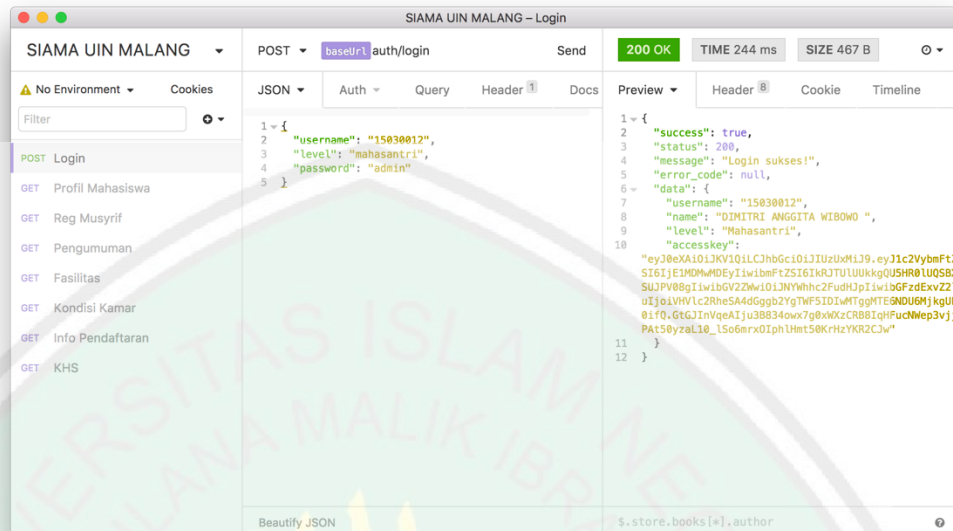
4.4.1. Metode Pengujian

Tahapan yang terakhir dalam pengembangan sistem adalah proses pengujian sistem secara menyeluruh. Pada tahapan ini akan dilakukan pengujian alpha dan pengujian metode *blackbox*.

1. Pengujian Alpha

Pengujian pertama yaitu adalah menguji fungsi-fungsi yang ada dalam *web service*. Apakah sudah berjalan dengan baik atau tidak, pada pengujian ini menggunakan *software testing* untuk *web service* menggunakan *Insomnia Application*. Pengujian ini dilakukan untuk mengecek kesesuaian parameter, fungsi serta operasi telah berjalan

dengan baik dan sesuai dengan yang diharapkan, berikut adalah tampilan aplikasi *Insomnia* untuk *testing web service*:



Gambar 4.10 Tampilan software Insomnia untuk *testing web service*

Hasil pengujian *web service mahad activity* dengan menggunakan *Insomnia REST application* dijelaskan secara detail pada pengujian berikut.

a. Pengujian data pada mahasantri *service*

Pada tabel 4.69 akan dilakukan pengujian dengan parameter data benar pada mahasantri *service*.

Tabel 4.69 Operasi *web services mahad testing* mahasantri data normal

Method operasi dan hasil uji (Data Normal)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	profil()	User Id : 15110013	<i>ProfilUser JSON Response</i>	Sesuai
2	absensi()	User Id : 15110013	<i>AbsensiUser JSON Response</i>	Sesuai
3	iqob()	User Id : 15110013	<i>DataIqob JSON Response</i>	Sesuai
		Detail : Talim Pagi		
		Id Kegiatan : 1		
4	tashih()	User Id : 15110013	<i>DataTashih JSON Response</i>	Sesuai
5	khs()	User Id : 15110013	<i>DataKHS JSON Response</i>	Sesuai
6	fasilitas()	User Id : 15110013	<i>DataFasilitas JSON Response</i>	Sesuai
7	kondisikamar()	User Id : 15110013	<i>DataKamar JSON Response</i>	Sesuai
8	infopendaftar an()	User Id : 15110013	<i>InfoPendaftar JSON Response</i>	Sesuai
9	regmusyrif()	User Id : 15110013	<i>RegMusyrif JSON Response</i>	Sesuai
10	pengumuman()	User Id : 15110013	<i>Pengumuman JSON Response</i>	Sesuai
11	sendlocation()	User Id : 15110013	<i>Success code</i>	Sesuai
		Longitude: - 122.0840		
		Latitude: 37.4220		

Kesimpulan pada hasil pengujian di tabel 4.69 pada mahasantri *service* untuk data benar, didapatkan hasil bahwa seluruh *request* mempunyai *output* yang sesuai dengan parameter dan *response* data yang

diharapkan. Maka pengujian selanjutnya adalah dengan parameter *input* data salah pada mahasantri *service* pada tabel 4.70 berikut.

Tabel 4.70 Operasi *web services mahad testing* mahasantri data salah

Method operasi dan hasil uji (Data Salah)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	profil()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
2	absensi()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
3	iqob()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
		Detail : Talim		
		Id Kegiatan : 999		
4	tashih()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
5	khs()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
6	fasilitas()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
7	kondisikamar()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
8	infopendaftaran()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
9	regmusyrif()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
10	pengumuman()	User Id : 1511	<i>ErrorResponse JSON</i>	Sesuai
11	sendlocation()	User Id : 1511	<i>Error code</i>	Sesuai
		Longitude: -xxx		
		Latitude: xxx		

Kesimpulan pada hasil pengujian di tabel 4.70 untuk data salah pada mahasantri *service*, didapatkan hasil bahwa seluruh *request* mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan.

b. Pengujian data pada *musyrif service*

Pada tabel 4.71 akan dilakukan pengujian dengan parameter data benar pada *musyrif service*.

Tabel 4.71 Operasi *web services mahad testing musyrif* data benar

Method operasi dan hasil uji (Data Normal)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	presensi()	User Id : 15000009	<i>DataPresensi JSON Response</i>	Sesuai
		Pertemuan ke : 1		
		Min. Kamar : 1		
		Max. Kamar : 15		
		Id kegiatan : 10		
2	updatepresensi()	Id presensi : 1	<i>Success response</i>	Sesuai
		Kehadiran : 1		
		Sakit : 10		
		Ijin : 0		
		Alpha : 1		
3	kelasbahasa()	User Id : 15000009	<i>DataPresensi JSON Response</i>	Sesuai

4	nilaites()	User Id : 15000009	<i>DataNilai JSON Response</i>	Sesuai
		Id kelas test : 5		
5	nilautsuas()	User Id : 15000009	<i>DataNilai JSON Response</i>	Sesuai
		Id kegiatan : 7		
6	mahasantripandai()	Id kegiatan : 5	<i>DataProfile JSON Response</i>	Sesuai
		Jenis kelas : bahasa		
7	kelasquran()	User Id : 15000009	<i>DataPresensi JSON Response</i>	Sesuai
8	ijinpulang()	User Id : 15000009	<i>PresensiIjin JSON Response</i>	Sesuai
9	pelanggaransantri()	User Id : 15000009	<i>DataPelanggaran JSON Response</i>	Sesuai
10	notifikasiiqob()	User Id : 15000009	<i>DataIqob JSON Response</i>	Sesuai
11	kelasafkar()	User Id : 15000009	<i>DataPresensi JSON Response</i>	Sesuai
12	listsantri()	User Id : 15000009	<i>ListProfil JSON Response</i>	Sesuai
13	detailfasiltas()	Id mahasantri : 15110013	<i>ListFasilitas JSON Response</i>	Sesuai
14	detailkondisikamar()	User Id : 15000009	<i>DetailKamar JSON Response</i>	Sesuai
		Id mahasantri : 15110013		
15	invenmenunggu()	User Id : 15000009	<i>DetailInvent JSON Response</i>	Sesuai
16	invenditolak()	User Id : 15000009	<i>DetailInvent JSON Response</i>	Sesuai
17	invendiproses()	User Id : 15000009	<i>DetailInvent JSON Response</i>	Sesuai
18	inventidakdirepon()	User Id : 15000009	<i>DetailInvent JSON Response</i>	Sesuai

Kesimpulan pada hasil pengujian di tabel 4.71 untuk data benar pada musyrif service, didapatkan hasil bahwa seluruh request

mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan. Maka pengujian selanjutnya adalah pengujian untuk *musyrif service* pada data salah dijelaskan pada tabel 4.72.

Tabel 4.72 Operasi *web services mahad testing* Musyrif data salah

Method operasi dan hasil uji (Data Salah)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	presensi()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
		Pertemuan ke : 999		
		Min. Kamar : 999		
		Max. Kamar : 999		
		Id kegiatan : 999		
2	updatepresensi()	Id presensi : 999	<i>ErrorResponse JSON</i>	Sesuai
		Kehadiran : 999		
		Sakit : 999		
		Ijin : 999		
		Alpha : 999		
3	kelasbahasa()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
4	nilaites()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
		Id kelas test : 999		
5	nilautsuas()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
		Id kegiatan : 999		

6	mahasantripan dai()	Id kegiatan : 999	<i>ErrorResponse JSON</i>	Sesuai
		Jenis kelas : xxx		
7	kelasquran()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
8	ijin pulang()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
9	pelanggaransa ntri()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
10	notifikasiqob()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
11	kelasafkar()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
12	listsantri()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
13	detailfasiltas()	Id mahasantri : 15110	<i>ErrorResponse JSON</i>	Sesuai
14	detailkondisika mar()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
		Id mahasantri : 15110		
15	invenmenungg u()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
16	inwenditolak()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
17	inwendiproses()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai
18	inventidakdire spon()	User Id : 15000	<i>ErrorResponse JSON</i>	Sesuai

Kesimpulan pada hasil pengujian di tabel 4.72 untuk data salah pada musyrif *service*, didapatkan hasil bahwa seluruh *request* mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan.

c. Pengujian data pada pengasuh service

Pada tabel 4.73 akan dilakukan pengujian dengan parameter data benar pada pengasuh service.

Tabel 4.73 Operasi *web services mahad testing* pengasuh data normal

Method operasi dan hasil uji (Data Normal)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	listsantrimabna()	-	<i>DataProfile JSON Response</i>	Sesuai
2	listiqob()	-	<i>DataProfile JSON Response</i>	Sesuai
3	listpresensi()	Id mabna : 10	<i>DataProfile JSON Response</i>	Sesuai
4	listpenjadwalanmualim()	Id mabna : 10	<i>JadwalMualim JSON Response</i>	Sesuai
5	listbulanmualim()	-	<i>JadwalMualim JSON Response</i>	Sesuai
6	listjurnalmualim()	Id jurnal : 12 Id bulan : 2	<i>JurnalMualim JSON Response</i>	Sesuai
7	listkehadiranbaik()	-	<i>DataProfile JSON Response</i>	Sesuai
8	listkehadiranburuk()	-	<i>DataProfile JSON Response</i>	Sesuai
9	mhspandai()	Jenis test : Afkar	<i>DataProfile JSON Response</i>	Sesuai
10	penerimaanmulyrif()	-	<i>DataProfile JSON Response</i>	Sesuai
11	penempatanmulyrif()	-	<i>DataProfile JSON Response</i>	Sesuai
12	listsetoran()	-	<i>DataProfile JSON Response</i>	Sesuai

Kesimpulan pada hasil pengujian di tabel 4.73 untuk data benar pada pengasuh service, didapatkan hasil bahwa seluruh request

mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan. Selanjutnya untuk pengujian pada data salah akan dijelaskan pada tabel 4.74.

Tabel 4.74 Operasi *web services mahad* testing pengasuh data salah

Method operasi dan hasil uji (Data Salah)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	listsantrimabna()	param	<i>ErrorResponse JSON</i>	Sesuai
2	listiqob()	param	<i>ErrorResponse JSON</i>	Sesuai
3	listpresensi()	Id mabna : 999	<i>ErrorResponse JSON</i>	Sesuai
4	listpenjadwalanmualim()	Id mabna : 999	<i>ErrorResponse JSON</i>	Sesuai
5	listbulanmualim()	param	<i>ErrorResponse JSON</i>	Sesuai
6	listjurnalmualim()	Id jurnal : 999	<i>ErrorResponse JSON</i>	Sesuai
		Id bulan : 999		
7	listkehadiranbairik()	param	<i>ErrorResponse JSON</i>	Sesuai
8	listkehadiranburuk()	param	<i>ErrorResponse JSON</i>	Sesuai
9	mhspandai()	Jenis test : xxx	<i>ErrorResponse JSON</i>	Sesuai
10	penerimaanmuysyirif()	param	<i>ErrorResponse JSON</i>	Sesuai
11	penempatanmuysyirif()	param	<i>ErrorResponse JSON</i>	Sesuai
12	listsetoran()	param	<i>ErrorResponse JSON</i>	Sesuai
13	listmabna()	param	<i>ErrorResponse JSON</i>	Sesuai

Kesimpulan pada hasil pengujian di tabel 4.74 untuk data salah pada pengasuh *service*, didapatkan hasil bahwa seluruh *request* mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan

d. Pengujian data pada global service

Pada tabel 4.75 akan dilakukan pengujian dengan parameter data benar pada *global service*.

Tabel 4.75 Operasi *web services mahad testing global service* data normal

Method operasi dan hasil uji (Data Normal)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	login()	Username : 1511	<i>Credential JSON Response</i>	Sesuai
		Password : 367295		
		Level : Pengasuh		

Kesimpulan pada hasil pengujian di tabel 4.75 untuk data benar pada *global service* untuk *method login*, didapatkan hasil bahwa *request* untuk *login* mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan. Selanjutnya untuk pengujian pada data salah akan dijelaskan pada tabel 4.76.

Tabel 4.76 Operasi *web services mahad testing global service* data salah

Method operasi dan hasil uji (Data Salah)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	login()	Username : 15	<i>ErrorResponse JSON</i>	Sesuai
		Password : 36		

		Level : param		
--	--	---------------	--	--

Kesimpulan pada hasil pengujian di tabel 4.76 untuk data salah pada *global service*, didapatkan hasil bahwa seluruh *request* mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan.

e. Pengujian data pada *news service*

Pada tabel 4.77 akan dilakukan pengujian dengan parameter data benar pada *news service*.

Tabel 4.77 Operasi *web services news feed testing* data benar

Method operasi dan hasil uji (Data Normal)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	getUrlFeed ()	Port : 80	<i>NewsData POJO</i>	Sesuai
		URL : <i>http://msaa.uin-malang.ac.id/feed</i>		

Kesimpulan pada hasil pengujian di tabel 4.77 untuk data benar pada *news service* untuk *method getUrlFeed()*, didapatkan hasil bahwa *request* tersebut mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan. Selanjutnya untuk pengujian pada data salah akan dijelaskan pada tabel 4.78.

Tabel 4.78 Operasi *web services news feed testing* data salah

Method operasi dan hasil uji (Data Salah)				
No	Nama Operasi	Parameter Input	Output yang diharapkan	Kesimpulan
1	getUrlFeed()	Port : 8080	<i>ErrorResponse</i>	Sesuai
		URL : <i>//ping.com</i>	<i>JSON</i>	

Kesimpulan pada hasil pengujian di tabel 4.78 untuk data salah pada *news service*, didapatkan hasil bahwa seluruh *request* mempunyai *output* yang sesuai dengan parameter dan *response* data yang diharapkan.

2. Pengujian *Blacbox*

Pengujian yang kedua pada penelitian ini yaitu pengujian pada sistem dengan menggunakan metode *blackbox*. *Blackbox* merupakan metode pengujian yang memfokuskan pada fungsionalitas dan hasil keluaran dari proses sebuah sistem. Skenario pengujian mengacu pada *use case* dalam sistem. Berikut tabel pengujian fungsionalitas sistem *mahad* dapat dilihat pada tabel 4.79.

Tabel 4.79 Tabel kriteria pengujian sistem

No	Nama Activity	Kriteria Evaluasi Hasil
1	Melihat <i>news feed</i>	Aplikasi dapat menampilkan halaman <i>news feed</i> berita dan update terbaru mengenai <i>mahad</i> sesuai dengan web resmi <i>mahad</i> .
2	<i>Login</i>	Aplikasi dapat menampilkan halaman dan form <i>login user</i> dan memeriksa apakah data login sudah benar.
3	Melihat <i>report</i> nilai ujian	Aplikasi dapat menampilkan <i>report</i> penilaian, baik nilai ujian UTS maupun UAS.
4	Melihat <i>report</i> inventaris	Aplikasi dapat menampilkan halaman <i>report</i> inventaris barang-barang yang berkaitan dengan <i>mabna</i> dan kamar santri.
5	Melihat <i>report</i> <i>mabna</i>	Aplikasi dapat menampilkan halaman <i>report</i> <i>mabna</i> yang berkaitan dengan presensi kehadiran, beserta mahsantri berprestasi.
6	Melihat <i>report</i> <i>musyrif</i>	Aplikasi dapat menampilkan halaman <i>report</i> <i>musyrif</i> berupa pengumuman penerimaan

		<i>musyrif</i> dan hasil ujian penerimaan <i>musyrif</i> .
7	Melihat <i>reportmurabbi</i>	Aplikasi dapat menampilkan halaman <i>report murabbi</i> berupa pembagian jadwal dan jurnal-jurnal <i>murabbi</i> .

Skenario ini dilakukan oleh pengguna dengan menginstall aplikasi *beta testing* pada *playstore* dan *email* yang sudah didaftarkan. Setelah itu pengguna mengisi kuesioner *online* yang telah diberikan. Pengujian terdiri dari pengujian fungsionalitas sistem dan antarmuka. Pengujian ini melibatkan 10 responden dari kalangan mahasiswa dan *musyrif*. Berikut hasil pengujian fungsionalitas sistem dan antarmuka sebagaimana terlihat pada tabel 4.80 dan tabel 4.81.

Tabel 4.80 Tabel pengujian fungsionalitas sistem

No	Nama Activity	SS	S	TS	STS
1	Aplikasi dapat menampilkan halaman <i>news feed</i> berita dan update terbaru mengenai <i>mahad</i> sesuai dengan web resmi <i>mahad</i> .	8	2	-	-
2	Aplikasi dapat menampilkan halaman dan form <i>login user</i> dan memeriksa apakah data <i>login</i> sudah benar.	1	9	-	-
3	Aplikasi dapat menampilkan <i>report</i> penilaian, baik nilai ujian UTS maupun UAS.	1	5	4	-
4	Aplikasi dapat menampilkan halaman <i>report</i> inventaris barang-barang yang berkaitan dengan <i>mabna</i> dan kamar santri.	1	5	4	-
5	Aplikasi dapat menampilkan halaman <i>report</i> <i>mabna</i> yang berkaitan dengan presensi kehadiran, beserta mahasiswa	1	4	5	-

	berprestasi.				
6	Aplikasi dapat menampilkan halaman <i>report musyrif</i> berupa pengumuman penerimaan <i>musyrif</i> dan hasil ujian penerimaan <i>musyrif</i> .	4	6	-	-
7	Aplikasi dapat menampilkan halaman <i>report murabbi</i> berupa pembagian jadwal dan jurnal-jurnal <i>murabbi</i> .	1	5	4	-
Total		17	36	17	

Tabel 4.81 Tabel pengujian antarmuka sistem

No	Nama Activity	SS	S	TS	STS
1	Layout antarmuka <i>user friendly</i> sehingga mempermudah user dalam menggunakan sistem	7	2	1	-
2	Pesan validasi form sangat mempermudah dalam proses menginput data	1	6	3	-
3	Menu navigasi mudah dipahami	5	4	1	-
4	Menu navigasi berfungsi dengan baik	7	3	-	-
Total		20	15	5	

Tabel 4.82 Tabel bobot nilai

Kategori	Nilai
SS	4
S	3
TS	2
STS	1

Dari hasil pengujian sistem yang diketahui melalui tabel 4.80 dan tabel 4.81, diperoleh hasil dari total 10 responden yang menjawab seluruh

pertanyaan. Untuk menghitung hasil dari seluruh responden tersebut, digunakan skala likert yang sudah dipaparkan pada bab sebelumnya.

Untuk mendapatkan hasil interpretasi, harus diketahui dulu skor tertinggi (X) dan angka terendah (Y) untuk item penilaian dengan rumus sebagai berikut :

- a. $Y = \text{Skor tertinggi likert} \times \text{jumlah responden (Angka Tertinggi 4)}.$
- b. $X = \text{Skor terendah likert} \times \text{jumlah responden (Angka Terendah 1)}.$

Jumlah skor tertinggi untuk item **SS** untuk fungsional ialah 4×10 (responden) = 40, sedangkan item **STS** ialah $1 \times 10 = 10$. Maka penilaian interpretasi responden terhadap kedua kasus tersebut adalah hasil nilai yang dihasilkan dengan menggunakan rumus Index %. Rumus Index % = Total Skor / Y x 100. Sebelum itu juga perlu diketahui rumus interval dengan rumus $I = 100 / \text{Jumlah Skor (Likert)}$, yaitu $100 / 4 = 25$, maka rentang kriterianya adalah sebagai berikut :

Angka 0% – 24,99% = Tidak setuju / Kurang baik

Angka 25% – 49,99% = Cukup / Netral

Angka 50% – 74,99% = Setuju/Baik/suka

Angka 75% – 100% = Sangat (setuju/Baik/Suka)

Maka penyelesaian akhir untuk fungsionalitas dan antarmuka adalah:

Tabel 4.83 Tabel hasil akhir penilaian fungsionalitas

No	Nama activity	SS	S	TS	ST S	Hasil
1	Aplikasi dapat menampilkan halaman <i>news feed</i> berita dan update	$8 \times 4 =$ 32	$2 \times 3 =$ 6	-	-	$38 / Y \times 100$ = $38 / 40 \times 100$ =

	terbaru mengenai <i>mahad</i> sesuai dengan web resmi <i>mahad</i> .					95% = Sangat Setuju
2	Aplikasi dapat menampilkan halaman dan form <i>login user</i> dan memeriksa apakah data <i>login</i> sudah benar.	1x4 = 4	9x3 = 27	-	-	31 / Y x 100 = 31 / 40 * 100 = 77,5 % = Sangat Setuju
3	Aplikasi dapat menampilkan <i>report</i> penilaian, baik nilai ujian UTS maupun UAS.	1x4 = 4	5x3 = 15	4x2 = 8	-	27 / Y x 100 = 27 / 40 * 100 = 67,5% = Setuju
4	Aplikasi dapat menampilkan halaman <i>report</i> inventaris barang-barang yang berkaitan dengan mabna dan kamar santri.	1x4 = 4	5x3 = 15	4x2 = 8	-	27 / Y x 100 = 27 / 40 * 100 = 67,5% = Setuju
5	Aplikasi dapat menampilkan halaman <i>report</i> mabna yang berkaitan dengan presensi kehadiran, beserta mahasiswa	1x4 = 4	4x3 = 12	5x2 = 10	-	26 / Y x 100 = 26 / 40 * 100 = 65% = Setuju

	berprestasi.					
6	Aplikasi dapat menampilkan halaman <i>report musyrif</i> berupa pengumuman penerimaan <i>musyrif</i> dan hasil ujian penerimaan <i>musyrif</i> .	4x4 = 16	6x3 = 18	-	-	34 / Y x 100 = 34 / 40 * 100 = 85% = Setuju
7	Aplikasi dapat menampilkan halaman <i>report murabbi</i> berupa pembagian jadwal dan jurnal-jurnal <i>murabbi</i> .	1x4 = 4	5x3 = 15	4x2 = 8	-	27 / Y x 100 = 27 / 40 * 100 = 67,5% = Setuju
Total						75% = Sangat Setuju

Tabel 4.84 Tabel hasil akhir penilaian antarmuka

No	Nama Activity	SS	S	TS	STS	Hasil
1	Layout antarmuka <i>user friendly</i> sehingga mempermudah user dalam menggunakan sistem	7x4 = 28	2x3 = 6	1x2 = 2	-	36 / Y x 100 = 36 / 40 * 100 = 90% = Sangat Setuju
2	Pesan validasi form sangat mempermudah dalam proses	1x4 = 4	6x3 = 18	3x2 = 6	-	28 / Y x 100 = 28 / 40 * 100 =

	menginput data					70% = Setuju
3	Menu navigasi mudah dipahami	5x4 = 20	4x3 = 12	1x2 = 2	-	34/ Y x 100 = 34/ 40 * 100 = 85% = Sangat Setuju
4	Menu navigasi berfungsi dengan baik	7x4 = 28	3x3 = 9	-	-	37/ Y x 100 = 37/ 40 * 100 = 92,5% = Sangat Setuju
Total						84,37% = Sangat Setuju

Berdasarkan hasil pengujian fungsionalitas dan antarmuka diatas maka didapat nilai 75%, yang berarti *user* sangat setuju bahwa fungsionalitas berjalan dengan baik dan 84% *user* setuju bahwa antarmuka berjalan dengan baik dan mudah digunakan.

Berdasarkan kesesuaian hasil pengujian *web service* juga dapat diambil kesimpulan bahwa sistem *mahad* dengan SOA telah bisa mengatasi permasalahan *loosely coupled* dan *interoperability* sistem. Selanjutnya, berdasarkan hasil pengujian sistem juga sudah layak untuk digunakan.

Tetapi perlu ada pengembangan yang lebih serius terkait dengan desain tampilan, level integrasi dan keamanan sistem untuk memperoleh hasil yang lebih baik dan lebih maksimal.

4.5. Integrasi Sistem Mahad Dalam Pandangan Islam

Di dalam ajaran Islam, seseorang muslim diharuskan untuk menghargai waktu dalam hidupnya, hal ini dijelaskan pula dalam surat QS. Al-Ashr : 1-3 bahwa seseorang manusia dalam kerugian apabila tidak memanfaatkan waktu yang telah diberikan oleh Allah SWT dengan baik.

وَالْعَصْرِ ۝ إِنَّ الْإِنْسَانَ لَفِي خُسْرٍ ۝ إِلَّا الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ
وَتَوَاصَوْا بِالْحَقِّ وَتَوَاصَوْا بِالصَّبْرِ ۝

Arti dan tafsir Jalalain :

1. *Demi masa* (atau zaman atau waktu yang dimulai dari tergelincirnya matahari hingga terbenamnya, yang dimaksud adalah waktu sholat ashar).
2. *Sesungguhnya manusia itu* (yang dimaksud adalah jenis manusia) *benar-benar dalam kerugian* (dalam perniagaan dan aktifitasnya),
3. *kecuali orang-orang yang beriman dan mengerjakan amal saleh* (mereka tidak termasuk orang-orang yang merugi dalam perniagaan dan aktifitasnya) *dan nasehat menasehati* (artinya sebagian diantara mereka menasehati sebagian lainnya) *supaya mentaati kebenaran* (yaitu iman) *dan nasehat menasehati supaya menetapi kesabaran* (yaitu dalam menjalankan amal ketaatan dan menjauhi kemaksiatan).

Pada seluruh integrasi sistem informasi mahad ini diharapkan bisa mempercepat proses dan membuat seluruh pekerjaan menjadi lebih efisien dan efektif sehingga bisa menjadi lebih baik.

Hal ini pula dijelaskan dan ditegaskan pula dalam ayat lain bahwa seorang muslim diharuskan menggunakan waktu dan dapat mengambil pelajaran dan selalu bersyukur, yaitu surat Al-Furqan ayat 62 berikut :

وَهُوَ الَّذِي جَعَلَ اللَّيْلَ وَالنَّهَارَ خِلْفَةً لِّمَنۢ أَرَادَ أَنۢ يَذَّكَّرَ أَوْ أَرَادَ شُكُورًا

Arti dan tafsir *Jalalain* :

Dan Dia (pula) yang menjadikan malam dan siang silih berganti (yakni satu sama lainnya saling silih berganti dengan yang lainnya) bagi orang yang ingin mengambil pelajaran (dapat dibaca Yadzdzakkara dan Yadzakura, yang pembahasannya sebagaimana pada ayat sebelumnya. Yakni, ia ingat akan kebaikan yang tidak dilakukan pada salah satu di antaranya, kemudian ia melakukan pada waktu yang lainnya, sebagai ganti dari apa yang tidak dilakukannya di waktu yang pertama tadi) atau orang yang ingin bersyukur (atas nikmat Rabb yang telah dilimpahkan kepadanya pada dua waktu itu).

BAB V

KESIMPULAN DAN SARAN

Pada bab terakhir ini, diperoleh kesimpulan dari hasil penelitian yang sudah dilakukan, beserta saran agar penelitian ini dapat dikembangkan lebih baik kedepannya.

5.1. Kesimpulan

Dari hasil pengamatan selama analisis perancangan, implementasi dan pengujian dapat diambil kesimpulan bahwa :

1. SOA dengan teknologi *web services (REST web service)* dapat diterapkan didalam pengembangan aplikasi sistem informasi mahad Al-Jamiah UIN Malang. Implementasi SOA dapat dilakukan dengan dua tahap proses, yaitu dengan melakukan *breakdown* proses bisnis sehingga terbentuk beberapa *service*, kemudian dilakukan *breakdown* terhadap *service* yang sudah terbentuk menjadi *subtask request*.
2. Pembagian layanan *service* dengan SOA dapat dilakukan dengan cara *identifikasi subtask request* yang sudah dibuat, dilanjutkan dengan pengkodean *web service*.
3. Implementasi aplikasi mobile *Smart-Mahad* dilakukan dengan cara menggunakan API yang sudah tersedia pada *web service*. Hasil pengujian *acceptance test* kepada *user* menunjukkan bahwa 75% fungsionalitas berjalan baik dan 84% memiliki tampilan yang mudah dipahami.

5.2. Saran

Sistem *mahad* yang telah dibangun ini tentunya tidak lepas dari kekurangan. Terutama dalam metode integrasi sistem, oleh karena itu peneliti menyarankan beberapa hal, sebagai berikut :

1. Menerapkan model EAI (*Enterprise Application Integration*) untuk mengatur pola integrasi dengan banyak *service*.
2. Aplikasi dan sistem *web service* yang digunakan untuk selalu di kembangkan menggunakan model *framework* terbaru agar keamanan sistem tetap terjaga.

DAFTAR PUSTAKA

- Al-Mahalli, Syaikh Jalaluddin Mukhammad, Al-Imam Jalaluddin Abdurrahman As-Suyuthi, 2010, *Tafsir Al-Jalalain*, Penerbit Eiba.
- Dudhe, Anil, 2014, *Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment*, India : Second National Conference on Recent Trends in Information Security, GHRCE, Nagpur, India.
- Erl, T., 2005, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, Upper Saddle River, New Jersey 07458.
- Fielding, Roy, 2000, *Representational State Transfer (REST)*, http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm , Diakses tanggal 22 Februari 2015.
- Hohpe, Gregor, 2003, *Hub and Spoke [or] Zen and the Art of Message Broker Maintenance*, http://www.eaipatterns.com/ramblings/03_hubandspoke.html diakses tanggal 16 Februari 2015.
- Hohpe, Gregor., Bobby, Wolf., 2003, *Enterprise Integration Pattern, Designing, Building, and Deploying Message Solution*, FlyHeart.com.
- Juric, M.B., Loganathan, R., Sarang, P., dan Jennings, F., 2007, *SOA Approach to Integration*, Packt Publishing, Birmingham, B27 6PA, UK.
- Juric, M.B., Chandrasekaran, S., Frece, A., Hertis, M., dan Srdic, G., 2010, *WS-BPEL 2.0 for SOA Composite Applications with IBM WebSphere 7*, Packt Publishing, Birmingham, B27 6PA, UK.
- Khoirul, Azizi, 2012, *Rancang Bangun Sistem Informasi Apotek Terintegrasi Menggunakan Service Oriented Architecture*, Yogyakarta : Skripsi Teknik Informatika UIN Sunan Kalijaga.
- Narangoda, Isuru Udana Loku, 2014, *Service Orchestration with WSO2 ESB*, <http://wso2.com/library/articles/2014/02/service-orchestration-with-wso2-esb/> diakses tanggal 22 Februari 2015.
- Oktalifryan, Aditya, 2011, *Rancang Bangun Aplikasi Instalasi Rawat Jalan Dengan Paradigma Pengembangan Terintegrasi Menggunakan Enterprise Service Bus (ESB)*, Undergraduate Thesis of Information System, RSSI 658.403 801 1.

- Putra, Charisma, 2011, *Implementasi Kriptografi Untuk Pengamanan Data Sensitif Pada Aplikasi Rekam Medis*, Undergraduate Thesis, Informations System, RSSI 005.82 Chai, 2011.
- Putra, Kholid Abdillah, 2015, *Sistem Informasi Kesantrian Ma'had Sunan Ampel Al-Aly*, Skripsi, Jurusan Teknik Informatika UIN Maulana Malik Ibrahim Malang.
- Siriwardena, Prabath, 2012, *WSO2 Enterprise Service Bus*, <https://svn.wso2.org/repos/wso2/people/prabath/esb/wso2-esb.pdf> diakses tanggal 16 Februari 2015.
- Tjakrawala, F.X Kurniawan, 2002, *Fenomena Kesenjangan Yang Terjadi Dalam Hubungan Antara Sistem Informasi Dan Teknologi Informasi Serta Upaya Menjembatannya: Suatu Tinjauan Dari Perspektif Teori Ekonomi*, Jakarta: Jurnal Ekonomi/Fe-Untar/01/2002.
- Utomo, Wiranto Herry, 2012, *Penerapan Enterprise Service Bus (ESB) Sebagai Middleware Integrasi Berbasis SOA*, Yogyakarta : Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012).