

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN
MASJID AL GHIFARI BERBASIS WEB**

SKRIPSI

Untuk memenuhi sebagian Persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Rezky Sayuthi Putra

NIM: 155150201111120



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2021



PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN MASJID AL GHIFARI
BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Rezky Sayuthi Putra
NIM: 155150201111120

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II




Nurudin Santoso, S.T., M.T.
NIP: 19740916 200012 1 001

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

Mengetahui

Ketua Jurusan Teknik Informatika




Achmad Basuki, S.T., M.MG., Ph.D.
NIP: 19741118 200312 1 002



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah dan disebutkan dalam daftar referensi.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2000, Pasal 25 ayat 2 dan Pasal 70).

Malang, 19 Juni 2021



Muhammad Rezky Sayuthi Putra



PRAKATA

Puji syukur kepada Allah SWT yang telah memberikan nikmat rahmat, taufik, kesehatan, dan rezeki sehingga laporan skripsi yang berjudul “Pengembangan Sistem Manajemen Masjid Al Ghifari Berbasis Web” ini dapat terselesaikan. Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Nurudin Santoso, S.T., M.T., dan bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku pembimbing skripsi yang telah sabar dan ikhlas membimbing serta mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Kedua orang tua penulis, Muh Sayuthi, S.E., dan Ir. Suwartini, M.M. serta keluarga besar atas segala kasih sayang, nasihat, perhatian, kesabaran, perhatian, dan doa yang sudah diberikan dalam mendidik penulis sehingga dapat menyelesaikan skripsi ini,
3. Bapak Adhitya Bhawiyuga, S.Kom., M.Sc. selaku Ketua Program Studi Teknik Informatika,
4. Bapak Achmad Basuki, S.T., M.Mg., Ph.D. selaku Ketua Jurusan Teknik Informatika,
5. Seluruh dosen Fakultas Ilmu Komputer yang sudah banyak memberikan bantuan ilmu dan dukungan selama penulis berada di bangku perkuliahan sehingga skripsi ini dapat terselesaikan,
6. Alm. Bapak Agus Junaedi selaku narasumber dari Masjid AL Ghifari yang sangat baik dan sabar dalam memberikan bantuan kepada penulis,
7. Raehan Azalia Hidayat yang memberikan semangat,
8. Teman-teman Teknik Informatika 2015, Dara Daeng, Dara Daeng 2015, CL BERSIN, korlap Olimpiade Brawijaya 2017, dan korlap PK2 MABA FILKOM yang menemani penulis dan memberikan warna-warni persahabatan selama di Malang. Teman-teman terkhusus, yaitu Hilmy, Abang, Fadel, Sadino, Capy, Commas, Fira, Kifli, Rais, Kak Sule, Dimas, Fina, dan Jeje yang selalu memberikan bantuan dan semangat sehingga penulis dapat menyelesaikan skripsi ini,

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang berikan dapat membangun penulis menjadi lebih baik. Akhir kata dari penulis semoga skripsi ini dapat membawa manfaat bagi seluruh pihak yang menggunakannya.

Malang, 19 Juni 2021

Penulis
Rezky Muhammad13@gmail.com

ABSTRAK

Muhammad Rezky Sayuthi Putra, Pengembangan Sistem Informasi Manajemen Masjid Al Ghifari Berbasis Web

Pembimbing: Nurudin Santoso, S.T., M.T., dan Tri Astoto Kurniawan, S.T., M.T., Ph.D.

Masjid adalah tempat yang biasa digunakan umat Islam untuk beribadah. Masjid juga dapat berfungsi sebagai pusat penyebaran pengetahuan agama Islam, pemberdayaan umat Islam, dan kegiatan masyarakat. Fungsi masjid harus berjalan dengan baik dan optimal, sehingga keberadaan masjid dapat memberikan manfaat bagi masyarakat. Hal ini harus dilakukan melalui proses tata kelola yang baik. Masjid Al Ghifari merupakan salah satu masjid yang ada di Kecamatan Lowokwaru, Kota Malang. Manajemen Masjid Al Ghifari terdiri manajemen kegiatan dan manajemen keuangan. Manajemen Kegiatan terdiri dari pengelolaan informasi kegiatan rutin, jadwal salat, dan kegiatan insidental. Manajemen keuangan terdiri dari pencatatan pemasukan dan pengeluaran keuangan masjid. Permasalahan pada manajemen kegiatan Masjid Al Ghifari salah satunya adalah publikasi informasi kegiatan yang ada menggunakan media sosial, yaitu Instagram dan Facebook. Kekurangan penyampaian informasi menggunakan Instagram dan Facebook tidak semua informasi tentang kegiatan dapat tersampaikan dan hanya orang-orang yang memiliki media sosial tersebut yang mendapatkan informasi. Permasalahan pada manajemen keuangan Masjid Al Ghifari salah satunya yaitu pencatatan pemasukan dan pengeluaran keuangan masih dicatat menggunakan buku. Kekurangan pencatatan laporan keuangan menggunakan buku yaitu hilang atau mengalami kerusakan. Solusi permasalahan yang ada, maka dikembangkan sebuah sistem informasi manajemen masjid berbasis web. Pengembangan sistem ini menggunakan model *waterfall* dan menggunakan pendekatan *object-oriented*, serta dibangun menggunakan *framework CodeIgniter*. Sistem ini sudah diuji menggunakan pengujian unit, integrasi, dan validasi. Pengujian unit dan integrasi menggunakan teknik pengujian *white-box testing*, serta pengujian validasi dilakukan menggunakan teknik pengujian *black-box testing* dengan masing-masing pengujian bernilai 100% valid. Sistem ini dapat dijalankan di berbagai macam *browser*, seperti *Google Chrome*, *Firefox*, dan *Opera*. Hasil dari penelitian ini adalah sistem informasi manajemen masjid berbasis web.

Kata Kunci: manajemen masjid, sistem informasi, *waterfall*, *object oriented*, web

ABSTRACT

Muhammad Rezky Sayuthi Putra, *Web-Based Al Ghifari Mosque Management Information System Development*

Supervisors: Nurudin Santoso, S.T., M.T., dan Tri Astoto Kurniawan, S.T., M.T., Ph.D.

Mosque is a place used by Muslims to pray. Mosques can also function as the centers of Islam, empowerment of Muslims, and community activities. The function of the mosque should run properly and optimally to provide benefits for the community. This must be done through a good governance process. Al Ghifari Mosque is one of the mosques in Lowokwaru District, Malang City. Al Ghifari Mosque management consists of activity management and financial management. Activity Management consists of managing information on routine activities, prayer schedules, and incidental activities. Financial management consists of recording the mosque's financial income and expenditure. The problem with the misuse of Al Ghifari Mosque's activity management is the publication of activity information on social media, namely Instagram and Facebook. Lack of information delivery using Instagram and Facebook not all information about activities can be conveyed and only people with these social media get the information. One of the problems in the financial management of the Al Ghifari Mosque is the recording of financial income and expenditure which is recorded using a manual book. The disadvantage of recording financial statements using books is that they are lost or damaged. To solve the existing problems, a web-based mosque management information system was developed. The development of this system uses the waterfall model and uses an object-oriented approach, and built using the CodeIgniter framework. The system has been tested using unit testing, integration, and validation. Unit and integration testing uses white-box testing techniques, and tests are validated using black-box testing techniques with each test being 100% valid. This system can be run on various browsers, such as Google Chrome, Firefox, and Opera. The result of this research is a web-based mosque management information system.

Keywords: *mosque management, information system, waterfall, object-oriented, web*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Manajemen Masjid Al Ghifari	6
2.3 Rekayasa Perangkat Lunak	10
2.3.1 Model <i>Waterfall</i>	11
2.3.2 Pendekatan Berorientasi Objek	15
2.3.2.1 <i>Use Case Diagram</i>	16
2.3.2.2 <i>Sequence Diagram</i>	19
2.3.2.3 <i>Class Diagram</i>	21
2.4 Teknologi Pengembangan Sistem	22
2.4.1 MySQL	22
2.4.2 <i>Framework CodeIgniter</i>	23
BAB 3 METODOLOGI PENELITIAN	25
3.1 Deskripsi Umum	25
3.2 Studi Literatur	25
3.3 Analisis Kebutuhan	26

3.4 Perancangan	26
3.5 Implementasi	27
3.6 Pengujian	27
3.7 Kesimpulan dan Saran	28
BAB 4 REKAYASA KEBUTUHAN.....	29
4.1 Elisitasi Kebutuhan.....	29
4.1.1 Identifikasi Aktor	29
4.1.2 Kebutuhan Fungsional.....	29
4.1.3 Kebutuhan Nonfungsional	32
4.2 Spesifikasi Kebutuhan	32
4.2.1 Use Case Diagram	41
4.2.2 Use Case Scenario	42
BAB 5 PERANCANGAN DAN IMPLEMENTASI	63
5.1 Perancangan.....	63
5.1.1 Perancangan Arsitektur	63
5.1.1.1 Pemodelan Sequence Diagram.....	63
5.1.1.2 Class Diagram.....	65
5.1.2 Perancangan Data.....	67
5.1.3 Perancangan Komponen.....	69
5.1.3.1 Perancangan Komponen Klas Kegiatan_Controller.....	69
5.1.3.2 Perancangan Komponen Klas Kegiatan_Controller.....	70
5.1.3.3 Perancangan Komponen Klas Keuangan_harian_Controller.....	71
5.1.4 Perancangan Antarmuka	72
5.1.4.1 Perancangan Antarmuka Halaman Tambah Kegiatan Insidental	72
5.1.4.2 Perancangan Antarmuka Halaman Ubah Kegiatan Insidental	74
5.1.4.3 Perancangan Antarmuka Halaman Tambah Keuangan Harian	77
5.2 Implementasi	79
5.2.1 Spesifikasi Sistem.....	79
5.2.2 Implementasi Kode Program.....	80
5.2.2.1 Implementasi method tambah_kegiatan_aksi dari klas	
Kegiatan_insidental_control.....	80
5.2.2.2 Implementasi method ubah_kegiatan_aksi dari klas	
Kegiatan_insidental_controller.....	82



5.2.2.3	Implementasi method tambah_harian_aksi dari kelas Keuangan_harian_control.....	83
5.2.3	Implementasi Data.....	84
5.2.4	Implementasi Antarmuka.....	85
5.2.4.1	Implementasi Antarmuka Halaman Tambah Kegiatan Insidental.....	85
5.2.4.2	Implementasi Antarmuka Halaman Ubah Kegiatan Insidental.....	85
5.2.4.3	Implementasi Antarmuka Halaman Tambah Keuangan Harian.....	86
BAB 6 PENGUJIAN.....		87
6.1	Pengujian Unit.....	87
6.1.1	Pengujian unit <i>method</i> tambah_kegiatan_insidental_aksi() kelas Kegiatan_insidental_control.....	87
6.1.2	Pengujian unit <i>method</i> ubah_kegiatan_aksi() kelas Kegiatan_insidental_control.....	91
6.1.3	Pengujian unit <i>method</i> tambah_harian_aksi() kelas Keuangan_harian_aksi.....	92
6.2	Pengujian Integrasi.....	95
6.2.1	Pengujian integrasi <i>method</i> tambah_kegiatan_aksi() kelas Kegiatan_insidental_control.....	95
6.2.2	Pengujian integrasi <i>method</i> ubah_kegiatan_aksi() kelas Kegiatan_insidental_control.....	99
6.2.3	Pengujian integrasi <i>method</i> tambah_harian_aksi() Kelas Keuangan_harian_control.....	101
6.3	Pengujian Validasi.....	104
6.3.1	Pengujian validasi <i>login</i>	104
6.3.2	Pengujian validasi <i>logout</i>	106
6.3.3	Pengujian validasi tambah artikel.....	107
6.3.4	Pengujian validasi ubah artikel.....	108
6.3.5	Pengujian validasi hapus artikel.....	109
6.3.6	Pengujian validasi tambah pengumuman.....	110
6.3.7	Pengujian validasi ubah pengumuman.....	111
6.3.8	Pengujian validasi hapus pengumuman.....	112
6.3.9	Pengujian validasi tambah foto.....	113

6.3.10	Pengujian validasi hapus foto.....	114
6.3.11	Pengujian validasi tambah kegiatan insidental.....	115
6.3.12	Pengujian validasi ubah kegiatan insidental.....	116
6.3.13	Pengujian validasi hapus kegiatan insidental.....	118
6.3.14	Pengujian validasi tambah kegiatan rutin.....	119
6.3.15	Pengujian validasi ubah kegiatan rutin.....	120
6.3.16	Pengujian validasi hapus kegiatan rutin.....	121
6.3.17	Pengujian validasi tambah imam salat.....	122
6.3.18	Pengujian validasi ubah imam salat.....	123
6.3.19	Pengujian validasi hapus imam salat.....	124
6.3.20	Pengujian validasi tambah jadwal salat.....	125
6.3.21	Pengujian validasi ubah jadwal salat.....	127
6.3.22	Pengujian validasi hapus jadwal salat.....	128
6.3.23	Pengujian validasi tambah keuangan harian.....	129
6.3.24	Pengujian validasi ubah keuangan harian.....	130
6.3.25	Pengujian validasi hapus keuangan harian.....	131
6.3.26	Pengujian validasi cetak keuangan harian.....	132
6.3.27	Pengujian validasi cetak laporan keuangan bulanan.....	133
6.3.28	Pengujian validasi lihat detail dan daftar kegiatan insidental jalur utama.....	134
6.3.29	Pengujian validasi lihat detail dan daftar kegiatan rutin jalur utama.....	135
6.3.30	Pengujian validasi lihat detail dan daftar pengumuman jalur utama.....	136
6.3.31	Pengujian validasi lihat detail dan daftar artikel jalur utama.....	137
6.3.32	Pengujian validasi lihat daftar foto jalur utama.....	137
6.3.33	Pengujian validasi lihat daftar imam salat jalur utama.....	138
6.3.34	Pengujian validasi lihat jadwal salat jalur utama.....	139
6.3.35	Pengujian validasi lihat laporan keuangan harian jalur utama.....	140
6.3.36	Pengujian validasi lihat laporan keuangan bulanan jalur utama.....	141
6.3.37	Pengujian <i>Compatibility</i>	142



BAB 7 KESIMPULAN DAN SARAN	143
7.1 Kesimpulan.....	143
7.2 Saran	143
DAFTAR REFERENSI.....	144
LAMPIRAN	146



DAFTAR TABEL

Tabel 2.1 Notasi Pada <i>Use Case Diagram</i>	17
Tabel 2.2 Bagian-bagian pada <i>use case scenario</i>	18
Tabel 2.3 Notasi pada <i>Sequence Diagram</i>	20
Tabel 2.4 Notasi pada <i>Class Diagram</i>	21
Tabel 4.1 Identifikasi Aktor	29
Tabel 4.2 Daftar Kebutuhan Fungsional Sistem Manajemen Masjid Al Ghifari....	30
Tabel 4.3 Spesifikasi Kebutuhan Fungsional	32
Tabel 4.4 Spesifikasi Kebutuhan Nonfungsional	40
Tabel 4.5 <i>Use case scenario</i> untuk <i>login</i>	42
Tabel 4.6 <i>Use case scenario</i> untuk <i>logout</i>	42
Tabel 4.7 <i>Use case scenario</i> untuk tambah artikel	43
Tabel 4.8 <i>Use case scenario</i> untuk ubah artikel.....	43
Tabel 4.9 <i>Use case scenario</i> untuk hapus artikel	44
Tabel 4.10 <i>Use case scenario</i> untuk tambah pengumuman	44
Tabel 4.11 <i>Use case scenario</i> untuk ubah pengumuman	45
Tabel 4.12 <i>Use case scenario</i> untuk hapus pengumuman.....	46
Tabel 4.13 <i>Use case scenario</i> untuk tambah foto.....	46
Tabel 4.14 <i>Use case scenario</i> untuk hapus foto.....	47
Tabel 4.15 <i>Use case scenario</i> untuk tambah kegiatan Insidental.....	47
Tabel 4.16 <i>Use case scenario</i> untuk ubah kegiatan insidental	48
Tabel 4.17 <i>Use case scenario</i> untuk hapus kegiatan insidental.....	49
Tabel 4.18 <i>Use case scenario</i> untuk tambah kegiatan rutin.....	49
Tabel 4.19 <i>Use case scenario</i> untuk ubah kegiatan rutin	50
Tabel 4.20 <i>Use case scenario</i> untuk hapus kegiatan rutin.....	51
Tabel 4.21 <i>Use case scenario</i> untuk tambah Imam Salat	51
Tabel 4.22 <i>Use case scenario</i> untuk ubah imam salat	52
Tabel 4.23 <i>Use case scenario</i> untuk hapus imam salat.....	52
Tabel 4.24 <i>Use case scenario</i> untuk tambah jadwal salat	53
Tabel 4.25 <i>Use case scenario</i> untuk ubah jadwal salat.....	54
Tabel 4.26 <i>Use case scenario</i> untuk hapus jadwal salat	54
Tabel 4.27 <i>Use case scenario</i> untuk tambah laporan keuangan harian	55

Tabel 4.28 <i>Use case scenario</i> untuk ubah laporan keuangan harian	55
Tabel 4.29 <i>Use case scenario</i> untuk hapus laporan keuangan harian	56
Tabel 4.30 <i>Use case scenario</i> untuk cetak laporan keuangan harian	57
Tabel 4.31 <i>Use case scenario</i> untuk cetak laporan keuangan bulanan	57
Tabel 4.32 <i>Use case scenario</i> untuk lihat detail kegiatan rutin	58
Tabel 4.33 <i>Use case scenario</i> untuk lihat Foto.....	58
Tabel 4.34 <i>Use case scenario</i> untuk lihat daftar dan detail artikel.....	58
Tabel 4.35 <i>Use case scenario</i> untuk lihat daftar dan detail kegiatan insidental ..	59
Tabel 4.36 <i>Use case scenario</i> untuk lihat daftar dan detail pengumuman	59
Tabel 4.37 <i>Use case scenario</i> untuk lihat daftar Imam Salat.....	60
Tabel 4.38 <i>Use case scenario</i> untuk lihat daftar jadwal salat	60
Tabel 4.39 <i>Use case scenario</i> untuk lihat laporan keuangan harian.....	60
Tabel 4.40 <i>Use case scenario</i> untuk lihat laporan keuangan bulanan.....	61
Tabel 5.1 <i>Pseudocode method</i> tambah_kegiatan_aksi.....	69
Tabel 5.2 <i>Pseudocode method</i> ubah_kegiatan_aksi.....	70
Tabel 5.3 <i>Pseudocode Method</i> tambah_harian_aksi.....	71
Tabel 5.4 Penjelasan Bagian Antarmuka Halaman Tambah Kegiatan Insidental ..	72
Tabel 5.5 Penjelasan Bagian Antarmuka Halaman Ubah Kegiatan Insidental.....	75
Tabel 5.6 Penjelasan Bagian Antarmuka Halaman Tambah Keuangan harian	77
Tabel 5.7 Spesifikasi Perangkat Keras	79
Tabel 5.8 Spesifikasi Perangkat Lunak	80
Tabel 5.9 <i>Sourcecode method</i> tambah_kegiatan_aksi	80
Tabel 5.10 <i>Sourcecode method</i> ubah_kegiatan_aksi.....	82
Tabel 5.11 <i>Sourcecode method</i> tambah_harian_aksi.....	83
Tabel 6.1 <i>Pseudocode method</i> tambah_kegiatan_aksi.....	87
Tabel 6.2 Kasus uji dan hasil uji <i>method</i> tambah_kegiatan_insidental_aksi.....	89
Tabel 6.3 <i>Pseudocode method</i> ubah_kegiatan_aksi.....	91
Tabel 6.4 Kasus uji dan hasil uji <i>method</i> ubah_kegiatan_aksi.....	92
Tabel 6.5 <i>Pseudocode method</i> tambah_harian_aksi.....	93
Tabel 6.6 Kasus uji dan hasil uji <i>method</i> tambah_harian_aksi	94
Tabel 6.7 <i>Pseudocode method</i> tambah_kegiatan_aksi.....	96
Tabel 6.8 Kasus uji dan hasil uji <i>method</i> tambah_kegiatan_insidental_aksi.....	98



Tabel 6.9 <i>Pseudocode method</i> ubah_kegiatan_aksi.....	99
Tabel 6.10 Kasus uji dan hasil uji <i>method</i> ubah_kegiatan_aksi.....	101
Tabel 6.11 <i>Pseudocode method</i> tambah_harian_aksi.....	101
Tabel 6.12 Kasus uji dan hasil uji <i>method</i> tambah_harian_aksi.....	103
Tabel 6.13 Kasus uji <i>Login</i> jalur utama.....	105
Tabel 6.14 Kasus uji <i>Login</i> jalur alternatif 1a.....	105
Tabel 6.15 Kasus uji <i>Login</i> jalur alternatif 1b.....	106
Tabel 6.16 Kasus uji <i>Logout</i> jalur utama.....	106
Tabel 6.17 Kasus uji <i>Logout</i> jalur alternatif 2a.....	106
Tabel 6.18 Kasus uji tambah artikel jalur utama.....	107
Tabel 6.19 Kasus uji tambah artikel jalur alternatif 5a.....	107
Tabel 6.20 Kasus uji ubah artikel jalur utama.....	108
Tabel 6.21 Kasus uji ubah artikel jalur alternatif 4a.....	108
Tabel 6.22 Kasus uji hapus artikel jalur utama.....	109
Tabel 6.23 Kasus uji hapus artikel jalur alternatif 3a.....	110
Tabel 6.24 Kasus uji tambah pengumuman jalur utama.....	110
Tabel 6.25 Kasus uji tambah pengumuman jalur alternatif 5a.....	111
Tabel 6.26 Kasus uji ubah pengumuman jalur utama.....	111
Tabel 6.27 Kasus uji ubah pengumuman jalur alternative 4a.....	112
Tabel 6.28 Kasus uji hapus pengumuman jalur utama.....	112
Tabel 6.29 Kasus uji hapus pengumuman jalur alternatif 3a.....	113
Tabel 6.30 Kasus uji tambah foto jalur utama.....	113
Tabel 6.31 Kasus uji tambah foto jalur alternatif 3a.....	114
Tabel 6.32 Kasus uji hapus foto jalur utama.....	114
Tabel 6.33 Kasus uji hapus foto jalur alternatif 4a.....	115
Tabel 6.34 Kasus uji tambah kegiatan insidental jalur utama.....	115
Tabel 6.35 Kasus uji tambah kegiatan insidental jalur alternatif 5a.....	116
Tabel 6.36 Kasus uji ubah kegiatan insidental jalur utama.....	117
Tabel 6.37 Kasus uji ubah kegiatan insidental jalur alternatif 4a.....	117
Tabel 6.38 Kasus uji hapus kegiatan insidental jalur utama.....	118
Tabel 6.39 Kasus uji hapus kegiatan insidental jalur alternatif 3a.....	118
Tabel 6.40 Kasus uji tambah kegiatan rutin jalur utama.....	119



Tabel 6.41 Kasus uji tambah kegiatan rutin jalur alternatif 5a.....	119
Tabel 6.42 Kasus uji ubah kegiatan rutin jalur utama.....	120
Tabel 6.43 Kasus uji ubah kegiatan rutin jalur alternatif 4a	120
Tabel 6.44 Kasus uji hapus kegiatan rutin jalur utama.....	121
Tabel 6.45 Kasus uji hapus kegiatan rutin jalur alternatif 3a.....	121
Tabel 6.46 Kasus uji tambah imam salat jalur utama.....	122
Tabel 6.47 Kasus uji tambah imam salat jalur alternative 5a	122
Tabel 6.48 Kasus uji ubah imam salat jalur utama	123
Tabel 6.49 Kasus uji ubah imam salat jalur alternatif 4a	124
Tabel 6.50 Kasus uji hapus imam salat jalur utama.....	124
Tabel 6.51 Kasus uji hapus imam salat jalur alternatif 3a	125
Tabel 6.52 Kasus uji tambah jadwal salat jalur utama.....	125
Tabel 6.53 Kasus uji tambah jadwal salat jalur alternatif 5a	126
Tabel 6.54 Kasus uji ubah jadwal salat jalur utama	127
Tabel 6.55 Kasus uji ubah jadwal salat jalur alternatif 4a.....	127
Tabel 6.56 Kasus uji hapus jadwal salat jalur utama	128
Tabel 6.57 Kasus uji hapus jadwal salat jalur alternatif 3a	128
Tabel 6.58 Kasus uji tambah keuangan harian jalur utama.....	129
Tabel 6.59 Kasus uji tambah keuangan harian jalur alternatif 5a	130
Tabel 6.60 Kasus uji ubah keuangan harian jalur utama	130
Tabel 6.61 Kasus uji ubah keuangan harian jalur alternatif 4a.....	131
Tabel 6.62 Kasus uji hapus keuangan harian jalur utama.....	131
Tabel 6.63 Kasus uji hapus keuangan harian jalur alternatif 3a	132
Tabel 6.64 Kasus uji mencetak laporan keuangan harian jalur utama.....	132
Tabel 6.65 Kasus uji mencetak laporan keuangan harian jalur alternative 3a	133
Tabel 6.66 Kasus uji mencetak laporan keuangan bulanan jalur utama	133
Tabel 6.67 Kasus uji mencetak laporan keuangan bulanan jalur alternative 3a	134
Tabel 6.68 Kasus uji melihat detail dan daftar kegiatan insidental jalur utama.	134
Tabel 6.69 Kasus uji melihat detail dan daftar kegiatan insidental jalur 3a.....	135
Tabel 6.70 Kasus uji melihat detail dan daftar kegiatan rutin jalur utama.....	135
Tabel 6.71 Kasus uji melihat detail dan daftar kegiatan rutin jalur 3a	136
Tabel 6.72 Kasus uji melihat detail dan daftar pengumuman jalur utama	136



Tabel 6.73 Kasus uji Melihat detail dan daftar pengumuman jalur 3a..... 136

Tabel 6.74 Kasus uji melihat detail dan daftar artikel jalur utama..... 137

Tabel 6.75 Kasus uji melihat detail dan daftar artikel jalur 3a 137

Tabel 6.76 Kasus uji melihat daftar foto jalur utama..... 138

Tabel 6.77 Kasus uji melihat daftar foto jalur 2a 138

Tabel 6.78 Kasus uji melihat daftar imam jalur utama..... 138

Tabel 6.79 Kasus uji melihat daftar imam salat jalur 2a 139

Tabel 6.80 Kasus uji melihat daftar jadwal salat jalur utama 139

Tabel 6.81 Kasus uji melihat daftar jadwal salat jalur 2a..... 139

Tabel 6.82 Kasus uji melihat laporan keuangan harian jalur utama..... 140

Tabel 6.83 Kasus uji melihat laporan keuangan harian jalur 2a 140

Tabel 6.84 Kasus uji melihat laporan keuangan bulanan jalur utama..... 141

Tabel 6.85 Kasus uji melihat laporan keuangan bulanan jalur 2a 141



DAFTAR GAMBAR

Gambar 2.1 Proses Bisnis Pencatatan Keuangan Harian <i>As-Is</i>	9
Gambar 2.2 Proses Bisnis Pencatatan Keuangan Harian <i>To-Be</i>	10
Gambar 2.3 Proses Bisnis Membuat Laporan Keuangan Bulanan <i>As-Is</i>	10
Gambar 2.4 Proses Bisnis Membuat Laporan Keuangan Bulanan <i>To-Be</i>	10
Gambar 2.5 Alur <i>Waterfall</i>	12
Gambar 2.6 Contoh <i>Use Case Diagram</i>	17
Gambar 2.7 Contoh <i>Sequence Diagram</i>	19
Gambar 2.8 Contoh <i>Class Diagram</i>	21
Gambar 3.1 Diagram Alir Metode Penelitian	25
Gambar 4.1 <i>Use Case Diagram</i> Sistem Informasi Manajemen Masjid Al Ghifari	41
Gambar 5.1 <i>Sequence diagram</i> Tambah kegiatan insidental	64
Gambar 5.2 <i>Sequance diagram</i> Ubah kegiatan insidental	64
Gambar 5.3 <i>Sequance diagram</i> Tambah laporan keuangan harian	65
Gambar 5.4 <i>Class diagram</i> umum	66
Gambar 5.5 <i>Class diagram controller</i>	67
Gambar 5.6 <i>Class diagram entity</i>	67
Gambar 5.7 Perancangan CDM (<i>Conceptual Data Model</i>)	68
Gambar 5.8 Perancangan PDM (<i>Physical Data Model</i>)	69
Gambar 5.9 Perancangan Antarmuka Halaman Tambah Kegiatan Insidental	72
Gambar 5.10 Perancangan Antarmuka Halaman Ubah Kegiatan	75
Gambar 5.11 Perancangan Antarmuka Halaman Tambah Keuangan harian	77
Gambar 5.12 Implementasi Data	84
Gambar 5.13 Implementasi Antarmuka Halaman Tambah Kegiatan	85
Gambar 5.14 Implementasi Antarmuka Halaman Ubah Kegiatan Insidental	85
Gambar 5.15 Implementasi Antarmuka Halaman Tambah Keuangan Harian	86
Gambar 6.1 <i>Flow Graph method</i> tambah_kegiatan_aksi()	88
Gambar 6.2 <i>Flow graph method</i> ubah_kegiatan_aksi()	91
Gambar 6.3 <i>Flow graph method</i> tambah_harian_aksi()	94
Gambar 6.4 <i>Flow Graph method</i> tambah_kegiatan_aksi()	97
Gambar 6.5 <i>Flow graph method</i> ubah_kegiatan_aksi()	100
Gambar 6.6 <i>Flow graph method</i> tambah_harian_aksi()	103

Gambar 6.7 Hasil Pengujian Compatibility Menggunakan Aplikasi SortSite 142



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Masjid merupakan tempat yang digunakan oleh umat Muslim untuk beribadah. Masjid juga bisa berfungsi sebagai pusat pengajaran ilmu Agama Islam, pemberdayaan umat Muslim, dan pusat kegiatan sosial (Qadaruddin, Nurkidam, & Firman, 2016). Fungsi masjid harus dapat berjalan dengan baik dan optimal, sehingga keberadaan masjid dapat memberikan manfaat bagi masyarakat dengan proses manajemen yang baik.

Masjid Al Ghifari merupakan salah satu masjid yang ada di Kecamatan Lowokwaru, Kota Malang. Manajemen Masjid Al Ghifari terdiri dari manajemen kegiatan dan manajemen keuangan. Manajemen kegiatan di Masjid Al Ghifari terdiri dari pengelolaan informasi kegiatan rutin dan kegiatan insidental, sedangkan untuk manajemen keuangan yaitu pencatatan pemasukan dan pengeluaran keuangan masjid. Kegiatan rutin yang ada di Masjid Al Ghifari terdiri dari kajian rutin tiap hari, buka puasa setiap hari Senin dan Kamis, kajian tematik pernikahan yang diadakan setiap dua minggu sekali, dan olahraga setiap hari Sabtu dan Minggu. Masyarakat atau pengurus masjid yang ingin membuat kegiatan di Masjid Al Ghifari harus melewati beberapa tahapan administrasi. Pengusul kegiatan di Masjid Al Ghifari diwajibkan mengajukan surat permohonan pelaksanaan kegiatan yang ditujukan ke pengurus masjid yang mengatur bagian kegiatan. Pengurus masjid akan bermusyawarah mengenai tema, judul kegiatan, pemateri, dan tanggal pelaksanaan. Pengurus masjid akan memberikan rekomendasi tanggal kepada pengusul kegiatan setelah dilakukan musyawarah dan melihat buku pencatatan kegiatan Masjid Al Ghifari agar tidak terjadi bentrok satu kegiatan dengan kegiatan lainnya. Manajemen keuangan Masjid Al Ghifari terdiri dari pencatatan pemasukan dan pengeluaran keuangan masjid. Pencatatan pemasukan dan pengeluaran di Masjid Al Ghifari masih menggunakan buku. Pencatatan pemasukan dan pengeluaran keuangan setiap hari langsung dicatat oleh Bendahara pada buku catatan laporan keuangan harian. Laporan keuangan bulan dibuat menggunakan aplikasi *Microsoft Excel* berdasarkan laporan keuangan harian yang sudah dicatat di buku, setelah laporan keuangan bulanan dibuat oleh Bendahara laporan tersebut ditempel di papan pengumuman masjid.

Masjid AL Ghifari masih menggunakan buku untuk mencatat semua informasi. Pencatatan dan penyimpanan informasi menggunakan buku masih memiliki kekurangan yaitu dapat hilang atau bisa mengalami kerusakan karena pengarsipan informasi masih berupa fisik. Permasalahan pada manajemen kegiatan masjid yaitu pengurus masjid kesulitan untuk dapat menginformasikan seluruh kegiatan yang ada di Masjid Al Ghifari. Sosialisasi dan pemberitahuan kegiatan di Masjid Al Ghifari masih menggunakan poster yang ditempel di papan pengumuman serta menggunakan media sosial seperti *Instagram* dan *Facebook*. Penggunaan media sosial untuk menyampaikan informasi kegiatan yang ada di Masjid Al Ghifari masih memiliki keterbatasan karena media sosial tersebut tidak bisa menampilkan

semua informasi secara detail. Penyampaian kegiatan menggunakan *Instagram* dan *Facebook* dapat menjangkau masyarakat lebih luas untuk mendapatkan informasi, tetapi terbatas hanya masyarakat yang memiliki dan terhubung langsung ke media sosial tersebut yang mendapatkan informasi. Permasalahan manajemen keuangan di Masjid Al Ghifari yaitu Ketua pengurus masjid kesulitan untuk mengontrol keuangan karena tidak punya akses langsung untuk dapat melihat laporan keuangan serta pencatatan pemasukan dan pengeluaran keuangan masjid masih menggunakan buku. Bendahara masjid Al Ghifari masih mencatat laporan keuangan harian masjid menggunakan buku. Buku catatan laporan keuangan harian masjid Al Ghifari pernah hilang yang membuat Bendahara masjid kesulitan dan terhambat untuk membuat laporan keuangan bulanan. Buku laporan keuangan harian menjadi dasar untuk membuat laporan keuangan bulanan. Ketua pengurus dan pengurus masjid yang lain kesulitan untuk mengakses informasi di Masjid Al Ghifari secara langsung.

Berdasarkan permasalahan yang ada di atas, solusi yang dapat diberikan untuk melakukan manajemen Masjid Al Ghifari secara efektif dan efisien, maka dikembangkan sebuah Sistem Informasi Manajemen Masjid Berbasis Web. Teknologi web dipilih karena memiliki beberapa manfaat antara lain menyediakan informasi yang lebih detail, jangkauan lebih luas ke seluruh dunia, dan dapat mengakses informasi dimana dan kapan saja. Keunggulan menggunakan teknologi web yaitu dapat menampung segala informasi-informasi yang berbentuk teks, gambar dan video (Supriyono, Noviandri, & Purnomo, 2017). Web dapat mempermudah Anggota Bidang Ubudiyah mensosialisasikan dan menambah data kegiatan kepada masyarakat dimana dan kapan saja, memudahkan Ketua Pengurus Masjid Al Ghifari untuk dapat mengontrol keuangan secara langsung, serta membantu Bendahara untuk dapat membuat laporan keuangan.

Penelitian ini mengambil judul "Pengembangan Sistem Informasi Manajemen Masjid Al Ghifari Berbasis Web". Sistem ini diharapkan dapat membantu Ketua pengurus masjid untuk mengontrol keuangan secara langsung, mempermudah Anggota Bidang Ubudiyah untuk menambahkan Kegiatan masjid serta dapat membantu Bendahara masjid untuk dapat membuat laporan keuangan dimana dan kapan saja.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka didapatkan rumusan masalah sebagai berikut:

1. Apa saja kebutuhan yang harus ada pada sistem informasi manajemen Masjid Al Ghifari berbasis web?
2. Bagaimana hasil perancangan dari sistem informasi manajemen Masjid Al Ghifari berbasis web?
3. Bagaimana hasil implementasi dari pengembangan sistem Informasi Manajemen Masjid Al Ghifari berbasis web?

4. Bagaimana hasil pengujian dari sistem informasi manajemen Masjid Al Ghifari berbasis web?

1.3 Tujuan

Tujuan dari penelitian ini antara lain:

1. Mengetahui hasil kebutuhan dari sistem informasi manajemen Masjid Al Ghifari berbasis web.
2. Merancang sistem informasi manajemen Masjid Al Ghifari berbasis web yang berdasarkan dengan kebutuhan yang sudah ditentukan sebelumnya.
3. Mengimplementasikan hasil dari perancangan sistem informasi manajemen Masjid Al Ghifari berbasis web.
4. Menguji sistem informasi manajemen Masjid Al Ghifari berbasis web.

1.4 Manfaat

Manfaat dari penelitian ini antara lain:

1. Memudahkan pengurus masjid untuk menyampaikan informasi yang ada di Masjid Al Ghifari.
2. Memudahkan pengurus masjid untuk mengontrol pencatatan keuangan dan kegiatan di Masjid Al Ghifari.
3. Mempermudah masyarakat untuk mengetahui tentang informasi dan kegiatan yang akan dilaksanakan oleh Masjid Al Ghifari

1.5 Batasan Masalah

Agar hasil diperoleh sesuai dengan apa yang diharapkan, maka dibuat batasan untuk membatasi pengembangan *website* Masjid Al Ghifari antara lain:

1. Pengembangan sistem ini hanya difokuskan untuk Masjid Al Ghifari Malang.
2. Pembuatan sistem berbasis web ini menggunakan metode pengembangan *waterfall*.
3. Sistem informasi manajemen masjid ini merupakan sistem yang berbasis web dan menggunakan *Framework Codeigniter* dalam proses pengembangan sistem

1.6 Sistematika Pembahasan

Sistematika penyusunan laporan ini adalah ditujukan untuk memberikan gambaran dari laporan secara garis besar yang meliputi beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi tentang kajian pustakan dari penelitian-penelitian sebelumnya dan landasan teori yang dipakai sebagai landasan dasar yang digunakan sebagai referensi untuk membuat sistem.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang metode dan langkah-langkah yang digunakan dalam penelitian ini untuk membangun sebuah sistem.

BAB IV REKAYASA KEBUTUHAN

Bab ini berisi tentang analisis kebutuhan sistem yang direpresentasikan pada gambaran umum sistem, identifikasi aktor, analisis kebutuhan fungsional dan nonfungsional, *use case diagram*, dan *use case scenario*

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi tentang perancangan sistem dan implementasi sistem yang sesuai dengan analisis kebutuhan sistem yang sudah ditentukan. Perancangan sistem terdiri dari perancangan arsitektur, perancangan komponen, perancangan data, dan perancangan antarmuka. Implementasi terdiri dari implementasi kode program, implementasi data, implementasi antarmuka

BAB VI PENGUJIAN

Bab ini berisi tentang hasil pengujian terhadap sistem yang telah dibuat berdasarkan kebutuhan yang sudah ditentukan. Pengujian yang dilakukan terdiri dari pengujian unit, pengujian integrasi, dan pengujian validasi.

BAB VII KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang didapatkan dari hasil penelitian yang sudah dilakukan. Serta memberika saran yang dapat digunakan untuk pengembangan selanjutnya dari penelitian "Pengembangan Sistem Informasi Manajemen Masjid Al Ghifari Berbasis Web".

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian sebelumnya yang serupa mengenai sistem manajemen masjid telah dilakukan oleh Reyanda pada tahun 2017 yang berjudul "*Perancangan Sistem Informasi Manajemen Masjid Baiturrahim Berbasis Website*" (Reyanda, 2017). Masjid Baiturrahim terletak di Desa Makamhaji, Kecamatan Kartasura, Kabupaten Sukoharjo, Jawa Tengah. Sistem ini dibuat untuk membantu pihak pengurus Masjid Baiturrahim untuk memberikan informasi mengenai kegiatan dan keuangan masjid kepada masyarakat secara tepat, akurat, dan bisa diakses dengan mudah. Sistem ini memiliki fitur informasi kegiatan, keuangan, dan *maps* untuk mempermudah masyarakat untuk mendapatkan informasi tentang Masjid Baiturrahim. Sistem ini diimplementasikan berbasis web dan menggunakan metode pengembangan perangkat lunak *waterfall* serta menggunakan pendekatan berorientasi objek (*Object Oriented*).

Penelitian yang serupa mengenai sistem manajemen masjid dilakukan oleh Eko Ariwibowo pada tahun 2013 yang berjudul "*Perancangan dan Implementasi Sistem Informasi Manajemen Kegiatan Masjid*" Studi kasus: Masjid Jogokariyan Yogyakarta (Eko, 2013). Sistem manajemen masjid yang dibuat untuk memudahkan Sekretaris dan Bendahara masjid untuk melakukan mengelola data jemaah, donatur, dan keuangan Masjid Jogokariyan. Sistem ini diimplementasikan menggunakan *Microsoft Visual Basic 6.0* dan menggunakan metode pengembangan perangkat lunak *waterfall* serta menggunakan pendekatan terstruktur.

Penelitian yang sama mengenai sistem manajemen masjid yang dilakukan oleh Dodyk Kristanto pada tahun 2018 yang berjudul "*Sistem Informasi Pengelolaan Masjid Ibaadurrahman Berbasis Web*" (Dodyk, 2018). Masjid Ibaadurrahman terletak di Desa Pabelan, Kecamatan Kartasura, Kabupaten Sukoharjo, Jawa Tengah. Sistem ini dibuat untuk membantu pengurus masjid secara efektif dan efisien untuk memberikan informasi kegiatan serta mengurus peminjaman masjid untuk kajian. Sistem ini memiliki fitur jadwal sholat, hitung zakat, peminjaman masjid. Sistem ini diimplementasikan berbasis web dan menggunakan metode pengembangan perangkat lunak *waterfall* serta menggunakan pendekatan berorientasi objek (*object oriented*).

Perbedaan penelitian ini dengan penelitian yang dilakukan oleh Eko Ariwibowo (2013) terletak pada implementasi, pemodelan, dan pendekatan. Penelitian yang dilakukan oleh Eko diimplementasikan menggunakan *Microsoft Visual Basic*, sedangkan pada penelitian ini diimplementasikan berbasis web menggunakan *Framework Codeigniter* dengan menggunakan bahasa pemrograman PHP. Penelitian yang dilakukan oleh Eko (2013) menggunakan pemodelan terstruktur, sedangkan pada penelitian ini menggunakan pemodelan

berorientasi objek (*Object Oriented*). Penelitian yang dilakukan oleh Eko (2013) menggunakan pemodelan DFD, sedangkan pada penelitian ini menggunakan pemodelan UML. Sistem yang akan dibuat untuk memudahkan ketua pengurus dan pengurus masjid yang lain untuk mengontrol keuangan dan kegiatan yang ada di Masjid Al Ghifari. Sistem ini juga memudahkan masyarakat untuk mendapatkan informasi-informasi yang ada di Masjid Al Ghifari. Sistem ini memiliki beberapa fitur yaitu keuangan, kegiatan rutin, kegiatan *insidental*, olahraga, fasilitas, pengumuman, dan artikel. Sistem ini menggunakan pendekatan berorientasi objek dan metode pengembangan perangkat lunak *waterfall*.

2.2 Manajemen Masjid Al Ghifari

Menurut Hasibuan Malayu (2014), manajemen masjid adalah usaha-usaha untuk mengembangkan dan mengatur fungsi-fungsi masjid sebagaimana fungsinya. Manajemen masjid merupakan suatu proses atau usaha untuk mencapai kemakmuran masjid yang ideal yang dilakukan oleh ketua pengurus masjid bersama staf dan jamaahnya melalui berbagai kegiatan yang positif. Tujuan dari *idarrah* atau manajemen masjid supaya masjid mampu mengembangkan kegiatan yang ada dan lebih berguna ke masyarakat. Manajemen di Masjid AL Ghifari terdiri dari manajemen kegiatan dan manajemen keuangan. Manajemen kegiatan yang ada terdiri kegiatan *insidental*, kegiatan rutin, dan olahraga. Manajemen keuangan terdiri dari pencatatan pemasukan dan pengeluaran keuangan masjid.

Masyarakat atau pengurus masjid yang ingin membuat kegiatan di Masjid Al Ghifari diwajibkan untuk membuat surat permohonan pelaksanaan kegiatan yang ditujukan untuk ketua pengurus masjid. Surat permohonan tersebut terdiri tema kegiatan, judul kegiatan, pemateri, dan tanggal pelaksanaan, sesudah itu pengurus masjid akan bermusyawarah untuk mengizinkan atau tidak kegiatan tersebut dilaksanakan. Pengurus masjid akan langsung menolak kegiatan tersebut jika tanggal acara bentrok dengan acara yang sudah tercatat. Pengurus masjid mencatat semua kegiatan yang akan dilakukan menggunakan buku. Kegiatan *insidental* yang sering dibuat dominan kegiatan ceramah dengan skala jamaah yang lebih banyak yang dibuat oleh Masjid Al Ghifari yang biasa dilakukan 1 atau 2 bulan sekali dengan mengundang ustaz untuk memberikan materi. Kegiatan rutin yang ada di Masjid Al Ghifari terdiri dari kajian singkat setelah sholat Subuh dan sholat Magrib, buka puasa setiap hari Senin dan Kamis, kajian muslimah ibu-ibu diadakan setiap hari Sabtu setelah sholat Asar, dan pesantren mahasiswa. Pesantren mahasiswa diwajibkan para pesertanya untuk bermalam di masjid dan program yang ada di pesantren mahasiswa ini terdiri dari menghafal Al-Qur'an, bahasa Arab, *sira nabawiyah*, dan menghafal hadis. Kegiatan olahraga yang ada di Masjid Al Ghifari terdiri dari olahraga bela diri dojo, memanah, berkuda, futsal, dan bulutangkis. Jadwal latihan olahraga dojo dilakukan setiap hari Satu dan Minggu serta untuk atlet setelah sholat Isya sekitar jam 7-9 malam dan untuk kelas umum hanya hari Sabtu dan Minggu sekitar jam 6-7 pagi. Jadwal latihan memanah dilakukan setiap hari Minggu pagi jam 6-9 pagi dan jadwal latihan berkuda hanya

dilakukan sebulan sekali dan tergantung peminatnya. Jadwal latihan futsal dan bulutangkis dilaksanakan setiap hari Minggu selesai sholat Subuh atau sholat Isya. Masjid Al Ghifari menggunakan aplikasi *Instagram* dan *Facebook* untuk membantu pengurus masjid untuk menginformasikan tentang kegiatan yang akan dilakukan. Fasilitas yang ada di Masjid Al Ghifari terdiri dari penyediaan mobil *ambulance*, *mualaf center*, dan *funeral service* atau penyediaan perlengkapan mayat. Mobil *ambulance*, *mualaf center*, dan *funeral service* bisa digunakan oleh masyarakat secara gratis dan untuk menggunakannya hanya dengan cara menghubungi pengurus masjid untuk di data lebih lanjut. *Mualaf center* digunakan untuk membimbing masyarakat yang baru masuk agama Islam untuk lebih memahami tentang ajaran agama Islam seperti cara sholat, membaca Al-Qur'an, berpuasa, wudu, dan lain-lain yang bersangkutan mengenai agama Islam.

Manajemen Keuangan adalah proses pengambilan keputusan yang berkaitan seberapa banyak dan apa saja aset yang dimiliki, serta meningkatkan modal untuk memaksimalkan nilai perusahaan atau lembaga nirlaba (Brigham & Houston, 2017). Menurut Nainggola (2012), manajemen keuangan terdiri dari pencatatan pemasukan dan pengeluaran serta evaluasi keuangan. Pencatatan keuangan di Masjid Al Ghifari masih menggunakan buku untuk mencatat pemasukan dan pengeluaran keuangan masjid. Pencatatan pemasukan dan pengeluaran keuangan setiap hari langsung dicatat oleh Bendahara pada buku catatan laporan keuangan harian. Laporan keuangan bulan dibuat menggunakan aplikasi *Microsoft Excel* berdasarkan laporan keuangan harian yang sudah dicatat di buku, setelah laporan keuangan bulanan dibuat oleh Bendahara laporan tersebut ditempel di papan pengumuman masjid. Proses untuk membuat laporan keuangan di Masjid Al Ghifari, pertama dalam membuat laporan keuangan mencatat semua pengeluaran dan pemasukan keuangan pada buku catatan keuangan harian. Proses kedua membuat laporan keuangan bulanan berdasarkan buku catatan keuangan harian, setelah itu laporan keuangan bulanan dibuat dengan menggunakan *Microsoft Excel* untuk dapat merincikan pemasukan dan pengeluaran keuangan Masjid Al Ghifari selama jangka waktu bulanan. Laporan keuangan Masjid Al Ghifari terbagi menjadi dua antara lain:

1. Laporan keuangan harian

Laporan keuangan harian adalah laporan keuangan yang berisi semua catatan transaksi pemasukan dan pengeluaran keuangan Masjid Al Ghifari selama jangka waktu harian. Cara membuat laporan keuangan harian dengan cara mencatat semua transaksi keuangan pada hari itu ke dalam buku keuangan harian dan sebelum dicatat semua transaksi harus disertai dengan bukti pembayaran transaksi pemasukan atau pengeluaran keuangan. Semua transaksi yang sudah dicatat pada buku keuangan harian nanti akan dipindahkan dan dikelompokkan ke dalam *Microsoft Excel* untuk dibuat menjadi laporan keuangan bulanan. Laporan keuangan harian Masjid Al Ghifari dapat dilihat pada (Lampiran C). Gambar 2.1 menunjukkan proses bisnis pencatatan laporan keuangan harian *As-Is* dan Gambar 2.2 menunjukkan

proses bisnis pencatatan laporan keuangan harian *To-Be*. Laporan keuangan terdiri dari empat kolom antara lain:

- a. Kolom pertama berisi keterangan transaksi
- b. Kolom kedua berisi tanggal transaksi
- c. Kolom ketiga berisi jumlah pemasukan atau debit
- d. Kolom keempat berisi jumlah pengeluaran atau kredit

2. Laporan keuangan bulanan

Laporan keuangan bulanan adalah laporan keuangan yang berisi semua catatan transaksi pemasukan dan pengeluaran keuangan Masjid Al Ghifari dalam jangka waktu bulanan. Laporan keuangan bulanan bisa dijadikan bahan evaluasi keuangan untuk pengurus masjid. Laporan keuangan bulanan dibuat berdasarkan laporan keuangan keuangan harian yang sudah dibuat. Cara membuat laporan keuangan bulanan dengan cara memindahkan semua transaksi pada buku catatan keuangan harian ke dalam *Microsoft Excel* dan mengelompokkannya sesuai dengan kelompok transaksi keuangannya. Laporan keuangan bulanan Masjid Al Ghifari dapat dilihat pada (Lampiran B). Gambar 2.3 menunjukkan proses bisnis membuat laporan keuangann bulanan *As-Is* dan Gambar 2.4 menunjukkan proses bisnis membuat laporan keuangan bulanan *To-Be* Laporan keuangan bulanan Masjid Al Ghifari terbagi menjadi beberapa bagian kelompok transaksi antara lain:

a. Pemasukan rutin

Pemasukan rutin adalah kumpulan pendapatan yang pasti didapat oleh Masjid Al Ghifari dalam jangka waktu bulanan seperti infak jumat, infak parkir, kontribusi Al Ghifari Mart, dan infak TPQ Jundullah.

b. Pemasukan nonrutin

Pemasukan nonrutin adalah kumpulan pendapatan yang belum pasti didapatkan oleh Masjid Al Ghifari dalam jangka waktu bulanan seperti infak untuk renovasi masjid, infak untuk renovasi pagar masjid, dan infak perbaikan masjid.

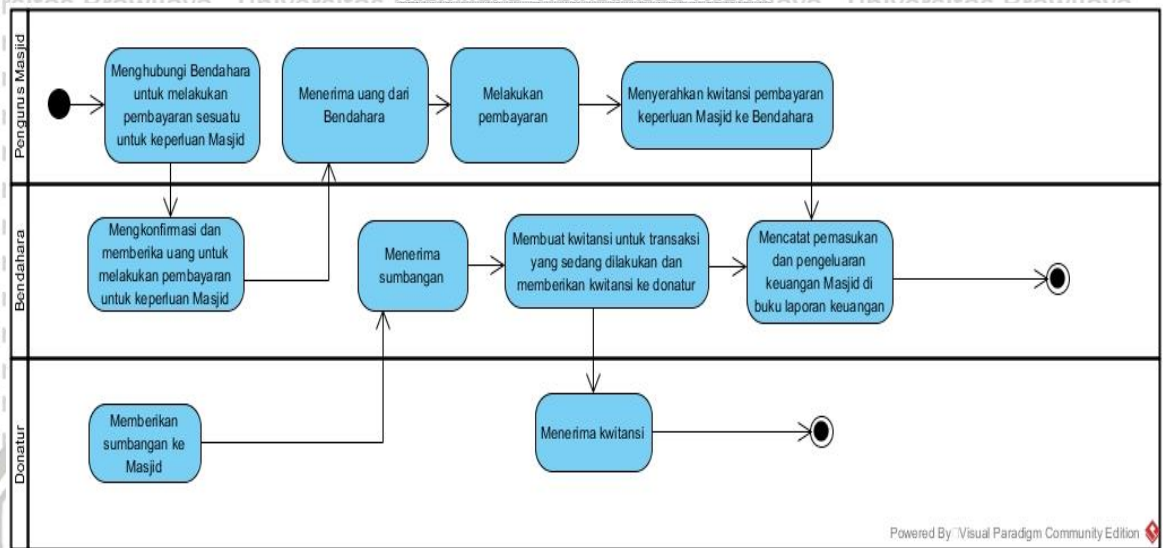
c. Pengeluaran rutin

Pengeluaran rutin adalah kumpulan pengeluaran yang sudah pasti dikeluarkan oleh Masjid Al Ghifari dalam jangka waktu bulanan seperti biaya guru TPQ, gaji imam, biaya khatib, biaya listrik, biaya internet, biaya operasional *ambulance*, biaya petugas LDPS.

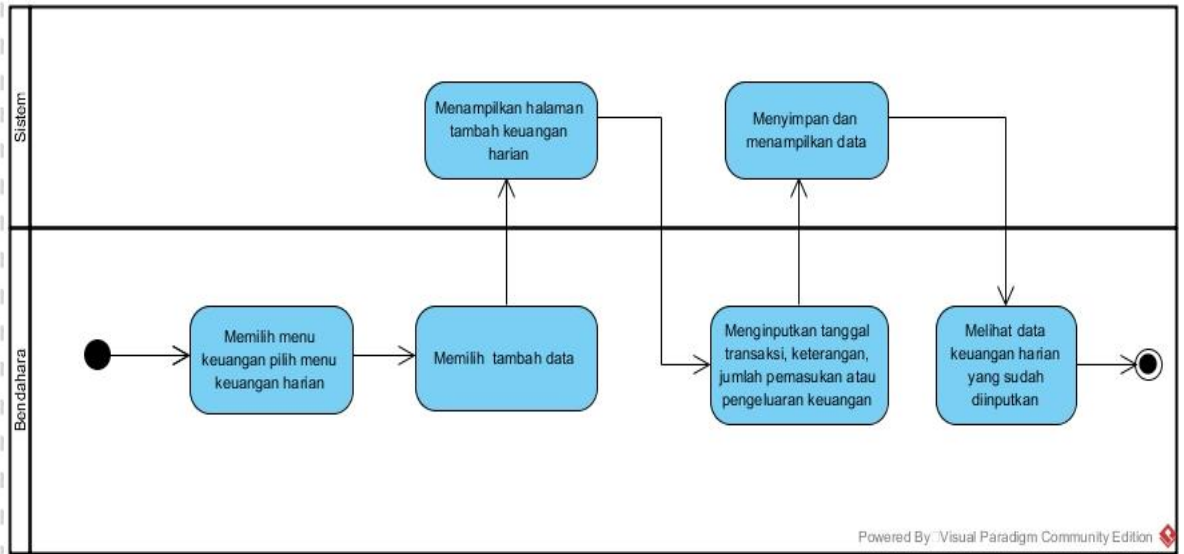
d. Pengeluaran nonrutin

Pengeluaran nonrutin adalah kumpulan pengeluaran yang belum pasti dikeluarkan oleh Masjid Al Ghifari dalam jangka waktu bulanan seperti biaya perbaikan kanopi masjid dan biaya mengganti keran air masjid.

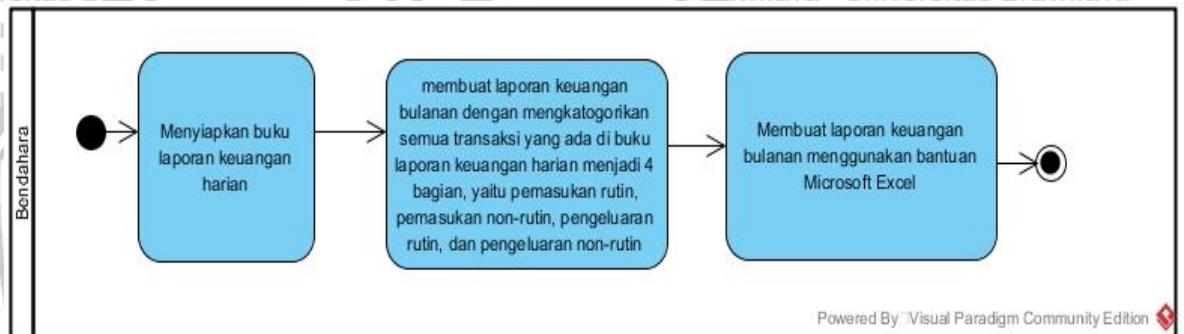
Perbedaan laporan keuangan harian dan bulanan, yaitu laporan keuangan harian digunakan untuk mencatat semua transaksi keuangan harian sedangkan laporan keuangan bulanan berisi rangkuman transaksi keuangan selama jangka waktu bulanan. Hubungan antara laporan keuangan harian dan bulanan, yaitu laporan keuangan harian yang sudah dibuat menjadi dasar pembuatan laporan keuangan bulanan. Laporan keuangan harian dan bulanan hanya bisa dibuat oleh Bendahara Masjid Al Ghifari.



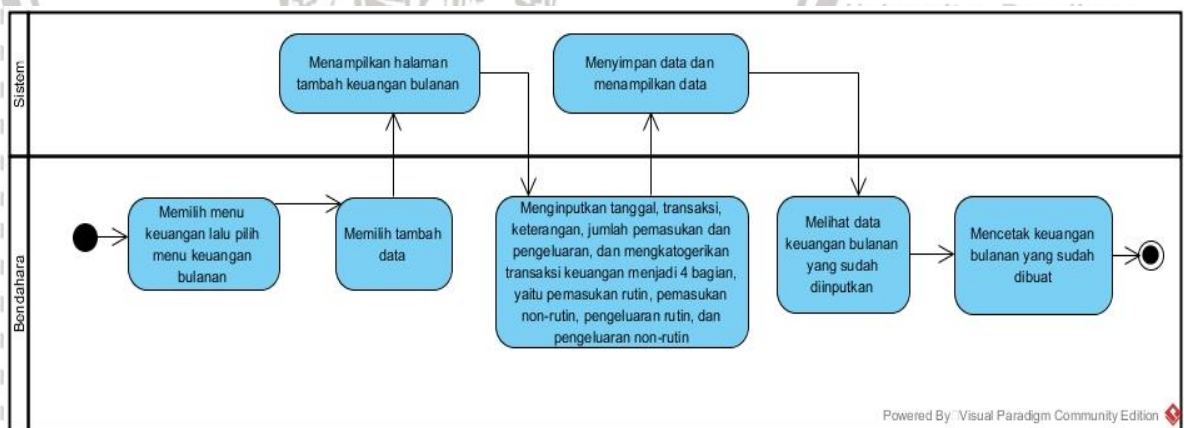
Gambar 2.1 Proses Bisnis Pencatatan Keuangan Harian AS-1s



Gambar 2.2 Proses Bisnis Pencatatan Keuangan Harian To-Be



Gambar 2.3 Proses Bisnis Membuat Laporan Keuangan Bulanan As-Is



Gambar 2.4 Proses Bisnis Membuat Laporan Keuangan Bulanan To-Be

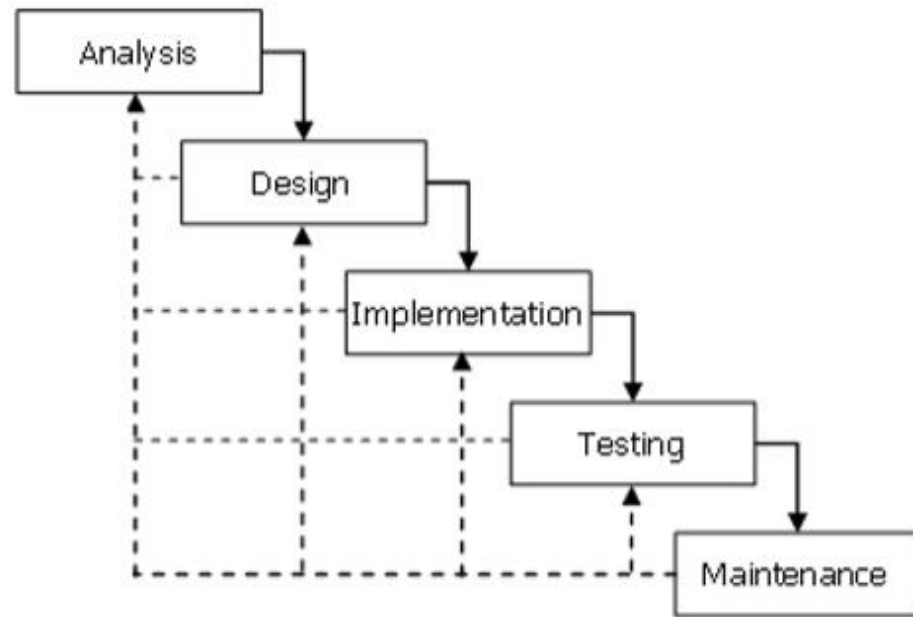
2.3 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak atau *Software Engineering* adalah penggunaan suatu prinsip (*engineering*) untuk membuat suatu perangkat lunak yang lebih ekonomis dan dapat bekerja secara efisien pada pengembangan suatu perangkat lunak (Pressman, 2010). Rekayasa perangkat lunak juga dibutuhkan agar sistem

perangkat lunak bisa digunakan dengan maksimal. Rekayasa perangkat lunak memiliki beberapa model yang digunakan dalam pengembangan perangkat lunak. Model pengembangan perangkat lunak digunakan untuk menjelaskan aliran segala aktivitas dan susunan pekerjaan yang harus dilakukan untuk membuat suatu perangkat lunak (Pressman, 2010). Model-model pengembangan perangkat lunak terdiri dari model *waterfall*, model *prototyping*, model *spiral*, model RAD (*rapid application development*), model *iterative*, dan model *extreme programming*. Penelitian ini menggunakan model pengembangan perangkat lunak *waterfall* karena kebutuhan sistem sudah diketahui dengan baik sebelum pengembangan perangkat lunak dimulai (Sommerville, 2011). Kebutuhan sistem dapat berubah selama proses pengembangan perangkat lunak berlangsung, namun kemungkinan perubahannya sangat kecil. Rekayasa perangkat lunak memiliki beberapa pendekatan yang digunakan dalam membuat sebuah perangkat lunak. Pendekatan perangkat lunak terdiri dari pendekatan berorientasi objek dan pendekatan terstruktur. Penelitian ini menggunakan pendekatan berorientasi objek karena menggunakan objek untuk mengirim data, memproses data, menerima pesan dari objek lain, dan kemudahan dalam melakukan perubahan.

2.3.1 Model *Waterfall*

Waterfall adalah salah satu model pengembangan perangkat lunak yang ada di dalam model SDLC (*Software Development Life Cycle*). Menurut Bassil (2012), *waterfall* memiliki karakteristik bertahap dalam proses alur kerja. *Waterfall* memiliki sifat rekursif, yaitu setiap tahap dapat diulang dan kembali ke tahap sebelumnya untuk mendapatkan hasil yang sempurna (Bassil, 2012). Model *waterfall* memiliki lima tahapan yang terdiri dari *analysis*, *design*, *implementation*, *testing*, dan *maintenance* (Bassil, 2012).



Gambar 2.5 Alur *Waterfall*

Sumber: Bassil (2012)

Gambar 2.5 menggambarkan tahapan-tahapan yang ada di dalam model *waterfall*, penjelasan setiap tahapan sebagai berikut:

1. Tahapan *Analysis*

Tahapan ini digunakan untuk mengidentifikasi kebutuhan yang ada di dalam sistem. Kebutuhan pada sebuah perangkat lunak diklasifikasikan menjadi dua, yaitu kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional merupakan kebutuhan yang harus disediakan di dalam sistem dan menggambarkan apa yang bisa dilakukan oleh sebuah sistem (Sommerville, 2011). Kebutuhan nonfungsional merupakan batasan layanan atau fungsi yang ditawarkan oleh sistem seperti batasan waktu, proses pengembangan, dan standar yang sudah ditentukan (Sommerville, 2011). Proses pada tahap rekayasa kebutuhan terbagi menjadi beberapa tahapan yang terdiri dari elisitasi kebutuhan, spesifikasi kebutuhan, dan validasi (Pressman, 2010).

- a. Elisitasi kebutuhan merupakan sebuah proses untuk menggali dan mengumpulkan kebutuhan dengan cara melalui komunikasi dengan pihak yang berkepentingan (Pressman, 2010). Elisitasi kebutuhan mempunyai beberapa macam teknik untuk mengumpulkan kebutuhan yang terdiri dari wawancara, kuisisioner, observasi, analisis, *prototyping*, dan lain-lain.
- b. Spesifikasi kebutuhan digunakan untuk menulis informasi yang sudah didapatkan dalam proses elisitasi kebutuhan menjadi lebih jelas, tidak ambigu, mudah dipahami, lengkap, dan konsisten (Sommerville, 2011).

c. Validasi adalah proses untuk memastikan kembali kebutuhan yang didapatkan dari proses spesifikasi kebutuhan yang sudah didefinisikan sesuai dengan keinginan pengguna (Sommerville, 2011).

2. Tahapan *Design*

Tahapan ini digunakan untuk merancang perangkat lunak berdasarkan kebutuhan yang sudah didapatkan. Tahapan perancangan yang dilakukan terdiri dari perancangan arsitektur, perancangan data, perancangan komponen, dan perancangan antarmuka (Pressman, 2010):

- Perancangan arsitektur digunakan untuk mengidentifikasi komponen-komponen perangkat lunak dan relasi antar komponen (Sommerville, 2011). Menurut Pressman (2010), perancangan arsitektur merupakan perancangan dasar untuk membuat perangkat lunak dan menggambarkan hubungan setiap komponen.

- Perancangan data digunakan untuk menggambarkan susunan struktur data berdasarkan informasi dari hasil analisis kebutuhan yang dipakai untuk implementasi perangkat lunak (Pressman, 2010). Struktur data dapat disimpan dalam *database* atau *file*. Struktur data yang disimpan dalam *database* dapat dimodelkan menggunakan CDM (*Conceptual Data Model*) dan PDM (*Physical Data Model*). CDM adalah model data yang berisi entitas, atribut, dan hubungan antar entitas lain (Birgi, Khaire & Hira, 2016). PDM adalah model data yang berisi struktur tabel, termasuk nama kolom, tipe data, *primary key*, *foreign key*, dan hubungan antar tabel (Birgi, Khaire & Hira, 2016). Pemodelan CDM akan ditransformasikan untuk dapat memodelkan entitas yang ada pada sistem, setelah itu pemodelan CDM digunakan untuk menjadi dasar dalam pemodelan PDM.

- Perancangan komponen digunakan mendefinisikan algoritme suatu komponen perangkat lunak (Pressman, 2010). Menurut Pressman dan Maxim (2015), perancangan komponen merupakan suatu proses untuk mentransformasi komponen yang didapatkan dalam perancangan arsitektur menjadi deskripsi komponen yang lebih detail.

- Perancangan antarmuka digunakan untuk merancang antarmuka perangkat lunak untuk pengguna (Pressman, 2010).

3. Tahapan *Implementation*

Tahapan ini digunakan untuk menerapkan hasil perancangan yang diubah menjadi bentuk bahasa pemrograman atau kode yang dapat dibaca oleh mesin. Implementasi dapat dilakukan dengan menggunakan bahasa pemrograman yang terdiri dari PHP, *Java*, C, dan lain-lain.

4. Tahapan *Testing*

Tahapan ini dikenal dengan verifikasi dan validasi yang digunakan untuk memeriksa spesifikasi kebutuhan sesuai dengan yang sudah didefinisikan (Bassil, 2012). Tahapan pengujian terdiri dari verifikasi dan validasi. Tahapan

verifikasi adalah suatu proses untuk memastikan perangkat lunak mengimplementasikan fungsi dengan benar (Bassil, 2012). Tahapan validasi adalah suatu proses memastikan perangkat lunak yang dibangun sesuai dengan kebutuhan yang sudah ditentukan (Bassil, 2012). Pengujian perangkat lunak memiliki teknik pengujian yang dilakukan untuk menjalankan strategi pengujian. Teknik pengujian terdiri dari teknik *white-box testing* dan *black-box testing* (Pressman & Maxim, 2015). Setiap teknik pengujian memiliki metode yang berisi langkah-langkah untuk mendefinisikan kasus uji.

White-box testing adalah pengujian yang menggunakan algoritme dari hasil perancangan komponen sebagai bahan untuk kasus uji (Pressman & Maxim, 2015). *White-box testing* memiliki beberapa metode yang dapat digunakan untuk menguji kasus uji. Metode teknik *white-box testing* yang digunakan dalam penelitian ini adalah metode *basis path testing*. Metode *basis path testing* adalah merancang kasus uji berdasarkan ukuran kompleksitas (*cyclometric complexity*) yang berasal dari algoritme hasil perancangan komponen (Pressman & Maxim, 2015). Ukuran sebuah kompleksitas dapat digunakan sebagai acuan untuk mendefinisikan semua jalur yang ada. Kasus uji yang sudah didefinisikan digunakan untuk menguji kode program dan setiap jalur dasar yang sudah didefinisikan akan dilewati minimal satu kali.

Langkah-langkah untuk melakukan *basis path testing* adalah:

- Membuat *flow graph* berdasarkan hasil perancangan komponen.
- Menentukan *cyclomatic complexity* $V(G)$:
 - o $V(G)$ = jumlah *region*,
 - o $V(G)$ = jumlah *edge* – jumlah *node* + 2,
 - o $V(G)$ = *predicate node* + 1
- Mendefinisikan kasus-kasus uji untuk setiap jalur dasar yang sudah ditentukan.

Black-box testing adalah pengujian yang dilakukan sesuai dengan spesifikasi kebutuhan. *Black-box testing* menguji input untuk mengetahui keluaran dari input sudah sesuai dengan yang diinginkan (Marsic, 2012). Pengujian *black-box testing* tidak perlu tahu bagaimana isi dari kode program perangkat lunak yang akan diuji (Somerville, 2011). Metode *black-box testing* yang digunakan dalam penelitian ini adalah *scenario based testing*. *Scenario based testing* merupakan salah metode pengujian yang berfokus pada apa yang dilakukan pengguna pada perangkat lunak (Pressman, 2010). Kasus uji yang digunakan dalam metode *scenario based testing* yaitu *use case scenario*.

Strategi pengujian adalah langkah-langkah yang akan direncanakan dan dilakukan sebagai bagian dari pengujian dengan menerapkan teknik dan metode pengujian yang ada (Pressman & Maxim, 2015). Strategi pengujian dapat dilakukan dengan cara pengujian unit, pengujian integrasi, dan pengujian validasi. Penjelasan masing-masing strategi pengujian sebagai berikut:

- Pengujian Unit

Pengujian unit adalah pengujian dilakukan dengan menggunakan kode program sebagai unit yang dapat berdiri sendiri (Pressman & Maxim). Menurut Pressman (2010), pengujian berfokus pada bagian terkecil dari suatu perangkat lunak, yaitu komponen. Pengujian unit menguji komponen secara mandiri. Unit dalam pengujian ini adalah komponen yang berasal dari perangkat lunak. Menurut Pressman (2010), unit dalam pendekatan terstruktur adalah module dan pada pendekatan berorientasi objek adalah klas. Pengujian unit pada pendekatan berorientasi objek dilakukan dengan cara menguji operasi (*method*) yang ada di dalam klas atau module secara mandiri. Pengujian unit dapat dilakukan dengan menggunakan teknik *white-box* dan *black-box testing*.

- Pengujian Integrasi

Pengujian integrasi adalah pengujian yang dilakukan pada operasi (*method*) yang ada di dalam klas atau module yang saling berhubungan (Pressman, 2010). Pengujian integrasi dapat dilakukan dengan menggunakan teknik *white-box testing* dan *black-box testing*.

- Pengujian Validasi

Pengujian validasi berfokus pada apa yang dilakukan pengguna dan keluaran yang dikenali oleh pengguna (Pressman & Maxim, 2015). Pengujian validasi berfokus pada spesifikasi kebutuhan perangkat lunak (Pressman, 2010). Pengujian validasi dilakukan untuk menguji semua kebutuhan perangkat lunak yang terdiri dari kebutuhan fungsional dan kebutuhan nonfungsional. Pengujian validasi dilakukan berdasarkan kebutuhan yang sudah didefinisikan dan pengujian validasi dapat dilakukan dengan menggunakan teknik *black-box testing* dengan metode *basis path testing*. Pengujian kebutuhan nonfungsional pada penelitian ini menggunakan pengujian *compatibility*. Pengujian *compatibility* dilakukan untuk mengetahui kemampuan dari sistem dapat dijalankan pada berbagai *browser web* (Pressman, 2010).

5. Tahapan *Maintenance*

Tahapan ini adalah proses memodifikasi perangkat lunak, memperbaiki output, memperbaiki kesalahan, dan meningkat kualitas dari perangkat lunak (Bassil, 2012). Pada penelitian ini tahap *maintenance* tidak digunakan.

2.3.2 Pendekatan Berorientasi Objek

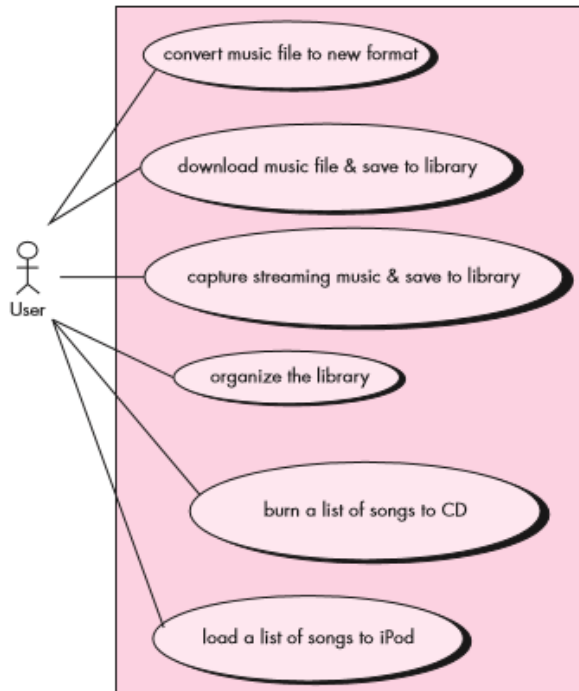
Pendekatan berorientasi objek merupakan suatu pendekatan dalam pengembangan perangkat lunak yang menjelaskan informasi sebagai sekumpulan objek yang memiliki atribut dan perilaku tertentu (Pressman, 2010). Pendekatan berorientasi objek dapat diterapkan pada tahapan analisis, perancangan, implemetasi dan pengujian (Pressman, 2010). Pengembangan perangkat lunak dengan menggunakan pendekatan OO (*Object Oriented*) dimulai dari tahapan

analisis atau yang disebut *Object Oriented Analysis* (OOA). OOA merupakan tahapan analisis domain masalah dengan pendekatan berorientasi objek (Pressman, 2010). Tahapan perancangan atau yang disebut dengan *Object Oriented Design* (OOD), merupakan tahapan merancang perangkat lunak berdasarkan hasil analisis dari hasil dari proses OOA (Pressman, 2010). OOD dapat dilakukan dengan perancangan arsitektur, perancangan data, perancangan komponen, dan perancangan antarmuka (Pressman, 2010). Tahapan implementasi atau yang disebut dengan *Object Oriented Programming* (OOP), merupakan mengubah hasil dari perancangan yang sudah didapatkan dari proses OOD menjadi bahasa pemrograman. Tahapan pengujian atau yang disebut *Object Oriented Testing* (OOT), merupakan suatu proses menemukan kesalahan dalam perangkat lunak (Pressman, 2010).

Pemodelan pada tahapan OOA dan OOD dapat dimodelkan dengan menggunakan *Unified Modelling Language* (UML). UML adalah sebuah standar yang digunakan untuk memvisualisasikan, mengidentifikasi, dan membuat dokumentasi dari sebuah sistem (Pressman & Maxim, 2015). Diagram-diagram UML yang digunakan dalam penelitian ini terdiri dari *Use Case Diagram*, *Sequence Diagram*, dan *Class Diagram*. Berikut penjelasan dari UML yang dipakai dalam penelitian ini:

2.3.2.1 Use Case Diagram

Use Case Diagram merupakan diagram yang menggambarkan aktivitas yang dapat dilakukan aktor pada sistem (Sommerville, 2011). *Use case diagram* mendeskripsikan sebuah interaksi satu atau lebih pengguna dengan sistem informasi yang akan dibuat (Sommerville, 2011). Contoh *use case diagram* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Contoh Use Case Diagram

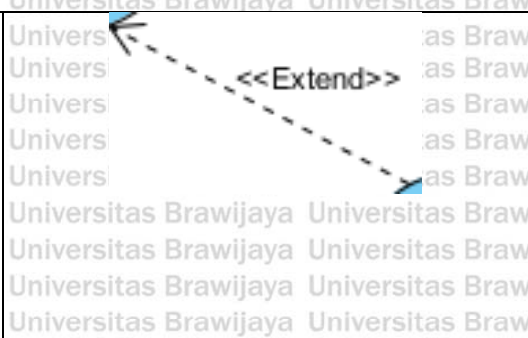
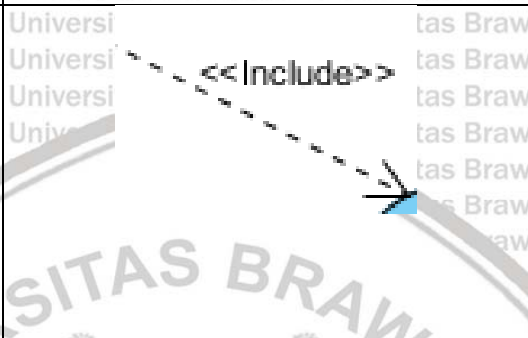
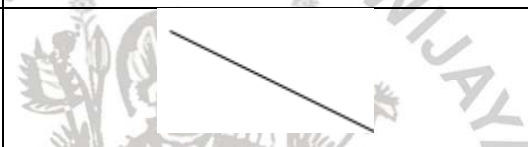
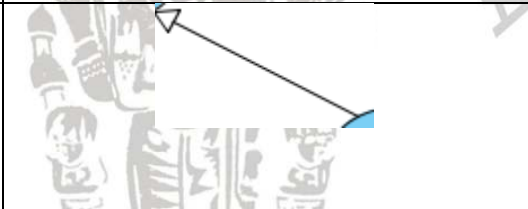
Sumber: Pressman (2010)

Contoh *use case diagram* yang diilustrasikan pada Gambar 2.6 memiliki aktor bernama *User*. Aktor tersebut mempunyai 6 *use case* yaitu *convert music file to new format*, *download music file & save to library*, *capture streaming music & save to library*, *organize the library*, *burn a list songs to CD*, dan *load a list of songs to ipad*. Aktor beserta *use case* yang dimiliki dihubungkan menggunakan relasi asosiasi. Notasi – notasi yang ada pada *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Notasi Pada Use Case Diagram

Nama simbol/notasi	Gambar	Deskripsi
<i>Actor</i>		<i>Actor</i> adalah suatu entitas yang berinteraksi dengan perangkat lunak, bisa berupa seseorang atau sistem lain.
<i>Use case</i>		<i>Use case</i> adalah sebuah representasi dari perilaku perangkat lunak yang dapat diakses oleh pengguna agar tujuan pengguna dapat tercapai.

Tabel 2.1 Notasi Pada Use Case Diagram (Lanjutan)

<p><i>Extend</i></p>		<p><i>Extend</i> adalah relasi yang menunjukkan sebuah <i>use case</i> (yang dituju oleh anak panah) merupakan <i>use case</i> yang dapat diperluas oleh <i>use case</i> lain (asal anak panah) dan bersifat <i>optional</i>.</p>
<p><i>Include</i></p>		<p><i>Include</i> adalah relasi yang menunjukkan sebuah <i>use case</i> memerlukan <i>use case</i> lain agar <i>use case</i> tersebut dapat terpenuhi. <i>Use case</i> yang dituju oleh anak panah merupakan asal mula dari <i>use case</i>.</p>
<p><i>Association</i></p>		<p><i>Association</i> adalah sebuah relasi antara aktor dan <i>use case</i>.</p>
<p><i>Generalization</i></p>		<p><i>Generalization</i> adalah relasi yang menunjukkan sebuah aktor memiliki hubungan dengan aktor lain.</p>

Sumber: Visual Paradigm (2018c)

Setiap *use case* pada *use case diagram* dapat dijelaskan secara lebih detail dengan menggunakan *use case scenario*. *Use case scenario* merupakan penjelasan secara tekstual langkah yang dilakukan oleh aktor ketika berinteraksi dengan perangkat lunak (Kurniawan, 2018). *Use case scenario* memiliki beberapa bagian-bagian yang akan dijelaskan pada Tabel 2.2:

Tabel 2.2 Bagian-bagian pada *use case scenario*

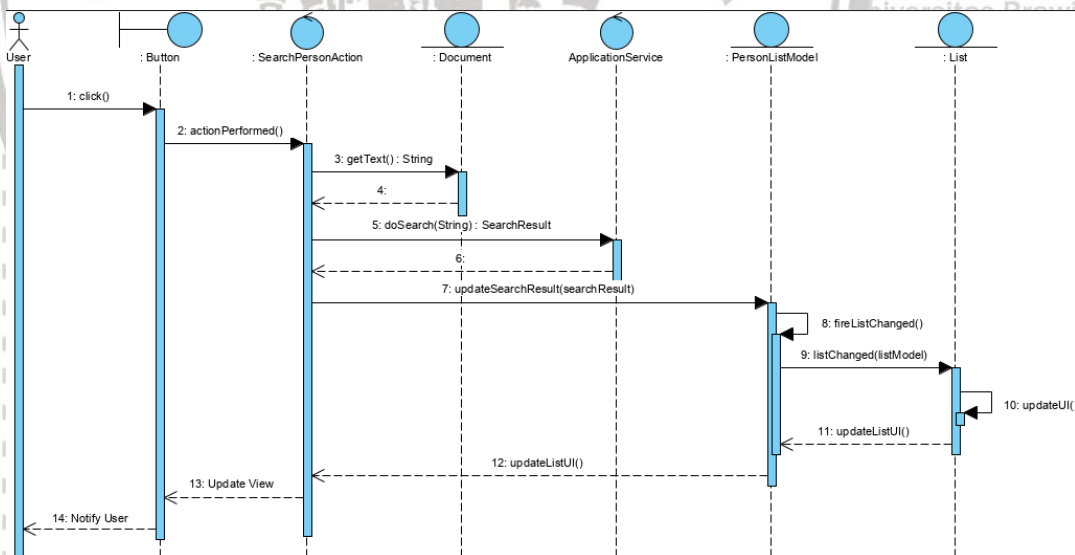
Nama Bagian	Deskripsi
<p>Aktor primer (<i>primary actor</i>)</p>	<p>Menjelaskan mengenai aktor yang menginisiasi layanan sistem untuk mencapai tujuan dari aktor tersebut.</p>
<p>Prakondisi (<i>preconditions</i>)</p>	<p>Menjelaskan mengenai kondisi spesifik yang harus terpenuhi sebelum sebuah <i>use case</i> bisa diinisiasi atau dieksekusi oleh aktor primer.</p>

Alur utama (<i>main flow</i>)	Menjelaskan mengenai jalur interaksi yang mengarahkan pada scenario yang berhasil sehingga tujuan dari aktor bisa terpenuhi.
Alur alternatif (<i>alternative flows</i>)	Menjelaskan mengenai jalur alternatif dari interaksi yang terjadi antar aktor dengan sistem yang mencakup percabangan (pilihan) maupun scenario yang gagal sehingga tujuan dari aktor tidak terpenuhi.
Kondisi akhir (<i>postcondition</i>)	Menjelaskan mengenai kondisi spesifik yang harus terjadi ketika <i>use case</i> berhasil dijalankan atau dieksekusi secara lengkap, sebagai representasi dari tujuan yang ingin dicapai oleh aktor primer.

Sumber: Kurniawan (2018)

2.3.2.2 Sequence Diagram

Sequence diagram merupakan diagram yang digunakan untuk memodelkan interaksi antara objek pada sebuah sistem, serta menggambarkan interaksi antara satu objek dengan objek lainnya (Sommerville, 2011). *Sequence diagram* digambarkan berdasarkan pada penjelasan lebih detail dari *use case scenario*. Contoh *sequence diagram* dapat dilihat pada Gambar 2.7 dan notasi – notasi yang ada pada *sequence diagram* dapat dilihat pada Tabel 2.3.



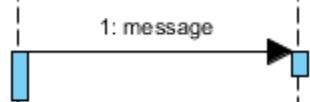

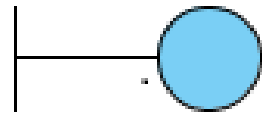




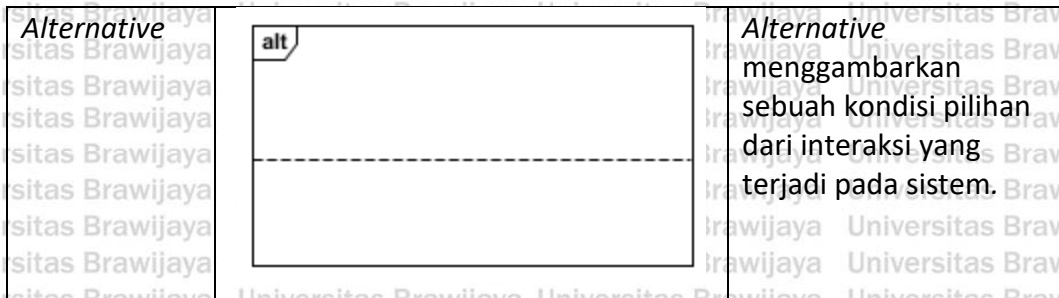
Gambar 2.7 Contoh Sequence Diagram

Sumber: Visual Paradigm (2010)

Contoh *sequence diagram* yang diilustrasikan pada Gambar 2.8 memiliki aktor bernama *User*, memiliki satu klas *boundary* bernama *Button* dan memiliki dua klas *control* bernama *SearchPersonAction* dan *ApplicationService*. *Sequence diagram* memiliki tiga klas *entity* bernama *Document*, *PersonListModel*, dan *List*. *Sequence diagram* juga memiliki beberapa notasi yang dijelaskan pada Tabel 2.3:

Tabel 2.3 Notasi pada Sequence Diagram

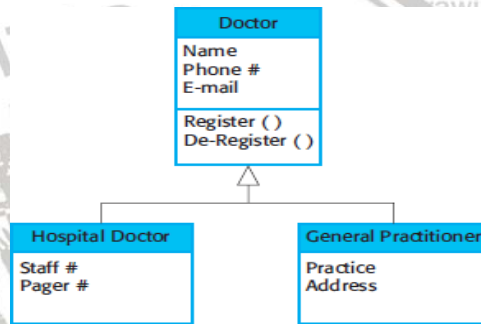
Nama simbol/notasi	Gambar	Deskripsi
<i>Actor</i>		<i>Actor</i> mewakili pengguna manusia atau subjek lainnya.
<i>Lifeline</i>		<i>Lifeline</i> menggambarkan garis waktu dalam bentuk garis putus-putus secara vertical.
<i>Message</i>		<i>Message</i> menggambarkan sebuah komunikasi antar objek.
<i>Return Message</i>		<i>Return Message</i> menggambarkan sebuah pesan kembalian dari sebuah operasi.
<i>Boundary</i>		<i>Boundary</i> menggambarkan sebuah objek yang berinteraksi dengan aktor. <i>Boundary</i> berupa antarmuka pengguna.
<i>Control</i>		<i>Control</i> menggambarkan sebuah objek yang mengatur alur kerja dari sebuah sistem. <i>Lifeline control</i> menjembatani antara <i>Boundary</i> dan <i>Entity</i> .
<i>Entity</i>		<i>Entity</i> menggambarkan sebuah objek yang dapat menyimpan data.



Sumber: Visual Paradigm (2018b)

2.3.2.3 Class Diagram

Class diagram merupakan sebuah diagram yang menggambarkan kelas-kelas yang saling berhubungan yang memiliki atribut dan *method* (Sommerville, 2011). Contoh dari *class diagram* dapat dilihat pada Gambar 2.9.



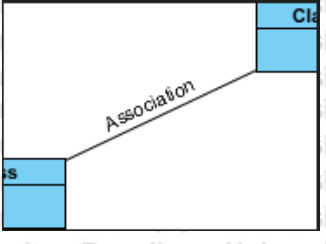
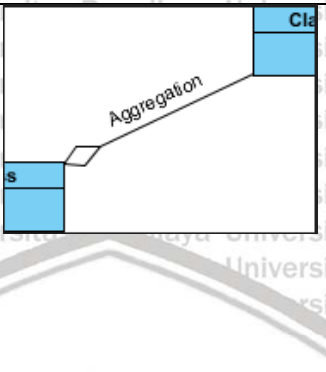
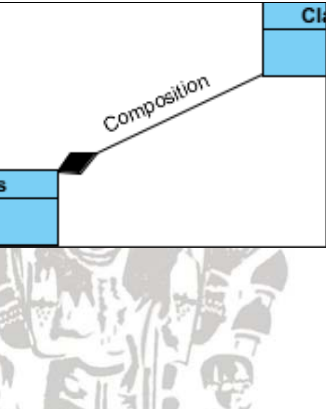
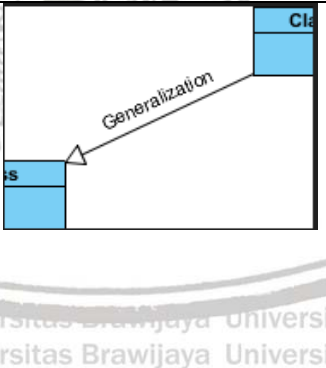
Gambar 2.8 Contoh Class Diagram

Sumber: Sommerville (2011)

Contoh *class diagram* yang diilustrasikan pada Gambar 2.8 memiliki tiga kelas, yaitu kelas *Doctor* sebagai *superclass* dan kelas *Hospital Doctor* serta kelas *General Practitioner* sebagai *subclass*. Kelas *Doctor* memiliki atribut *Name*, *Phone*, dan *E-mail* serta memiliki operasi *Register()* dan *De-Register()*. Kelas *General Practitioner* dan *Hospital Doctor* mewarisi atribut dan operasi dari kelas *Doctor*. *Class diagram* memiliki beberapa notasi yang dijelaskan pada Tabel 2.4:

Tabel 2.4 Notasi pada *Class Diagram*

Nama simbol/notasi	Gambar	Deskripsi
<i>Class</i>		<i>Class</i> menggambarkan kumpulan objek yang memiliki atribut dan operasi (<i>method</i>).

<p><i>Association Relationship</i></p>		<p><i>Association Relationship</i> menggambarkan hubungan antara satu kelas dengan kelas yang lain.</p>
<p><i>Aggregation Relationship</i></p>		<p><i>Aggregation Relationship</i> menggambarkan relasi antar kelas yang menunjukkan sebuah kelas menjadi bagian dari kelas yang lain, namun objek dari suatu kelas dapat berdiri sendiri.</p>
<p><i>Composition Relationship</i></p>		<p><i>Composition Relationship</i> menggambarkan relasi antar kelas yang lebih kuat dari <i>aggregation relationship</i>. Objek dari suatu kelas tidak dapat berdiri sendiri tanpa adanya objek dari kelas yang lain.</p>
<p><i>Generalization Relationship</i></p>		<p><i>Generalization Relationship</i> menggambarkan relasi antar kelas dimana sebuah kelas (asal panah) mewarisi atribut dan operasi dari kelas yang lain (kelas yang ditunjuk anak panah).</p>

Sumber: Visual Paradigm (2018a)

2.4 Teknologi Pengembangan Sistem

2.4.1 MySQL

Basis data merupakan suatu kumpulan data yang terhubung yang disimpan secara bersama-sama pada suatu aplikasi (Date, 2010). Menurut Date (2010), suatu tempat yang dapat untuk menampung sekumpulan data yang terkait dan

terkomputerisasi yang bertujuan untuk dapat memuat informasi disebut DBMS (*Database Management System*).

Menurut Date (2010), MySQL (*MyStructured Query Language*) adalah sebuah DBMS yang bersifat relasional, yaitu dapat menyimpan data ke dalam tabel-tabel yang saling berhubungan. MySQL digunakan sebagai basis data untuk menyimpan data ketika mengembangkan perangkat lunak karena sifat dari MySQL, yaitu dapat terdistribusi secara gratis (*open source*), konsisten (*reliable*), banyak pengguna (*multiuser*). Perancangan basis data yang dilakukan pada penelitian ini menggunakan MySQL. CDM dan PDM pada perancangan data dapat dijadikan acuan untuk dapat merancang basis data pada MySQL.

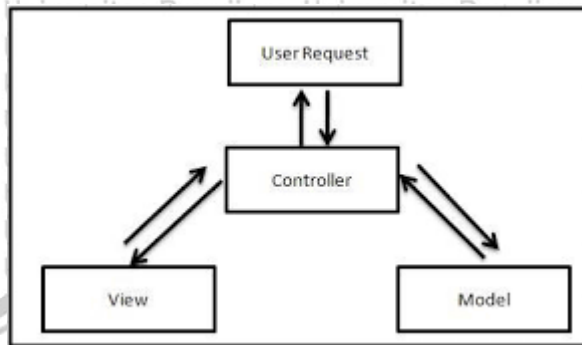
2.4.2 Framework CodeIgniter

Framework merupakan kumpulan instruksi yang dikumpulkan dalam *class* dengan fungsi masing-masing untuk dapat memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama secara berulang-ulang serta dapat menghemat waktu (Betha, 2012).

CodeIgniter merupakan sebuah *framework* yang digunakan untuk membuat aplikasi yang berbasis web yang menggunakan bahasa pemrograman PHP dan bersifat *open source* (Betha, 2012). Hal-hal yang mendasari *CodeIgniter* menjadi *open source* adalah untuk dapat memudahkan pengembang suatu perangkat lunak untuk mengerjakan dengan cepat dan efisien dibandingkan dengan menuliskan seluruh kode program dari awal. *CodeIgniter* menggunakan konsep MVC (*Model, View, Controller*), yaitu memisahkan bagian logika dari kode program dengan tampilan. Penggunaan metode MVC juga digunakan untuk memudahkan *developer* atau pengguna dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal (Betha, 2012). *CodeIgniter* memiliki beberapa kelas yang berbentuk sebuah *library* yang berfungsi untuk membantu *programmer* dalam mengembangkan suatu aplikasi yang berbasis web. *Model* pada MVC dengan *entity* pada ECB (*entity, control, boundary*) berbeda karena *model* hanya berkaitan dengan *database* sedangkan *entity* berkaitan dengan objek sehingga bisa mengelola informasi. *Controller* pada MVC berbeda juga dengan *control* pada ECB karena *controller* mengelola interaksi antar *view* dan *model* sedangkan *control* mengelola interaksi antar objek, yaitu *boundary* dan *entity*. Penerapan konsep ECB pada *framework codeigniter* menyesuaikan *configuration file* (penempatan *file*) dengan menempatkan *entity* ke dalam *model*, *control* ke dalam *controller*, dan *boundary* ke dalam *view*.

Konsep pemrograman berorientasi objek pada *codeigniter* menggunakan konsep dua kelas induk, yaitu *CI_Controller* dan *CI_Model*. *CI_Controller* dan *CI_Model* memiliki masing-masing *method construct* dan atribut. *CI_Model* dan *CI_Controller* merupakan sebuah dasar pengaturan *library model* dan *controller* dari *framework codeigniter*. Setiap *class controller* harus memiliki *method* dan atribut dari kelas *CI_Controller* sehingga semua *class controller* perlu melakukan *extend* pada kelas *CI_Controller*. Setiap *class model* juga harus memiliki *method* dan atribut dari kelas *CI_Model* sehingga semua kelas *model* perlu melakukan *extend*

pada kelas *CI_Model*. Kedua kelas induk harus membuat hak akses setiap *method* dan *attribute* menjadi *protected* atau *public* sehingga dapat digunakan oleh setiap kelas turunannya yang menjadikan *codeigniter* menerapkan konsep *inheritance*. Konsep *inheritance* yang digunakan untuk menghindari duplikasi kode program. Teknik pemrograman MVC terbagi menjadi tiga bagian penting antara lain: *model*, *view*, dan *controller*, seperti Gambar 2.9.



Gambar 2.9 Bentuk MVC

Sumber: Betha (2012)

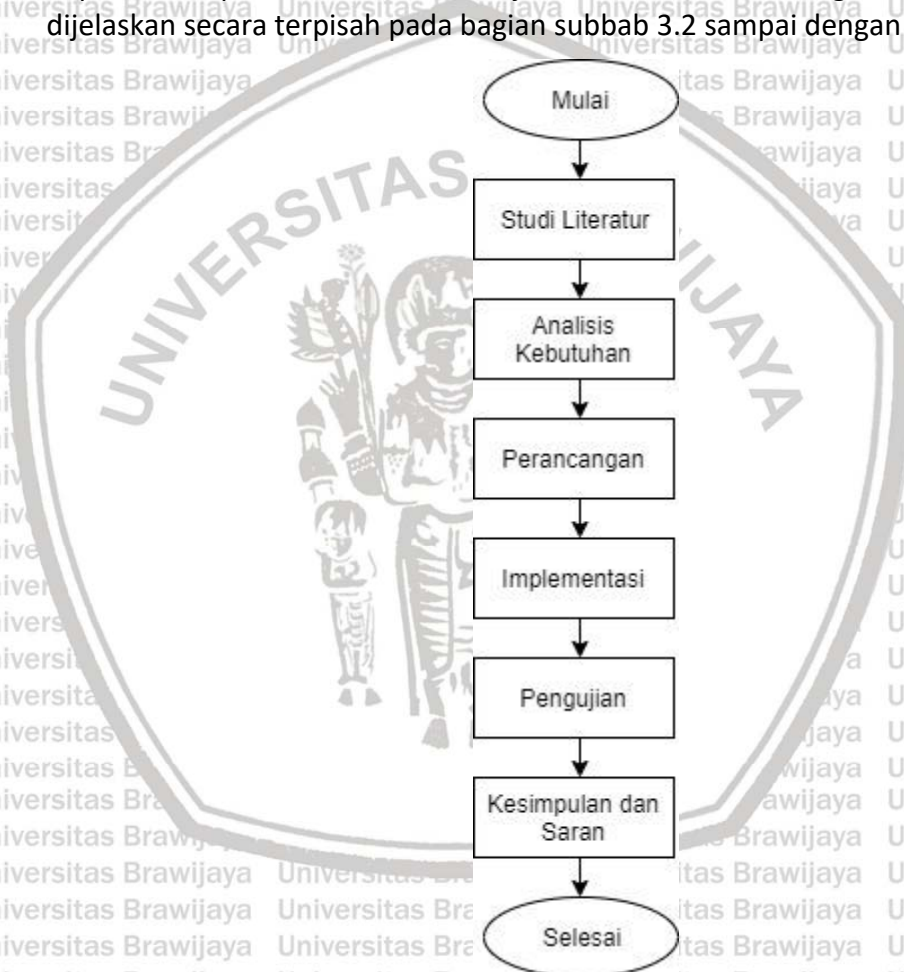
MVC terdiri dari 3 bagian:

1. *Model*: Merupakan kode program yang digunakan untuk menghubungkan dengan database MySQL sekaligus untuk memanipulasinya (*input, edit, delete*).
2. *View*: Merupakan komponen yang menampilkan antarmuka (*interface*) kepada pengguna sistem.
3. *Controller*: Merupakan kode program yang digunakan untuk mengontrol aliran data atau dengan kata lain sebagai penghubung *model* dan *view*.

BAB 3 METODOLOGI PENELITIAN

3.1 Deskripsi Umum

Tahapan ini dilakukan untuk membahas langkah-langkah yang dilakukan dalam pengembangan sistem informasi manajemen Masjid Al Ghifari berbasis web yang dimulai dari tahap studi literatur sampai dengan tahap kesimpulan dan saran. Penelitian ini menggunakan pendekatan model *waterfall* dalam mengembangkan sistem yang akan dibuat. Tahapan yang digunakan pada pengembangan sistem informasi manajemen Masjid Al Ghifari menggunakan pendekatan berbasis objek. Diagram alir yang digunakan dalam penelitian ini dapat dilihat pada Gambar 3.1. Penjelasan detail dari masing-masing tahapan ini dijelaskan secara terpisah pada bagian subbab 3.2 sampai dengan 3.7.



Gambar 3.1 Diagram Alir Metode Penelitian

3.2 Studi Literatur

Studi literatur dilakukan untuk mendapatkan teori-teori yang relevan berdasarkan hasil penelitian sebelumnya dan diperlukan dalam pengembangan sebuah sistem yang akan dibuat. Sumber literatur yang bisa dijadikan sebuah rujukan terdiri dari artikel atau jurnal ilmiah dan buku. Teori-teori yang berkaitan dengan penelitian ini terdiri dari:

1. Kajian pustaka
2. Manajemen Masjid Al Ghifari
3. Rekayasa perangkat lunak
 - a. Model *waterfall*
 - b. Pendekatan berorientasi objek
4. Teknologi Pengembangan Sistem
 - a. *Framework Codeigniter*
 - b. MySQL

3.3 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mendapatkan kebutuhan dari sistem informasi manajemen Masjid Al Ghifari. Tahapan yang dilakukan dalam analisis kebutuhan terdiri dari elisitasi kebutuhan, spesifikasi kebutuhan, dan validasi. Elisitasi kebutuhan digunakan untuk mengidentifikasi permasalahan yang ada di Masjid Al Ghifari, menggali kebutuhan untuk keperluan sistem, dan mengidentifikasi aktor-aktor yang terlibat dalam sistem yang akan dibangun. Teknik elisitasi kebutuhan yang digunakan adalah wawancara dan observasi. Wawancara dilakukan secara langsung bersama Kepala Kerumahtanggan Masjid Al Ghifari. Observasi dilakukan dengan cara datang ke Masjid Al Ghifari melakukan pengamatan secara langsung untuk melihat kegiatan yang ada di Masjid Al Ghifari serta bagaimana proses kegiatan itu dilakukan. Spesifikasi kebutuhan digunakan untuk menjelaskan kebutuhan sistem yang sudah didefinisikan sebelumnya secara lebih detail dan tepat. Validasi dilakukan untuk memastikan kebutuhan yang sudah didefinisikan sudah benar dan lengkap sesuai dengan keinginan pengguna. Kebutuhan yang sudah didefinisikan dapat dimodelkan dengan menggunakan diagram UML yaitu *use case diagram* dan *use case scenario*.

3.4 Perancangan

Perancangan dilakukan untuk membuat rancangan dari sistem informasi manajemen Masjid Al Ghifari berdasarkan hasil analisis kebutuhan yang telah dilakukan. Perancangan yang dilakukan terdiri dari perancangan arsitektur, perancangan data, perancangan komponen, dan perancangan antarmuka:

a. Perancangan Arsitektur

Perancangan arsitektur dilakukan untuk mengidentifikasi klas-klas dalam sistem informasi manajemen Masjid Al Ghifari dan relasi antar klas yang ada. Perancangan arsitektur dimodelkan menggunakan pemodelan *sequence diagram* dan *class diagram*. Pemodelan *sequence diagram* dibuat berdasarkan *use case scenario* dan *class diagram* dibuat berdasarkan hasil pemodelan *sequence diagram*. Sampel yang digunakan pada pemodelan *sequence diagram* menggunakan tiga fungsional utama. *Class diagram* digunakan untuk mendefinisikan klas-klas beserta relasinya.

b. Perancangan Data

Perancangan data dilakukan untuk merancang struktur data pada *database* yang digunakan di sistem informasi manajemen Masjid Al Ghifari. CDM digunakan untuk menggambarkan entitas yang memiliki atribut dan hubungan antar entitas lain. CDM direpresentasikan dalam bentuk *Entity Relationship Diagram* (ERD). PDM digunakan untuk menggambarkan tabel yang berisi atribut, tipe data setiap kolom, *primary key*, *foreign key*, dan hubungan antar tabel. PDM dibuat berdasarkan CDM yang sudah valid. PDM merupakan gambaran secara detail yang merepresentasikan *database* pada sistem informasi manajemen Masjid Al Ghifari yang akan dibangun.

c. Perancangan Komponen

Perancangan komponen dilakukan untuk merancang algoritme operasi/*method* yang terdapat pada klas sistem informasi manajemen Masjid Al Ghifari dalam bentuk *pseudocode*. Sampel yang digunakan pada perancangan komponen adalah tiga fungsional utama.

d. Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk merancang tampilan sistem informasi manajemen Masjid Al Ghifari sesuai dengan kebutuhan sistem.

3.5 Implementasi

Implementasi dilakukan untuk menerapkan hasil perancangan ke dalam bentuk sebuah bahasa pemrograman. Implementasi arsitektur dan algoritme dilakukan berdasarkan pemodelan *class diagram* dan perancangan komponen dengan menggunakan bahasa pemrograman PHP. Implementasi basis data dilakukan berdasarkan pemodelan PDM dengan menggunakan MySQL. Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka dengan menggunakan HTML.

3.6 Pengujian

Pengujian dilakukan untuk memastikan bahwa sistem yang dibangun sudah sesuai dengan kebutuhan dan hasil perancangan. Pengujian yang dilakukan terdiri dari pengujian unit, pengujian integrasi, dan pengujian validasi:

a. Pengujian Unit

Pengujian unit dilakukan dengan teknik *white-box testing* menggunakan metode *basis path testing*. Kasus uji yang digunakan pada pengujian unit berdasarkan dari hasil perancangan komponen. Pengujian unit menggunakan tiga sampel uji.

b. Pengujian Integrasi

Pengujian integrasi dilakukan dengan teknik *white-box testing* menggunakan metode *basis path testing*. Kasus uji yang digunakan pada pengujian integrasi

berdasarkan klas-klas yang saling berhubungan pada perancangan komponen. Pengujian integrasi menggunakan tiga sampel uji.

c. Pengujian Validasi

Pengujian validasi dilakukan dengan teknik *black-box testing* menggunakan metode *scenario-based testing*. Pengujian validasi dilakukan untuk memastikan seluruh kebutuhan sistem sudah diimplementasi baik dan benar. Kasus uji yang digunakan pada pengujian validasi berdasarkan *use case scenario*. Pengujian kebutuhan nonfungsional pada penelitian ini menggunakan pengujian *compatibility*. Pengujian *compatibility* dilakukan dengan bantuan *software SortSiteEval* dan pengujian *compatibility* akan dilakukan pada berbagai macam *browser* seperti *Chrome, Mozilla, dan Opera*.

3.7 Kesimpulan dan Saran

Kesimpulan dan saran dilakukan setelah semua tahapan sudah dilakukan. Kesimpulan dilakukan untuk menjawab rumusan masalah yang sudah dibuat. Saran dilakukan untuk memberikan pertimbangan dalam pengembangan sistem selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

4.1 Elisitasi Kebutuhan

Elisitasi kebutuhan dilakukan dengan teknik wawancara dan observasi. Wawancara dilakukan bersama Kepala Kerumahtanggaan Masjid AL Ghifari, yaitu Bapak Agus Juanedi. Pertanyaan yang diajukan kepada Bapak Agus Junaedi meliputi apa saja kegiatan yang ada di Masjid Al Ghifari, cara menginformasikan kegiatan, pencatatan keuangan, dan permasalahan yang ada di Masjid Al Ghifari. Observasi dilakukan untuk mengamati aktivitas dan pengelolaan kegiatan serta keuangan di Masjid Al Ghifari. Elisitasi kebutuhan menghasilkan aktor-aktor yang terlibat dengan sistem serta mendapatkan kebutuhan fungsional dan nonfungsional sistem.

4.1.1 Identifikasi Aktor

Identifikasi aktor digunakan untuk mengetahui aktor-aktor yang terlibat atau berinteraksi dengan sistem. Aktor yang teridentifikasi menggunakan sistem berjumlah empat aktor, yaitu Bendahara, Pengguna, dan Ketua Pengurus Masjid. Deskripsi masing-masing aktor dijelaskan pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1	Anggota Bidang Ubudiyah	Aktor yang mengurus bagian syiar dan dakwah serta membuat kegiatan kajian di Masjid Al Ghifari. Aktor sudah teridentifikasi oleh sistem
2	Bendahara	Aktor yang mengurus keuangan di Masjid Al Ghifari. Aktor sudah teridentifikasi oleh sistem.
3	Pengguna	Aktor yang belum teridentifikasi pada sistem.
4	Ketua Pengurus Masjid	Aktor yang mengontrol keuangan dan kegiatan yang ada di Masjid Al Ghifari. Aktor sudah teridentifikasi oleh sistem.

4.1.2 Kebutuhan Fungsional

Daftar kebutuhan fungsional yang terdapat pada sistem manajemen Masjid Al Ghifari yang dikelompokkan berdasarkan masing-masing aktor dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Fungsional Sistem Manajemen Masjid Al Ghifari

No	Aktor	Kebutuhan Fungsional
1.	Bendahara	<ul style="list-style-type: none"> - Sistem harus mampu menyediakan fungsi untuk menambahkan laporan keuangan harian - Sistem harus mampu menyediakan fungsi untuk mengubah laporan keuangan harian - Sistem harus mampu menyediakan fungsi untuk menghapus laporan keuangan harian - Sistem harus mampu menyediakan fungsi untuk mencetak laporan keuangan harian - Sistem harus mampu menyediakan fungsi untuk mencetak laporan keuangan bulanan - Sistem harus mampu menyediakan fungsi untuk <i>logout</i>
2.	Pengguna	<ul style="list-style-type: none"> - Sistem harus mampu menyediakan fungsi untuk <i>login</i> - Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail kegiatan insidental - Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail kegiatan rutin - Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail pengumuman - Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail artikel - Sistem harus mampu menyediakan fungsi untuk melihat foto - Sistem harus mampu menyediakan fungsi untuk melihat daftar imam sholat - Sistem harus mampu menyediakan fungsi untuk melihat jadwal sholat

3.	Ketua Pengurus Masjid	<ul style="list-style-type: none"> - Sistem harus mampu untuk lihat laporan keuangan harian - Sistem harus mampu untuk lihat laporan keuangan bulanan - Sistem harus mampu menyediakan fungsi untuk <i>logout</i>
4	Anggota Bidang Ubudiyah	<ul style="list-style-type: none"> - Sistem harus mampu menyediakan fungsi untuk menambahkan pengumuman - Sistem harus mampu menyediakan fungsi untuk mengubah pengumuman - Sistem harus mampu menyediakan fungsi untuk menghapus pengumuman - Sistem harus mampu menyediakan fungsi untuk menambahkan foto - Sistem harus mampu menyediakan fungsi untuk menghapus foto - Sistem harus mampu menyediakan fungsi untuk menambahkan artikel - Sistem harus mampu menyediakan fungsi untuk mengubah artikel - Sistem harus mampu menyediakan fungsi untuk menghapus artikel - Sistem harus mampu menyediakan fungsi untuk menambahkan kegiatan insidental - Sistem harus mampu menyediakan fungsi untuk mengubah kegiatan insidental - Sistem harus mampu menyediakan fungsi untuk menghapus kegiatan insidental - Sistem harus mampu menyediakan fungsi untuk menambahkan jadwal imam sholat - Sistem harus mampu menyediakan fungsi untuk mengubah jadwal imam sholat - Sistem harus mampu menyediakan fungsi untuk menghapus jadwal imam sholat - Sistem harus mampu menyediakan fungsi untuk menambahkan jadwal sholat

	<ul style="list-style-type: none"> - Sistem harus mampu menyediakan fungsi untuk menghapus jadwal sholat - Sistem harus mampu menyediakan fungsi untuk mengubah jadwal sholat - Sistem harus mampu menyediakan fungsi untuk menambahkan kegiatan rutin - Sistem harus mampu menyediakan fungsi untuk mengubah kegiatan rutin - Sistem harus mampu menyediakan fungsi untuk menghapus kegiatan rutin - Sistem harus mampu menyediakan fungsi untuk <i>logout</i>
--	---

4.1.3 Kebutuhan Nonfungsional

Kebutuhan nonfungsional yang digunakan pada sistem manajemen Masjid Al Ghifari adalah *compatibility*. *Compatibility* merupakan kemampuan perangkat lunak yang dapat dijalankan diberbagai jenis web *browser*. Web *browser* yang digunakan dalam pengujian *compatibility*, yaitu *Google Chrome*, *Mozilla Firefox* dan *Opera Mini*.

4.2 Spesifikasi Kebutuhan

Spesifikasi kebutuhan digunakan untuk menjelaskan kebutuhan fungsional secara detail. Setiap kebutuhan fungsional akan diberikan kode "SIMAGFARI_F_XX" dan spesifikasinya diberikan kode "SIMAGFARI_F_XX_YY". Kode "SIMAGFARI" merupakan singkatan dari Sistem Informasi Manajemen Masjid Al Ghifari. Kode "F" menunjukkan kebutuhan fungsional dan kode "NF" menunjukkan kebutuhan nonfungsional. Kode "XX" menunjukan nomor kebutuhan dan kode "YY" menunjukkan nomor spesifikasi kebutuhan. Tabel 4.3 menunjukkan spesifikasi kebutuhan fungsional.

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

Kode Kebutuhan	Kebutuhan fungsional dan spesifikasi	Aktor	Use case
SIMAGFARI_F_01	Sistem harus mampu menyediakan fungsi untuk <i>login</i> .	Pengguna	<i>Login</i>
SIMAGFARI_F_02	Sistem harus mampu menyediakan fungsi untuk <i>logout</i> .	Bendahara, Ketua pengurus masjid,	<i>Logout</i>

		Anggota Bidang Ubudiyah	
SIMAGFARI_F_03	Sistem harus mampu menyediakan fungsi untuk menambahkan artikel.	Anggota Bidang Ubudiyah	Tambah artikel
SIMAGFARI_F_03_01	Sistem menyediakan <i>form</i> untuk menambahkan artikel yang terdiri dari judul artikel, isi artikel, gambar artikel, tanggal artikel.		
SIMAGFARI_F_04	Sistem harus mampu menyediakan fungsi untuk mengubah artikel.	Anggota Bidang Ubudiyah	Ubah artikel
SIMAGFARI_F_04_01	Sistem menyediakan form untuk mengubah data artikel yang terdiri dari judul artikel, isi artikel, dan tanggal artikel.		
SIMAGFARI_F_05	Sistem harus mampu menyediakan fungsi untuk menghapus artikel.	Anggota Bidang Ubudiyah	Hapus artikel
SIMAGFARI_F_05_01	Sistem menyediakan fungsi untuk menghapus artikel.		
SIMAGFARI_F_06	Sistem harus mampu menyediakan fungsi untuk menambahkan pengumuman.	Anggota Bidang Ubudiyah	Tambah pengumuman
SIMAGFARI_F_06_01	Sistem menyediakan <i>form</i> untuk menambahkan pengumuman yang terdiri dari judul pengumuman, isi pengumuman, gambar pengumuman, tanggal pengumuman.		
SIMAGFARI_F_07	Sistem harus mampu menyediakan fungsi untuk mengubah pengumuman.	Anggota Bidang Ubudiyah	Ubah pengumuman

SIMAGFARI_F_07_01	Sistem menyediakan form untuk mengubah data pengumuman yang terdiri dari judul pengumuman, isi pengumuman, dan tanggal pengumuman.		
SIMAGFARI_F_08	Sistem harus mampu menyediakan fungsi untuk menghapus pengumuman.	Anggota Bidang Ubudiyah	Hapus pengumuman
SIMAGFARI_F_08_01	Sistem menyediakan fungsi untuk menghapus pengumuman.		
SIMAGFARI_F_09	Sistem harus mampu menyediakan fungsi untuk menambahkan foto.	Anggota Bidang Ubudiyah	Tambah foto
SIMAGFARI_F_09_01	Sistem menyediakan <i>form</i> untuk menambahkan foto yang terdiri dari nama foto dan <i>upload</i> foto.		
SIMAGFARI_F_10	Sistem harus mampu menyediakan fungsi untuk menghapus foto.	Anggota Bidang Ubudiyah	Hapus foto
SIMAGFARI_F_10_01	Sistem menyediakan fungsi untuk menghapus foto.		
SIMAGFARI_F_11	Sistem harus mampu menyediakan fungsi untuk menambahkan kegiatan insidental.	Anggota Bidang Ubudiyah	Tambah kegiatan insidental
SIMAGFARI_F_11_01	Sistem menyediakan <i>form</i> untuk menambahkan kegiatan insidental yang terdiri dari judul kegiatan, isi kegiatan, gambar kegiatan, tanggal kegiatan.		
SIMAGFARI_F_12	Sistem harus mampu menyediakan fungsi untuk mengubah kegiatan insidental.	Anggota Bidang Ubudiyah	Ubah kegiatan insidental
SIMAGFARI_F_12_01	Sistem menyediakan form untuk mengubah data kegiatan insidental yang		

	terdiri dari judul kegiatan, isi kegiatan, dan tanggal kegiatan.		
SIMAGFARI_F_13	Sistem harus mampu menyediakan fungsi untuk menghapus kegiatan insidental.	Anggota Bidang Ubudiyah	Hapus kegiatan insidental
SIMAGFARI_F_13_01	Sistem menyediakan fungsi untuk menghapus kegiatan insidental.		
SIMAGFARI_F_14	Sistem harus mampu menyediakan fungsi untuk menambahkan kegiatan rutin.	Anggota Bidang Ubudiyah	Tambah kegiatan rutin
SIMAGFARI_F_14_01	Sistem menyediakan <i>form</i> untuk menambahkan kegiatan rutin yang terdiri dari judul kegiatan, isi kegiatan, gambar kegiatan, tanggal kegiatan.		
SIMAGFARI_F_15	Sistem harus mampu menyediakan fungsi untuk mengubah kegiatan rutin.	Anggota Bidang Ubudiyah	Ubah kegiatan rutin
SIMAGFARI_F_15_01	Sistem menyediakan form untuk mengubah data kegiatan rutin yang terdiri dari judul kegiatan, isi kegiatan, dan tanggal kegiatan.		
SIMAGFARI_F_16	Sistem harus mampu menyediakan fungsi untuk menghapus kegiatan rutin.	Anggota Bidang Ubudiyah	Hapus kegiatan rutin
SIMAGFARI_F_16_01	Sistem menyediakan fungsi untuk menghapus kegiatan rutin.		
SIMAGFARI_F_17	Sistem harus mampu menyediakan fungsi untuk menambah jadwal imam sholat	Anggota Bidang Ubudiyah	Tambah jadwal imam sholat

SIMAGFARI_F_17_01	Sistem menyediakan <i>form</i> untuk menambahkan jadwal imam sholat yang terdiri dari imam Zuhur, imam Asar, imam Magrib, imam Isya, imam Subuh, tanggal imam sholat.		
SIMAGFARI_F_18	Sistem harus mampu menyediakan fungsi untuk mengubah jadwal imam sholat.	Anggota Bidang Ubudiyah	Ubah jadwal Imam Sholat
SIMAGFARI_F_18_01	Sistem menyediakan form untuk mengubah data jadwal imam sholat yang terdiri dari imam Zuhur, imam Asar, imam Magrib, imam Isya, imam Subuh, tanggal imam sholat.		
SIMAGFARI_F_19	Sistem harus mampu menyediakan fungsi untuk menghapus jadwal imam sholat.	Anggota Bidang Ubudiyah	Hapus jadwal imam sholat
SIMAGFARI_F_19_01	Sistem menyediakan fungsi untuk menghapus jadwal imam sholat.		
SIMAGFARI_F_20	Sistem harus mampu menyediakan fungsi untuk menambah jadwal sholat.	Anggota Bidang Ubudiyah	Tambah jadwal sholat
SIMAGFARI_F_20_01	Sistem menyediakan <i>form</i> untuk menambahkan jadwal sholat yang terdiri dari jadwal Zuhur, jadwal Asar, jadwal Magrib, jadwal Isya, jadwal Subuh, tanggal jadwal sholat.		
SIMAGFARI_F_21	Sistem harus mampu menyediakan fungsi untuk mengubah jadwal sholat.	Anggota Bidang Ubudiyah	Ubah jadwal sholat
SIMAGFARI_F_21_01	Sistem menyediakan form untuk mengubah data jadwal sholat yang terdiri dari jadwal Zuhur, jadwal		

	Asar, jadwal Magrib, jadwal Isya, jadwal Subuh, tanggal jadwal sholat.		
SIMAGFARI_F_22	Sistem harus mampu menyediakan fungsi untuk menghapus jadwal sholat.	Anggota Bidang Ubudiyah	Hapus jadwal sholat
SIMAGFARI_F_22_01	Sistem menyediakan fungsi untuk menghapus jadwal sholat.		
SIMAGFARI_F_23	Sistem harus mampu menyediakan fungsi untuk menambahkan laporan keuangan harian.	Bendahara	Tambah laporan keuangan harian
SIMAGFARI_F_23_01	Sistem menyediakan <i>form</i> untuk menambahkan laporan keuangan harian yang terdiri dari keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran.		
SIMAGFARI_F_24	Sistem harus mampu menyediakan fungsi untuk mengubah data laporan keuangan harian.	Bendahara	Ubah laporan keuangan harian
SIMAGFARI_F_24_01	Sistem menyediakan form untuk mengubah data laporan keuangan harian yang terdiri dari keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran.		
SIMAGFARI_F_25	Sistem harus mampu menyediakan fungsi untuk menghapus laporan keuangan harian.	Bendahara	Hapus laporan keuangan harian
SIMAGFARI_F_25_01	Sistem menyediakan fungsi untuk menghapus laporan keuangan harian.		

SIMAGFARI_F_26	Sistem harus mampu menyediakan fungsi untuk mencetak laporan keuangan harian.	Bendahara	Cetak laporan keuangan harian
SIMAGFARI_F_26_01	Sistem menyediakan fungsi untuk mencetak laporan keuangan harian.		
SIMAGFARI_F_27	Sistem harus mampu menyediakan fungsi untuk mencetak laporan keuangan bulanan.	Bendahara	Cetak laporan keuangan bulanan
SIMAGFARI_F_27_01	Sistem menyediakan fungsi untuk mencetak laporan keuangan bulanan.		
SIMAGFARI_F_28	Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail kegiatan insidental.	Pengguna	Lihat daftar dan detail kegiatan insidental
SIMAGFARI_F_28_01	Sistem menampilkan daftar dan detail kegiatan insidental.		
SIMAGFARI_F_29	Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail kegiatan rutin	Pengguna	Lihat daftar dan detail kegiatan rutin
SIMAGFARI_F_29_01	Sistem menampilkan daftar dan detail kegiatan rutin.		
SIMAGFARI_F_30	Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail pengumuman.	Pengguna	Lihat daftar dan detail pengumuman
SIMAGFARI_F_30_01	Sistem menampilkan daftar dan detail pengumuman.		

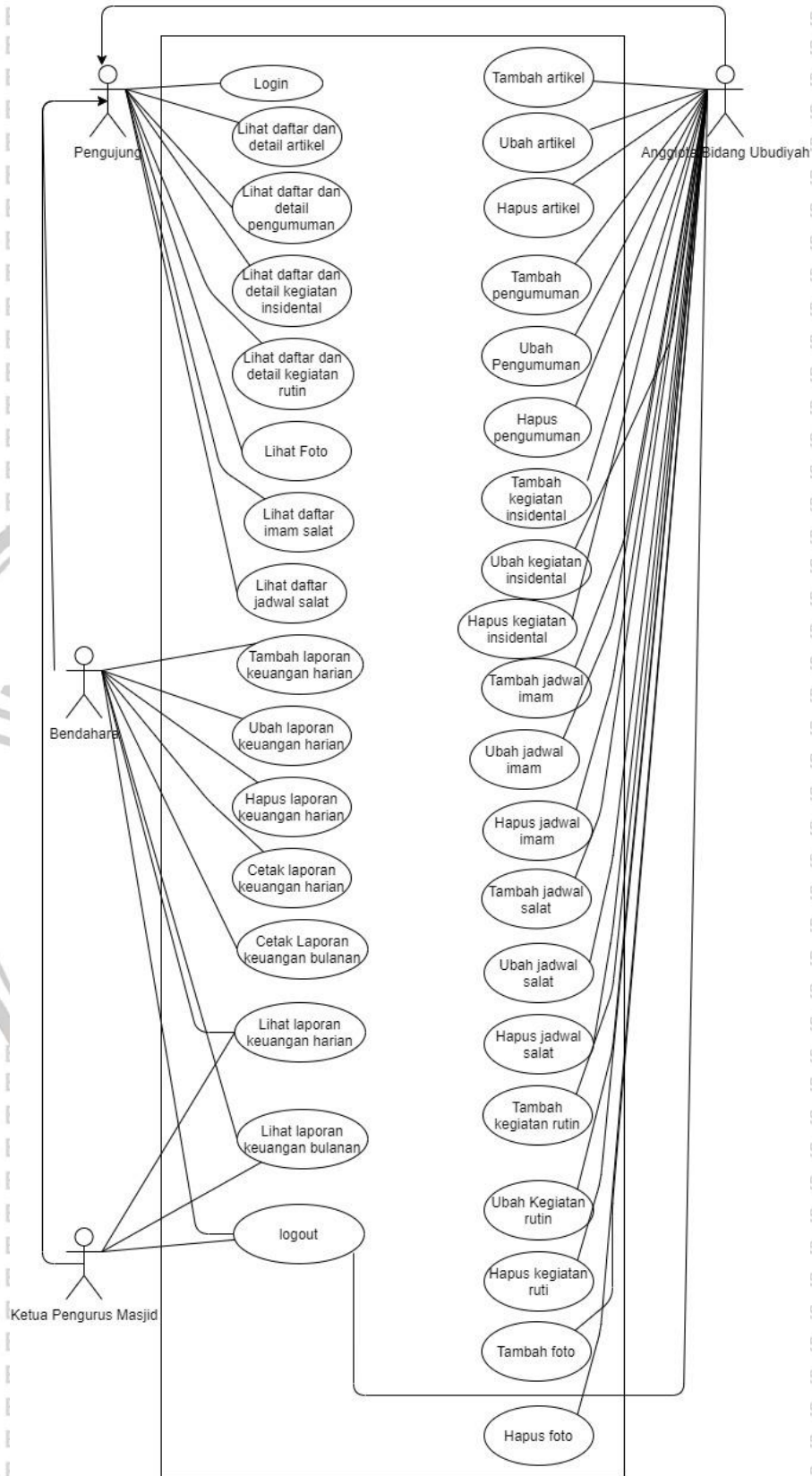
SIMAGFARI_F_31	Sistem harus mampu menyediakan fungsi untuk melihat daftar dan detail artikel.	Pengguna	Lihat daftar dan detail artikel
SIMAGFARI_F_31_01	Sistem menampilkan daftar dan detail artikel.		
SIMAGFARI_F_32	Sistem harus mampu menyediakan fungsi untuk melihat foto.	Pengguna	Lihat foto
SIMAGFARI_F_32_01	Sistem menampilkan daftar foto.		
SIMAGFARI_F_33	Sistem harus mampu menyediakan fungsi untuk melihat daftar imam sholat.	Pengguna	Lihat daftar imam sholat
SIMAGFARI_F_33_01	Sistem menampilkan daftar imam sholat.		
SIMAGFARI_F_34	Sistem harus mampu menyediakan fungsi untuk melihat daftar jadwal sholat.	Pengguna	Lihat daftar jadwal sholat
SIMAGFARI_F_34_01	Sistem menampilkan daftar jadwal sholat.		
SIMAGFARI_F_35	Sistem harus mampu untuk mengontrol laporan keuangan harian.	Ketua pengurus masjid,	Lihat laporan keuangan harian
SIMAGFARI_F_35_01	Sistem dapat mengontrol laporan keuangan harian yang sudah dibuat.	Bendahara, dan Anggota Bidang Ubudiyah	
SIMAGFARI_F_36	Sistem harus mampu untuk mengontrol laporan keuangan bulanan yang sudah dibuat.	Ketua pengurus masjid,	Lihat laporan keuangan bulanan
SIMAGFARI_F_36_01	Sistem dapat mengontrol laporan bulanan yang sudah dibuat.	Bendahara, dan Anggota Bidang Ubudiyah	

Tabel 4.4 Spesifikasi Kebutuhan Nonfungsional

No	Kode	Deskripsi	Parameter
1	SIMAGFARI _NF_01	Sistem dapat dijalankan diberbagai jenis browser seperti <i>Firefox</i> , <i>Google Chrome</i> , dan <i>Opera</i>	<i>Compatibility</i>



4.2.1 Use Case Diagram



Gambar 4.1 Use Case Diagram Sistem Informasi Manajemen Masjid Al Ghifari

4.2.2 Use Case Scenario

Tabel 4.5 Use case scenario untuk login

Login	
Objective	Aktor masuk ke dalam sistem sesuai dengan hak aksesnya
Actors	Pengguna
Preconditions	Sistem menampilkan halaman <i>login</i> yang sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> dan <i>password</i> dan meminta sistem untuk memproses 2. Sistem menampilkan halaman utama aktor sesuai dengan hak aksesnya
Alternative Flows	<ol style="list-style-type: none"> 1a. Jika <i>username</i> dan/atau <i>password</i> kosong, maka sistem akan menampilkan “Harap isi <i>Username</i> dan/atau <i>Password</i>”. 1b. Jika <i>username</i> dan/atau <i>password</i> tidak valid, maka sistem akan menampilkan pesan “<i>Username</i> dan/atau <i>Password</i> yang anda masukkan salah, coba lagi!!!”
Postconditions	Halaman sistem sistem terbuka sesuai dengan hak akses dari aktor yang sudah dikenali oleh sistem

Tabel 4.6 Use case scenario untuk logout

Logout	
Objective	Aktor keluar dari sistem
Actors	Bendahara, Anggota Bidang Ubudiyah dan Ketua Pengurus Masjid
Preconditions	Aktor sudah berada di dalam sistem
Main Flow	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk <i>logout</i> 2. Sistem akan menampilkan sebuah pesan “Apakah anda yakin keluar dari sistem” dan setelah itu sistem memunculkan pilihan “Batalkan” atau “Ya, saya yakin” 3. Aktor memerintahkan sistem untuk memproses 4. Sistem menampilkan halaman <i>login</i>
Alternative Flows	2a. Jika aktor memilih “Batalkan” maka aktor tidak berhasil keluar dari sistem dan menampilkan halaman awal sistem

<i>Postconditions</i>	Aktor berhasil keluar dari sistem
-----------------------	-----------------------------------

Tabel 4.7 Use case scenario untuk tambah artikel

Tambah artikel	
<i>Objective</i>	Aktor tambah artikel
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman artikel 2. Sistem menampilkan halaman artikel yang berisi menu tambah 3. Aktor memilih menu tambah artikel 4. Sistem menampilkan <i>form</i> tambah artikel yang berisi kolom tanggal artikel, judul artikel, isi artikel, dan foto 5. Aktor mengisi <i>form</i> tambah artikel secara lengkap dan memerintahkan sistem untuk memproses 6. Sistem menampilkan pesan "Data Tersimpan" dan sistem menampilkan artikel yang sudah ditambahkan dalam bentuk tabel artikel yang berisi tanggal artikel, judul artikel, isi artikel, dan foto
<i>Alternative Flows</i>	5a. Jika kolom pada <i>form</i> tambah artikel belum diisi maka sistem akan memunculkan pesan " <i>Field is required</i> " pada kolom <i>form</i> tersebut
<i>Postconditions</i>	Aktor berhasil menambahkan artikel dan menampilkan daftar artikel yang sudah dibuat

Tabel 4.8 Use case scenario untuk ubah artikel

Ubah artikel	
<i>Objective</i>	Aktor ubah data artikel kecuali gambar
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman artikel
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar artikel yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih artikel yang akan diubah dan memilih menu ubah artikel 3. Sistem menampilkan <i>form</i> ubah artikel yang terdiri dari kolom tanggal artikel, judul artikel, isi artikel



	<ol style="list-style-type: none"> 4. Aktor melakukan perubahan data artikel secara lengkap dan memerintahkan sistem untuk memproses 5. Sistem menampilkan pesan "Data berhasil diubah" dan menampilkan artikel yang sudah diubah pada tabel yang berisi tanggal artikel, judul artikel, isi artikel, dan foto
<i>Alternative Flows</i>	4a. Jika aktor tidak mengisi bagian form secara tidak lengkap (terdapat kolom yang belum terisi), maka sistem akan memunculkan pesan "Field is required"
<i>Postconditions</i>	Aktor berhasil mengubah artikel yang sudah dipilih

Tabel 4.9 Use case scenario untuk hapus artikel

Hapus artikel	
<i>Objective</i>	Aktor hapus artikel
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman artikel
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar artikel yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih artikel yang akan dihapus 3. Sistem akan menampilkan sebuah pesan "Apakah anda yakin menghapus artikel ini" dan setelah itu sistem memunculkan pilihan "Batalkan" atau "Ya, saya yakin" 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	3a. Jika aktor memilih "Batalkan" maka artikel tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus artikel yang sudah dipilih

Tabel 4.10 Use case scenario untuk tambah pengumuman

Tambah pengumuman	
<i>Objective</i>	Aktor tambah pengumuman
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman pengumuman 2. Sistem menampilkan halaman pengumuman yang berisi menu tambah

	<ol style="list-style-type: none"> 3. Aktor memilih menu tambah pengumuman 4. Sistem menampilkan <i>form</i> tambah pengumuman yang berisi kolom tanggal pengumuman, judul pengumuman, isi pengumuman, dan foto 5. Aktor mengisi <i>form</i> tambah pengumuman secara lengkap dan memerintahkan sistem untuk memproses 6. Sistem menampilkan pesan “Data Tersimpan” dan sistem menampilkan pengumuman yang sudah ditambahkan dalam bentuk tabel artikel yang berisi tanggal pengumuman, judul pengumuman, isi pengumuman, dan foto
<i>Alternative Flows</i>	5a. Jika ada salah satu kolom pada <i>form</i> tambah pengumuman belum diisi maka sistem akan memunculkan pesan “ <i>Field is required</i> ” pada kolom <i>form</i> tersebut
<i>Postconditions</i>	Aktor berhasil menambahkan pengumuman dan menampilkan daftar pengumuman yang sudah dibuat

Tabel 4.11 Use case scenario untuk ubah pengumuman

Ubah pengumuman	
<i>Objective</i>	Aktor ubah data pengumuman kecuali gambar
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman pengumuman
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar pengumuman yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih pengumuman yang akan diubah dan memilih menu ubah pengumuman 3. Sistem menampilkan <i>form</i> ubah pengumuman yang terdiri dari kolom tanggal pengumuman, judul pengumuman, isi pengumuman 4. Aktor melakukan perubahan data pengumuman secara lengkap dan memerintahkan sistem untuk memproses 5. Sistem menampilkan pesan “Data berhasil diubah” dan menampilkan pengumuman yang sudah diubah pada tabel yang berisi tanggal pengumuman, judul pengumuman, isi pengumuman, dan foto
<i>Preconditions</i>	4a. Jika aktor tidak mengisi bagian form secara tidak lengkap (terdapat kolom yang belum terisi), maka

	sistem akan memunculkan pesan “Field is required”
<i>Preconditions</i>	Aktor berhasil mengubah pengumuman yang sudah dipilih

Tabel 4.12 Use case scenario untuk hapus pengumuman

Hapus pengumuman	
<i>Objective</i>	Aktor hapus pengumuman
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman pengumuman
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar pengumuman yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih pengumuman yang akan dihapus 3. Sistem akan menampilkan sebuah pesan “Apakah anda yakin menghapus pengumuman ini” dan setelah itu sistem memunculkan pilihan “Batalkan” atau “Ya, saya yakin” 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	3a. Jika aktor memilih “Batalkan” maka pengumuman tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus pengumuman yang sudah dipilih

Tabel 4.13 Use case scenario untuk tambah foto

Tambah foto	
<i>Objective</i>	tambah foto
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman foto 2. Sistem menampilkan <i>form</i> tambah foto yang terdiri dari nama foto dan <i>upload</i> foto 3. Aktor mengisi <i>form</i> foto secara lengkap dan memerintahkan sistem untuk memproses 4. Sistem menampilkan foto yang sudah ditambahkan dalam bentuk tabel
<i>Alternative Flows</i>	3a. Jika ada salah satu kolom pada <i>form</i> tambah foto belum diisi maka sistem akan memunculkan pesan “Field is required” pada kolom <i>form</i> tersebut



<i>Postconditions</i>	Aktor berhasil menambahkan foto dan menampilkan daftar foto yang sudah dibuat
-----------------------	---

Tabel 4.14 Use case scenario untuk hapus foto

Hapus foto	
<i>Objective</i>	Aktor hapus foto
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman foto
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar foto yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih foto yang akan dihapus 3. Sistem akan menampilkan sebuah pesan “Apakah anda yakin menghapus foto ini” dan setelah itu sistem memunculkan pilihan “Batalkan” atau “Ya, saya yakin” 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	4a. Jika Aktor memilih “Batalkan” maka foto tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus foto yang sudah dipilih

Tabel 4.15 Use case scenario untuk tambah kegiatan Insidental

Tambah kegiatan insidental	
<i>Objective</i>	Aktor tambah kegiatan insidental
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman kegiatan insidental 2. Sistem menampilkan halaman kegiatan insidental yang berisi menu tambah kegiatan insidental dan daftar kegiatan insidental yang sudah ditambahkan yang terdiri dari tanggal kegiatan, judul kegiatan, isi kegiatan, dan foto 3. Aktor memilih menu tambah kegiatan insidental 4. Sistem menampilkan <i>form</i> tambah kegiatan insidental yang berisi kolom tanggal kegiatan insidental, judul insidental, isi kegiatan insidental, dan foto

	<p>5. Aktor mengisi <i>form</i> tambah kegiatan insidental secara lengkap dan memerintahkan sistem untuk memproses</p> <p>6. Sistem menampilkan pesan “Data Tersimpan” dan sistem menampilkan kegiatan insidental yang sudah ditambahkan dalam bentuk tabel kegiatan insidental yang berisi tanggal insidental, judul insidental, isi kegiatan insidental, dan foto</p>
<i>Alternative Flows</i>	5a. Jika ada salah satu kolom pada <i>form</i> tambah kegiatan insidental belum diisi maka sistem akan memunculkan pesan “ <i>Field is required</i> ” pada kolom <i>form</i> tersebut
<i>Postconditions</i>	Aktor berhasil menambahkan kegiatan insidental dan menampilkan daftar kegiatan insidental yang sudah dibuat

Tabel 4.16 Use case scenario untuk ubah kegiatan insidental

Ubah kegiatan insidental	
<i>Objective</i>	Aktor ubah data kegiatan insidental kecuali gambar
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman kegiatan insidental
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar kegiatan insidental yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih kegiatan insidental yang akan diubah dan memilih menu ubah kegiatan insidental 3. Sistem menampilkan <i>form</i> ubah kegiatan insidental yang terdiri dari kolom tanggal kegiatan insidental, judul kegiatan insidental, isi kegiatan insidental 4. Aktor melakukan perubahan data kegiatan insidental secara lengkap dan memerintahkan sistem untuk memproses 5. Sistem menampilkan pesan “Data berhasil diubah” dan menampilkan kegiatan insidental yang sudah diubah pada tabel yang berisi tanggal kegiatan insidental, judul kegiatan insidental, isi kegiatan insidental, dan foto
<i>Alternative Flows</i>	4a. Jika aktor tidak mengisi bagian <i>form</i> secara tidak lengkap (terdapat <i>field</i> yang belum terisi), maka sistem akan memunculkan pesan “ <i>Field is required</i> ”

<i>Postconditions</i>	Aktor berhasil mengubah kegiatan insidental yang sudah dipilih
-----------------------	--

Tabel 4.17 Use case scenario untuk hapus kegiatan insidental

Hapus kegiatan insidental	
<i>Objective</i>	Aktor hapus kegiatan insidental
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman kegiatan insidental
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar kegiatan insidental yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih kegiatan insidental yang akan dihapus 3. Sistem akan menampilkan sebuah pesan “Apakah anda yakin menghapus kegiatan insidental ini” dan setelah itu sistem memunculkan pilihan “Batalkan” atau “Ya, saya yakin” 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	3a. Jika aktor memilih “Batalkan” maka kegiatan insidental tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus kegiatan insidental yang sudah dipilih

Tabel 4.18 Use case scenario untuk tambah kegiatan rutin

Tambah kegiatan rutin	
<i>Objective</i>	Aktor tambah kegiatan rutin
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman kegiatan rutin 2. Sistem menampilkan halaman kegiatan rutin yang berisi menu tambah 3. Aktor memilih menu tambah kegiatan rutin 4. Sistem menampilkan <i>form</i> tambah kegiatan rutin yang berisi kolom tanggal kegiatan rutin, judul rutin, dan isi kegiatan 5. Aktor mengisi <i>form</i> tambah kegiatan rutin secara lengkap dan memerintahkan sistem untuk memproses



	6. Sistem menampilkan pesan “Data Tersimpan” dan sistem menampilkan rutin yang sudah ditambahkan dalam bentuk tabel kegiatan rutin yang berisi tanggal rutin, judul rutin, isi kegiatan , dan foto
<i>Alternative Flows</i>	5a. Jika ada salah satu kolom pada <i>form</i> tambah kegiatan rutin belum diisi maka sistem akan memunculkan pesan “ <i>Field is required</i> ” pada kolom <i>form</i> tersebut
<i>Postconditions</i>	Aktor berhasil menambahkan kegiatan rutin dan menampilkan daftar kegiatan insidental yang sudah dibuat

Tabel 4.19 Use case scenario untuk ubah kegiatan rutin

Ubah kegiatan rutin	
<i>Objective</i>	Aktor ubah data kegiatan rutin
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman kegiatan rutin
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar kegiatan rutin yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih kegiatan rutin yang akan diubah dan memilih menu ubah kegiatan rutin 3. Sistem menampilkan <i>form</i> ubah kegiatan rutin yang terdiri dari kolom tanggal kegiatan rutin, judul kegiatan rutin, isi kegiatan rutin 4. Aktor melakukan perubahan data kegiatan rutin secara lengkap dan memerintahkan sistem untuk memproses 5. Sistem menampilkan pesan “Data berhasil diubah” dan menampilkan kegiatan rutin yang sudah diubah pada tabel yang berisi tanggal kegiatan rutin, judul kegiatan rutin, isi kegiatan rutin, dan foto
<i>Alternative Flows</i>	4a. Jika aktor tidak mengisi bagian form secara tidak lengkap (terdapat kolom yang belum terisi), maka sistem akan memunculkan pesan “ <i>Field is required</i> ”
<i>Postconditions</i>	Aktor berhasil mengubah kegiatan rutin yang sudah dipilih

Tabel 4.20 Use case scenario untuk hapus kegiatan rutin

Hapus kegiatan rutin	
<i>Objective</i>	Aktor hapus kegiatan rutin
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman kegiatan rutin
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar rutin yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih artikel yang akan dihapus 3. Sistem akan menampilkan sebuah pesan “Apakah anda yakin menghapus rutin ini” dan setelah itu sistem memunculkan pilihan “Batalkan” atau “Ya, saya yakin” 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	3a. Jika aktor memilih “Batalkan” maka kegiatan rutin tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus kegiatan rutin yang sudah dipilih

Tabel 4.21 Use case scenario untuk tambah Imam Salat

Tambah imam salat	
<i>Objective</i>	Aktor tambah imam salat
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman jadwal imam salat 2. Sistem menampilkan halaman jadwal imam salat yang berisi menu tambah 3. Aktor memilih menu tambah jadwal imam salat 4. Sistem menampilkan <i>form</i> tambah jadwal imam salat yang terdiri dari tanggal, imam Subuh, Zuhur, imam Asar, imam Magrib, imam Isya 5. Aktor mengisi <i>form</i> tambah jadwal imam salat secara lengkap dan memerintahkan sistem untuk memproses 6. Sistem menampilkan pesan “Data Tersimpan” dan sistem menampilkan jadwal imam salat yang sudah ditambahkan dalam bentuk tabel artikel yang berisi tanggal, imam Subuh, Zuhur, imam Asar, imam Magrib, imam Isya

<i>Alternative Flows</i>	5a. Jika ada salah satu kolom pada <i>form</i> tambah jadwal imam salat belum diisi maka sistem akan memunculkan pesan “ <i>Field is required</i> ” pada kolom <i>form</i> tersebut
<i>Postconditions</i>	Aktor berhasil menambahkan jadwal imam salat dan menampilkan daftar jadwal imam salat yang sudah dibuat

Tabel 4.22 Use case scenario untuk ubah imam salat

Ubah imam salat	
<i>Objective</i>	Aktor ubah jadwal imam salat
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman jadwal imam salat
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar jadwal imam salat yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih jadwal imam salat yang akan diubah dan memilih menu ubah jadwal imam salat 3. Sistem menampilkan <i>form</i> jadwal ubah imam salat yang terdiri dari kolom tanggal imam, imam Subuh, Zuhur, imam Asar, imam Magrib, imam Isya 4. Aktor melakukan perubahan data jadwal imam salat secara lengkap dan memerintahkan sistem untuk memproses 5. Sistem menampilkan pesan “Data berhasil diubah” dan menampilkan jadwal imam salat yang sudah diubah pada tabel yang berisi tanggal tanggal imam, imam Subuh, Zuhur, imam Asar, imam Magrib, imam Isya
<i>Alternative Flows</i>	4a. Jika aktor tidak mengisi bagian form secara tidak lengkap (terdapat kolom yang belum terisi), maka sistem akan memunculkan pesan “ <i>Field is required</i> ”
<i>Postconditions</i>	Aktor berhasil mengubah jadwal imam salat yang sudah dipilih

Tabel 4.23 Use case scenario untuk hapus imam salat

Hapus jadwal imam salat	
<i>Objective</i>	Aktor hapus jadwal imam salat
<i>Actors</i>	Anggota Bidang Ubudiyah

<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman imam sholat
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar imam sholat yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih imam sholat yang akan dihapus 3. Sistem akan menampilkan sebuah pesan “Apakah anda yakin menghapus imam sholat ini” dan setelah itu sistem memunculkan pilihan “Batalkan” atau “Ya, saya yakin” 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	3a. Jika aktor memilih “Batalkan” maka imam salat tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus imam sholat yang sudah dipilih

Tabel 4.24 Use case scenario untuk tambah jadwal salat

Tambah jadwal salat	
<i>Objective</i>	Aktor tambah jadwal salat
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman jadwal salat 2. Sistem menampilkan halaman jadwal salat yang berisi menu tambah 3. Aktor memilih menu tambah jadwal salat 4. Sistem menampilkan <i>form</i> tambah jadwal salat yang terdiri dari tanggal, jadwal Subuh, Zuhur, jadwal Asar, jadwal Magrib, jadwal Isya 5. Aktor mengisi <i>form</i> tambah jadwal salat secara lengkap dan memerintahkan sistem untuk memproses 6. Sistem menampilkan pesan “Data Tersimpan” dan sistem menampilkan jadwal salat yang sudah ditambahkan dalam bentuk tabel jadwal salat yang berisi tanggal, jadwal Subuh, jadwal Zuhur, jadwal Asar, jadwal Magrib, jadwal Isya
<i>Alternative Flows</i>	5a. Jika ada salah satu kolom pada <i>form</i> tambah jadwal salat belum diisi maka sistem akan memunculkan pesan “ <i>Field is required</i> ” pada kolom <i>form</i> tersebut
<i>Postconditions</i>	Aktor berhasil menambahkan jadwal salat dan menampilkan daftar jadwal salat yang sudah dibuat

Tabel 4.25 Use case scenario untuk ubah jadwal salat

Ubah jadwal salat	
<i>Objective</i>	Aktor ubah jadwal salat
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman jadwal salat
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar jadwal salat yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih jadwal salat yang akan diubah dan memilih menu ubah jadwal salat 3. Sistem menampilkan <i>form</i> ubah jadwal salat yang terdiri dari kolom tanggal jadwal salat, jadwal Subuh, jadwal Zuhur, imam Asar, imam Magrib, imam Isya 4. Aktor melakukan perubahan data imam salat secara lengkap dan memerintahkan sistem untuk memproses 5. Sistem menampilkan pesan "Data berhasil diubah" dan menampilkan artikel yang sudah diubah pada tabel yang berisi tanggal tanggal imam, imam Subuh, Zuhur, imam Asar, imam Magrib, imam Isya
<i>Alternative Flows</i>	4a. Jika aktor tidak mengisi bagian form secara tidak lengkap (terdapat kolom yang belum terisi), maka sistem akan memunculkan pesan " <i>Field is required</i> "
<i>Postconditions</i>	Aktor berhasil mengubah imam salat yang sudah dipilih.

Tabel 4.26 Use case scenario untuk hapus jadwal salat

Hapus jadwal salat	
<i>Objective</i>	Aktor hapus jadwal salat
<i>Actors</i>	Anggota Bidang Ubudiyah
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman jadwal salat
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar jadwal sholat yang sudah ditambahkan oleh Anggota Bidang Ubudiyah 2. Aktor memilih jadwal sholat yang akan dihapus 3. Sistem akan menampilkan sebuah pesan "Apakah anda yakin menghapus jadwal sholat ini" dan setelah itu sistem memunculkan pilihan "Batalkan" atau "Ya, saya yakin" 4. Aktor memerintahkan sistem untuk memproses

<i>Alternative Flows</i>	4a. Jika aktor memilih “Batalkan” maka jadwal salat tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus jadwal sholat yang sudah dipilih

Tabel 4.27 Use case scenario untuk tambah laporan keuangan harian

Tambah laporan keuangan harian	
<i>Objective</i>	Aktor tambah laporan keuangan harian
<i>Actors</i>	Bendahara
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman laporan keuangan harian 2. Sistem menampilkan halaman laporan keuangan harian yang berisi menu tambah transaksi keuangan harian 3. Aktor memilih menu tambah laporan keuangan harian 4. Sistem menampilkan <i>form</i> tambah laporan keuangan harian yang berisi dari keterangan transaksi, tanggal transaksi, pemasukan, pengeluaran, serta kelompok transaksi 5. Aktor mengisi <i>form</i> tambah laporan keuangan harian secara lengkap dan memerintahkan sistem untuk memproses 6. Sistem menampilkan pesan “Data Tersimpan” dan sistem menampilkan laporan keuangan harian yang sudah ditambahkan dalam bentuk tabel laporan keuangan harian yang berisi keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran
<i>Alternative Flows</i>	5a. Jika ada salah satu kolom pada <i>form</i> tambah artikel belum diisi maka sistem akan memunculkan pesan “ <i>Field is required</i> ” pada kolom <i>form</i> tersebut
<i>Postconditions</i>	Aktor berhasil menambahkan laporan keuangan harian dan menampilkan daftar laporan keuangan harian yang sudah dibuat

Tabel 4.28 Use case scenario untuk ubah laporan keuangan harian

Ubah laporan keuangan harian	
<i>Objective</i>	Aktor ubah laporan keuangan harian
<i>Actors</i>	Bendahara

<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman laporan keuangan harian
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar laporan keuangan harian yang sudah ditambahkan oleh Bendahara 2. Aktor memilih laporan keuangan harian yang akan diubah dan memilih menu ubah laporan keuangan harian 3. Sistem menampilkan <i>form</i> ubah laporan keuangan harian yang terdiri dari kolom keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran 4. Aktor melakukan perubahan data laporan keuangan harian secara lengkap dan memerintahkan sistem untuk memproses 5. Sistem menampilkan pesan “Data berhasil diubah” dan menampilkan laporan keuangan harian yang sudah diubah pada tabel yang berisi keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran
<i>Alternative Flows</i>	4a. Jika aktor tidak mengisi bagian form secara tidak lengkap (terdapat kolom yang belum terisi), maka sistem akan memunculkan pesan “ <i>Field is required</i> ”
<i>Postconditions</i>	Aktor berhasil mengubah laporan keuangan harian yang sudah dipilih

Tabel 4.29 Use case scenario untuk hapus laporan keuangan harian

Hapus laporan keuangan harian	
<i>Objective</i>	Aktor hapus laporan keuangan harian
<i>Actors</i>	Bendahara
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman laporan keuangan harian
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar laporan keuangan harian yang sudah ditambahkan oleh Bendahara 2. Aktor memilih laporan keuangan harian yang akan dihapus 3. Sistem akan menampilkan sebuah pesan “Apakah anda yakin menghapus laporan keuangan harian ini” dan setelah itu sistem memunculkan pilihan “Batalkan” atau “Ya, saya yakin” 4. Aktor memerintahkan sistem untuk memproses

<i>Alternative Flows</i>	3a. Jika aktor memilih “Batalkan” maka laporan keuangan harian tidak akan terhapus
<i>Postconditions</i>	Aktor berhasil menghapus laporan keuangan harian yang sudah dipilih

Tabel 4.30 Use case scenario untuk cetak laporan keuangan harian

Mencetak laporan keuangan harian	
<i>Objective</i>	Aktor cetak laporan keuangan harian
<i>Actors</i>	Bendahara
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman laporan keuangan harian
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman laporan keuangan harian yang berisi menu cetak laporan keuangan harian 2. Aktor memilih menu cetak laporan keuangan harian 3. Sistem menampilkan laporan keuangan dan memunculkan pilihan “Simpan” atau “Batalkan” 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	3a. Jika aktor memilih “Batalkan” maka sistem tidak mencetak laporan keuangan harian
<i>Postconditions</i>	Aktor berhasil mencetak laporan keuangan harian

Tabel 4.31 Use case scenario untuk cetak laporan keuangan bulanan

Mencetak laporan keuangan bulanan	
<i>Objective</i>	cetak laporan keuangan bulanan
<i>Actors</i>	Bendahara
<i>Preconditions</i>	Aktor sudah masuk ke dalam halaman laporan keuangan bulanan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman laporan keuangan bulanan yang berisi menu cetak laporan keuangan harian 2. Aktor memilih menu cetak laporan keuangan bulanan 3. Sistem menampilkan laporan keuangan dan memunculkan pilihan “Simpan” atau “Batalkan” 4. Aktor memerintahkan sistem untuk memproses
<i>Alternative Flows</i>	3a. Jika aktor memilih “Batalkan” maka sistem tidak mencetak laporan keuangan bulanan

<i>Postconditions</i>	Aktor berhasil mencetak laporan keuangan bulanan
-----------------------	--

Tabel 4.32 Use case scenario untuk lihat detail kegiatan rutin

Lihat detail kegiatan rutin	
<i>Objective</i>	Aktor lihat daftar kegiatan rutin
<i>Actors</i>	Pengunjung
<i>Preconditions</i>	Aktor sudah berada di halaman utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu kegiatan 2. Aktor memilih menu kegiatan rutin 3. Sistem menampilkan daftar kegiatan rutin
<i>Alternative Flows</i>	3a. Jika Anggota Bidang Ubudiyah belum menambahkan kegiatan rutin, maka aktor tidak bisa melihat daftar kegiatan rutin
<i>Postconditions</i>	Aktor berhasil melihat detail kegiatan rutin

Tabel 4.33 Use case scenario untuk lihat Foto

Lihat foto	
<i>Objective</i>	Aktor lihat foto
<i>Actors</i>	Pengunjung
<i>Preconditions</i>	Aktor sudah berada di halaman utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu foto 2. Sistem menampilkan daftar foto
<i>Alternative Flows</i>	2a. Jika Anggota Bidang Ubudiyah belum menambahkan foto, maka aktor tidak bisa melihat daftar foto
<i>Postconditions</i>	Aktor berhasil melihat daftar foto

Tabel 4.34 Use case scenario untuk lihat daftar dan detail artikel

Lihat daftar dan detail artikel	
<i>Objective</i>	Aktor lihat daftar dan detail artikel
<i>Actors</i>	Pengunjung
<i>Preconditions</i>	Aktor sudah berada di halaman utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar artikel 2. Aktor memilih artikel yang akan dilihat 3. Aktor memerintahkan sistem untuk memproses

	4. Sistem menampilkan detail artikel yang dipilih oleh aktor
<i>Alternative Flows</i>	1a. Jika Anggota Bidang Ubudiyah belum menambahkan artikel, maka aktor tidak bisa melihat detail artikel
<i>Postconditions</i>	Aktor berhasil melihat detail artikel

Tabel 4.35 Use case scenario untuk lihat daftar dan detail kegiatan insidental

Lihat detail kegiatan insidental	
<i>Objective</i>	Aktor melihat detail kegiatan insidental
<i>Actors</i>	Pengunjung
<i>Preconditions</i>	Aktor sudah berada di halaman utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu kegiatan 2. Aktor memilih menu kegiatan insidental 3. Sistem menampilkan daftar kegiatan insidental
<i>Alternative Flows</i>	3a. Jika Anggota Bidang Ubudiyah belum menambahkan kegiatan insidental, maka aktor tidak bisa melihat detail kegiatan insidental
<i>Postconditions</i>	Aktor berhasil melihat detail kegiatan insidental

Tabel 4.36 Use case scenario untuk lihat daftar dan detail pengumuman

Lihat detail pengumuman	
<i>Objective</i>	Aktor lihat daftar dan detail pengumuman
<i>Actors</i>	Pengunjung
<i>Preconditions</i>	Aktor sudah berada di halaman utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar pengumuman 2. Aktor memilih pengumuman yang akan dilihat 3. Aktor memerintahkan sistem untuk memproses 4. Sistem menampilkan detail prngumuman yang dipilih oleh aktor
<i>Alternative Flows</i>	1a. Jika Anggota Bidang Ubudiyah belum menambahkan pengumuman, maka aktor tidak bisa daftar pengumuman
<i>Postconditions</i>	Aktor berhasil melihat daftar pengumuman

Tabel 4.37 Use case scenario untuk lihat daftar Imam Salat

Lihat daftar imam salat	
<i>Objective</i>	Aktor lihat daftar imam salat
<i>Actors</i>	Pengunjung
<i>Preconditions</i>	Aktor sudah berada di halaman utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu imam salat 2. Sistem menampilkan daftar imam salat dalam bentuk tabel yang terdiri dari tanggal, imam Subuh, imam Zuhur, imam Asar, imam Maghrib, imam Isya
<i>Alternative Flows</i>	2a. Jika Anggota Bidang Ubudiyah belum menambahkan imam salat, maka sistem menampilkan data tabel imam salat yang kosong
<i>Postconditions</i>	Aktor berhasil melihat daftar imam salat

Tabel 4.38 Use case scenario untuk lihat daftar jadwal salat

Lihat daftar jadwal salat	
<i>Objective</i>	Aktor lihat daftar jadwal salat
<i>Actors</i>	Pengunjung
<i>Preconditions</i>	Aktor sudah berada di halaman utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu jadwal salat 2. Sistem menampilkan daftar imam salat dalam bentuk tabel yang terdiri dari tanggal, jadwal Subuh, jadwal Zuhur, jadwal Asar, jadwal Maghrib, jadwal Isya
<i>Alternative Flows</i>	2a. Jika Anggota Bidang Ubudiyah belum menambahkan jadwal salat, maka sistem menampilkan data tabel jadwal salat yang kosong
<i>Postconditions</i>	Aktor berhasil melihat daftar jadwal salat

Tabel 4.39 Use case scenario untuk lihat laporan keuangan harian

Lihat laporan keuangan harian	
<i>Objective</i>	Aktor Lihat laporan keuangan harian pada durasi waktu yang ditentukan
<i>Actors</i>	Ketua Pengurus Masjid dan Bendahara
<i>Preconditions</i>	Aktor sudah berada pada halaman utama

<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman laporan keuangan harian 2. Sistem menampilkan halaman laporan keuangan harian berisi filter keuangan yang terdiri dari kolom pilihan tanggal awal dan tanggal akhir data keuangan yang akan dilihat 3. Aktor memilih rentang tanggal, bulan, dan tahun laporan keuangan yang ingin ditampilkan serta memerintahkan sistem untuk memproses 4. Sistem menampilkan detail laporan keuangan harian terdiri dari tabel keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran berdasarkan rentang tanggal yang dipilih oleh aktor
<i>Alternative Flows</i>	<ol style="list-style-type: none"> 3a. Jika transaksi keuangan pada rentang tanggal yang dipilih aktor belum ditambahkan oleh Bendahara, maka sistem menampilkan pesan “Maaf data keuangan yang anda cari tidak ada”
<i>Postconditions</i>	Aktor berhasil lihat laporan keuangan harian

Tabel 4.40 Use case scenario untuk lihat laporan keuangan bulanan

Lihat laporan keuangan bulanan	
<i>Objective</i>	Aktor lihat laporan keuangan bulanan
<i>Actors</i>	Ketua Pengurus Masjid dan Bendahara
<i>Preconditions</i>	Aktor sudah berada pada halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memerintahkan sistem untuk menampilkan halaman laporan keuangan bulanan 2. Sistem menampilkan halaman laporan keuangan bulanan yang berisi tabel keterangan transaksi, kelompok transaksi jumlah pemasukan, dan jumlah pengeluaran 3. Aktor memilih rentang bulan dan tahun laporan keuangan yang ingin ditampilkan serta memerintahkan sistem untuk memproses 4. Sistem menampilkan detail laporan keuangan bulanan terdiri dari tabel keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran berdasarkan rentang bulan dan tahun yang dipilih oleh aktor
<i>Alternative Flows</i>	<ol style="list-style-type: none"> 2a. Jika Bendahara belum menambahkan laporan keuangan harian, maka sistem menampilkan data pada tabel laporan keuangan bulanan kosong

Postconditions

Aktor berhasil lihat laporan keuangan bulanan



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Perancangan merupakan sebuah tahapan yang dilakukan setelah semua kebutuhan sudah didapatkan dari proses analisis kebutuhan. Perancangan sistem yang dilakukan terdiri dari perancangan arsitektur, perancangan data, perancangan komponen, dan perancangan antarmuka.

5.1.1 Perancangan Arsitektur

Perancangan arsitektur yang digunakan untuk memodelkan perangkat lunak menggunakan *sequence diagram* dan *class diagram*. *Sequence diagram* digunakan untuk menggambarkan secara rinci alur pertukaran pesan antar objek yang berada di dalam sistem, sedangkan *class diagram* digunakan untuk menggambarkan hubungan antar kelas.

5.1.1.1 Pemodelan *Sequence Diagram*

Pemodelan *sequence diagram* digunakan untuk mengidentifikasi objek-objek yang saling berinteraksi berdasarkan *use case scenario* yang sudah dijelaskan. Pemodelan ini menggunakan tiga *use case scenario* yang sudah dibuat antara lain Tambah kegiatan insidental, Ubah kegiatan insidental, Tambah laporan keuangan harian.

1. *Sequence diagram* Tambah kegiatan insidental

Sequence diagram untuk menambah kegiatan insidental dapat dilihat pada Gambar 5.1. Aktor yang terlibat pada *sequence diagram* tambah kegiatan insidental adalah Anggota Bidang Ubudiyah. Objek-objek yang terdapat dalam *sequence diagram* ini adalah dua objek *boundary*, yaitu “Tambah_kegiatan_insidental” dan “Kegiatan_insidental_v”, satu objek *controller*, yaitu “Kegiatan_insidental_controller”, dan satu objek *entity*, yaitu “Kegiatan”.

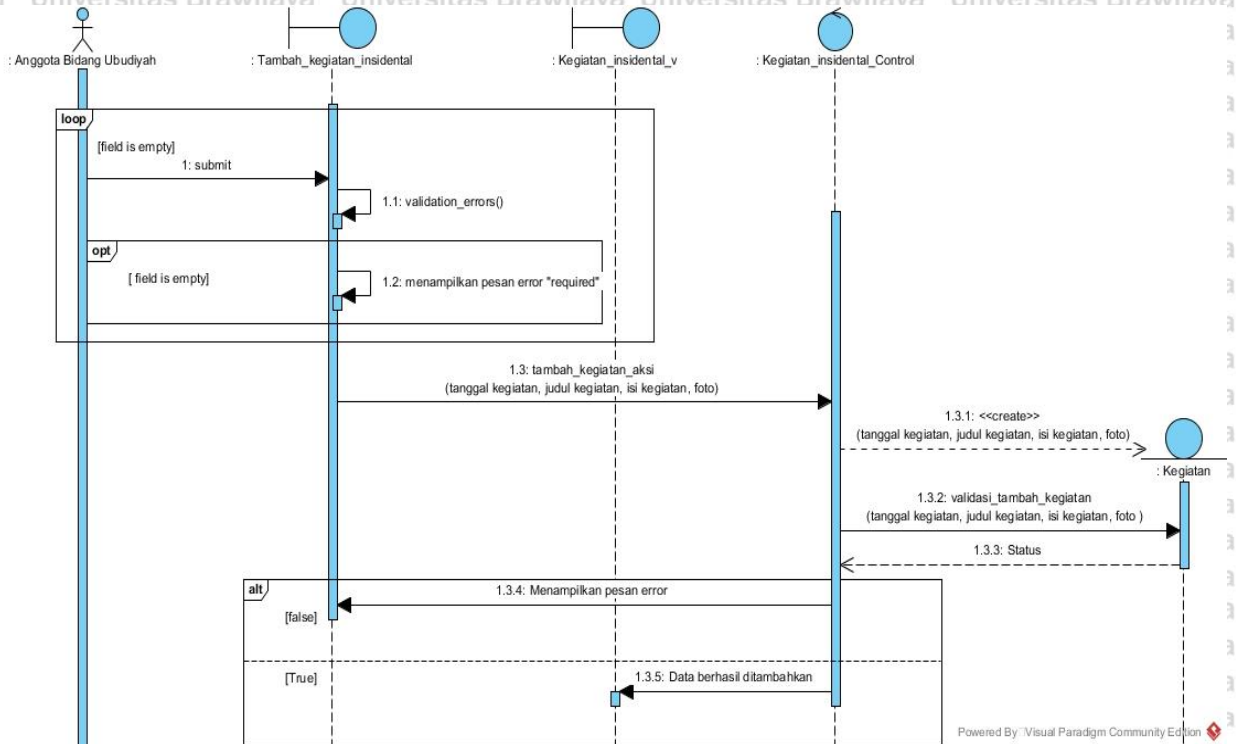
2. *Sequence diagram* Ubah kegiatan insidental

Sequence diagram untuk menambah kegiatan insidental dapat dilihat pada Gambar 5.2. Aktor yang terlibat pada *sequence diagram* tambah kegiatan insidental adalah Anggota Bidang Ubudiyah. Objek-objek yang terdapat dalam *sequence diagram* ini adalah dua objek *boundary*, yaitu “Ubah_kegiatan_insidental” dan “Kegiatan_insidental_v”, satu objek *controller*, yaitu “Kegiatan_insidental_controller”, dan satu objek *entity*, yaitu “Kegiatan”.

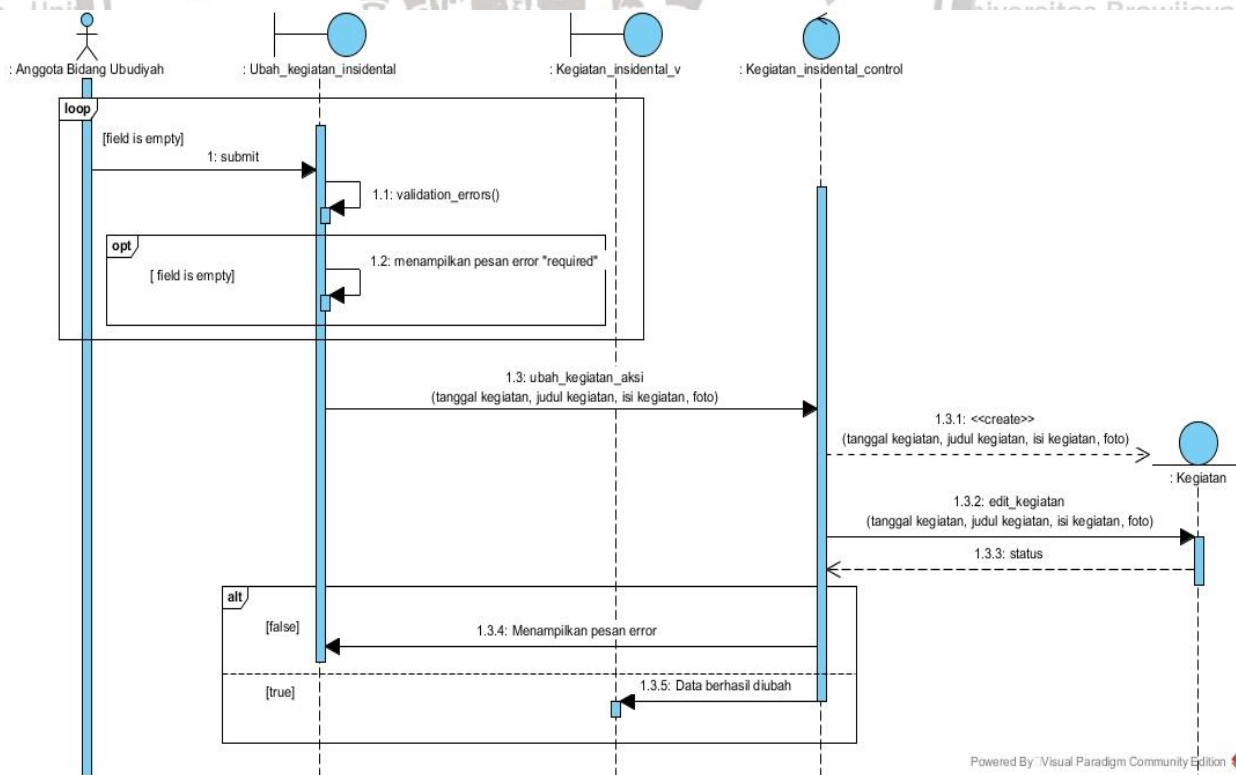
3. *Sequence diagram* Tambah laporan keuangan harian

Sequence diagram untuk menambah laporan keuangan harian dapat dilihat pada Gambar 5.3. Aktor yang terlibat pada *sequence diagram* tambah laporan keuangan harian adalah Bendahara. Objek-objek yang terdapat dalam *sequence diagram* ini adalah dua objek *boundary*, yaitu “Tambah_keuangan_harian” dan “Keuangan_harian_v” satu objek *controller*,

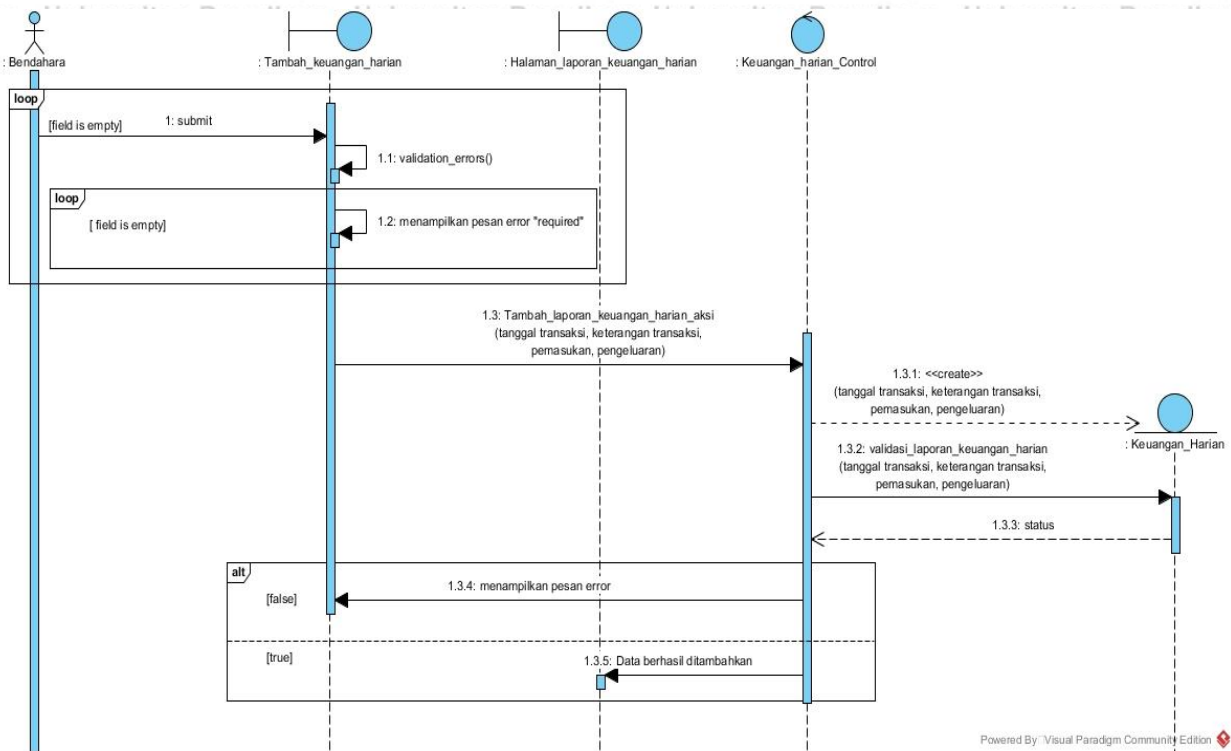
yaitu "Tambah_Laporan_Keuangan_Harian_Controller", dan satu objek *entity*, yaitu "Keuangan_Harian".



Gambar 5.1 Sequence diagram Tambah kegiatan insidental



Gambar 5.2 Sequence diagram Ubah kegiatan insidental

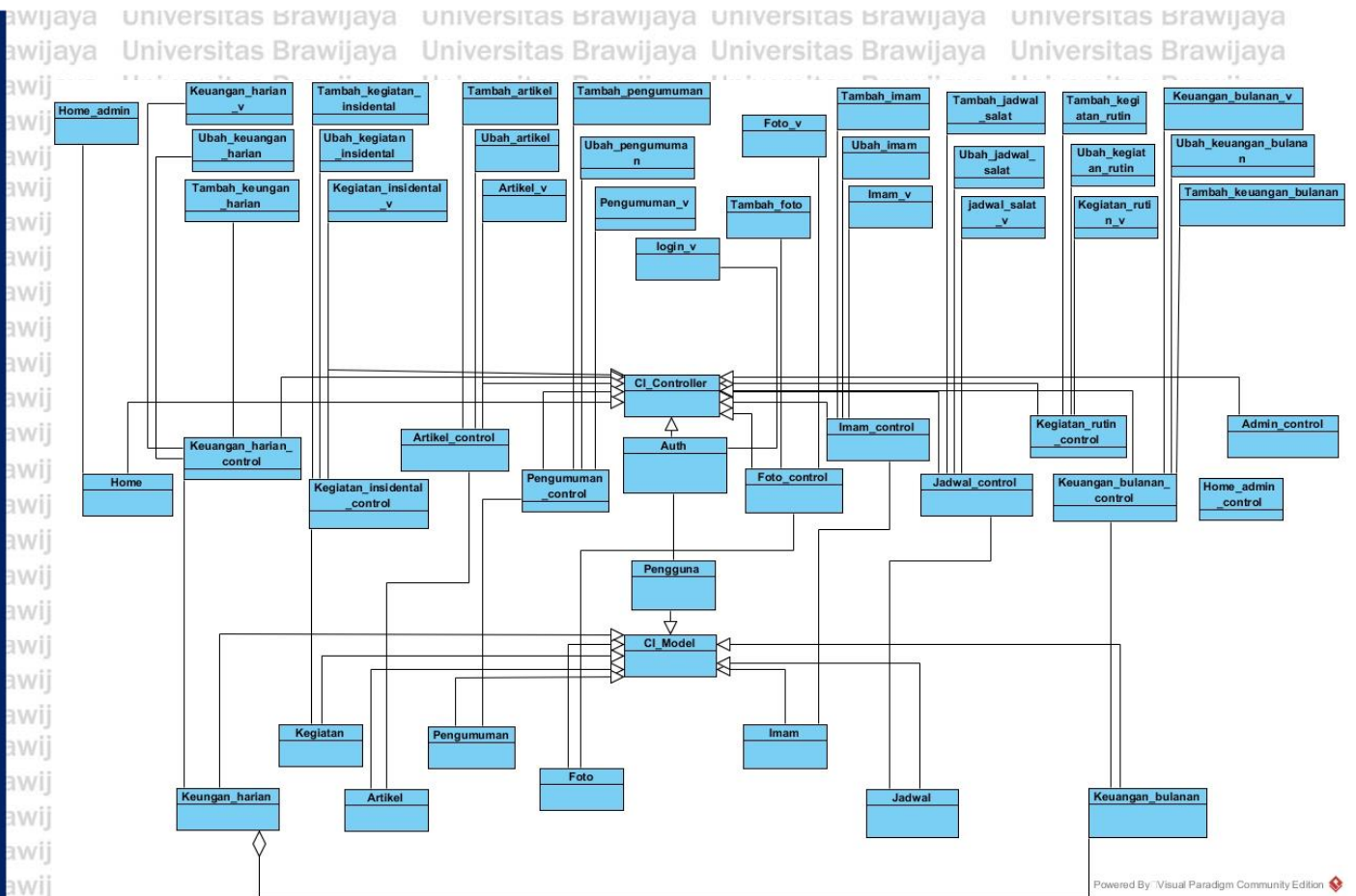


Gambar 5.3 Sequence diagram Tambah laporan keuangan harian

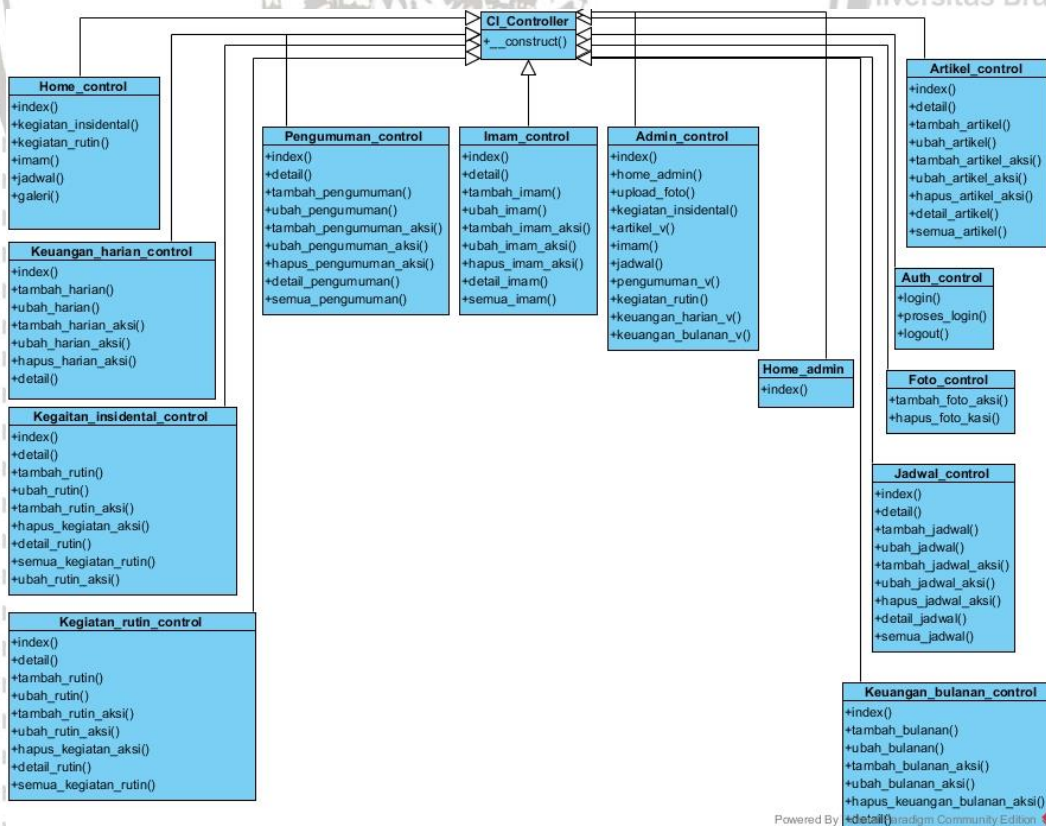
5.1.1.2 Class Diagram

Pemodelan *class* diagram dilakukan untuk menggambarkan relasi antar kelas pada perangkat lunak. *Class* diagram umum yang diilustrasikan ada Gambar 5.4, terdiri dari *CI_Model*, *CI_Controller*, 9 klas *entity*, 12 klas *controller*, dan 27 klas *boundary*.

Pemodelan *class* diagram yang diilustrasikan pada Gambar 5.4 menggambarkan relasi antara klas boudary, klas controller, dan klas entity. Klas *CI_Model* memiliki 9 klas turunan, yaitu Keuangan harian, Artikel, Pengumuman, Foto, Kegiatan, Imam, Jadwal, dan Keuangan bulanan. Klas *CI_Controller* memiliki 13 klas turunan, yaitu *Home*, *Keuangan_harian_control*, *Kegiatan_insidental_control*, *Artikel_control*, *Pengumuman_control*, *Auth*, *Foto_control*, *Imam_control*, *Jadwal_control*, *Kegiatan_rutin_control*, *Admin_control*, *Keuangan_bulanan_control*, *Home_admin_control*. Class boundary memiliki 28 klas, yaitu *Home_admin*, *Keuangan_harian_v*, *Tambah_keuangan_harian*, *Ubah_keuangan_harian*, *Kegiatan_insidental_v*, *Tambah_kegiatan_insidental*, *Ubah_kegiatan_insidental*, *Artikel_v*, *Tambah_artikel*, *Ubah_artikel*, *Pengumuman_v*, *Tambah_pengumuman*, *Ubah_pengumuman*, *Login_v*, *Tambah_foto*, *Foto_v*, *Tambah_imam*, *Ubah_imam*, *Imam_v*, *Tambah_jadwal_salat*, *Ubah_jadwal_salat*, *Jadwal_v*, *Tambah_kegiatan_rutin*, *Ubah_kegiatan_rutin*, *Kegiatan_rutin_v*, *Keuangan_bulanan_v*, *Tambah_keuangan_bulanan*, *Ubah_Keuangan_bulanan*.

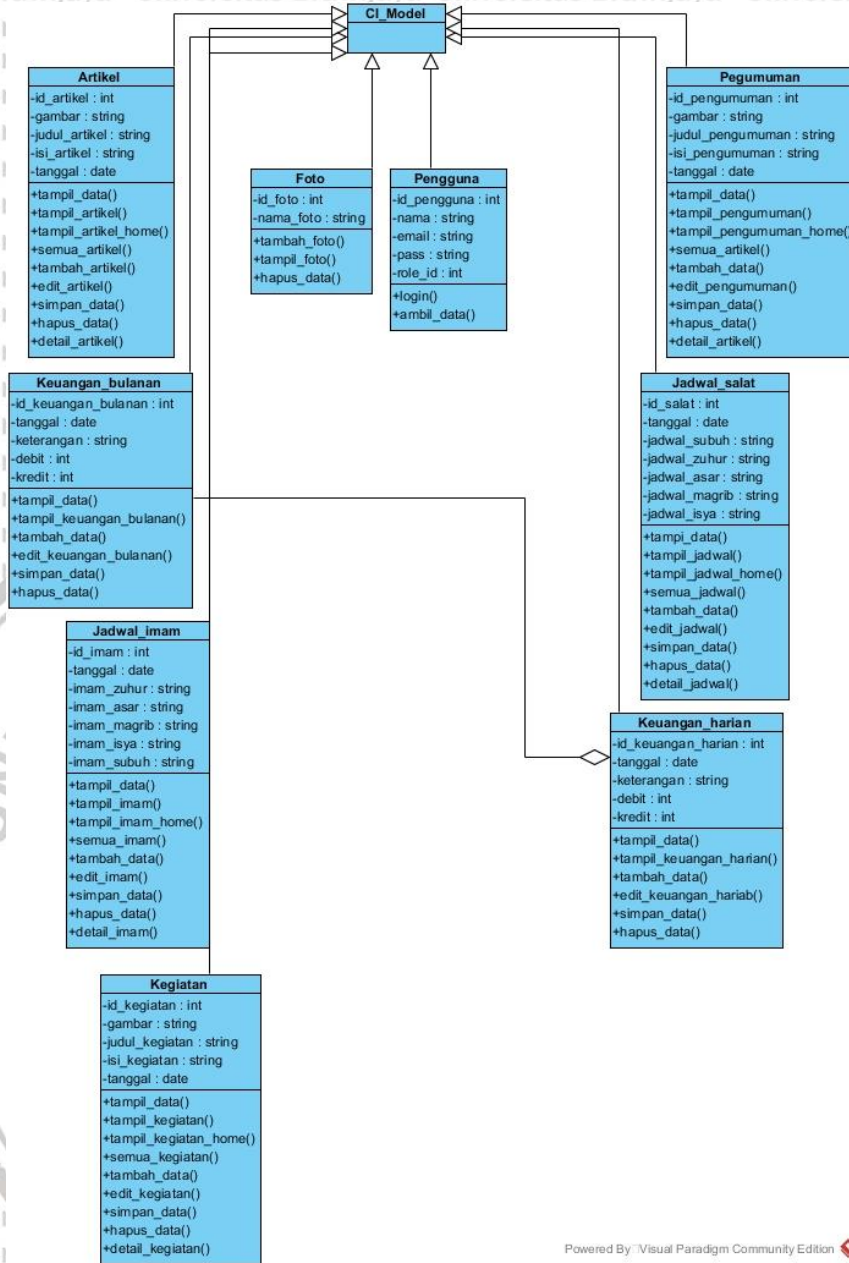


Gambar 5.4 Class diagram umum



Powered By Visual Paradigm Community Edition

Gambar 5.5 Class diagram controller

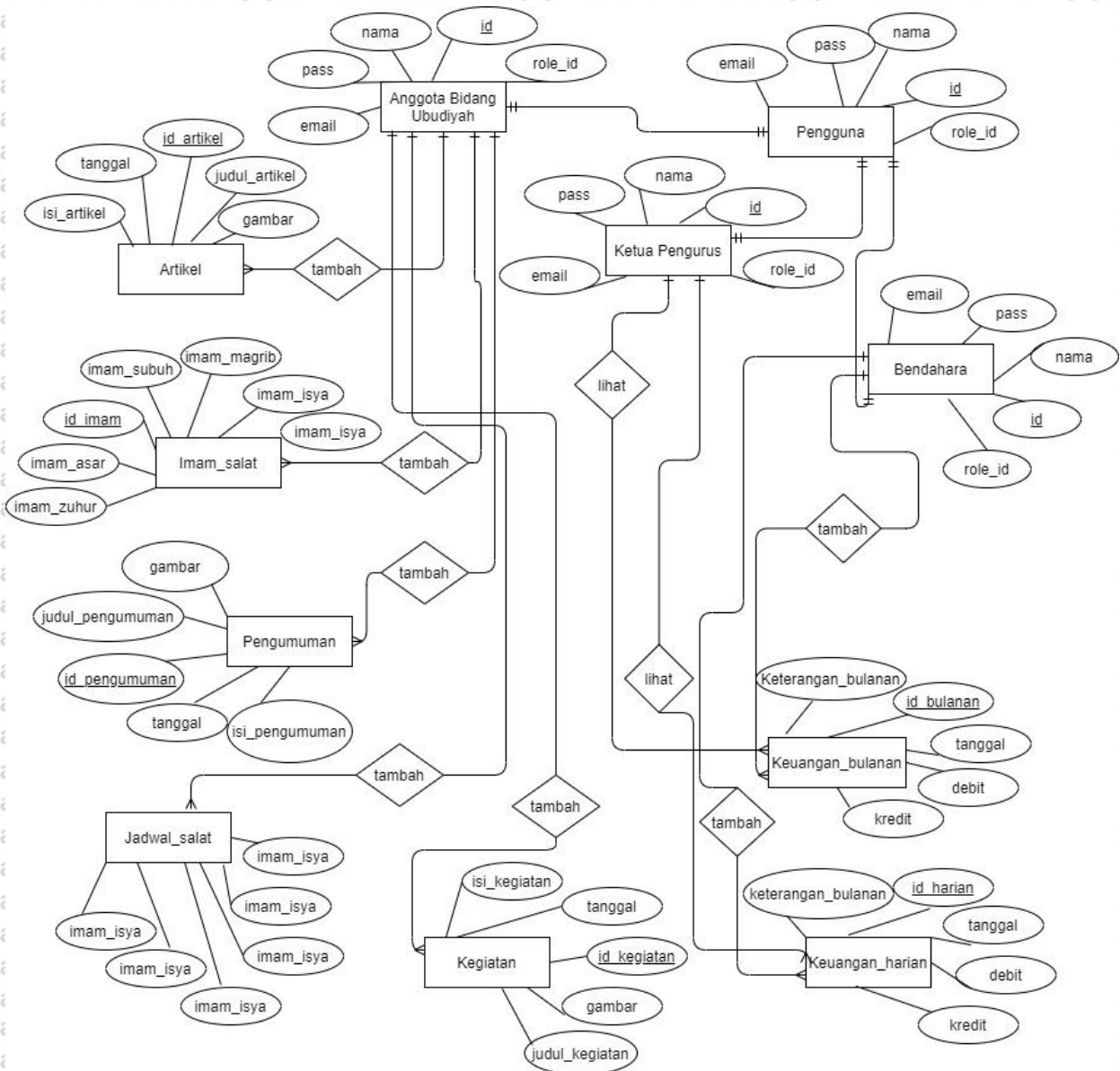


Gambar 5.6 Class diagram entity

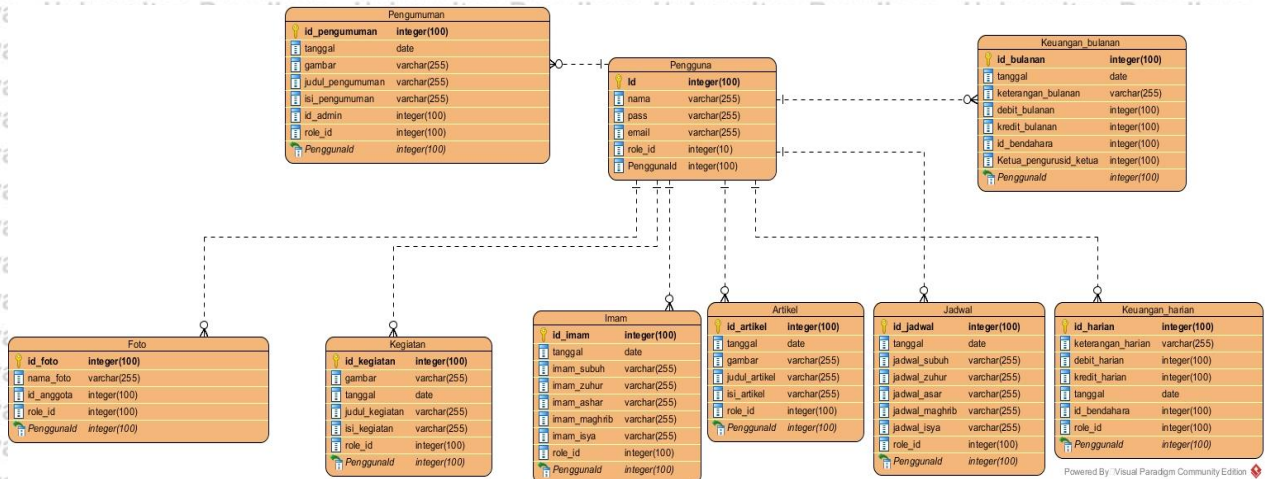
5.1.2 Perancangan Data

Perancangan data dilakukan untuk memodelkan data secara konseptual dan fisik. Pemodelan menggunakan *Conceptual Data Model* (CDM) dapat direpresentasikan dalam bentuk *Entity Relationship Diagram* (ERD) dan *Physical Data Model* (PDM) dapat direpresentasikan dalam bentuk skema database yang akan digunakan. Pemodelan CDM akan ditransformasikan untuk dapat memodelkan entitas yang ada pada sistem, pada entitas tersebut memiliki atribut dan relasi antara entitas. Pemodelan CDM digunakan untuk menjadi dasar dalam pemodelan PDM, serta pada pemodelan PDM merupakan pemodelan

lebih lengkap dikarenakan sudah mempunyai *primary key*, *foreign key*, nama tabel, nama kolom, serta tipe data dari setiap kolom. Permodelan CDM dapat dilihat pada Gambar 5.7 dan permodelan PDM dapat dilihat pada Gambar 5.8.



Gambar 5.7 Perancangan CDM (Conceptual Data Model)



Gambar 5.8 Perancangan PDM (Physical Data Model)

5.1.3 Perancangan Komponen

Perancangan komponen digunakan untuk menjelaskan secara rinci algoritme dalam sebuah *method* dari suatu kelas. Algoritme dalam sebuah *method* dituliskan dalam bentuk *pseudocode*. Perancangan komponen terdiri dari tiga method, yaitu `tambah_kegiatan_aksi()`, `ubah_kegiatan_aksi()`, `tambah_harian_aksi()`.

5.1.3.1 Perancangan Komponen Klas Kegiatan_Controller

Nama *method*: `tambah_kegiatan_aksi`

Algoritme:

Tabel 5.1 Pseudocode method `tambah_kegiatan_aksi`

No	Pseudocode
1	Start
2	Inisialisasi nilai <code>judul_kegiatan</code>
3	Inisialisasi nilai <code>isi_kegiatan</code>
4	Inisialisasi nilai <code>tanggal_kegiatan</code>
5	If (<code>form_validation == false</code>) then
6	Memanggil method <code>tambah_kegiatan()</code>
7	Else
8	Inisialisasi nilai variabel <code>foto = nilai field</code>
9	<code>foto_kegiatan</code>
10	If (<code>upload_foto == true</code>) then
11	upload foto berdasarkan data <code>'file_name'</code>
12	inisialisasi <code>\$judul_kegiatan = input</code> berdasarkan data <code>'judul_kegiatan'</code>
13	inisialisasi <code>\$isi_kegiatan = input</code> berdasarkan data <code>'isi_kegiatan'</code>
14	



```

15      inisialisasi $tanggal_kegiatan = input
16      berdasarkan data 'tanggal_kegiatan'
17      inisialisasi $data = array (
18          nilai variabel 'judul_kegiatan' =>
19          $judul_kegiatan
20          nilai variabel 'isi_kegiatan' =>
21          $isi_kegiatan
22          nilai variabel 'tanggal_kegiatan' =>
23          $tanggal_kegiatan
24      memanggil method tambah_data untuk menyimpan
25      nilai kedalam tabel kegiatan_model
26      menampilkan pesan 'Data Tersimpan'
27      memanggil method kegiatan()
28      else
29          $error
30      End if
31  End if
32  End

```

5.1.3.2 Perancangan Komponen Klas Kegiatan_Controller

Nama *method*: ubah_kegiatan_aksi

Algoritme:

Tabel 5.2 Pseudocode method ubah_kegiatan_aksi

No	Pseudocode
1	Start
2	Inisialisasi nilai judul_kegiatan = input berdasarkan data 'judul_kegiatan'
3	Inisialisasi nilai isi_kegiatan = input berdasarkan data 'isi_kegiatan'
4	Inisialisasi nilai tanggal_kegiatan = input berdasarkan data 'tanggal_kegiatan'
5	Inisialisasi nilai gambar_kegiatan = input berdasarkan data 'file_name'
6	Inisialisasi \$data = array (
7	Nilai judul_kegiatan = \$judul_kegiatan
8	Nilai isi_kegiatan = \$isi_kegiatan
9	Nilai tanggal_kegiatan = \$tanggal_kegiatan
10	Where data array berdasarkan 'id_kegiatan'
11	Memanggil method simpan_data() berdasarkan 'id_kegiatan' ke dalam tabel 'kegiatan_model'

15	Menampilkan pesan 'Data Berhasil Diubah'
16	Memanggil method kegiatan()
17	End

5.1.3.3 Perancangan Komponen Klas Keuangan_harian_Controller

Nama *method*: tambah_harian_aksi

Algoritme:

Tabel 5.3 Pseudocode Method tambah_harian_aksi

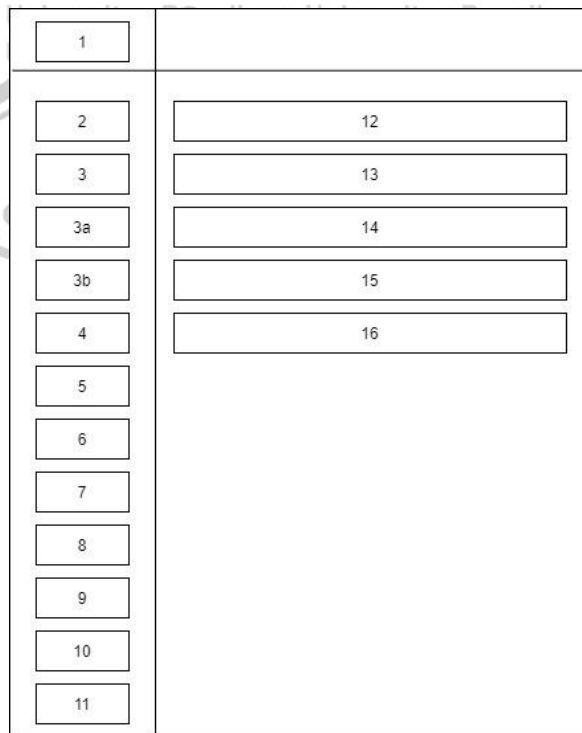
No	Pseudocode
1	Start
2	Inisialisasi nilai keterangan_harian
3	Inisialisasi nilai debit_harian
4	Inisialisasi nilai kredit_harian
5	Inisialisasi nilai tanggal_harian
6	If (form_validation == false)then
7	Memanggil method tambah_harian()
8	Else
9	Inisialisasi nilai variabel \$keterangan_harian =
10	berdasarkan input data 'keterangan_harian'
11	Inisialisasi nilai variabel \$debit_harian =
12	berdasarkan input data 'debit_harian'
13	Inisialisasi nilai variabel \$kredit_harian =
14	berdasarkan input data 'kredit_harian'
15	Inisialisasi nilai variabel \$tanggal_harian =
16	berdasarkan input data 'tanggal_jurnal'
17	Inisialisasi \$data = array (
18	Nilai 'keterangan_harian' => \$keterangan_harian
19	Nilai 'debit_harian' => \$debit_harian
20	Nilai 'kredit_harian' => \$kredit_harian
21	Nilai 'tanggal_harian' => \$tanggal_harian
22	Memanggil method tambah_data() untuk menyimpan nilai \$data ke dalam tabel keuangan_harian
23	Menampilkan 'Data Tersimpan'
24	Memanggil method kegiatan()
25	End if
26	End

5.1.4 Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk membuat desain tampilan antarmuka untuk pengguna. Perancangan antarmuka pada penelitian ini terdiri dari tiga sampel yang diambil dari fungsionalitas utama. Tiga sampel yang digunakan terdiri dari perancangan antarmuka halaman Tambah kegiatan insidental, Ubah kegiatan insidental, dan Tambah keuangan harian.

5.1.4.1 Perancangan Antarmuka Halaman Tambah Kegiatan Insidental

Gambar 5.9 merupakan perancangan antarmuka dari halaman tambah kegiatan insidental. Penjelasan dari rancangan tersebut dapat dilihat pada Tabel 5.4.



Gambar 5.9 Perancangan Antarmuka Halaman Tambah Kegiatan Insidental

Tabel 5.4 Penjelasan Bagian Antarmuka Halaman Tambah Kegiatan Insidental

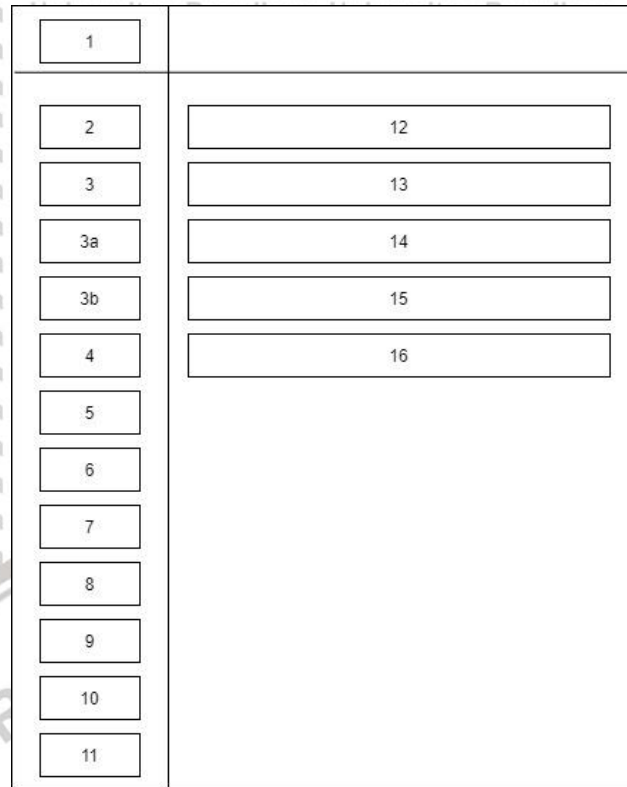
No	Nama Objek	Tipe	Keterangan
1	Logo sistem	<i>Button Logo</i>	Berfungsi untuk untuk menampilkan logo sistem berupa <i>text</i>
2	Pilihan menu beranda	Menu	Menu untuk menampilkan halaman beranda

3	Pilihan menu keuangan	Menu Tab	Menu untuk menampilkan submenu yang dimiliki menu keuangan
3a	Menu jurnal umum	Submenu Tab	Menu untuk masuk ke halaman jurnal umum
3b	Menu buku besar	Submenu Tab	Menu untuk masuk ke halaman buku besar
4	Pilihan menu <i>upload</i> foto	Menu	Menu untuk masuk ke halaman <i>upload</i> foto
5	Pilihan menu kegiatan incidental	Menu	Menu untuk masuk ke halaman kegiatan incidental
6	Pilihan menu kegiatan rutin	Menu	Menu untuk masuk ke halaman kegiatan rutin
7	Pilihan menu pengumuman	Menu	Menu untuk masuk ke halaman pengumuman
8	Pilihan menu artikel	Menu	Menu untuk masuk ke halaman artikel
9	Pilihan menu imam sholat	Menu	Menu untuk masuk ke halaman imam sholat
10	Pilihan menu <i>logout</i>	Menu	Tombol untuk keluar dari sistem

11	Pilihan menu	<i>Button</i>	Berfungsi untuk untuk mengubah ukuran menu tab
12	Judul kegiatan	<i>Text</i>	Berfungsi untuk memasukkan nama judul kegiatan
13	Isi kegiatan	<i>Text</i>	Berfungsi untuk memasukkan isi kegiatan
14	Gambar	<i>Image</i>	Berfungsi untuk menampilkan foto acara/poster kegiatan
15	Tanggal kegiatan	<i>Date</i>	Berfungsi untuk memasukkan tanggal kegiatan
16	Tombol simpan	<i>Button</i>	Tombol untuk menyimpan data yang sudah dimasukkan

5.1.4.2 Perancangan Antarmuka Halaman Ubah Kegiatan Insidental

Gambar 5.10 merupakan perancangan antarmuka dari halaman ubah kegiatan insidental. Penjelasan dari rancangan tersebut dapat dilihat pada Tabel 5.5.



Gambar 5.10 Perancangan Antarmuka Halaman Ubah Kegiatan

Tabel 5.5 Penjelasan Bagian Antarmuka Halaman Ubah Kegiatan Insidental

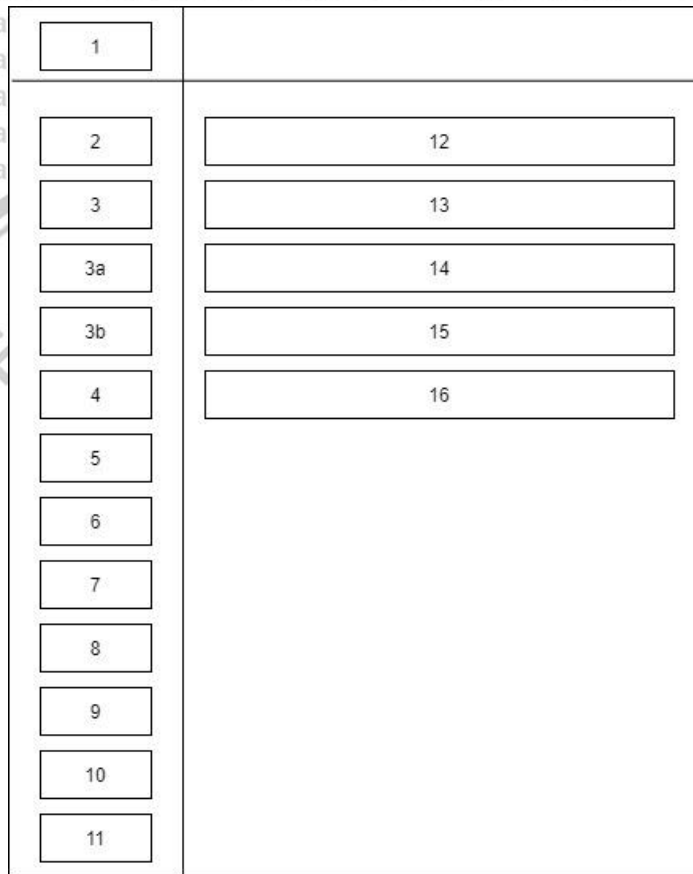
No	Nama Objek	Tipe	Keterangan
1	Logo sistem	<i>Button Logo</i>	Berfungsi untuk untuk menampilkan logo sistem berupa <i>text</i>
2	Pilihan menu beranda	Menu	Menu untuk menampilkan halaman beranda
3	Pilihan menu keuangan	Menu Tab	Menu untuk menampilkan submenu yang dimiliki menu keuangan
3a	Menu keuangan harian	<i>Submenu Tab</i>	Menu untuk masuk ke halaman keuangan harian

3b	Menu keuangan bulanan	<i>Submenu Tab</i>	Menu untuk masuk ke halaman keuangan_bulanan
4	Pilihan menu <i>upload</i> foto	Menu	Menu untuk masuk ke halaman <i>upload</i> foto
5	Pilihan menu kegiatan insidental	Menu	Menu untuk masuk ke halaman kegiatan insidental
6	Pilihan menu kegiatan rutin	Menu	Menu untuk masuk ke halaman kegiatan rutin
7	Pilihan menu pengumuman	Menu	Menu untuk masuk ke halaman pengumuman
8	Pilihan menu artikel	Menu	Menu untuk masuk ke halaman artikel
9	Pilihan menu imam sholat	Menu	Menu untuk masuk ke halaman imam sholat
10	Pilihan menu <i>logout</i>	Menu	Tombol untuk keluar dari sistem
11	Pilihan menu	<i>Button</i>	Berfungsi untuk mengubah ukuran menu tab
12	Judul kegiatan	<i>Text</i>	Berfungsi untuk mengubah nama judul kegiatan
13	Isi kegiatan	<i>Text</i>	Berfungsi untuk mengubah isi kegiatan
14	Gambar	<i>Image</i>	Berfungsi untuk menampilkan foto acara/poster kegiatan
15	Tanggal kegiatan	<i>Date</i>	Berfungsi untuk mengubah tanggal kegiatan

16	Tombol simpan	Button	Tombol untuk menyimpan data yang sudah dimasukkan
----	---------------	--------	---

5.1.4.3 Perancangan Antarmuka Halaman Tambah Keuangan Harian

Gambar 5.11 merupakan perancangan antarmuka dari halaman tambah keuangan harian. Penjelasan dari rancangan tersebut dapat dilihat pada Tabel 5.6.



Gambar 5.11 Perancangan Antarmuka Halaman Tambah Keuangan harian

Tabel 5.6 Penjelasan Bagian Antarmuka Halaman Tambah Keuangan harian

No	Nama Objek	Tipe	Keterangan
1	Logo sistem	Button Logo	Berfungsi untuk untuk menampilkan logo sistem berupa text

2	Pilihan menu beranda	Menu	Menu untuk menampilkan halaman beranda
3	Pilihan menu keuangan	Menu Tab	Menu untuk menampilkan submenu yang dimiliki menu keuangan
3a	Menu keuangan harian	Submenu Tab	Menu untuk masuk ke halaman keuangan harian
3b	Menu keuangan bulanan	Submenu Tab	Menu untuk masuk ke halaman keuangan bulanan
4	Pilihan menu <i>upload</i> foto	Menu	Menu untuk masuk ke halaman <i>upload</i> foto
5	Pilihan menu kegiatan insidental	Menu	Menu untuk masuk ke halaman kegiatan insidental
6	Pilihan menu kegiatan rutin	Menu	Menu untuk masuk ke halaman kegiatan rutin
7	Pilihan menu pengumuman	Menu	Menu untuk masuk ke halaman pengumuman
8	Pilihan menu artikel	Menu	Menu untuk masuk ke halaman artikel
9	Pilihan menu imam sholat	Menu	Menu untuk masuk ke halaman imam sholat

10	Pilihan menu <i>logout</i>	Menu	Tombol untuk keluar dari sistem
11	Pilihan menu	<i>Button</i>	Berfungsi untuk untuk mengubah ukuran menu tab
12	Tanggal harian	<i>Date</i>	Berfungsi untuk memasukkan tanggal harian
13	Keterangan harian	<i>Text</i>	Berfungsi untuk memasukkan keterangan harian
14	Debit	<i>Number</i>	Berfungsi untuk memasukkan debit
15	Kredit	<i>number</i>	Berfungsi untuk memasukkan kredit
16	Tombol simpan	<i>Button</i>	Tombol untuk menyimpan data yang sudah dimasukkan

5.2 Implementasi

Implementasi merupakan sebuah tahapan yang dilakukan setelah semua proses perancangan sudah dilakukan. Perancangan sistem yang dilakukan terdiri dari implementasi kode program, implementasi basis data, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Tabel 5.7 merupakan spesifikasi perangkat keras yang digunakan dalam pengembangan sistem.

Tabel 5.7 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi Perangkat Keras
Laptop	Asus A455L
RAM	4GB
Hard disk	500GB

Tabel 5.8 merupakan spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem.

Tabel 5.8 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi Perangkat Lunak
Sistem Operasi	Windows 10
Editor Dokumentasi	Microsoft Word 2013
Editor Perancangan	Visual Paradigm dan draw.io
Editor Pemrograman	Sublime Text 3
Bahasa Pemrograman	HTML, PHP, CSS, Javacript
Framework	Codeigniter
DBMS	Xampp MySQL
Browser	Mozilla Firefox dan Google Chrome

5.2.2 Implementasi Kode Program

Implementasi kode program dilakukan berdasarkan hasil perancangan komponen yang sudah dibuat. Implementasi kode program menggunakan tiga sampel yang berasal dari perancangan komponen. Tiga sampel tersebut terdiri dari *method* `tambah_kegiatan_aksi()` dari kelas `Kegiatan_insidental_control`, *method* `ubah_kegiatan_aksi()` dari kelas `Kegiatan_insidental_control`, dan *method* `tambah_harian_aksi()` dari kelas `Kegiatan_harian_control`.

5.2.2.1 Implementasi *method* `tambah_kegiatan_aksi` dari kelas `Kegiatan_insidental_control`

Tabel 5.9 merupakan implementasi kode program dari *method* `tambah_kegiatan_aksi` dari kelas `Kegiatan_insidental_control`.

Tabel 5.9 Sourcecode *method* `tambah_kegiatan_aksi`

No	Sourcecode
1	<code>public function tambah_kegiatan_aksi(){</code>
2	<code> \$this->form_validation-</code>
3	<code> ->set_rules('judul_kegiatan', 'Judul', 'trim required');</code>
4	<code> \$this->form_validation-</code>
5	<code> ->set_rules('isi_kegiatan', 'Isi Kegiatan',</code>
6	<code> 'trim required');</code>
7	<code> \$this->form_validation-</code>
8	<code> ->set_rules('tanggal_kegiatan', 'Tanggal', 'trim required');</code>


```
9 // $this->form_validation-  
10 >set_rules('gambar_kegiatan','Gambar','trim|required');  
11  
12 if($this->form_validation->run()==FALSE){  
13     $this->tambah_kegiatan();  
14 }else{  
15     $config['upload_path'] = './uploads';  
16     $config['allowed_types'] = 'gif|jpg|png|jpeg|JPG';  
17     $config['max_size'] = '10240';  
18     $config['max_width'] = '8000';  
19     $config['max_height'] = '10000';  
20     $nama_foto = "foto_kegiatan-".time();  
21     $config['file_name']=$nama_foto;  
22  
24     $this->load->library('upload', $config);  
25  
26     if($this->upload->do_upload('gambar_kegiatan') == true) {  
27  
28         $post['gambar_kegiatan'] = $this->  
29         >upload->data('file_name');  
30         $judul_kegiatan = $this->  
31         >input->post('judul_kegiatan');  
32         $isi_kegiatan = $this->input->post('isi_kegiatan');  
33         $tanggal_kegiatan = $this->input->post('tanggal_kegiatan');  
34  
35         $data = array(  
36             'judul_kegiatan' => $judul_kegiatan,  
37             'isi_kegiatan' => $isi_kegiatan,  
38             'gambar' => $post['gambar_kegiatan'],  
39             'tanggal' => $tanggal_kegiatan,  
40         );  
41  
42         $this->kegiatan_model->tambah_data('kegiatan', $data);  
43  
44  
45  
46  
47
```

```

48     $this->session-
49     >set_flashdata('pesan','Data Tersimpan!');
50     redirect('admin/kegiatan');
51     } else {
52         $error;
53     }
}

```

5.2.2.2 Implementasi *method* ubah_kegiatan_aksi dari kelas Kegiatan_incidental_controller

Tabel 5.10 merupakan implementasi kode program dari *method* ubah_kegiatan dari *class* Kegiatan_control.

Tabel 5.10 Sourcecode *method* ubah_kegiatan_aksi

No	Sourcecode
1	public function edit_kegiatan_aksi(\$id){
2	
3	
4	\$this->upload->do_upload('gambar_kegiatan');
5	
6	\$jjudul_kegiatan = \$this->input-
7	>post('judul_kegiatan');
8	\$iisi_kegiatan = \$this->input-
9	>post('isi_kegiatan');
10	\$tanggal_kegiatan = \$this->input-
11	>post('tanggal_kegiatan');
12	\$post['gambar_kegiatan'] = \$this->upload-
13	>data('file_name');
14	\$data = array(
15	'judul_kegiatan' => \$jjudul_kegiatan,
16	'isi_kegiatan' =>
17	\$iisi_kegiatan,
18	'gambar' =>
19	\$post['gambar_kegiatan'],
20	'tanggal' =>
21	\$tanggal_kegiatan,
22);
23	\$where = array('id_kegiatan'=>\$id);
24	\$this->kegiatan_model-
25	>simpan_data(\$where,'kegiatan', \$data);

24	<code>\$this->session->set_flashdata('pesan', 'Data</code>
25	<code>Berhasil Diubah!');</code>
26	<code>redirect('Admin/kegiatan');</code>
	<code>}</code>

5.2.2.3 Implementasi *method* tambah_harian_aksi dari kelas *Keuangan_harian_control*

Tabel 5.11 merupakan implementasi kode program dari *method* tambah_harian_aksi dari kelas *Keuangan_harian_control*.

Tabel 5.11 Sourcecode *method* tambah_harian_aksi

No	Sourcecode
1	<code>\$this->form_validation->set_rules('keterangan_harian',</code>
2	<code>'Keterangan', 'trim required');</code>
3	<code>\$this->form_validation-</code>
4	<code>>set_rules('debit_harian', 'Debit harian',</code>
5	<code>'trim required');</code>
6	<code>\$this->form_validation-</code>
7	<code>>set_rules('kredit_harian', 'Kredit harian',</code>
8	<code>'trim required');</code>
9	<code>\$this->form_validation-</code>
10	<code>>set_rules('tanggal_harian', 'harian', 'trim required');</code>
11	<code>if(\$this->form_validation->run()==FALSE){</code>
12	<code>\$this->tambah_harian();</code>
13	<code>}else{</code>
14	<code>\$keterangan_harian</code>
15	<code>\$this->input->post('keterangan_harian');</code>
16	<code>\$debit_harian</code>
17	<code>\$this->input->post('debit_harian');</code>
18	<code>\$kredit_harian</code>
19	<code>\$this->input->post('kredit_harian');</code>
20	<code>\$tanggal_harian = \$this-</code>
21	<code>>input->post('tanggal_harian');</code>
22	<code>\$data = array(sitas Brawijaya Universitas Brawijaya</code>
23	<code>'keterangan_harian' =></code>
24	<code>\$keterangan_harian,</code>
25	<code>'debit_harian' =></code>
26	<code>\$debit_harian,</code>
27	<code>'tanggal' =></code>
28	<code>\$tanggal_harian,</code>

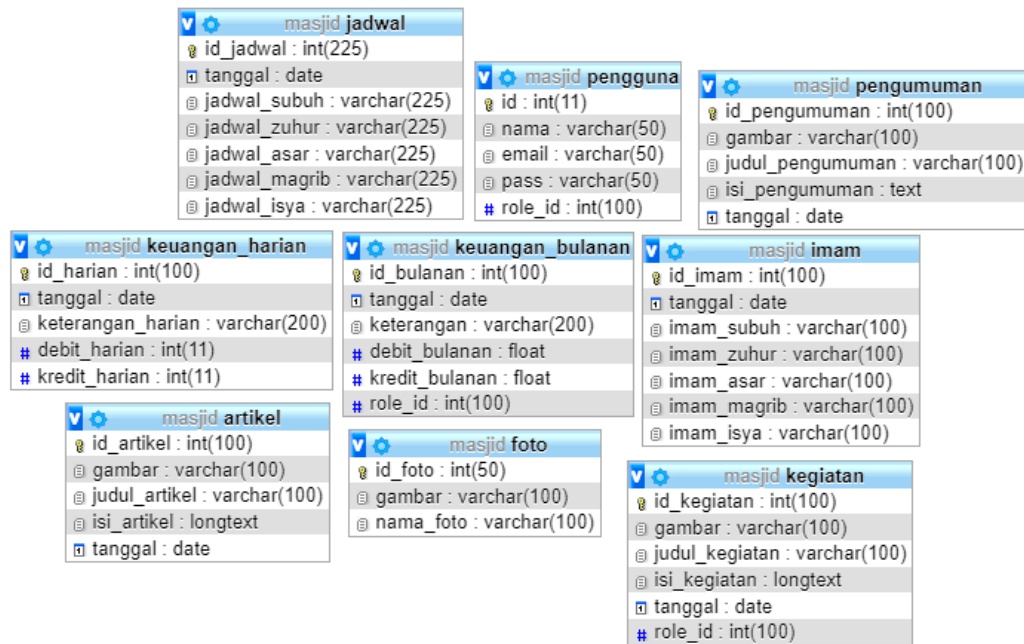
```

29         'kredit_harian' =>
30         $kredit_harian
31     );
32     $this->Keuangan_harian-
33     >tambah_data('keuangan_harian', $data);
34     $this->session->set_flashdata('pesan',
35     'Data berhasil ditambahkan!');
36     redirect('admin/Keuangan_harian_v');
37 }

```

5.2.3 Implementasi Data

Implementasi data dibuat berdasarkan perancangan data yang telah dibuat. Gambar 5.12 menggambarkan semua tabel dalam basis data sistem beserta kolom dan tipe data disetiap kolomnya.



Gambar 5.12 Implementasi Data

5.2.4 Implementasi Antarmuka

5.2.4.1 Implementasi Antarmuka Halaman Tambah Kegiatan Insidental

Gambar 5.26 merupakan hasil implementasi antarmuka halaman tambah kegiatan insidental.

The screenshot shows a web interface for 'MASJID AL GHIFARI'. On the left is a blue sidebar menu with options: Beranda, Keuangan, Upload Foto, Kegiatan Insidental, Kegiatan Rutin, Jadwal Sholat, Pengumuman, Artikel, Imam Sholat, and Logout. The main content area is titled 'TAMBAH KEGIATAN INSIDENTAL' and contains the following form elements:

- Judul Kegiatan:** A text input field.
- Isi Kegiatan:** A larger text area for content.
- Gambar:** A file upload section with a 'Pilih File' button and the text 'Tidak ada file yang dipilih'.
- Tanggal Kegiatan:** A date input field.
- Jenis Kegiatan:** A dropdown menu with 'Pilih' selected.
- Buttons:** A blue 'Simpan' button at the bottom right.

Gambar 5.13 Implementasi Antarmuka Halaman Tambah Kegiatan

5.2.4.2 Implementasi Antarmuka Halaman Ubah Kegiatan Insidental

Gambar 5.27 merupakan hasil implementasi antarmuka halaman ubah kegiatan insidental.

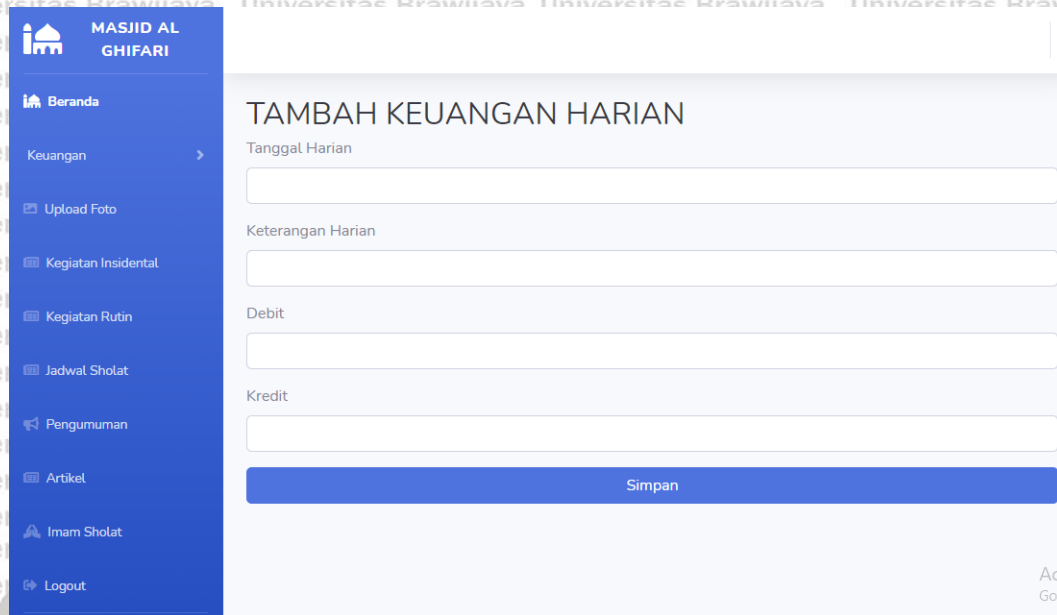
The screenshot shows the 'MASJID AL GHIFARI' web interface for editing an activity. The sidebar menu is identical to the previous screenshot. The main content area is titled 'EDIT KEGIATAN INSIDENTAL' and contains the following form elements:

- Judul Kegiatan:** A text input field containing 'sdsd'.
- Isi Kegiatan:** A text area containing 'dfdfdf'.
- Tanggal Kegiatan:** A date input field containing '2021-01-21'.
- Buttons:** A blue 'Simpan' button at the bottom right.

Gambar 5.14 Implementasi Antarmuka Halaman Ubah Kegiatan Insidental

5.2.4.3 Implementasi Antarmuka Halaman Tambah Keuangan Harian

Gambar 5.28 merupakan hasil implementasi antarmuka halaman tambah keuangan harian.



Gambar 5.15 Implementasi Antarmuka Halaman Tambah Keuangan Harian



BAB 6 PENGUJIAN

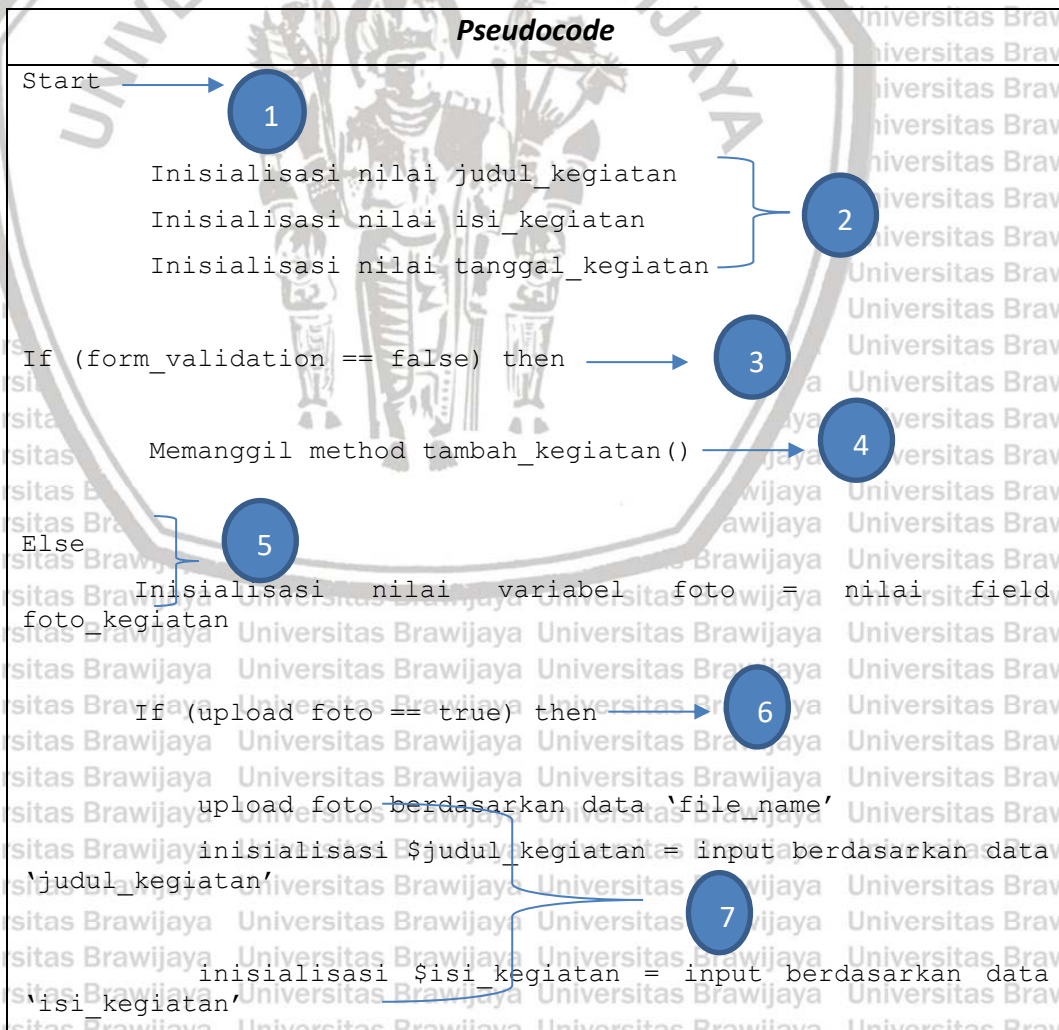
6.1 Pengujian Unit

Pengujian unit digunakan untuk menguji semua setiap unit yang ada pada sistem. Pengujian unit dilakukan menggunakan teknik pengujian *white box testing* dan metode *basis path testing*. Pengujian unit menggunakan klas *driver* untuk menguji *method* yang akan diuji. Pengujian unit menggunakan tiga *method* sampel sebagai bahan untuk melakukan pengujian yang berasal dari hasil perancangan komponen. Tiga *method* yang digunakan untuk pengujian terdiri dari `tambah_kegiatan_aksi()`, `ubah_kegiatan_aksi()`, dan `tambah_harian_aksi()`.

6.1.1 Pengujian unit *method* `tambah_kegiatan_insidental_aksi()` klas `Kegiatan_insidental_control`

Tabel 6.1 merupakan *pseudocode* dari *method* `tambah_kegiatan_aksi()` dari klas `Kegiatan_insidental_control`. Gambar 6.1 merupakan *flow graph* dari *method* `tambah_kegiatan_aksi()` dari klas `Kegiatan_insidental_control`.

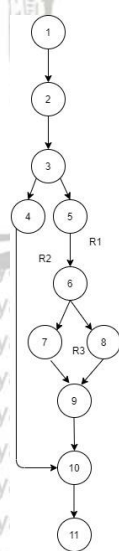
Tabel 6.1 Pseudocode method `tambah_kegiatan_aksi`



```

inialisasi $tanggal_kegiatan = input berdasarkan
data `tanggal_kegiatan`
inialisasi $data = array (
    nilai variabel `judul_kegiatan` => $judul_kegiatan
    nilai variabel `isi_kegiatan` => $isi_kegiatan
    nilai variabel `tanggal_kegiatan` =>
    $tanggal_kegiatan
)
memanggil method tambah_kegiatan_insidental untuk
menyimpan nilai kedalam tabel kegiatan
menampilkan pesan `Data Tersimpan`
redirect ke halaman Kegiatan_insidental_v
else
    $error
End if
End if
End
    
```

Flow Graph



Gambar 6.1 Flow Graph method tambah_kegiatan_aksi()

Cyclomatic Complexity (V(G))

- $V(G) = \text{jumlah region} = 3$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (12 - 11) + 2 = 3$
- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

Independent Path

- Jalur 1 = 1-2-3-4-10-11
- Jalur 2 = 1-2-5-6-7-9-10-11
- Jalur 3 = 1-2-3-5-6-8-9-10-11

Tabel 6.2 Kasus uji dan hasil uji *method* tambah_kegiatan_incidental_aksi

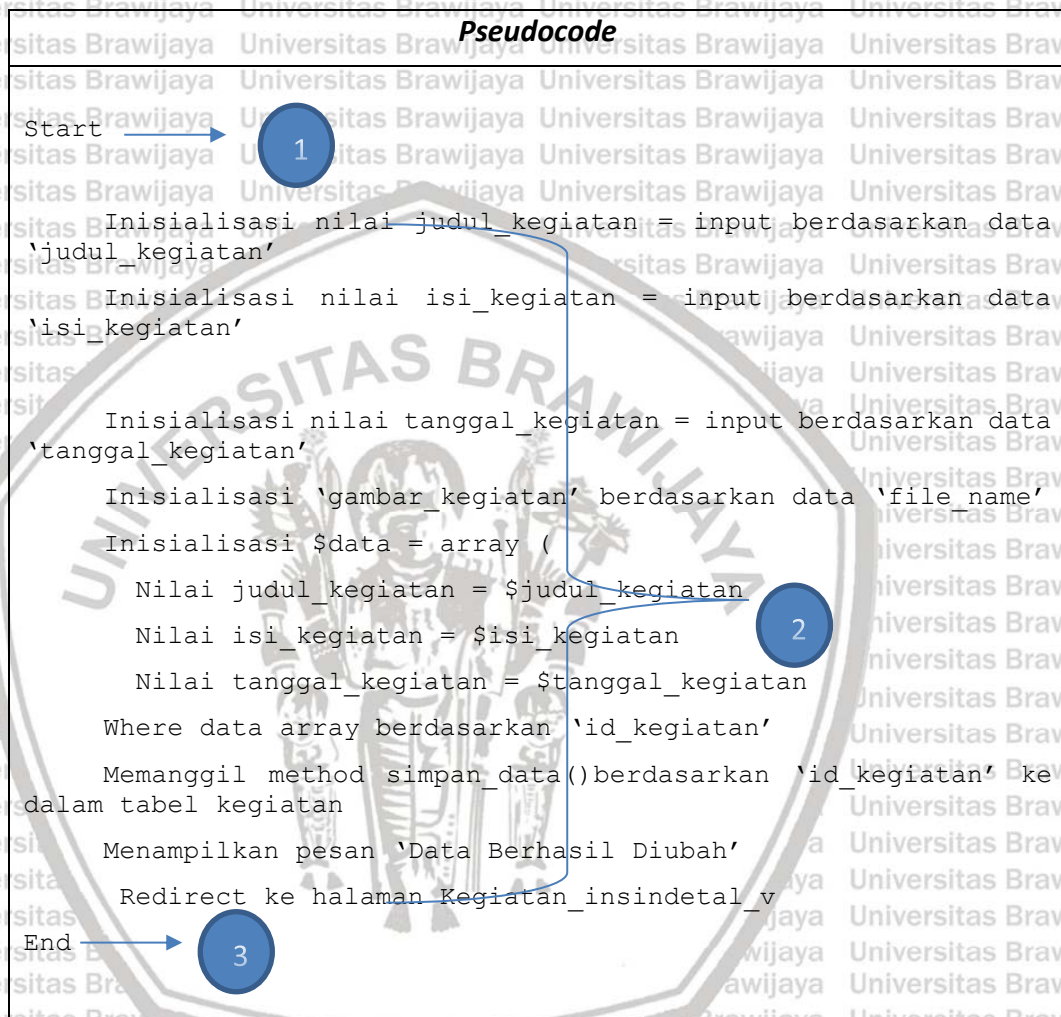
No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1	Jalur 1 = 1-2-3-4- 10-11	Klas <i>driver</i> menjalankan <i>method</i> tambah_kegiatan_aksi() dengan variabel form_validation = <i>false</i> dan memasukan nilai: Judul kegiatan = 'persiapan pernikahan' Isi kegiatan = 'assalamualaikum' Tanggal kegiatan = '2019-12-20' Gambar kegiatan = 'foto_kegiatan-1569860638.jpg'	Mengembalikan nilai <i>false</i> dan menampilkan pesan 'data gagal dimasukkan'	Mengembalikan nilai <i>False</i> dan menampilkan pesan 'data gagal dimasukkan'	Valid
2	Jalur 2 = 1-2-3-5- 6-8-9- 10-11	Klas <i>driver</i> menjalankan <i>method</i> tambah_kegiatan_aksi() dengan variabel form_validation = <i>true</i> , variabel gambar = <i>true</i> , dan memasukan nilai:	Akan mengembalikan nilai <i>True</i> dan menampilkan pesan 'berhasil menyimpan data'	mengembalikan nilai <i>true</i> dan menampilkan pesan 'berhasil menyimpan data'	Valid

		<p>Judul kegiatan = 'persiapan pernikahan'</p> <p>Isi kegiatan = 'assalamualaikum'</p> <p>Tanggal kegiatan = '2019-12-20'</p> <p>Gambar kegiatan = 'foto_kegiatan-1569860638.jpg'</p>			
3	Jalur 3 = 1-2-5-6-7-9-10-11	<p>Klas <i>driver</i> menjalankan <i>method</i> <code>tambah_kegiatan_aksi()</code> dengan variabel <code>form_validation = true</code>, variabel <code>gambar = false</code>, dan memasukkan nilai:</p> <p>Judul kegiatan = 'persiapan pernikahan'</p> <p>Isi kegiatan = 'assalamualaikum'</p> <p>Tanggal kegiatan = '2019-12-20'</p> <p>Gambar kegiatan = 'foto_kegiatan-1569860638.jpg'</p>	Akan menampilkan pesan 'gambar tidak terunggah'	Menampilkan pesan 'gambar tidak terunggah'	Valid

6.1.2 Pengujian unit *method* ubah_kegiatan_aksi() klas Kegiatan_insidental_control

Tabel 6.3 merupakan *pseudocode* dari *method* ubah_kegiatan_aksi() dari klas Kegiatan_insidental_control. Gambar 6.2 merupakan *flow graph* dari *method* ubah_kegiatan_aksi() dari klas Kegiatan_insidental_control.

Tabel 6.3 Pseudocode method ubah_kegiatan_aksi



Flow Graph



Gambar 6.2 Flow graph method ubah_kegiatan_aksi()

Cyclomatic Complexity (V(G))

- $V(G) = \text{jumlah region} = 1$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (2 - 3) + 2 = 1$
- $V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$

Independent Path

- Jalur 1 = 1-2-3-4-5-6

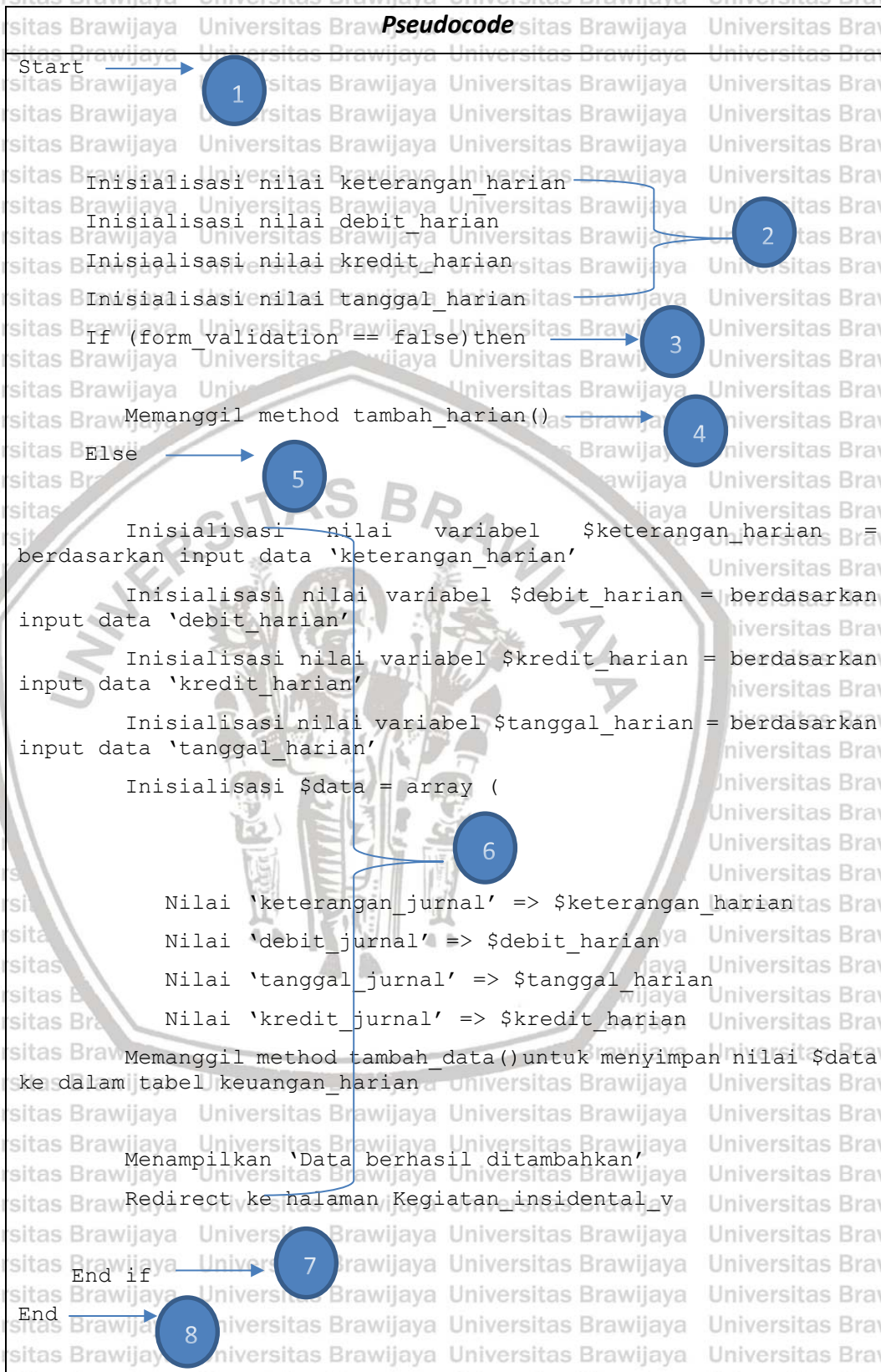
Tabel 6.4 Kasus uji dan hasil uji *method* ubah_kegiatan_aksi

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1	Jalur 1 = 1-2-3	Klas <i>driver</i> menjalankan <i>method</i> ubah_kegiatan_aksi dengan memasukkan nilai: Judul kegiatan: "Tabligh Akbar Menyambut Bulan Suci Ramadhan" Isi kegiatan: "Acara akan dilaksanakan di Masjid Al Ghifari Tanggal 23 April 2021" Tanggal: "2021-04-01"	Return value bernilai <i>true</i>	Return value bernilai <i>true</i>	Valid

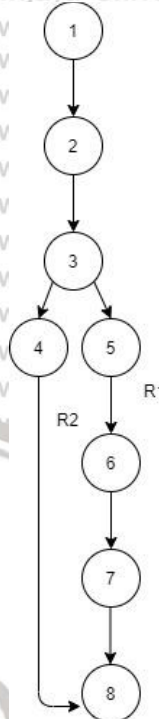
6.1.3 Pengujian unit *method* tambah_harian_aksi() klas Keuangan_harian_aksi

Tabel 6.5 merupakan *pseudocode* dari *method* tambah_harian_aksi() dari klas Keuangan_harian_aksi. Gambar 6.3 merupakan *flow graph* dari *method* tambah_harian_aksi() dari klas Keuangan_harian_control.

Tabel 6.5 Pseudocode method tambah_harian_aksi



Flow Graph



Gambar 6.3 Flow graph method tambah_harian_aksi()

Cyclomatic Complexity (V(G))

- V(G) = jumlah region = 2
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (8 - 8) + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

Independent Path

- Jalur 1 = 1-2-3-4-7-8
- Jalur 2 = 1-2-3-5-6-7-8

Tabel 6.6 Kasus uji dan hasil uji method tambah_harian_aksi

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1	Jalur 1 = 1-2-3-4-7-8	Klas <i>driver</i> menjalankan <i>method</i> <code>tambah_harian_aksi()</code> dengan variabel <code>form_validation = false</code> dan memasukan nilai:	Mengembalikan nilai <code>false</code> dan menampilkan pesan 'data gagal dimasukkan'	Mengembalikan nilai <code>false</code> dan menampilkan pesan 'data gagal dimasukkan'	Valid

		<p>Keterangan = "Sumbangan Masjid"</p> <p>Tanggal = "2021-02-02"</p> <p>Debit = "200000"</p> <p>Kredit = "-"</p>			
2	Jalur 2 = 1-2-3-5-6-7-8	<p>Klas <i>driver</i> menjalankan <i>method</i> <i>tambah_harian_aksi()</i> dengan variabel <i>form_validation = true</i> dan memasukkan nilai:</p> <p>Keterangan = "Sumbangan Masjid"</p> <p>Tanggal = "2021-02-02"</p> <p>Debit = "200000"</p> <p>Kredit = "-"</p>	Mengembalikan nilai <i>True</i> dan menampilkan pesan 'berhasil menyimpan data'	Mengembalikan nilai <i>True</i> dan menampilkan pesan 'berhasil menyimpan data'	Valid

6.2 Pengujian Integrasi

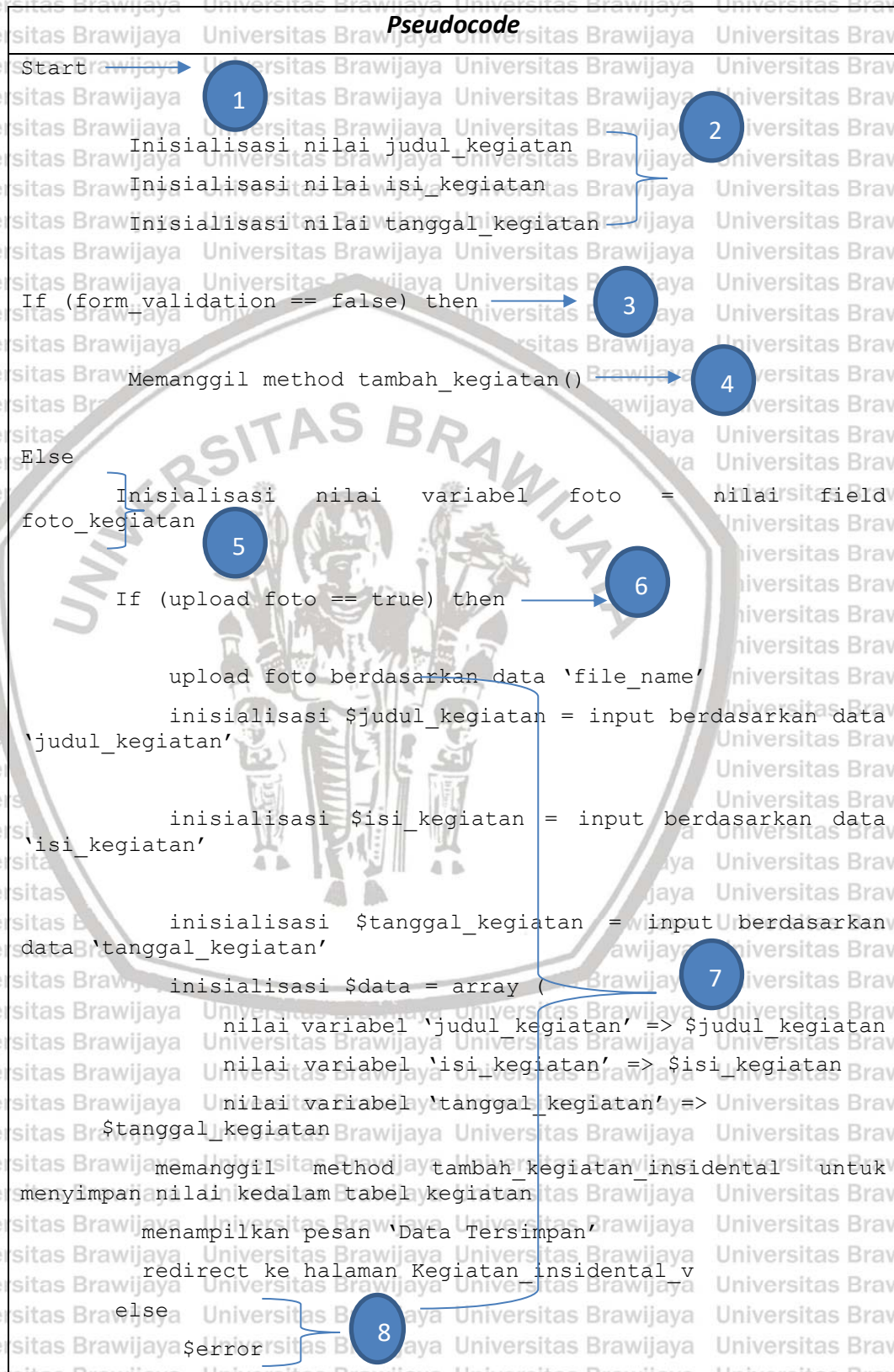
Pengujian integrasi dilakukan untuk menguji klas-klas yang saling berhubungan. Pengujian integrasi dilakukan pada tiga *method* sampel algoritme sebagai bahan pengujian yang berasal dari perancangan komponen. Tiga sampel algoritme yang digunakan, yaitu *method* *tambah_kegiatan_aksi()* dari klas *Kegiatan_insidental_control* dengan *method* *tambah_kegiatan_insidental()* dari klas *Kegiatan*, *method* *ubah_kegiatan_aksi()* dari klas *Kegiatan_insidental_control* dengan *method* *simpan_data()* dari klas *Kegiatan*, dan *method* *tambah_harian_aksi()* dari klas *Kuangan_harian_control* dengan *method* *tambah_data()* dari klas *Kuangan_harian*.

6.2.1 Pengujian integrasi *method* *tambah_kegiatan_aksi()* klas *Kegiatan_insidental_control*

Method *tambah_kegiatan_aksi()* dari klas *Kegiatan_insidental_control* memanggil *method* *tambah_kegiatan_insidental()* dari klas *Kegiatan*. Tabel 6.7 menunjukkan *pseudocode* dari *method* *tambah_kegiatan_aksi()* yang berasal dari

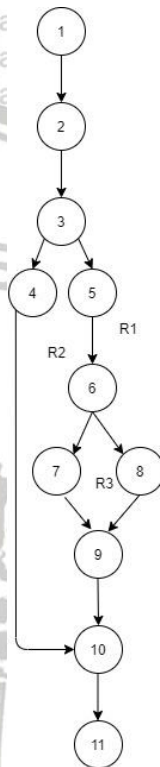
klas Kegiatan_insidental_control. Gambar 6.4 menunjukkan *flow graph* dari *method* tambah_kegiatan_aksi() dari klas Kegiatan_Insidental_control.

Tabel 6.7 Pseudocode method tambah_kegiatan_aksi





Flow Graph



Gambar 6.4 Flow Graph method tambah_kegiatan_aksi()

Cyclomatic Complexity (V(G))

- $V(G) = \text{jumlah region} = 3$

- $V(G) = (\text{jumlah edge} - \text{jumah node}) + 2 = (10 - 9) + 2 = 3$

- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

Independent Path

- Jalur 1 = 1-2-3-4-10-11

- Jalur 2 = 1-2-5-6-7-9-10-11

- Jalur 3 = 1-2-3-5-6-8-9-10-11

Tabel 6.8 Kasus uji dan hasil uji *method* tambah_kegiatan_insidental_aksi

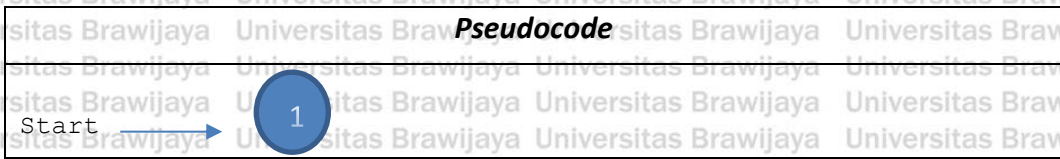
No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1	Jalur 1 = 1-2-3-4- 10-11	Klas <i>driver</i> menjalankan <i>method</i> tambah_kegiatan_aksi() dengan variabel form_validation = <i>false</i> dan memasukan nilai: Judul kegiatan = 'persiapan pernikahan' Isi kegiatan = 'assalamualaikum' Tanggal kegiatan = '2019-12-20' Gambar kegiatan = 'foto_kegiatan-1569860638.jpg'	Mengembalikan nilai <i>false</i> dan menampilkan pesan 'data gagal dimasukkan'	Mengembalikan nilai <i>False</i> dan menampilkan pesan 'data gagal dimasukkan'	Valid
2	Jalur 2 = 1-2-3-5- 6-8-9- 10-11	Klas <i>driver</i> menjalankan <i>method</i> tambah_kegiatan_aksi() dengan variabel form_validation = <i>true</i> , variabel gambar = <i>true</i> , dan memasukan nilai: Judul kegiatan = 'persiapan pernikahan' Isi kegiatan = 'assalamualaikum' Tanggal kegiatan = '2019-12-20' Gambar kegiatan = 'foto_kegiatan-1569860638.jpg'	Akan mengembalikan nilai <i>True</i> dan menampilkan pesan 'berhasil menyimpan data'	mengembalikan nilai <i>true</i> dan menampilkan pesan 'berhasil menyimpan data'	Valid

3	Jalur 3 = 1-2-5-6- 7-9-10- 11	Klas <i>driver</i> menjalankan <i>method</i> <i>tambah_kegiatan_aksi()</i> dengan variabel <i>form_validation = true</i> , variabel <i>gambar = false</i> , dan memasukan nilai: Judul kegiatan = 'persiapan pernikahan' Isi kegiatan = 'assalamualaikum' Tanggal kegiatan = '2019-12-20' Gambar kegiatan = 'foto_kegiatan-1569860638.jpg'	Akan menampilkan pesan 'gambar tidak terunggah'	Menampilkan pesan 'gambar tidak terunggah'	Valid
---	--	--	---	--	-------

6.2.2 Pengujian integrasi *method* ubah_kegiatan_aksi() klas Kegiatan_insidental_control

Method *ubah_kegiatan_aksi()* dari klas *Kegiatan_insidental_control* memanggil *method* *simpan_data()* dari klas *Kegiatan*. Tabel 6.9 menunjukkan *pseudocode* dari *method* *ubah_kegiatan_aksi()* yang berasal dari klas *Kegiatan_insidental_control*. Gambar 6.5 menunjukkan *Flow graph* dari *method* *ubah_kegiatan_aksi()* dari klas *Kegiatan_Insidental_control*.

Tabel 6.9 Pseudocode *method* ubah_kegiatan_aksi



```

Inisialisasi nilai judul_kegiatan = input berdasarkan data
'judul_kegiatan'

Inisialisasi nilai isi_kegiatan = input berdasarkan data
'isi_kegiatan'

Inisialisasi nilai tanggal_kegiatan = input berdasarkan data
'tanggal_kegiatan'

Inisialisasi 'gambar_kegiatan' berdasarkan data 'file_name'

Inisialisasi $data = array (
    Nilai judul_kegiatan = $judul_kegiatan
    Nilai isi_kegiatan = $isi_kegiatan
    Nilai tanggal_kegiatan = $tanggal_kegiatan
)

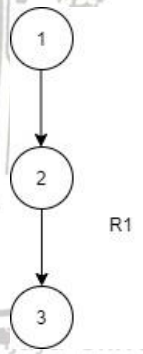
where data array berdasarkan 'id_kegiatan'
Memanggil method simpan_data() berdasarkan 'id_kegiatan' ke
dalam tabel kegiatan

Menampilkan pesan 'Data Berhasil Diubah'

Redirect ke halaman Kegiatan insindetal_v

End
    
```

Flow Graph



Gambar 6.5 Flow graph method ubah_kegiatan_aksi()

Cyclomatic Complexity (V(G))

- $V(G) = \text{jumlah region} = 1$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (2 - 3) + 2 = 1$
- $V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$

Independent Path

- Jalur 1 = 1-2-3-4-5-6



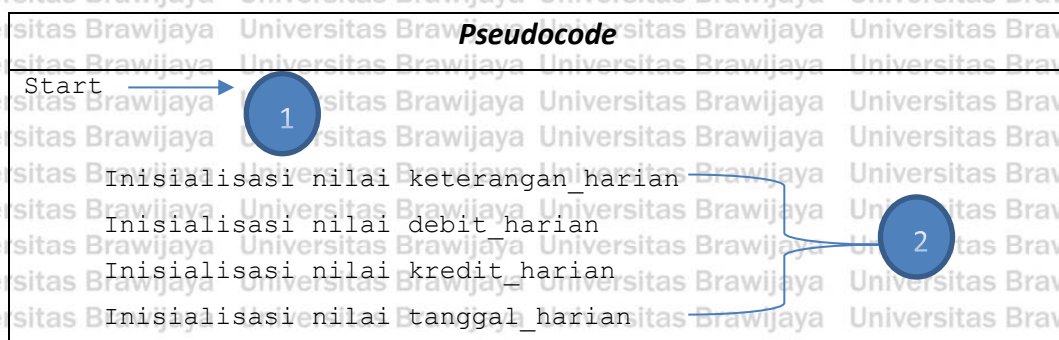
Tabel 6.10 Kasus uji dan hasil uji *method* ubah_kegiatan_aksi

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1	Jalur 1 = 1-2-3	Klas <i>driver</i> menjalankan <i>method</i> ubah_kegiatan_aksi dengan memasukkan nilai: Judul kegiatan: "Tabligh Akbar Menyambut Bulan Suci Ramadhan" Isi kegiatan: "Acara akan dilaksanakan di Masjid Al Ghifari Tanggal 23 April 2021" Tanggal: "2021-04-01"	<i>Return value</i> bernilai <i>true</i>	<i>Return value</i> bernilai <i>true</i>	Valid

6.2.3 Pengujian integrasi *method* tambah_harian_aksi() Klas Keuangan_harian_control

Method tambah_harian_aksi() dari klas Keuangan_harian_control memanggil *method* tambah_data() dari klas Keuangan_harian. Tabel 6.11 menunjukkan *pseudocode* dari *method* tambah_harian_aksi() yang berasal dari klas Keuangan_harian_control. Gambar 6.6 menunjukkan *Flow graph* dari *method* tambah_harian_aksi() dari klas Keuangan_harian_control.

Tabel 6.11 *Pseudocode method* tambah_harian_aksi



```

If (form_validation == false) then
    Memanggil method tambah_harian()
Else
    Inisialisasi nilai variabel $keterangan_harian =
    berdasarkan input data 'keterangan_harian'
    Inisialisasi nilai variabel $debit_harian = berdasarkan
    input data 'debit_harian'
    Inisialisasi nilai variabel $kredit_harian = berdasarkan
    input data 'kredit_harian'
    Inisialisasi nilai variabel $tanggal_harian = berdasarkan
    input data 'tanggal_harian'
    Inisialisasi $data = array (
        Nilai 'keterangan_jurnal' => $keterangan_harian
        Nilai 'debit_jurnal' => $debit_harian
        Nilai 'tanggal_jurnal' => $tanggal_harian
        Nilai 'kredit_jurnal' => $kredit_harian
    )
    Memanggil method tambah_data() untuk menyimpan nilai $data
    ke dalam tabel keuangan_harian
    Menampilkan 'Data berhasil ditambahkan'
    Redirect ke halaman kegiatan_insidental_v
End if
End

```

3

4

5

6

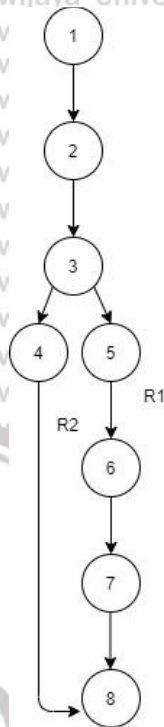
6

7

8



Flow Graph



Gambar 6.6 Flow graph method tambah_harian_aksi()

Cyclomatic Complexity (V(G))

- V(G) = jumlah region = 2
- $V(G) = (\text{jumlah edge} - \text{jumah node}) + 2 = (8 - 8) + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

Independent Path

- Jalur 1 = 1-2-3-4-7-8
- Jalur 2 = 1-2-3-5-6-7-8

Tabel 6.12 Kasus uji dan hasil uji method tambah_harian_aksi

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1	Jalur 1 = 1-2-3-4-7-8	Klas driver menjalankan method tambah_harian_aksi() dengan variabel form_validation =	Mengembalikan nilai false dan menampilkan pesan 'data gagal dimasukkan'	Mengembalikan nilai false dan menampilkan pesan 'data gagal dimasukkan'	Valid

		<p><i>false</i> dan memasukkan nilai:</p> <p>Keterangan = "Sumbangan Masjid"</p> <p>Tanggal = "2021-02-02"</p> <p>Debit = "200000"</p> <p>Kredit = "-"</p>			
2	Jalur 2 = 1-2-3-5-6-7-8	<p>Klas <i>driver</i> menjalankan <i>method</i> <i>tambah_harian_aksi()</i> dengan variabel <i>form_validation = true</i> dan memasukkan nilai:</p> <p>Keterangan = "Sumbangan Masjid"</p> <p>Tanggal = "2021-02-02"</p> <p>Debit = "200000"</p> <p>Kredit = "-"</p>	Mengembalikan nilai <i>True</i> dan menampilkan pesan 'berhasil menyimpan data'	Mengembalikan nilai <i>True</i> dan menampilkan pesan 'berhasil menyimpan data'	Valid

6.3 Pengujian Validasi

Pengujian validasi dilakukan menggunakan teknik pengujian *black box* dengan metode *scenario-based testing*. Pengujian validasi dilakukan untuk memeriksa sistem yang sudah dibuat sesuai dengan *use case scenario* yang sudah ada. Pengujian validasi dilakukan pada semua kebutuhan fungsional dan nonfungsional sistem. Pengujian kebutuhan nonfungsional pada penelitian ini menggunakan pengujian *compatibility*.

6.3.1 Pengujian validasi login

Kasus uji *login* jalur utama ditunjukkan pada Tabel 6.13, kasus uji *login* jalur alternative 1a ditunjukkan pada Tabel 6.14, serta kasus uji *login* jalur alternative 1b ditunjukkan pada Tabel 6.15.

Tabel 6.13 Kasus uji *Login* jalur utama

Kode kebutuhan	SIMAGFARI_F_01
Nama kasus uji	<i>Login</i> jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman login 2. Memasukkan: Nama: "malik" Password: "123456789" Untuk masuk ke dalam sistem dengan hak akses Anggota Bidang Ubudiyah 3. Menekan tombol <i>login</i>
Hasil yang diharapkan	Halaman sistem terbuka dan aktor masuk ke dalam sistem sesuai dengan hak aksesnya
Hasil yang didapatkan	Halaman sistem terbuka dan aktor masuk ke dalam sistem sesuai dengan hak aksesnya
Status validasi	Valid

Tabel 6.14 Kasus uji *Login* jalur alternatif 1a

Kode kebutuhan	SIMAGFARI_F_01
Nama kasus uji	<i>Login</i> jalur alternative 1a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman login 2. Memasukkan: Nama: "" Password: "" Untuk masuk ke dalam sistem dengan hak akses Anggota Bidang Ubudiyah 3. Menekan tombol <i>login</i>
Hasil yang diharapkan	Sistem akan menampilkan pesan eror "Harap isi nama dan Password"
Hasil yang didapatkan	Sistem akan menampilkan pesan eror "Harap isi nama dan Password"
Status validasi	Valid

Tabel 6.15 Kasus uji Login jalur alternatif 1b

Kode kebutuhan	SIMAGFARI_F_01
Nama kasus uji	Login jalur alternatif 1b
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman login 2. Memasukkan: Nama: "Malik44" Password: "123456789" Untuk masuk ke dalam sistem dengan hak akses Anggota Bidang Ubudiyah 3. Menekan tombol <i>login</i>
Hasil yang diharapkan	Sistem akan menampilkan error "Nama atau Password yang anda masukkan salah, coba lagi!!!"
Hasil yang didapatkan	Sistem akan menampilkan error "Nama atau Password yang anda masukkan salah, coba lagi!!!"
Status validasi	Valid

6.3.2 Pengujian validasi *logout*

Kasus uji *logout* jalur utama ditunjukkan pada Tabel 6.16 dan kasus uji *logout* jalur *alternative* 2a ditunjukkan pada Tabel 6.17

Tabel 6.16 Kasus uji Logout jalur utama

Kode kebutuhan	SIMAGFARI_F_02
Nama kasus uji	Logout jalur utam
Prosedur	<ol style="list-style-type: none"> 1. Mengakses sistem dengan cara <i>login</i> sesuai dengan hak akses 2. Menekan tombol <i>logout</i> 3. Memilih pilihan "Ya, saya yakin"
Hasil yang diharapkan	Sistem menampilkan halaman <i>login</i>
Hasil yang didapatkan	Sistem menampilkan halaman <i>login</i>
Status validasi	Valid

Tabel 6.17 Kasus uji Logout jalur alternatif 2a

Kode kebutuhan	SIMAGFARI_F_02
Nama kasus uji	Logout jalur alternatif 2a

Prosedur	<ol style="list-style-type: none"> 1. Mengakses sistem dengan cara <i>login</i> sesuai dengan hak akses 2. Menekan tombol <i>logout</i> 3. Memilih pilihan “Batalkan”
Hasil yang diharapkan	Sistem menampilkan halaman awal sistem
Hasil yang didapatkan	Sistem menampilkan halaman awal sistem
Status validasi	Valid

6.3.3 Pengujian validasi tambah artikel

Kasus uji tambah artikel jalur utama ditunjukkan pada Tabel 6.18 dan kasus uji tambah artikel jalur *alternative* 5a ditunjukkan pada Tabel 6.19.

Tabel 6.18 Kasus uji tambah artikel jalur utama

Kode kebutuhan	SIMAGFARI_F_03
Nama kasus uji	Tambah artikel jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman artikel 2. Menekan tombol tambah data 3. Mengisi form tambah artikel dengan data sebagai berikut: Judul artikel : “Berkah Ramadhan” Isi artikel : “Ramadhan” Gambar : “foto_artikel-1610637268.jpeg” Tanggal : “2021-01-01” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman artikel dan menampilkan artikel yang sudah dibuat
Hasil yang didapatkan	Sistem mengarahkan ke halaman artikel dan menampilkan artikel yang sudah dibuat
Status validasi	Valid

Tabel 6.19 Kasus uji tambah artikel jalur alternatif 5a

Kode kebutuhan	SIMAGFARI_F_03
Nama kasus uji	Tambah artikel jalur alternatif 5a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman artikel 2. Menekan tombol tambah data

	3. Mengisi form tambah artikel dengan data sebagai berikut: Judul artikel : “ ” Isi artikel : “ ” Gambar: “ ” Tanggal : “ ”
	4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Hasil yang didapatkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Status validasi	Valid

6.3.4 Pengujian validasi ubah artikel

Kasus uji ubah artikel jalur utama ditunjukkan pada Tabel 6.20 dan kasus uji ubah artikel jalur *alternative* 4a ditunjukkan pada Tabel 6.21.

Tabel 6.20 Kasus uji ubah artikel jalur utama

Kode kebutuhan	SIMAGFARI _F_04
Nama kasus uji	Ubah artikel jalur utama
Prosedur	1. Mengakses halaman artikel 2. Menekan tombol ubah artikel 3. Mengisi form ubah artikel dengan data sebagai berikut: Judul artikel : “Ramadhan” Isi artikel : “ Marhaban Ramadhan” Tanggal : “2021-01-01” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman artikel dan menampilkan artikel yang sudah diubah
Hasil yang didapatkan	Sistem mengarahkan ke halaman artikel dan menampilkan artikel yang sudah diubah
Status validasi	Valid

Tabel 6.21 Kasus uji ubah artikel jalur alternatif 4a

Kode kebutuhan	SIMAGFARI _F_04
Nama kasus uji	Ubah artikel jalur alternatif 4a

Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman artikel 2. Menekan tombol ubah artikel 3. Mengisi form ubah artikel dengan data sebagai berikut: Judul artikel : “ ” Isi artikel : “ ” Tanggal : “ ” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Hasil yang didapatkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Status validasi	Valid

6.3.5 Pengujian validasi hapus artikel

Kasus uji hapus artikel jalur utama ditunjukkan pada Tabel 6.22 dan kasus uji hapus artikel jalur *alternative* 3a ditunjukkan pada Tabel 6.23.

Tabel 6.22 Kasus uji hapus artikel jalur utama

Kode kebutuhan	SIMAGFARI _F_05
Nama kasus uji	Hapus artikel jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman artikel 2. Memilih artikel yang akan dihapus 3. Menekan tombol hapus artikel 4. Sistem menampilkan pesan “Apakah anda yakin menghapus artikel ini” 5. Mengklik “Ya, saya yakin” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman artikel
Hasil yang didapatkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman artikel
Status validasi	Valid

Tabel 6.23 Kasus uji hapus artikel jalur alternatif 3a

Kode kebutuhan	SIMAGFARI_F_05
Nama kasus uji	Hapus artikel jalur alternatif 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman artikel 2. Memilih artikel yang akan dihapus 3. Menekan tombol hapus artikel 4. Sistem menampilkan pesan “Apakah anda yakin menghapus artikel ini” 5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Artikel tidak terhapus dan sistem menampilkan halaman artikel
Hasil yang didapatkan	Artikel tidak terhapus dan sistem menampilkan halaman artikel
Status validasi	Valid

6.3.6 Pengujian validasi tambah pengumuman

Kasus uji tambah pengumuman jalur utama ditunjukkan pada Tabel 6.24 dan kasus uji tambah pengumuman jalur *alternative* 5a ditunjukkan pada Tabel 6.24.

Tabel 6.24 Kasus uji tambah pengumuman jalur utama

Kode kebutuhan	SIMAGFARI_F_06
Nama kasus uji	Tambah pengumuman jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Menekan tombol tambah data 3. Mengisi form tambah pengumuman dengan data sebagai berikut: Judul pengumuman : “Pendaftaran Tabligh Akbar Ustadz Yusuf Mansur” Isi pengumuman : “Ustadz Yusuf Mansur” Gambar: “foto_pengumuman-1609272737.jpg” Tanggal : “2021-01-01” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman pengumuman dan menampilkan pengumuman yang sudah dibuat

Hasil yang didapatkan	Sistem mengarahkan ke halaman artikel dan menampilkan artikel yang sudah dibuat
Status validasi	Valid

Tabel 6.25 Kasus uji tambah pengumuman jalur alternatif 5a

Kode kebutuhan	SIMAGFARI_F_06
Nama kasus uji	Tambah pengumuman jalur alternatif 5a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Menekan tombol tambah data 3. Mengisi form tambah pengumuman dengan data sebagai berikut: Judul pengumuman : “ ” Isi pengumuman : “ ” Gambar: “ ” Tanggal : “ ” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “field is required”
Hasil yang didapatkan	Sistem menampilkan pesan “field is required”
Status validasi	Valid

6.3.7 Pengujian validasi ubah pengumuman

Kasus uji ubah pengumuman jalur utama ditunjukkan pada Tabel 6.26 dan kasus uji ubah pengumuman jalur *alternative* 4a ditunjukkan pada Tabel 6.27.

Tabel 6.26 Kasus uji ubah pengumuman jalur utama

Kode kebutuhan	SIMAGFARI_F_07
Nama kasus uji	Ubah pengumuman jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Menekan tombol ubah pengumuman 3. Mengisi form ubah pengumuman dengan data sebagai berikut: Judul pengumuman : “Ramadhan” Isi pengumuman : “ Marhaban Ramadhan” Tanggal : “2021-01-01”

	4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman pengumuman dan menampilkan pengumuman yang sudah diubah
Hasil yang didapatkan	Sistem mengarahkan ke halaman pengumuman dan menampilkan pengumuman yang sudah diubah
Status validasi	Valid

Tabel 6.27 Kasus uji ubah pengumuman jalur alternative 4a

Kode kebutuhan	SIMAGFARI_F_07
Nama kasus uji	Ubah pengumuman jalur alternatif 4a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Menekan tombol ubah pengumuman 3. Mengisi form ubah pengumuman dengan data sebagai berikut: Judul pengumuman : “ ” Isi pengumuman : “ ” Tanggal : “ ” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Hasil yang didapatkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Status validasi	Valid

6.3.8 Pengujian validasi hapus pengumuman

Kasus uji hapus pengumuman jalur utama ditunjukkan pada Tabel 6.28 dan kasus uji hapus pengumuman jalur *alternative* 3a ditunjukkan pada Tabel 6.29.

Tabel 6.28 Kasus uji hapus pengumuman jalur utama

Kode kebutuhan	SIMAGFARI_F_08
Nama kasus uji	Hapus pengumuman jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Memilih pengumuman yang akan dihapus 3. Menekan tombol hapus pengumuman 4. Sistem menampilkan pesan “Apakah anda yakin menghapus pengumuman ini”

	5. Mengklik “Ya, saya yakin” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman pengumuman
Hasil yang didapatkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman pengumuman
Status validasi	Valid

Tabel 6.29 Kasus uji hapus pengumuman jalur alternatif 3a

Kode kebutuhan	SIMAGFARI_F_08
Nama kasus uji	Hapus pengumuman jalur alternatif 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Memilih pengumuman yang akan dihapus 3. Menekan tombol hapus pengumuman 4. Sistem menampilkan pesan “Apakah anda yakin menghapus pengumuman ini” 5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Pengumuman tidak terhapus dan sistem menampilkan halaman pengumuman
Hasil yang didapatkan	Pengumuman tidak terhapus dan sistem menampilkan halaman pengumuman
Status validasi	Valid

6.3.9 Pengujian validasi tambah foto

Kasus uji tambah foto jalur utama ditunjukkan pada Tabel 6.30 dan kasus uji tambah foto jalur *alternative 3a* ditunjukkan pada Tabel 6.31.

Tabel 6.30 Kasus uji tambah foto jalur utama

Kode kebutuhan	SIMAGFARI_F_09
Nama kasus uji	Tambah foto jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman foto 2. Menekan tombol tambah data 3. Mengisi form tambah foto dengan data sebagai berikut: Nama foto : “Berkah Ramadhan”

	Gambar: "foto_kegiatan-1612183655.jpeg" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman foto dan menampilkan foto yang sudah dibuat
Hasil yang didapatkan	Sistem mengarahkan ke halaman foto dan menampilkan foto yang sudah dibuat
Status validasi	Valid

Tabel 6.31 Kasus uji tambah foto jalur alternatif 3a

Kode kebutuhan	SIMAGFARI_F_09
Nama kasus uji	Tambah foto jalur alternatif 3a
Prosedur	1. Mengakses halaman foto 2. Menekan tombol tambah data 3. Mengisi form tambah foto dengan data sebagai berikut: Nama foto : " " Gambar: " " 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.10 Pengujian validasi hapus foto

Kasus uji hapus foto jalur utama ditunjukkan pada Tabel 6.32 dan kasus uji hapus foto *alternative* 4a ditunjukkan pada Tabel 6.33.

Tabel 6.32 Kasus uji hapus foto jalur utama

Kode kebutuhan	SIMAGFARI_F_10
Nama kasus uji	Hapus foto jalur utama
Prosedur	1. Mengakses halaman foto 2. Memilih foto yang akan dihapus 3. Menekan tombol hapus foto 4. Sistem menampilkan pesan "Apakah anda yakin menghapus foto ini"

	5. Mengklik “Ya, saya yakin” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman foto
Hasil yang didapatkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman foto
Status validasi	Valid

Tabel 6.33 Kasus uji hapus foto jalur alternatif 4a

Kode kebutuhan	SIMAGFARI_F_10
Nama kasus uji	Hapus foto jalur alternatif 4a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman foto 2. Memilih foto yang akan dihapus 3. Menekan tombol hapus pengumuman 4. Sistem menampilkan pesan “Apakah anda yakin menghapus foto ini” 5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Pengumuman tidak terhapus dan sistem menampilkan halaman foto
Hasil yang didapatkan	Pengumuman tidak terhapus dan sistem menampilkan halaman foto
Status validasi	Valid

6.3.11 Pengujian validasi tambah kegiatan insidental

Kasus uji tambah kegiatan insidental jalur utama ditunjukkan pada Tabel 6.34 dan kasus uji tambah kegiatan insidental jalur *alternative* 5a ditunjukkan pada Tabel 6.35.

Tabel 6.34 Kasus uji tambah kegiatan insidental jalur utama

Kode kebutuhan	SIMAGFARI_F_11
Nama kasus uji	Tambah kegiatan insidental jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan insidental 2. Menekan tombol tambah data 3. Mengisi form tambah kegiatan insidental dengan data sebagai berikut:

	Judul kegiatan: "Problematika Umat" Isi kegiatan : "Bismillahirrahmanirrahim HIJRAH UNITED PRESENT" Gambar: "foto_kegiatan-1569861907.jpg" Tanggal : "2021-01-01" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman kegiatan insidental dan menampilkan kegiatan insidental yang sudah dibuat
Hasil yang didapatkan	Sistem mengarahkan ke halaman kegiatan insidental dan menampilkan kegiatan insidental yang sudah dibuat
Status validasi	Valid

Tabel 6.35 Kasus uji tambah kegiatan insidental jalur alternatif 5a

Kode kebutuhan	SIMAGFARI _F_11
Nama kasus uji	Tambah kegiatan insidental jalur alternatif 5a
Prosedur	1. Mengakses halaman kegiatan insidental 2. Menekan tombol tambah data 3. Mengisi form tambah kegiatan insidental dengan data sebagai berikut: Judul kegiatan: "" Isi kegiatan : "" Gambar: "" Tanggal : "" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.12 Pengujian validasi ubah kegiatan insidental

Kasus uji ubah kegiatan insidental jalur utama ditunjukkan pada Tabel 6.36 dan kasus uji ubah kegiatan insidental jalur *alternative* 4a ditunjukkan pada Tabel 6.37.

Tabel 6.36 Kasus uji ubah kegiatan insindetal jalur utama

Kode kebutuhan	SIMAGFARI_F_12
Nama kasus uji	Ubah kegiatan insindetal jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan insidental 2. Menekan tombol ubah kegiatan insindetal 3. Mengisi form ubah kegiatan insidental dengan data sebagai berikut: Judul kegiatan : “Ramadhan” Isi kegiatan : “ Marhaban Ramadhan” Tanggal : “2021-01-01” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman kegiatan insidental dan menampilkan kegiatan insidental yang sudah diubah
Hasil yang didapatkan	Sistem mengarahkan ke halaman kegiatan insidental dan menampilkan kegiatan insidental yang sudah diubah
Status validasi	Valid

Tabel 6.37 Kasus uji ubah kegiatan insidental jalur alternatif 4a

Kode kebutuhan	SIMAGFARI_F_12
Nama kasus uji	Ubah kegiatan insidental jalur alternatif 4a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan insidental 2. Menekan tombol ubah kegiatan insidental 3. Mengisi form ubah kegiatan insidental dengan data sebagai berikut: Judul kegiatan: “ ” Isi kegiatan: “ ” Tanggal : “ ” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “field is required”
Hasil yang didapatkan	Sistem menampilkan pesan “field is required”
Status validasi	Valid

6.3.13 Pengujian validasi hapus kegiatan insidental

Kasus uji hapus kegiatan insidental jalur utama ditunjukkan pada Tabel 6.38 dan kasus uji hapus kegiatan insidental jalur *alternative* 3a ditunjukkan pada Tabel 6.39.

Tabel 6.38 Kasus uji hapus kegiatan insidental jalur utama

Kode kebutuhan	SIMAGFARI_F_13
Nama kasus uji	Hapus kegiatan insidental jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan insidental 2. Memilih kegiatan insidental yang akan dihapus 3. Menekan tombol hapus kegiatan insidental 4. Sistem menampilkan pesan “Apakah anda yakin menghapus kegiatan insidental ini” 5. Mengklik “Ya, saya yakin” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman kegiatan insidental
Hasil yang didapatkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman kegiatan insidental
Status validasi	Valid

Tabel 6.39 Kasus uji hapus kegiatan insidental jalur alternatif 3a

Kode kebutuhan	SIMAGFARI_F_13
Nama kasus uji	Hapus kegiatan insidental jalur alternatif 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan insidental 2. Memilih kegiatan insidental yang akan dihapus 3. Menekan tombol hapus kegiatan insidental 4. Sistem menampilkan pesan “Apakah anda yakin menghapus kegiatan insidental ini” 5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Pengumuman tidak terhapus dan sistem menampilkan halaman kegiatan insidental
Hasil yang didapatkan	Pengumuman tidak terhapus dan sistem menampilkan halaman kegiatan insidental

Status validasi	Valid
-----------------	-------

6.3.14 Pengujian validasi tambah kegiatan rutin

Kasus uji tambah kegiatan rutin jalur utama ditunjukkan pada Tabel 6.40 dan kasus uji tambah kegiatan jalur *alternative* 5a ditunjukkan pada Tabel 6.41.

Tabel 6.40 Kasus uji tambah kegiatan rutin jalur utama

Kode kebutuhan	SIMAGFARI_F_14
Nama kasus uji	Tambah kegiatan rutin jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan rutin 2. Menekan tombol tambah data 3. Mengisi form tambah kegiatan rutin dengan data sebagai berikut: Judul kegiatan: "Buka Puasa Senin dan Kamis" Isi kegiatan : "Kegiatan tersebut diadakan setiap hari Senin dan Kamis" Tanggal : "2021-01-01" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman kegiatan rutin dan menampilkan kegiatan rutin yang sudah dibuat
Hasil yang didapatkan	Sistem mengarahkan ke halaman kegiatan rutin dan menampilkan kegiatan rutin yang sudah dibuat
Status validasi	Valid

Tabel 6.41 Kasus uji tambah kegiatan rutin jalur alternatif 5a

Kode kebutuhan	SIMAGFARI_F_14
Nama kasus uji	Tambah kegiatan rutin jalur alternatif 5a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan rutin 2. Menekan tombol tambah data 3. Mengisi form tambah kegiatan rutin dengan data sebagai berikut: Judul kegiatan: "" Isi kegiatan : "" Tanggal : "" 4. Menekan tombol simpan

Hasil yang diharapkan	Sistem menampilkan pesan “field is required”
Hasil yang didapatkan	Sistem menampilkan pesan “field is required”
Status validasi	Valid

6.3.15 Pengujian validasi ubah kegiatan rutin

Kasus uji ubah kegiatan rutin jalur utama ditunjukkan pada Tabel 6.42 dan kasus uji ubah kegiatan jalur *alternative* 4a ditunjukkan pada Tabel 5.43.

Tabel 6.42 Kasus uji ubah kegiatan rutin jalur utama

Kode kebutuhan	SIMAGFARI _F_15
Nama kasus uji	Ubah kegiatan rutin jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan rutin 2. Menekan tombol ubah kegiatan rutin 3. Mengisi form ubah kegiatan rutin dengan data sebagai berikut: Judul kegiatan: “Kajian Sore” Isi kegiatan: “Kajian dilaksanakan setelah salat Ashar” Tanggal: “2021-01-01” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman kegiatan rutin dan menampilkan kegiatan rutin yang sudah diubah
Hasil yang didapatkan	Sistem mengarahkan ke halaman kegiatan rutin dan menampilkan kegiatan rutin yang sudah diubah
Status validasi	Valid

Tabel 6.43 Kasus uji ubah kegiatan rutin jalur alternatif 4a

Kode kebutuhan	SIMAGFARI _F_15
Nama kasus uji	Ubah kegiatan rutin jalur alternatif 4a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan rutin 2. Menekan tombol ubah kegiatan rutin 3. Mengisi form ubah kegiatan rutin dengan data sebagai berikut: Judul kegiatan: “ ” Isi kegiatan: “ ”

	Tanggal : “ ”
	4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Hasil yang didapatkan	Sistem menampilkan pesan “ <i>field is required</i> ”
Status validasi	Valid

6.3.16 Pengujian validasi hapus kegiatan rutin

Kasus uji hapus kegiatan rutin jalur utama ditunjukkan pada Tabel 6.44 dan kasus uji hapus kegiatan rutin jalur *alternative* 3a ditunjukkan pada Tabel 6.45.

Tabel 6.44 Kasus uji hapus kegiatan rutin jalur utama

Kode kebutuhan	SIMAGFARI _F_16
Nama kasus uji	Hapus kegiatan rutin jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan rutin 2. Memilih kegiatan rutin yang akan dihapus 3. Menekan tombol hapus kegiatan rutin 4. Sistem menampilkan pesan “Apakah anda yakin menghapus kegiatan rutin ini” 5. Mengklik “Ya, saya yakin” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman kegiatan rutin
Hasil yang didapatkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman kegiatan rutin
Status validasi	Valid

Tabel 6.45 Kasus uji hapus kegiatan rutin jalur alternatif 3a

Kode kebutuhan	SIMAGFARI _F_16
Nama kasus uji	Hapus kegiatan rutin jalur alternatif 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan rutin 2. Memilih kegiatan rutin yang akan dihapus 3. Menekan tombol hapus kegiatan rutin 4. Sistem menampilkan pesan “Apakah anda yakin menghapus kegiatan rutin ini”

	5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Pengumuman tidak terhapus dan sistem menampilkan halaman kegiatan rutin
Hasil yang didapatkan	Pengumuman tidak terhapus dan sistem menampilkan halaman kegiatan rutin
Status validasi	Valid

6.3.17 Pengujian validasi tambah imam salat

Kasus uji tambah imam salat jalur utama ditunjukkan pada Tabel 6.46 dan kasus uji tambah imam salat jalur *alternative* 5a ditunjukkan pada Tabel 6.47.

Tabel 6.46 Kasus uji tambah imam salat jalur utama

Kode kebutuhan	SIMAGFARI_F_17
Nama kasus uji	Tambah imam salat jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman imam salat 2. Menekan tombol tambah data 3. Mengisi form tambah imam salat dengan data sebagai berikut: Tanggal imam : “2021-01-01” Imam Subuh “Ustad Furqon” Imam Zuhur: “Ustad Furqon” Imam Asar:”Ustad Agus Junaedi” Imam Maghrib:”Ustad Agus Junaedi” Imam Isya:”Ustad Furqon” 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman imam salat dan menampilkan imam salat yang sudah dibuat
Hasil yang didapatkan	Sistem mengarahkan ke halaman imam salat dan menampilkan imam salat yang sudah dibuat
Status validasi	Valid

Tabel 6.47 Kasus uji tambah imam salat jalur alternative 5a

Kode kebutuhan	SIMAGFARI_F_17
Nama kasus uji	Tambah imam salat jalur alternatif 5a

Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman imam salat 2. Menekan tombol tambah data 3. Mengisi form tambah imam salat dengan data sebagai berikut: Tanggal imam : "" Imam Subuh "" Imam Zuhur: "" Imam Asar: "" Imam Maghrib: "" Imam Isya: "" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.18 Pengujian validasi ubah imam salat

Kasus uji ubah imam salat jalur utama ditunjukkan pada Tabel 6.48 dan kasus uji ubah imam salat *alternative* 4a ditunjukkan pada Tabel 6.49.

Tabel 6.48 Kasus uji ubah imam salat jalur utama

Kode kebutuhan	SIMAGFARI_F_18
Nama kasus uji	Ubah imam salat jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman imam salat 2. Menekan tombol ubah imam salat 3. Mengisi form ubah imam salat dengan data sebagai berikut: Tanggal imam : "2021-01-01" Imam Subuh "Ustad Furqon" Imam Zuhur: "Ustad Agus Junaedi" Imam Asar: "Ustad Agus Junaedi" Imam Maghrib: "Ustad Agus Junaedi" Imam Isya: "Ustad Agus Juneadi" 4. Menekan tombol simpan

Hasil yang diharapkan	Sistem mengarahkan ke halaman imam salat dan menampilkan imam salat yang sudah diubah
Hasil yang didapatkan	Sistem mengarahkan ke halaman imam salat dan menampilkan imam salat yang sudah diubah
Status validasi	Valid

Tabel 6.49 Kasus uji ubah imam salat jalur alternatif 4a

Kode kebutuhan	SIMAGFARI_F_18
Nama kasus uji	Ubah imam salat jalur alternatif 4a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman imam salat 2. Menekan tombol ubah imam salat 3. Mengisi form ubah imam salat dengan data sebagai berikut: Tanggal imam : "" Imam Subuh "" Imam Zuhur: "" Imam Asar: "" Imam Maghrib: "" Imam Isya: "" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.19 Pengujian validasi hapus imam salat

Kasus uji hapus imam salat jalur utama ditunjukkan pada Tabel 6.50 dan kasus uji hapus imam salat jalur *alternative* 3a ditunjukkan pada Tabel 6.51.

Tabel 6.50 Kasus uji hapus imam salat jalur utama

Kode kebutuhan	SIMAGFARI_F_19
Nama kasus uji	Hapus imam salat jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman imam salat 2. Memilih imam salat yang akan dihapus 3. Menekan tombol hapus imam salat

	4. Sistem menampilkan pesan “Apakah anda yakin menghapus imam salat ini” 5. Mengklik “Ya, saya yakin” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman imam salat
Hasil yang didapatkan	Sistem menampilkan pesan “Data Terhapus” dan menampilkan halaman imam salat
Status validasi	Valid

Tabel 6.51 Kasus uji hapus imam salat jalur alternatif 3a

Kode kebutuhan	SIMAGFARI_F_19
Nama kasus uji	Hapus imam salat jalur alternatif 3a
Prosedur	1. Mengakses halaman imam salat 2. Memilih imam salat yang akan dihapus 3. Menekan tombol hapus imam salat 4. Sistem menampilkan pesan “Apakah anda yakin menghapus imam salat ini” 5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Pengumuman tidak terhapus dan sistem menampilkan halaman imam salat
Hasil yang didapatkan	Pengumuman tidak terhapus dan sistem menampilkan halaman imam salat
Status validasi	Valid

6.3.20 Pengujian validasi tambah jadwal salat

Kasus uji tambah jadwal salat jalur utama ditunjukkan pada Tabel 6.52 dan kasus uji tambah jadwal salat jalur alternative 5a ditunjukkan pada Tabel 6.53.

Tabel 6.52 Kasus uji tambah jadwal salat jalur utama

Kode kebutuhan	SIMAGFARI_F_20
Nama kasus uji	Tambah jadwal salat jalur utama
Prosedur	1. Mengakses halaman jadwal salat 2. Menekan tombol tambah data

	<p>3. Mengisi form tambah jadwal salat dengan data sebagai berikut:</p> <p>Tanggal: "2021-01-01"</p> <p>Jadwal Subuh "04:15"</p> <p>Jadwal Zuhur: "11:43"</p> <p>Jadwal Asar:" 14:49"</p> <p>Jadwal Maghrib:" 17:52"</p> <p>Jadwal Isya:" 19:03"</p> <p>4. Menekan tombol simpan</p>
Hasil yang diharapkan	Sistem mengarahkan ke halaman jadwal salat dan menampilkan jadwal salat yang sudah dibuat
Hasil yang didapatkan	Sistem mengarahkan ke halaman jadwal salat dan menampilkan jadwal salat yang sudah dibuat
Status validasi	Valid

Tabel 6.53 Kasus uji tambah jadwal salat jalur alternatif 5a

Kode kebutuhan	SIMAGFARI _F_20
Nama kasus uji	Tambah jadwal salat jalur alternatif 5a
Prosedur	<p>1. Mengakses halaman jadwal salat</p> <p>2. Menekan tombol tambah data</p> <p>3. Mengisi form tambah jadwal salat dengan data sebagai berikut:</p> <p>Jadwal Subuh: ""</p> <p>Jadwal Subuh ""</p> <p>Jadwal Zuhur: ""</p> <p>Jadwal Asar:""</p> <p>Jadwal Maghrib:""</p> <p>Jadwal Isya:""</p> <p>4. Menekan tombol simpan</p>
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.21 Pengujian validasi ubah jadwal salat

Kasus uji ubah jadwal salat jalur utama ditunjukkan pada Tabel 6.54 dan kasus uji ubah jadwal salat jalur *alternative* 4a ditunjukkan pada Tabel 6.55.

Tabel 6.54 Kasus uji ubah jadwal salat jalur utama

Kode kebutuhan	SIMAGFARI_F_21
Nama kasus uji	Ubah jadwal salat jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman jadwal salat 2. Menekan tombol ubah jadwal salat 3. Mengisi form ubah jadwal salat dengan data sebagai berikut: Tanggal: "2021-01-05" Jadwal Subuh "04:20" Jadwal Zuhur: "11:40" Jadwal Asar:" 14:50" Jadwal Maghrib:" 17:50" Jadwal Isya:" 19:05" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman jadwal salat dan menampilkan jadwal salat yang sudah diubah
Hasil yang didapatkan	Sistem mengarahkan ke halaman jadwal salat dan menampilkan jadwal salat yang sudah diubah
Status validasi	Valid

Tabel 6.55 Kasus uji ubah jadwal salat jalur alternatif 4a

Kode kebutuhan	SIMAGFARI_F_21
Nama kasus uji	Ubah jadwal salat jalur alternatif 4a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman jadwa salat 2. Menekan tombol ubah jadwal salat 3. Mengisi form ubah jadwal salat dengan data sebagai berikut: Tanggal: "" Jadwal Subuh ""

	Jadwal Zuhur: "" Jadwal Asar: "" Jadwal Maghrib: "" Jadwal Isya: "" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.22 Pengujian validasi hapus jadwal salat

Kasus uji hapus jadwal salat jalur utama ditunjukkan pada Tabel 6.56 dan kasus uji hapus jadwal salat jalur *alternative* 3a ditunjukkan pada Tabel 6.57.

Tabel 6.56 Kasus uji hapus jadwal salat jalur utama

Kode kebutuhan	SIMAGFARI_F_22
Nama kasus uji	Hapus jadwal salat jalur utama
Prosedur	1. Mengakses halaman jadwal salat 2. Memilih jadwal salat yang akan dihapus 3. Menekan tombol hapus jadwal salat 4. Sistem menampilkan pesan "Apakah anda yakin menghapus jadwal salat ini" 5. Mengklik "Ya, saya yakin" pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Sistem menampilkan pesan "Data Terhapus" dan menampilkan halaman jadwal salat
Hasil yang didapatkan	Sistem menampilkan pesan "Data Terhapus" dan menampilkan halaman jadwal salat
Status validasi	Valid

Tabel 6.57 Kasus uji hapus jadwal salat jalur alternatif 3a

Kode kebutuhan	SIMAGFARI_F_22
Nama kasus uji	Hapus jadwal salat jalur alternatif 3a
Prosedur	1. Mengakses halaman jadwal salat 2. Memilih imam salat yang akan dihapus 3. Menekan tombol hapus jadwal salat

	4. Sistem menampilkan pesan “Apakah anda yakin menghapus jadwal salat ini” 5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Pengumuman tidak terhapus dan sistem menampilkan halaman jadwal salat
Hasil yang didapatkan	Pengumuman tidak terhapus dan sistem menampilkan halaman jadwal salat
Status validasi	Valid

6.3.23 Pengujian validasi tambah keuangan harian

Kasus uji tambah keuangan harian jalur utama ditunjukkan pada Tabel 6.58 dan kasus uji tambah keuangan harian jalur *alternative* 5a ditunjukkan pada Tabel 6.59.

Tabel 6.58 Kasus uji tambah keuangan harian jalur utama

Kode kebutuhan	SIMAGFARI_F_23
Nama kasus uji	Tambah keuangan harian jalur utama
Prosedur	<ol style="list-style-type: none"> Mengakses halaman keuangan harian Menekan tombol tambah data Mengisi form tambah keuangan harian dengan data sebagai berikut: Keterangan: “Sumbangan Masjid” Tanggal : “2021-01-01” Debit: “50000” Kredit: “-” Kelompok transaksi: “Pemasukan nonrutin” Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman keuangan harian dan menampilkan keuangan harian yang sudah dibuat
Hasil yang didapatkan	Sistem mengarahkan ke halaman keuangan harian dan menampilkan keuangan harian yang sudah dibuat
Status validasi	Valid

Tabel 6.59 Kasus uji tambah keuangan harian jalur alternatif 5a

Kode kebutuhan	SIMAGFARI_F_23
Nama kasus uji	Tambah keuangan harian jalur alternatif 5a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Menekan tombol tambah data 3. Mengisi form tambah keuangan harian dengan data sebagai berikut: Keterangan:"" Tanggal : "" Debit:"" Kredit:"" Kelompok transaksi:"" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.24 Pengujian validasi ubah keuangan harian

Kasus uji ubah keuangan harian jalur utama ditunjukkan pada Tabel 6.60 dan kasus uji ubah keuangan harian jalur *alternative* 4a ditunjukkan pada Tabel 6.61.

Tabel 6.60 Kasus uji ubah keuangan harian jalur utama

Kode kebutuhan	SIMAGFARI_F_24
Nama kasus uji	Ubah keuangan harian jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Menekan tombol ubah keuangan harian 3. Mengisi form ubah keuangan harian dengan data sebagai berikut: Keterangan:"Sumbangan Masjid" Tanggal : "2021-01-05" Debit:"50000" Kredit:"-"

	4. Menekan tombol simpan
Hasil yang diharapkan	Sistem mengarahkan ke halaman keuangan harian dan menampilkan keuangan harian yang sudah diubah
Hasil yang didapatkan	Sistem mengarahkan ke halaman keuangan harian dan menampilkan keuangan harian yang sudah diubah
Status validasi	Valid

Tabel 6.61 Kasus uji ubah keuangan harian jalur alternatif 4a

Kode kebutuhan	SIMAGFARI_F_24
Nama kasus uji	Ubah keuangan harian jalur alternatif 4a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Menekan tombol ubah keuangan harian 3. Mengisi form ubah keuangan harian dengan data sebagai berikut: Keterangan: "" Tanggal : "" Debit: "" Kredit: "" 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan "field is required"
Hasil yang didapatkan	Sistem menampilkan pesan "field is required"
Status validasi	Valid

6.3.25 Pengujian validasi hapus keuangan harian

Kasus uji hapus keuangan harian jalur utama ditunjukkan pada Tabel 6.62 dan kasus uji hapus keuangan harian jalur *alternative* 3a ditunjukkan pada Tabel 6.63.

Tabel 6.62 Kasus uji hapus keuangan harian jalur utama

Kode kebutuhan	SIMAGFARI_F_25
Nama kasus uji	Hapus keuangan harian jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Memilih keuangan harian yang akan dihapus

	<ol style="list-style-type: none"> 3. Menekan tombol hapus keuangan harian 4. Sistem menampilkan pesan “Apakah anda yakin menghapus keuangan harian ini” 5. Mengklik “Ya, saya yakin” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Laporan keuangan harian terhapus dan sistem menampilkan halaman keuangan harian
Hasil yang didapatkan	Laporan keuangan harian terhapus dan sistem menampilkan halaman keuangan harian
Status validasi	Valid

Tabel 6.63 Kasus uji hapus keuangan harian jalur alternatif 3a

Kode kebutuhan	SIMAGFARI_F_25
Nama kasus uji	Hapus keuangan harian jalur alternatif 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Memilih keuangan harian yang akan dihapus 3. Menekan tombol hapus keuangan harian 4. Sistem menampilkan pesan “Apakah anda yakin menghapus keuangan harian ini” 5. Mengklik “Batalkan” pada pesan konfirmasi penghapusan
Hasil yang diharapkan	Pengumuman tidak terhapus dan sistem menampilkan halaman keuangan harian
Hasil yang didapatkan	Pengumuman tidak terhapus dan sistem menampilkan halaman keuangan harian
Status validasi	Valid

6.3.26 Pengujian validasi cetak keuangan harian

Kasus uji cetak keuangan harian jalur utama ditunjukkan pada Tabel 6.64 dan kasus uji cetak keuangan harian jalur *alternative* 3a ditunjukkan pada Tabel 6.64.

Tabel 6.64 Kasus uji mencetak laporan keuangan harian jalur utama

Kode kebutuhan	SIMAGFARI_F_26
Nama kasus uji	Mencetak laporan keuangan harian jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian

	<ol style="list-style-type: none"> 2. Menampilkan halaman laporan keuangan harian yang berisi menu cetak laporan keuangan 3. Menekan menu cetak laporan keuangan harian 4. Sistem menampilkan laporan keuangan harian dan memunculkan pilihan “Simpan” atau “Batalkan” 5. Memilih menu “Simpan” pada pesan konfirmasi
Hasil yang diharapkan	Laporan keuangan harian berhasil dicetak
Hasil yang didapatkan	Laporan keuangan harian berhasil dicetak
Status validasi	Valid

Tabel 6.65 Kasus uji mencetak laporan keuangan harian jalur alternative 3a

Kode kebutuhan	SIMAGFARI_F_26
Nama kasus uji	Mencetak laporan keuangan harian jalur alternatif 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Menampilkan halaman laporan keuangan harian yang berisi menu cetak laporan keuangan 3. Menekan menu cetak laporan keuangan harian 4. Sistem menampilkan laporan keuangan harian dan memunculkan pilihan “Simpan” atau “Batalkan” 5. Memilih menu “Batalkan” pada pesan konfirmasi
Hasil yang diharapkan	Laporan keuangan harian tidak berhasil dicetak
Hasil yang didapatkan	Laporan keuangan harian tidak berhasil dicetak
Status validasi	Valid

6.3.27 Pengujian validasi cetak laporan keuangan bulanan

Kasus uji cetak keuangan bulanan jalur utama ditunjukkan pada Tabel 6.66 dan kasus uji cetak keuangan bulanan jalur *alternative* 3a ditunjukkan pada Tabel 6.67.

Tabel 6.66 Kasus uji mencetak laporan keuangan bulanan jalur utama

Kode kebutuhan	SIMAGFARI_F_30
Nama kasus uji	Mencetak laporan keuangan bulanan jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan bulanan

	<ol style="list-style-type: none"> 2. Menampilkan halaman laporan keuangan bulanan yang berisi menu cetak laporan keuangan 3. Menekan menu cetak laporan keuangan bulanan 4. Sistem menampilkan laporan keuangan bulanan dan memunculkan pilihan “Simpan” atau “Batalkan” 5. Memilih menu “Simpan” pada pesan konfirmasi
Hasil yang diharapkan	Laporan keuangan bulanan berhasil dicetak
Hasil yang didapatkan	Laporan keuangan bulanan berhasil dicetak
Status validasi	Valid

Tabel 6.67 Kasus uji mencetak laporan keuangan bulanan jalur alternative 3a

Kode kebutuhan	SIMAGFARI_F_30
Nama kasus uji	Mencetak laporan keuangan harian jalur alternatif 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan bulanan 2. Menampilkan halaman laporan keuangan bulanan yang berisi menu cetak laporan keuangan 3. Menekan menu cetak laporan keuangan bulanan 4. Sistem menampilkan laporan keuangan bulanan dan memunculkan pilihan “Simpan” atau “Batalkan” 5. Memilih menu “Batalkan” pada pesan konfirmasi
Hasil yang diharapkan	Laporan keuangan bulanan tidak berhasil dicetak
Hasil yang didapatkan	Laporan keuangan bulanan tidak berhasil dicetak
Status validasi	Valid

6.3.28 Pengujian validasi lihat detail dan daftar kegiatan insidental jalur utama

Kasus uji lihat detail dan daftar kegiatan insidental jalur utama ditunjukkan pada Tabel 6.68 dan kasus uji lihat detail dan daftar kegiatan insidental jalur *alternative 3a* ditunjukkan pada Tabel 6.69.

Tabel 6.68 Kasus uji melihat detail dan daftar kegiatan insidental jalur utama

Kode kebutuhan	SIMAGFARI_F_31
Nama kasus uji	Melihat detail dan daftar kegiatan insidental jalur utama

Prosedur	1. Mengakses halaman kegiatan 2. Mengakses halaman kegiatan insidental 3. Menampilkan detail dan daftar kegiatan insidental
Hasil yang diharapkan	Menampilkan detail dan daftar kegiatan insidental
Hasil yang didapatkan	Menampilkan detail dan daftar kegiatan insidental
Status validasi	Valid

Tabel 6.69 Kasus uji melihat detail dan daftar kegiatan insidental jalur 3a

Kode kebutuhan	SIMAGFARI_F_31
Nama kasus uji	Melihat detail dan daftar kegiatan insidental jalur 3a
Prosedur	1. Mengakses halaman kegiatan 2. Mengakses halaman kegiatan insidental 3. Menampilkan halaman kegiatan insidental yang kosong
Hasil yang diharapkan	Tidak menampilkan detail dan daftar kegiatan insidental
Hasil yang didapatkan	Tidak menampilkan detail dan daftar kegiatan insidental
Status validasi	Valid

6.3.29 Pengujian validasi lihat detail dan daftar kegiatan rutin jalur utama

Kasus uji lihat detail dan daftar kegiatan rutin jalur utama ditunjukkan oleh Tabel 6.70 dan kasus uji lihat detail dan daftar kegiatan rutin jalur *alternative* 3a ditunjukkan pada Tabel 6.71.

Tabel 6.70 Kasus uji melihat detail dan daftar kegiatan rutin jalur utama

Kode kebutuhan	SIMAGFARI_F_32
Nama kasus uji	Melihat detail dan daftar kegiatan rutin jalur utama
Prosedur	1. Mengakses halaman kegiatan 2. Mengakses halaman kegiatan rutin 3. Menampilkan detail dan daftar kegiatan rutin
Hasil yang diharapkan	Menampilkan detail dan daftar kegiatan rutin
Hasil yang didapatkan	Menampilkan detail dan daftar kegiatan valid

Status validasi	Valid
-----------------	-------

Tabel 6.71 Kasus uji melihat detail dan daftar kegiatan rutin jalur 3a

Kode kebutuhan	SIMAGFARI_F_32
Nama kasus uji	Melihat detail dan daftar kegiatan rutin jalur 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman kegiatan 2. Mengakses halaman kegiatan rutin 3. Menampilkan halaman kegiatan rutin yang kosong
Hasil yang diharapkan	Tidak menampilkan detail dan daftar kegiatan rutin
Hasil yang didapatkan	Tidak menampilkan detail dan daftar kegiatan rutin
Status validasi	Valid

6.3.30 Pengujian validasi lihat detail dan daftar pengumuman jalur utama

Kasus uji lihat detail dan daftar pengumuman jalur utama ditunjukkan pada Tabel 6.72 dan kasus uji lihat detail dan daftar pengumuman jalur *alternative* 3a ditunjukkan pada Tabel 6.73.

Tabel 6.72 Kasus uji melihat detail dan daftar pengumuman jalur utama

Kode kebutuhan	SIMAGFARI_F_33
Nama kasus uji	Melihat detail dan daftar pengumuman jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Menampilkan halaman pengumuman 3. Menampilkan detail dan daftar pengumuman
Hasil yang diharapkan	Menampilkan detail dan daftar kegiatan insidental
Hasil yang didapatkan	Menampilkan detail dan daftar kegiatan insidental
Status validasi	Valid

Tabel 6.73 Kasus uji Melihat detail dan daftar pengumuman jalur 3a

Kode kebutuhan	SIMAGFARI_F_33
Nama kasus uji	Melihat detail dan daftar pengumuman jalur 3a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman pengumuman 2. Mengakses halaman pengumuman

	3. Menampilkan halaman pengumuman yang kosong
Hasil yang diharapkan	Tidak menampilkan detail dan daftar pengumuman
Hasil yang didapatkan	Tidak menampilkan detail dan daftar pengumuman
Status validasi	Valid

6.3.31 Pengujian validasi lihat detail dan daftar artikel jalur utama

Kasus uji lihat detail dan daftar artikel jalur utama ditunjukkan pada Tabel 6.74 dan kasus uji lihat detail dan daftar artikel jalur *alternative* 3a ditunjukkan pada Tabel 6.75.

Tabel 6.74 Kasus uji melihat detail dan daftar artikel jalur utama

Kode kebutuhan	SIMAGFARI_F_34
Nama kasus uji	Melihat detail dan daftar artikel jalur utama
Prosedur	1. Mengakses halaman artikel 2. Menampilkan halaman artikel 3. Menampilkan detail dan daftar artikel
Hasil yang diharapkan	Menampilkan detail dan daftar artikel
Hasil yang didapatkan	Menampilkan detail dan daftar artikel
Status validasi	Valid

Tabel 6.75 Kasus uji melihat detail dan daftar artikel jalur 3a

Kode kebutuhan	SIMAGFARI_F_34
Nama kasus uji	Melihat detail dan daftar artikel jalur 3a
Prosedur	1. Mengakses halaman artikel 2. Mengakses halaman artikel 3. Menampilkan halaman artikel yang kosong
Hasil yang diharapkan	Tidak menampilkan detail dan daftar artikel
Hasil yang didapatkan	Tidak menampilkan detail dan daftar artikel
Status validasi	Valid

6.3.32 Pengujian validasi lihat daftar foto jalur utama

Kasus uji lihat daftar foto jalur utama ditunjukkan pada Tabel 6.76 dan kasus uji lihat daftar foto jalur *alternative* 2a ditunjukkan pada Tabel 6.77.

Tabel 6.76 Kasus uji melihat daftar foto jalur utama

Kode kebutuhan	SIMAGFARI_F_35
Nama kasus uji	Melihat daftar foto jalur utama
Prosedur	1. Mengakses halaman foto 2. Menampilkan halaman foto 3. Menampilkan daftar foto
Hasil yang diharapkan	Menampilkan daftar foto
Hasil yang didapatkan	Menampilkan daftar foto
Status validasi	Valid

Tabel 6.77 Kasus uji melihat daftar foto jalur 2a

Kode kebutuhan	SIMAGFARI_F_35
Nama kasus uji	Melihat daftar foto jalur 3a
Prosedur	1. Mengakses halaman foto 2. Menampilkan halaman foto yang kosong
Hasil yang diharapkan	Tidak menampilkan daftar foto
Hasil yang didapatkan	Tidak menampilkan daftar foto
Status validasi	Valid

6.3.33 Pengujian validasi lihat daftar imam salat jalur utama

Kasus uji lihat daftar imam salat jalur utama ditunjukkan pada Tabel 6.78 dan kasus uji lihat daftar imam salat jalur *alternative* 2a ditunjukkan pada Tabel 6.79.

Tabel 6.78 Kasus uji melihat daftar imam jalur utama

Kode kebutuhan	SIMAGFARI_F_36
Nama kasus uji	Melihat daftar imam salat jalur utama
Prosedur	1. Mengakses halaman imam salat 2. Menampilkan halaman imam salat 3. Menampilkan daftar imam salat dalam bentuk tabel yang terdiri dari tanggal, imam Subuh, imam Zuhur, imam Asar, imam Magrib, dan imam Isya
Hasil yang diharapkan	Menampilkan daftar imam salat
Hasil yang didapatkan	Menampilkan daftar imam salat

Status validasi	Valid
-----------------	-------

Tabel 6.79 Kasus uji melihat daftar imam salat jalur 2a

Kode kebutuhan	SIMAGFARI_F_36
Nama kasus uji	Melihat daftar imam salat jalur 2a
Prosedur	1. Mengakses halaman imam salat 2. Menampilkan halaman imam salat yang kosong
Hasil yang diharapkan	Tidak menampilkan daftar imam salat
Hasil yang didapatkan	Tidak menampilkan daftar imam salat
Status validasi	Valid

6.3.34 Pengujian validasi lihat jadwal salat jalur utama

Kasus uji lihat jadwal saat jalur utama ditunjukkan pada Tabel 6.80 dan kasus uji lihat daftar jadwal salat jalur *alternative* 2a ditunjukkan pada Tabel 6.81.

Tabel 6.80 Kasus uji melihat daftar jadwal salat jalur utama

Kode kebutuhan	SIMAGFARI_F_37
Nama kasus uji	Melihat daftar jadwal salat jalur utama
Prosedur	1. Mengakses halaman jadwal salat 2. Menampilkan halaman jadwal salat 3. Menampilkan daftar jadwal salat dalam bentuk tabel yang terdiri dari tanggal, jadwal Subuh, jadwal Zuhur, jadwal Asar, jadwal Magrib, dan jadwal Isya
Hasil yang diharapkan	Menampilkan daftar jadwal salat
Hasil yang didapatkan	Menampilkan daftar jadwal salat
Status validasi	Valid

Tabel 6.81 Kasus uji melihat daftar jadwal salat jalur 2a

Kode kebutuhan	SIMAGFARI_F_37
Nama kasus uji	Melihat daftar jadwal salat jalur 2a
Prosedur	1. Mengakses halaman jadwal salat 2. Menampilkan halaman jadwal salat yang kosong
Hasil yang diharapkan	Tidak menampilkan daftar jadwal salat

Hasil yang didapatkan	Tidak menampilkan daftar jadwal salat
Status validasi	Valid

6.3.35 Pengujian validasi lihat laporan keuangan harian jalur utama

Kasus uji lihat keuangan harian jalur utama ditunjukkan pada Tabel 6.82 dan kasus uji lihat keuangan harian jalur *alternative* 2a ditunjukkan pada Tabel 6.83.

Tabel 6.82 Kasus uji melihat laporan keuangan harian jalur utama

Kode kebutuhan	SIMAGFARI_F_38
Nama kasus uji	Melihat lihat laporan keuangan harian jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Menampilkan halaman keuangan harian terdapat filter keuangan yang terdiri dari kolom pilihan tanggal awal dan kolom tanggal akhir 3. Memilih rentang tanggal, bulan, dan tahun laporan keuangan yang ingin ditampilkan 4. Menampilkan detail laporan keuangan harian yang terdiri dari tabel keterangan transaksi, tanggal transaksi, pemasukan, dan pengeluaran berdasarkan rentang tanggal yang dipilih
Hasil yang diharapkan	Menampilkan laporan keuangan harian
Hasil yang didapatkan	Menampilkan laporan keuangan harian
Status validasi	Valid

Tabel 6.83 Kasus uji melihat laporan keuangan harian jalur 2a

Kode kebutuhan	SIMAGFARI_F_38
Nama kasus uji	Melihat lihat laporan keuangan harian jalur 2a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan harian 2. Menampilkan halaman keuangan harian yang kosong
Hasil yang diharapkan	Tidak menampilkan laporan keuangan harian
Hasil yang didapatkan	Tidak Menampilkan laporan keuangan harian
Status validasi	Valid

6.3.36 Pengujian validasi lihat laporan keuangan bulanan jalur utama

Kasus lihat keuangan bulanan jalur utama ditunjukkan pada Tabel 6.84 dan kasus uji lihat keuangan bulanan jalur *alternative 2a* ditunjukkan pada Tabel 6.85.

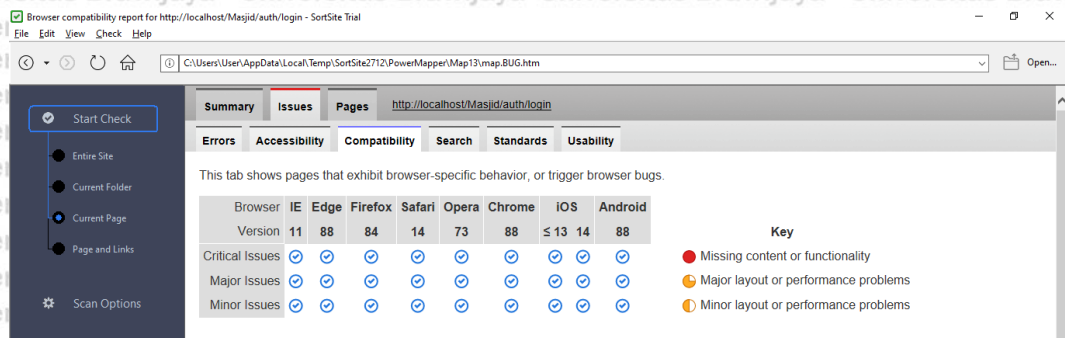
Tabel 6.84 Kasus uji melihat laporan keuangan bulanan jalur utama

Kode kebutuhan	SIMAGFARI_F_39
Nama kasus uji	Melihat lihat laporan keuangan bulanan jalur utama
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan bulanan 2. Menampilkan halaman keuangan bulanan dan terdapat filter keuangan yang terdiri dari kolom pilihan bulan dan kolom tahun 3. Memilih rentang bulan dan tahun laporan keuangan yang ingin ditampilkan sebagai berikut: <ul style="list-style-type: none"> - Kolom Bulan: "Juli" - Kolom Tahun: "2020" 4. Menampilkan detail laporan keuangan bulanan yang terdiri dari tabel keterangan transaksi, tanggal transaksi, pemasukan, pengeluaran, serta kelompok transaksi berdasarkan rentang bulan dan tahun yang dipilih
Hasil yang diharapkan	Menampilkan laporan keuangan bulanan
Hasil yang didapatkan	Menampilkan laporan keuangan bulanan
Status validasi	Valid

Tabel 6.85 Kasus uji melihat laporan keuangan bulanan jalur 2a

Kode kebutuhan	SIMAGFARI_F_39
Nama kasus uji	Melihat lihat laporan keuangan bulanan jalur 2a
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman keuangan bulanan 2. Menampilkan halaman keuangan bulanan yang kosong
Hasil yang diharapkan	Tidak menampilkan laporan keuangan bulanan
Hasil yang didapatkan	Tidak Menampilkan laporan keuangan bulanan
Status validasi	Valid

6.3.37 Pengujian *Compatibility*



Gambar 6.7 Hasil Pengujian *Compatibility* Menggunakan Aplikasi *SortSite*

Gambar 6.7 merupakan hasil dari pengujian *compatibility* dari aplikasi *SortSite*. Hasil pengujian *compatibility* menunjukkan sistem informasi manajemen Masjid Al Ghifari dapat dijalankan diberbagai jenis *browser* yang terdiri dari *Firefox*, *Opera*, *Chrome*, dan lain-lain.



BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

1. Tahap analisis kebutuhan menghasilkan 4 aktor dan 39 kebutuhan fungsional serta 1 kebutuhan nonfungsional. Aktor tersebut terdiri dari Ketua Pengurus Masjid, Bendahara, Anggota Bidang Ubudiyah, dan Pengguna. Kebutuhan fungsional utama pada penelitian ini meliputi Tambah kegiatan insidental, Ubah kegiatan insidental, dan Tambah keuangan harian.
2. Tahap perancangan menghasilkan pemodelan *sequence diagram* dan *class diagram* yang berasal dari hasil perancangan arsitektur. ERD dan skema *database* didapatkan dari perancangan data. *Pseudocode* didapatkan dari hasil perancangan komponen serta rancangan antarmuka didapatkan dari hasil perancangan antarmuka.
3. Tahap implementasi menghasilkan implementasi kode program, basis data, dan antarmuka. Implementasi kode program menggunakan bahasa pemrograman PHP dan *framework Codeigniter* yang didapatkan dari hasil perancangan komponen. Implementasi basis data menggunakan MySQL yang didapatkan dari skema *database* dan implementasi antarmuka didapatkan berdasarkan hasil perancangan antarmuka.
4. Tahap pengujian dilakukan pada level unit, integrasi, dan validasi. Pengujian unit dan integrasi menggunakan teknik pengujian *whitebox* dengan metode *basis path testing*. Pengujian unit dan integrasi menggunakan tiga sampel pengujian yang didapatkan dari hasil perancangan komponen dan menghasilkan nilai valid. Pengujian validasi menggunakan teknik pengujian *blackbox* dengan metode *scenario based testing*. Pengujian validasi menguji 39 kebutuhan fungsional yang menghasilkan nilai valid. Pengujian *compatibility* dilakukan untuk menguji kebutuhan nonfungsional dengan menggunakan bantuan *software SortSiteEval* untuk menguji perangkat lunak dapat dijalankan diberbagai macam *browser* seperti *Mozilla*, *Opera*, dan *Google Chrome*.

7.2 Saran

Saran terkait untuk pengembangan sistem informasi manajemen Masjid Al Ghifari berikutnya, yaitu menambahkan fitur perhitungan zakat serta dan fitur membuat grafik dalam perhitungan pemasukan dan pengeluaran keuangan masjid setiap bulan untuk dapat memudahkan Bendahara dalam membuat laporan bulanan.

DAFTAR REFERENSI

- Bassil, Y., 2012. A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (IJET)*.
- Betha Sidik., 2012, Pemrograman Web dengan PHP, Informatika, Bandung.
- Birgi, J. S., Khaire, M., & Hira, S., 2016. *Data Model: A Blueprint for Data Warehouse*. International Journal of Scientific and Research Publications – Vol. 6(1), January 2016.
- Blanco, J. A. & Upton, D., 2009. Codeigniter 1.7. [pdf] Birmingham: Packt Publishing. Tersedia di: <https://thoni.files.wordpress.com/2011/11/codeigniter.pdf?ved=2ahUKEwicr9XP5uPhAhWd6XMBHRgCNoQFjAGegQICBAB&usq=AOvVaw3E7lhQtg032qQhiNpTH_R&csid=1555939707968> [Diakses 19 Januari 2019]
- Brigham, E. F., & Houston, J. F. 2017. *Fundamentals of Financial Management: Concise (9th ed.)*. Boston: Cengage Learning.
- Date, C. J., 2010. An Introduction to Database Systems 8th Edition. [pdf] USA: Dartmouth Publishing, Inc. Tersedia di: <<https://lc.fie.umich.mx/~rodrigo/BD/An%20Introduction%20to%20Databases%20Systems%20e%20By%20C%20J%20Date.pdf>> [Diakses 21 Mei 2019]
- Destiningrum, Mara., 2017. *Sistem Informasi Penjadwalan Dokter Berbasis Web dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)*. Jurnal TEKNOINFO, 11, 30–77.
- Hasibuan Malayu S.P. 2014. *Manajemen Dasar, Pengertian dan Masalah*. Bumi Aksara: Jakarta.
- Hustinawati, Himawan, A. K. & Latifah, 2014. *Performance Analysis Framework Codeigniter and Cake Php in Website Creation*. Internasional Journal of Computer Applications, Volume 94, p. 6.
- Kristanto, Dodyk., 2018. *Sistem Informasi Pengelolaan Masjid Ibaadurrahman Berbasis Web*. Surakarta: Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta.
- Kurniawan, T. A., 2018. *Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan dalam Praktik*. JTIK, 5, pp.77–86.
- Marsic, I., 2012. *Software Engineering*. New Jersey: Rutgers University.
- Nainggolan, P. 2012. *Manajemen Keuangan Lembaga Nirlaba*. Jakarta: Yayasan Bina Integrasi Edukasi.
- Pressman, R.S., 2010. *Software Engineering a Practitioner's Approach. The McGraw-Hill Companies* : New York.
- Pressman, R. S. & Maxim, B. R., 2015. *Software engineering : a practitioner's approach*. 8 ed. New York: McGraw-Hill.

Putra, Reyanda Dwi Y., 2017. *Perancangan Sistem Informasi Manajemen Masjid Baiturrahim*. Surakarta: Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta.

Silalahi, M., Wahyudi, D., 2018. *Perbandingan Performansi Database MongoDB dan MySQL dalam Aplikasi File Multimedia Berbasis Web*. Computer Based Informasion System Journal, Vol. 06, No. 01, Maret 2018

Sommerville, I., 2011. *Software Engineering - 9th Ed.*. Boston, Massachusetts: Pearson Education.

Visual Paradigm, 2018a. Class Diagram. Tersedia di: <https://www.visualparadigm.com/VPGallery/diagrams/Class.html>

Visual Paradigm, 2018b. Sequence Diagram. Tersedia di <https://www.visualparadigm.com/VPGallery/diagrams/Sequence.html>

Visual Paradigm, 2018c. Uml Use Case Diagram Notations Guide. Tersedia di https://www.visualparadigm.com/support/documents/vpuserguide/94/2575/84257_usecasediagr.html

Visual Paradigm, 2020. How to Model MVC Frameork with UML Sequence Diagram. Tersedia di <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/how-to-model-mvc-with-uml-sequence-diagram/>

Wandana, T.I., & Ariwibowo. E. 2013. *Perancangan Dan Implementasi Sistem Manajemen Kegiatan Masjid Studi Kasus: Masjid Jogokariyan*. Yogyakarta: Jurnal Sarjana Teknik Informatika.

LAMPIRAN

Lampiran A Wawancara

Narasumber : Bapak Agus Junaedi (Ketua Bidang Ketatalaksanaan Rumah Tangga & Pemeliharaan Sarana dan Bangunan)

Hari, tanggal : 24 Juli 2019

Tempat : Masjid Al Ghifari Kota Malang

Pewawancara : Muhammad Rezky Sayuthi Putra

1. Apa saja kegiatan yang ada di Masjid Al Ghifari

Jawaban: Kegiatan yang ada di Masjid Al Ghifari terbagi menjadi 2 yaitu kegiatan rutin dan kegiatan insidental.

2. Perbedaan dari kegiatan rutin dan kegiatan insidental?

Jawaban: Kegiatan rutin merupakan kegiatan yang sering diadakan ataupun dilakukan oleh Masjid Al Ghifari, sedangkan kegiatan insidental merupakan kegiatan yang diadakan dalam waktu tertentu.

3. Apa-apa saja kegiatan rutin dan kegiatan insidental yang ada di Masjid AL Ghifari?

Jawaban:

- Kegiatan rutin:

1. Kajian rutin yang dilakukan setiap hari setelah sholat Maghrib dan sholat Subuh
2. Kajian muslimah untuk ibu-ibu yang dilakukan setiap hari Sabtu setelah sholat Asar
3. Buka Puasa setiap hari Senin dan Kamis
4. Pesantren Mahasiswa
5. Kajian pernikahan yang dilakukan sebulan dua kali setiap hari Sabtu di pekan 1 & 3 setelah sholat Asar
6. Olahraga dojo dilaksanakan 2 kali, untuk atlit dilakukan setelah sholat Isya jam 7-9 malam, serta untuk umum dilakukan pada hari Sabtu dan Minggu jam 7-9 pagi.

7. Olahraga memanah dilakukan pada hari Minggu jam 6-9 pagi.
8. Olahraga berkuda dilakukan sebulan sekali.
9. Olahraga futsal dan bulutangkis dilakukan setiap 2 hari sekali dalam seminggu setelah sholat Asar.
10. Kelas tahsin Qur'an, bahasa arab, dan sironabawiyah yang dilakukan setiap hari setelah sholat Isya.
11. Sedekah Jumat dilakukan untuk menghidangkan makan siang setelah sholat Jumat dan dibuka untuk umum

Kegiatan insidental:

1. Kajian akbar mengundang ustaz dari luar malang dan biasanya dihadiri oleh 300-400 orang di Masjid Al Ghifari
2. Mualaf center untuk dapat membantu orang baru masuk agama Islam supaya dapat belajar dan memahami tentang agama Islam
3. Funeral service untuk dapat membantu keluarga yang muslim yang meninggal dari menyiapkan alat-alat mayat sampai membantu untuk menguburkannya
4. Bagaimana menginformasikan tentang kegiatan yang ada di Masjid Al Ghifari?

Jawaban: untuk menginformasikan kegiatan-kegiatan yang ada di Masjid Al Ghifari menggunakan sosial media *Instagram* dan *facebook* serta menempelkan poster kegiatan di papan pengumuman masjid.

5. Apakah orang umum dapat membuat usulan kegiatan di Masjid Al Ghifari dan bagaimana caranya?

Jawaban: Boleh, cara untuk membuat kegiatan di Masjid Al Ghifari antara lain:

1. Masyarakat atau pengurus masjid yang ingin membuat kegiatan di Masjid Al Ghifari diwajibkan untuk membuat surat permohonan kegiatan
2. Surat permohonan tersebut terdiri dari tema kegiatan, judul kegiatan, pemateri, dan tanggal kegiatan.
3. Pengurus masjid akan bermusyawarah untuk mengizinkan atau tidak

4. Pengurus masjid akan menolak suratnya jika acara tersebut tanggalnya bertrok dengan kegiatan yang sudah tercatat.

6. Bagaimana cara membuat laporan keuangan di Masjid Al Ghifari

Jawaban:

- Uang masuk dan keluar wajib dicatat oleh Bendahara di buku laporan keuangan harian

- Uang yang masuk dan keluar wajib juga harus ada notanya

- Setelah semua pemasukan dan pengeluaran dicatat oleh Bendahara dibuka laporan keuangan harian nanti setelah itu akan membuat laporan keuangan bulanan berdasarkan laporan keuangan harian yang sudah ditulis dibuka.

- Laporan bulanan dibuat menggunakan *Microsoft Excel* dan setelah laporan keuangan bulanan jadi laporan tersebut ditempel dipapan pengumuman masjid supaya masyarakat dapat mengetahui jumlah pemasukan dan pengeluaran masjid Al Ghifari.

7. Siapa saja yang dapat mengakses buku laporan keuangan harian tersebut?

Jawaban: Ketua Pengurus Masjid dan Bendahara

8. Apa kendala Ketua Pengurus Masjid dalam mengontrol dan memeriksa keuangan di Masjid Al Ghifari?

Jawaban: Ketua Pengurus Masjid kesulitan untuk dapat memeriksa

keuangan di Masjid Al Ghifari pertama Ketua Pengurus juga memiliki

kesibukan sebagai dokter dan harus membagi waktu sebagai dokter dan

Ketua pengurus masjid sehingga seringkali jika sewaktu-waktu Ketua

pengurus masjid ingin memeriksa keuangan masjid pada hari tertentu harus

membuat janji dengan Bendahara dikarenakan buku laporan keuangan

tersebut di pegang oleh Bendahara.

9. Apa saja kendala yang dialami oleh Bendahara dalam membuat laporan keuangan?

Jawaban: Kendala yang dialami oleh Bendahara dalam membuat laporan

keuangan pernah beberapa tahun yang lalu Bendahara lupa menyimpan

buku catatan keuangan masjid sehingga laporan keuangan terlambat dibuat dikarenakan laporan bulanan yang dibuat Bendahara berdasarkan seluruh catatan laporan keuangan yang berada di dalam buku tersebut.

LAMPIRAN B LAPORAN KEUANGAN BULANAN MASJID AL GHIFARI



Lembaga Dakwah Pendidikan dan Sosial (LDPS)
ABU DZAR AL-GHIFARI
 Memperkokoh Aqidah Memperkuat Ukhuwah
 Jl. Soekarno-Hatta (Griyashanta) Blok E No. 219 A Mojolangu, Lowokwaru, Kota Malang
 Telp. 0822 4444 4282 | Email. ldps.alghifari.malang@gmail.com

LAPORAN KEUANGAN
 November 2019

Nama Rekening	Debet	Kredit
Pemasukan Rutin		
Sisa Saldo Oktober 2019	64,633,492 + 3 Dirham + US\$ 2.600	
Infaq Jumat	38,840,000	
Infaq Umum	4,258,390	
Infaq GIFT	7,332,392	
Infaq Nasi Jumat (Sedekah Jumat)		
Infaq TPQ Jundulah	200,000	
Infaq PESMA		
Kontribusi Al Ghifari Mart		
Kontribusi Al Ghifari Food		
Kontribusi Kuttab Al Fatih		
Infaq Parkir	2,127,000	
Pemasukan Non Rutin		
Infaq Renovasi Lantai	200,099	
Pengeluaran Rutin		
Biaya Kafalah Guru TPQ		2,750,000
Biaya Kafalah Guru PESMA		
Biaya Kafalah Petugas LDPS		11,600,000
Biaya Kafalah Ustadz		15,450,000
Biaya Air Listrik Telp Internet		3,922,900
Biaya Kebersihan		541,100
Biaya Puasa Senin Kamis (GIFT)		14,919,000
Biaya Sedekah Jumat		650,000
Biaya Muamalah		6,000,000
Biaya Operasional Ambulance		770,000
Biaya ATK Fotokopi		
		44,500
Biaya Lain		4,417,873
Pengeluaran Non Rutin		
Biaya Tukang Renov Lantai		1,620,000
Biaya Tukang Kanopi		12,500,000
Biaya Pengadaan Video		4,450,000

Capture		
Total	117,591,373	79,653,373
SALDO	37,956,000 + 3 Dirham + US\$ 2.600	



LAMPIRAN C LAPORAN KEUANGAN HARIAN MASJID AL GHIFARI

TGL	NO. CHEQ	URAIAN	DEBIT	KREDIT	SALDO
25/11		Payer Utang (Kaf Ut Acha)	500.000		2.623.700
25/11		Bayar Utang (Kaf Khatib)	300.000		2.323.700
25/11		Infag	500.000		2.823.700
25/11		Kaf Ut Acha	500.000		2.323.700
25/11		Infag	200.000		2.123.700
25/11		Donasi Sarungin utk Kaf Praga	200.000		2.323.700
25/11		Kunci P. Hamsa	98.500		2.224.200
25/11		Panitia Zakat	2.000.000		224.200
25/11		Infag LED	2.300.000		2.594.200
26/11		Kaf Uang x Khatib EO	300.000		2.294.200
26/11		Kaf SATPAM W. Gey	6.000.000		1.294.200
30/11		Uang Kotak Amal	7.490.000		8.684.200
30/11		KAF BUDS x PELUPESHA	5.800.000		2.884.200
30/11		KAF Khatib Jumat	700.000		2.184.200
30/11		BANKIRAN UT Ut. Lpanto	150.000		9.034.200
			14		
26/11		PATRYCASH	2.500.000		6.534.200
26/11		Mekamalan	2.500.000		4.034.200
26/11		Kaf Papi Kuning	450.000		3.584.200
26/11		CLCilan 10-4	300.000		3.284.200
26/11		Nony Khatib	200.000		3.084.200
26/11		P. n. sifkar x Banner	290.000		2.794.200
3/11		met	687.600		2.106.600
7/11		Kaf Khatib	300.000		1.806.600
8/11		Utang	1.204.000		602.600
8/11		PATRYCASH	500.000		102.600
9/11		Kunci P. Hamsa	50.000		52.600
9/11		Caf Tukang (Khatib)	1.000.000		1.002.600
10/11		Kaf Khatib	300.000		702.600
13/11		Payer Tukang	1.705.000		1.002.600
13/11		PATRYCASH	342.000		660.600
15/11		Bahan Tukang	500.000		160.600
17/11		Payer Tukang	600.000		160.600
17/11		Bahan Tukang	600.000		160.600
17/11		TIP P. Roldmad	2.300.000		-2.549.000
17/11		PATRYCASH	272.000		17.600

LAMPIRAN D ROADMAP SKRIPSI

No	Fitur	Milestone	Progress	Status
1	Laporan Keuangan			
	- Laporan keuangan harian		100%	Done
	- Filter tanggal laporan keuangan harian		100%	Done
	- Cetak laporan keuangan harian		100%	Done
	- Laporan keuangan bulanan		100%	Done
	- Filter berdasarkan bulan untuk laporan keuangan bulanan		100%	Done

	- Cetak Laporan keuangan bulanan		100%	Done
2	Kegiatan			
	- List kegiatan insidental		100%	Done
	- List kegiatan rutin		100%	Done
	- List foto-foto kegiatan		100%	Done
3	Berita			
	- List artikel		100%	Done
	- List pengumuman		100%	Done
4	Imam Salat			
	- List jadwal imam salat		100%	Done
5	Waktu Sholat			
	- List jadwal Sholat		100%	Done
		TOTAL PROGRESS	100%	On Progress

