

**PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN MASJID
NURUL IKHLAS KECAMATAN PADANG UTARA BERBASIS WEB**

LAPORAN TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Program Strata-I Pada
Departemen Sistem Informasi Fakultas Teknologi Informasi Universitas Andalas

Oleh :

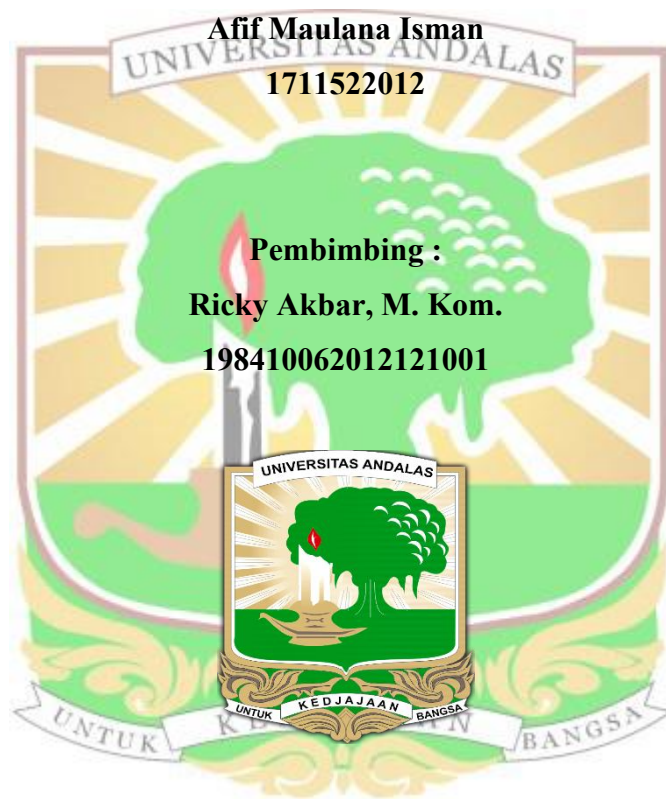
Afif Maulana Isman

1711522012

Pembimbing :

Ricky Akbar, M. Kom.

198410062012121001



**DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

PADANG

2023

HALAMAN PENGESAHAN

**PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN MASJID
NURUL IKHLAS KECAMATAN PADANG UTARA BERBASIS WEB**

Oleh :

Afif Maulana Isman

1711522012

LULUS SIDANG TUGAS AKHIR

14 September 2023

Padang, 14 November 2023

Telah diperiksa dan disetujui oleh

Pembimbing Tugas Akhir

Ricky Akbar, M. Kom.

NIP. 198410062012121001

Mengetahui,

Ketua Departemen Sistem Informasi

Husnil Kamil, M.T.

NIP. 198201182008121002

PERNYATAAN

Saya menyatakan bahwa laporan Tugas Akhir yang berjudul **“Pembangunan Sistem Informasi Pengelolaan Masjid Nurul Ikhlas Kecamatan Padang Utara Berbasis Web”** ini tidak terdapat karya yang pernah diajukan sebagai salah satu syarat menyelesaikan matakuliah Tugas Akhir di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Padang, 14 November 2023

Afif Maulana Isman



KATA PENGANTAR

Puji syukur penulis ucapkan atas kehadiran Allah SWT yang telah memberikan nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan pengerjaan proposal tugas akhir ini yang berjudul **“PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN MASJID NURUL IKHLAS KECAMATAN PADANG UTARA BERBASIS WEB”** dengan baik dan lancar.

Dalam pengerjaan tugas akhir ini, penulis banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Oleh sebab itu, penulis ingin Meyampaikan rasa hormat dan terima kasih kepada :

1. Bapak Husnil Kamil, M.T. sebagai ketua departemen sistem informasi.
2. Bapak Ricky Akbar, M.Kom. sebagai dosen pembimbing dalam tugas akhir ini.
3. Seluruh pihak yang terlibat dan yang telah membantu dan memberikan dukungan kepada penulis baik secara moril maupun materil dalam pembuatan proposal tugas akhir ini.

Penulis menyadari bahwa dalam penyusunan proposal tugas akhir ini masih terdapat kekurangan. Untuk itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari pembaca melalui email afif.maulana.isman@gmail.com agar penelitian selanjutnya dapat dilakukan lebih baik lagi.

Padang, September 2023

Penulis,

Afif Maulana Isman

DAFTAR ISI

PERNYATAAN	iii
KATA PENGANTAR	iv
DAFTAR ISI	iii
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	x
ABSTRAK	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1. Pengertian dan Fungsi Masjid	6
2.1.1. Pengertian Masjid	6
2.1.2. Fungsi Masjid Dalam Kehidupan Masyarakat	6
2.2. Pengertian Sistem Informasi	7
2.3. Aplikasi Web	7
2.4. Alat Analisis dan Perancangan Sistem	8
2.4.1. <i>Unified Modelling Language (UML)</i>	8
2.4.2. <i>Business Process Modelling Notation (BPMN)</i>	8
2.4.3. <i>Use Case Diagram</i>	10
2.4.4. Entity Relationship Diagram (ERD)	11
2.5. Perangkat Lunak yang digunakan	13

2.5.1.	Database MySQL	13
2.5.2.	<i>Framework</i>	14
2.5.3.	<i>Laravel</i>	15
2.5.4.	<i>Laragon</i>	15
BAB III METODE PENELITIAN		16
3.1.	Objek Penelitian	16
3.2.	Metode Pengumpulan Data	16
3.3.	Metode Pembangunan Aplikasi	17
3.4.	<i>Flowchart</i> Penelitian	18
BAB IV ANALISIS DAN PERANCANGAN SISTEM		21
4.1	Analisis Sistem	21
4.1.1	Sistem yang Sedang Berjalan	21
4.1.2	Sistem yang Diusulkan	25
4.1.3	Analisis Kebutuhan Fungsional	29
4.1.4	Use Case Diagram	30
4.1.5	Definisi Tugas Aktor	31
4.1.6	<i>Use Case Scenario</i>	34
4.1.6.1	<i>Use Case Scenario</i> Pencatatan Keuangan	35
4.1.6.2	<i>Use Case Scenario</i> Pendaftaran Santri	36
4.1.6.3	<i>Use Case Scenario</i> Pengajuan Kegiatan	36
4.1.7	<i>Sequence Diagram</i>	37
4.1.7.1	<i>Sequence Diagram</i> Pengajuan Kegiatan (Tambah)	38
4.1.7.2	<i>Sequence Diagram</i> Pencatatan Keuangan (Tambah)	39
4.1.7.3	<i>Sequence Diagram</i> Mendaftarkan Santri	40
4.1.8	<i>Class Analysis</i>	41
4.1.8.1	<i>Class Analysis</i> Pencatatan Keuangan (Tambah)	41
4.1.8.2	<i>Class Analysis</i> Pengajuan Kegiatan (Tambah)	41

4.1.8.3	<i>Class Analysis</i> Pendaftaran Santri	42
4.2	Perancangan Sistem	43
4.2.1	Perancangan <i>Database</i>	43
4.2.2	Struktur Tabel dan Basis Data	45
4.2.3	Arsitektur Aplikasi	46
4.2.4	Class Diagram Aplikasi	47
4.2.5	Perancangan Antarmuka	50
4.2.5.1	Perancangan Antarmuka Aplikasi Web	50
BAB V	IMPLEMENTASI DAN PENGUJIAN SISTEM	53
5.1.	Implementasi Sistem	53
5.1.1.	<i>Source Code</i> Program	54
5.1.1.1.	<i>Source Code Routing</i> Aplikasi	54
5.1.1.2.	<i>Source Code Model</i> Aset	57
5.1.1.3.	<i>Source Code Controller</i> Santri	58
5.1.1.4.	<i>Source Code View Index</i> Laporan Keuangan	61
5.1.2.	Implementasi Antarmuka Aplikasi <i>Web</i>	65
5.1.2.1.	Halaman <i>Login</i>	65
5.1.2.2.	Halaman <i>Dashboard</i>	66
5.1.2.3.	Halaman Lihat Laporan Keuangan	67
5.2.	Pengujian Sistem	68
5.2.1.	Fokus Pengujian	68
5.2.2.	Kasus Hasil Pengujian	70
5.2.2.1.	Pengujian Pendaftaran Santri	70
5.2.2.2.	Pengujian Menambahkan Laporan Keuangan	72
5.2.2.3.	Pengujian Pengajuan Kegiatan Masjid	75
5.3.	Kesimpulan Hasil Pengujian	79
BAB VI	KESIMPULAN DAN SARAN	84

6.1. Kesimpulan.....	84
6.2. Saran.....	84
DAFTAR PUSTAKA.....	86



DAFTAR GAMBAR

Gambar 2.1 Simbol Event (Dewi, dkk., 2012).....	9
Gambar 2.2 Simbol Activity dan Gateway (Dewi, dkk., 2012).....	9
Gambar 2.3 Connection Object (Dewi, dkk., 2012).....	10
Gambar 2.4 Contoh Swimlane (Dewi, dkk., 2012).....	10
Gambar 3.1 Metode Waterfall (Sommerville, 2011).....	18
Gambar 3.2 Flowchart Penelitian.....	20
Gambar 4.1 BPMN Pencatatan Keuangan yang Sedang Berjalan.....	22
Gambar 4.2 BPMN Pencatatan Kegiatan yang Sedang Berjalan.....	23
Gambar 4.3 Pencatatan Aset yang Sedang Berjalan.....	24
Gambar 4.4 BPMN Pencatatan Santri yang Sedang Berjalan.....	25
Gambar 4.5 BPMN Pencatatan Keuangan yang Diusulkan.....	26
Gambar 4.6 BPMN Pencatatan kegiatan yang Diusulkan.....	27
Gambar 4.7 BPMN Pencatatan Aset yang Diusulkan.....	28
Gambar 4.8 BPMN Pencatatan Santri Masjid yang Diusulkan.....	29
Gambar 4.9 Use Case Diagram.....	31
Gambar 4.10 Sequence Diagram Pengajuan Kegiatan.....	38
Gambar 4.11 Sequence Diagram Pencatatan Keuangan.....	39
Gambar 4.12 Sequence Diagram Pencatatan Santri.....	40
Gambar 4.13 Class Analysis Pencatatan Keuangan.....	41
Gambar 4.14 Class Analysis pengajuan Kegiatan.....	42
Gambar 4.15 Class Analysis Pendaftaran Santri.....	42
Gambar 4.16 Rancangan ERD.....	44
Gambar 4.17 Rancangan Basis Data.....	45
Gambar 4.18 Arsitektur Aplikasi (MVC).....	47
Gambar 4.19 Class Diagram Aplikasi.....	49
Gambar 4.20 Antarmuka Halaman Awal.....	50
Gambar 4.21 Antarmuka Laporan Keuangan.....	51
Gambar 4.22 Antarmuka Aset Masjid.....	52
Gambar 5.1 Halaman Login.....	66
Gambar 5.2 Halaman Dashboard.....	67

Gambar 5.3 Laporan Keuangan	68
Gambar 5.4 Mendaftarkan Santri (Benar)	71
Gambar 5.5 Mendaftarkan Santri (Benar)	71
Gambar 5.6 Mendaftarkan Santri (Benar)	72
Gambar 5.7 Menambahkan Santri (Alternatif)	72
Gambar 5.8 Menambahkan Laporan Keuangan (Benar)	74
Gambar 5.9 Menambahkan Laporan Keuangan (Benar)	74
Gambar 5.10 Menambahkan Laporan Keuangan (Benar)	75
Gambar 5.11 Menambahkan Laporan Keuangan (Alternatif)	75
Gambar 5.12 Pengajuan Kegiatan Masjid (Benar)	77
Gambar 5.13 Pengajuan Kegiatan Masjid (Benar)	77
Gambar 5.14 Pengajuan Kegiatan Masjid (Benar)	78
Gambar 5.15 Pengajuan Kegiatan Masjid (Alternatif)	78



DAFTAR TABEL

Tabel 2.1 Simbol-simbol Use Case Diagram dan keterangan (Fauzi, 2011).....	10
Tabel 2.2 Simbol ERD (Rosa dan Shalahuddin, 2018).....	11
Tabel 4.1 Definisi Tugas Aktor.....	32
Tabel 4.2 Use Case Scenario Pencatatan Keuangan.....	35
Tabel 4.3 Use Case Scenario Registrasi Santri.....	36
Tabel 4.4 Use Case Scenario Pengajuan Kegiatan.....	37
Tabel 4.5 Struktur Tabel Neraca.....	45
Tabel 4.6 Struktur Tabel Kegiatan.....	46
Tabel 4.7 Struktur Tabel Aset.....	46
Tabel 5.1 Kode Program Routing Aplikasi.....	55
Tabel 5.2 Kode Program Model Aset.....	57
Tabel 5.3 Kode Program SantriController.....	58
Tabel 5.4 Kode Program Laporan Keuangan.....	62
Tabel 5.5 Fokus Pengujian.....	69
Tabel 5.6 Kasus dan Hasil Uji Pendaftaran Santri (Benar).....	70
Tabel 5.7 Kasus dan Hasil Uji Pendaftaran Santri (Alternatif).....	70
Tabel 5.8 Kasus dan Hasil Uji Menambahkan Laporan Keuangan (Benar).....	73
Tabel 5.9 Kasus dan Hasil Uji Menambahkan Laporan Keuangan (Alternatif)...	73
Tabel 5.10 Kasus dan Hasil Uji Pengajuan kegiatan Masjid (Benar).....	76
Tabel 5.11 Kasus dan Hasil Uji Pengajuan kegiatan Masjid (Alternatif).....	76
Tabel 5.12 Hasil Pengujian.....	79

DAFTAR LAMPIRAN

LAMPIRAN A	89
LAMPIRAN B	103
LAMPIRAN C	113
LAMPIRAN D	121
LAMPIRAN E	125
LAMPIRAN F	133
LAMPIRAN G	147
LAMPIRAN H	156



ABSTRAK

Masjid Nurul Ikhlas merupakan masjid besar yang terletak di Jalan Pinang Sori No. 10 RT 02 RW 02 Kelurahan Air Tawar Timur, Kecamatan Padang Utara. Selama ini proses pengelolaan kegiatan dan keuangan masjid menerapkan Sistem Informasi Pembukuan & Keuangan Ummat (SIPKU). SIPKU merupakan aplikasi open access yang dibuat oleh CSR Pegadaian yang awalnya digunakan oleh Masyarakat Ekonomi Syariah dan dapat dimanfaatkan untuk pencatatan keuangan masjid secara sederhana. Namun, SIPKU hanya memiliki fitur pengelolaan keuangan masjid dan tidak memiliki fitur pengelolaan masjid lainnya. Hal ini menyebabkan pengelolaan lain seperti kegiatan, aset, dan pendaftaran santri masjid masih dilakukan secara manual. Oleh karena itu, dibutuhkan sebuah sistem informasi yang memiliki seluruh fitur pengelolaan masjid sehingga kegiatan pengelolaan masjid dapat dilakukan pada satu sistem yang terintegrasi. Sistem informasi ini dibangun dalam bentuk web application dengan menggunakan framework Laravel dan basis data MySQL. Aplikasi ini dibangun berdasarkan metode pengembangan sistem waterfall yang memiliki beberapa tahapan yaitu analysis, design, pengkodean, dan testing. Hasil dari penelitian ini berupa sistem informasi yang dapat digunakan pengurus masjid dalam mengelola seluruh kegiatan dan keuangan masjid.

Kata Kunci : sistem informasi, kegiatan dan keuangan, framework laravel, metode waterfall

BAB I PENDAHULUAN

1.1 Latar Belakang

Masjid adalah tempat ibadah bagi umat Islam, khususnya dalam melakukan sholat. Masjid juga sering disebut *Baitullah* (Rumah Allah), yang fungsi utamanya sebagai tempat sholat, berdo'a, mengaji Al-Qur'an, pengajian dan ibadah lainnya (Rifa'i, 2016). Selain itu masjid juga berperan sebagai pusat kegiatan masyarakat dan komunitas muslim, diantaranya kegiatan pembinaan, pendidikan, pembelajaran, pemberdayaan umat dan kegiatan perayaan hari besar (Kristanto, 2018).

Masjid Nurul Ikhlas merupakan masjid besar yang terletak di Jalan Pinang Sori Induk No. 10 RT 02 RW 02 Kelurahan Air Tawar Timur, Kecamatan Padang Utara. Masjid ini mulai dibangun pada tanggal 10 november 1996 dan diresmikan pada tanggal 27 Desember 1997 oleh Drs. Zuen Rais, M.S, yang saat itu menjabat sebagai wali kota. Masjid ini dibangun karena penduduk kelurahan tersebut semakin bertambah dan mushala yang tersedia tidak mampu lagi menampung jamaah untuk beribadah. Sejak tahun 1997 sampai dengan tahun 2021, Masjid Nurul Ikhlas sudah memiliki 10 periode kepengurusan. Sejak kepengurusan periode ke-9 (2017-2020) status masjid berubah menjadi Masjid Besar Padang Utara. Masjid Besar adalah kriteria yang diberikan oleh pemerintah daerah setingkat Camat, atas rekomendasi dari Kepala KUA (Kantor Urusan Agama) Kecamatan, untuk menjadi pusat kegiatan keagamaan di wilayah kecamatan.

Selain memfasilitasi kegiatan beribadah, Masjid Nurul Ikhlas juga memiliki kegiatan rutin seperti wirid mingguan yang dilakukan setiap hari Selasa dan Jumat, *Majlis Ta'lim* Kaum Ibu setiap hari Sabtu, Wirid Remaja, Pesantren Kilat di setiap bulan *Ramadhan*, dan Didikan Subuh setiap hari Minggu. Masjid juga menerima donasi dan sedekah seperti masjid lain pada umumnya. Hasil donasi akan digunakan untuk pembangunan masjid, sedangkan sedekah akan digunakan untuk pembangunan TPA (Tempat Pendidikan Al'Qur'an) dan membantu anak yatim dan fakir/miskin.

Pada kepengurusan H. Yempita Efendi, pengurus masjid saat ini mencoba membuat situs web masjid yang berfungsi untuk memberikan informasi masjid kepada masyarakat. Situs web ini dibuat dengan menggunakan *Google Sites*, sebuah aplikasi dari *Google* untuk pembuatan situs web. Isi dari situs web ini terdiri dari profil masjid, sejarah masjid, struktur kepengurusan masjid, visi, misi, dan strategi masjid, daftar donatur bulanan masjid, dan kegiatan-kegiatan yang dilakukan masjid. Selain itu, pengurus masjid mencoba menerapkan Sistem Informasi Pembukuan & Keuangan Ummat (SIPKU). SIPKU merupakan aplikasi open access yang dibuat oleh CSR Pegadaian yang awalnya digunakan oleh Masyarakat Ekonomi Syariah dan dapat dimanfaatkan untuk pencatatan keuangan masjid secara sederhana. Sistem Informasi ini memiliki fitur *dashboard* total debit (pemasukkan masjid), total kredit (pengeluaran masjid), dan sisa saldo (sisa uang masjid) serta pengelolaan debit dan kredit masjid.

Namun, Sistem Informasi ini masih memiliki kekurangan, seperti tidak memiliki sistem pelaporan keuangan yang detail. Selain itu, SIPKU juga tidak memiliki fitur untuk mengelola kegiatan masjid. Masjid Nurul Ikhlas masih melakukan pengelolaan kegiatan secara manual, sehingga beresiko adanya kegiatan yang memerlukan dana tidak tercatat pada SIPKU.

Untuk mendukung penelitian ini penulis merujuk ke beberapa penelitian terdahulu, antara lain penelitian Eka Chandra Ramdhani dkk. tahun 2021 yang berjudul “Sipas (Sistem Informasi Pengelolaan Dana Masjid)”. Penelitian ini bertujuan untuk meningkatkan kualitas manajemen masjid, mengingat teknologi informasi dapat berperan penting dalam menyebarkan informasi, sehingga informasi mengenai kegiatan masjid, pertanggungjawaban dana masjid dapat diakses oleh umat.

Selanjutnya penelitian T. Indra Wardana dan Eko Aribowo tahun 2013 yang berjudul “Perancangan dan Implementasi Sistem Informasi Manajemen Kegiatan Masjid, Studi Kasus : Masjid Jogokarian Yogyakarta”. Penelitian ini bertujuan memudahkan kinerja sekretaris dan bendahara dalam proses pengelolaan pendataan jamaah, donatur arsip, mubaligh dan data keuangan, menghemat waktu yang dibutuhkan dalam memasukkan data, mengurangi resiko kesalahan dalam pembuatan laporan, dan pengelolaan kegiatan dapat dipertanggungjawabkan.

Dari berbagai penelitian tersebut dapat dijelaskan bahwa sistem informasi masjid dapat meningkatkan kualitas pengelolaan masjid. Kesalahan dalam memasukkan data kegiatan dan keuangan masjid dapat diminimalisir serta dapat dipertanggungjawabkan. Maka berdasarkan uraian tersebut penulis membuat penelitian dalam bentuk tugas akhir yang berjudul **“PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN MASJID NURUL IKHLAS KECAMATAN PADANG UTARA BERBASIS WEB”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan bahwa permasalahan yang dibahas pada penelitian ini adalah bagaimana membangun sistem informasi pada Masjid Nurul Ikhlas sehingga seluruh kegiatan pengelolaan masjid dapat dilakukan pada satu sistem informasi.

1.3 Batasan Masalah

Agar pembahasan ini tidak terlalu luas, maka penelitian dibatasi sebagai berikut:

1. Sistem hanya dapat diakses oleh pengurus masjid yang sudah terdaftar pada aplikasi.
2. Sistem Informasi di khususkan untuk dapat digunakan oleh Masjid Nurul Ikhlas yang terletak di Jalan Pinang Sori No. 10 RT 02 RW 02 Kelurahan Air Tawar Timur, Kecamatan Padang Utara.
3. Sistem Informasi hanya berfokus pada kegiatan pengelolaan aset, pengelolaan kegiatan, pengelolaan santri, dan pencatatan keuangan masjid.
4. Pengelolaan aset pada aplikasi hanya sebatas input data aset masjid saja dan tidak mengelola transaksi aset masjid.
5. Pengelolaan santri pada aplikasi hanya sebatas input data kedalam sistem.
6. Pengujian aplikasi hanya sebatas memeriksa ketersediaan fungsional dan kesesuaian dengan rancangan sistem yang diusulkan.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Melakukan Analisis terhadap kebutuhan-kebutuhan sistem yang akan dirancang.
2. Melakukan perancangan dan pembangunan Aplikasi Pengelolaan Masjid berbasis web.
3. Melakukan pengujian terhadap aplikasi yang dibangun, sehingga aplikasi betul-betul bisa digunakan dan dimanfaatkan oleh pihak-pihak yang membutuhkan.
4. Melalui Aplikasi ini dapat digunakan pengurus masjid Nurul Ikhlas dalam mengelola kegiatan dan keuangan masjid.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah agar aplikasi ini dapat diterapkan di Masjid Nurul Ikhlas sehingga pengelola dan pengurus Masjid Nurul Ikhlas dapat mengontrol kegiatan dan keuangan masjid serta memantau pengalokasian dananya untuk kegiatan masjid dapat lebih efektif.

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan pendahuluan mengenai pelaksanaan penelitian ini seperti latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan dari laporan ini.

BAB II TINJAUAN PUSTAKA

Bab ini berisi landasan teori dan informasi pendukung serta penunjang dalam melaksanakan penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang objek penelitian, metode pengumpulan data, metode penelitian serta *flowchart* penelitian.

BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi tentang pemodelan analisis sistem menggunakan tools seperti, *Business Process Model Notation (BPMN)*, *use case diagram*, *use case*

scenario, sequence diagram, perancangan basis data, struktur basis data dan tabel, *class diagram*, arsitektur aplikasi dan perancangan antarmuka.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi tentang pengimplementasian aplikasi ke dalam bahasa pemrograman berdasarkan analisis dan perancangan, serta pengujian terhadap hasil implementasi sistem.

BAB VI PENUTUP

Berisi tentang kesimpulan terhadap hasil penelitian dan saran untuk pengembangan sistem kedepannya.



BAB II TINJAUAN PUSTAKA

Pada bab ini, menjelaskan mengenai teori-teori pendukung dalam melaksanakan penelitian ini dimana terdiri dari penjelasan pengertian dan fungsi masjid, pengertian sistem informasi, alat analisis dan perancangan, dan *web application*.

2.1. Pengertian dan Fungsi Masjid

Sub bab ini menjelaskan tentang teori dan pengertian masjid, serta fungsi dan peranan masjid dalam kehidupan masyarakat.

2.1.1. Pengertian Masjid

Menurut Syafe'i (2016), secara etimologis masjid diambil dari kata sujud yang berarti taat, patuh, tunduk dengan penuh rasa hormat dan takzim. Sedangkan secara terminologis, dalam hukum Islam (*fiqh*), sujud adalah meletakkan dahi berikut ujung hidung, kedua telapak tangan, kedua lutut dan kedua ujung jari kaki ke tanah, yang merupakan salah satu rukun shalat. Sujud dalam pengertian ini merupakan bentuk lahiriyah yang paling nyata dari makna-makna diatas. Itulah sebabnya, tempat khusus penyelenggaraan shalat disebut dengan masjid.

Bedasarkan pengertian sujud secara terminologis diatas, Syafe'i (2016) mendefinisikan masjid sebagai “suatu bangunan, gedung atau suatu lingkungan yang memiliki batas yang jelas (benteng/pagar) yang didirikan secara khusus sebagai tempat beribadah umat Islam kepada Allah SWT, khususnya untuk menunaikan shalat”.

2.1.2. Fungsi Masjid Dalam Kehidupan Masyarakat

Selain tempat ibadah, masjid juga merupakan pusat kehidupan komunitas muslim. Kegiatan – kegiatan perayaan hari besar, diskusi, kajian agama, ceramah dan belajar Al Qur'an sering dilaksanakan di Masjid. Bahkan dengan seiring perkembangan zaman dan juga agama Islam di Indonesia, kegiatan yang bisa di lakukan di lingkungan Masjid pun mulai banyak dan beragam, mulai dari kegiatan keagamaan sehari-hari umat Islam sampai kegiatan-kegiatan yang

bisa menunjang kelancaran ibadah (Yudhasmara, 2016). Adapun beberapa fungsi masjid antara lain (Rifa'i, 2016):

- a. Masjid sebagai sarana dakwah
- b. Masjid sebagai pusat pengembangan moral dan sosial
- c. Masjid sebagai pusat pendidikan
- d. Masjid sebagai pusat pengembangan ekonomi
- e. Masjid sebagai pusat pengembangan politik

2.2. Pengertian Sistem Informasi

Sistem adalah suatu kumpulan objek-objek yang saling berhubungan dan berinteraksi satu sama lain serta menjadi satu kesatuan untuk mencapai suatu tujuan. Karakteristik sistem terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem, dan sasaran sistem (Amalia 2017). Sedangkan Informasi menurut Hermawan dan Hidayat (2016), berpendapat informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang. Sistem informasi menurut Anggraeini dan Irviani (2017) merupakan suatu kombinasi teratur dari orang-orang, hardware, software, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa sistem informasi adalah suatu proses mengolah, menampilkan, pendukung pengambilan keputusan, dan menyimpulkan sesuatu yang terdiri dari kumpulan-kumpulan data yang memiliki arti yang saling berhubungan dan bekerjasama untuk mencapai tujuan tertentu.

2.3. Aplikasi Web

Menurut Remick Jarel (2010) aplikasi web (*web application*) merupakan sebuah aplikasi yang menggunakan teknologi *browser* untuk menjalankan aplikasi dan diakses melalui jaringan komputer. Sedangkan menurut Margaret Rouse (2011) aplikasi *web* adalah sebuah program yang disimpan di *server* dan dikirim melalui internet dan diakses melalui antarmuka browser.

Berdasarkan beberapa pendapat diatas dapat disimpulkan bahwa *web application* adalah sebuah aplikasi berbasis *internet* yang disimpan kedalam *server* dan dapat dibuka dengan menggunakan *browser*. Aplikasi web menggunakan *HTML (HyperText Markup Language)* untuk membuat tampilan depan (*frontend*) dari aplikasi.

2.4. Alat Analisis dan Perancangan Sistem

Alat-alat analisis dan perancangan sistem yang akan digunakan sebagai tahap awal dalam pembuatan adalah sebuah sistem informasi yang terdiri dari Unified Process Modelling Language (UML), *Business Process Modelling Notation (BPMN)*, *Use case Diagram*, dan *Entity Relationship Diagram (ERD)*.

2.4.1. Unified Modelling Language (UML)

UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997.

UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software*.

UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor *scalability*, *robustness*, *security*, dan sebagainya (Dicoding, 2021).

2.4.2. Business Process Modelling Notation (BPMN)

Business Process Modelling Notation (BPMN) adalah bahasa standar untuk mendeskripsikan proses bisnis. *BPMN* dirancang agar mudah digunakan dan dipahami serta memiliki kemampuan untuk memodelkan proses bisnis yang kompleks dan secara spesifik dirancang dengan mempertimbangkan *web services*. Notasi yang digunakan dalam *BPMN* mudah dipahami oleh semua pengguna bisnis. *BPMN* lebih cenderung menggambarkan proses dari pandangan seorang analisis bisnis yang lebih mengarah penyampaian kebutuhan yang akan dideskripsikan kepada analisis IT dan pengembang *software*. *BPMN* menyediakan

BPD (Business Process Diagram), yang berlandaskan pada teknik flowchart yang digunakan untuk membuat model proses bisnis (Aini, 2013).

Sebuah BPD terdiri dari sekumpulan elemen grafis, dan kategori dasar dari elemen BPD adalah (Dewi, dkk., 2012) :

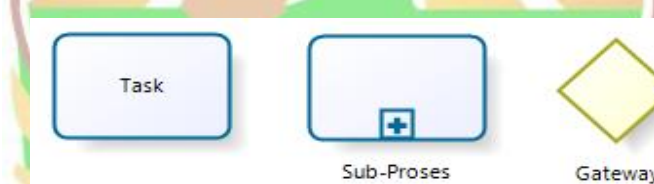
1. *Flow Object*

Flow object terdiri dari *event*, *activity*, dan *gateway*. *Event* adalah notasi yang menggambarkan suatu kondisi atau keadaan. Ada tiga jenis event yaitu *start* untuk memulai proses, *intermediate* untuk menginterupsi proses, dan *end* untuk mengakhiri proses. *Event* dapat dilihat pada gambar 2.1.



Gambar 2.1 Simbol Event (Dewi, dkk., 2012)

Activity adalah notasi pekerjaan yang dilakukan oleh perusahaan. Gateway merupakan notasi percabangan alur. Ada empat tipe gateway, yaitu *exclusive decisions*, *parallel execution*, *inclusive decisions*, dan *rework*. Simbol *Activity* dan *Gateway* dapat dilihat pada gambar 2.2.



Gambar 2.2 Simbol Activity dan Gateway (Dewi, dkk., 2012)

2. *Connecting Object*

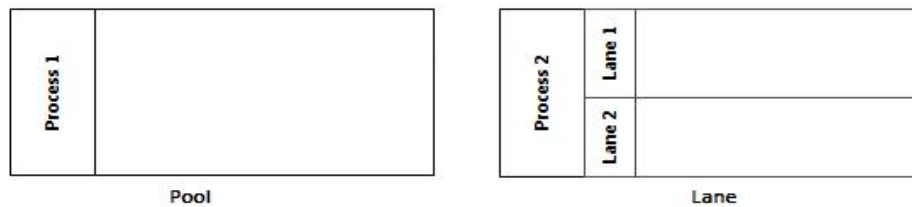
Connecting Object adalah notasi garis yang menghubungkan *flow object*. *Connecting Object* memiliki 3 jenis elemen, yaitu *Sequence Flow*, *Message Flow*, dan *Association*. *Sequence Flow* digunakan untuk menunjukkan urutan yang kegiatannya dilakukan dalam sebuah proses. *Message Flow* Digunakan untuk menunjukkan aliran pesan antara dua entitas yang siap untuk mengirim dan menerima. *Association* digunakan untuk asosiasi data, informasi, dan artefak dengan aliran benda. Contoh tipe-tipe *connection object* dapat dilihat pada gambar 2.3 berikut :



Gambar 2.3 Connection Object (Dewi, dkk., 2012)

3. Swimlanes

Elemen *Swimlane* adalah notasi berbentuk garis yang memisahkan atau mengelompokkan aktor. *Swimlane* digunakan untuk mengkategorikan fungsional atau tanggung jawab yang berbeda pada bisnis. Contoh simbol *pool* dan *lane* dalam *swimlanes* dapat dilihat pada gambar 2.4 berikut.



Gambar 2.4 Contoh Swimlane (Dewi, dkk., 2012)

2.4.3. Use Case Diagram

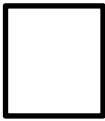
Rosa dan Shalahuddin (2018), *Use case diagram* atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

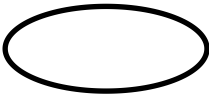


Berikut adalah simbol-simbol *use case diagram* yang dapat dilihat pada tabel 2.1 berikut ini :

Tabel 2.1 Simbol-simbol Use Case Diagram dan keterangan (Fauzi, 2011)

Simbol	Keterangan
--------	------------

	Sistem yaitu gambar batasan (<i>boundaries</i>) sebuah sistem.
---	--

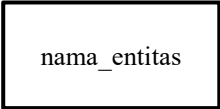
Tabel 2.1 Simbol-simbol Use Case Diagram dan keterangan (Fauzi, 2011)
(lanjutan)

	<i>Use Case</i> , merupakan deskripsi dari urutan aksi-aksi.
	<i>Actor</i> , yaitu para pengguna (<i>user</i>) dari sebuah sistem. Aktor adalah seseorang atau sesuatu yang harus berinteraksi dengan sistem.
	<i>Relationship</i> , yaitu gambaran hubungan/relasi antara aktor dengan <i>use case</i> .

2.4.4. Entity Relationship Diagram (ERD)

Menurut Rosa dan Shalahuddin(2018), “ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional”. Simbol-simbol yang digunakan dalam ERD dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol ERD (Rosa dan Shalahuddin, 2018)

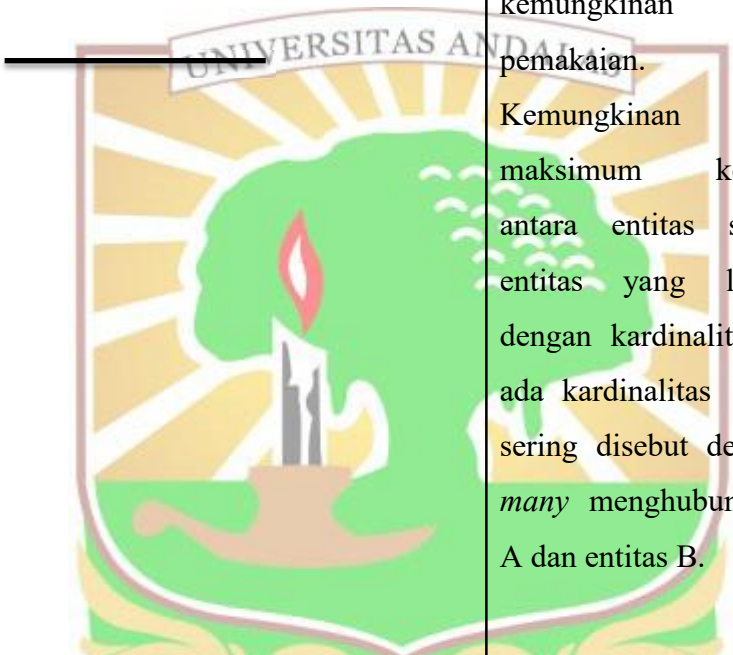
Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum

	merupakan nama tabel.
--	-----------------------

Tabel 2.2 Simbol ERD (Rosa dan Shalahuddin, 2018) (Lanjutan)

<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p>
<p>Atribut kunci primer</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan, biasanya berupa <i>id</i>, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik.</p>
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja.</p>

Tabel 2.2 Simbol ERD (Rosa dan Shalahuddin, 2018) (Lanjutan)

<p>Asosiasi / <i>Association</i></p> 	<p>Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.</p> <p>Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.</p>
--	---

2.5. Perangkat Lunak yang digunakan

Pada sub bab ini dijelaskan mengenai teori-teori pendukung dalam membangun aplikasi berbasis web yang terdiri dari *database* MySQL, *framework*, *Laravel*, dan *Laragon*.

2.5.1. Database MySQL

My Structured Query Language (MySQL) adalah salah satu jenis *database* server yang menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya*. *MySQL* termasuk jenis *RDBMS (Relational Database Management System)*. Pada *MySQL*, sebuah *database* mengandung satu atau sejumlah tabel.

Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

SQL adalah bahasa standar yang digunakan untuk berkomunikasi dengan basis data relasional, dan juga merupakan bahasa yang digunakan oleh banyak aplikasi atau *tools* untuk berinteraksi dengan server basis data. *SQL* adalah bahasa fungsional yang tidak mengenal iterasi dan tidak bersifat prosedural. *SQL* menggunakan perintah-perintah dengan kata-kata sederhana dan mirip dengan bahasa manusia sehari-hari (Arifudzaki, B., Somantri, M., & FR, A. 2010).

2.5.2. *Framework*

Menurut Naista (2017), *framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah website yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat website lebih singkat dan memudahkan dalam melakukan perbaikan.

Menurut Rony Setiawan (2021) *framework* memiliki fungsi utama untuk memudahkan web developer dalam membuat sebuah website. Selain itu, *framework* juga memiliki fungsi lain. Berikut di antaranya:

- a) Membuat kode program menjadi lebih terstruktur

Framework biasanya memiliki pola arsitektur dalam menuliskan kode. Sehingga, kode yang dituliskan lebih mudah dan struktur. Sehingga kesalahan dapat langsung ditemukan dan diperbaiki.

- b) Meningkatkan keamanan

Framework dapat meningkatkan keamanan website. Seperti contohnya *framework Laravel* yang sudah mengadopsi berbagai sistem keamanan seperti autentikasi, enkripsi, dan *hashing*.

- c) Mempercepat pembuatan *website*

Framework dapat mempercepat pembuatan *website*. Hal itu karena pengembang dapat menggunakan komponen-komponen yang sudah disediakan dan tidak perlu menulis kode dari awal.

- d) Pemeliharaan dan perawatan *website* lebih mudah

Perbaikan *bug*, *maintenance* menambah fitur dan meningkatkan keamanan *website* akan jadi lebih mudah karena kebanyakan *framework* sudah menggunakan pola arsitektur yang beragam.

2.5.3. *Laravel*

Laravel adalah *framework* berbasis *PHP* yang dirilis di bawah lisensi *MIT* dan dibangun dengan konsep (*Model View Controller*). *Laravel* merupakan pengembangan *website* berbasis *MVP* yang ditulis dalam *PHP* yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan aplikasi yang menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu (Supardi & Sulaeman, 2019).

Penelitian ini menggunakan *Laravel* karena *Laravel* bersifat *MVC* yang memisahkan antara *backend* dan *frontend* aplikasi, sehingga mempermudah pembuatan dan pemahaman logika aplikasi tersebut. *Laravel* juga memiliki beberapa ekstensi yang dapat memberikan fitur tambahan untuk *Laravel* sesuai dengan kebutuhan aplikasi.

2.5.4. *Laragon*

Laragon adalah *universal development environment* untuk *PHP*, *Node.js*, *Python*, *Java*, *Go*, dan *Ruby* yang dikembangkan oleh Leo Khoa (Khoa, 2019). Fungsi *Laragon* sama seperti *XAMPP*, yaitu memfasilitasi perangkat-perangkat yang diperlukan untuk membuat program berbasis *web*, seperti *HTTP Server*, *Database Management System (DBMS)*, dan menerjemahkan bahasa pemrograman *PHP* untuk bisa dibaca oleh *browser*. Penelitian ini menggunakan *Laragon* karena *Laragon* tidak memiliki dependensi dengan *Windows Operating System*, sehingga kinerja *Laragon* lebih cepat daripada *XAMPP*.

BAB III METODE PENELITIAN

Pada bab ini dijelaskan mengenai metodologi penelitian yang dilakukan terhadap penelitian ini dimana terdiri dari objek penelitian, metode pengumpulan data, metode penelitian, dan *flowchart* penelitian.

3.1. Objek Penelitian

Objek kajian dari penelitian ini adalah pembangunan sistem informasi pengelolaan kegiatan dan keuangan masjid Nurul Ikhlas. Objek ini diambil karena belum adanya aplikasi untuk menampilkan informasi keuangan masjid dan tidak adanya sistem informasi pendataan kegiatan masjid serta korelasinya dengan keuangan masjid yang menyebabkan kurang maksimalnya proses manajemen di Masjid Nurul Ikhlas

3.2. Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan dalam penelitian ini adalah dengan melakukan studi lapangan, yang terdiri dari wawancara, analisis dokumen, serta studi literatur.

a. Wawancara

Wawancara dilakukan untuk memperoleh data dan informasi yang lebih rinci mengenai penelitian. Wawancara difokuskan terhadap kebutuhan dari masjid dan proses yang sedang berlangsung di masjid. Wawancara dilakukan dengan sesi tanya jawab kepada pengurus masjid.

b. Analisis Dokumen

Analisis dokumen adalah tahap pengumpulan data dalam bentuk dokumen. Dokumen yang dianalisis adalah neraca keuangan masjid, laporan donatur pembelian *vacuum cleaner*, jadwal penceramah wirid selasa malam masjid, dan data santri TPQ.

c. Studi literatur

Studi literatur dilakukan dengan mempelajari dan memahami literatur dari berbagai sumber seperti situs internet, jurnal ilmiah dan bacaan lain yang mendukung penelitian.

3.3. Metode Pembangunan Aplikasi

Metode yang digunakan untuk membangun aplikasi ini adalah metode *Waterfall*. Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan yang berawal dari analisis dan definisi kebutuhan, perancangan perangkat lunak, implementasi dan *unit testing*, integrasi dan *system testing*, dan berakhir dengan mengoperasikan dan pemeliharaan. Dalam penelitian ini, tahapan yang digunakan hanya dilakukan dari tahap analisis dan definisi kebutuhan hingga tahap implementasi dan *unit testing* (Sommerville, 2011). Metode *waterfall* dapat dilihat pada Gambar 3.1. Tahapan dari metode *waterfall* yaitu :

a. *Requirement Analysis and Definition*

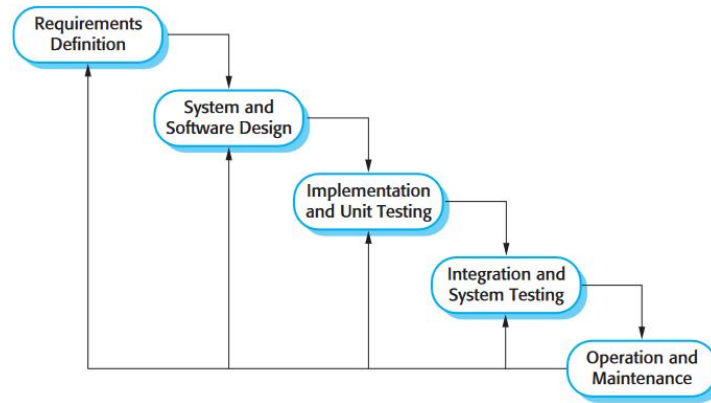
Tahap analisis dan definisi kebutuhan adalah tahap penetapan layanan, kendala, dan tujuan sistem melalui konsultasi dengan pengguna sistem yang kemudian didefinisikan secara rinci dan menghasilkan spesifikasi sistem.

b. *System and Software Design*

Tahap perancangan sistem dan perangkat lunak adalah tahap membangun arsitektur sistem secara keseluruhan. Desain perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak mendasar dan hubungannya.

c. *Implementation and Unit Testing*

Pada tahap implementasi, desain perangkat lunak direalisasikan sebagai sekumpulan program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.



Gambar 3.1 Metode Waterfall (Sommerville, 2011)

3.4. **Flowchart Penelitian**

Flowchart dari penelitian ini menggambarkan langkah-langkah yang dilakukan dalam pembuatan aplikasi. *Flowchart* penelitian ini terlihat pada gambar 3.2.

Berdasarkan *flowchart* tersebut, dapat diuraikan langkah-langkah yang dilakukan pada penelitian ini, diantaranya:

1. Studi Literatur dan Identifikasi Masalah

Pada tahap ini menganalisa proses yang sedang berjalan dan mengidentifikasi masalah yang terjadi dalam hal pengelolaan informasi terkait kegiatan dan keuangan yang dilakukan oleh pengurus Masjid Nurul Ikhlas serta mencari data pendukung penelitian dalam bentuk jurnal, artikel, buku dan situs internet yang terpercaya.

2. Pengumpulan Data

Pengumpulan data, baik data primer maupun sekunder, yang dapat digunakan dalam pembangunan aplikasi.

3. Analisis dan Definisi Kebutuhan

Melakukan analisis kebutuhan *user* dan pemodelan proses komunikasi menggunakan *BPMN*.

4. Perancangan Sistem

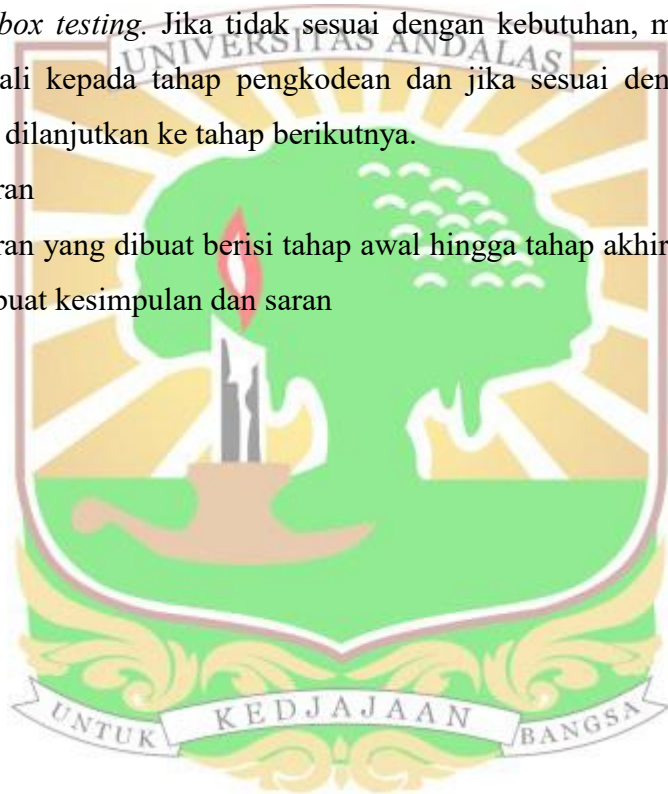
Melakukan perancangan *Entity Relationship Diagram*, perancangan *user interface*, arsitektur aplikasi dan perancangan menggunakan *tools UML*.

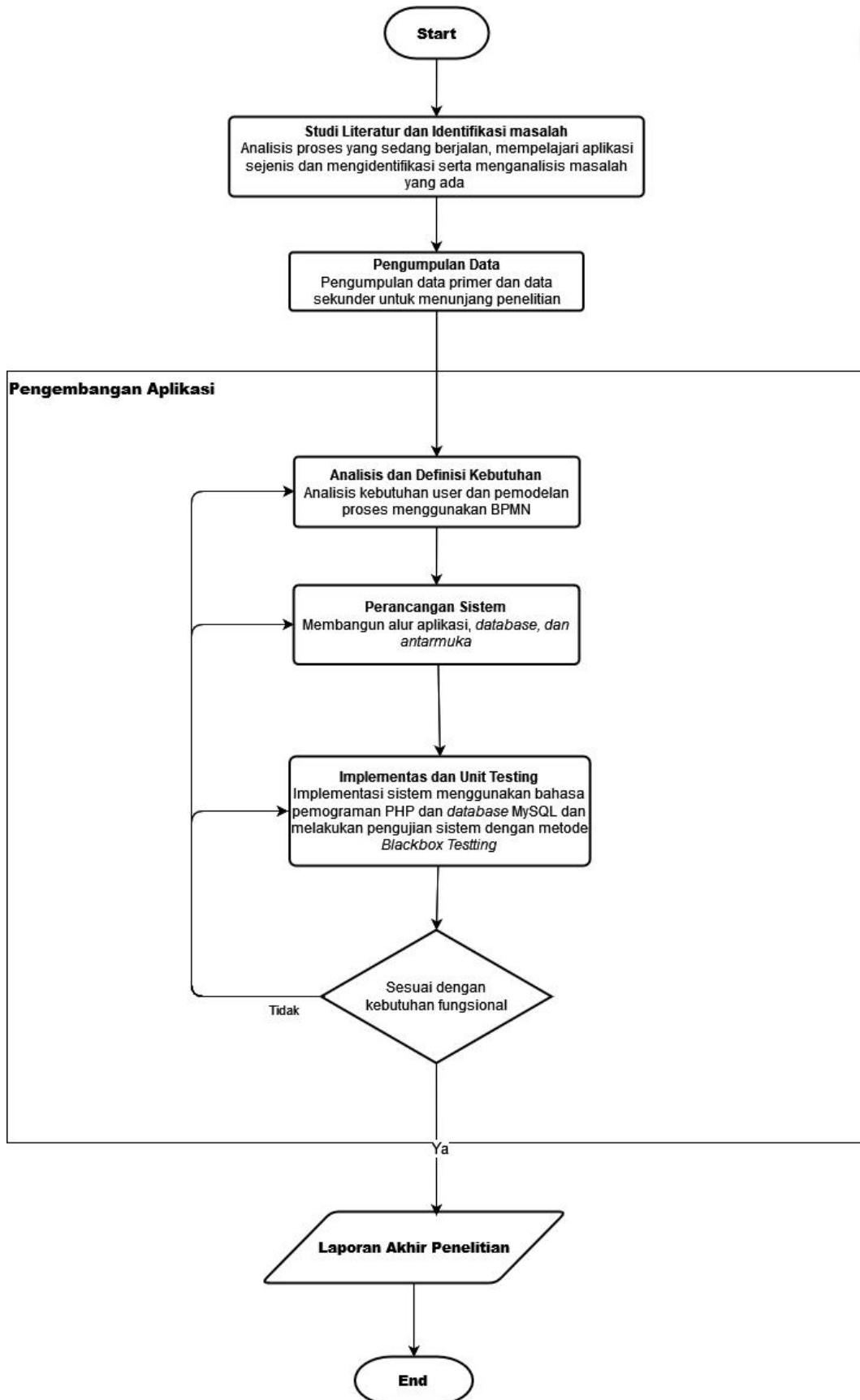
5. Implementasi dan *unit testing*

Pada tahap ini dilakukan pembangunan aplikasi dengan bahasa pemrograman PHP dan *database MySQL* dengan menggunakan *framework Laravel*. Pengujian dilakukan untuk melihat kesesuaian kebutuhan fungsional dengan aplikasi yang dirancang menggunakan *blackbox testing*. Jika tidak sesuai dengan kebutuhan, maka pengerjaan kembali kepada tahap pengkodean dan jika sesuai dengan kebutuhan, maka dilanjutkan ke tahap berikutnya.

6. Laporan

Laporan yang dibuat berisi tahap awal hingga tahap akhir penelitian serta membuat kesimpulan dan saran





Gambar 3.2 Flowchart Penelitian

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan hasil dari analisis kebutuhan dan perancangan sistem informasi pengelolaan kegiatan dan keuangan Masjid Nurul Ikhlas Kecamatan Padang Utara berbasis web. Analisis sistem ini menggunakan tools *Bussines Process Model and Notation (BPMN)*, analisis kebutuhan fungsional sistem, *use case diagram*, *use case scenario*, *sequence diagram*, dan *class analysis*. Sedangkan pada perancangan sistem terdiri dari perancangan basis data, struktur tabel dan basis data, arsitektur aplikasi, *class diagram*, dan user interface.

4.1 Analisis Sistem

Tahapan analisis sistem menjelaskan tentang kedudukan sistem saat ini, sistem yang diusulkan dan analisis sistem yang dimodelkan menggunakan *BPMN (Business Process Model Notation)*. Tools yang digunakan dalam analisis sistem ini menggunakan UML (*Unified Modelling Language*) antara lain : *use case diagram*, *use case scenario*, *sequence diagram*, dan *class analysis*.

4.1.1 Sistem yang Sedang Berjalan

Bagian ini membahas mengenai proses bisnis yang sedang berjalan pada Masjid Nurul Ikhlas dengan menggunakan *Bussiness Process Model Notation (BPMN)*, meliputi proses pencatatan keuangan masjid, proses pencatatan kegiatan, proses pencatatan aset, dan pencatatan santri masjid.

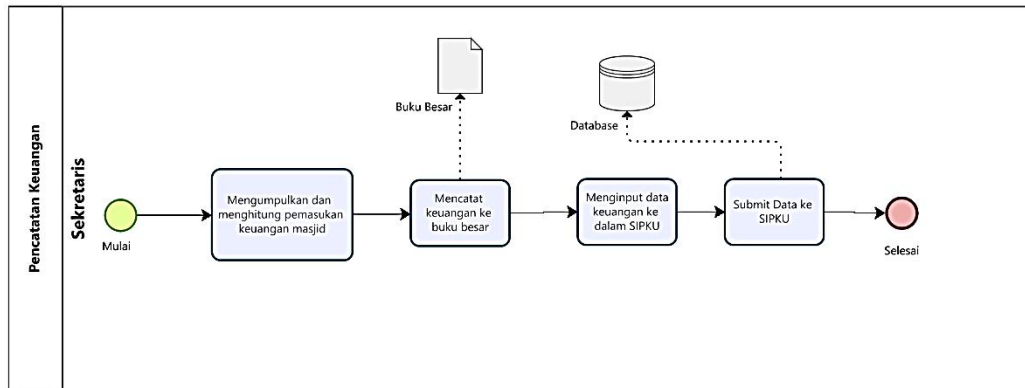
a. Pencatatan Keuangan

Berikut penjelasan pencatatan keuangan yang sedang berjalan di Masjid Nurul Ikhlas :

1. Sekretaris masjid mengumpulkan dan menghitung pemasukkan keuangan masjid.
2. Sekretaris lalu mencatat hasil keuangan tersebut kedalam buku besar.
3. Sekretaris membuka sistem keuangan masjid dan menginput hasil keuangan kedalam sistem.

4. Sekretaris meng-*submit* data kedalam sistem.

BPMN untuk proses pencatatan keuangan bisa dilihat pada gambar 4.1 berikut:



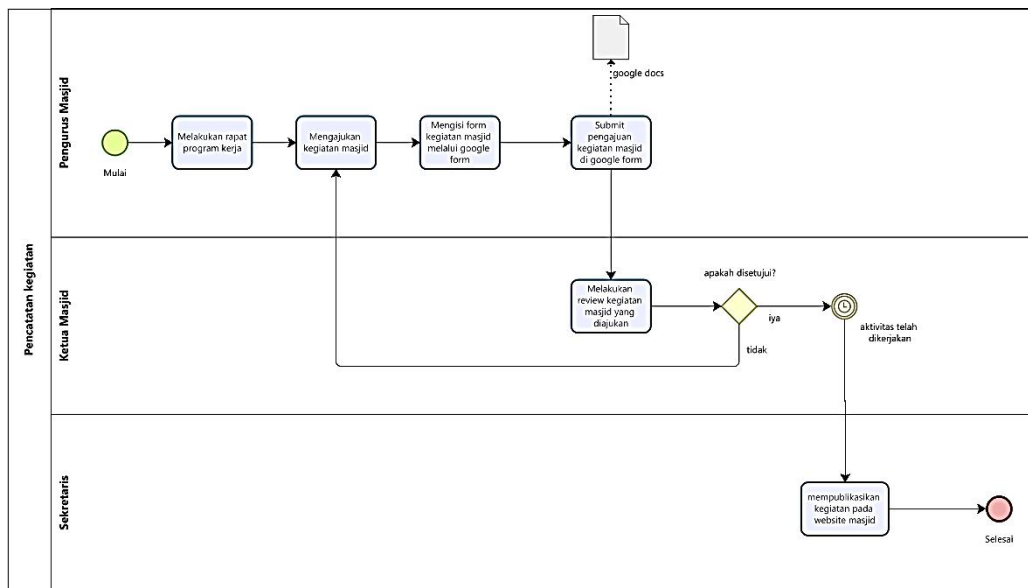
Gambar 4.1 BPMN Pencatatan Keuangan yang Sedang Berjalan

b. Pencatatan Kegiatan

Berikut penjelasan pencatatan kegiatan yang sedang berjalan di Masjid Nurul Ikhlas :

1. Pengurus masjid melakukan rapat program kerja setiap awal masa kepengurusan masjid.
2. Anggota lalu mengajukan kegiatan masjid sesuai dengan hasil rapat program kerja dengan mengisi dan *meng-submit* formulir kegiatan masjid melalui *google form*.
3. Ketua masjid akan melakukan tinjauan terhadap kegiatan masjid yang diajukan. Jika pengajuan kegiatan ditolak, maka ketua masjid memberi tahu kepada pengurus masjid dan dapat mengajukan kegiatan masjid kembali. Jika pengajuan kegiatan disetujui, maka sekretaris masjid akan mempublikasikan kegiatan pada *website* masjid.

BPMN untuk proses pembuatan kegiatan ini bisa dilihat pada gambar 4.2 berikut:



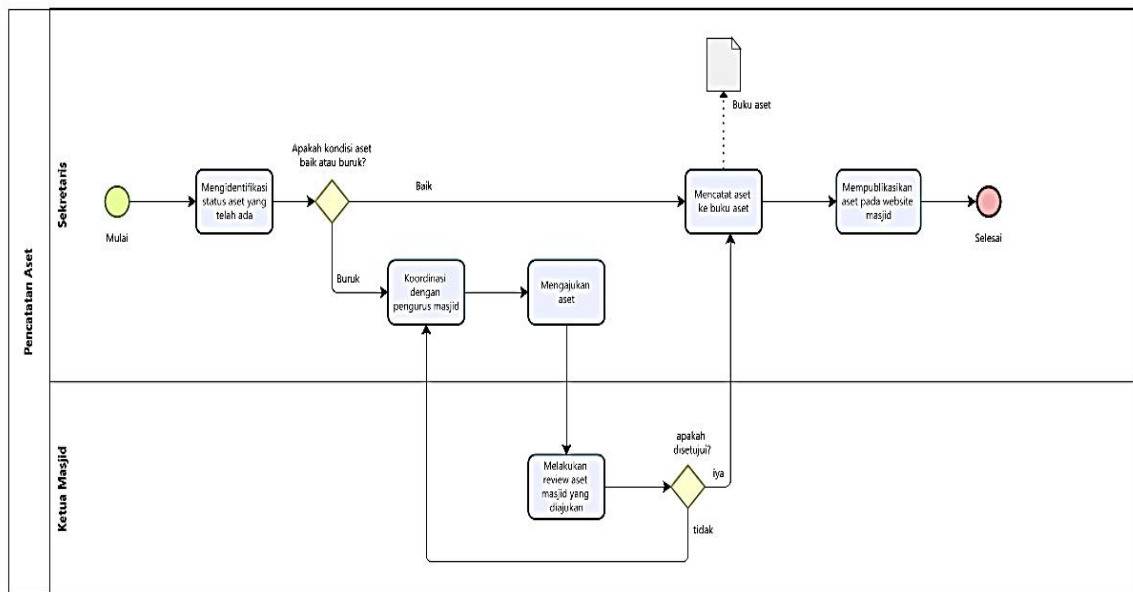
Gambar 4.2 BPMN Pencatatan Kegiatan yang Sedang Berjalan

c. Pencatatan Aset

Berikut penjelasan pencatatan aset yang sedang berjalan di Masjid Nurul Ikhlas :

1. Sekretaris masjid mengidentifikasi status aset yang telah ada. Jika kondisi aset baik-baik saja maka sekretaris akan mencatat aset tersebut dan mempublikasikan aset kedalam *website* masjid.
2. Jika kondisi aset ada yang buruk, maka sekretaris akan berkoordinasi dengan pengurus masjid.
3. Sekretaris lalu akan mengajukan aset yang baru kepada ketua masjid
4. Ketua masjid akan melakukan tinjauan terhadap aset masjid yang diajukan. Jika pengajuan aset ditolak, maka ketua masjid memberi tahu sekretaris dan dapat mengajukan kegiatan masjid kembali. Jika pengajuan aset disetujui, maka sekretaris masjid akan mempublikasikan aset pada *website* masjid.

BPMN untuk proses pencatatan aset ini bisa dilihat pada gambar 4.3 berikut:



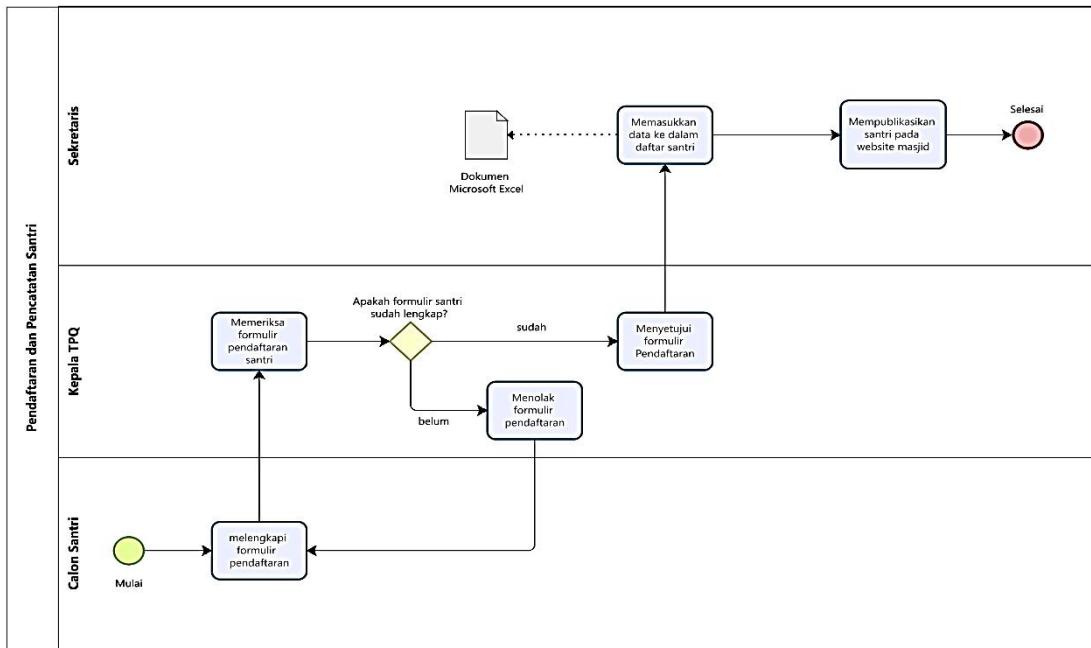
Gambar 4.3 Pencatatan Aset yang Sedang Berjalan

d. Pencatatan Santri Masjid

Berikut penjelasan pencatatan santri yang sedang berjalan di Masjid Nurul Ikhlas :

1. Calon santri melengkapi formulir pendaftaran dan mengirim formulir pendaftaran tersebut kepada pengurus masjid
2. Ketua masjid menerima formulir dan memeriksa formulir tersebut. Jika formulir santri belum lengkap, maka ketua masjid akan menolak formulir pendaftaran calon santri dan calon santri dapat melengkapi formulir pendaftaran kembali. Jika formulir santri sudah lengkap, maka ketua masjid menyetujui formulir pendaftaran dan calon santri akan menerima surat diterima sebagai santri.
3. Sekretaris memasukkan data santri ke dalam dokumen.
4. Sekretaris mempublikasikan santri pada *website* masjid.

BPMN untuk proses pencatatan aset ini bisa dilihat pada gambar 4.4 berikut:



Gambar 4.4 BPMN Pencatatan Santri yang Sedang Berjalan

4.1.2 Sistem yang Diusulkan

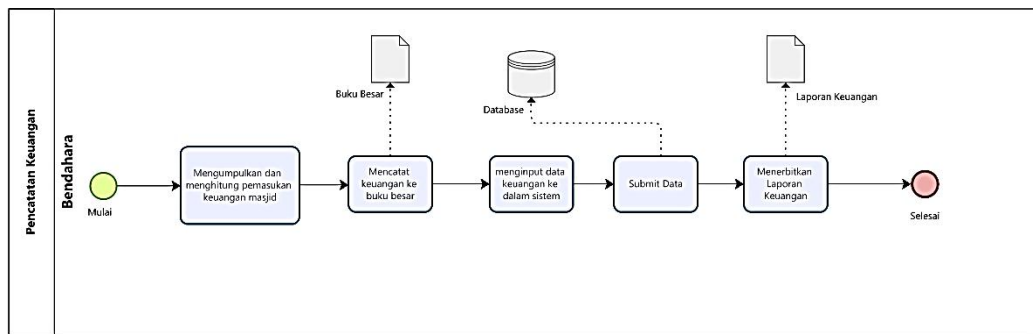
Bagian ini membahas mengenai proses bisnis yang diusulkan dengan menggunakan *Business Process Model Notation (BPMN)*, meliputi proses pencatatan keuangan masjid, proses pencatatan kegiatan, proses pencatatan aset, dan pencatatan santri masjid.

a. Pencatatan Keuangan

Berikut penjelasan pencatatan keuangan yang sedang berjalan di Masjid Nurul Ikhlas :

1. Bendahara masjid mengumpulkan dan menghitung pemasukkan keuangan masjid.
2. Bendahara lalu mencatat hasil keuangan tersebut kedalam buku besar.
3. Bendahara membuka sistem keuangan masjid dan menginput hasil keuangan ke dalam sistem.
4. Bendahara meng-*submit* data kedalam sistem.

BPMN untuk proses pencatatan kegiatan bisa dilihat pada gambar 4.5 berikut:



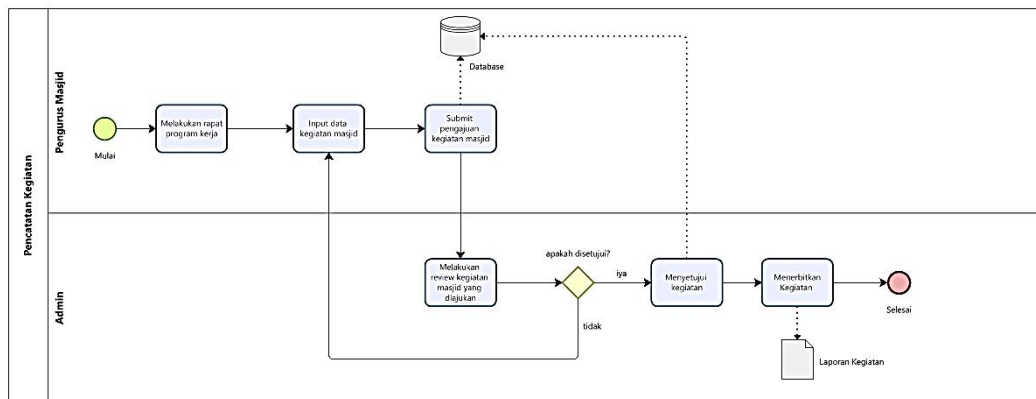
Gambar 4.5 BPMN Pencatatan Keuangan yang Diusulkan

b. Pencatatan Kegiatan

Berikut penjelasan pencatatan kegiatan yang sedang berjalan di Masjid Nurul Ikhlas :

1. Pengurus masjid melakukan rapat program kerja setiap awal masa kepengurusan masjid.
2. Pengurus masjid lalu mengajukan kegiatan masjid sesuai dengan hasil rapat program kerja dengan mengisi dan meng-submit form kegiatan masjid pada sistem.
3. Admin akan melakukan tinjauan terhadap kegiatan masjid yang diajukan. Jika pengajuan kegiatan ditolak, maka sistem akan memberikan status “ditolak” pada kegiatan yang diajukan dan pengurus masjid dapat mengajukan kegiatan kembali. Jika pengajuan kegiatan disetujui, maka sistem akan memberikan status “disetujui” pada kegiatan yang diajukan dan sistem akan menyimpan data kegiatan tersebut.
4. Admin akan mempublikasikan kegiatan pada *website* masjid.

BPMN untuk proses pembuatan kegiatan ini bisa dilihat pada gambar 4.6 berikut:



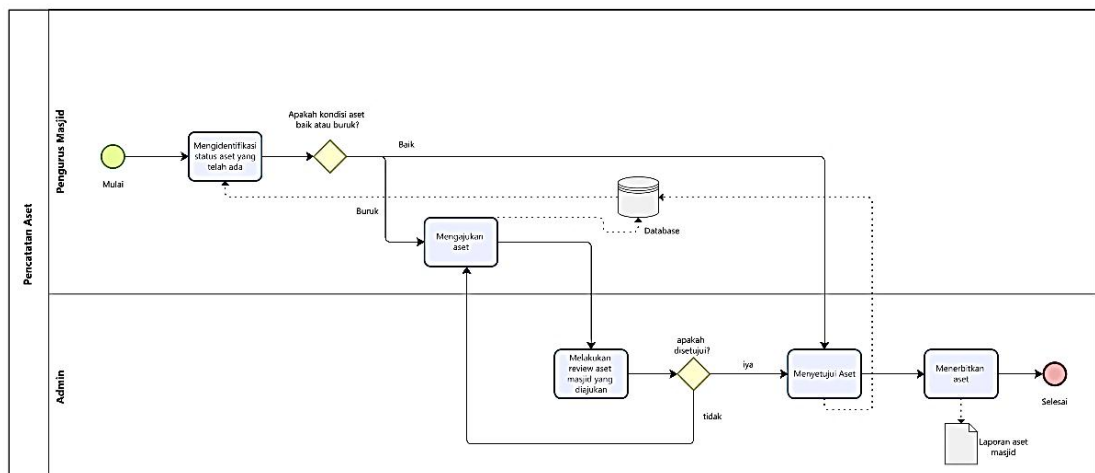
Gambar 4.6 BPMN Pencatatan kegiatan yang Diusulkan

c. Pencatatan Aset

Berikut penjelasan pencatatan aset yang sedang berjalan di Masjid Nurul Ikhlas :

1. Pengurus masjid mengidentifikasi status aset yang telah ada. Jika kondisi aset baik-baik saja maka admin akan mencatat aset ke dalam sistem dan mempublikasikan aset kedalam *website* masjid.
2. Jika kondisi aset ada yang buruk, maka pengurus akan melakukan *input* pengajuan aset yang baru kedalam sistem.
3. Admin akan melakukan tinjauan terhadap aset masjid yang diajukan. Jika pengajuan aset ditolak, maka sistem akan memberikan status “ditolak” pada aset yang diajukan dan sekretaris dapat mengajukan aset kembali. Jika pengajuan aset disetujui, maka sistem akan memberikan status “disetujui” pada kegiatan yang diajukan dan sistem akan menyimpan data aset tersebut.
4. Sekretaris masjid akan mempublikasikan aset pada *website* masjid.

BPMN untuk proses pencatatan aset ini bisa dilihat pada gambar 4.7 berikut:



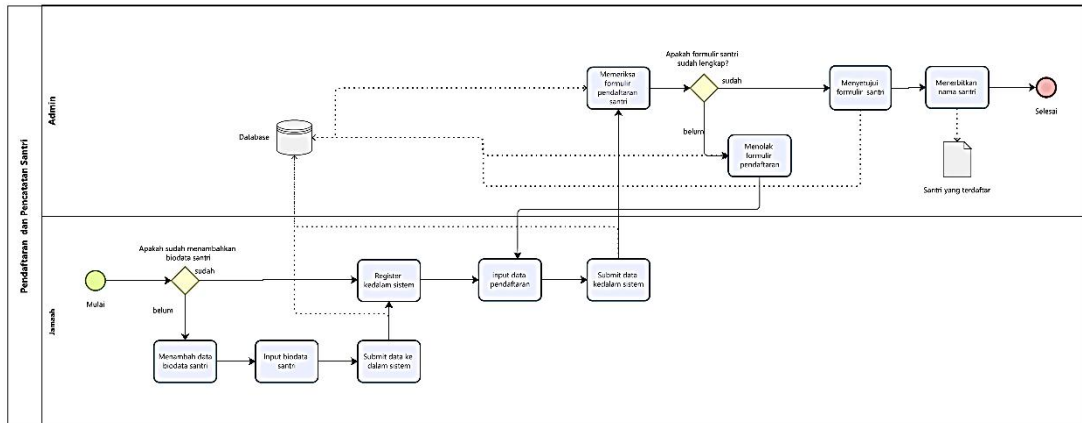
Gambar 4.7 BPMN Pencatatan Aset yang Diusulkan

d. Pencatatan Santri Masjid

Berikut penjelasan pencatatan santri yang sedang berjalan di Masjid Nurul Ikhlas :

1. Jamaah terlebih dahulu mengecek apakah jamaah sudah memasukkan biodata calon santri kedalam sistem. Jika belum, jamaah harus melengkapi formulir biodata calon santri terlebih dahulu dan menyimpannya ke dalam sistem.
2. Jamaah lalu melengkapi formulir pendaftaran dan mengirim formulir pendaftaran tersebut kepada pengurus masjid
3. Admin menerima formulir dan memeriksa formulir tersebut. Jika formulir santri belum lengkap, maka ketua masjid akan menolak formulir pendaftaran calon santri dan calon santri dapat melengkapi formulir pendaftaran kembali. Jika formulir santri sudah lengkap, maka ketua masjid menyetujui formulir pendaftaran dan calon santri akan menerima surat diterima sebagai santri. Santri akan dinyatakan terdaftar di dalam sistem.

BPMN untuk proses pencatatan aset ini bisa dilihat pada gambar berikut:



Gambar 4.8 BPMN Pencatatan Santri Masjid yang Diusulkan

4.1.3 Analisis Kebutuhan Fungsional

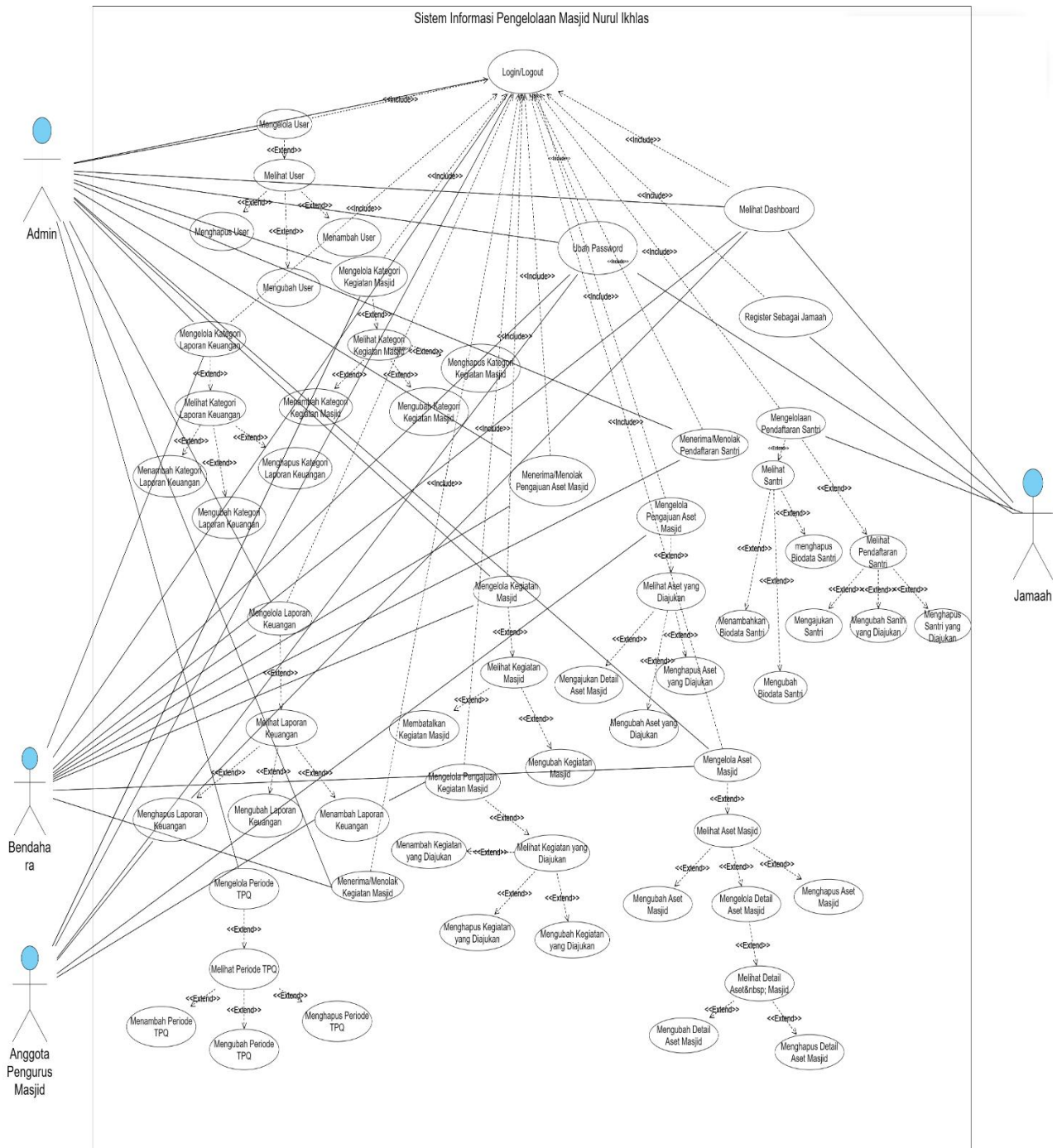
Berdasarkan analisis sistem yang diusulkan, maka kebutuhan fungsional dapat dirumuskan sebagai berikut :

1. Admin dapat menambah dan mengelola user.
2. Admin dapat mengelola periode TPQ.
3. Pengurus masjid dapat mengajukan kegiatan masjid.
4. Pengurus masjid dapat mengajukan pengadaan aset baru.
5. Jamaah dapat mendaftarkan santri.
6. Admin dan bendahara dapat menyetujui atau menolak pengajuan kegiatan.
7. Admin dan bendahara dapat menyetujui atau menolak pengajuan pengadaan aset baru.
8. Admin dan bendahara dapat menyetujui atau menolak pendaftaran santri.
9. Admin dan bendahara dapat mengelola kegiatan masjid.
10. Admin dan bendahara dapat mengelola keuangan masjid.
11. Admin dan bendahara dapat mengelola aset masjid.
12. Admin, bendahara, pengurus masjid, dan jamaah masjid dapat melihat dashboard sistem kegiatan dan keuangan masjid.
13. Admin, bendahara, pengurus masjid, dan jamaah masjid dapat *login* dan *logout* pada sistem.

4.1.4 Use Case Diagram

Use case diagram menggambarkan hubungan antara fungsional dengan aktor yang terlibat di dalam sistem. Pada perancangan sistem informasi kegiatan dan keuangan masjid terdapat tiga aktor yang terlibat langsung dalam sistem, yaitu admin, sekretaris masjid, dan jamaah masjid. Setiap aktor diharuskan untuk *login* sebelum menggunakan sistem. Aktor admin memiliki 11 fungsional, aktor sekretaris memiliki 9 fungsional dan aktor jamaah memiliki 4 fungsional. Fungsional dari masing-masing aktor dapat dilihat pada gambaran *use case diagram* dari aplikasi ini pada Gambar 4.9.





Gambar 4.9 Use Case Diagram

4.1.5 Definisi Tugas Aktor

Aktor yang terlibat dalam sistem ini memiliki peran dan tugasnya masing-masing seperti yang sudah digambarkan pada *use case diagram*. Definisi dari tugas masing-masing aktor tersebut dapat dilihat pada tabel 4.1.

Tabel 4.1 Definisi Tugas Aktor

No.	Nama Aktor	Deskripsi Tugas
1.	Admin	<ul style="list-style-type: none"> • Mengelola <i>website profile</i> masjid. Admin dapat mengunggah informasi keuangan, kegiatan, aset, dan jumlah santri masjid. • Melihat <i>dashboard</i> sistem kegiatan dan keuangan masjid. • Melakukan <i>login</i> dan <i>logout</i>. Admin dapat login untuk mengakses pengelolaan pada sistem kegiatan dan keuangan masjid. • Menambah dan mengelola user. Admin dapat menambah, mengubah, dan menghapus, user pada sistem. • Mengelola periode TPQ. Admin dapat menambah, mengubah, dan menghapus, periode TPQ pada sistem. • Mengelola keuangan masjid. Admin dapat menambah, mengubah, dan menghapus data keuangan masjid pada sistem. Admin dapat menambah, mengubah, dan menghapus kategori keuangan masjid pada sistem. • Mengelola kegiatan masjid. Admin dapat mengubah jadwal kegiatan, mengubah status kegiatan, dan menghapus kegiatan pada sistem. Admin dapat menambah, mengubah, atau menghapus kategori kegiatan masjid pada sistem. Menyetujui atau menolak pengajuan kegiatan masjid.

Tabel 4.1 Definisi Tugas Aktor (Lanjutan)

		<ul style="list-style-type: none"> • Mengelola aset masjid. Admin dapat mengubah kondisi aset, dan menghapus aset pada sistem. • Menyetujui atau menolak pengajuan aset baru masjid. • Menyetujui atau menolak pendaftaran santri.
2.	Pengurus masjid	<ul style="list-style-type: none"> • Melihat dashboard sistem kegiatan dan keuangan masjid. • Melakukan <i>login</i> dan <i>logout</i>. Pengurus masjid dapat login untuk mengakses pengelolaan pada sistem kegiatan dan keuangan masjid. • Mengajukan kegiatan masjid. Pengurus masjid dapat menambah, mengubah, dan menghapus kegiatan masjid yang diajukan pada sistem. • Mengajukan aset masjid. Pengurus masjid dapat menambah, mengubah, dan menghapus aset masjid yang diajukan pada sistem.
3.	Jamaah	<ul style="list-style-type: none"> • Melakukan <i>login</i> dan <i>logout</i>. Jamaah dapat login untuk mengakses pendaftaran calon santri. • Melihat dashboard sistem kegiatan dan keuangan masjid. • Mengelola biodata calon santri. Jamaah dapat menambah, mengubah, dan menghapus biodata calon santri dengan mengisi form yang ada. • Mendaftarkan santri. Jamaah dapat melakukan pendaftaran santri dengan mengisi form yang ada, form tersebut nantinya akan diperiksa oleh admin. Jamaah dapat mengelola data santri selama belum disetujui oleh admin.

Tabel 4.1 Definisi Tugas Aktor (Lanjutan)

4.	Bendahara	<ul style="list-style-type: none"> • Melihat dashboard sistem kegiatan dan keuangan masjid. • Melakukan <i>login</i> dan <i>logout</i>. Pengurus masjid dapat login untuk mengakses pengelolaan pada sistem kegiatan dan keuangan masjid. • Mengelola keuangan masjid. Pengurus masjid dapat menambah, mengubah, dan menghapus data keuangan masjid pada sistem Pengurus Masjid dapat menambah, mengubah, dan menghapus kategori keuangan masjid pada sistem. • Mengelola kegiatan masjid. Pengurus masjid dapat mengubah jadwal kegiatan, mengubah status kegiatan, dan menghapus kegiatan pada sistem. • Pengurus masjid dapat menambah, mengubah, atau menghapus kategori kegiatan masjid pada sistem. • Mengelola aset masjid. Pengurus masjid dapat menambah mengubah kondisi aset, dan menghapus aset pada sistem.
----	-----------	--

4.1.6 Use Case Scenario

Pada sub bab ini dibahas beberapa skenario dari *use case* yang terdapat di dalam pembuatan Sistem Informasi Pengelolaan Kegiatan Dan Keuangan Masjid Nurul Ikhlas Kecamatan Padang Utara Berbasis Web. *Use case scenario* yang ditampilkan pada sub bab ini hanya tiga, yaitu *use case scenario* pencatatan keuangan, pencatatan santri, dan pengajuan kegiatan. *Use case scenario* yang lain dapat dilihat pada lampiran A.

4.1.6.1 Use Case Scenario Pencatatan Keuangan

Use case scenario pencatatan keuangan adalah proses yang dilakukan oleh pengurus masjid dan admin untuk menambah data keuangan masjid. *Use case scenario* pencatatan keuangan dapat dilihat pada tabel 4.2.

Tabel 4.2 Use Case Scenario Pencatatan Keuangan

<i>Use Case</i>	Pencatatan keuangan (tambah)
<i>Actor</i>	Bendahara Admin
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “laporan keuangan masjid” 2. Sistem akan menampilkan halaman laporan keuangan masjid dan menampilkan data keuangan masjid. 3. <i>Actor</i> mengklik tombol “tambah” 4. Sistem akan menampilkan halaman tambah keuangan masjid dan menampilkan <i>form</i> tambah keuangan masjid. 5. <i>Actor</i> meng-<i>input</i> form keuangan masjid. 6. <i>Actor</i> mengklik tombol “submit” saat data benar. 7. Sistem akan menyimpan data keuangan masjid dan menampilkan halaman keuangan masjid serta menampilkan data keuangan masjid.
<i>Scenario Alternative</i>	<ol style="list-style-type: none"> 8. <i>Actor</i> menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan. 10. <i>Actor</i> menekan tombol “Submit” saat kategori keuangan kosong. 11. Sistem menampilkan notifikasi kesalahan.
<i>Exit Scenario</i>	<i>Actor</i> berhasil menambahkan data keuangan masjid.

4.1.6.2 Use Case Scenario Pendaftaran Santri

Use case scenario pendaftaran santri adalah proses yang dilakukan oleh jamaah untuk mendaftarkan diri menjadi santri. *Use case scenario* pencatatan keuangan dapat dilihat pada tabel 4.3.

Tabel 4.3 Use Case Scenario Registrasi Santri

Use Case	Pendaftaran santri masjid (tambah)
Actor	Jamaah
Entry Condition	<i>Actor</i> telah <i>login</i>
Flow of Event	<ol style="list-style-type: none">1. <i>Actor</i> membuka halaman “pendaftaran santri”.2. Sistem akan menampilkan halaman pendaftaran santri dan menampilkan <i>form</i> tambah pencatatan santri.3. <i>Actor</i> meng-<i>input</i> form pendaftaran santri.4. <i>Actor</i> mengklik tombol “submit” saat data benar.5. Sistem akan menyimpan data pendaftaran santri dan kembali ke halaman <i>dashboard</i>.
Scenario Alternative	<ol style="list-style-type: none">6. Actor menekan tombol “Submit” saat inputan data salah atau kosong.7. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	Actor berhasil menambahkan data santri.

4.1.6.3 Use Case Scenario Pengajuan Kegiatan

Use case scenario pengajuan kegiatan adalah proses yang dilakukan oleh pengurus masjid untuk mengajukan kegiatan yang akan dilakukan oleh masjid. *Use case scenario* pencatatan keuangan dapat dilihat pada tabel 4.4.

Tabel 4.4 Use Case Scenario Pengajuan Kegiatan

Use Case	Pengajuan Kegiatan (Tambah)
Actor	Pengurus masjid
Entry Condition	<i>Actor</i> telah <i>login</i>
Flow of Event	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “mengajukan kegiatan”. 2. Sistem akan menampilkan halaman pengajuan kegiatan dan menampilkan data pengajuan kegiatan. 3. <i>Actor</i> mengklik tombol “tambah” 4. Sistem akan menampilkan halaman tambah pengajuan kegiatan dan <i>form</i> tambah pengajuan kegiatan. 5. <i>Actor</i> meng-<i>input</i> data pengajuan kegiatan. 6. <i>Actor</i> mengklik tombol “submit” saat data benar. 7. Sistem akan menyimpan data kegiatan dengan menambahkan status pada data sebagai “diajukan”. Lalu Sistem akan kembali ke halaman pengajuan kegiatan dan menampilkan data pengajuan kegiatan.
Scenario Alternative	<ol style="list-style-type: none"> 8. <i>Actor</i> menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	<i>Actor</i> berhasil menambahkan pengajuan kegiatan.

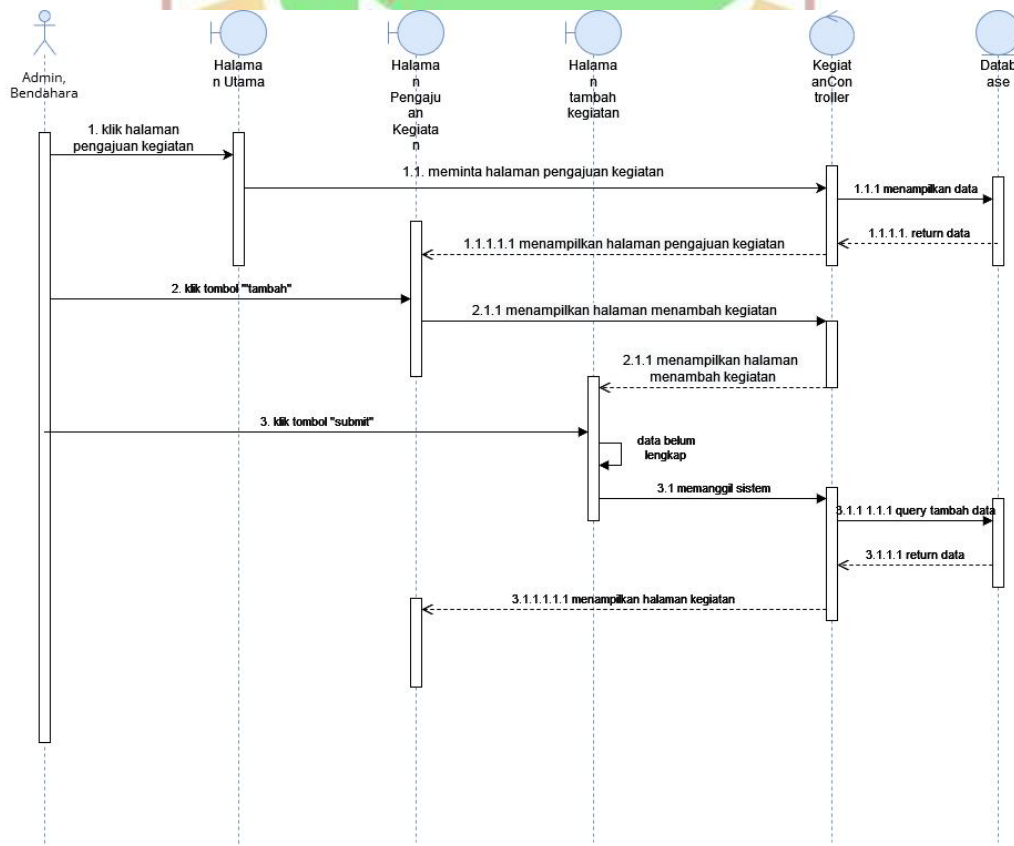
4.1.7 Sequence Diagram

Pada bagian ini digambarkan *sequence diagram* yang mencakup semua proses yang berlangsung di dalam sistem. *Sequence diagram* ini dibuat berdasarkan hasil analisis *use case diagram*, dan kemudian dikembangkan sampai dengan proses terkecil yang ada pada setiap *use case*. Pada beberapa proses diasumsikan bahwa *user* telah melakukan *login* terlebih dahulu sebelum masuk proses berikutnya. *Sequence diagram* yang ditampilkan pada sub bab ini hanya ada tiga, yaitu *sequence diagram* pengajuan kegiatan (tambah),

pencatatan keuangan (tambah) dan pencatatan santri. Untuk *sequence diagram* lainnya dijelaskan pada bagian lampiran B.

4.1.7.1 Sequence Diagram Pengajuan Kegiatan (Tambah)

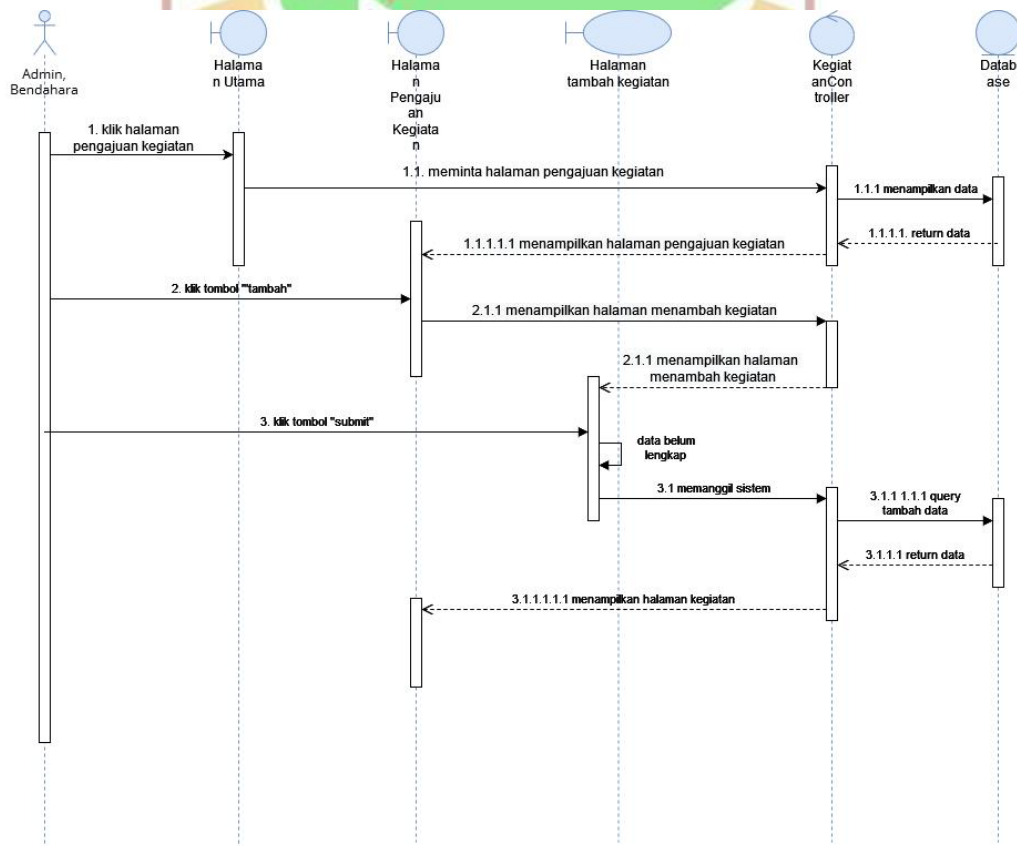
Pada fungsional pengajuan kegiatan ini, pengurus masjid mengajukan kegiatan masjid. Proses pengajuan kegiatan dimulai dengan memilih halaman pengajuan kegiatan, mengklik tombol “tambah”, mengisi *form* pengajuan kegiatan dan terakhir mengklik tombol *submit* pada halaman tambah kegiatan. *Controller* pada input data pengajuan kegiatan akan memproses data sesuai perintah yang dijalankan dari *view* ke *model* dengan mengirimkan data pengajuan kegiatan untuk disimpan dalam *database*. Jika saat mengklik tombol *submit* ada data kegiatan yang masih kosong maka ketika data dikirimkan, akan muncul peringatan di tempat input dari sistem bahwa data yang dimasukkan belum lengkap. *Sequence diagram* pengajuan kegiatan dapat dilihat pada gambar 4.10.



Gambar 4.10 Sequence Diagram Pengajuan Kegiatan

4.1.7.2 Sequence Diagram Pencatatan Keuangan (Tambah)

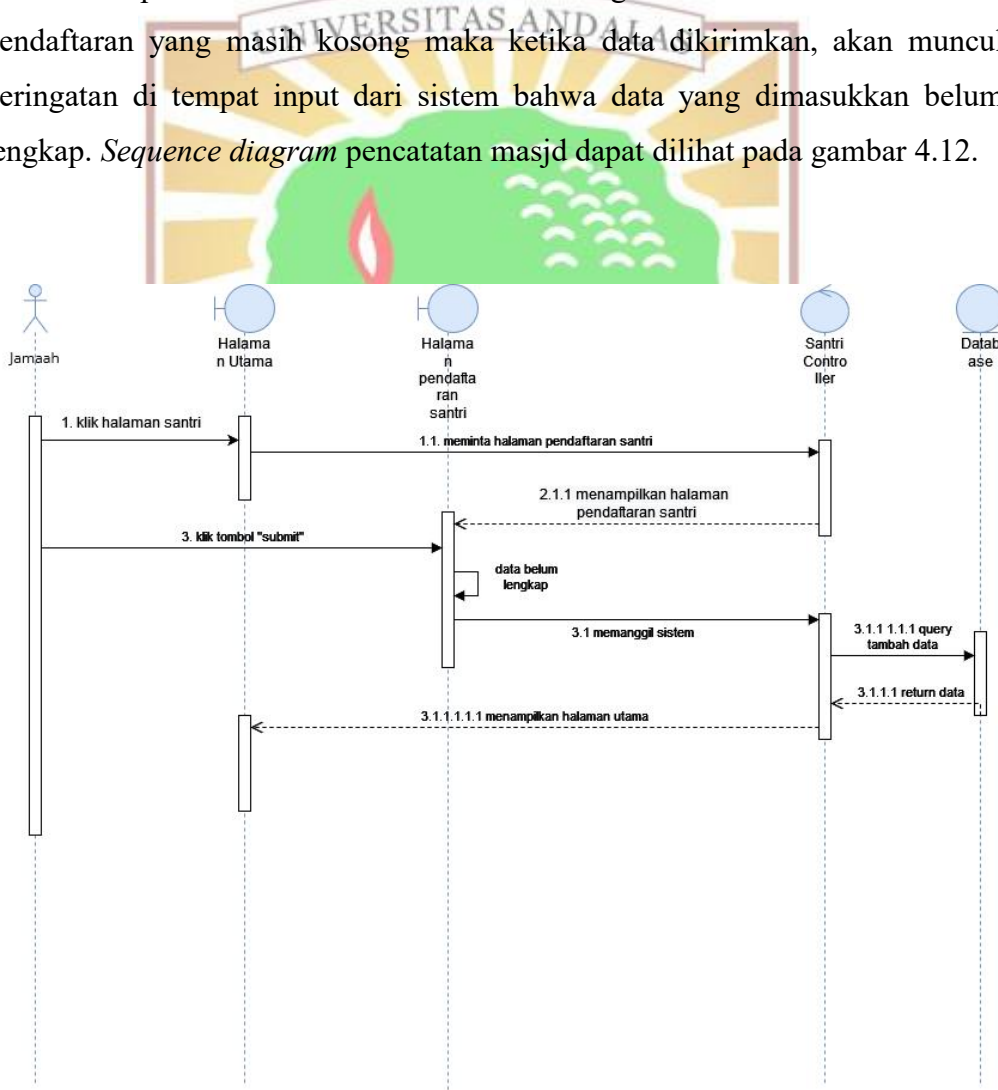
Sequence diagram pencatatan keuangan adalah proses yang dilakukan oleh admin atau pengurus masjid untuk menambah data keuangan masjid. Proses pencatatan keuangan dimulai dengan memilih halaman debet & kredit, mengklik tombol “tambah”, mengisi *form* pencatatan keuangan dan terakhir mengklik tombol *submit* pada halaman tambah kegiatan. *Controller* pada input data pengajuan kegiatan akan memproses data sesuai perintah yang dijalankan dari *view* ke *model* dengan mengirimkan data pengajuan kegiatan untuk disimpan dalam *database*. Jika saat mengklik tombol *submit* ada data keuangan yang masih kosong maka ketika data dikirimkan, akan muncul peringatan di tempat input dari sistem bahwa data yang dimasukkan belum lengkap. *Sequence diagram* menambah pencatatan keuangan dapat dilihat pada gambar 4.11.



Gambar 4.11 Sequence Diagram Pencatatan Keuangan

4.1.7.3 Sequence Diagram Mendaftarkan Santri

Sequence diagram pencatatan santri adalah proses yang dilakukan oleh jamaah untuk mendaftar santri. Proses pencatatan keuangan dimulai dengan memilih halaman pendaftaran santri, mengisi *form* pendaftaran santri, dan terakhir mengklik tombol *submit* pada halaman tambah kegiatan. *Controller* pada input data pengajuan kegiatan akan memproses data sesuai perintah yang dijalankan dari *view* ke *model* dengan mengirimkan data pengajuan kegiatan untuk disimpan dalam *database*. Jika saat mengklik tombol *submit* ada data pendaftaran yang masih kosong maka ketika data dikirimkan, akan muncul peringatan di tempat input dari sistem bahwa data yang dimasukkan belum lengkap. *Sequence diagram* pencatatan masjid dapat dilihat pada gambar 4.12.



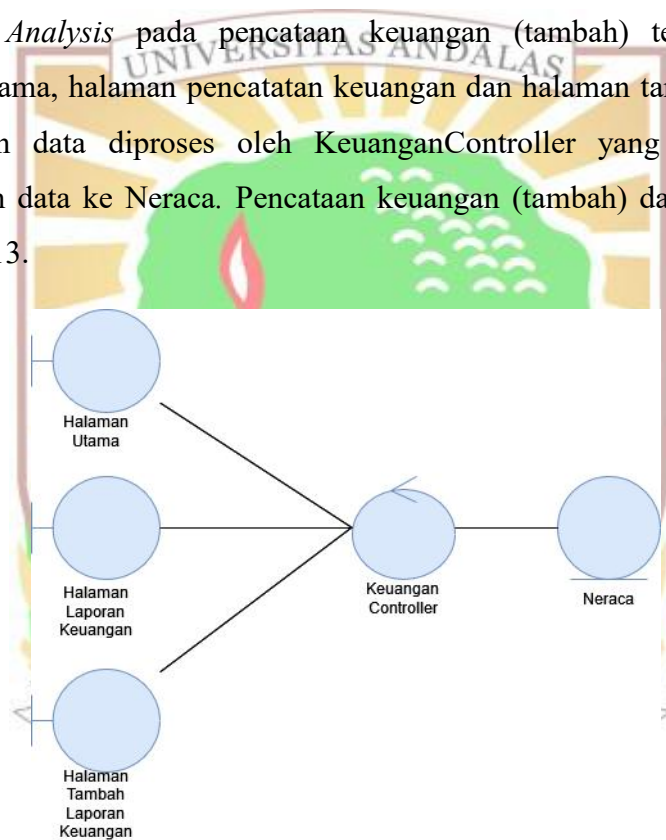
Gambar 4.12 Sequence Diagram Pencatatan Santri

4.1.8 Class Analysis

Class analysis berfungsi untuk menggambarkan semua kelas yang terlibat di dalam setiap proses dan dibuat berdasarkan *sequence diagram* sistem yang telah dirancang. *Class analysis* yang ditampilkan pada sub bab ini ada tiga, yaitu *class analysis* pengajuan pencatatan keuangan (tambah), pengajuan kegiatan (tambah), dan pendaftaran santri. Untuk *class analysis* lainnya dipaparkan pada bagian lampiran C.

4.1.8.1 Class Analysis Pencatatan Keuangan (Tambah)

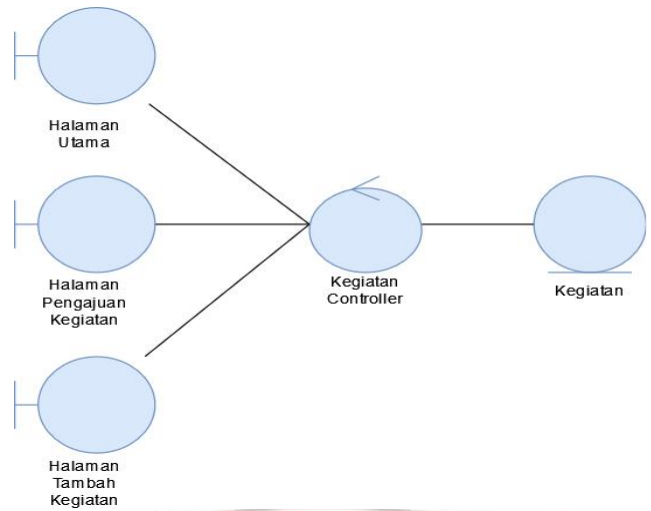
Class Analysis pada pencatatan keuangan (tambah) terdiri atas satu halaman utama, halaman pencatatan keuangan dan halaman tambah keuangan. Pengelolaan data diproses oleh *KeuanganController* yang mengirim dan menyimpan data ke *Neraca*. Pencatatan keuangan (tambah) dapat dilihat pada Gambar 4.13.



Gambar 4.13 Class Analysis Pencatatan Keuangan

4.1.8.2 Class Analysis Pengajuan Kegiatan (Tambah)

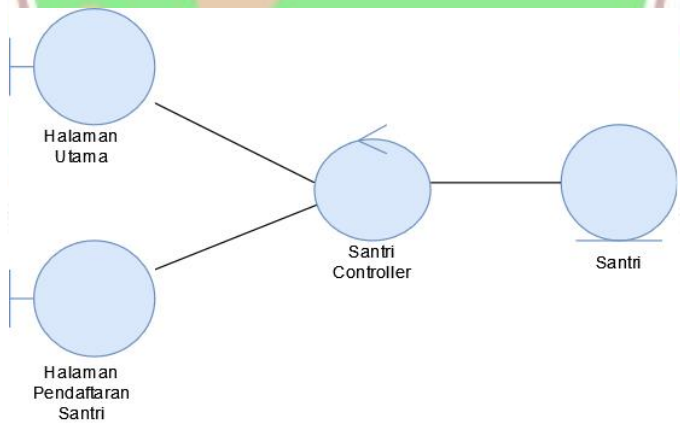
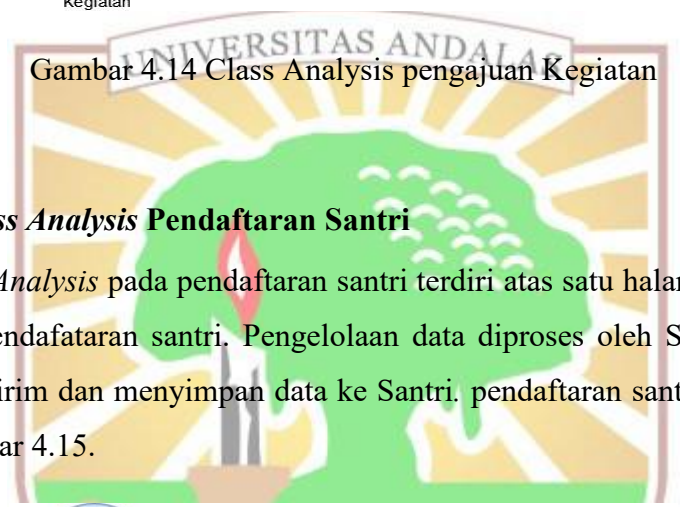
Class Analysis pada pengajuan kegiatan (tambah) terdiri atas satu halaman utama, halaman pengajuan kegiatan dan halaman tambah kegiatan. Pengelolaan data diproses oleh *KegiatanController* yang mengirim dan menyimpan data ke *Kegiatan*. Pencatatan keuangan (tambah) dapat dilihat pada Gambar 4.14.



Gambar 4.14 Class Analysis pengajuan Kegiatan

4.1.8.3 Class Analysis Pendaftaran Santri

Class Analysis pada pendaftaran santri terdiri atas satu halaman utama dan halaman pendaftaran santri. Pengelolaan data diproses oleh SantriController yang mengirim dan menyimpan data ke Santri. pendaftaran santri dapat dilihat pada Gambar 4.15.



Gambar 4.15 Class Analysis Pendaftaran Santri

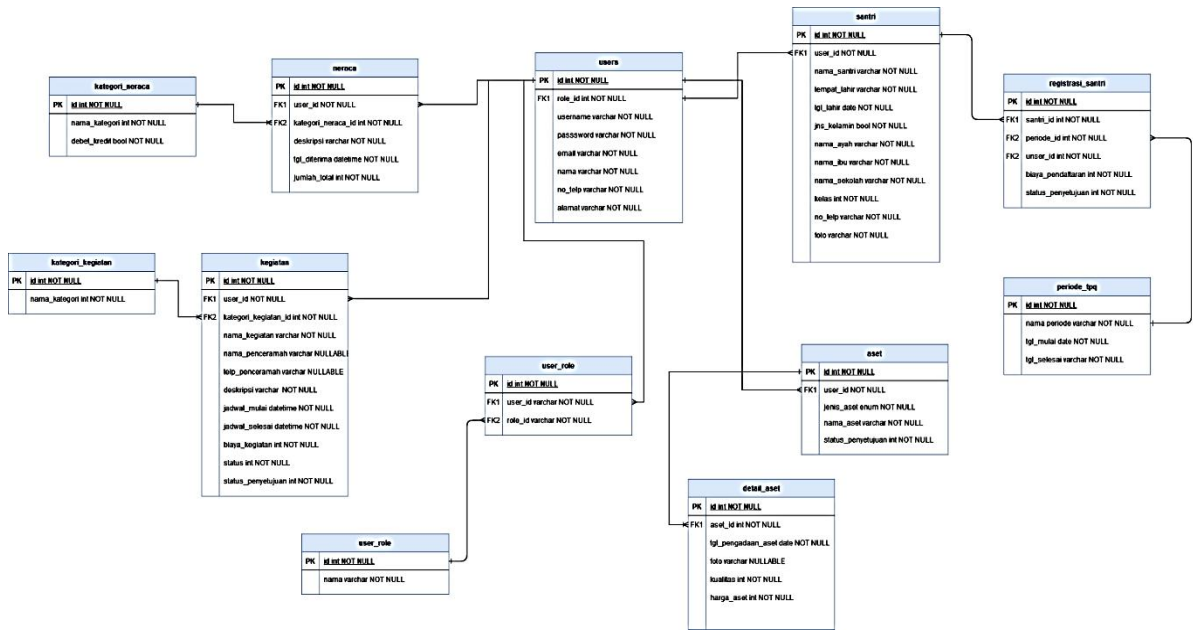
4.2 Perancangan Sistem

Berdasarkan tahapan analisis proses bisnis yang sedang berjalan, alur sistem, dan kebutuhan sistem informasi yang dibangun maka diperoleh hasil yang menjadi dasar dan tolak ukur untuk melakukan perancangan sistem. Perancangan sistem ini meliputi perancangan *database*, struktur tabel dan basis data, arsitektur aplikasi antarmuka (*user interface*), dan diagram kelas (*class diagram*).

4.2.1 Perancangan Database

Perancangan *database* dimulai dengan menyusun struktur *database* berdasarkan entitas yang digunakan dan hubungannya dengan entitas-entitas lain. Entitas beserta hubungannya digambarkan dalam sebuah ERD (*Entity Relationship Diagram*). Rancangan ERD dapat dilihat pada gambar 4.16.





Gambar 4.17 Rancangan Basis Data

4.2.2 Struktur Tabel dan Basis Data

Struktur tabel dan basis data merupakan representasi tabel masing-masing relasi beserta status atributnya. Status tersebut adalah *primary key*, *foreign key*, *tipe data*, nama atribut, dan nama tabel. Berikut adalah uraian tabel neraca, tabel kegiatan, dan tabel aset yang dijelaskan pada tabel 4.5 sampai dengan tabel 4.7, dan untuk uraian tabel lainnya dapat dilihat pada lampiran D.

Tabel 4.5 Struktur Tabel Neraca

Nama Atribut	Tipe Data	Keterangan
id	Int(10)	PRIMARY KEY
user_id	Int(10)	FOREIGN KEY
kategori_neraca_id	Int(10)	FOREIGN KEY
deskripsi	Varchar(255)	
tgl_diterima	date	
jumlah_total	Int(10)	

Tabel 4.6 Struktur Tabel Kegiatan

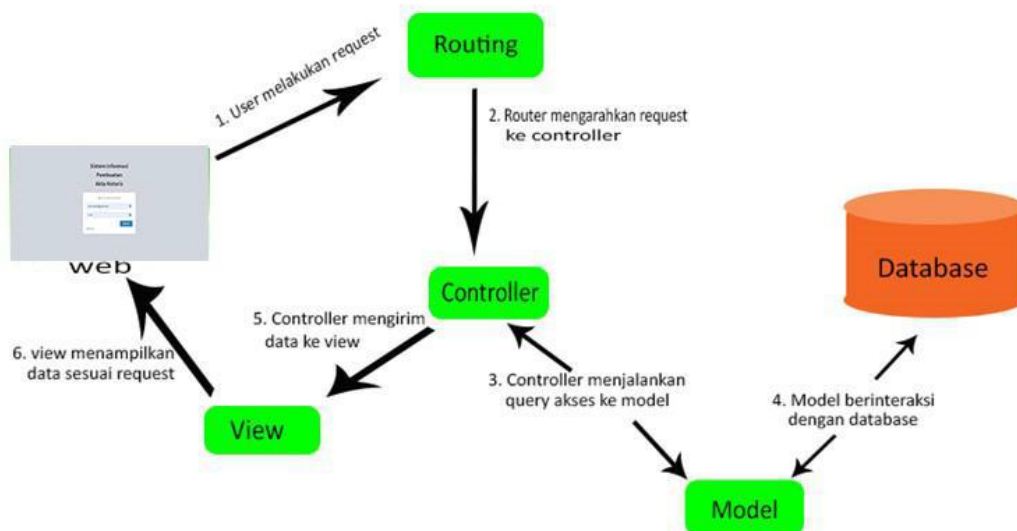
Nama Atribut	Tipe Data	Keterangan
id	Int(10)	PRIMARY KEY
user_id	Int(10)	FOREIGN KEY
kategori_kegiatan_id	Int(10)	FOREIGN KEY
nama_kegiatan	Varchar(10)	
nama_penceramah	Varchar(10)	NULLABLE
telp_penceramah	Varchar(10)	NULLABLE
deskripsi	Varchar(255)	
jadwal_mulai	Datetime	
jadwal_selesai	Datetime	
status	Int(2)	
status_penyetujuan	Int(2)	
biaya kegiatan	Int(10)	

Tabel 4.7 Struktur Tabel Aset

Nama Atribut	Tipe Data	Keterangan
id	Int(10)	PRIMARY KEY
user_id	Int(10)	FOREIGN KEY
jenis_aset	enum('Aset Lancar', 'Aset Tidak Lancar')	
nama_aset	Varchar(10)	
status_penyetujuan	Int(2)	

4.2.3 Arsitektur Aplikasi

Arsitektur aplikasi yang digunakan dalam perancangan dan pembangunan aplikasi ini adalah arsitektur aplikasi MVC (*Model View Controller*) dengan metode OOP (*Object Oriented Programming*) menggunakan *framework* Laravel untuk pembangunan aplikasi web. Arsitektur pembangunan aplikasi dapat dilihat pada Gambar 4.18.



Gambar 4.18 Arsitektur Aplikasi (MVC)

Berdasarkan Gambar 4.18 mengenai arsitektur aplikasi terlihat bahwa pada perancangan aplikasi web digunakan arsitektur yang memisahkan antara data (*model*), tampilan antar muka (*view*), dan pengarah (*controller*). Ketika *user* melakukan permintaan (*request*) terhadap sistem, maka *routing* mengarahkan kepada *controller* bersama dengan *method* yang dibutuhkan. *Controller* kemudian memanggil data melalui *model*. Kemudian *model* berinteraksi dan mengambil data dari *database*. Data yang didapatkan *model* dikirim kembali ke *controller* untuk kemudian ditampilkan oleh *view* ke halaman *user*.

4.2.4 Class Diagram Aplikasi

Class diagram merupakan gambaran mengenai struktur kelas-kelas dari suatu sistem dan merupakan salah satu diagram yang banyak dipakai. Dengan adanya *class diagram* diharapkan pembangunan sistem dapat terbantu dalam membangun sistem pada tahapan implementasi. Pada *class diagram* terdapat *view*, *model*, dan *controller*. *Class diagram* juga memaparkan atribut dari masing-masing kelas beserta fungsi-fungsi yang terkandung didalamnya. Fungsi yang ada pada masing-masing kelas memiliki tugas tersendiri sesuai kebutuhan

sistem. Untuk membangun sistem yang terintegrasi, masing-masing kelas memiliki relasi dengan kelas lainnya. *Class diagram* aplikasi dapat dilihat pada gambar 4.19.

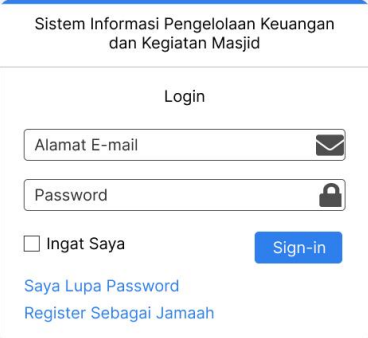


4.2.5 Perancangan Antarmuka

Antarmuka merupakan sistem komunikasi antara pengguna (*user*) dengan sistem. Antarmuka pengguna (*user interface*) dapat menerima dan memberikan informasi kepada pengguna untuk membantu mengarahkan alur pencarian masalah sampai ditemukan suatu solusi. Penjelasan beberapa antarmuka aplikasi *web* dan yang dibangun dapat dilihat pada gambar 4.20, 4.21, 4.22. Antarmuka aplikasi lainnya dapat dilihat pada lampiran E.

4.2.5.1 Perancangan Antarmuka Aplikasi Web

Perancangan antarmuka aplikasi *web* yang dijelaskan pada bagian ini adalah antarmuka halaman awal, halaman laporan keuangan dan halaman aset masjid. Perancangan antarmuka halaman awal dapat dilihat pada gambar 4.19.



The image shows a login form for a web application. At the top, the title reads "Sistem Informasi Pengelolaan Keuangan dan Kegiatan Masjid". Below the title is the word "Login". There are two input fields: "Alamat E-mail" with an envelope icon and "Password" with a lock icon. Below the password field is a checkbox labeled "Ingat Saya" and a blue "Sign-in" button. At the bottom, there are two links: "Saya Lupa Password" and "Register Sebagai Jamaah".

Gambar 4.20 Antarmuka Halaman Awal

Berdasarkan antarmuka halaman awal dari gambar 4.20, halaman awal merupakan halaman yang ditampilkan saat mengakses aplikasi. Halaman awal menampilkan *form login* sebelum *user* masuk ke dalam sistem. Setelah *user* berhasil *login*, sistem akan menampilkan halaman utama. Selanjutnya antarmuka halaman laporan keuangan dapat dilihat pada gambar 4.20.

≡ Profil User
Logout

● User Name

Navigasi

- Dashboard
- Laporan Keuangan
- Kegiatan Masjid
- Aset Masjid
- Santri Masjid

Laporan Keuangan

Pilih Data Bulan --Semua Bulan-- Tahun 2023 Submit

Rp 7.000.500

TOTAL DEBET

Rp 1.500.500

TOTAL KREDIT

Rp 7.000.500

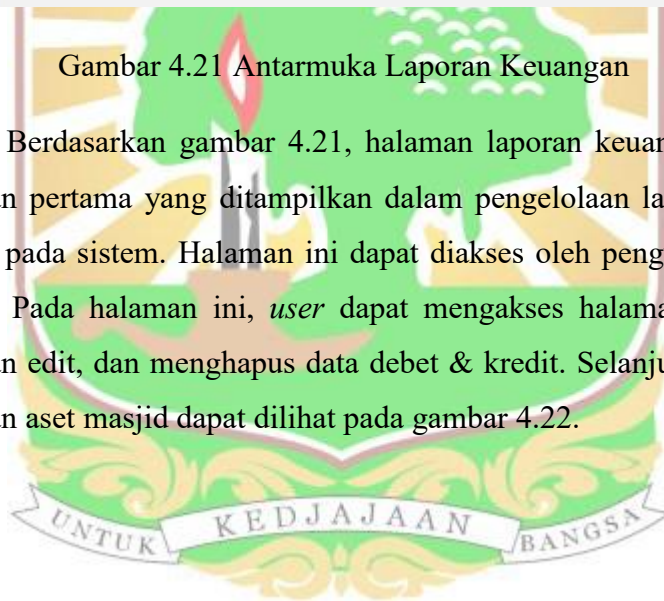
SISA SALDO

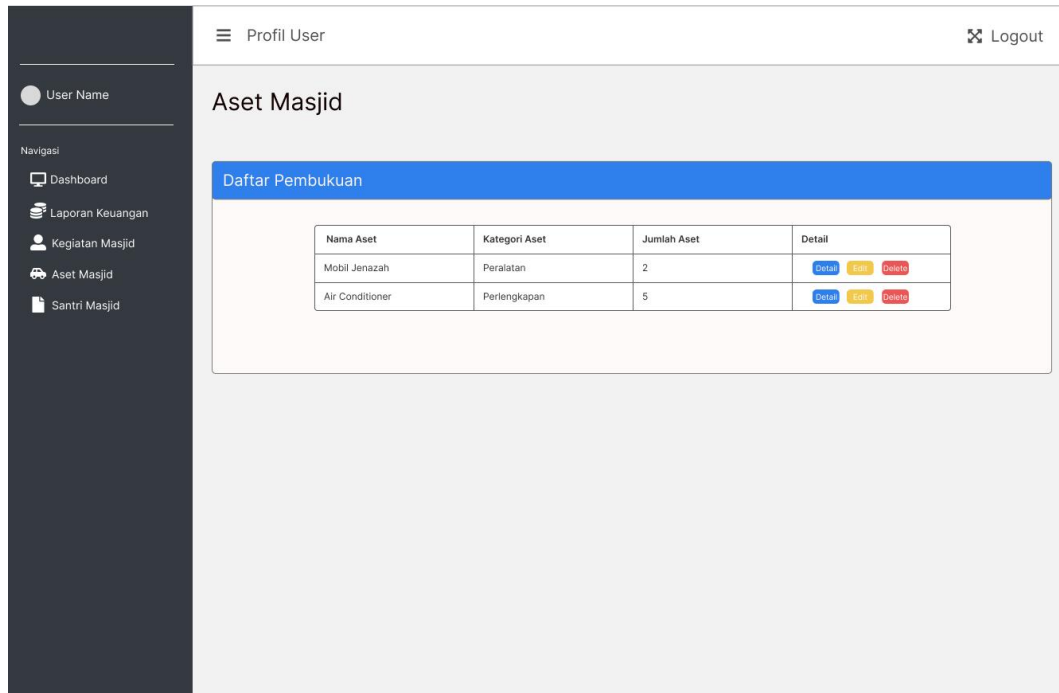
Daftar Pembukuan

Tgl. Input	Deskripsi	Kategori	No.Invoice	Tanggal Diterima	Debet (Rp)	Kredit (Rp)	Saldo (Rp)	Detail
19 March 2023 13:45:26	Uang Garin	Uang Garin	04	14 February 2023	0	Rp 400.000	- Rp 400.000	Edit Delete
19 March 2023 13:47:21	Infak	Infak	01	04 January 2023	Rp 2.000.000	0	Rp 2.000.000	Edit Delete
19 March 2023 14:06:25	Perfengkapan Masjid	Perfengkapan Masjid	05	19 March 2023	0	Rp 1.100.000	- Rp 1.100.000	Edit Delete
25 March 2023 12:44:49	Donasi	Donasi	03	25 March 2023	Rp 5.000.500	0	Rp 5.000.500	Edit Delete

Gambar 4.21 Antarmuka Laporan Keuangan

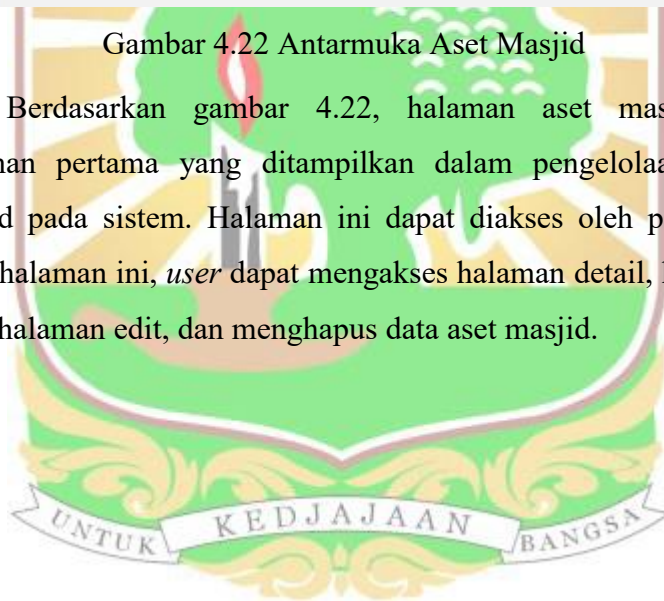
Berdasarkan gambar 4.21, halaman laporan keuangan merupakan halaman pertama yang ditampilkan dalam pengelolaan laporan keuangan masjid pada sistem. Halaman ini dapat diakses oleh pengurus masjid dan admin. Pada halaman ini, *user* dapat mengakses halaman tambah data, halaman edit, dan menghapus data debit & kredit. Selanjutnya antarmuka halaman aset masjid dapat dilihat pada gambar 4.22.





Gambar 4.22 Antarmuka Aset Masjid

Berdasarkan gambar 4.22, halaman aset masjid merupakan halaman pertama yang ditampilkan dalam pengelolaan laporan aset masjid pada sistem. Halaman ini dapat diakses oleh pengurus masjid. Pada halaman ini, *user* dapat mengakses halaman detail, halaman tambah data, halaman edit, dan menghapus data aset masjid.



BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini dijelaskan tentang implementasi sistem sesuai dengan perancangan yang dilakukan pada bab sebelumnya dan pengujian kesesuaian sistem yang dibangun dengan alur proses yang diusulkan. Pada implementasi ini dijelaskan tentang hal-hal yang berhubungan dengan alur perangkat lunak dan spesifikasi yang dibutuhkan pada saat penelitian, lalu pada pengujian menjelaskan tentang fungsional yang ada pada aplikasi apakah berjalan sesuai perancangan atau tidak. Tahap implementasi dan tahap pengujian sistem dilakukan pada aplikasi *web*.

5.1. Implementasi Sistem

Aplikasi pengelolaan kegiatan dan keuangan masjid ini difungsikan sebagai aplikasi yang menyimpan dan mengelola data keuangan dan kegiatan masjid. Pada sistem ini terdapat tiga aktor yang terlibat dan empat belas fungsional yang dapat dijalankan. Fungsional tersebut mewakili setiap tugas dari *user* dan diharapkan dapat memberikan efisiensi dan efektifitas terhadap semua proses pada sistem.

Implementasi sistem ini dilakukan dengan menggunakan perangkat keras komputer dengan spesifikasi sebagai berikut:

1. Komputer dengan *processor AMD Ryzen 5 4600H (12 CPUs), 3.0 GHz*.
2. Memori Komputer (RAM) 16 GB.
3. *Harddisk* Komputer dengan kapasitas 500 GB.

Spesifikasi perangkat lunak yang digunakan dalam implementasi sistem ini adalah sebagai berikut:

1. Sistem Operasi *Windows 11 Home Single Language*.
2. *Web browser* yang digunakan adalah *Mozilla Firefox* versi 111.0.1.

Berikut ini spesifikasi sistem yang dibutuhkan untuk implementasi sistem:

1. Sistem operasi Windows 7, 8, 8.1, 10, 11.
2. *Web server* dengan modul PHP 8.1.7 atau setelahnya.
3. *Web browser* yang mendukung HTML 5.

5.1.1. Source Code Program

Pada bagian *source code* program ini dijelaskan kode program yang dibuat berdasarkan arsitektur aplikasi yang digunakan. Bagian *view*, *model*, dan *controller* dijelaskan secara terpisah berdasarkan fungsi dan kegunaan masing-masing. Berikut dijelaskan fungsi dan kegunaan masing-masing bagian pada aplikasi. Untuk penjelasan kode program lebih lanjut dapat dilihat pada lampiran F.

5.1.1.1. Source Code Routing Aplikasi

Routing merupakan bagian yang mengatur arah perpindahan permintaan (*request*) dari *user*. *Route* dapat mengendalikan semua perintah yang dideklarasikan dan mengirimkannya kepada *controller* serta *method* yang sesuai. Kode program *routing* aplikasi dapat dilihat pada Tabel 5.1.



```

Route::get('/', function () {
    return view('auth.login');
});
Route::middleware('auth')->group(function(){
    Route::get('/santri_registration/accepted_list',
[SantriRegistrationController::class, 'acceptedList'])-
>name('santri_registration.accepted_list');
    Route::post('/santri_registration/search_list',
[SantriRegistrationController::class, 'searchList'])-
>name('santri_registration.search_list');
});
Route::get('/home', [DashboardController::class, 'index']->middleware('auth')-
>name('dashboard');
Route::get('/password_update', [ProfileController::class, 'password_update']-
>name('password_update');
Route::resource('/profile', ProfileController::class->middleware('auth')-
>only(['index']));
Route::prefix('admin')->middleware(['auth', 'auth.accessAdmin']->name('admin.'))-
>group(function(){
    Route::resource('/user', UserController::class);
    Route::patch('/accept_santri/accept_checked', [AcceptSantriController::class,
'accept_checked']->name('accept_santri.accept_checked'));
    Route::patch('/accept_santri/accept/{id}', [AcceptSantriController::class,
'accept']->name('accept_santri.accept'));
    Route::patch('/accept_santri/deny/{id}', [AcceptSantriController::class,
'deny']->name('accept_santri.deny'));
});

```

Tabel 5.1 Kode Program Routing Aplikasi



Tabel 5.1 Kode Program Routing Aplikasi (lanjutan)

```

Route::resource('/accept_santri', AcceptSantriController::class)-
>only(['index', 'show']);
Route::patch('/accept_asset/accept_checked', [AcceptAssetController::class,
'accept_checked']->name('accept_asset.accept_checked');
Route::patch('/accept_asset/accept/{id}', [AcceptAssetController::class,
'accept']->name('accept_asset.accept');
Route::patch('/accept_asset/deny/{id}', [AcceptAssetController::class,
'deny']->name('accept_asset.deny');
Route::resource('/accept_asset', AcceptAssetController::class)-
>only(['index', 'show']);
Route::patch('/accept_activity/accept_checked',
[AcceptActivityController::class, 'accept_checked']->
>name('accept_activity.accept_checked');
Route::patch('/accept_activity/accept/{id}',
[AcceptActivityController::class, 'accept']->name('accept_activity.accept');
Route::patch('/accept_activity/deny/{id}', [AcceptActivityController::class,
'deny']->name('accept_activity.deny');
Route::resource('/accept_activity', AcceptActivityController::class)-
>only(['index', 'show']);
});
Route::middleware(['auth', 'auth.accessSekre']->group(function(){
Route::get('/asset/propose', [AssetController::class, 'propose']->
>name('asset.propose');
Route::get('/activity/propose', [ActivityController::class, 'propose']->
>name('activity.propose');
});
Route::middleware(['auth', 'auth.accessJamaah']->group(function(){
Route::get('/santri/propose', [SantriController::class, 'propose']->
>name('santri.propose');
Route::resource('/santri', SantriController::class);
Route::resource('/santri_registration',
SantriRegistrationController::class);
});
Route::middleware(['auth', 'auth.accessAdminAndBendahara']->group(function(){
Route::resource('/balance', BalanceController::class);
Route::resource('/balance_categories', BalanceCategoryController::class);
Route::post('/balance/search', [BalanceController::class, 'search']->
>name('balance.search');
Route::resource('/asset', AssetController::class);
Route::resource('/asset_categories', AssetCategoryController::class);
Route::resource('/asset_detail', AssetDetailController::class);
Route::patch('/activity/cancel/{id}', [ActivityController::class,
'cancel']->name('activity.cancel');
Route::post('/activity/search', [ActivityController::class, 'search']->
>name('activity.search');
Route::resource('/activity', ActivityController::class);
Route::resource('/activity_categories', ActivityCategoryController::class);
Route::resource('/tpq_period', TpqPeriodController::class);
});

```

5.1.1.2. Source Code Model Aset

Model merupakan bagian yang berfungsi memanggil data dari *database* serta mengirimkan data ke *controller* dengan menggunakan syntax SQL (*Structured Query Language*). Salah satu *model* yang dibangun adalah *model* Aset untuk memanggil entitas akta pada *database*. Kode program *model* Aset dapat dilihat pada Tabel 5.2.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Asset extends Model
{
    protected $fillable = ['user_id',
'asset_name', 'submission_status', 'asset_categories_id'];
    public function AssetCategory(){
        return $this->belongsTo('App\Models\AssetCategory',
'asset_categories_id');
    }
    public function AssetDetail(){
        return $this->hasMany('App\Models\AssetDetail', 'asset_id', 'id');
    }
    public function totalAsset($id){
        $total = $this->AssetDetail()->where('asset_id', $id)->count();
        if ($total < 1) {
            return "Data Aset belum dimasukkan";
        } else {
            return $total;
        }
    }
    public function users()
    {
        return $this->belongsTo('App\Models\User', 'user_id');
    }
    use HasFactory;
}
```

Tabel 5.2 Kode Program Model Aset

5.1.1.3. Source Code Controller Santri

Controller merupakan bagian yang mengatur seluruh alur dan *method* pada aplikasi sistem informasi dengan menggunakan arsitektur MVC. Salah satu *controller* yang dibangun adalah *SantriController*. Kode program *SantriController* dapat dilihat pada Tabel 5.3.

```
use App\Models\Santri;
use App\Models\User;
use Illuminate\Http\Request;
class SantriController extends Controller
{
    public function index()
    {
        $user = User::find(auth()->user()->id);
        $santries = $user->Santries()->get();
        return view('backend.santri.index', compact('santries'));
    }
    public function create()
    {
        return view('backend.santri.create');
    }
    public function store(Request $request)
    {
        $santri = new Santri;
        $santri->user_id = $request->user_id;
        $santri->santri_name = $request->santri_name;
        $santri->tpq_grade = $request->tpq_grade;
        $santri->birth_place = $request->birth_place;
        $santri->birth_date = $request->birth_date;
        $santri->sex = $request->sex;
        $santri->address = $request->address;
        $santri->father_name = $request->father_name;
        $santri->mother_name = $request->mother_name;
        $santri->school_name = $request->school_name;
        $santri->school_grade = $request->school_grade;
        $santri->telp_number = $request->telp_number;
        if ( isset($request->photo) ) {
            $path=$request->file('photo')->store('santri_photos', 'public');
            $santri->photo = '../../storage/'.$path;
        }$santri->save();return redirect()->route('santri.propose')->with('success',
'Data berhasil ditambahkan');}
```

Tabel 5.3 Kode Program SantriController




```

public function show(Santri $santri)
{
    $santri->find($santri->id);
    return view('backend.santri.show', compact('santri'));
}
public function edit(Santri $santri)
{
    $santri->find($santri->id);
    // dd($santri);
    return view('backend.santri.edit', compact('santri'));
}
public function update(Request $request, Santri $santri)
{
    // dd($santri->id);
    // Santri::find($santri->id)->update($request->all());
    $santri->user_id = $request->user_id;
    $santri->santri_name = $request->santri_name;
    $santri->tpq_grade = $request->tpq_grade;
    $santri->birth_place = $request->birth_place;
    $santri->birth_date = $request->birth_date;
    $santri->sex = $request->sex;
    $santri->address = $request->address;
    $santri->father_name = $request->father_name;
    $santri->mother_name = $request->mother_name;
    $santri->school_name = $request->school_name;
    $santri->school_grade = $request->school_grade;
    $santri->telp_number = $request->telp_number;
    if ( isset($request->photo) ) {
        $path=$request->file('photo')->store('santri_photos', 'public');
        $santri->photo = '../../storage/'.$path;
    }$santri->save();return redirect()->route('santri.propose')->with('success',
'Data berhasil diubah');
}
public function destroy(Santri $santri)
{
    Santri::destroy($santri->id);
    return redirect()->route('santri.index')->with('error', 'Data berhasil
dihapus');
}
public function propose()
{
    $user = User::find(auth()->user()->id);
    $santries = $user->Santries()->get();

    return view('backend.santri.propose', compact('santries'));
}

```

5.1.1.4. *Source Code View Index Laporan Keuangan*

View merupakan antar muka aplikasi yang berinteraksi langsung dengan user. Data dari database dan perintah dari controller kemudian ditampilkan pada *view*. Salah satu *view* yang dibangun adalah *view* neraca `index.blade.php` yang berfungsi untuk menampilkan halaman “laporan keuangan”. Kode program laporan keuangan dapat dilihat pada Tabel 5.4.



```

@extends('layouts/main')
@section('title')
    <title>SMKK | Laporan Keuangan</title>
@endsection
@section('page_name')
    <h1>Laporan Keuangan</h1>
@endsection
@section('breadcrumb')
    {!!
        breadcrumb([
            'Laporan Keuangan' => 'Laporan Keuangan'
        ])
    !!}
@endsection

@section('content')
    <div class="card">
        <div class="card-box">
            <form action="{{ route('balance.search') }}" method="POST">
                @csrf
                <div class="text-center mt-5">
                    <div class="form-group">
                        <div class="row">
                            <h5 class="col-2 mt-1">Pilih Data</h5>
                            <div class="col-4 ">
                                <div class="input-group input-group-sm">
                                    <div class="input-group-prepend">
                                        <span class="input-group-text">Bulan</span>
                                    </div>
                                    <select class="custom-select" name="month[]"
id="month">
                                        @foreach ( monthNameArray() as $value =>
$month_name)
                                            <option value="{{ $value }}" {{ $month ==
$value ? 'selected' : '' }} {{ $month_name }}</option>

```

Tabel 5.4 Kode Program Laporan Keuangan

```

        @endforeach
    </select>
</div>
</div>
</div>
<div class="col-4">
    <div class="input-group input-group-sm">
        <div class="input-group-prepend">
            <span class="input-group-text">Tahun</span>
        </div>
        <select class="custom-select" name="year[]" id="year">
            @foreach ($years as $year_list => $value)
                <option value="{{ $value }}" {{ $value ==
$year ? 'selected' : '' }} >{{ $value }}</option>
            @endforeach
        </select>
<div class="card card-primary">
    <div class="card-header">
        <h5 class="card-title">Daftar Pembukuan</h5>
    </div>
    <div class="card-body">
        <table id="example1" class="table table-bordered table-hover dataTable dtr-
inline" role="grid" aria-describedby="example1_info">
            <thead>
                <tr role="row">
                    <th class="sorting sorting_asc" tabindex="0" aria-
controls="example1" rowspan="1" colspan="1" aria-label="Rendering engine: activate to
sort column descending" aria-sort="ascending">Tgl Input</th>
                    <th class="sorting" tabindex="0" aria-controls="example1"
rowspan="1" colspan="1" aria-label="Browser: activate to sort column
ascending">Deskripsi</th><th class="sorting" tabindex="0" aria-controls="example1"
rowspan="1" colspan="1" aria-label="Platform(s): activate to sort column
ascending">Kategori</th><th class="sorting" tabindex="0" aria-controls="example1"
rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">No.
Invoice</th><th class="sorting" tabindex="0" aria-controls="example1" rowspan="1"
colspan="1" aria-label="Engine version: activate to sort column ascending">Tanggal
Diterima</th><th class="sorting" tabindex="0" aria-controls="example1" rowspan="1"
colspan="1" aria-label="CSS grade: activate to sort column ascending">Debet
(Rp)</th><th class="sorting" tabindex="0" aria-controls="example1" rowspan="1"
colspan="1" aria-label="CSS grade: activate to sort column ascending">kredit
(Rp)</th><th class="sorting" tabindex="0" aria-controls="example1" rowspan="1"
colspan="1" aria-label="CSS grade: activate to sort column ascending">Saldo
(Rp)</th><th class="sorting" tabindex="0" aria-controls="example1" rowspan="1"
colspan="1" aria-label="CSS grade: activate to sort column ascending">Detail</th>
                </tr>
            </thead>

```

```
|  |
| --- |
| Edit |

```

Tabel 5.7: Kode Program Laporan Keuangan (Lanjutan)

Tabel 5.4 Kode Program Laporan Keuangan (Lanjutan)

```
@push('child-scripts')
<script>
  $(function(){
    $('#example1').DataTable({
      "paging": true,
      "lengthChange": true,
      "searching": true,
      "ordering": true,
      "info": true,
      "autoWidth": false,
      "responsive": true,
    });
  });
</script>
<script>
  function search(){
    var month = $('#month').val();
    var year = $('#year').val();
    $.ajax({
      type: "Get",
      url: "{{ url('balance/search') }}",
      data: "month" +month
    })
  }
</script>
@endpush
```

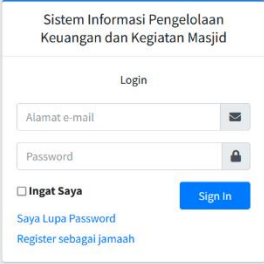
5.1.2. Implementasi Antarmuka Aplikasi Web

Aplikasi *web* yang dibangun dapat diakses oleh *user* menggunakan *web browser*. Berikut dijelaskan tampilan aplikasi *web* pengelolaan kegiatan dan keuangan masjid sesuai dengan implementasi sistem. Untuk penjelasan antarmuka lain dapat dilihat pada lampiran G.

5.1.2.1. Halaman Login

Halaman *login* merupakan halaman *website* yang pertama kali ditampilkan ketika *user* membuka aplikasi *web*. Hal ini dimaksudkan untuk menjaga agar aplikasi ini hanya dapat digunakan oleh *user* yang sudah terdaftar

dalam sistem pengelolaan kegiatan dan keuangan masjid. Halaman *login* dapat dilihat pada Gambar 5.1.



Sistem Informasi Pengelolaan
Keuangan dan Kegiatan Masjid

Login

Alamat e-mail

Password

Ingat Saya

[Sign In](#)

[Saya Lupa Password](#)

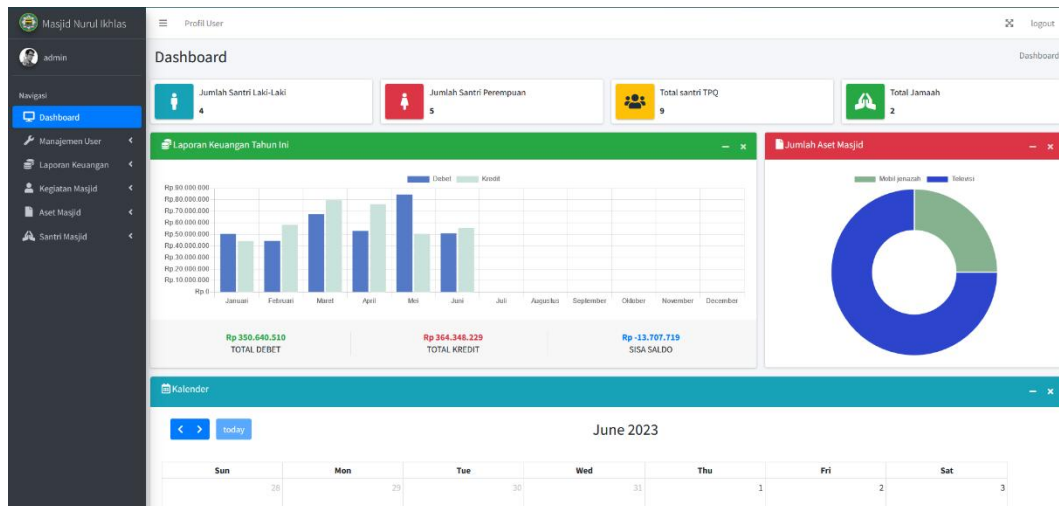
[Register sebagai jamaah](#)

Gambar 5.1 Halaman Login

Berdasarkan Gambar 5.1 *user* terlebih dahulu harus membuat akun sebelum bisa melakukan *login*. Setelah membuat akun *user* dapat masuk kedalam aplikasi dengan men-*input*-kan alamat *email* dan *password* yang telah di *input*-kan sebelumnya lalu menekan tombol “Sign In”. Jika *user* berhasil *login* maka sistem menampilkan halaman utama aplikasi.

5.1.2.2. Halaman *Dashboard*

Halaman *dashboard* adalah halaman utama yang ditampilkan kepada *user* setelah *login*. Tampilan *dashboard* dapat dilihat pada Gambar 5.2.



Gambar 5.2 Halaman Dashboard

Berdasarkan gambar 5.2, user dapat melihat beberapa informasi berikut:

1. Jumlah santri dan jamaah yang dimiliki oleh masjid.
2. Grafis data keuangan masjid.
3. Jadwal kegiatan masjid.
4. Jumlah aset yang dimiliki masjid.

Pada bagian navigasi, *user* dapat mengakses fitur-fitur aplikasi web sesuai dengan *role* yang dimiliki *user*.

5.1.2.3. Halaman Lihat Laporan Keuangan

Pada halaman lihat laporan keuangan, *user* dapat melihat laporan keuangan masjid secara detail. Halaman lihat laporan keuangan dapat dilihat pada Gambar 5.3.

Tgl Input	Deskripsi	Kategori	Tanggal Diterima	Debet	Kredit	Saldo	Opsi
06 September 2023 16:46:51	Honor Inam & Petugas Masjid/ Guru	Pengeluaran Kegiatan	01 Agustus 2023	Rp 5.000.000	0	- Rp 5.000.000	Edit Delete
06 September 2023 16:53:47	Sisa Kas Pembangunan	Pemasukkan Pembangunan	01 Agustus 2023	0	Rp 46.509.937	Rp 46.509.937	Edit Delete
06 September 2023 16:55:15	Sisa Infak Anak Yatim	Infak Anak Yatim	01 Agustus 2023	0	Rp 18.690.363	Rp 18.690.363	Edit Delete
06 September 2023 23:27:42	Kasak Infak	Pemasukkan Pembangunan	06 September 2023	0	Rp 1.511.100	Rp 1.511.100	Edit Delete

Gambar 5.3 Laporan Keuangan

Berdasarkan gambar 5.3, *user* dapat melihat dan mencari data keuangan masjid secara spesifik dengan memilih bulan dan tahun data keuangan. Sistem akan menampilkan total debet, total kredit, sisa saldo, dan data keuangan sesuai dengan bulan dan tahun yang diinputkan.

5.2. Pengujian Sistem

Pada tahapan ini dilakukan pengujian pada semua bagian aplikasi yang telah dibangun. Proses ini dilakukan agar aplikasi yang dibangun berjalan sesuai dengan perancangan sistem. Pengujian ini dilakukan dengan menggunakan metode *black box testing*, yaitu metode pengujian yang berfokus pada pemeriksaan ketersediaan fungsional yang sudah dirancang pada aplikasi. Dengan menggunakan metode ini diketahui apakah fungsional aplikasi sudah berjalan sebagaimana mestinya dan sesuai dengan harapan atau tidak. Kelebihan metode ini dibandingkan metode *white box testing* adalah pada metode *black box testing* tidak diperlukan pengetahuan khusus tentang kode program aplikasi, sedangkan pada metode *white box testing* pengujian diharuskan untuk memeriksa komponen internal kode program aplikasi. Pengujian ini dilakukan oleh penulis.

5.2.1. Fokus Pengujian

Fokus pengujian aplikasi ini menggunakan data uji berdasarkan data yang telah didapatkan dari aplikasi *web*. Pada pengujian ini terdapat 12 item uji. Fokus Pengujian dapat dilihat pada Tabel 5.5 berikut ini

No.	Item Uji	Platform	Pengguna	Detail Pengujian
1.	Mengelola User	<i>Website</i>	Admin	<i>create, edit, delete</i>
2.	Mengelola periode TPQ	<i>Website</i>	Admin	<i>index, create, detail, edit, delete</i>
3.	Mendaftar sebagai Jamaah	<i>Website</i>	Seluruh user	<i>create</i>
4.	Fitur lupa password	<i>Website</i>	Seluruh user	ubah password
5.	Mengelola Laporan Keuangan Masjid	<i>Website</i>	Admin, Bendahara	<i>index, create, edit, delete, search</i>
6.	Mengajukan Kegiatan masjid	<i>Website</i>	Pengurus masjid	<i>index, create, detail, edit, delete</i>
7.	Mengelola Kategori Kegiatan Masjid	<i>Website</i>	Admin, pengurus masjid, bendahara	<i>index, create, edit, delete</i>
8.	Mengajukan Aset masjid	<i>Website</i>	Pengurus masjid	<i>index, create, detail, edit, delete</i>
9.	Mengelola Biodata Santri	<i>Website</i>	Jamaah	<i>index, create, edit, delete</i>
10.	Mengelola Pendaftaran Santri	<i>Website</i>	Jamaah	<i>index, create, detail, edit, delete, search</i>
11.	Mengelola Penerimaan Santri	<i>Website</i>	Admin, bendahara	<i>index, detail, menyetujui, menolak</i>

Tabel 5.5 Fokus Pengujian

5.2.2. Kasus Hasil Pengujian

Bagian ini membahas tentang kasus-kasus serta hasil dari pengujian sistem. Pengujian yang dilakukan berpedoman pada fokus pengujian yang telah ditentukan sebelumnya. Pengujian ini dilakukan berdasarkan fungsional dengan memperhatikan masukan ke sistem dan keluaran yang dihasilkan sistem. Pada bagian ini dijelaskan tiga kasus pengujian, yaitu pendaftaran santri, menambahkan laporan keuangan, dan mengajukan kegiatan masjid

5.2.2.1. Pengujian Pendaftaran Santri

Pada pengujian ini dilakukan aktifitas pengujian pendaftaran santri. Hasil pengujian pendaftaran santri dapat dilihat pada Tabel 5.6 dan Tabel 5.7.

Kasus dan Hasil Uji (Benar)	
Data Masukan	Seluruh data yang dibutuhkan dalam halaman pendaftaran santri.
Yang Diharapkan	Sistem menyimpan dan menampilkan data santri yang telah diinputkan.
Pengamatan	Sistem berhasil menyimpan data santri pada database dan menampilkan notifikasi “Data Berhasil disimpan”
Hasil	Sesuai.

Tabel 5.6 Kasus dan Hasil Uji Pendaftaran Santri (Benar)

Kasus dan Hasil Uji (Alternatif)	
Data Masukan	User menekan tombol “submit” sebelum mengisi formulir.
Yang Diharapkan	Sistem menampilkan notifikasi kesalahan.
Pengamatan	User tidak bisa mendaftarkan santri.
Hasil	Sesuai.

Tabel 5.7 Kasus dan Hasil Uji Pendaftaran Santri (Alternatif)

A. Mendaftarkan Santri (Benar)

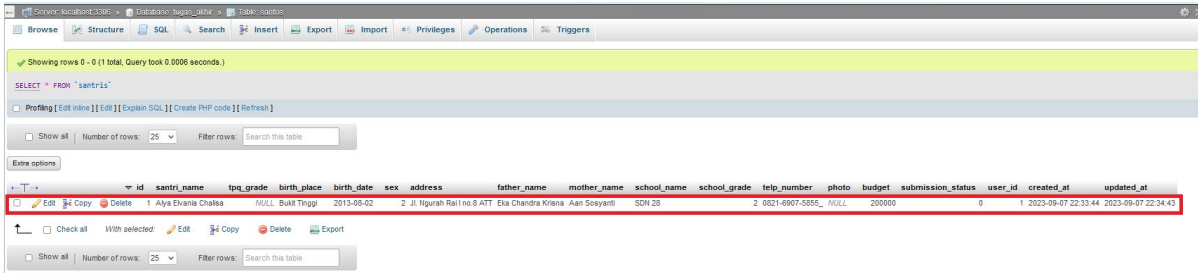
Pengujian aplikasi dilakukan dengan menginput data untuk menambahkan santri pada halaman mendaftarkan santri dan menekan tombol

“submit”. Jika berhasil maka sistem menampilkan halaman lihat daftar santri. Hasil pengujian pengajuan santri (benar) dapat dilihat pada Gambar 5.4, Gambar 5.5, dan Gambar 5.6.

Gambar 5.4 Mendaftarkan Santri (Benar)

Nama Santri	Sekolah	Kelas	Jenis Kelamin	Biaya Pendaftaran (Rp)	Status	Opsi
Alya Elvania Chalisa	SDN 28	2	Perempuan	Rp 200.000	Sedang diproses	Detail Edit Delete

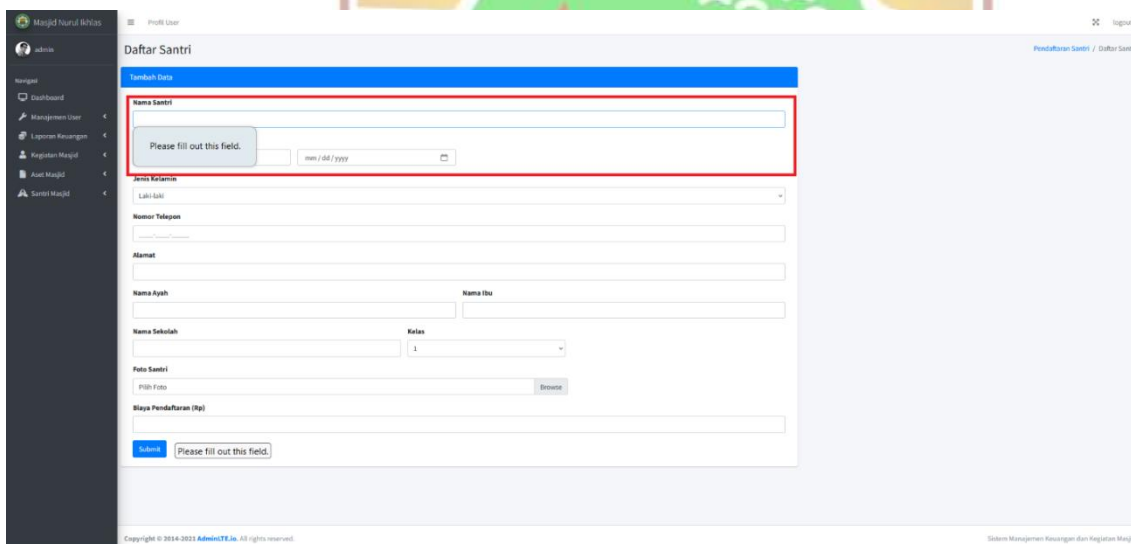
Gambar 5.5 Mendaftarkan Santri (Benar)



Gambar 5.6 Mendaftarkan Santri (Benar)

B. Mendaftarkan Santri (Alternatif)

Pengujian aplikasi dilakukan dengan menekan tombol “*submit*” sebelum mengisi *form*, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian pengajuan mendaftar santri (alternatif) dapat dilihat pada Gambar 5.6.



Gambar 5.7 Menambahkan Santri (Alternatif)

5.2.2.2. Pengujian Menambahkan Laporan Keuangan

Pada pengujian ini dilakukan aktifitas pengujian penambahan laporan keuangan. Hasil pengujian menambahkan laporan keuangan dapat dilihat pada Tabel 5.8 dan Tabel 5.9.

Kasus dan Hasil Uji (Benar)	
Data Masukan	Seluruh data yang dibutuhkan dalam halaman menambahkan laporan keuangan.
Yang Diharapkan	Sistem menyimpan dan menampilkan data keuangan yang telah diinputkan.
Pengamatan	Sistem berhasil menyimpan data santri pada database dan menampilkan notifikasi “Data Berhasil disimpan”
Hasil	Sesuai.

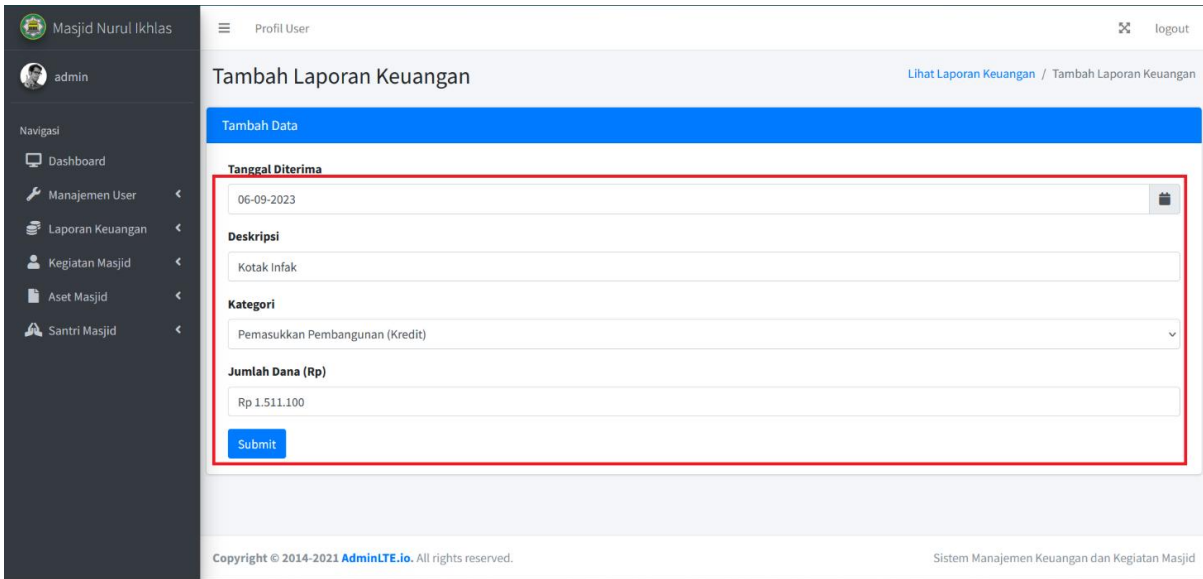
Tabel 5.8 Kasus dan Hasil Uji Menambahkan Laporan Keuangan (Benar)

Kasus dan Hasil Uji (Alternatif)	
Data Masukan	User menekan tombol “submit” sebelum mengisi formulir.
Yang Diharapkan	Sistem menampilkan notifikasi kesalahan.
Pengamatan	User tidak bisa mendaftarkan santri.
Hasil	Sesuai.

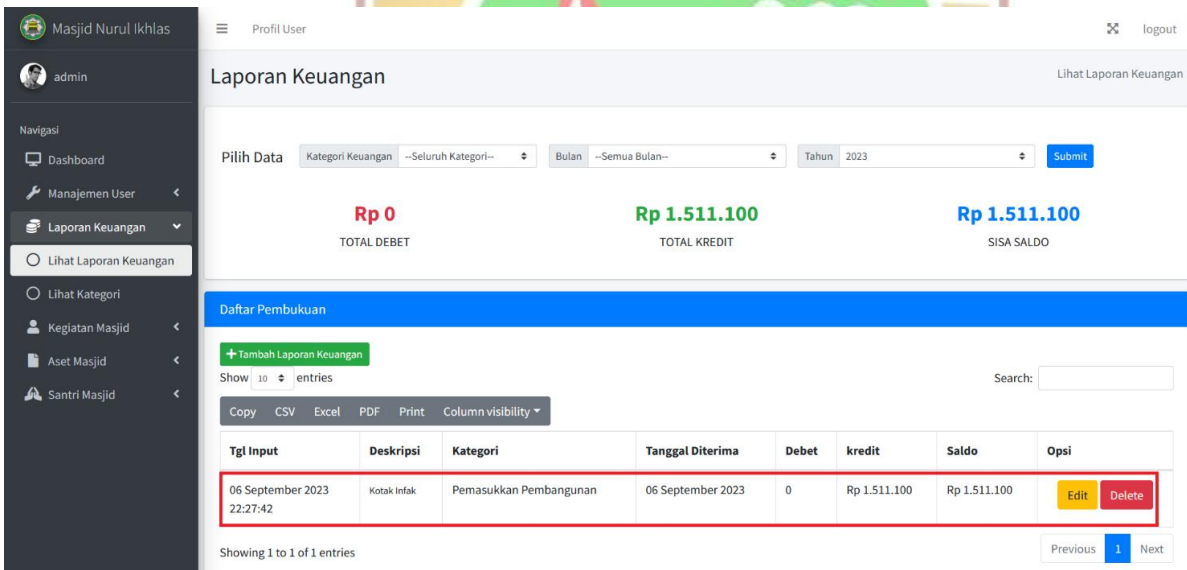
Tabel 5.9 Kasus dan Hasil Uji Menambahkan Laporan Keuangan (Alternatif)

A. Menambahkan Laporan Keuangan (Benar)

Pengujian aplikasi dilakukan dengan menginput data untuk menambahkan laporan keuangan pada halaman menambahkan laporan keuangan dan menekan tombol “*submit*”. Jika berhasil maka sistem menampilkan halaman laporan keuangan. Hasil pengujian menambahkan laporan keuangan(benar) dapat dilihat pada Gambar 5.8, Gambar 5.9 dan Gambar 5.10.



Gambar 5.8 Menambahkan Laporan Keuangan (Benar)



Gambar 5.9 Menambahkan Laporan Keuangan (Benar)

Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

SELECT * FROM `balances`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	description	no_invoice	date_received	total_amount	user_id	balance_category_id	created_at	updated_at
<input type="checkbox"/>	1	Honor Imam & Petugas Masjid/ Guru	NULL	2023-08-01 00:00:00	5000000	1	4	2023-09-06 16:46:51	2023-09-06 16:54:30
<input type="checkbox"/>	2	Sisa Kas Pembangunan	NULL	2023-08-01 00:00:00	46509937	1	5	2023-09-06 16:53:47	2023-09-06 16:55:32
<input type="checkbox"/>	3	Sisa Infak Anak Yatim	NULL	2023-08-01 00:00:00	18690363	1	1	2023-09-06 16:55:15	2023-09-06 16:55:15
<input type="checkbox"/>	4	Kotak Infak	NULL	2023-09-06 00:00:00	1511100	1	5	2023-09-06 22:27:42	2023-09-06 22:27:42

Check all | With selected: Edit Copy Delete Export

Gambar 5.10 Menambahkan Laporan Keuangan (Benar)

B. Menambahkan Laporan Keuangan (Alternatif)

Pengujian aplikasi dilakukan dengan menekan tombol “*submit*” sebelum mengisi *form*, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian pengajuan menambahkan laporan keuangan (alternatif) dapat dilihat pada Gambar 5.11.

AdminLTE 3 | Profil User | Laporan Keuangan

Tambah Data

Tanggal Diterima: 04 / 18 / 2023

Deskripsi: Please fill out this field.

Kategori: Donasi

Debet/Kredit: Debet

Jumlah Dana (Rp): 500000

Submit

Copyright © 2014-2021 AdminLTE.io. All rights reserved. | Sistem Manajemen Keuangan dan Kegiatan Masjid

Gambar 5.11 Menambahkan Laporan Keuangan (Alternatif)

5.2.2.3. Pengujian Pengajuan Kegiatan Masjid

Pada pengujian ini dilakukan aktifitas pengujian pengajuan kegiatan masjid. Hasil pengujian pengajuan kegiatan masjid dapat dilihat pada Tabel 5.10 dan Tabel 5.11.

Tabel 5.10 Kasus dan Hasil Uji Pengajuan kegiatan Masjid (Benar)

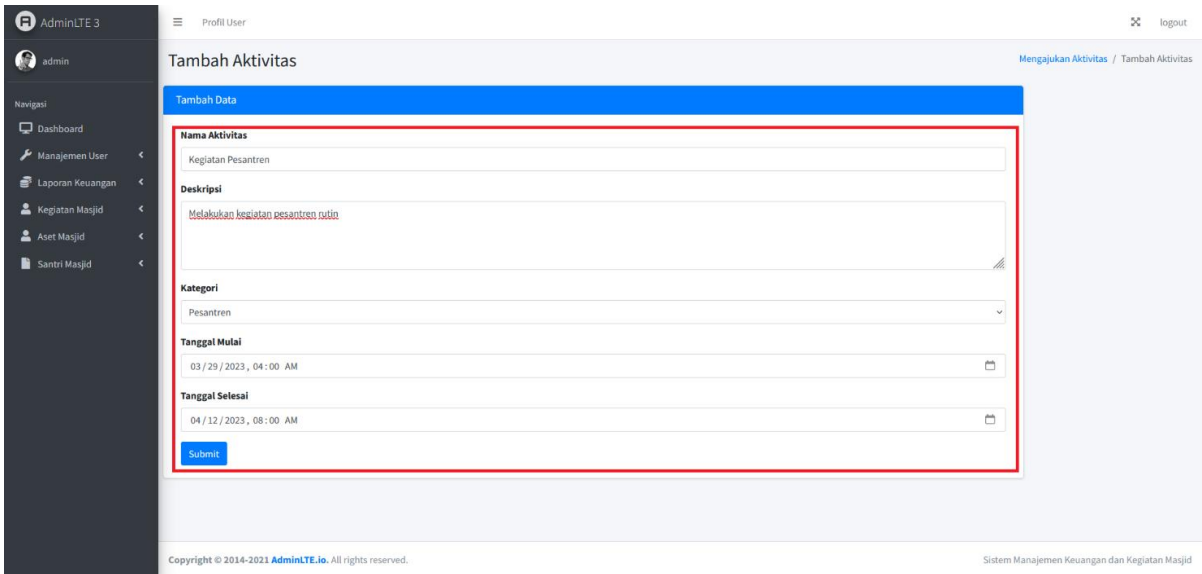
Kasus dan Hasil Uji (Benar)	
Data Masukan	Seluruh data yang dibutuhkan dalam halaman pengajuan kegiatan masjid.
Yang Diharapkan	Sistem menyimpan dan menampilkan data kegiatan yang telah diinputkan.
Pengamatan	Sistem berhasil menyimpan data santri pada database dan menampilkan notifikasi “Data Berhasil disimpan”
Hasil	Sesuai.

Tabel 5.11 Kasus dan Hasil Uji Pengajuan kegiatan Masjid (Alternatif)

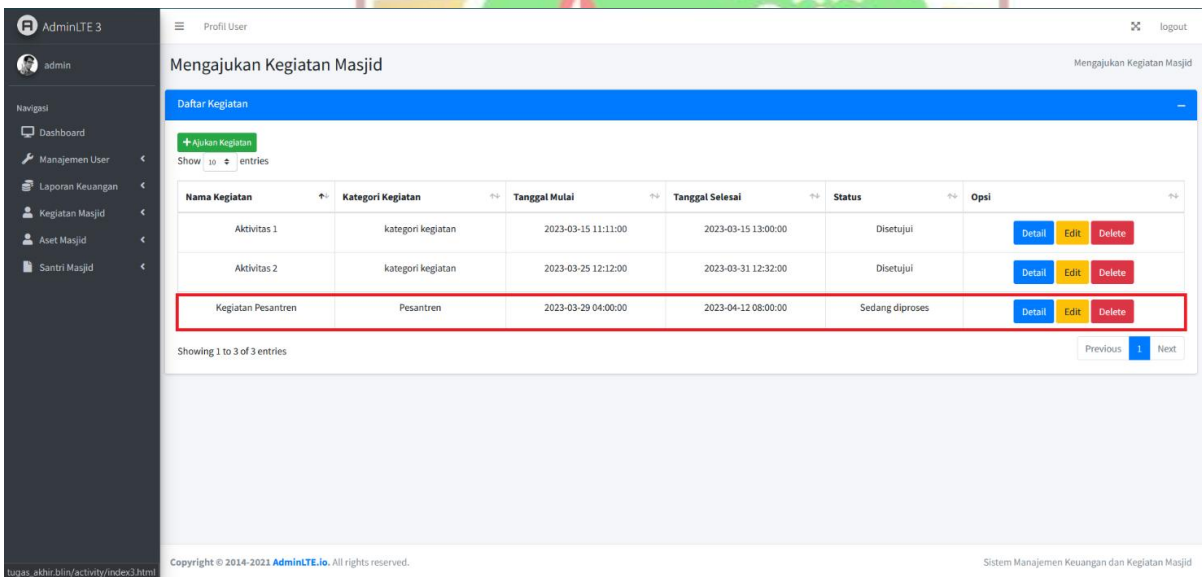
Kasus dan Hasil Uji (Alternatif)	
Data Masukan	User menekan tombol “submit” sebelum mengisi formulir.
Yang Diharapkan	Sistem menampilkan notifikasi kesalahan.
Pengamatan	User tidak bisa mendaftarkan santri.
Hasil	Sesuai.

A. Pengajuan Kegiatan Masjid (Benar)

Pengujian aplikasi dilakukan dengan menginput data untuk pengajuan kegiatan pada halaman mengajukan kegiatan masjid dan menekan tombol “submit”. Jika berhasil maka sistem menampilkan halaman laporan keuangan. Hasil pengujian pengajuan kegiatan masjid (benar) dapat dilihat pada Gambar 5.12, Gambar 5.13 dan Gambar 5.14.



Gambar 5.12 Pengajuan Kegiatan Masjid (Benar)



Gambar 5.13 Pengajuan Kegiatan Masjid (Benar)

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM `activities`
```

Number of rows: 25 | Filter rows: Search this table

id	activity_name	description	schedule_start	schedule_end	status	submission_status	user_id	activity_categories_id	created_at	updated_at
1	Kegiatan Pesantren	Melakukan Kegiatan pesantren rutin	2023-03-29 04:00:00	2023-04-12 08:00:00	2	0	1	1	2023-05-07 21:14:19	2023-05-07 21:14:19

Gambar 5.14 Pengajuan Kegiatan Masjid (Benar)

B. Pengajuan Kegiatan Masjid (Alternatif)

Pengujian aplikasi dilakukan dengan menekan tombol “*submit*” sebelum mengisi *form*, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian pengajuan mendaftarkan kegiatan masjid (alternatif) dapat dilihat pada Gambar 5.15.

AdminLTE 3 | Profil User | logout

Tambah Aktivitas

Mengajukan Aktivitas / Tambah Aktivitas

Tambah Data

Nama Aktivitas

Please fill out this field.

Kategori

kategori kegiatan

Tanggal Mulai

mm / dd / yyyy, --:-- --

Tanggal Selesai

mm / dd / yyyy, --:-- --

Submit

Copyright © 2014-2021 AdminLTE.io. All rights reserved. | Sistem Manajemen Keuangan dan Kegiatan Masjid

Gambar 5.15 Pengajuan Kegiatan Masjid (Alternatif)

5.3. Kesimpulan Hasil Pengujian

Pengujian yang dilakukan terhadap aplikasi yang dibangun berfokus pada kesesuaian fungsional sistem yang diuji secara manual. Berdasarkan pengujian yang telah dilakukan diperoleh sebuah hasil antara perancangan dengan keluaran sistem, selain itu juga tidak ditemukan kegagalan dalam setiap proses pada masing-masing fungsional. Dapat disimpulkan bahwa aplikasi web untuk sistem informasi pengelolaan kegiatan dan keuangan Masjid Nurul Ikhlas Kecamatan Padang Utara telah berjalan sesuai dengan fungsional yang dirancang. Hasil pengujian secara lengkap dijelaskan pada lampiran H. Kesimpulan dari hasil pengujian sistem dapat dilihat pada Tabel 5.12

Tabel 5.12 Hasil Pengujian

No	Fungsional	Platform	Pengguna	Skenario Pengujian		Hasil yang Diharapkan		Hasil
				Benar	Alternatif	Benar	Alternatif	
1.	Mengelola User	Website	Admin	Admin membuka menu kelola user, admin dapat menjalankan fungsi tambah, detail, edit, dan hapus	Admin membuka menu kelola user, admin tidak bisa menambah data jika form tambah user belum diisi, admin tidak dapat men-uncheck seluruh role yang dimiliki user.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data, sistem menampilkan notifikasi kesalahan pada saat perubahakan dilakukan.	Sesuai

Tabel 5.12 Hasil Pengujian (Lanjutan)

2.	Mengelola Periode TPQ	Website	Admin	Admin membuka menu kelola periode, <i>admin</i> dapat menjalankan fungsi tambah, <i>detail</i> , <i>edit</i> , dan hapus	Admin membuka menu kelola periode, <i>admin</i> tidak bisa menambah data jika <i>form</i> tambah periode belum diisi.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data, sistem menampilkan notifikasi kesalahan pada saat perubahan dilakukan.	Sesuai
3.	Mendaftar sebagai Jamaah	Website	Seluruh user	User menekan <i>link</i> "Register Sebagai Jamaah", user dapat mendaftar dengan mengisi <i>form</i> pendaftaran	User tidak bisa menambah data jika <i>form</i> laporan keuangan belum diisi.	Sistem menyimpan data dan terdaftar sebagai jamaah dalam sistem.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data.	Sesuai
4.	Fitur lupa password	Website	Seluruh user	User dapat mengubah <i>password</i> pada halaman <i>login</i> dengan menekan <i>link</i> "Saya Lupa Password"	User tidak dapat mengubah <i>password</i> jika data yang dimasukkan pada <i>form</i> "password" tidak sama dengan <i>form</i> "confirm password"	Sistem mengubah <i>password</i> dan sistem kembali ke halaman <i>login</i> .	Sistem menampilkan notifikasi kesalahan pada saat men- <i>submit</i> data.	Sesuai

Tabel 5.12 Hasil Pengujian (Lanjutan)

5.	Mengelola Laporan Keuangan Masjid	Website	Admin, Pengurus masjid	User membuka menu laporan keuangan, user dapat menjalankan fungsi tambah, detail, edit, dan hapus.	user membuka menu data laporan keuangan, user tidak bisa menambah data jika form laporan keuangan belum diisi.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data keuangan.	Sesuai
6.	Mengajukan Kegiatan masjid	Website	Pengurus masjid	Pengurus masjid membuka menu mengajukan kegiatan masjid, pengurus masjid dapat menjalankan fungsi tambah, detail, edit, dan hapus.	Pengurus masjid membuka menu mengajukan kegiatan masjid, user tidak bisa menambah data jika form mengajukan kegiatan masjid belum diisi.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data kategori aset masjid.	Sesuai
7.	Mengelola Kategori Kegiatan Masjid	Website	Admin, Pengurus masjid	User membuka menu kategori kegiatan masjid, user dapat menjalankan fungsi tambah, detail, edit, dan hapus.	User membuka menu kategori kegiatan masjid, user tidak bisa menambah data jika form kategori kegiatan	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data kategori kegiatan masjid.	Sesuai

					masjid belum diisi.			
--	--	--	--	--	---------------------	--	--	--

Tabel 5.12 Hasil Pengujian (Lanjutan)

8.	Mengelola Kategori Kegiatan Masjid	Website	Admin, Pengurus masjid	User membuka menu kategori kegiatan masjid, user dapat menjalankan fungsi tambah, detail, edit, dan hapus.	User membuka menu kategori kegiatan masjid, user tidak bisa menambah data jika form kategori kegiatan masjid belum diisi.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data kategori kegiatan masjid.	Sesuai
9.	Mengajukan Aset Masjid	Website	Pengurus masjid	Pengurus masjid membuka menu mengajukan aset masjid, user dapat menjalankan fungsi tambah, detail, edit, dan hapus.	Pengurus masjid membuka menu mengajukan aset masjid, user tidak bisa menambah data jika form kategori aset masjid belum diisi.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data kategori kegiatan masjid.	Sesuai
10.	Mengelola Biodata Santri	Website	Jamaah	Jamaah membuka menu biodata santri, jamaah dapat menjalankan fungsi tambah, detail, edit, dan hapus	Jamaah membuka menu data biodata santri, jamaah tidak bisa menambah data jika form pendaftaran santri belum diisi.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data biodata santri.	Sesuai

Tabel 5.12 Hasil Pengujian (Lanjutan)

11.	Mengelola Pendaftaran Santri	Website	Jamaah	Jamaah membuka menu pendaftaran santri, jamaah dapat menjalankan fungsi tambah, <i>detail, edit, dan hapus.</i>	Jamaah membuka menu data pendaftaran santri, jamaah tidak bisa menambah data jika <i>form</i> pendaftaran santri belum diisi.	Sistem menampilkan data sesuai dengan fungsional yang dilakukan.	Sistem menampilkan notifikasi kesalahan pada saat menambahkan data santri.	Sesuai
-----	------------------------------	---------	--------	---	---	--	--	--------



BAB VI

KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang kesimpulan dan saran pada laporan tugas akhir ini. Kesimpulan merupakan pencapaian dari tujuan yang telah ditetapkan pada penelitian, sedangkan saran merupakan harapan yang diinginkan kedepannya terkait penelitian yang telah dilakukan.

6.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa:

1. Pembangunan sistem informasi Masjid Nurul Ikhlas Kecamatan Padang Utara berbasis web telah berhasil dibangun dengan menggunakan metode *waterfall* yang terdiri dari tahapan analisis kebutuhan fungsional aplikasi, *design* yang dilakukan terdiri dari perancangan ERD (*Entity Relational Diagram*), arsitektur perangkat lunak, perancangan antarmuka (*user interface*), pengkodean dengan menggunakan bahasa pemrograman *javascript* dan *php* dengan *laravel* sebagai *framework* aplikasi, dan *testing* dilakukan dengan menggunakan metode *black box testing*.
2. Berdasarkan sistem informasi yang telah dibangun, sistem dapat mempermudah proses pelaporan keuangan masjid, pencatatan kegiatan masjid, pencatatan aset masjid, dan pendaftaran santri masjid.
3. Dengan adanya sistem informasi ini, seluruh pengguna aplikasi dapat melihat dan mengetahui informasi umum masjid dari halaman *dashboard* aplikasi.
4. Dengan adanya sistem informasi ini, pengelola dan pengurus masjid Nurul Ikhlas dapat mengelola dan memantau kegiatan dan keuangan masjid secara efektif karena seluruh pengelolaan dapat dilakukan di dalam satu sistem.

6.2. Saran

Aplikasi sistem informasi pengelolaan kegiatan dan keuangan Masjid Nurul Ikhlas Kecamatan Padang Utara berbasis web ini masih membutuhkan pengembangan lebih lanjut yang sejalan dengan kebutuhan fungsional yang telah dibangun. Pada saat ini, sistem masih dihubungkan dengan *link* pada *website*

profile masjid. Diharapkan kedepannya adanya pengembangan lebih lanjut dengan *website profile* masjid dan sistem informasi yang telah dibuat digabungkan menjadi satu aplikasi. Diharapkan juga adanya pengembangan aplikasi ini dikembangkan menjadi aplikasi *multiplatform* sehingga aplikasi dapat menjangkau semua sistem operasi yang digunakan *user*.



DAFTAR PUSTAKA

- Rifa'i, Ahmad. 2016. "Revitalisasi Fungsi Masjid dalam Kehidupan Masyarakat Modern". Kediri: Jurnal Universum.
- Wardana, T. Indra. Eko Aribowo. 2013. "Perancangan dan Implementasi Sistem Informasi Manajemen Kegiatan Masjid, Studi Kasus : Masjid Jogokarian Yogyakarta". Yogyakarta: Universitas Ahmad Dahlan.
- Ramdhani, Eka Chandra. Muhamad Ilham Destama Putra. Asep. Fransisca Natalia. 2021. "SIPAS (Sistem Informasi Pengelolaan Dana Masjid)". Jakarta: Universitas Bina Sarana Informatika
- Kristanto, Dodyk. 2018. "Sistem Informasi Pengelolaan Masjid Ibaadurrahman Berbasis Web". Surakarta: Universitas Muhammadiyah Surakarta.
- Syafe'I, Makhmud. 2016. "Masjid Dalam Perspektif Sejarah dan Hukum Islam". Bandung: Universitas Pendidikan Indonesia.
- Amalia, Y. S. Endang. 2017. "Kependudukan Sebagai Pengembangan Egovernment," Informasi, Sist. Tek. Fak. Widyatama, Univ., vol. 2, no. 1, pp. 1-5.
- Hermawan, R. Hidayat, A. dan Gayuh Utomo, V. 2016. "Sistem Informasi Penjadwalan Kegiatan Belajar Mengajar Berbasis Web (Studi Kasus : Yayasan Ganesha Operation Semarang)." 4: 14-16.
- Anggraeni, Elisabet Yunaeti & Rita Irviani. 2017. *Pengantar Sistem Informasi*. Yogyakarta: Andi. 1:1-2.
- Yudhasmara, A. 2015. "Fungsi Masjid Sebagai Tempat Ibadah dan Fungsi Sosial Lainnya". Dikutip dari <http://islamislam.com/2015/11/27/masjid-tempat-ibadah-umat-muslim/> .
- Keputusan Dirjen Bimbingan Masyarakat Islam. No. DJ.11/802 tahun 2014 tentang "Standar Pembinaan Manajemen Masjid".
- Jarel, Remick. 2010. "Meet Jarel Remick: The Web AppStorm Editor". Appstorm (<http://web-appstorm.net/general/interviews/meet-jarel-remick-the-web-appstorm-editor/>).

- Rouse, Margaret. 2011. Web Application (Web App). TechTarget Magazine (<http://searchsoftwarequality.techtarget.com/definition/Web-applicationWeb-app>).
- Yuniar, Supardi & Sulaeman. 2019. *Semua Bisa Menjadi Programmer Laravel Basic*. Jakarta: Penerbit PT Elex Media Komputindo.
- Arifudzaki, B., Somantri, M., & FR, A. 2010. “Aplikasi Sistem Informasi Persediaan Barang pada Perusahaan Ekspor Hasil Laut Berbasis WEB“. Semarang: Universitas Diponegoro.
- Aini, F. N. 2013. “Pemodelan Arsitektur Enterprise Menggunakan TOGAF ADM untuk mendukung Sistem Informasi Promosi pada Perguruan Tinggi (Studi Kasus Universitas Respati Yogyakarta)”. Yogyakarta: Universitas Atmajaya Yogyakarta.
- Dewi, L. P., Indahyanti, U., & S, Y. H. 2012. “Pemodelan Proses Bisnis Menggunakan Activity Diagram UML DAN BPMN (Studi Kasus FRS ONLINE)”. Surabaya: Petra Christian University.
- Dicoding. 2021. Apa itu UML? Beserta Pengertian dan Contohnya. (<https://www.dicoding.com/blog/apa-itu-uml/>). Diakses pada tanggal 08 September 2023
- Fauzi, I. 2011. “Penggunaan Algoritma Dijkstra dalam Pencairan Rute Tercepat dan Rute Terpendek.” Studi Kasus pada Jalan Raya antara Wilayah Blok M dan Kota.
- Pressman, R. S. & Brian R. Maxim. 2020. *Software Engineering: A Practitioner's Approach ninth edition*. New York: Mc-Graw Hill.
- Somerville, Ian. 2011. *Software Engineering: ninth edition*. Boston: Pearson.
- S. ,Rosa A. . M. Shalahuddin. 2018. “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek ”. Bandung : Informatika Bandung.
- Sabirin, Abdul & Sulfia (2018). “Sistem Informasi Jasa Pemesanan Kamar Kost Online Di Kota Baubau”. Baubau : Universitas Dayanu Ikhsanuddin Baubau.
- Naista, D. (2017). *Codeigniter Vs Laravel*. Yogyakarta: CV. Lokomedia.

Setiawan, Rony. 2021. “Apa itu framework? Developer Wajib Tahu”. Dicoding (<https://www.dicoding.com/blog/apa-itu-framework/>). Diakses pada tanggal 08 September 2023

Khoa, Leo (2019). “Laragon Documentation”. Laragon (<https://laragon.org/docs/>)





LAMPIRAN A

User case scenario aplikasi

1. *User case scenario* menambah laporan keuangan

<i>Use Case</i>	Pencatatan keuangan (tambah)
<i>Actor</i>	Bendahara Admin
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<p>12. <i>Actor</i> membuka halaman “laporan keuangan masjid”.</p> <p>13. Sistem akan menampilkan halaman laporan keuangan masjid dan menampilkan data keuangan masjid.</p> <p>14. <i>Actor</i> mengklik tombol “edit”</p> <p>15. Sistem akan menampilkan halaman tambah keuangan masjid dan menampilkan <i>form</i> tambah keuangan masjid.</p> <p>16. <i>Actor</i> meng-<i>input</i> form keuangan masjid.</p> <p>17. <i>Actor</i> mengklik tombol “submit” saat data benar.</p> <p>18. Sistem akan menyimpan data keuangan masjid dan menampilkan halaman laporan keuangan masjid serta menampilkan data keuangan masjid.</p>
<i>Scenario Alternative</i>	<p>19. <i>Actor</i> menekan tombol “Submit” saat inputan data salah atau kosong.</p> <p>20. Sistem menampilkan notifikasi kesalahan.</p> <p>21. <i>Actor</i> menekan tombol “Submit” saat kategori keuangan kosong.</p> <p>22. Sistem menampilkan notifikasi kesalahan.</p>
<i>Exit Scenario</i>	<i>Actor</i> berhasil menambahkan data keuangan masjid.

2. *Use case scenario* mengubah laporan keuangan

<i>Use Case</i>	Pencatatan keuangan (ubah)
<i>Actor</i>	Bendahara Admin
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “laporan keuangan masjid”. 2. Sistem akan menampilkan halaman laporan keuangan masjid dan menampilkan data keuangan masjid. 3. <i>Actor</i> mengklik tombol “edit” 4. Sistem akan menampilkan halaman ubah keuangan masjid dan menampilkan <i>form</i> ubah keuangan masjid. 5. <i>Actor</i> meng-<i>input</i> form keuangan masjid. 6. <i>Actor</i> mengklik tombol “submit” saat data benar. 7. Sistem akan mengubah data keuangan masjid dan menampilkan halaman laporan keuangan masjid serta menampilkan data keuangan masjid.
<i>Scenario Alternative</i>	<ol style="list-style-type: none"> 8. <i>Actor</i> menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan. 10. <i>Actor</i> menekan tombol “Submit” saat kategori keuangan kosong. 11. Sistem menampilkan notifikasi kesalahan.
<i>Exit Scenario</i>	<i>Actor</i> berhasil mengubah data keuangan masjid.

3. Use case scenario hapus laporan keuangan

<i>Use Case</i>	Pencatatan keuangan (hapus)
<i>Actor</i>	Bendahara Admin
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “laporan keuangan masjid”. 2. Sistem akan menampilkan halaman laporan keuangan masjid dan menampilkan data keuangan masjid. 3. <i>Actor</i> mengklik tombol “hapus” 4. Sistem akan menghapus data keuangan masjid dan menampilkan halaman laporan keuangan masjid serta menampilkan data keuangan masjid.
<i>Exit Scenario</i>	<i>Actor</i> berhasil mengubah data keuangan masjid.

4. Use case scenario tambah kegiatan masjid

<i>Use Case</i>	Pengajuan kegiatan (tambah)
<i>Actor</i>	Pengurus masjid
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 10. <i>Actor</i> membuka halaman “mengajukan kegiatan”. 11. Sistem akan menampilkan halaman pengajuan kegiatan dan menampilkan data pengajuan kegiatan. 12. <i>Actor</i> mengklik tombol “tambah” 13. Sistem akan menampilkan halaman tambah pengajuan kegiatan dan <i>form</i> tambah pengajuan kegiatan. 14. <i>Actor</i> meng-<i>input</i> data pengajuan kegiatan. 15. <i>Actor</i> mengklik tombol “submit” saat data benar. 16. Sistem akan menyimpan data kegiatan dengan menambahkan status pada data sebagai “diajukan”. Lalu Sistem akan kembali ke halaman pengajuan kegiatan dan menampilkan data pengajuan

	kegiatan.
Scenario Alternative	17. Actor menekan tombol “Submit” saat inputan data salah atau kosong. 18. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	Actor berhasil menambahkan pengajuan kegiatan.

5. Use case scenario ubah kegiatan masjid

Use Case	Pengajuan kegiatan (ubah)
Actor	Pengurus masjid
Entry Condition	Actor telah login
Flow of Event	<ol style="list-style-type: none"> 1. Actor membuka halaman “mengajukan kegiatan”. 2. Sistem akan menampilkan halaman pengajuan kegiatan dan menampilkan data pengajuan kegiatan. 3. Actor mengklik tombol “edit” 4. Sistem akan menampilkan halaman ubah pengajuan kegiatan dan form ubah pengajuan kegiatan. 5. Actor meng-input data pengajuan kegiatan. 6. Actor mengklik tombol “submit” saat data benar. 7. Sistem akan mengubah data kegiatan. Lalu Sistem akan kembali ke halaman pengajuan kegiatan dan menampilkan data pengajuan kegiatan.
Scenario Alternative	<ol style="list-style-type: none"> 8. Actor menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	Actor berhasil mengubah pengajuan kegiatan.

6. Use case scenario membatalkan kegiatan masjid

<i>Use Case</i>	Pencatatan kegiatan (membatalkan)
<i>Actor</i>	Pengurus masjid Admin
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “mengajukan kegiatan”. 2. Sistem akan menampilkan halaman mengajukan kegiatan masjid dan menampilkan data kegiatan masjid. 3. <i>Actor</i> mengklik tombol “batalkan” 4. Sistem akan menghapus data kegiatan masjid dan menampilkan halaman mengajukan kegiatan serta menampilkan data kegiatan masjid. 5. Sistem akan mengubah status kegiatan menjadi “dibatalkan” dan sistem menampilkan halaman mengajukan kegiatan serta menampilkan data kegiatan masjid.
<i>Exit Scenario</i>	<i>Actor</i> berhasil menghapus data kegiatan masjid.

7. Use case scenario tambah aset masjid

<i>Use Case</i>	Pengajuan aset (tambah)
<i>Actor</i>	Pengurus masjid
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “mengajukan aset”. 2. Sistem akan menampilkan halaman pengajuan aset dan menampilkan data pengajuan aset. 3. <i>Actor</i> mengklik tombol “tambah” 4. Sistem akan menampilkan halaman tambah pengajuan aset dan <i>form</i> tambah pengajuan aset. 5. <i>Actor</i> meng-<i>input</i> data pengajuan aset. 6. <i>Actor</i> mengklik tombol “submit” saat data benar. 7. Sistem akan menyimpan data aset dengan menambahkan status pada

	data sebagai “diajukan”. Lalu Sistem akan kembali ke halaman pengajuan aset dan menampilkan data pengajuan aset.
Scenario Alternative	8. Actor menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	Actor berhasil menambahkan pengajuan aset.

8. Use case scenario ubah aset masjid

Use Case	Pengajuan aset (ubah)
Actor	Pengurus masjid
Entry Condition	Actor telah login
Flow of Event	<ol style="list-style-type: none"> 1. Actor membuka halaman “mengajukan aset”. 2. Sistem akan menampilkan halaman pengajuan aset dan menampilkan data pengajuan aset. 3. Actor mengklik tombol “edit” 4. Sistem akan menampilkan halaman ubah pengajuan aset dan form ubah pengajuan aset. 5. Actor meng-input data pengajuan aset. 6. Actor mengklik tombol “submit” saat data benar. 7. Sistem akan mengubah data aset. Lalu Sistem akan kembali ke halaman pengajuan aset dan menampilkan data pengajuan aset.
Scenario Alternative	8. Actor menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	Actor berhasil mengubah pengajuan kegiatan.

9. Use case scenario hapus aset masjid

<i>Use Case</i>	Pengajuan aset (hapus)
<i>Actor</i>	Pengurus masjid
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “mengajukan aset”. 2. Sistem akan menampilkan halaman mengajukan aset masjid dan menampilkan data aset masjid. 3. <i>Actor</i> mengklik tombol “hapus” 4. Sistem akan menghapus data aset masjid dan menampilkan halaman mengajukan aset serta menampilkan data aset masjid.
<i>Exit Scenario</i>	<i>Actor</i> berhasil menghapus data aset masjid.

10. Use case scenario tambah detail aset masjid

<i>Use Case</i>	Mengelola detail aset (tambah)
<i>Actor</i>	Pengurus masjid
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “mengajukan aset”. 2. Sistem akan menampilkan halaman pengajuan aset dan menampilkan data pengajuan aset. 3. <i>Actor</i> mengklik tombol “detail” 4. Sistem akan menampilkan halaman detail aset dan <i>form</i> tambah detail aset. 5. <i>Actor</i> meng-<i>input</i> data detail aset. 6. <i>Actor</i> mengklik tombol “submit” saat data benar. 7. Sistem akan menyimpan data aset dan sistem akan kembali ke halaman detail aset.
<i>Exit Scenario</i>	<i>Actor</i> berhasil menambahkan pengajuan aset.

11. Use case scenario ubah detail aset masjid

Use Case	Mengelola detail aset (ubah)
Actor	Pengurus masjid
Entry Condition	<i>Actor</i> telah <i>login</i>
Flow of Event	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “mengajukan aset”. 2. Sistem akan menampilkan halaman pengajuan aset dan menampilkan data pengajuan aset. 3. <i>Actor</i> mengklik tombol “detail” 4. Sistem akan menampilkan halaman detail aset dan <i>form</i> tambah detail aset. 5. <i>Actor</i> mengklik tombol “edit” 6. Sistem akan menampilkan halaman ubah detail aset dan <i>form</i> ubah detail aset. 7. <i>Actor</i> mengubah data detail aset. 8. <i>Actor</i> mengklik tombol “submit” saat data benar. 9. Sistem akan mengubah data aset. Lalu Sistem akan kembali ke halaman pengajuan aset dan menampilkan data pengajuan aset.
Exit Scenario	<i>Actor</i> berhasil mengubah data detail aset.

12. Use case scenario hapus detail aset masjid

Use Case	Mengelola detail aset (hapus)
Actor	Pengurus masjid Admin
Entry Condition	<i>Actor</i> telah <i>login</i>
Flow of Event	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “mengajukan aset”. 2. Sistem akan menampilkan halaman mengajukan aset masjid dan menampilkan data aset masjid. 3. <i>Actor</i> mengklik tombol “detail” 4. Sistem akan menampilkan halaman detail aset dan <i>form</i> tambah detail aset. 5. <i>Actor</i> mengklik tombol “hapus”

	6. Sistem akan menghapus data aset masjid dan menampilkan halaman mengajukan aset serta menampilkan data aset masjid.
<i>Exit Scenario</i>	Actor berhasil menghapus data aset masjid.

13. Use case scenario menambahkan biodata santri masjid

Use Case	Biodata santri (tambah)
Actor	Jamaah
Entry Condition	<i>Actor</i> telah <i>login</i>
Flow of Event	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “biodata santri”. 2. Sistem akan menampilkan halaman pendaftaran santri dan menampilkan data santri. 3. Actor mengklik tombol “tambah santri” 4. Sistem akan menampilkan halaman pengajuan santri dan <i>form</i> tambah pengajuan santri. 5. <i>Actor</i> meng-<i>input</i> data santri. 6. <i>Actor</i> mengklik tombol “submit” saat data benar. 7. Sistem akan menyimpan data santri dengan. Lalu Sistem akan kembali ke halaman biodata santri dan menampilkan data santri.
Scenario Alternative	<ol style="list-style-type: none"> 8. Actor menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	Actor berhasil menambahkan pengajuan aset.

14. Use case scenario ubah biodata santri masjid

Use Case	Biodata santri (ubah)
Actor	Jamaah
Entry Condition	<i>Actor</i> telah <i>login</i>
Flow of Event	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “biodata santri”. 2. Sistem akan menampilkan halaman pendaftaran santri dan menampilkan data santri.

	<ol style="list-style-type: none"> 3. Actor mengklik tombol “edit” 4. Sistem akan menampilkan halaman ubah santri dan <i>form</i> ubah santri. 5. <i>Actor</i> mengubah data santri. 6. <i>Actor</i> mengklik tombol “submit” saat data benar. 7. Sistem akan mengubah data santri. Lalu Sistem akan kembali ke halaman pendaftaran santri dan menampilkan data santri.
Scenario Alternative	<ol style="list-style-type: none"> 8. Actor menekan tombol “Submit” saat inputan data salah atau kosong. 9. Sistem menampilkan notifikasi kesalahan.
Exit Scenario	Actor berhasil mengubah data santri.

15. Use case scenario hapus biodata santri

<i>Use Case</i>	Biodata santri (hapus)
<i>Actor</i>	Jamaah
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “biodata santri”. 2. Sistem akan menampilkan halaman tambah santri dan menampilkan data santri. 3. Actor mengklik tombol “hapus” 4. Sistem akan menghapus data dan menampilkan halaman biodata santri serta menampilkan data santri.
<i>Exit Scenario</i>	Actor berhasil menghapus data santri.

16. Use case scenario mendaftarkan santri masjid

<i>Use Case</i>	Mendaftarkan santri (tambah)
<i>Actor</i>	Jamaah
<i>Entry Condition</i>	<i>Actor</i> telah <i>login</i>
<i>Flow of Event</i>	<ol style="list-style-type: none"> 10. <i>Actor</i> membuka halaman “pendaftaran santri”. 11. Sistem akan menampilkan halaman pendaftaran santri dan menampilkan data pendaftaran. 12. Actor mengklik tombol “ajukan santri” 13. Sistem akan menampilkan halaman

	<p>pengajuan santri dan <i>form</i> tambah pengajuan santri.</p> <p>14. <i>Actor</i> meng-<i>input</i> data pengajuan santri.</p> <p>15. <i>Actor</i> mengklik tombol “submit” saat data benar.</p> <p>16. Sistem akan menyimpan data santri dengan menambahkan status pada data sebagai “diajukan”. Lalu Sistem akan kembali ke halaman pendaftaran santri dan menampilkan data pendaftarab.</p>
Scenario Alternative	<p>17. Actor menekan tombol “Submit” saat inputan data salah atau kosong.</p> <p>18. Sistem menampilkan notifikasi kesalahan.</p>
Exit Scenario	Actor berhasil menambahkan pengajuan santri.

17. Use case scenario ubah santri masjid

Use Case	Mendaftarkan santri (ubah)
Actor	Jamaah
Entry Condition	<i>Actor</i> telah <i>login</i>
Flow of Event	<p>10. <i>Actor</i> membuka halaman “pendaftaran santri”.</p> <p>11. Sistem akan menampilkan halaman pendaftaran santri dan menampilkan data santri.</p> <p>12. Actor mengklik tombol “edit”</p> <p>13. Sistem akan menampilkan halaman ubah santri dan <i>form</i> ubah santri.</p> <p>14. <i>Actor</i> mengubah data pendaftaran santri.</p> <p>15. <i>Actor</i> mengklik tombol “submit” saat data benar.</p> <p>16. Sistem akan mengubah data pendaftaran. Lalu Sistem akan kembali ke halaman pendaftaran santri dan menampilkan data santri.</p>
Scenario Alternative	<p>17. Actor menekan tombol “Submit” saat inputan data salah atau kosong.</p> <p>18. Sistem menampilkan notifikasi kesalahan.</p>
Exit Scenario	Actor berhasil mengubah data pendaftaran.

18. Use case scenario hapus santri yang didaftarkan

<i>Use Case</i>	Mendaftarkan santri (hapus)
<i>Actor</i>	Jamaah
<i>Entry Condition</i>	<i>Actor</i> telah login
<i>Flow of Event</i>	<ol style="list-style-type: none"> 5. <i>Actor</i> membuka halaman “pendaftaran santri”. 6. Sistem akan menampilkan halaman mengajukan santri dan menampilkan data pendaftaran. 7. <i>Actor</i> mengklik tombol “hapus” 8. Sistem akan menghapus data santri dan menampilkan halaman mengajukan santri serta menampilkan data pendaftaran.
<i>Exit Scenario</i>	<i>Actor</i> berhasil menghapus data pendaftaran.

19. Use case scenario menerima santri yang didaftarkan

<i>Use Case</i>	Menerima santri
<i>Actor</i>	Admin Bendahara
<i>Entry Condition</i>	<i>Actor</i> telah login
<i>Flow of Event</i>	<ol style="list-style-type: none"> 1. <i>Actor</i> membuka halaman “menerima santri”. 2. Sistem akan menampilkan halaman menerima santri dan menampilkan data santri yang diajukan dan data santri yang telah disetujui. 3. <i>Actor</i> mengklik tombol “terima data diceklis” 4. Sistem akan mengubah status santri menjadi “disetujui” dan sistem menampilkan halaman menerima santri serta menampilkan data santri.
<i>Exit Scenario</i>	<i>Actor</i> berhasil menerima santri.

20. Use case scenario menyetujui kegiatan yang diusulkan

<i>Use Case</i>	Menyetujui kegiatan
<i>Actor</i>	Admin Bendahara
<i>Entry Condition</i>	<i>Actor</i> telah login
<i>Flow of Event</i>	<p>5. <i>Actor</i> membuka halaman “menyetujui kegiatan”.</p> <p>6. Sistem akan menampilkan halaman menyetujui kegiatan dan menampilkan data kegiatan yang diajukan dan data kegiatan yang telah disetujui.</p> <p>7. <i>Actor</i> mengklik tombol “terima data diceklis”</p> <p>8. Sistem akan mengubah status kegiatan menjadi “disetujui” dan sistem menampilkan halaman menyetujui kegiatan serta menampilkan data kegiatan.</p>
<i>Exit Scenario</i>	<i>Actor</i> berhasil menyetujui kegiatan.

21. Use case scenario menyetujui aset yang didaftarkan

<i>Use Case</i>	Menyetujui aset
<i>Actor</i>	Admin Bendahara
<i>Entry Condition</i>	<i>Actor</i> telah login
<i>Flow of Event</i>	<p>9. <i>Actor</i> membuka halaman “menyetujui aset”.</p> <p>10. Sistem akan menampilkan halaman menyetujui aset dan menampilkan data aset yang diajukan dan data aset yang telah disetujui.</p> <p>11. <i>Actor</i> mengklik tombol “terima data diceklis”</p> <p>12. Sistem akan mengubah status aset menjadi “disetujui” dan sistem menampilkan halaman menerima aset serta menampilkan data aset.</p>

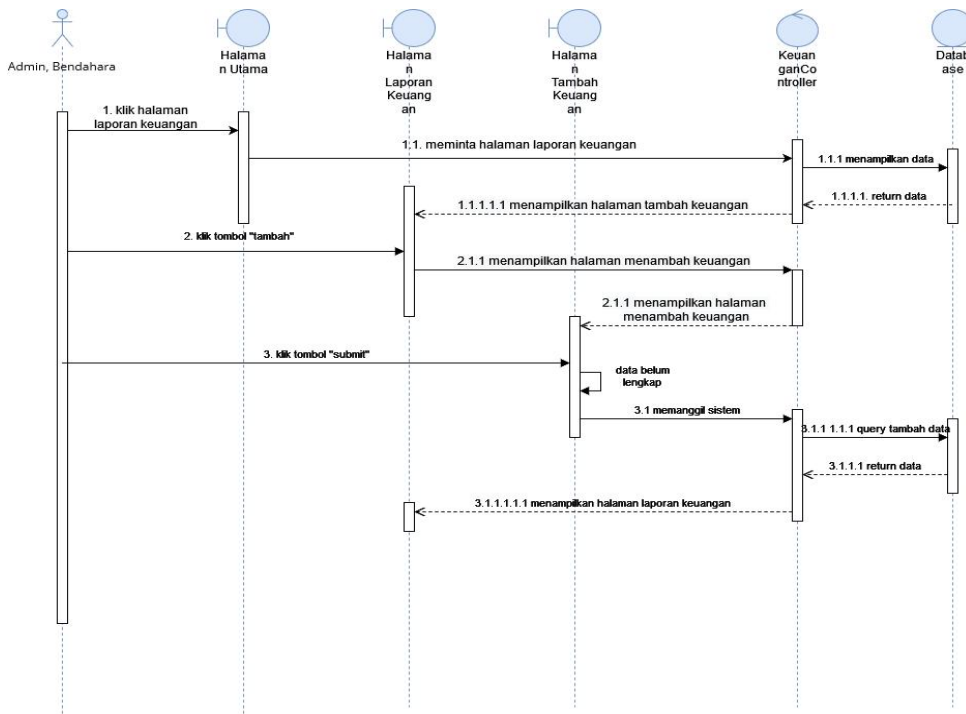
Exit Scenario

Actor berhasil menyetujui aset.



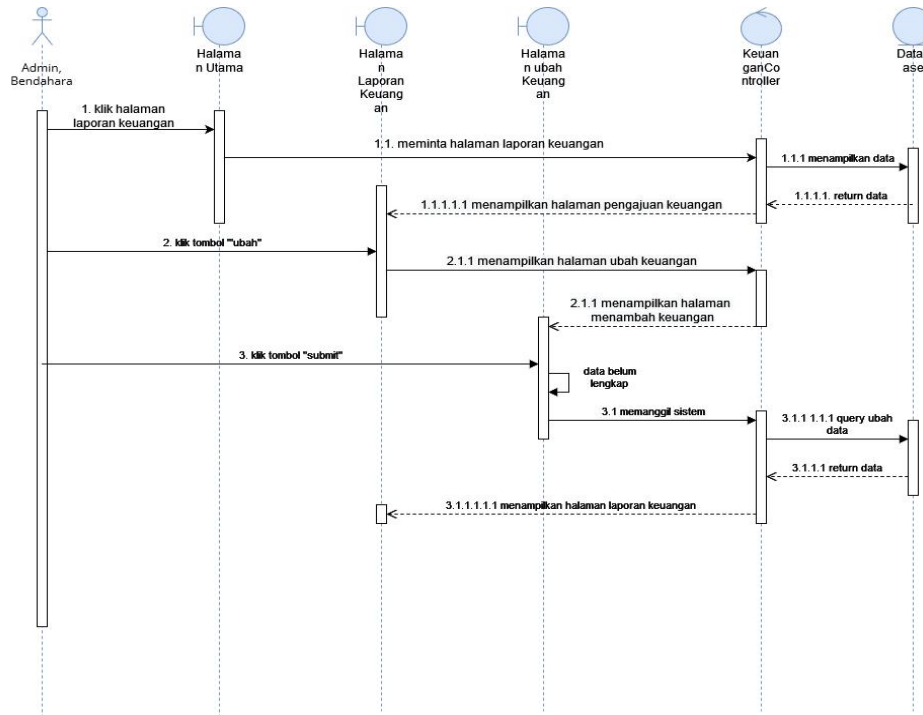
Sequence diagram aplikasi

1. Tambah laporan keuangan masjid

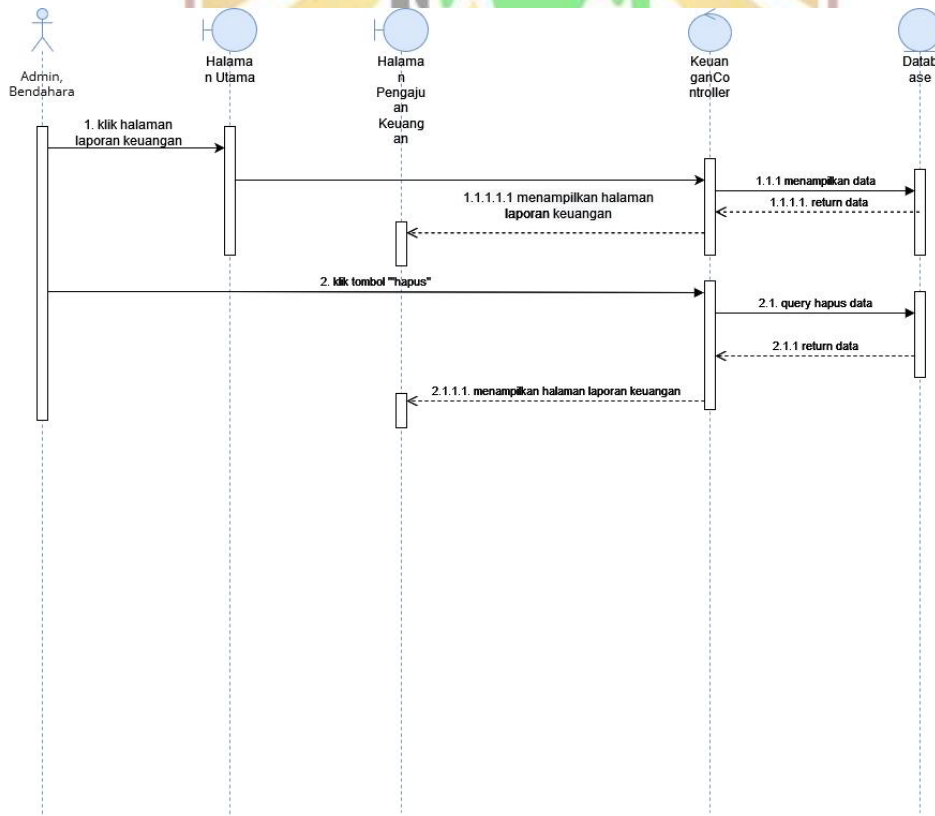


2. Ubah laporan keuangan masjid

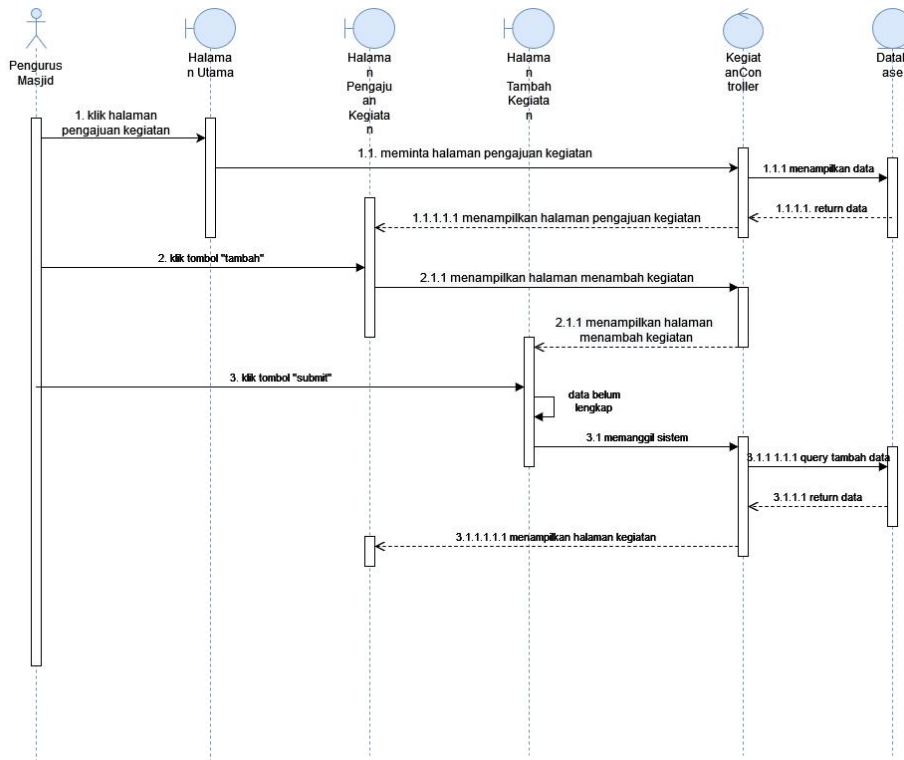




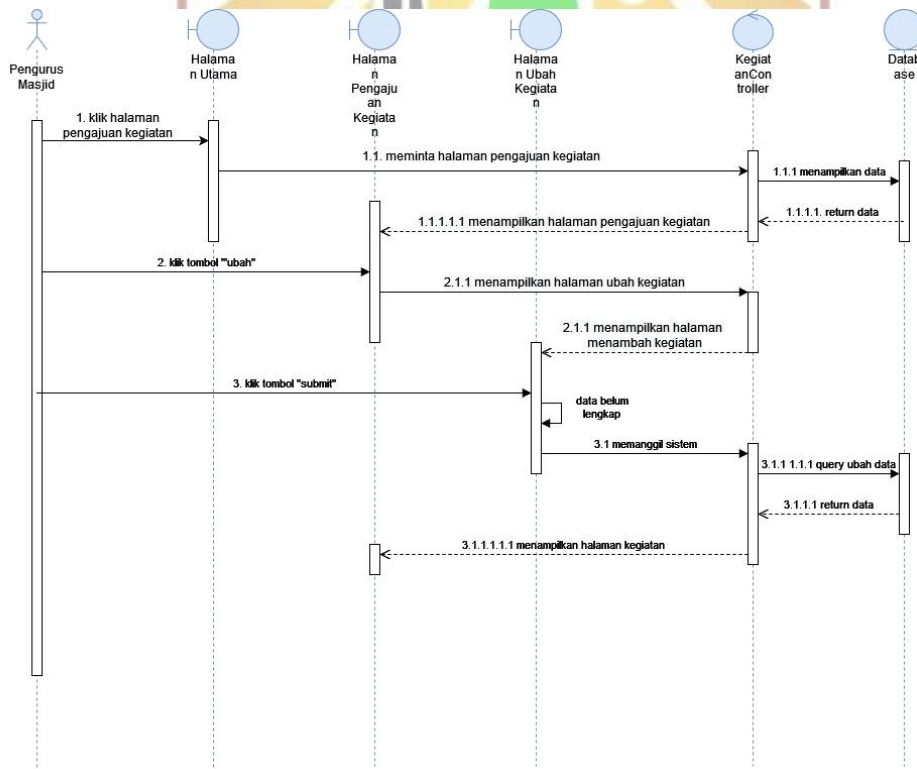
3. Hapus laporan keuangan masjid



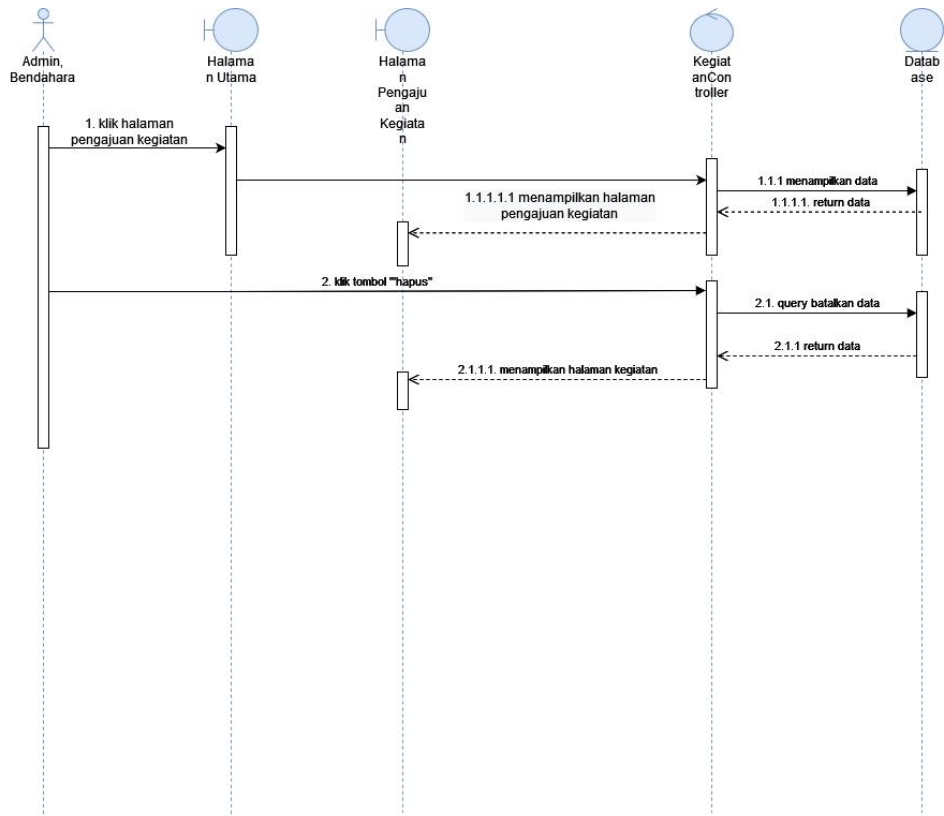
4. Tambah kegiatan masjid



5. Ubah kegiatan masjid

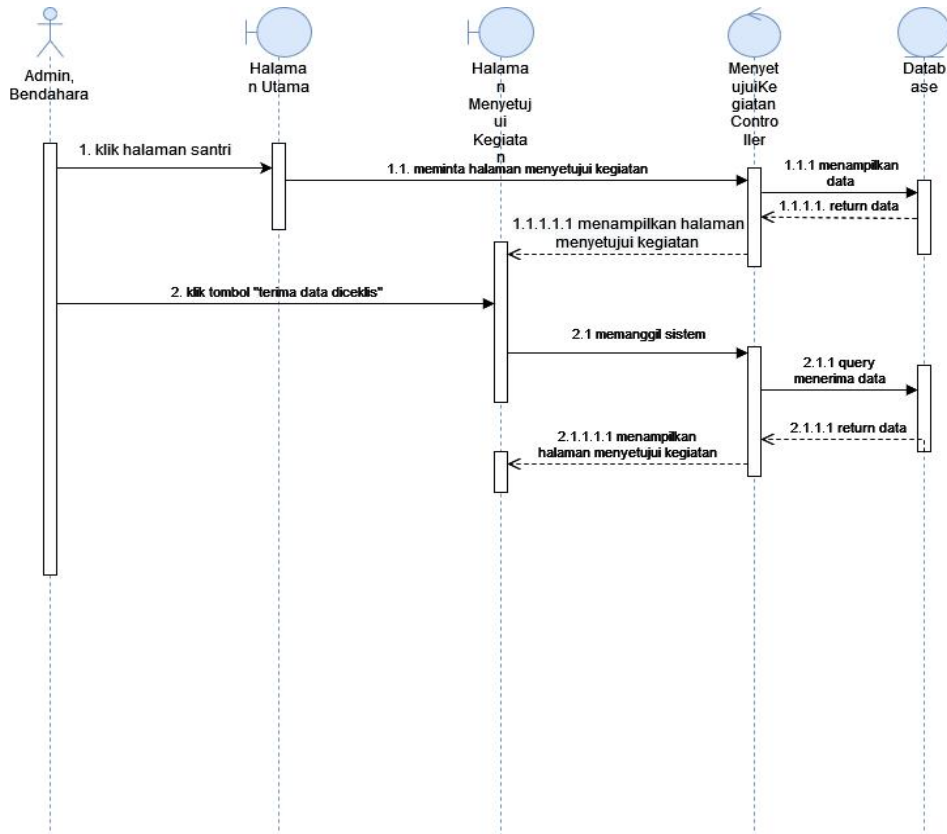


6. Membatalkan kegiatan masjid

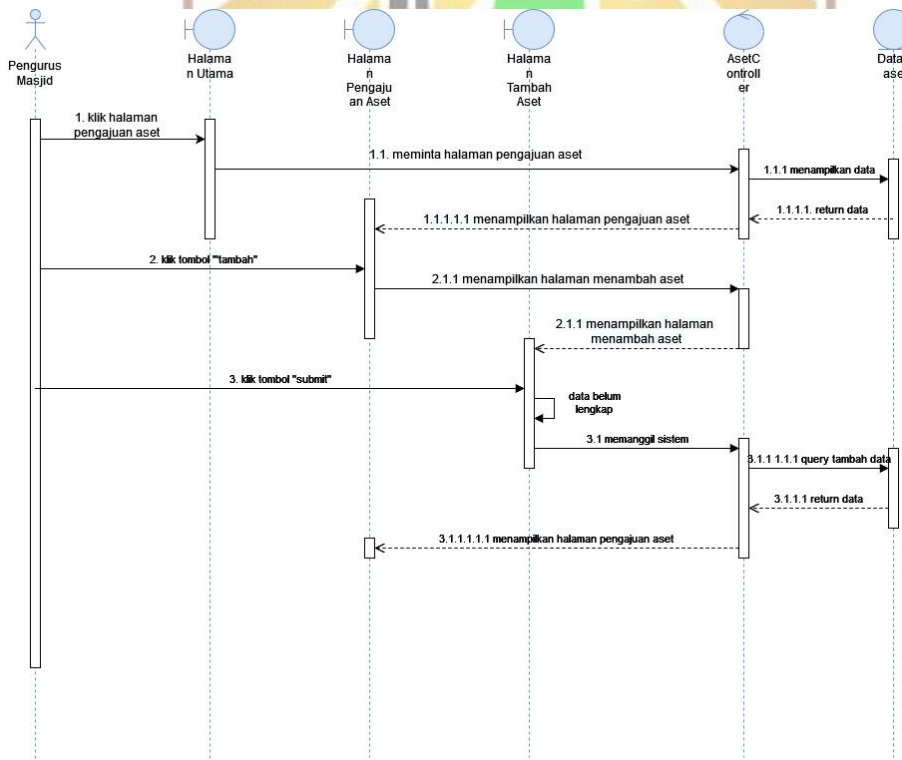


7. Menyetujui kegiatan masjid

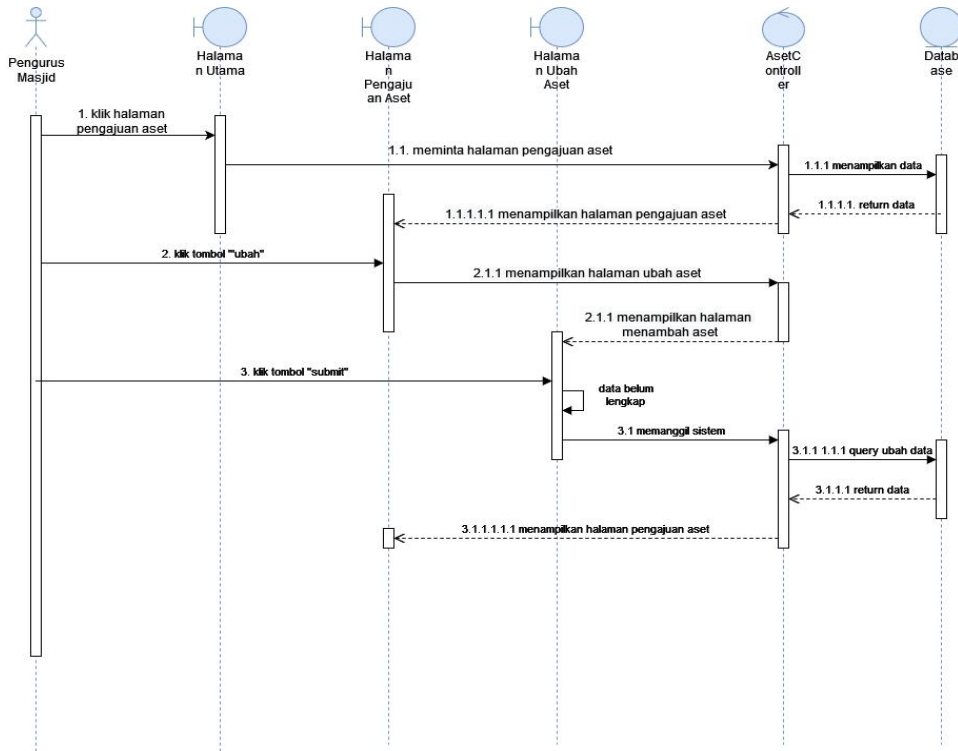




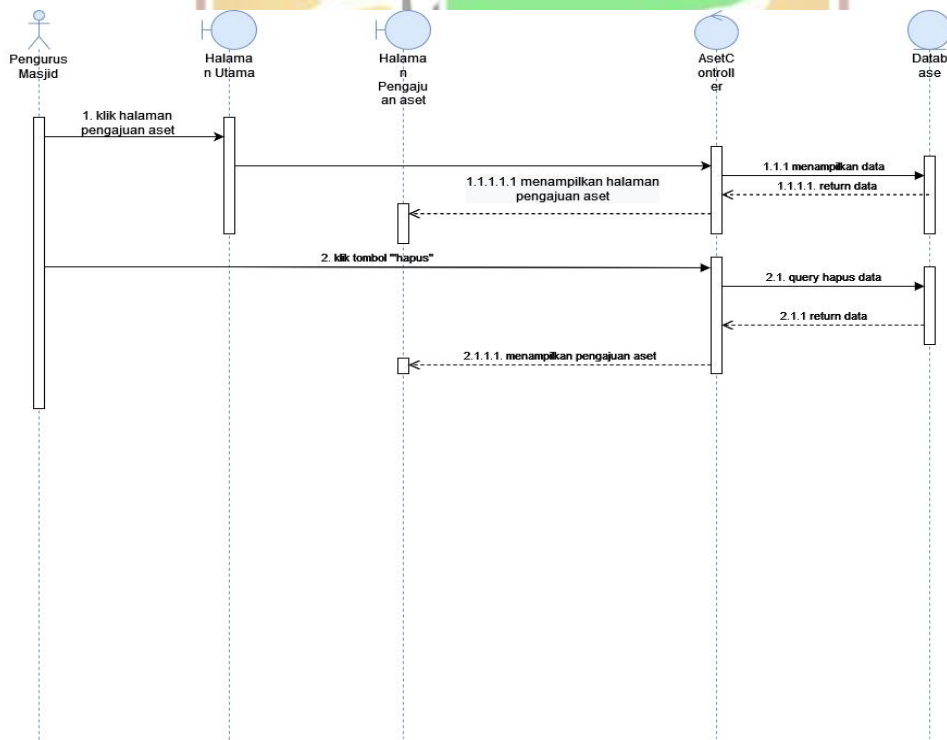
8. Menambah aset masjid



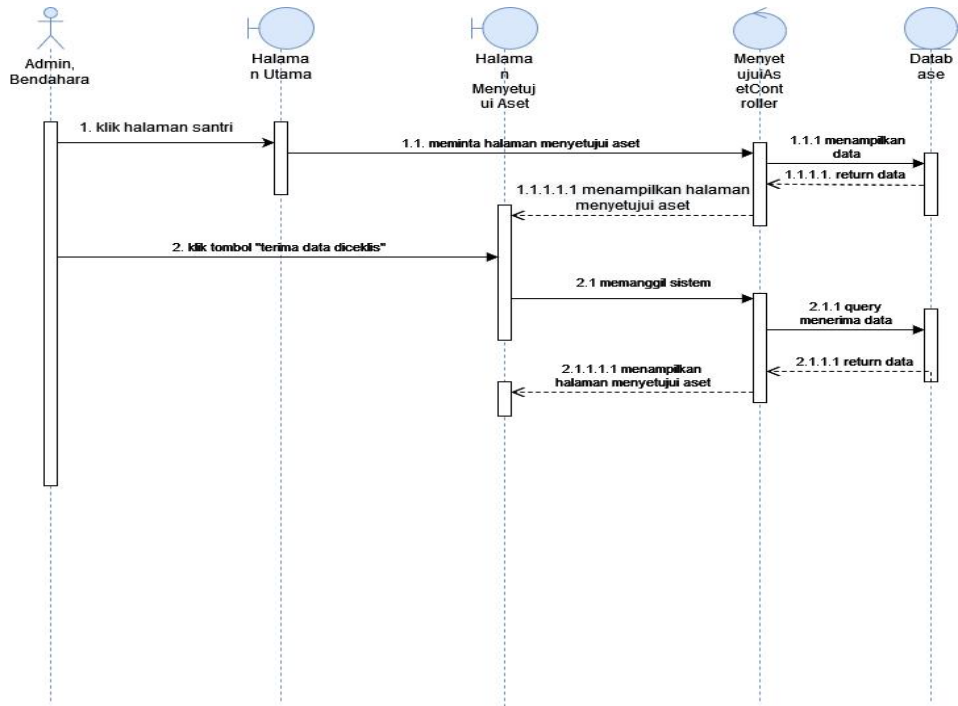
9. Mengubah aset masjid



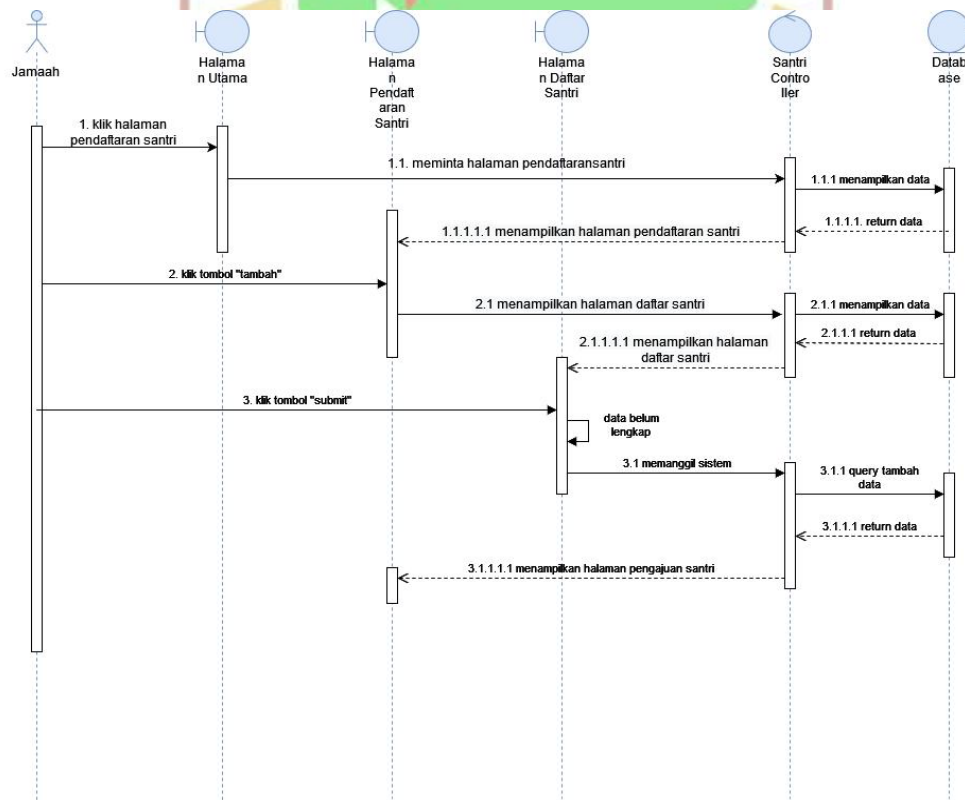
10. Hapus aset masjid



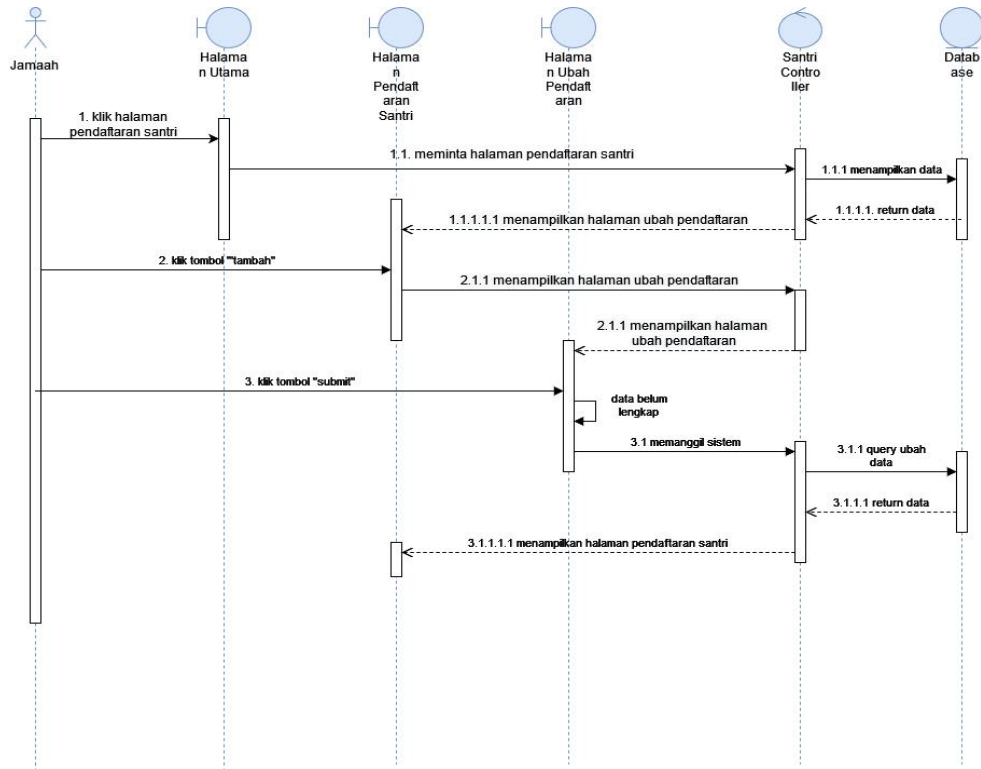
11. Menyetujui aset masjid



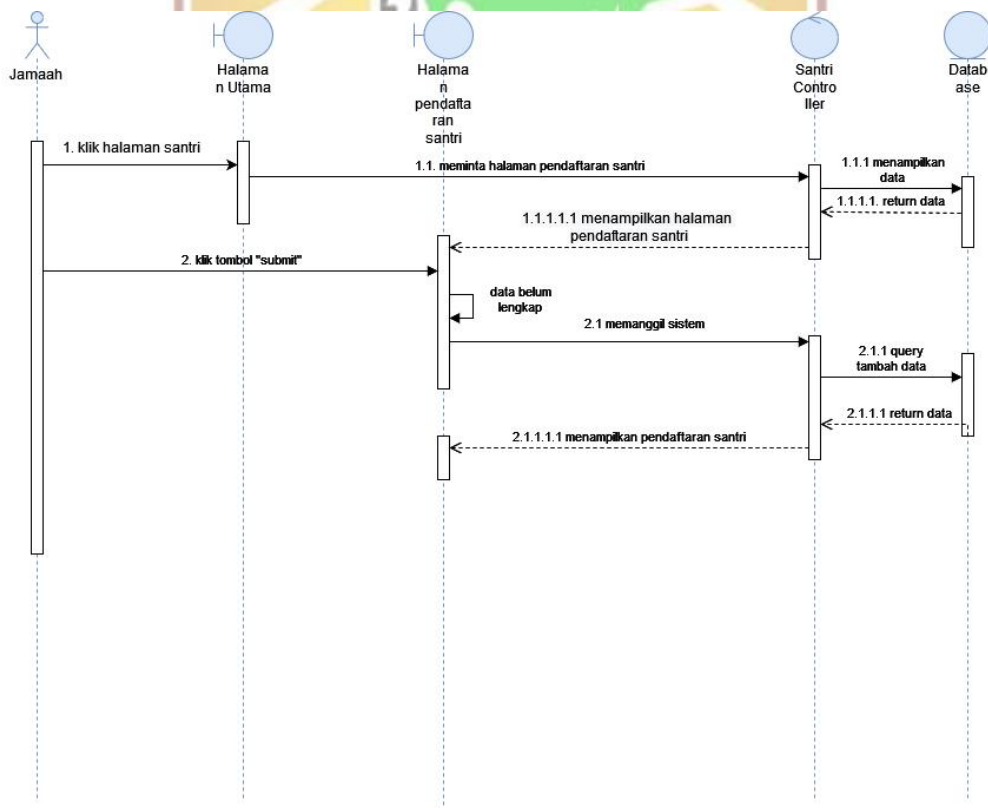
12. Mendaftarkan santri



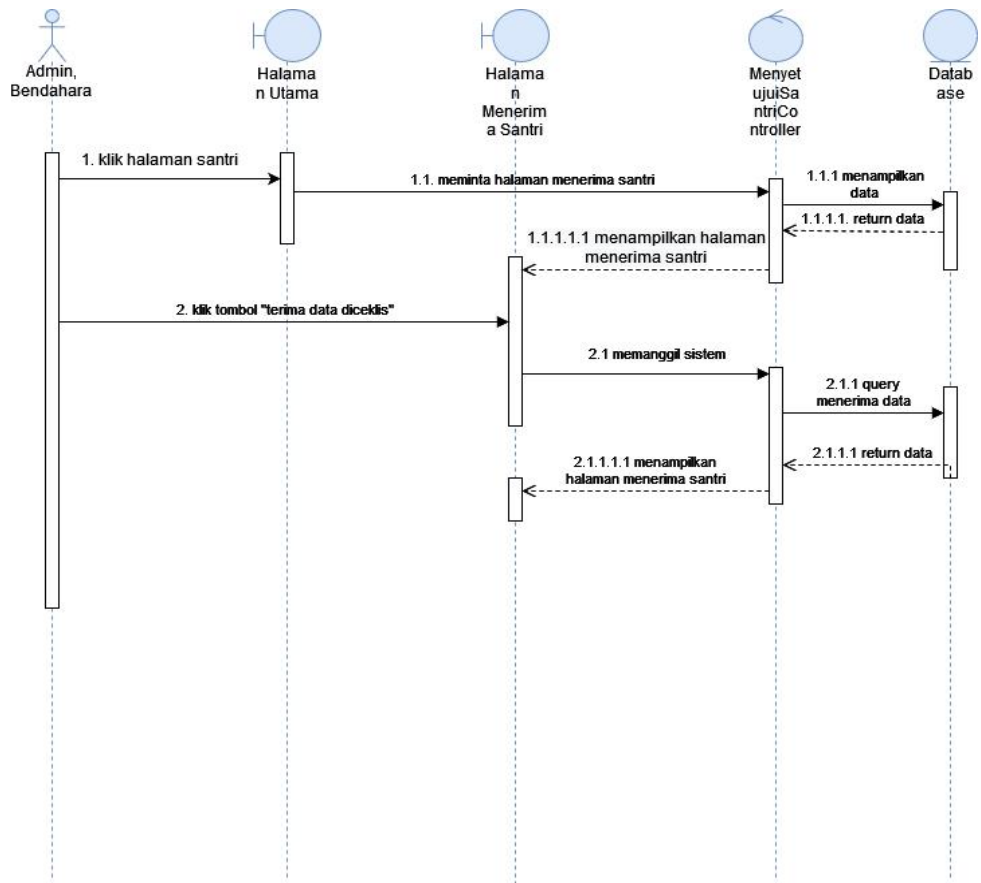
13. Mengubah data pendaftaran santri



14. Menghapus data pendaftaran santri



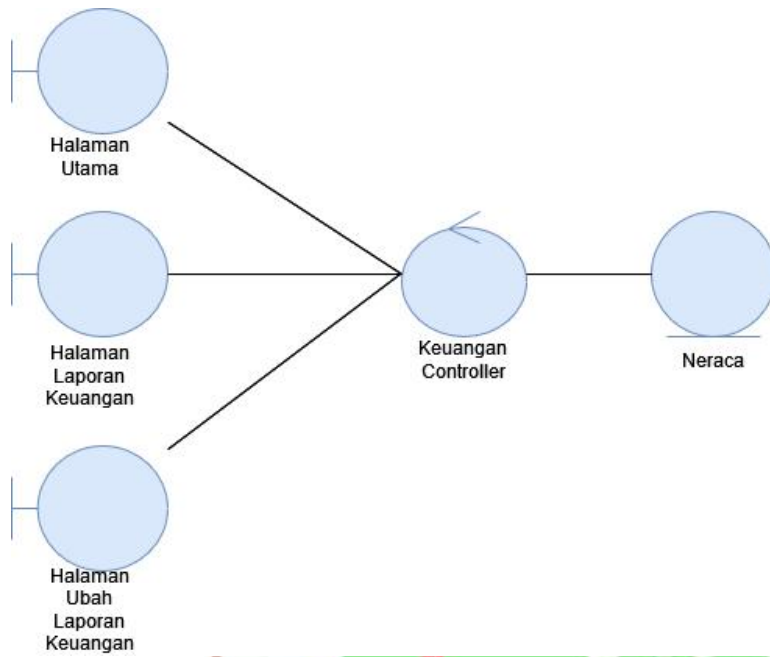
15. Menerima pendaftaran santri



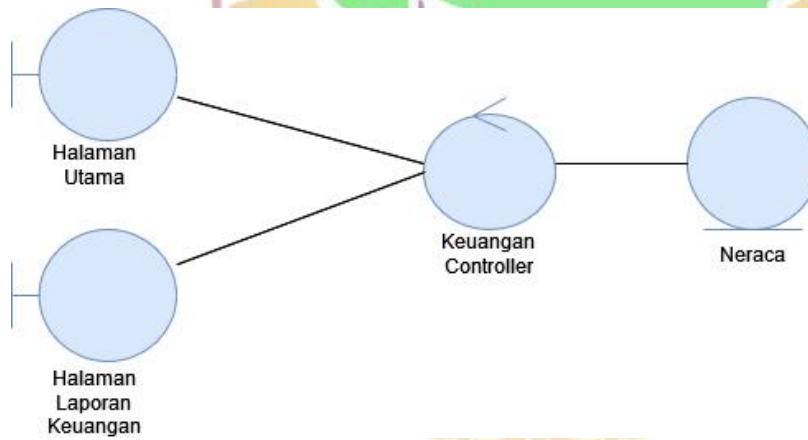


Class Analysis aplikasi

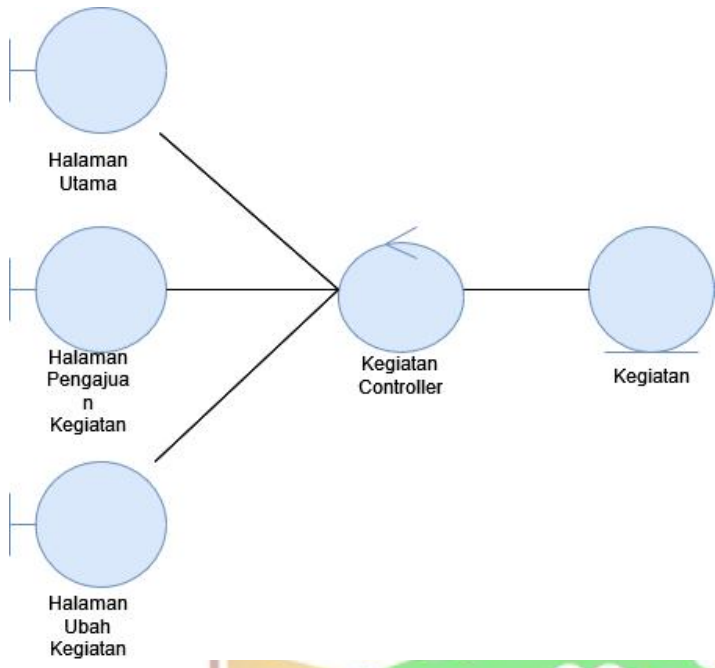
1. Ubah laporan keuangan



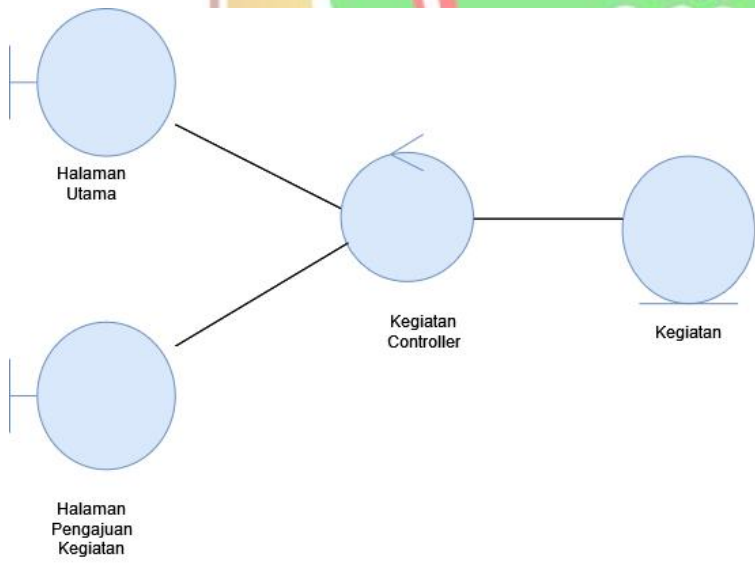
2. Hapus laporan keuangan



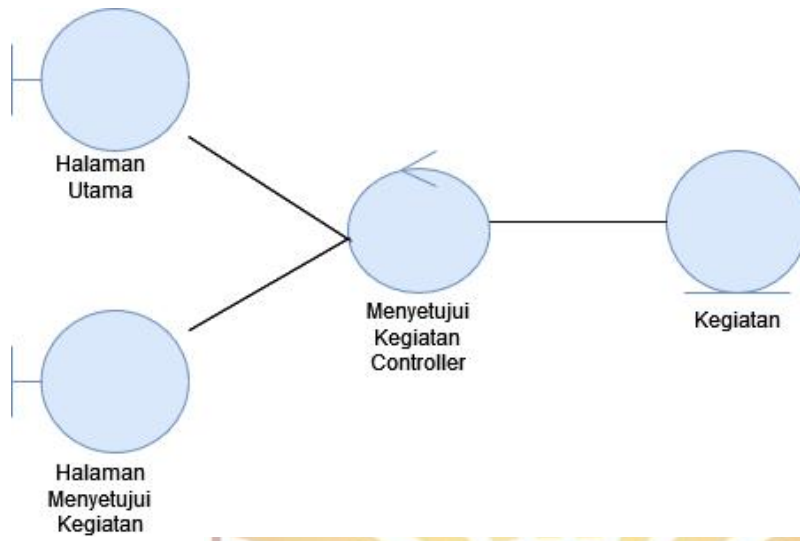
3. Ubah kegiatan masjid



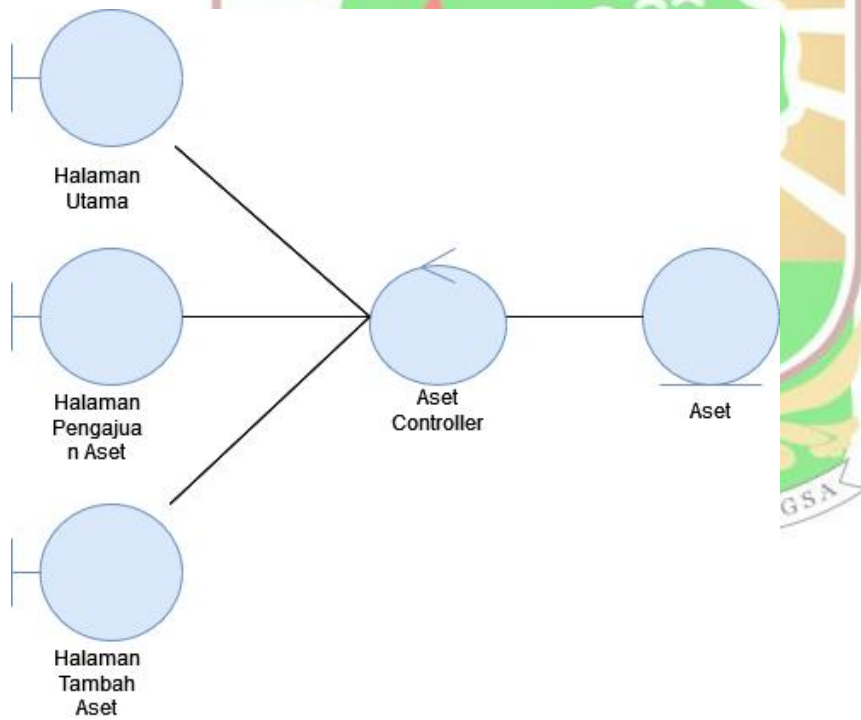
4. Hapus kegiatan masjid



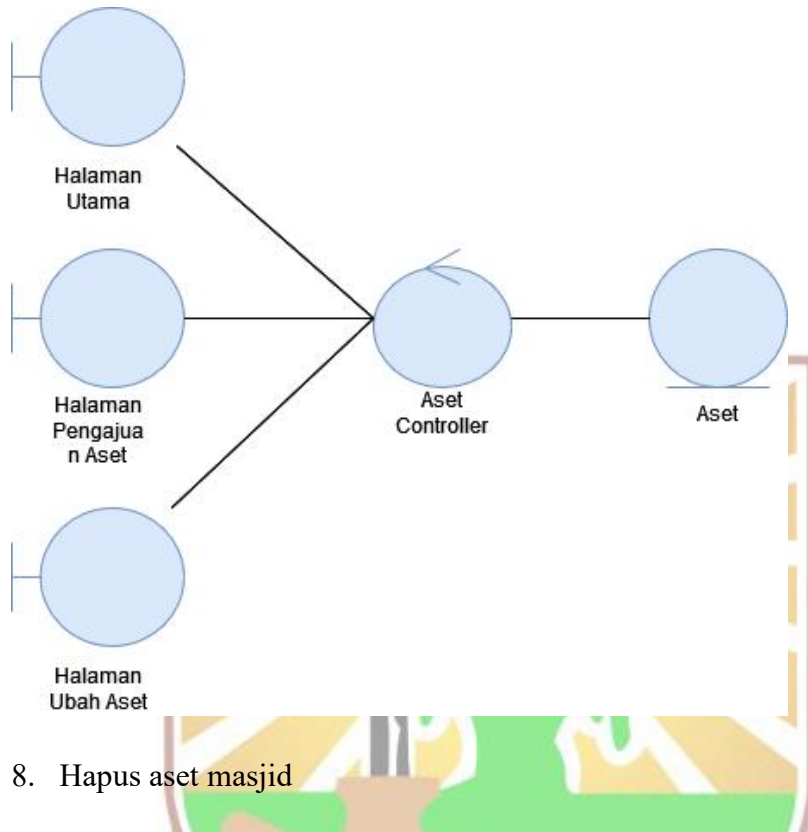
5. Menyetujui kegiatan masjid



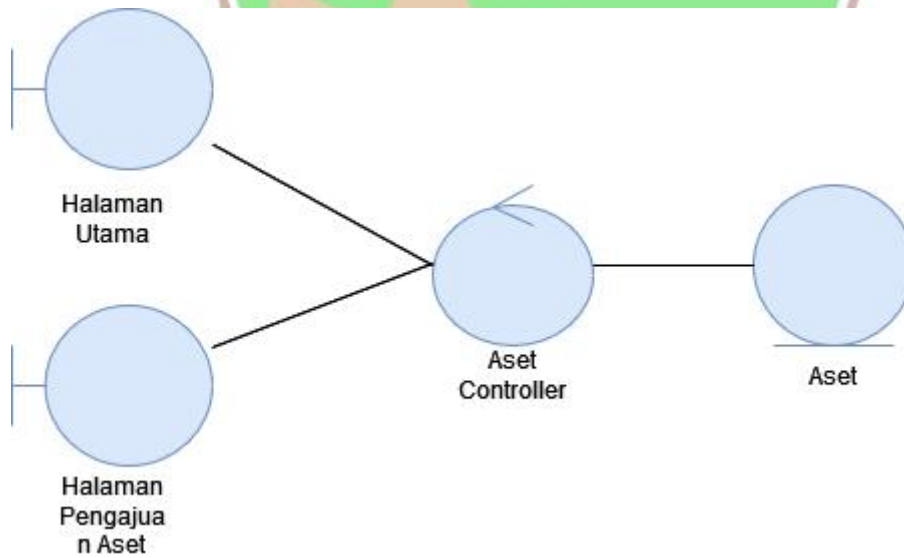
6. Mengajukan aset masjid



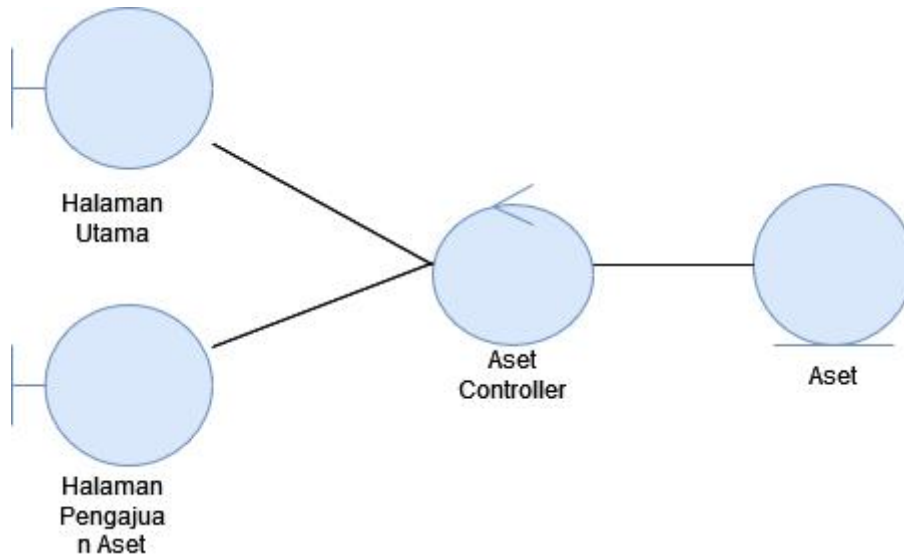
7. Ubah aset masjid



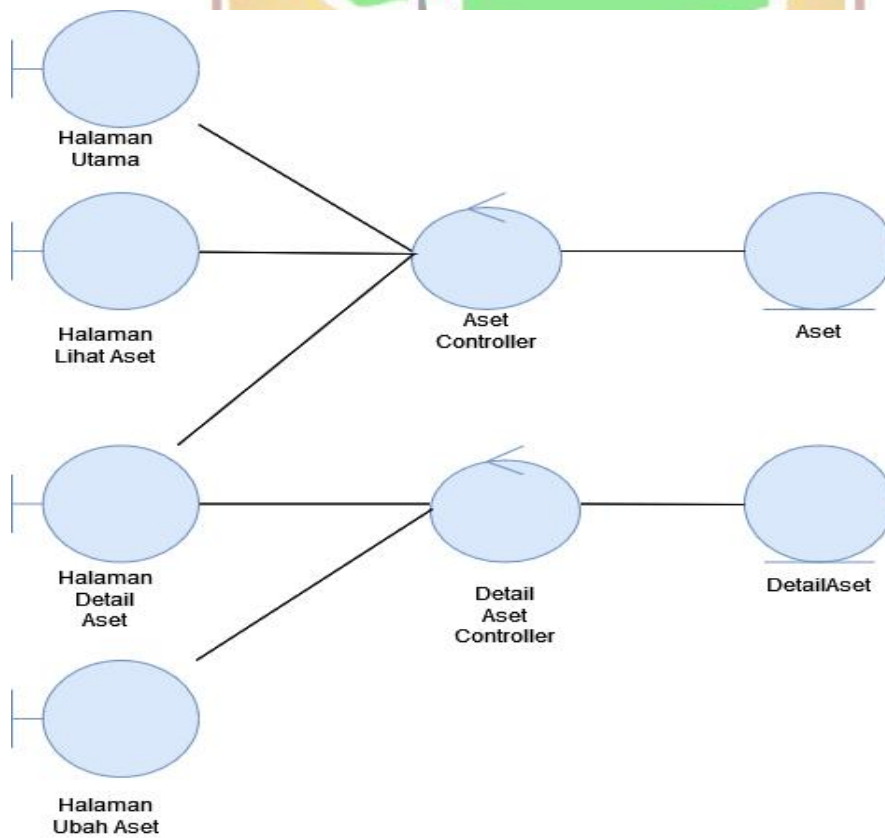
8. Hapus aset masjid



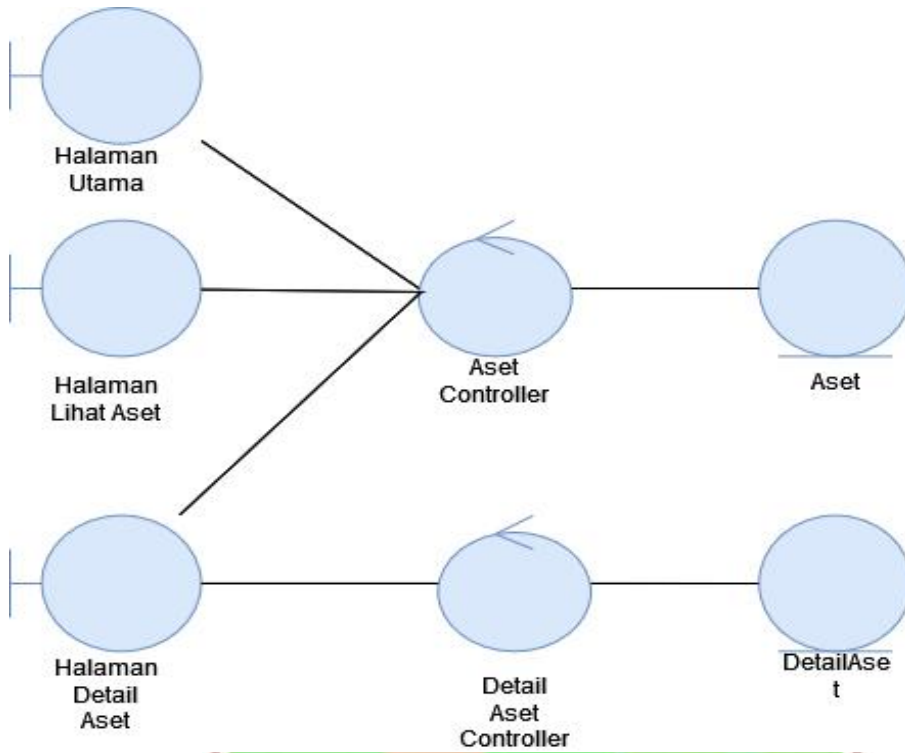
9. Menambah detail aset masjid



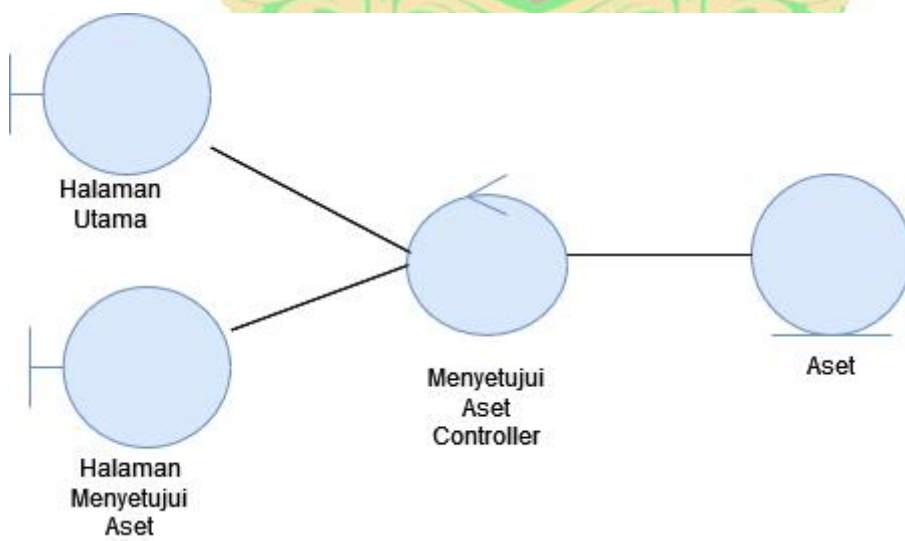
10. Ubah detail aset



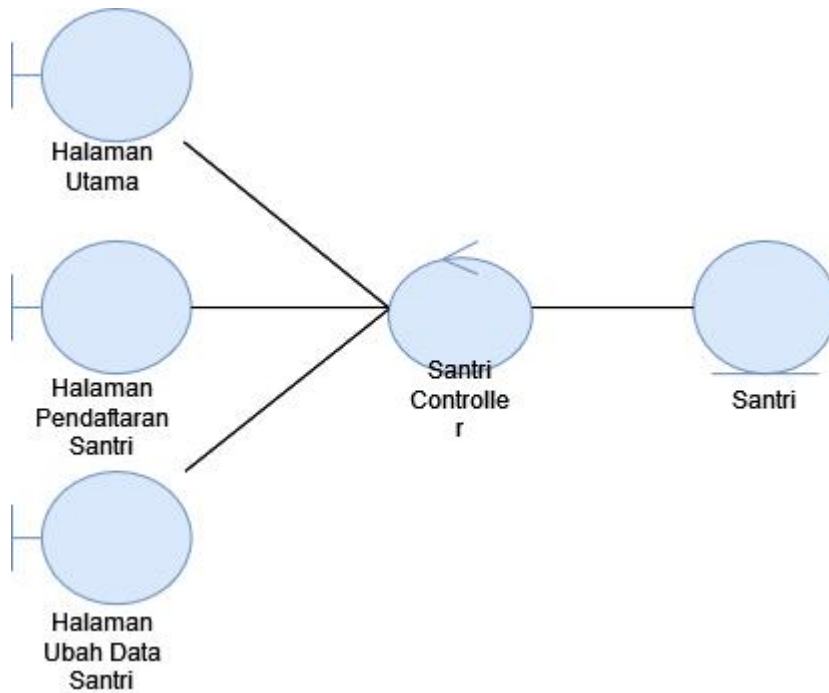
11. Hapus detail aset



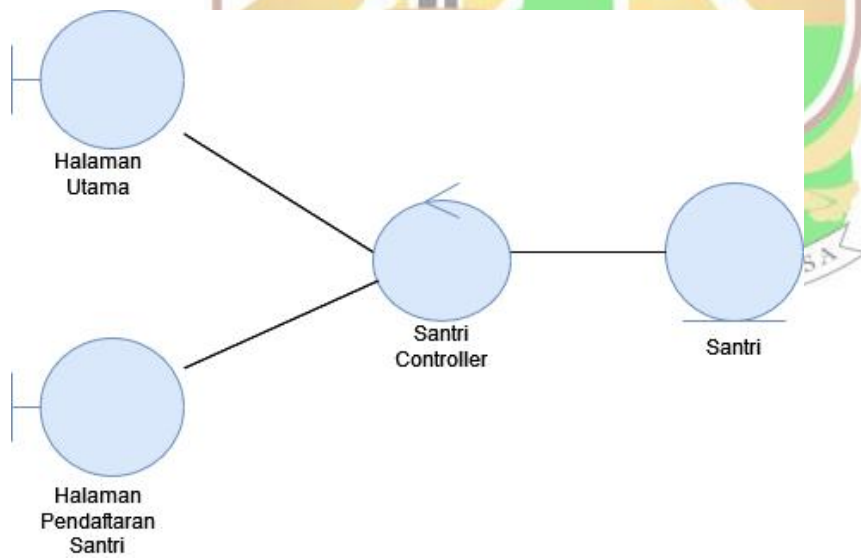
12. Menyetujui aset masjid



13. Mengubah data pendaftaran santri



14. Hapus data pendaftaran santri





LAMPIRAN D

Struktur tabel dan basis data aplikasi

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
user_id	int(10)	FOREIGN KEY
nama_santri	varchar(10)	
tempat_lahir	varchar (10)	
tgl_lahir	date(10)	
jns_kelamin	boolean	
nama_ayah	varchar (10)	
nama_ibu	varchar (10)	
nama_sekolah	varchar (10)	
kelas	int (10)	
no_telp	varchar (10)	
foto	varchar (10)	NULLABLE

1. Santri

2. Periode TPQ

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
nama_periode	varchar(10)	
tgl_mulai	date	
tgl_selesai	date	

3. Registrasi Santri

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
santri_id	int(10)	FOREIGN KEY
tpq_period_id	int(10)	FOREIGN KEY
status_penyetujuan	int (10)	
biaya_pendaftaran	int(10)	

4. User

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
user_id	int(10)	FOREIGN KEY
username	varchar(10)	
password	varchar (10)	
email	varchar(10)	

5. Role

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
Nama_role	varchar(10)	
role_id	int(10)	

6. Role User

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
user_id	int(10)	FOREIGN KEY
role_id	int(10)	

7. Detail Aset

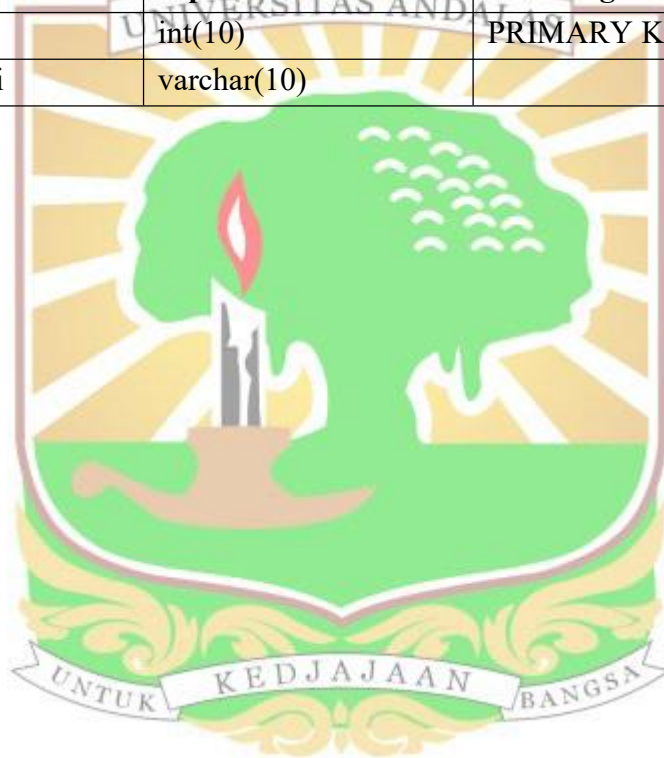
Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
aset_id	int(10)	FOREIGN KEY
tgl_pengadaan_aset	date	
kualitas	int(10)	
foto	varchar (10)	
harga_aset	int(10)	

8. Kategori Neraca

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
nama_kategori	varchar(10)	
debet_kredit	boolean	

9. Kategori Kegiatan

Nama Atribut	Tipe Data	Keterangan
id	int(10)	PRIMARY KEY
nama_kategori	varchar(10)	





LAMPIRAN E

Perancangan antarmuka aplikasi

1. Lihat laporan keuangan masjid

Profil User Logout

Laporan Keuangan

Pilih Data

Rp 7.000.500 TOTAL DEBET **Rp 1.500.500** TOTAL KREDIT **Rp 7.000.500** SISA SALDO

Daftar Pembukuan

Tgl. Input	Deskripsi	Kategori	No.Invoice	Tanggal Diterima	Debet (Rp)	Kredit (Rp)	Saldo (Rp)	Opsi
19 March 2023 13:45:26	Uang Garin	Uang Garin	04	14 February 2023	0	Rp 400.000	- Rp 400.000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
19 March 2023 13:47:21	Infak	Infak	01	04 January 2023	Rp 2.000.000	0	Rp 2.000.000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
19 March 2023 14:06:25	Perlengkapan Masjid	Perlengkapan Masjid	05	19 March 2023	0	Rp 1.100.000	- Rp 1.100.000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
25 March 2023 12:44:49	Donasi	Donasi	03	25 March 2023	Rp 5.000.500	0	Rp 5.000.500	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

2. Tambah laporan keuangan masjid

Profil User Logout

Laporan Keuangan

Tambah Data

Tanggal Diterima

Deskripsi

Nomor Invoice

Kategori

Debet/Kredit

Jumlah Dana (Rp)

3. Lihat kegiatan masjid

Profil User Logout

Kegiatan Masjid

Pilih Data Tahun

Daftar Kegiatan

Nama Kegiatan	Kategori Kegiatan	Tanggal Mulai	Tanggal Selesai	Detail
Kegiatan Pesantren	Pesantren	29 Maret 2023 04:00	12 April 2023 08:00	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Ceramah Mingguan	Kegiatan Mingguan	29 Maret 2023 19:00	29 Maret 2023 19:30	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Ceramah Mingguan	Kegiatan Mingguan	30 Maret 2023 19:00	30 Maret 2023 19:30	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

4. Mengajukan kegiatan masjid

Profil User Logout

Mengajukan Kegiatan Masjid

Daftar Kategori

Nama Kegiatan	Kategori Kegiatan	Tanggal Mulai	Tanggal Selesai	Status	Opsi
Pesantren Ramadhan	Pesantren	19-04-2023 00:08:56	14-04-2023 03:46:53	Disetujui	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mengaji Bersama	Kegiatan Mingguan	19-04-2023 18:49:07	24-04-2023 20:16:18	Disetujui	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

5. Tambah kegiatan masjid

Profil User Logout

User Name

Navigasi

- Dashboard
- Laporan Keuangan
- Kegiatan Masjid
- Aset Masjid
- Santri Masjid

Tambah Kegiatan

Tambah Data

Nama Kegiatan

Deskripsi

Kategori

Tanggal Mulai

Tanggal Selesai

Status

6. Menyetujui kegiatan masjid

Profil User Logout

User Name

Navigasi

- Dashboard
- Laporan Keuangan
- Kegiatan Masjid
- Aset Masjid
- Santri Masjid

Menyetujui Kegiatan Masjid

Daftar Kegiatan

Nama Aset	Kategori Aset	Jumlah Aset	Nama Pengaju	Status	Opsi
<input type="checkbox"/> Mobil Jenazah	Peralatan	2	Aff Maulana	Sedang Diproses	<input type="button" value="Detail"/>
<input type="checkbox"/> Air Conditioner	Peralatan	5	Aff Maulana	Sedang Diproses	<input type="button" value="Detail"/>

Ceklis Seluruh Data

7. Lihat aset masjid

Profil User Logout

Aset Masjid

Daftar Aset

Nama Aset	Kategori Aset	Jumlah Aset	Opsi
Mobil Jenazah	Peralatan	2	Detail Edit Delete
Air Conditioner	Peralatan	5	Detail Edit Delete

8. Mengajukan aset masjid

Profil User Logout

Mengajukan Aset Masjid

Daftar Kategori

[+ Ajukan Aset](#)

Nama Aset	Kategori Aset	Jumlah Aset	Status Aset	Opsi
Mobil Jenazah	Perlengkapan	2	Sedang Diproses	Detail Edit Delete

9. Tambah aset masjid

Profil User Logout

Tambah Aset

Tambah Data

Nama Aset

Jenis Aset

10. Detail aset masjid

Profil User Logout

Aset Masjid

Tambah Data

Tanggal Pengadaan Aset

Kualitas Aset

Foto Aset

Daftar Kategori

ID Aset	Tgl Pengadaan Aset	Foto Aset	Kualitas Aset	Opsi
1	25 Juni 2023		Baik	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	25 Juni 2023		Baik	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

11. Ubah detail aset masjid

Profil User Logout

Ubah Detail Aset Masjid

Ubah Data

Kualitas Aset
Baik

Foto Aset

12. Biodata santri

Profil User Logout

Biodata Santri

Daftar Santri

Nama Santri	Sekolah	Kelas	Jenis Kelamin	Opsi
Aliya Elvania Chalisa	SDN 26	2	Perempuan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Aqlia Queensa Rafia	SDN 26	2	Perempuan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

13. Tambah santri

Profil User Logout

User Name

Navigasi

- Dashboard
- Laporan Keuangan
- Kegiatan Masjid
- Aset Masjid
- Santri Masjid

Daftar Santri

Tambah Data

Nama Santri

Tempat / Tanggal Lahir

Jenis Kelamin

Nomor Telepon

Alamat

Nama Ayah Nama Ibu

Nama Sekolah Kelas

Foto Santri

14. Menerima santri

Profil User Logout

User Name

Navigasi

- Dashboard
- Laporan Keuangan
- Kegiatan Masjid
- Aset Masjid
- Santri Masjid

Menerima Santri

Daftar Santri

Nama Santri	Sekolah	Kelas	Jenis Kelamin	Nama Pengaju	Status	Opsi
<input type="checkbox"/> Aliya Elvania Chalisa	SDN 26	2	Perempuan	Afif Maulana Isman	Sedang Diproses	<input type="button" value="Detail"/>
<input type="checkbox"/> Aqila Queensa Rafia	SDN 26	2	Perempuan	Afif Maulana Isman	Sedang Diproses	<input type="button" value="Detail"/>

Ceklis Seluruh Data



LAMPIRAN F

Kode aplikasi

1. AcceptActivityController

```
<?php
namespace App\Http\Controllers;

use App\Models\Activity;
use Illuminate\Http\Request;

class AcceptActivityController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $x_activities = Activity::where('submission_status', 0)->get();
        $y_activities= Activity::where('submission_status', '!=',0)
        ->get();
        return view('backend.admin.accept_activity.index',
        compact('x_activities', 'y_activities'));
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $activity= Activity::find($id);
        return view('backend.admin.accept_activity.show', compact('activity'));
    }

    public function accept($id)
    {
        Activity::find($id)->update(['submission_status' => 1]);
        return redirect()->route('admin.accept_activity.index')
        ->with('success', 'Data berhasil diubah');
    }

    public function deny($id)
    {
        Activity::find($id)->update(['submission_status' => 2]);
        return redirect()->route('admin.accept_activity.index')
        ->with('success', 'Data berhasil diubah');
    }

    public function accept_checked(Request $request)
    {
        Activity::whereIn('id', $request->activity_id)
        ->update(array('submission_status' => $request->accept_checked));
        return redirect()->route('admin.accept_activity.index')
        ->with('success', 'Data berhasil diubah');
    }
}
```

2. AcceptAssetController

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Asset;
use App\Models\AssetDetail;

class AcceptAssetController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $x_assets = Asset::where('submission_status', 0)->get();
        $y_assets= Asset::where('submission_status', '!=',0)->get();
        return view('backend.admin.accept_asset.index',
            compact('x_assets', 'y_assets'));
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $asset = Asset::find($id);
        $as_details= AssetDetail::where('asset_id',$id)->get();
        return view('backend.admin.accept_asset.show',
            compact('asset', 'as_details'));
    }

    public function accept($id){
        Asset::find($id)->update(['submission_status' => 1]);
        return redirect()->route('admin.accept_asset.index');
    }

    public function deny($id){
        Asset::find($id)->update(['submission_status' => 2]);
        return redirect()->route('admin.accept_asset.index');
    }

    public function accept_checked(Request $request){
        Asset::whereIn('id', $request->asset_id)->
            update(array('submission_status' => $request->accept_checked));
        return redirect()->route('admin.accept_asset.index')
            ->with('success', 'Data berhasil diubah');
    }
}
```

3. AcceptSantriController

```
<?php
namespace App\Http\Controllers;

use App\Models\Santri;
use Illuminate\Http\Request;

class AcceptSantriController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $santries = Santri::all();
        $x_santries = Santri::where('submission_status', 0)->get();
        $y_santries= Santri::where('submission_status', '!=',0)->get();
        return view('backend.admin.accept_santri.index',
            compact('x_santries', 'y_santries'));
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $santri= Santri::find($id);
        return view('backend.admin.accept_santri.show', compact('santri'));
    }

    public function accept(Request $request, $id){
        Santri::find($id)->update(['submission_status' => 1]);
        return redirect()->route('admin.accept_santri.index');
    }

    public function deny(Request $request, $id){
        Santri::find($id)->update(['submission_status' => 2]);
        return redirect()->route('admin.accept_santri.index');
    }

    public function accept_checked(Request $request){
        Santri::whereIn('id', $request->santri_id)-
        >update(array('submission_status'
            => $request->accept_checked));
        return redirect()->route('admin.accept_santri.index')
            ->with('success', 'Data berhasil diubah');
    }
}
```

4. ActivityCategoryController

```
<?php
namespace App\Http\Controllers;

use App\Models\ActivityCategory;
use Illuminate\Http\Request;

class ActivityCategoryController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $ac_categories = ActivityCategory::all();
        return view('backend.activity.categories.index', compact('ac_categories'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('backend.activity.categories.create');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        ActivityCategory::create($request->all());
        return redirect()->route('activity_categories.index')
            ->with('success', 'Data berhasil ditambahkan');
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param \App\Models\ActivityCategory $activityCategory
     * @return \Illuminate\Http\Response
     */
    public function edit(ActivityCategory $activityCategory)
    {
        $ac_category = ActivityCategory::find($activityCategory->id);
        // dd($ac_category);
        return view('backend.activity.categories.edit', compact('ac_category'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param \App\Models\ActivityCategory $activityCategory
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, ActivityCategory $activityCategory)
    {
        ActivityCategory::find($activityCategory->id)->update($request->all());
    }
}
```

```

        return redirect()->route('activity_categories.index')
        ->with('success', 'Data berhasil diubah');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  \App\Models\ActivityCategory $activityCategory
     * @return \Illuminate\Http\Response
     */
    public function destroy(ActivityCategory $activityCategory)
    {
        ActivityCategory::destroy($activityCategory->id);
        return redirect()->route('activity_categories.index')
        ->with('error', 'Data berhasil dihapus');
    }
}

```

5. ActivityController

```

<?php
namespace App\Http\Controllers;

use App\Models\User;
use App\Models\Activity;
use App\Models\ActivityCategory;
use Illuminate\Http\Request;

class ActivityController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $new_activity = new Activity;
        $years = $new_activity->getYear();
        $year = $years[0];
        $month = 0;

        $new_activities = Activity::oldest();
        $activities = $new_activities->whereMonth('schedule_start', date('m'))
        ->whereYear('schedule_start', $year)->orderBy('schedule_start', 'DESC')->get();
        return view('backend.activity.index', compact('activities', 'month', 'year', 'years', ));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */

    public function create()
    {
        $categories = ActivityCategory::pluck('id', 'category_name');
        return view('backend.activity.create', compact('categories'));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
}

```

```

public function store(Request $request)
{
    Activity::create($request->all());
    return redirect()->route('activity.propose')
        ->with('success', 'Data berhasil ditambahkan');
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Activity $activity
 * @return \Illuminate\Http\Response
 */
public function show(Activity $activity)
{
    $activity->find($activity->id);
    return view('backend.activity.show', compact('activity'));
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\Models\Activity $activity
 * @return \Illuminate\Http\Response
 */
public function edit(Activity $activity)
{
    $activity = Activity::find($activity->id);
    $categories = ActivityCategory::pluck('id', 'category_name');
    $activity_status = config('constants.activity_status');
    return view('backend.activity.edit', compact('activity', 'categories'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Activity $activity
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, Activity $activity)
{
    Activity::find($activity->id)->update($request->all());
    return redirect()->route('activity.propose')
        ->with('success', 'Data berhasil diubah');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Activity $activity
 * @return \Illuminate\Http\Response
 */
public function destroy(Activity $activity)
{
    Activity::destroy($activity->id);
    return redirect()->route('activity.index')
        ->with('error', 'Data berhasil dihapus');
}

public function propose()
{
    $user = User::find(auth()->user()->id);
    $activities = $user->Activities()->get();
    return view('backend.activity.propose', compact('activities'));
}

public function search(Request $request){
    $new_activity = new Activity;
    $years = $new_activity->getYear();
}

```



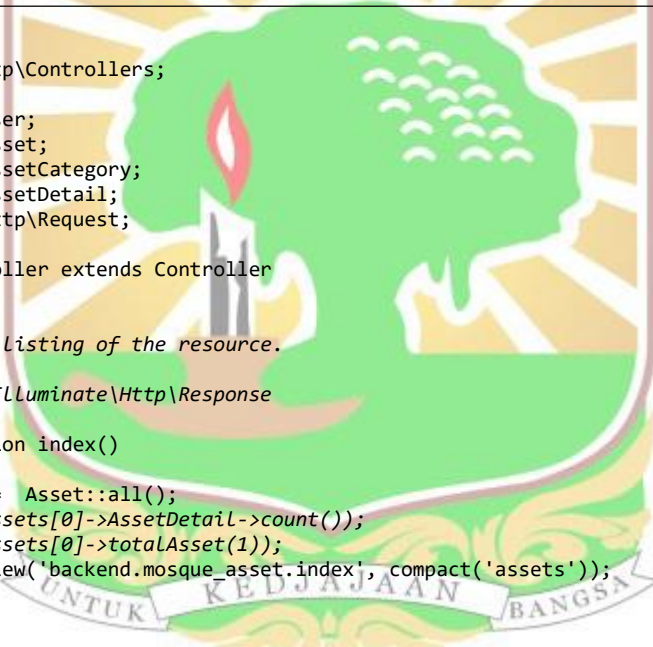
```

    $month = $request->month[0];
    $year = $request->year[0];
    if ($month != 0) {
        $new_activities = Activity::oldest();
        $activities = $new_activities
        ->whereMonth('schedule_start', $month)->whereYear('schedule_start', $year)
        ->orderBy('schedule_start', 'DESC')->get();
        return view('backend.activity.index',
            compact('activities', 'month', 'year', 'years', ));
    }

    $new_activities = Activity::oldest();
    $activities = $new_activities->whereYear('schedule_start', $year)
    ->orderBy('schedule_start', 'DESC')->get();
    return view('backend.activity.index',
        compact('activities', 'month', 'year', 'years', ));
}
}
}

```

6. AssetController



```

<?php
namespace App\Http\Controllers;

use App\Models\User;
use App\Models\Asset;
use App\Models\AssetCategory;
use App\Models\AssetDetail;
use Illuminate\Http\Request;

class AssetController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $assets = Asset::all();
        // dd($assets[0]->AssetDetail->count());
        // dd($assets[0]->totalAsset(1));
        return view('backend.mosque_asset.index', compact('assets'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $categories = AssetCategory::pluck('id', 'category_name');
        return view('backend.mosque_asset.create', compact('categories'));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        Asset::create($request->all());
        return redirect()->route('asset.index')
    }
}

```

```

->with('success', 'Data berhasil ditambahkan');
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Asset $asset
 * @return \Illuminate\Http\Response
 */
public function show(Asset $asset)
{
    $as_details= AssetDetail::where('asset_id',$asset->id)->get();
    $as_id = $asset->id;
    // dd($asset_details[0]->qualitytext());
    return view('backend.mosque_asset.show',compact('as_details', 'as_id'));
}

```

7. AssetDetailController

```

<?php
namespace App\Http\Controllers;

use App\Models\Asset;
use App\Models\AssetDetail;
use Illuminate\Http\Request;

class AssetDetailController extends Controller
{
    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        // dd(isset($request->photo));
        $asset = new AssetDetail;
        $asset->quality = $request->quality;
        $asset->asset_id = $request->asset_id;
        $asset->procurement_date = $request->procurement_date;

        if ( isset($request->photo) ) {
            $path=$request->file('photo')->store('mosque_assets', 'public');
            $asset->photo = '../../storage/'.$path;
        }
        $asset->save();
        return redirect()->back()->with('success', 'Data berhasil ditambahkan');
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param \App\Models\AssetDetail $assetDetail
     * @return \Illuminate\Http\Response
     */
    public function edit(AssetDetail $assetDetail)
    {
        $as_detail = AssetDetail::find($assetDetail->id);
        $as_id = $as_detail;
    }
}

```

```

        return view('backend.mosque_asset.details.edit',
            compact('as_detail', 'as_id'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param \App\Models\AssetDetail $assetDetail
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, AssetDetail $assetDetail)
    {
        $assetDetail->quality = $request->quality;
        if ( isset($request->photo) ) {
            $path=$request->file('photo')->store('mosque_assets', 'public');
            $assetDetail->photo = '../..../storage/'.$path;
        }
        $assetDetail->save();
        return redirect()
            ->route('asset.show', ['asset' => $assetDetail->asset_id])
            ->with('success', 'Data berhasil diubah');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param \App\Models\AssetDetail $assetDetail
     * @return \Illuminate\Http\Response
     */
    public function destroy(AssetDetail $assetDetail)
    {
        // dd($assetDetail);
        // $as_id = $assetDetail->asset_id;
        AssetDetail::destroy($assetDetail->id);
        return redirect()->back()->with('error', 'Data berhasil dihapus');
    }
}

```

8. BalanceCategoryController

```

<?php

namespace App\Http\Controllers;

use App\Models\BalanceCategory;
use Illuminate\Http\Request;

class BalanceCategoryController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $b_categories = BalanceCategory::all();
        // dd($b_categories[0]->id);
        return view('backend.balance.categories.index', compact('b_categories'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('backend.balance.categories.create');
    }
}

```

```

}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    BalanceCategory::create($request->all());
    return redirect()->route('balance_categories.index')
        ->with('success', 'Data berhasil ditambahkan');
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\Models\BalanceCategory $balanceCategories
 * @return \Illuminate\Http\Response
 */
public function edit(BalanceCategory $balanceCategories, $id)
{
    $b_category = BalanceCategory::find($id);
    return view('backend.balance.categories.edit', compact('b_category'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\BalanceCategory $balanceCategories
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, BalanceCategory $balanceCategories, $id)
{
    BalanceCategory::find($id)->update($request->all());
    return redirect()->route('balance_categories.index')
        ->with('success', 'Data berhasil diubah');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\BalanceCategory $balanceCategories
 * @return \Illuminate\Http\Response
 */
public function destroy(BalanceCategory $balanceCategories, $id)
{
    BalanceCategory::destroy($id);
    return redirect()->route('balance_categories.index')
        ->with('error', 'Data berhasil dihapus');
}
}

```

9. BalanceController

```

<?php

namespace App\Http\Controllers;

use App\Models\Balance;
use App\Models\BalanceCategory;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class BalanceController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
}

```

```

*/
public function index()
{
    $new_balance = new Balance;
    $years = $new_balance->getYear();
    $year = $years[0];
    $month = 0;

    $model_balances = Balance::oldest();
    $balances = $model_balances->whereMonth('date_received', date('m'))
->whereYear('date_received', date('Y'))->get();
    $sum_debit = $balances->where('debit_credit', 0)->sum('total_amount');
    $sum_credit = $balances->where('debit_credit', 1)->sum('total_amount');
    $total_sum = $sum_debit-$sum_credit;

    return view('backend.balance.index',
compact('month', 'year', 'balances', 'years',
'sum_debit', 'sum_credit', 'total_sum'));
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    $categories = BalanceCategory::pluck('id', 'category_name');
    return view('backend.balance.create', compact('categories'));
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $balance = Balance::create($request->all());
    return redirect()->route('balance.index')
->with('success', 'Data berhasil ditambahkan');
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\Models\Balance $balance
 * @return \Illuminate\Http\Response
 */
public function edit(Balance $balance)
{
    $balance->find($balance->id);
    $categories = BalanceCategory::pluck('id', 'category_name');
    return view('backend.balance.edit', compact('balance', 'categories'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Balance $balance
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, Balance $balance)
{
    Balance::find($balance->id)->update($request->all());
    return redirect()->route('balance.index')
->with('success', 'Data berhasil diubah');
}

/**

```

```

* Remove the specified resource from storage.
*
* @param \App\Models\Balance $balance
* @return \Illuminate\Http\Response
*/
public function destroy(Balance $balance)
{
    Balance::destroy($balance->id);
    return redirect()->route('balance.index')
        ->with('error', 'Data berhasil dihapus');
}

public function search(Request $request){
    $new_balance = new Balance;
    $years = $new_balance->getYear();

    $month = $request->month[0];
    $year = $request->year[0];
    if ($month != 0) {
        $model_balances = Balance::oldest();
        $balances = $model_balances->whereMonth('date_received', $month)
            ->whereYear('date_received', $year)->get();
        $sum_debit = $balances->where('debit_credit', 0)->sum('total_amount');
        $sum_credit = $balances->where('debit_credit', 1)->sum('total_amount');
        $total_sum = $sum_debit-$sum_credit;
        return view('backend.balance.index',
            compact('balances', 'month', 'year', 'years',
                'sum_debit', 'sum_credit', 'total_sum'));
    }

    $model_balances = Balance::oldest();
    $balances = $model_balances->whereYear('date_received', $year)->get();
    $sum_debit = $balances->where('debit_credit', 0)->sum('total_amount');
    $sum_credit = $balances->where('debit_credit', 1)->sum('total_amount');
    $total_sum = $sum_debit-$sum_credit;
    return view('backend.balance.index',
        compact('balances', 'month', 'year', 'years',
            'sum_debit', 'sum_credit', 'total_sum'));
}
}
}

```

10. SantriController

```

<?php

namespace App\Http\Controllers;

use App\Models\Santri;
use App\Models\User;
use Illuminate\Http\Request;

class SantriController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = User::find(auth()->user()->id);
        $santries = $user->Santries()->get();
        // dd($santries);
        return view('backend.santri.index', compact('santries'));
    }
    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {

```

```

    return view('backend.santri.create');
}
/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    Santri::create($request->all());
    return redirect()->route('santri.index')
        ->with('success', 'Data berhasil ditambahkan');
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Santri $santri
 * @return \Illuminate\Http\Response
 */
public function show(Santri $santri)
{
    $santri->find($santri->id);
    return view('backend.santri.show', compact('santri'));
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\Models\Santri $santri
 * @return \Illuminate\Http\Response
 */
public function edit(Santri $santri)
{
    $santri->find($santri->id);
    // dd($santri);
    return view('backend.santri.edit', compact('santri'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Santri $santri
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, Santri $santri)
{
    Santri::find($santri->id)->update($request->all());
    return redirect()->route('santri.index')
        ->with('success', 'Data berhasil diubah');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Santri $santri
 * @return \Illuminate\Http\Response
 */
public function destroy(Santri $santri)
{
    Santri::destroy($santri->id);
    return redirect()->route('santri.index')
        ->with('error', 'Data berhasil dihapus');
}
}

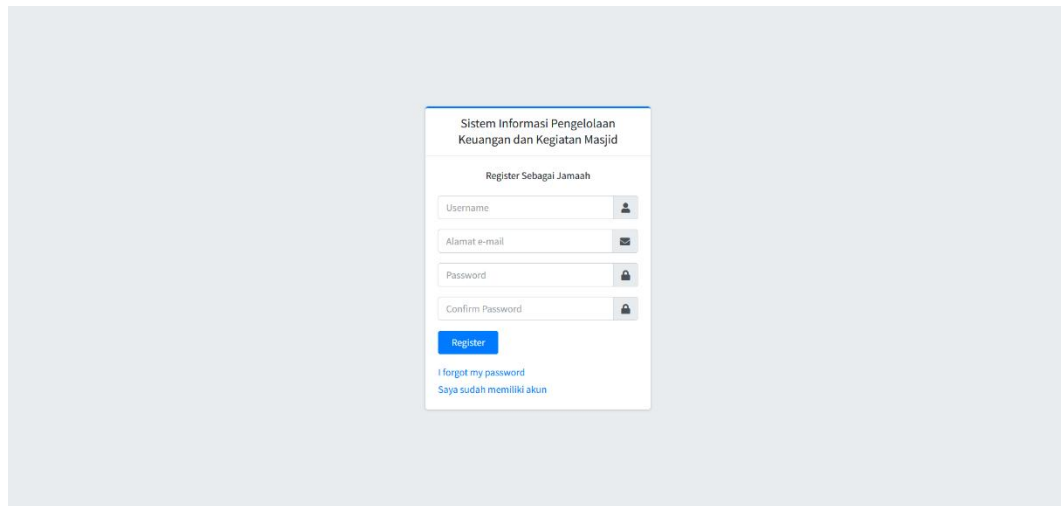
```



LAMPIRAN G

Implementasi antarmuka aplikasi

1. Register sebagai jamaah



Sistem Informasi Pengelolaan Keuangan dan Kegiatan Masjid

Register Sebagai Jamaah

Username

Alamat e-mail

Password

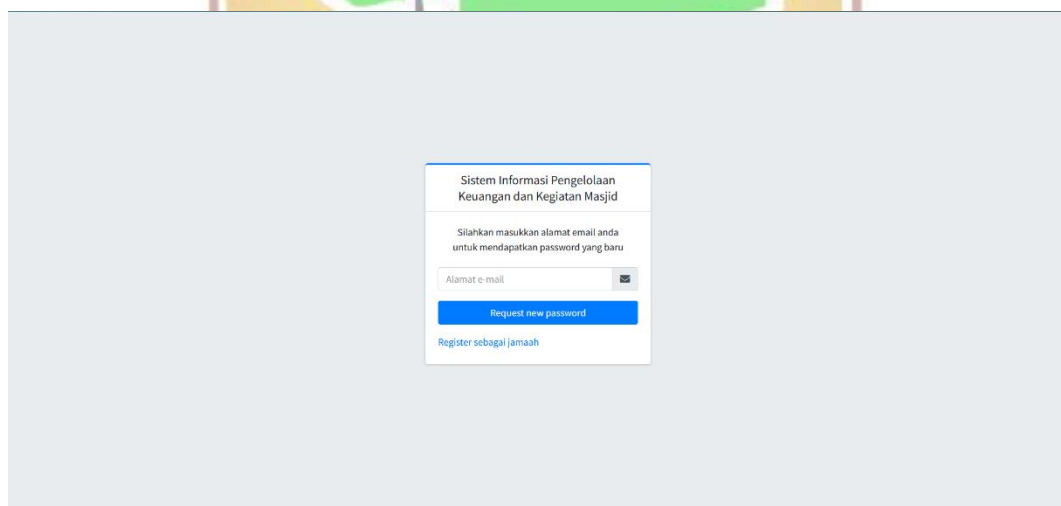
Confirm Password

[Register](#)

[I forgot my password](#)

[Saya sudah memiliki akun](#)

2. Lupa password



Sistem Informasi Pengelolaan Keuangan dan Kegiatan Masjid

Silahkan masukkan alamat email anda untuk mendapatkan password yang baru

Alamat e-mail

[Request new password](#)

[Register sebagai jamaah](#)

3. Tambah laporan keuangan

Masjid Nurul Ikhlas | Profil User | Logout

Tambah Laporan Keuangan | Lihat Laporan Keuangan / Tambah Laporan Keuangan

Tambah Data

Tanggal Diterima

Deskripsi

Nomor Invoice

Kategori
SPP

Debet/Kredit
Debet

Jumlah Dana (Rp)

Submit

4. Ubah laporan keuangan

Masjid Nurul Ikhlas | Profil User | Logout

Ubah Laporan Keuangan | Lihat Laporan Keuangan / Ubah Laporan Keuangan

Ubah Data

Tanggal Diterima
26-06-2023

Deskripsi
wakaf

Nomor Invoice
7

Kategori
WAKAF

Debet/Kredit
Kredit

Jumlah Dana (Rp)
Rp 1.975.446

Submit

5. Mengajukan aset

Masjid Nurul Ikhlas

admin

Profil User

logout

Mengajukan Aset Masjid

Mengajukan Aset

Daftar Aset

+ Ajukan Aset

Show 10 entries

Nama Aset	Kategori Aset	Jumlah Aset	Total Harga Aset	Status Aset	Opsi
Vacuum Cleaner	Perlengkapan	1	Rp 1.000.000	Disetujui	Detail Edit Delete

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Sistem Manajemen Keuangan dan Kegiatan Masjid

6. Menambah aset masjid

Masjid Nurul Ikhlas

admin

Profil User

logout

Tambah Aset Masjid

Mengajukan Aset / Tambah Aset

Tambah Data

Nama Aset

Jenis Aset

Kendaraan

Submit

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Sistem Manajemen Keuangan dan Kegiatan Masjid

7. Ubah aset masjid

Masjid Nurul Ikhlas

admin

Ubah Aset Masjid

Mengajukan Aset / Ubah Aset

Ubah Data

Nama Aset

Mobil jenazah

Jenis Aset

Kendaraan

Submit

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Sistem Manajemen Keuangan dan Kegiatan Masjid

8. Lihat detail aset masjid

Masjid Nurul Ikhlas

admin

Ases Masjid

Ases Masjid / Detail Aset

Tambah Data


Tanggal Pengambilan Aset

Kategori Aset

Harga Aset (Rp)

Foto Aset

Detail Aset

ID Aset	Tgl Pengambilan Aset	Foto Aset	Kategori Aset	Harga Aset (Rp)	Opsi
1	23 Agustus 2023		Bak	Rp 1.000.000	Edit Delete

Showing 1 to 1 of 1 entries

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Sistem Manajemen Keuangan dan Kegiatan Masjid

9. Ubah detail aset masjid

Masjid Nurul Ikhlas

admin

Ubah Detail Aset Masjid

Ubah Data

Tanggal Pengadaan Aset
26-06-2023

Kualitas Aset
Baik

Foto Aset
Pilih Foto

Copyright © 2014-2021 AdminLTE.Jo. All rights reserved. Sistem Manajemen Keuangan dan Kegiatan Masjid

10. Lihat kegiatan masjid

Masjid Nurul Ikhlas

admin

Kegiatan Masjid

Pilih Data Tahun 2023

Daftar Kegiatan

Show 5 entries

Copy CSV Excel PDF Print Column visibility

Nama Kegiatan	Kategori Kegiatan	Tanggal Mulai	Tanggal Selesai	Dana Kegiatan	Status Kegiatan	Opsi
Ceramah Wirid	Ceramah Minggu Selasa Malam	29 Agustus 2023 19:00	29 Agustus 2023 19:30	Rp 130.000	Sudah berjalan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Batalan"/>
Ceramah Wirid	Ceramah Minggu Selasa Malam	22 Agustus 2023 19:00	22 Agustus 2023 19:30	Rp 140.000	Sudah berjalan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Batalan"/>
Ceramah Wirid	Ceramah Minggu Selasa Malam	15 Agustus 2023 19:00	15 Agustus 2023 19:30	Rp 170.000	Sudah berjalan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Batalan"/>
Ceramah Wirid	Ceramah Minggu Selasa Malam	08 Agustus 2023 19:00	08 Agustus 2023 19:30	Rp 190.000	Sudah berjalan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Batalan"/>
Ceramah Wirid	Ceramah Minggu Selasa Malam	01 Agustus 2023 19:00	01 Agustus 2023 19:30	Rp 110.000	Sudah berjalan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Batalan"/>

Showing 1 to 5 of 5 entries

11. Mengajukan kegiatan masjid

The screenshot shows a dashboard for 'Mengajukan Kegiatan Masjid'. The main content is a table listing activities. The table has the following columns: Nama Kegiatan, Kategori Kegiatan, Tanggal Mulai, Tanggal Selesai, Dana Kegiatan, Status, and Opsi. The data rows are as follows:

Nama Kegiatan	Kategori Kegiatan	Tanggal Mulai	Tanggal Selesai	Dana Kegiatan	Status	Opsi
Ceramah Wihid	Ceramah-Mingguan Selasa Malam	01 September 2023 18:00	05 September 2023 19:30	Rp 50.000	Sedang diproses	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Selasa Malam	15 September 2023 19:30	19 September 2023 19:30	Rp 300.000	Sedang diproses	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Jumat Malam	20 Agustus 2023 19:00	25 Agustus 2023 19:30	Rp 220.000	Sedang diproses	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Selasa Malam	11 September 2023 19:00	12 September 2023 19:30	Rp 200.000	Sedang diproses	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Selasa Malam	08 Agustus 2023 20:00	08 Agustus 2023 19:30	Rp 190.000	Ditetujui	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Selasa Malam	15 Agustus 2023 20:00	15 Agustus 2023 19:30	Rp 170.000	Ditetujui	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Jumat Malam	03 Agustus 2023 19:00	03 Agustus 2023 19:30	Rp 150.000	Sedang diproses	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Selasa Malam	22 Agustus 2023 19:00	22 Agustus 2023 19:30	Rp 140.000	Ditetujui	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Selasa Malam	20 Agustus 2023 19:00	20 Agustus 2023 19:30	Rp 130.000	Ditetujui	Detail Edit Batal
Ceramah Wihid	Ceramah-Mingguan Jumat Malam	04 Agustus 2023 18:00	04 Agustus 2023 19:30	Rp 120.000	Sedang diproses	Detail Edit Batal

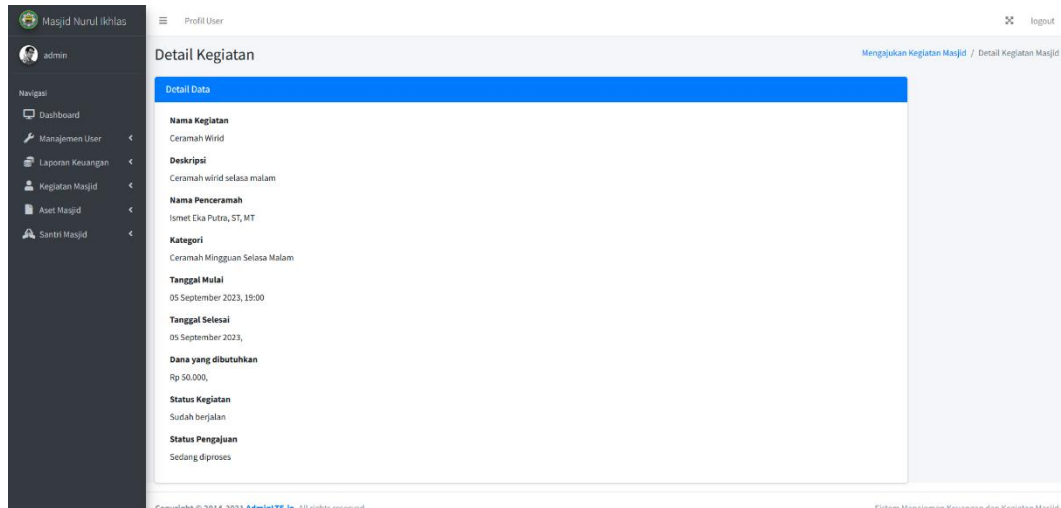
Showing 1 to 10 of 13 entries

12. Tambah kegiatan masjid

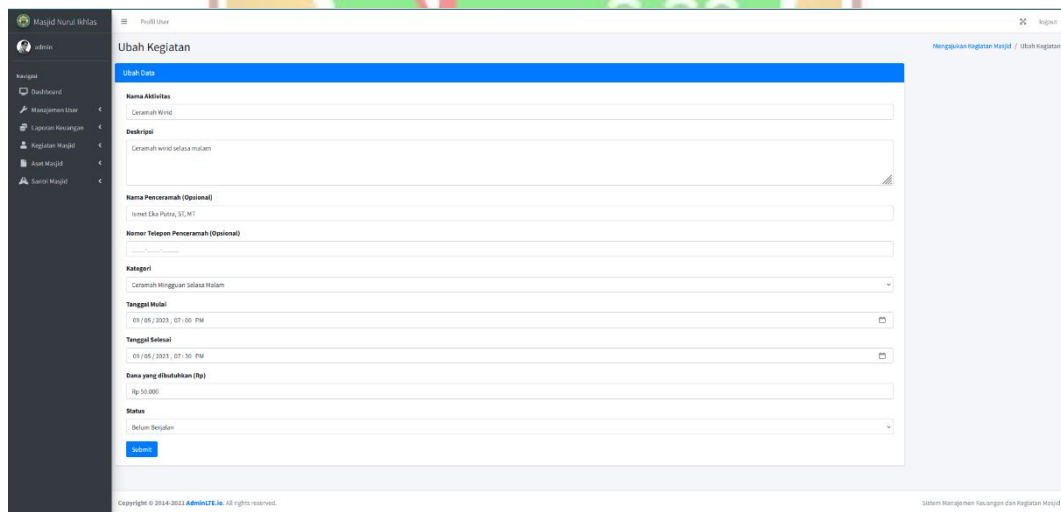
The screenshot shows a form titled 'Tambah Kegiatan'. The form fields are as follows:

- Tambah Data** (Section Header)
- Nama Aktivitas** (Text input field)
- Nama Penceramah (Optional)** (Text input field)
- Nomor Telepon Penceramah (Optional)** (Text input field)
- Deskripsi** (Text area)
- Kategori** (Dropdown menu, currently showing 'Penceramah')
- Tanggal Mulai** (Date picker)
- Tanggal Selesai** (Date picker)
- Dana yang dibutuhkan (Rp)** (Text input field)
- Submit** (Blue button)

13. Detail kegiatan masjid



14. Ubah kegiatan masjid



15. Ubah data santri

The screenshot shows the 'Ubah Santri' (Edit Santri) form. The form fields are as follows:

- Nama Santri:** Alya Elvania Chalisa
- Tempat / Tanggal Lahir:** Bukit Tinggi, mm/dd/yyyy
- Jenis Kelamin:** Perempuan
- Nomor Telepon:** 0821-8907-8855
- Alamat:** Jl. Ngarah Rai I no.8 ATT
- Nama Ayah:** Eka Chandra Kartana
- Nama Ibu:** Ann Sosyanti
- Nama Sekolah:** SDN 28
- Kelas:** 2
- Foto Santri:** Pilih Foto (Browse)
- Harga Aset (Rp):** Rp.200.000

A 'Submit' button is located at the bottom left of the form.

16. Menerima santri

The screenshot shows the 'Menerima Santri' (Receive Santri) page. It features a table with the following data:

Nama Santri	Sekolah	Kelas	Jenis Kelamin	Nama Pengaju	Biaya Pendaftaran (Rp)	Status	Opsi
<input type="checkbox"/> Alya Elvania Chalisa	SDN 28	2	Perempuan	admin	Rp.200.000	Sedang diproses	Detail

Below the table, there are buttons for 'Terima data diceklis' (Accept data checked) and 'Tolak data diceklis' (Reject data checked). A green bar at the bottom indicates 'Data yang Sudah Diproses' (Data that has been processed).



LAMPIRAN H

Pengajuan Sistem

1. Admin

Form Pengujian Sistem

Nama Aplikasi : Sistem Informasi Pengelolaan Kegiatan Dan Keuangan Masjid Nurul Ikhlas Kecamatan Padang Utara Berbasis Web

Nama Penguji : *Yengita Ezerck*

Jabatan : *Ketua Umum Masjid Nurul Ikhlas*

Pengajuan Untuk Fungsi *Admin*

No.	Nama Fungsional	Skenario Pengujian	Hasil yang Diharapkan	Sesuai	Tidak Sesuai
1.	Mengelola User				
	a. Melihat user	Memilih "Lihat User" pada menu <i>sidebar</i> aplikasi.	Membuka halaman "Lihat User" dan menampilkan <i>list user</i> aplikasi.	✓	
	b. Menambah user	Menekan tombol "Tambah Pengguna" dan mengisi form tambah <i>user</i> , lalu menekan tombol "submit".	Sistem menampilkan form tambah <i>user</i> . Lalu, sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah user	Menekan tombol "Edit" pada <i>user</i> yang ingin diubah,	Sistem menampilkan data yang telah diperbaharui.	✓	



		mengubah data <i>user</i> , dan menekan tombol "submit".			
	d. Menghapus user	Menekan tombol "Delete" pada <i>user</i> yang ingin diubah.	Sistem menghapus data user	✓	
2.	Mengelola Laporan Keuangan Masjid				
	a. Melihat laporan keuangan masjid	Memilih "Lihat Laporan Keuangan" pada menu <i>sidebar</i> aplikasi.	Membuka halaman " Laporan Keuangan" dan menampilkan <i>list</i> laporan keuangan.	✓	
	b. Menambah laporan keuangan masjid	Menekan tombol "Tambah Laporan Keuanagn" dan mengisi form, lalu menekan tombol "submit".	Sistem menampilkan form tambah laporan keuangan. Lalu, sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah laporan keuangan masjid	Menekan tombol "Edit" pada laporan keuangan yang ingin diubah, mengubah data laporan keuangan, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	d. Menghapus laporan keuangan masjid	Menekan tombol "Delete" pada laporan keuangan yang ingin diubah.	Sistem menghapus data laporan keuangan	✓	

3.	Mengelola Kategori Laporan Keuangan Masjid				
	a. Melihat kategori laporan keuangan masjid	Memilih "Lihat Kategori" pada menu <i>sidebar</i> aplikasi.	Membuka halaman " Kategori Laporan Keuangan" dan menampilkan <i>list</i> kategori laporan keuangan.	✓	
	b. Menambah kategori laporan keuangan masjid	Menekan tombol "Tambah Kategori" dan mengisi form, lalu menekan tombol "submit".	Sistem menampilkan form tambah kategori laporan keuangan. Lalu, sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah kategori laporan keuangan masjid	Menekan tombol "Edit" pada kategori laporan keuangan yang ingin diubah, mengubah data kategori, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	d. Menghapus kategori laporan keuangan masjid	Menekan tombol "Delete" pada laporan keuangan yang ingin diubah.	Sistem menghapus data kategori laporan keuangan	✓	
4.	Mengelola Kegiatan Masjid				
	a. Melihat kegiatan masjid	Memilih "Lihat Kegiatan" pada menu <i>sidebar</i> aplikasi.	Membuka halaman "Lihat Kegiatan" dan menampilkan <i>list</i>	✓	


			kegiatan masjid yang telah disetujui.		
	b. Mengubah kegiatan masjid	Menekan tombol "Edit" pada kegiatan yang ingin diubah, mengubah data kegiatan, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	c. Menghapus kegiatan masjid	Menekan tombol "Delete" pada kegiatan yang ingin diubah.	Sistem menghapus data kegiatan	✓	
	d. Menyetujui kegiatan masjid	Memilih "Menyetujui Kegiatan" pada menu <i>sidebar</i> . Menceklis data yang ingin disetujui, lalu menekan tombol "Terima data diceklis".	Membuka halaman "Menyetujui Kegiatan" dan menampilkan <i>list</i> kegiatan. Sistem mengubah status data menjadi "disetujui".	✓	
5.	Mengelola Kategori Kegiatan Masjid				
	a. Melihat kategori kegiatan masjid	Memilih "Lihat Kategori Kegiatan" pada menu <i>sidebar</i> aplikasi.	Membuka halaman "Kategori Kegiatan" dan menampilkan <i>list</i> kategori kegiatan.	✓	
	b. Menambah kategori kegiatan masjid	Menekan tombol "Tambah Kategori" dan mengisi form, lalu menekan tombol "submit".	Sistem menampilkan form tambah kategori kegiatan. Lalu, sistem	✓	

			menampilkan data yang telah diinputkan.		
	c. Mengubah kategori kegiatan masjid	Menekan tombol "Edit" pada kategori kegiatan yang ingin diubah, mengubah data kategori, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	d. Menghapus kategori kegiatan masjid	Menekan tombol "Delete" pada kegiatan yang ingin dihapus.	Sistem menghapus data kategori kegiatan.	✓	
6.	Mengelola Aset Masjid				
	a. Melihat aset masjid	Memilih "Lihat Aset" pada menu <i>sidebar</i> aplikasi.	Membuka halaman "Lihat Aset" dan menampilkan <i>list</i> kegiatan masjid yang telah disetujui.	✓	
	b. Mengubah aset masjid	Menekan tombol "Edit" pada aset yang ingin diubah, mengubah data aset, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	c. Menghapus aset masjid	Menekan tombol "Delete" pada aset yang ingin diubah.	Sistem menghapus data aset	✓	
	d. Menyetujui aset masjid	Memilih "Menyetujui Aset" pada menu <i>sidebar</i> . Menceklis data yang ingin disetujui, lalu	Membuka halaman "Menyetujui Aset" dan menampilkan <i>list</i> aset.	✓	

		menekan tombol "Terima data diceklis".	Sistem mengubah status data menjadi "disetujui".		
7.	Mengelola Kategori Aset Masjid				
	a. Melihat kategori aset masjid	Memilih "Lihat Kategori Aset" pada menu <i>sidebar</i> aplikasi.	Membuka halaman " Kategori Aset" dan menampilkan <i>list</i> kategori aset.	✓	
	b. Menambah kategori aset masjid	Menekan tombol "Tambah Kategori" dan mengisi form, lalu menekan tombol "submit".	Sistem menampilkan form tambah kategori kegiatan. Lalu, sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah kategori aset masjid	Menekan tombol "Edit" pada kategori kegiatan yang ingin diubah, mengubah data kategori, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	d. Menghapus kategori aset masjid	Menekan tombol "Delete" pada kategori aset yang ingin dihapus.	Sistem menghapus data kategori aset.	✓	
8.	Mengelola Penerimaan Santri				
	a. Melihat santri	Memilih "Lihat Santri" pada menu <i>sidebar</i> aplikasi.	Membuka halaman "Lihat Santri" dan menampilkan <i>list</i> santri.	✓	

	b. Menerima santri	Memilih "Menerima Santri" pada menu <i>sidebar</i> . Menceklis data yang ingin disetujui, lalu menekan tombol "Terima data diceklis".	Membuka halaman "Menerima Santri" dan menampilkan <i>list</i> kegiatan. Sistem mengubah status data menjadi "disetujui".	✓	
--	--------------------	---	--	---	--

Padang, 21 Juni 2023
Penguji



2. Pengurus Masjid

Form Pengujian Sistem

Nama Aplikasi : Sistem Informasi Pengelolaan Kegiatan Dan Keuangan Masjid Nurul Ikhlas Kecamatan Padang Utara Berbasis Web

Nama Penguji : Yemrita Ependi

Jabatan : Ketua Umum Masjid Nurul Ikhlas

Pengajuan Untuk Fungsi Pengurus Masjid

No.	Nama Fungsional	Skenario Pengujian	Hasil yang Diharapkan	Sesuai	Tidak Sesuai
1.	Mengelola Pengajuan Kegiatan Masjid				
	a. Melihat kegiatan yang diajukan	Memilih "Mengajukan Kegiatan" pada menu <i>sidebar</i> aplikasi.	Membuka halaman " Mengajukan Kegiatan" dan menampilkan <i>list</i> kegiatan yang diajukan.	✓	
	b. Mengajukan kegiatan	Menekan tombol "Ajukan Kegiatan" dan mengisi form pengajuan kegiatan, lalu menekan tombol "submit".	Sistem menampilkan form pengajuan kegiatan. Lalu, sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah kegiatan yang diajukan	Menekan tombol "Edit" pada data yang ingin diubah,	Sistem menampilkan data yang telah diperbaharui.	✓	

		mengubah data <i>user</i> , dan menekan tombol "submit".			
	d. Menghapus kegiatan yang diajukan	Menekan tombol "Delete" pada data yang ingin diubah.	Sistem menghapus data kegiatan yang diajukan.	✓	
2.	Mengelola Pengajuan Aset Masjid				
	a. Melihat aset yang diajukan	Memilih "Mengajukan Aset" pada menu <i>sidebar</i> aplikasi.	Membuka halaman " Mengajukan Aset" dan menampilkan <i>list</i> aset yang diajukan.	✓	
	b. Mengajukan aset	Menekan tombol "Ajukan Aset" dan mengisi form pengajuan aset, lalu menekan tombol "submit".	Sistem menampilkan form pengajuan aset. Lalu, sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah aset yang diajukan	Menekan tombol "Edit" pada data yang ingin diubah, mengubah data <i>user</i> , dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	d. Menghapus aset yang diajukan	Menekan tombol "Delete" pada data yang ingin diubah.	Sistem menghapus data aset yang diajukan.	✓	
3.	Mengelola Detail Aset Masjid				

a. Melihat kategori aset masjid	Menekan tombol "Detail" pada aset yang diinginkan	Membuka halaman "Detail Aset" dan menampilkan <i>list</i> detail aset.	✓	
b. Menambah detail aset masjid	menentukan kualitas aset dan meunggah foto aset jika ada, lalu menekan tombol "submit".	Sistem menampilkan data yang telah diinputkan.	✓	
c. Mengubah detail aset masjid	Menekan tombol "Edit" pada detail aset yang ingin diubah, mengubah data detail, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
d. Menghapus detail aset masjid	Menekan tombol "Delete" pada detail aset yang ingin dihapus.	Sistem menghapus data detail aset.	✓	



Padang, 11 Juni 2023

Penguji

3. Jamaah



Form Pengujian Sistem

Nama Aplikasi : Sistem Informasi Pengelolaan Kegiatan Dan Keuangan Masjid Nurul Ikhlas Kecamatan Padang Utara Berbasis Web

Nama Penguji : Yempto Ependi

Jabatan : Ketua Umum Masjid Nurul Ikhlas

Pengajuan Untuk Fungsi Jamaah

No.	Nama Fungsional	Skenario Pengujian	Hasil yang Diharapkan	Sesuai	Tidak Sesuai
1.	Mendaftar Sebagai Jamaah	Memilih Opsi "Register Sebagai Jamaah" pada halaman <i>login</i> dan mengisi form registrasi, lalu menekan tombol "submit"	Sistem menambahkan user sebagai jamaah kedalam sistem. Lalu sistem menampilkan halaman <i>dashboard</i>	✓	
2.	Mengelola Pendaftaran Santri				
	a. Melihat santri yang diajukan	Memilih "Mengajukan Santri" pada menu <i>sidebar</i> aplikasi.	Membuka halaman " Mengajukan Santri" dan menampilkan <i>list</i> santri yang diajukan.	✓	
	b. Mengajukan santri	Menekan tombol "Ajukan Santri" dan mengisi form	Sistem menampilkan form pengajuan santri. Lalu, sistem	✓	

		pengajuan santri, lalu menekan tombol "submit".	menampilkan data yang telah diinputkan.		
	c. Mengubah santri yang diajukan	Menekan tombol "Edit" pada data yang ingin diubah, mengubah data <i>user</i> , dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	d. Menghapus santri yang diajukan	Menekan tombol "Delete" pada data yang ingin diubah.	Sistem menghapus data santri yang diajukan.	✓	
3.	Mengelola Pengajuan Aset Masjid				
	a. Melihat aset yang diajukan	Memilih "Mengajukan Aset" pada menu <i>sidebar</i> aplikasi.	Membuka halaman " Mengajukan Aset" dan menampilkan <i>list</i> aset yang diajukan.	✓	
	b. Mengajukan aset	Menekan tombol "Ajukan Aset" dan mengisi form pengajuan aset, lalu menekan tombol "submit".	Sistem menampilkan form pengajuan aset. Lalu, sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah aset yang diajukan	Menekan tombol "Edit" pada data yang ingin diubah, mengubah data <i>user</i> , dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	

	d. Menghapus aset yang diajukan	Menekan tombol "Delete" pada data yang ingin diubah.	Sistem menghapus data aset yang diajukan.	✓	
4.	Mengelola Detail Aset Masjid				
	a. Melihat kategori aset masjid	Menekan tombol "Detail" pada aset yang diinginkan	Membuka halaman "Detail Aset" dan menampilkan <i>list</i> detail aset.	✓	
	b. Menambah detail aset masjid	menentukan kualitas aset dan meunggah foto aset jika ada, lalu menekan tombol "submit".	Sistem menampilkan data yang telah diinputkan.	✓	
	c. Mengubah detail aset masjid	Menekan tombol "Edit" pada detail aset yang ingin diubah, mengubah data detail, dan menekan tombol "submit".	Sistem menampilkan data yang telah diperbaharui.	✓	
	d. Menghapus detail aset masjid	Menekan tombol "Delete" pada detail aset yang ingin dihapus.	Sistem menghapus data detail aset.	✓	

Padang, 21 Juni 2023
 Penguji

