

LAPORAN PRAKTIK KERJA LAPANGAN

PENGEMBANGAN SISTEM INFORMASI SURAT MASUK DAN KELUAR DI KEMENTERIAN AGAMA KOTA MALANG

Diajukan Untuk Memenuhi Sebagian Persyaratan Kurikulum



Disusun Oleh:

Achmad Syarifudin	135150200111065
Faris Febrianto	135150201111221
Sigit Pangestu	135150201111188

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016**

LEMBAR PERSETUJUAN
LAPORAN PRAKTIK KERJA LAPANGAN

Diajukan Untuk Memenuhi Sebagian Persyaratan Kurikulum
Program Studi Teknik Informatika
Jurusan Teknik Informatika
Universitas Brawijaya

**PENGEMBANGAN SISTEM INFORMASI SURAT MASUK DAN KELUAR
DI KEMENTERIAN AGAMA KOTA MALANG**

Dilaksanakan di:

KEMENTERIAN AGAMA KOTA MALANG

Tanggal:

22 Agustus 2016 – 21 Oktober 2016

Disusun oleh:

Achmad Syarifudin	135150200111065
Faris Febrianto	135150201111221
Sigit Pangestu	135150201111188

Mengetahui,
Ketua Jurusan Teknik Informatika

Menyetujui,
Dosen Pembimbing

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

Agus Wahyu Widodo, S.T., M.Cs
NIP. 19740805 200112 1 001

LEMBAR PERSETUJUAN
LAPORAN PRAKTIK KERJA LAPANGAN

Diajukan Untuk Memenuhi Sebagian Persyaratan Kurikulum
Program Studi Teknik Informatika
Jurusan Teknik Informatika
Universitas Brawijaya

**PENGEMBANGAN SISTEM INFORMASI SURAT MASUK DAN KELUAR
DI KEMENTERIAN AGAMA KOTA MALANG**

Dilaksanakan di:

KEMENTERIAN AGAMA KOTA MALANG

Tanggal:

22 Agustus 2016 – 21 Oktober 2016


Disusun oleh:

Achmad Syarifudin	135150200111065
Faris Febrianto	135150201111221
Sigit Pangestu	135150201111188

Kepala Sub Bagian Tata Usaha


Muhajir, S.Pd., M.Ag
NIP. 19730413 199903 1 003

Pembimbing PKL


Jainul Mustofah
NIP. 19780704 200901 1 010

KATA PENGANTAR

Puji Syukur kami panjatkan kehadiran Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan hidayah-Nya, sehingga kami telah berhasil menyelesaikan laporan hasil Praktik Kerja Lapangan di Kementerian Agama Kota Malang.

Kami mengangkat tema Pengembangan “SIMAK” (Sistem Informasi Surat Masuk Dan Keluar) karena kami melihat adanya ketidakefisienan dan efektifitasnya waktu yang digunakan dalam proses pengarsipan surat masuk maupun surat keluar pada instansi Kementerian Agama Kota Malang.

Kami mengucapkan terima kasih kepada:

1. Kedua orang tua kami, yang telah memberikan dukungan dan doa.
2. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer, Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T., MT., Ph.D selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya sekaligus dosen pembimbing Praktik Kerja Lapangan kami.
5. Bapak Jainul Mustofah selaku pembimbing di Kementerian Agama Kota Malang.
6. Seluruh pegawai Kementerian Agama Kota Malang yang telah menginspirasi dan membantu kami dalam kegiatan kerja praktik ini.

Kami menyadari bahwa laporan ini masih memiliki banyak kekurangan, untuk itu saran dan kritik yang membangun sangat diharapkan demi perbaikan laporan selanjutnya. Kami berharap laporan ini dapat bermanfaat bagi penulis sendiri dan para pembaca pada umumnya.

Malang, 10 Oktober 2016

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PERSETUJUAN	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR TABEL	vii
DAFTAR GAMBAR	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan masalah	2
1.3. Tujuan	2
1.3.1. Tujuan Umum	2
1.3.2. Tujuan Khusus	2
1.4. Manfaat	3
1.5. Batasan Masalah	3
1.6. Sistematika Pembahasan	4
1.7. Pelaksanaan Praktik Kerja Lapangan	4
BAB II LANDASAN KEPUSTAKAAN	5
2.1. Profil Instansi	5
2.1.1. Sejarah Kementerian Agama	5
2.1.2. Visi dan Misi	6
2.2. Sistem Informasi	6
2.3. Surat Masuk dan Surat Keluar	7
2.3.1. SOP Administrasi Surat Masuk dan Keluar	7
2.3.2. SOP Penomoran Surat	9
2.4. Basis Data	10
2.5. Pemodelan Berorientasi Objek	11
2.5.1. Unified Modelling Language (UML)	11
2.5.2. Use Case Diagram	12
2.5.3. Sequence Diagram	12

2.5.4. Class Diagram	13
2.6. PHP.....	14
2.7. CodeIgniter.....	14
2.8. Black Box Testing	15
BAB III METODOLOGI PENELITIAN	16
3.1. Studi Pustaka.....	16
3.2. Analisis Kebutuhan.....	17
3.3. Perancangan Sistem.....	18
3.4. Implementasi	19
3.5. Pengujian	19
3.6. Kesimpulan.....	19
BAB IV ANALISIS DAN PERANCANGAN.....	20
4.1. Analisis Kebutuhan.....	20
4.1.1. Gambaran Umum Sistem	20
4.1.2. Identifikasi Aktor	21
4.1.3. Daftar Kebutuhan Sistem	22
4.2. Pemodelan Kebutuhan	25
4.2.1. Use Case	25
4.2.2. Use Case Scenario.....	25
4.2.3. Sequence Diagram.....	37
4.2.4. Class Diagram	57
4.3. Perancangan Sistem.....	58
4.3.1. Perancangan Komponen	58
4.3.2. Perancangan Data	63
4.3.3. Perancangan Desain Antarmuka	63
BAB V IMPLEMENTASI	76
5.1. Batasan Implementasi	76
5.2. Implementasi Komponen.....	76
5.2.1. Klasifikasi Surat.....	76
5.2.2. Kelola Surat Masuk	78
5.2.3. Kelola Surat Keluar	80

5.2.4. Disposisi Surat	82
5.2.5. Cetak Agenda.....	84
5.2.6. Cetak Disposisi	84
5.3. Implementasi Data	85
5.4. Implementasi Antarmuka	88
5.4.1. Halaman Login	88
5.4.2. Halaman Home	89
5.4.3. Halaman Manage User	90
5.4.4. Halaman Setting Instansi	91
5.4.5. Halaman Ganti Password	91
5.4.6. Halaman Kelola Surat Masuk.....	92
5.4.7. Halaman Disposisi Surat	94
5.4.8. Halaman Cetak Disposisi	94
5.4.9. Halaman Kelola Surat Keluar	95
5.4.10. Halaman Klasifikasi Surat	96
5.4.11. Halaman Cetak Agenda Surat	96
BAB VI PENGUJIAN	98
6.1. Perangkat Lunak Pengujian.....	98
6.2. Perangkat Keras Pengujian	98
6.3. Pengujian Validasi	99
6.3.1. Pengujian Validasi Kebutuhan Fungsional.....	99
6.3.2. Pengujian Validasi Kebutuhan Non Fungsional.....	104
BAB VII PENUTUP.....	105
7.1. Kesimpulan.....	105
7.2. Saran	106
DAFTAR PUSTAKA.....	107

DAFTAR TABEL

Tabel 2.1 Tahapan Administrasi Surat Masuk dan Keluar	7
Tabel 2.2 Tahapan Penomoran Surat.....	9
Tabel 2.3 Elemen-elemen Use Case	12
Tabel 4.1 Identifikasi Aktor	21
Tabel 4.2 Kebutuhan Fungsional	22
Tabel 4.3 Kebutuhan Non Fungsional	24
Tabel 4.4 Use Case Scenario Login	25
Tabel 4.5 Use Case Scenario Logout	26
Tabel 4.6 Use Case Scenario Ganti Password	26
Tabel 4.7 Use Case Scenario Klasifikasi Surat	27
Tabel 4.8 Use Case Scenario Cetak Agenda Surat Masuk	27
Tabel 4.9 Use Case Scenario Cetak Agenda Surat Keluar	28
Tabel 4.10 Use Case Scenario Kelola Surat Masuk	28
Tabel 4.11 Use Case Scenario Cari Surat Masuk	29
Tabel 4.12 Use Case Scenario Tambah Surat Masuk	29
Tabel 4.13 Use Case Scenario Edit Surat Masuk	30
Tabel 4.14 Use Case Scenario Hapus Surat Masuk	30
Tabel 4.15 Use Case Scenario Disposisi Surat Masuk	31
Tabel 4.16 Use Case Scenario Cetak Disposisi Surat.....	31
Tabel 4.17 Use Case Scenario Kelola Surat Keluar	31
Tabel 4.18 Use Case Scenario Cari Surat Keluar	32
Tabel 4.19 Use Case Scenario Tambah Surat Keluar.....	32
Tabel 4.20 Use Case Scenario Edit Surat Keluar	33
Tabel 4.21 Use Case Scenario Hapus Surat Keluar.....	33
Tabel 4.22 Use Case Scenario Setting Instansi.....	34
Tabel 4.23 Use Case Scenario Manage User	34
Tabel 4.24 Use Case Scenario Tambah User	35
Tabel 4.25 Use Case Scenario Edit User	35
Tabel 4.26 Use Case Scenario Hapus User	36

Tabel 4.27 Use Case Scenario Cari User	36
Tabel 4.28 Keterangan Rancangan Halaman Login.....	64
Tabel 4.29 Keterangan Rancangan Halaman Home Admin	65
Tabel 4.30 Keterangan Rancangan Halaman Home User	66
Tabel 4.31 Keterangan Rancangan Halaman Manage User.....	66
Tabel 4.32 Keterangan Rancangan Halaman Manage User – Tambah data	67
Tabel 4.33 Keterangan Rancangan Halaman <i>Setting</i> Instansi	68
Tabel 4.34 Keterangan Rancangan Halaman Ganti Password	69
Tabel 4.35 Keterangan Rancangan Halaman Kelola Surat Masuk	69
Tabel 4.36 Rancangan Halaman Tambah Surat Masuk	70
Tabel 4.37 Keterangan Rancangan Halaman Disposisi Surat.....	71
Tabel 4.38 Keterangan Rancangan Halaman Kelola Surat Keluar.....	72
Tabel 4.39 Keterangan Rancangan Halaman Tambah Data Surat Keluar	73
Tabel 4.40 Keterangan Rancangan Halaman Klasifikasi Surat	74
Tabel 4.41 Keterangan Rancangan Halaman Cetak Agenda Surat	75
Tabel 5.1 <i>Source Code</i> Klasifikasi Surat	77
Tabel 5.2 <i>Source Code</i> Kelola Surat Masuk	78
Tabel 5.3 <i>Source Code</i> Kelola Surat Keluar	80
Tabel 5.4 <i>Source Code</i> Disposisi Surat.....	82
Tabel 5.5 <i>Source Code</i> Cetak Agenda	84
Tabel 5.6 <i>Source Kode</i> Cetak Disposisi	84
Tabel 6.1 Perangkat Lunak Pengujian	98
Tabel 6.2 Perangkat Keras Pengujian	98
Tabel 6.3 Pengujian Validasi Kebutuhan Fungsional	99
Tabel 6.4 Pengujian Validasi Kebutuhan Non Fungsional.....	104

DAFTAR GAMBAR

Gambar 2.1 <i>Flowchart</i> Administrasi Surat Masuk dan Keluar	8
Gambar 2.2 <i>Flowchart</i> SOP Penomoran Surat Surat	10
Gambar 2.3 Diagram UML.....	11
Gambar 2.4 Contoh Sequence Diagram.....	13
Gambar 2.5 Arsitektur <i>Framework</i> CodeIgniter	15
Gambar 3.1 Diagram Alir Penelitian.....	16
Gambar 4.1 Use Case Diagram SIMAK	25
Gambar 4.2 Sequence Diagram Login	37
Gambar 4.3 Sequence Diagram Logout	37
Gambar 4.4 Sequence Diagram Ganti Password	38
Gambar 4.5 Sequence Diagram Klasifikasi Surat	39
Gambar 4.6 Sequence Diagram Cetak Agenda Surat Masuk	40
Gambar 4.7 Sequence Diagram Cetak Agenda Surat Keluar	41
Gambar 4.8 Sequence Diagram Kelola Surat Masuk	42
Gambar 4.9 Sequence Diagram Cari Surat Masuk	42
Gambar 4.10 Sequence Diagram Tambah Surat Masuk	43
Gambar 4.11 Sequence Diagram Edit Surat Masuk	44
Gambar 4.12 Sequence Diagram Hapus Surat Masuk	45
Gambar 4.13 Sequence Diagram Disposisi Surat Masuk	46
Gambar 4.14 Cetak Disposisi Surat	47
Gambar 4.15 Sequence Diagram Kelola Surat Keluar	48
Gambar 4.16 Cari Surat Keluar.....	49
Gambar 4.17 Tambah Surat Keluar	50
Gambar 4.18 Edit Surat Keluar	51
Gambar 4.19 Hapus Surat Keluar	52
Gambar 4.20 Setting Instansi	52
Gambar 4.21 Manage User	53
Gambar 4.22 Tambah User	54
Gambar 4.23 Edit User	55

Gambar 4.24 Hapus User	56
Gambar 4.25 Cari User	56
Gambar 4.26 Class Diagram SIMAK.....	57
Gambar 4.27 <i>Conceptual Data Model</i> SIMAK	63
Gambar 4.28 Rancangan Halaman Login	64
Gambar 4.29 Rancangan Halaman Home Admin	64
Gambar 4.30 Rancangan Halaman Home User.....	65
Gambar 4.31 Rancangan Halaman Manage User	66
Gambar 4.32 Rancangan Halaman Manage User – Tambah data	67
Gambar 4.33 Rancangan Halaman <i>Setting</i> Instansi	68
Gambar 4.34 Rancangan Halaman Ganti Password	68
Gambar 4.35 Rancangan Halaman Kelola Surat Masuk.....	69
Gambar 4.36 Rancangan Halaman Tambah Surat Masuk	70
Gambar 4.37 Rancangan Halaman Disposisi Surat	71
Gambar 4.38 Rancangan Halaman Cetak Disposisi	71
Gambar 4.39 Rancangan Halaman Kelola Surat Keluar	72
Gambar 4.40 Rancangan Halaman Tambah Data Surat Keluar	73
Gambar 4.41 Rancangan Halaman Klasifikasi Surat	74
Gambar 4.42 Rancangan Halaman Cetak Agenda Surat.....	74
Gambar 4.43 Rancangan Hasil Mencetak Agenda Surat	75
Gambar 5.1 <i>Physical Data Model</i> SIMAK	85
Gambar 5.2 Halaman Login.....	89
Gambar 5.3 Halaman Home untuk Admin	89
Gambar 5.4 Halaman Home untuk User	90
Gambar 5.5 Halaman Manage User	90
Gambar 5.6 Halaman Input Data User	91
Gambar 5.7 Halaman Setting Instansi.....	91
Gambar 5.8 Halaman Ganti Password	92
Gambar 5.9 Halaman Kelola Surat Masuk	92
Gambar 5.10 Halaman Form Surat Masuk.....	93

Gambar 5.11 Halaman Disposisi Surat.....	94
Gambar 5.12 Halaman Cetak Disposisi	94
Gambar 5.13 Halaman Kelola Surat Keluar	95
Gambar 5.14 Halaman Form Surat Keluar	95
Gambar 5.15 Halaman Klasifikasi Surat	96
Gambar 5.16 Halaman Pilih Rentang Waktu Agenda Surat.....	97
Gambar 5.17 Halaman Agenda yang Dicetak	97

BAB I

PENDAHULUAN

Praktik Kerja Lapangan merupakan salah satu mata kuliah di Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya, sebagai sarana untuk pelatihan, pengembangan, dan penerapan ilmu pengetahuan yang diperoleh saat kuliah. Selain itu dengan Praktik Kerja Lapangan mahasiswa akan memperoleh gambaran yang jelas tentang berbagai hal yang berkaitan dengan berbagai masalah, khususnya masalah pengaturan sistem informasi dan aplikasi pada saat Praktik Kerja Lapangan.

1.1. Latar Belakang

Kementerian Agama Kota Malang memiliki fungsi untuk menjalankan tugas dalam mencapai tujuan yang sudah ditetapkan. Salah satu fungsinya adalah pelaksanaan hubungan dengan pemerintah daerah, instansi terkait, dan lembaga masyarakat dalam rangka pelaksanaan tugas Kementerian Agama di Kota Malang dengan melalui media persuratan. Dengan media persuratan, baik secara langsung maupun melalui tidak langsung (media elektronik), dapat memberikan segala informasi yang dibutuhkan oleh Kementerian Agama Kota Malang.

Sub Bagian Tata Usaha, Divisi Umum Kementerian Agama Kota Malang merupakan salah satu divisi yang mengelola segala keperluan dokumen-dokumen penting yang bersangkutan dengan Kementerian Agama Kota Malang. Dokumen-dokumen tersebut harus disimpan dalam jangka waktu tertentu, untuk menghindari kerusakan dokumen dalam bentuk *hardcopy*, diperlukan sebuah sistem yang berfungsi untuk menyimpan dokumen dalam bentuk *softcopy*. Salah satu sistem yang diperlukan adalah sistem informasi surat masuk dan keluar, sistem informasi ini berfungsi untuk menyimpan sekaligus menata segala jenis surat yang masuk maupun keluar dari dan ke Kementerian Agama Kota Malang.

1.2. Rumusan masalah

1. Bagaimana analisis kebutuhan pengguna untuk merancang sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang?
2. Bagaimana rancangan sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang berdasarkan analisis kebutuhan yang telah dilakukan?
3. Bagaimana implementasi sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang dari hasil rancangan yang telah dibuat?
4. Bagaimana hasil pengujian sistem informasi surat masuk dan keluar dengan kesesuaian kebutuhan pengguna?

1.3. Tujuan

Tujuan pengembangan laporan ini terdiri dari tujuan umum dan tujuan khusus, yaitu sebagai berikut:

1.3.1. Tujuan Umum

Mengembangkan sistem informasi surat masuk dan keluar bagi Kementerian Agama Kota Malang khususnya Sub Bagian Tata Usaha, Divisi Umum.

1.3.2. Tujuan Khusus

Secara khusus, Praktik Kerja Lapangan ini bertujuan untuk:

1. Melakukan analisis kebutuhan fungsional dan non-fungsional dari sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang.
2. Melakukan perancangan sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang berdasarkan kebutuhan yang telah ditentukan.
3. Melakukan implementasi sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang berdasarkan hasil rancangan yang telah dibuat.
4. Melakukan pengujian terhadap sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang menggunakan *black box testing*.

1.4. Manfaat

Hasil dari pengembangan sistem informasi ini diharapkan dapat memberikan manfaat yang dapat diambil yaitu sebagai berikut:

1. Memperoleh hasil berupa sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang yang dapat digunakan untuk manajemen surat pada Sub Bagian Tata Usaha, Divisi Umum.
2. Membantu untuk menemukan surat baik surat masuk maupun surat keluar pada tanggal tertentu atau sesuai kata kunci yang diinginkan.
3. Memperoleh informasi spesifik mengenai suatu surat masuk atau surat keluar yang pernah diinputkan melalui peramban web desktop.
4. Meminimalisir resiko kehilangan data surat masuk dan surat keluar karena secara tidak langsung data surat masuk dan keluar juga disalin pada sistem informasi.

1.5. Batasan Masalah

Agar memperoleh hasil pembahasan yang sesuai, maka perlu diberikan pembatasan masalah pada pengembangan sistem informasi ini, yaitu:

1. Pengembangan difokuskan pada proses rekayasa perangkat lunak (meliputi analisis dan perancangan, implementasi, dan pengujian) terhadap pengembangan sistem informasi surat masuk dan keluar Kementerian Agama Kota Malang.
2. Informasi yang disediakan oleh sistem informasi dalam ruang lingkup Kementerian Agama Kota Malang Sub Bagian Tata Usaha, Divisi Umum.
3. Data surat masuk yang dipilih hanya dari instansi luar yang masuk ke Sub Bagian Tata Usaha, Divisi Umum, Kementerian Agama Kota Malang.
4. Data surat keluar yang dipilih hanya meliputi sub-sub bagian kerja yang ada di Kementerian Agama Kota Malang setelah mendapatkan persetujuan dari Sub Bagian Tata Usaha, Divisi Umum Kementerian Agama Kota Malang.
5. Pengembangan sistem informasi dibuat dengan menggunakan *framework* CodeIgniter berdasarkan *standard operating procedure* (SOP) surat masuk dan keluar Kementerian Agama Kota Malang.

1.6. Sistematika Pembahasan

Sistematika penulisan laporan merupakan gambaran dan uraian dari laporan Praktik Kerja Lapangan secara garis besar meliputi beberapa bab, sebagai berikut:

1. Bab I Pendahuluan

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

2. Bab II Landasan Kepustakaan

Bab ini membahas tentang sejarah, visi, dan misi Kementerian Agama Kota Malang serta menguraikan kajian pustaka dan referensi yang mendasari perancangan sistem.

3. Bab III Metodologi Penelitian

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam proses perancangan dan implementasi pada pelaksanaan Praktik Kerja Lapangan.

4. Bab IV Analisis dan Perancangan

Menguraikan analisis kebutuhan serta perancangan sistem yang menjadi objek studi kasus Praktik Kerja Lapangan.

5. Bab V Implementasi

Menguraikan proses implementasi dari dasar teori yang telah dipelajari sesuai analisis dan perancangan sistem

6. Bab VI Pengujian

Menguraikan proses pengujian hasil implementasi menggunakan *black box testing*.

7. Bab VII Penutup

Menguraikan kesimpulan yang diperoleh dari pengembangan dan pengujian sistem informasi yang dikembangkan dalam Praktik Kerja Lapangan ini serta saran-saran untuk pengembangan lebih lanjut.

1.7. Pelaksanaan Praktik Kerja Lapangan

Praktik Kerja Lapangan (PKL) dilaksanakan mulai tanggal 22 Agustus 2016 sampai 21 Oktober 2016. PKL dilaksanakan di Kementerian Agama Kota Malang, Jl. Panji Suroso No. 2, Purwodadi, Blimbing, Malang.

BAB II

LANDASAN KEPUSTAKAAN

2.1. Profil Instansi

2.1.1. Sejarah Kementerian Agama

Departemen Agama dibentuk berdasarkan usulan dari Anggota Komite Nasional Indonesia Pusat, agar urusan agama tidak ditangani secara sambilan maka perlu dibentuk Departemen Agama, sekitar 5 bulan kemudian tepatnya 3 Januari 1946 lahirlah Departemen Agama dengan penetapan Pemerintah Nomor 1/SD Tahun 1946. Berdasarkan penetapan menteri Agama No. 6 tahun 1956, tanggal 1 Maret 1956 maka tanggal 3 Januari 1946 ditetapkan sebagai hari berdirinya Departemen Agama RI. Pada peringatan ulang tahun Departemen Agama ke-34, tanggal 3 Januari 1980, peringatan tersebut diubah sebutannya menjadi “Hari Amal Bhakti Departemen Agama” disingkat “HAB Depag” dengan motto “Ikhlās Beramal”.

Sejarah awal sebelum terbentuknya nama Departemen Agama Kota Malang menurut KMA nomor 6 tahun 1977 yang ditindak lanjuti dengan KMA nomor 45 tahun 1981 tentang Susunan Organisasi dan Tata Kerja Kantor Wilayah Departemen Agama, Kantor Departemen Agama Kabupaten/Kota dan Balai Diklat Pendidikan Teknis Keagamaan. Saat itu jauh sebelumnya Departemen Agama telah mengalami beberapa pergantian nama, mulai dari nama Kantor Kepenghuluan Kabupaten dan selanjutnya berubah lagi menjadi Kantor Urusan Agama Tingkat II. Hal ini berdasarkan KMA nomor 47 tahun 1963 tentang Perencanaan Organisasi dan Tata Kerja Departemen Agama, dan terakhir pergantian nama hingga saat ini menjadi Kementerian Agama Kota Malang mulai tanggal 28 Januari 2010 sesuai dengan PMA nomor 1 tahun 2010 tentang perubahan penyebutan Departemen Agama menjadi Kementerian Agama. Mengawali kegiatan perkantoran pada tahun 1981 Departemen Agama Kota Malang bertempat di Jl. Aris Munandar nomor 35 (saat ini difungsikan sebagai rumah dinas kepala kantor) dan baru pada tahun 1987 pindah tempat hingga saat ini menempati perkantoran di Jl. Raden Panji Suroso No. 2 Malang.

2.1.2. Visi dan Misi

Visi Kementerian Agama Kota Malang yaitu terwujudnya masyarakat Indonesia yang taat beragama, rukun, cerdas, dan sejahtera lahir batin dalam rangka mewujudkan Indonesia yang berdaulat, mandiri, dan berkepribadian berlandaskan gotong royong.

Untuk mewujudkan visi Kementerian Agama Kota Malang maka misi Kementerian Agama Kota Malang sebagai berikut:

1. Meningkatkan pemahaman dan pengamalan ajaran agama.
2. Memantapkan kerukunan intra dan antar umat bergama.
3. Menyediakan pelayanan kehidupan beragama yang merata dan berkualitas.
4. Meningkatkan pemanfaatan dan kualitas pengelolaan potensi ekonomi keagamaan.
5. Mewujudkan penyelenggaraan ibadah haji dan umrah yang berkualitas dan akuntabel.
6. Meningkatkan akses dan kualitas pendidikan umum berciri agama, pendidikan agama pada satuan pendidikan umum dan pendidikan keagamaan.
7. Mewujudkan tata kelola pemerintahan yang bersih, akuntabel dan terpercaya.

2.2. Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Jogiyanto, 2005).

Pada prinsipnya terdapat beberapa elemen yang ada didalam sistem, antara lain sebagai berikut:

1. Objek dapat berupa bagian elemen, variabel atau dapat berupa benda fisik, abstrak, atau keduanya sekaligus tergantung dengan sifat dari sistem tersebut.
2. Atribut, merupakan penentuan kualitas atau sifat kepemilikan sistem dan objeknya.

3. Hubungan internal diantara objek-objeknya.
4. Lingkungan, merupakan tempat dimana sistem tersebut berada.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.3. Surat Masuk dan Surat Keluar

Surat masuk adalah surat-surat yang diterima oleh suatu organisasi/perusahaan yang berasal dari seseorang atau dari suatu organisasi. Sedangkan surat keluar adalah surat-surat yang dikeluarkan/dibuat suatu organisasi/perusahaan untuk dikirimkan kepada pihak lain, baik perseorangan maupun kelompok.

Di Kementerian Agama Kota Malang, terdapat SOP (*standard operating procedure*) tersendiri dalam penanganan surat masuk dan keluar yakni:

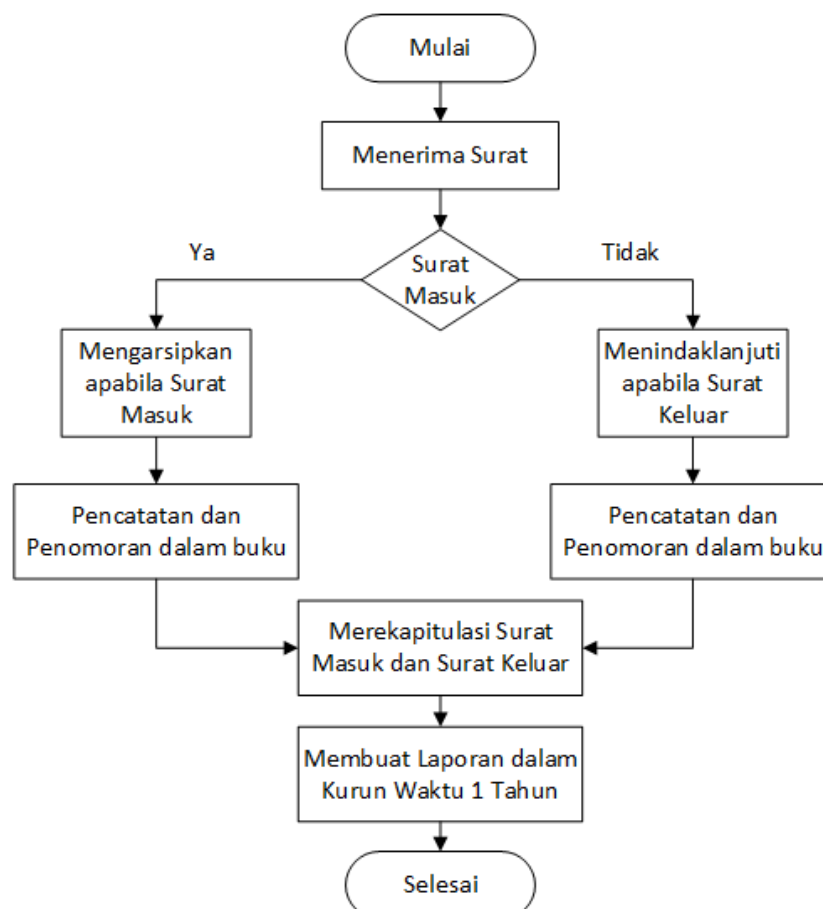
2.3.1. SOP Administrasi Surat Masuk dan Keluar

Tabel 2.1 Tahapan Administrasi Surat Masuk dan Keluar

No	Aktivitas	Pelaksana			Mutu Baku		
		1	2	3	Persyaratan	Output	Waktu
1	Menerima surat yang didisposisikan Pimpinan	Kabag	Kasubag	Staf	Tata Persuratan Dinas	Tertatanya administrasi surat	10 menit
2	Mencatat surat masuk dan surat keluar	Staf			Buku pencatatan surat	Tercatatnya surat masuk dan surat keluar	15 menit
3	Mengarsipkan surat masuk dan membuat surat keluar sesuai isi disposisi pimpinan	Kabag	Kasubag	Staf	Tata persuratan Dinas	Tersusunnya surat keluar dan terarsipkannya surat masuk	30 menit
4	Memberi nomor surat masuk dan surat keluar	Staf			Tata persuratan Dinas	Terdapatnya nomor surat masuk dan keluar	15 menit

5	Merekapitulasi Surat Masuk dan Surat Keluar	Staf			Tata persura tan Dinas	Tersusunnya rekapitulasi surat masuk dan keluar	1 jam
6	Membuat Laporan Surat Masuk dan Surat Keluar dalam kurun waktu 1 tahun	Staf				Tersusunnya laporan surat masuk dan surat keluar	1 Minggu

Aktifitas pertama merupakan penerimaan surat ke divisi umum, sub bagian tata usaha, jika surat adalah surat keluar maka akan ditindaklanjuti oleh staf untuk dicatat dan diberi penomoran dalam buku. Jika berjenis surat masuk maka surat tersebut diarsipkan dengan memberi catatan serta penomoran dalam buku. Setelah memberi nomor pada surat maka staf divisi umum akan merekapitulasi surat masuk dan keluar, dan terakhir dalam kurun satu tahun staf akan membuat laporan tahunan (agenda surat). Aktifitas administrasi surat masuk dan keluar masuk tersebut dapat juga dilihat pada Gambar 2.1.



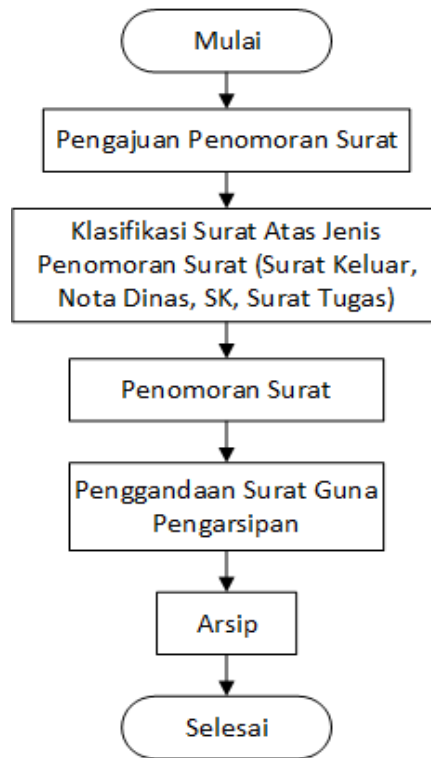
Gambar 2.1 Flowchart Administrasi Surat Masuk dan Keluar

2.3.2. SOP Penomoran Surat

Tabel 2.2 Tahapan Penomoran Surat

No	Aktivitas	Pelaksana			Mutu Baku		
		1	2	3	Persyaratan	Output	Waktu
1	Pengajuan Penomoran Surat	Pengadm inistrasi Umum Unit Pengolah			Surat Keluar/ LHA-STL yang telah ditandatangani Pejabat	Tersusunnya secara urut surat-surat yang siap dengan penomoran sesuai kurtu kendali	15 Menit
2	Pemilahan surat atas Jenis Penomoran Surat (Surat Keluar, Nota Dinas, SK, Surat Tugas)	Pengadm inistrasi Umum Unit Tata Usaha			Surat Keluar/ LHA-STL yang telah ditandatangani Pejabat	Terklasifikasinya surat hingga penomorannya	15 Menit
3.	Penomoran Surat	Pengadm inistrasi Umum Unit tata Usaha			Buku Kendali, Surat Keluar, nota Dinas, SK, Surat Tugas	Terciptanya surat keluar yang telah ditandatangani dan memiliki tanggal dan nomor Surat	10 Menit
4	Penggandaan Surat guna pengarsipan	Pengadm inistrasi Umum Unit Pengolah			Surat Keluar yang telah ditandatangani dan memiliki tanggal dan nomor surat	Tersedianya arsip Surat pada Unit Pengolah dan Unit tata Usaha	10 Menit

Untuk SOP penomoran surat, sub bagian atau divisi lain di Kementerian Agama Kota Malang melakukan pengajuan penomoran surat keluar, nota dinas, SK, atau surat tugas ke divisi umum. Dalam waktu 15 menit divisi umum melakukan klasifikasi surat berdasarkan jenisnya, jika sudah maka dilakukan penomoran surat dilanjut dengan penggandaan surat menggunakan perangkat *scanner* untuk diarsipkan. Untuk lebih jelasnya dapat dilihat pada Gambar 2.2.



Gambar 2.2 Flowchart SOP Penomoran Surat Surat

2.4. Basis Data

Basis data merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan pada perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

Sistem basis data yang kami pakai yakni MariaDB. MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL dan tersedia dengan lisensi GPL v2. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Michael Monty Widenius sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB.

Secara teori semua program yang dibangun menggunakan MySQL bisa berjalan pula dengan MariaDB. Terdapat beberapa API (*Application Programming Interface*) tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses basis data MariaDB antara lain: bahasa pemrograman C, C++, C#, Smalltalk, Java, Lisp, Perl, dan PHP. MariaDB memiliki beberapa keunggulan diantaranya *powerfull*, mudah dalam

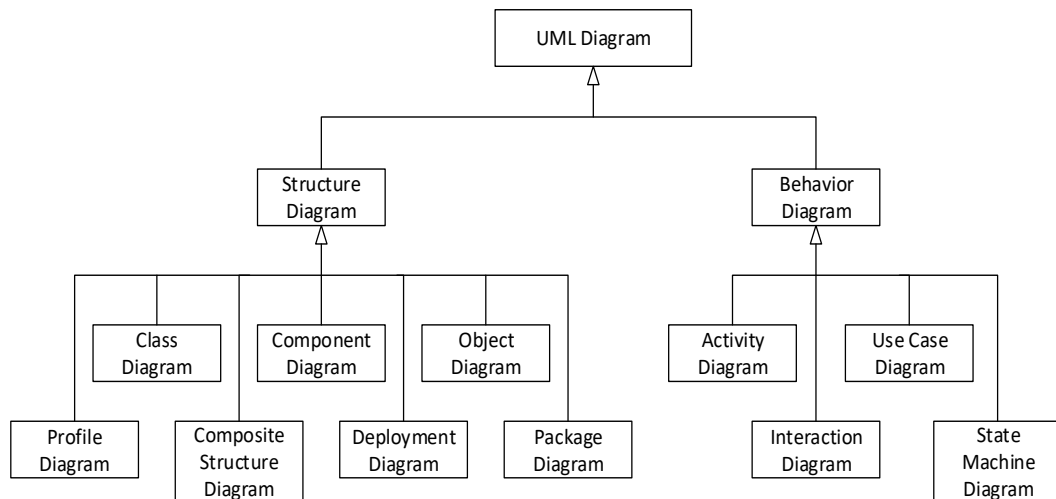
pemasangan dan penggunaan, serta tersedia untuk multiplatform (Bartholomew, 2015).

2.5. Pemodelan Berorientasi Objek

Pemodelan berorientasi objek adalah suatu teknik yang menggunakan perspektif dari objek sebagai *problem domain*-nya. Pemodelan ini mendeskripsikan dunia nyata dengan objek, lengkap dengan *attributes*, *services* dan *relationships*-nya. Keuntungan menggunakan pemodelan berorientasi objek diantaranya *reusable* (bisa digunakan kembali) dan *measurable* (terukur).

2.5.1. Unified Modelling Language (UML)

Menurut Nugroho, UML (*Unified Modelling Language*) adalah metodologi kolaborasi antara Booch, OMT (*Object Modelling Technique*), OOSE (*Object Oriented Software Engineer*), dan beberapa metode lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek (OOP) (Nugroho, 2009).



Gambar 2.3 Diagram UML

Dilihat dari Gambar 2.3 (Fowler, 2004), *Structure diagram* merupakan kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.


2.5.2. Use Case Diagram

Menurut Shelly dan Rosenblatt, *use case* adalah sebuah pola yang merepresentasikan langkah-langkah kedalam fungsi spesifik bisnis atau proses (Sibero, 2011).

Menurut Satziger, Jackson dan Burd, *use case* adalah sebuah daftar yang memberikan apa yang sistem lakukan saat kejadian terjadi (Satzinger, 2010).

Jadi kesimpulan dari pengertian *use case* adalah sebuah daftar yang membentuk sebuah pola yang memberikan apa yang sistem lakukan pada saat kejadian terjadi. Setiap *use case* dilengkapi dengan use case scenario.

Tabel 2.3 Elemen-elemen Use Case

No.	Simbol	Nama	Keterangan
1.		Aktor	Merepresentasikan pelaku use case
2.		Use Case	Merepresentasikan suatu hal yang dilakukan oleh aktor.

2.5.3. Sequence Diagram

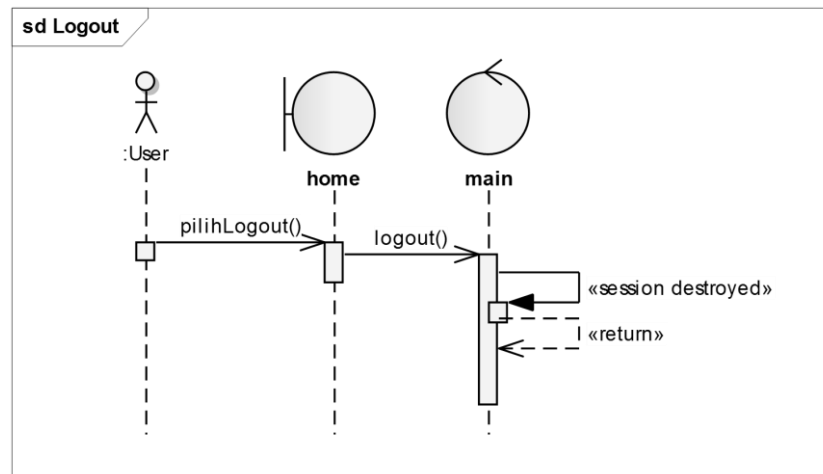
Sequence diagram menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metode dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki

sifat khusus, standar UML mendefinisikan ikon khusus untuk objek *boundary*, *controller* dan *entity* (Pressman, 2010).



Gambar 2.4 Contoh Sequence Diagram

2.5.4. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan *objek* beserta hubungan satu sama lain yaitu:

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.
2. Generalisasi ('*is-a*'), yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya.
3. Agregasi, yaitu hubungan yang menyatakan bagian ('*part-of*').
4. Komposisi, hubungan yang hampir sama dengan agregasi tetapi lebih spesifik. Komposisi biasanya memiliki ketergantungan yang kuat antara *container class* dan instance dari *container class*. Jika *container class* dihilangkan maka setiap *instance* yang ada didalamnya otomatis akan

hilang. Contoh hubungan antara bangunan (*container class*) dengan ruangan (*instance*).

2.6. PHP

Menurut Sibero, PHP adalah pemrograman yang memproses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan (Sibero, 2011). Sedangkan menurut Adhi Prasetyo, PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman web *server-side* yang bersifat *open source* (Prasetyo, 2010).

PHP memiliki kemampuan untuk memisahkan kode dari HTML. Beberapa kelebihan dari PHP sebagai berikut:

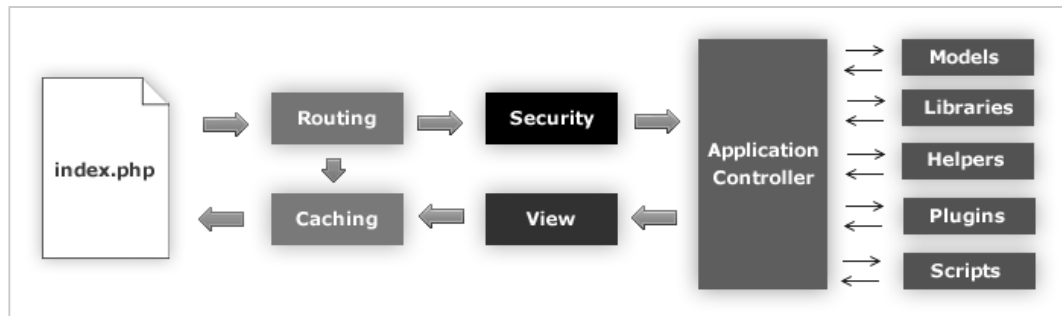
1. Kesederhanaan, mudah dipelajari karena mempunyai banyak referensi serta bisa digunakan untuk membuat website dinamis.
2. PHP bersifat *open source*, sehingga PHP mudah didapatkan secara gratis dan tersedia secara versi-versi baru dalam jangka waktu yang cepat.
3. Stabilitas dan kompatibilitas, PHP stabil diberbagai sistem operasi selain itu, PHP juga terintegrasi secara baik dengan berbagai macam webserver termasuk 2 yang paling populer yaitu IIS dan Apache.
4. Kemampuan proses cepat dalam menampilkan halaman web serta mampu berinteraksi dengan banyak basis data.

2.7. CodeIgniter

CodeIgniter adalah sebuah *framework* PHP. *Framework* itu sendiri adalah suatu kerangka kerja yang berupa sekumpulan folder yang memuat file-file PHP yang menyediakan kelas *libraries*, *helpers*, *plugins*, dan lainnya. *Framework* menyediakan konfigurasi dan teknik koding tertentu (Sofwan, 2007).

CodeIgniter menggunakan konsep MVC secara fleksibel. Konsep MVC (*Model-View-Controller*) adalah konsep pemisahan antara algoritma pemrograman (*controller*) dengan tampilan (*view*) dan basis data (*model*). Manfaat konsep ini adalah mempermudah dalam pengembangan perangkat lunak, karena membuat programmer dapat bekerja secara terpisah dengan

designer. Programmer mengerjakan *controller*, sedangkan *designer* berkeutatan dengan *view*.



Gambar 2.5 Arsitektur *Framework CodeIgniter*

1. Index.php berfungsi sebagai *front controller*, menginisialisasi sumber daya dasar yang dibutuhkan untuk menjalankan CodeIgniter.
2. Routing memeriksa HTTP *request* untuk menentukan apa yang harus dilakukan dengan *request* tersebut.
3. Jika terdapat file yang akan di-cache, maka langsung dikirim ke browser, melewati eksekusi sistem secara normal.
4. *Security*. Sebelum *controller* sistem dimuat, data yang dikirim oleh HTTP *request* dan oleh setiap pengguna disaring untuk keamanan.
5. *Controller* memuat model, core libraries, helpers, dan setiap sumber daya lainnya yang dibutuhkan untuk memproses *request* khusus.
6. *View* yang sudah lengkap langsung di-render dan dikirim ke web browser untuk dilihat. Jika cache diaktifkan maka *view* akan di-cache pertama kali sehingga *request* selanjutnya dapat diproses.

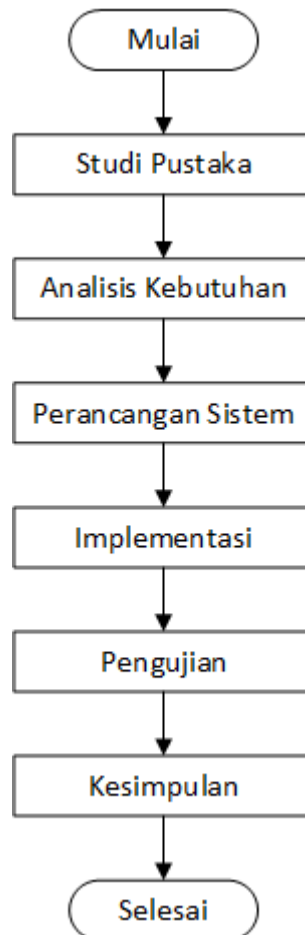
2.8. Black Box Testing

Dengan mengetahui spesifikasi fungsi dari sistem yang dibangun, pengujian dilakukan untuk menunjukkan bahwa setiap fungsi beroperasi sepenuhnya. Pada umumnya pengujian *black box* lebih sering digunakan dalam rekayasa perangkat lunak yang tidak terlalu kompleks dengan alasan lebih mudah digunakan. Perangkat lunak diperlakukan seperti *black box*, dimana hanya bisa diamati dari *input* yang diberikan dan *output* yang dikeluarkan (Sommerville, 2011).

BAB III

METODOLOGI PENELITIAN

Tahapan metode penelitian yang digunakan pada laporan PKL ini adalah analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, dan penarikan kesimpulan. Adapun diagram alir analisis kebutuhan ditunjukkan oleh Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1. Studi Pustaka

Studi pustaka ini digunakan untuk memperoleh dasar-dasar teori dan pengetahuan terhadap domain masalah. Studi Pustaka ini nantinya menjadi sumber acuan dalam penulisan laporan dan pengembangan sistem. Teori-teori pendukung penulisan serta pemahaman tentang materi pengembangan perangkat lunak ini diperoleh dari buku, jurnal, *e-book*, dan penelitian sebelumnya.

yang terkait dengan subjek penelitian yang akan dibuat. Adapun teori dan pustaka yang berkaitan dengan penelitian ini meliputi:

1. Sistem Informasi
2. Surat Masuk dan Surat Keluar
3. Basis Data
4. Pemodelan Berorientasi Objek
 - a. *Unified Modelling Language* (UML)
 - b. Use Case Diagram
 - c. Sequence Diagram
 - d. Class Diagram
5. PHP
6. CodeIgniter
7. *Black Box Testing*

3.2. Analisis Kebutuhan

Dalam tahap ini dilakukan elisitasi dan analisis terhadap kebutuhan yang akan dipenuhi oleh sistem. Proses ini dilakukan untuk menggali domain sistem, fitur-fitur sistem apa saja yang harus disediakan, unjuk kerja sistem, dan batasan dari sistem. Analisis kebutuhan dilakukan dengan melakukan observasi terhadap dokumen *strandart operating procedure* surat masuk dan keluar Kementerian Agama Kota Malang serta observasi langsung dengan melihat proses manajemen surat masuk dan keluar secara *offline* yang sudah ada di Sub Bagian Umum Tata Usaha, Divisi Umum.

Analisis kebutuhan menggunakan pendekatan OO (*object oriented*). Hasil analisis kebutuhan ini berupa gambaran umum sistem, definisi kebutuhan, serta pemodelan kebutuhan dari sistem. Hasilnya nanti akan didokumentasikan sesuai dengan SRS (*software requirement specification*). Hasil analisis kebutuhan ini akan dijadikan sebagai dasar dalam melakukan perancangan dan pengujian sistem.

3.3. Perancangan Sistem

Perancangan adalah proses mentransformasi hasil analisis kebutuhan yang telah didefinisikan dan dispesifikasikan kedalam model perancangan. Dalam membuat model perancangan tersebut menggunakan OOD (*object oriented design*) Model perancangan ini yang nantinya digunakan untuk melakukan tahapan implementasi. Adapun tahapan-tahapan dalam perancangan sistem yang akan dibangun dalam laporan ini yakni:

1. Perancangan Komponen

Dalam hal ini dilakukan pendekomposisian subsistem menjadi komponen-komponen sistem yang lebih detil berupa algoritma yang siap untuk diimplementasikan. Algoritma dari beberapa fungsi direpresentasikan dalam bentuk *pseudocode* untuk persiapan implementasi kedalam bentuk kode program.

2. Perancangan Data

Dalam hal ini dilakukan transformasi domain informasi dari hasil analisis menjadi struktur data yang dibutuhkan untuk menyimpan seluruh informasi yang dibutuhkan oleh sistem. Perancangan data ini dilakukan dengan membuat perancangan basis data berupa *conseptual data model* dengan mendefinisikan dan menspesifikasikan tabel-tabel basis data untuk menyimpan data sistem.

3. Perancangan Antarmuka

Dalam hal ini dilakukan perancangan sketsa interface atau antarmuka pengguna dari sistem yang akan dikembangkan serta perancangan antarmuka interaksi antara objek-objek yang ada dalam sistem. Perancangan antarmuka pengguna dilakukan dengan membuat desain UI (*user interface*) halaman-halaman yang ada pada sistem.

3.4. Implementasi

Pada tahap ini dilakukan proses implementasi rancangan sistem yang telah dibuat pada tahap sebelumnya menjadi suatu sistem informasi tata persuratan. Implementasi sistem berupa kode komponen dan *user interface* dibuat dengan menggunakan bahasa pemrograman PHP menggunakan *framework* CodeIgniter dengan memanfaatkan media penyimpanan basis data MariaDB sehingga akan menghasilkan suatu sistem informasi tata persuratan yang berbasis web.

3.5. Pengujian

Pengujian yang digunakan yaitu pengujian validasi dengan menggunakan metode *black box testing*. Pengujian validasi merupakan pengujian untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan analisis kebutuhan baik kebutuhan fungsional dan non fungsional

Metode *black box testing* adalah pengujian yang menganggap sistem sebagai kotak hitam yang menerima input berupa *test case* dan mengeluarkan output berupa *result*, jika hasilnya sesuai dengan *expected result* maka *case* tersebut dianggap valid. Uji coba ini dilakukan oleh tim internal dan oleh calon pengguna untuk mengetahui kelebihan dan kekurangan sistem.

3.6. Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan analisis dan perancangan, implementasi, dan pengujian Sistem Informasi Surat Masuk dan Keluar telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Didalam kesimpulan juga terdapat saran yang merupakan ide atau pikiran untuk penulis dan *stakeholder* untuk pengembangan sistem selanjutnya.

BAB IV

ANALISIS DAN PERANCANGAN

Pada bab ini membahas tentang analisis kebutuhan dan perancangan perangkat lunak Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang. Pada tahap perancangan ini dilakukan dengan tiga tahap yaitu proses analisis kebutuhan, pemodelan kebutuhan, dan perancangan sistem.

4.1. Analisis Kebutuhan

Analisis kebutuhan perangkat lunak akan menjelaskan tentang kebutuhan dari Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang. Metode analisis menggunakan UML (*Unified Modelling Language*) berorientasi objek.

4.1.1. Gambaran Umum Sistem

Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang dirancang dan dibuat dengan melibatkan bantuan dari pihak divisi umum Kementerian Agama Kota Malang selaku pengguna dari sistem ini. Pada proses perancangan sistem, pihak divisi umum Kementerian Agama Kota Malang memberitahukan informasi mengenai peraturan dan alur persuratan baik surat masuk maupun surat keluar.

Penulis juga diberikan informasi tentang proses disposisi surat masuk dan keluar yang dilakukan di Kementerian Agama Kota Malang pada umumnya dan disposisi surat masuk pada divisi umum khususnya. Kebutuhan informasi lain yang dibutuhkan dalam pengembangan sistem persuratan ini juga disampaikan oleh kepala divisi umum Kementerian Agama Kota Malang.

Sistem informasi yang kami buat adalah Sistem Informasi Surat Masuk dan Keluar yang kami sebut sebagai SIMAK. SIMAK sendiri merupakan sebuah sistem informasi berbasis web yang bertujuan untuk memudahkan pengarsipan dan penyedia informasi surat masuk dan keluar tanpa membutuhkan resource komputasi yang besar.

Sistem informasi ini dapat membantu pengguna khususnya divisi umum yang membutuhkan manajemen persuratan untuk mempermudah proses

penataan serta pengarsipan surat masuk dan keluar secara digital dari Kementerian Agama Kota Malang yang melalui divisi umum.

Sebelum sistem ini dibuat, pada divisi umum menggunakan proses penginputan dan pendataan surat masih dilakukan secara manual menggunakan buku agenda. Sistem mengubah proses pendataan surat yang semula masih dikerjakan secara manual (tuliskan tangan), menjadi lebih efisien karena hanya dengan menginputkan poin-poin penting yang selanjutnya akan diproses oleh sistem tersebut.

Sistem ini juga dapat meningkatkan produktivitas dan efisiensi kerja serta dapat mengurangi jumlah penggunaan kertas. Sistem persuratan ini dapat diakses oleh siapapun, kapanpun, dan dimanapun selama pengguna diberikan hak akses untuk menggunakan sistem persuratan ini.

Akses yang dilakukan pada sistem persuratan ini terdiri dari dua macam tipe yaitu Admin dan User. User biasa hanya dapat mengakses sistem tanpa bisa memanajemen user lainnya. Sedangkan Admin bisa mengakses seluruh fitur aplikasi.

4.1.2. Identifikasi Aktor

Pada bagian ini dilakukan identifikasi aktor yang akan berinteraksi dengan sistem persuratan. Aktor merupakan gambaran dari pengguna. Pada kolom aktor akan disebutkan aktor-aktor yang berperan didalam sistem dan dikolom lainnya akan dideskripsikan seperti apa aktor yang akan berperan di dalam sistem. Tabel 4.1 akan menunjukkan aktor-aktor yang berperan didalam sistem beserta penjelasannya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi Aktor
User	User merupakan aktor pengguna sistem yang memiliki akses untuk ganti password, kelola surat masuk, kelola surat keluar, klasifikasi surat, cetak agenda surat masuk, dan cetak agenda surat keluar.
Admin	Admin merupakan aktor pengguna sistem yang memiliki semua akses seperti user ditambah bisa fitur <i>me-manage</i> user, dan setting instansi.

4.1.3. Daftar Kebutuhan Sistem

Tahapan ini bertujuan untuk menjelaskan kebutuhan sistem yang harus dipenuhi oleh sistem saat aktor melakukan sebuah aksi. Daftar kebutuhan dibagi menjadi dua yaitu:

1. Kebutuhan Fungsional

Tabel 4.2 Kebutuhan Fungsional

Nomor SRS	Aktor	Kebutuhan	Use Case
SRS-F-SIMAK-01	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat masuk kedalam sistem	Login
SRS-F-SISTA-02	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat keluar dari sistem	Logout
SRS-F-SISTA-03	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mengganti password	Ganti Password
SRS-F-SISTA-04	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mengklasifikasikan surat keluar	Klasifikasi Surat
SRS-F-SISTA-05	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mencetak agenda surat masuk	Cetak Agenda Surat Masuk
SRS-F-SISTA-06	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mencetak agenda surat keluar	Cetak Agenda Surat Keluar
SRS-F-SISTA-07	Admin dan User	Sistem menyediakan fitur bagi aktor untuk mengelola surat masuk	Kelola Surat Masuk
SRS-F-SISTA-08	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mencari surat masuk berdasarkan tanggal atau kata kunci tertentu	Cari Surat Masuk

SRS-F-SISTA-09	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat menambah surat masuk	Tambah Surat Masuk
SRS-F-SISTA-10	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mengedit surat masuk	Edit Surat Masuk
SRS-F-SISTA-11	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat menghapus surat masuk	Hapus Surat Masuk
SRS-F-SISTA-12	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mendisposisi suatu surat masuk	Disposisi Surat Masuk
SRS-F-SISTA-13	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mencetak disposisi surat	Cetak Disposisi Surat
SRS-F-SISTA-14	Admin dan User	Sistem menyediakan fitur bagi aktor untuk mengelola surat keluar	Kelola Surat Keluar
SRS-F-SISTA-15	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mencari surat keluar berdasarkan tanggal atau kata kunci tertentu	Cari Surat Keluar
SRS-F-SISTA-16	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat menambah surat keluar	Tambah Surat Keluar
SRS-F-SISTA-17	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat mengedit surat keluar	Edit Surat Keluar
SRS-F-SISTA-18	Admin dan User	Sistem menyediakan fitur bagi aktor untuk dapat menghapus surat keluar	Hapus Surat Keluar
SRS-F-SISTA-19	Admin	Sistem menyediakan fitur bagi aktor untuk dapat mengatur nama,	Setting Instansi

		alamat, kepala, NIP kepala, dan logo instansi	
SRS-F-SISTA-20	Admin	Sistem menyediakan fitur bagi aktor untuk dapat mengelola user yang ada dalam sistem	Manage User
SRS-F-SISTA-21	Admin	Sistem menyediakan fitur bagi aktor untuk dapat menambah user yang ada dalam sistem	Tambah User
SRS-F-SISTA-22	Admin	Sistem menyediakan fitur bagi aktor untuk dapat mengedit informasi user berupa username, nama lengkap, serta NIP/NIK	Edit User
SRS-F-SISTA-23	Admin	Sistem menyediakan fitur bagi aktor untuk dapat menghapus user	Hapus User
SRS-F-SISTA-24	Admin	Sistem menyediakan fitur bagi aktor untuk dapat mencari user berdasarkan nama	Cari User

2. Kebutuhan Non Fungsional

Tabel 4.3 Kebutuhan Non Fungsional

Nomor SRS	Parameter	Kebutuhan
SRS-NF-SISTA-25	<i>Responsiveness</i>	Sistem dapat dijalankan diberbagai resolusi layar browser Mozilla Firefox atau Google Chrome.
SRS-NF-SISTA-26	Performa	Sistem harus bisa diakses dalam waktu kurang dari 5 detik.

Keterangan:

- SRS : *Software Requirements Spesification*
- F : Kode Representasi Kebutuhan fungsional
- NF : Kode Representasi Kebutuhan Non Fungsional
- SIMAK : Singkatan Nama Sistem
- 01-26 : Nomor Urut Representasi Kebutuhan

4.2. Pemodelan Kebutuhan

4.2.1. Use Case



Gambar 4.1 Use Case Diagram SIMAK

Berdasarkan dari rancangan *use case* pada Gambar 4.1, level admin dapat mengakses seluruh fitur dalam sistem. Sedangkan user hanya dapat mengakses keseluruhan sistem tanpa bisa *manage* user dan instansi.

4.2.2. Use Case Scenario

1. Login (SRS-F-SIMAK-01)

Tabel 4.4 Use Case Scenario Login

Kode Kebutuhan	SRS_SIMAK_F_01
Objective	Menggambarkan aktifitas aktor untuk masuk kedalam sistem informasi
Aktor	Admin dan User

Pre-Conditions	1. Aktor berada pada halaman login 2. Aktor belum ter- <i>autentifikasi</i>
Main Flow	1. Aktor memasukkan <i>username</i> 2. Aktor memasukkan password dan menekan button login 3. Session aktor dibuat 4. Sistem akan mengarahkan aktor ke halaman home sesuai level aktor
Alternative Flow	Apabila username dan password Aktor salah, sistem akan mengarahkan kembali ke halaman login dengan menampilkan pesan bahwa username atau password tidak <i>valid</i>
Post-Conditions	Aktor berhasil ter- <i>autentifikasi</i> kedalam sistem

2. Logout (SRS-F-SISTA-02)

Tabel 4.5 Use Case Scenario Logout

Kode Kebutuhan	SRS_SIMAK_F_02
Objective	Menggambarkan aktifitas aktor yang keluar dari sistem
Aktor	Admin dan User
Pre-Conditions	Aktor masih ter- <i>autentifikasi</i>
Main Flow	1. Aktor memilih submenu Logout 2. Session aktor dihapus 3. Sistem mengarahkan ke halaman login
Alternative Flow	-
Post-Conditions	Aktor keluar dari sistem dan menjadi tidak ter- <i>autentifikasi</i>

3. Ganti Password (SRS_SIMAK_F_03)

Tabel 4.6 Use Case Scenario Ganti Password

Kode Kebutuhan	SRS_SIMAK_F_03
Objective	Menggambarkan aktifitas aktor mengganti password
Aktor	Admin dan User
Pre-Conditions	Aktor sudah ter- <i>autentifikasi</i>
Main Flow	1. Aktor memilih submenu ganti password 2. Aktor memasukkan password lama 3. Aktor memasukkan password baru 4. Aktor mengulangi memasukkan password baru 5. Sistem menyimpan password baru

Alternative Flow	1. Jika aktor salah memasukkan password lama, sistem akan menampilkan informasi “password lama tidak sama” 2. Jika aktor menekan tombol kembali, sistem akan mengembalikan ke halaman sebelumnya 3. Jika konfirmasi password baru tidak sama dengan password baru sebelumnya, sistem menampilkan informasi “Password baru 1 dan 2 tidak cocok”
Post-Conditions	-

4. Klasifikasi Surat (SRS_SIMAK_F_04)

Tabel 4.7 Use Case Scenario Klasifikasi Surat

Kode Kebutuhan	SRS_SIMAK_F_04
Objective	Menggambarkan aktifitas aktor dalam mengklasifikasikan surat keluar
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter-authentifikasi 2. Aktor mengetahui kategori klasifikasi surat
Main Flow	1. Aktor memilih menu klasifikasi 2. Aktor memilih tombol edit data pada klasifikasi surat 3. Aktor memasukkan data yang dirubah 4. Sistem menyimpan data klasifikasi surat
Alternative Flow	1. Jika aktor memilih tombol Delete Data, sistem akan menampilkan informasi konfirmasi 2. Jika aktor memilih “OK”, sistem melanjutkan dalam menghapus klasifikasi surat 3. Jika aktor memilih “Cancel”, sistem tidak jadi menghapus klasifikasi surat
Post-Conditions	-

5. Cetak Agenda Surat Masuk (SRS_SIMAK_F_05)

Tabel 4.8 Use Case Scenario Cetak Agenda Surat Masuk

Kode Kebutuhan	SRS_SIMAK_F_05
Objective	Menggambarkan aktifitas aktor mencetak agenda surat masuk
Aktor	Admin dan User
Pre-Conditions	Aktor telah ter-authentifikasi
Main Flow	1. Aktor memilih submenu Agenda Surat Masuk 2. Aktor memilih tanggal awal surat yang dicetak 3. Aktor memilih tanggal akhir surat yang dicetak 4. Aktor mengatur pengaturan printer 5. Agenda surat masuk berhasil dicetak

Alternative Flow	1. Jika aktor belum memasukkan tanggal awal/akhir surat yang dicetak, sistem akan menampilkan informasi "Please fill out this field" sesuai field yang belum diisi 2. Jika aktor telah memilih tanggal awal/akhir surat yang dicetak, kemudian menekan tombol clear, sistem akan menghapus isian yang telah dimasukkan user
Post-Conditions	-

6. Cetak Agenda Surat Keluar (SRS_SIMAK_F_06)

Tabel 4.9 Use Case Scenario Cetak Agenda Surat Keluar

Kode Kebutuhan	SRS_SIMAK_F_06
Objective	Menggambarkan aktifitas aktor mencetak agenda surat keluar
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih submenu Agenda Surat Keluar 2. Aktor memilih tanggal awal surat yang dicetak 3. Aktor memilih tanggal akhir surat yang dicetak 4. Aktor mengatur pengaturan printer 5. Agenda surat masuk berhasil dicetak
Alternative Flow	1. Jika aktor belum memasukkan tanggal awal/akhir surat yang dicetak, sistem akan menampilkan informasi "Please fill out this field" sesuai field yang belum diisi 2. Jika aktor telah memilih tanggal awal/akhir surat yang dicetak, kemudian menekan tombol clear, sistem akan menghapus isian yang telah dimasukkan user
Post-Conditions	-

7. Kelola Surat Masuk (SRS_SIMAK_F_07)

Tabel 4.10 Use Case Scenario Kelola Surat Masuk

Kode Kebutuhan	SRS_SIMAK_F_07
Objective	Menggambarkan aktifitas aktor mengakses menu kelola surat masuk
Aktor	Admin dan User
Pre-Conditions	Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih menu Kelola Surat Masuk 2. Sistem menampilkan halaman kelola surat masuk
Alternative Flow	-
Post-Conditions	Sistem menampilkan daftar surat masuk

8. Cari Surat Masuk (SRS_SIMAK_F_08)

Tabel 4.11 Use Case Scenario Cari Surat Masuk

Kode Kebutuhan	SRS_SIMAK_F_08
Objective	Menggambarkan aktifitas aktor mencari surat masuk
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i> 2. Aktor telah berada pada halaman Kelola Surat Masuk
Main Flow	1. Aktor memilih menu Kelola Surat Masuk 2. Aktor memasukkan kata kunci pencarian dan/atau tanggal surat yang dicari
Alternative Flow	-
Post-Conditions	Sistem menampilkan daftar surat masuk berdasarkan tanggal atau kata kunci

9. Tambah Surat Masuk (SRS_SIMAK_F_09)

Tabel 4.12 Use Case Scenario Tambah Surat Masuk

Kode Kebutuhan	SRS_SIMAK_F_09
Objective	Menggambarkan aktifitas aktor menambahkan data surat masuk
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih menu Kelola Surat Masuk 2. Aktor menekan tombol Tambah Data 3. Sistem menampilkan form data surat masuk 4. Aktor memasukkan nomor surat masuk 5. Aktor memasukkan asal surat masuk 6. Aktor memasukkan isi surat masuk 7. Aktor memasukkan tanggal surat masuk 8. Aktor memasukkan file gambar hasil scan surat masuk 9. Aktor memasukkan keterangan surat masuk 10. Aktor melakukan submit data 11. Sistem menyimpan data surat masuk
Alternative Flow	1. Jika aktor menekan tombol kembali, sistem akan mengembalikan ke halaman sebelumnya 2. Jika aktor menekan tombol Clear, sistem akan mengembalikan isi data semua kolom <i>properties</i> surat masuk ke keadaan semula
Post-Conditions	-

10. Edit Surat Masuk (SRS_SIMAK_F_10)

Tabel 4.13 Use Case Scenario Edit Surat Masuk

Kode Kebutuhan	SRS_SIMAK_F_10
Objective	Menggambarkan aktifitas aktor mengedit surat masuk
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i> 2. Aktor telah berada pada halaman Kelola Surat Masuk
Main Flow	1. Aktor memilih menu Kelola Surat Masuk 2. Aktor menekan tombol Edit Data pada surat yang akan diedit 3. Aktor mengubah data pada kolom yang perlu dirubah 4. Sistem menyimpan data surat masuk yang telah diedit
Alternative Flow	1. Jika aktor menekan tombol kembali, sistem akan mengembalikan ke halaman sebelumnya 2. Jika aktor menekan tombol Clear, sistem akan mengembalikan isi data semua kolom <i>properties</i> surat masuk ke keadaan semula
Post-Conditions	-

11. Hapus Surat Masuk (SRS_SIMAK_F_11)

Tabel 4.14 Use Case Scenario Hapus Surat Masuk

Kode Kebutuhan	SRS_SIMAK_F_11
Objective	Menggambarkan aktifitas aktor menghapus data surat masuk
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i> 2. Ada data surat masuk
Main Flow	1. Aktor memilih menu Kelola Surat Masuk 2. Aktor menekan tombol 'Hapus Data' pada surat masuk yang akan dihapus 3. Sistem menampilkan konfirmasi penghapusan data 4. Aktor mengkonfirmasi penghapusan data surat masuk 5. Sistem menampilkan informasi bahwa data surat masuk telah terhapus
Alternative Flow	Jika aktor menekan tombol 'Cancel', sistem tidak jadi menghapus surat masuk
Post-Conditions	Data surat masuk telah terhapus

12. Disposisi Surat Masuk (SRS_SIMAK_F_12)

Tabel 4.15 Use Case Scenario Disposisi Surat Masuk

Kode Kebutuhan	SRS_SIMAK_F_12
Objective	Menggambarkan aktifitas aktor membuat disposisi suatu surat masuk
Aktor	Admin dan User
Pre-Conditions	Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih icon Disposisi Surat Masuk 2. Aktor memilih tambah data 3. Aktor memasukkan data yang diperlukan Disposisi Surat 4. Aktor memilih pilihan simpan
Alternative Flow	-
Post-Conditions	Disposisi surat masuk telah disimpan sistem

13. Cetak Disposisi Surat (SRS_SIMAK_F_13)

Tabel 4.16 Use Case Scenario Cetak Disposisi Surat

Kode Kebutuhan	SRS_SIMAK_F_13
Objective	Menggambarkan aktifitas aktor mencetak disposisi surat
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i> 2. Aktor telah mengetahui data disposisi surat yang akan dicetak 3. Aktor telah berada pada halaman Kelola Surat Masuk
Main Flow	1. Aktor memilih menu Kelola Surat Masuk 2. Aktor menekan tombol Cetak Disposisi Surat pada surat yang akan dicetak 3. Aktor mengatur pengaturan printer
Alternative Flow	-
Post-Conditions	Disposisi surat masuk siap untuk dicetak

14. Kelola Surat Keluar (SRS_SIMAK_F_14)

Tabel 4.17 Use Case Scenario Kelola Surat Keluar

Kode Kebutuhan	SRS_SIMAK_F_14
Objective	Menggambarkan aktifitas aktor mengakses menu kelola surat keluar
Aktor	Admin dan User
Pre-Conditions	Aktor telah ter- <i>authentifikasi</i>

Main Flow	1. Aktor memilih menu Kelola Surat Keluar 2. Sistem menampilkan halaman kelola surat keluar
Alternative Flow	-
Post-Conditions	Sistem menampilkan daftar surat keluar

15. Cari Surat Keluar (SRS_SIMAK_F_15)

Tabel 4.18 Use Case Scenario Cari Surat Keluar

Kode Kebutuhan	SRS_SIMAK_F_15
Objective	Menggambarkan aktifitas aktor mencari surat keluar
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i> 2. Aktor telah berada pada halaman Kelola Surat Keluar
Main Flow	1. Aktor memilih menu Kelola Surat Keluar 2. Aktor memasukkan kata kunci pencarian dan/atau tanggal surat yang dicari
Alternative Flow	-
Post-Conditions	Sistem menampilkan daftar surat masuk berdasarkan tanggal atau kata kunci

16. Tambah Surat Keluar (SRS_SIMAK_F_16)

Tabel 4.19 Use Case Scenario Tambah Surat Keluar

Kode Kebutuhan	SRS_SIMAK_F_16
Objective	Menggambarkan aktifitas aktor menambahkan data surat keluar
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih menu Kelola Surat Keluar 2. Aktor menekan tombol Tambah Data 3. Sistem menampilkan form data surat keluar 4. Aktor memasukkan nomor surat keluar 5. Aktor memasukkan tujuan surat keluar 6. Aktor memasukkan isi surat keluar 7. Aktor memasukkan tanggal surat keluar 8. Aktor memasukkan file gambar hasil scan surat keluar 9. Aktor memasukkan keterangan surat keluar 10. Aktor melakukan submit data 11. Sistem menyimpan data surat keluar
Alternative Flow	1. Jika aktor menekan tombol kembali, , sistem akan mengembalikan ke halaman sebelumnya

	2. Jika aktor menekan tombol Clear, sistem akan mengembalikan isi data semua kolom <i>properties</i> surat keluar ke keadaan semula
Post-Conditions	-

17. Edit Surat Keluar (SRS_SIMAK_F_17)

Tabel 4.20 Use Case Scenario Edit Surat Keluar

Kode Kebutuhan	SRS_SIMAK_F_17
Objective	Menggambarkan aktifitas aktor mengedit surat masuk
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>autentifikasi</i> 2. Aktor telah berada pada halaman Kelola Surat keluar
Main Flow	1. Aktor memilih menu Kelola Surat keluar 2. Aktor menekan tombol Edit Data pada surat yang akan diedit 3. Aktor mengubah data pada kolom yang perlu dirubah 4. Sistem menyimpan data surat keluar yang telah diedit
Alternative Flow	1. Jika aktor menekan tombol kembali, sistem akan mengembalikan ke halaman sebelumnya 2. Jika aktor menekan tombol Clear, sistem akan mengembalikan isi data semua kolom <i>properties</i> surat keluar ke keadaan semula
Post-Conditions	-

18. Hapus Surat Keluar (SRS_SIMAK_F_18)

Tabel 4.21 Use Case Scenario Hapus Surat Keluar

Kode Kebutuhan	SRS_SIMAK_F_18
Objective	Menggambarkan aktifitas aktor menghapus data surat keluar
Aktor	Admin dan User
Pre-Conditions	1. Aktor telah ter- <i>autentifikasi</i> 2. Ada data surat keluar
Main Flow	1. Aktor memilih menu Kelola Surat Keluar 2. Aktor menekan tombol 'Hapus Data' pada surat keluar yang akan dihapus 3. Sistem menampilkan konfirmasi penghapusan data 4. Aktor mengkonfirmasi penghapusan data surat keluar 5. Sistem menampilkan informasi bahwa data surat keluar telah terhapus
Alternative Flow	Jika aktor menekan tombol 'Cancel', sistem tidak jadi menghapus surat keluar

Post-Conditions	Data surat keluar telah terhapus
------------------------	----------------------------------

19. Setting Instansi (SRS_SIMAK_F_19)

Tabel 4.22 Use Case Scenario Setting Instansi

Kode Kebutuhan	SRS_SIMAK_F_19
Objective	Menggambarkan aktifitas aktor mengatur data dari instansi Kementerian Agama Kota Malang
Aktor	Admin
Pre-Conditions	Aktor telah ter- <i>authentifikasi</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu 'Pengaturan Instansi' pada menu <i>dropdown</i> 'Pengaturan' 2. Aktor memasukkan nama instansi 3. Aktor memasukkan alamat instansi 4. Aktor memasukkan nama kepala instansi 5. Aktor memasukkan NIP kepala instansi 6. Aktor memasukkan logo instansi 7. Aktor men-<i>submit</i> data 8. Sistem menyimpan data instansi
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol kembali, , sistem akan mengembalikan ke halaman sebelumnya 2. Jika aktor menekan tombol Clear, sistem akan mengembalikan isi data semua kolom <i>properties</i> surat keluar ke keadaan semula
Post-Conditions	-

20. Manage User (SRS_SIMAK_F_20)

Tabel 4.23 Use Case Scenario Manage User

Kode Kebutuhan	SRS_SIMAK_F_20
Objective	Menggambarkan aktifitas aktor dalam manajemen User pada sistem
Aktor	Admin
Pre-Conditions	Aktor telah ter- <i>authentifikasi</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu 'Manajemen User' pada menu <i>dropdown</i> 'Pengaturan' 2. Sistem menampilkan halaman manajemen user
Alternative Flow	-
Post-Conditions	Sistem menampilkan daftar user yang terdaftar pada sistem

21. Tambah User (SRS_SIMAK_F_21)

Tabel 4.24 Use Case Scenario Tambah User

Kode Kebutuhan	SRS_SIMAK_F_21
Objective	Menggambarkan aktifitas aktor menambahkan user baru ke dalam sistem
Aktor	Admin
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i> 2. Aktor telah berada pada halaman manajemen user
Main Flow	1. Aktor memilih tombol 'Tambah User' 2. Sistem menampilkan form data user 3. Aktor memasukkan username user baru 4. Aktor memasukkan password user baru 5. Aktor memasukkan lagi password user baru 6. Aktor memasukkan nama user baru 7. Aktor memasukkan Username user baru 8. Aktor memasukkan NIP user baru 9. Aktor memasukkan jenis dari user baru 10. Aktor men- <i>submit</i> data user baru 11. Sistem menyimpan data user baru
Alternative Flow	1. Jika aktor menekan tombol kembali, sistem akan mengembalikan ke halaman sebelumnya 2. Jika aktor menekan tombol Clear, sistem akan mengembalikan isi data semua kolom <i>properties</i> surat keluar ke keadaan semula
Post-Conditions	Sistem menampilkan <i>list</i> user

22. Edit User (SRS_SIMAK_F_22)

Tabel 4.25 Use Case Scenario Edit User

Kode Kebutuhan	SRS_SIMAK_F_22
Objective	Menggambarkan aktifitas aktor mengedit data user
Aktor	Admin
Pre-Conditions	Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih submenu 'Manajemen user' pada menu <i>dropdown</i> 'Pengaturan' 2. Aktor menekan tombol 'Edit' pada user yang akan diedit 3. Sistem menampilkan form data user 4. Aktor melakukan perubahan data pada kolom yang perlu dirubah 5. Aktor men- <i>submit</i> data user 6. Sistem menyimpan data user

Alternative Flow	1. Jika aktor menekan tombol kembali, , sistem akan mengembalikan ke halaman sebelumnya 2. Jika aktor menekan tombol Clear, sistem akan mengembalikan isi data semua kolom <i>properties</i> surat keluar ke keadaan semula
Post-Conditions	Sistem menampilkan <i>list</i> user

23. Hapus User (SRS_SIMAK_F_23)

Tabel 4.26 Use Case Scenario Hapus User

Kode Kebutuhan	SRS_SIMAK_F_23
Objective	Menggambarkan aktifitas aktor menghapus user
Aktor	Admin
Pre-Conditions	1. Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih submenu 'Manajemen user' pada menu <i>dropdown</i> 'Pengaturan' 2. Aktor menekan tombol 'Delete' pada user yang akan dihapus 3. Sistem menampilkan konfirmasi penghapusan user 4. Aktor mengkonfirmasi penghapusan user 5. Sistem menghapus user
Alternative Flow	Jika aktor menekan tombol 'Cancel' pada saat konfirmasi penghapusan, sistem akan mengembalikan ke halaman manajemen user
Post-Conditions	Sistem menampilkan <i>list</i> user

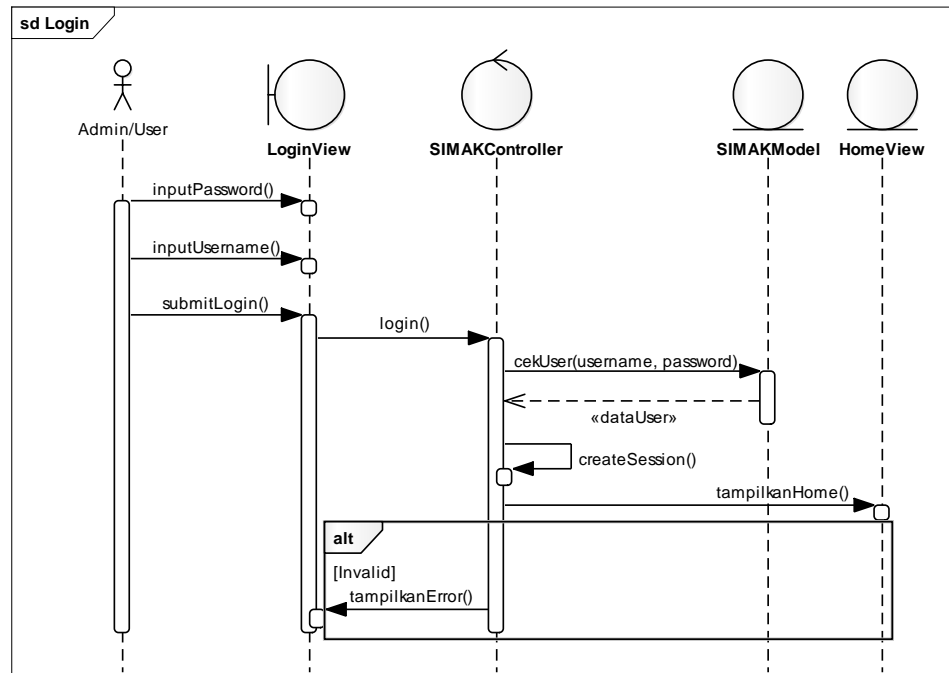
24. Cari User (SRS_SIMAK_F_24)

Tabel 4.27 Use Case Scenario Cari User

Kode Kebutuhan	SRS_SIMAK_F_24
Objective	Menggambarkan aktifitas aktor mencari user
Aktor	Admin
Pre-Conditions	Aktor telah ter- <i>authentifikasi</i>
Main Flow	1. Aktor memilih menu 'pengaturan' kemudian memilih submenu 'manajemen user' 2. Aktor memasukkan kata kunci pencarian user yang dicari 3. Sistem menampilkan hasil pencarian user
Alternative Flow	-
Post-Conditions	Sistem menampilkan daftar hasil pencarian user

4.2.3. Sequence Diagram

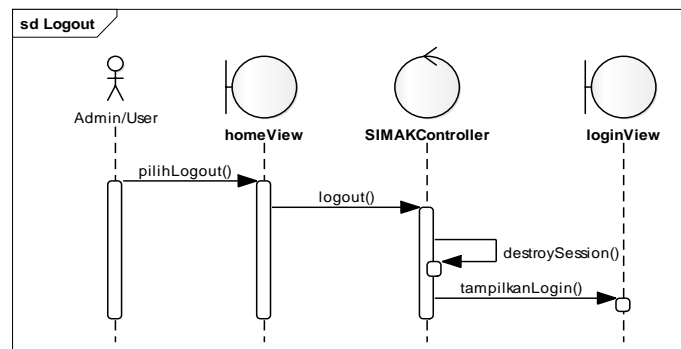
1. Login (SRS-F-SIMAK-01)



Gambar 4.2 Sequence Diagram Login

Pada Gambar 4.2 yang menggambarkan sequence diagram dari proses login yang dilakukan oleh aktor admin dan user, pertama aktor menginputkan password, kemudian menginputkan *username* yang kemudian ditujukan kepada boundary LoginView. Kemudian method *login()* pada SIMAKController akan mencocokkan *username* dan password yang diinputkan oleh user pada data yang terdapat pada *database*. Jika data valid maka method *createSession()* dipanggil dan mengarahkan aktor ke boundary HomeView. Dan bila data tidak valid maka user mendapatkan pesan yang menyatakan bahwa login telah gagal dilakukan.

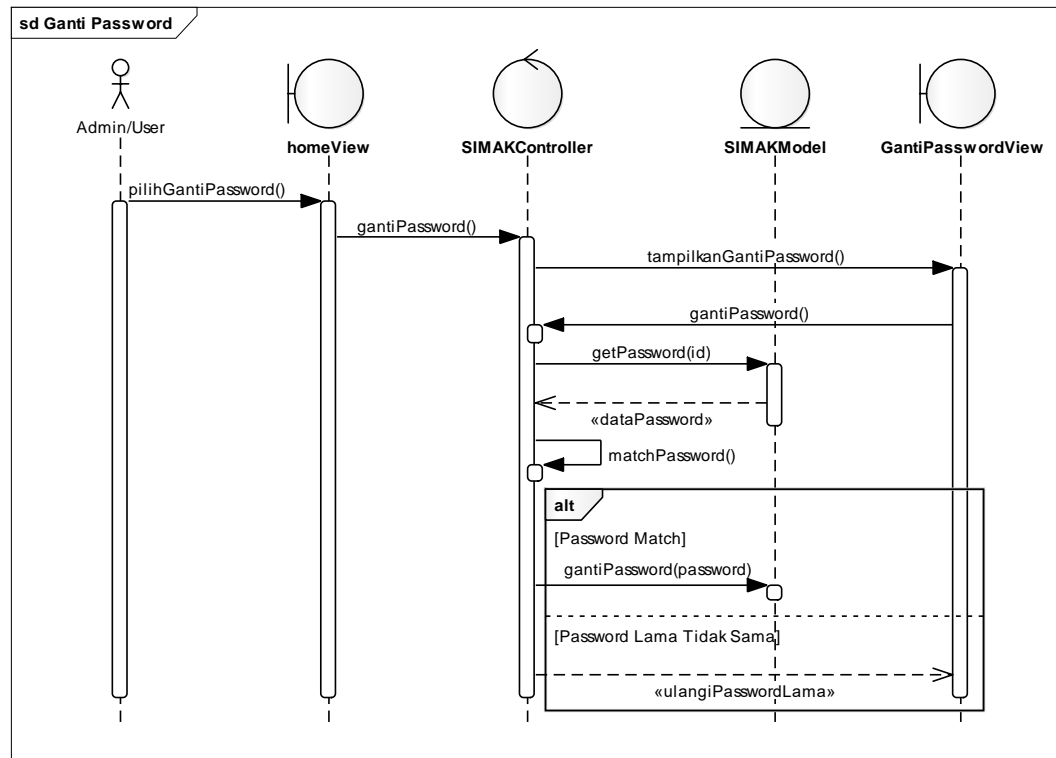
2. Logout (SRS-F-SIMAK-02)



Gambar 4.3 Sequence Diagram Logout

Pada Gambar 4.3 yang menggambarkan sequence diagram dari proses logout yang dilakukan oleh aktor admin dan user, pertama aktor memilih submenu 'Logout' yang kemudian ditujukan kepada boundary homeView. Kemudian method logout() pada SIMAKController akan menghapus *session* yang diset didalam sistem. Kemudian sistem mengarahakan aktor ke halaman login.

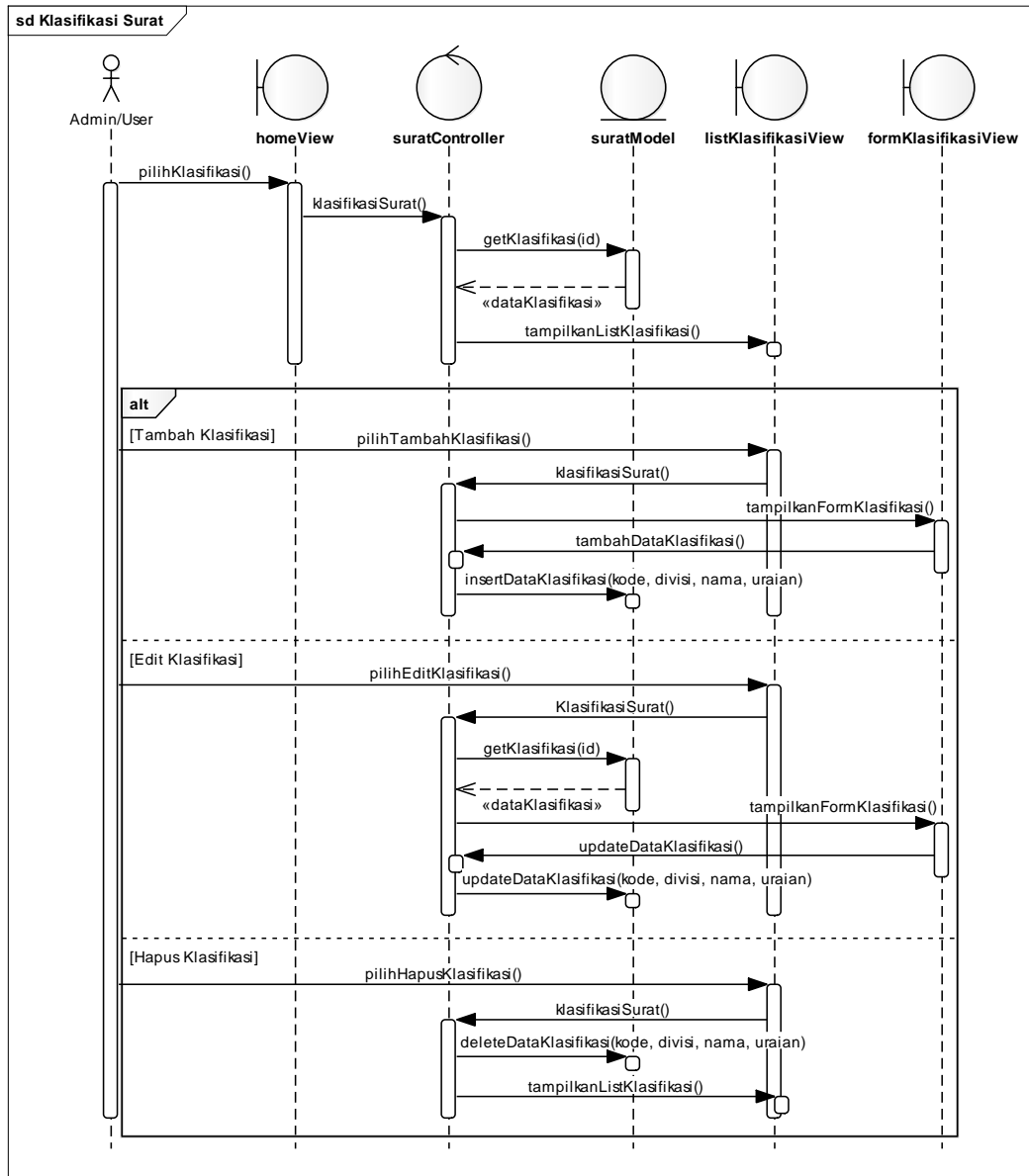
3. Ganti Password (SRS-F-SIMAK-03)



Gambar 4.4 Sequence Diagram Ganti Password

Pada Gambar 4.4 yang menggambarkan sequence diagram dari proses mengganti password yang dilakukan oleh aktor admin dan user, pertama aktor memilih submenu 'Ganti Password' yang kemudian ditujukan kepada boundary homeView. Kemudian method gantiPassword() pada SIMAKController dipanggil, yang selanjutnya akan memanggil method tampilkanGantiPassword() pada GantiPasswordView, kemudian sistem menampilkan form ganti password dan aktor memasukkan data password lama dan password baru. Bila password lama benar maka sistem memanggil method gantiPassword(password) untuk mengganti password dengan password baru. Bila password lama tidak benar maka sistem meminta untuk memasukkan pasword lama kembali.

4. Klasifikasi Surat (SRS-F-SIMAK-04)



Gambar 4.5 Sequence Diagram Klasifikasi Surat

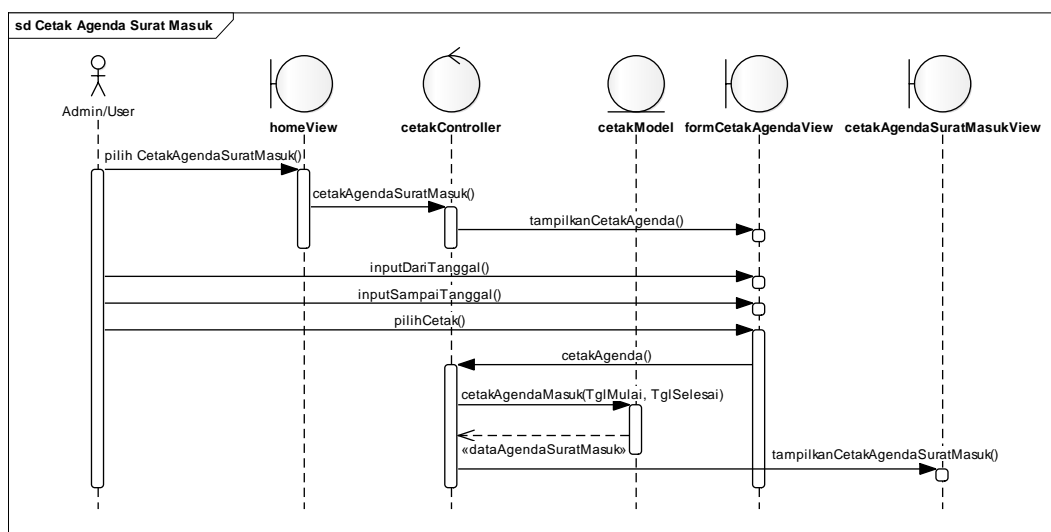
Pada Gambar 4.5 yang menggambarkan sequence diagram dari proses klasifikasi surat yang dilakukan oleh aktor admin dan user, pertama aktor memilih menu 'Klasifikasi Surat' yang kemudian ditujukan kepada boundary homeView. Kemudian method klasifikasiSurat() pada suratController, yang selanjutnya akan memanggil method getKlasifikasi(id) pada suratModel, selanjutnya sistem menampilkan halaman kelola klasifikasi surat. Selanjutnya jika aktor memilih tombol 'Tambah klasifikasi' kemudian sistem menampilkan form klasifikasi, aktor

mengisi data (kode, divisi, nama, uraian) dan klik tombol 'simpan' untuk menyimpan data klasifikasi baru.

Jika aktor memilih tombol 'Edit klasifikasi' sistem menampilkan form edit klasifikasi surat, mengisi data (kode, divisi, nama, uraian), memilih tombol 'Simpan', Kemudian suratModel memanggil method `updateDataKlasifikasi(kode, divisi, nama, uraian)` untuk memasukkan data kedalam *database*.

Jika aktor memilih tombol 'Hapus Klasifikasi' pada klasifikasi surat akan dihapus, yang kemudian ditujukan kepada boundary `listKlasifikasiView`. Kemudian method `KlasifikasiSurat()` pada `suratController` dipanggil, yang selanjutnya akan memanggil method `dleteDataKlasifikasi(kode, divisi, nama, uraian)` pada `suratModel` untuk menghapus data klasifikasi surat dari *database*. Kemudian sistem menampilkan halaman kelola klasifikasi surat.

5. Cetak Agenda Surat Masuk (SRS-F-SIMAK-05)

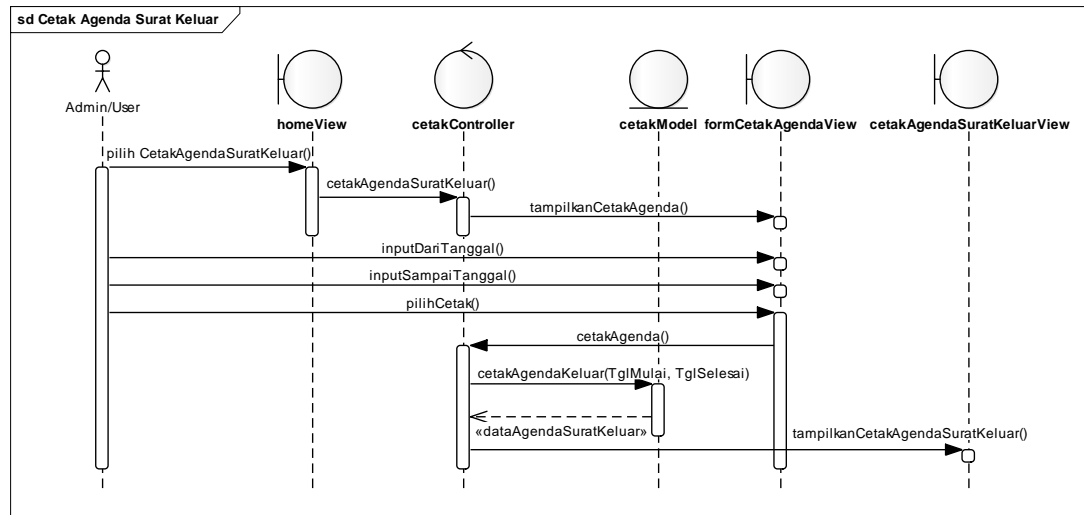


Gambar 4.6 Sequence Diagram Cetak Agenda Surat Masuk

Pada Gambar 4.6 yang menggambarkan sequence diagram dari proses mencetak agenda surat masuk yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat masuk, kemudian aktor memilih submenu 'Cetak Agenda Surat Masuk', yang kemudian ditujukan kepada boundary `homeView`. Kemudian method `cetakAgendaSuratMasuk()` pada `cetakController` dipanggil, yang selanjutnya akan memanggil method `tampilkanCetakAgenda()` untuk menampilkan form cetak agenda surat masuk, selanjutnya aktor mengisikan

tanggal awal agenda dicetak, tanggal akhir agenda dicetak, kemudian menekan tombol 'Cetak'. Kemudian cetakModel memanggil method cetakAgendaMasuk(TglMulai, TglSelesai) untuk mengambil data surat yang akan dicetak, kemudian sistem menampilkan halaman Cetak Agenda Surat Masuk kepada aktor untuk selanjutnya dilakukan pencetakan agenda surat masuk.

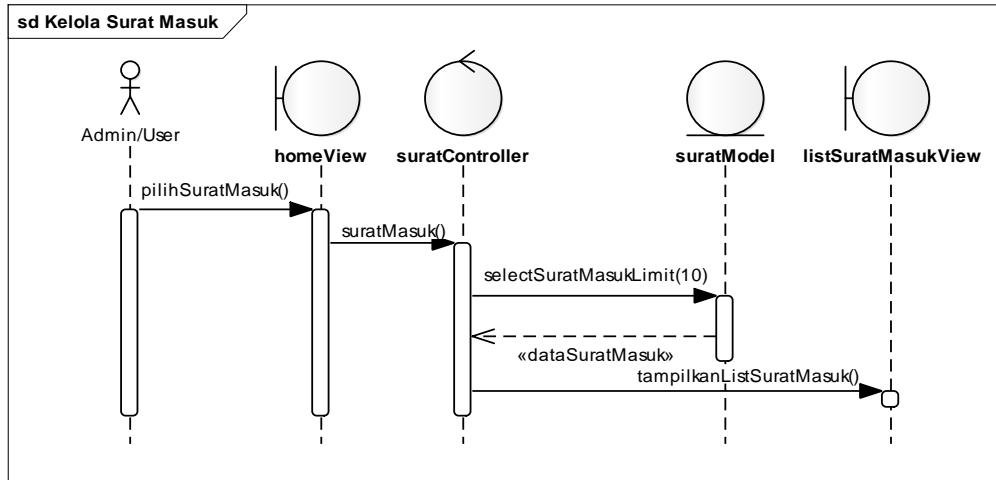
6. Cetak Agenda Surat Keluar (SRS-F-SIMAK-06)



Gambar 4.7 Sequence Diagram Cetak Agenda Surat Keluar

Pada Gambar 4.7 yang menggambarkan sequence diagram dari proses mencetak agenda surat keluar yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat masuk, kemudian aktor memilih submenu 'Cetak Agenda Surat Keluar', yang kemudian ditujukan kepada boundary homeView. Kemudian method cetakAgendaSuratKeluar() pada cetakController dipanggil, yang selanjutnya akan memanggil method tampilkanCetakAgenda() untuk menampilkan form cetak agenda surat keluar, selanjutnya aktor mengisi tanggal awal agenda dicetak, tanggal akhir agenda dicetak, kemudian menekan tombol 'Cetak'. Kemudian cetakModel memanggil method cetakAgendaKeluar(TglMulai, TglSelesai) untuk mengambil data surat yang akan dicetak, kemudian sistem menampilkan halaman Cetak Agenda Surat Keluar kepada aktor untuk selanjutnya dilakukan pencetakan agenda surat keluar.

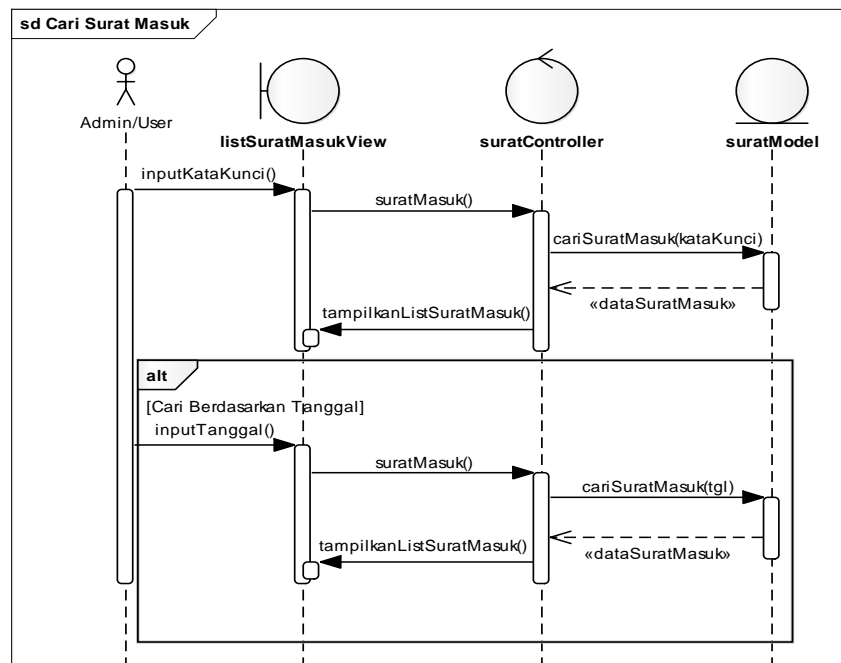
7. Kelola Surat Masuk (SRS-F-SIMAK-07)



Gambar 4.8 Sequence Diagram Kelola Surat Masuk

Pada Gambar 4.8 yang menggambarkan sequence diagram dari proses mengelola surat masuk yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman home, kemudian aktor memilih menu 'Surat Masuk', yang kemudian ditujukan kepada boundary homeView. Kemudian method suratMasuk() pada suratController dipanggil, yang selanjutnya akan memanggil method selectSuratMasuk(limit10) pada suratModel, kemudian sistem akan menampilkan halaman kelola surat masuk kepada aktor.

8. Cari Surat Masuk (SRS-F-SIMAK-08)

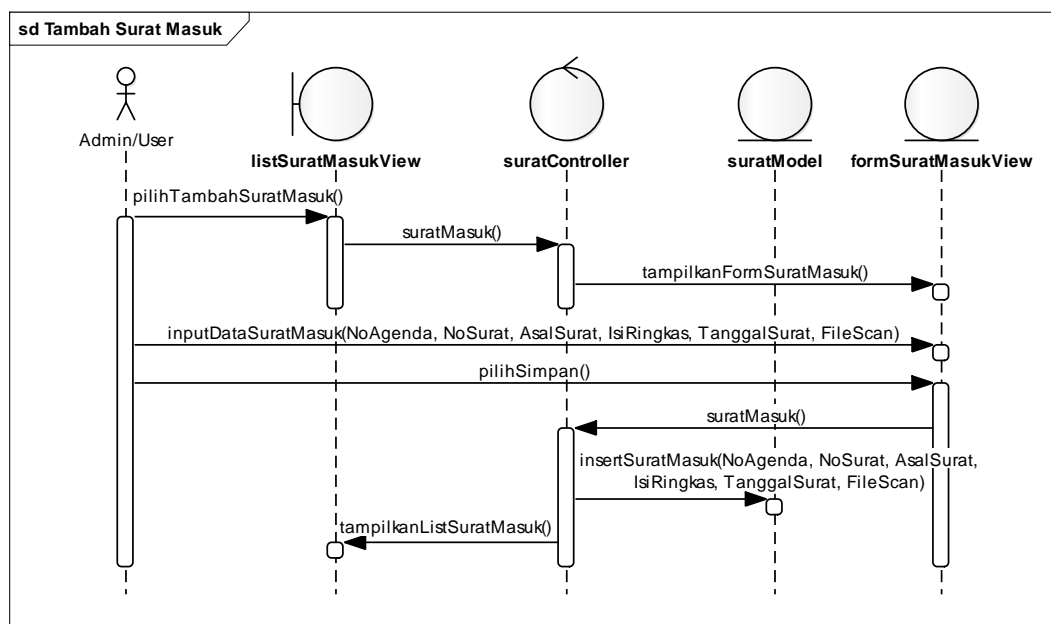


Gambar 4.9 Sequence Diagram Cari Surat Masuk

Pada Gambar 4.9 yang menggambarkan sequence diagram dari proses mencari file surat masuk yang dilakukan oleh aktor admin dan user, pertama aktor menginputkan kata kunci, yang kemudian ditujukan kepada boundary listSuratMasukView. Kemudian method suratMasuk() pada suratController dipanggil, yang selanjutnya akan memanggil method cariSuratMasuk(kataKunci) pada suratModel untuk mendapatkan data surat masuk yang sesuai kata kunci. Kemudian menampilkan hasilnya pada boundary listManageUserView.

Jika aktor mencari berdasarkan tanggal surat langkahnya yaitu pertama aktor menginputkan tanggal surat yang dicari, yang kemudian ditujukan kepada boundary listSuratMasukView. Kemudian method suratMasuk() pada suratController dipanggil, yang selanjutnya akan memanggil method cariSuratMasuk(tgl) pada suratModel untuk mendapatkan data surat masuk yang sesuai tanggal yang dimasukkan. Kemudian menampilkan hasilnya pada boundari listManageUserView.

9. Tambah Surat Masuk (SRS-F-SIMAK-09)

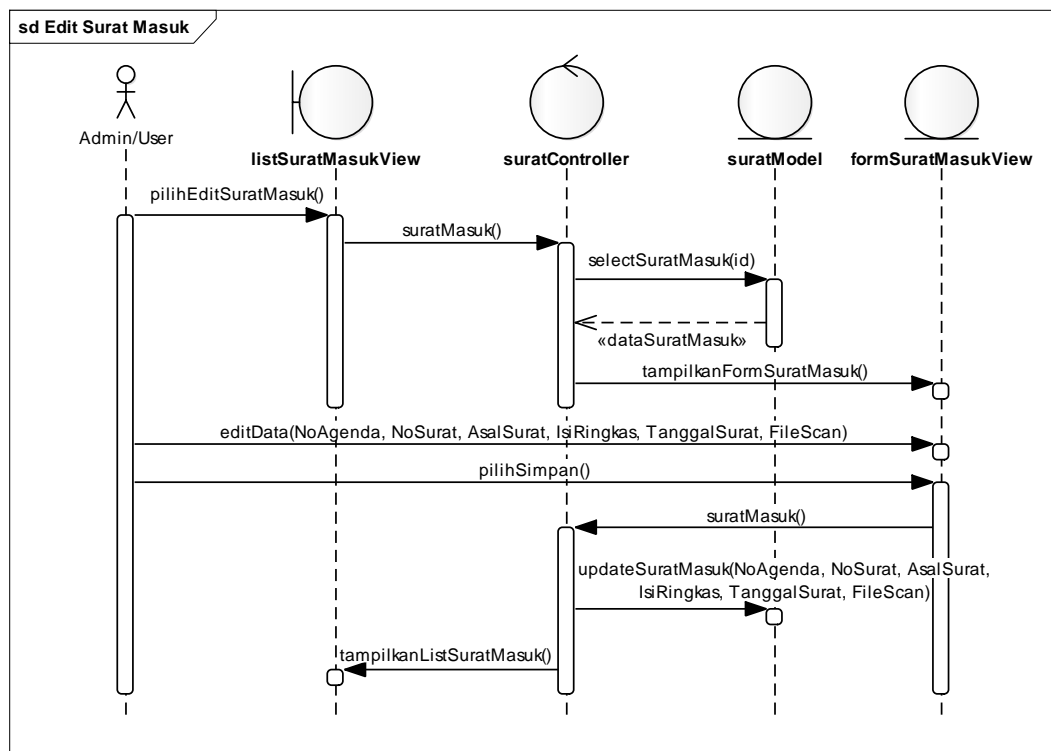


Gambar 4.10 Sequence Diagram Tambah Surat Masuk

Pada Gambar 4.10 menggambarkan sequence diagram dari proses menambah surat masuk yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat masuk, kemudian aktor memilih tombol

‘Tambah Data’, yang kemudian ditujukan kepada boundary `listSuratMasukView`. Kemudian method `suratMasuk()` pada `suratController` dipanggil, yang selanjutnya akan memanggil method `tampilkanFormSuratMasuk()` pada `formSuratMasukView`, kemudian sistem menampilkan form tambah surat masuk dan aktor memasukkan data surat masuk (`NoAgenda`, `noSurat`, `asalSurat`, `isiRingkas`, `tanggalSurat`, `fileScan`) kemudian menekan tombol ‘simpan’. Kemudian `suratModel` memanggil method `insertSuratMasuk(NoAgenda, noSurat, asalSurat, isiRingkas, tanggalSurat, fileScan)` untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan ke halaman kelola surat masuk.

10. Edit Surat Masuk (SRS-F-SIMAK-10)

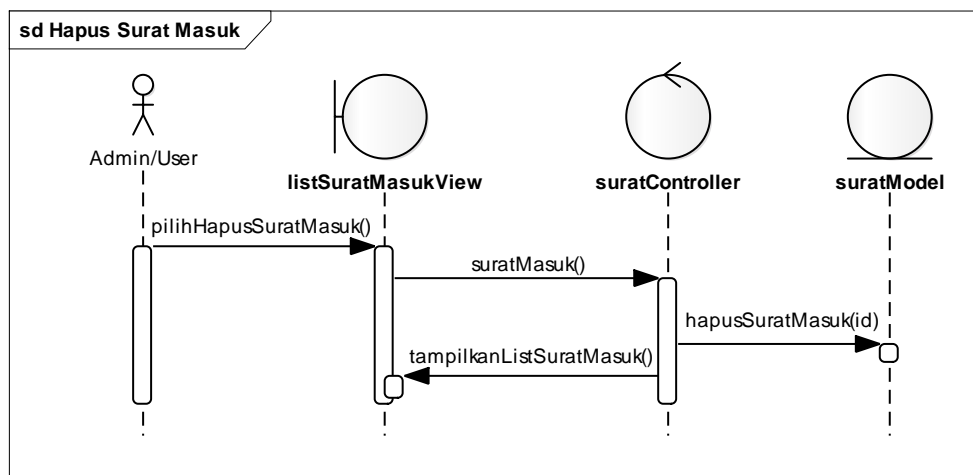


Gambar 4.11 Sequence Diagram Edit Surat Masuk

Pada Gambar 4.11 yang menggambarkan sequence diagram dari proses mengedit surat masuk yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat masuk, kemudian aktor memilih tombol ‘Edit’ pada surat yang akan dirubah datanya, yang kemudian ditujukan kepada boundary `listSuratMasukView`. Kemudian method `suratMasuk()` pada `suratController` dipanggil, yang selanjutnya akan memanggil method

selectSuratMasuk(id) pada suratModel, kemudian sistem menampilkan form edit surat masuk dan aktor memasukkan data surat masuk (NoAgenda, noSurat, asalSurat, isiRingkas, tanggalSurat, fileScan) kemudian menekan tombol 'simpan'. Kemudian suratModel memanggil method updateSuratMasuk(NoAgenda, noSurat, asalSurat, isiRingkas, tanggalSurat, fileScan) untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan ke halaman kelola surat masuk.

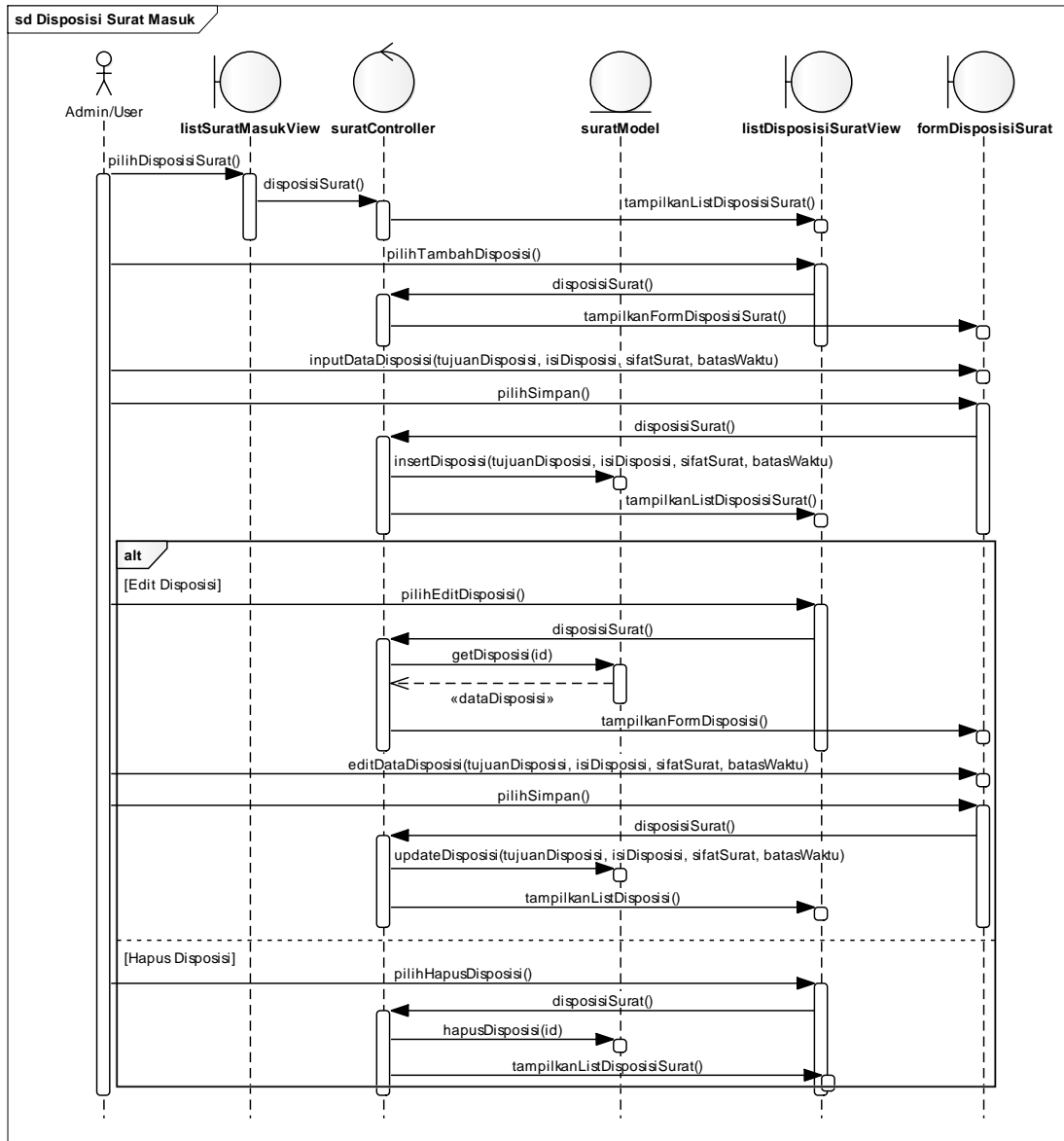
11. Hapus Surat Masuk (SRS-F-SIMAK-11)



Gambar 4.12 Sequence Diagram Hapus Surat Masuk

Pada Gambar 4.12 yang menggambarkan sequence diagram dari proses menghapus surat masuk yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat masuk, kemudian aktor memilih tombol 'hapus' pada surat yang akan dihapus, yang kemudian ditujukan kepada boundary listSuratMasukView. Kemudian method suratMasuk() pada suratController dipanggil, yang selanjutnya akan memanggil method hapusSuratMasuk(id) pada suratModel untuk menghapus data surat keluar dari *database*. Kemudian sistem menampilkan halaman kelola surat keluar.

12. Disposisi Surat Masuk (SRS-F-SIMAK-12)



Gambar 4.13 Sequence Diagram Disposisi Surat Masuk

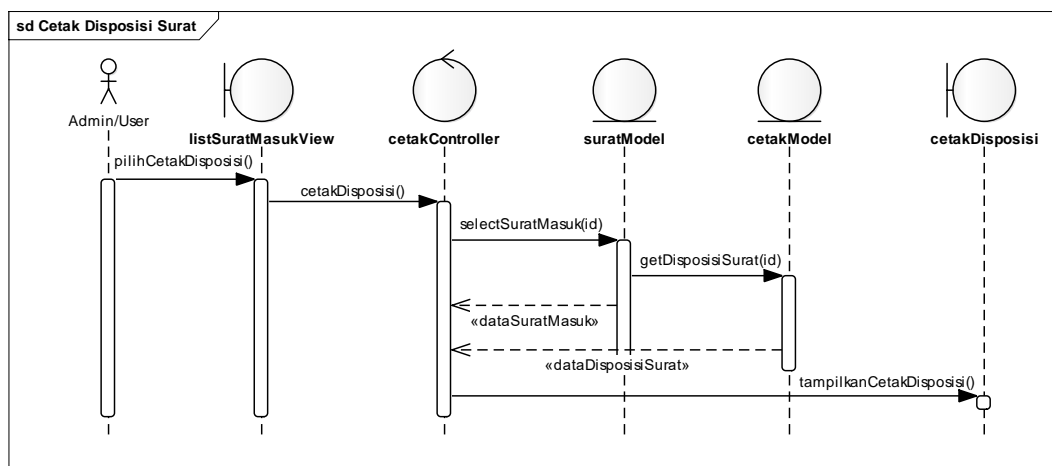
Pada Gambar 4.13 yang menggambarkan sequence diagram dari proses disposisi surat masuk yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat masuk, kemudian aktor memilih tombol 'Disposisi Surat' pada surat yang akan didisposisi, yang kemudian ditujukan kepada boundary `listSuratMasukView`. Kemudian method `disposisiSurat()` pada `suratController`, yang selanjutnya akan memanggil method `tampilkanListDisposisiSurat()` pada `listDisposisiSuratView`, selanjutnya aktor memilih tombol 'Tambah Disposisi' kemudian sistem menampilkan form tambah

disposisi surat, aktor mengisi data (tujuanDisposisi, isiDisposisi, sifatSURat, batasWaktu) kemudian memilih tombol 'Simpan', Kemudian suratModel memanggil method insertDisposisi(tujuanDisposisi, isiDisposisi, sifatSURat, batasWaktu) untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan kehalaman kelola disposisi surat.

Jika aktor memilih tombol 'Edit Disposisi' kemudian sistem menampilkan form edit disposisi surat, kemudian aktor mengisi data (tujuanDisposisi, isiDisposisi, sifatSURat, batasWaktu) kemudian memilih tombol 'Simpan', Kemudian suratModel memanggil method updateDisposisi(tujuanDisposisi, isiDisposisi, sifatSURat, batasWaktu) untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan kehalaman kelola disposisi surat.

Jika aktor memilih tombol 'Hapus Disposisi' pada disposisi yang akan dihapus, yang kemudian ditujukan kepada boundary listDisposisiSuratView. Kemudian method disposisiSurat() pada suratController dipanggil, yang selanjutnya akan memanggil method hapusDisposisi(id) pada suratModel untuk menghapus data disposisi surat masuk dari *database*. Kemudian sistem menampilkan halaman kelola disposisi surat.

13. Cetak Disposisi Surat (SRS-F-SIMAK-13)

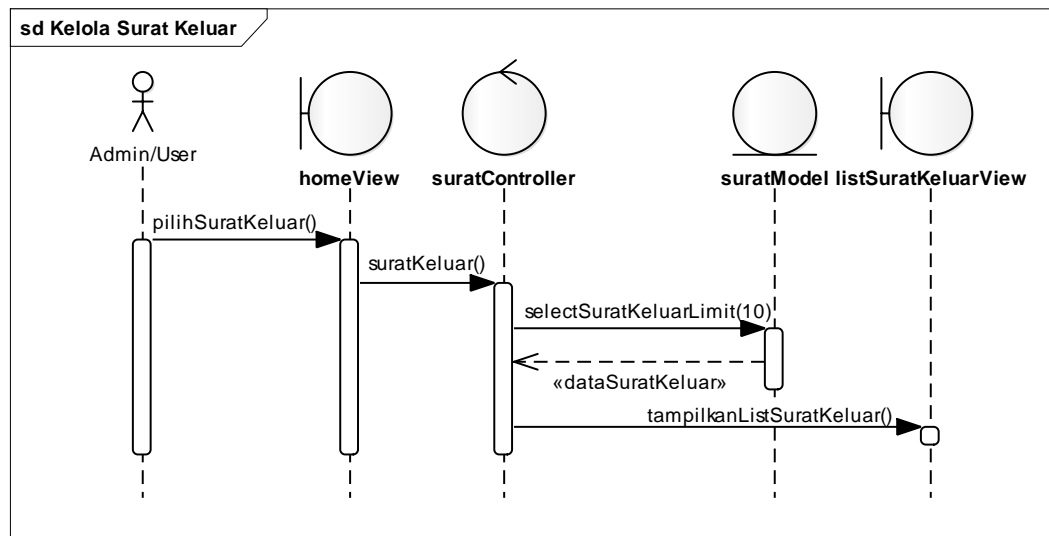


Gambar 4.14 Cetak Disposisi Surat

Pada Gambar 4.14 yang menggambarkan sequence diagram dari proses mencetak disposisi surat yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat masuk, kemudian aktor memilih tombol

‘Cetak Disposisi’ pada disposisi surat yang akan dicetak, yang kemudian ditujukan kepada boundary listSuratMasukView. Kemudian method cetakDisposisi() pada cetakController dipanggil, yang selanjutnya akan memanggil method selectSuratMasuk(id) pada suratModel, selanjutnya memanggil method getDisposisiSurat(id) untuk mengambil data surat yang akan dicetak, kemudian sistem menampilkan halaman Cetak Disposisi kepada aktor untuk selanjutnya dilakukan pencetakan disposisi surat.

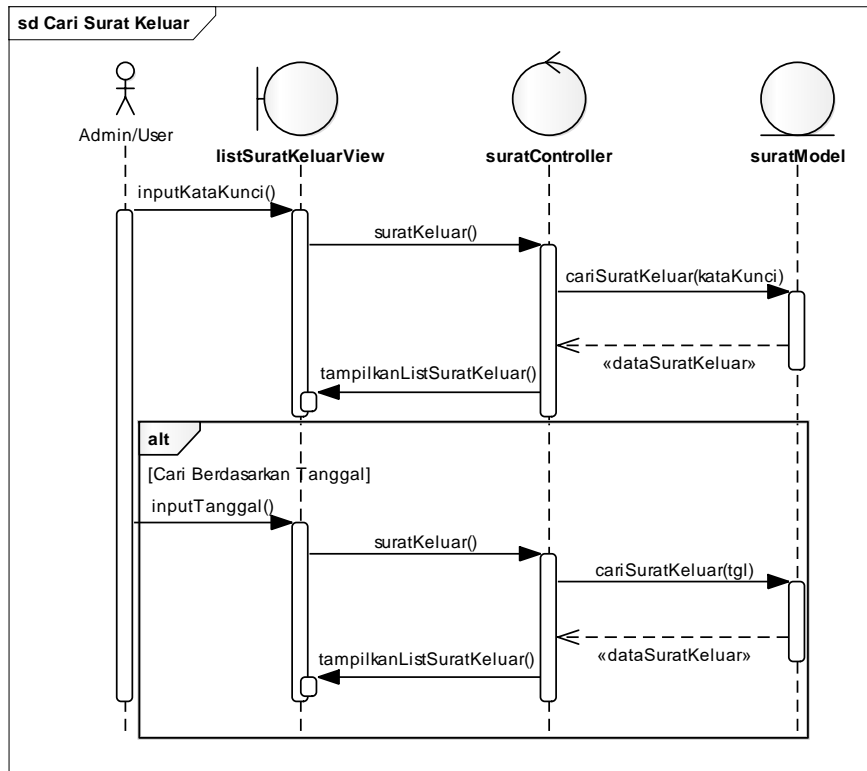
14. Kelola Surat Keluar (SRS-F-SIMAK-14)



Gambar 4.15 Sequence Diagram Kelola Surat Keluar

Pada Gambar 4.15 yang menggambarkan sequence diagram dari proses mengelola surat keluar yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman home, kemudian aktor memilih menu ‘Surat Keluar’, yang kemudian ditujukan kepada boundary homeView. Kemudian method suratKeluar() pada suratController dipanggil, yang selanjutnya akan memanggil method selectSuratKeluar(limit10) pada suratModel, kemudian sistem akan menampilkan halaman kelola surat keluar kepada aktor.

15. Cari Surat Keluar (SRS-F-SIMAK-15)

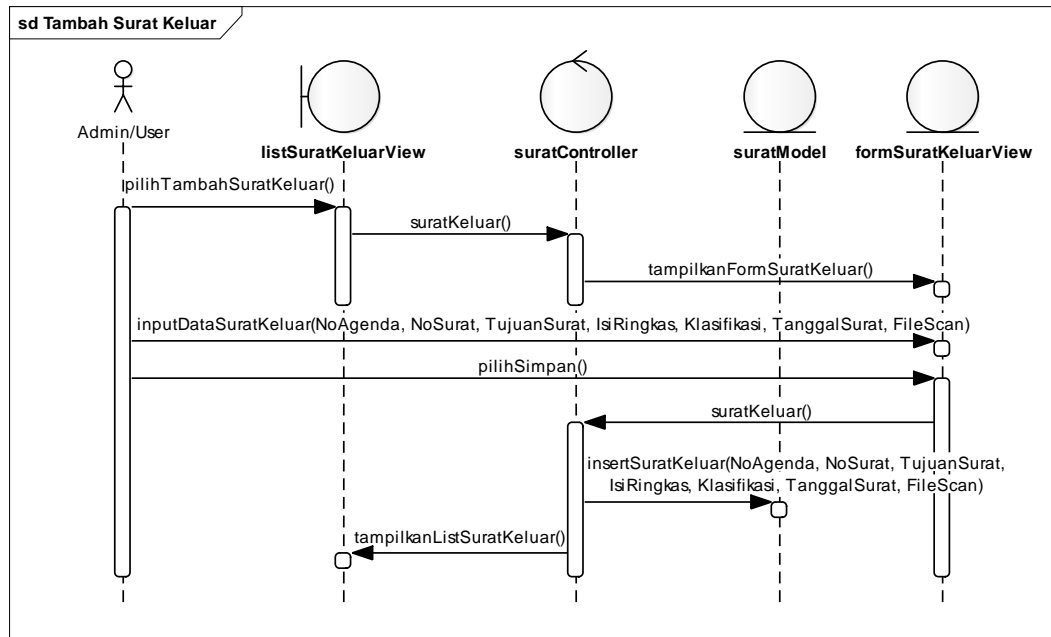


Gambar 4.16 Cari Surat Keluar

Pada Gambar 4.16 yang menggambarkan sequence diagram dari proses mencari file surat keluar yang dilakukan oleh aktor admin dan user, pertama aktor menginputkan kata kunci, yang kemudian ditujukan kepada boundary listSuratKeluarView. Kemudian method suratKeluar() pada suratController dipanggil, yang selanjutnya akan memanggil method cariSuratKeluar(kataKunci) pada suratModel untuk mendapatkan data surat keluar yang sesuai kata kunci. Kemudian menampilkan hasilnya pada boundary listManageUserView.

Jika aktor mencari berdasarkan tanggal surat langkahnya yaitu pertama aktor menginputkan tanggal surat yang dicari, yang kemudian ditujukan kepada boundary listSuratKeluarView. Kemudian method suratKeluar() pada suratController dipanggil, yang selanjutnya akan memanggil method cariSuratKeluar(tgl) pada suratModel untuk mendapatkan data surat keluar yang sesuai tanggal yang dimasukkan. Kemudian menampilkan hasilnya pada boundari listManageUserView.

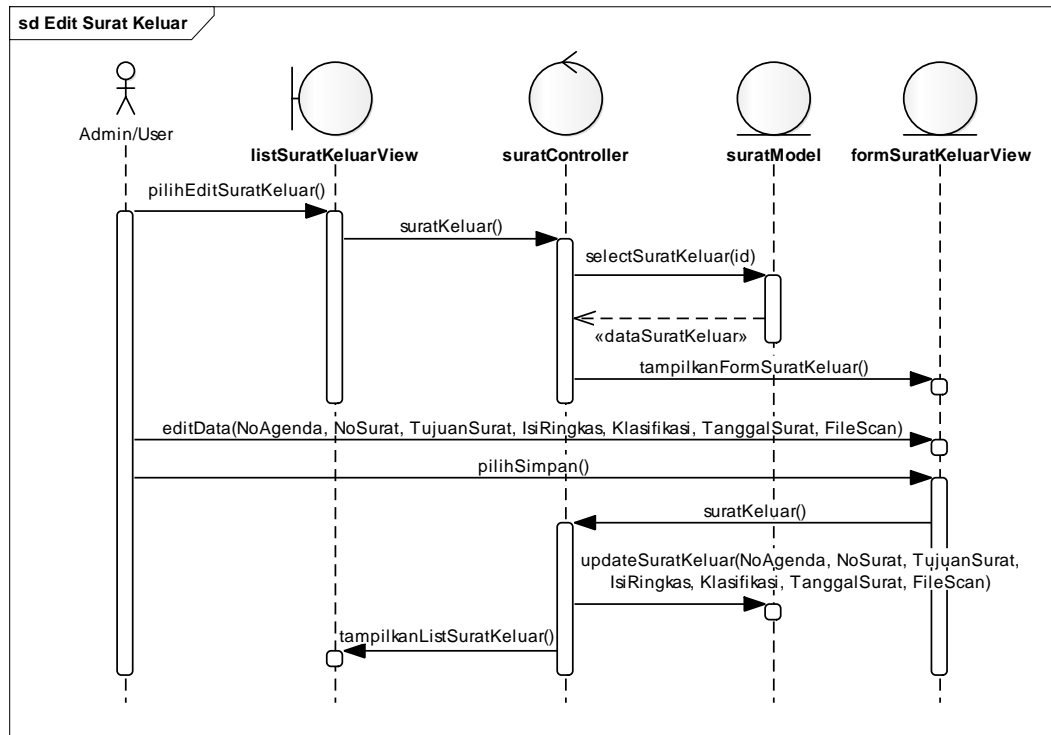
16. Tambah Surat Keluar (SRS-F-SIMAK-16)



Gambar 4.17 Tambah Surat Keluar

Pada Gambar 4.17 yang menggambarkan sequence diagram dari proses menambah surat keluar yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat keluar, kemudian aktor memilih tombol 'Tambah Data', yang kemudian ditujukan kepada boundary listSuratKeluarView. Kemudian method suratKeluar() pada suratController dipanggil, yang selanjutnya akan memanggil method tampilkanFormSuratKeluar() pada formSuratKeluarView, kemudian sistem menampilkan form tambah surat keluar dan aktor memasukkan data surat keluar (NoAgenda, noSurat, tujuanSurat, isiRingkas, klasifikasi, tanggalSurat, fileScan) kemudian menekan tombol 'simpan'. Kemudian suratModel memanggil method insertSuratKeluar(NoAgenda, noSurat, tujuanSurat, isiRingkas, klasifikasi, tanggalSurat, fileScan) untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan kehalaman kelola surat keluar.

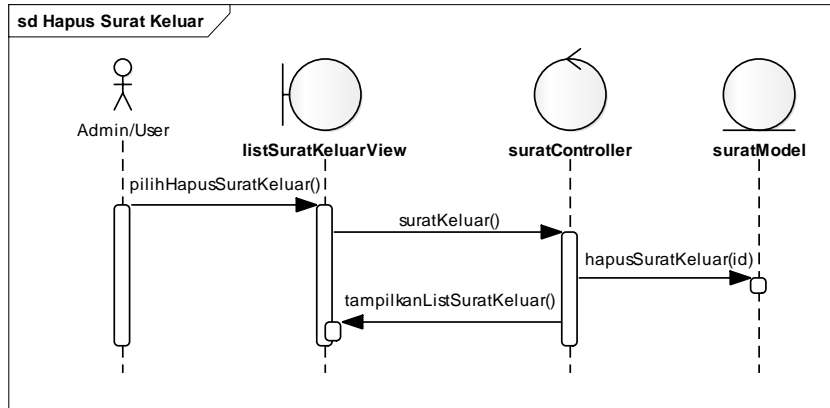
17. Edit Surat Keluar (SRS-F-SIMAK-17)



Gambar 4.18 Edit Surat Keluar

Pada Gambar 4.18 yang menggambarkan sequence diagram dari proses mengedit surat keluar yang dilakukan oleh aktor admin dan user, pertama aktor telah berada pada halaman kelola surat keluar, kemudian aktor memilih tombol 'Edit' pada surat yang akan dirubah datanya, yang kemudian ditujukan kepada boundary `listSuratKeluarView`. Kemudian method `suratKeluar()` pada `suratController` dipanggil, yang selanjutnya akan memanggil method `selectSuratKeluar(id)` pada `suratModel`, kemudian sistem menampilkan form edit surat keluar dan aktor memasukkan data surat keluar (NoAgenda, noSurat, tujuanSurat, isiRingkas, klasifikasi, tanggalSurat, fileScan) kemudian menekan tombol 'simpan'. Kemudian `suratModel` memanggil method `updateSuratKeluar(NoAgenda, noSurat, tujuanSurat, isiRingkas, klasifikasi, tanggalSurat, fileScan)` untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan kehalaman kelola surat keluar.

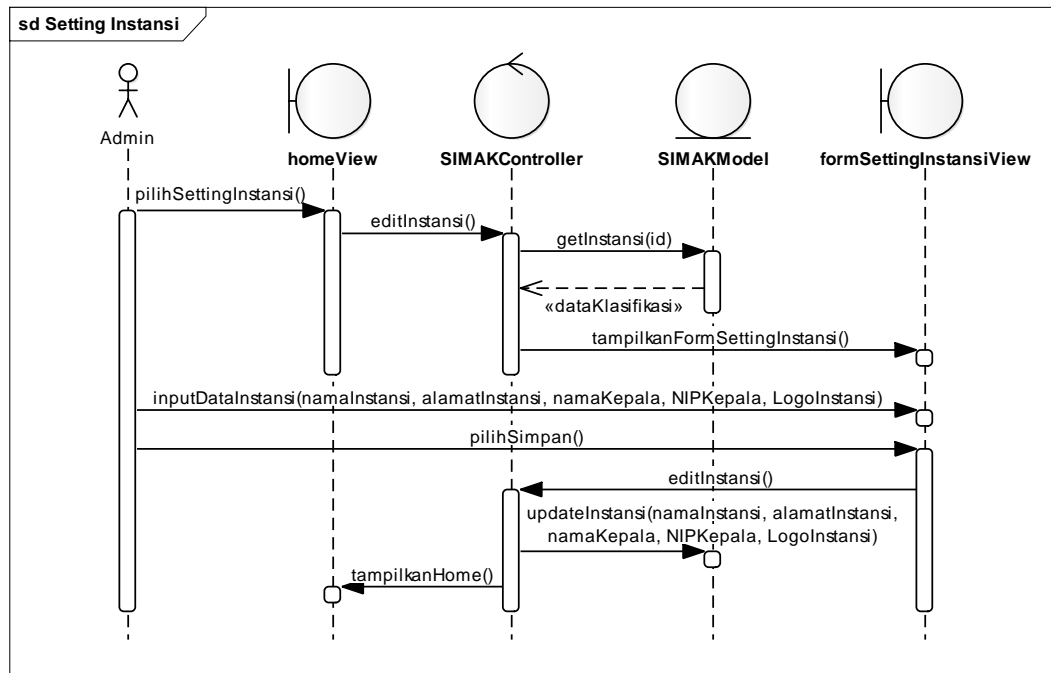
18. Hapus Surat Keluar (SRS-F-SIMAK-18)



Gambar 4.19 Hapus Surat Keluar

Pada Gambar 4.19 yang menggambarkan sequence diagram dari proses menghapus surat keluar yang dilakukan oleh aktor, pertama aktor telah berada pada halaman kelola surat keluar, aktor memilih 'hapus' pada surat yang akan dihapus, yang ditujukan kepada boundary `listSuratKeluarView`. method `suratKeluar()` pada `suratController` dipanggil, yang selanjutnya memanggil method `hapusSuratKeluar(id)` pada `suratModel` untuk menghapus data surat keluar dari *database*. Pada akhirnya sistem menampilkan halaman kelola surat keluar.

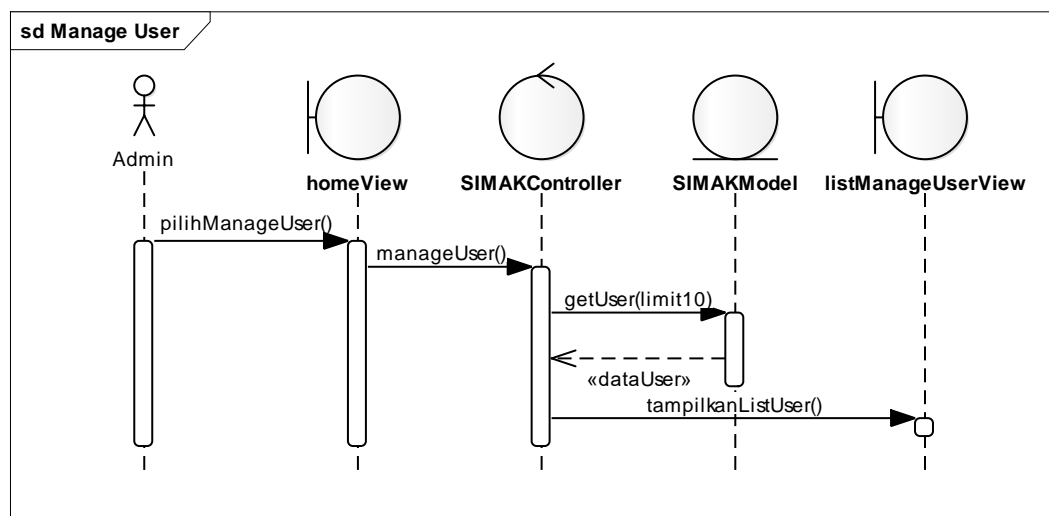
19. Setting Instansi (SRS-F-SIMAK-19)



Gambar 4.20 Setting Instansi

Pada Gambar 4.20 yang menggambarkan sequence diagram dari proses setting instansi yang dilakukan oleh aktor admin, pertama aktor telah berada pada halaman home, kemudian aktor memilih submenu 'Setting Instansi', yang kemudian ditujukan kepada boundary homeView. Kemudian method editInstansi() pada SIMAKController dipanggil, yang selanjutnya akan memanggil method getInstance(id) pada SIMAKModel, kemudian sistem menampilkan form setting instansi dan aktor memasukkan data instansi (namaInstansi, alamatInstansi, namaKepala, NIPKepala, logoInstansi) kemudian menekan tombol 'simpan'. Kemudian SIMAKModel memanggil method updateInstansi(namaInstansi, alamatInstansi, namaKepala, NIPKepala, logoInstansi) untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan kehalaman home.

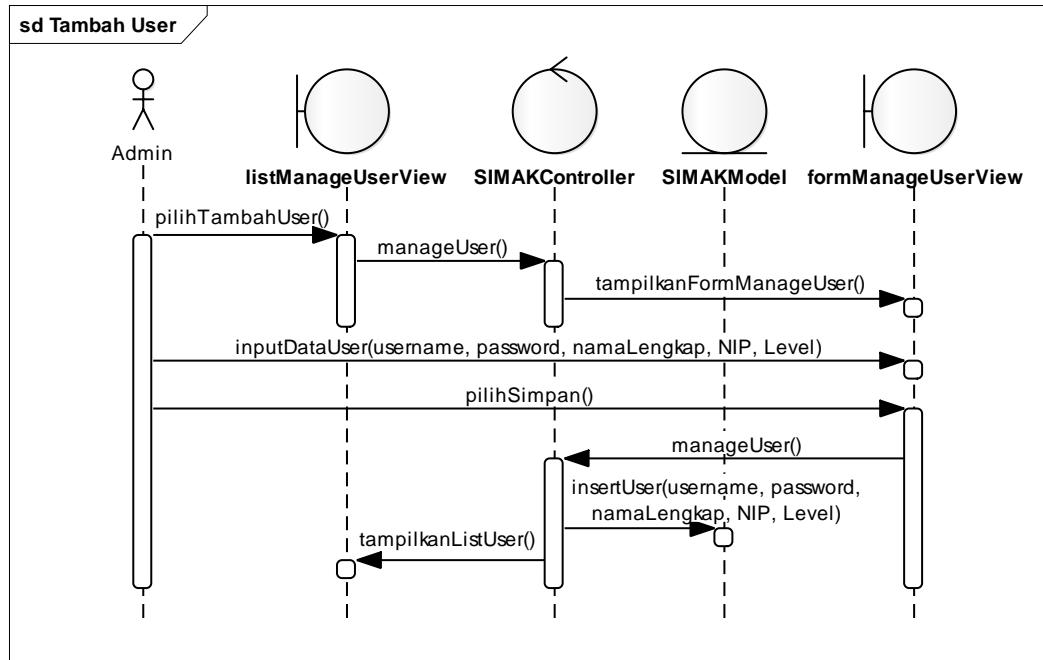
20. Manage User (SRS-F-SIMAK-20)



Gambar 4.21 Manage User

Pada Gambar 4.21 yang menggambarkan sequence diagram dari proses manage user yang dilakukan oleh aktor admin, pertama aktor telah berada pada halaman home, kemudian aktor memilih submenu 'Manage User', yang kemudian ditujukan kepada boundary homeView. Kemudian method manageUser() pada SIMAKController dipanggil, yang selanjutnya akan memanggil method getUser(limit10) pada SIMAKModel, kemudian sistem akan menampilkan halaman manage user kepada aktor.

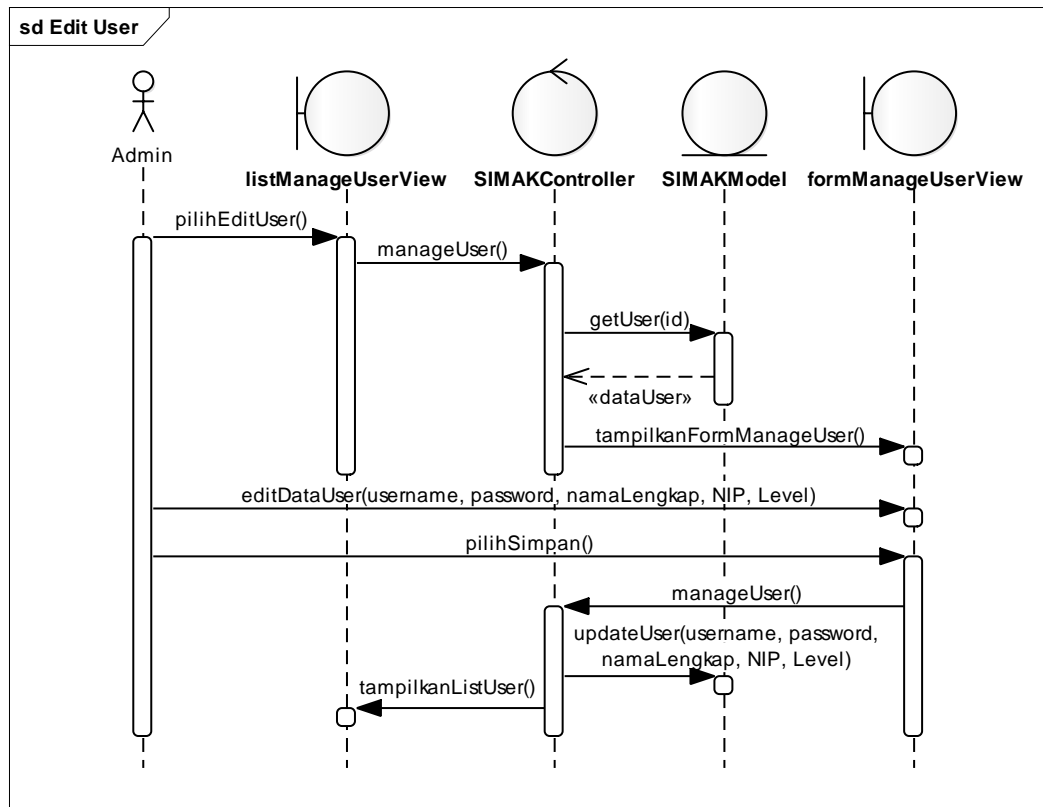
21. Tambah User (SRS-F-SIMAK-21)



Gambar 4.22 Tambah User

Pada Gambar 4.22 yang menggambarkan sequence diagram dari proses menambah user yang dilakukan oleh aktor admin, pertama aktor telah berada pada halaman kelola user, kemudian aktor memilih tombol ‘tambah user’, yang kemudian ditujukan kepada boundary `listManageUserView`. Kemudian method `manageUser()` pada `SIMAKController` dipanggil, yang selanjutnya akan memanggil method `tampilkanFormManageUser()` pada boundary `formManageUserView`, kemudian aktor memasukkan data user baru (username, password, namaLengkap, NIP, dan Level) kemudian menekan tombol ‘simpan’. Kemudian `SIMAKModel` memanggil method `insertUser(username, password, namaLengkap, NIP, Level)` untuk memasukkan data kedalam *database*. Dan kemudian aktor diarahkan kehalaman kelola user.

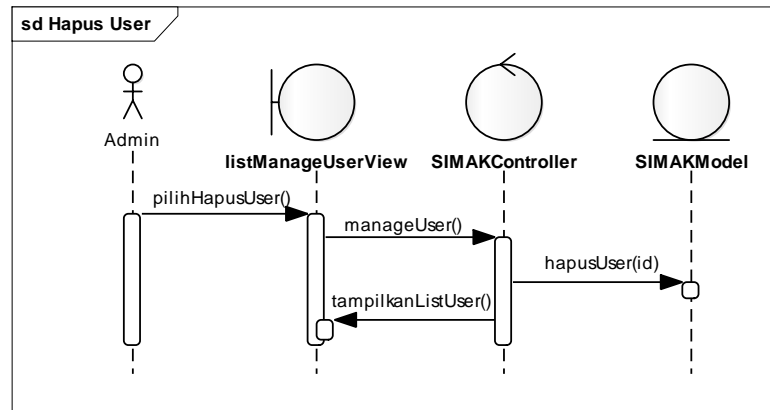
22. Edit User (SRS-F-SIMAK-22)



Gambar 4.23 Edit User

Pada Gambar 4.23 menggambarkan sequence diagram dari proses mengedit data user yang dilakukan oleh aktor admin, pertama aktor telah berada pada halaman kelola user, kemudian aktor memilih tombol edit pada user yang akan diganti *properties*-nya, yang kemudian ditujukan kepada boundary `listManageUserView`. Kemudian method `manageUser()` pada `SIMAKController` dipanggil, yang selanjutnya akan memanggil method `getUser(id)` pada `SIMAKModel` untuk mengambil data data user dari *database*. Kemudian `SIMAKController` menampilkan data user pada boundary `formManageUserView`. Setelah aktor mengedit data dan menekan tombol simpan, maka method `manageUser()` dipanggil yang kemudian memanggil method `update(username, password, namaLengkap, NIP, Level)` pada `SIMAKModel` untuk menyimpan hasil perubahan data. Selanjutnya sistem mengarahkan aktor pada halaman kelola user lagi.

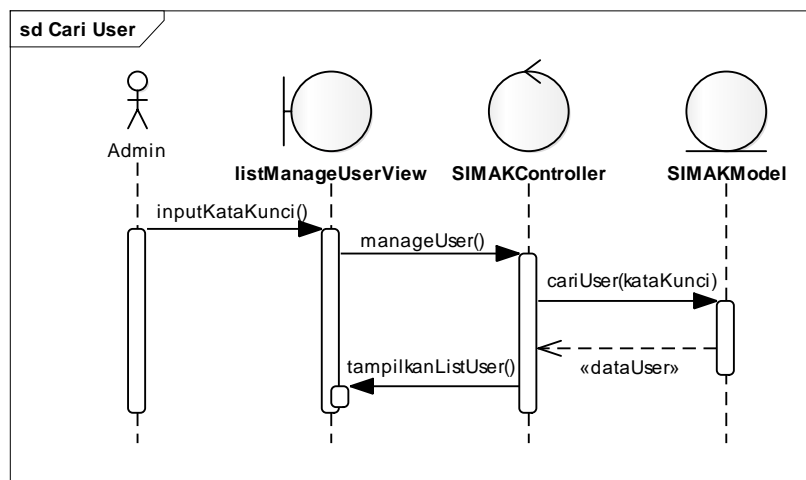
23. Hapus User (SRS-F-SIMAK-23)



Gambar 4.24 Hapus User

Pada Gambar 4.24 yang menggambarkan sequence diagram dari proses menghapus user yang dilakukan oleh aktor admin, pertama aktor telah berada pada halaman kelola user, kemudian aktor memilih tombol hapus pada user yang akan dihapus, yang kemudian ditujukan kepada boundary listManageUserView. Kemudian method `manageUser()` pada `SIMAKController` dipanggil, yang selanjutnya akan memanggil method `hapusUser(id)` pada `SIMAKModel` untuk menghapus data user dari *database*. Kemudian menampilkan hasilnya pada boundary listManageUserView.

24. Cari User (SRS-F-SIMAK-24)



Gambar 4.25 Cari User

Pada Gambar 4.25 yang menggambarkan sequence diagram dari proses mencari user yang dilakukan oleh aktor admin, pertama aktor menginputkan kata

4.2.4. Class Diagram



Pada diagram kelas SIMAK terdapat 3 macam kelas yaitu kelas model, view, dan controller (MVC) karena kami menggunakan framework CodeIgniter dalam implementasinya. Dengan MVC maka kelas-kelas controller sebagai kelas pusat dari dan menuju kelas-kelas view serta kelas-kelas model.

Semua kelas-kelas view dan kelas-kelas model mempunyai multiplicity one to one (1 to 1) karena memang masing-masing komponen mempunyai bentuk riil sebanyak satu buah kelas (satu file) untuk masing-masing kelas pada implementasi nanti. Penggunaan framework CodeIgniter mengharuskan setiap kelas controller meng-extend kelas CI_Controller dan kelas model meng-extend kelas CI_Model.

4.3. Perancangan Sistem

Pada perancangan sistem dilakukan perancangan komponen, data, dan perancangan desain antarmuka sistem.

4.3.1. Perancangan Komponen

1. Klasifikasi Surat

START

Ambil Variabel post(id_post)

Ambil Variabel post(kode)

Ambil Variabel post(divisi)

Ambil Variabel post(nama)

Ambil Variabel post(uraian)

Ambil Variabel post(cari)

Tampilkan view header

IF (tujuan == hapus)

 Panggil model delete_klasifikasi_surat_id

ELSE IF (tujuan == cari)

 Panggil model cari_klasifikasi_surat

 Tampilkan view list_klas_surat

ELSE IF (tujuan == tambah)

 Tampilkan view form_klas_surat

 Panggil model insert_data_klasifikasi

ELSE IF (tujuan == edit)

 Panggil model get_data_klasifikasi

```

    Tampilkan view form_klas_surat
    Panggil model edit_data_klasifikasi
ELSE
    Panggil model get_data_klasifikasi
    Tampilkan view list_klas_surat

Tampilkan data judul Klasifikasi Surat
END

```

2. Kelola Surat Masuk

```

START
Ambil Variabel post(id_post)
Ambil Variabel post(no_agenda)
Ambil Variabel post(indeks_berkas)
Ambil Variabel post(kode)
Ambil Variabel post(dari)
Ambil Variabel post(no_surat)
Ambil Variabel post(tgl_surat)
Ambil Variabel post(uraian)
Ambil Variabel post(keterangan)
Ambil Variabel post(cari)
Ambil Variabel post(cari_tgl)

Tampilkan view header
IF (tujuan == hapus)
    Panggil model delete_surat_masuk_id
ELSE IF (tujuan == cari)
    IF (cari == tgl)
        Panggil model cari_surat_masuk_tgl
    ELSE
        Panggil model cari_surat_masuk_key
    Tampilkan view list_surat_masuk
ELSE IF (tujuan == tambah)
    Tampilkan view form_surat_masuk
    Panggil model insert_data_surat_masuk
ELSE IF (tujuan == edit)

```

```

    Panggil model select_surat_masuk_id
    Tampilkan view form_surat_masuk
    Panggil model edit_data_surat_masuk
ELSE
    Panggil model select_surat_masuk
    Tampilkan view list_surat_masuk

Tampilkan data judul Surat Masuk
END

```

3. Kelola Surat Keluar

```

START
Ambil Variabel post(id_post)
Ambil Variabel post(no_agenda)
Ambil Variabel post(kode)
Ambil Variabel post(tujuan)
Ambil Variabel post(no_surat)
Ambil Variabel post(tgl_surat)
Ambil Variabel post(uraian)
Ambil Variabel post(keterangan)
Ambil Variabel post(cari)
Ambil Variabel post(cari_tgl)

Tampilkan view header
IF (tujuan == hapus)
    Panggil model delete_surat_keluar_id
ELSE IF (tujuan == cari)
    IF (cari == tgl)
        Panggil model cari_surat_keluar_tgl
    ELSE
        Panggil model cari_surat_keluar_key
    Tampilkan view list_surat_keluar
ELSE IF (tujuan == tambah)
    Tampilkan view form_surat_keluar
    Panggil model insert_data_surat_keluar
ELSE IF (tujuan == edit)

```



```

    Panggil model select_surat_keluar_id
    Tampilkan view form_surat_keluar
    Panggil model edit_data_surat_keluar
ELSE
    Panggil model select_surat_keluar
    Tampilkan view list_surat_keluar

Tampilkan data judul Surat Keluar
END

```

4. Disposisi Surat

```

START
Ambil Variabel post(id_post)
Ambil Variabel post(id_surat)
Ambil Variabel post(kepada)
Ambil Variabel post(isi_disposisi)
Ambil Variabel post(sifat)
Ambil Variabel post(batas_waktu)
Ambil Variabel post(catatan)

Tampilkan view header
IF (tujuan == hapus)
    Panggil model delete_disposisi_id
ELSE IF (tujuan == tambah)
    Tampilkan view form_disposisi
    Panggil model insert_data_disposisi
ELSE IF (tujuan == edit)
    Panggil model select_disposisi_id
    Tampilkan view form_disposisi
    Panggil model edit_data_disposisi
ELSE
    Panggil model select_disposisi
    Tampilkan view list_disposisi

Tampilkan data judul Disposisi Surat
END

```

5. Cetak Agenda

```
START
Ambil Variabel post(tipe)
Ambil Variabel post(tgl_start)
Ambil Variabel post(tgl_end)

Tampilkan view header
IF (jenis surat == cetak agenda masuk)
    Panggil model cetak_agenda_surat_masuk
    Tampilkan view cetak_agenda_masuk
    Tampilkan data judul Cetak Agenda Surat Masuk
ELSE
    Panggil model cetak_agenda_surat_keluar
    Tampilkan view cetak_agenda_keluar
    Tampilkan data judul Cetak Agenda Surat Keluar
END
```

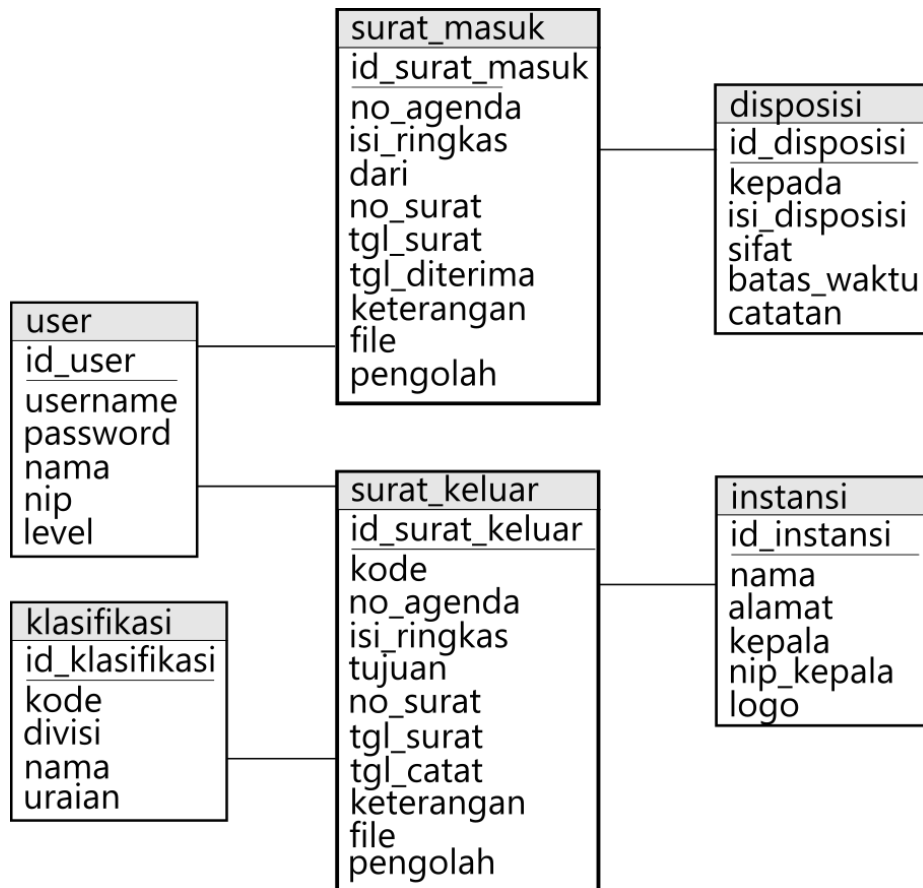
6. Cetak Disposisi

```
START
Ambil Variabel post(tipe)
Ambil Variabel post(tgl_start)
Ambil Variabel post(tgl_end)

Tampilkan view header
Panggil model select_surat_masuk_id
Panggil model get_disposisi_cetak_id
Tampilkan view cetak_disposisi
Tampilkan data judul Cetak Disposisi
END
```

4.3.2. Perancangan Data

Perancangan data diperlukan sebagai gambaran bagaimana data yang diperlukan akan disimpan. Perancangan data direpresentasikan dengan menggunakan *conceptual data model* yang sudah dinormalisasi menjadi *physical data model* pada tahap implementasi.



Gambar 4.27 Conceptual Data Model SIMAK

Penjelasan:

Setiap entitas diperoleh dari pengembangan kelas model pada class diagram.

BAB I SIMAK_model menjadi 2 entitas yaitu user dan instansi

BAB II surat_model dan cetak_model menjadi 4 entitas yaitu surat_masuk, surat_keluar, klasifikasi, serta disposisi

4.3.3. Perancangan Desain Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan desain antarmuka Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang.

1. Halaman Login

Gambar 4.28 Rancangan Halaman Login

Tabel 4.28 Keterangan Rancangan Halaman Login

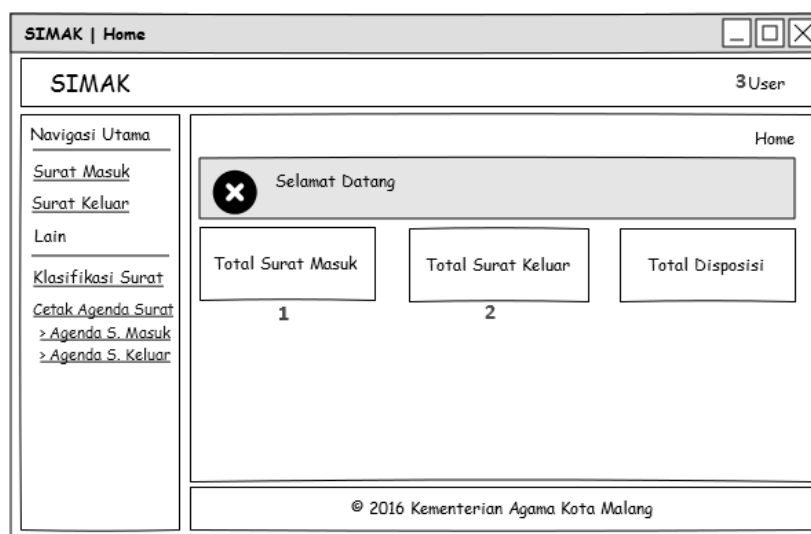
No.	Nama objek	Tipe	Keterangan
1	Username	Text	Menerima input berupa <i>username</i>
2	Password	Password	Menerima input berupa password
3	Sign In	Button	Jika diklik dan benar maka akan menuju halaman home dari aktor

2. Halaman Home

Gambar 4.29 Rancangan Halaman Home Admin

Tabel 4.29 Keterangan Rancangan Halaman Home Admin

No.	Nama objek	Tipe	Keterangan
1	SIMAK	Hyperlink	Jika diklik akan menuju halaman home aktor
2	Surat Masuk	Hyperlink	Jika diklik akan menuju halaman kelola surat masuk
3	Surat Keluar	Hyperlink	Jika diklik akan menuju halaman kelola surat keluar
4	Klasifikasi Surat	Hyperlink	Jika diklik akan menuju halaman klasifikasi surat
5	Cetak Agenda Surat	Dropdown	Jika diklik akan muncul submenu agenda surat masuk
6	Agenda S. Masuk	Hyperlink	Jika diklik akan menuju halaman agenda surat masuk
7	Agenda S. Keluar	Hyperlink	Jika diklik akan menuju halaman agenda surat keluar
8	Pengaturan	Dropdown	Jika diklik akan muncul submenu manage user dan pengaturan instansi
9	Menu Aktor	Dropdown	Jika diklik akan muncul submenu ganti password dan logout
10	Total Surat Masuk	Button	Jika diklik akan menuju halaman kelola surat masuk
11	Total Surat Keluar	Button	Jika diklik akan menuju halaman kelola surat keluar

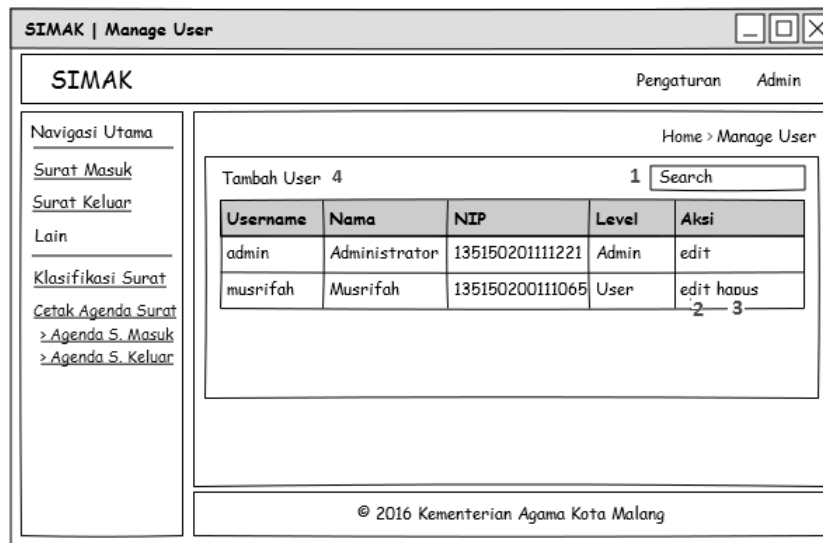


Gambar 4.30 Rancangan Halaman Home User

Tabel 4.30 Keterangan Rancangan Halaman Home User

No.	Nama objek	Tipe	Keterangan
1	Total Surat Masuk	Button	Jika diklik akan menuju halaman kelola surat masuk
2	Total Surat Keluar	Button	Jika diklik akan menuju halaman kelola surat keluar
3	Menu Aktor	Dropdown	Jika diklik akan muncul submenu ganti password dan logout

3. Halaman Manage User



Gambar 4.31 Rancangan Halaman Manage User

Tabel 4.31 Keterangan Rancangan Halaman Manage User

No.	Nama objek	Tipe	Keterangan
1	Search	Text	Menerima input berupa text untuk pencarian user
2	Edit	Button	Jika diklik akan menuju halaman edit user
3	Hapus	Button	Jika diklik akan muncul konfirmasi penghapusan user
4	Tambah User	Button	Jika diklik akan menuju form tambah user

The screenshot shows a web application window titled "SIMAK | Manage User". The main content area is titled "SIMAK" and "Pengaturan Admin". On the left, there is a sidebar with navigation links: "Navigasi Utama", "Surat Masuk", "Surat Keluar", "Lain", "Klasifikasi Surat", "Cetak Agenda Surat", "> Agenda S. Masuk", and "> Agenda S. Keluar". The main content area has a breadcrumb "Home > Manage User" and a form titled "Data User". The form contains the following fields and buttons:

- Username (1)
- Nama Lengkap (4)
- Password Baru (2)
- NIP (5)
- Ulangi Password Baru (3)
- Pilih Level (6) (Dropdown menu)
- Simpan (7)
- Kembali (8)

At the bottom of the window, there is a copyright notice: "© 2016 Kementerian Agama Kota Malang".

Gambar 4.32 Rancangan Halaman Manage User – Tambah data

Tabel 4.32 Keterangan Rancangan Halaman Manage User – Tambah data

No.	Nama objek	Tipe	Keterangan
1	Username	Text	Menerima input berupa username
2	Password Baru	Password	Menerima input berupa password baru
3	Ulangi Password Baru	Password	Menerima input berupa password baru
4	Nama Lengkap	Text	Menerima input berupa nama lengkap
5	NIP	Text	Menerima input berupa NIP
6	Level	Dropdown	Jika diklik akan muncul pilihan level yang harus dipilih
7	Simpan	Button	Jika diklik dan benar maka akan disimpan dan kembali ke halaman kelola user
8	Kembali	Button	Jika diklik akan kembali ke halaman sebelumnya

4. Halaman Setting Instansi

Gambar 4.33 Rancangan Halaman *Setting* Instansi

Tabel 4.33 Keterangan Rancangan Halaman *Setting* Instansi

No.	Nama objek	Tipe	Keterangan
1	Nama Instansi	Text	Menerima input berupa nama instansi
2	Alamat Instansi	Text	Menerima input berupa alamat instansi
3	Nama Kepala	Text	Menerima input berupa nama kepala
4	NIP Kepala	eText	Menerima input berupa NIP kepala
5	Logo Instansi	Button	Jika diklik akan meminta aktor untuk memilih logo dari instansi
6	Simpan	Button	Jika diklik dan benar maka akan kembali ke halaman pengaturan instansi
7	Kembali	Button	Jika diklik akan kembali ke halaman sebelumnya

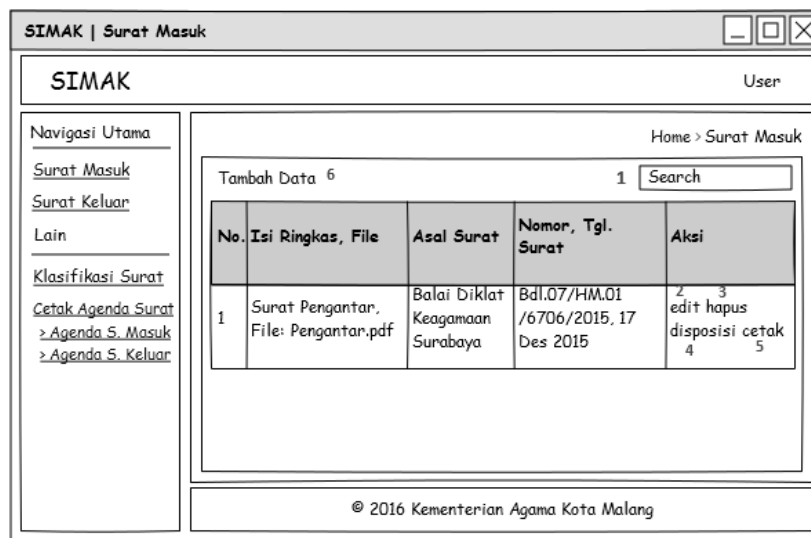
5. Halaman Ganti Password

Gambar 4.34 Rancangan Halaman Ganti Password

Tabel 4.34 Keterangan Rancangan Halaman Ganti Password

No.	Nama objek	Tipe	Keterangan
1	Password Lama	Password	Menerima input berupa password lama
2	Password Baru1	Password	Menerima input berupa password baru1
3	Password Baru2	Password	Menerima input berupa password baru2
4	Simpan	Button	Jika diklik dan benar maka akan kembali ke halaman kelola user
5	Kembali	Button	Jika diklik akan kembali ke halaman sebelumnya

6. Halaman Kelola Surat Masuk



Gambar 4.35 Rancangan Halaman Kelola Surat Masuk

Tabel 4.35 Keterangan Rancangan Halaman Kelola Surat Masuk

No.	Nama objek	Tipe	Keterangan
1	Search	Text	Menerima input berupa text untuk pencarian surat
2	Edit	Button	Jika diklik akan menuju halaman edit surat masuk
3	Hapus	Button	Jika diklik akan muncul konfirmasi penghapusan surat
4	Disposisi	Button	Jika diklik akan menuju halaman disposisi surat
5	Cetak	Button	Jika diklik akan menuju halaman cetak surat
4	Tambah Data	Button	Jika diklik akan menuju form tambah surat masuk

Gambar 4.36 Rancangan Halaman Tambah Surat Masuk

Tabel 4.36 Rancangan Halaman Tambah Surat Masuk

No.	Nama objek	Tipe	Keterangan
1	No. Agenda	Text	Menerima input berupa nomor agenda
2	Nomor Surat	Text	Menerima input berupa nomor surat
3	Asal Surat	Text	Menerima input berupa asal surat
4	Isi Ringkas	Text	Menerima input berupa isi ringkas surat
5	Tanggal Surat	Text	Menerima input berupa tanggal surat
6	File Surat	Button	Jika diklik akan meminta aktor untuk memilih file dari surat
7	Keterangan	Text	Menerima input berupa keterangan surat
8	Simpan	Button	Jika diklik dan benar maka akan kembali ke halaman kelola surat masuk
9	Kembali	Button	Jika diklik akan kembali ke halaman sebelumnya

7. Halaman Disposisi Surat

Gambar 4.37 Rancangan Halaman Disposisi Surat

Tabel 4.37 Keterangan Rancangan Halaman Disposisi Surat

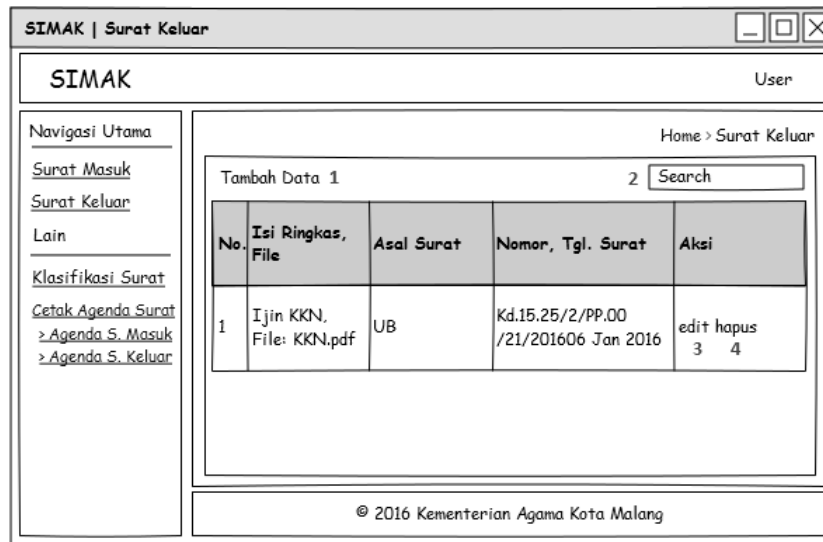
No.	Nama objek	Tipe	Keterangan
1	Tujuan Disposisi	Text	Menerima input berupa tujuan disposisi
2	Isi Disposisi	Text	Menerima input berupa isi disposisi
3	Sifat Surat	Text	Menerima input berupa sifat surat
4	Batas Waktu	Text	Menerima input berupa batas waktu
5	Catatan	Text	Menerima input berupa catatan surat
6	Simpan	Button	Jika diklik dan benar maka akan kembali ke halaman kelola disposisi surat
7	Kembali	Button	Jika diklik akan kembali ke halaman sebelumnya

8. Halaman Cetak Disposisi

Gambar 4.38 Rancangan Halaman Cetak Disposisi

Karena halaman 4.38 hanya dicetak menurut data yang telah ada sebelumnya tanpa perlu inputan dari aktor, maka tidak ada keterangan berupa tabel keterangan rancangan halaman cetak disposisi.

9. Halaman Kelola Surat Keluar



Gambar 4.39 Rancangan Halaman Kelola Surat Keluar

Tabel 4.38 Keterangan Rancangan Halaman Kelola Surat Keluar

No.	Nama objek	Tipe	Keterangan
1	Tambah Data	Button	Jika diklik akan menuju form tambah surat keluar
2	Search	Text	Menerima input berupa text untuk pencarian surat keluar
3	Edit	Button	Jika diklik akan menuju halaman edit surat keluar
4	Hapus	Button	Jika diklik akan muncul konfirmasi penghapusan surat

Gambar 4.40 Rancangan Halaman Tambah Data Surat Keluar

Tabel 4.39 Keterangan Rancangan Halaman Tambah Data Surat Keluar

No.	Nama objek	Tipe	Keterangan
1	No. Agenda	Text	Menerima input berupa nomor agenda
2	Nomor Surat	Text	Menerima input berupa nomor surat
3	Tujuan Surat	Text	Menerima input berupa tujuan surat
4	Isi Ringkas	Text	Menerima input berupa isi ringkas surat
5	Kode Klasifikasi	Dropdown	Jika diklik akan muncul pilihan kode klasifikasi surat
6	Tanggal Surat	Text	Menerima input berupa tanggal surat
7	File Surat	Button	Jika diklik akan meminta aktor untuk memilih file dari surat
8	Keterangan	Text	Menerima input berupa keterangan surat
9	Simpan	Button	Jika diklik dan benar maka akan disimpan dan kembali ke halaman kelola surat keluar
10	Kembali	Button	Jika diklik akan kembali ke halaman sebelumnya

10. Halaman Klasifikasi Surat

SIMAK | Klasifikasi Surat

Pengaturan Admin

Home > Klasifikasi Surat

Tambah Data 1 2 Search

Kode	Nama	Bobot	Aksi
KU.00	ANGGARAN	Rencana dan Penyusunan Anggaran	edit hapus 3 4
KU.00.	RENCANA ANGGARAN	Berkenaan dgn perencanaan anggaran seperti RAKIP	edit hapus

© 2016 Kementerian Agama Kota Malang

Gambar 4.41 Rancangan Halaman Klasifikasi Surat

Tabel 4.40 Keterangan Rancangan Halaman Klasifikasi Surat

No.	Nama objek	Tipe	Keterangan
1	Tambah Data	Button	Jika diklik akan menuju form tambah data klasifikasi surat
2	Search	Text	Menerima input berupa text untuk pencarian klasifikasi surat
3	Edit	Button	Jika diklik akan menuju halaman edit klasifikasi surat
4	Hapus	Button	Jika diklik akan muncul konfirmasi penghapusan klasifikasi surat

11. Halaman Cetak Agenda Surat

SIMAK | Cetak Agenda Surat Masuk

User

Home > Cetak Agenda Surat Masuk

Waktu Agenda

Dari Tanggal 1

Sampai Tanggal 2

Cetak 3 4 Kembali

© 2016 Kementerian Agama Kota Malang

Gambar 4.42 Rancangan Halaman Cetak Agenda Surat

Tabel 4.41 Keterangan Rancangan Halaman Cetak Agenda Surat

No.	Nama objek	Tipe	Keterangan
1	Dari Tanggal	Date	Menerima input berupa tanggal awal mencetak agenda surat
2	Sampai Tanggal	Date	Menerima input berupa tanggal akhir mencetak agenda surat
3	Simpan	Button	Jika diklik dan benar maka akan disimpan dan kembali ke halaman kelola surat keluar
4	Kembali	Button	Jika diklik akan kembali ke halaman sebelumnya

SIMAK Cetak Agenda Surat Masuk						
AGENDA SURAT MASUK				Dari 05 Jan 2016 sampai 07 Des 2016		
No.	Isi Ringkas	Dari	No. Surat	Tgl. Surat	Pengolah	Tgl. Disposisi
1	Permohonan Liputan Haji	Radar Malang	106/RI06-MIG/XII/2015	04 Jan 2016	Admin	05 Jan 2016
2	Pendirian TPQ	TPQ BABUSSALAM	02/SP/TPQ-B5/I/2016	04 Jan 2016	Admin	05 Jan 2016
3	Pendistribusian Al-Qur'an dan Juz' Amma 2016	KANWIL	Kw.15.6/4/HK.03.2/7500/2016	05 Jan 2016	Admin	05 Jan 2016

Gambar 4.43 Rancangan Hasil Mencetak Agenda Surat

Karena halaman 4.43 hanya dicetak menurut data yang telah ada sebelumnya tanpa perlu inputan dari aktor, maka tidak ada keterangan berupa tabel keterangan rancangan halaman cetak agenda surat.

BAB V

IMPLEMENTASI

Bab ini menjelaskan tentang implementasi pengembangan Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang. Implementasi ini mengacu pada perancangan sistem yang telah dibahas pada bab Analisis dan Perancangan.

Tahap implementasi merupakan tahapan yang dilakukan setelah tahapan analisis dan perancangan sistem selesai dilakukan. Tujuan dari tahapan ini adalah membuat sistem yang dapat dioperasikan sesuai dengan perancangan. Pada tahap ini akan dijelaskan sistem yang dirancang dan cara penggunaanya.

5.1. Batasan Implementasi

Batasan implementasi dalam pengembangan Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang adalah sebagai berikut:

1. *Input* yang diterima oleh sistem adalah data yang diinputkan secara manual oleh operator divisi umum Kementerian Agama Kota Malang (Admin maupun user).
2. *Output* yang dikeluarkan oleh sistem berupa *list* surat masuk, list data disposisi, list data klasifikasi, hak akses ganti password, serta cetak agenda surat masuk dan keluar untuk admin dan user. Sementara tambahan untuk admin, sistem akan mengeluarkan pilihan berupa *manage* user dan setting instansi.

5.2. Implementasi Komponen

Berikut implementasi beberapa proses yang ada di dalam Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang.

5.2.1. Klasifikasi Surat

Sebelum data surat keluar disimpan, maka data klasifikasi surat harus dipilih untuk memudahkan pengelompokan surat keluar dan kejelasan divisi mana yang mengeluarkannya. Kode controller untuk klasifikasi surat ditunjukkan dalam Tabel 5.1.

Tabel 5.1 Source Code Klasifikasi Surat

1	public function klas_surat() {
2	if (\$this->session->user_valid == FALSE && \$this->session->user_id == "") {
3	redirect("simak/login");
4	}
5	
6	/* pagination */
7	\$total_row = \$this->surat_model->get_total_row();
8	\$per_page = 10;
9	\$awal = \$this->uri->segment(4, 0);
10	\$akhir = \$per_page;
11	\$data['pagi'] = _page(\$total_row, \$per_page, 4, site_url('surat/klas_surat/page'));
12	
13	//ambil variabel URL
14	\$go_to = \$this->uri->segment(3);
15	\$id_url = \$this->uri->segment(4);
16	
17	//ambil variabel Postingan
18	\$id_post = addslashes(\$this->input->post('id_post'));
19	\$kode = addslashes(\$this->input->post('kode'));
20	\$divisi = addslashes(\$this->input->post('divisi'));
21	\$nama = addslashes(\$this->input->post('nama'));
22	\$uraian = addslashes(\$this->input->post('uraian'));
23	\$cari = addslashes(\$this->input->post('q'));
24	
25	if (\$go_to == "del") {
26	\$this->surat_model->delete_klasifikasi_surat(\$id_url);
27	\$this->session->set_flashdata('message', message_box('Data telah dihapus'));
28	redirect('surat/klas_surat');
29	} else if (\$go_to == "cari") {
30	\$data['data'] = \$this->surat_model->cari_klas_surat(\$cari);
31	\$data['page'] = "surat/list_klas_surat";
32	} else if (\$go_to == "add") {
33	\$data['page'] = "surat/form_klas_surat";
34	} else if (\$go_to == "edit") {
35	\$data['datapil'] = \$this->surat_model->get_data_id(\$id_url);
36	\$data['page'] = "surat/form_klas_surat";
37	} else if (\$go_to == "act_add") {
38	\$this->surat_model->insert_data_klasifikasi(\$kode, \$divisi, \$nama, \$uraian);
39	\$this->session->set_flashdata('message', message_box('Data telah ditambah'));
40	redirect('surat/klas_surat');
41	} else if (\$go_to == "act_edit") {
42	\$this->surat_model->edit_data_klasifikasi(\$kode, \$divisi, \$nama, \$uraian, \$id_post);
43	\$this->session->set_flashdata('message', message_box('Data telah diperbaharui'));

44	redirect('surat/klas_surat');
45	} else {
46	\$data['data'] = \$this->surat_model->get_data_limit(\$awal, \$akhir);
47	\$data['page'] = "surat/list_klas_surat";
48	}
49	\$data['title'] = "Klasifikasi Surat";
50	\$this->load->view('simak/header', \$data);
51	}

5.2.2. Kelola Surat Masuk

Kode ini berisi sekumpulan percabangan apakah admin/user memilih menambah surat masuk, mengedit surat masuk, menghapus surat masuk, mendisposisi surat, mencetak disposisi, atau mencari surat masuk. Semua inputan pada view surat_masuk diolah pada fungsi ini, seperti ditunjukkan pada Tabel 5.2

Tabel 5.2 Source Code Kelola Surat Masuk

1	public function surat_masuk() {
2	if (\$this->session->user_valid == FALSE && \$this->session->user_id == "") {
3	redirect("simak/login");
4	}
5	
6	/* pagination */
7	\$total_row = \$this->surat_model->get_total_row_surat_masuk();
8	\$per_page = 10;
9	\$awal = \$this->uri->segment(4, 0);
10	\$akhir = \$per_page;
11	\$data['pagi'] = _page(\$total_row, \$per_page, 4, site_url('surat/surat_masuk/page'));
12	
13	//ambil variabel URL
14	\$mau_ke = \$this->uri->segment(3);
15	\$id_url = \$this->uri->segment(4);
16	
17	//ambil variabel Postingan
18	\$id_post = addslashes(\$this->input->post('id_post'));
19	\$no_agenda = addslashes(\$this->input->post('no_agenda'));
20	\$indek_berkas = \$no_agenda;
21	//\$indek_berkas = addslashes(\$this->input->post('indek_berkas'));
22	\$kode = addslashes(\$this->input->post('kode'));
23	\$dari = addslashes(\$this->input->post('dari'));
24	\$no_surat = addslashes(\$this->input->post('no_surat'));
25	\$tgl_surat = addslashes(\$this->input->post('tgl_surat'));
26	\$uraian = addslashes(\$this->input->post('uraian'));
27	\$ket = addslashes(\$this->input->post('ket'));

28	\$tglcari = addslashes(\$this->input->post('t'));
29	\$cari = addslashes(\$this->input->post('q'));
30	
31	//upload config
32	\$config['upload_path'] = './upload/surat_masuk';
33	\$config['allowed_types'] = 'jpg png jpeg pdf doc docx';
34	\$config['max_size'] = '10000';
35	\$config['max_width'] = '10000';
36	\$config['max_height'] = '10000';
37	\$this->load->library('upload', \$config);
38	
39	if (\$mau_ke == "del") {
40	\$this->surat_model->delete_surat_masuk(\$id_url);
41	\$this->session->set_flashdata('message', message_box('Data telah dihapus'));
42	redirect('surat/surat_masuk');
43	} else if (\$mau_ke == "cari") {
44	if (\$cari == "") {
45	\$data['data'] = \$this->surat_model->cari_surat_masuk_tgl(\$tglcari);
46	\$data['page'] = "surat/list_surat_masuk";
47	} else if (\$tglcari == "") {
48	\$data['data'] = \$this->surat_model->cari_surat_masuk_key(\$cari);
49	\$data['page'] = "surat/list_surat_masuk";
50	} else {
51	\$data['data'] = \$this->surat_model->cari_surat_masuk_tgl_key(\$tglcari, \$cari);
52	\$data['page'] = "surat/list_surat_masuk";
53	}
54	} else if (\$mau_ke == "add") {
55	\$data['page'] = "surat/form_surat_masuk";
56	} else if (\$mau_ke == "edit") {
57	\$data['datapil'] = \$this->surat_model->select_surat_masuk_id(\$id_url);
58	\$data['page'] = "surat/form_surat_masuk";
59	} else if (\$mau_ke == "act_add") {
60	if (\$this->upload->do_upload('file_surat')) {
61	\$up_data = \$this->upload->data('file_name');
62	\$this->surat_model->insert_surat_masuk_with_file(\$kode, \$no_agenda, \$indek_berkas, \$uraian, \$dari, \$no_surat, \$tgl_surat, \$ket, \$up_data);
63	} else {
64	\$this->surat_model->insert_surat_masuk(\$kode, \$no_agenda, \$indek_berkas, \$uraian, \$dari, \$no_surat, \$tgl_surat, \$ket);
65	}
66	\$this->session->set_flashdata('message', message_box('Data telah ditambah. ' . \$this->upload->display_errors()));
67	redirect('surat/surat_masuk');
68	} else if (\$mau_ke == "act_edit") {
69	if (\$this->upload->do_upload('file_surat')) {

70	\$up_data = \$this->upload->data('file_name');
71	\$this->surat_model->update_surat_masuk_with_file(\$kode, \$no_agenda,
72	\$indek_berkas, \$uraian, \$dari, \$no_surat, \$tgl_surat, \$ket, \$up_data, \$id_post);
73	} else {
74	\$this->surat_model->update_surat_masuk(\$kode, \$no_agenda, \$indek_berkas,
75	\$uraian, \$dari, \$no_surat, \$tgl_surat, \$ket, \$id_post);
76	\$this->session->set_flashdata('message', message_box('Data telah diperbaharui. ' .
77	\$this->upload->display_errors()));
78	redirect('surat/surat_masuk');
79	} else {
80	\$data['data'] = \$this->surat_model->select_surat_masuk_limit(\$awal, \$akhir);
81	\$data['page'] = "surat/list_surat_masuk";
82	}
83	\$data['title'] = "Surat Masuk";
	\$this->load->view('simak/header', \$data);
	}

5.2.3. Kelola Surat Keluar

Kode pada Tabel 5.3 isinya berupa sekumpulan percabangan apakah admin/user ingin menambah surat keluar, mengedit surat keluar, menghapus surat keluar, dan mencari surat masuk.

Tabel 5.3 Source Code Kelola Surat Keluar

1	public function surat_keluar() {
2	if (\$this->session->user_valid == FALSE && \$this->session->user_id == "") {
3	redirect("simak/login");
4	}
5	
6	/* pagination */
7	\$total_row = \$this->surat_model->get_total_row_surat_keluar();
8	\$per_page = 10;
9	\$awal = \$this->uri->segment(4, 0);
10	\$akhir = \$per_page;
11	\$data['pagi'] = _page(\$total_row, \$per_page, 4, site_url('surat/surat_keluar/page'));)
12	
13	//ambil variabel URL
14	\$mau_ke = \$this->uri->segment(3);
15	\$id_url = \$this->uri->segment(4);
16	
17	//ambil variabel Postingan
18	\$id_post = addslashes(\$this->input->post('id_post'));
19	\$no_agenda = addslashes(\$this->input->post('no_agenda'));
20	\$kode = addslashes(\$this->input->post('kode'));
21	\$dari = addslashes(\$this->input->post('dari'));

22	\$no_surat = addslashes(\$this->input->post('no_surat'));
23	\$tgl_surat = addslashes(\$this->input->post('tgl_surat'));
24	\$suraian = addslashes(\$this->input->post('uraian'));
25	\$ket = addslashes(\$this->input->post('ket'));
26	\$tglcari = addslashes(\$this->input->post('t'));
27	\$cari = addslashes(\$this->input->post('q'));
28	
29	//upload config
30	\$config['upload_path'] = './upload/surat_keluar';
31	\$config['allowed_types'] = 'jpg png pdf jpeg doc docx';
32	\$config['max_size'] = '10000';
33	\$config['max_width'] = '10000';
34	\$config['max_height'] = '10000';
35	\$this->load->library('upload', \$config);
36	
37	if (\$mau_ke == "del") {
38	\$this->surat_model->delete_surat_keluar(\$id_url);
39	\$this->session->set_flashdata('message', message_box('Data telah dihapus'));
40	redirect('surat/surat_keluar');
41	} else if (\$mau_ke == "cari") {
42	if (\$cari == "") {
43	\$data['data'] = \$this->surat_model->cari_surat_keluar_tgl(\$tglcari);
44	\$data['page'] = "surat/list_surat_keluar";
45	} else if (\$tglcari == "") {
46	\$data['data'] = \$this->surat_model->cari_surat_keluar_key(\$cari);
47	\$data['page'] = "surat/list_surat_keluar";
48	} else {
49	\$data['data'] = \$this->surat_model->cari_surat_keluar_tgl_key(\$tglcari, \$cari);
50	\$data['page'] = "surat/list_surat_keluar";
51	}
52	} else if (\$mau_ke == "add") {
53	\$data['data'] = \$this->surat_model->select_surat_keluar_div();
54	\$data['page'] = "surat/form_surat_keluar";
55	} else if (\$mau_ke == "edit") {
56	\$data['data'] = \$this->surat_model->select_surat_keluar_div();
57	\$data['datpil'] = \$this->surat_model->select_surat_keluar_id(\$id_url);
58	\$data['page'] = "surat/form_surat_keluar";
59	} else if (\$mau_ke == "act_add") {
60	if (\$this->upload->do_upload('file_surat')) {
61	\$up_data = \$this->upload->data(file_name);
62	\$this->surat_model->insert_surat_keluar_with_file(\$kode, \$no_agenda,
63	\$suraian, \$dari, \$no_surat, \$tgl_surat, \$ket, \$up_data);
64	} else {
	\$this->surat_model->insert_surat_keluar(\$kode, \$no_agenda, \$suraian, \$dari,
	\$no_surat, \$tgl_surat, \$ket);

65	}
66	\$this->session->set_flashdata('message', message_box('Data telah ditambah'));
67	redirect('surat/surat_keluar');
68	} else if (\$mau_ke == "act_edit") {
69	if (\$this->upload->do_upload('file_surat')) {
70	\$up_data = \$this->upload->data(file_name);
71	\$this->surat_model->update_surat_keluar_with_file(\$no_agenda, \$kode, \$uraian, \$dari, \$no_surat, \$tgl_surat, \$ket, \$up_data, \$id_post);
72	} else {
73	\$this->surat_model->update_surat_keluar(\$no_agenda, \$kode, \$uraian, \$dari, \$no_surat, \$tgl_surat, \$ket, \$id_post);
74	}
75	\$this->session->set_flashdata('message', message_box('Data telah diperbaharui' . \$this->upload->display_errors()));
76	redirect('surat/surat_keluar');
77	} else {
78	\$data['data'] = \$this->surat_model->select_surat_keluar_limit(\$awal, \$akhir);
79	\$data['page'] = "surat/list_surat_keluar";
80	}
81	\$data['title'] = "Surat Keluar";
82	\$this->load->view('simak/header', \$data);
83	}

5.2.4. Disposisi Surat

Saat sebuah surat masuk selesai diinputkan, langkah selanjutnya yaitu mendisposisi surat masuk tersebut. Disposisi surat berisi penerima disposisi, isi disposisi, sifat, dan batas waktu pelaksanaan. Kode untuk disposisi surat dapat dilihat pada Tabel 5.4.

Tabel 5.4 Source Code Disposisi Surat

1	public function surat_disposisi() {
2	if (\$this->session->user_valid == FALSE && \$this->session->user_id == "") {
3	redirect("simak/login");
4	}
5	
6	//ambil variabel URL
7	\$go_to = \$this->uri->segment(4);
8	\$id_url1 = \$this->uri->segment(3);
9	\$id_url2 = \$this->uri->segment(5);
10	
11	// pagination //
12	\$total_row = \$this->surat_model->get_total_row_surat_disposisi(\$id_url1);
13	\$per_page = 10;
14	\$awal = \$this->uri->segment(4, 0);

```

15     $akhir = $per_page;
16     $data['pagi'] = _page($total_row, $per_page, 4, site_url("surat/surat_disposisi/" .
17     $id_url1 . "/page"));
18     //ambil variabel Postingan
19     $id_post = addslashes($this->input->post('id_post'));
20     $id_surat = addslashes($this->input->post('id_surat'));
21     $kepada = addslashes($this->input->post('kepada'));
22     $isi_disposisi = addslashes($this->input->post('isi_disposisi'));
23     $sifat = addslashes($this->input->post('sifat'));
24     $batas_waktu = addslashes($this->input->post('batas_waktu'));
25     $catatan = addslashes($this->input->post('catatan'));
26     //$cari = addslashes($this->input->post('q'));
27     //judul surat
28     $judul_surat = gval("surat_masuk", "id", "isi_ringkas", $id_url1);
29     $this->session->set_flashdata('judul_surat', message_box('Perihal: ' . $judul_surat));
30
31     if ($go_to == "del") {
32         $this->surat_model->delete_disposisi($id_url2);
33         $this->session->set_flashdata('message', message_box('Data telah dihapus'));
34         redirect('surat/surat_disposisi/' . $id_url1);
35     } else if ($go_to == "add") {
36         $data['page'] = "surat/form_surat_disposisi";
37     } else if ($go_to == "edit") {
38         $data['datapil'] = $this->surat_model->get_disposisi_id($id_url2);
39         $data['page'] = "surat/form_surat_disposisi";
40     } else if ($go_to == "act_add") {
41         $this->surat_model->insert_disposisi($id_surat, $kepada, $isi_disposisi, $sifat,
42         $batas_waktu, $catatan);
43         $this->session->set_flashdata('message', message_box('Data telah ditambah'));
44         redirect('surat/surat_disposisi/' . $id_surat);
45     } else if ($go_to == "act_edit") {
46         $this->surat_model->update_disposisi($kepada, $isi_disposisi, $sifat,
47         $batas_waktu, $catatan, $id_post);
48         $this->session->set_flashdata('message', message_box('Data telah diperbaharui'));
49         redirect('surat/surat_disposisi/' . $id_surat);
50     } else {
51         $data['data'] = $this->surat_model->get_disposisi_limit($id_url1, $awal, $akhir);
52         $data['page'] = "surat/list_surat_disposisi";
53     }
54     $data['title'] = "Disposisi Surat";
55     $this->load->view('simak/header', $data);
56 }

```

5.2.5.Cetak Agenda

Surat masuk dan surat keluar yang telah diinputkan kedalam sistem akan disimpan dalam basis data. Untuk mencetak data surat masuk atau surat keluar dalam selang waktu tertentu maka dibuat fungsi cetak agenda. Tabel 5.5 menunjukkan proses controller untuk mencetak agenda surat masuk atau keluar dengan menampilkan view agenda_surat_masuk atau agenda surat keluar.

Tabel 5.5 Source Code Cetak Agenda

1	public function cetak_agenda() {
2	\$jenis_surat = \$this->input->post('tipe');
3	\$ttl_start = \$this->input->post('ttl_start');
4	\$ttl_end = \$this->input->post('ttl_end');
5	
6	\$data['ttl_start'] = \$ttl_start;
7	\$data['ttl_end'] = \$ttl_end;
8	
9	if (\$jenis_surat == "cetak_agenda_masuk") {
10	\$data['data'] = \$this->cetak_model->cetak_agenda_surat_masuk(\$ttl_start,
11	\$ttl_end);
12	\$data['title'] = "Cetak Agenda Surat Masuk";
13	\$this->load->view('cetak/cetak_agenda_masuk', \$data);
14	} else {
15	\$data['data'] = \$this->cetak_model->cetak_agenda_surat_keluar(\$ttl_start,
16	\$ttl_end);
17	\$data['title'] = "Cetak Agenda Surat Keluar";
18	\$this->load->view('cetak/cetak_agenda_keluar', \$data);
19	}
20	}

5.2.6.Cetak Disposisi

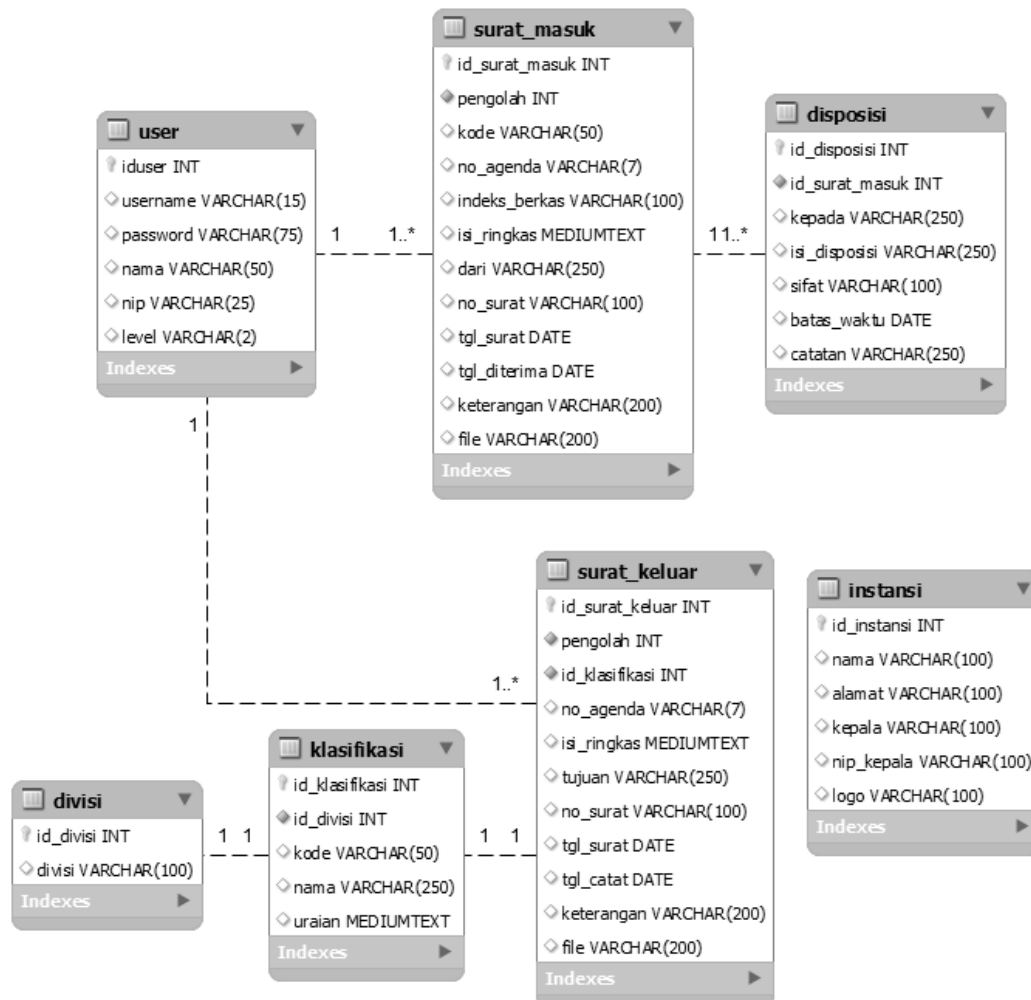
Cetak disposisi adalah fungsi pada cetak controller untuk mencetak disposisi surat dengan mengambil data dari tabel surat_masuk dan tabel disposisi. Kode untuk prosesnya ditunjukkan pada Tabel 5.6.

Tabel 5.6 Source Kode Cetak Disposisi

1	public function cetak_disposisi() {
2	\$id_url = \$this->uri->segment(3);
3	\$this->load->model('surat_model');
4	\$data['datpil1'] = \$this->surat_model->select_surat_masuk_id(\$id_url);
5	\$data['datpil2'] = \$this->cetak_model->get_disposisi_cetak(\$id_url);
6	\$data['title'] = "Cetak Disposisi";
7	\$this->load->view('cetak/cetak_disposisi', \$data);}

5.3. Implementasi Data

Implementasi penyimpanan data dilakukan dengan *database management system* MariaDB. Hasil implementasi dimodelkan dalam tabel-tabel data. Implementasi data pada sistem ini sebagai berikut:



Gambar 5.1 Physical Data Model SIMAK

Keterangan tabel:

1. Tabel divisi

Tabel ini digunakan untuk menyimpan divisi yang ada pada instansi Kemenag untuk diinputkan kedalam sistem terutama pada saat membuat surat keluar.

2. Tabel klasifikasi

Tabel ini digunakan untuk menyimpan klasifikasi nomor surat keluar yang ada pada instansi terkait.

3. Tabel instansi

Tabel ini digunakan untuk menyimpan informasi mengenai instansi Kemenag berupa nama instansi, alamat instansi, nama kepala kantor, NIP kepala kantor, dan logo instansi.

4. Tabel user

Tabel ini digunakan untuk menyimpan akun pengguna yang hanya bisa dikelola oleh admin. Didalam tabel user terdapat atribut username, password, nama untuk menyimpan nama lengkap pengguna, NIP, dan level untuk menyimpan level pengguna dimana dalam atribut ini mempunyai 2 level yaitu sebagai admin dan sebagai user.

5. Tabel disposisi

Tabel ini digunakan untuk menyimpan informasi atau isi dari disposisi surat. Didalamnya terdapat atribut:

- a. id_surat_masuk sebagai foreign key yang mengarah pada surat masuk.
- b. kepada digunakan untuk menyimpan data surat tersebut akan diteruskan kepada siapa.
- c. isi_disposisi merupakan atribut yang digunakan untuk menyimpan isi disposisi surat yang dimasukkan oleh user kedalam sistem.
- d. sifat merupakan atribut yang digunakan untuk menyimpan bagaimana sifat atau tindakan yang harus dilakukan terhadap sebuah surat masuk.
- e. batas_waktu yakni atribut yang digunakan untuk menyimpan batas waktu surat tersebut akan selesai diproses.
- f. catatan adalah atribut yang berisi catatan tentang sebuah disposisi surat.

6. Tabel surat_masuk

Tabel ini digunakan untuk menyimpan informasi atau detail dari surat masuk. Berikut penjelasan setiap atribut pada tabel:

- a. Kode, merupakan atribut yang digunakan untuk menyimpan kode surat masuk dari instansi luar.
- b. pengolah, merupakan atribut foreign_key yang digunakan untuk menyimpan data pengolah surat (tabel user) yang terdapat pada sistem.

- c. `no_agenda`, merupakan atribut yang digunakan untuk menyimpan nomor agenda surat secara urut.
 - d. `indeks_berkas`, merupakan atribut yang digunakan untuk menyimpan indeks berkas surat.
 - e. `isi_ringkas`, merupakan atribut yang digunakan untuk menyimpan perihal surat yang masuk.
 - f. `dari`, merupakan atribut yang digunakan untuk menyimpan pengirim dari surat masuk.
 - g. `no_surat`, merupakan atribut yang digunakan untuk menyimpan nomor surat yang masuk ke kantor Kementerian Agama Kota Malang.
 - h. `tgl_surat`, merupakan atribut yang digunakan untuk menyimpan data surat itu dikeluarkan atau dibuat oleh instansi terkait.
 - i. `tgl_diterima`, merupakan atribut yang digunakan untuk menyimpan tanggal surat itu diterima atau diinputkan kedalam aplikasi.
 - j. `keterangan`, merupakan atribut yang digunakan untuk menyimpan keterangan dari sebuah surat.
 - k. `file`, merupakan atribut yang digunakan untuk menyimpan file *scan* surat dalam bentuk *.jpeg/.png* yang akan diinputkan oleh user.
7. Tabel `surat_keluar`
- Tabel ini digunakan untuk menyimpan informasi atau detail dari surat keluar.
- a. `pengolah`, merupakan atribut foreign key yang digunakan untuk menyimpan data pengolah surat yang terdapat pada sistem.
 - b. `id_klasifikasi`, merupakan atribut foreign key yang digunakan untuk menyimpan kode surat yang dipilih oleh user saat menginputkan data. Dimana kode surat ini bias dipilih sesuai dengan divisi kerja yang mengeluarkan surat tersebut.
 - c. `no_agenda`, merupakan atribut yang digunakan untuk menyimpan nomor agenda surat secara urut.
 - d. `isi_ringkas`, merupakan atribut yang digunakan untuk menyimpan perihal surat keluar yang akan diinput oleh admin.

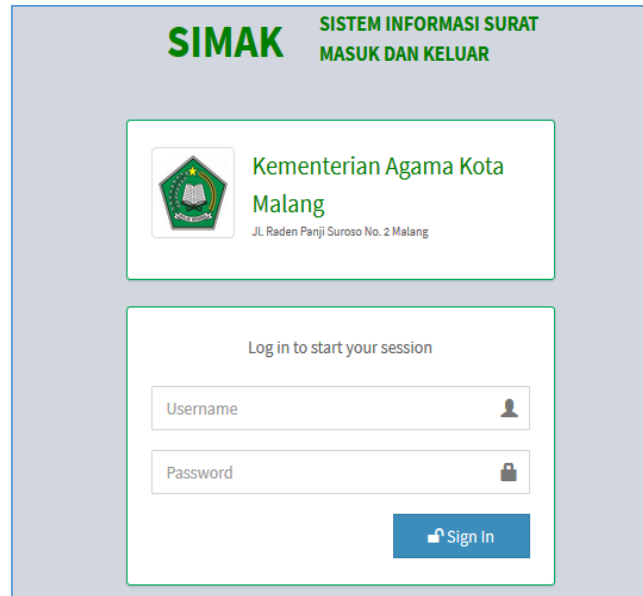
- e. tujuan, merupakan atribut yang digunakan untuk menyimpan tujuan dari surat keluar.
- f. no_surat, merupakan atribut yang digunakan untuk menyimpan nomor surat yang keluar dari kantor Kementerian Agama Kota Malang.
- g. tgl_surat, merupakan atribut yang menyimpan data surat itu dikeluarkan atau dibuat.
- h. tgl_catat, merupakan atribut menyimpan data tanggal surat itu dicatat atau diinputkan kedalam aplikasi.
- i. keterangan, merupakan atribut yang digunakan untuk menyimpan keterangan dari sebuah surat.
- j. file, merupakan atribut yang digunakan untuk menyimpan file *scan* surat dalam bentuk .jpeg/.png yang akan diinputkan oleh user.

5.4. Implementasi Antarmuka

Implementasi antarmuka Sistem Informasi Surat Masuk dan Keluar Kementerian Agama Kota Malang terdiri dari implementasi untuk admin dan untuk user. Antarmuka untuk user hanya ada beberapa *point* yang berbeda. Berikut adalah fitur-fitur yang dapat diakses oleh admin maupun user pada sistem, antara lain sebagai berikut:

5.4.1. Halaman Login

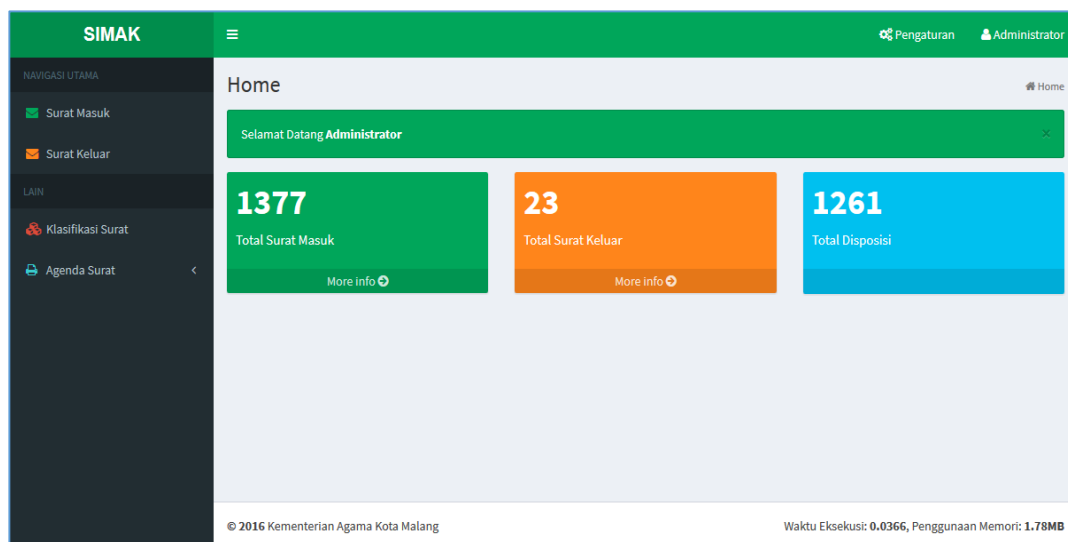
Admin atau user melakukan login pada halaman login dengan memasukkan username dan password yang dimiliki, kemudian klik tombol Sign in dan sistem langsung masuk pada halaman utama jika data yang diinputkan valid.



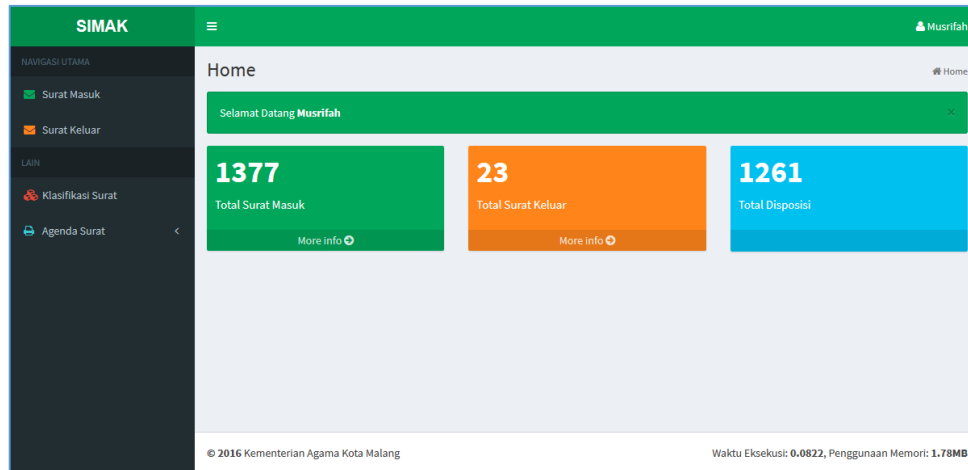
Gambar 5.2 Halaman Login

5.4.2. Halaman Home

Halaman home memiliki pilihan menu yang berbeda yaitu user tidak dapat memilih menu manage user dan setting instansi. Halaman home berisi ucapan selamat datang dan *counter* jumlah surat masuk, keluar dan disposisi yang telah diinputkan.

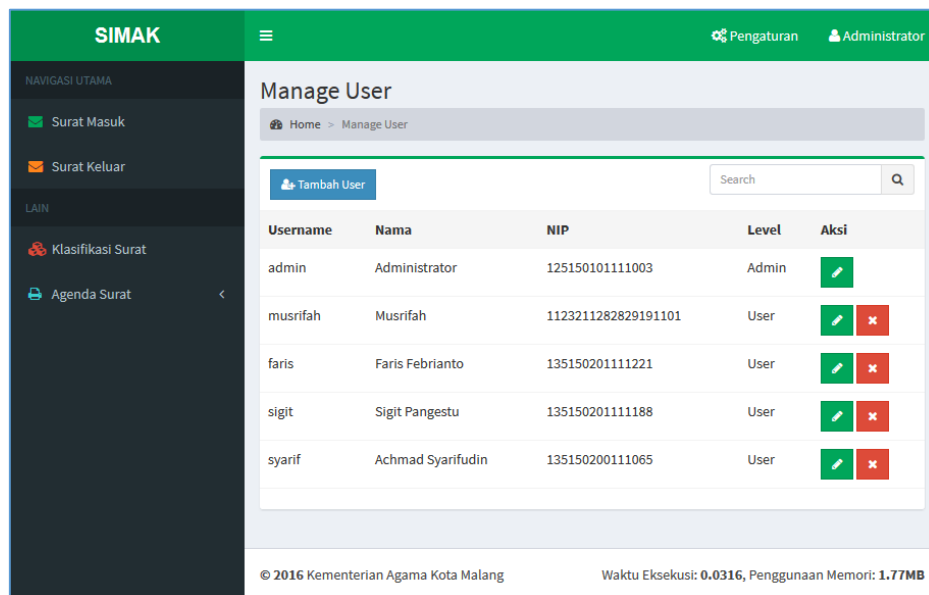


Gambar 5.3 Halaman Home untuk Admin



Gambar 5.4 Halaman Home untuk User

5.4.3. Halaman Manage User



Gambar 5.5 Halaman Manage User

Manage user hanya dapat dilakukan oleh admin. Ketika memilih menu tambah data, maka admin dapat menginputkan data user yang akan didaftarkan berupa username, password baru, ulangi password, nama lengkap, NIP, dan pilih level. Selain itu admin juga dapat mengedit data user serta menghapus user yang ada pada daftar user.

SIMAK Pengaturan Administrator

NAVIGASI UTAMA

- Surat Masuk
- Surat Keluar

LAIN

- Klasifikasi Surat
- Agenda Surat

Manage User

Home > Manage User

Data User

Username

Nama Lengkap

Password baru

NIP

Ulangi Password baru

Pilih Level

Simpan Kembali Clear

© 2016 Kementerian Agama Kota Malang Waktu Eksekusi: 0.0837, Penggunaan Memori: 1.76MB

Gambar 5.6 Halaman Input Data User

5.4.4. Halaman Setting Instansi

Setting instansi hanya dilakukan oleh admin. Pada halaman setting instansi, admin dapat menginput data instansi berupa nama instansi, alamat instansi, nama kepala, NIP kepala, dan logo instansi.

SIMAK Pengaturan Administrator

NAVIGASI UTAMA

- Surat Masuk
- Surat Keluar

LAIN

- Klasifikasi Surat
- Agenda Surat

Setting Instansi

Home > Setting Instansi

Data Instansi

Nama Instansi

Kementerian Agama Kota Malang

Nama Kepala

Drs. H. IMRON, M.Ag

Alamat Instansi

Jl. Raden Panji Suroso No. 2 Malang

NIP Kepala

196008161994031004

Logo Instansi

Browse... No file selected.

Logo harus berekstensi .jpg/.jpeg/.png

Simpan Kembali Clear

© 2016 Kementerian Agama Kota Malang Waktu Eksekusi: 0.0455, Penggunaan Memori: 1.81MB

Gambar 5.7 Halaman Setting Instansi

5.4.5. Halaman Ganti Password

Untuk mengganti password, baik admin maupun user dapat memilih menu ganti password. User mengisi form rubah password dan bila berhasil maka password baru akan tersimpan kedalam sistem secara otomatis.

Gambar 5.8 Halaman Ganti Password

5.4.6. Halaman Kelola Surat Masuk

Bagian surat masuk dapat diakses oleh admin maupun user. Berikut tampilan dan penjelasan dari bagian-bagian yang dapat diakses oleh admin maupun user. Admin dan user dapat mengakses bagian dari tata persuratan surat masuk, yaitu: tambah data, edit data, hapus surat masuk, disposisi surat masuk, edit posisi, cetak disposisi dan cari surat.

No.	Isi Ringkas, File	Asal Surat	Nomor, Tgl. Surat	Aksi
1	Surat Pengantar File: Image_9.jpg	Balai Diklat Keagamaan Surabaya	Bdl.07/HM.01/6706/2015 17 Des 2015	[Edit] [Delete] [Print] [Lock]
02	PENAWARAN PEMASANGAN IKLAN File: Image.jpg	RADIO ANDALUS FM	0005/MAR/ALS/I/2016 04 Des 2015	[Edit] [Delete] [Print] [Lock]
03	Permohonan Liputan Haji File: Admin_Data_Thn_20168.jpg	Radar Malang	106/RI06-MIG/XII/2015 04 Jan 2016	[Edit] [Delete] [Print] [Lock]
04	Pendirian Taman Pendidikan Al-Qur'an (TPQ) File: Admin_Data_Thn_20167.jpg	TPQ BABUSSALAM	02/SP/TPQ-BS/I/2016 04 Jan 2016	[Edit] [Delete] [Print] [Lock]
05	Permohonan ijin magang File: Admin_Data_Thn_20166.jpg	Universitas Islam Negeri Maulana Malik Ibrahim...	Un.03.2/PP.07/1294/2016 05 Jan 2016	[Edit] [Delete] [Print] [Lock]
06	Pendistribusian kitab suci Al-Qur'an dan Juz'amma... File: Admin_Data_Thn_20165.jpg	KANWIL	Kw.15.6/4/HK.03.2/7500/2016 05 Jan 2016	[Edit] [Delete] [Print] [Lock]
7	Pemberitahuan retur SP2D File: Image_5.jpg	KPPN	S-5447/WPB.16/KP.032/2016 05 Jan 2016	[Edit] [Delete] [Print] [Lock]

Gambar 5.9 Halaman Kelola Surat Masuk

The screenshot displays the 'Surat Masuk' form in the SIMAK application. The sidebar on the left lists navigation options: 'Surat Masuk' (active), 'Surat Keluar', 'Klasifikasi Surat', and 'Agenda Surat'. The main form area is titled 'Surat Masuk' and contains the following fields and controls:

- No. Agenda:** A text input field containing 'p100'.
- Tanggal Surat:** A date selection field.
- Nomor Surat:** A text input field.
- File Surat:** A section with a 'Browse...' button, the text 'No file selected.', and a note 'File harus berekstensi .pdf/.jpg/.jpeg/.png'.
- Asal Surat:** A text input field.
- Keterangan:** A text input field.
- Isi Ringkas:** A large text area for a summary.
- Buttons:** 'Simpan' (Save), 'Kembali' (Back), and 'Clear'.

The footer of the application shows '© 2016 Kementerian Agama Kota Malang' and system performance data: 'Waktu Eksekusi: 0.0830, Penggunaan Memori: 1.96MB'.

Gambar 5.10 Halaman Form Surat Masuk

Jika dipilih tambah data, pada form surat masuk, user dapat menambah data surat masuk dengan mengisi form tambah surat masuk. Selain itu juga ada menu untuk melampirkan hasil scan surat masuk berupa gambar (.jpg, .jpeg, .png) atau dalam bentuk pdf.

Pada Edit Surat Masuk, dapat mengedit informasi surat masuk yang sudah tersimpan dalam sistem.

Hapus Surat Masuk, dapat menghapus surat masuk yang tersimpan dalam sistem. Setelah button “OK” dipilih, maka surat masuk yang sudah terpilih akan terhapus dari sistem.

Cari surat masuk ada dua cara, yaitu dengan menginputkan tanggal atau agar lebih spesifik lagi dengan menginputkan kata kunci pencarian.

5.4.7. Halaman Disposisi Surat

SIMAK Pengaturan Administrator

NAVIGASI UTAMA

- Surat Masuk
- Surat Keluar

LAIN

- Klasifikasi Surat
- Agenda Surat

Disposisi Surat Home > Disposisi Surat

Perihal: Pendistribusian kitab suci Al-Qur'an dan Juz'Amma Th 2016

Disposisi Surat

Tujuan Disposisi

Tujuan Disposisi

Sifat Surat

Sifat Surat

Isi Disposisi

Isi Disposisi

Batas Waktu

Batas Waktu

Catatan

Catatan


Simpan Kembali Clear

© 2016 Kementerian Agama Kota Malang Waktu Eksekusi: 0.0380, Penggunaan Memori: 1.87MB

Gambar 5.11 Halaman Disposisi Surat

Admin maupun user dapat mengedit isi disposisi dengan cara memilih menu edit pada surat masuk yang diinginkan. Kemudian sistem akan menyimpan dan menampilkan data terbaru disposisi yang telah diedit.

5.4.8. Halaman Cetak Disposisi

 **Kementerian Agama Kota Malang**
Jl. Raden Panji Suroso No. 2 Malang

LEMBAR DISPOSISI

No. Surat: Bdl.07/HM.01/6706/2015

Tgl. Surat: 17 Des 2015 Sifat: BiasaBiasa

Diterima Tanggal: 04 Jan 2016

No. Agenda: 1

Dari: Balai Diklat Keagamaan Surabaya

Perihal: Surat Pengantar

Isi Disposisi: lanjutkantolong diarsipkan

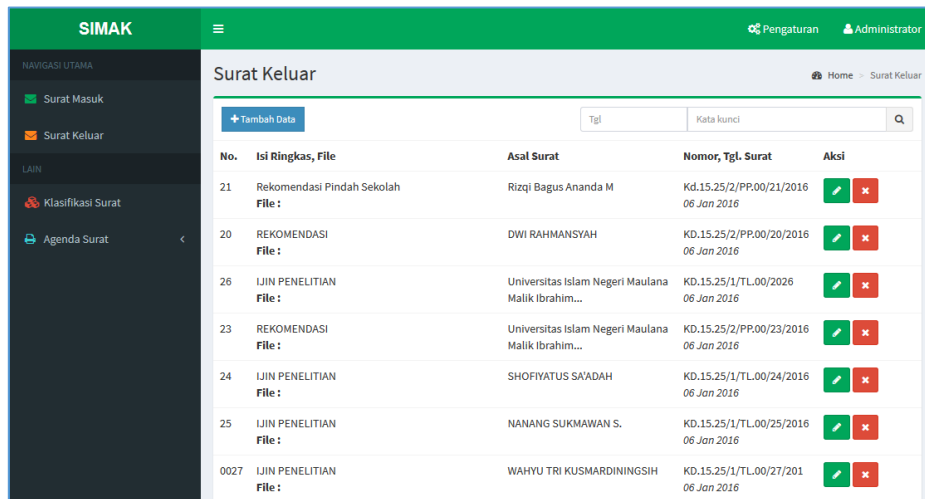
Batas Waktu: 2016-01-052016-01-11

Diteruskan kepada: Kasubag TUSaudara koordinator umum

Gambar 5.12 Halaman Cetak Disposisi

Admin maupun User biasa dapat mencetak langsung disposisi beserta keterangan surat masuk dengan memilih menu cetak disposisi surat yang diinginkan.

5.4.9. Halaman Kelola Surat Keluar

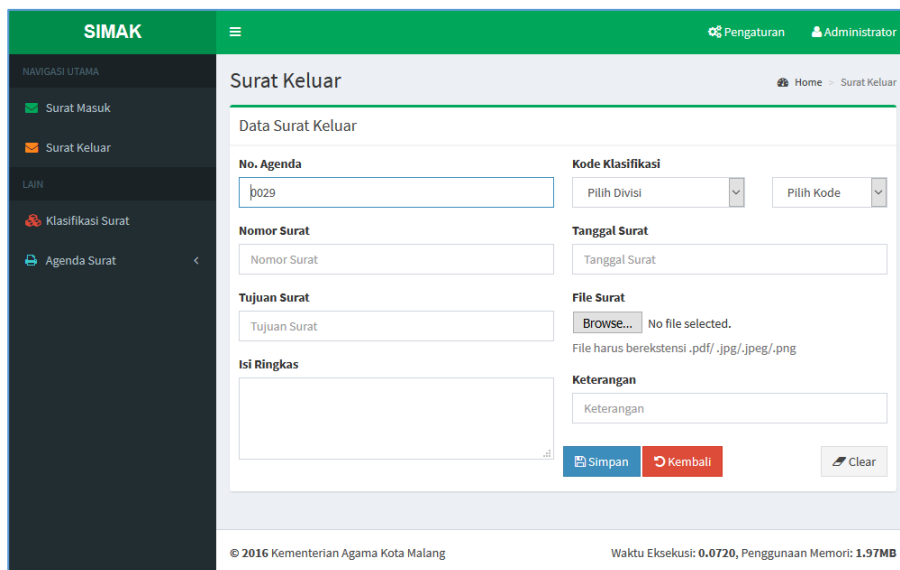


No.	Isi Ringkas, File	Asal Surat	Nomor, Tgl. Surat	Aksi
21	Rekomendasi Pindah Sekolah File :	Rizqi Bagus Ananda M	Kd.15.25/2/PP.00/21/2016 06 Jan 2016	
20	REKOMENDASI File :	DWI RAHMANSYAH	KD.15.25/2/PP.00/20/2016 06 Jan 2016	
26	IJIN PENELITIAN File :	Universitas Islam Negeri Maulana Malik Ibrahim...	KD.15.25/1/TL.00/2026 06 Jan 2016	
23	REKOMENDASI File :	Universitas Islam Negeri Maulana Malik Ibrahim...	KD.15.25/2/PP.00/23/2016 06 Jan 2016	
24	IJIN PENELITIAN File :	SHOFIYATU SA'ADAH	KD.15.25/1/TL.00/24/2016 06 Jan 2016	
25	IJIN PENELITIAN File :	NANANG SUKMAWAN S.	KD.15.25/1/TL.00/25/2016 06 Jan 2016	
0027	IJIN PENELITIAN File :	WAHYU TRI KUSMARDININGSIH	KD.15.25/1/TL.00/27/201 06 Jan 2016	

Gambar 5.13 Halaman Kelola Surat Keluar

Bagian tata persuratan surat keluar dapat diakses oleh admin maupun user. Berikut tampilan dan penjelasan dari bagian-bagian yang dapat diakses oleh admin maupun user.

Admin dan user dapat mengakses bagian dari tata persuratan surat keluar, yaitu: tambah data, edit data, hapus surat keluar, dan cari surat keluar.



Surat Keluar

Data Surat Keluar

No. Agenda:

Kode Klasifikasi:

Nomor Surat:

Tanggal Surat:

Tujuan Surat:

File Surat: No file selected.
File harus berekstensi .pdf/.jpg/.jpeg/.png

Isi Ringkas:

Keterangan:

© 2016 Kementerian Agama Kota Malang Waktu Eksekusi: 0.0720, Penggunaan Memori: 1.97MB

Gambar 5.14 Halaman Form Surat Keluar

Tambah data surat keluar, pada form surat keluar, dapat menambah data surat keluar dengan mengisi form tambah surat keluar. Selain itu juga ada menu untuk melampirkan surat keluar berupa gambar (.jpg, .jpeg, .png) serta pdf. Pada form kode klasifikasi, pilih salah satu divisi, kemudian akan tampil kode-kode yang ada pada divisi tersebut. Setiap kode mewakili fungsi dan divisi tersendiri.

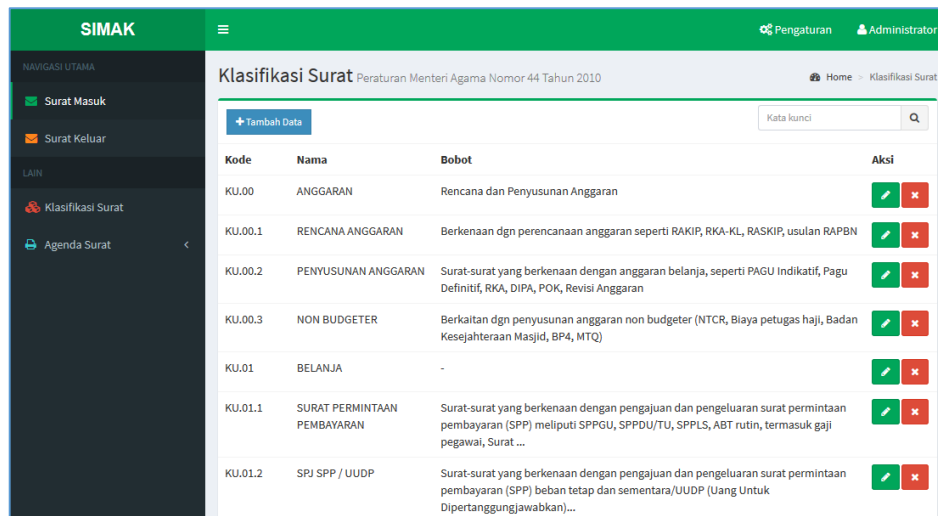
Edit surat keluar, untuk mengedit informasi surat keluar dengan cara mengisi form edit pada surat keluar yang akan diedit. Sistem akan menyimpan dan menampilkan data surat yang baru diedit.

Hapus Surat Keluar, untuk menghapus surat keluar yang tersimpan dalam sistem. Setelah menu “ok” dipilih, maka surat keluar yang sudah terpilih akan terhapus dari sistem.

Cari surat, keluar ada dua cara, yaitu dengan menginputkan tanggal atau agar lebih spesifik lagi dengan menginputkan kata kunci pencarian.

5.4.10. Halaman Klasifikasi Surat

Klasifikasi surat dapat dilakukan oleh admin maupun user.



Kode	Nama	Bobot	Aksi
KU.00	ANGGARAN	Rencana dan Penyusunan Anggaran	[Edit] [Hapus]
KU.00.1	RENCANA ANGGARAN	Berkenaan dgn perencanaan anggaran seperti RAKIP, RKA-KL, RASKIP, usulan RAPBN	[Edit] [Hapus]
KU.00.2	PENYUSUNAN ANGGARAN	Surat-surat yang berkenaan dengan anggaran belanja, seperti PAGU Indikatif, Pagu Definitif, RKA, DIPK, POK, Revisi Anggaran	[Edit] [Hapus]
KU.00.3	NON BUDGETER	Berkaitan dgn penyusunan anggaran non budgeter (NCR, Biaya petugas haji, Badan Kesejahteraan Masjid, BP4, MTQ)	[Edit] [Hapus]
KU.01	BELANJA	-	[Edit] [Hapus]
KU.01.1	SURAT PERMINTAAN PEMBAYARAN	Surat-surat yang berkenaan dengan pengajuan dan pengeluaran surat permintaan pembayaran (SPP) meliputi SPPGU, SPPDU/TU, SPPLS, ABT rutin, termasuk gaji pegawai, Surat ...	[Edit] [Hapus]
KU.01.2	SPJ SPP / UUDP	Surat-surat yang berkenaan dengan pengajuan dan pengeluaran surat permintaan pembayaran (SPP) beban tetap dan sementara/UUDP (Uang Untuk Dipertanggungjawabkan)...	[Edit] [Hapus]

Gambar 5.15 Halaman Klasifikasi Surat

5.4.11. Halaman Cetak Agenda Surat

Admin dapat melihat disposisi surat masuk dengan cara memilih bar Agenda Surat → Surat Masuk, dan kemudian menentukan rentang waktu yang diinginkan. Maka sistem akan menampilkan halaman agenda disposisi surat masuk yang diinginkan.

SIMAK

Pengaturan

Administrator

NAVIGASI UTAMA

Surat Masuk

Surat Keluar

LAIN

Klasifikasi Surat

Agenda Surat

Cetak Agenda

Cetak Agenda Surat Masuk

Home > Cetak Agenda

Waktu Agenda

Dari Tanggal:

2016-01-01

Sampai Tanggal:

2016-12-02

Cetak

Kembali

Clear

© 2016 Kementerian Agama Kota Malang Waktu Eksekusi: 0.0871, Penggunaan Memori: 1.69MB

Gambar 5.16 Halaman Pilih Rentang Waktu Agenda Surat

AGENDA SURAT MASUK							
Dari 01 Jan 2016 sampai 02 Des 2016							
No	Isi Ringkas	Dari	Nomor Surat	Tgl. Surat	Pengolah	Tgl. Disposisi	Ket
1	Surat Pengantar	Balai Diklat Keagamaan Surabaya	BdL07/HM.01/6706/2015	17 Des 2015	Administrator	04 Jan 2016	Masih Naik ke Kepala
2	PENAWARAN PEMASANGAN IKLAN	RADIO ANDALUS FM	0005/MAR/ALS/I/2016	04 Des 2015	Administrator	04 Jan 2016	
3	Permohonan Liputan Haji	Radar Malang	106/RI06-MIG/XII/2015	04 Jan 2016	Administrator	05 Jan 2016	
4	Pendirian Taman Pendidikan Al-Qur'an (TPQ)	TPQ BABUSSALAM	02/SP/TPQ-B5/I/2016	04 Jan 2016	Administrator	05 Jan 2016	
5	Permohonan ijin magang	Universitas Islam Negeri Maulana Malik Ibrahim Malang	Un.03.2/PP.07/1294/2016	05 Jan 2016	Administrator	05 Jan 2016	
6	Pendistribusian kitab suci Al-Qur'an dan Juz'amma Th 2016	KANWIL	Kw.15.6/4/HK.03.2/7500/2016	05 Jan 2016	Administrator	05 Jan 2016	
7	Pemberitahuan retur SP2D	KPPN	S-5447/WPB.16/KP.032/2016	05 Jan 2016	Administrator	05 Jan 2016	

Gambar 5.17 Halaman Agenda yang Dicetak

97

BAB VI

PENGUJIAN

Pada tahap pengujian kami menggunakan pengujian validasi dengan metode *black box testing* yaitu pengujian yang menganggap sistem sebagai kotak hitam yang menerima input berupa *test case* dan mengeluarkan output berupa *result*, jika hasil *result* sesuai dengan *expected result* maka *case* tersebut dianggap valid. Tujuan dari pengujian adalah untuk menguji fungsionalitas sistem apakah implementasi sistem telah berjalan dengan benar sesuai dengan rancangan yang telah dibuat.

6.1. Perangkat Lunak Pengujian

Tabel 6.1 Perangkat Lunak Pengujian

No.	Jenis Pengujian	Perangkat Lunak Pengujian
1.	<i>Responsiveness</i>	Google Chrome dan Mozilla Firefox untuk PC
2.	Performa	CodeIgniter Benchmarking Class

6.2. Perangkat Keras Pengujian

Spesifikasi perangkat keras untuk pengujian kebutuhan fungsional atau non-fungsional pada tabel 6.2.

Tabel 6.2 Perangkat Keras Pengujian

No.	Asus A451L Notebook	
1.	Prosesor	Intel Core i5 3210M
2.	Sistem operasi	Microsoft Windows 10 64 bit
3.	Kartu grafis	AMD Radeon HD 7670M - 2GB
4.	Layar	14" HD 1366x768
5.	RAM	4GB DDR3
6.	Storage	500GB HDD

6.3. Pengujian Validasi

6.3.1. Pengujian Validasi Kebutuhan Fungsional

Tabel 6.3 Pengujian Validasi Kebutuhan Fungsional

No.	Test Name	Test case	Expected Result	Result	Status
1	Pengujian Login	Pengujian login dengan memasukkan username dan password yang benar	Sistem me- <i>redirect</i> ke halaman home dan menampilkan pesan selamat datang pada halaman utama.	Sistem me- <i>redirect</i> ke halaman home dan menampilkan pesan selamat datang pada halaman utama	Valid
2	Pengujian Logout	Pengujian logout dengan memilih menu logout pada pojok kanan atas layar pada antarmuka home	Sistem menutup antarmuka home dan <i>redirect</i> ke halaman login	Sistem menutup antarmuka home dan <i>redirect</i> ke halaman login	Valid
3	Pengujian Ganti Password	Pengujian dilakukan dengan memilih menu ganti password dan aktor memasukkan password lama dan baru, kemudian menekan button simpan	Sistem memberi pesan bahwa password berhasil diperbaharui	Sistem memberi pesan bahwa password berhasil diperbaharui	Valid
4	Pengujian Klasifikasi Surat	Pengujian dilakukan dengan menambah, mengubah, serta menghapus salah satu klasifikasi surat	Sistem berhasil menambah, mengubah, dan menghapus klasifikasi surat	Sistem berhasil menambah, mengubah, dan menghapus klasifikasi surat	Valid
5	Pengujian Cetak Agenda Surat Masuk	Pengujian dilakukan dengan mencetak agenda surat masuk dari tanggal 1 Januari	Sistem menampilkan agenda surat masuk yang siap dicetak pada	Sistem menampilkan agenda surat masuk yang siap dicetak pada	Valid

		2016 – 31 Oktober 2016	rentang waktu yang ditentukan	rentang waktu yang ditentukan	
6	Pengujian Cetak Agenda Surat Keluar	Pengujian dilakukan dengan mencetak agenda surat keluar dari tanggal 1 Januari 2016 – 30 September 2016	Sistem menampilkan agenda surat keluar yang siap dicetak pada rentang waktu yang ditentukan	Sistem menampilkan agenda surat keluar yang siap dicetak pada rentang waktu yang ditentukan	Valid
7	Pengujian Kelola Surat Masuk	Pengujian dengan memilih link surat masuk untuk melihat semua surat masuk yang ada beserta aksi yang dapat dilakukan	Sistem menampilkan list surat masuk beserta aksi (tambah, edit, hapus, cari, disposisi, dan cetak disposisi)	Sistem menampilkan list surat masuk beserta aksi (tambah, edit, hapus, cari, disposisi, dan cetak disposisi)	Valid
8	Pengujian Cari Surat Masuk	Pengujian dilakukan dengan mengisi kolom pencarian berdasarkan tanggal atau kata kunci	Sistem menampilkan list surat masuk hasil pencarian	Sistem menampilkan list surat masuk hasil pencarian	Valid
9	Pengujian Tambah Surat Masuk	Pengujian dilakukan dengan menekan button tambah data, mengisikan data surat masuk dengan benar kemudian menekan tombol simpan	Sistem menyimpan data surat masuk baru serta menampilkan pesan bahwa surat masuk berhasil ditambah	Sistem menyimpan data surat masuk baru serta menampilkan pesan bahwa surat masuk berhasil ditambah	Valid
10	Pengujian Edit Surat Masuk	Pengujian dilakukan dengan memilih button edit pada surat masuk yang diinginkan, mengisi kolom yang perlu diubah, kemudian menekan button simpan	Sistem menyimpan hasil edit surat masuk dan menampilkan pesan bahwa surat masuk berhasil diperbaharui	Sistem menyimpan hasil edit surat masuk dan menampilkan pesan bahwa surat masuk berhasil diperbaharui	Valid

11	Hapus Surat Masuk	Pengujian dilakukan dengan memilih button hapus pada surat masuk yang diinginkan, kemudian menekan pilihan OK untuk konfirmasi	Sistem menghapus surat masuk dan menampilkan pesan bahwa surat masuk berhasil dihapus dari sistem	Sistem menghapus surat masuk dan menampilkan pesan bahwa surat masuk berhasil dihapus dari sistem	Valid
12	Pengujian Disposisi Surat Masuk	Pengujian dilakukan dengan menekan button disposisi surat masuk yang diinginkan pada list surat masuk, menekan tombol tambah data, kemudian mengisikan data disposisi dengan benar kemudian menekan tombol simpan	Sistem menyimpan data disposisi surat masuk serta menampilkan pesan bahwa disposisi surat masuk berhasil ditambah	Sistem menyimpan data disposisi surat masuk serta menampilkan pesan bahwa disposisi surat masuk berhasil ditambah	Valid
13	Pengujian Cetak Disposisi Surat	Pengujian dilakukan dengan memilih button cetak disposisi pada surat masuk yang diinginkan	Sistem menampilkan halaman cetak disposisi surat masuk	Sistem menampilkan halaman cetak disposisi surat masuk	Valid
14	Pengujian Kelola Surat Keluar	Pengujian dengan memilih link surat keluar untuk melihat semua surat keluar yang ada beserta aksi yang dapat dilakukan	Sistem menampilkan list surat keluar beserta aksi (tambah, edit, hapus, dan cari)	Sistem menampilkan list surat keluar beserta aksi (tambah, edit, hapus, dan cari)	Valid
15	Pengujian Cari Surat Keluar	Pengujian dilakukan dengan mengisi kolom pencarian berdasarkan	Sistem menampilkan list surat keluar hasil pencarian	Sistem menampilkan list surat masuk hasil pencarian	Valid

		tanggal atau kata kunci			
16	Pengujian Tambah Surat Keluar	Pengujian dilakukan dengan menekan button tambah data, mengisikan data surat keluar dengan benar kemudian menekan tombol simpan	Sistem menyimpan data surat keluar baru serta menampilkan pesan bahwa surat keluar berhasil ditambah	Sistem menyimpan data surat keluar baru serta menampilkan pesan bahwa surat keluar berhasil ditambah	Valid
17	Pengujian Edit Surat Keluar	Pengujian dilakukan dengan memilih button edit pada surat keluar yang diinginkan, mengisi kolom yang perlu diubah, kemudian menekan button simpan	Sistem menyimpan hasil edit surat keluar dan menampilkan pesan bahwa surat keluar berhasil diperbaharui	Sistem menyimpan hasil edit surat keluar dan menampilkan pesan bahwa surat keluar berhasil diperbaharui	Valid
18	Pengujian Hapus Surat Keluar	Pengujian dilakukan dengan memilih button hapus pada surat keluar yang diinginkan, kemudian menekan pilihan OK untuk konfirmasi	Sistem menghapus surat keluar dan menampilkan pesan bahwa surat keluar berhasil dihapus dari sistem	Sistem menghapus surat keluar dan menampilkan pesan bahwa surat keluar berhasil dihapus dari sistem	Valid
19	Pengujian Setting Instansi	Pengujian dengan memilih menu setting instansi, mengisi kolom yang ingin diganti, kemudian admin menekan tombol simpan	Sistem menyimpan pengaturan instansi dan menampilkan pesan bahwa pengaturan berhasil disimpan	Sistem menyimpan pengaturan instansi dan menampilkan pesan bahwa pengaturan berhasil disimpan	Valid
20	Pengujian Manage User	Pengujian dengan memilih menu manage user	Sistem menampilkan list user beserta aksi	Sistem menampilkan list user beserta aksi	Valid

		masuk supaya admin dapat melihat user yang ada beserta aksi yang dapat dilakukan	(tambah, edit, hapus, dan cari)	(tambah, edit, hapus, dan cari)	
21	Pengujian Tambah User	Pengujian dilakukan dengan menekan button tambah user, mengisikan data user baru dengan benar kemudian menekan tombol simpan	Sistem menyimpan data user baru serta menampilkan pesan bahwa user berhasil ditambahkan dalam sistem	Sistem menyimpan data user baru serta menampilkan pesan bahwa user berhasil ditambahkan dalam sistem	Valid
22	Pengujian Edit User	Pengujian dilakukan dengan memilih button edit pada user yang diinginkan, mengisi kolom yang perlu diubah, kemudian menekan button simpan	Sistem menyimpan hasil edit user dan menampilkan pesan bahwa user berhasil diperbaharui	Sistem menyimpan hasil edit user dan menampilkan pesan bahwa user berhasil diperbaharui	Valid
23	Pengujian Hapus User	Pengujian dilakukan dengan memilih button hapus pada user yang diinginkan, kemudian menekan pilihan OK untuk konfirmasi	Sistem menghapus user dan menampilkan pesan bahwa user berhasil dihapus dari sistem	Sistem menghapus user dan menampilkan pesan bahwa user berhasil dihapus dari sistem	Valid
24	Pengujian Cari User	Pengujian dilakukan dengan mengisi kolom pencarian user berdasarkan kata kunci	Sistem menampilkan list user hasil pencarian	Sistem menampilkan list user hasil pencarian	Valid

6.3.2. Pengujian Validasi Kebutuhan Non Fungsional

Tabel 6.4 Pengujian Validasi Kebutuhan Non Fungsional

No.	Test Name	Parameter Uji	Result	Status
1.	<i>Responsiveness</i>	Sistem dapat menyesuaikan tampilan diberbagai resolusi layar browser Mozilla Firefox atau Google Chrome.	Tampilan sistem dapat menyesuaikan resolusi layar browser Mozilla Firefox atau Google Chrome.	Valid
2.	Performa	Akses sistem dalam waktu dari 5 detik.	Sistem dapat diakses dalam waktu 5 detik.	Valid

BAB VII

PENUTUP

7.1. Kesimpulan

Dari hasil pengembangan aplikasi “SIMAK” Kementerian Agama Kota Malang, dapat diambil beberapa disimpulkan sebagai berikut:

1. Sistem ini dibuat berdasarkan latar belakang proses tata persuratan yang dilakukan secara manual oleh pegawai Kementerian Agama Kota Malang bagian tata usaha divisi umum sehingga dapat dilakukan komputerisasi dengan bantuan aplikasi “SIMAK” untuk mempermudah dan menghemat waktu.
2. Perancangan aplikasi tata persuratan dibuat sesuai dengan kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan dengan pendekatan *object oriented* menggunakan pemodelan dengan *Use Case*, *Sequence Diagram*, *Class Diagram* dan perancangan basis data menggunakan *Conseptual Data Model*.
3. Aplikasi tata persuratan ini diimplementasikan menggunakan bahasa pemrograman PHP dan basis data MariaDB dan disesuaikan dengan proses perancangan dan analisis kebutuhan sistem, sehingga bisa sesuai dengan kebutuhan pengguna.
4. Alasan menggunakan bahasa pemrograman PHP dan basis data MariaDB adalah karena lebih fleksibel dalam penggunaan, mengurangi *recource* komputasi, *open source*, serta memungkinkan dapat diakses *online* maupun *offline* di berbagai *platform*.
5. Penggunaan *framework* CodeIgniter bertujuan untuk memisahkan antara data (Model), tampilan (View) dan cara bagaimana memprosesnya (Controller) agar mudah dilakukan perombakan dikemudian hari.
6. Hasil pengujian sistem dengan menggunakan metode *blackbox testing* menunjukkan bahwa fungsi-fungsi yang ada didalam aplikasi “SIMAK” secara keseluruhan sudah berjalan sebagaimana mestinya.

7. Berdasarkan dari aplikasi SIMAK pada Kementerian Agama Kota Malang yang telah dibuat, memungkinkan pihak yang bersangkutan untuk menambah menu baru untuk mendukung proses tata persuratan yang lebih kompleks agar sistem ini bisa terus berkembang dan bermanfaat untuk pihak Kementerian Agama Kota Malang khususnya pada bagian tata usaha divisi Umum.

7.2. Saran

Setelah mendemonstrasikan aplikasi “SIMAK” kepada pembimbing PKL, terdapat beberapa saran, yaitu:

1. Penambahan nomor surat dan fungsi editing pada bagian klasifikasi nomor surat.
2. Memiliki *user interface* aplikasi yang sederhana agar memudahkan penggunaan oleh pegawai Kementerian Agama Kota Malang.

DAFTAR PUSTAKA

- Bartholomew, Daniel. 2015. *Getting Started with MariaDB 2nd*. Packt Publishing. Birmingham, UK.
- Fowler, M. 2004. UML Distilled 3th, *Panduan Singkat Bahasa Pemodelan Objek Standar*. Diterjemahkan dari Bahasa Inggris oleh Tim Penerjemah Andi. Andi, Yogyakarta.
- Hidayat, Rahmat. 2010. *Cara Praktis Membangun Website Gratis*. Elex Media Kompuindo, Jakarta.
- Jogiyanto, H.M. 2005. *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Andi, Yogyakarta.
- Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Andi, Yogyakarta.
- Pressman, R.G. 2010. *Software Engineering: A Practitioner's Approach 7th*. McGraw-Hill, New York, USA.
- Prasetio, Adhi. 2010. *Cara Mudah Mendesain Web untuk Pemula*. Media kita, Jakarta.
- Satzinger, Jackson, Burd. 2010. *System Analysis and Design with the Unified Process*. Course Technology, USA.
- Sibero, Alexander F.K., 2011. *Kitab Suci Web Programming*. Mediakom, Yogyakarta.
- STAF IF. 2015. *Spesifikasi Kebutuhan Perangkat Lunak*. Jurusan Teknik Informatika UB, Malang.
- Sofwan, Akhmad. 2007. *Belajar PHP dengan Framework CodeIgniter*. Komunitas eLearning ImuKomputer.Com.
- Sommerville, Ian. 2011. *Software Engineering 9th*. Addison-Wesley, USA.
- Shelly G.B & Rosenblatt H.J., 2012. *System Analysis and Design 9th*. Course Technology, Boston.