

**RANCANG BANGUN SISTEM INFORMASI  
PENGARSIPAN SURAT BANK RAKYAT INDONESIA  
KANTOR WILAYAH MALANG**

**SKRIPSI**

**LABORATORIUM SISTEM INFORMASI**

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun oleh :

**A.MARDHATILLAH SUKANDAR**

**NIM. 105060807111077**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
PROGAM STUDI INFORMATIKA/ILMU KOMPUTER  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA**

**MALANG**

**2014**

**LEMBAR PERSETUJUAN**

**RANCANG BANGUN SISTEM INFORMASI PENGARSIPAN SURAT  
BANK RAKYAT INDONESIA KANTOR WILAYAH MALANG**

**SKRIPSI**

**LABORATORIUM SISTEM INFORMASI**

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

**UNIVERSITAS BRAWIJAYA**



Disusun oleh :

**A.MARDHATILLAH SUKANDAR**

**NIM. 105060807111077**

Skripsi ini telah disetujui dosen pembimbing pada tanggal 17 September 2014.

Malang, 17 September 2014

Dosen Pembimbing I

Dosen Pembimbing II

Diah Priharsari, ST., MT.

Aryo Pinandito, ST., M.MT.

NIK. 83051916110374

**LEMBAR PENGESAHAN**

**RANCANG BANGUN SISTEM INFORMASI PENGARSIPAN SURAT  
BANK RAKYAT INDONESIA KANTOR WILAYAH MALANG**

**SKRIPSI**

**LABORATORIUM SISTEM INFORMASI**

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh :

**A.MARDHATILLAH SUKANDAR**

**NIM. 105060807111077**

Setelah dipertahankan di depan Majelis Penguji pada tanggal 2 Oktober 2014 dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana dalam bidang Ilmu Komputer

Penguji 1

Penguji 2

Aditya Rachmadi, S.ST., MTI  
NIK. 860421 16 1 1 0426

Eriq Muhammad Adams J,ST.,M.Kom  
NIP. 19850410 201212 1 001

Penguji 3

Denny Sagita Rusdianto, S.Kom., M.Kom  
NIK. 851124 06 1 1 0250

Mengetahui  
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.  
NIP. 19670801 199203 1 001

**PERNYATAAN****ORISINALITAS SKRIPSI**

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

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiat, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku yaitu UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70.

Undang-undang No. 20 Tahun 2003 Pasal 25 ayat 2 yang berisi “Lulusan perguruan tinggi yang karya ilmiahnya digunakan untuk memperoleh gelar akademik, profesi, atau vokasi terbukti merupakan jiplakan dicabut gelarnya.” dan Pasal 70 yang berisi “Lulusan yang karya ilmiah yang digunakannya untuk mendapatkan gelar akademik, profesi, atau vokasi sebagaimana dimaksud dalam Pasal 25 ayat (2) terbukti merupakan jiplakan dipidana dengan pidana penjara paling lama dua tahun dan/atau pidana denda paling banyak Rp200.000.000,00 (dua ratus juta rupiah).”

Malang, 17 September 2014

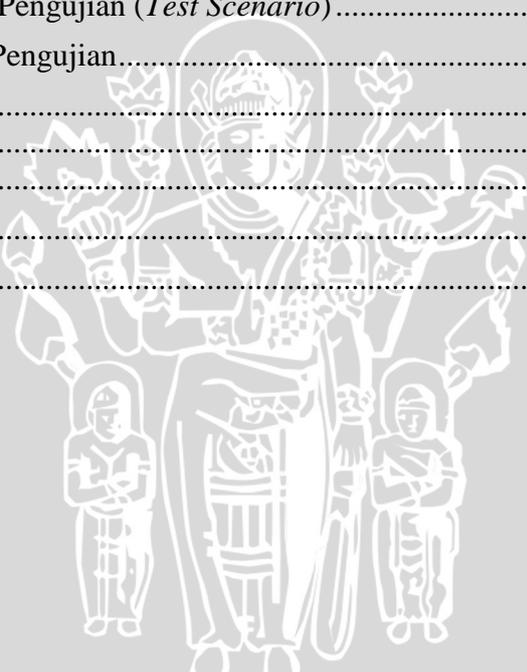
Mahasiswa,

A.Mardhatillah Sukandar  
NIM. 105060807111077

## DAFTAR ISI

|   |     |
|---|-----|
| KATA PENGANTAR .....                            | i   |
| ABSTRAK .....                                   | ii  |
| ABSTRACT .....                                  | iii |
| BAB I .....                                     | 1   |
| PENDAHULUAN .....                               | 1   |
| 1.1. Latar Belakang .....                       | 1   |
| 1.2. Rumusan Masalah .....                      | 2   |
| 1.3. Batasan Masalah .....                      | 3   |
| 1.4. Tujuan .....                               | 3   |
| 1.5. Manfaat .....                              | 3   |
| 1.6. Sistematika Penulisan .....                | 4   |
| BAB II .....                                    | 5   |
| DASAR TEORI .....                               | 5   |
| 2.1. Arsip Surat BRI Kanwil Malang .....        | 5   |
| 2.2. Perancangan Aplikasi Berbasis Objek .....  | 8   |
| 2.1.1. <i>Use Case Diagram</i> .....            | 8   |
| 2.1.2. <i>Activity Diagram</i> .....            | 9   |
| 2.1.3. <i>Class Diagram</i> .....               | 10  |
| 2.1.4. <i>Entity Relationship Diagram</i> ..... | 10  |
| 2.3. Pengujian Aplikasi Berbasis Objek .....    | 10  |
| BAB III .....                                   | 12  |
| METODE PENELITIAN .....                         | 12  |
| 3.1. Pengumpulan Data .....                     | 13  |
| 3.2. Analisis Kebutuhan .....                   | 13  |
| 3.3. Perancangan .....                          | 14  |
| 3.4. Implementasi .....                         | 15  |
| 3.5. Pengujian .....                            | 15  |
| 3.6. Kesimpulan dan Saran .....                 | 16  |
| BAB IV .....                                    | 18  |
| ANALISIS DAN PERANCANGAN .....                  | 18  |
| 4.1. Spesifikasi Kebutuhan .....                | 18  |
| 4.1.1. Pengumpulan Data .....                   | 18  |
| 4.1.2. Analisis Kebutuhan .....                 | 22  |
| 4.2. Diagram Perancangan .....                  | 25  |
| 4.2.1. <i>Use Case Diagram</i> .....            | 25  |

|                                 |   |      |
|---------------------------------|---|------|
| 4.2.2.                          | Activity Diagram.....                   | 28   |
| 4.2.3.                          | Sequence Diagram.....                   | 37   |
| 4.2.4.                          | Class Diagram.....                      | 48   |
| 4.2.5.                          | Entity Relationship Diagram.....        | 57   |
| 4.3.                            | Desain Tampilan Aplikasi.....           | 64   |
| BAB V.....                      |   | 70   |
| IMPLEMENTASI DAN PENGUJIAN..... |   | 70   |
| 5.1.                            | Implementasi.....                       | 70   |
| 5.1.1.                          | Implementasi Database.....              | 70   |
| 5.1.2.                          | Implementasi Perangkat Lunak.....       | 72   |
| 5.2.                            | Pengujian.....                          | 81   |
| 5.2.1.                          | Rencana Pengujian (Test Plan).....      | 82   |
| 5.2.2.                          | Skenario Pengujian (Test Scenario)..... | 82   |
| 5.2.3.                          | Analisis Pengujian.....                 | 85   |
| BAB VI.....                     |   | 89   |
| PENUTUP.....                    |   | 89   |
| 6.1.                            | Kesimpulan.....                         | 89   |
| 6.2.                            | Saran.....                              | 90   |
| DAFTAR PUSTAKA.....             |   | DP-1 |



## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2. 1 Contoh <i>Use Case</i> Diagram.....   | 8  |
| Gambar 2. 2 Contoh <i>Activity</i> Diagram.....   | 9  |
| Gambar 2. 3 Contoh <i>Entity Relationship</i> Diagram (ERD).....  | 11 |
| Gambar 3. 1 Diagram Alir Metode Penelitian .....  | 12 |
| Gambar 4. 1 <i>Use Case</i> Diagram Sistem Informasi Pengarsipan Surat BRI Kanwil<br>Malang .....                                       | 26 |
| Gambar 4. 2 <i>Activity</i> Diagram Pengguna Melihat Surat Masuk .....  | 29 |
| Gambar 4. 3 <i>Activity</i> Diagram Sekretaris Tambah Surat Masuk .....   | 31 |
| Gambar 4. 4 <i>Activity</i> Diagram Seksi Terima Surat Masuk .....  | 32 |
| Gambar 4. 5 <i>Activity</i> Diagram Seksi Membuat Surat Keluar .....  | 34 |
| Gambar 4. 6 <i>Activity</i> Diagram Seksi Membuat Surat Balasan.....  | 36 |
| Gambar 4. 7 <i>Sequence</i> Diagram Pengguna Melihat Surat Masuk .....  | 39 |
| Gambar 4. 8 <i>Sequence</i> Diagram Cari Surat Masuk.....   | 40 |
| Gambar 4. 9 <i>Sequence</i> Diagram Sekretaris Tambah Surat Masuk .....   | 44 |
| Gambar 4. 10 <i>Sequence</i> Diagram Seksi Terima Surat Masuk .....   | 45 |
| Gambar 4. 11 <i>Sequence</i> Diagram Seksi Membuat Surat Keluar .....   | 46 |
| Gambar 4. 12 <i>Sequence</i> Diagram Seksi Membalas Surat Masuk .....   | 47 |
| Gambar 4. 13 <i>Class</i> Diagram Sistem Informasi Pengarsipan Surat Bank Rakyat<br>Indonesia Kantor Wilayah Malang.....                | 51 |
| Gambar 4. 14 <i>Entity Relationship</i> Diagram Sistem Informasi Pengarsipan Surat<br>Bank Rakyat Indonesia Kantor Wilayah Malang ..... | 58 |
| Gambar 4. 15 Desain <i>View</i> ‘suratMasuk’ Akses Dilakukan Sekretaris.....  | 65 |
| Gambar 4. 16 Desain <i>View</i> ‘detailSurat’ dan ‘perbaikiDetailSurat’ .....   | 68 |
| Gambar 4. 17 Desain <i>View</i> ‘balasSuratMasuk’ .....   | 69 |
| Gambar 4. 18 Desain <i>View</i> ‘terimaSuratMasuk’ .....  | 69 |
| Gambar 5. 1 Tabel-tabel <i>Database</i> ‘bri_db’ .....  | 71 |
| Gambar 5. 2 <i>View</i> ‘suratKeluar’ .....   | 73 |
| Gambar 5. 3 Tampilan <i>View</i> ‘detailSuratKeluar’ .....  | 77 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 4. 1 Spesifikasi Kebutuhan Fungsional Sistem Informasi Pengarsipan Surat<br>BRI Kanwil Malang ..... | 23 |
| Tabel 4. 2 Fitur-Fitur Aplikasi Pada <i>Use Case</i> Diagram .....  | 27 |
| Tabel 4. 3 Kelas dan Metode Pada Proses Pengguna Melihat Surat Masuk.....                                 | 41 |
| Tabel 4. 4 Kelas dan Metode Ketika Terdapat Surat Masuk .....   | 41 |
| Tabel 4. 5 Kelas dan Metode Ketika Seksi Membuat Surat Keluar.....  | 42 |
| Tabel 4. 6 Kelas dan Metode Pada Proses Membalas Surat Masuk .....  | 43 |
| Tabel 4. 7 Metode atau Fungsi Kelas ‘ <i>controllersibri</i> ’ Sebagai <i>Controller</i> .....            | 48 |
| Tabel 4. 8 Metode atau Fungsi Kelas ‘ <i>controllerSuratMasuk</i> ’ Sebagai <i>Controller</i><br>.....    | 48 |
| Tabel 4. 9 Metode atau Fungsi Kelas ‘ <i>controllerSuratKeluar</i> ’ Sebagai <i>Controller</i><br>.....   | 49 |
| Tabel 4. 10 Metode / Fungsi Kelas ‘ <i>modelsibri</i> ’ Sebagai <i>Model</i> .....                        | 52 |
| Tabel 4. 11 Metode / Fungsi Kelas ‘ <i>modelSurat</i> ’ Sebagai <i>Model</i> .....                        | 53 |
| Tabel 4. 12 Metode / Fungsi Kelas ‘ <i>modelSuratMasuk</i> ’ Sebagai <i>Model</i> .....                   | 53 |
| Tabel 4. 13 Metode / Fungsi Kelas ‘ <i>modelSuratKeluar</i> ’ Sebagai <i>Model</i> .....                  | 55 |
| Tabel 4. 14 Kelas-kelas Sebagai <i>View</i> .....   | 57 |
| Tabel 4. 15 Penjelasan Kolom Tabel ‘ <i>pengguna</i> ’ .....  | 59 |
| Tabel 4. 16 Penjelasan Kolom Tabel ‘ <i>seksi</i> ’ .....   | 60 |
| Tabel 4. 17 Penjelasan Kolom Tabel ‘ <i>surat</i> ’ .....   | 60 |
| Tabel 4. 18 Penjelasan Kolom Tabel ‘ <i>surat_keluar</i> ’ .....  | 62 |
| Tabel 4. 19 Penjelasan Kolom Tabel ‘ <i>surat_masuk</i> ’ .....   | 63 |
| <br>  |    |
| Tabel 5. 1 Skenario Pengujian 7 .....   | 84 |
| Tabel 5. 2 Rekapitulasi Hasil Skenario Pengujian Aplikasi .....   | 87 |

**DAFTAR KODE**

Kode 5. 1 *Query* Untuk Membuat Tabel ‘surat’, ‘surat\_masuk’, dan ‘surat\_keluar’ ..... 72

Kode 5. 2 Cuplikan Kode *View* ‘suratKeluar’ ..... 74

Kode 5. 3 Cuplikan Kode Program Fungsi ‘tambahSuratKeluar’ Kelas ‘controllerSuratKeluar’ ..... 75

Kode 5. 4 Cuplikan Kode Program Fungsi ‘no\_surat’ Kelas ‘modelSuratKeluar’ 76

Kode 5. 5 Cuplikan Kode Program *View* ‘detailSuratKeluar’ ..... 77

Kode 5. 6 Cuplikan Kode Program Fungsi ‘reg’ Kelas ‘modelSurat’ ..... 79

Kode 5. 7 Cuplikan Kode Program Fungsi ‘tambahSuratKeluar’ Kelas ‘controllerSuratKeluar’ ..... 80

Kode 5. 8 Cuplikan Kode Program Fungsi ‘suratKeluar’ Kelas ‘modelSuratKeluar’ ..... 81

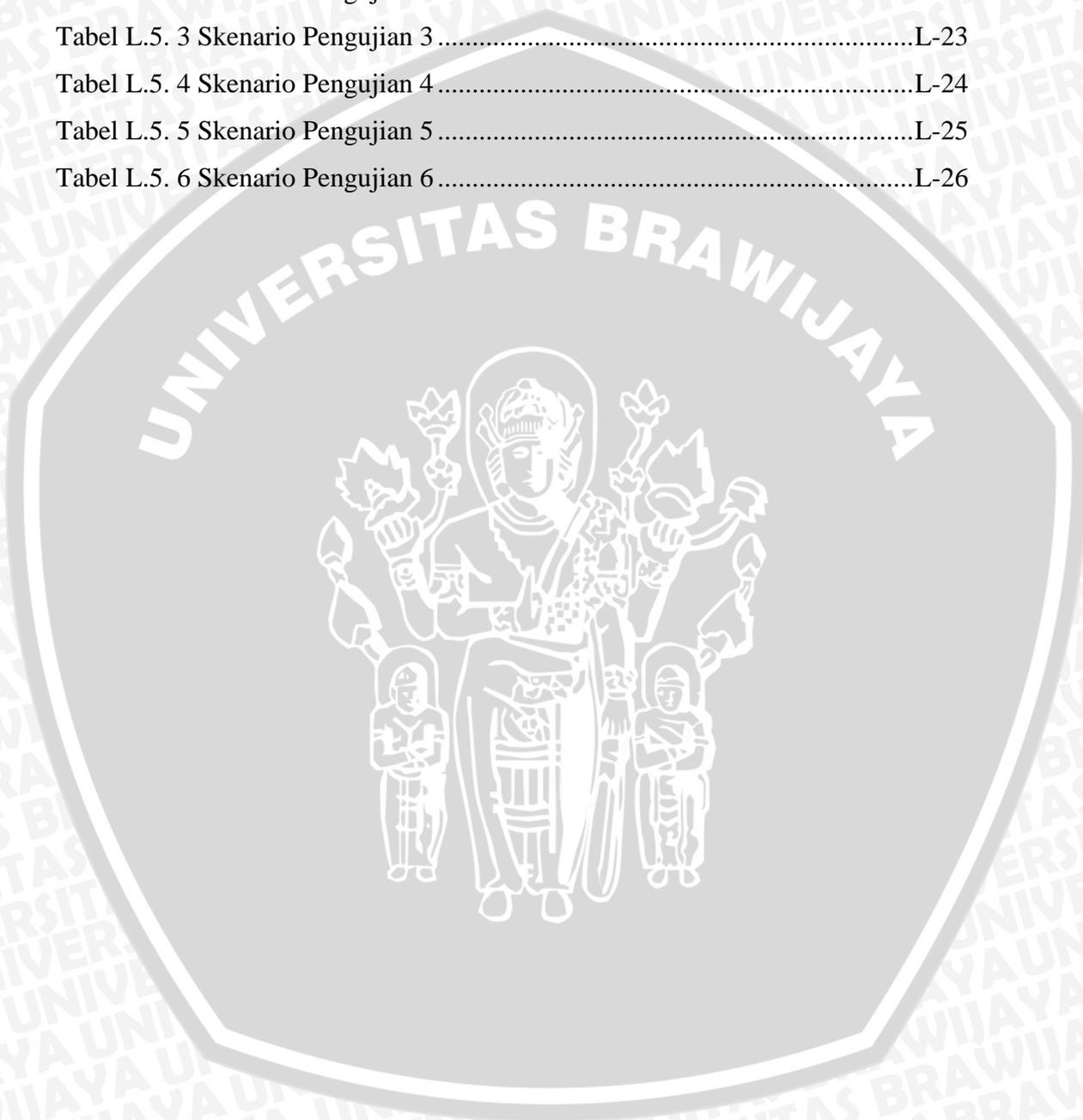


## DAFTAR LAMPIRAN

|   |      |
|---|------|
| LAMPIRAN .....  | L-1  |
| LAMPIRAN 1  |      |
| ACTIVITY DIAGRAM.....   | L-2  |
| Gambar L.1. 1 Activity Diagram <i>Log In</i> .....  | L-3  |
| Gambar L.1. 2 Activity Diagram Pengguna Melihat Surat Keluar.....                                   | L-3  |
| LAMPIRAN 2  |      |
| PERBANDINGAN PROSEDUR PENGARSIPAN SURAT.....  | L-4  |
| Gambar L.2. 1 Perbandingan Prosedur Lama dan Prosedur Baru Proses Ketika Terdapat Surat Masuk ..... | L-5  |
| Gambar L.2. 2 Perbandingan Prosedur Lama dan Prosedur Baru Proses Seksi Membuat Surat Keluar .....  | L-6  |
| Gambar L.2. 3 Perbandingan Prosedur Lama dan Prosedur Baru Proses Seksi Membuat Surat Balasan ..... | L-7  |
| LAMPIRAN 3  |      |
| SEQUENCE DIAGRAM .....  | L-8  |
| Gambar L.3. 1 Sequence Diagram <i>Log In</i> .....  | L-9  |
| Gambar L.3. 2 Sequence Diagram Pengguna Melihat Surat Keluar.....                                   | L-10 |
| Gambar L.3. 3 Sequence Diagram Cari Surat Keluar.....   | L-11 |
| LAMPIRAN 4  |      |
| DESAIN TAMPILAN APLIKASI.....   | L-12 |
| Gambar L.4. 1 Desain View 'login' .....   | L-13 |
| Gambar L.4. 2 Desain View 'dataUser' Akses Dilakukan Sekretaris.....                                | L-13 |
| Gambar L.4. 3 Desain View 'dataUser' Akses Dilakukan Seksi .....                                    | L-13 |
| Gambar L.4. 4 Desain View 'perbaruiDataUser' .....  | L-13 |
| Gambar L.4. 5 Desain View 'suratMasuk' Akses Dilakukan Seksi.....                                   | L-14 |
| Gambar L.4. 6 Desain View 'suratKeluar' Akses Dilakukan Sekretaris.....                             | L-15 |
| Gambar L.4. 7 Desain View 'suratKeluar' Akses Dilakukan Seksi.....                                  | L-16 |
| Gambar L.4. 8 Desain View 'detailSuratKeluar' dan 'perbaikiDetailSuratKeluar' .....                 | L-17 |

LAMPIRAN 5

|   |      |
|---|------|
| SKENARIO PENGUJIAN.....                 | L-18 |
| Tabel L.5. 1 Skenario Pengujian 1 ..... | L-19 |
| Tabel L.5. 2 Skenario Pengujian 2 ..... | L-22 |
| Tabel L.5. 3 Skenario Pengujian 3 ..... | L-23 |
| Tabel L.5. 4 Skenario Pengujian 4 ..... | L-24 |
| Tabel L.5. 5 Skenario Pengujian 5 ..... | L-25 |
| Tabel L.5. 6 Skenario Pengujian 6 ..... | L-26 |



## KATA PENGANTAR

Alhamdulillah, puji syukur kepada Allah SWT yang telah memberikan rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Informasi Pengarsipan Surat Bank Rakyat Indonesia Kantor Wilayah Malang”. Skripsi ini diajukan kepada Universitas Brawijaya Malang untuk memenuhi salah satu persyaratan dalam memperoleh gelar Sarjana Komputer.

Selesaiannya skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu dengan kerendahan hati saya menyampaikan banyak terima kasih kepada:

1. A.Sukandar Rifa'i dan Supik Rifa'i selaku orang tua saya yang selalu memberikan dukungan moral dan materi sehingga diselesaikannya skripsi ini.
2. Seluruh keluarga saya yang ada dimalang yang selalu memberikan yang terbaik untuk saya selama menempuh masa perkuliahan dan khususnya dalam proses penyelesaian skripsi.
3. Seluruh staf dan dosen Program Teknologi Informasi dan Ilmu Komputer yang telah membina saya selama menjadi mahasiswa Universitas Brawijaya Malang.
4. Diah Priharsari, ST., MT dan Aryo Pinandito, ST., M.MT selaku pembimbing I dan pembimbing II yang telah memberikan bimbingan dan arahan dengan penuh kesabaran sehingga skripsi ini dapat diselesaikan.
5. Bank Rakyat Indonesia Kantor Wilayah Malang yang telah menyediakan tempat dan memperbolehkan saya untuk melakukan penelitian.
6. Seluruh teman-teman saya.

Saya menyadari sepenuhnya bahwa skripsi ini masih belum sempurna, oleh karena itu dengan segala kerendahan hati saya mengharapkan kritik dan saran untuk kesempurnaan skripsi ini.

Malang, 17 September 2014

Penulis

A.Mardhatillah Sukandar . 2014 . Rancang Bangun Sistem Informasi Pengarsipan Surat Bank Rakyat Indonesia Kantor Wilayah Malang . Program Studi Informatika/Ilmu Komputer . Program Teknologi Informasi dan Ilmu Komputer . Pembimbing: Diah Priharsari, ST., MT. dan Aryo Pinandito, ST., M.MT.

### ABSTRAK

Bank Rakyat Indonesia Kantor Wilayah (BRI Kanwil) Malang mengalami beberapa permasalahan terkait pengarsipan surat. Permasalahan pengarsipan surat yang terjadi adalah BRI Kanwil Malang membutuhkan ruang besar untuk menyimpan arsip surat, pemborosan kertas karena arsip surat asli dipegang oleh sekretaris dan fotokopi arsip surat dipegang oleh bagian seksi, sekretaris dan seksi mengalami kesulitan untuk menemukan arsip surat, dan terdapat kesalahan melakukan penomoran surat (nomor surat atau nomor register). Permasalahan pengarsipan yang terjadi di BRI Kanwil Malang tentu dapat diselesaikan dengan menggunakan teknologi yang ada saat ini. Untuk menyelesaikan masalah-masalah tersebut langkah pertama adalah mengumpulkan data yang dibutuhkan, analisis kebutuhan, perancangan, implementasi, dan pengujian. Penelitian ini menghasilkan aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang serta hasil pengujian menunjukkan kemampuan aplikasi dalam menyelesaikan masalah yang ada.

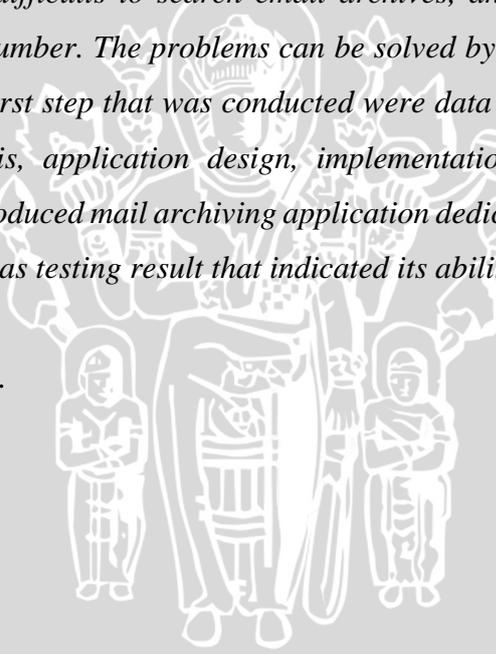
Kata kunci : arsip surat.

A.Mardhatillah Sukandar . 2014 . *Design and Build Mail Archiving Application Dedicated to BRI Malang Regional Office* . Program Studi Informatika/Ilmu Komputer . Program Teknologi Informasi dan Ilmu Komputer . Advisors: Diah Priharsari, ST., MT. and Aryo Pinandito, ST., M.MT.

### **ABSTRACT**

*Bank Rakyat Indonesia (BRI) Malang Regional Office had some issues related to mail archiving. The problems in Bank Rakyat Indonesia Malang Regional Office were needs a large space for storing mail archives, wastes of paper because the original records of letters held by each section and another copy of the letters held by the secretary, difficults to search email archives, and faces problem in generatif unique mail number. The problems can be solved by using technologies. To solve the problem, first step that was conducted were data gathering, followed by requirement analysis, application design, implementation, and application testing. The research produced mail archiving application dedicated to BRI Malang Regional Office as well as testing result that indicated its ability to solve identified problems.*

*Keyword: mail archives.*



## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Arsip merupakan salah satu alat otentik di sebuah instansi atau perusahaan. Semua dokumen atau arsip akan dipelihara dengan baik dan akan digunakan kembali jika suatu saat diperlukan. Keberadaan arsip dalam organisasi menjadi salah satu faktor yang sangat berperan penting dan juga merupakan penentu dalam proses pelaksanaan tugas organisasi (Tahir, 2007).

Bank Rakyat Indonesia Kantor Wilayah Malang (BRI Kanwil Malang) mengalami permasalahan pengarsipan terkait surat. Permasalahan terjadi karena proses surat menyurat yang terjadi di Bank Rakyat Indonesia masih menggunakan cara konvensional. Permasalahan pengarsipan surat yang muncul adalah membutuhkan ruang besar untuk menyimpan arsip surat karena arsip surat asli dipegang oleh sekretaris dan fotokopi surat dipegang oleh bagian seksi. Oleh karena itu hal ini akan menyebabkan pemborosan kertas, ketika seksi atau sekretaris membutuhkan arsip surat karena arsip surat menumpuk maka arsip surat sulit ditemukan bahkan tidak bisa ditemukan, dan terjadi kesalahan dalam melakukan penomoran surat.

Permasalahan ruang penyimpanan arsip surat fisik membuat BRI kanwil Malang harus selalu mempersiapkan ruang tambahan untuk menyimpan arsip surat dan melakukan pemusnahan terhadap arsip surat-surat lama secara berkala meskipun bisa saja sewaktu-waktu surat yang dimusnahkan itu dibutuhkan. Permasalahan penyimpanan arsip surat fisik tersebut menimbulkan permasalahan baru ketika pegawai BRI Kanwil Malang harus menemukan arsip surat dari beberapa bulan atau beberapa tahun yang lalu. Bukan hanya kesulitan dalam menemukan bisa saja arsip surat yang dicari tidak dapat ditemukan karena arsip surat rusak, hilang, atau sudah dimusnahkan. Permasalahan dalam proses surat menyurat juga muncul ketika sekretaris salah dalam memberikan nomor surat atau nomor register. Kesalahan dalam penomoran ini dapat mengakibatkan kesalahan rekapitulasi arsip surat dan kesalahan letak penyimpanan surat.

Permasalahan pengarsipan yang terjadi di BRI Kanwil Malang tentu dapat diselesaikan dengan menggunakan teknologi yang ada saat ini. Beberapa aplikasi sistem informasi komersial telah tersedia di pasaran untuk mengatasi permasalahan arsip surat seperti *e-filling* dari *digital-sense*, dan Sistem Informasi Disposisi Surat dari cabikhosting. Aplikasi *e-filling* memiliki manajemen atau prosedur pengarsipan surat yang telah tertanam dan memiliki tabel rekapitulasi arsip berbeda dengan yang dibutuhkan oleh BRI Kanwil Malang (Ikhsan, 2010) (Editor, 2012). Aplikasi Sistem Informasi Disposisi Surat hanya digunakan untuk pengarsipan dan pengecekan disposisi pada surat keluar saja (Editor, 2013).

Aplikasi sistem informasi komersial di pasaran yang telah disebutkan di atas tidak digunakan oleh BRI Kanwil Malang karena terdapat dua alasan utama. Alasan pertama adalah aplikasi-aplikasi tersebut tidak memenuhi kebutuhan BRI Kanwil Malang sehingga permasalahan juga tidak dapat diselesaikan, seperti tidak cocoknya tabel rekapitulasi arsip surat seperti pada aplikasi *e-filling* serta tidak ada fitur untuk keperluan arsip surat masuk pada aplikasi Sistem Informasi Disposisi Surat. Maka dibutuhkan aplikasi pengarsipan surat yang benar-benar sesuai dengan kebutuhan BRI Kanwil Malang. Alasan kedua kenapa aplikasi di pasaran tidak digunakan oleh BRI Kanwil Malang adalah karena aplikasi-aplikasi tersebut telah memiliki prosedur pengarsipan surat sendiri yang jauh berbeda dengan prosedur yang terdapat di BRI Kanwil Malang seperti dalam aplikasi *e-filling*. Oleh karena itu BRI Kanwil Malang membutuhkan aplikasi sistem informasi pengarsipan surat yang mengadopsi prosedur pengarsipan surat yang terdapat di BRI Kanwil Malang. Mengadopsi prosedur yang lama dimaksudkan bahwa ketika proses pengarsipan surat di BRI Kanwil Malang beralih dari proses manual ke proses komputerisasi maka prosedur pengarsipan surat yang digunakan nantinya tidak jauh berbeda dengan prosedur pengarsipan yang lama.

## 1.2. Rumusan Masalah

Adapun rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana permasalahan pengarsipan surat di BRI Kanwil Malang dapat diatasi dengan aplikasi sistem informasi pengarsipan surat?

2. Bagaimana prosedur baru pengarsipan surat di BRI Kanwil Malang dengan adanya sistem informasi pengarsipan surat?

### 1.3. Batasan Masalah

Penelitian ini memiliki batasan masalah sebagai berikut:

1. Seksi adalah bidang kerja di BRI Kanwil Malang. Dalam suatu seksi yang dapat melakukan akses terhadap aplikasi sistem informasi pengarsipan surat adalah salah satu orang yang memiliki tugas mengarsipkan surat di seksi tersebut.
2. Permasalahan pengarsipan yang dimaksud adalah arsip surat kantor BRI Kanwil Malang sesuai dengan definisi BRI Kanwil Malang bukan milik seksi.
3. Pada penelitian ini menekankan pada fungsional sistem sehingga desain antar muka digunakan untuk menunjukkan fungsional sistem saja.

### 1.4. Tujuan

Tujuan dari penelitian ini adalah :

1. Membuat aplikasi yang dapat menemukan berkas surat, melakukan penomoran surat secara otomatis, dan menyimpan arsip surat secara digital sesuai kebutuhan dan prosedur BRI Kanwil Malang.
2. Membuat prosedur pengarsipan surat baru yang telah disesuaikan dengan sistem informasi pengarsipan surat BRI Kanwil Malang.

### 1.5. Manfaat

Manfaat yang diperoleh dari sistem informasi hasil penelitian ini adalah :

1. Bank Rakyat Indonesia Kantor Wilayah Malang hanya perlu ruang penyimpanan atau brankas arsip fisik surat asli.
2. Sekretaris tidak perlu melakukan pencatatan nomor surat secara manual karena nomor surat akan dibuat secara otomatis oleh aplikasi.
3. Sekretaris dan para seksi dapat lebih efisien dan efektif dalam mencari arsip surat.

## **1.6. Sistematika Penulisan**

### **BAB I PENDAHULUAN**

Memuat latar belakang penelitian ini, rumusan masalah tentang beberapa masalah yang ingin diselesaikan, tujuan dari penelitian, batasan masalah agar penelitian tidak melebar pada bidang lain dan fokus penelitian tetap pada masalah yang ingin diselesaikan, manfaat penelitian, dan sistematika penulisan.

### **BAB II DASAR TEORI**

Berisikan penjelasan dasar teori dari berbagai sumber pustaka (literatur) untuk menunjang penyelesaian masalah pengarsipan surat di Bank Rakyat Indonesia Kantor Wilayah Malang.

### **BAB III METODE PENELITIAN**

Membahas mengenai proses penelitian sehingga didapatkan kebutuhan aplikasi dari Bank Rakyat Indonesia Kantor Wilayah Malang.

### **BAB IV ANALISIS DAN PERANCANGAN**

Menjelaskan proses perancangan sistem informasi pengarsipan surat untuk Bank Rakyat Indonesia Kantor Wilayah Malang mulai dari pengumpulan data hingga perancangan aplikasi.

### **BAB V IMPLEMENTASI DAN PENGUJIAN**

Membahas tentang implementasi pembuatan aplikasi dan pengujian pada aplikasi yang dilakukan berdasarkan metode penelitian dan perancangan dari sistem yang telah dibuat lalu menguraikan hasil analisis dari pengujian.

### **BAB VI PENUTUP**

Memuat kesimpulan dari keseluruhan penelitian dan saran untuk pengembangan lebih lanjut.

## BAB II

### DASAR TEORI

Tahap ini berisikan penjelasan dasar teori dari berbagai sumber pustaka (literatur) untuk menunjang penyelesaian masalah pengarsipan surat di Bank Rakyat Indonesia Kantor Wilayah Malang.

#### 2.1. Arsip Surat BRI Kanwil Malang

Arsip surat Bank Rakyat Indonesia (BRI) dibedakan menjadi dua bagian yaitu arsip surat Kantor dan arsip surat seksi. Perbedaan dari arsip surat kantor dan arsip surat seksi terletak pada kepemilikan arsip. Arsip kantor BRI merupakan arsip kantor yang dalam proses pengarsipannya memungkinkan seksi terlibat di dalamnya, arsip surat seksi hanya merupakan pengarsipan surat milik seksi dengan proses serta ketentuan berbeda tiap seksi dan tidak ada campur tangan sekretaris kantor BRI karena arsip ini merupakan otoritas dari masing-masing seksi. Arsip surat seksi digunakan untuk keperluan surat menyurat antar seksi dalam satu kantor. Namun surat yang ditujukan secara spesifik pada pegawai maka surat itu diarsipkan dalam arsip surat kantor BRI. Surat masuk dan surat keluar milik seksi dalam arsip surat seksi tidak dianggap surat masuk dan surat keluar pada arsip surat Kantor. Pada penelitian ini arsip surat yang dimaksud adalah arsip surat kantor BRI Kanwil Malang.

Dalam ruang lingkup arsip surat, BRI atau khususnya BRI Kanwil Malang memiliki definisi surat yang digunakan lebih spesifik terkait surat masuk, surat keluar, dan surat balasan. Definisi surat masuk bagi BRI Kanwil Malang adalah surat yang ditujukan kepada BRI Kanwil Malang atau Pimpinan BRI Kanwil Malang, diterima oleh sekretaris dan harus memenuhi syarat-syarat tertentu. Syarat surat untuk dikatakan sebagai surat masuk BRI Kanwil Malang adalah sebagai berikut:

1. Mencantumkan asal surat (nama atau instansi pengirim).
2. Harus jelas surat tersebut ditujukan ke mana, bisa secara umum surat ditujukan ke BRI Kanwil Malang, atau secara spesifik ditujukan pada Pimpinan BRI

Kanwil Malang atau langsung pada seksi tertentu. Ketika terjadi kasus surat masuk ditujukan secara umum pada BRI Kanwil Malang atau pada Pimpinan BRI Kanwil Malang maka Pimpinan BRI Kanwil Malang berhak menyimpan surat tersebut tanpa dilakukan pengarsipan atau secara langsung memberi instruksi pada sekretaris agar dilakukan pengarsipan untuk kemudian ditindak lanjuti oleh seksi tertentu.

3. Perihal surat harus jelas. Perihal surat dapat secara tertulis terdapat dalam surat apabila surat tersebut adalah surat formal (seperti surat dari instansi) atau semi formal (seperti surat lamaran kerja atau permohonan magang). Namun perihal surat juga dapat disimpulkan dari isi surat tersebut apabila surat tersebut adalah surat tidak formal (seperti surat dari nasabah).

BRI Kanwil Malang membagi surat masuk menjadi dua, yaitu surat masuk rahasia dan surat masuk biasa. Surat masuk rahasia adalah surat masuk yang hanya diketahui asal surat atau pengirim surat dan tujuan surat dalam hal ini tujuan surat adalah BRI Kanwil Malang atau Pimpinan BRI Kanwil Malang. Pada surat masuk rahasia biasanya terdapat stempel "RAHASIA" dan sekretaris tidak boleh membuka amplop surat. Namun jika tidak terdapat stempel "RAHASIA" maka sekretaris akan melakukan konfirmasi pada pengirim surat apakah surat itu boleh dibuka atau tidak, apabila surat tidak boleh dibuka dapat disimpulkan surat tersebut adalah surat rahasia. Surat rahasia harus diserahkan pada Pimpinan BRI Kanwil Malang. Pimpinan BRI Kanwil Malang berhak menyimpan surat rahasia tersebut tanpa dilakukan pengarsipan atau secara langsung memberi instruksi pada sekretaris agar dilakukan pengarsipan untuk kemudian ditindak lanjuti oleh seksi tertentu. Sedangkan surat masuk biasa adalah surat masuk secara umum memenuhi syarat surat masuk milik BRI Kanwil Malang.

Surat masuk yang diterima oleh BRI Kanwil Malang memungkinkan untuk ditindak lanjuti oleh lebih dari satu seksi. Kasus ini mungkin terjadi karena memang surat masuk ditujukan lebih dari satu seksi. Pada kasus lain hal ini bisa terjadi karena surat masuk secara umum ditujukan pada BRI Kanwil Malang atau secara khusus pada Pimpinan BRI Kanwil Malang dan Pimpinan BRI Kanwil Malang dapat menunjuk seksi mana saja yang harus menindak lanjuti surat masuk tersebut. Ketika menerima surat masuk harus diterima atau ditindaklanjuti lebih dari satu

seksi maka surat tersebut harus difotokopi dan jumlah surat masuk yang diterima dianggap sejumlah seksi yang harus menindaklanjuti surat. Jadi misalkan terdapat satu surat masuk berupa surat edaran yang harus ditindaklanjuti oleh tiga seksi maka surat masuk tersebut difotokopi dan dianggap terdapat tiga surat masuk yang harus diarsipkan.

Surat keluar adalah surat yang dibuat oleh seksi, dikirim oleh BRI Kawil Malang melalui sekretaris dan harus memenuhi syarat-syarat tertentu. Surat balasan merupakan surat keluar yang membalas surat masuk, jadi dapat disimpulkan surat balasan adalah bagian dari surat keluar. Syarat surat untuk dikatakan sebagai surat keluar BRI Kanwil Malang adalah sebagai berikut:

1. Mencantumkan seksi atau nama dan jabatan yang mengirimkan surat. Dalam melakukan pengarsipan surat keluar sekretaris hanya mencatat seksi yang bertanggung jawab terhadap surat keluar yang dibuat.
2. Harus jelas surat tersebut ditujukan ke mana.
3. Perihal surat harus jelas dan tertulis pada kolom perihal.

Setelah surat masuk dan surat keluar memenuhi syarat dan ketentuan di atas maka pengarsipan surat dapat dilakukan. Dalam melakukan pengarsipan surat dikenal dua proses penomoran, yaitu nomor register dan nomor surat. Nomor register digunakan secara umum untuk surat masuk ataupun surat keluar untuk menjadi kode singkat tentang hubungan antara urutan proses surat atau total surat dalam arsip dan waktu ketika surat diproses. Sebagai contoh total arsip surat (surat masuk dan surat keluar) BRI Kanwil Malang pada tanggal 15 bulan September tahun 2009 berjumlah 401 surat, pada tanggal tersebut terdapat surat masuk maka nomor register surat masuk tersebut adalah 20090915402 dan setelah itu terdapat surat keluar maka nomor register surat keluar tersebut adalah 20090915403. Nomor register juga dapat digunakan untuk mengetahui hubungan antara surat masuk dan surat keluar karena surat keluar akan memiliki nomor register yang sama dengan salah satu surat masuk apabila surat keluar itu untuk membalas surat masuk tersebut. Sebagai contoh misalkan surat masuk yang akan dibalas memiliki nomor register 20090915403 maka surat balasan yang dibuat memiliki nomor register 20090915403 karena surat balasan dianggap dibuat untuk memproses surat masuk tersebut meskipun surat balasan dibuat bukan pada waktu yang sama seperti surat

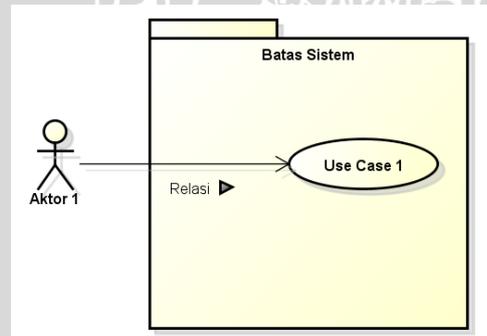
masuk yang diterima. Nomor surat digunakan secara khusus hanya untuk penomoran surat keluar termasuk surat balasan. Nomor surat ini berurutan, digunakan untuk mengetahui urutan surat keluar dan total surat yang telah dikeluarkan.

## 2.2. Perancangan Aplikasi Berbasis Objek

Perancangan yang dilakukan untuk membuat aplikasi berbasis objek cukup menggunakan *use case* diagram, *activity* diagram, *class* diagram (Nugroho, 2009:14). Sedangkan karena aplikasi menyimpan data dalam bentuk digital di dalam *database* maka *entity relationship* diagram penting digunakan untuk menggambarkan *database* yang digunakan oleh aplikasi.

### 2.1.1. Use Case Diagram

Pressman (2001: 280) menyatakan bahwa sebuah *use case* diagram adalah skenario yang menggambarkan bagaimana perangkat lunak yang akan digunakan dalam situasi tertentu. Hasil dari analisis kebutuhan direpresentasikan dengan pembuatan *use case* diagram yang sesungguhnya merupakan deskripsi perangkat tinggi bagaimana perangkat lunak akan digunakan oleh penggunanya.

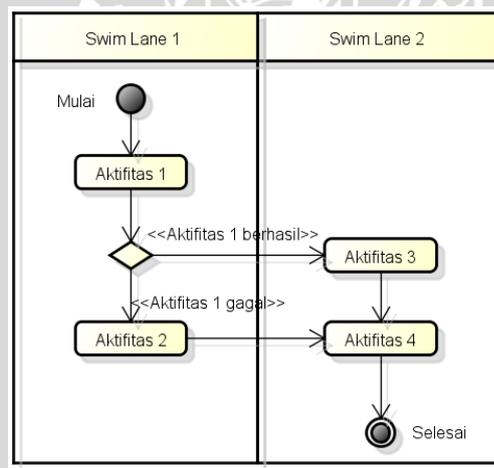


**Gambar 2. 1 Contoh Use Case Diagram**

Dalam *use case* diagram dikenal 4 istilah yang biasa digunakan, yaitu aktor, *use case*, *system boundary* (batas sistem), dan relasi seperti dalam Gambar 2.1. Aktor merupakan suatu perwujudan dari pengguna yang memiliki peran terhadap aplikasi menggunakan fitur dalam aplikasi berupa *use case* dihubungkan dengan garis relasi. Beberapa *use case* biasanya berada dalam satu *system boundary* (batas sistem) untuk menunjukkan bahwa suatu aplikasi memiliki batasan fungsionalitas.

Garis relasi juga dapat digunakan antar *use case* untuk menunjukkan hubungan antar *use case*. Seperti relasi *include*, yaitu relasi yang menggambarkan suatu *use case* adalah bagian dari *use case* lainnya. Relasi ini menunjukkan bahwa ketika suatu *use case* (*parent use case*) dijalankan maka *use case* lain (*child use case*) pasti akan dijalankan juga. Relasi ini berupa garis putus-putus dengan kepala panah menghadap pada *child use case*. Kemudian terdapat relasi *extend*, yaitu relasi yang menunjukkan bahwa suatu *use case* memiliki fungsionalitas tambahan. Relasi ini menunjukkan bahwa ketika suatu *use case* (*parent use case*) dijalankan maka aktor dapat menjalankan *use case* lainnya (*child use case*). Relasi ini berupa garis putus-putus dengan kepala panah menghadap pada *parent use case*.

### 2.1.2. Activity Diagram



**Gambar 2. 2 Contoh Activity Diagram**

Meskipun *use case* merupakan gambaran menyeluruh dari aplikasi namun kurang terperinci, maka dari itu *activity diagram* digunakan untuk mendapatkan gambaran yang lebih menyeluruh (Nugroho, 2009). *Activity diagram* digunakan untuk menggambarkan urutan aktivitas atau prosedur suatu aplikasi seperti dalam Gambar 2.2. Pada diagram ini juga terdapat *swim lane* (garis tegak lurus dari atas ke bawah) yang menunjukkan *state* berada pada objek atau kelas tertentu. *Swim lane* tersebut akan mempermudah adalah menentukan kelas untuk aplikasi berbasis objek seperti *codeigniter*.

### 2.1.3. Class Diagram

*Class* diagram digunakan untuk mengelompokkan atribut dan fungsi yang terdapat dalam kelas. Terdapat dua cara untuk menentukan atribut dan metode atau fungsi yang terlibat dalam suatu kelas yaitu bisa menggunakan *data flow* diagram (DFD) (Lee dan Tepfenhart, 1999) atau menggunakan *sequence* diagram (Nugroho, 2009). Selain atribut dan metode atau fungsi pada *class* diagram bisa ditentukan berdasarkan skenario *data flow* diagram (DFD) atau menggunakan *sequence* diagram, atribut dan metode dapat ditambahkan berdasarkan pertimbangan bahwa memang diperlukan dalam aplikasi.

### 2.1.4. Entity Relationship Diagram

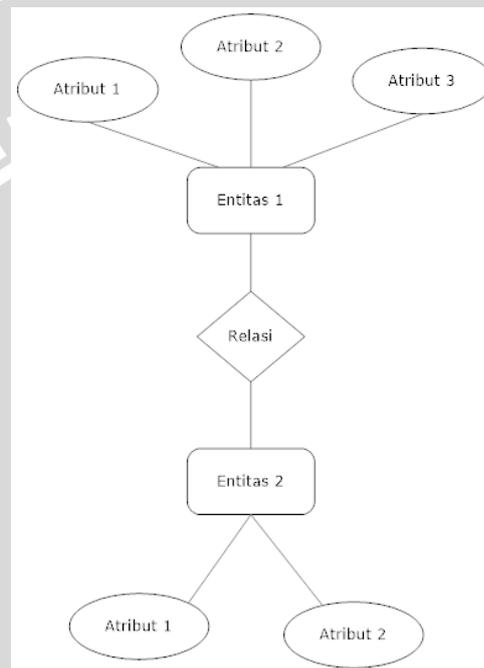
*Entity relationship* diagram (ERD) digunakan untuk menggambarkan model logika dari *database* yang akan digunakan dalam aplikasi. Biasanya dalam ERD juga disertakan analisis terkait entitas yang akan menjadi tabel beserta atribut (kolom) dalam *database*. Antar entitas memiliki hubungan atau relasi yang memungkinkan terjadi interaksi antar entitas. Berikut dalam Gambar 2.3 menunjukkan contoh ERD.

## 2.3. Pengujian Aplikasi Berbasis Objek

Pendekatan dasar terhadap pengujian berorientasi objek dibagi menjadi pengujian kelas objek dan pengujian integrasi objek (Sommerville, 2001). Pengujian kelas objek merupakan pengujian kotak putih yang disebut juga pengujian unit namun diperluas untuk mencakup rangkaian operasi yang berhubungan. Pengujian ini berhubungan dengan pemberian jaminan bahwa *statement* program dieksekusi paling tidak sekali dan bahwa semua jalur dalam program dieksekusi untuk mengetahui kesalahan dalam program (Sommerville, 2007).

Pengujian integrasi objek atau pengujian *release* (Sommerville, 2007) merupakan salah satu tipe pengujian sistem. Pengujian sistem digunakan pada aplikasi yang mengintegrasikan dua atau lebih komponen yang melaksanakan fungsi-fungsi atau fitur sistem. Pengujian sistem dibagi menjadi *integration testing*

(pengujian integrasi) dan *release testing* (pengujian integrasi objek). Aplikasi berorientasi objek sangat sulit untuk menerapkan pengujian integrasi yang berbasis *top-down* atau *bottom-up* karena pada aplikasi berorientasi objek tidak ada hierarki objek yang jelas (Sommerville, 2001). Oleh karena itu *release testing* (pengujian integrasi objek) yang merupakan pengujian kotak hitam lebih tepat digunakan, pengujian ini dilakukan dengan pendekatan berdasarkan operasi atau fitur dalam sistem atau aplikasi.

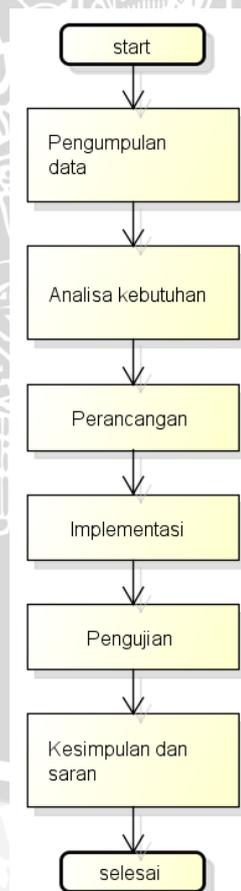


**Gambar 2. 3 Contoh *Entity Relationship Diagram* (ERD)**

### BAB III

#### METODE PENELITIAN

Pada tahap ini akan dijelaskan mengenai metode penelitian yang digunakan untuk menjawab permasalahan penelitian yang telah disebutkan sebelumnya sehingga penelitian ini dapat mencapai tujuan yang diharapkan. Untuk dapat membangun aplikasi sistem informasi pengarsipan surat yang dapat menyelesaikan permasalahan pengarsipan surat di BRI Kanwil Malang maka harus dipahami terlebih dahulu permasalahan arsip surat yang terjadi, dan kebutuhan BRI Kanwil Malang terkait aplikasi sistem informasi pengarsipan surat. Setelah itu dilakukan perancangan dan pembuatan aplikasi sistem informasi pengarsipan surat sesuai kebutuhan dari BRI Kanwil Malang.



**Gambar 3. 1 Diagram Alir Metode Penelitian**

Aplikasi yang dibuat kemudian diuji untuk membuktikan apakah aplikasi tersebut telah sesuai dengan kebutuhan BRI Kanwil Malang atau tidak, apabila telah sesuai dengan kebutuhan maka aplikasi ini dapat menyelesaikan permasalahan pengarsipan surat yang terjadi di BRI Kanwil Malang karena kebutuhan berawal dari permasalahan yang terjadi sebelumnya. Oleh karena itu metode penelitian yang digunakan ditunjukkan seperti dalam Gambar 3.1.

### 3.1. Pengumpulan Data

Tahap pertama yang dilakukan dalam penelitian adalah pengumpulan data. Pada tahap ini data akan dikumpulkan dari BRI Kanwil Malang dengan cara wawancara. Data yang akan dikumpulkan adalah prosedur pengarsipan surat, pembukuan laporan arsip surat, permasalahan pengarsipan surat yang terjadi, dan spesifikasi kebutuhan aplikasi sistem informasi pengarsipan surat menurut BRI Kanwil Malang.

### 3.2. Analisis Kebutuhan

Analisis kebutuhan dilakukan berdasarkan dari hasil yang diperoleh dari pengumpulan data. Analisis kebutuhan bertujuan untuk mengetahui dan memahami semua kebutuhan yang diperlukan oleh aplikasi yang akan dibangun sehingga dapat mencapai tujuan yang dimaksud. Pada tahap ini maka kebutuhan akan di bedakan menjadi dua, yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional adalah penjelasan mengenai apa saja yang harus dapat dilakukan oleh aplikasi. Kebutuhan fungsional tergantung pada aplikasi yang akan dikembangkan dan sesuai dengan espektasi dari pengguna aplikasi (Sommerville, 2011). Sedangkan kebutuhan non-fungsional menjelaskan bagaimana aplikasi akan melakukan fungsinya. Kebutuhan non-fungsional adalah kebutuhan yang tidak secara langsung berkaitan dengan fungsi spesifik yang terdapat dalam aplikasi. Kebutuhan non-fungsional diperoleh dari kebutuhan pengguna (*product requierment*), aturan organisasi dan faktor eksternal (Sommerville, 2011).

### 3.3.Perancangan

Perancangan dilakukan setelah semua data dan spesifikasi kebutuhan diketahui. Perancangan dilakukan untuk mengetahui apa yang sedang dibuat. Untuk membuat aplikasi berbasis objek cukup menggunakan *use case* diagram, *activity* diagram, *class* diagram (Nugroho, 2009:14). Sedangkan karena aplikasi menyimpan data dalam bentuk digital di dalam *database* maka *entity relationship* diagram penting digunakan untuk menggambarkan *database* yang digunakan oleh aplikasi. Pressman (2001: 280) menyatakan bahwa sebuah *use case* diagram adalah skenario yang menggambarkan bagaimana perangkat lunak yang akan digunakan dalam situasi tertentu. Hasil dari analisis kebutuhan direpresentasikan dengan pembuatan *use case* diagram yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak akan digunakan oleh penggunanya.

Meskipun *use case* merupakan gambaran menyeluruh dari aplikasi namun kurang terperinci, maka dari itu *activity* diagram digunakan untuk mendapatkan gambaran yang lebih menyeluruh (Nugroho, 2009). *Activity* diagram digunakan untuk menggambarkan urutan aktivitas atau prosedur pengarsipan surat baru yang mengadopsi prosedur pengarsipan surat yang lama sehingga prosedur pengarsipan surat dapat disesuaikan dengan keberadaan aplikasi sistem informasi pengarsipan surat dan pada diagram ini juga terdapat *swimlane* (garis tegak lurus dari atas ke bawah) yang menunjukkan *state* berada pada objek atau kelas tertentu. *Swim lane* tersebut akan mempermudah adalah menentukan kelas untuk aplikasi berbasis objek seperti *codeigniter*.

*Class* diagram digunakan untuk mengelompokkan atribut dan fungsi yang terdapat dalam kelas. Untuk menentukan fungsi-fungsi yang terlibat dalam suatu kelas maka dibutuhkan *sequence* diagram (Nugroho, 2009). Sedangkan atribut dan metode atau fungsi pada *class* diagram bisa ditentukan berdasarkan skenario pada *sequence* diagram atau dapat ditambahkan berdasarkan pertimbangan bahwa memang diperlukan dalam aplikasi. Aplikasi yang dibuat menggunakan *database* sebagai penyimpanan data sehingga perlu dipetakan tabel dan karakteristik data yang tersimpan di dalamnya dengan menggunakan *entity relationship* diagram.

Pemetaan *database* yang dilakukan berdasarkan pada tabel-tabel yang digunakan oleh BRI Kanwil Malang untuk melakukan rekapitulasi pengarsipan surat.

Dalam perancangan ini juga akan dijelaskan mengenai desain dari tampilan yang akan digunakan dalam aplikasi. Pembuatan desain ini menggunakan *wireframe* dan desain yang dibuat berdasarkan dari *view* yang diperoleh dari hasil pembuatan *class diagram*. Dalam desain *view* menggunakan *wireframe* akan menjelaskan komponen-komponen dalam tampilan aplikasi.

### 3.4.Implementasi

Implementasi dilakukan berdasarkan pada hasil perancangan sistem sebelumnya. Implementasi pada aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang ini berbasis *Web* dan berorientasi objek dengan *framework codeigniter* yang merupakan model MVC (*Model, View, Controller*) menggunakan PHP sesuai dengan kriteria aplikasi dari BRI Kanwil Malang. Pada tahap ini akan dijelaskan tentang lingkungan dari implementasi dan proses implementasi itu sendiri untuk membuat aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang sehingga bisa menangani masalah pengarsipan surat BRI Kanwil Malang. Pada bagian ini juga akan dijelaskan secara spesifik fungsi-fungsi dari tiap kelas yang sangat penting dalam melakukan fungsi menemukan berkas surat, melakukan penomoran surat secara otomatis, dan menyimpan arsip surat secara digital sesuai kebutuhan dan prosedur BRI Kanwil Malang.

### 3.5.Pengujian

Pengujian dilakukan untuk memastikan apakah tujuan dari pembuatan sistem informasi pengarsipan surat BRI Kanwil Malang tercapai dan memenuhi semua kebutuhan dari BRI Kanwil Malang. Karena aplikasi yang dibuat berorientasi objek maka pengujian yang dilakukan juga merupakan pengujian berorientasi objek. Pendekatan dasar terhadap pengujian berorientasi objek dibagi menjadi pengujian kelas objek dan pengujian integrasi objek (Sommerville, 2001).

Pengujian kelas objek merupakan pengujian kotak putih yang disebut juga pengujian unit namun diperluas untuk mencakup rangkaian operasi yang berhubungan. Pengujian ini berhubungan dengan pemberian jaminan bahwa

*statement* program dieksekusi paling tidak sekali dan bahwa semua jalur dalam program dieksekusi untuk mengetahui kesalahan dalam program (Sommerville, 2007).

Pengujian integrasi objek atau pengujian *release* (Sommerville, 2007) merupakan salah satu tipe pengujian sistem. Pengujian sistem digunakan pada aplikasi yang mengintegrasikan dua atau lebih komponen yang melaksanakan fungsi-fungsi atau fitur sistem. Pengujian sistem dibagi menjadi *integration testing* (pengujian integrasi) dan *release testing* (pengujian integrasi objek).

Untuk aplikasi berorientasi objek sangat sulit untuk menerapkan pengujian integrasi yang berbasis *top-down* atau *bottom-up* karena pada aplikasi berorientasi objek tidak ada hierarki objek yang jelas (Sommerville, 2001). Oleh karena itu *release testing* (pengujian integrasi objek) yang merupakan pengujian kotak hitam lebih tepat digunakan, pengujian ini dilakukan dengan pendekatan berdasarkan operasi atau fitur dalam sistem atau aplikasi.

Kedua pengujian ini yaitu pengujian kelas objek dan pengujian integrasi objek memiliki tujuan yang sama yaitu untuk menjamin bahwa aplikasi yang dibuat telah sesuai dengan kebutuhan BRI Kanwil Malang, prosedur baru yang dibuat dengan adanya aplikasi tidak jauh berbeda dengan prosedur yang lama dan tidak mengubah isi arsip serta aturan penomoran, dengan kata lain isi arsip surat sama saat proses pengarsipan dilakukan dengan prosedur lama karena prosedur baru hanya menyesuaikan pengarsipan dengan adanya aplikasi. Oleh karena itu pengujian yang akan digunakan cukup pengujian integrasi objek (*release testing*) karena lebih mudah untuk membuktikan pencapaian tujuan. Pada akhir pengujian akan dilakukan analisis pembahasan terhadap hasil pengujian yang telah dilakukan. Analisis ini dilakukan untuk mengetahui kelebihan dan kekurangan yang dimiliki oleh aplikasi, dengan mengetahui kelebihan dan kekurangan aplikasi maka akan lebih mudah untuk mengembangkan aplikasi selanjutnya. Hasil dari pembahasan ini akan digunakan untuk pengambilan kesimpulan dan saran.

### **3.6. Kesimpulan dan Saran**

Kesimpulan diberikan setelah semua tahapan selesai dilakukan, mulai dari pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi,

pengujian dan hingga analisis hasil uji. Kesimpulan penelitian akan menjawab permasalahan penelitian. Hasil kesimpulan diberikan berdasarkan keseluruhan proses pembuatan aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang untuk menunjukkan apakah masalah penelitian terjawab dan tujuan penelitian tercapai. Selain kesimpulan, disertakan pula saran untuk kepentingan pengembangan aplikasi.



## BAB IV

### ANALISIS DAN PERANCANGAN

Pada tahap ini akan dibahas perancangan sistem informasi pengarsipan surat untuk Bank Rakyat Indonesia Kantor Wilayah Malang dengan tahapan pengumpulan data terlebih dahulu kemudian dilakukan analisis untuk mendapatkan kebutuhan lalu melakukan perancangan sesuai kebutuhan dan data yang didapat.

#### 4.1. Spesifikasi Kebutuhan

Dalam mendapatkan spesifikasi kebutuhan proses pertama yang dilakukan adalah pengumpulan data kemudian dilakukan analisis kebutuhan.

##### 4.1.1. Pengumpulan Data

Pengumpulan data dilakukan dengan metode wawancara dengan salah satu tim *Information Technology* (IT) BRI Kantor Wilayah Malang, salah satu bidang naungan dari seksi Manajemen Sumber Daya Masyarakat (MSDM). Pada tahap ini diperoleh prosedur pengarsipan surat yang terjadi di Bank Rakyat Indonesia Kantor Wilayah Malang (BRI Kanwil Malang). Mengetahui prosedur pengarsipan surat di BRI Kanwil Malang sangat penting untuk gambaran awal proses bisnis yang terjadi yang nantinya akan di adopsi oleh aplikasi. Prosedur pengarsipan surat yang ada di BRI Kanwil Malang adalah prosedur surat masuk, surat keluar dan prosedur surat balasan.

#### A. Prosedur Pengarsipan Surat

Prosedur pengarsipan surat masuk di BRI Kanwil Malang menjelaskan bahwa jika terdapat surat masuk adalah seperti pada penjelasan berikut :

1. Surat masuk diterima oleh sekretaris.
2. Kemudian surat tersebut diberi nomor register surat setelah sebelumnya sekretaris harus mengecek nomor register terakhir pada pembukuan yang telah digunakan.

3. Informasi umum surat masuk baru tersebut kemudian harus dicatat dalam pembukuan surat masuk. Informasi umum yang dimaksud adalah:
  - a. Diturunkan pada,
  - b. Perihal,
  - c. Jumlah lampiran.
4. Setelah itu surat tersebut diserahkan pada seksi yang dituju.
5. Seksi tersebut melakukan pembukuan surat masuk kembali di masing-masing pembukuan seksi.
6. Seksi yang menerima surat masuk harus memfotokopi surat tersebut untuk arsip.
7. Terakhir seksi tersebut harus menyerahkan hasil fotokopi tersebut pada sekretaris.

Prosedur pengarsipan surat keluar di BRI Kanwil Malang menjelaskan bahwa jika terdapat surat keluar maka seperti pada penjelasan berikut :

1. Seksi bersangkutan yang akan mengeluarkan surat harus melakukan pengecekan nomor register surat dan nomor surat yang bisa digunakan ke sekretaris.
2. Mencatat detail surat keluar yang akan dibuat ke pembukuan arsip surat.
3. Setelah itu seksi dapat membuat surat keluar.
4. Surat keluar yang telah jadi difotokopi.
5. Lalu seksi memberi nomor register pada arsip surat keluar.
6. Hasil fotokopi surat keluar tersebut harus diserahkan pada sekretaris untuk menjadi arsip.
7. Kemudian surat keluar dikirim.

Prosedur pengarsipan surat balasan di BRI Kanwil Malang menjelaskan bahwa jika terdapat surat keluar balasan untuk surat masuk sebelumnya maka seperti pada penjelasan berikut :

1. Seksi bersangkutan yang akan mengeluarkan surat harus melakukan pengecekan nomor register surat masuk yang akan dibalas nomor surat yang bisa digunakan ke sekretaris.
2. Mencatat detail surat keluar balasan yang akan dibuat ke pembukuan arsip surat.
3. Setelah itu seksi dapat membuat surat keluar balasan.
4. Surat keluar yang telah jadi difotokopi.
5. Lalu seksi memberi nomor register pada arsip surat keluar sesuai dengan nomor register surat masuk yang dibalas.
6. Hasil fotokopi surat keluar balasan tersebut harus diserahkan pada sekretaris untuk menjadi arsip.
7. Kemudian surat balasan dikirim.

## **B. Pembukuan Laporan Arsip Surat**

Pada proses pengarsipan surat di BRI Kanwil Malang terdapat tabel pembukuan laporan arsip surat. Tabel ini menyimpan data surat masuk dan surat keluar dan tabel ini dimiliki oleh sekretaris. Berikut adalah nama kolom yang terdapat pada tabel pembukuan laporan arsip surat masuk dan surat keluar.

### **Surat Masuk**

1. REG (nomor register menunjukkan urutan surat, dan surat masuk maupun surat keluar menggunakan satu urutan penomoran nomor register),
2. Tanggal (tanggal surat masuk diterima oleh sekretaris),
3. Pengirim,
4. Tertuju,
5. Perihal,
6. Dari Agenda (menjelaskan bahwa surat berasal dari agenda apa),
7. Dengan Agenda (menjelaskan surat memiliki agenda atau tidak),
8. Lampiran,
9. Catatan.

### Surat Keluar

1. REG (nomor register menunjukkan urutan surat, dan surat masuk maupun surat keluar menggunakan satu urutan penomoran nomor register),
2. Nomor Surat (nomor yang menunjukkan urutan surat keluar),
3. Tanggal (tanggal surat keluar ketika dikeluarkan oleh seksi),
4. Pengirim,
5. Tertuju,
6. Perihal,
7. Dari Agenda (menjelaskan bahwa surat berasal dari agenda apa),
8. Dengan Agenda (menjelaskan surat memiliki agenda atau tidak),
9. Lampiran,
10. Catatan.

Pada pembukuan surat masuk kolom REG, tanggal, pengirim, tertuju, perihal dan kolom pada surat keluar yaitu REG, nomor surat, tanggal, pengirim, tertuju, perihal adalah kolom-kolom yang harus terisi. Kolom-kolom tersebut merupakan kolom-kolom minimal yang harus terisi untuk dapat mengidentifikasi surat masuk maupun surat keluar di Bank Rakyat Indonesia.

### C. Kriteria Aplikasi

Pada proses wawancara disebutkan oleh tim *Information Technology* BRI bahwa pihak BRI Kanwil Malang mengharapkan bahwa aplikasi yang dibuat nantinya berbasis *web* menggunakan PHP dan memenuhi kriteria sebagai berikut:

1. Dapat menangani pengarsipan surat masuk dan surat keluar.
2. Hanya dapat digunakan oleh seorang sekretaris dan lima orang seksi.
3. Sekretaris dapat mengakses semua surat masuk dan surat keluar.
4. Sekretaris bisa memasukkan data surat masuk berupa:
  - a. Pengirim surat,
  - b. Ditujukan pada,
  - c. Perihal.
5. Seksi hanya dapat mengakses surat masuk dan surat keluar milik seksi yang bersangkutan.

6. Tanggal terima surat masuk dan tanggal keluar surat keluar dicatat secara otomatis oleh aplikasi.
7. Nomor register pada surat masuk maupun keluar menggunakan format *yyyymmddxxx* dengan keterangan sebagai berikut:
  - a. *yyyy* adalah tahun,
  - b. *mm* adalah bulan,
  - c. *dd* adalah hari/tanggal,
  - d. *xxx* adalah angka yang bertambah secara otomatis.
8. Nomor surat bisa bertambah secara otomatis karena nantinya akan disesuaikan dengan format nomor surat sesuai aturan administrasi BRI.
9. *Database* menyesuaikan tabel pembukuan laporan arsip surat.
10. Proses bisnis sistem yang baru disesuaikan dengan prosedur milik BRI.
11. Surat di-*scan* lalu disimpan dalam *database*.
12. *Database* terpusat di *server* lokal BRI.

#### 4.1.2. Analisis Kebutuhan

Analisis Kebutuhan diperlukan untuk menyimpulkan hasil data yang berhasil dikumpulkan sehingga dapat dibuat suatu solusi atas permasalahan yang terjadi terkait pengarsipan surat. Kebutuhan yang dimaksud adalah kebutuhan fungsional dan kebutuhan non-fungsional.

##### A. Kebutuhan Fungsional

Analisis Kebutuhan diperlukan untuk menyimpulkan hasil data yang berhasil dikumpulkan sehingga dapat dibuat suatu solusi atas permasalahan yang terjadi terkait pengarsipan surat. Kriteria aplikasi seperti yang disebutkan oleh tim *Information Technology* BRI dan hasil dari pengumpulan data sebelumnya kita simpulkan menjadi kebutuhan fungsional. Berikut ini adalah Tabel 4.1 yang berisi kebutuhan fungsional dan penjabarannya sesuai hasil pengumpulan data. Pada Tabel 4.1 ini menghubungkan antara kebutuhan fungsional dan fitur-fitur pada aplikasi yang digambarkan di *use case* diagram Gambar 4.1 pada tahap perancangan.

**Tabel 4. 1 Spesifikasi Kebutuhan Fungsional Sistem Informasi Pengarsipan Surat BRI Kanwil Malang**

| ID Kebutuhan Fungsional | Deskripsi   | Use Case ID | Use Case                     |
|-------------------------|---|-------------|------------------------------|
| KF-01                   | Sekretaris dapat melihat seluruh surat masuk yang ditujukan pada seluruh seksi  | UC-01       | Melihat surat masuk          |
| KF-02                   | Seksi dapat melihat seluruh surat masuk yang ditujukan padanya  |             |                              |
| KF-03                   | Sekretaris dapat melihat seluruh surat keluar yang dibuat oleh seluruh seksi  | UC-02       | Melihat surat keluar         |
| KF-04                   | Seksi dapat melihat seluruh surat keluar yang dibuat olehnya  |             |                              |
| KF-05                   | Pengguna dapat melakukan pencarian menurut masing-masing kolom pada arsip surat masuk dan surat keluar  | UC-03       | Pencarian                    |
| KF-06                   | Pengguna dapat melakukan pengurutan menurut masing-masing kolom pada arsip surat masuk dan surat keluar   | UC-04       | Pengurutan                   |
| KF-07                   | <p>Sekretaris menerima surat masuk dan memasukkan data umum surat masuk dan lalu tanggal surat masuk di buat secara otomatis ke dalam <i>database</i>. Data umum yang dimaksud adalah:</p> <ol style="list-style-type: none"> <li>Pengirim surat,</li> <li>Ditujukan pada,</li> <li>Perihal.</li> </ol> <p>Nomor register di buat juga secara otomatis ketika data umum surat masuk dimasukkan, nomor register dibuat dengan format <i>yyyymmddxxx</i> dengan keterangan sebagai berikut:</p> <ol style="list-style-type: none"> <li><i>yyyy</i> adalah tahun,</li> <li><i>mm</i> adalah bulan,</li> <li><i>dd</i> adalah hari/tanggal,</li> <li><i>xxx</i> adalah angka yang bertambah secara otomatis.</li> </ol> | UC-05       | Tambah surat masuk           |
|                         |   | UC-06       | Memberi nomor register surat |
| KF-08                   | Seksi memasukkan detail surat masuk yang ditujukan padanya  | UC-07       | Terima surat masuk           |

|       |  |       |                                      |
|-------|--|-------|--------------------------------------|
|       |  | UC-08 | Unggah hasil <i>scan</i> arsip surat |
| KF-09 | Seksi dapat membuat surat keluar, tanggal surat keluar dan nomor surat keluar dibuat secara otomatis oleh aplikasi. Nomor register di buat juga secara otomatis ketika surat keluar dibuat dengan format <i>yyyymmddxxx</i> dengan keterangan sebagai berikut:<br>a. <i>yyyy</i> adalah tahun,<br>b. <i>mm</i> adalah bulan,<br>c. <i>dd</i> adalah hari/tanggal,<br>d. <i>xxx</i> adalah angka yang bertambah secara otomatis.  | UC-09 | Membuat surat keluar                 |
|       |  | UC-10 | Memberi nomor surat                  |
|       |  | UC-06 | Memberi nomor register surat         |
|       |  | UC-08 | Unggah hasil <i>scan</i> arsip surat |
| KF-10 | Seksi dapat membuat surat keluar untuk membalas surat masuk yang tertuju pada seksi bersangkutan, tanggal surat keluar dan nomor surat keluar dibuat secara otomatis oleh aplikasi. Nomor register surat keluar balasan dibuat sama dengan surat masuk yang dibalas dengan format <i>yyyymmddxxx</i> dengan keterangan sebagai berikut:<br>a. <i>yyyy</i> adalah tahun,<br>b. <i>mm</i> adalah bulan,<br>c. <i>dd</i> adalah hari/tanggal,<br>d. <i>xxx</i> adalah angka yang bertambah secara otomatis. | UC-11 | Membuat surat balasan                |
|       |  | UC-06 | Memberi nomor register surat         |
|       |  | UC-08 | Unggah hasil <i>scan</i> arsip surat |
|       |  | UC-10 | Memberi nomor surat                  |

## B. Kebutuhan Non-Fungsional

Hasil dari pengumpulan data selain menghasilkan kebutuhan fungsional juga menghasilkan kebutuhan non-fungsional. Dari hasil pengumpulan data diperoleh kriteria bahwa aplikasi harus membedakan hak akses pengguna. Berikut adalah dua kriteria aplikasi yang disebutkan pada proses pengumpulan data yaitu:

1. Sekretaris dapat mengakses semua surat masuk dan surat keluar.
2. Seksi hanya dapat mengakses surat masuk dan surat keluar milik seksi yang bersangkutan.

Dari kriteria tersebut maka dapat disimpulkan bahwa aplikasi harus dapat mengidentifikasi pengguna yang melakukan akses lalu menentukan hak dan izin akses pengguna tersebut. Oleh karena itu muncul kebutuhan non-fungsional yang disebut kebutuhan keamanan. Identifikasi pengguna dilakukan aplikasi pada proses *log in* lalu aplikasi menentukan hak dan izin akses pengguna terhadap arsip surat dilakukan setiap pengguna melakukan permintaan akses dengan cara pengecekan kondisi.

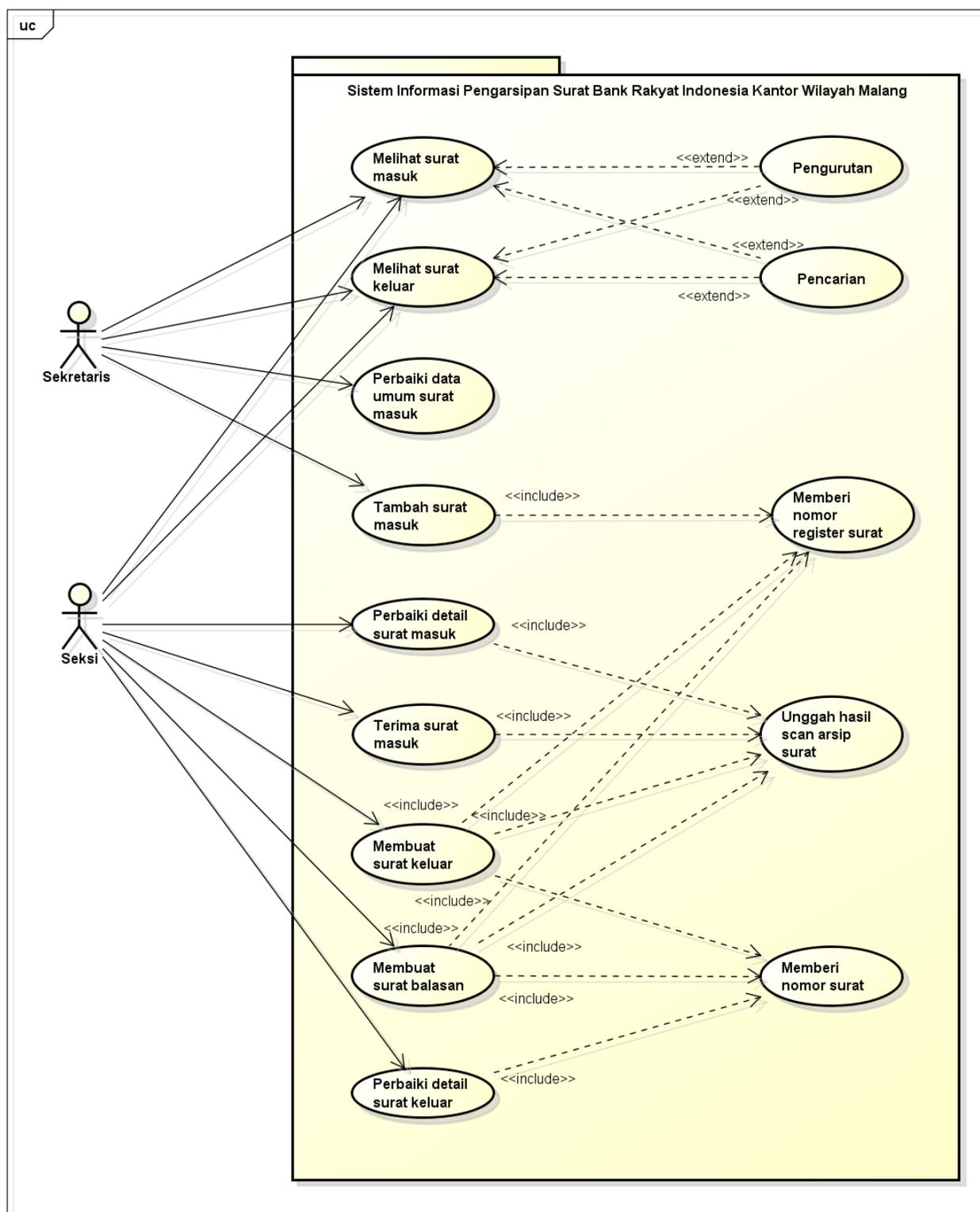
## 4.2. Diagram Perancangan

Pada tahap perancangan ini dibuat *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Entity Relationship Database Diagram* untuk menggambarkan dan merancang aplikasi sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang.

### 4.2.1. Use Case Diagram

Aplikasi sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang memiliki fitur sesuai dengan kebutuhan fungsional pada Tabel 4.1 dan digambarkan menggunakan *use case diagram* seperti dalam Gambar 4.1 di bawah ini untuk menjelaskan fitur-fitur pengarsipan surat dalam aplikasi.

Dari Gambar 4.1 dapat dilihat bahwa aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang ini dibuat untuk menangani pengarsipan surat masuk dan surat keluar. Nomor surat untuk surat keluar dan surat keluar balasan dibuat secara otomatis dalam fungsi yang sama sehingga akan berurutan. Nomor register dibuat secara otomatis untuk menomori surat masuk dan surat keluar dalam fungsi yang sama sehingga akan berurutan. Fitur-fitur dalam aplikasi yang dibuat lebih detail akan dijelaskan pada Tabel 4.2.



Gambar 4. 1 Use Case Diagram Sistem Informasi Pengarsipan Surat BRI Kanwil Malang

Tabel 4. 2 Fitur-Fitur Aplikasi Pada *Use Case Diagram*

| ID    | <i>Use Case</i>                      | Penjelasan Fitur   |
|-------|--------------------------------------|--|
| UC-01 | Melihat surat masuk                  | Pengguna dapat melihat surat masuk sesuai hak akses pengguna masing-masing.  |
| UC-02 | Melihat surat keluar                 | Pengguna dapat melihat surat keluar sesuai hak akses pengguna masing-masing.   |
| UC-03 | Pencarian                            | Pada tampilan surat masuk dan surat keluar pengguna dapat melakukan pencarian untuk mempermudah menemukan arsip surat.   |
| UC-04 | Pengurutan                           | Pada tampilan surat masuk dan surat keluar pengguna dapat melakukan pengurutan sesuai kolom tertentu untuk mempermudah menemukan arsip surat.                                |
| UC-05 | Tambah surat masuk                   | Sekretaris dapat menambah dan memasukkan data umum surat masuk ke dalam aplikasi.  |
| UC-06 | Memberi nomor register surat         | Aplikasi secara otomatis membuat nomor register ketika aplikasi menyimpan surat masuk, surat keluar atau surat balasan.  |
| UC-07 | Terima surat masuk                   | Seksi dapat menerima surat masuk dari sekretaris dan memasukkan detail surat ke dalam aplikasi.  |
| UC-08 | Unggah hasil <i>scan</i> arsip surat | Aplikasi akan menyimpan hasil <i>scan</i> surat sebagai arsip ketika seksi mengisi detail surat saat menerima surat masuk, membuat surat keluar, atau membuat surat balasan. |
| UC-09 | Membuat surat keluar                 | Seksi dapat membuat surat keluar dan memasukkan detail surat keluar.   |
| UC-10 | Memberi nomor surat                  | Aplikasi secara otomatis membuat nomor surat ketika terdapat permintaan dari seksi untuk membuat surat keluar atau surat balasan.  |
| UC-11 | Membuat surat balasan                | Seksi dapat membuat surat balasan dan memasukkan detail surat balasan.   |

|       |                                |   |
|-------|--------------------------------|---|
| UC-12 | Perbaiki data umum surat masuk | Sekretaris dapat memperbaiki data umum surat masuk. |
| UC-13 | Perbaiki detail surat masuk    | Seksi dapat memperbaiki detail surat masuk.         |
| UC-14 | Perbaiki detail surat keluar   | Seksi dapat memperbaiki detail surat keluar.        |

#### 4.2.2. Activity Diagram

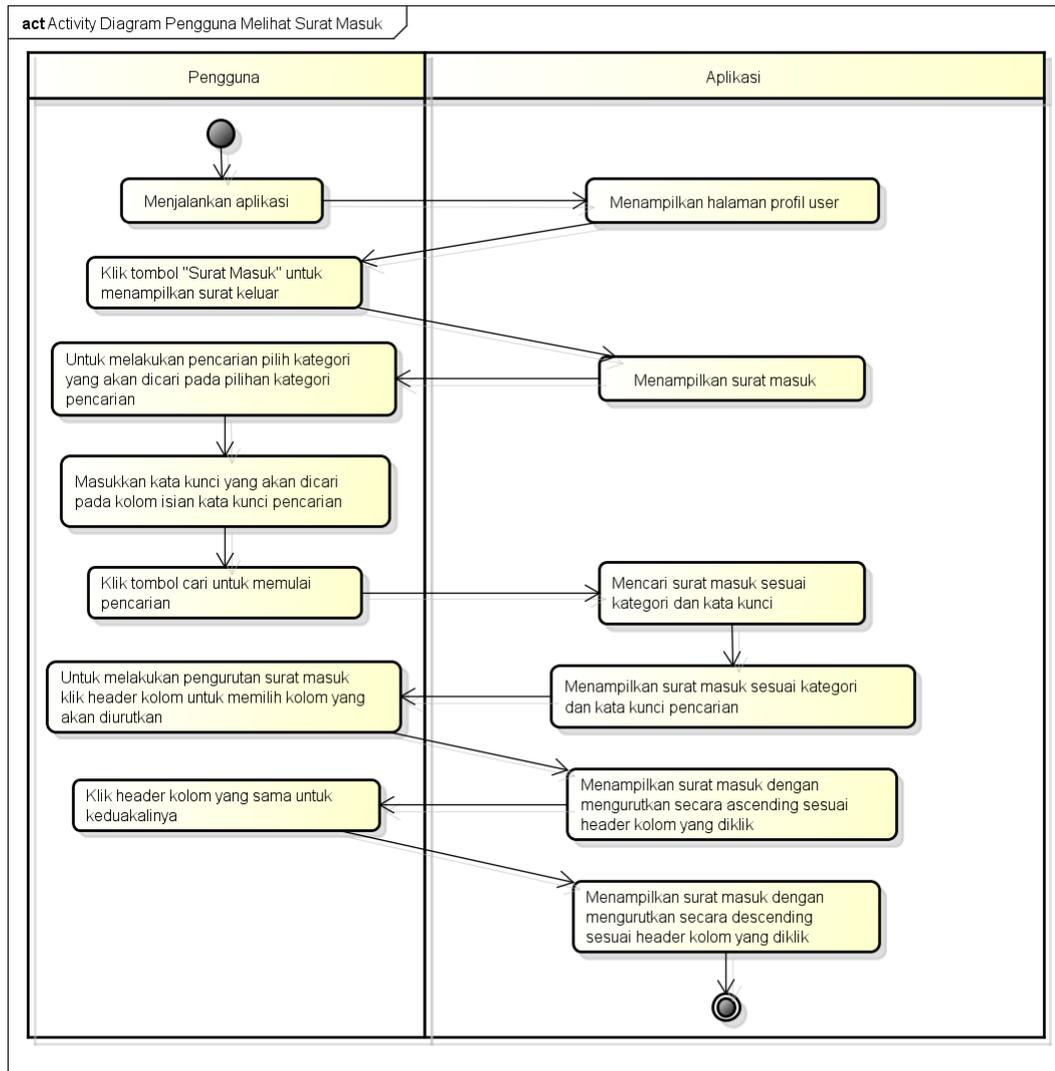
Dalam melakukan pengarsipan surat menggunakan aplikasi sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang adapun urutan aktivitas dalam proses bisnis merupakan penjabaran dari fitur-fitur utama aplikasi yang yang tercantum pada *use case* diagram dan merupakan penyesuaian dengan prosedur pengarsipan surat yang sebelumnya telah ada.

*Activity* diagram digunakan untuk menjabarkan lebih detail fitur-fitur dan interaksi pengguna terhadap aplikasi dan menggambarkan urutan aktivitas atau prosedur pengarsipan surat baru yang mengadopsi prosedur pengarsipan surat yang lama sehingga prosedur pengarsipan surat dapat disesuaikan dengan keberadaan aplikasi sistem informasi pengarsipan surat Terdapat 6 proses bisnis dalam pengarsipan surat menggunakan sistem informasi pengarsipan surat BRI Kanwil Malang, yaitu pengguna melihat surat masuk, pengguna melihat surat keluar, sekretaris menambahkan surat masuk, seksi menerima surat masuk, seksi membalas surat masuk (membuat surat balasan), dan seksi membuat surat keluar. Sedangkan *activity* diagram proses masuk aplikasi (*log in*) yang disebutkan pada kebutuhan non-fungsional dapat dilihat pada dalam Lampiran 1 Gambar L.1.1 *activity* diagram *log in*. Jadi diasumsikan pada setiap *activity* diagram pada tahapan ini pengguna telah melakukan *log in* terhadap aplikasi.

##### A. Proses Pengguna Melihat Surat

Dalam Gambar 4.2 menunjukkan proses bagaimana pengguna dapat melihat surat masuk. Proses dalam menampilkan surat masuk atau surat keluar hampir mirip, *activity* diagram melihat surat keluar dapat dilihat dalam Lampiran 1 Gambar L.1.2 *activity* diagram pengguna melihat surat keluar. Setelah pengguna masuk ke dalam aplikasi maka pengguna akan diteruskan pada halaman profil. Seperti dalam

Gambar 4.2 terlihat untuk menuju halaman surat masuk pengguna harus mengklik tombol ‘Surat Masuk’ sedangkan untuk masuk ke halaman surat keluar pengguna harus mengklik tombol ‘Surat Keluar’. Dalam melakukan akses terhadap aplikasi pengguna memiliki hak akses yang berbeda-beda. Sekretaris dapat melihat seluruh surat masuk dan surat keluar, sedangkan seksi hanya dapat menampilkan surat masuk dan surat keluar milik seksi yang bersangkutan.



powered by Astah

**Gambar 4. 2 Activity Diagram Pengguna Melihat Surat Masuk**

Pada tampilan surat masuk maupun surat keluar pengguna dapat menggunakan fitur pencarian atau pengurutan untuk mempermudah pengguna dalam menemukan atau menampilkan surat yang diinginkan. Pencarian dapat dilakukan dengan cara pengguna memilih kategori dan memasukkan kata kunci

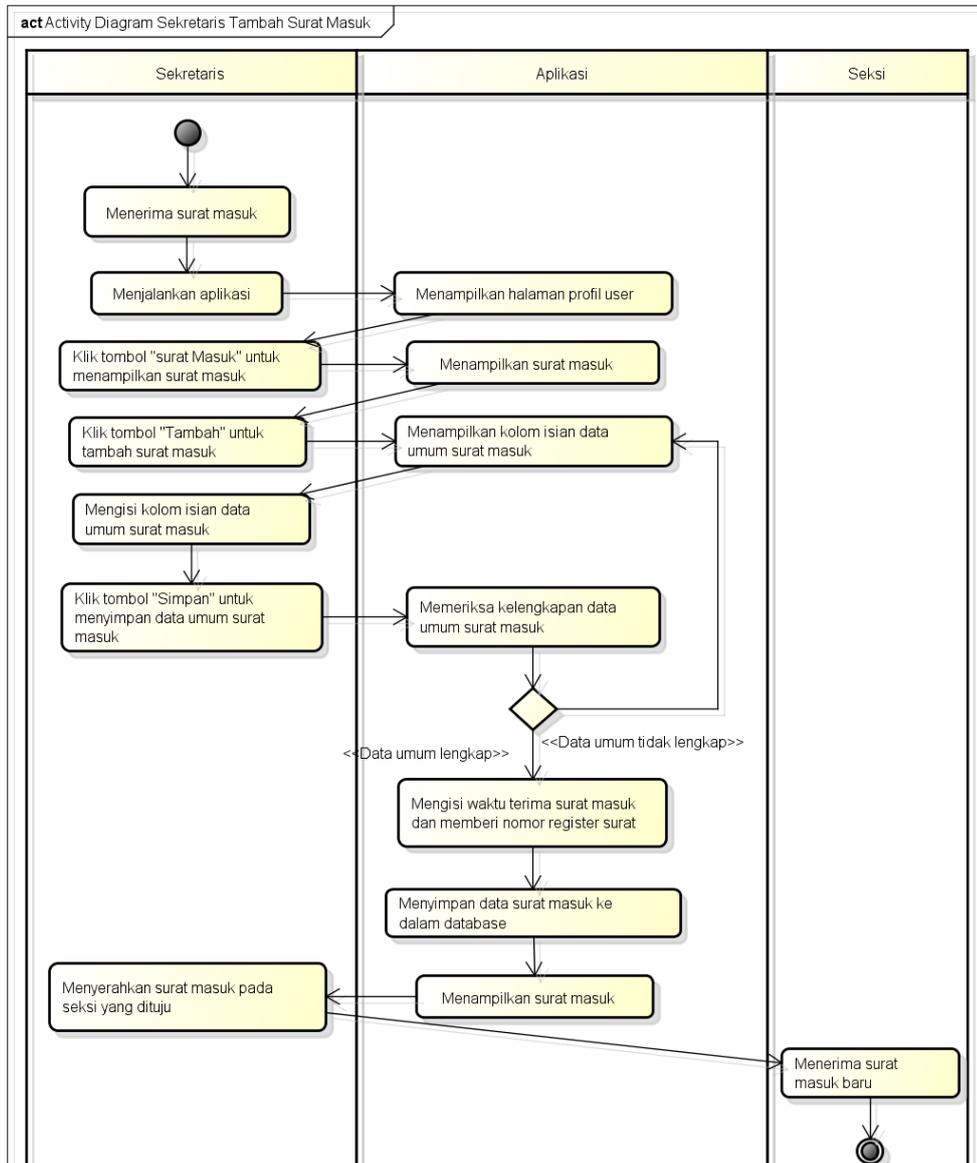
arsip surat yang akan dicari, contohnya surat akan dicari adalah surat masuk berdasarkan kategori 'Pengirim' dengan kata kunci 'Pemerintah', maka aplikasi akan menampilkan surat masuk dengan kategori pengirim yang memiliki kata pemerintah. Sedangkan proses pengurutan dapat dilakukan cukup melakukan klik pada *header* kolom maka aplikasi akan memberikan tampilan berurutan.

## B. Proses Ketika Terdapat Surat Masuk

Proses ketika terdapat surat masuk pertama kali surat diterima oleh sekretaris lalu diteruskan pada seksi. Proses bisnis yang terjadi adalah sebagai berikut:

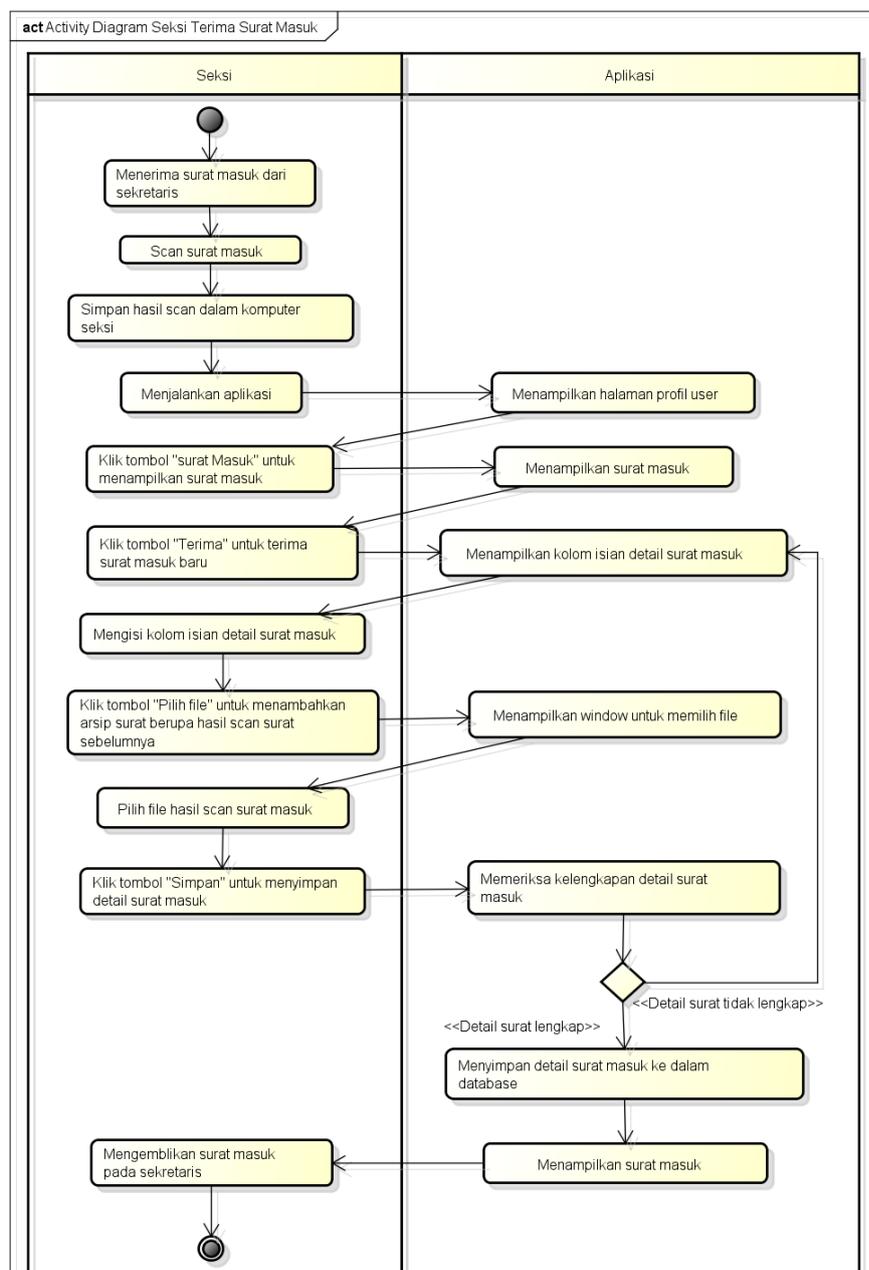
1. Surat masuk diterima oleh sekretaris,
2. Sekretaris melakukan klik tombol 'Tambah' untuk menambah surat masuk pada tampilan surat masuk,
3. Sekretaris memasukkan data umum surat masuk ke dalam aplikasi. Data umum yang dimaksud berupa:
  - a. Pengirim surat,
  - b. Ditujukan pada,
  - c. Perihal,
4. Tanggal surat diterima dan nomor register surat akan secara otomatis dibuat oleh aplikasi,
5. Setelah itu sekretaris memberikan surat tersebut pada seksi yang dituju,
6. Seksi yang menerima surat masuk dan melakukan *scan* pada surat masuk lalu menyimpan hasil *scan*,
7. Kemudian seksi memeriksa surat masuk yang tertuju padanya, lalu seksi mengklik tombol 'Terima' pada surat masuk baru.
8. Seksi menyimpan detail dan hasil *scan* surat masuk ke dalam aplikasi. Detail surat tersebut adalah:
  - a. Dari agenda,
  - b. Dengan agenda,
  - c. Lampiran,
  - d. Catatan,

- e. Arsip hasil *scan* surat,
- 9. Kemudian seksi mengembalikan arsip surat masuk asli pada sekretaris.



**Gambar 4. 3 Activity Diagram Sekretaris Tambah Surat Masuk**

Lebih detail mengenai alur proses ketika terdapat surat masuk bisa dilihat dalam Gambar 4.3 yaitu proses yang dilakukan sekretaris ketika terdapat surat masuk dan Gambar 4.4 proses yang dilakukan oleh seksi ketika menerima surat masuk dari sekretaris.



**Gambar 4. 4 Activity Diagram Seksi Terima Surat Masuk**

Proses yang dijelaskan dalam Gambar 4.3 dan Gambar 4.4 yang dilakukan oleh aktor berbeda merupakan satu kesatuan proses ketika terdapat surat masuk baru. Sekretaris berperan untuk menambahkan surat masuk baru, lalu seksi menerima surat itu untuk kemudian ditindak lanjuti. Apabila terjadi kesalahan dalam memasukkan data umum surat masuk maka sekretaris dapat menjalankan fitur perbaiki data umum surat masuk dengan proses hampir sama seperti dalam Gambar 4.3 hanya saja sekretaris bukan mengklik tombol “Tambah” tapi mengklik



tombol “Perbaiki”. Begitu juga seksi dapat melakukan perbaikan pada detail surat masuk yang sudah dimasukkan dengan mengklik tombol “Perbaiki”

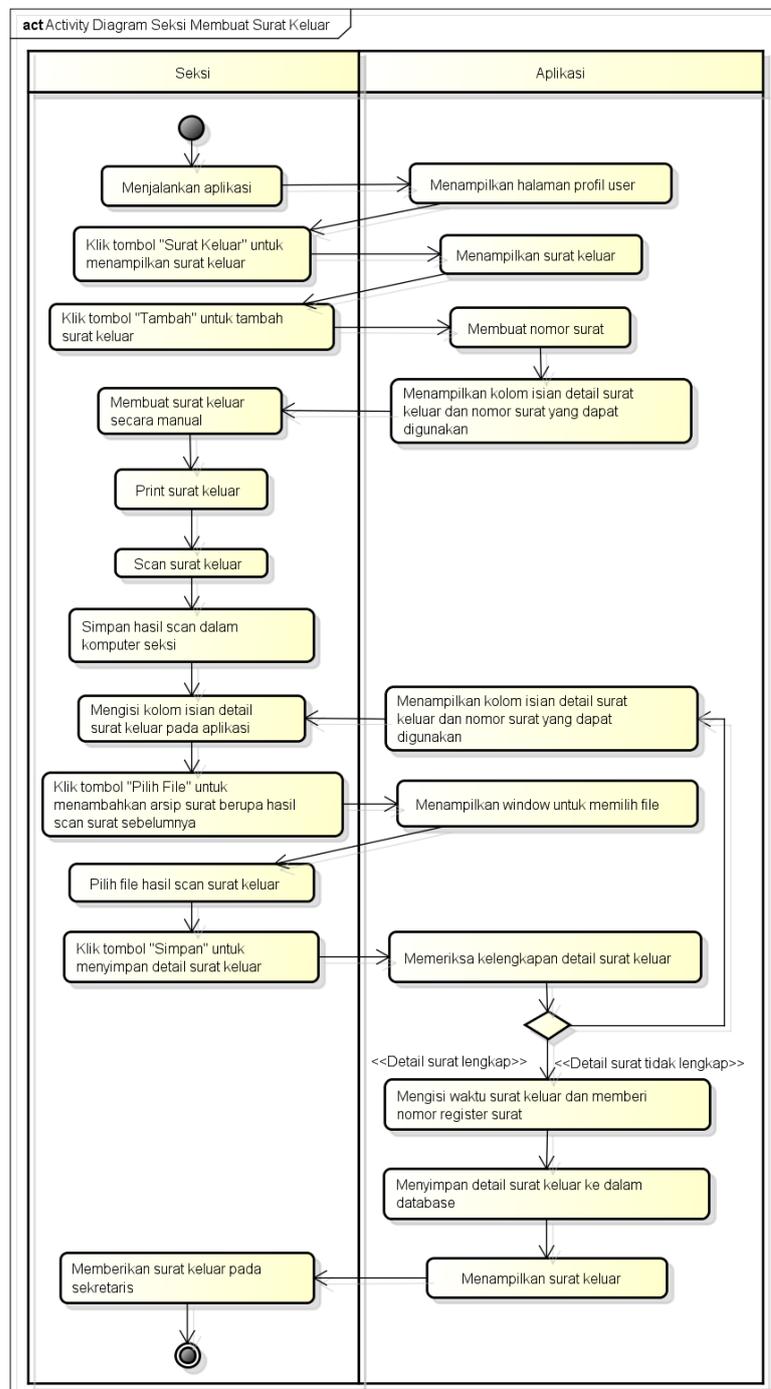
Pada prosedur baru untuk ketika terdapat surat masuk aktivitas sekretaris mengecek nomor register terakhir, sekretaris memberi nomor register pada surat, dan sekretaris menyimpan fotokopi surat masuk untuk arsip tidak dilakukan lagi. Kemudian terdapat beberapa aktivitas yang disesuaikan dengan keberadaan aplikasi. Lebih jelas mengenai perbandingan prosedur ketika terdapat surat masuk antara yang lama dan yang baru maka dapat dilihat dalam Lampiran 2 Gambar L.2.1.

### C. Proses Seksi Membuat Surat Keluar

Pembuatan surat keluar hanya dapat dilakukan oleh seksi. Surat keluar dibuat dengan alur sebagai berikut:

1. Seksi bersangkutan yang akan membuat surat keluar melakukan klik tombol ‘Tambah’ untuk membuat surat keluar pada tampilah surat keluar,
2. Pada aplikasi akan muncul nomor surat yang dapat digunakan oleh seksi untuk surat keluar,
3. Seksi membuat surat keluar secara manual menggunakan nomor surat yang muncul,
4. Setelah surat dibuat lalu *scan* surat keluar yang telah dibuat dan simpan hasil *scan* ke dalam komputer,
5. Kemudian seksi memasukkan detail surat keluar ke dalam aplikasi berupa:
  - a. Tertuju,
  - b. Perihal,
  - c. Dari Agenda,
  - d. Dengan Agenda,
  - e. Jumlah lampiran,
  - f. Catatan,
  - g. Arsip hasil *scan* surat.
6. Simpan detail surat dalam aplikasi,

7. Tanggal surat keluar dan nomor register surat akan dibuat secara otomatis oleh aplikasi, Surat keluar diserahkan pada sekretaris dan siap dikirim.



Gambar 4.5 Activity Diagram Seksi Membuat Surat Keluar

Dalam Gambar 4.5 di bawah ini akan menjelaskan lebih detail bagaimana proses membuat surat keluar yang dilakukan oleh seksi. Apabila terjadi kesalahan

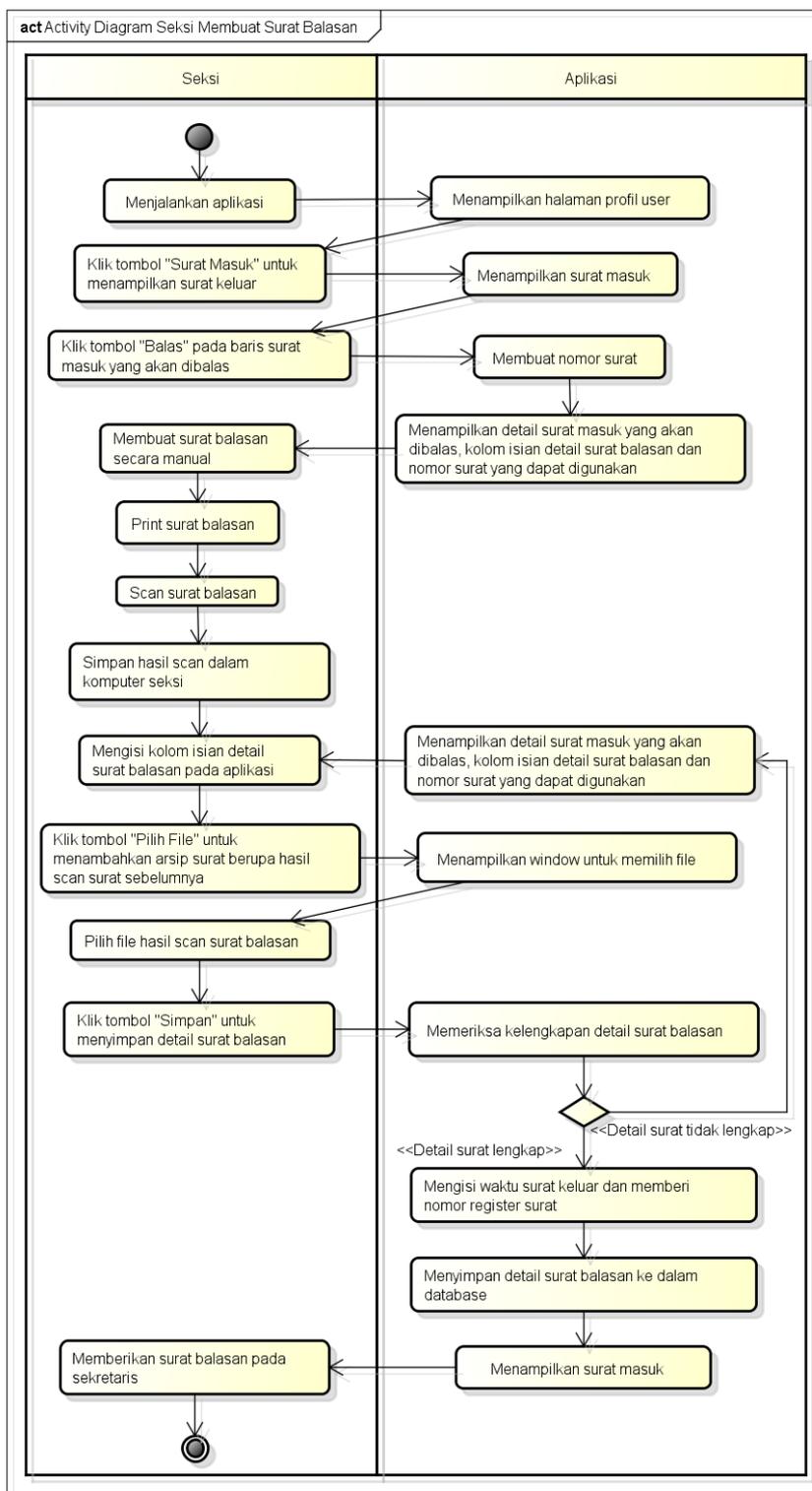
dalam memasukkan detail surat keluar maka seksi dapat menjalankan fitur perbaikan detail surat keluar dengan dengan proses hampir sama seperti dalam Gambar 4.5 hanya saja sekretaris bukan mengklik tombol “Tambah” tapi mengklik tombol “Perbaiki”.

Pada prosedur baru untuk seksi membuat surat keluar aktivitas seksi mengecek nomor register serta nomor surat yang bisa digunakan ke sekretaris, dan seksi memberi nomor register pada arsip surat tidak dilakukan lagi. Kemudian terdapat beberapa aktivitas yang disesuaikan dengan keberadaan aplikasi. Lebih jelas mengenai perbandingan prosedur seksi membuat surat keluar antara yang lama dan yang baru maka dapat dilihat dalam Lampiran 2 Gambar L.2.2.

#### **D. Proses Seksi Membuat Surat Balasan**

Seksi dapat membalas surat masuk yang tertuju padanya. Berikut adalah proses yang dilakukan untuk membuat surat balasan:

1. Seksi bersangkutan yang akan membalas surat masuk melakukan klik tombol ‘Balas’ untuk membalas surat masuk pada tampilan surat masuk,
2. Pada aplikasi akan menampilkan detail surat masuk yang akan dibalas dan nomor surat yang dapat digunakan oleh seksi untuk surat balasan,
3. Seksi membuat surat balasan menggunakan nomor surat yang muncul
4. *Scan* surat balasan yang telah dibuat dan simpan hasil *scan* dalam komputer,
5. Kemudian seksi menyimpan detail surat balasan ke dalam aplikasi berupa:
  - a. Tertuju,
  - b. Perihal,
  - c. Dari Agenda,
  - d. Dengan Agenda,
  - e. Jumlah lampiran,
  - f. Catatan,
  - g. Arsip hasil *scan* surat.
6. Tanggal surat keluar dan nomor register surat balasan akan dibuat sama dengan nomor register surat masuk yang dibalas secara otomatis oleh aplikasi.
7. Surat balasan diserahkan pada sekretaris dan siap dikirim.



Gambar 4. 6 Activity Diagram Seksi Membuat Surat Balasan

Untuk lebih detail alur proses membalas surat masuk yang dilakukan oleh seksi dapat dilihat dalam Gambar 4.6 di atas. Pada prosedur baru untuk seksi membuat surat balasan aktivitas seksi mengecek nomor register surat masuk yang

akan dibalas serta nomor surat yang bisa digunakan ke sekretaris, dan seksi memberi nomor register pada arsip surat tidak dilakukan lagi. Kemudian terdapat beberapa aktivitas yang disesuaikan dengan keberadaan aplikasi. Lebih jelas mengenai perbandingan prosedur seksi membuat surat balasan antara yang lama dan yang baru maka dapat dilihat dalam Lampiran 2 Gambar L.2.3.

#### 4.2.3. Sequence Diagram

*Sequence* diagram bermanfaat untuk menentukan metode-metode (fungsi) yang terlibat dalam suatu kelas (Nugroho, 2009:16). Para ahli seperti Russel J. Abbot dan Grady Booch menggunakan pernyataan-pernyataan dari spesifikasi kebutuhan atau *activity* diagram yang dikembangkan dari suatu *use case* untuk mendapatkan pernyataan informal mengenai aplikasi yang dikembangkan. Jadi dapat disimpulkan bahwa pernyataan kebutuhan dan *activity* diagram yang merupakan deskripsi terperinci dari suatu *use case* diagram, kalimat-kalimat yang dijumpai merupakan kandidat kelas dan dapat diterjemahkan dalam bentuk *sequence* diagram untuk mempermudah dalam pembuatan kelas diagram. Karena aplikasi ini menggunakan codeigniter yang berupa *framework* MVC, maka kelas yang digunakan dibagi menjadi tiga tipe, yaitu *controller*, *model*, dan *view*.

Untuk *sequence* diagram *log in* sendiri untuk melakukan identifikasi pengguna yang masuk dalam aplikasi dapat dilihat dalam Lampiran 3 Gambar L.3.1 *sequence* diagram *log in*. Jika sekretaris masuk dalam aplikasi maka aplikasi akan menampilkan profil seluruh pengguna sedangkan apabila yang masuk adalah salah satu seksi maka aplikasi akan menampilkan profil pengguna tersebut. Untuk perbedaan akses pengguna pada arsip surat akan lebih detail dijelaskan pada diagram-diagram berikut ini.

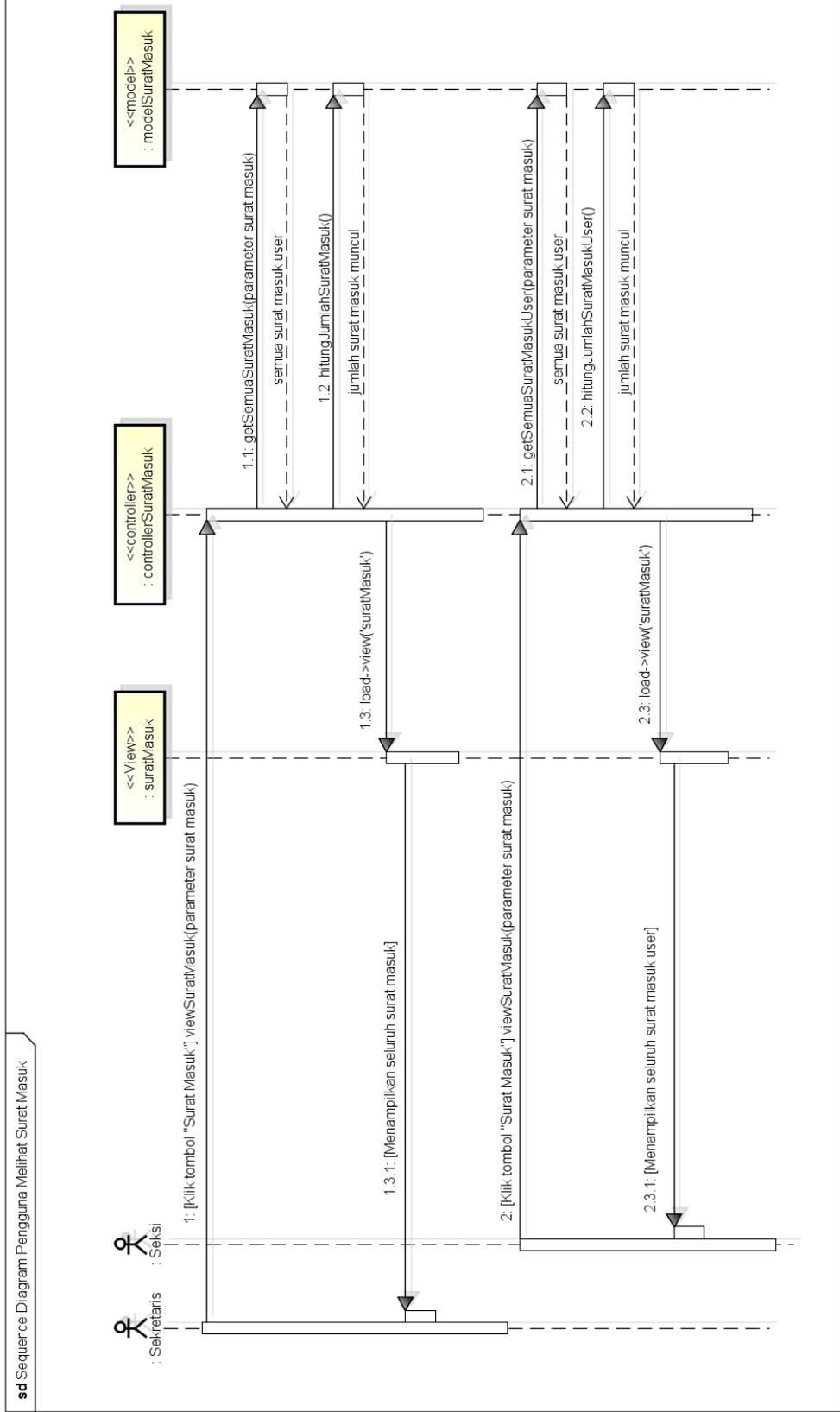
##### A. Pengguna Melihat Surat

Seperti yang telah dijelaskan pada spesifikasi kebutuhan fungsional, sekretaris dan seksi memiliki hak akses arsip surat berbeda. Sekretaris bisa menampilkan seluruh surat masuk sedangkan sedangkan seksi hanya bisa menampilkan surat masuk yang berhubungan dengan seksi tersebut. Berikut *sequence* diagram menampilkan surat masuk dalam Gambar 4.7 yang dilakukan

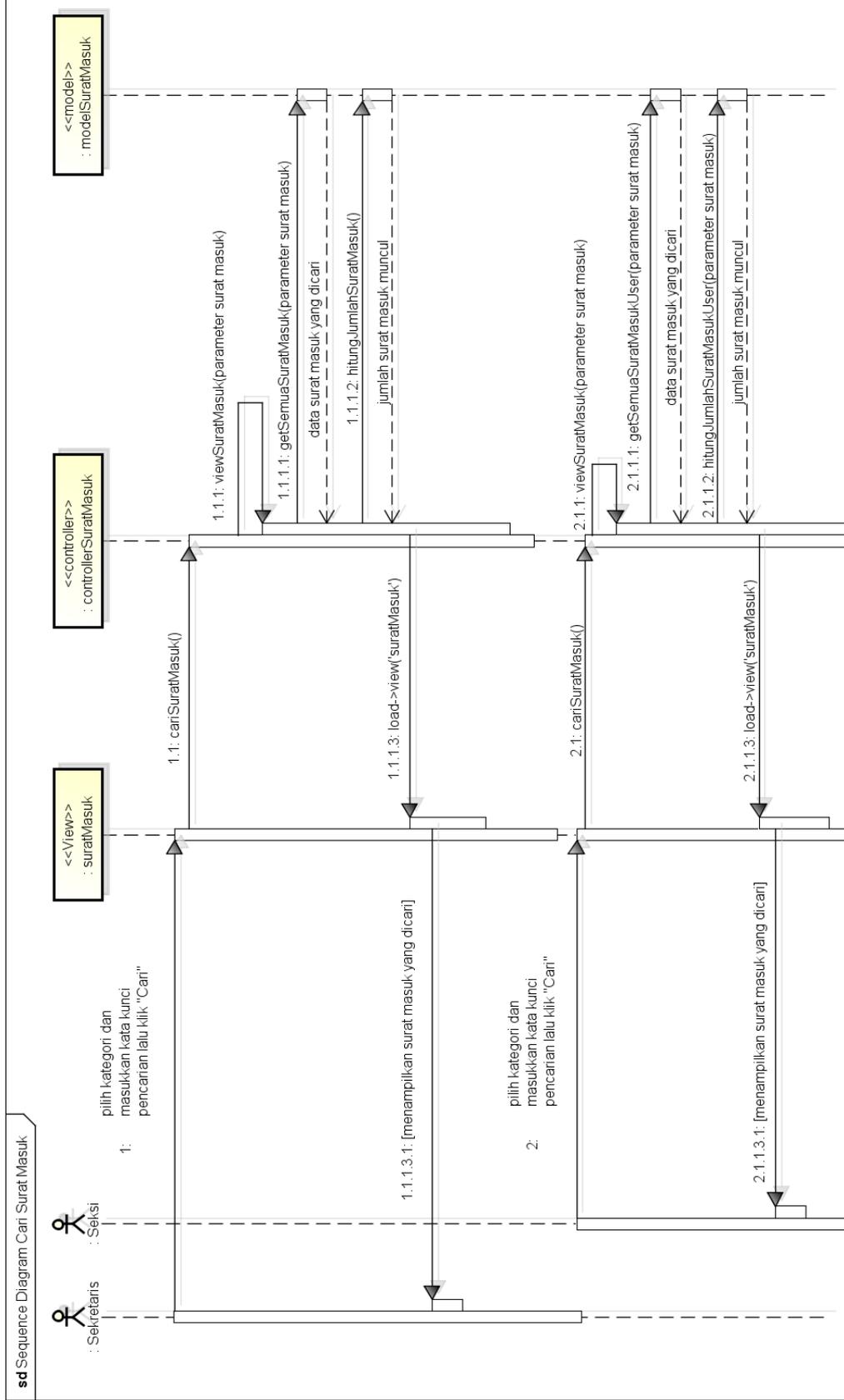
oleh sekretaris atau seksi dan memiliki proses yang sama dengan *sequence* diagram pengguna melihat surat keluar dalam Lampiran 3 Gambar L.3.2. Kedua proses tersebut hanya dibedakan oleh beberapa fungsi. *Sequence* diagram ini dibuat berdasarkan alur pada *activity* diagram pengguna melihat surat masuk dan pengguna melihat surat keluar.

Hak akses untuk menampilkan surat masuk tersebut dibedakan pada fungsi yang digunakan dari kelas *controllerSuratMasuk* dan kelas *modelSuratMasuk* untuk menampilkan surat. Dalam Gambar 4.7 operasi 1.1 *getSemuaSuratMasuk*(parameter surat masuk) adalah fungsi kelas *modelSuratMasuk* yang digunakan untuk menampilkan semua surat masuk dalam *database*. Sedangkan pada operasi 1.4 *getSemuaSuratMasukUser*(parameter surat masuk) adalah fungsi kelas *modelSuratMasuk* yang digunakan untuk menampilkan semua surat masuk milik seksi yang melakukan akses. Fungsi yang dijalankan pada aplikasi untuk melihat surat keluar adalah *getSemuaSuratKeluar*(parameter surat keluar) apabila yang melakukan akses adalah sekretaris dan apabila pengguna adalah salah satu dari seksi maka fungsi yang dijalankan adalah *getSemuaSuratKeluarUser*(parameter surat keluar).

Pada tampilan surat masuk atau surat keluar pengguna dapat melakukan pencarian arsip surat. Gambar 4.8 menunjukkan proses pencarian surat masuk dalam aplikasi sedangkan proses pencarian surat keluar dapat dilihat dalam Lampiran 3 Gambar L.3.3. Karena hak akses yang berbeda maka dalam fitur ini aplikasi hanya memberikan hasil pencarian surat masuk sesuai hak akses surat masuk masing-masing. Fitur untuk pengurutan surat masuk menggunakan metode atau fungsi yang sama untuk menampilkan surat yaitu *viewSuratMasuk*(parameter surat masuk) pada *controllerSuratMasuk* lalu pada *modelSuratMasuk* yaitu fungsi *getSemuaSuratMasuk*(parameter surat masuk) untuk sekretaris dan *getSemuaSuratMasukUser*(parameter surat masuk) untuk seksi. Pengurutan pada aplikasi hanya akan mengubah parameter surat masuk pada fungsi-fungsi tersebut. Kelas dan metode atau fungsi serta parameter yang diperoleh dari *sequence* diagram pengguna melihat surat dan cari surat untuk surat masuk maupun surat keluar di atas akan disebutkan pada Tabel 4.3 di bawah ini.



Gambar 4. 7 Sequence Diagram Pengguna Melihat Surat Masuk



Gambar 4. 8 Sequence Diagram Cari Surat Masuk

**Tabel 4. 3 Kelas dan Metode Pada Proses Pengguna Melihat Surat Masuk**

| Tipe           | Kelas                | Metode / Fungsi  |
|----------------|----------------------|--|
| <<controller>> | controllerSuratMasuk | + cariSuratMasuk() : void  |
|                |                      | + viewSuratMasuk(\$sort_by : String, \$sort_order : String, \$offset : int, \$cari_array : ArrayList) : void   |
|                |                      | + cariSuratKeluar() : void   |
|                |                      | + viewSuratKeluar(\$sort_by : String, \$sort_order : String, \$offset : int, \$cari_array : ArrayList) : void  |
| <<model>>      | modelSuratMasuk      | + getAllSuratMasuk(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$cari_array : ArrayList) : ArrayList                       |
|                |                      | + getAllSuratMasukUser(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$user : String, \$cari_array : ArrayList) : ArrayList  |
|                |                      | + getAllSuratKeluar(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$cari_array : ArrayList) : ArrayList                      |
|                |                      | + getAllSuratKeluarUser(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$user : String, \$cari_array : ArrayList) : ArrayList |
| <<view>>       | suratMasuk           |  |
|                | suratKeluar          |  |

## B. Ketika Terdapat Surat Masuk

Ketika terdapat surat masuk sesuai pada *activity* diagram surat diterima oleh sekretaris terlebih dahulu lalu diserahkan pada seksi, jadi keseluruhan proses ketika terdapat surat masuk dibedakan proses yang dilakukan oleh sekretaris terlebih dahulu untuk menambahkan data surat masuk seperti dalam Gambar 4.9 lalu proses yang dilakukan seksi untuk menerima surat masuk dan menambahkan detail surat masuk seperti dalam Gambar 4.10.

**Tabel 4. 4 Kelas dan Metode Ketika Terdapat Surat Masuk**

| Tipe           | Kelas                | Metode / Fungsi             |
|----------------|----------------------|-----------------------------|
| <<controller>> | controllerSuratMasuk | + tambahSuratMasuk() : void |
|                |                      | + terimaSuratMasuk() : void |

|           |                  |  |
|-----------|------------------|--|
|           |                  | + viewSuratMasuk(\$sort_by : String, \$sort_order : String, \$offset : int, \$cari_array : ArrayList) : void |
| <<model>> | modelSurat       | + reg() : String   |
|           | modelSuratMasuk  | + suratMasuk(\$dataSurat : ArrayList) : void   |
|           |                  | + perbaikiSuratMasuk(\$data : ArrayList, \$reg : String) : void  |
| <<view>>  | suratMasuk       |  |
|           | detailSurat      |  |
|           | terimaSuratMasuk |  |

Dalam Gambar 4.9 sekretaris akan menjalankan operasi 1.1 tambahSuratMasuk() pada kelas *controllerSuratMasuk* lalu menyimpan data surat masuk ke dalam *database* menggunakan operasi 1.1.1.1.1.3 suratMasuk() pada kelas *modelSuratMasuk*. Sedangkan dalam Gambar 4.10 seksi menjalankan operasi 1.1 terimaSuratMasuk() pada kelas *controllerSuratMasuk* lalu menyimpan detail surat masuk ke dalam *database* menggunakan operasi 1.1.1.1.1.2 perbaikiSuratMasuk() pada kelas *modelSuratMasuk*. Pada dasarnya operasi perbaikiSuratMasuk() digunakan untuk mengedit data surat masuk pada *database* yang telah dimasukkan oleh sekretaris. Kelas dan metode atau fungsi serta parameter yang diperoleh dari *sequence* diagram ini disebutkan pada Tabel 4.4 di bawah ini.

### C. Seksi Membuat Surat Keluar

Pengguna yang bisa membuat surat keluar hanyalah seksi. Dalam Gambar 4.11 di atas terlihat seksi membuat surat keluar menggunakan operasi 1.1 tambahSuratKeluar() pada kelas *controllerSuratKeluar* dan menggunakan operasi 1.1.2.1.1.1.2 suratKeluar() untuk menyimpan detail surat keluar ke *database* surat keluar. Kelas dan metode atau fungsi serta parameter yang diperoleh dari *sequence* diagram proses ini disebutkan pada Tabel 4.5 di bawah ini.

**Tabel 4. 5 Kelas dan Metode Ketika Seksi Membuat Surat Keluar**

| Tipe           | Kelas                 | Metode / Fungsi   |
|----------------|-----------------------|---|
| <<controller>> | controllerSuratKeluar | + tambahSuratKeluar() : void  |
|                |                       | + viewSuratKeluar(\$sort_by : String, \$sort_order : String, \$offset : int, \$cari_array : ArrayList) : void |
| <<model>>      | modelSurat            | + reg() : String  |

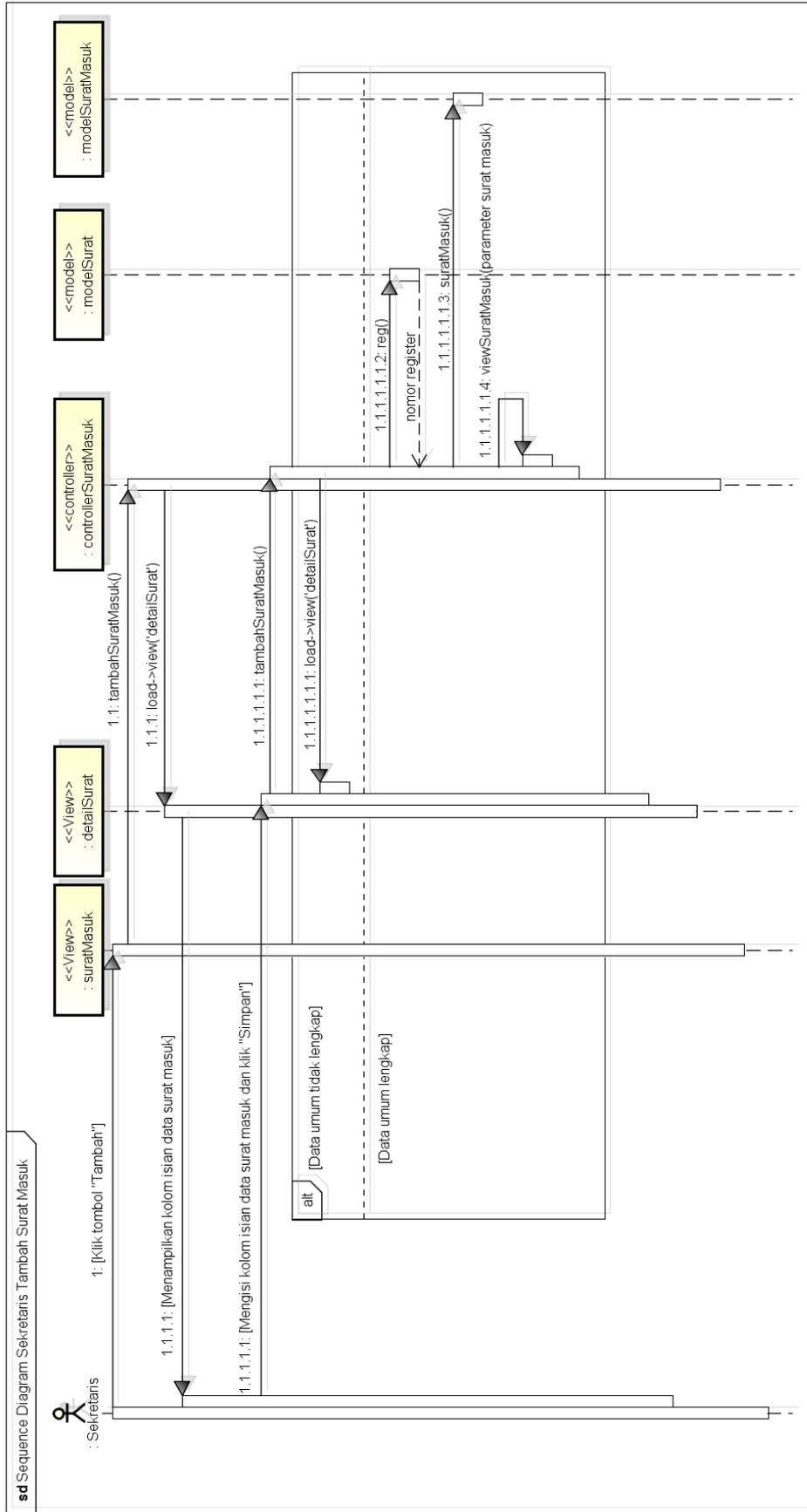
|          |                   |   |
|----------|-------------------|---|
|          | modelSuratKeluar  | + no_surat() : String                         |
|          |                   | + suratKeluar(\$dataSurat : ArrayList) : void |
| <<view>> | suratKeluar       |   |
|          | detailSuratKeluar |   |

#### D. Seksi Membuat Surat Balasan

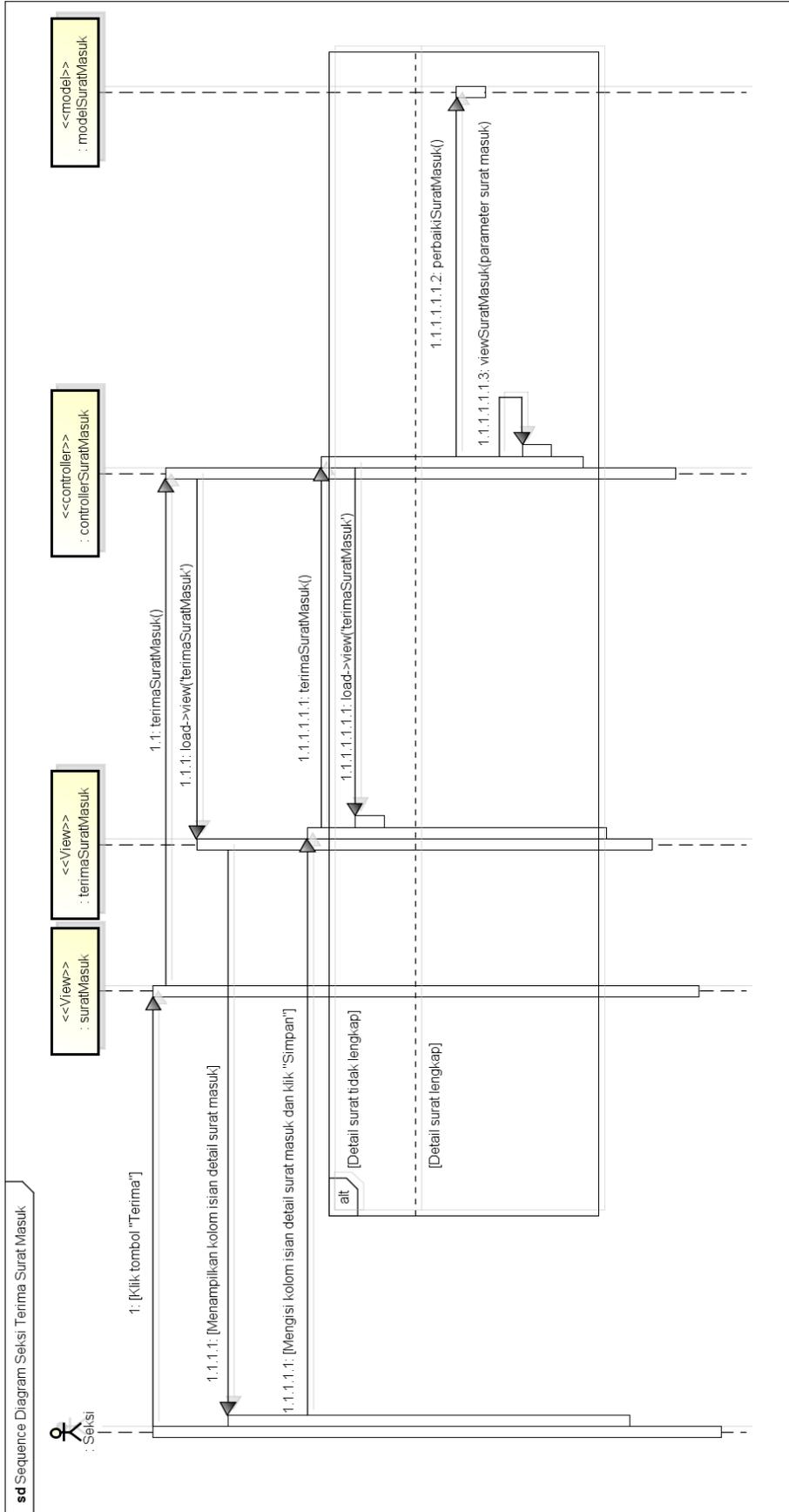
Seksi yang menerima surat masuk bisa membalas surat tersebut. Surat balasan termasuk surat keluar oleh karena itu menggunakan operasi 1.1.3.1.1.3 suratKeluar() untuk menyimpan detail surat balasan ke *database* surat keluar. Ketika membalas surat masuk seksi akan diperlihatkan detail surat masuk yang akan dibalas menggunakan operasi 1.1.1 cek(). Keseluruhan alur dalam aplikasi dapat dilihat pada *sequence* diagram dalam Gambar 4.12. Kelas dan metode atau fungsi serta paramter yang diperoleh dari *sequence* diagram proses ini disebutkan pada Tabel 4.6 di bawah ini. Sedikit berbeda dengan membuat surat keluar, pada fitur membuat surat balasan tidak memanggil fungsi reg() untuk membuat nomor register surat karena nomor register yang digunakan untuk surat keluar berupa surat balasan adalah nomor register surat masuk yang dibalas.

**Tabel 4. 6 Kelas dan Metode Pada Proses Membalas Surat Masuk**

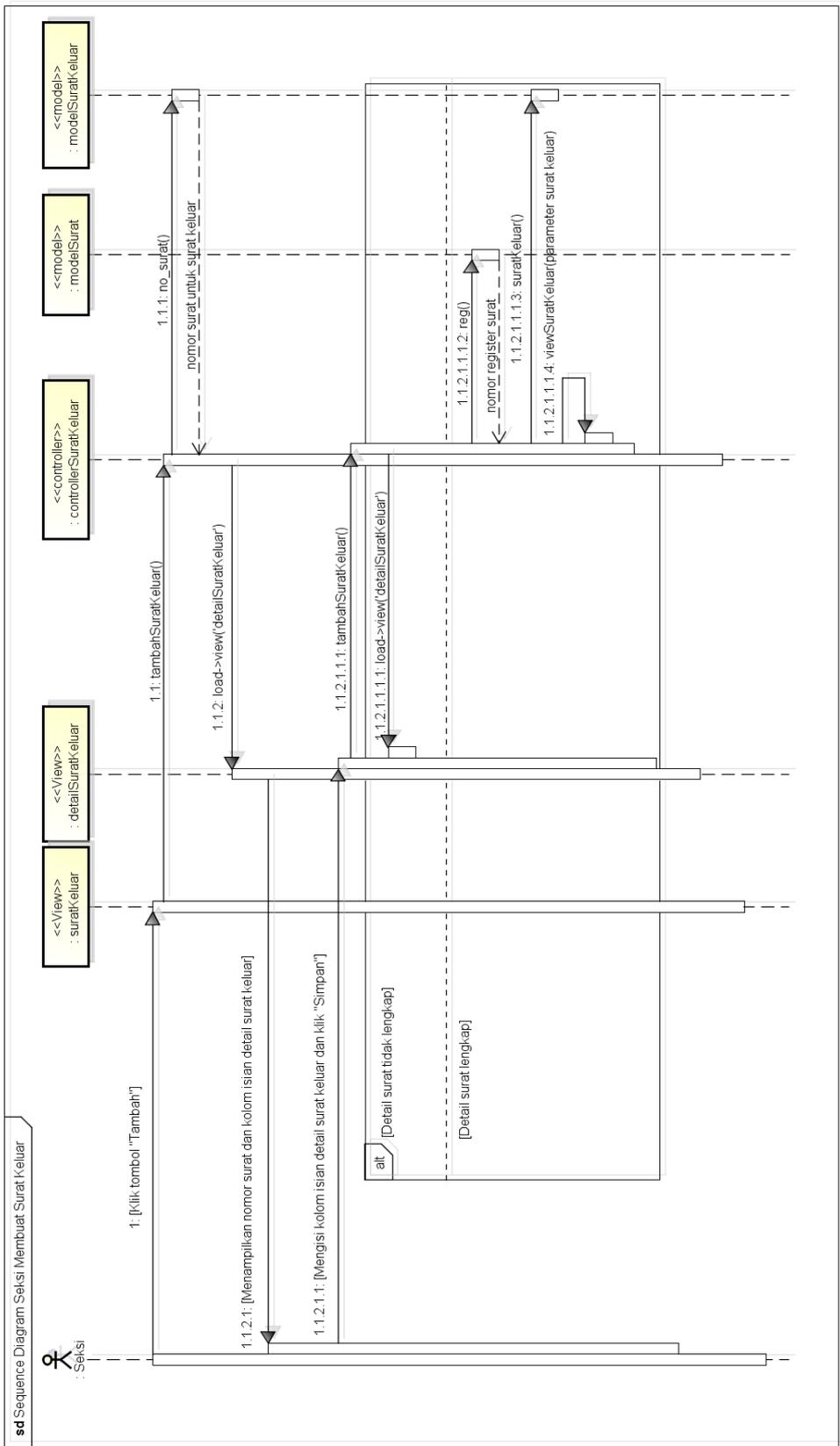
| Tipe           | Kelas                 | Metode / Fungsi  |
|----------------|-----------------------|--|
| <<controller>> | controllerSuratKeluar | + balasSuratMasuk() : void   |
|                | controllerSuratMasuk  | + viewSuratMasuk(\$sort_by : String, \$sort_order : String, \$offset : int, \$cari_array : ArrayList) : void |
| <<model>>      | modelSuratMasuk       | + cek(\$reg : String) : ArrayList  |
|                |                       | + perbaikiSuratMasuk(\$data : ArrayList, \$reg : String) : void  |
|                | modelSuratKeluar      | + no_surat() : String  |
|                |                       | + suratKeluar(\$dataSurat : ArrayList) : void  |
| <<view>>       | suratMasuk            |  |
|                | balasSuratMasuk       |  |



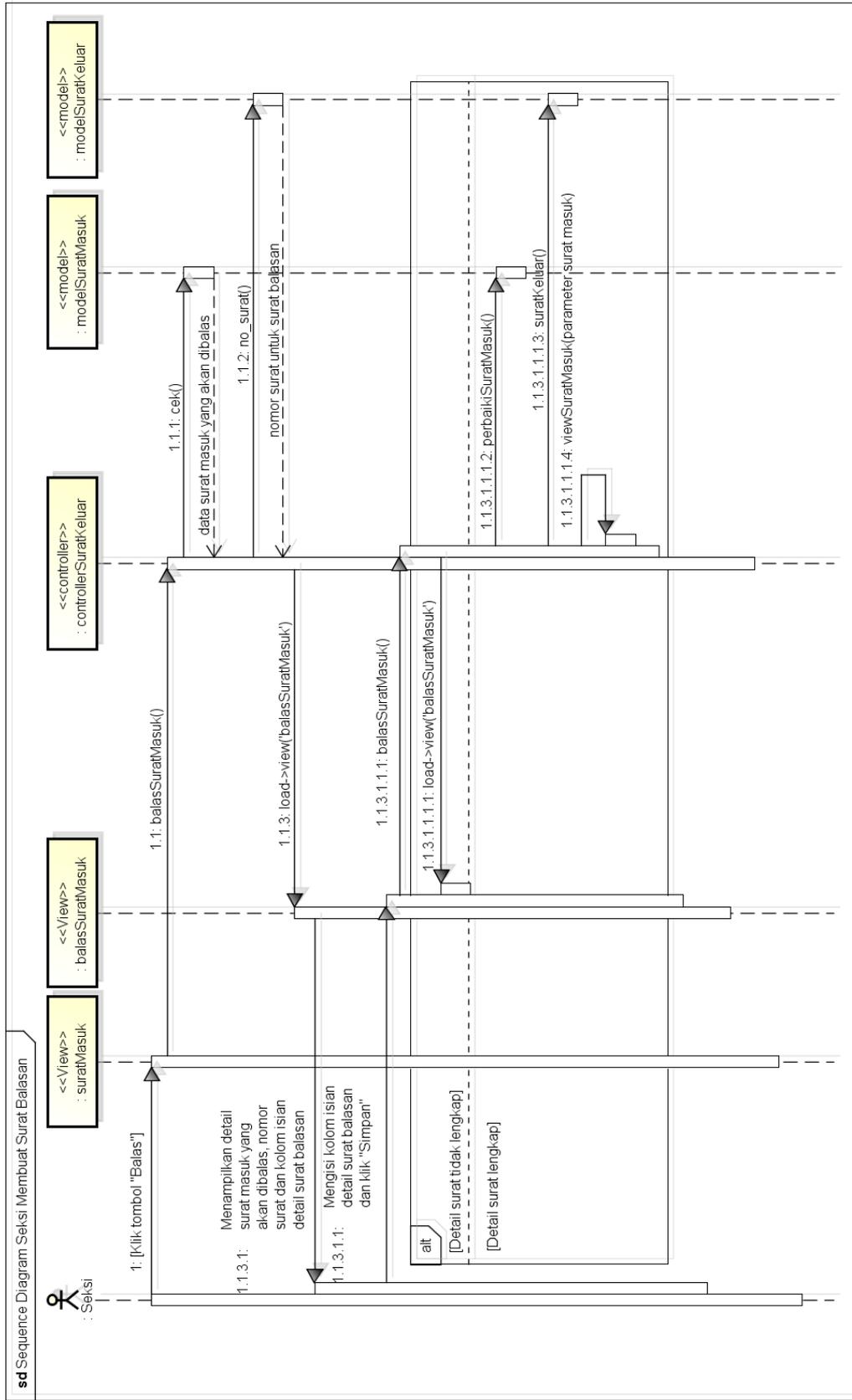
Gambar 4. 9 Sequence Diagram Sekretaris Tambah Surat Masuk



Gambar 4.10 Sequence Diagram Seksi Terima Surat Masuk



Gambar 4. 11 Sequence Diagram Seksi Membuat Surat Keluar



Gambar 4. 12 Sequence Diagram Seksi Membalas Surat Masuk

#### 4.2.4. Class Diagram

Dalam Gambar 4.13 adalah *class* diagram sistem informasi pengarsipan surat BRI Kanwil Malang yang menunjukkan kelas, metode atau fungsi serta hubungan antar kelas dalam aplikasi. *Class* diagram digunakan untuk mengelompokkan atribut dan metode / fungsi yang terdapat dalam kelas. Untuk menentukan fungsi-fungsi yang terlibat dalam suatu kelas maka dibutuhkan *sequence* diagram (Nugroho, 2009). Namun penambahan fungsi atau metode di luar *sequence* diagram bisa dilakukan apabila dianggap diperlukan oleh aplikasi.

Dari *sequence* diagram yang telah dibuat sebelumnya terlihat bahwa kelas akan dibedakan menjadi tiga bagian karena aplikasi ini menggunakan *framework* MVC, yaitu ada yang bertindak sebagai kelas *controller*, kelas *model*, dan *view*. Pembuatan *class* diagram dapat mempermudah dalam proses pembuatan aplikasi pada tahap implementasi. *Class* diagram yang dibuat juga menunjukkan hubungan asosiasi antar kelas. Terlihat dalam Gambar 4.13 kelas yang dibuat dikelompokkan dalam *package* *controller*, *model* dan *view* untuk mempermudah memahami peran masing-masing kelas.

**Tabel 4. 7 Metode atau Fungsi Kelas ‘controllersibri’ Sebagai Controller**

| Metode / Fungsi             | Penjelasan  |
|-----------------------------|---|
| + viewData() : void         | Digunakan untuk menampilkan seluruh data pengguna dan tidak memiliki nilai pengembalian.                          |
| + loginDaftar() : void      | Digunakan pengguna untuk masuk ke dalam aplikasi dan tidak memiliki nilai pengembalian.                           |
| + logout() : void           | Digunakan pengguna untuk keluar dari aplikasi dan tidak memiliki nilai pengembalian.                              |
| + perbaruiDataUser() : void | Digunakan seksi untuk mengedit <i>user name</i> dan <i>password</i> mereka dan tidak memiliki nilai pengembalian. |

**Tabel 4. 8 Metode atau Fungsi Kelas ‘controllerSuratMasuk’ Sebagai Controller**

| Metode / Fungsi  | Penjelasan   |
|--|--|
| + viewSuratMasuk(\$sort_by : String, \$sort_order : String, \$offset : int, \$cari_array : ArrayList) : void | Digunakan untuk menampilkan surat masuk dan tidak memiliki nilai |

|                             |  |
|-----------------------------|--|
|                             | <p>pengembalian. Fungsi ini memiliki parameter sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. <code>\$sort_by</code> : String<br/>Parameter untuk menentukan tampilan surat masuk berurutan berdasarkan kolom tertentu.</li> <li>2. <code>\$sort_order</code> : String<br/>Parameter untuk menentukan tampilan surat masuk berurutan secara <i>ascending</i> atau <i>descending</i>.</li> <li>3. <code>\$offset</code> : int<br/>Parameter untuk menunjukkan posisi arsip surat masuk yang ditampilkan pada <i>pagination</i>.</li> <li>4. <code>\$cari_array</code> : ArrayList<br/>Parameter yang digunakan untuk menyimpan kolom dan kata kunci pencarian arsip surat masuk.</li> </ol> |
| + cariSuratMasuk() : void   | Digunakan untuk melakukan pencarian surat masuk dan tidak memiliki nilai pengembalian.   |
| + tambahSuratMasuk() : void | Digunakan sekretaris untuk menambah surat masuk dan tidak memiliki nilai pengembalian.   |
| + perbaiki() : void         | Digunakan sekretaris untuk mengedit data umum surat masuk dan tidak memiliki nilai pengembalian.   |
| + terimaSuratMasuk() : void | Digunakan seksi untuk menerima surat masuk dan tidak memiliki nilai pengembalian.  |

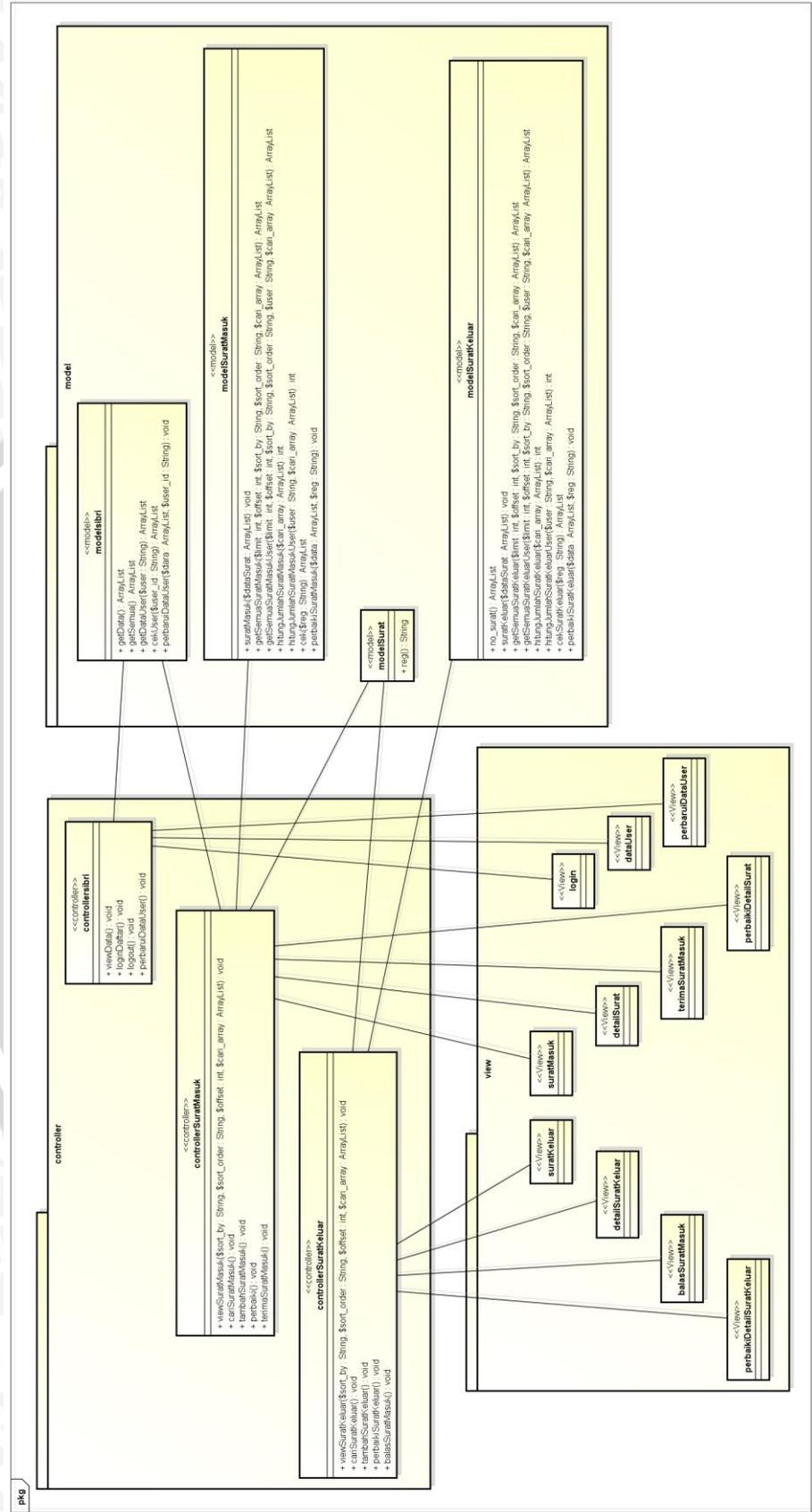
**Tabel 4. 9 Metode atau Fungsi Kelas ‘controllerSuratKeluar’ Sebagai Controller**

| Metode / Fungsi   | Penjelasan  |
|---|---|
| + viewSuratKeluar(\$sort_by : String, \$sort_order : String, \$offset : int, \$cari_array : ArrayList) : void | <p>Digunakan untuk menampilkan surat keluar dan tidak memiliki nilai pengembalian. Fungsi ini memiliki parameter sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. <code>\$sort_by</code> : String<br/>Parameter untuk menentukan tampilan surat keluar berurutan berdasarkan kolom tertentu.</li> <li>2. <code>\$sort_order</code> : String<br/>Parameter untuk menentukan tampilan surat keluar berurutan secara <i>ascending</i> atau <i>descending</i>.</li> <li>3. <code>\$offset</code> : int</li> </ol> |

|                                |   |
|--------------------------------|---|
|                                | Parameter untuk menunjukkan posisi arsip surat keluar yang ditampilkan pada <i>pagination</i> .<br>4. \$cari_array : ArrayList<br>Parameter yang digunakan untuk menyimpan kolom dan kata kunci pencarian arsip surat keluar. |
| + cariSuratKeluar() : void     | Digunakan untuk melakukan pencarian surat keluar dan tidak memiliki nilai pengembalian.   |
| + tambahSuratKeluar() : void   | Digunakan seksi untuk menambah atau membuat surat keluar dan tidak memiliki nilai pengembalian.   |
| + perbaikiSuratKeluar() : void | Digunakan seksi untuk mengedit detail surat keluar atau surat balasan dan tidak memiliki nilai pengembalian.  |
| + balasSuratMasuk() : void     | Digunakan seksi untuk membalas surat masuk dan tidak memiliki nilai pengembalian.   |

Kelas yang bertindak sebagai *controller* adalah kelas ‘*controllersibri*’, ‘*controllerSuratMasuk*’ dan ‘*controllerSuratKeluar*’. Metode-metode atau fungsi pada kelas ini didapat dari *sequence* diagram sebelumnya dan beberapa metode tambahan yang diperkirakan aplikasi akan membutuhkannya. Metode atau fungsi pada kelas ‘*controllersibri*’ disebutkan dan dijelaskan pada Tabel 4.7, kelas ini digunakan untuk operasi yang bersangkutan dengan data pengguna. Metode atau fungsi pada kelas ‘*controllerSuratMasuk*’ disebutkan dan dijelaskan pada Tabel 4.8, kelas ini digunakan untuk operasi yang bersangkutan dengan arsip surat masuk. Metode atau fungsi pada kelas ‘*controllerSuratKeluar*’ disebutkan dan dijelaskan pada Tabel 4.9, kelas ini digunakan untuk operasi yang bersangkutan dengan arsip surat keluar.

Sedangkan kelas yang bertindak sebagai *model* adalah kelas ‘*modelsibri*’, ‘*modelSurat*’, ‘*modelSuratMasuk*’ dan ‘*modelSuratKeluar*’. Kelas ini menghubungkan aplikasi dengan *database*. Mayoritas metode atau fungsi pada kelas *model* ini memiliki nilai pengembalian *ArrayList* yaitu tipe data mirip *array* tapi memiliki daya tampung dinamis karena ketika meminta data dari *database* aplikasi tidak mengetahui jumlah data yang akan dikembalikan dari *database*.



Gambar 4. 13 Class Diagram Sistem Informasi Pengarsipan Surat Bank Rakyat Indonesia Kantor Wilayah Malang

Metode atau fungsi pada kelas ‘modelsibri’ disebutkan dan dijelaskan pada Tabel 4.10, kelas ini digunakan untuk operasi database yang berhubungan dengan data pengguna. Metode atau fungsi pada kelas ‘modelSurat’ disebutkan dan dijelaskan pada Tabel 4.11, kelas ini digunakan untuk operasi database yang berhubungan dengan nomor register yang dimiliki oleh surat masuk atau pun surat keluar. Metode atau fungsi pada kelas ‘modelSuratMasuk’ disebutkan dan dijelaskan pada Tabel 4.12, kelas ini digunakan untuk operasi database yang berhubungan dengan data arsip surat masuk. Metode atau fungsi pada kelas ‘modelsibriSuratKeluar’ disebutkan dan dijelaskan pada Tabel 4.13, kelas ini digunakan untuk operasi database yang berhubungan dengan data arsip surat keluar.

**Tabel 4. 10 Metode / Fungsi Kelas ‘modelsibri’ Sebagai Model**

| Metode / Fungsi   | Penjelasan  |
|---|---|
| + getData() : ArrayList                                     | Digunakan untuk mendapatkan <i>user name</i> dan <i>password</i> dari <i>database</i> , memiliki nilai pengembalian bertipe data <i>ArrayList</i> .   |
| + getAll() : ArrayList                                      | Digunakan untuk mendapatkan data seluruh pengguna dari <i>database</i> , memiliki nilai pengembalian bertipe data <i>ArrayList</i> .  |
| + getUser(\$user : String) : ArrayList                      | Digunakan untuk mendapatkan data pengguna sesuai parameter \$user bertipe data <i>String</i> dari <i>database</i> , memiliki nilai pengembalian bertipe data <i>ArrayList</i> .   |
| + checkUser(\$user_id : String) : ArrayList                 | Digunakan untuk mengambil data pengguna sesuai parameter \$user_id bertipe data <i>String</i> yang menyimpan <i>id</i> pengguna yang akan ditampilkan. Fungsi ini memiliki nilai pengembalian bertipe data <i>ArrayList</i> .   |
| + updateUser(\$data : ArrayList, \$user_id : String) : void | Digunakan untuk mengedit <i>user name</i> dan <i>password</i> pengguna dalam <i>database</i> . Parameter \$data bertipe data <i>ArrayList</i> berisi <i>user name</i> dan <i>password</i> baru sedangkan \$user_id bertipe data <i>String</i> menyimpan <i>id</i> pengguna yang akan mengedit data. Fungsi ini tidak memiliki nilai pengembalian. |

Tabel 4. 11 Metode / Fungsi Kelas 'modelSurat' Sebagai Model

| Metode / Fungsi  | Penjelasan   |
|------------------|--|
| + reg() : String | Digunakan untuk membuat nomor register surat dan memiliki nilai pengembalian bertipe data <i>string</i> karena nomor register dalam <i>database</i> bertipe data <i>string</i> . |

Tabel 4. 12 Metode / Fungsi Kelas 'modelSuratMasuk' Sebagai Model

| Metode / Fungsi   | Penjelasan  |
|---|---|
| + suratMasuk(\$dataSurat : ArrayList) : void  | Digunakan untuk menyimpan data surat masuk yang dibuat ke dalam <i>database</i> . Fungsi ini tidak memiliki nilai pengembalian dan memiliki parameter \$dataSurat bertipe data <i>ArrayList</i> yang menyimpan data surat masuk hasil masukkan dari pengguna.   |
| + getAllSuratMasuk(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$cari_array : ArrayList) : ArrayList                      | Digunakan untuk mendapatkan semua surat masuk dari <i>database</i> . Memiliki nilai pengembalian bertipe data <i>ArrayList</i> . Fungsi ini memiliki parameter: <ol style="list-style-type: none"> <li>1. \$limit : int<br/>Parameter ini digunakan untuk membatasi surat masuk yang ditampilkan di setiap halaman</li> <li>2. \$offset : int<br/>Parameter untuk menunjukkan posisi arsip surat masuk yang ditampilkan pada <i>pagination</i>.</li> <li>3. \$sort_by : String<br/>Parameter untuk menentukan tampilan surat masuk berurutan berdasarkan kolom tertentu.</li> <li>4. \$sort_order : String<br/>Parameter untuk menentukan tampilan surat masuk berurutan secara <i>ascending</i> atau <i>descending</i>.</li> <li>5. \$cari_array : ArrayList<br/>Parameter yang digunakan untuk menyimpan kolom dan kata kunci pencarian arsip surat masuk.</li> </ol> |
| + getAllSuratMasukUser(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$user : String, \$cari_array : ArrayList) : ArrayList | Digunakan untuk mendapatkan semua surat masuk milik seksi dari <i>database</i> . Memiliki nilai pengembalian bertipe data <i>ArrayList</i> . Fungsi ini memiliki parameter: <ol style="list-style-type: none"> <li>1. \$limit : int<br/>Parameter ini digunakan untuk membatasi surat masuk yang ditampilkan di setiap halaman</li> <li>2. \$offset : int</li> </ol>  |

|  |   |
|--|---|
|  | <p>Parameter untuk menunjukkan posisi arsip surat masuk yang ditampilkan pada <i>pagination</i>.</p> <p>3. \$sort_by : String<br/>Parameter untuk menentukan tampilan surat masuk berurutan berdasarkan kolom tertentu.</p> <p>4. \$sort_order : String<br/>Parameter untuk menentukan tampilan surat masuk berurutan secara <i>ascending</i> atau <i>descending</i>.</p> <p>5. \$cari_array : ArrayList<br/>Parameter yang digunakan untuk menyimpan kolom dan kata kunci pencarian arsip surat masuk.</p> |
| <pre>+ hitungJumlahSuratMasuk(\$cari_array : ArrayList) : int</pre>                      | <p>Digunakan untuk menghitung total baris surat masuk yang akan ditampilkan. Fungsi ini memiliki nilai pengembalian <i>integer</i> dan memiliki parameter \$cari_array bertipe data <i>ArrayList</i> yang digunakan apabila menghitung total baris surat masuk hasil pencarian. Fungsi ini digunakan untuk <i>pagination</i>.</p>   |
| <pre>+ hitungJumlahSuratMasukUser(\$user : String, \$cari_array : ArrayList) : int</pre> | <p>Digunakan untuk menghitung total baris surat masuk milik seksi yang akan ditampilkan. Fungsi ini memiliki nilai pengembalian <i>integer</i> dan memiliki parameter \$cari_array bertipe data <i>ArrayList</i> yang digunakan apabila menghitung total baris surat masuk hasil pencarian serta \$user bertipe data <i>String</i> untuk menyimpan data pengguna yang melakukan akses. Fungsi ini digunakan untuk <i>pagination</i>.</p>  |
| <pre>+ cek(\$reg : String) : ArrayList</pre>   | <p>Digunakan untuk mendapatkan surat masuk sesuai nomor register (\$reg) yang menjadi parameter pada fungsi ini. Fungsi ini digunakan ketika seksi akan membuat surat balasan. Fungsi ini memiliki nilai pengembalian bertipe data <i>ArrayList</i>.</p>  |
| <pre>+ perbaikiSuratMasuk(\$data : ArrayList, \$reg : String) : void</pre>               | <p>Digunakan untuk mengedit data surat masuk dalam <i>database</i>. Fungsi ini tidak memiliki nilai pengembalian dan memiliki parameter \$data bertipe data <i>ArrayList</i> yang menyimpan data surat yang akan diganti dan \$reg bertipe data <i>String</i> yang menyimpan nomor register surat keluar yang akan diganti.</p>   |

Tabel 4. 13 Metode / Fungsi Kelas 'modelSuratKeluar' Sebagai Model

| Metode / Fungsi  | Penjelasan  |
|--|---|
| + no_surat() : String  | Digunakan untuk membuat nomor surat dan memiliki nilai pengembalian bertipe data <i>string</i> karena nomor surat dalam <i>database</i> bertipe data <i>string</i> .  |
| + suratKeluar(\$dataSurat : ArrayList) : void  | Digunakan untuk menyimpan data surat keluar yang dibuat ke dalam <i>database</i> . Fungsi ini tidak memiliki nilai pengembalian dan memiliki parameter \$dataSurat bertipe data <i>ArrayList</i> yang menyimpan data surat keluar hasil masukkan dari pengguna.   |
| + getSemuaSuratKeluar(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$cari_array : ArrayList) : ArrayList                      | Digunakan untuk mendapatkan semua surat keluar dari <i>database</i> . Memiliki nilai pengembalian bertipe data <i>ArrayList</i> . Fungsi ini memiliki parameter: <ol style="list-style-type: none"> <li>1. \$limit : int<br/>Parameter ini digunakan untuk membatasi surat keluar yang ditampilkan di setiap halaman</li> <li>2. \$offset : int<br/>Parameter untuk menunjukkan posisi arsip surat keluar yang ditampilkan pada <i>pagination</i>.</li> <li>3. \$sort_by : String<br/>Parameter untuk menentukan tampilan surat keluar berurutan berdasarkan kolom tertentu.</li> <li>4. \$sort_order : String<br/>Parameter untuk menentukan tampilan surat keluar berurutan secara <i>ascending</i> atau <i>descending</i>.</li> <li>5. \$cari_array : ArrayList<br/>Parameter yang digunakan untuk menyimpan kolom dan kata kunci pencarian arsip surat keluar.</li> </ol> |
| + getSemuaSuratKeluarUser(\$limit : int, \$offset : int, \$sort_by : String, \$sort_order : String, \$user : String, \$cari_array : ArrayList) : ArrayList | Digunakan untuk mendapatkan semua surat keluar dari <i>database</i> . Memiliki nilai pengembalian bertipe data <i>ArrayList</i> . Fungsi ini memiliki parameter: <ol style="list-style-type: none"> <li>1. \$limit : int<br/>Parameter ini digunakan untuk membatasi surat keluar yang ditampilkan di setiap halaman</li> <li>2. \$offset : int<br/>Parameter untuk menunjukkan posisi arsip surat keluar yang ditampilkan pada <i>pagination</i>.</li> <li>3. \$sort by : String</li> </ol>  |

|  |  |
|--|--|
|  | <p>Parameter untuk menentukan tampilan surat keluar berurutan berdasarkan kolom tertentu.</p> <p>4. \$sort_order : String<br/>Parameter untuk menentukan tampilan surat keluar berurutan secara <i>ascending</i> atau <i>descending</i>.</p> <p>5. \$cari_array : ArrayList<br/>Parameter yang digunakan untuk menyimpan kolom dan kata kunci pencarian arsip surat keluar.</p>  |
| + hitungJumlahSuratKeluar(\$cari_array : ArrayList) : int                      | Digunakan untuk menghitung total baris surat keluar yang akan ditampilkan. Fungsi ini memiliki nilai pengembalian <i>integer</i> dan memiliki parameter \$cari_array bertipe data <i>ArrayList</i> yang digunakan apabila menghitung total baris surat keluar hasil pencarian. Fungsi ini digunakan untuk <i>pagination</i> .  |
| + hitungJumlahSuratKeluarUser(\$user : String, \$cari_array : ArrayList) : int | Digunakan untuk menghitung total baris surat keluar milik seksi yang akan ditampilkan. Fungsi ini memiliki nilai pengembalian <i>integer</i> dan memiliki parameter \$cari_array bertipe data <i>ArrayList</i> yang digunakan apabila menghitung total baris surat keluar hasil pencarian serta \$user bertipe data <i>String</i> untuk menyimpan data pengguna yang melakukan akses. Fungsi ini digunakan untuk <i>pagination</i> . |
| + cekSuratKeluar(\$reg : String) : ArrayList                                   | Digunakan untuk mengambil data surat keluar sesuai parameter \$reg bertipe data <i>String</i> yang menyimpan nomor register surat keluar yang akan ditampilkan. Fungsi ini memiliki nilai pengembalian bertipe data <i>ArrayList</i> .   |
| + perbaikiSuratKeluar(\$data : ArrayList, \$reg : String) : void               | Digunakan untuk mengedit data surat keluar dalam <i>database</i> . Fungsi ini tidak memiliki nilai pengembalian dan memiliki parameter \$data bertipe data <i>ArrayList</i> yang menyimpan data surat yang akan diganti dan \$reg bertipe data <i>String</i> yang menyimpan nomor register surat keluar yang akan diganti.   |

*View* merupakan kelas untuk mengatur *interface* (antar muka) aplikasi yang menghubungkan pengguna dan aplikasi itu sendiri. Pada aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang ini menggunakan beberapa kelas *view* untuk mempermudah pengguna dalam berinteraksi dengan aplikasi. Kelas-kelas yang bertindak sebagai *view* akan dijelaskan pada Tabel 4.14. Setelah mengetahui kelas

dan metode yang terlibat dalam aplikasi maka dibuatlah *class* diagram untuk mempermudah dalam melakukan implementasi.

**Tabel 4. 14 Kelas-kelas Sebagai View**

| No | Kelas                     | Penjelasan   |
|----|---------------------------|--|
| 1  | login                     | Memberikan tampilan ketika pengguna melakukan <i>log in</i>  |
| 2  | dataUser                  | Memberi tampilan ketika menampilkan detail akun pengguna.  |
| 3  | perbaruiDataUser          | Memberi tampilan ketika pengguna memperbarui <i>user name</i> dan <i>password</i> .  |
| 4  | suratMasuk                | Memberi tampilan ketika pengguna menampilkan surat masuk.  |
| 5  | suratKeluar               | Memberi tampilan ketika pengguna menampilkan surat keluar.   |
| 6  | detailSurat               | Memberi tampilan ketika sekretaris menambah surat masuk dan mengisi data surat masuk.  |
| 7  | perbaikiDetailSurat       | Memberi tampilan ketika sekretaris memperbaiki data surat masuk.   |
| 8  | terimaSuratMasuk          | Memberi tampilan ketika seksi menerima surat masuk dan mengisi detail surat masuk.   |
| 9  | detailSuratKeluar         | Memberi tampilan ketika seksi membuat surat keluar dan mengisi detail surat keluar.  |
| 10 | perbaikiDetailSuratKeluar | Memberi tampilan ketika seksi memperbaiki detail surat keluar.   |
| 11 | balasSuratMasuk           | Memberi tampilan ketika seksi membalas surat masuk, menampilkan detail surat masuk yang akan dibalas dan mengisi detail surat balasan. |

#### 4.2.5. Entity Relationship Diagram

*Entity relationship* diagram dibuat untuk mempermudah dalam proses implementasi *database*. Dalam Gambar 4.14 rancangan *database* yang dibuat berdasarkan tabel pembukuan laporan arsip surat BRI Kanwil Malang dengan ditambahkan beberapa tabel dan kolom yang dibutuhkan untuk menunjang kebutuhan fungsional aplikasi. Tabel ‘pengguna’ adalah tabel yang menyimpan data pengguna yang dibedakan menjadi tabel ‘sekretaris’ dan ‘seksi’. Perbedaan tabel-tabel ini dilakukan karena objek sekretaris dan seksi ataupun surat keluar dan surat masuk memiliki beberapa yang berbeda.



**Gambar 4. 14 Entity Relationship Diagram Sistem Informasi Pengarsipan Surat Bank Rakyat Indonesia Kantor Wilayah Malang**

Atribut yang sama disimpan pada tabel induk yaitu tabel ‘pengguna’ untuk sub-tabel ‘sekreteraris’ dan ‘seksi’. Sedangkan untuk sub-tabel ‘surat\_keluar’ dan ‘surat\_masuk’ memiliki tabel induk ‘surat’. Tabel ‘seksi’ memiliki relasi satu ke banyak terhadap tabel ‘surat’, ini menunjukkan bahwa satu seksi bisa menerima banyak surat masuk dan bisa membuat banyak surat keluar. Tabel 4.15 di bawah ini akan menjelaskan lebih rinci kolom-kolom dalam tabel ‘pengguna’. Kolom-kolom pada tabel ‘pengguna’ harus terisi lengkap untuk dapat mengidentifikasi pengguna aplikasi. Tabel 4.16 menjelaskan lebih rinci kolom-kolom pada tabel ‘seksi’, sedangkan tabel ‘sekreteraris’ hanya menyimpan kolom ‘id\_pengguna’ seperti kolom pada tabel induk yaitu pada tabel ‘pengguna’.

Desain *database* dibuat sedemikian rupa seperti dalam Gambar 4.14 karena dengan menggunakan desain seperti itu maka diasumsikan dalam menjalankan fungsi *database* dalam aplikasi akan lebih efisien. Desain *database* akan lebih efisien dibandingkan dengan desain *database* apabila terdapat *database* yang dinormalisasi seperti berikut, yaitu dengan contoh terdapat tabel ‘surat’ yang berisi surat masuk dan surat keluar. Lalu terdapat tabel ‘tipe\_surat’ yang menyimpan *id* surat masuk dan surat keluar. Jadi dalam tabel ‘surat’ data surat masuk dan surat keluar dibedakan dengan suatu *id*. Dengan desain normalisasi seperti ini maka akan terjadi penggunaan kolom atau atribut yang tidak sesuai karena atribut untuk surat masuk dan surat keluar berbeda. Dengan desain seperti dalam Gambar 4.14 maka dapat dilihat dengan jelas perbedaan atribut yang dimiliki oleh ‘sekretaris’ dan ‘seksi’ yang merupakan sub-tabel dari tabel ‘pengguna’ dan tabel ‘surat\_masuk’ serta ‘surat\_keluar’ yang merupakan sub-tabel dari tabel ‘surat’.

**Tabel 4. 15 Penjelasan Kolom Tabel ‘pengguna’**

| Kolom                        | Tipe Data   | Kosong ( <i>null</i> ) | Penjelasan   | Contoh Data                    |
|------------------------------|-------------|------------------------|--|--------------------------------|
| id_pengguna<br>(primary key) | VARCHAR(8)  | Tidak boleh            | Digunakan untuk menyimpan <i>user name</i> untuk <i>log in</i> ke dalam aplikasi. Tipe data yang digunakan adalah <i>varchar</i> dengan alokasi maksimal 8 karakter.   | seksi_1                        |
| password                     | VARCHAR(8)  | Tidak boleh            | Digunakan untuk menyimpan <i>user name</i> untuk <i>log in</i> ke dalam aplikasi. Tipe data yang digunakan adalah <i>varchar</i> dengan alokasi karakter maksimal 8 karena <i>password</i> dibuat maksimal 8 karakter. | abcd1234                       |
| nama                         | VARCHAR(70) | Tidak boleh            | Digunakan untuk menyimpan nama pengguna. Tipe data yang digunakan adalah <i>varchar</i> dengan alokasi karakter maksimal 70 (Cabinet, 2001).   | A.Mardh<br>atillah<br>Sukandar |

Tabel 4. 16 Penjelasan Kolom Tabel ‘seksi’

| Kolom                        | Tipe Data   | Kosong<br>( <i>null</i> ) | Penjelasan   | Contoh Data                            |
|------------------------------|-------------|---------------------------|--|--|
| id_pengguna<br>(primary key) | VARCHAR(8)  | Tidak<br>boleh            | Digunakan untuk menyimpan <i>user name</i> untuk <i>log in</i> ke dalam aplikasi. Tipe data yang digunakan adalah <i>varchar</i> dengan alokasi maksimal 8 karakter.   | seksi_1                                |
| bidang                       | VARCHAR(30) | Tidak<br>boleh            | Digunakan untuk menyimpan nama bidang pengguna untuk aplikasi. Tipe data yang digunakan adalah <i>varchar</i> dengan alokasi karakter maksimal 30 karena nama bidang terpanjang saat ini adalah Manajemen Sumber Daya Manusia (MSDM) dengan 28 karakter. Maka 30 karakter cukup. | Manajemen<br>Sumber<br>Daya<br>Manusia |

Tabel ‘surat\_masuk’ digunakan untuk menyimpan data surat masuk yang diterima oleh BRI Kanwil Malang. Sedangkan tabel ‘surat\_keluar’ digunakan untuk menyimpan data surat keluar yang dikeluarkan oleh BRI Kanwil Malang. Tabel ‘surat\_masuk’ dan ‘surat\_keluar’ ini dapat diakses oleh seluruh pengguna, dengan batasan hak akses yang diatur oleh aplikasi.

Atribut-atribut sama milik surat masuk dan surat keluar disimpan pada tabel induk ‘surat’. Penjelasan lebih rinci tentang kolom-kolom pada tabel ‘surat’ dapat dilihat pada Tabel 4.17, lalu Tabel 4.18 menjelaskan kolom-kolom pada tabel ‘surat\_masuk’ dan Tabel 4.19 menjelaskan kolom-kolom pada tabel ‘surat\_keluar’. Aturan tidak boleh kosong (*not null*) pada kolom-kolom tabel surat masuk dan surat keluar diperoleh dari aturan pembukuan arsip surat.

Tabel 4. 17 Penjelasan Kolom Tabel ‘surat’

| Kolom                     | Tipe Data | Kosong<br>( <i>null</i> ) | Penjelasan   | Contoh Data |
|---------------------------|-----------|---------------------------|--|-------------|
| id_surat<br>(primary key) | SMALLINT  | Tidak<br>boleh            | Otomatis bertambah ketika terdapat arsip surat masuk atau surat keluar | 1           |

|               |                    |             |  |  |
|---------------|--------------------|-------------|--|--|
|               |                    |             | yang tersimpan. Kolom ini digunakan untuk membedakan data arsip surat. Tipe data menggunakan <i>SMALLINT</i> karena dengan pertimbangan bahwa dalam 1 tahun nomor ini tidak akan melebihi angka 65535.   |  |
| id_pengguna   | <i>VARCHAR(8)</i>  | Tidak boleh | Data ini diperoleh dari relasi antara tabel 'seksi' dan 'surat'. Data <i>id</i> ini digunakan untuk mengidentifikasi siapa yang bertanggung jawab terhadap arsip surat masuk yang diterima atau surat yang dikeluarkan.  | seksi_1  |
| reg           | <i>VARCHAR(11)</i> | Tidak boleh | Digunakan untuk menyimpan nomor register surat. Tipe data yang digunakan adalah <i>varchar</i> dengan alokasi karakter maksimal 11 karena nomor register surat memiliki format <i>yyyymmddxxx</i> dengan keterangan sebagai berikut: <ol style="list-style-type: none"> <li>1. <i>yyyy</i> adalah tahun,</li> <li>2. <i>mm</i> adalah bulan,</li> <li>3. <i>dd</i> adalah hari/tanggal,</li> <li>4. <i>xxx</i> adalah angka yang bertambah secara otomatis.</li> </ol> | 20146271   |
| perihal       | <i>MEDIUMTEXT</i>  | Tidak boleh | Digunakan untuk menyimpan perihal surat.   | Kerjasama Pendidikan   |
| dari_agenda   | <i>MEDIUMTEXT</i>  | Boleh       | Digunakan untuk menyimpan surat yang diterima atau dikirim dari agenda apa.  | Rapat terbuka bertempat di Widyaloka Universitas Brawijaya hari Minggu 22 Juni 2014. |
| dengan_agenda | <i>MEDIUMTEXT</i>  | Boleh       | Digunakan untuk menyimpan surat yang   | Batas akhir keputusan kerja sama   |

|               |                   |       |   |   |
|---------------|-------------------|-------|---|---|
|               |                   |       | diterima atau dikirim dengan agenda apa.  | ditunggu tanggal 30 Juni 2014 pada jam kerja.   |
| lampiran      | <i>TINYINT</i>    | Boleh | Digunakan untuk menyimpan jumlah lampiran pada surat. Tipe data yang digunakan adalah <i>tinyint</i> ( <i>tiny integer</i> ) karena <i>tinyint</i> dapat menyimpan data berupa angka dengan rentang paling kecil antara 0 sampai 255 dengan ukuran penyimpanan paling kecil 1 <i>byte</i> . | 0   |
| catatan       | <i>MEDIUMTEXT</i> | Boleh | Digunakan untuk menyimpan catatan untuk surat.  | Penting untuk segera dirapatkan kembali oleh petinggi BRI Kanwil Malang, apabila diperlukan maka segera meminta pertimbangan pada BRI Kampus Jakarta. |
| location_file | <i>MEDIUMTEXT</i> | Boleh | Digunakan untuk menyimpan lokasi penyimpanan arsip surat.   | *./uploads/s eks_1/Surat Keluar/1/1.jpg*  |

Tabel 4. 18 Penjelasan Kolom Tabel ‘surat\_keluar’

| Kolom                     | Tipe Data       | Kosong ( <i>null</i> ) | Penjelasan  | Contoh Data |
|---------------------------|-----------------|------------------------|---|-------------|
| id_surat<br>(primary key) | <i>SMALLINT</i> | Tidak boleh            | Data ini diperoleh dari relasi antara tabel ‘surat’ dan tabel ‘surat_keluar’  | 2           |
| no_surat                  | <i>SMALLINT</i> | Tidak boleh            | Digunakan untuk menyimpan nomor surat untuk surat keluar yang dibuat. Kolom ini menjadi <i>primary Key</i> pada tabel ‘surat_keluar’. Tipe data yang digunakan adalah | 58          |

|            |                   |             |   |  |
|------------|-------------------|-------------|---|--|
|            |                   |             | <i>smallint</i> ( <i>small integer</i> ) karena <i>smallint</i> dapat menyimpan data berupa angka dengan rentang antara 0 sampai 65535 dengan ukuran penyimpanan 2 <i>bytes</i> . Tipe data ini cukup untuk menyimpan nomor surat selama 1 tahun dengan asumsi 1 hari terdapat 5 surat keluar maka total surat keluar selama 1 tahun sebanyak 1825 nomor surat. Angka 1825 lebih dari <i>tinyint</i> (0-255) dan kurang dari 65535. |  |
| tgl_keluar | <i>TIMESTAMP</i>  | Tidak boleh | Kolom ini digunakan untuk menyimpan waktu surat dikeluarkan. Tipe data yang digunakan adalah <i>timestamp</i> karena <i>timestamp</i> memiliki format yang lebih fleksibel lengkap dengan detail jam.   | 2014-06-28<br>09:38:34   |
| tertuju    | <i>MEDIUMTEXT</i> | Tidak boleh | Digunakan untuk menyimpan nama orang atau instansi yang dituju surat keluar.  | Ketua<br>Department<br>Hubungan<br>Luar<br>Negeri<br>PT.Indora |

**Tabel 4. 19 Penjelasan Kolom Tabel ‘surat\_masuk’**

| Kolom                     | Tipe Data         | Kosong ( <i>null</i> ) | Penjelasan   | Contoh Data              |
|---------------------------|-------------------|------------------------|--|--------------------------|
| id_surat<br>(primary key) | <i>SMALLINT</i>   | Tidak boleh            | Data ini diperoleh dari relasi antara tabel ‘surat’ dan tabel ‘surat_masuk’  | 1                        |
| tgl_terima                | <i>TIMESTAMP</i>  | Tidak boleh            | Kolom ini menyimpan waktu surat diterima. Tipe data yang digunakan adalah <i>timestamp</i> karena <i>timestamp</i> memiliki format yang lebih fleksibel lengkap dengan detail jam. | 2014-06-27<br>13:45:56   |
| pengirim                  | <i>MEDIUMTEXT</i> | Tidak boleh            | Digunakan untuk menyimpan nama pengirim surat.   | Universitas<br>Brawijaya |
| status_balasan            | <i>BOOLEAN</i>    | Boleh                  | Membantu penngguna untuk mengetahui apakah surat   | 0                        |

|               |                |       |  |   |
|---------------|----------------|-------|--|---|
|               |                |       | masuk yang diterima telah dibalas atau tidak. Tipe data yang digunakan adalah <i>boolean</i> , bernilai <i>false</i> (0) apabila tidak dibalas dan bernilai <i>true</i> (1) apabila sudah dibalas.   |   |
| status_terima | <i>BOOLEAN</i> | Boleh | Digunakan untuk menandakan apakah surat masuk yang diterima telah diterima oleh seksi atau tidak. Tipe data yang digunakan adalah <i>boolean</i> , bernilai <i>false</i> (0) apabila belum diterima dan bernilai <i>true</i> (1) apabila sudah diterima. | 0 |

Mayoritas dari kolom-kolom dengan masukan berupa karakter atau kalimat menggunakan tipe data *mediumtext*. Tipe data *mediumtext* ini sangat fleksibel dengan jumlah karakter maksimum 16.777.215 atau dengan kapasitas maksimal 16 MB. Meskipun pada tipe data ini disediakan alokasi data maksimal 16 MB namun apabila data yang disimpan hanya 100 karakter maka yang tersimpan adalah 100 *byte* (Editor, 2014). Pemilihan tipe data ini agar menjamin bahwa semua data tersimpan untuk kepentingan pengarsipan surat.

#### 4.3.Desain Tampilan Aplikasi

Pada tahap ini akan dijelaskan mengenai desain dari tampilan (*view*) yang akan digunakan oleh pengguna dalam tampilan *web* untuk tampilan aplikasi pengarsipan surat BRI Kanwil Malang. Dalam desain *view* akan dijelaskan komponen-komponen dalam tampilan yang akan diberikan oleh aplikasi. Terdapat 11 rancangan yaitu 'login', 'dataUser', 'perbaruiDataUser', 'suratMasuk', 'suratKeluar', 'detailSurat', 'perbaikiDetailSurat', 'terimaSuratMasuk', 'detailSuratKeluar', 'perbaikiDetailSuratKeluar', dan 'balasSuratMasuk'.

Tambah

Data Pengguna

Kategori

Surat Masuk

Kategori

Surat Keluar

Kata Kunci

Surat Keluar

Cari

| No             | REG | TANGGAL SURAT DITERIMA | PENGIRIM | DITUJUKAN PADA | PERIHAL | DARI AGENDA | DENGAN AGENDA | LAMPIRAN | CATATAN | ARSIP | BALASAN |
|----------------|-----|------------------------|----------|----------------|---------|-------------|---------------|----------|---------|-------|---------|
| 1              |     |                        |          |                |         |             |               |          |         | 1     | 2       |
| 2              |     |                        |          |                |         |             |               |          |         |       |         |
| 3              |     |                        |          |                |         |             |               |          |         |       |         |
| 4              |     |                        |          |                |         |             |               |          |         |       |         |
| 5              |     |                        |          |                |         |             |               |          |         |       |         |
| dan seterusnya |     |                        |          |                |         |             |               |          |         |       |         |

Perbaiki

Perbaiki

Perbaiki

Perbaiki

Perbaiki

Perbaiki

Previous

1

2

3

Next

Keluar

Gambar 4. 15 Desain View 'suratMasuk' Akses Dilakukan Sekretaris

Pada desain *view* 'login' akan disediakan kolom untuk memasukkan *id* pengguna dan *password* lalu terdapat tombol 'Masuk' yang harus diklik oleh pengguna untuk dapat masuk ke dalam aplikasi. Desain dari *view* 'login' dapat dilihat dalam Lampiran 4 Gambar L.4.1. *View* berikutnya adalah 'dataUser'. *View* ini dapat menampilkan profil dari seluruh pengguna dalam bentuk tabel apabila yang melakukan akses adalah sekretaris, sedangkan apabila yang melakukan akses adalah seksi maka *view* ini akan menampilkan profil seksi yang melakukan akses dalam bentuk *list*.

Desain *view* 'dataUser' apabila yang melakukan akses adalah sekretaris dapat dilihat dalam Lampiran 4 Gambar L.4.2 dan apabila akses dilakukan seksi maka desain dapat dilihat dalam Lampiran 4 Gambar L.4.3. Dalam tampilan *view* 'dataUser' pengguna dapat memperbaiki profil mereka dengan menekan tombol 'Perbarui' lalu aplikasi akan meneruskan pengguna pada tampilan *view* 'perbaruiDataUser'. Desain untuk *view* 'perbaruiDataUser' dapat dilihat dalam Lampiran 4 Gambar L.4.4. Pada tampilan *view* 'perbaruiDataUser' ini pengguna dapat memperbaiki nama, *id* pengguna, atau *password*.

Terdapat 3 tampilan utama dari aplikasi yang dapat diakses oleh aplikasi salah satunya ada *view* 'dataUser', lalu sisanya adalah *view* 'suratMasuk' untuk menampilkan arsip surat masuk dan 'suratKeluar' untuk menampilkan arsip surat keluar. Oleh karena itu pada setiap *view* tersebut terdapat 4 tombol utama yaitu tombol 'Data Pengguna', 'Surat Masuk', 'Surat Keluar' dan terdapat tombol 'Keluar' yang dapat digunakan untuk keluar dari aplikasi. Tombol 'Data Pengguna' digunakan untuk menampilkan *view* 'dataUser', tombol 'Surat Masuk' untuk menampilkan *view* 'suratMasuk', dan tombol 'Surat Keluar' untuk menampilkan *view* 'suratKeluar'. Pada *view* 'suratMasuk' dan 'suratKeluar' memiliki kesamaan karena arsip surat masuk dan surat keluar ditampilkan dalam tabel.

Kolom-kolom pada tampilan data arsip surat masuk maupun surat keluar didapat dari hasil perancangan database (ERD). Dalam Gambar 4.15 menunjukkan desain tampilan *view* 'suratMasuk' dengan akses yang dilakukan oleh sekretaris sedangkan untuk desain *view* 'suratMasuk' dengan akses yang dilakukan oleh seksi dapat dilihat dalam Lampiran 4 Gambar L.4.5. Pada *view* 'suratMasuk' sekretaris dapat menambahkan surat masuk dengan cara klik tombol 'Tambah'. Tombol

dengan kode 1 pada kolom 'ARSIP' digunakan untuk menampilkan arsip surat tersebut dan tombol kode 2 pada kolom 'BALASAN' digunakan untuk melihat surat balasan dari surat masuk.

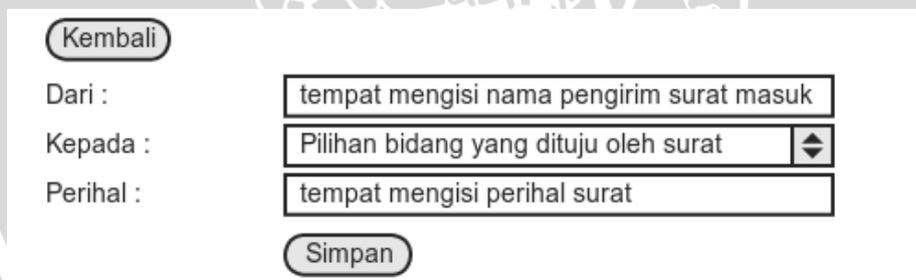
Pada tampilan *view* 'suratMasuk' terdapat tombol 'Perbaiki' yang dapat digunakan sekretaris untuk memperbaiki data umum surat masuk yang telah disimpan. Sedangkan dalam tampilan *view* 'suratMasuk' ketika akses dilakukan oleh seksi tombol ini awalnya adalah tombol 'Terima' yang digunakan untuk menerima surat masuk dan seksi bersangkutan dapat memasukkan detail surat masuk yang diterima. Setelah detail surat masuk tersimpan tombol tersebut diganti dengan tombol 'Perbaiki' yang dapat digunakan oleh seksi untuk memperbaiki detail surat masuk yang telah tersimpan. Sebelah kanan dari tombol ini terdapat tombol 'Balas' yang muncul setelah detail surat masuk telah disimpan. Tombol balas ini digunakan seksi untuk membalas surat masuk yang dimaksud. Pada *view* 'suratMasuk' dan 'suratKeluar' terdapat *pagination* untuk membagi tampilan arsip surat sesuai batas maksimal yang diatur dalam aplikasi.

Sama dengan desain *view* 'suratMasuk', untuk *view* 'suratKeluar' arsip surat akan ditampilkan dalam tabel. Sedikit perbedaan pada tampilan *view* 'suratKeluar', karena tombol 'Tambah' yang sebelumnya pada *view* 'suratMasuk' ada di tampilan untuk akses yang dilakukan oleh sekretaris namun pada *view* 'suratKeluar' tombol tersebut ada di tampilan untuk akses oleh seksi yang dapat digunakan untuk menambahkan arsip surat keluar. Pada tampilan *view* 'suratKeluar' dengan akses yang dilakukan oleh seksi terdapat tombol 'Perbaiki' untuk memperbaiki surat keluar yang telah tersimpan, sedangkan pada tampilan *view* 'suratKeluar' yang diakses oleh sekretaris tidak terdapat tambahan tombol lain selain tombol utama. Desain *view* 'suratKeluar' dapat dilihat dalam Lampiran 4 Gambar L.4.6 dan Lampiran 4 Gambar L.4.7.

Apabila tombol 'Tambah' pada tampilan *view* 'suratMasuk' diklik oleh sekretaris maka sekretaris akan diteruskan pada *view* 'detailSurat' untuk memasukkan data umum surat masuk. Desain tampilan *view* 'detailSurat' sama dengan tampilan *view* 'perbaikiDetailSurat' yang muncul apabila tombol 'Perbaiki' diklik oleh sekretaris pada *view* 'suratMasuk'. Desain *view* 'detailSurat' dan 'perbaikiDetailSurat' dapat dilihat dalam Gambar 4.16.

Dalam Gambar 4.18 menunjukkan tampilan *view* terimaSuratMasuk yang digunakan oleh seksi untuk memasukkan detail surat masuk yang ditujukan pada seksi yang bersangkutan. Tombol ‘Pilih File’ digunakan untuk memilih *file* hasil *scan* arsip surat masuk yang diterima. Sedangkan tombol ‘Pilih File’ pada tampilan *view* detailSuratKeluar, perbaikiDetailSuratKeluar, dan balasSuratMasuk digunakan untuk memilih *file* hasil *scan* arsip surat keluar ataupun surat balasan yang dibuat.

Desain *view* dari detailSuratKeluar dan perbaikiDetailSuratKeluar dapat dilihat dalam Lampiran 4 Gambar L.4.8, sedangkan desain *view* dari balasSuratMasuk dapat dilihat dalam Gambar 4.17. Desain tampilan *view* detailSuratKeluar, perbaikiDetailSuratKeluar memiliki desain yang sama, hanya saja pada kolom isian detail surat pada *view* perbaikiDetailSuratKeluar akan menampilkan detail surat sebelumnya yang pernah disimpan sebagai acuan dalam memperbaiki isi detail surat. Dalam Gambar 4.17 untuk desain tampilan *view* balasSuratMasuk hampir sama dengan desain detailSuratKeluar namun pada tampilan tersebut terdapat lokasi untuk menampilkan detail surat masuk yang akan dibalas sebagai acuan seksi dalam mengisi detail surat balasan yang dibuat.



|  |   |
|--|---|
| <input type="button" value="Kembali"/> |   |
| Dari :                                 | <input type="text" value="tempat mengisi nama pengirim surat masuk"/> |
| Kepada :                               | <input type="text" value="Pilihan bidang yang dituju oleh surat"/>    |
| Perihal :                              | <input type="text" value="tempat mengisi perihal surat"/>             |
|  | <input type="button" value="Simpan"/>                                 |

**Gambar 4. 16 Desain View ‘detailSurat’ dan ‘perbaikiDetailSurat’**

[Kembali](#)

Tempat menampilkan detail surat masuk yang akan dibalas

No. Surat :            tempat nomor surat ditampilkan

Kepada :               

Perihal :               

Dari Agenda :        

Dengan Agenda :    

Lampiran :            

Catatan :               

Arsip :                

**Gambar 4. 17 Desain View ‘balasSuratMasuk’**

[Kembali](#)

Reg :                      tempat nomor register surat ditampilkan

Tanggal Terima Surat :    tempat tanggal surat diterima ditampilkan

Dari :                     tempat nama pengirim surat ditampilkan

Kepada :                 tempat menampilkan bidang yang dituju oleh surat

Perihal :                 tempat perihal surat ditampilkan

Dari Agenda :           

Dengan Agenda :       

Lampiran :              

Catatan :               

Arsip :                

**Gambar 4. 18 Desain View ‘terimaSuratMasuk’**

## BAB V

### IMPLEMENTASI DAN PENGUJIAN

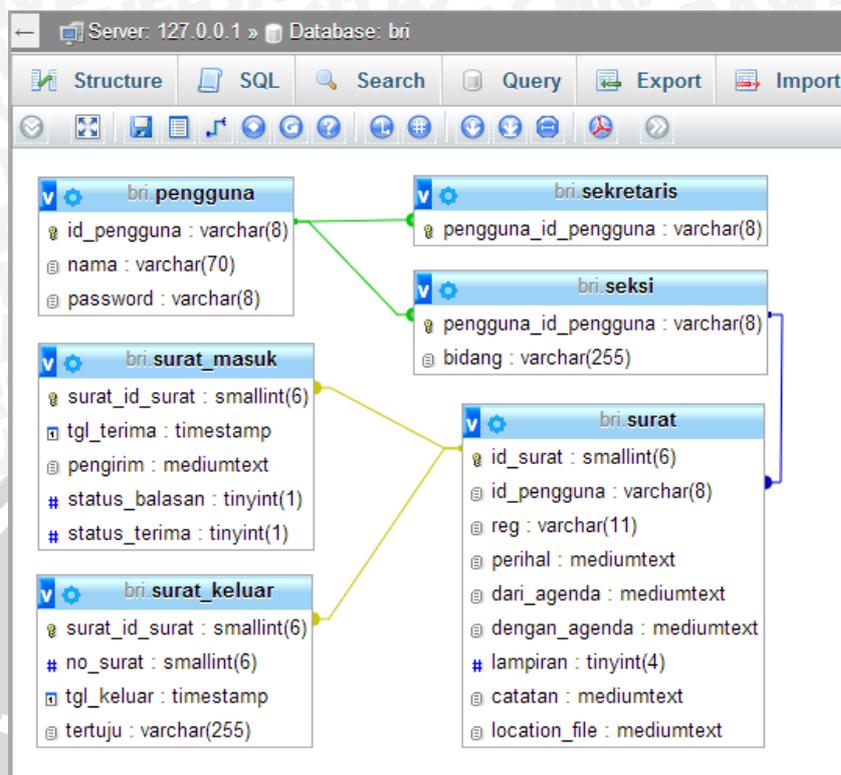
Pada bab ini akan dibahas tentang implementasi pembuatan aplikasi dan pengujian pada aplikasi yang dilakukan berdasarkan metode penelitian dan perancangan dari sistem yang telah dibuat lalu menguraikan hasil analisis dari pengujian.

#### 5.1. Implementasi

Berdasarkan perancangan dan analisis kebutuhan yang telah dijelaskan sebelumnya, berikut adalah penjelasan tentang implementasi dari tiap tahapan proses. Implementasi yang dilakukan dibedakan menjadi implementasi yang dilakukan pada *database* dan implementasi kode program dalam perangkat lunak (aplikasi) itu sendiri.

##### 5.1.1. Implementasi Database

Penggunaan *database* dalam pengarsipan surat dapat menyelesaikan permasalahan boros penggunaan kertas untuk keperluan arsip, dan penggunaan ruang penyimpanan arsip yang semakin lama semakin menumpuk. Implementasi pada penyimpanan data dilakukan dengan MySQL. Hasil implementasi penyimpanan data ini berupa *file database* dengan format *file extension* \*.sql. Hasil implementasi *database* dengan MySQL ini dimodelkan dalam ERD diagram. Pada konseptual ERD diagram terdapat hubungan relasi antar tabel. Dalam Gambar 4.14 menggambarkan diagram konseptual *entity relationship* Diagram (ERD) dari aplikasi sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang. Setelah pada bab sebelumnya telah dijelaskan tentang ERD yaitu tentang perancangan *database* dari sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang, maka dari hasil perancangan tersebut menghasilkan *database* dengan nama 'bri\_db'. Berikut Gambar 5.1 adalah hasil screenshot dari tabel-tabel database 'bri\_db'.



**Gambar 5.1** Tabel-tabel *Database* 'bri\_db'

Seperti yang terlihat pada gambar di atas terdapat 6 tabel di dalam *database* 'bri\_db'. Tabel 'pengguna' adalah tabel yang menyimpan data pengguna yang dibedakan menjadi tabel 'sekretaris' dan 'seksi'. Perbedaan tabel-tabel ini dilakukan karena objek sekretaris dan seksi ataupun surat keluar dan surat masuk memiliki beberapa yang berbeda. Atribut yang sama disimpan pada tabel induk yaitu tabel 'pengguna' untuk sub-tabel 'sekretaris' dan 'seksi'. Sedangkan untuk sub-tabel 'surat\_keluar' dan 'surat\_masuk' memiliki tabel induk 'surat'. Proses pembuatan *database* menggunakan *query* sesuai dengan perancangan pada ERD dalam Gambar 4.14. Berikut dalam Kode 5.1 merupakan cuplikan *query* untuk membuat tabel 'surat', 'surat\_masuk', dan 'surat\_keluar' yang merupakan beberapa tabel dari *database* 'bri\_db'.

```

1 CREATE TABLE IF NOT EXISTS `surat` (
2   `id_surat` smallint(6) NOT NULL AUTO_INCREMENT,
3   `id_pengguna` varchar(8) NOT NULL,
4   `reg` varchar(11) NOT NULL,
5   `perihal` mediumtext NOT NULL,
6   `dari_agenda` mediumtext,
7   `dengan_agenda` mediumtext,

```

```
8  `lampiran` tinyint(4) DEFAULT '0',
9  `catatan` mediumtext,
10 `location_file` mediumtext,
11 PRIMARY KEY (`id_surat`),
12 KEY `fk_surat_seksi1_idx` (`id_pengguna`)
13 );
14 CREATE TABLE IF NOT EXISTS `surat_keluar` (
15   `surat_id_surat` smallint(6) NOT NULL,
16   `no_surat` smallint(6) NOT NULL,
17   `tgl_keluar` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
18   `tertuju` varchar(255) NOT NULL,
19   PRIMARY KEY (`surat_id_surat`)
20 );
21 CREATE TABLE IF NOT EXISTS `surat_masuk` (
22   `surat_id_surat` smallint(6) NOT NULL,
23   `tgl_terima` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
24   `pengirim` mediumtext NOT NULL,
25   `status_balasan` tinyint(1) DEFAULT '0',
26   `status_terima` tinyint(1) DEFAULT '0',
27   PRIMARY KEY (`surat_id_surat`)
28 );
```

**Kode 5.1 Query Untuk Membuat Tabel ‘surat’, ‘surat\_masuk’, dan ‘surat\_keluar’**

### 5.1.2. Implementasi Perangkat Lunak

Pada tahapan ini akan dijelaskan mengenai implementasi kode program yang dilakukan untuk membangun aplikasi Sistem Informasi Pengarsipan Surat BRI Kanwil Malang. Aplikasi Sistem Informasi Pengarsipan Surat BRI Kanwil Malang dibangun dari fitur-fitur yang telah dijelaskan pada perancangan diagram *use case* diagram sebelumnya yaitu pada tabel 4.2 menggunakan *framework* CodeIgniter 2.1.4. Seluruh fitur tersebut diimplementasikan dalam aplikasi dan akan dijelaskan dalam tahapan ini. Proses implementasi dilakukan berdasarkan *sequence* dan *class* diagram sesuai dengan fitur yang akan dibangun.

Tambah

Data Pengguna

Surat Masuk

Surat Keluar

Cari

REG

Ditemukan 3 surat keluar

| REG      | NOMOR SURAT | TANGGAL SURAT DIKIRIM ▲ | PENGIRIM | DITUJUKAN PADA | PERIHAL     | JAWABAN DARI AGENDA NOMOR | JAWABAN DENGAN AGENDA NOMOR | LAMPIRAN | CATATAN        | ARSIIP                              | BALASAN DARI | Perbaiki |
|----------|-------------|-------------------------|----------|----------------|-------------|---------------------------|-----------------------------|----------|----------------|-------------------------------------|--------------|----------|
| 20147153 | 1           | 2014-07-15<br>02:42:54  | bidang2  | tesKepada3     | tesPerihal3 |                           |                             | 1        | Penting banget | Halaman 1<br>Halaman 2              |              | Perbaiki |
| 20147142 | 2           | 2014-07-15<br>06:10:16  | bidang2  | tesPengirim2   | tesPerihal4 |                           |                             | 2        |                | Halaman 1<br>Halaman 2<br>Halaman 3 | 20147142     | Perbaiki |
| 20147142 | 3           | 2014-07-15<br>06:35:51  | bidang2  | tesPengirim2   | tesPerihal5 |                           |                             | 0        | Susulan        | Halaman 1                           | 20147142     | Perbaiki |

Keluar

Gambar 5. 2 View 'suratKeluar'

Fitur-fitur yang terdapat dalam aplikasi sangat erat hubungannya dengan hak akses pengguna terhadap aktivitas yang dapat dilakukan pada arsip surat maka aplikasi ditambahkan fitur *login* untuk masuk ke dalam aplikasi sehingga aplikasi dapat mengetahui siapa yang melakukan akses terhadap aplikasi dan *logout* untuk keluar dari aplikasi. Ketika melakukan *login* pengguna akan memasukkan ID Pengguna dan *Password* ke dalam kolom isian pada *view* 'login'. Setelah pengguna mengklik tombol 'Masuk' maka data isian hasil masukkan dari pengguna tersebut akan dikirim ke kelas 'controller' fungsi 'loginDaftar'. Dalam fungsi 'loginDaftar' data isian dari pengguna tersebut dikirimkan ke kelas 'modelsibri' fungsi 'getData' untuk memperoleh hasil apakah ID Pengguna dan *Password* yang dimasukkan oleh pengguna cocok dengan salah satu data dalam *database*.

```

... ..
125 foreach($hasil as $row)
126 {
127 $file = explode(" ", $row->location_file,-1);
128 $shalaman = "Halaman ";
129 $surutan = 0;
130 echo '
131 <tr>
132 <td>'.$row->reg.'</td>
133 <td>'.$row->no_surat.'</td>
134 <td>'.$row->tgl_keluar.'</td>
135 <td>'.$row->pengirim.'</td>
136 <td>'.$row->tertuju.'</td>
137 <td>'.$row->perihal.'</td>
138 <td>'.$row->dari_agenda.'</td>
139 <td>'.$row->dengan_agenda.'</td>
140 <td>'.$row->lampiran.'</td>
141 <td>'.$row->catatan.'</td>
... ..

```

**Kode 5. 2 Cuplikan Kode View 'suratKeluar'**

Selanjutnya apabila ditemukan kecocokan maka fungsi ini mengidentifikasi pengguna ini memiliki hak akses sekretaris atau seksi bidang tertentu lalu mengembalikan hasilnya pada fungsi 'loginDaftar' pada kelas 'controller' untuk kemudian akan menentukan apabila terdapat kecocokan maka fungsi 'loginDaftar' akan meneruskan pengguna ke dalam aplikasi namun apabila pengguna tidak dapat diidentifikasi maka fungsi akan menampilkan *view* 'login' kembali. Ketika aplikasi

dapat mengidentifikasi pengguna yang melakukan akses maka pengguna dapat menggunakan fitur-fitur dalam aplikasi sesuai hak akses masing-masing. Seperti yang tertulis pada Tabel 4.2 yaitu terdapat 11 fitur dari aplikasi Sistem Informasi Pengarsipan Surat BRI Kanwil Malang.

```

...
138 if($this->input->post('nomor_surat') <> null &&
139 $_FILES['upload']['tmp_name']['0'] == null)
140 {
141     $data_kirim['berita'] = "Arsip Belum Terisi atau perihal
dan kepada kosong";
142     $data_kirim['nomor_surat'] = $this->input-
>post('nomor_surat');
143     $this->load->view('detailSuratKeluar',$data_kirim);
144 }
145 else
146 {
147     $nomor_surat = $this->modelsibri->no_surat();
148     $data_kirim = array(
149         'nomor_surat' => $nomor_surat,
150         'berita' => '*'
151     );
152     $this->load->view('detailSuratKeluar',$data_kirim);
153 }
...

```

### Kode 5. 3 Cuplikan Kode Program Fungsi ‘tambahSuratKeluar’ Kelas ‘controllerSuratKeluar’

Dalam tahapan ini akan dijelaskan proses implementasi fitur membuat surat keluar (UC-09) sebagai contoh proses implementasi dari semua fitur dalam aplikasi. Fitur ini dipilih karena menggunakan beberapa fitur sekaligus. Implementasi untuk fitur ini dibuat berdasarkan *sequence* diagram dalam Gambar 4.11 dan *class* diagram dalam Gambar 4.13. Ketika aplikasi sedang menampilkan surat keluar pada tampilan *view* ‘suratKeluar’ seperti dalam Gambar 5.2 terdapat tombol ‘Tambah’ apabila yang melakukan akses adalah seksi. Tombol tersebut digunakan untuk menjalankan fitur membuat surat keluar (UC-09). Dalam Kode 5.2 menunjukkan bagaimana *View* ‘suratKeluar’ menampilkan seluruh data arsip surat keluar. Data arsip surat keluar yang diperoleh dari *database* merupakan dalam bentuk *arraylist* lalu ditampilkan menggunakan pengulangan *foreach*.

```
... ..
4 public function no_surat()
5 {
6     $query = $this->db->query('SELECT no_surat FROM
    surat_keluar order by surat_id_surat DESC limit 1');
7     if($query->result() == null)
8     {
9         $nomor_surat = 1;
10    }
11    else
12    {
13        $nomor_surat = $query->row()->no_surat;
14        $nomor_surat = $nomor_surat + 1;
15    }
16    return $nomor_surat;
17 }
... ..
```

**Kode 5. 4 Cuplikan Kode Program Fungsi ‘no\_surat’ Kelas  
‘modelSuratKeluar’**

Ketika tombol ‘Tambah’ diklik maka fungsi ‘tambahSuratKeluar’ pada kelas ‘controllerSuratKeluar’ akan dijalankan. Fungsi ini tidak membutuhkan parameter dan tidak memberikan *return value*. Dalam Kode 5.3 terlihat saat fungsi ‘tambahSuratKeluar’ dijalankan maka aplikasi akan membuat nomor surat yang dapat digunakan oleh seksi dalam membuat surat dengan menjalankan fungsi ‘no\_surat’ pada kelas ‘modelSuratKeluar’ yang merupakan fitur memberi nomor surat (UC-10). Potongan kode program dalam Kode 5.3 akan dijalankan kembali apabila kolom isian pada *view* ‘detailSuratKeluar’ tidak lengkap. Fungsi ‘no\_surat’ dalam kelas ‘modelSuratKeluar’ seperti dalam Kode 5.4 ini akan melakukan pengecekan terhadap nomor surat yang terakhir digunakan ditambah 1 sehingga nomor surat yang digunakan pasti akan berurutan.

**Kembali**

Nomor Surat : 4

Kepada :  \*)

Perihal :  \*)

Dari agenda :

Dengan agenda :

Lampiran :

Catatan :

Arsip :  No files selected. \*)

\*)Harus diisi

**Gambar 5.3** Tampilan View ‘detailSuratKeluar’

```

..    ...
.
45    <tr>
46    <td><?php echo form_label('Nomor Surat'); ?></td>
47    <td> : </td>
48    <td>
49    <?php
50    echo form_label($nomor_surat);
51    echo form_hidden('nomor_surat', $nomor_surat);
52    ?>
53    </td>
54    </tr>
55    <tr>
56    <td><?php echo form_label('Kepada'); ?></td>
57    <td> : </td>
58    <td><?php echo form_input('tertuju', ''); echo "*"");
59    ?></td>
60    </tr>
..    ...
.

```

**Kode 5.5** Cuplikan Kode Program View ‘detailSuratKeluar’

Setelah nomor surat dibuat maka nomor surat itu akan dikembalikan pada fungsi ‘tambahSuratKeluar’ untuk kemudian ditampilkan pada view ‘detailSuratKeluar’ seperti dalam Gambar 5.3 dan dengan cuplikan kode program dalam Kode 5.5. Pada view ini juga akan ditampilkan kolom isian untuk diisi oleh seksi terkait detail surat keluar yang dibuat lalu seksi dapat mengunggah hasil scan arsip surat (UC-09) yang telah dibuat lalu klik ‘Simpan’. Seperti dalam Kode 5.5,

kolom isian detail surat keluar menggunakan *form\_input* kecuali untuk kolom fitur UC-09 unggah hasil *scan* arsip surat. Fitur mengunggah hasil *scan* arsip surat (UC-09) ini dibuat agar pengguna dapat menyimpan arsip surat secara digital dalam *database* sehingga lebih mudah dalam menemukan dan efisien tempat penyimpanan. Setiap *file* hasil *scan* arsip surat diunggah maka *file* akan disimpan dalam *storage* penyimpanan pada komputer *server* sedangkan dalam *database* disimpan alamat lokasi penyimpanan.

Setelah pengguna mengklik tombol simpan maka detail surat itu akan dikirimkan pada fungsi 'tambahSuratKeluar' lagi untuk memberi nomor register surat (UC-06) dengan menjalankan fungsi 'reg' pada kelas 'modelSurat' seperti dalam Kode 5.6, membuat folder untuk menyimpan file arsip surat, menyimpan file arsip surat dalam folder, lalu menyimpan seluruh detail surat dalam 1 variabel *array* dengan proses seperti dalam Kode 5.7. Pada fungsi 'reg' awalnya dilakukan pengecekan nomor register terakhir dalam *database* digunakan untuk data pada tabel 'surat\_masuk' ataukah 'surat\_keluar'. Setelah nomor register terakhir digunakan telah ditemukan maka nomor register tersebut dijadikan acuan untuk membuat nomor register berikutnya. Nomor register dibuat dengan format *yyyymmddxxx* yang telah dijelaskan pada ketentuan nomor register surat pada rancangan *use case* diagram. Nilai *xxx* itulah yang akan dilanjutkan dari nomor register sebelumnya sedangkan nilai *yyyymmdd* diperoleh dari tahun, bulan dan tanggal ketika surat akan disimpan dalam *database*. Setelah nomor register dibuat oleh fungsi 'reg' dalam kelas 'modelSurat' maka fungsi akan mengirimkan kembali nomor register surat yang telah dibuat ke fungsi atau fitur yang menjalankannya.

```
... ..
50 else if($query->row()->compare == 0)
51 {
    $result1 = $this->db->query("SELECT CONCAT( EXTRACT( YEAR
    FROM (SELECT tgl_terima FROM surat_masuk order by
    tgl_terima DESC limit 1) ), EXTRACT( MONTH FROM (SELECT
52 tgl_terima FROM surat_masuk order by tgl_terima DESC limit
    1) ), EXTRACT( DAY FROM (SELECT tgl_terima FROM
    surat_masuk order by tgl_terima DESC limit 1) ) ) as reg
    ");
53 $find = $result1->row()->reg;
54 $result2 = $this->db->query('SELECT reg FROM surat order
    by id surat DESC limit 1');
```

```

55  $reg = $result2->row()->reg;
56  $no_reg = str_replace($find,"",$reg);
57  $no_reg = intval($no_reg);
58  $no_reg = $no_reg + 1;
59  $no_reg = strval($no_reg);
60  }
61  $result = $this->db->query("SELECT CONCAT( EXTRACT( YEAR
FROM CURDATE( ) ) , EXTRACT( MONTH FROM CURDATE( ) ) ,
EXTRACT( DAY FROM CURDATE( ) ) , '". $no_reg.'"') as reg ");
62  $temp = $result->row();
63  $data = $temp->reg;
64  return strval($data);
...  ...

```

#### Kode 5. 6 Cuplikan Kode Program Fungsi ‘reg’ Kelas ‘modelSurat’

Setelah nomor register dibuat, file arsip surat telah disimpan dalam folder dan seluruh detail surat telah disimpan dalam 1 variabel maka variabel itu dikirimkan pada fungsi ‘suratKeluar’ dalam kelas ‘modelSuratKeluar’. Fungsi ‘suratKeluar’ tersebut digunakan untuk menyimpan data detail surat keluar ke dalam *database*. Berikut dalam Kode 5.8 adalah cuplikan program fungsi ‘suratKeluar’. Pada fungsi tersebut terdapat 2 proses penyimpanan, yaitu penyimpanan detail surat ke dalam tabel ‘surat’ dan penyimpanan detail surat ke dalam tabel ‘surat\_keluar’. Terdapat 2 proses dalam menyimpan detail surat karena tabel ‘surat’ pada *database* ‘bri\_db’ menyimpan data umum surat, sedangkan pada tabel ‘surat\_keluar’ menyimpan detail surat secara spesifik hanya milik surat keluar. Setelah tersimpan maka aplikasi akan menjalankan fungsi ‘viewSuratKeluar’ kembali untuk menampilkan surat keluar.

```

...  ...
101  $this->load->model('modelsibri');
102  $dataSurat['reg'] = $this->modelsibri->reg();
103  $dataSurat['no_surat'] = $this->input->post('nomor_surat');
104  $dataSurat['pengirim'] = $user;
105  $dataSurat['tertuju'] = $this->input->post('tertuju');
106  $dataSurat['perihal'] = $this->input->post('perihal');
107  $dataSurat['dari_agenda'] = $this->input-
>post('dari_agenda');
108  $dataSurat['dengan_agenda'] = $this->input-
>post('dengan_agenda');
109  $lampiran = $this->input->post('lampiran');

```

```

110 $lampiran = intval($lampiran);
111 $dataSurat['lampiran'] = $lampiran;
112 $dataSurat['catatan'] = $this->input->post('catatan');
113 //membuat folder location_file
114 $file_upload = "";
115 $user = $this->modelsibri->getDataUser($user);
116 $user = $user['pengguna'][0]->bidang;
117 $no_surat = strval($dataSurat['no_surat']);
118 mkdir("./uploads/".$user."/Surat Keluar/".$no_surat."/");
119 $path = "./uploads/".$user."/Surat Keluar/".$no_surat."/";
120 //menyimpan file ke folder location_file
121 $i=0;$a=1;
122 foreach($_FILES['upload']['tmp_name'] as $tmp)
123 { move_uploaded_file($tmp,
124 $path.$_FILES['upload']['name'][$i]);
125 $extension =
126 pathinfo($_FILES['upload']['name'][$i],PATHINFO_EXTENSION);
127 rename($path.$_FILES['upload']['name'][$i],
128 $path.$a."".$extension);
129 $file_upload = $file_upload.$path.$a."".$extension."*";
130 $i++;$a++;}
131 $dataSurat['no_surat'] = intval($dataSurat['no_surat']);
132 $dataSurat['location_file'] = $file_upload;
133 $this->modelsibri->suratKeluar($dataSurat);
134 $this->viewSuratKeluar();
135 ...

```

**Kode 5. 7 Cuplikan Kode Program Fungsi ‘tambahSuratKeluar’ Kelas ‘controllerSuratKeluar’**

```

...
19 public function suratKeluar($dataSurat)
20 {
21 $query1 = "INSERT INTO surat
22 (reg,id_pengguna,perihal,dari_agenda,dengan_agenda,lampiran,catatan,location_file) VALUES(?, ?, ?, ?, ?, ?, ?, ?)";
23 $result = $this->db->query($query1,
24 array(
25 $dataSurat['reg'],
26 $dataSurat['pengirim'],
27 $dataSurat['perihal'],
28 $dataSurat['dari_agenda'],
29 $dataSurat['dengan_agenda'],

```

```
29  $dataSurat['lampiran'],
30  $dataSurat['catatan'],
31  $dataSurat['location_file']
32  )
33  ) or die("Cannot Insert Data Into user");
34
35  $query2 = $this->db->query('SELECT id_surat FROM surat
36  order by id_surat DESC limit 1');
37  $id_surat = $query2->result();
38
39  $query3 = "INSERT INTO surat_keluar
40  (surat_id_surat,no_surat,tertuju) VALUES(?, ?, ?)";
41  $result = $this->db->query($query3,
42  array(
43  intval($id_surat[0]->id_surat),
44  $dataSurat['no_surat'],
45  $dataSurat['tertuju']
46  )
47  ) or die("Cannot Insert Data Into user");
48  }
49  ...
50  ...
```

**Kode 5. 8 Cuplikan Kode Program Fungsi ‘suratKeluar’ Kelas  
‘modelSuratKeluar’**

## 5.2. Pengujian

Pada tahap ini akan dilakukan pengujian terhadap aplikasi yang telah dibuat menggunakan metode pengujian integrasi objek atau pengujian *release* (Sommerville, 2007) yang merupakan salah satu tipe pengujian sistem. Pengujian sistem digunakan pada aplikasi yang mengintegrasikan dua atau lebih komponen yang melaksanakan fungsi-fungsi atau fitur sistem. Pengujian sistem dibagi menjadi *integration testing* (pengujian integrasi) dan *release testing* (pengujian integrasi objek).

Untuk aplikasi berorientasi objek sangat sulit untuk menerapkan pengujian integrasi yang berbasis *top-down* atau *bottom-up* karena pada aplikasi berorientasi objek tidak ada hierarki objek yang jelas (Sommerville, 2001). Oleh karena itu *release testing* (pengujian integrasi objek) yang merupakan pengujian kotak hitam lebih tepat digunakan, pengujian ini dilakukan dengan pendekatan berdasarkan operasi atau fitur dalam sistem atau aplikasi. Pengujian ini memiliki tujuan untuk

menjamin bahwa aplikasi yang dibuat telah sesuai dengan kebutuhan BRI Kanwil Malang dan prosedur baru yang dibuat dengan adanya aplikasi tidak mengubah isi dari arsip surat, dengan kata lain isi arsip surat sama saat proses pengarsipan dilakukan dengan prosedur lama karena prosedur baru hanya menyesuaikan pengarsipan dengan adanya aplikasi.

### 5.2.1. Rencana Pengujian (*Test Plan*)

Seperti yang diketahui bahwa pengujian integrasi objek adalah pengujian kotak hitam dengan pendekatan berdasarkan operasi atau fitur dalam sistem atau aplikasi, dengan kata lain pengujian dilakukan pada kemampuan aplikasi dalam mengidentifikasi pengguna dengan fitur *log in* dan fitur-fitur yang terdapat dalam diagram *use case* seperti yang telah digambarkan dalam Gambar 4.1. Fitur-fitur tersebut diuji dengan segala kemungkinan yang ada untuk menjamin bahwa aplikasi yang dibuat telah sesuai dengan kebutuhan (fungsional dan non-fungsional) dari BRI Kanwil Malang dan prosedur baru yang telah dibuat dengan adanya aplikasi tidak mengubah isi dari arsip surat.

Perangkat keras yang digunakan dalam pengujian sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang adalah:

1. Prosesor Intel Core 2 Duo @2,4 GHz,
2. 2 GB DDR2 RAM,
3. Ruang penyimpanan 320 GB.

Sedangkan perangkat lunak yang digunakan dalam pengembangan sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang adalah:

1. Windows 8.1,
2. Xampp 1.8.3,
3. Code Igniter 2.1.4,
4. Google Chrome 34.0.1847.137.

### 5.2.2. Skenario Pengujian (*Test Scenario*)

Skenario pengujian disesuaikan dengan masing-masing fitur yang akan diuji. Fitur-fitur yang akan diuji adalah identifikasi pengguna (*log in*), melihat surat masuk (UC-01), melihat surat keluar (UC-02), pencarian (UC-03), pengurutan

(UC-04), tambah surat masuk (UC-05), memberi nomor register surat (UC-06), terima surat masuk (UC-07), unggah hasil *scan* arsip surat (UC-08), membuat surat keluar (UC-09), memberi nomor surat (UC-10), dan membuat surat balasan (UC-11). Tidak dilakukan pengujian pada fitur perbaiki data umum surat masuk (UC-12), perbaiki detail surat masuk (UC-13) dan fitur perbaiki detail surat keluar (UC-14) karena fitur-fitur tersebut identik dengan beberapa fitur lain. Fitur UC-12 sama dengan fitur UC-05, fitur UC-13 sama dengan fitur UC-07 dan fitur UC-14 sama dengan fitur UC-09.

Pada masing-masing skenario pengujian memiliki tujuan dan kasus pengujian yang berbeda, disesuaikan dengan fitur apa saja yang akan diuji dalam skenario tersebut. Dalam skenario pengujian suatu kasus uji dianggap sukses apabila hasil yang sebenarnya dari pengujian sesuai dengan hasil yang diharapkan. Setiap skenario pengujian bisa mencakup beberapa fitur untuk diuji, karena ketika menjalankan suatu fitur maka beberapa fitur lain dijalankan juga. Dalam Tabel 5.1 adalah skenario pengujian 7 yang melakukan pengujian terhadap fitur membuat surat balasan (UC-11) dan juga menjalankan fitur memberi nomor register surat (UC-06), memberi nomor surat (UC-10) dan unggah hasil *scan* arsip surat (UC-08).

Skenario pengujian untuk fitur lainnya terdapat pada halaman Lampiran. Pengujian yang dilakukan untuk fitur identifikasi pengguna (*log in*), melihat surat masuk (UC-01) dan melihat surat keluar (UC-02) dapat dilihat dalam Lampiran 4 Tabel L.5.1. Pengujian untuk fitur melihat surat masuk (UC-01), melihat surat keluar (UC-02) dan pencarian (UC-03) dalam Lampiran 4 Tabel L.5.2. Pengujian untuk fitur melihat surat masuk (UC-01), melihat surat keluar (UC-02) dan pencarian (UC-04) dalam Lampiran 4 Tabel L.5.3. Pengujian untuk fitur tambah surat masuk (UC-05) dan memberi nomor register surat (UC-06) ada dalam Lampiran 4 Tabel L.5.4. Pengujian fitur terima surat masuk (UC-07) dan unggah hasil *scan* arsip surat (UC-08) dapat dilihat dalam Lampiran 4 Tabel L.5.5. Pengujian yang dilakukan untuk fitur membuat surat keluar (UC-09), memberi nomor register surat (UC-06), memberi nomor surat (UC-10) dan unggah hasil *scan* arsip surat (UC-08) dapat dilihat dalam Lampiran 4 Tabel L.5.6.

**Tabel 5. 1 Skenario Pengujian 7**

|                       |   |
|-----------------------|---|
| ID Pengujian          | SP-07   |
| Tujuan                | <ol style="list-style-type: none"> <li>1. Membuktikan fitur membuat surat balasan (UC-11) dapat dijalankan.</li> <li>2. Membuktikan bahwa fitur memberi nomor surat (UC-10) dapat dijalankan nomor surat yang dibuat benar.</li> <li>3. Membuktikan bahwa fitur unggah hasil <i>scan</i> arsip surat (UC-08) dapat dijalankan dan <i>file</i> hasil <i>scan</i> tersimpan dalam <i>database</i> tanpa mengubah isi surat.</li> <li>4. Membuktikan bahwa ketika detail surat balasan berhasil disimpan maka aplikasi memberi nomor register surat (UC-06) pada surat tersebut.</li> <li>5. Nomor register surat yang dibuat sesuai dengan aturan yang telah dibuat pada proses perancangan.</li> </ol>   |
| Skenario              | Aplikasi diakses menggunakan ID Pengguna dan <i>Password</i> milik seksi5. Pada pengujian fitur ini dilakukan proses memasukkan detail surat balasan ke dalam aplikasi dengan cara mengisi kolom isian detail surat balasan dan mengunggah <i>file</i> hasil <i>scan</i> arsip surat balasan. Lalu klik tombol ‘Simpan’.  |
| Kasus pengujian       | <p>KP-07-1. Membiarkan kosong kolom isian detail surat balasan yang harus diisi.<br/>                     Kepada : kosong (harus diisi),<br/>                     Perihal : kosong (harus diisi),<br/>                     Dari agenda : kosong,<br/>                     Dengan agenda : kosong,<br/>                     Lampiran : kosong,<br/>                     Catatan : kosong,<br/>                     Arsip dibiarkan kosong (harus diisi).</p> <p>KP-07-2. Mengisi seluruh kolom isian detail surat balasan yang harus diisi.<br/>                     Kepada : tesPengirimcoba (harus diisi),<br/>                     Perihal : tesPerihal (harus diisi),<br/>                     Dari agenda : kosong,<br/>                     Dengan agenda : kosong,<br/>                     Lampiran : kosong,<br/>                     Catatan : kosong,<br/>                     Arsip diisi dengan <i>file</i> bernama ‘surat1.jpg’ (harus diisi).</p> |
| <i>Pre-conditions</i> | <ol style="list-style-type: none"> <li>1. Masuk ke dalam aplikasi menggunakan hak akses seksi5.</li> <li>2. Klik tombol ‘Surat Masuk’ untuk menjalankan fitur melihat surat masuk (UC-01).</li> </ol>   |

|                        |   |
|------------------------|---|
|                        | 3. Klik tombol 'Balas' pada data surat masuk yang akan dibalas untuk menjalankan fitur membuat surat balasan (UC-11).   |
| Hasil yang diharapkan  | <p>KP-07-1.</p> <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat balasan merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat balasan tidak dapat disimpan dan terdapat peringatan berupa "Arsip belum terisi atau kolom kepada dan perihal belum terisi".</li> </ol> <p>KP-07-2.</p> <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat balasan merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat balasan berhasil tersimpan.</li> <li>3) <i>File</i> arsip surat balasan dapat ditampilkan.</li> <li>4) Data arsip surat keluar tersebut tampil pada barisan paling atas dengan nomor register sama dengan nomor register surat masuk yang dibalas.</li> </ol> |
| Hasil yang sebenarnya  | <p>KP-07-1.</p> <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat balasan merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat balasan tidak dapat disimpan dan terdapat peringatan berupa "Arsip belum terisi atau kolom kepada dan perihal belum terisi".</li> </ol> <p>KP-07-2.</p> <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat balasan merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat balasan berhasil tersimpan.</li> <li>3) <i>File</i> arsip surat balasan dapat ditampilkan.</li> <li>4) Data arsip surat keluar tersebut tampil pada barisan paling atas dengan nomor register sama dengan nomor register surat masuk yang dibalas.</li> </ol> |
| <i>Post conditions</i> | Apabila kolom isian yang harus diisi sudah diisi maka ketika tombol 'Simpan' diklik aplikasi akan menampilkan <i>view</i> 'viewSuratMasuk'  |

### 5.2.3. Analisis Pengujian

Aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang diharapkan menjadi jawaban dari permasalahan pengarsipan surat yang terjadi di BRI Kanwil Malang. Permasalahan pengarsipan surat yang terjadi di BRI Kanwil Malang adalah membutuhkan ruang besar untuk menyimpan arsip surat, terjadi

pemborosan kertas karena arsip surat asli dipegang oleh sekretaris dan fotokopi arsip surat dipegang oleh bagian seksi, sekretaris dan seksi mengalami kesulitan untuk menemukan arsip surat, dan terjadi kesalahan penomoran surat (nomor surat atau nomor register).

Hasil dari skenario pengujian yang telah dilakukan menunjukkan bahwa kasus pengujian yang diberikan pada aplikasi 100% sukses diselesaikan seperti yang ditunjukkan dalam Tabel 5.2. Seluruh fitur yang dimiliki oleh aplikasi secara penuh bisa dijalankan oleh aplikasi yang mana seluruh fitur tersebut adalah hasil dari penyimpulan kebutuhan Bank Rakyat Indonesia Kantor Wilayah Malang yang sebelumnya telah dijelaskan pada tahap perancangan. Isi dari arsip surat tidak berubah karena selain detail surat dimasukkan oleh pengguna ke dalam *database*, aplikasi juga menyimpan *file* hasil *scan* arsip surat dalam *database* dan dapat dilihat oleh pengguna ketika menjalankan fitur melihat surat masuk (UC-01) atau melihat surat keluar (UC-02). Dengan memperhatikan hasil Pengujian dapat disimpulkan kelebihan dan kekurangan aplikasi ini. Aplikasi ini memiliki kelebihan sebagai berikut:

1. Aplikasi menyimpan arsip surat dalam *database* tanpa menggunakan kertas, sehingga tidak membutuhkan ruang besar untuk menyimpan arsip surat dan tidak perlu secara rutin menyediakan ruangan baru untuk menyimpan arsip surat fisik. Karena arsip surat disimpan dalam *database* maka memungkinkan sekretaris dan seksi melihat surat melalui aplikasi (UC-01 dan UC-02), hal ini dibuktikan pada skenario pengujian 1 dalam Lampiran 4 Tabel L.5.1.
2. Tidak lagi terjadi pemborosan kertas untuk keperluan arsip, karena aplikasi menyimpan arsip surat secara digital (UC-05, UC-07, UC-09, dan UC-11) dan dalam bentuk *file scan* arsip surat yaitu fitur (UC-08). Solusi ini dibuktikan pada skenario pengujian 4 dalam Lampiran 4 Tabel L.5.4, skenario pengujian 5 dalam Lampiran 4 Tabel L.5.5, skenario pengujian 6 dalam Lampiran 4 Tabel L.5.6 dan skenario pengujian 7 dalam Tabel 5.1.
3. Sekretaris atau seksi tidak lagi kesulitan dalam menemukan arsip surat karena terdapat fitur pencarian (UC-03) dan pengurutan (UC-04), yang dibuktikan dengan skenario pengujian 2 dalam Lampiran 4 Tabel L.5.2 dan skenario pengujian 3 dalam Lampiran 4 Tabel L.5.3.

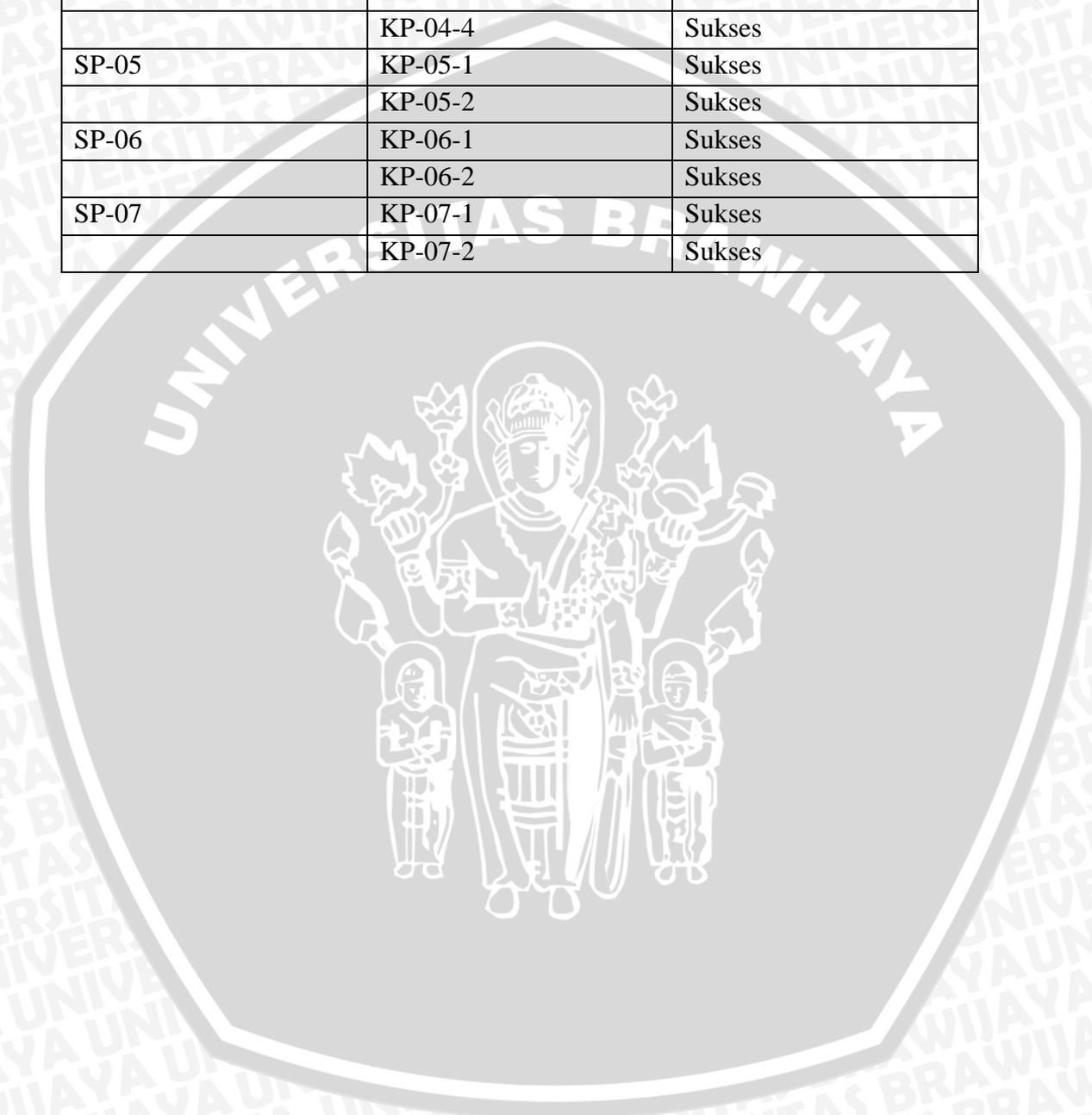
4. Sistem penomoran surat (UC-10) dan nomor register surat (UC-06) secara otomatis dibuat oleh aplikasi sehingga mengurangi kemungkinan terjadinya kesalahan dalam melakukan penomoran, fitur ini dibuktikan pada skenario pengujian 4 dalam Lampiran 4 Tabel L.5.4, skenario pengujian 6 dalam Lampiran 4 Tabel L.5.6, dan skenario pengujian 7 dalam Tabel 5.1.
5. Penomoran surat untuk surat keluar dan surat balasan (UC-10) atau penomoran register surat (UC-06) telah sesuai dengan aturan milik BRI Kanwil Malang, hal ini dibuktikan pada skenario pengujian 4 dalam Lampiran 4 Tabel L.5.4, skenario pengujian 6 dalam Lampiran 4 Tabel L.5.6, dan skenario pengujian 7 dalam Tabel 5.1.

Sedangkan kekurangan dari aplikasi ini adalah aplikasi ini sangat bergantung pada ketersediaan *database* karena apabila terjadi masalah pada *database* maka sistem surat menyurat menggunakan aplikasi ini tidak bisa dilakukan seperti yang terjadi pada skenario pengujian 1 dalam Lampiran 4 Tabel L.5.1. Secara keseluruhan aplikasi sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang ini telah dapat menyelesaikan permasalahan pengarsipan surat yang terjadi di BRI Kanwil Malang tersebut.

**Tabel 5. 2 Rekapitulasi Hasil Skenario Pengujian Aplikasi**

| Skenario Pengujian | Kasus Pengujian | Status |
|--------------------|-----------------|--------|
| SP-01              | KP-01-1         | Sukses |
|                    | KP-01-2         | Sukses |
|                    | KP-01-3         | Sukses |
|                    | KP-01-4         | Sukses |
|                    | KP-01-5         | Sukses |
|                    | KP-01-6         | Sukses |
|                    | KP-01-7         | Sukses |
|                    | KP-01-8         | Sukses |
|                    | KP-01-9         | Sukses |
|                    | KP-01-10        | Sukses |
| SP-02              | KP-02-1         | Sukses |
|                    | KP-02-2         | Sukses |
|                    | KP-02-3         | Sukses |
|                    | KP-02-4         | Sukses |
| SP-03              | KP-03-1         | Sukses |
|                    | KP-03-2         | Sukses |

|       |         |        |
|-------|---------|--------|
|       | KP-03-3 | Sukses |
|       | KP-03-4 | Sukses |
| SP-04 | KP-04-1 | Sukses |
|       | KP-04-2 | Sukses |
|       | KP-04-3 | Sukses |
|       | KP-04-4 | Sukses |
| SP-05 | KP-05-1 | Sukses |
|       | KP-05-2 | Sukses |
| SP-06 | KP-06-1 | Sukses |
|       | KP-06-2 | Sukses |
| SP-07 | KP-07-1 | Sukses |
|       | KP-07-2 | Sukses |



## BAB VI

### PENUTUP

Pada tahap ini akan dijelaskan mengenai kesimpulan dan saran terkait sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang (BRI Kanwil Malang). Kesimpulan yang diambil merupakan pernyataan untuk menjawab apakah sistem informasi pengarsipan surat BRI Kanwil Malang menyelesaikan permasalahan pengarsipan surat yang terjadi di BRI Kanwil Malang atau tidak sesuai dengan rumusan masalah penelitian. Selain kesimpulan, disertakan pula saran untuk kepentingan pengembangan aplikasi.

#### 6.1. Kesimpulan

1. Aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang dapat mengatasi permasalahan pengarsipan surat di BRI Kanwil Malang dengan menyimpan arsip surat dalam *database* dan pengguna (sekretaris dan seksi) dapat menggunakan fitur-fitur dalam aplikasi yang dibuat sesuai kebutuhan BRI Kanwil Malang. Fitur-fitur tersebut adalah melihat surat masuk (UC-01), melihat surat keluar (UC-02), pencarian (UC-03), pengurutan (UC-04), tambah surat masuk (UC-05), memberi nomor register surat (UC-06), terima surat masuk (UC-07), unggah hasil *scan* arsip surat (UC-08), membuat surat keluar (UC-09), memberi nomor surat (UC-10), membuat surat balasan (UC-11), perbaiki data umum surat masuk (UC-12), perbaiki detail surat masuk (UC-13) dan fitur perbaiki detail surat keluar (UC-14). Kemudian dari hasil pengujian menunjukkan bahwa fitur-fitur tersebut sukses menjalankan fungsionalitasnya, sehingga dianggap aplikasi ini dapat mengatasi permasalahan pengarsipan surat di BRI Kanwil Malang dan dari prosedur pengarsipan surat yang baru setelah terdapat aplikasi menunjukkan peningkatan efisiensi.
2. Prosedur baru pengarsipan surat di BRI Kanwil Malang lebih efisien dengan adanya sistem informasi pengarsipan surat BRI Kanwil Malang yaitu seperti yang telah dirancang dalam Gambar 4.2, Gambar 4.3, Gambar 4.4, Gambar 4.5, Gambar 4.6 dan dalam Lampiran 1 Gambar L.1.1 serta Gambar L.1.2. Prosedur

pengarsipan surat yang baru diadopsi dari prosedur pengarsipan surat yang lama. Peningkatan efisiensi prosedur pengarsipan surat baru dari prosedur pengarsipan surat lama dapat dilihat dalam Lampiran 2. Kemudian dapat dilihat dalam Lampiran 2 tersebut bahwa prosedur pengarsipan surat baru merupakan prosedur lama dengan menghilangkan beberapa aktivitas akibat terdapat beberapa aktivitas yang telah dilakukan secara otomatis oleh aplikasi, sehingga sekretaris maupun seksi tidak perlu melakukan aktivitas itu lagi.

## 6.2. Saran

Berdasarkan skenario pengujian 1 dalam Lampiran 4 Tabel L.5.1 menunjukkan bahwa aplikasi sangat bergantung atas koneksi dan ketersediaan *database*, maka kerusakan *database* adalah ancaman terbesar untuk aplikasi sistem pengarsipan surat di BRI Kanwil Malang. Oleh karena itu Tim IT dari BRI Kanwil Malang harus selalu mengawasi keadaan atau kondisi *server database* serta selalu melakukan *backup* terhadap data arsip surat. Sehingga aplikasi dapat selalu digunakan dan tidak perlu khawatir arsip surat hilang.

Selain ancaman *database* rusak, aplikasi ini memiliki keterbatasan jumlah karakter yang dapat disimpan dalam *database*. Kebanyakan dari informasi yang tersimpan dalam aplikasi ini tersimpan dalam bentuk teks, meskipun tipe data dalam *database* yang digunakan adalah MEDIUMTEXT dengan jumlah karakter maksimum 16.777.215 namun tetap saja masih ada kemungkinan informasi tidak tersimpan secara penuh apabila teks yang disimpan melebihi kapasitas maksimum karakter. Maka dalam pengembangan aplikasi ini perlu dipertimbangkan cara untuk menyimpan informasi arsip surat tanpa ada kekhawatiran informasi tersebut tidak tersimpan dengan penuh.

Setelah dalam jangka waktu tertentu dalam penggunaan aplikasi sistem informasi pengarsipan surat BRI Kanwil Malang ini maka perlu dilakukan evaluasi. Evaluasi ini secara rutin harus dilakukan untuk mengetahui apakah terdapat permasalahan pengarsipan surat yang belum teratasi dengan adanya aplikasi ini, atau mungkin permasalahan baru yang muncul. Selain itu evaluasi baik dilakukan secara rutin setelah jangka waktu tertentu untuk menjamin semua fitur dalam aplikasi masih bekerja sesuai dengan kebutuhan BRI Kanwil Malang.

## DAFTAR PUSTAKA

Cabinet Office. 2001. *Government Data Standards Catalogue Volume 2 – Data Types Standards*. Mei. Inggris.

Editor. 2014. *Data Type Storage Requirements*.

<http://dev.mysql.com/doc/refman/5.7/en/storage-requirements.html>. 16 Juni 2014.

Editor. 2013. *Sistem Informasi Disposisi Surat*.

[http://cabikhosting.com/file/disposisi\\_sistem\\_informasi\\_disposisi.pdf](http://cabikhosting.com/file/disposisi_sistem_informasi_disposisi.pdf). 19 Juni 2013.

Ikhsan. 2010. *e-Filing: Tata laksana dokumen elektronik (tnde) dan persuratan*.

<http://www.digital-sense.net/e-filing-sistem-pengarsipan>. 19 Juni 2013.

Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Andi. Yogyakarta.

Pressman, Roger S. 2001. *Software Engineering: a Practitioner's Approach*. Edisi Kelima. Mc. GrowHill. Los Angeles.

Sommerville, Ian, 2001, *Software Engineering*, Edisi 6, Pereson Education, United Kingdom, Dra. Yuhilza Hanum, M.Eng, 2003, *Software Engineering (Rekayasa Perangkat Lunak)*, Edisi 6/Jilid 2, Erlangga, Jakarta.

Sommerville, Ian. 2007. *Sortware Engineering*. Edisi 8. Pereson Education. United Kingdom.

Tahir, Irmawanti. 2007. *Efektifitas Pelaksanaan Sistem Kearsipan Pada Kantor Badan Kepegawaian Dan Diklat Daerah Kabupaten Takalar*. *Skripsi*. Universitas Hasanuddin. Makassar.

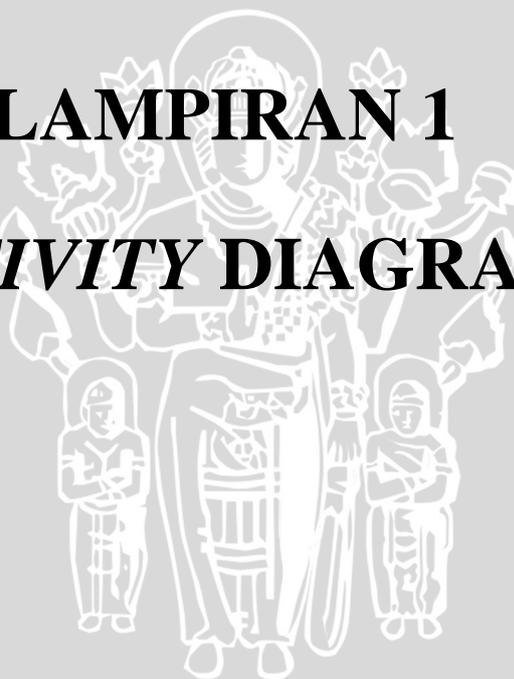


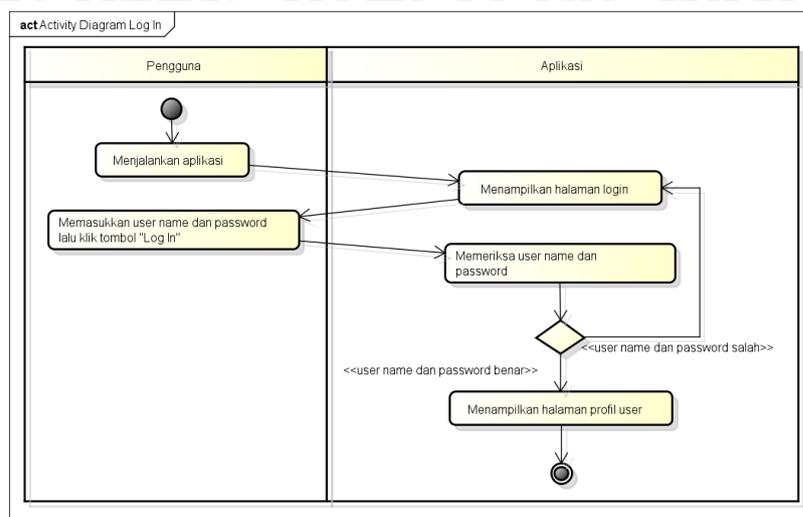
# LAMPIRAN



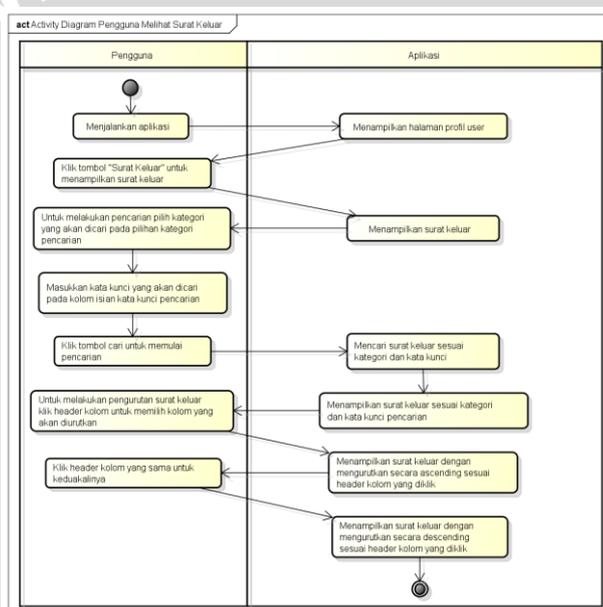
UNIVERSITAS BRAWIJAYA

**LAMPIRAN 1**  
**ACTIVITY DIAGRAM**





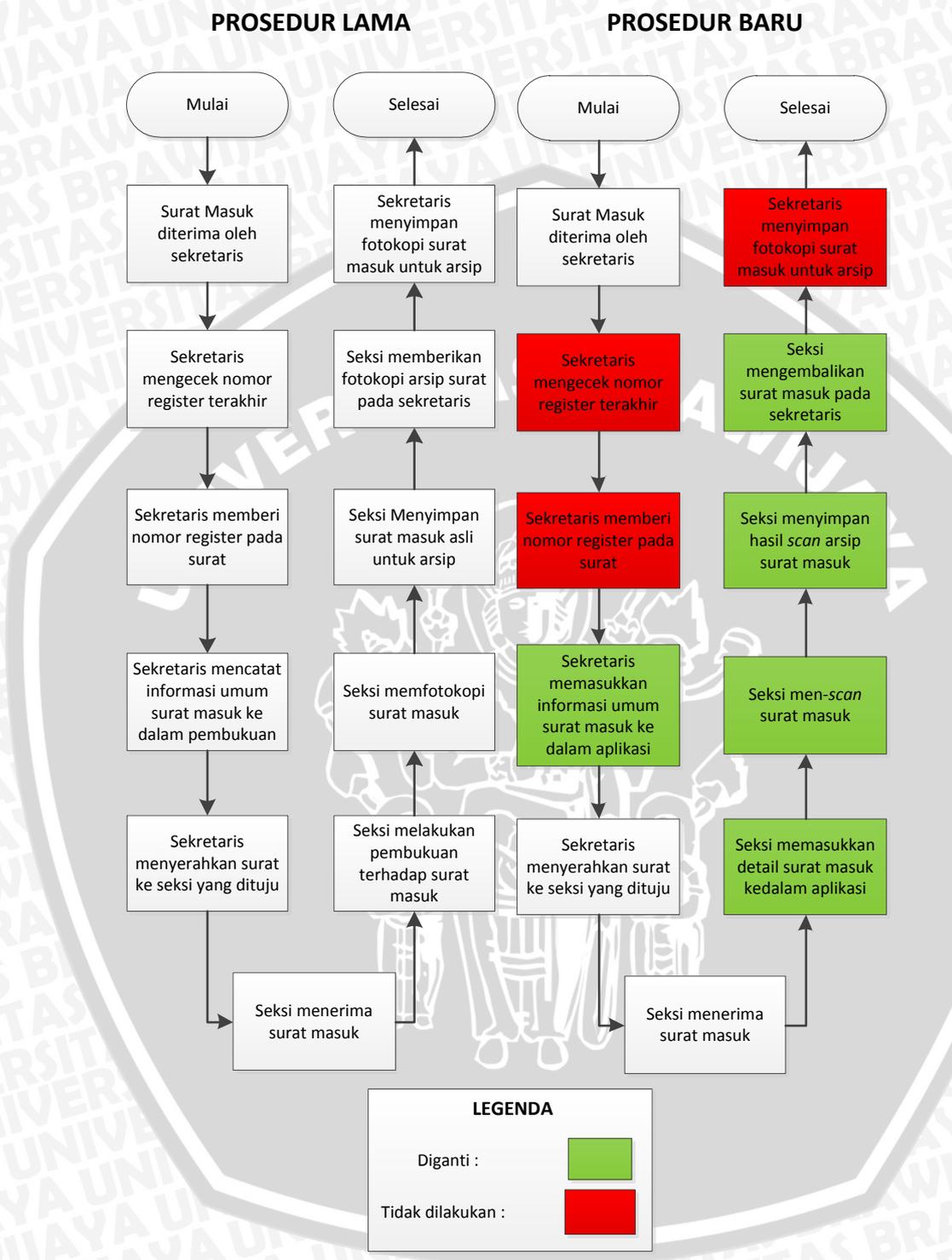
Gambar L.1. 1 Activity Diagram Log In



Gambar L.1. 2 Activity Diagram Pengguna Melihat Surat Keluar



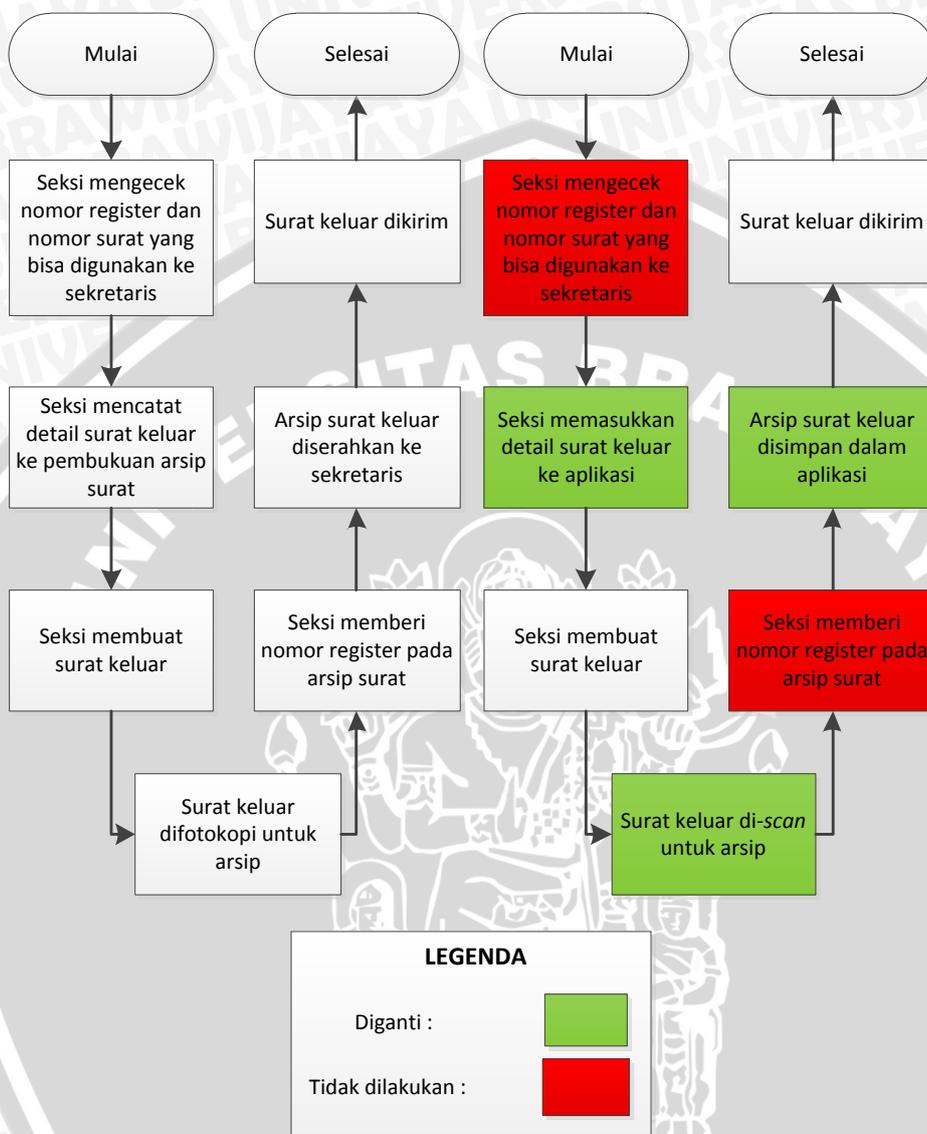
**LAMPIRAN 2**  
**PERBANDINGAN PROSEDUR**  
**PENGARSIPAN SURAT**



**Gambar L.2. 1 Perbandingan Prosedur Lama dan Prosedur Baru Proses Ketika Terdapat Surat Masuk**

**PROSEDUR LAMA**

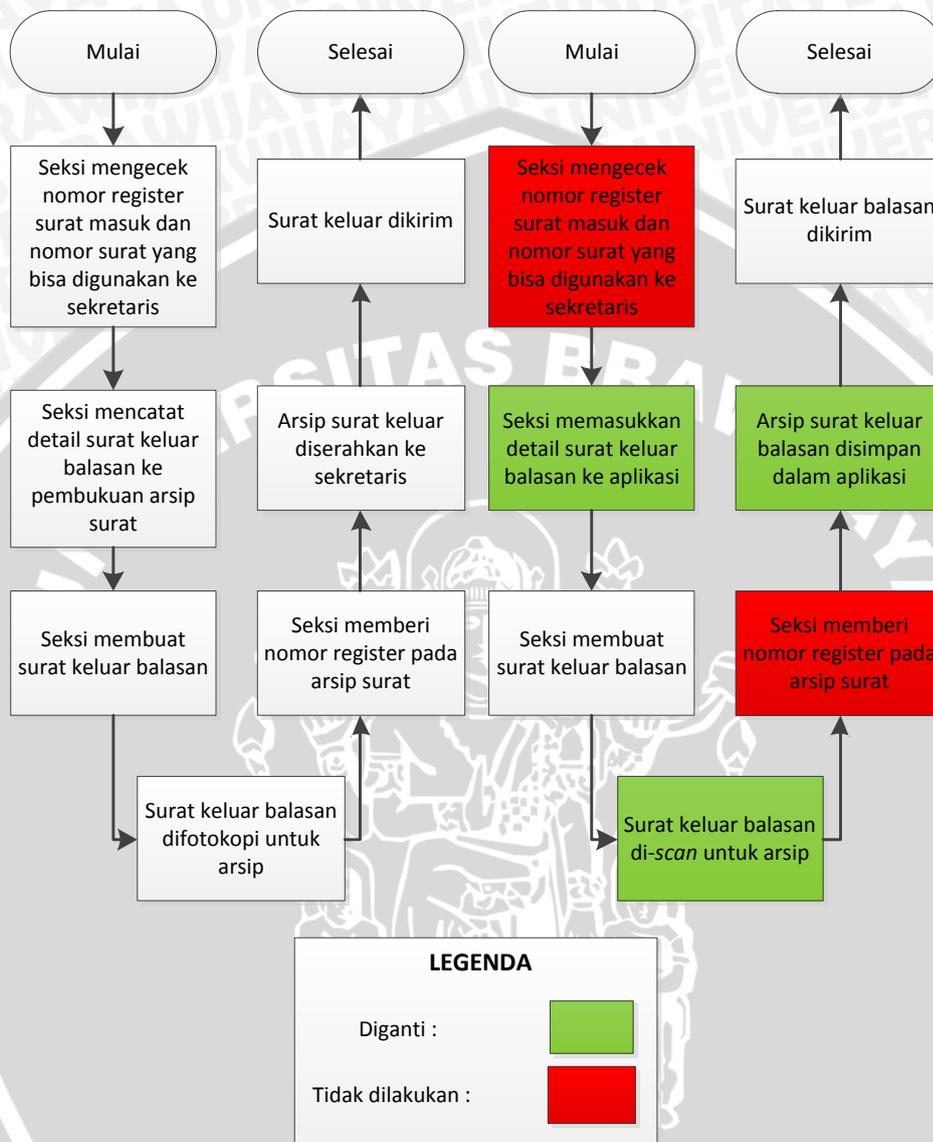
**PROSEDUR BARU**



**Gambar L.2. 2 Perbandingan Prosedur Lama dan Prosedur Baru Proses Seksi Membuat Surat Keluar**

**PROSEDUR LAMA**

**PROSEDUR BARU**



**Gambar L.2. 3 Perbandingan Prosedur Lama dan Prosedur Baru Proses Seksi Membuat Surat Balasan**

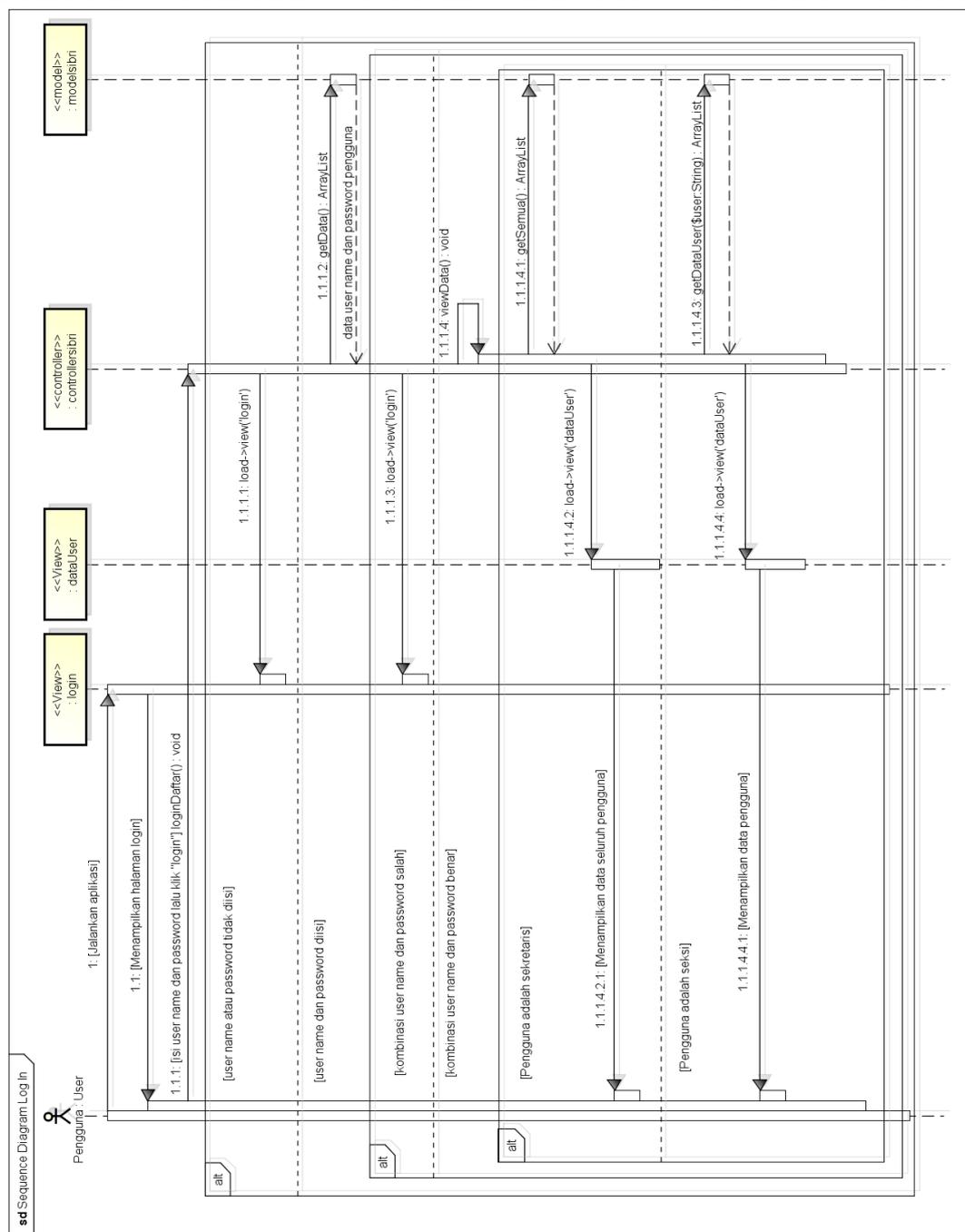


UNIVERSITAS BRAWIJAYA

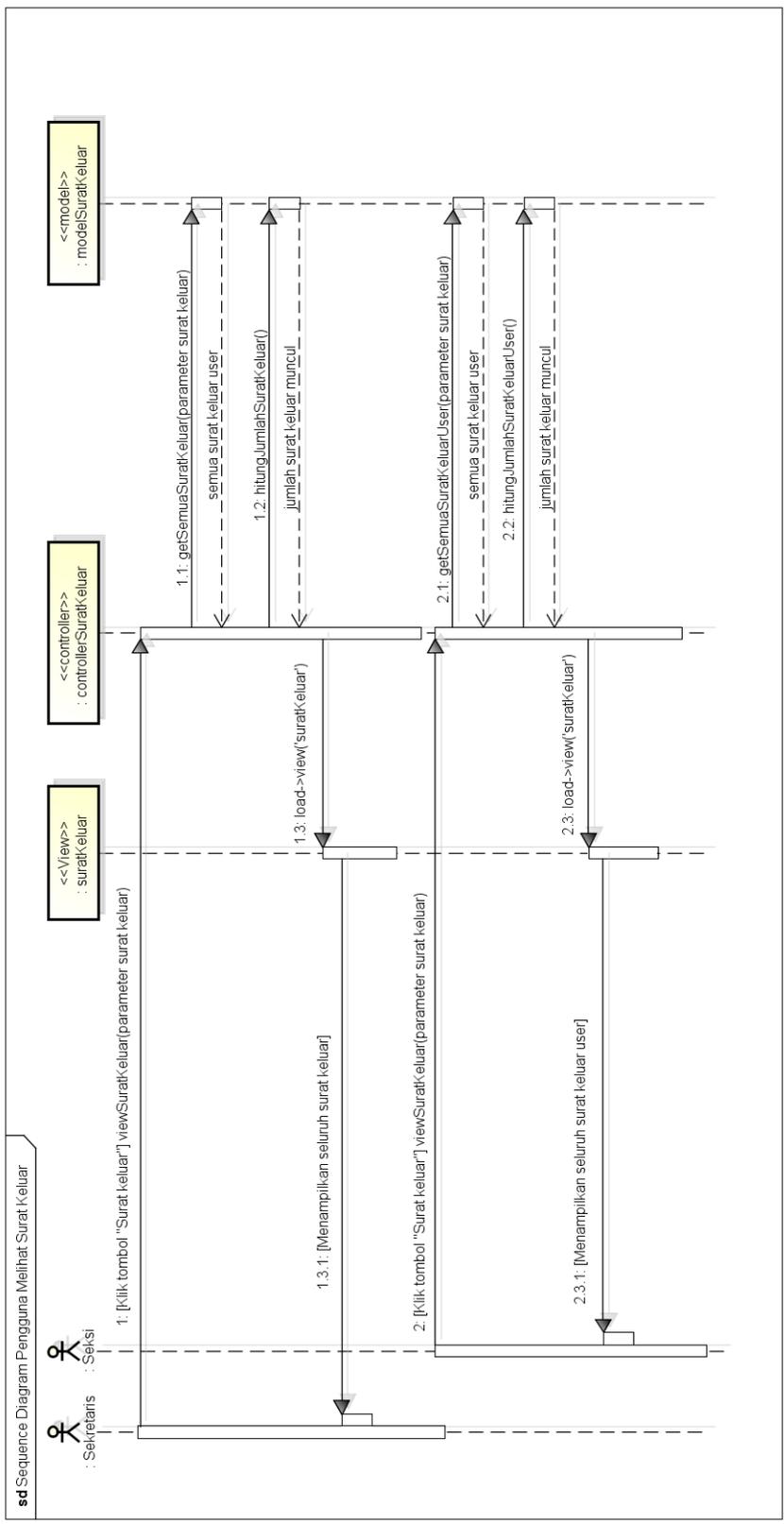
## LAMPIRAN 3

# *SEQUENCE DIAGRAM*



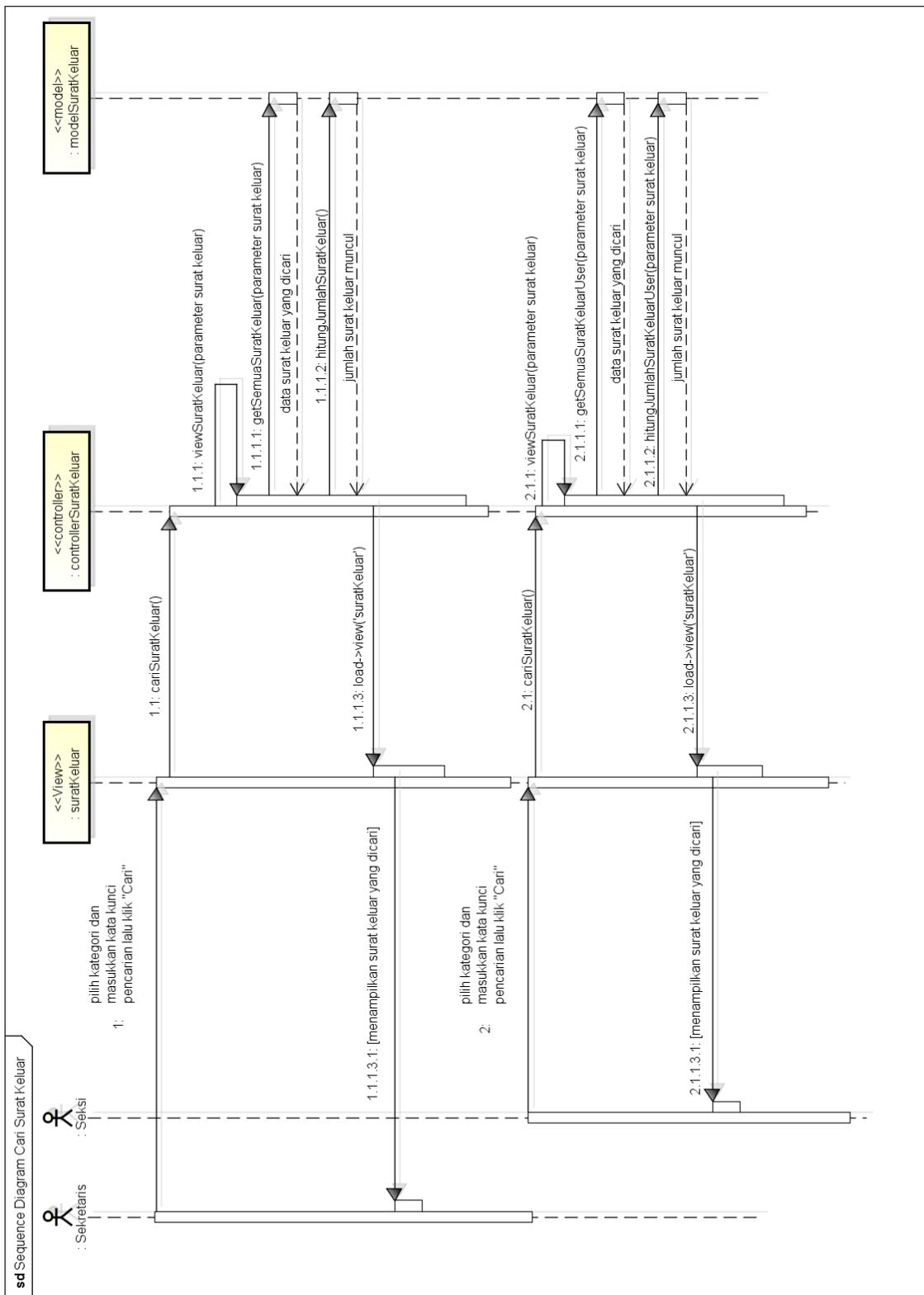


Gambar L.3. 1 Sequence Diagram Log In



powered by Astah

Gambar L.3. 2 Sequence Diagram Pengguna Melihat Surat Keluar



Gambar L.3.3 Sequence Diagram Cari Surat Keluar

UNIVERSITAS BRAWIJAYA

**LAMPIRAN 4**  
**DESAIN TAMPILAN**  
**APLIKASI**



ID Pengguna :

Password :

**Gambar L.4. 1 Desain View ‘login’**

Nama :            contoh nama

ID Pengguna : IDcontoh

| No | Nama | Bidang |
|----|------|--------|
| 1  |      |        |
| 2  |      |        |
| 3  |      |        |
| 4  |      |        |
| 5  |      |        |

**Gambar L.4. 2 Desain View ‘dataUser’ Akses Dilakukan Sekretaris**

Nama :            contoh nama

ID Pengguna : IDcontoh

Bidang :            contoh keterangan

**Gambar L.4. 3 Desain View ‘dataUser’ Akses Dilakukan Seksi**

Nama :

Password :

**Gambar L.4. 4 Desain View ‘perbaruiDataUser’**

Data Pengguna

Surat Masuk

Surat Keluar

Kategori

Kata Kunci

Cari

PENGIRIM

DITUJUKAN PADA

PERIHAL

DARI AGENDA

DENGAN AGENDA

LAMPIRAN

CATATAN

ARSIP

BALASAN

1

2

3

4

5

dan seterusnya

1

2

3

Next

Keluar

Balas

Balas

Balas

Balas

Balas

Balas

Perbaiki / Temna

| No             | REG | TANGGAL SURAT DITERIMA | PENGIRIM | DITUJUKAN PADA | PERIHAL | DARI AGENDA | DENGAN AGENDA | LAMPIRAN | CATATAN | ARSIP | BALASAN |
|----------------|-----|------------------------|----------|----------------|---------|-------------|---------------|----------|---------|-------|---------|
| 1              |     |                        |          |                |         |             |               |          |         | 1     | 2       |
| 2              |     |                        |          |                |         |             |               |          |         |       |         |
| 3              |     |                        |          |                |         |             |               |          |         |       |         |
| 4              |     |                        |          |                |         |             |               |          |         |       |         |
| 5              |     |                        |          |                |         |             |               |          |         |       |         |
| dan seterusnya |     |                        |          |                |         |             |               |          |         |       |         |

Gambar L.4. 5 Desain View ‘suratMasuk’ Akses Dilakukan Seksi

UNIVERSITAS  
**BRAWIJAYA**

UNIVERSITAS  
**BRAWIJAYA**

Data Pengguna

Surat Masuk

Surat Keluar

Kategori

Kata Kunci

Cari

| No             | REG | TANGGAL SURAT DIKIRIM | PENGIRIM | DITUJUKAN PADA | PERIHAL | DARI/AGENDA | DENGAN AGENDA | LAMPIRAN | CATATAN | ARSIP | BALASAN DARI |
|----------------|-----|-----------------------|----------|----------------|---------|-------------|---------------|----------|---------|-------|--------------|
| 1              |     |                       |          |                |         |             |               |          |         | 1     | 2            |
| 2              |     |                       |          |                |         |             |               |          |         |       |              |
| 3              |     |                       |          |                |         |             |               |          |         |       |              |
| 4              |     |                       |          |                |         |             |               |          |         |       |              |
| 5              |     |                       |          |                |         |             |               |          |         |       |              |
| dan seterusnya |     |                       |          |                |         |             |               |          |         |       |              |

Previous

1

2

3

Next

Keluar

Gambar L.4. 6 Desain View 'suratKeluar' Akses Dilakukan Sekretaris

Tambah

Data Pengguna

Kategori

Surat Masuk

Surat Keluar

Kata Kunci

Cari

| No             | REG | TANGGAL SURAT DIKIRIM | PENGIRIM | DITUJUKAN PADA | PERIHAL | DARI AGENDA | DENGAN AGENDA | LAMPIRAN | CATATAN | JARISIP | BALASAN DARI |
|----------------|-----|-----------------------|----------|----------------|---------|-------------|---------------|----------|---------|---------|--------------|
| 1              |     |                       |          |                |         |             |               |          |         | 1       | 2            |
| 2              |     |                       |          |                |         |             |               |          |         |         |              |
| 3              |     |                       |          |                |         |             |               |          |         |         |              |
| 4              |     |                       |          |                |         |             |               |          |         |         |              |
| 5              |     |                       |          |                |         |             |               |          |         |         |              |
| dan seterusnya |     |                       |          |                |         |             |               |          |         |         |              |

Previous

Keluar

1

2

3

Next

Perbaiki

Perbaiki

Perbaiki

Perbaiki

Perbaiki

Perbaiki

Gambar L.4. 7 Desain View 'suratKeluar' Akses Dilakukan Seksi

No. Surat : tempat nomor surat ditampilkan

Kepada :

Perihal :

Dari Agenda :

Dengan Agenda :

Lampiran :

Catatan :

Arsip :

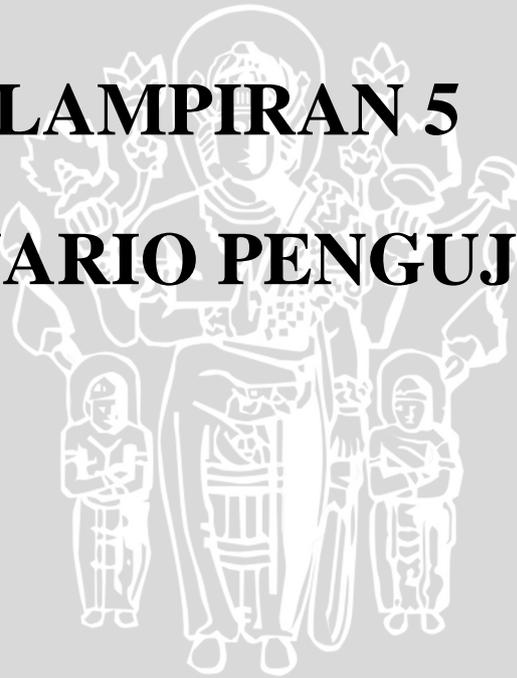
**Gambar L.4. 8 Desain View ‘detailSuratKeluar’ dan ‘perbaikiDetailSuratKeluar’**



UNIVERSITAS BRAWIJAYA

## LAMPIRAN 5

# SKENARIO PENGUJIAN



Tabel L.5. 1 Skenario Pengujian 1

| ID Pengujian | SP-01   |
|--------------|---|
| Tujuan       | <ol style="list-style-type: none"> <li>1. Menunjukkan bahwa aplikasi sangat bergantung pada <i>database</i>.</li> <li>2. Membuktikan bahwa fitur identifikasi pengguna (<i>log in</i>) pada aplikasi dapat memberikan peringatan apabila ID Pengguna dan <i>Password</i> yang dimasukkan oleh pengguna salah.</li> <li>3. Membuktikan bahwa aplikasi dapat mengidentifikasi pengguna dan memberikan hak akses sesuai aturan hak akses arsip surat, yaitu : <ol style="list-style-type: none"> <li>a. Sekretaris dapat mengakses semua surat masuk dan surat keluar.</li> <li>b. Seksi hanya dapat mengakses surat masuk dan surat keluar milik seksi yang bersangkutan.</li> </ol> <p>Aturan hak akses sekretaris dan seksi tersebut dapat dibuktikan dengan menjalankan fitur melihat surat masuk (UC-01), melihat surat keluar (UC-02).</p> </li> <li>4. Membuktikan bahwa arsip surat tersimpan dalam <i>database</i> dan dapat ditampilkan kembali.</li> </ol>  |
| Skenario     | <p>Segala kemungkinan kondisi dari pengisian ID Pengguna dan <i>Password</i> akan dicoba. ID Pengguna dan <i>Password</i> dinyatakan benar apabila sesuai dengan data yang terdapat pada <i>database</i> 'bri_db'. ID Pengguna dan <i>Password</i> yang terdapat pada 'beri_db' untuk pengujian ini adalah sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. Sekretaris <ul style="list-style-type: none"> <li>• ID Pengguna : id1</li> <li>• <i>Password</i> : 12345678</li> </ul> </li> <li>2. Seksi 1 <ul style="list-style-type: none"> <li>• ID Pengguna : id2</li> <li>• <i>Password</i> : 12345678</li> </ul> </li> <li>3. Seksi 2 <ul style="list-style-type: none"> <li>• ID Pengguna : id3</li> <li>• <i>Password</i> : 12345678</li> </ul> </li> <li>4. Seksi 3 <ul style="list-style-type: none"> <li>• ID Pengguna : id4</li> <li>• <i>Password</i> : 12345678</li> </ul> </li> <li>5. Seksi 4 <ul style="list-style-type: none"> <li>• ID Pengguna : id5</li> <li>• <i>Password</i> : 12345678</li> </ul> </li> <li>6. Seksi 5 <ul style="list-style-type: none"> <li>• ID Pengguna : id6</li> <li>• <i>Password</i> : 12345678</li> </ul> </li> </ol> |

|                       |   |
|-----------------------|---|
|                       | <p>Pada pengujian ini tidak semua ID Pengguna dan <i>Password</i> akan digunakan, cukup hanya ID Pengguna dan <i>Password</i> milik sekretaris, dan 2 ID Pengguna dan <i>Password</i> milik seksi. Data arsip surat masuk dan surat keluar telah diisi dengan data arsip percobaan.</p>   |
| Kasus pengujian       | <p>KP-01-1. Memutuskan sambungan aplikasi ke <i>database</i>.</p> <p>KP-01-2. ID Pengguna : kosong<br/><i>Password</i> : kosong</p> <p>KP-01-3. ID Pengguna : kosong<br/><i>Password</i> : 1234567</p> <p>KP-01-4. ID Pengguna : id8<br/><i>Password</i> : kosong</p> <p>KP-01-5. ID Pengguna : id8<br/><i>Password</i> : 1234567</p> <p>KP-01-6. ID Pengguna : id1<br/><i>Password</i> : 1234567</p> <p>KP-01-7. ID Pengguna : id<br/><i>Password</i> : 12345678</p> <p>KP-01-8. ID Pengguna : id1 (sekretaris)<br/><i>Password</i> : 12345678 (sekretaris)</p> <p>KP-01-9. ID Pengguna : id2 (seksi1)<br/><i>Password</i> : 12345678 (seksi1)</p> <p>KP-01-10. ID Pengguna : id3 (seksi2)<br/><i>Password</i> : 12345678 (seksi2)</p> |
| <i>Pre-conditions</i> | <ol style="list-style-type: none"> <li>1. Untuk KP-01-1 MySQL tidak perlu dijalankan, cukup hanya menjalankan Apache sebagai <i>web server</i>.</li> <li>2. Untuk KP-01-2 sampai dengan KP-01-10 Apache dan MySQL harus sudah dijalankan sehingga <i>database</i> sudah terhubung dengan aplikasi,</li> <li>3. <i>Browser</i> dijalankan,</li> <li>4. Menjalankan aplikasi sistem informasi pengarsipan surat Bank Rakyat Indonesia Kantor Wilayah Malang.</li> </ol>   |
| Hasil yang diharapkan | <p>KP-01-1. Terdapat peringatan “A Database Error Occurred” dan aplikasi tidak bisa dijalankan.</p> <p>KP-01-2. Terdapat peringatan berupa “ID Pengguna dan <i>Password</i> tidak boleh kosong”.</p> <p>KP-01-3. Terdapat peringatan berupa “ID Pengguna dan <i>Password</i> tidak boleh kosong”.</p> <p>KP-01-4. Terdapat peringatan berupa “ID Pengguna dan <i>Password</i> tidak boleh kosong”.</p> <p>KP-01-5. Terdapat peringatan berupa “ID Pengguna tidak ditemukan”.</p>  |

|                        |   |
|------------------------|---|
|                        | <p>KP-01-6. Terdapat peringatan berupa “Kombinasi ID Pengguna dan <i>Password</i> salah”.</p> <p>KP-01-7. Terdapat peringatan berupa “ID Pengguna tidak ditemukan”.</p> <p>KP-01-8. Sekretaris diteruskan pada tampilan <i>view</i> ‘dataUser’, lalu sekretaris dapat mengakses seluruh surat masuk dan surat keluar.</p> <p>KP-01-9. Seksi 1 diteruskan pada tampilan <i>view</i> ‘dataUser’, lalu seksi 1 hanya dapat mengakses surat masuk dan surat keluar milik seksi yang bersangkutan.</p> <p>KP-01-10. Seksi 2 diteruskan pada tampilan <i>view</i> ‘dataUser’, lalu seksi 2 hanya dapat mengakses surat masuk dan surat keluar milik seksi yang bersangkutan.</p>  |
| Hasil yang sebenarnya  | <p>KP-01-1. Terdapat peringatan “A Database Error Occurred” dan aplikasi tidak bisa dijalankan.</p> <p>KP-01-2. Terdapat peringatan berupa “ID Pengguna dan <i>Password</i> tidak boleh kosong”.</p> <p>KP-01-3. Terdapat peringatan berupa “ID Pengguna dan <i>Password</i> tidak boleh kosong”.</p> <p>KP-01-4. Terdapat peringatan berupa “ID Pengguna dan <i>Password</i> tidak boleh kosong”.</p> <p>KP-01-5. Terdapat peringatan berupa “ID Pengguna tidak ditemukan”.</p> <p>KP-01-6. Terdapat peringatan berupa “Kombinasi ID Pengguna dan <i>Password</i> salah”.</p> <p>KP-01-7. Terdapat peringatan berupa “ID Pengguna tidak ditemukan”.</p> <p>KP-01-8. Sekretaris diteruskan pada tampilan <i>view</i> ‘dataUser’, lalu sekretaris dapat mengakses seluruh surat masuk dan surat keluar.</p> <p>KP-01-9. Seksi 1 diteruskan pada tampilan <i>view</i> ‘dataUser’, lalu seksi 1 hanya dapat mengakses surat masuk dan surat keluar milik seksi yang bersangkutan.</p> <p>KP-01-10. Seksi 2 diteruskan pada tampilan <i>view</i> ‘dataUser’, lalu seksi 2 hanya dapat mengakses surat masuk dan surat keluar milik seksi yang bersangkutan.</p> |
| <i>Post conditions</i> | Apabila identifikasi pengguna ( <i>log in</i> ) berhasil maka aplikasi akan menampilkan <i>view</i> ‘dataUser’ dan pengguna dapat menjalankan fitur-fitur lain yang terdapat dalam aplikasi.  |

Tabel L.5. 2 Skenario Pengujian 2

|                        |   |
|------------------------|---|
| ID Pengujian           | SP-02   |
| Tujuan                 | Membuktikan bahwa ketika pengguna melihat surat masuk (UC-01) atau melihat surat keluar (UC-02) mereka dapat menjalankan fitur pencarian (UC-03).   |
| Skenario               | Pengujian untuk fitur pencarian (UC-03) akan dilakukan pada salah satu kolom yang tersedia dalam fitur pencarian baik dengan kata kunci yang datanya tersedia ataupun tidak tersedia. Data arsip surat masuk dan surat keluar telah diisi dengan data arsip percobaan.  |
| Kasus pengujian        | <p>KP-02-1. Pencarian pada arsip surat masuk kolom 'BALASAN' dengan kata kunci 'Belum'.</p> <p>KP-02-2. Pencarian pada arsip surat masuk kolom 'DITUJUKAN PADA' dengan kata kunci 'Bidang3'.</p> <p>KP-02-3. Pencarian pada arsip surat keluar kolom 'PERIHAL' dengan kata kunci 'testPerihal4'.</p> <p>KP-02-4. Pencarian pada arsip surat keluar kolom 'LAMPIRAN' dengan kata kunci '10'.</p> |
| <i>Pre-conditions</i>  | <p>Untuk kasus pengujian kode KP-02-1 dan KP-02-2 aplikasi harus dalam keadaan menampilkan arsip surat masuk (<i>view</i> 'viewSuratMasuk').</p> <p>Untuk kasus pengujian kode KP-02-3 dan KP-02-4 aplikasi harus dalam keadaan menampilkan arsip surat keluar (<i>view</i> 'viewSuratKeluar').</p>   |
| Hasil yang diharapkan  | <p>KP-02-1. Aplikasi menampilkan hasil pencarian karena data memang tersedia.</p> <p>KP-02-2. Aplikasi tidak menampilkan data yang dicari karena data memang tidak tersedia.</p> <p>KP-02-3. Aplikasi menampilkan hasil pencarian karena data memang tersedia.</p> <p>KP-02-4. Aplikasi tidak menampilkan data yang dicari karena data memang tidak tersedia.</p>                               |
| Hasil yang sebenarnya  | <p>KP-02-1. Aplikasi menampilkan hasil pencarian karena data memang tersedia.</p> <p>KP-02-2. Aplikasi tidak menampilkan data yang dicari karena data memang tidak tersedia.</p> <p>KP-02-3. Aplikasi menampilkan hasil pencarian karena data memang tersedia.</p> <p>KP-02-4. Aplikasi tidak menampilkan data yang dicari karena data memang tidak tersedia.</p>                               |
| <i>Post conditions</i> | Aplikasi menampilkan hasil pencarian.   |

Tabel L.5. 3 Skenario Pengujian 3

|                       |   |
|-----------------------|---|
| ID Pengujian          | SP-03   |
| Tujuan                | Membuktikan bahwa ketika pengguna melihat surat masuk (UC-01) atau melihat surat keluar (UC-02) mereka dapat menjalankan fitur pengurutan (UC-04).  |
| Skenario              | Pengujian pada fitur pengurutan (UC-04) akan dilakukan pada tampilan arsip surat masuk dan surat keluar. Akan diambil sampel masing-masing 1 kolom untuk diurutkan secara <i>ascending</i> dan 1 kolom untuk diurutkan secara <i>descending</i> . Data arsip surat masuk dan surat keluar telah diisi dengan data arsip percobaan.  |
| Kasus pengujian       | <p>KP-03-1. Pada tampilan arsip surat masuk klik judul kolom 'REG' satu kali.</p> <p>KP-03-2. Pada tampilan arsip surat masuk klik judul kolom 'REG' satu kali lagi.</p> <p>KP-03-3. Pada tampilan arsip surat keluar klik judul kolom 'LAMPIRAN' satu kali.</p> <p>KP-03-4. Pada tampilan arsip surat keluar klik judul kolom 'LAMPIRAN' satu kali lagi.</p>   |
| <i>Pre-conditions</i> | <p>Untuk kasus pengujian kode KP-03-1 dan KP-03-2 aplikasi harus dalam keadaan menampilkan arsip surat masuk (<i>view</i> 'viewSuratMasuk').</p> <p>Untuk kasus pengujian kode KP-03-3 dan KP-03-4 aplikasi harus dalam keadaan menampilkan arsip surat keluar (<i>view</i> 'viewSuratKeluar').</p>   |
| Hasil yang diharapkan | <p>KP-03-1. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>ascending</i> menurut kolom 'REG'.</p> <p>KP-03-2. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>descending</i> menurut kolom 'REG'.</p> <p>KP-03-3. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>ascending</i> menurut kolom 'LAMPIRAN'.</p> <p>KP-03-4. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>descending</i> menurut kolom 'LAMPIRAN'.</p> |
| Hasil yang sebenarnya | <p>KP-03-1. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>ascending</i> menurut kolom 'REG'.</p> <p>KP-03-2. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>descending</i> menurut kolom 'REG'.</p>  |

|                        |  |
|------------------------|--|
|                        | <p>KP-03-3. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>ascending</i> menurut kolom 'LAMPIRAN'.</p> <p>KP-03-4. Aplikasi menampilkan data arsip surat masuk yang diurutkan secara <i>descending</i> menurut kolom 'LAMPIRAN'.</p> |
| <i>Post conditions</i> | Aplikasi Menampilkan data arsip surat sesuai kondisi pengurutan yang diinginkan oleh pengguna.   |

**Tabel L.5. 4 Skenario Pengujian 4**

|                 |  |
|-----------------|--|
| ID Pengujian    | SP-04  |
| Tujuan          | <ol style="list-style-type: none"> <li>1. Membuktikan bahwa fitur tambah surat masuk (UC-05) yang dilakukan oleh sekretaris dapat digunakan.</li> <li>2. Membuktikan bahwa surat masuk akan tersimpan hanya apabila seluruh kolom terisi.</li> <li>3. Membuktikan bahwa ketika surat masuk ditambahkan maka aplikasi memberi nomor register surat (UC-06) pada surat tersebut.</li> <li>4. Nomor register surat yang dibuat sesuai dengan aturan yang telah dibuat pada proses perancangan.</li> <li>5. Membuktikan bahwa nomor register yang dibuat oleh aplikasi sesuai dengan nomor register yang dibuat secara manual pada prosedur pengarsipan sebelum ada aplikasi.</li> </ol> |
| Skenario        | Aplikasi diakses menggunakan ID Pengguna dan <i>Password</i> milik sekretaris. Pada pengujian fitur ini pengguna menambahkan surat masuk ke dalam aplikasi dan mengisi kolom isian data umum surat masuk dengan segala kemungkinan. Lalu klik tombol 'Simpan'.   |
| Kasus pengujian | <p>KP-04-1. Dari : kosong<br/>Kepada : bidang3<br/>Perihal : kosong</p> <p>KP-04-2. Dari : CV. Indah Motor<br/>Kepada : bidang3<br/>Perihal : kosong</p> <p>KP-04-3. Dari : kosong<br/>Kepada : bidang3<br/>Perihal : Penundaan Pembayaran Iuran Kredit</p> <p>KP-04-4. Dari : CV. Indah Motor<br/>Kepada : bidang3<br/>Perihal : Penundaan Pembayaran Iuran Kredit</p>  |

|                        |   |
|------------------------|---|
|                        | Nomor register yang dibuat manual adalah 20148188.  |
| <i>Pre-conditions</i>  | <ol style="list-style-type: none"> <li>1. Masuk ke dalam aplikasi menggunakan hak akses sekretaris.</li> <li>2. Klik tombol ‘Surat Masuk’ untuk menjalankan fitur melihat surat masuk (UC-01).</li> <li>3. Klik tombol ‘Tambah’ untuk menjalankan fitur tambah surat masuk (UC-05).</li> </ol>  |
| Hasil yang diharapkan  | <p>KP-04-1. Aplikasi menampilkan kolom isian data umum surat masuk kembali.</p> <p>KP-04-2. Aplikasi menampilkan kolom isian data umum surat masuk kembali.</p> <p>KP-04-3. Aplikasi menampilkan kolom isian data umum surat masuk kembali.</p> <p>KP-04-4. Data umum surat masuk baru berhasil tersimpan dan data arsip surat masuk tersebut tampil pada barisan paling atas dengan nomor register 20148188.</p> |
| Hasil yang sebenarnya  | <p>KP-04-1. Aplikasi menampilkan kolom isian data umum surat masuk kembali.</p> <p>KP-04-2. Aplikasi menampilkan kolom isian data umum surat masuk kembali.</p> <p>KP-04-3. Aplikasi menampilkan kolom isian data umum surat masuk kembali.</p> <p>KP-04-4. Data umum surat masuk baru berhasil tersimpan dan data arsip surat masuk tersebut tampil pada barisan paling atas dengan nomor register 20148188.</p> |
| <i>Post conditions</i> | Apabila data umum surat masuk lengkap maka aplikasi akan menampilkan data arsip surat masuk pada tampilan <i>view</i> ‘viewSuratMasuk’ namun apabila data umum tidak lengkap maka kolom isian data umum surat masuk akan ditampilkan kembali.   |

Tabel L.5. 5 Skenario Pengujian 5

|              |   |
|--------------|---|
| ID Pengujian | SP-05   |
| Tujuan       | <ol style="list-style-type: none"> <li>1. Membuktikan fitur terima surat masuk (UC-07) dapat dijalankan.</li> <li>2. Membuktikan bahwa proses penyimpanan pada fitur terima surat masuk (UC-07) tidak dapat dilakukan apabila kolom isian yang harus diisi masih kosong.</li> <li>3. Membuktikan fitur unggah hasil <i>scan</i> arsip surat (UC-08) dapat dijalankan dan <i>file</i> hasil <i>scan</i> tersimpan dalam <i>database</i> tanpa mengubah isi surat.</li> </ol> |
| Skenario     | Aplikasi diakses menggunakan ID Pengguna dan <i>Password</i> milik seksi3. Pada pengujian fitur ini   |

|                        |  |
|------------------------|--|
|                        | dilakukan proses memasukkan detail surat masuk tersebut ke dalam aplikasi dengan cara mengisi kolom isian detail surat masuk dengan kemungkinan mengunggah <i>file</i> hasil <i>scan</i> arsip surat masuk atau tidak. Lalu klik tombol ‘Simpan’.  |
| Kasus pengujian        | <p>KP-05-1. Dari agenda : kosong,<br/>         Dengan agenda : kosong,<br/>         Lampiran : 0,<br/>         Catatan : kosong,<br/>         Arsip dibiarkan kosong.</p> <p>KP-05-2. Dari agenda : kosong,<br/>         Dengan agenda : kosong,<br/>         Lampiran : 0,<br/>         Catatan : kosong,<br/>         Arsip diisi dengan <i>file</i> bernama ‘scan-1.jpg’.</p> |
| <i>Pre-conditions</i>  | <ol style="list-style-type: none"> <li>1. Masuk ke dalam aplikasi menggunakan hak akses seksi3.</li> <li>2. Klik tombol ‘Surat Masuk’ untuk menjalankan fitur melihat surat masuk (UC-01).</li> <li>3. Klik tombol ‘Terima’ pada data surat masuk yang akan diterima untuk menjalankan fitur terima surat masuk (UC-07).</li> </ol>  |
| Hasil yang diharapkan  | <p>KP-05-1. Detail surat masuk tidak dapat disimpan dan terdapat peringatan berupa “Arsip belum terisi”.</p> <p>KP-05-2. Detail surat masuk berhasil tersimpan dan <i>file</i> arsip dapat ditampilkan.</p>  |
| Hasil yang sebenarnya  | <p>KP-05-1. Detail surat masuk tidak dapat disimpan dan terdapat peringatan berupa “Arsip belum terisi”.</p> <p>KP-05-2. Detail surat masuk berhasil tersimpan dan <i>file</i> arsip dapat ditampilkan.</p>  |
| <i>Post conditions</i> | Ketika arsip surat masuk diisi (diunggah) maka detail surat masuk tersimpan dan aplikasi menampilkan <i>view</i> ‘veiwSuratMasuk’.   |

Tabel L.5. 6 Skenario Pengujian 6

|              |   |
|--------------|---|
| ID Pengujian | SP-06   |
| Tujuan       | <ol style="list-style-type: none"> <li>1. Membuktikan fitur membuat surat keluar (UC-09) dapat dijalankan.</li> <li>2. Membuktikan bahwa fitur memberi nomor surat (UC-10) dapat dijalankan dan nomor surat yang dibuat benar.</li> <li>3. Membuktikan bahwa fitur unggah hasil <i>scan</i> arsip surat (UC-08) dapat dijalankan dan <i>file</i> hasil <i>scan</i></li> </ol> |

|                       |  |
|-----------------------|--|
|                       | <p>tersimpan dalam <i>database</i> tanpa mengubah isi surat.</p> <ol style="list-style-type: none"> <li>4. Membuktikan bahwa ketika detail surat keluar berhasil disimpan maka aplikasi memberi nomor register surat (UC-06) pada surat tersebut.</li> <li>5. Nomor register surat yang dibuat sesuai dengan aturan yang telah dibuat pada proses perancangan.</li> <li>6. Membuktikan bahwa nomor register yang dibuat oleh aplikasi sesuai dengan nomor register yang dibuat secara manual pada prosedur pengarsipan sebelum ada aplikasi.</li> </ol>  |
| Skenario              | Aplikasi diakses menggunakan ID Pengguna dan <i>Password</i> milik seksi4. Pada pengujian fitur ini dilakukan proses memasukkan detail surat keluar ke dalam aplikasi dengan cara mengisi kolom isian detail surat keluar dan mengunggah <i>file</i> hasil <i>scan</i> arsip surat keluar. Lalu klik tombol ‘Simpan’.  |
| Kasus pengujian       | <p>KP-06-1. Membiarkan kosong kolom isian detail surat keluar yang harus diisi.<br/>         Kepada : kosong (harus diisi),<br/>         Perihal : kosong (harus diisi),<br/>         Dari agenda : kosong,<br/>         Dengan agenda : kosong,<br/>         Lampiran : kosong,<br/>         Catatan : kosong,<br/>         Arsip dibiarkan kosong (harus diisi).</p> <p>KP-06-2. Mengisi seluruh kolom isian detail surat keluar yang harus diisi.<br/>         Kepada : KepadaCoba1 (harus diisi),<br/>         Perihal : PerihalCoba1 (harus diisi),<br/>         Dari agenda : kosong,<br/>         Dengan agenda : kosong,<br/>         Lampiran : kosong,<br/>         Catatan : kosong,<br/>         Arsip diisi dengan <i>file</i> bernama ‘Surat Perjanjian Pinjaman.jpg’ (harus diisi).<br/>         Nomor register yang dibuat manual adalah 20148189.</p> |
| <i>Pre-conditions</i> | <ol style="list-style-type: none"> <li>1. Masuk ke dalam aplikasi menggunakan hak akses seksi4.</li> <li>2. Klik tombol ‘Surat Keluar’ untuk menjalankan fitur melihat surat keluar (UC-02).</li> <li>3. Klik tombol ‘Tambah’ untuk menjalankan fitur membuat surat keluar (UC-09).</li> </ol>   |
| Hasil yang diharapkan | KP-07-1.   |

|                        |  |
|------------------------|--|
|                        | <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat keluar baru merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat keluar tidak dapat disimpan dan terdapat peringatan berupa “Arsip Belum Terisi atau perihal dan kepada kosong”.</li> </ol> <p>KP-07-2.</p> <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat keluar baru merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat keluar berhasil tersimpan.</li> <li>3) <i>File</i> arsip surat keluar dapat ditampilkan.</li> <li>4) Data arsip surat keluar tersebut tampil pada barisan paling atas dengan nomor register 20148189.</li> </ol>                |
| Hasil yang sebenarnya  | <p>KP-06-1.</p> <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat keluar baru merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat keluar tidak dapat disimpan dan terdapat peringatan berupa “Arsip belum terisi atau kolom kepada dan perihal belum terisi”.</li> </ol> <p>KP-06-2.</p> <ol style="list-style-type: none"> <li>1) Nomor surat yang digunakan untuk detail surat keluar baru merupakan lanjutan dari nomor surat yang terakhir digunakan.</li> <li>2) Detail surat keluar berhasil tersimpan.</li> <li>3) <i>File</i> arsip dapat ditampilkan.</li> <li>4) Data arsip surat keluar tersebut tampil pada barisan paling atas dengan nomor register 20148189.</li> </ol> |
| <i>Post conditions</i> | Apabila kolom isian yang harus diisi sudah diisi maka ketika tombol ‘Simpan’ diklik aplikasi akan menampilkan <i>view</i> ‘viewSuratKeluar’.   |