

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan teknologi informasi memberi pengaruh besar pada manajemen perkantoran terutama dalam hal membangun *E-Office*. *E-Office* bertujuan untuk untuk menutupi meningkatnya penggunaan komputer berbasis teknologi informasi untuk pekerjaan kantor, terutama di tahun 1980-an. Itu adalah kata kunci pemasaran pada saat itu, tapi sekarang tidak begitu banyak digunakan karena semua kantor modern adalah kantor elektronik. *E-office* merupakan *software* dan digunakan untuk mengatur pola pekerjaan yang sudah atau akan dilakukan oleh pegawai, menyimpan, dan

mengontrol konten-konten yang ada didalam suatu organisasi secara elektronik.

Pendapat lain mengatakan *Electronic Office* atau *E-Office* adalah suatu sistem terotomasi yang mengelola administrasi perkantoran dan memusatkan komponen-komponen sebuah organisasi, dimana data, informasi, dan komunikasi dilakukan melalui media telekomunikas, Istilah lain untuk *e-office* adalah *office automation*.

Menurut Robert. M (2001), *Electronic Office (e-office)* adalah suatu sistem yang berhubungan dengan administrasi secara maya memusatkan komponen komponen sebuah organisasi dimana data, informasi dan komunikasi dibuat melalui media telekomunikasi. Sedangkan menurut Bowen. J., *e-office* adalah istilah yang

diciptakan untuk menutupi meningkatnya kebutuhan penggunaan komputer berbasis teknologi informasi untuk pekerjaan kantor. Penerapan *e-office* sangat dibutuhkan oleh suatu kantor atau perusahaan pada skala yang lebih besar, hal ini sangat dibutuhkan karena mengingat persaingan yang semakin ketat sehingga menuntut perusahaan untuk dapat bergerak cepat dan memaksimalkan kinerja perusahaannya. Selain itu mahalanya biaya operasional juga mempengaruhi kondisi suatu perusahaan, sehingga perusahaan berlomba-lomba menekan biaya operasional tanpa mempengaruhi kinerja perusahaan itu sendiri. *E-office* merupakan sebuah solusi dalam membantu perusahaan dalam menghadapi ketatnya persaingan.

Adapun pada Dinas Perhubungan Provinsi Jawa Barat telah terkomputerisasi dengan menggunakan *microsoft excel*, ada beberapa tahapan yang masih melakukan proses manualisasi dalam beberapa hal antara lain yaitu dalam hal administrasi perkantoran.

Dengan adanya permasalahan tersebut, diperlukanlah sebuah solusi yang dapat mengatasi problematika yang dihadapi. Sehingga timbullah ide untuk membuat “Rancang Bangun Aplikasi *E-Office* Berbasis Web Dengan Menggunakan Metode *Rapid Application Development* (Studi Kasus Dinas Perhubungan Provinsi Jawa Barat)” dimana aplikasi ini akan membantu dalam hal mengelola administrasi perkantoran pada Dinas Perhubungan Provinsi Jawa Barat.

1.2 RUMUSAN MASALAH

Dari latar belakang diatas dapat disimpulkan rumusan masalah antara lain:

1. Bagaimana cara mengelola surat masuk dan surat keluar dengan terkomputerisasi pada Dinas Perhubungan Provinsi Jawa Barat.
2. Bagaimana cara menentukan disposisi surat secara terkomputerisasi pada Dinas Perhubungan Provinsi Jawa Barat.
3. Bagaimana menentukan schedule atau jadwal pegawai pada Dinas Perhubungan Provinsi Jawa Barat.

1.3 BATASAN MASALAH

Agar penelitian ini lebih terarah, fokus dan tidak meluas, penulis membatasi

penelitian yaitu hanya membahas tentang pengelolaan surat masuk dan keluar, penentuan disposisi surat, dan penentuan schedule atau jadwal pegawai berbasis teknologi informas.

1.4 TUJUAN PEMBUATAN PENELITIAN

Adapun tujuan dari penelitian ini sebagai berikut :

1. Dapat membantu proses pengelolaan surat masuk dan surat keluar.
2. Dapat membantu dalam penentuam disposisi surat.
3. Dapat membantu menentukan schedule atau jadwail pegawai.

1.5 MANFAAT PENELITIAN

Penelitian ini diharapkan dapat memberikan manfaat yang berguna diantaranya yaitu :

1. Menghasilkan sebuah aplikasi yang dapat membantau keseluruhan progres dalam mengelola surat masuk dan surat keluar.
2. Dapat membantu dalam menentukan disposisi surat.
3. Dapat membantu dalam menentukan schedule atau jadwal pegawai.

1.6 LANDASAN TEORI

1.6.1 Sistem Informasi

E-Office bertujuan untuk untuk menutupi meningkatnya penggunaan komputer berbasis teknologi informasi untuk pekerjaan kantor, terutama di tahun 1980-an. Itu adalah kata kunci pemasaran pada saat itu, tapi sekarang tidak begitu banyak digunakan karena semua kantor modern adalah kantor elektronik. *E-office*

merupakan *software* dan digunakan untuk mengatur pola pekerjaan yang sudah atau akan dilakukan oleh pegawai, menyimpan, dan mengontrol konten-konten yang ada didalam suatu organisasi secara elektronik.

Pendapat lain mengatakan *Electronic Office* atau *e-Office* adalah suatu sistem terotomasi yang mengelola administrasi perkantoran dan memusatkan komponen-komponen sebuah organisasi, dimana data, informasi, dan komunikasi dilakukan melalui media telekomunikas, Istilah lain untuk *e-office* adalah *office automation*. Menurut Robert. M (2001), *Electronic Office (e-office)* adalah suatu sistem yang berhubungan dengan administrasi secara maya memusatkan komponen komponen sebuah organisasi dimana data, informasi

dan komunikasi dibuat melalui media telekomunikasi. Sedangkan menurut Bowen. J., *e-office* adalah istilah yang diciptakan untuk menutupi meningkatnya kebutuhan penggunaan komputer berbasis teknologi informasi untuk pekerjaan kantor. Penerapan *e-office* sangat dibutuhkan oleh suatu kantor atau perusahaan pada skala yang lebih besar, hal ini sangat dibutuhkan karena mengingat persaingan yang semakin ketat sehingga menuntut perusahaan untuk dapat bergerak cepat dan memaksimalkan kinerja perusahaannya. Selain itu mahalanya biaya operasional juga mempengaruhi kondisi suatu perusahaan, sehingga perusahaan berlomba-lomba menekan biaya operasional tanpa mempengaruhi kinerja perusahaan itu

sendiri. *E-office* merupakan sebuah solusi dalam membantu perusahaan dalam menghadapi ketatnya persaingan

1.7 RINGKASAN ISI PENELITIAN

Pada Intrensi 1 peneliti akan membuat suatu rancang bangun aplikasi *E-Office* pada Dinas Perhubungan Provinsi Jawa Barat. Dimana dalam rancang bangun aplikasi ini berbasis *web* karna pengaplikasi melalui *web*, maka dari itu setiap bagian atau pegawai yang bersangkutan dapat dengan mudah mengakses data atau surat yang masuk dan memeriksa jadwal acara dari setiap surat yang masuk dan keluar melalui aplikasi *web* ini.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.6.1 E-Office

Menurut Robert. M (2001), *Electronic Office (e-office)* adalah suatu sistem yang berhubungan dengan administrasi secara maya memusatkan komponen komponen sebuah organisasi dimana data, informasi dan komunikasi dibuat melalui media telekomunikasi. Sedangkan menurut Bowen. J., *e-office* adalah istilah yang diciptakan

untuk menutupi meningkatnya kebutuhan penggunaan komputer berbasis teknologi informasi untuk pekerjaan kantor. Penerapan *e-office* sangat dibutuhkan oleh suatu kantor atau perusahaan pada skala yang lebih besar, hal ini sangat dibutuhkan karena mengingat persaingan yang semakin ketat sehingga menuntut perusahaan untuk dapat bergerak cepat dan memaksimalkan kinerja perusahaannya. Selain itu mahal nya biaya operasional juga

mempengaruhi kondisi suatu perusahaan, sehingga perusahaan berlomba-lomba menekan biaya operasional tanpa mempengaruhi kinerja perusahaan itu sendiri. *E-office* merupakan sebuah solusi dalam membantu perusahaan dalam menghadapi ketatnya persaingan.

2.6.2 Database

Kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware

komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu.

2.6.3 PHP

PHP (akronim dari PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman yang berfungsi untuk membuat *website* dinamis maupun aplikasi *web*. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website*.

2.6.4 CodeIgniter

CodeIgniter adalah sebuah *framework* PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi *web*. Dilengkapi banyak library dan helper yang berguna di dalamnya dan tentunya mempermudah proses development. Sedangkan Twitter Bootstrap adalah sebuah alat bantu (*framework*) HTML dan CSS untuk membuat sebuah tampilan halaman *website* yang elegan dan *support* segala macam *device*. Dengan implementasi *Twitter Bootstrap* pada

framework CodeIgniter akan mempermudah dan mempercepat pembuatan maupun pengembangan *website* dinamis tanpa harus kesulitan membuat desain.

2.6.5 MySQL

Pengenalan MySQL
MySQL adalah salah satu jenis *database server* yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses *database* nya. Lisensi Mysql adalah FOSS *License Exception* dan ada juga yang versi komersial nya. Tag Mysql adalah “*The World's most popular open source database*”. MySQL

tersedia untuk beberapa *platform*, di antara nya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap Mysql, anda dapat menggunakan *software* tertentu, di antara nya adalah phpmyadmin dan mysql yog. Pada kesempatan kali ini, kita akan menggunakan phpmyadmin, yang terdapat dalam bundle xampp.

2.6.6 RAD (Rapid Application Development)

Rapid Application Development (RAD) adalah salah satu metode

pengembangan suatu sistem informasi dengan waktu yang relatif singkat. Untuk pengembangan suatu sistem informasi yang normal membutuhkan waktu minimal 180 hari, akan tetapi dengan menggunakan metode RAD suatu sistem dapat diselesaikan hanya dalam waktu 30-90 hari.

Tujuan utama dari semua metode sistem development adalah memberikan suatu sistem yang dapat memenuhi harapan dari para pemakai, akan tetapi sering kali di dalam melakukan pengembangan suatu sistem tidak melibatkan para pemakai

sistem secara langsung, sehingga hal ini menyebabkan sistem informasi yang dibuat jauh dari harapan pemakai yang dapat berakibat sistem tersebut walaupun dapat diterima tetapi para pemakainya enggan untuk menggunakannya atau bahkan para pemakai menolak untuk menggunakannya.

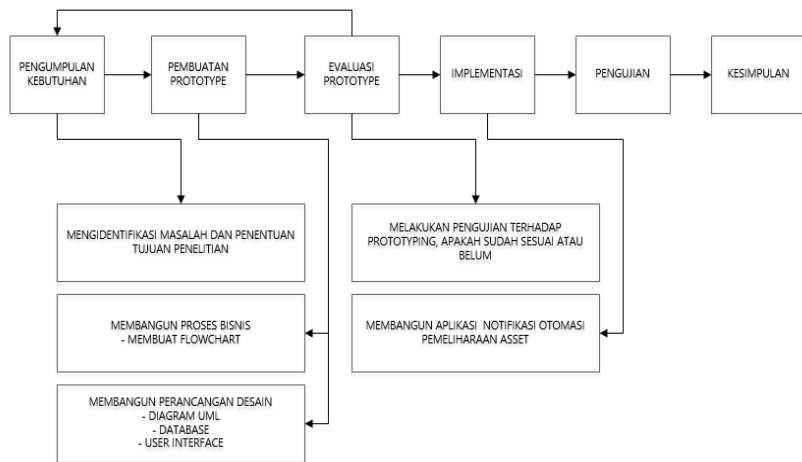
Pada saat RAD diimplementasikan, maka para pemakai bisa menjadi bagian dari keseluruhan proses pengembangan sistem dengan bertindak sebagai pengambil keputusan pada setiap tahapan pengembangan. RAD bisa menghasilkan suatu sistem dengan cepat karena sistem yang dikembangkan dapat

memenuhi keinginan dari para
pemakai sehingga dapat
mengurangi waktu untuk
pengembangan ulang setelah
tahap implementasi.

BAB IV

METODOLOGI PENELITIAN

4.1 Diagram Alur Metodologi Penelitian



Gambar 4. 1 Diagram Alur Metodologi Penelitian

4.2 Tahapan – tahapan Diagram Alur Metodologi Penelitian

Berikut adalah tahapan – tahapan diagram alur metodologi penelitian berdasarkan gambar sebelumnya :

4.2.1 Pengumpulan Kebutuhan

Pengumpulan kebutuhan dilakukan untuk menentukan permasalahan yang terjadi yang akan diteliti sehingga dapat ditentukan tujuan dari penelitian yang dilakukan

4.2.2 Membangun Prototyping

4.2.2.1 Membangun Proses Bisnis

Pada tahap ini dibuat proses bisnis atau analisis sistem yang bertujuan mengidentifikasi masalah dan hubungan antarproses yang terlibat dalam sistem, yang meliputi proses bisnis yang sedang berjalan dan proses bisnis yang akan dibangun.

4.2.2.2 Membangun Perancangan Desain

Pada tahap ini, dibuat rincian sistem agar dapat dimengerti oleh *user* yang terdiri dari tiga tahap, yaitu:

1. Tahap merancang gambaran sistem atau proses yang akan dibangun yang dimodelkan menggunakan diagram UML.
2. Tahap merancang *database* yang akan digunakan.
3. Tahap merancang tampilan *user interface* dan struktur menu sistem yang akan digunakan.

4.2.3 Evaluasi *Prototyping*

Setelah melakukan pembangunan *prototyping*, dilakukan evaluasi untuk mengetahui apakah *prototyping* yang akan digunakan sudah sesuai dengan kebutuhan atau belum. Jika sudah sesuai, maka akan

lanjut ke langkah selanjutnya. Dan apabila belum sesuai, maka akan direvisi dan kembali lagi untuk mengumpulkan kebutuhan.

4.2.4 Implementasi

Implementasi merupakan tahap pembuatan sistem berdasarkan hasil perancangan yang telah dibuat sebelumnya sehingga sistem dapat difungsikan dalam keadaan yang sebenarnya dan dapat diketahui apakah sistem yang dibuat sudah berhasil mencapai tujuan yang sebenarnya atau belum.

4.2.5 Pengujian

Setelah implementasi sistem selesai, langkah selanjutnya adalah pengujian terhadap sistem yang telah dibuat menggunakan pengujian *Black-box*. Pengujian bertujuan mengetahui apakah sistem yang telah dibuat sudah sesuai dengan kebutuhan *user* dan mengetahui apakah fungsi *input* dan

output sistem sudah sesuai dengan yang diharapkan atau tidak.

4.2.6 Kesimpulan dan Saran

Dari hasil pengujian sistem yang telah dibuat, maka penulis dapat membuat kesimpulan dari hasil penelitian yang dilakukan sehingga penulis bisa memberikan saran terhadap sistem yang telah dibuat.

BAB V

ANALISIS DAN PERANCANGAN SISTEM

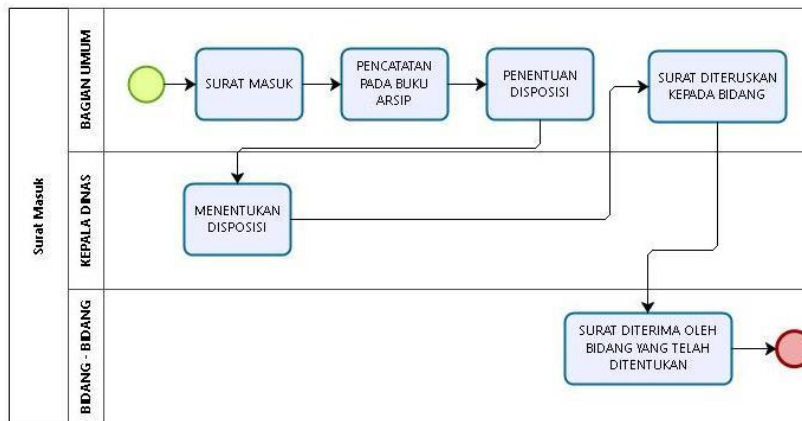
5.1 Analisis dan Perancangan Sistem

5.1.1 Analisis yang berjalan

Tahapan yang di perlukan dalam pembuatan suatu program yaitu menganalisa sistem yang telah ada, dimana analisa sistem merupakan proses mempelajari suatu sistem dengan cara menguraikan sistem tersebut kedalam elemen yang membentuknya. Selanjutnya mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang terjadi serta kebutuhan yang diperlukan, sehingga dapat di usulkan pembuatannya.

5.1.1.1 Analisis Prosedur yang berjalan

1. Analisis Prosedur Surat Masuk Yang Sedang Berjalan

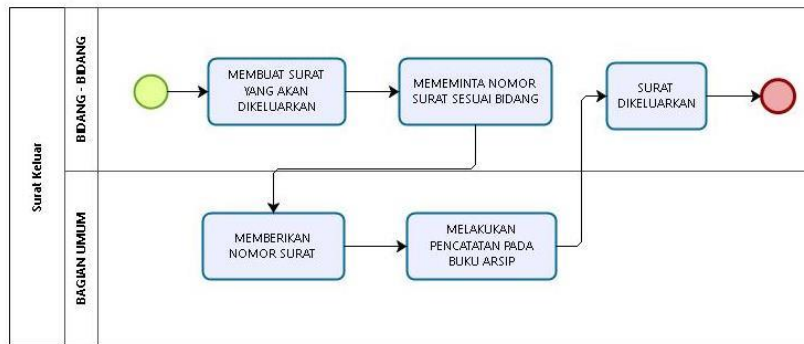


Gambar 5. 1 BPMN Prosedur Surat Masuk yang Sedang Berjalan

Pada gambar 5.1 BPMN diatas adalah penjelasan alur yang berjalan pada proses surat masuk dinas perhubungan. Mulai dari

menerima surat, setelah itu pencatatan pada buku arsip, penentuan disposisi oleh Kepala Dinas Setelah itu surat diteruskan kepada bidang yang sesuai dengan disposisi yang sudah ditentukan. Lalu bidang yang dituju akan menerima surat.

2. Analisis Prosedur Surat Keluar Yang Sedang Berjalan



Gambar 5. 2 BPMN Prosedur Surat Keluar yang Sedang Berjalan

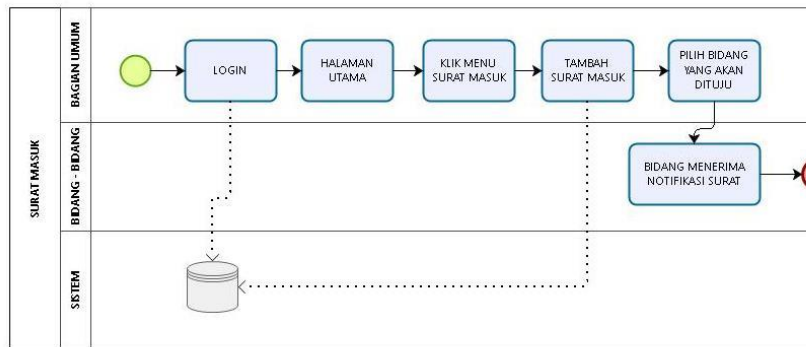
Pada gambar 5.2 BPMN diatas adalah penjelasan alur yang berjalan pada proses surat keluar dinas perhubungan. Mulai dari pembuatan surat, meminta nomor surat sesuai bidang, melakukan pencatatan pada buku arsip. Lalu surat dikeluarkan.

5.1.2 Analisis yang akan dibangun

Dalam membangun sistem perlu dibuat sebuah perancangan. Perancangan sistem ini ditujukan agar dalam pembentukan sistem dapat dihasilkan suatu aplikasi yang berfungsi secara optimal dan memberi kemudahan bagi penggunaanya.

5.1.2.1 Analisis Prosedur yang akan dibangun

1. Analisis BPMN Surat Masuk Yang Akan Dibangun

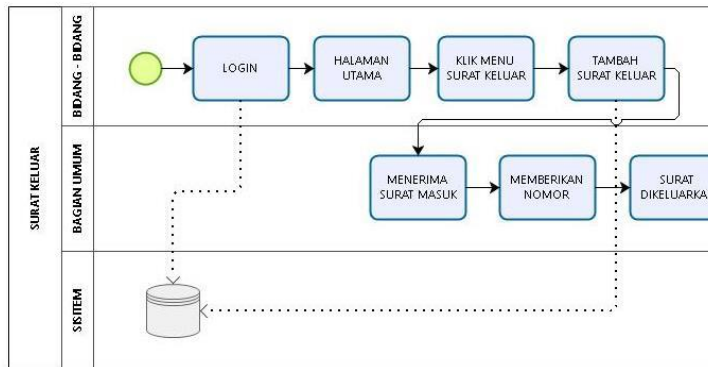


Gambar 5. 3 BPMN Surat Masuk Yang Akan Dibangun

Pada gambar 5.3 BPMN diatas adalah menjelaskan alur proses diagram sistem yang akan dibangun pada surat masuk yaitu bagian umum melakukan login terlebih dahulu, jika valid maka akan masuk ke halaman utama jika tidak valid maka akan kembali lagi ke halaman login. Lalu bagian umum menginput surat masuk pada menu surat masuk dan setelah itu

disimpan ke dalam database. Lalu bidang - bidang melakukan login untuk masuk ke halaman bidang. Dan bidang - bidang menerima surat masuk.

2. Analisis BPMN Surat Keluar Yang Akan Dibangun

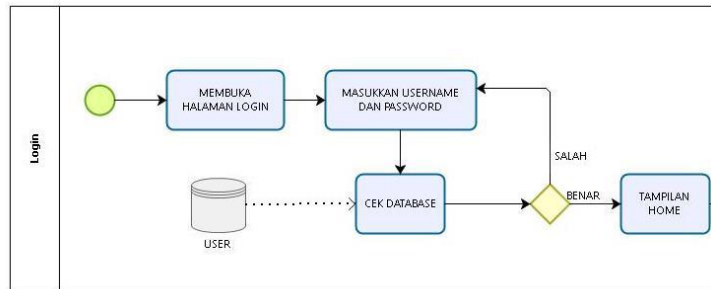


Gambar 5. 4 BPMN Surat Keluar Yang Akan Dibangun

Pada gambar 5.4 BPMN diatas adalah menjelaskan alur proses diagram sistem yang akan dibangun pada surat keluar yaitu bidang melakukan login terlebih dahulu, jika valid maka akan masuk ke halaman utama jika tidak

valid maka akan kembali lagi ke halaman login. Lalu bidang menginput surat keluar pada menu surat keluar dan setelah itu disimpan ke dalam database. Lalu bagian umum melakukan login untuk masuk ke halaman bagian umum. Dan bagian umum menerima surat keluar dan memberikan nomor, kemudian surat dikeluarkan.

3. Analisis BPMN Yang Akan Dibangun Pada *Login Medical*



Gambar 5. 5 BPMN Yang Akan Dibangun Pada Login

Pada prosedur *login*, *user* akan masuk dengan menggunakan *username* dan *password* lalu akan dicek oleh sistem apakah masukan yang diberi oleh *user* sesuai atau tidak. Bila sesuai maka sistem akan otomatis beralih ke halaman home sesuai dengan hak aksesnya.

Bila tidak sesuai maka *user* diharuskan mengisi kembali *username* dan *passwordnya*.

5.2 Perancangan

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi mengonfigurasi komponen – komponen perangkat lunak dan perangkat keras dari suatu sistem.

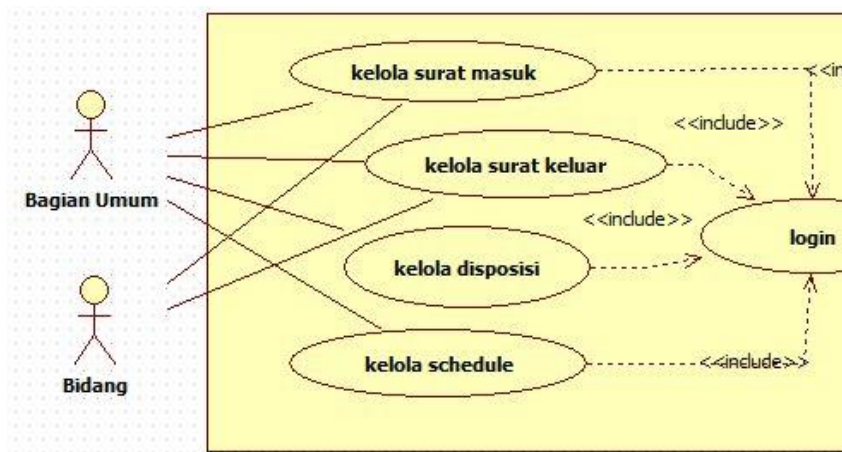
5.2.1 Perancangan UML

Perancangan merupakan penjelasan perancangan sistem/aplikasi/alat yang akan dibuat terdiri dari perancangan alir program (*flow chart*), algoritma, data, maupun perancangan input/output sistem/aplikasi/alat. Perancangan yang akan dibuat yaitu menggunakan object oriented UML terdiri dari *use case diagram*, *class diagram*,

sequence diagram, collaboration diagram, activity diagram, statechart diagram, component diagram, deployment diagram, objek diagram, structure program, dan perancangan user interface.

5.2.1.1 Use Case Diagram

Use case diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antara aktor dengan aktivitas yang terjadi pada sistem dan alat. berikut ini menggambarkan tentang proses yang terjadi secara umum dalam Sistem Tagihan :



Gambar 5. 6 Use Case Diagram Sistem Tagihan.

5.2.1.1.1 Definisi Aktor

Tabel 3.3 Definisi Aktor

No	Aktor	Keterangan
1	Bagian Umum	Melakukan kelola surat masuk, surat keluar, disposisi, dan jadwal / schedule.

2	Bidang - bidang	Melakukan kelola surat masuk dan surat keluar.
---	--------------------	--

5.2.1.1.2 Definisi *Use Case*

Tabel 3.4 Definisi *Use Case*

N o	Use Case	Deskripsi
1.	<i>Login</i>	Melakukan proses <i>login</i>
2.	Kelola Surat Masuk	Merupakan serangkaian aktivitas yang dilakukan untuk mengelola surat masuk.
3.	Kelola Surat Keluar	Merupakan serangkaian aktivitas yang dilakukan untuk mengelola surat keluar.
4.	Kelola Disposisi	Merupakan serangkaian aktivitas yang dilakukan untuk mengelola disposisi.
5.	Kelola Jadwal / Schedule	Merupakan serangkaian aktivitas yang dilakukan untuk mengelola jadwal dan schedule pegawai.

5.2.1.1.3 *Use case Diagram Skenario*

Use Case Skenario

mendesripsikan urutan langkah-langkah dalam proses sistem, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berikut ini penjelasan dari masing-masing skenario tersebut :

5.2.1.1.3.1 Skenario *Use Case Login Bagian Umum*

Tabel 3.5 Skenario *Use Case Login Bagian Umum*

Identifikasi	
Nomor	UC1
Nama	<i>Login bagian umum</i>
Tujuan	Memberikan hak akses <i>bagian</i> terhadap sistem dengan melakukan validasi terhadap <i>username</i> dan <i>password</i> yang dimasukan oleh <i>umum</i> .
Deskripsi	
Aktor	<i>Bagian Umum</i>
Skenario Utama	

Kondisi Awal	<i>Form Login</i> sudah tersedia
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan password	2. <i>Form Login</i> akan menampilkan <i>textbox username, password</i> dan <i>password</i> ditampilkan dalam bentuk kode ‘•’ pada layar untuk menjamin keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username, password</i> yang telah dimasukkan dengan menekan tombol <i>Login</i> .	4. Aplikasi melakukan validasi terhadap <i>username, password</i> yang telah dimasukkan oleh pengguna dengan melakukan pengecekan basis data.

Kondisi Akhir	Jika pada akhir interaksi <i>username</i> dan <i>password</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.

5.2.1.1.3.2 Skenario *Use Case Login Bidang* - bidang

Tabel 3.6 Skenario *Use Case Login Bidang - Bidang*

Identifikasi	
Nomor	UC2
Nama	<i>Login Bidang-bidang</i>
Tujuan	Memberikan hak akses <i>bidang-</i> terhadap sistem dengan melakukan validasi terhadap <i>username</i> dan <i>password</i> yang dimasukan oleh <i>bidang-bidang</i> .
Deskripsi	
Aktor	<i>Bidang-bidang</i>
Skenario Utama	
Kondisi Awal	<i>Form Login</i> sudah tersedia
Aksi Aktor	Reaksi Sistem

<p>1. Memasukkan <i>username</i> dan password</p>	<p>2. <i>Form Login</i> akan menampilkan <i>textbox username, password</i> dan <i>password</i> ditampilkan dalam bentuk kode ‘•’ pada layar untuk menjamin keamanan.</p>
<p>3. Aktor melakukan konfirmasi persetujuan terhadap <i>username, password</i> yang telah dimasukkan dengan menekan tombol <i>Login</i>.</p>	<p>4. Aplikasi melakukan validasi terhadap <i>username, password</i> yang telah dimasukkan oleh pengguna dengan melakukan pengecekan basis data.</p>

Kondisi Akhir	Jika pada akhir interaksi <i>username</i> dan <i>password</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.
Alur Alternatif kondisi akhir a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal login.	

5.2.1.1.3.3 Skenario *Use Case* Kelola Surat Masuk

Tabel 3.8 Skenario *Use Case* Kelola Data
User

Identifikasi	
Nomor	UC3
Nama	Kelola Surat Masuk
Tujuan	Melakukan Imputan Surat Masuk
Deskripsi	
Aktor	Bagian Umum dan Bidang-bidang
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama jika tidak <i>valid</i>

	akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih <i>link</i> ke halaman surat masuk.	2. Aplikasi menampilkan tampilan sesuai pilihan pengguna.
3. Pengelolaan isi <i>record</i> dengan memilih tombol tambah, ubah atau hapus.	4. Aplikasi menyimpan hasil kelola pada basis data tabel data masuk atau yang terkait..

Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel data masuk yang dilakukan pengelolaan mengalami perubahan.
----------------------	--

5.2.1.1.3.4 Skenario *Use Case* Kelola Surat Keluar

Tabel 3.9 Skenario *Use Case* Kelola Surat Keluar

Identifikasi	
Nomor	UC4
Nama	Kelola Surat Keluar
Tujuan	Melakukan Imputan Surat Keluar
Deskripsi	
Aktor	Bagian Umum dan Bidang-bidang
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .

Aksi Aktor	Reaksi Sistem
1. Pengguna memilih <i>link</i> ke halaman surat keluar.	2. Aplikasi menampilkan tampilan sesuai pilihan pengguna.
3. Pengelolaan isi <i>record</i> dengan memilih tombol tambah, ubah atau hapus.	4. Aplikasi menyimpan hasil ke pada basis data tabel data keluar atau yang terkait..
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel data keluar yang dilakukan pengelolaan mengalami perubahan.

5.2.1.1.3.5 Skenario *Use Case* Kelola Disposisi

Tabel 3.10 Skenario *Use Case* Kelola Disposisi

Identifikasi	
Nomor	UC5
Nama	Kelola Disposisi
Tujuan	Melakukan penentuan disposisi
Deskripsi	
Aktor	Bagian Umum
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem

1. Pengguna memilih <i>link</i> ke halaman disposisi.	2. Aplikasi menampilkan tampilan sesuai pilihan pengguna.
3. Pengelolaan isi <i>record</i> dengan memilih tombol tambah, ubah atau hapus.	4. Aplikasi menyimpan hasil kelola pada basis data tabel data disposisi atau yang terkait..
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel data disposisi yang dilakukan pengelolaan mengalami perubahan.

5.2.1.1.3.6 Skenario *Use Case* Kelola Jadwal / Schedule

Tabel 3. 11 Skenario *Use Case* Kelola Jadwal / Schedule

Identifikasi	
Nomor	UC6
Nama	Kelola Jadwal
Tujuan	Melakukan penentuan jadwal / schedule pegawai
Deskripsi	
Aktor	Bagian Umum
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama jika tidak <i>valid</i>

	akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih <i>link</i> ke halaman jadwal / schedule.	2. Aplikasi menampilkan tampilan sesuai pilihan pengguna.
3. Pengelolaan isi <i>record</i> dengan memilih tombol tambah, ubah atau hapus.	4. Aplikasi menyimpan hasil kelola pada basis data tabel da jadwal atau yang terkait..

Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel data jadwal yang dilakukan pengelolaan mengalami perubahan.
----------------------	---

5.2.2 Class Diagram

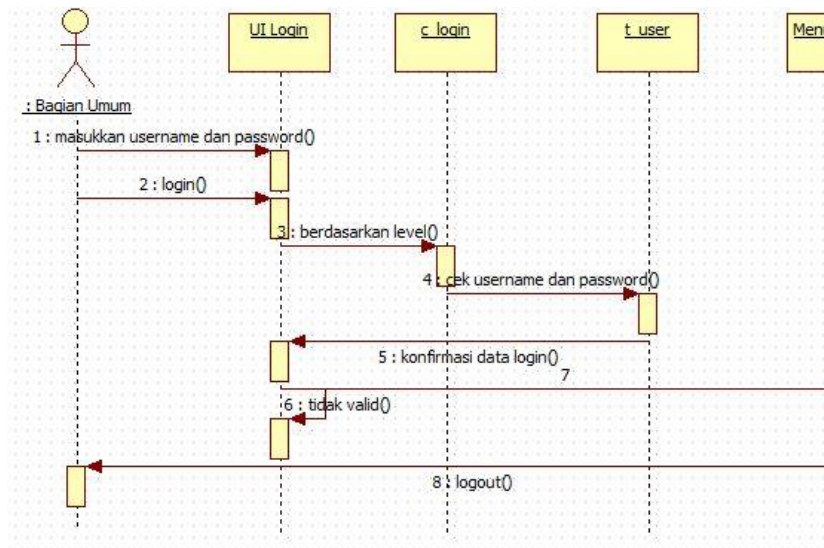
Class Diagram menggambarkan *class* dan hubungan antar *class* didalam sistem. *Class* digambarkan dengan sebuah kotak dibagi menjadi tiga bagian. Bagian paling atas diisikan nama *class*, bagian tengah diisikan *variabel* yang dimiliki *class*, dan bagian bawah diisikan *method-method* dari *class*.

Dalam proses pembuatan aplikasi ini *class diagram* yang akan digunakan adalah sebagai berikut :

5.2.3 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram, pada diagram ini, dimensi vertikal merepresentasikan waktu.

5.2.3.1 *Sequence Diagram Login Bagian Umum*

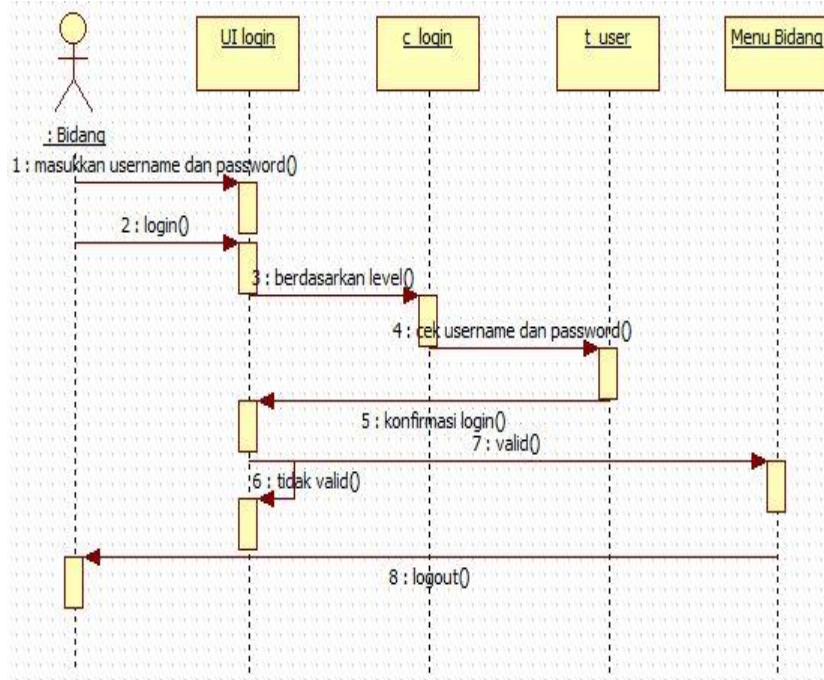


Gambar 5.9 *Sequence Diagram Login Bagian Umum*

Pada *sequence diagram* ini menjelaskan proses *login*. Bagian Umum mulai menjalankan aplikasi akan tampil *form*

login. Kemudian bagian umum memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di table *user*. Jika sesuai maka akan mendapatkan tampilan halaman utama bagian umum.

5.2.3.2 Sequence Diagram Login Bidang-bidang

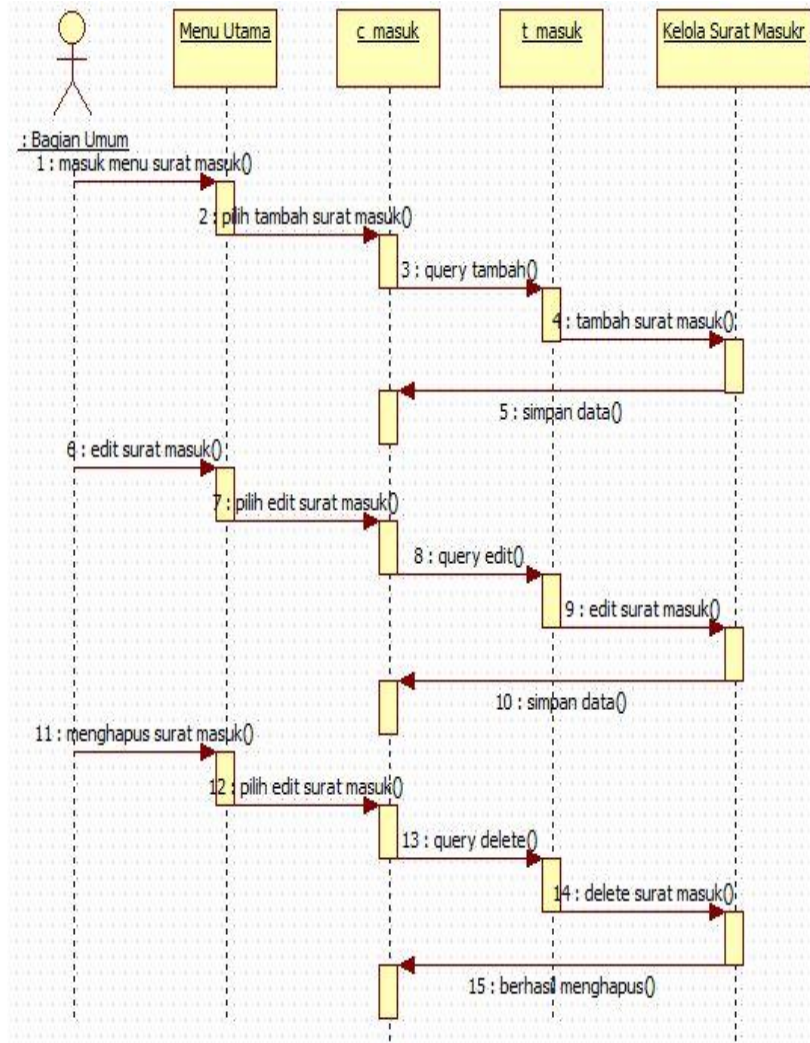


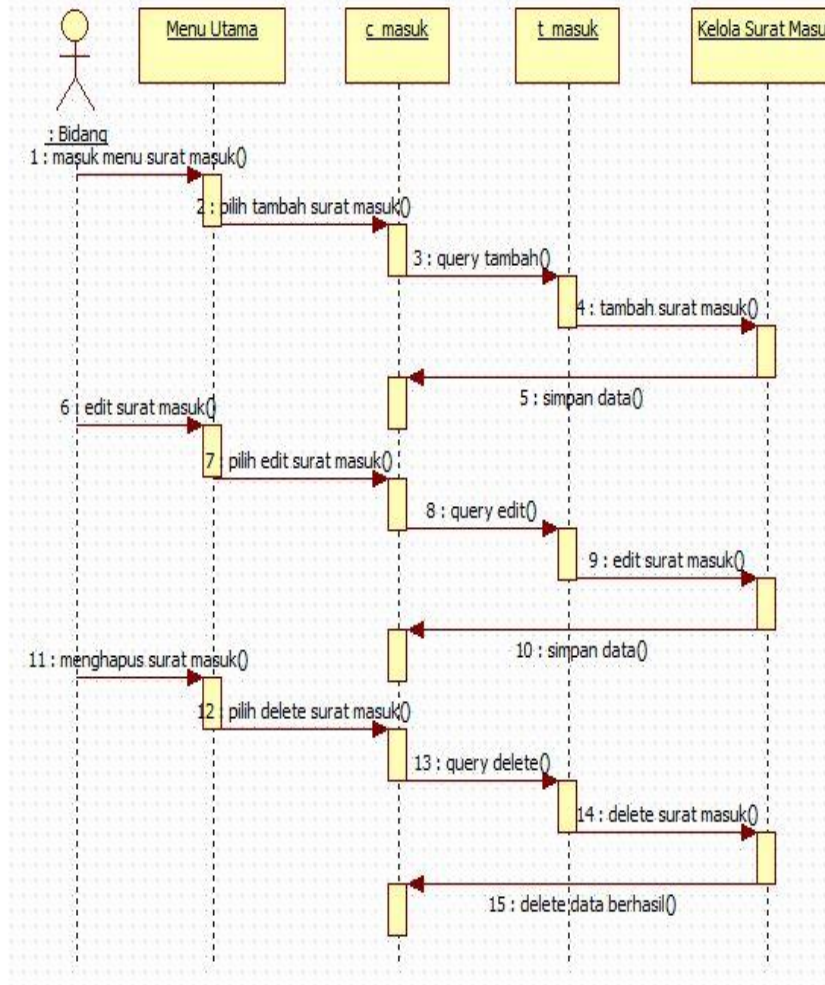
Gambar 5.10 Sequence Diagram Login
Bidang-bidang

Pada *sequence diagram* ini menjelaskan proses *login*. Bidang mulai

menjalankan aplikasi akan tampil *form login*. Kemudian bidang memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di table *user*. Jika sesuai maka akan mendapatkan tampilan halaman utama bidang.

5.2.3.4 *Sequence Diagram* Kelola Surat Masuk

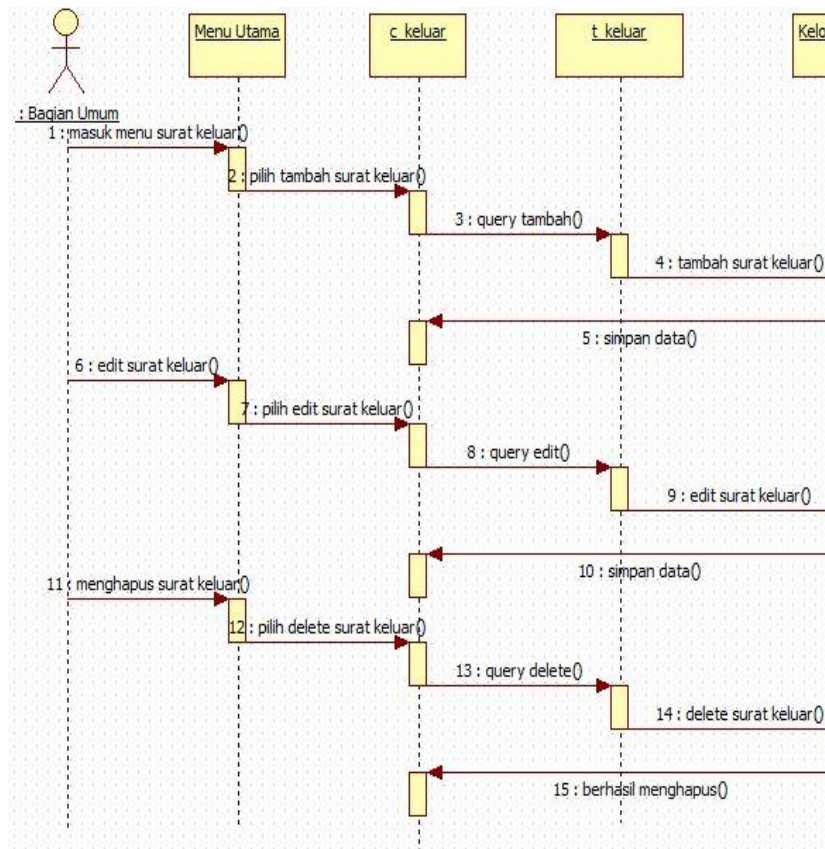


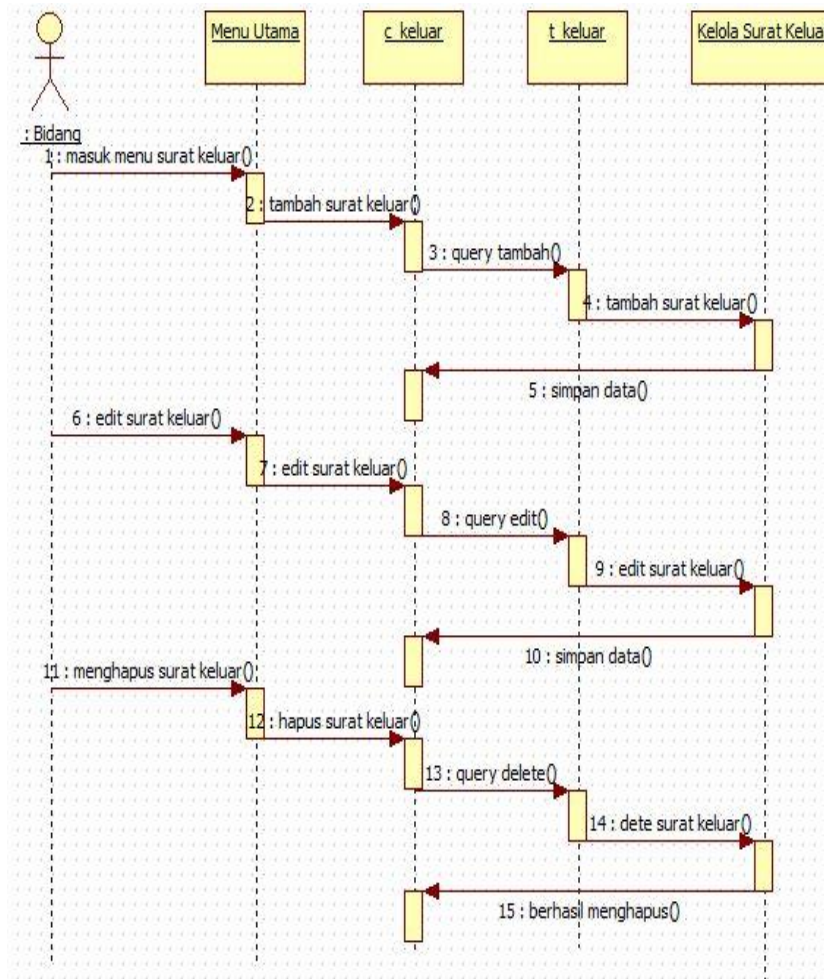


Gambar 5.12 *Sequence Diagram* Kelola Surat
Masuk

Pada *sequence diagram* ini menjelaskan proses mengelola data *user* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu kelola data user, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, bagian umum dan bidang dapat melakukan operasi tambah data, edit data dan hapus data. Selanjutnya sistem akan menyimpan kembali pada *database*.

5.2.3.5 Sequence Diagram Kelola Surat Keluar





Gambar 5.13 *Sequence Diagram* Kelola Surat keluar

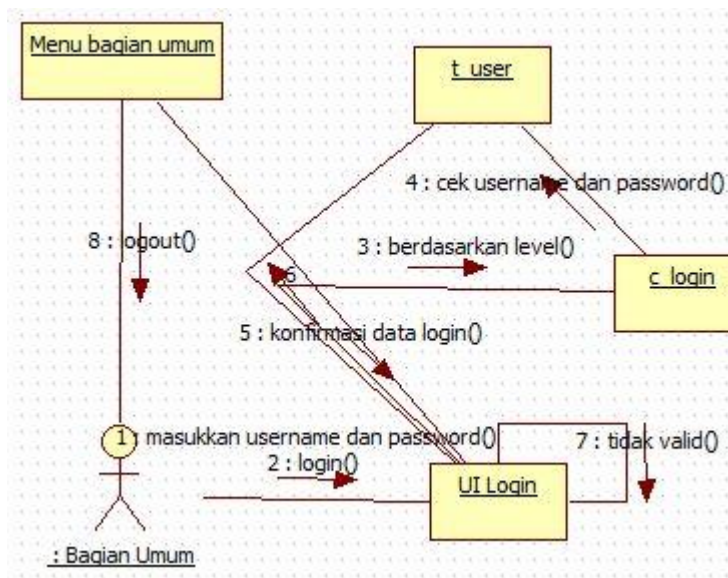
Pada *sequence diagram* ini menjelaskan proses mengelola data *user* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu kelola data user, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, bagian umum dan bidang dapat melakukan operasi tambah data, edit data dan hapus data. Selanjutnya sistem akan menyimpan kembali pada *database*.

5.2.4 Collaboration Diagram

Collaboration Diagram adalah suatu diagram yang memperlihatkan atau menampilkan pengorganisasian interaksi yang terdapat disekitar objek (seperti halnya

sequence diagram) dan hubungannya terhadap yang lainnya. Collaboration Diagram lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan/message.

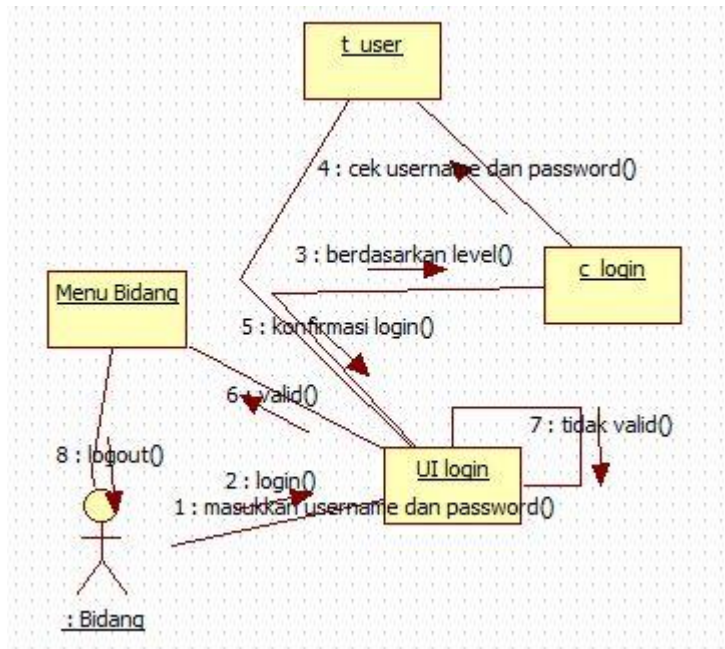
5.2.4.1 Collaboration Diagram Login Bagian Umum



Gambar 5.18 *Collaboration Diagram Login*
Bagian Umum

Pada *collaboration diagram* ini menjelaskan proses *login*. Bagian umum mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian bagian umum memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di tabel *user*. Jika sesuai maka akan mendapatkan tampilan *form* halaman utama bagian umum.

5.2.4.2 Collaboration Diagram Login
Bidang-bidang

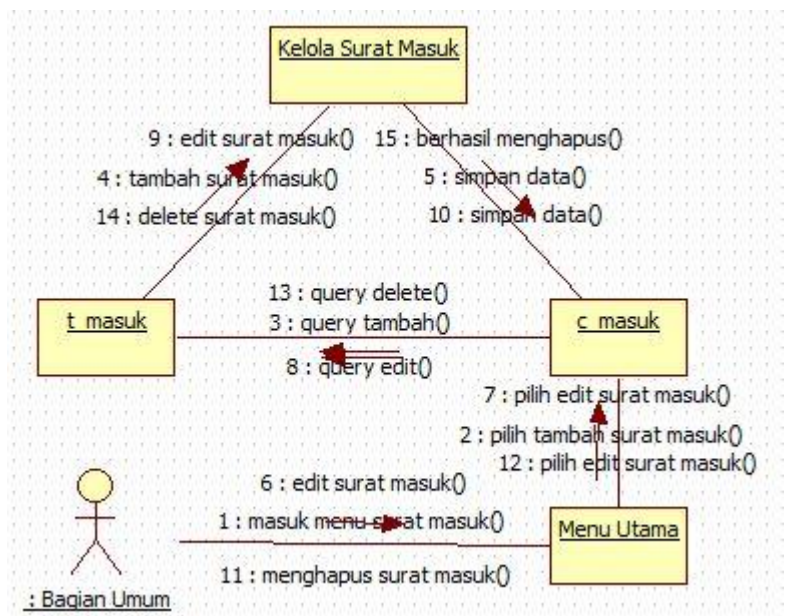


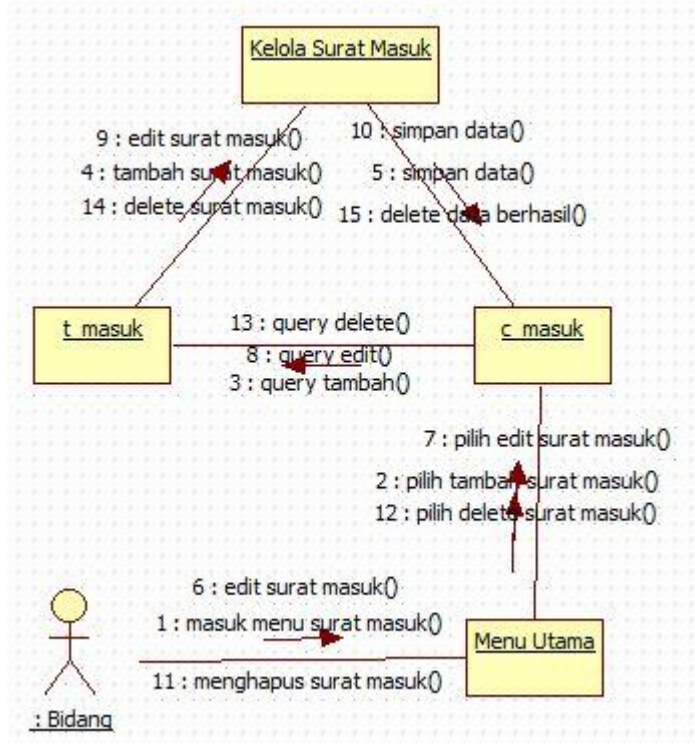
Gambar 5.19 *Collaboration Diagram Login*
Bidang

Pada *collaboration diagram* ini menjelaskan proses *login*. Bidang mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian bidang memasukkan

username dan password, pada interface login data yang di input akan di cek di tabel user. Jika sesuai maka akan mendapatkan tampilan form halaman utama bidang.

5.2.4.4 Collaboration Diagram *Kelola Surat Masuk*



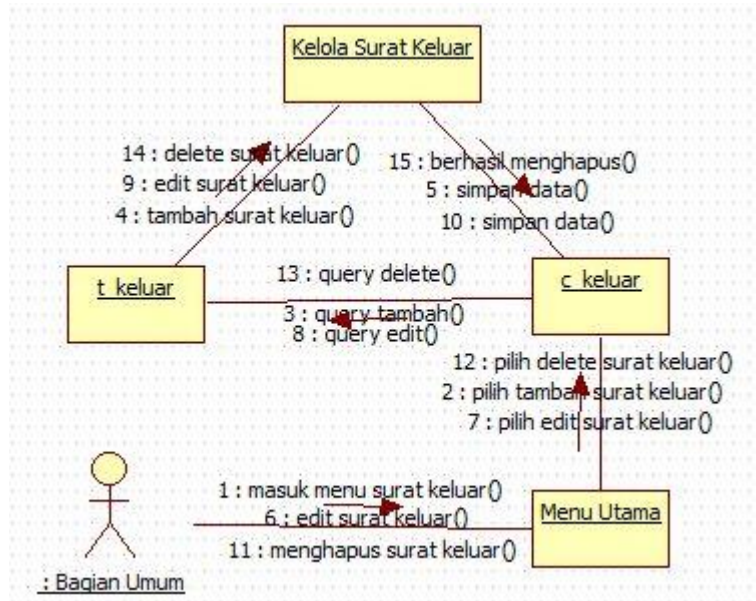


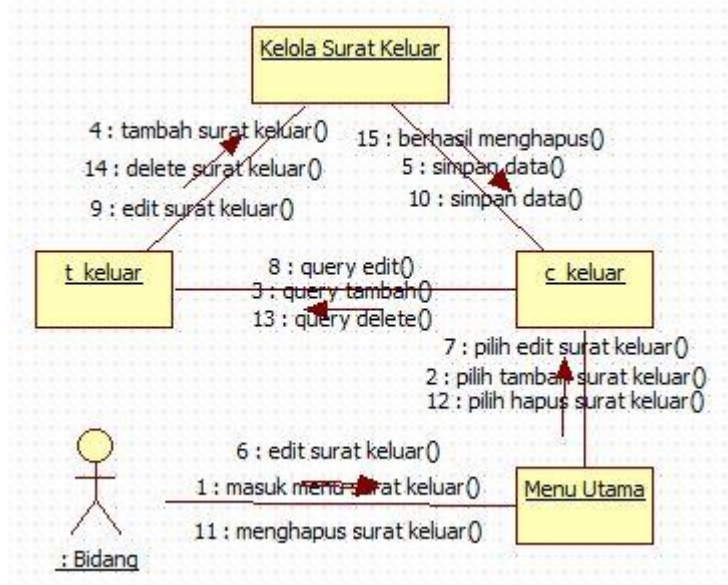
Gambar 5.21 *Collaboration Diagram* Kelola Surat Masuk

Pada *collaboration diagram* ini menjelaskan proses mengelola surat masuk.

Bagian umum dan bidang melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu surat masuk, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *user* dapat melakukan operasi create, edit dan delete data. Selanjutnya sistem akan menyimpan perubahan tersebut.

5.2.4.5 Collaboration Diagram *Kelola Surat Keluar*





Gambar 5.22 *Collaboration Diagram* Kelola Surat Keluar

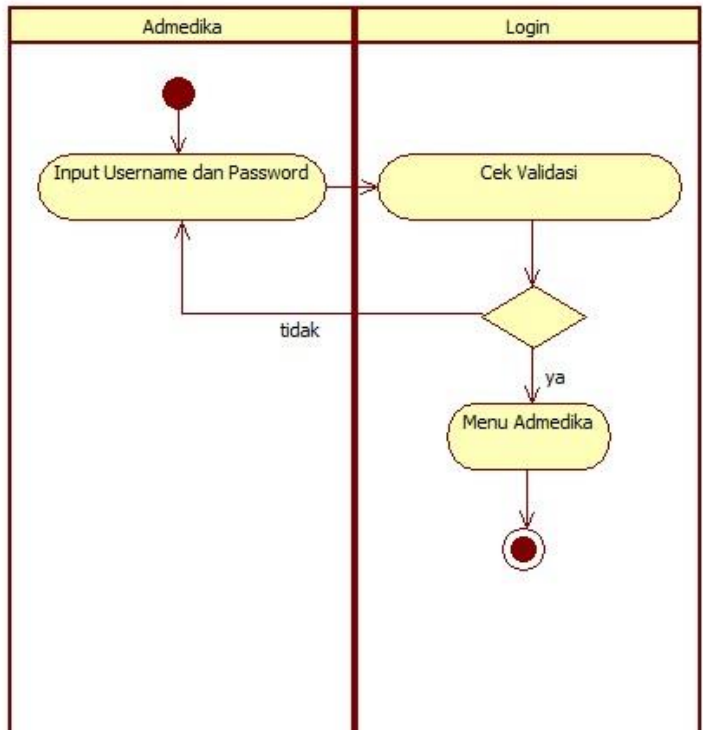
Pada *collaboration diagram* ini menjelaskan proses mengelola surat keluar. Bagian umum dan bidang melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau

bidang. Untuk memulai kelola ini, masuk ke menu surat keluar, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *user* dapat melakukan operasi create, edit dan delete data. Selanjutnya sistem akan menyimpan perubahan tersebut.

5.2.5 Activity Diagram

Activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

5.2.5.1 Activity Diagram Login Admedika

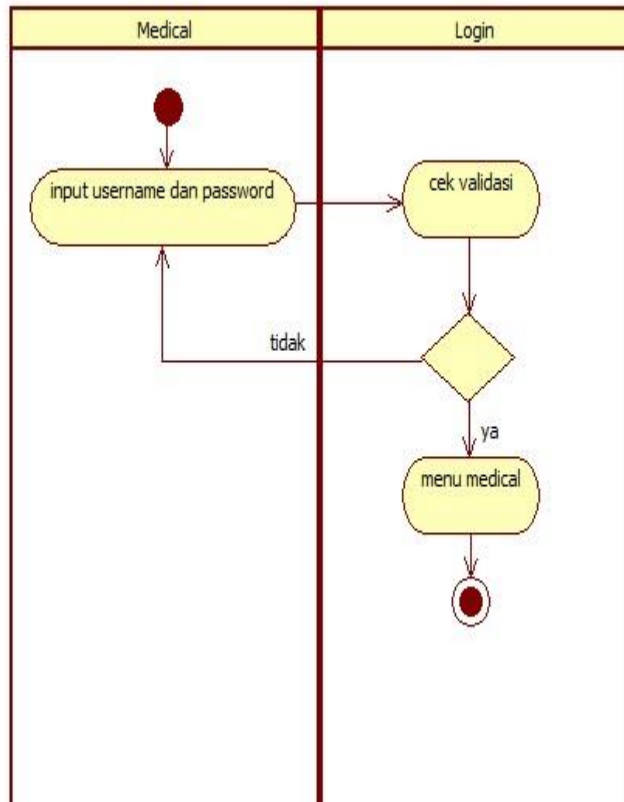


Gambar 5.27 Activity Diagram Login
Admedika

Actor yang dapat login adalah admedika.
Admedika mulai dengan memasukkan

username dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil halaman utama admedika, tetapi jika tidak *valid* akan kembali ke *form login*.

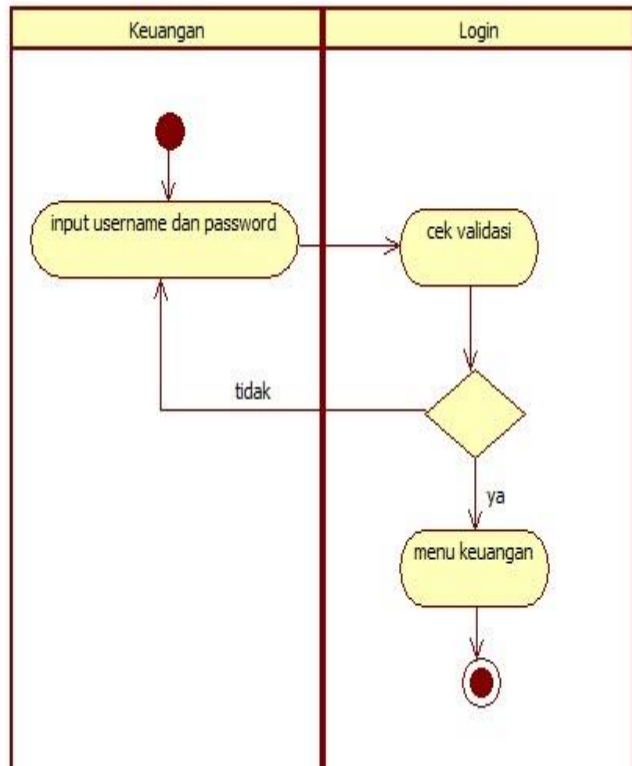
5.2.5.2 Activity Diagram Login Medical



Gambar 5.28 Activity Diagram Login
Medical

Actor yang dapat *login* adalah *medical*. *Medical* mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil halaman utama *medical*, tetapi jika tidak *valid* akan kembali ke *form login*.

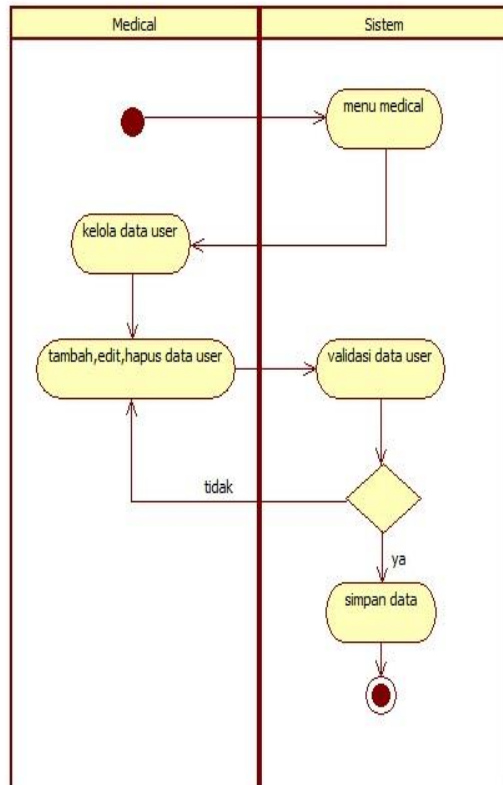
5.2.5.3 Activity Diagram Login Keuangan



Gambar 5.29 Activity Diagram Login Keuangan

Actor yang dapat *login* adalah keuangan. Keuangan mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil halaman utama keuangan, tetapi jika tidak *valid* akan kembali ke *form login*.

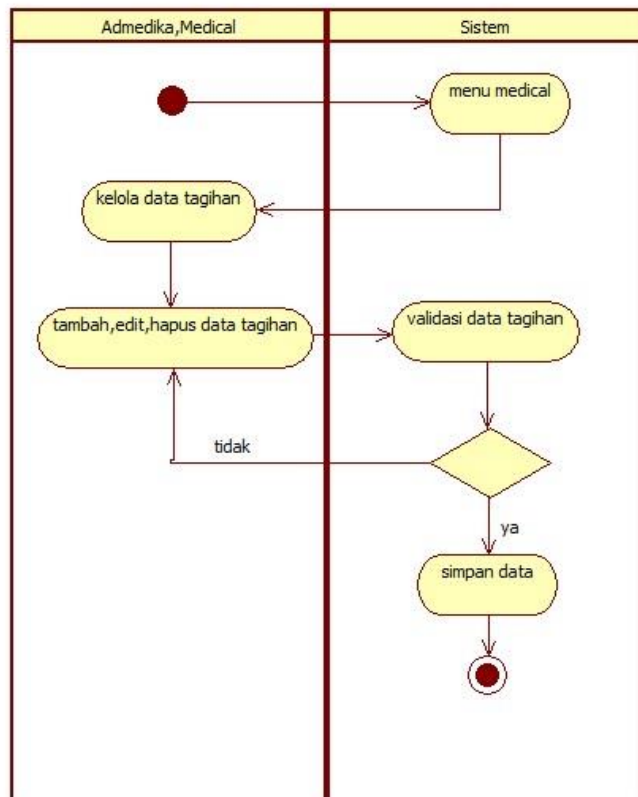
5.2.5.4 Activity Diagram Kelola Data User



Gambar 5.30 Activity Diagram Kelola Data User

Pada *activity diagram* ini menjelaskan proses mengelola data user. *Medical* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama *medical*. Untuk memulai kelola ini, masuk ke menu kelola data user, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *actor* dapat melakukan operasi create, edit data dan delete data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

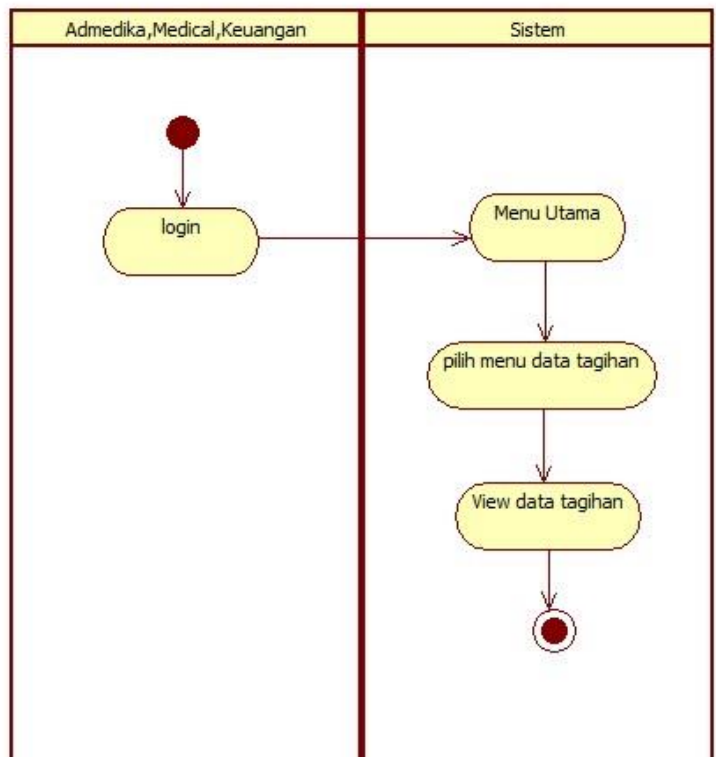
5.2.5.5 Activity Diagram Kelola Data Tagihan



Gambar 5.31 *Activity Diagram* Kelola Data
Tagihan

Pada *activity diagram* ini menjelaskan proses mengelola data tagihan. *Medical* dan *admedika* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama *medical* dan *admedika*. Untuk memulai kelola ini, masuk ke menu kelola data tagihan, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *actor* dapat melakukan operasi *create*, *edit* data dan *delete* data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

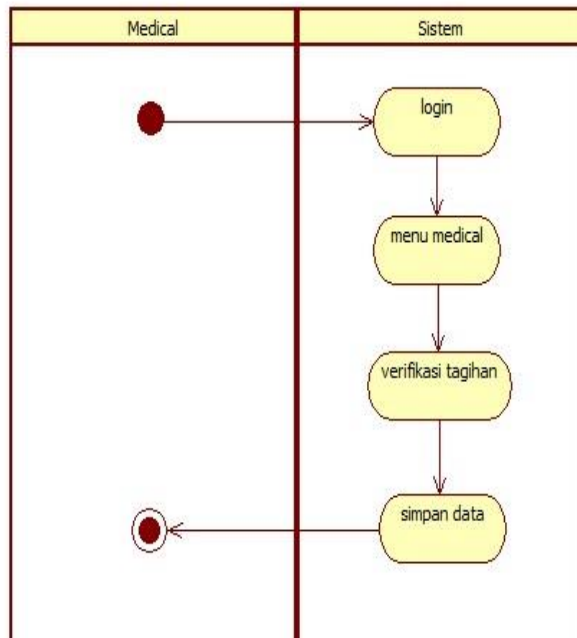
5.2.5.6 Activity Diagram View Data Tagihan



Gambar 5.32 Activity Diagram View Data Tagihan

Pada *activity diagram* ini menjelaskan proses melihat data tagihan. Aktor memilih menu data pesanan. Namun sebelumnya lakukan login terlebih dahulu. Selanjutnya sistem akan melakukan koneksi *database*. Selanjutnya system akan memvalidasi. Selanjutnya sistem memuat tampilan data tagihan.

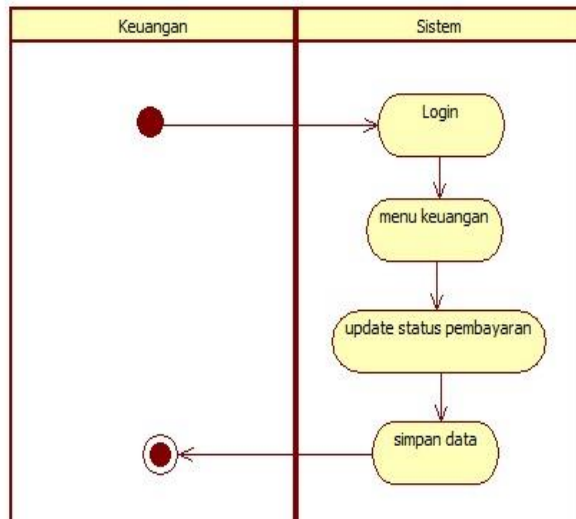
5.2.5.7 Activity Diagram Verifikasi Tagihan



Gambar 5.33 Activity Diagram Verifikasi Tagihan

Setelah login, medical memilih menu verifikasi tagihan. Kemudian sistem akan menampilkan verifikasi tagihan. Aktor akan melakukan proses verifikasi tagihan, selanjutnya di sistem akan disimpan.

5.2.5.8 Activity Diagram Update Status Pembayaran

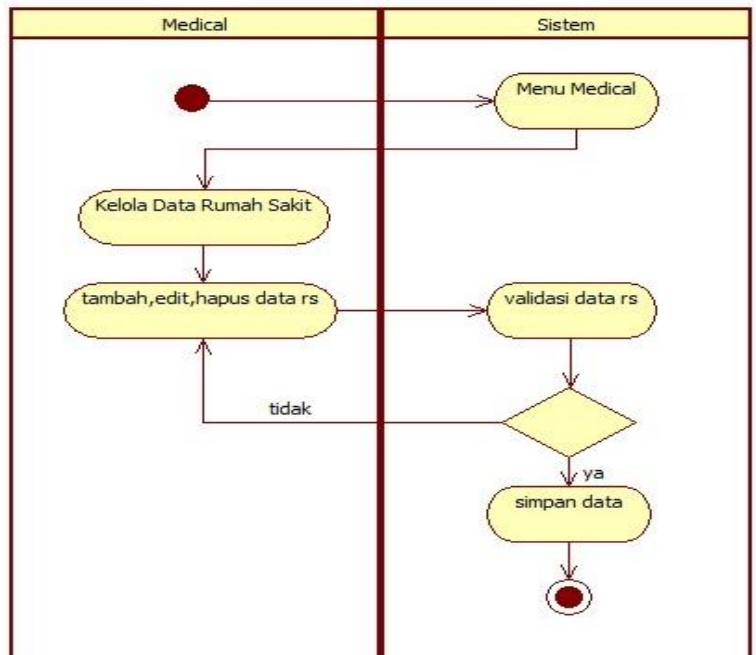


Gambar 5.34 Activity Diagram Update Status Pembayaran

Setelah *login*, keuangan memilih menu update status tagihan. Kemudian sistem akan menampilkan update status pembayaran. Aktor akan melakukan proses *update* status

pembayaran, selanjutnya di sistem akan disimpan.

5.2.5.9 Activity Diagram Kelola Data Rumah Sakit



Gambar 5.35 Activity Diagram Kelola Data Rumah Sakit

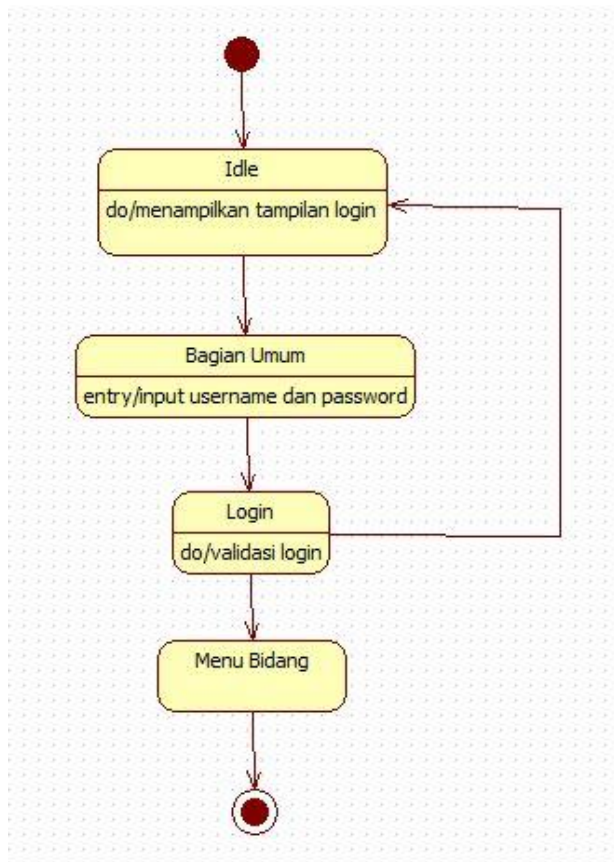
Pada *activity diagram* ini menjelaskan proses mengelola data rumah sakit. *Medical*

melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama *medical*. Untuk memulai kelola ini, masuk ke menu kelola data rumah sakit, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *actor* dapat melakukan operasi create, edit data dan delete data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

5.2.6 Statechart Diagram

Statechart Diagram menggambarkan transisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu objek pada *system* sebagai akibat dari stimulan yang diterima *statechart diagram* mendeskripsikan bagaimana suatu objek mengalami perubahan status adanya *trigger* dari *event-event*. Menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek.

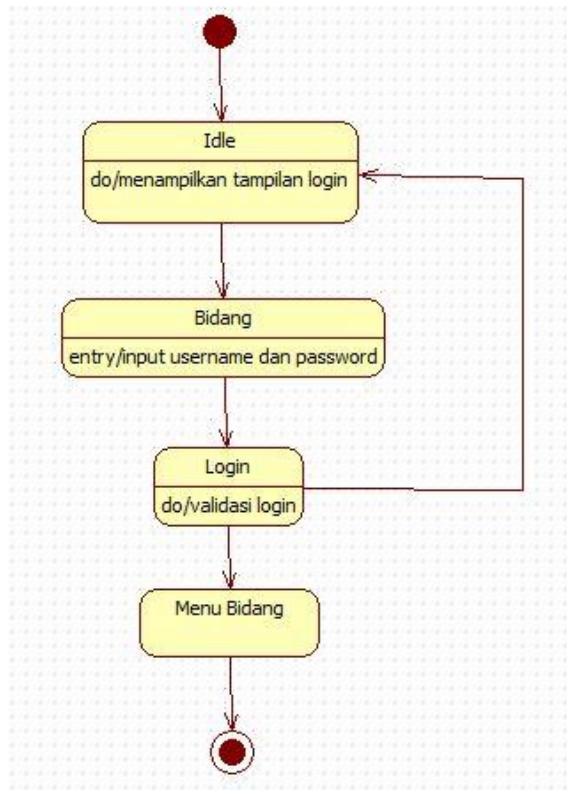
5.2.6.1 Statechart Diagram Login *Bagian Umum*



Gambar 5.36 *Statechart Diagram Login*
Admedika

Bagian umum masuk ke menu *login* menginputkan *username* dan *password* jika *valid* maka akan masuk ke halaman utama bagian umum.

5.2.6.2 *Statechart Diagram Login* Bidang-bidang

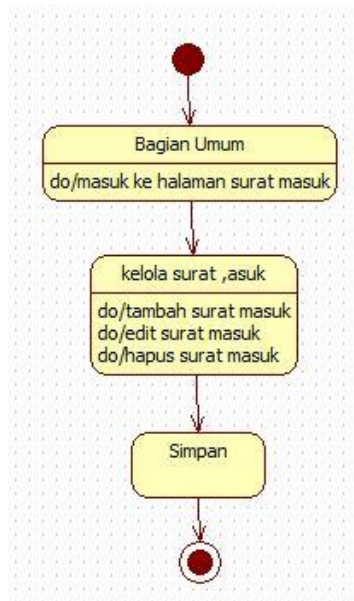


Gambar 5.37 *Statechart Diagram Login*
Medical

Bidang masuk ke menu *login*
menginputkan *username* dan *password* jika

valid maka akan masuk ke halaman utama bidang.

5.2.6.3 *Statechart Diagram* Kelola Surat Masuk

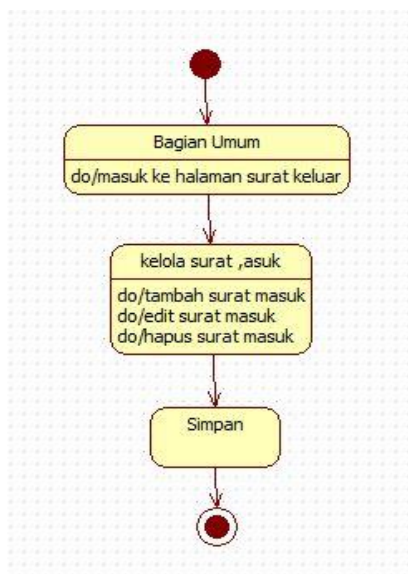




Gambar 5.39 *Statechart Diagram* Kelola Surat Masuk

Bagian umum dan Bidang mengelola data masuk yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.6.4 *Statechart Diagram* Kelola Data Surat Keluar





Gambar 5.40 *Statechart Diagram* Kelola Surat Keluar

Bagian Umum dan Bidang mengelola data keluar yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.7 Component Diagram

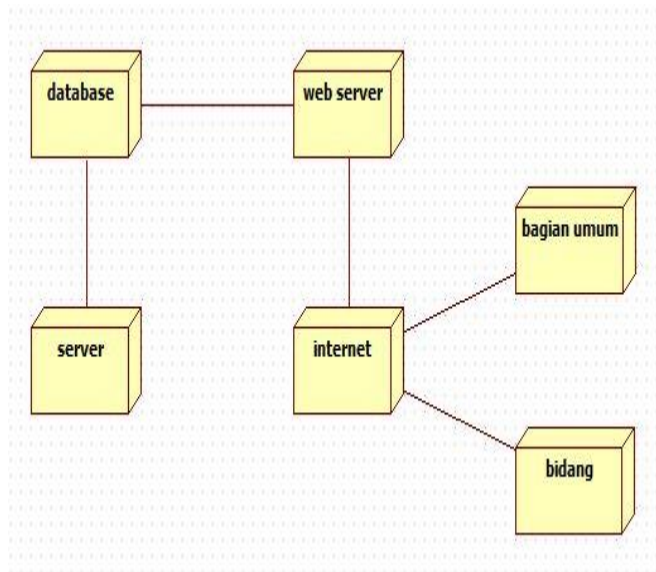
Component diagram yaitu salah satu jenis *diagram* pada *UML* yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa *file* data atau *.exe*, *source code*, *table*, dokumen, dll.

Berikut ini merupakan *component diagram* yang menjelaskan semua komponen yang ada pada sistem.

5.2.8 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal seperti sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan

hardware, sistem *client/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi



Gambar 5.46 *Deployment Diagram*

5.2.9 Perancangan Antar Muka

5.2.9.1 *User Interface Login* Bagian Umum/Bidang-bidang



DINAS PERHUBUNGAN PROVINSI JAWA BARAT

The image shows a login interface within a rectangular frame. At the top, the text 'DINAS PERHUBUNGAN PROVINSI JAWA BARAT' is centered. Below this, there are two horizontal input fields, one above the other, for username and password. At the bottom, there is a rectangular button with the text 'Login' centered on it.

Gambar 5.47 User Interface Login

Keterangan Gambar :

Textbox 1 : *Input Username*

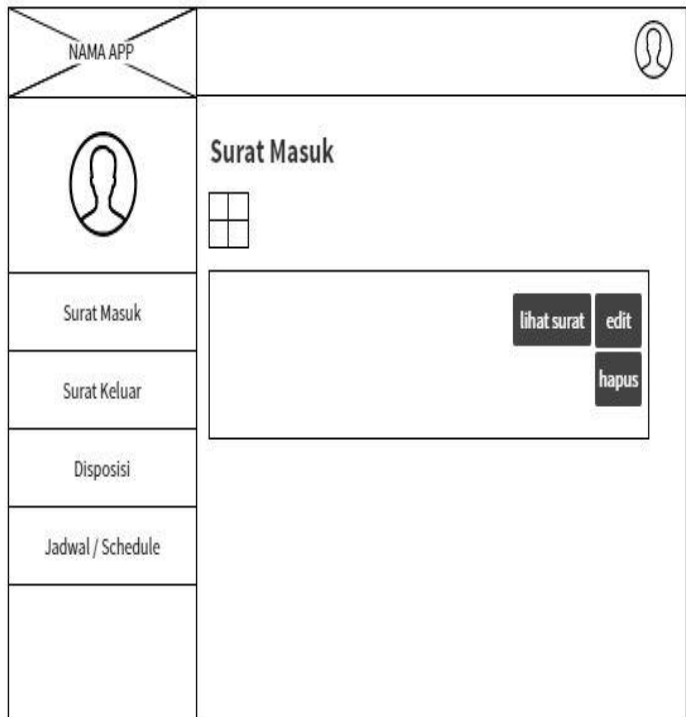
Textbox 2 : *Input Password*

Button 1 : *Tombol Login*

Label 1 : Dinas Perhubungan

Provinsi Jawa Barat

5.2.9.2 User Interface Halaman Surat Masuk

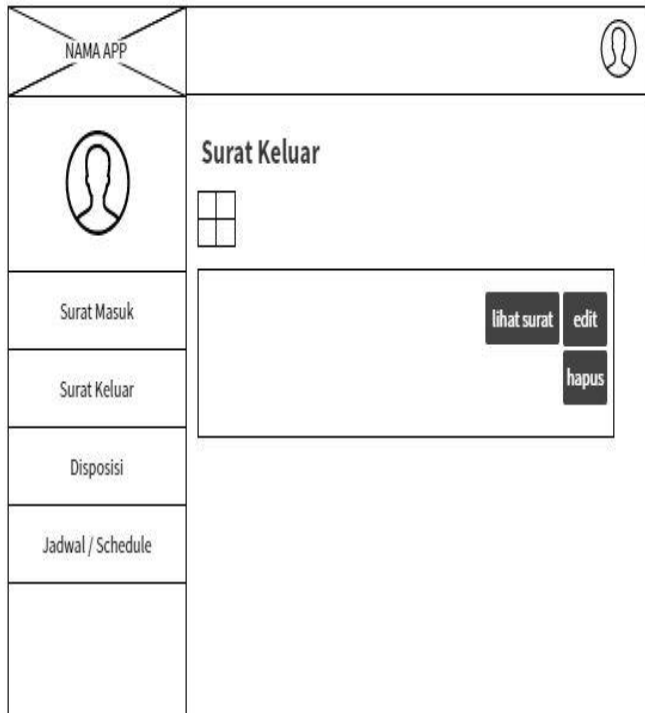


Gambar 5.48 *User Interface* Halaman Surat
Masuk

Keterangan Gambar :

<i>Label 1</i>	: Surat Masuk
<i>Label 2</i>	: Surat Keluar
<i>Label 3</i>	: Disposisi
<i>Label 4</i>	: Jadwal / Schedule
<i>Button</i> Lihat Surat	: Tambah, Edit, Hapus,
<i>Profile Picture</i>	
<i>Logo</i>	

5.2.9.3 User Interface Halaman Surat Keluar



Gambar 5.49 *User Interface* Halaman Surat Keluar

Keterangan Gambar :

<i>Label 1</i>	: Surat Masuk
<i>Label 2</i>	: Surat Keluar
<i>Label 3</i>	: Disposisi
<i>Label 4</i>	: Jadwal / Schedule
<i>Button</i> Lihat Surat	: Tambah, Edit, Hapus,
<i>Profile Picture</i>	
<i>Logo</i>	

5.2.9.5 *User Interface* Halaman Tambah Surat Masuk

[illegible]

Gambar 5.51 *User Interface* Halaman
Tambah Surat Masuk

Keterangan Gambar :

<i>Textbox 1</i>	: Nomor Surat
<i>Datepicker</i>	: Tanggal Surat
<i>Textbox 2</i>	: Sifat Surat
<i>Textbox 3</i>	: Isi Ringkas
<i>Textbox 4</i>	: Surat Dari
<i>Textbox 5</i>	: Kepada
<i>Textbox 6</i>	: Diteruskan
<i>Textbox 7</i>	: Pengolah
<i>Button</i>	: Tambah

5.2.9.6 User Interface Halaman Tambah Surat Keluar

NAMA APP	
	Tambah Surat Masuk
Surat Masuk	<input type="text"/> <div> <input type="text" value="12 May 2016"/> <input type="button" value="📅"/> </div>
Surat Keluar	<input type="text"/>
Disposisi	<input type="text"/>
Jadwal / Schedule	<input type="text"/>
	<input type="button" value="Tambah"/>

Gambar 5.52 *User Interface* Halaman
Tambah Data *User*

Keterangan Gambar :

Textbox 1 : Nomor Surat

Datepicker : Tanggal Surat

Textbox 2 : Sifat Surat

Textbox 3 : Prihal

Textbox 4 : Surat Dari

Textbox 5 : Kepada

Button : Tambah

BAB VI

IMPLEMENTASI DAN PENGUJIAN

Implementasi adalah sistem/aplikasi/alat yang dibuat dengan merinci komponen-komponen pendukung berupa program, Lingkungan Implementasi, Tampilan Antarmuka, Petunjuk Pemakaian, Petunjuk Instalasi.

Pengujian adalah cara untuk mengetahui apakah sistem/aplikasi/alat yang dibuat sesuai dengan rancangan dan menuliskan hasil ujinya.

6.1 Lingkungan Implementasi

Perancangan aplikasi ini dapat berjalan dengan baik karena didukung oleh perangkat pendukung yaitu perangkat lunak dan perangkat keras.

6.1.1 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut :

Tabel 6. 1 Deskripsi Perangkat Lunak

No.	Jenis		Keterangan
1	Sistem Operasi	:	Microsoft Windows 10
2	Bahasa Pemrograman	:	PHP

3	<i>Database</i>	:	MySQL
4	<i>Framework</i>	:	CodeIgniter

6.1.2 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut :

Tabel 6. 2 Deskripsi Perangkat Keras

No.	Jenis		Keterangan
1	Prosesor	:	Intel® Core™ i5
2	Memori (RAM)	:	4 GB
3	<i>Monitor</i>	:	LCD 14,0 inci
4	<i>Mouse dan keyboard</i>	:	<i>Standard</i>

6.2 Pengujian Dan Hasil Pengujian

6.2.1 Identifikasi dan Rencana Pengujian

Tabel 6. 3 Identifikasi Dan Rencana

Kelas Uji	Butir Uji	Identifikasi SKPL	Tingkat Pengujian	Jenis Pengujian	Jadwal
Validasi Login	Validasi Login Pelanggan	UC01	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
Tampil Form Utama	Tampil Menu Surat Masuk	UC02	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
	Tampil Menu Surat Keluar	UC03	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
	Tampil Menu Disposisi	UC04	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
	Tampil Menu Jadwal / Schedule	UC05	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
Logout	Logout Bagian Umum dan Bidang	UC07	Pengujian Sistem	<i>Black Box</i>	25 – 09-2017

Pengujian

6.2.1 Deskripsi dan Hasil Uji

6.2.2.1 Pengujian Validasi Login

Tabel 6. 4 Pengujian Validasi Login

Iden tifik asi	Deskri psi	Prosed ur Penguji an	Masukan	Keluaran Yang Diharapk an	Kriteria Evaluasi Hasil	Has il	
UC 01	Mengu ji Login.	Klik Button OK	Masukkan User Name : Bagian Umum Password :*****	Tampil Form Utama Bagian Umum atau Bidang.	Tampil Form Utama Bagian Umum atau Bidang.	OK	D r

6.2.2.2 Pengujian Tampilan Utama
Pelanggan

Tabel 6. 5 Pengujian Tampilan Utama
Pelanggan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil	
UC02	Menguji Menu Surat Masuk	Tampil Data	- Data Surat Masuk	- Data yang ada di database muncul	Tampil di Halaman Surat Masuk	OK	
UC03	Menguji Menu Surat Keluar	Tampil Data	-Data Surat Keluar	- Data yang ada di database muncul	Tampil di Halaman Surat Keluar	OK	

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil	Revisi
UC04	Menguji Menu Delete Surat Masuk dan Keluar	Delete	-Data Surat	- Proses modifikasi berhasil kembali ke halaman Surat	Tampil halaman Surat	OK	Revisi
UC05	Menguji Menu Disposisi	Tampil Data	Data Disposisi	- Data yang ada di database muncul	Tampil di Halaman Disposisi	OK	Revisi
UC06	Menguji Menu Jadwal / Schedule	Tampil Data	-Data Jadwal	- Data yang ada di database muncul	Tampil di Halaman Disposisi	OK	Revisi

6.2.2.3 Pengujian Logout

Tabel 6. 6 Pengujian Logout

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil	Kesimpulan
SKPL 11	Menguji Logout	Klik menu logout lalu pilih logout	-	Tampil Form Bidang Umum atau Bidang	Tampil Form utama	OK	Diterima