

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.

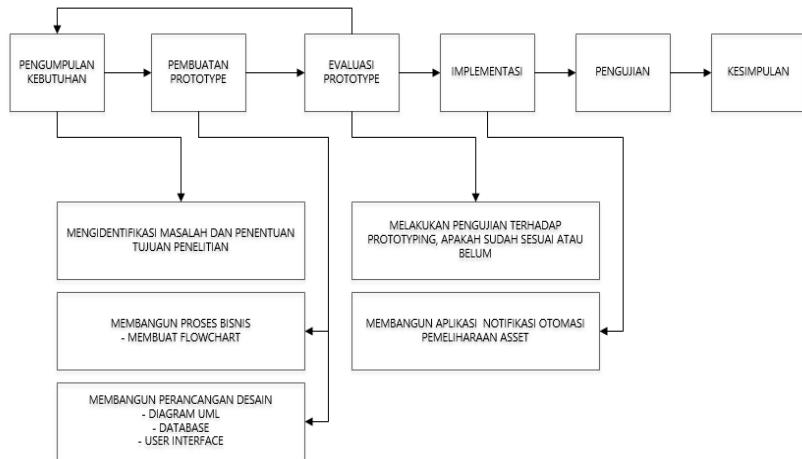
Pada saat RAD diimplementasikan, maka para pemakai bisa menjadi bagian dari keseluruhan proses pengembangan sistem dengan bertindak sebagai pengambil keputusan pada setiap tahapan pengembangan. RAD bisa menghasilkan suatu sistem dengan cepat karena sistem yang dikembangkan dapat

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

BAB III

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 IV

ANALISIS DAN PERANCANGAN SISTEM

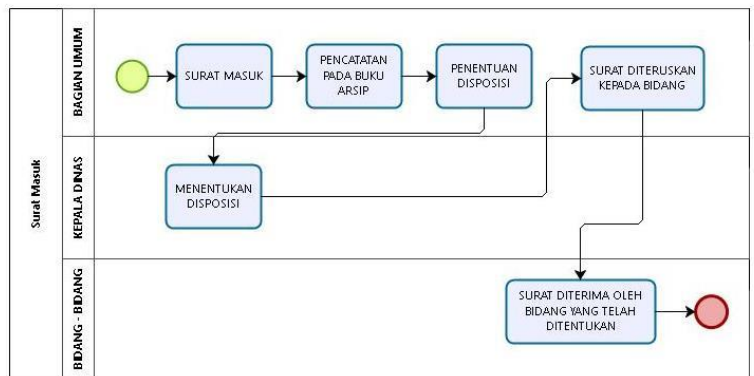
4.1 Analisis dan Perancangan Sistem

4.1.1 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

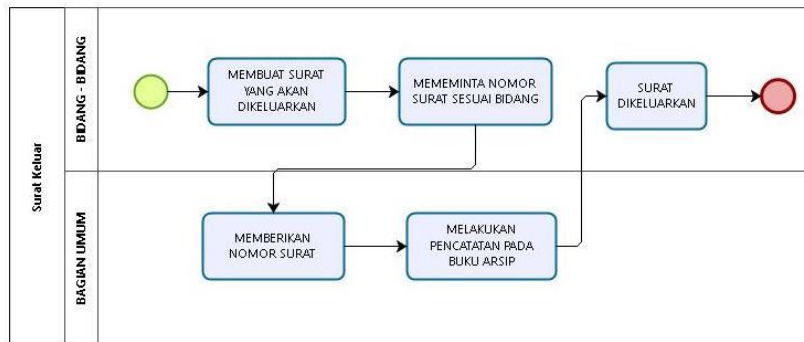


Gambar 5. 1 BPMN Prosedur Surat Masuk yang Sedang Berjalan

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

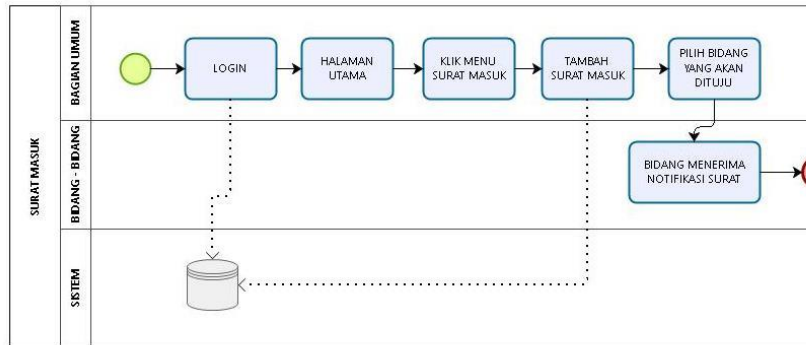
Pada gambar 5.2 BPMN diatas adalah penjelasan alur yang berjalan pada proses surat keluar dinas perhubungan. Mulai dari pembuatan surat, meminta nomor surat sesuai bidang, melakukan pencatatan pada buku arsip. Lalu surat dikeluarkan.

4.1.2 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

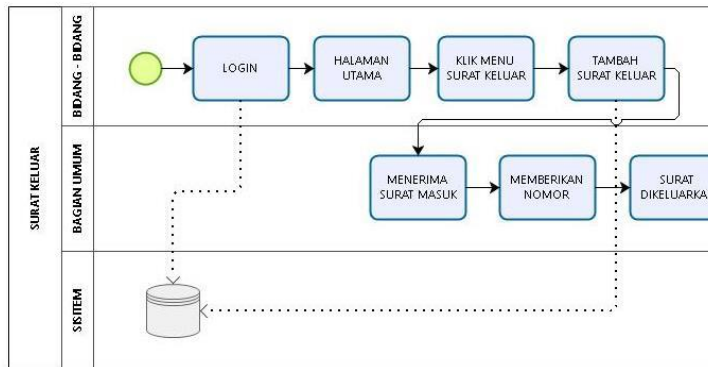


Gambar 5. 3 BPMN Surat Masuk Yang Akan Dibangun

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

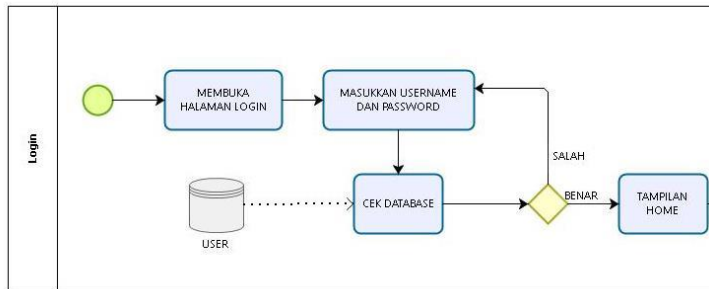


Gambar 5. 4 BPMN Surat Keluar Yang Akan Dibangun

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.

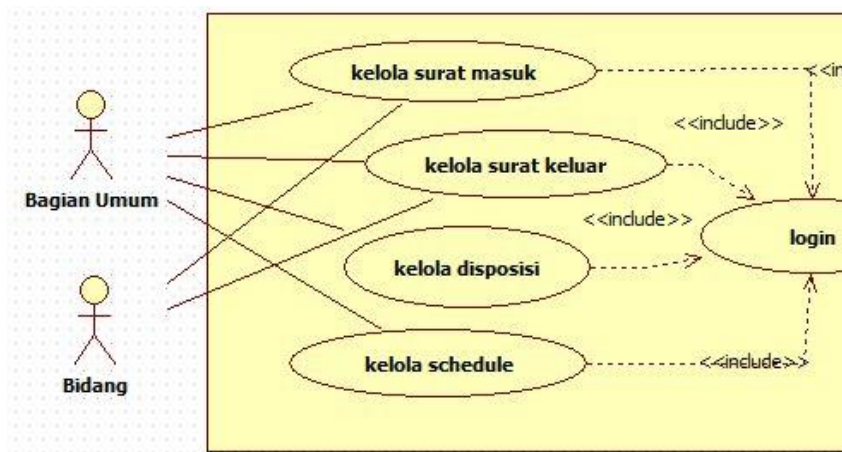
5.2.1 Perancangan UML

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 surat.
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>user</i> memasukkan <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

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.
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

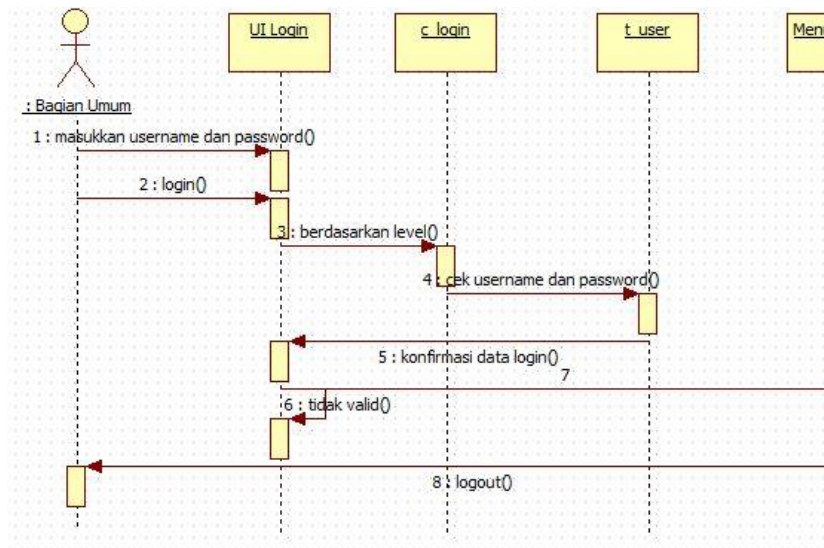
Class Diagram menggambarkan *class* dan hubungan antar *class* didalam sistem. *Class* digambarkan dengan sebuah kotak dibagi menjadi tiga bagian. Bagian paling atas diisikan nama *class*, bagian tengah diisikan *variabel* yang dimiliki *class*, dan bagian bawah diisikan *method-method* dari *class*.

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*

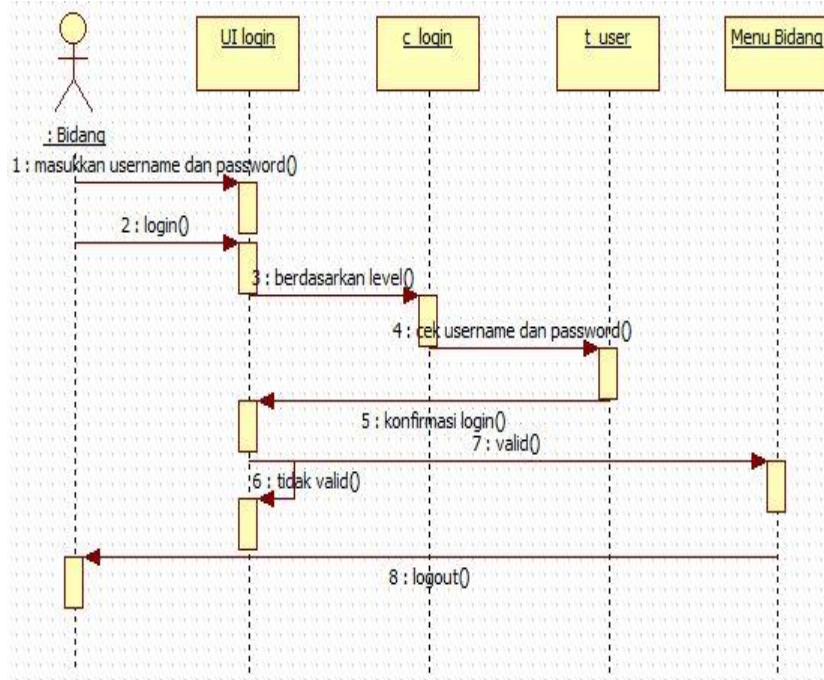


Gambar 5.9 *Sequence Diagram Login Bagian Umum*

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

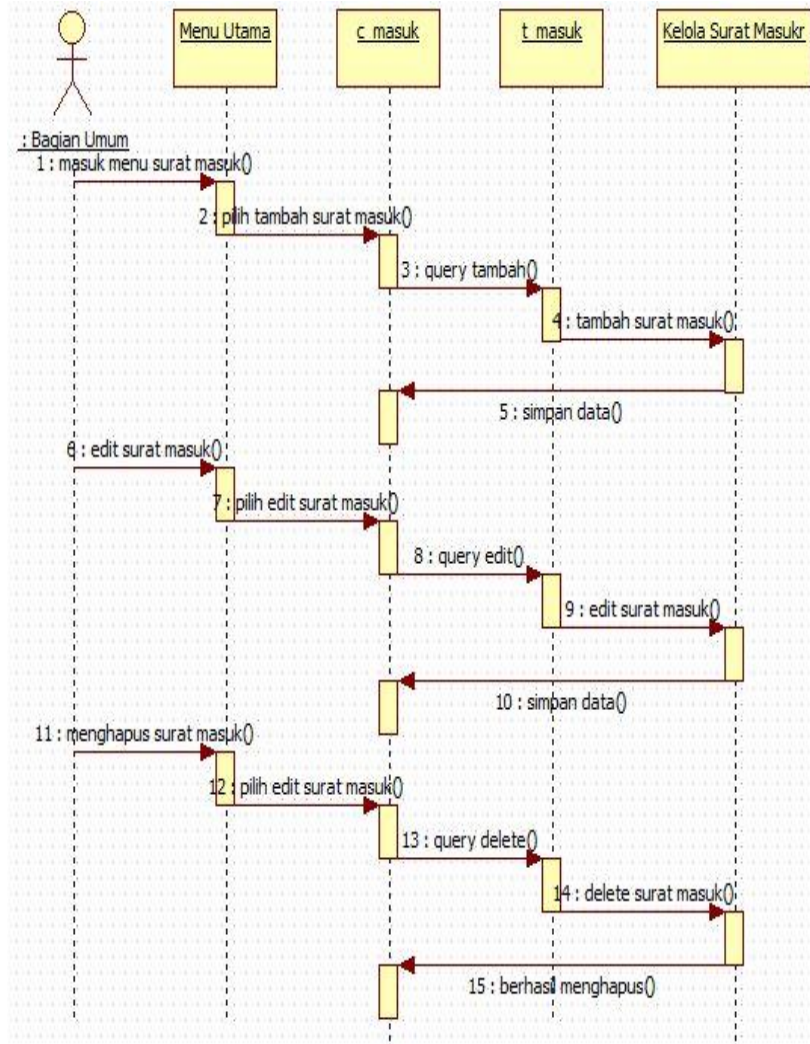


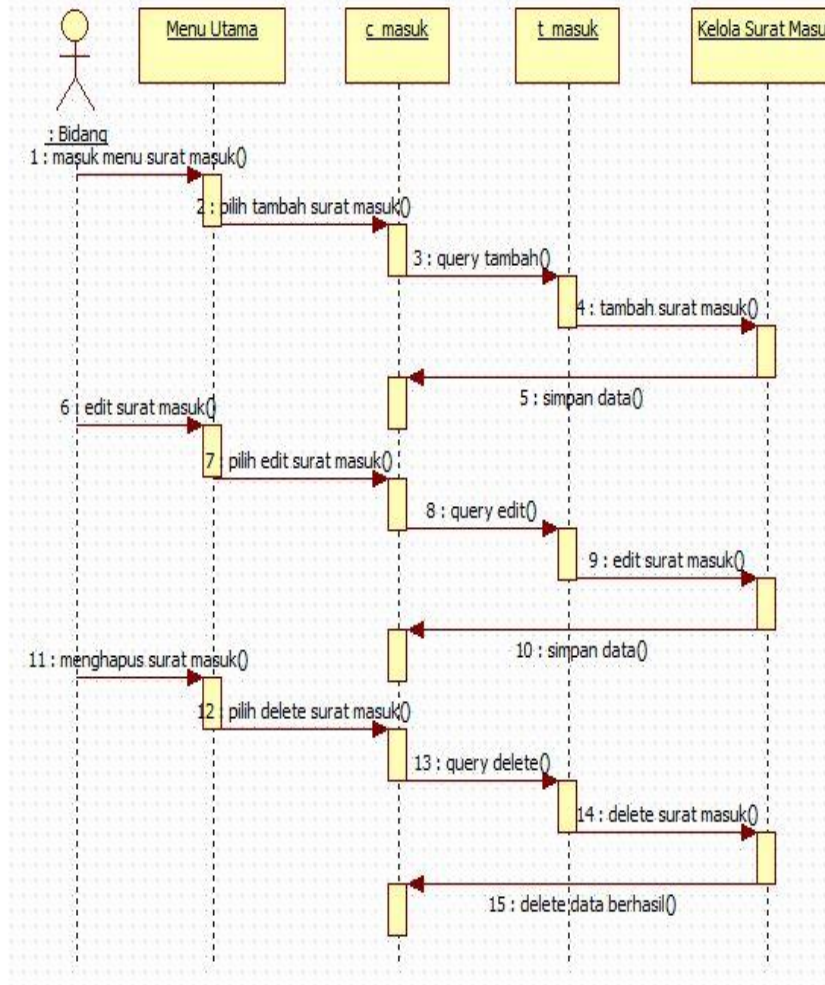
Gambar 5.10 Sequence Diagram Login
Bidang-bidang

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

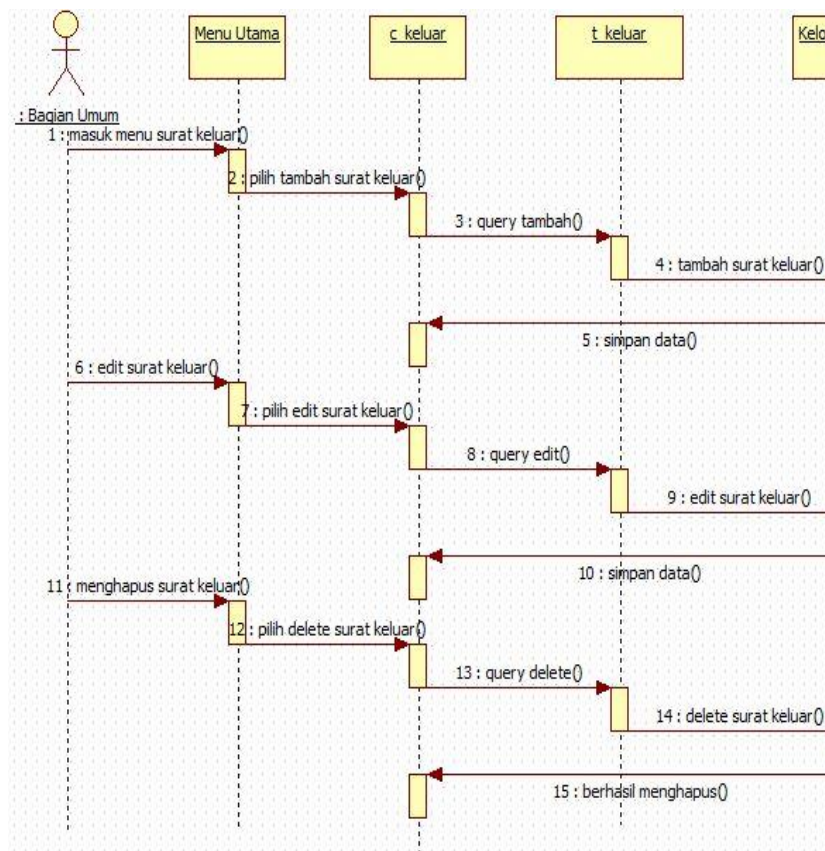


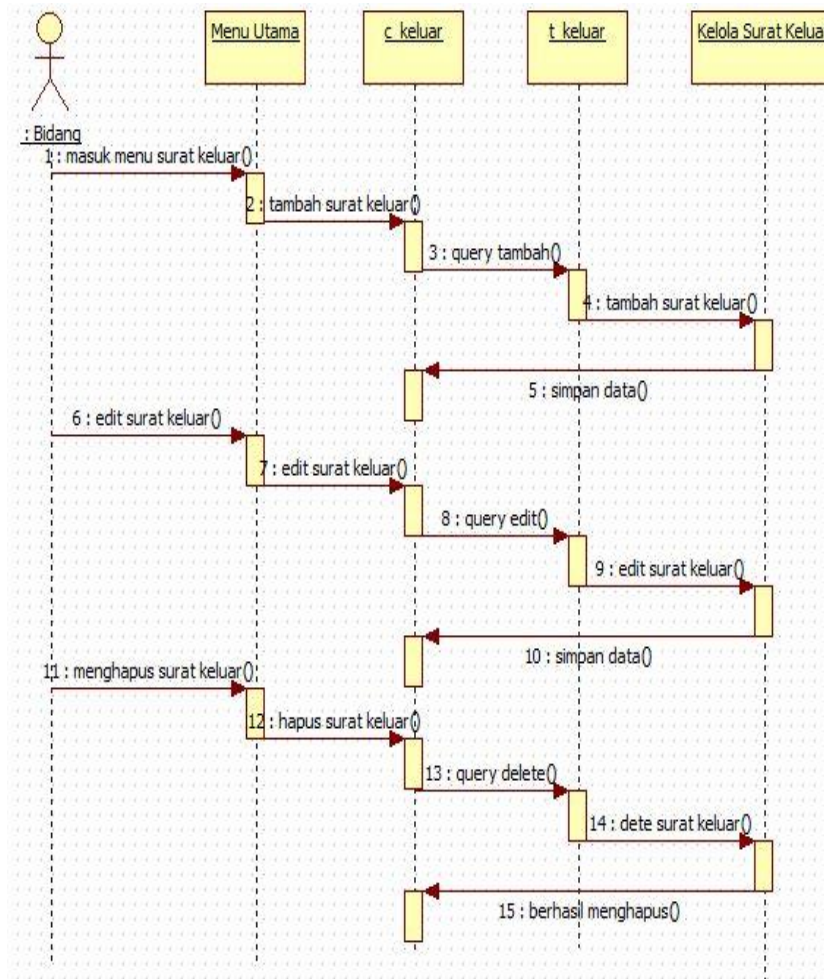


Gambar 5.12 *Sequence Diagram* Kelola Surat
Masuk

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

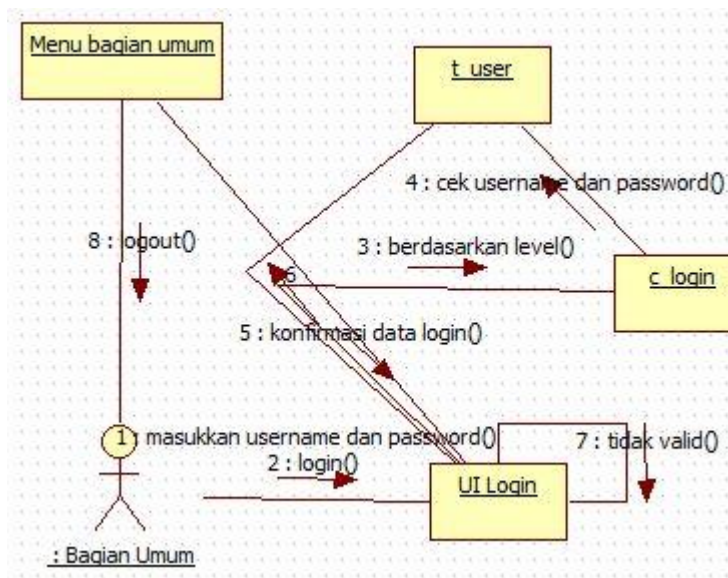
Pada *sequence diagram* ini menjelaskan proses mengelola data *user* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu kelola data user, kemudian sistem akan melakukan koneksi *database*. Setelah *form* 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.

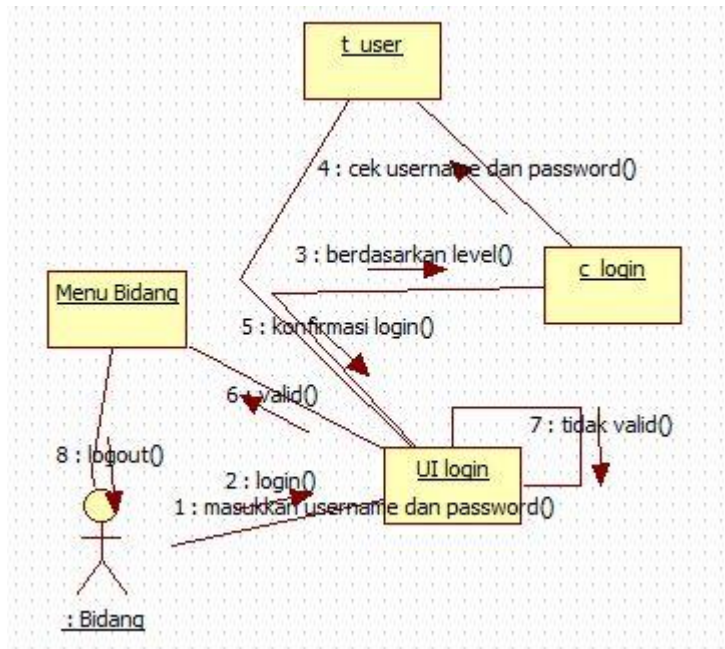
5.2.4.1 Collaboration Diagram Login Bagian Umum



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

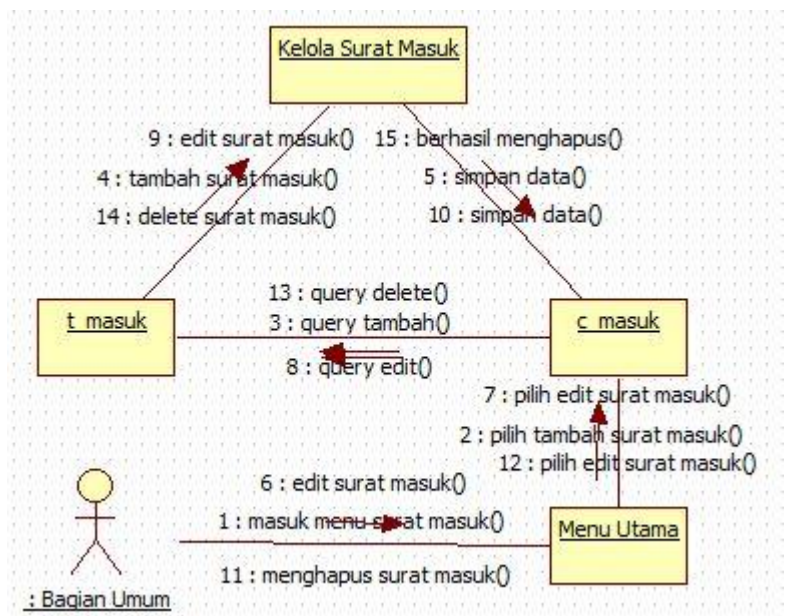


Gambar 5.19 *Collaboration Diagram Login*
Bidang

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

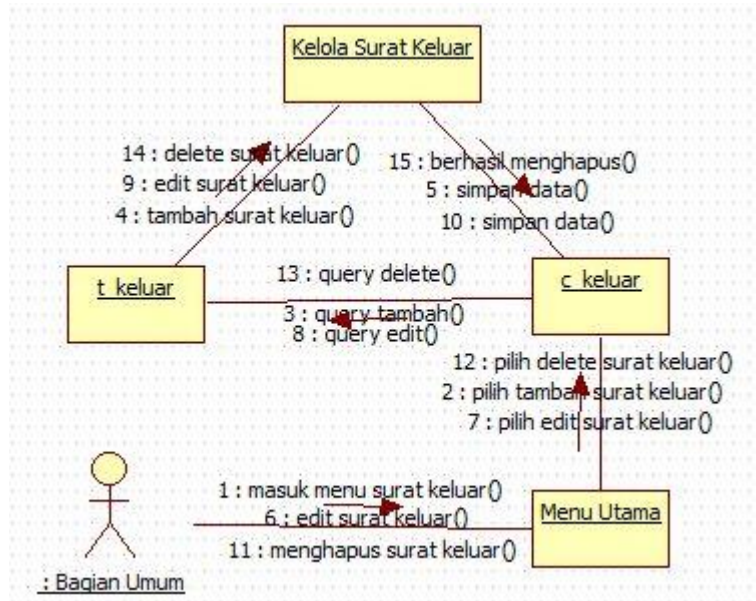
username dan password, pada interface login data yang di input akan di cek di tabel user. Jika sesuai maka akan mendapatkan tampilan form halaman utama bidang.

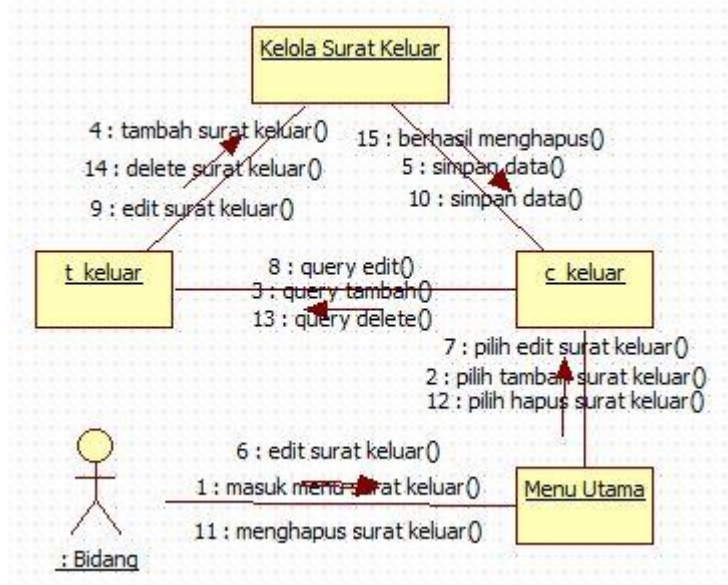
5.2.4.4 Collaboration Diagram *Kelola Surat Masuk*



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

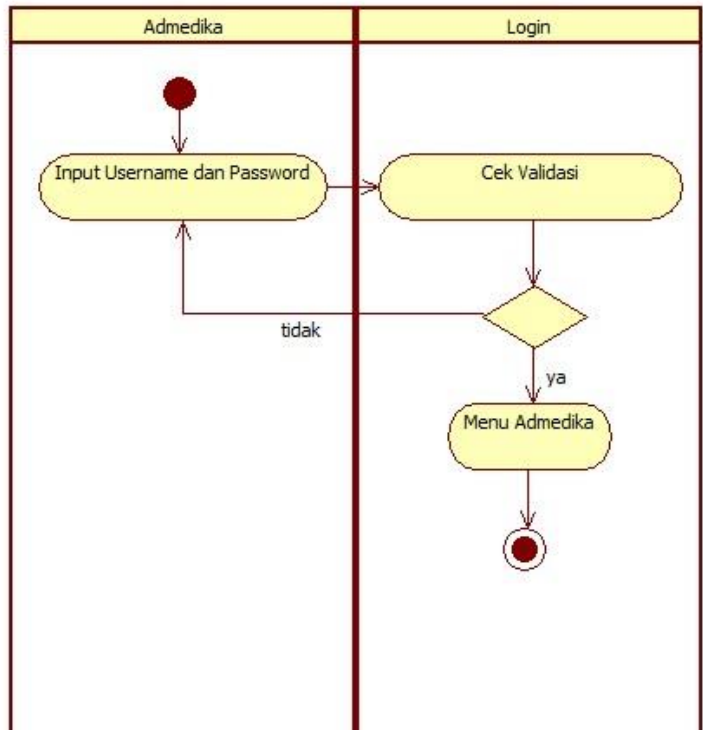
Pada *collaboration diagram* ini menjelaskan proses mengelola surat keluar. Bagian umum dan bidang melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau

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

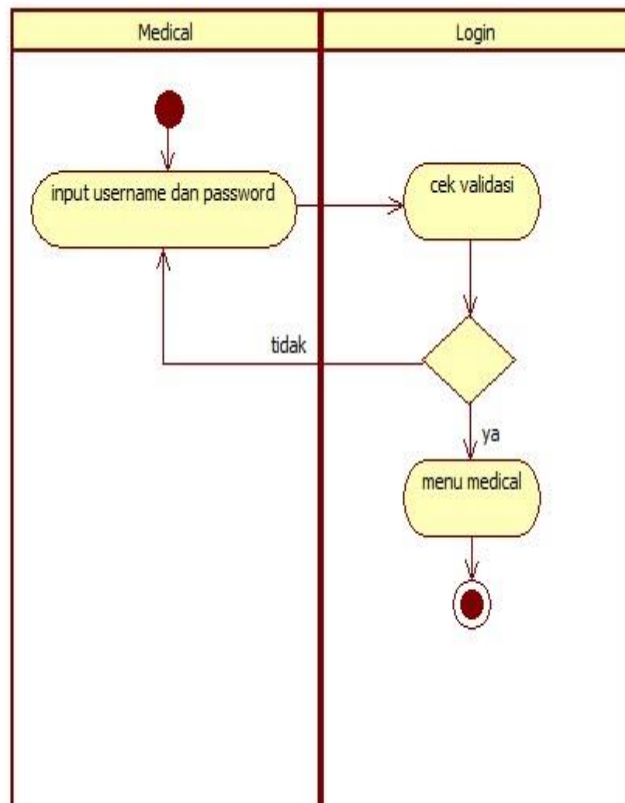


Gambar 5.27 Activity Diagram Login
Admedika

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*.

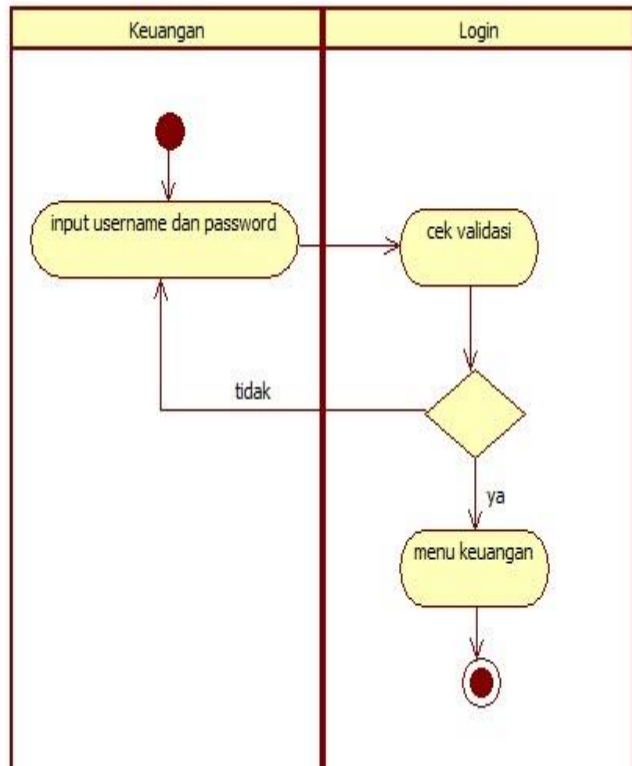
5.2.5.2 Activity Diagram Login Medical



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*.

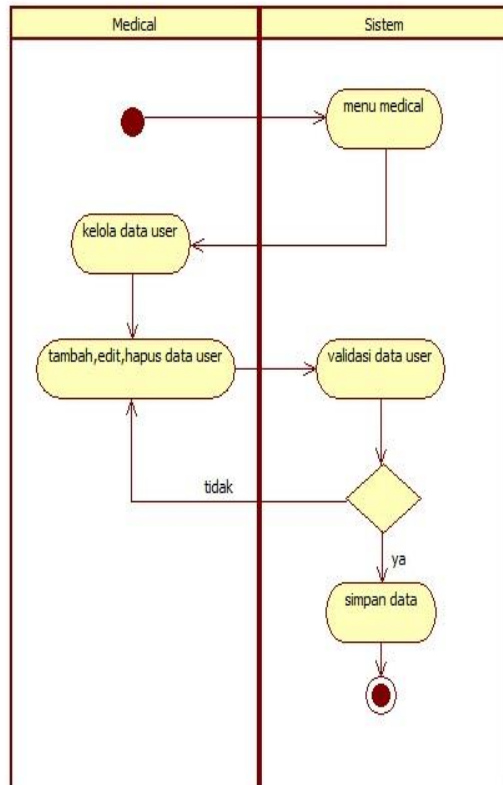
5.2.5.3 Activity Diagram Login Keuangan



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*.

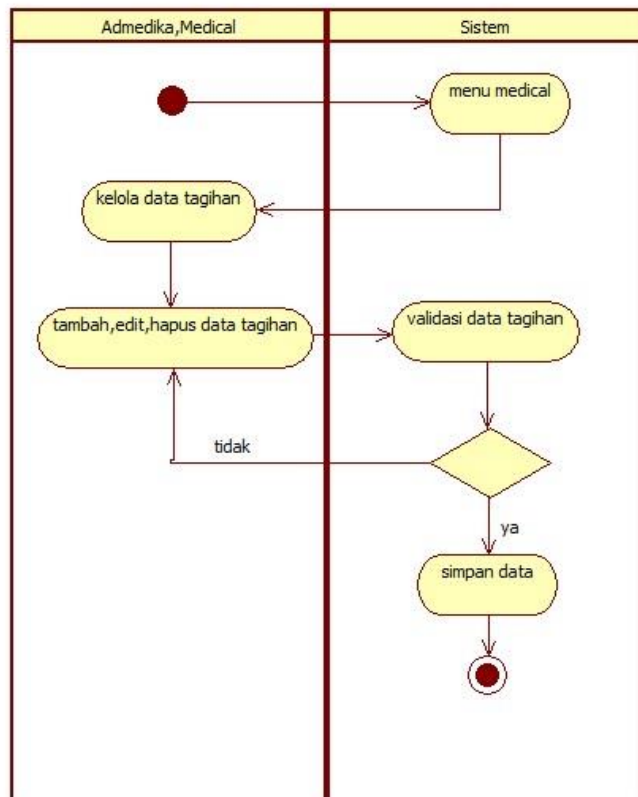
5.2.5.4 Activity Diagram Kelola Data User



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.

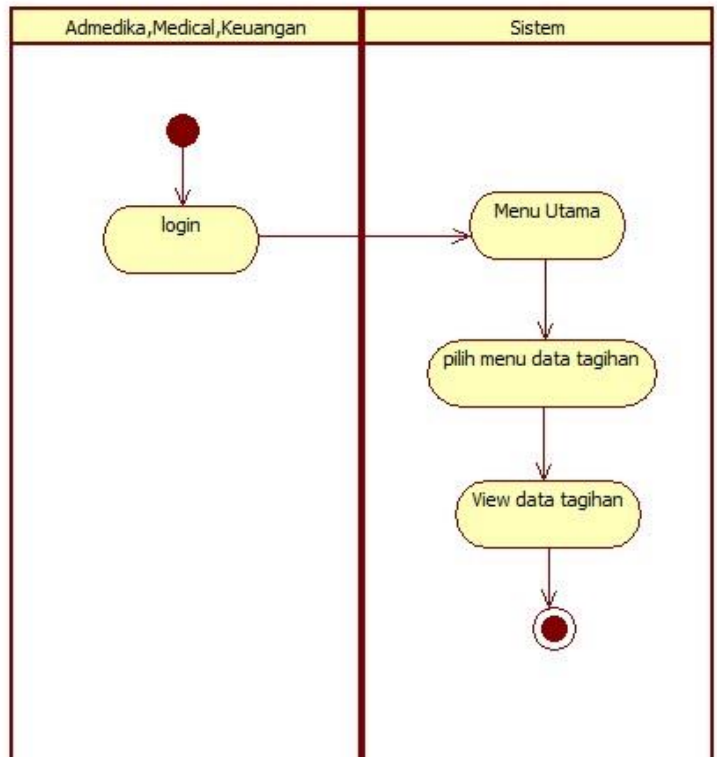
5.2.5.5 Activity Diagram Kelola Data Tagihan



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.

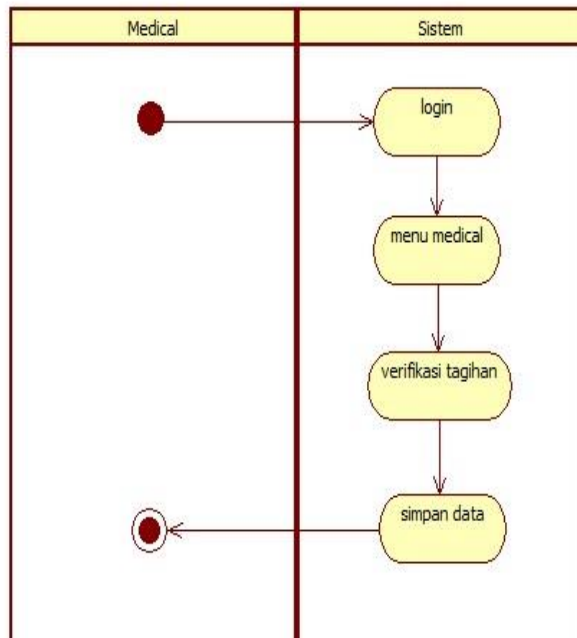
5.2.5.6 Activity Diagram View Data Tagihan



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.

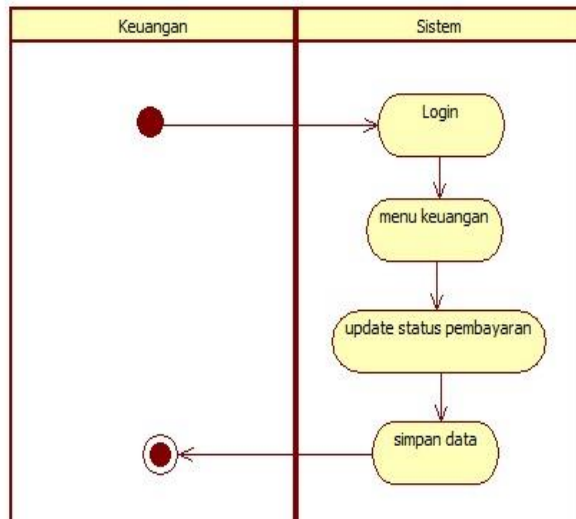
5.2.5.7 Activity Diagram Verifikasi Tagihan



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

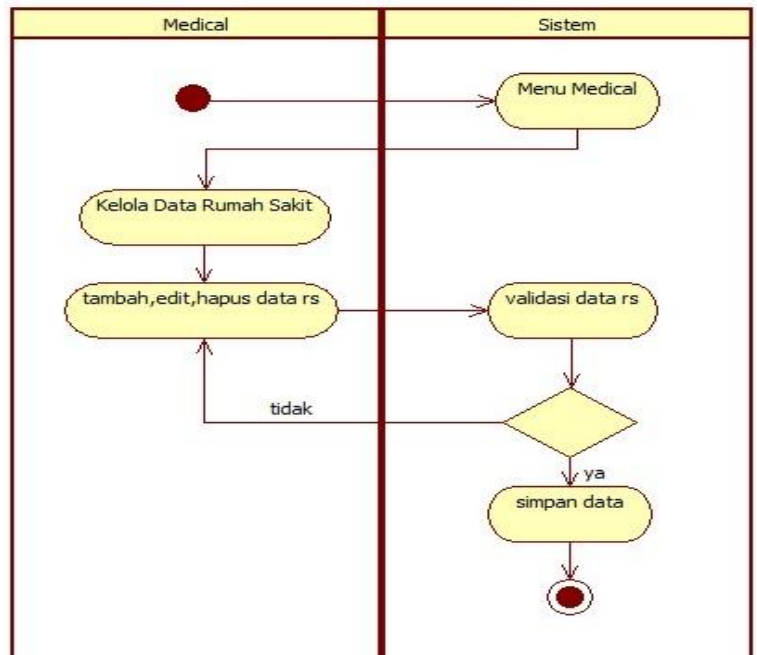


Gambar 5.34 Activity Diagram Update Status Pembayaran

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

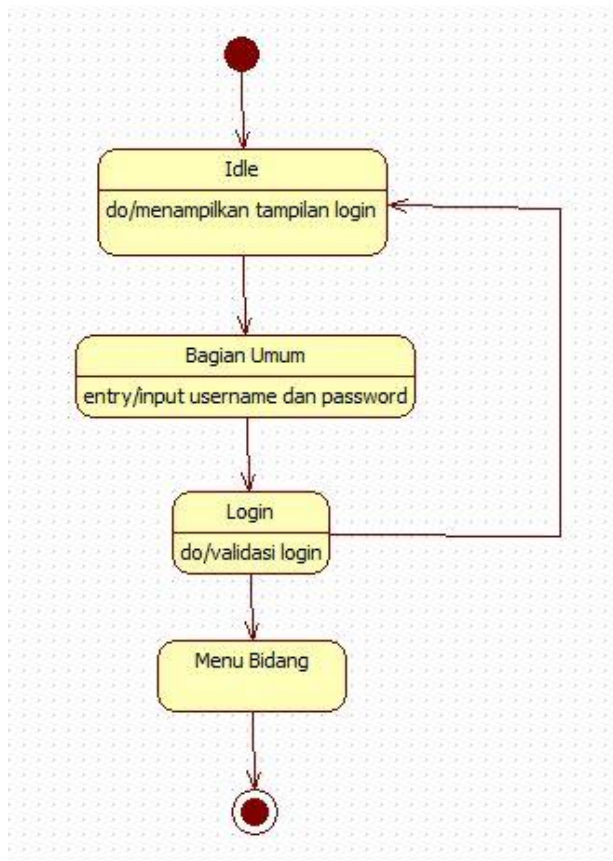
Pada *activity diagram* ini menjelaskan proses mengelola data rumah sakit. *Medical*

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.

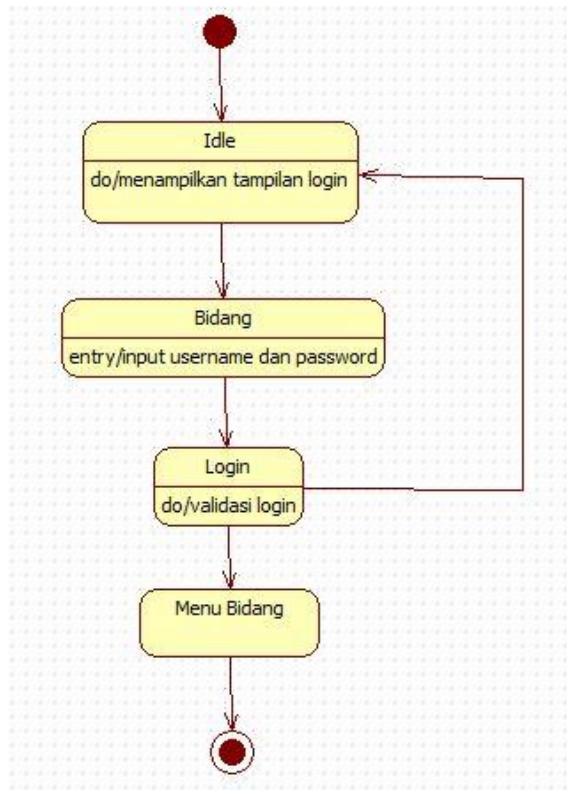
5.2.6.1 Statechart Diagram Login *Bagian Umum*



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*

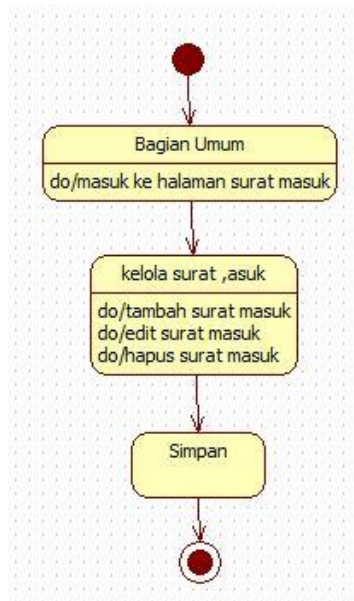


Gambar 5.37 *Statechart Diagram Login Medical*

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

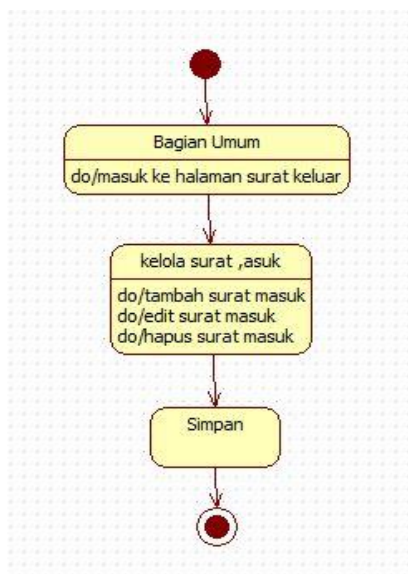




Gambar 5.39 *Statechart Diagram* Kelola Surat Masuk

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

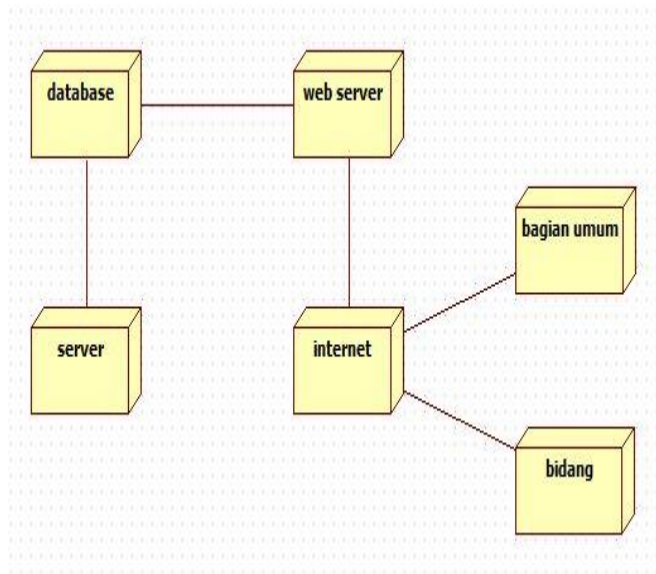
Component diagram yaitu salah satu jenis *diagram* pada *UML* yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa *file* data atau *.exe*, *source code*, *table*, dokumen, dll.

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 empty rectangular input fields stacked vertically. 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*

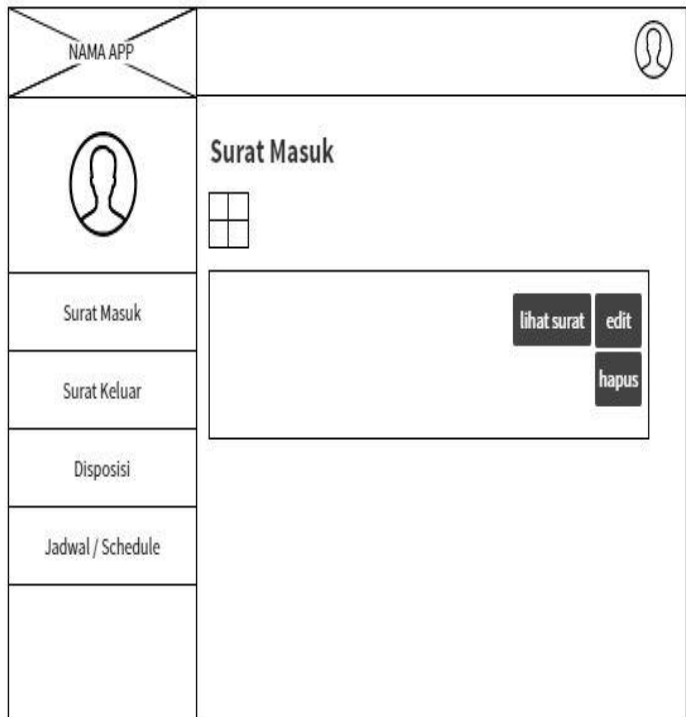
Textbox 2 : *Input Password*

Button 1 : *Tombol Login*

Label 1 : Dinas Perhubungan

Provinsi Jawa Barat

5.2.9.2 User Interface Halaman Surat Masuk

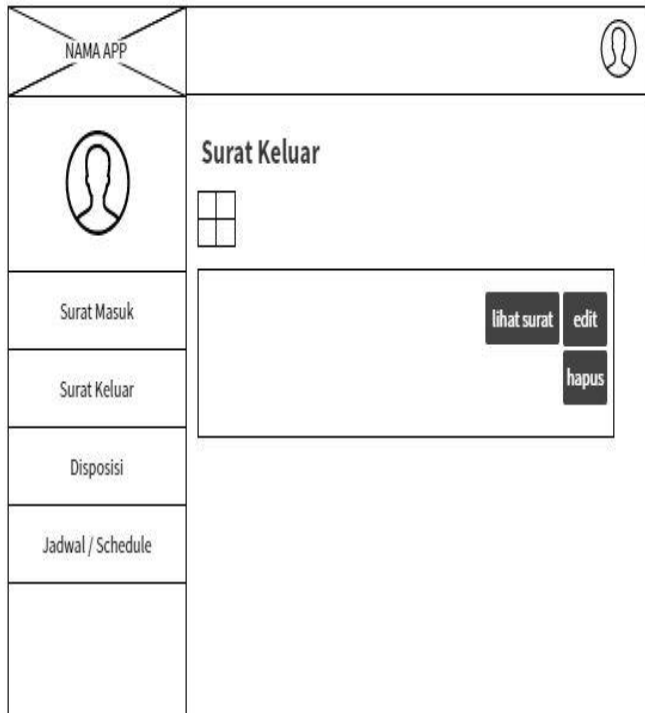


Gambar 5.48 User Interface Halaman Surat
Masuk

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




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

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"/> 12 May 2016 	
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
	<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 Pengujian

Kelas Uji	Butir Uji	Identifikasi SKPL	Tingkat Pengujian	Jenis Pengujian	Jadwal
Validasi Logi n	Validasi Login Pelanggan	UC01	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
Tampil Form Uta ma	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	UC05	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019

	Jadwal / Schedule				
Log out	Logout Bagian Umum dan Bidang	UC0 7	Penguji an Sistem	<i>Black Box</i>	25 – 09- 2017

6.2.1 Deskripsi dan Hasil Uji

6.2.2.1 Pengujian Validasi Login

Tabel 6. 4 Pengujian Validasi Login

Ide nti fik asi	Desk ripsi	Prose dur Peng ujian	Masuk an	Keluar an Yang Dihara pkan	Kriteri a Evalua si Hasil	H as il	Ke si mp ula n
UC 01	Men guji Logi n.	Klik Butto n OK	Masukk an User Name : Bagian Umum	Tampil Form Utama Bagian Umum atau	Tampil Form Utama Bagian Umum atau	O K	Dit eri ma

			Passwor d :****	Bidang .	Bidang .		
--	--	--	--------------------	-------------	-------------	--	--

6.2.2.2 Pengujian Tampilan Utama Pelanggan

Tabel 6. 5 Pengujian Tampilan Utama Pelanggan

Ide ntif ika si	Desk ripsi	Prose dur Pengu jian	Masu kan	Keluara n yang Dihara pkan	Krite ria Evalu asi Hasil	H as il	Ke si m pu lan
UC 02	Meng uji Menu Surat Masu k	Tampi l Data	-Data Surat Masu k	- Data yang ada di databa se munc ul	Tampi l di Halam an Surat Masu k	O K	Dit eri ma

Ide ntif ika si	Desk ripsi	Prose dur Pengu jian	Masu kan	Keluara n yang Dihara pkan	Krite ria Evalu asi Hasil	H as il	Ke si m pu lan
UC 03	Meng uji Menu Surat Kelua r	Tampi l Data	-Data Surat Kelua r	- Data yang ada di databa se munc ul	Tampi l di Halam an Surat Kelua r	O K	Dit eri ma
UC 04	Meng uji Menu Delet e Surat Masu k dan Kelua r	Delete	-Data Surat	- Proses modif ikasi berhas il kemb ali ke halam an Surat	Tampi l halam an Surat	O K	Dit eri ma

Ide ntif ika si	Desk ripsi	Prose dur Pengu jian	Masu kan	Keluara n yang Dihara pkan	Krite ria Evalu asi Hasil	H as il	Ke si m pu lan
UC 05	Meng uji Menu Dispo sisi	Tampi l Data	Data Dispo sisi	- Data yang ada di databa se munc ul	Tampi l di Halam an Dispo sisi	O K	Dit eri ma
UC 06	Meng uji Menu Jadw al / Sche dule	Tampi l Data	-Data Jadw al	- Data yang ada di databa se munc ul	Tampi l di Halam an Dispo sisi	O K	Dit eri ma

6.2.2.3 Pengujian Logout

Tabel 6. 6 Pengujian Logout

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil	Kesimpulan
SKP L11	Meng uji Logout	Klik menu logout lalu pilih logout	-	Tampil Form Bidang Umum atau Bidang	Tampil Form utama	OK	Diterima

Mengenal Data

Data merupakan fakta atau observasi mental yang biasanya mengenai fenomena fisik atau transaksi bisnis. Data pun merupakan suatu ukuran objektif dari atribut (karakteristik) dari entitas. Dalam pemrograman tempat penyimpanan data disebut juga dengan basis data. Basis data merupakan suatu focus utama pada aplikasi, pada awal tahun 1960 Charles Bachman di perusahaan General electric mendesain generasi pertama DBMS yang disebut dengan penyimpanan data terintegrasi (Integrated Data Store) karena hampir semua data computer disimpan pada pita magnetic, namun karena pita hanya dapat di proses secara berurut, maka data harus disimpan dalam bentuk daftar (atau biasa disebut file sekuensial. Database terdiri dari kelompok table-tabel yang berhubungan. Pada kebanyakan kasus, masing-masing table berisi data dengan satu tema. Hubungan antara tabletabel direpresentasikan dengan cara memberikan masing- IV-117 V-117 masing baris ID yang unik dan menggunakan ID tersebut untuk menghubungkan sebuah baris

dalam satu table dengan sebuah baris dalam table baris kedua. Key pada database merupakan kolom yang mengidentifikasi baris yang unik. Kemudian foreign key merupakan sebuah kolom dalam satu table yang merupakan identifier unik pada table kedua. Foreign key digunakan untuk memperlihatkan hubungan diantara table. Table dapat dikombinasikan dengan menggunakan SQL yang merupakan bahasa standar industry untuk pemrosesan table. Sebuah sistem database terdiri dari empat unsur dasar yaitu :
a. User b. Aplikasi database c. DBMS d. Database User menggunakan sistem database untuk melakukan pekerjaannya. Aplikasi hasilkan form, queries dan report, mengeksekusi logika aplikasi dan mengendalikan pemrosesan DBMS menciptakan, memproses, dan mengadministrasikan database. IV-118 V-118
Sebuah database adalah sekumpulan records terintegrasi yang menggambarkan dirinya sendiri. Database terdiri dari data user, metadata, indeks, prosedur tersimpan (stored procedur), trigger, dan metadata aplikasi database. Trigger merupakan prosedur yang aktif saat tindakan tertentu terjadi. Teknologi

database dapat digunakan dalam aplikasi yang luas. Sebagian database digunakan oleh satu individu, sementara sebagian lagi digunakan oleh workgroup, dan yang lainnya digunakan oleh organisasi besar. Seperti sistem informasi, sistem database dapat dikembangkan dengan menggunakan sebuah proses yang mencakup tiga fase, yaitu : a. Requirement b. Desain c. Implementasi. Selama fase requirement, sebuah model data atau representasi logis dari struktur database, dikembangkan. Model data merupakan bagian penting karena desain database dan aplikasinya diturunkan dari model data.

IV-119 V-119 Diagram entity-relationship merupakan sebuah tool yang digunakan untuk merepresentasikan model data. Model data ditransformasikan menjadi table dan hubungan selama fase desain. Indeks, batasan, prosedur tersimpan, dan trigger juga dirancang dalam fase ini. Diagram struktur data kadang-kadang digunakan untuk mendokumentasikan table dan hubungannya. Model Database Pertama, dengan suksesnya media penyimpanan disk pada tahun 1960an, maka kita dapat memiliki akses non-sekuensial atau langsung ke records (Perekaman). Dalam hal

ini, database dirancang untuk menghilangkan masalah pemrosesan file sekuensial. Terdapat dua arsitektur atau model yang awalnya sukses. IBM mengembangkan dan mempromosikan DL/I atau Data language one yang membuat model data database dalam bentuk pohon hirarki.

IV-120 V-120 Gambar DBMS Hirarki

Model Relasional Model database relasional pertama kali diajukan oleh E.F. Codd pada tahun 1970. Codd bekerja untuk IBM dan setelah 10 tahun melakukan penelitian, pengembangan dan lobi, dia dan temannya yang lain berhasil meyakinkan IBM untuk mengembangkan produk DBMS berdasarkan model relasional, model yang paling terkenal dalam produk ini yaitu DB2, yaitu DBMS yang masih aktif digunakan hingga saat ini. Kemudian perusahaan seperti Oracle, Ingres, Sybase dan Informix mengembangkan juga produk DBMS berdasarkan model relasional.

IV-121 V-121 Gambar Database Relasional

Dasar untuk model data jaringan terbentuk lalu di standarisasi oleh CODASYL (Conference on Data System Languages). Pada tahun 1973 Bachman menerima penghargaan CM Turing Award yaitu nobel pada ilmu computer. Menurut sejarah, system pemrosesan basis data terbentuk setelah masa system pemrosesan manual dan system pemrosesan berkas. System pemrosesan manual (Berbasis kertas) merupakan bentuk pemrosesan yang menggunakan dasar berupa setumpuk rekaman yang disimpan pada rak-

rak berkas. Jika berkas-berkas tersebut diperlukan, berkas tersebut harus dicari pada rak- IV-122 V-122 rak tersebut. System pemrosesan berkas merupakan system komputer, dimana sekelompok rekaman disimpan pada sejumlah berkas secara terpisah. Perancangan system ini didasarkan pada kebutuhan individual pengguna, bukan kebutuhan sejumlah pengguna. Sehingga setiap aplikasi menuliskan data sendiri, alhasil ada kemungkinan data yang sama terdapat pada berkasberkas lain yang digunakan oleh program aplikasi lain. Sejarah DBMS (Database Management System), generasi pertama DBMS didesain oleh Charles Bachman di perusahaan General Electric pada awal tahun 1960, disebut sebagai penyimpanan data terintegrasi (Integrated Data Store). Dibentuk dasar untuk model data jaringan yang kemudian distandardisasi oleh CODASYL (Conference on Data System Languages). Pada akhir 1960, IBM mengembangkan system manajemen informasi (Information Management System) DBMS. IMS dibentuk dari representasi data pada kerangka kerja yang disebut dengan

model data hirarki. Dalam waktu yang sama, dikembangkan system SABRE sebagai hasil kerjasama antara IBM dengan perusahaan IV-123 V-123 penerbangan Amerika. System ini memungkinkan user untuk mengakses data yang sama pada jaringan komputer. Pada tahun 1985, Microsoft dan IBM mengumumkan perjanjian kerjasama jangka panjang untuk mengembangkan system operasi dan produk-produk perangkat lunak lainnya. Pengumuman ini adalah permulaan dimulainya OS/2, sebuah system operasi setelah masa kejayaan MS-DOS. OS/2 ini akan lebih hebat dan lebih canggih daripada MS-DOS, ia akan mampu menangani multitasking application dengan memanfaatkan kemampuan processor Intel yang terbaru yaitu 80286. OS/2 secara resmi diumumkan pada bulan april 1987, dan dijanjikan akan tersedia buat end-user pada akhir tahun tersebut. Tetapi dalam waktu yang sangat singkat setelah pengumuman perjanjian kerjasama tersebut, IBM mengumumkan peluncuran sebuah versi spesial OS/2 yang disebut OS/2 Extended Edition. Versi lebih powerfull ini akan menyertakan sebuah

Database SQL yang disebut OS/2 Database Manager, OS/2 Database Manager akan sangat berguna bagi pengembangan aplikasi yang sederhana dan kompatibel dengan DB/2 , IV-124 V-124 sebuah Database server milik IBM yang beroperasi pada mainframe. OS/2 Database manager juga akan menyertakan SNA (System Network Architecture) communication service, yang disebut OS/2 Communication Manager. Sebagai bagian dari SSA (System Application Architecture)-nya. IBM menjanjikan semua produk-produk tersebut dapat saling bekerjasama pada masa yang akan datang. Hal inilah yang membuat Microsoft segera mencari solusi sendiri. Pada tahun 1986, Microsoft sudah menghasilkan 197 JutaUS\$ per tahun, dengan 1153 pegawai. (Sepuluh tahun kemudian, Microsoft telah mendapatkan 6 Milyar US\$ dari bisnis softwarentya, dengan hampir sekitar 18.000 pegawai). Produk-produk Microsoft hampir semuanya terfokus pada aplikasi desktop dengan produk utamanya adalah MS-DOS. Komputasi Client/Server pada saat itu belum menjadi fokus utama Microsoft dan industri komputer. Manajemen data pada sebuah PC

hanyalah menjadi sebuah fantasi pada masa itu, User biasanya hanya menggunakan LOTUS 1-2-3 untuk menyimpan data. Produk dBASE buatan Ashton Tate segera menjadi sangat IV-125 V-125 populer setelah diluncurkan, Selanjutnya Anca Software merilis Paradox dan Micro Rim dengan produk RBase-nya. Pada tahun 1986 tersebut, Microsoft belum mempunyai produk manajemen Database sendiri. (Tetapi pada tahun 1992, Microsoft mendapatkan sukses yang luar biasa dari produk manajemen database desktop-nya dengan Microsoft Access dan Microsoft FoxPro). Tetapi IBM Database Manager sangatlah berbeda dari dBASE, Paradox atau RBase. Produk IBM ini walaupun tidak begitu user-friendly, tetapi ia mempunyai SQL query processor-nya sendiri dan bekerja berdasarkan transactions, hampir sama dengan Database Server yang berjalan diatas Minicomputer dan mainframe, seperti DB/2, Oracle ataupun Informix. Microsoft membutuhkan produk DBMS (Database Management System) sekaliber ini dan membutuhkannya segera!. Microsoft berpaling kepada Sybase, Inc. Sebuah perusahaan pembuat software DBMS

yang sedang 'Naik daun', yang merilis produk Data Server-nya pada bulan Mei 1987 untuk Sun Microsystems yang berjalan di atas UNIX. Data Server mendapatkan reputasi berkat inovasi dan kreatifitasnya dengan Stored Procedure dan Trigger IV-126 V-126 serta paradigma baru dalam dunia komputasi yaitu: Client/Server. Basis data merupakan kumpulan dari elemen data logis yang saling berhubungan. Basis data mengonsolidasi banyak catatan yang sebelumnya disimpan dalam file terpisah. Basis data pun merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh organisasi. Maka basis data merupakan tempat penyimpanan data yang besar dimana dapat digunakan oleh banyak pengguna. Seluruh item basis data tidak dimiliki oleh satu departemen, melainkan menjadi sumber daya perusahaan yang dapat digunakan bersama.

1.6.1 Object Oriented DBMS (OODBMS)

Pemrograman berorientasi objek (object oriented) mulai digunakan pada tahun 1980 dan akan dikembangkan menjadi produk DBMS

berorientasi objek. Tujuan produk ini untuk menyimpan objek pemrograman berorientasi objek seperti C++ atau java, dalam sebuah database tanpa harus mentransformasikannya ke format relasional. IV-127 V-127 1.6.2

MySQL MySQL merupakan sebuah perangkat lunak yang mengatur basis data relasional (RDBMS) yang di distribusikan gratis dengan lisensi GPL (General Public License). Pengguna dapat menggunakan MySQL secara bebas namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan bersifat komersial. MySQL adalah server yang melayani database. Untuk membuat dan mengolah database, pada dMySQL ini dapat mempelajari pemrograman khusus yang disebut query (perintah) SQL. Database diperlukan jika akan menginput data dari user menggunakan form HTML untuk diolah PHP agar dapat disimpan kedalam database MySQL. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya. SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau

seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. IV-128 V-128 MySQL mendukung operasi basisdata transaksional maupun operasi basisdata nontransaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus nontransaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

Sejarah PHP (Hypertext Preprocessor)

Pertama kali PHP dikenal dengan kependekan dari (Personal Home Page) atau Situs Personal yang dibuat oleh Rasmus Lerdorf pada Tahun 1995. Saat pertama kali dibuat PHP masih bernama Form Interpreted (FI) yang berupa sekumpulan skrip untuk pengolahan data formulir dari web. Logo PHP Pertama Kemudian Rasmus merilis kode sumber tersebut pada bulan November tahun 1997 untuk umum dengan memberikan nama PHP/FI. Dengan merilis kode tersebut menjadi kode sumber terbuka sehingga banyak IV-93 V-93 pemrograman yang tertarik untuk mengembangkan PHP tersebut. Di tahun yang sama, sebuah perusahaan bernama Zend menulis kembali interpreter PHP menjadi lebih cepat, rapih dan lebih baik. Sehingga pada bulan Juni tahun 1998, perusahaan tersebut merilis kembali interpreter baru untuk PHP dan meresmikan perilisan tersebut dengan nama PHP 3.0 serta mengubah nama menjadi akronim berulang yang diberi nama PHP (Hypertext Preprocessing). Pada pertengahan tahun 1999, perusahaan Zend merilis kembali interpreter PHP dengan nama

PHP 4.0 pada versi 4.0 ini banyak digunakan pada awal abad ke-21. Karena versi ini memiliki kemampuan untuk membangun aplikasi web kompleks tetapi tetap memiliki stabilitas dan kecepatan yang tinggi. Setelah 5 tahun berlalu pada akhirnya Zend merilis kembali PHP dengan versi 5.0 pada bulan Juni tahun 2004. Pada versi ini, PHP memasukkan model pemrograman berorientasi objek untuk mengembangkan bahasa pemrograman ke arah paradigma berorientasi objek. IV-94 V-94 Gambar logo PHP 5 Di tanggal 18 Desember pada tahun 2014 Zend merilis bahasa pemrograman PHP menjadi versi 5.6.4 yang diperuntukan agar mempermudah pengembang menjalankan kode PHP tanpa menginstall software server terlebih dahulu. Tidak lama kemudian pada tanggal 17 Februari 2017 versi 5.6.4 dikembangkan kembali menjadi versi terbaru dan stabil pada bahasa pemrograman PHP saat ini menjadi versi 7.0.16 dan 7.1.2. IV-95 V-95 Gambar Logo PHP 7 1.5 Menenal PHP PHP merupakan bahasa pemrograman scripting karena bahasa pemrograman PHP diproses pada web server, sehingga semua kode-kode yang ditulis menggunakan PHP akan dijalan

dan disimpan di web server maka PHP menyatu dengan HTML (Kode dasar website) dan dijalankan pada server side. Artinya, semua sintaks PHP yang diberikan akan sepenuhnya dijalankan IV-96 V-96 pada server, sedangkan yang dikirimkan ke browser hanya hasilnya saja. PHP digunakan untuk membuat website dinamis dan interaktif. Dinamis yaitu website tersebut dapat berubahubah tampilan dan kontennya sesuai kondisi tertentu. Terdapat dua acara yang dapat dilakukan untuk menulis kode-kode PHP untuk membangun sebuah aplikasi dinamis, seperti menggunakan PHP native dan framework.

- a. PHP Native PHP Native merupakan metode penulisan kode-kode bahasa pemrograman PHP yang benar-benar dimulai dari awal atau dari nol. Pada PHP native ini programmer akan membuat kerangka kerjanya sendiri untuk mengembangkan website dinamis. Biasanya pemula akan disarankan untuk mempelajari PHP Native.
- b. Framework PHP Untuk memudahkan pengembangan pada aplikasi web dengan menggunakan bahasa pemrograman PHP maka gunakanlah framework PHP, karena terdapat beberapa IV-

97 V-97 keuntungan yang bisa didapatkan ketika membuat website menggunakan framework, diantaranya : 1) Tidak perl melakukan pengulangan coding 2) Aplikasi web yang dibangun dengan framework akan lebih stabil 3) Mengembangkan aplikasi web dapat lebih cepat 4) Membantu pemula dalam membuat aplikasi Adapun beberapa framework PHP yang sering didengar oleh para developer web, yaitu :

1) CodeIgniter Gambar Logo
CodeIgniter

Variabel

Variabel merupakan suatu lokasi penyimpanan yang berisikan nilai atau informasi yang nilainya tidak diketahui maupun telah diketahui. Nilai dari variabel dapat diisi dengan informasi yang diinginkan dan dapat diubah nilainya pada saat kode program sedang berjalan. Sebuah variabel memiliki nama yang digunakan untuk mengakses nilai dari variabel itu. Jika anda memiliki pengetahuan dasar tentang bahasa pemrograman, tentunya tidak asing dengan istilah variabel.

- Variabel digunakan untuk menyimpan sebuah value, informasi atau data
- Penamaan variable diawali dengan tanda \$
- Panjang yang digunakan tidak ada batas maksimum penggunaan IV-102 V-102
- Setelah menggunakan tanda \$ diawali oleh huruf atau under-score (_). Maka karakter yang berikutnya bias terdiri dari angka, huruf dan karakter tertentu yang diperbolehkan.
- Case yang digunakan bersifat case-sensitive

PHP membedakan variabel yang ditulis dengan huruf besar dan kecil (bersifat case sensitive), contoh \$nilai_kuadrat tidak sama dengan \$NilaiKuadrat dan

\$NILAIKUADRAT ketiganya akan dianggap sebagai variabel yang berbeda. f. Tidak boleh mengandung spasi g. Tidak perlu di deklarasikan.

Tipe Data

Pada PHP tipe data variable secara otomatis ditentukan oleh interpreter PHP. Tetapi, PHP mendukung 8 buah tipe data primitive, seperti :

- a. Integer Integer merupakan semua angka bulat baik itu positif ataupun negative bukan pecahan, seperti : 1,2,3,5,dst
- b. Float Float atau floating point atau yang disebut dengan nomor pecahan atau juga bilangan real, seperti : 0.1,0.25,0.2, 0.314E1,dst.
- c. Boolean Boolean merupakan tipe data standar, Boolean hanya menyatakan kebenaran dengan kata kunci perintah TRUE (benar) atau FALSE (salah).
- d. String String merupakan rangkaian karakter, sehingga karakter tersebut sama seperti byte. String dapat didefinisikan dengan cara yang paling mudah yaitu menggunakan tanda satu petik (‘’) terdapat 255 karakter yang dapat dijadikan string.
- e. Array Array merupakan jenis tipe

data yang unik, berbeda dengan tipe data lainnya, tipe data array dapat menyimpan satu atau lebih data dalam variabel tunggal. Tipe data array berguna untuk menyimpan banyak data dalam satu variabel. f. Resource IV-104 V-104 Resource merupakan jenis tipe data yang menyimpan sebuah referensi external resource seperti bagian koneksi database. g. Object Object merupakan keluaran dari class. Object dapat menampilkan atau mengelola isi class. h. Null Null mendeklarasikan variabel tanpa value. Sebuah variabel dapat menjadi null jika : a) Variabel belum diset ke nilai apapun b) Menerima unset() c) Dialokasikan konstanta NULL.

Konstanta

Dalam bahasa pemrograman, Konstanta atau constant merupakan suatu lokasi penyimpanan yang berisikan nilai yang memiliki sifat tetap dan tidak dapat berubah sepanjang program berjalan. Konstanta memiliki fungsi yang sama seperti variabel namun nilainya statis/konstan dan tidak bisa berubah. 1.5.3.1 Aturan Penulisan Konstanta PHP 1. Cara Pendefinisikan Konstanta dalam PHP IV-105 V-105 Jika variabel pada PHP dibuat dengan menambahkan “tanda dollar”, seperti : \$nilai_kuadrat. Untuk membuat konstanta PHP memiliki 2 cara : a. Menggunakan kata kunci (Keyword) const b. Menggunakan fungsi define 2. Konstanta PHP bersifat case sensitive Sama seperti variabel, konstanta dalam PHP bersifat case sensitif, sehingga perbedaan huruf besar dan kecil dianggap berbeda. GAJI, GaJi, dan gaji merupakan 3 konstanta yang berbeda. 3. Nilai konstanta PHP tidak dapat diubah 4. Konstanta hanya dapat berisi tipe data tertentu Konstanta dalam PHP hanya dapat berisi tipe data sederhana, yaitu : a. Boolean b. Integer c. Float d. String Hal ini berbeda dengan variabel yang dapat

berisi tipe data turunan seperti array, objek atau resources. 5. Konstanta Sistem PHP (Predefined Constant) IV-106 V-106 Sama seperti variabel, PHP juga telah membuat beberapa konstanta yang telah didefinisikan dan tidak bisa di ubah nilainya. Namun karena banyaknya modul yang dapat ditambahkan kedalam PHP, Predefined Constant dalam PHP akan bertambah tergantung modul yang ada. Namun sebagai contoh, berikut adalah Predefined Constant dalam sistem inti PHP: PHP_VERSION,PHP_MAJOR_VERSION,PHP_MINOR_VERSION,PHP_RELEASE_VERSION,PHP_EXTRA_VERSION,PHP_ZTS,PHP_DEBUG,PHP_MAXPATHLEN,PHP_OS,PHP_SAPI,PHP_EOL,PHP_INT_MAX,PHP_INT_SIZE,DEFAULT_INCLUDE_PATH,PEAR_INSTALL_DIR,PEAR_EXTENSION_DIR,PHP_EXTENSION_DIR,PHP_PREFIX,PHP_BIN_DIR,PHP_BINARY,PHP_MANDIR,PHP_LIBDIR,PHP_DATADIR, __LINE__ , __FILE__ , __DIR__ , __FUNCTION__ , __CLASS__ ,

__TRAIT__, __METHOD__,
__NAMESPACE__,

Kelebihan PHP

Sebagai salahsatu bahasa pemrograman PHP memiliki banyak kelebihan diantaranya : a. Komunitas yang besar Komunitas PHP sangat besar dan tidak dapat dipungkiri bahwa komunitas ini tersebar diseluruh dunia. Tidak hanya di social media seperti Facebook bahkan komunitas PHP ini berada di Telegram, WhatsApp, dll. b. Mudah dipelajari PHP merupakan bahasa pemrograman semua orang yang hampir semua orang yang bergelut dengan dunia Web IV-111 V-111 Development pernah menggunakannya atau mencobanya. c. Resouces yang melimpah Dengan banyaknya komunitas yang besar maka akan berdampak pada kemudahan mencari resources yang berhubungan dengan PHP baik itu permasalahan yang terjadi di library, software, CMS hingga framework PHP. d. Simpel PHP memiliki syntax yang sangat sederhana dan mudah untuk dipelajari. Untuk memulai belajar PHP tida perlu

melakukan pengaturan apapun cukup hanya install XAMPP ataupun WAMP maka dapat memulai langsung belajar PHP. e. Maintenance mudah Sekali web yang menggunakan PHP berjalan, programmer dapat dengan mudah melakukan update dari software PHP dengan mudah jika memang diperlukan. karena sifat PHP yang merupakan interpreter. Aplikasi web yang dibuat dengan menggunakan PHP dapat dengan IV-112 V-112 mudah diupgrade versi PHP tanpa harus melakukan kompilasi ulang source code. berbeda sekali dengan bahasa pemrograman lain yang membutuhkan kompilasi ulang jika melakukan upgrade versi dari bahasa pemrograman. PHP juga dapat berjalan pada berbagai macam web server seperti apache, nginx, dan IIS. f. Open Source PHP merupakan sebuah projek Open source dengan license yang dikeluarkan oleh PHP group yaitu PHP license V3.01. Inti dari license ini adalah setiap pengguna program PHP bebas menggunakan PHP secara gratis tanpa harus memberikan royalty apapun ke PHP group namun tetap wajib mencantumkan licensi atas PHP yang dimiliki PHP Group. Dengan kata lain selama pemakai program PHP tidak

mengakui produk PHP adalah buaatannya maka perjual belian program yang menggunakan PHP diperbolehkan tanpa harus membayar licensi apapun. g. Perkembangan pesat IV-113 V-113 Karena sifat PHP yang open source, banyak sekali bermunculan projek projek open source besar yang menggunakan PHP seperti Prestashop, WordPress, Drupal, dan lain lain. Hal ini menjadi keunggulan yang sangat besar bagi orang yang menguasai pemrograman PHP. Dengan sangat luasnya perkembangan PHP, maka kesempatan untuk bisnis ataupun kerja pada bidang pemrograman PHP sangatlah luas.

Kekurangan PHP

Kekurangan utama PHP yaitu PHP merupakan bahasa yang weak type dimana variabel tidak memiliki tipe data sehingga menyulitkan ketika melakukan debugging. Pada weak type ini menyebabkan terjadinya juggling dimana variable yang terjadi berisi nilai integer misalnya dapat berubah menjadi nilai string atau tipe data lainnya. Berikut merupakan kekurangan dalam bahasa pemrograman PHP

: a. Banyak kompetisi Komunitas yang banyak tentu membawa kompetisi yang ketat. Para web developer yang menguasai PHP tiap hari semakin bertambah. Namun kekurangan ini IV-114 V-114 seharusnya menjadi pemacu bagi para pebisnis yang menginginkan produk IT untuk menggunakan PHP sebagai bahasa pemrograman yang digunakan untuk mengembangkan aplikasi bisnisnya karena terbukanya para programmer PHP yang sangat kompetitif dan tiap hari semakin banyak b. Mudah di bajak Karena sifat PHP yang merupakan interpreter, source code dari aplikasi php dapat dengan mudah di modifikasi dan diubah fungsinya. hal ini membuat PHP tidak cocok untuk digunakan mengembangkan aplikasi jika pemilik aplikasi memiliki source code yang ingin dijaga kerahasiaannya. Meskipun ada cara untuk mengamankan source code yang menggunakan bahasa pemrograman PHP, namun dibutuhkan sebuah ekstensi yang berbayar yang dikeluarkan oleh Zend sebuah korporasi di bidang pemrograman PHP. c. Terkesan kurang prestigious Entry level yang berada pada tingkat pemula, yakni mudah dipelajari oleh programmer pemula membuat

bahasa pemrograman PHP terkesan kurang prestisius jika IV-115 V-115 dibandingkan dengan bahasa pemrograman web lain yang terkesan lebih sulit untuk digunakan. Aplikasi web yang dihasilkan dari penggunaan bahasa pemrograman PHP terkesan kurang aman dan memiliki celah. Namun sebenarnya hal ini disebabkan karena faktor pengembang yang mungkin belum mempelajari secara penuh bagaimana standar dan cara membuat aplikasi yang benar dengan menggunakan PHP. Dari sisi performa, pemrograman PHP dan pemrograman lainnya jika digunakan dengan standar dan penerapan yang benar, akan menghasilkan sebuah aplikasi web yang berkualitas. d. Tidak adanya type data pada PHP PHP tidak memiliki type data. hal ini membuat kadang pada saat menggunakan bahasa pemrograman PHP muncul bug bug yang tidak diinginkan berkaitan dengan tidak adanya standar type data pada php. contohnya adalah data "1000" dan "1e3" jika dibandingkan akan memiliki type data yang sama karena secara implisit data tersebut dirubah menjadi floating point. namun kekurangan ini berkaitan sekali dengan pengalaman dari developer yang

menggunakan bahasa pemrograman PHP. IV-116 V-116 developer yang sudah ahli tentunya sudah paham betul bagaimana mengatasi permasalahan ini.