BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa tinjauan pustaka yang digunakan sebagai dasar implementasi aplikasi kepegawaian CV. Ladapedas Surakarta berbasis *web* antara lain:

Penelitian pertama dilakukan oleh Rahayu Kurniawati pada tahun 2017 dengan judul Sistem Presensi Berbasis Web Anggota Polresta Surakarta. Penelitian tersebut memfokuskan pada presensi pegawai yang masih dilakukan secara manual dan dianggap tidak efisien baik dari segi waktu maupun tenaga. Proses penyimpanan data presensi anggota tersebut bisa dipermudah apabila mempunyai sebuah sistem aplikasi berbasis komputer. Presensi anggota yang ada sekarang masih dilakukan secara manual menggunakan Ms. Excel dengan memasukkan data presensi anggota satu per satu ke dalam komputer, namun kadang kala terjadi kecurangan dimana presensi bisa dititipkan ke temannya dan anggota opsnal atau bekerja di lapangan. Berdasarkan data dan kasus yang didapatkan maka perlu membuat sistem yang diharapkan dapat membantu Sie Propam Polresta Surakarta dalam melakukan presensi. Sistem presensi tersebut dibangun dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai databasenya. Metode yang digunakan dalam merancang sistem ini adalah metode *Waterfall*.

Pada tahun 2013 dalam artikel ilmiah Pricillia dan Erwin Ciputra Salim menciptakan perancangan basis data berbasis web untuk sistem presensi dan penggajian karyawan pada CV. Utama Teknik. Masalah yang terjadi pada penelitian ini adalah presensi karyawan masuk dan pulang menggunakan alat bantu mesin *checkclock*. Mesin *checkclock* digunakan dengan cara memasukkan kartu presensi karyawan kemudian mesin akan mencetak waktu masuk dan waktu pulang. Dari kartu presensi *checkclock*, *finance* akan menghitung gaji serta uang lembur berdasarkan jumlah jam kerja yang tertera pada kartu presensi karyawan. Hal ini bersifat manual karena *finance* harus memeriksa satu persatu dan menghitung dengan kalkulator. Sistem ini masih bersifat manual sehingga waktu

yang dibutuhkan dalam menjalankan proses menjadi lebih lama. Dari permasalahan tersebut peneliti bertujuan untuk menganalisis dan merancang sistem yang dapat mempercepat proses pengisian presensi serta mempermudah dalam menghitung gaji karyawan. Metode yang digunakan dalam merancang sistem ini adalah metode analisis dan metode perancangan basis data dan aplikasi.

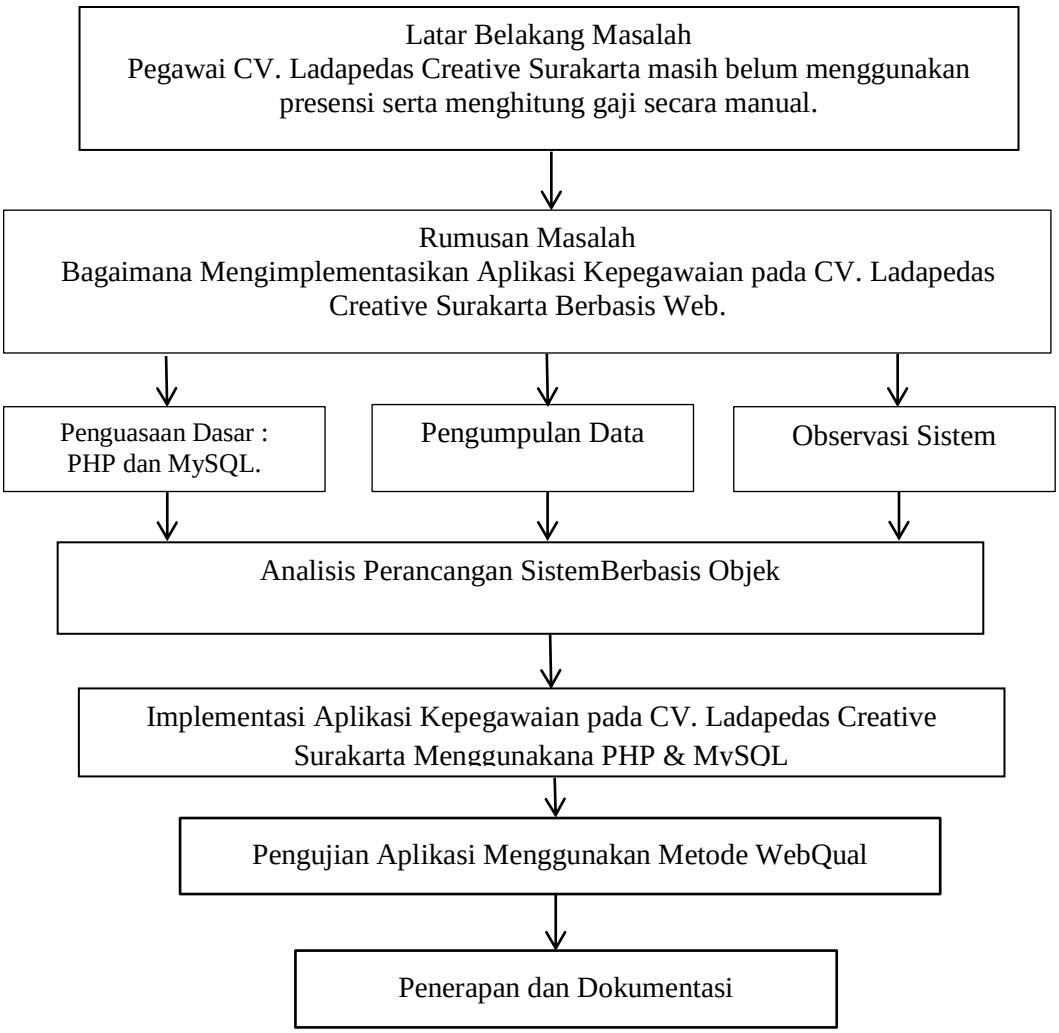
Penelitian yang ketiga dilakukan oleh Warkim dan Hafiz Novanda Ichwan pada jurnal sistem informasi tahun 2015 yang berjudul Analisa dan Desain Sistem Kehadiran Pegawai Pada Pusat Penelitian Perkembangan IPTEK Lembaga Ilmu Pengetahuan Indonesia dalam jurnal sistem informasi. Pada penelitian ini mereka membangun sebuah sistem informasi kehadiran pegawai berbasis web agar informasi kehadiran dapat diakses oleh pegawai melalui jaringan lokal maupun internet. Sistem ini dibangun untuk melengkapi sistem yang lama dimana sistem yang lama menggunakan aplikasi *finger print attendance management*. Aplikasi *finger print attendance management* merupakan aplikasi yang berfungsi dapat membaca data kehadiran dan ketidakhadiran pegawai dari alat *finger print* (sidik jari). Sistem *finger print attendance management* dibangun dengan menggunakan database *Microsoft Access*, dimana database ini hanya bersifat *client*. Karena itu, bagian Tata Usaha terutama pada Sub Bagian Kepegawaian dan Umum memerlukan suatu aplikasi yang berbasikan web, agar infromasi kehadiran dapat diakses oleh pegawai. Berdasarkan masalah tersebut peneliti bertujuan untuk menganalisis dan medesain aplikasi kehadiran pegawai sehingga informasi kehadiran tersebut dapat diakses oleh pegawai baik dari jaringan lokal maupun internet. Metode pengembangan sistem yang digunakan adalah metode RAD (*Rafid Aplication Development*). RAD adalah sekumpulan strategi, metodologi dan alat terintegrasi yang terdapat di dalam suatu kerangka kerja yang disebut rekayasa informasi.

Pada tahun 2018 Yunita Yolanda melakukan penelitian yang berjudul Analisis dan Perancangan Aplikasi Presensi Pegawai CV. Ladapedas Creative Surakarta Berbasis Web. Penelitian tersebut memfokuskan pada presensi pegawai yang belum diterapkan sama sekali oleh pegawai CV. Ladapedas Creative Surakarta dimana pegawai datang dan pulang tanpa mengisikan presensi. Hal ini

membuat pemimpin perusahaan tidak bisa memantau dan mengetahui informasi kehadiran pegawainya. Metode yang digunakan dalam membangun aplikasi ini adalah metode *Waterfall*. Kelebihan dari aplikasi ini adalah mudah digunakan oleh pegawai serta dapat menghemat waktu dalam pengisian presensi. Aplikasi ini dibangun menggunakan bahasa pemrograman PHP dan MySQL sebagai databasenya.

2.2 Kerangka Pemikiran

Berikut ini adalah tahapan kerangka pemikiran dalam melakukan implementasi Aplikasi Kepegawaian CV. Ladapedas Creative Surakarta Berbasis Web.



Gambar 2.1 Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah:

1. Latar Belakang Masalah

Pegawai CV. Ladapedas Creative Surakarta belum menggunakan presensi sama sekali hal ini menyebabkan kurang disiplinnya pegawai dan sulitnya mendapatkan informasi tentang kehadiran pegawai, karena seiring dengan berkembangnya perusahaan maka pegawai perusahaanpun akan semakin banyak jika tidak dilakukan proses presensi maka pemimpin perusahaan tidak akan mengetahui bagaimana perkembangan pegawainya. Analisis dan perancangan aplikasi sudah dilakukan pada penelitian sebelumnya namun belum ada implementasi sistem. Adanya aplikasi presensi ini diharapkan dapat membantu dan mempermudah proses pengisian presensi di CV. Ladapedas Creative Surakarta yang saat ini masih belum menggunakan presensi serta dapat membantu dalam masalah perhitungan gaji pegawai.

2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalahnya adalah "Bagaimana Mengimplementasikan Aplikasi Kepegawaian CV. Ladapedas Creative Surakarta Berbasis Web?".

3. Penguasaan Dasar

Penguasaan dasar PHP, MySQL, dan CodeIgniter merupakan kunci untuk membangun Aplikasi Kepegawaian CV. Ladapedas Creative Surakarta Berbasis Web.

4. Pengumpulan Data

Tahap pengumpulan data dalam penelitian ini menggunakan metode observasi, wawancara dan studi literatur. Tujuan dari pengumpulan data ini adalah untuk mengetahui kebutuhan dari CV. Ladapedas Creative Surakarta.

5. Observasi Sistem

Pada penelitian ini dilakukan pengamatan terhadap aplikasi yang sejenis sebagai referensi dalam membangun aplikasi ini.

6. Analisis Perancangan Berbasis Objek

Analisis dan perancangan pada penelitian ini dilakukan dengan bagaimana aplikasi ini nantinya dibuat untuk memecahkan permasalahan yang ada.

7. Implementasi Aplikasi Kepegawaian pada CV. Ladapedas Creative Surakarta Berbasis Web

Implementasi aplikasi merupakan tahap membuat aplikasi kepegawaian menggunakan bahasa pemrograman PHP serta membuat database dengan menggunakan MySQL untuk menyimpan data-data yang dibutuhkan sistem.

8. Pengujian Sistem

Tahap pengujian sistemini menggunakan metode WebQual. Pengujian sistem bertujuan untuk mengetahui kelebihan dan kekurangan aplikasi serta untuk mengetahui apakah aplikasi tersebut layak untuk digunakan.

9. Penerapan Sistem dan Dokumentasi

Tahap penerapan sistem berarti aplikasi kepegawaian telah lulus uji aplikasi dan telah digunakan pada CV. Ladapedas Creative Surakarta untuk membantu proses pengisian presensi dan proses perhitungan gaji pegawai, serta membuat dokumentasi dari seluruh kegiatan penyusunan Tugas Akhir.

2.3 Teori Pendukung

2.3.1 Implementasi

Implementasi adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci. Implementasi biasanya dilakukan setelah perencanaan sudah dianggap *fix*. Secara sederhana implementasi bisa diartikan pelaksanaan atau penerapan (Rahma Farah Ningrum, 1 Agustus 2014)

2.3.2 Aplikasi

Menurut Sukatmi: 2018 dalam jurnalnya yang berjudul Aplikasi Absensi Siswa Berbasis Web Dengan Dukungan SMS Gateway pada SMK Krida Wisata Bandar Lampung aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows, permainan dan sebagainya (Sukamti, 2018).

2.3.3 Kepegawaian

Pegawai merupakan kekayaan utama suatu perusahaan karena tanpa keikutsertaan mereka, aktifitas tidak akan terjadi. Kepegawaian merupakan suatu badan yang mengurusi administrasi pegawai. Dimana keberadaan kepegawaian ini lebih berfungsi pada urusan administrasi pengankatan, kepangkatan, penggajian, mutasi, pemberhentian dan pensiunan (Nurbaity, 2010).

2.3.4 Presensi

Menurut Sofika Enggari: 2016 dalam jurnalnya yang berjudul Perancangan Sistem Informasi Absensi Siswa MTsN Pariaman Selatan Dengan Menggunakan PHP MySQL dan SMS Gateway presensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi atau komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila seawktu-waktu diperlukan oleh pihak yang berkepentingan (Enggari, 2016).

2.3.5 Gaji

Gaji merupakan pembayaran atas jasa yang dilakukan oleh karyawan sebagai pengganti atas pekerjaannya. Gaji adalah suatu hal yang penting bagi setiap karyawan yang bekerja dalam suatu perusahaan, karena dengan gaji yang diperoleh seseorang dapat memenuhi kebutuhan hidupnya (Gumilar, 2018).

2.3.6 Analisis Sistem

Analisis sistem merupakan sebah tahapan dalam pengembangan sistem yang akan menghasilkan berbagai dokumen yang menyajikan rencana pekerjaan yang akan dilaksanakan untuk mengembangkan sistem tersebut. Sedangkan analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui.

Maka suatu analisis sistem merupakan penguraian dari sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-

kesempatan, hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Ada beberapa langkah-langkah dasar dari analisis sistem berikut ini:

1. *Indentify*, yaitu mengindentifikasi masalah.
2. *Understand*, yaitu memahami kerja sistem yang ada.
3. *Analyze*,yaitu menganalisa sistem (Robert Marco & Hanif Al Fatta, 2015).

2.3.7 Perancangan Sistem

Perancangan sistem mulanya diawali dengan menentukan segala keperluan yang akan memenuhi apa yang dibutuhkan oleh sistem, siapa yang mengambil langkah dan bagaimana cara menyesuaikannya. Pada dasarnya perancangan sistem bergerak dari input menuju ke *output* sistem, yang terdiri dari *reports* dan *file* untuk memenuhi kebutuhan organisasi. Perancangan sistem merupakan sebuah penentuan proses data yang diperlukan oleh sistem baru, dan tahap-tahap dalam merancang sistem, meliputi:

1. Menyiapkan perancangan sistem secara rinci
   Analisis berkerjasama dengan pemakai dan mendokumentasikan rancangan sistem baru menggunakan peralatan tertentu.
2. Mengidentifikasi alternatife konfigurasi sistem
   Analisis harus mengidentifikasi konfigurasi peralatan komputer yang memberi hasil sesuai dengan yang diperlukan untuk menyelesaikan proses.
3. Mengevaluasi alternatife konfigurasi sistem
   Analisis berkerjasama dengan manager untuk mengevaluasi alternatife.
4. Memilih konfigurasi terbaik.
5. Menyiapkan usulan implementasi.
6. Menyiapkan usulan penerapan yang memberi ringkasan tugas-tugas penerapan yang harus dilakukan dari dokumentasi perancangan.
7. Menyetujui dan menolak penerapan sistem (Robert Marco & Hanif Al Fatta, 2015).

2.3.8 Analisis Berorientasi Objek

Pada jurnal Ilmu Komputer karya Gushelmi dan Deded Ramad Kamda analisis berorientasi objek (OOA-*Object Oriented Analysis*) adalah tahapan perangkat lunak dengan menentukan spesifikasi sistem, sering orang menyebutnya sebagai SRS (*System Requirement Spesification*) dan mengidentifikasi kelas-kelas serta hubungannya satu terhadap yang lain (Gushelmi & Deded Ramad Kamda, 2012).

2.3.9 Perancangan Berorientasi Objek

Sasaran dari perancangan berorientasi objek (OOD-*Object Oriented Design*) adalah merancang kelas-kelas yang teridentifikasi selama tahap analisis dan antarmuka pengguna. Selama tahap ini mengidentifikasi dan menambah beberapa objek dan kelas yang mendukung implementasi dan spesifikasi kebutuhan. Perancangan berbasis objek dan analisis berorientasi objek adalah topik-topik yang terpisah namun keduanya saling bekerja sama dengan erat. Aktifitas dan fokus dari analisis berorientasi objek dan perancangan berbasis objek saling bekerja sama, saling melengkapi (Gushelmi & Deded Ramad Kamda, 2012).

2.3.10 *Object Oriented Programming* (OOP)

Pemrograman berorientasi objek merupakan pradigma pemrograman yang berorientasikan kepada objek. Ini adalah jenis pemrograman di mana programmer mendefinisikan tidak hanya tipe data dari sebuah struktur data, tetapi juga jenis operasi (fungsi) yang dapat diterapkan pada struktur data. Dengan cra ini, struktur data menjadi objek yang meliputi data dan fungsi. Selain itu, pemrogram dapat membuat hubungan antara satu benda dan lainnya. Sebagai contoh, objek dapat mewarisi karakteristik dari objek lain (Subiyantoro, 2013).

2.3.11 *Unified Modeling Language (UML)*

Menurut (Rosa & Shalahuddin, 2016) dijelaskan pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisari agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berosientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (UML).

Menurut (Rosa & Shalahuddin, 2016) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Tujuan utama pemodelan *use case* adalah :

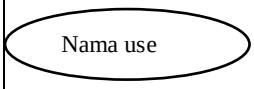
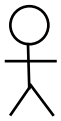
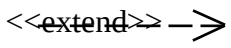
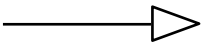
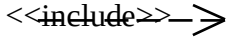
1. Memutuskan dan mendeskripsikan kebutuhan-kebutuhan fungsionalitas sistem.
2. Memberikan deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan, sehingga model *use case* digunakan diseluruh proses pengembangan untuk komunikasi dan menyediakan basis untuk pemodelan berikutnya yang mengacu sistem harus memberikan fungsionalitas yang dimodelkan pada *use case*.
3. Menyediakan basis untuk melakukan pengujian sistem yang memverifikasi sistem. Menguji apakah sistem telah memberikan fungsionalitas yang diminta.
4. Menyediakan kemampuan melacak kebutuhan fungsionalitas menjadi kelas-kelas dan operasi-operasi aktual di sistem.

Setiap *use case* dilengkapi dengan skenario. Skenario *use case* adalah alur jalannya proses use case dari sisi aktor dan sistem.

Ada 6 (enam) macam diagram dalam *Unified Modeling Language* (UML), yaitu :

1. *Use Case Diagram*

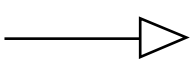
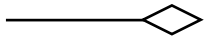
Tabel 2.1 Simbol-simbol *Use Case Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		*Use case*	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama *use case.*
2.		*Actor*	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanyanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.		*Association*	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
4.		*Extend*	Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.
5.		*Generalization*	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.		*Include*	Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini.

2. *Class Diagram*

Menurut (Rosa & Shalahuddin, 2016) Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas.Berikut merupakan tabel penjelasan mengenai simbol *class diagram* dapat dilihat pada Tabel 2.2.

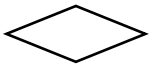
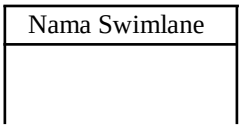
Tabel 2.2 Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.	Nama kelas + Atribut + Operasi()	*Class*	Kelas pada struktur sistem
2.		*Interface*	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.		*Association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity.*
4.		*Directed association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan *multiplicity.*
5.		*Generalization*	Relasi antar kelas dengan makna generalisasi - spesialisai (umum – khusus).
6.		*Dependency*	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.		*Aggregation*	Relasi antar kelas dengan makna semua bagian (*whole-part*).

3. *Activity Diagram*

Menurut (Rosa & Shalahuddin, 2016) Diagram aktivitas atau *activity diagram* menggambarkan aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol *Activity Diagram* ditujukan pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Activity Diagram*

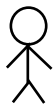
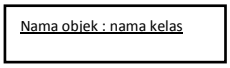
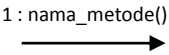
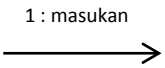
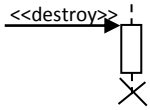
NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		*Decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		*Join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Status akhir	Status akhir yang dilakukan sistem.
6.		*Swimlane*	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

4. *Sequence Diagram*

Menurut (Rosa & Shalahuddin, 2016) *Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen diagram maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang

diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. *Sequence diagram* dapat dilihat pada Tabel 2.4.

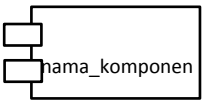
Tabel 2.4 Tabel *Sequence Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanyanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		*Lifeline*	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5.		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
8.		Pesan tipe *destroy*	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

5. *Component Diagram*

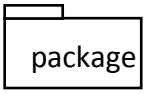
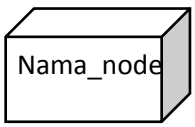
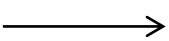
Menurut (Rosa & Shalahuddin, 2016) Diagram komponen atau *component* diagram menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Simbol *component* diagram dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol *Component Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1.		*Component*	Komponen sistem.
2.		*Dependency*	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
3.		*Interface*	Sama dengan konsep *interface* pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
4.		*Link*	Relasi antar komponen.

6. *Deployment Diagram*

Tabel 2.6 Simbol *Deployment Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		*Package*	*Package* merupakan sebuah bungkusan dari satu atau lebih *node.*
2.		*Node*	Biasanya mengacu pada perangkat keras (*hardware*), perangkat lunak yang tidak dibuat sendiri (*software*), jika didalam *node* disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelum pada diagram komponen.
3.		*Dependency*	Kebergantungan antar *node*, arah panah mengarah pada *node* yang dipakai.

Menurut (Rosa & Shalahuddin, 2016) *Deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan sistem tambahan yang menggambarkan rancangan device, node, dan hardware.Sistem client, sistem terdistribusi murni, rekayasa ulang aplikasi. Berikut merupakan tabel mengenai Simbol *deployment* diagram yang ditunjukkan pada Tabel 2.6.

2.3.12 *Website*

Website merupakan kumpulan dari halaman-halaman web yang berhubungan dengan file-file lain yang terkait. Dalam sebuah website terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi website. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah ke halaman lain yang terdapat dalam website tersebut (Hendrianto, 2014).

2.3.13 PHP

Pada Indonesian Journal on Networking and Security, PHP adalah bahasa *server-sidescripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan sepenuhnya akan dijalankan di server tetapi disertakan pada dokumen HTML. Pembuatan web ini merupakan kombinasi antara PHP sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman web (Hendrianto, 2014).

2.3.14 Database

Database adalah kumpulan informasi yang disusun dan merupakan suatu kesatuan yang utuh yang disimpan di dalam perangkat keras (komputer) secara sistematis sehingga dapat diolah menggunakan perangkat lunak. Dengan sistem tersebut data yang terhimpun dalam suatu database dapat menghasilkan informasi yang berguna (Ganda Yoga Swara & Yunes Pebriadi, 2016).

2.3.15 MySQL

Pada Indonesian Journal on Networking and Security, MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client server* melibatkan *server daemon* MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan disisi *client*.MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TEX, mengaku mampu menyimpan data lebih dari 40 *databases*, 10.000 tabel, dan sekitar 7.000.000 baris totalnya kurang lebih 100 Gigabyte data (Hendrianto, 2014).

2.3.16 *Codeigniter*

Menurut Mara Destiningrum: 2017 dalam jurnalnya yang berjudul Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan *Framework Codeigniter* (Studi Kasus: Rumah Sakit Yukum Medical Centre), *Codeigniter* adalah sebuah *framework* PHP yang bersifat *open source* dan mengunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal. Dalam situs resmi *codeigniter*, (Official Website Codigniter,2002) menyebutkan bahwa *codeigniter* merupakan *framework* PHP yang kuat dan sedikit *bug*. *Codeigniter* ini dibangun untuk para pengembang dengan bahasa pemrograman PHP yang membutuhkan alat untuk membuat web dengan fitur lengkap (Mara Destiningrum & Qhadli Jafar Adrian, 2017).

2.3.17 *Bootstrap*

Menurut (Wahyu, 2014) *Bootstrap* merupakan sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dapat dibuat oleh alat bantu ini memiliki halaman tampilan yang sama atau mirip dengan tampilan halaman *twitter* atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. Tampilan *website* yang dibuat *bootstrap*akan menyesuaikan ukuran layar dari

browser yang kita gunakan baik *desktop*, *tablet* ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun di non-aktifkan sesuai keinginan. Dengan *bootstrap* kita juga bisa membangun web dinamis ataupun statis.

2.3.18 Pengujian Sistem

2.3.18.1 Metode WebQual

Pengujian perangkat lunak adalah sebuah elemen topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan atau *customer* (Rosa & Shalahuddin, 2016).

Metode WebQual merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari metode servqual yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. Ada banyak faktor (variabel) yang menentukan kualitas layanan website, namun variabel yang akan digunakan dalam penelitian ini mengacu pada teori webqual untuk mengukur kualitas layanan websit edari perspektif pengguna. Terdapat tiga dimensi yang mewakili kualitas suatu *website*, yaitu kegunaan (*usability*), kualitas informasi (*informationquality*) dan interaksi layanan (*serviceinteraction*). Masing-masing dimensi terdiri dari beberapa peryataan yang ditunjukkan oleh tabel di bawah ini:

Tabel 2.7 Dimensi Kegunaan

No	Deskripsi Indikator
1.	Pengguna merasa mudah untuk mempelajari pengoperasian *website*.
2.	Interaksi antara *website* dengan pengguna jelas dan mudah dipahami.
3.	Pengguna merasa mudah untuk bernavigasi dalam *website.*
4.	Pengguna merasa *website* mudah digunakan.
5.	*Website* memiliki tampilan yang menarik.
6.	Desain sesuai dengan jenis *website.*
7.	*Website* mengandung kompetensi.
8.	*Website* menciptakan pengalaman positif bagi pengguna.

Tabel 2.8 Dimensi Kualitas Informasi

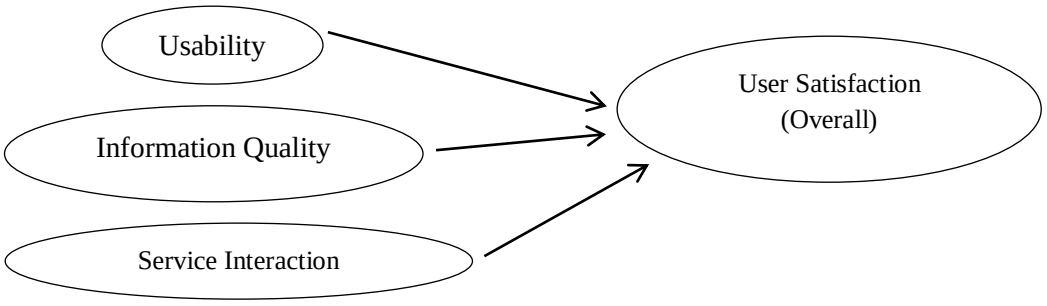
No	Deskripsi Indikator
1.	*Website* menyediakaninformasiyang akurat.
2.	*Website* menyajikaninformasiyangterpercaya.
3.	*Website* menyediakaninformasi *upto date.*
4.	*Website* meyediakaninformasiyangrelevan.
5.	*Website*menyediakan informasi yang mudah dimengerti.
6.	*Website* menyediakaninformasiyangdetail.
7.	*Website*menyajikan informasi dengan format yangsesuai.

Tabel 2.9 Dimensi Interaksi Layanan

No	Deskripsi Indikator
1.	*Website* memilikireputasiyangbaik.
2.	Pengguna merasa aman untuk melakukan transaksi.
3.	Pengguna merasa aman terhadap informasi pribadinya.
4.	*Website* memberi ruang untuk personalisasi.
5.	*Website* memberi ruang untuk komunitas.
6.	*Website* memberikan kemudahan untuk berkomunikasidengan organisasi.
7.	Penggunamerasayakinbahwabarang/jasaakan dikirim sebagaimanayangtelah dijanjikan.

Tabel 2.10 Dimensi Keseluruhan

No.	Deskripsi Indikator
1.	Pendapat secara umum tentang *website* ini.



Gambar 2.2 Hipotesis Penelitian (Silalahi, 2015)

Beberapa tahap yang harus dilakuakan untuk melakukan pengujian WebQual antara lain:

1. Uji Instrumen

Menurut (Surya, 2017) uji instrumen digunakan untuk mengetahui deskripsi mengenai variabel-variabel dalam penelitian, uji instrumen terdiri dari uji validitas dan uji reliabilitas.

1.1 Uji Validitas

Validitas mengandung dua bagian yaitu bahwa instrumen pengukuran adalah mengukur secara aktual konsep dalam pertanyaan dan bukan beberapa konsep yang lain; dan bahwa konsep dapat diukur secara akurat. Oleh karena itu, suatu instrumen pengukur bisa dikatakan valid jika mengukur apa yang hendak diukur dan mempu mengungkap data tentang karakteristik gejala yang diteliti secara tepat (Bailey, dalam Silalahi, 2013).

1.2 Reliabilitas

Reliabilitas adalah derajat sejauh mana ukuran menciptakan respon yang sama sepanjang waktu dan lintas situasi. Suatu alat ukur dikatakan reliabel jika hasil pengukuran dari alat ukur tersebut stabil dan konsisten (Silalahi, 2012). Dengan demikian reliabel ádalah suatu keadaan di mana instrumen penelitian tersebut akan tetap menghasilkan data yang sama meskipun disebarkan pada sampel yang berbeda dan pada waktu yang berbeda. Uji reliabilitas akan dilakukan dengan menggunakan uji statistik *cronbach's alpha* (α) dengan ketentuan bahwa variabel yang diteliti dinyatakan *reliabel* apabila nilai *cronbach's alpha* (α) adalah di atas 0,6 (Ghozali, 2014).

2. Uji Asumsi Klasik

2.1 Uji Normalitas

Menurut (Surya, 2017) uji normalitas ini bertujuan untuk menguji apakah dalam model regresi, variabel pengganggu atau residual memiliki distribusi secara normal. Uji T dan Uji F mengasumsikan bahwa residual mengikuti distribusi secara normal, kalau asumsi ini dilanggar maka uji statik menjadi tidak valid.

Salah satu cara untuk mendeteksi apakah residual berdistribusi normal atau tidak yaitu dengan uji statik *one sample* Kolmogorov-Smirnov. Dasar penggambilan keputusannya adalah:

a) Jika hasil *one sample* Kolmogorov-Smirnov diatas tingkat signifikansi 0,05 menunjukkan pola distribusi normal, maka model regresi tersebut memenuhi asumsi normalitas
b) Jika hasil *one sample* Kolmogorov-Smirnov dibawah tingkat signifikansi 0,05 menunjukkan pola distribusi normal, maka model regresi tersebut tidak memenuhi asumsi normalitas

2.2 Uji Multikolinearitas

Menurut (Surya, 2017), pengujian ini bertujuan untuk menguji apakah model regresi ditemukan adanya korelasi antara variabel bebas (*independent)*. Model regresi yang baik seharusnya tidak terjadi korelasi diantara variabel-variabel bebas. Jika variabel saling berkorelasi maka variabel-variabel ini tidak ortogonal.

Varibel ortogonal merupakan variabel bebas (*independent)* yang nilai korelasi antar sesama variabel bebas sama dengan nol. Salah satu cara untuk mengetahui ada tidaknya multikolinearitas pada suata model adalah dengan melihat nilai tolerance dan VIF (Variance Inflation Factor)

a) Jika nilai tolerance > 0,10 dan VIF < 0,10, maka dapat diartikan bahwa tidak terdapat multikolinearitas pada penelitian tersebut
b) Jika nilai tolerance < 0,10 dan VIF > 0,10, maka dapat diartikan bahwa terdapat gangguan multikolinearitas pada penelitian tersebut

2.3 Uji Heteroskedastisitas

Menurut (Surya, 2017) pengujian ini bertujuan untuk menguji apakah dalam model regresi terjadi ketidaksamaan varian dari residual satu pengamatan ke pengamatan yang lain. Jika variance dari residual satu ke pengamatan lain tetap, maka disebut homoskedastisitas dan jika berbeda disebut heteroskedastisitas adalah dengan melihat ada atau tidak adanya pola tertentu pada grafik Scatter Plot dengan ketentuan:

a) Jika terdapat pola tertentu, seperti titik-titik yang ada membentuk pola tertentu yang teratur maka menunjukkan telah terjadi heteroskedastisitas
b) Jika tidak ada pola yang kelas, serta titik-titik menyebar diatas dan dibawah angka 0 pada sumbu Y, maka tidak terjadi heteroskedastisitas

3. Analisis Regresi Linier Berganda

Menurut (Surya, 2017) metode analisis regresi linear berganda berfungsi untuk mengetahui pengaruh atau hubungan variabel *independent* dengan variabel *dependent*. Pengujian regresi linear berganda ini dilakukan dengan melakukan uji koefisien determinasi, uji parsial (Uji T) dan uji simultan (Uji F).

3.1 Uji F (Uji Simultan)

Menurut (Ghozali, 2011), Uji F digunakan untuk menunjukkan apakah semua variabel *indenpendent* yang dimasukkan dalam model mempunyai pengaruh secara bersama-sama terhadap variabel *dependent*. Dasar penerimaan atau penolakan hipotesis dapat dilihat dengan membandingkan F hitung dengan F tabel, jika F hitung > F tabel maka Ho ditolak dan Ha diterima.

3.2 Uji T (Uji Persial)

Menurut (Ghozali, 2011) uji T pada dasarnya menunjukkan seberapa jauh pengaruh suatu variabel *independent* secara individual dalam menerangkan variabel *dependent*. Penerimaan atau penolakan hipotesis dilakukan dengan kriteria sebagai berikut :

a. Jika nilai *sig*<0,05 atau t hitung > t tabel maka terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terkait).
b. Jika nilai sig>0,05 atau t hitung < t tabel maka tidak terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terkait).

3.3 Koefisien Determinasi (R^2)

Koefisien determinasi merupakan koefisian yang nilainya dimaksudkan untuk mengetahui seberapa besar variasi perubahan dalam satu variabel independen. Koefisien korelasi digunakan untuk menentukan koefisien determinasi. Pada konteks ini, koefisien determinasi merupakan kuadrat dari koefisien korelasi yang dinotasi dengan R^2 (Surya, 2017). Oleh karena itu, semakin kuat korelasi diantara variabel yang diamati maka semakin besar pula koefisien determinasi yang dihasilkan. Koefisien determinasi dinyatakan dalam persen (%) sehingga harus dikalikan dengan 100%. Artinya adalah bahwa persentase dari variasi perubahan dalam variabel Y adalah disebabkan oleh adanya variasi perubahan dalam variabel X.