

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa tinjauan pustaka yang digunakan sebagai dasar implementasi sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta berbasis *web* antara lain:

Artikel ilmiah yang berjudul Otomatisasi Sistem Pembuatan Berita Acara dan Pelaporan Honor Ujian Skripsi Fakultas Teknologi Informasi Universitas Kristen Satya Wacana (Wahid, 2016). Penelitian pada Universitas Kristen Satya Wacana tersebut memfokuskan pada pembuatan otomatisasi sistem berita acara dikarenakan proses pembuatan dokumen pelaporan honor dosen memakan waktu yang lama. Dimana dokumen berita acara ujian dibuat oleh bagian TA FTI UKSW berdasarkan data pendaftar ujian yang disimpan didalam *microsoft excel* kemudian membuat dokumen berita acara satu persatu didalam *microsoft word*. Setelah dilakukan pengujian Tugas Akhir, bagian TA FTI UKSW akan merekapitulasi data hasil ujian yang telah dilakukan. Rekapitulasi ini mencakup nilai dari dosen pembimbing, nilai dari dosen penguji, banyaknya menguji yang dilakukan oleh setiap dosen penguji, IPK mahasiswa, jumlah mahasiswa bimbingan dari setiap dosen serta menghitung pengujian yang dilakukan setiap dosen.

Proses ini memakan waktu 2-3 minggu untuk membuat dokumen pelapor honor semua dosen. Pembuatan berita acara ujian, perhitungan nilai hasil ujian dan perhitungan total honor setiap dosen penguji berdasarkan proses yang ada saat ini menimbulkan kesalahan atau kekeliruan dalam pembuatan berita acara ujian. Sehingga dibutuhkan sebuah sistem yang mampu mempercepat proses dan mengurangi kesalahan dalam pembuatan berita acara ujian, perhitungan nilai ujian, dan perhitungan total honor setiap dosen penguji.

Berdasarkan permasalahan tersebut penulis mengembangkan aplikasi otomatisasi sistem pembuatan berita acara dan pelaporan honor ujian skripsi FTI UKSW berbasis *website* sehingga aplikasi ini dapat digunakan diberbagai

platform. Aplikasi tersebut dibangun menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*).

Penelitian kedua dilakukan oleh (Andi & Inggih, 2016) yang menciptakan sistem informasi pencatatan surat masuk Kantor Camat Kampar Kiri Kabupaten Kampar Provinsi Riau yang ditulis dalam jurnal rekayasa dan manajemen sistem informasi. Pada penelitian ini mereka membangun sebuah sistem pencatatan surat masuk pada kantor Camat Kampar Kiri, sistem tersebut dibangun dengan menggunakan teknik *Object Oriented Analysis and Design (OOAD)*. *Tool* yang digunakan adalah empat buah diagram *Unified Modeling Language (UML)*, yaitu: *use case diagram*, *sequence diagram*, *activity diagram* dan *class diagram*. Melalui penelitian tersebut terciptalah sebuah sistem yang mampu menyelesaikan masalah yang ada mengenai tata kelola surat dengan baik pada kantor Camat Kampar Kiri Kabupaten Kampar Provinsi Riau.

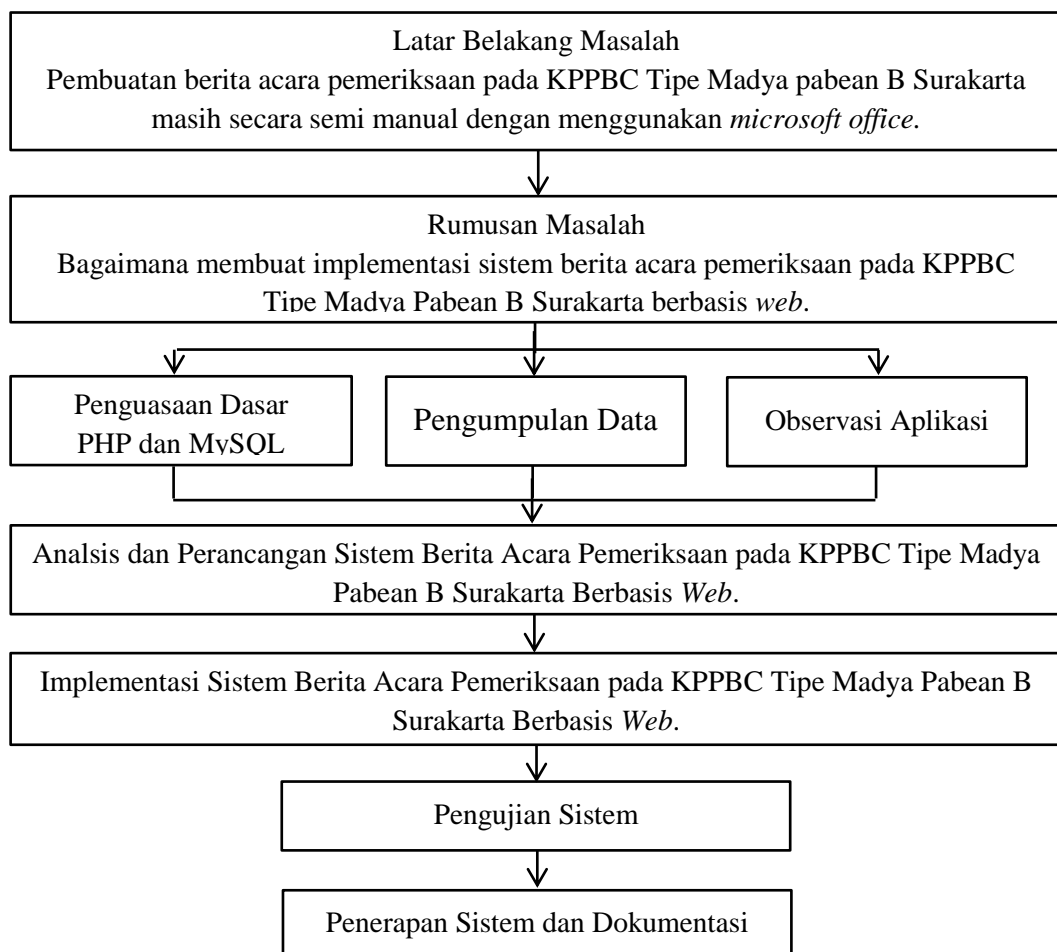
Penelitian ketiga dilakukan oleh (Sukron & Kondar, 2016) yang berjudul analisis dan perancangan sistem informasi manajemen arsip berbasis *web* pada Sekolah Tinggi Ilmu Tarbiah (STIT) Kabupaten Tebo melalui jurnal manajemen sistem informasi. Masalah yang terjadi pada penelitian ini adalah penggunaan teknologi komputer pada bagian tata usaha kampus STIT Kabupaten Tebo yang belum optimal dan dalam penataan arsip kampus masih dilakukan secara manual. Hal tersebut terlihat dari penggunaan aplikasi dalam penataan arsip-arsip masih menggunakan aplikasi *microsoft office*. Petugas sering mengalami kesulitan ketika harus mencari arsip yang sudah lama, karena harus membongkar gudang arsip untuk mencari surat yang dibutuhkan.

Media penyimpanan yang digunakan adalah *flashdisk*, *compact disk* dan *hard disk*. Serta pengarsipan yang dilakukan yaitu dengan menyimpan dokumen didalam sebuah bindex (*file box*) dan ditempatkan di ruang program studi. Banyaknya arsip akan menyulitkan pihak yang hendak meminjam dokumen tersebut, karena harus mencari pada tumpukan arsip. Dari permasalahan tersebut peneliti bertujuan untuk menganalisis dan merancang aplikasi manajemen arsip yang terintegrasi yang bertujuan untuk mengurangi resiko hilangnya arsip, mempercepat dalam proses pencarian arsip. Metode pendekatan yang digunakan

peneliti adalah pendekatan *object oriented* yang dalam menggambarkan seluruh proses dan objeknya menggunakan UML (*Unified Modeling Language*), yaitu *use case diagram*, *class diagram* dan *activity diagram*.

2.2 Kerangka Pemikiran

Berikut ini adalah tahapan kerangka pemikiran dalam melakukan implementasi sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta berbasis *web*.



Gambar 2.1 Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah :

1. Latar Belakang Masalah

Pembuatan berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta masih dilakukan secara semi manual. Pembuatan berita acara

menggunakan *microsoft office* yaitu *microsoft word* dan *microsoft excel* untuk melakukan pencatatan dan pengolahan data hasil pemeriksaan. Walaupun penulisan berita acara telah terkomputerisasi namun hal tersebut masih kurang efektif, dikarenakan petugas harus menginputkan data satu persatu pada *microsoft excel*, sedangkan petugas harus mengisi formulir pemeriksaan tersebut secara manual dengan menulis tangan dan harus membawa formulir tersebut kemanapun saat bertugas. Karena banyaknya data yang masuk mengharuskan petugas untuk mengolah data secara cepat dan akurat. Pengolahan data yang dilakukan secara semi manual menjadi kendala terselesaikannya pekerjaan tersebut seperti keterlambatan dalam pengolahan data.

2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalahnya adalah “Bagaimana membuat implementasi sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta berbasis *web*?”.

3. Penguasaan dasar PHP dan MySQL

Penguasaan dasar PHP dan MySQL merupakan suatu kunci terciptanya sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta berbasis *web* tersebut, karena pembuatan sistem berita acara pemeriksaan ini menggunakan bahasa pemrograman PHP dan menggunakan *database* MySQL sebagai media penyimpan datanya.

4. Pengumpulan data

Tahap pengumpulan data pada penelitian ini melalui observasi, wawancara dan studi literatur. Pengumpulan data bertujuan untuk mengetahui permasalahan dan kebutuhan suatu instansi.

5. Observasi aplikasi

Observasi aplikasi bertujuan untuk mencari gambaran sistem yang akan dibuat dengan melihat beberapa aplikasi yang sudah ada.

6. Analisis dan perancangan sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta berbasis *web*

Analisis dan perancangan sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta bertujuan untuk mengetahui kelemahan serta kekurangan dari sistem yang telah berjalan saat ini, serta membuat perancangan sistem baru yang diharapkan dapat membantu menyelesaikan suatu pekerjaan agar lebih efektif dan efisien. Perancangan sistem pada penelitian ini menggunakan UML (*Unified Modeling Language*).

7. Implementasi sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta berbasis *web*

Implementasi sistem berita acara pemeriksaan pada KPPBC Tipe Madya Pabean B Surakarta menggunakan bahasa pemrograman PHP dan database MySQL sebagai media penyimpanan datanya.

8. Pengujian Sistem

Pengujian sistem merupakan tahap setelah sistem berhasil dibuat. Pengujian sistem bertujuan untuk mengetahui kelebihan, kelemahan serta mengetahui kelayakan suatu sistem untuk digunakan. Pengujian sistem pada penelitian ini menggunakan metode *WebQual*.

9. Penerapan Sistem dan Dokumentasi

Tahap penerapan sistem disini berarti sistem berita acara pemeriksaan telah lolos uji aplikasi dan telah digunakan pada KPPBC Tipe Madya Pabean B Surakarta untuk membantu menyelesaikan suatu pekerjaan, serta membuat dokumentasi dari seluruh kegiatan penyusunan Tugas Akhir.

2.3 Teori Pendukung

2.3.1 Berita Acara

Berita acara adalah keterangan mengenai kejadian atau peristiwa yang dibuat mengenai waktu terjadi, tempat, keterangan, dan petunjuk lain tentang

suatu perkara atau peristiwa. (Kamus Besar Bahasa Indonesia edisi keempat, 2014).

2.3.2 Sistem

Menurut (Abdul, 2014) sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Hal (2001) dalam Abdul Khadir (2014 : 62-63) tujuan utama sistem yang umum ada tiga macam yaitu:

1. Untuk mendukung fungsi kepengurusan manajemen.
2. Untuk mendukung pengambilan keputusan manajemen.
3. Untuk mendukung kegiatan operasi perusahaan.

2.3.3 Analisis Sistem

Menurut (Muhamad & Oktafianto, 2016) Analisis sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut berkerja dan berinteraksi untuk mencapai tujuan mereka. Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi dalam menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Tahapan ini sangat penting karena menentukan bentuk sistem yang harus dibangun. Tahapan ini bisa merupakan tahap yang mudah jika klien sangat paham dengan masalah yang dihadapi dalam organisasinya dan tahu betul fungsionalitas dari sistem informasi yang akan dibuat. Tetapi tahap ini bisa menjadi tahap paling sulit jika klien tidak bisa mengidentifikasi kebutuhannya dan tertutup pihak luar yang ingin mengetahui detail proses-proses bisnisnya.

2.3.4 Analisis Berorientasi Objek

Menurut (Rosa & Shalahuddin, 2016) Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) adalah tahapan untuk menganalisis spesifikasi atau kebutuhan akan sistem yang akan dibangun dengan konsep berorientasi

objek, apakah benar kebutuhan yang ada dapat diimplementasikan menjadi sebuah sistem berorientasi objek. Analisis ini sebaiknya dilakukan oleh orang-orang yang benar-benar memahami implementasi sistem yang berbasis atau berorientasi objek, karena tanpa pemahaman itu maka sistem yang dihasilkan bisa jadi tidak realistis jika diimplementasikan dengan berbasis objek. OOA biasanya menggunakan kartu CRC (*Component, Responsibility, Collaboration*) untuk membangun kelas-kelas yang akan digunakan atau menggunakan UML (*Unified Modeling Language*) pada bagian diagram *use case*, diagram kelas, dan diagram objek.

2.3.5 Perancangan Berorientasi Objek

Menurut (Rosa & Shalahuddin, 2016) Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini sangat beragam (aplikasi bisnis, *real-time*, *utility* dan sebagainya) dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi.

Saat ini sudah banyak bahasa pemrograman berorientasi objek. Banyak orang berfikir bahwa pemrograman berorientasi objek identik dengan bahasa Java. Memang bahasa Java merupakan bahasa yang paling konsisten dalam mengimplementasikan paradigma pemrograman berorientasi objek. Namun

sebenarnya bahasa pemrograman yang mendukung pemrograman berorientasi objek tidak hanya bahasa Java. Bahasa pemrograman yang mendukung pemrograman berorientasi objek antara lain: bahasa pemrograman *Smalltalk*, *Eiffel*, C++, PHP dan Java.

2.3.6 Unified Modeling Language (UML)

Menurut (Rosa & Shalahuddin, 2016) pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

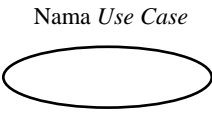


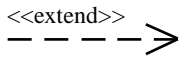
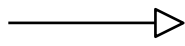
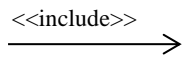
UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi penelitian.

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

2.3.6.1 Use Case Diagram

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. Simbol *use case diagram* dapat dilihat pada Tabel 2.1.

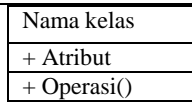






Tabel 2.1 Simbol *Use Case Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Use case</i>	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 <i>use case</i> .
2		<i>Actor</i>	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; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2.3.6.2 Class Diagram

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. (Rosa & Shalahuddin, 2016). Simbol *class diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Class Diagram*

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

2.3.6.3 Activity Diagram

Tabel 2.3 Simbol *Activity Diagram*


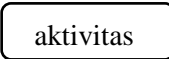
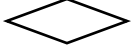


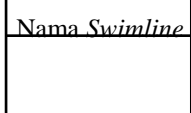
NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih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



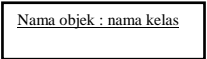

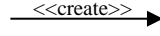
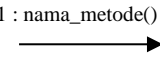
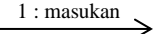
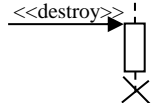
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh

sistem. (Rosa & Shalahuddin, 2016) Simbol *Activity Diagram* ditunjukkan pada Tabel 2.3.

2.3.6.4 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. (Rosa & Shalahuddin, 2016). Simbol *sequence* diagram ditunjukkan pada Tabel 2.4.

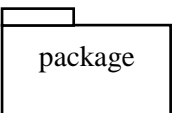
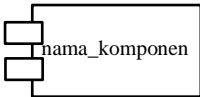



Tabel 2.4 Simbol *Sequence* Diagram

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; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2		<i>Lifeline</i>	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 dikirim.
8		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

2.3.6.5 Component Diagram

Diagram komponen atau *component* diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. *Component* diagram fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. *Component* diagram juga dapat digunakan untuk memodelkan *source code* program perangkat lunak, komponen *executable* yang dilepas ke *user*, basis data secara fisik, sistem yang harus beradaptasi dengan sistem lain dan *framework* sistem. (Rosa & Shalahuddin, 2016) Simbol *component* diagram dapat dilihat pada Tabel 2.5.

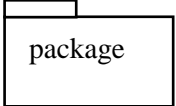
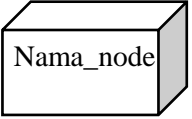
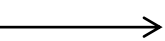

Tabel 2.5 Simbol *Component* Diagram.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
2		<i>Component</i>	Komponen sistem
3		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4		<i>Interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
5		<i>Link</i>	Relasi antar komponen

2.3.6.6 Deployment Diagram

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. (Rosa & Shalahuddin, 2016). Simbol *deployment* diagram ditunjukkan pada Tabel 2.6.

Tabel 2.6 Simbol *Deployment Diagram*

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

2.3.7 Website

Menurut (Fritz, 2014) *website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman ke halaman lain (*hyper text*), baik diantara halaman yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Halaman diakses dan dibaca melalui *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* dan aplikasi *browser* lainnya.

2.3.8 HTML

HTML atau *Hypertext Markup Language* merupakan bahasa pemrograman yang bisa digunakan untuk membuat aplikasi berbasis *web*. Bahasa pemrograman ini ditulis dalam berkas format ASCII, supaya dapat menghasilkan tampilan wujud yang terintegrasi. HTML adalah bahasa pemrograman *web* yang umum dan bisa digunakan serta mudah dipelajari. (Maya, Bowo, HS, & Irawan, 2015)

2.3.9 *PHP Hypertext Preprocessor (PHP)*

Menurut (Hakim, 2014), PHP merupakan bahasa pemrograman yang bisa digunakan untuk membuat halaman HTML. File .php yang dibuat akan diproses di dalam *server*, sedangkan halaman yang akan dikirim ke *browser* pengunjung hanyalah tampilan HTML-nya. Dengan PHP, halaman *website* yang dibuat akan menjadi dinamis, yakni dapat selalu berubah tanpa harus mengubah isi website secara manual. Informasi akan diproses ulang oleh *web server* sehingga akan didapatkan isi paling mutakhir dari halaman *web*.

2.3.10 *Database*

Menurut (Rosa & Shalahuddin, 2016) *database* atau basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System (DBMS)*.

2.3.11 *MySQL*

Menurut (Edy & Zaki, 2014) MySQL adalah sebuah *software database*. MySQL adalah sebuah database yang dapat berjalan sebagai *server* maupun *client*. Produk *database* MySQL ini memiliki *General Public Licence*, dan bersifat *open source*, sehingga bebas untuk digunakan, diedarkan, maupun dikembangkan kembali tanpa harus khawatir dengan hak cipta. MySQL merupakan tipe data rasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan.

2.3.12 *Cascading Style Sheet (CSS)*

Menurut (Hartono, 2013) CSS (*Cascading Style Sheet*) adalah kumpulan aturan-aturan pemformatan yang mengontrol tampilan konten dalam sebuah halaman *web*. Terdapat tiga jenis CSS, yaitu:

- a. *Inline style sheet* : cukup menambahkan atribut *style* di *tag* yang ingin kita berikan pemformatan.
- b. *Internal style sheet* : meletakkan aturan pemformatan dengan CSS dibagi `<head>` dari *html* dengan tambahan *tag* `<style>`.
- c. *External style sheet* : memisahkan antara file CSS dengan file HTML-nya.

2.3.13 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 dibuat oleh alat bantu ini memiliki tampilan halaman 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.

2.4 Pengujian WebQual

Menurut (Rosa & Shalahuddin, 2016) 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 (*customer*). Penelitian ini menggunakan model pengujian *WebQual*.

Menurut (Iman, 2012), *WebQual* merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari SERVQUAL yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. *WebQual* sudah mulai dikembangkan sejak tahun

1998 dan telah mengalami beberapa interaksi dalam penyusunan dimensi dan butir pertanyaannya.

Menurut teori *WebQual* pada jurnal yang disusun oleh (Anwariningsih, 2011), terdapat tiga dimensi yang mewakili kualitas suatu *website*, yaitu kegunaan (*usability*), kualitas informasi (*information quality*) dan interaksi layanan (*service interaction*). Masing-masing dimensi terdiri dari beberapa pernyataan yang ditunjukkan pada Tabel 2.7 sampai Tabel 2.9.

Tabel 2.7 Dimensi Kegunaan (*usability quality*)

No	Deskripsi Indikator
1	<i>Website</i> mudah untuk dioperasikan.
2	Interaksi dengan <i>website</i> sangat mudah dimengerti dan tidak membingungkan.
3	Pengguna merasa mudah untuk bernavigasi dalam <i>website</i> .
4	<i>Website</i> mudah untuk digunakan.
5	<i>Website</i> memiliki tampilan yang menarik.
6	Desain <i>website</i> sesuai dengan tipe <i>website</i> .
7	<i>Website</i> menunjukkan kemampuannya.
8	<i>Website</i> dapat memberikan pengaruh/ pengalaman positif bagi pengguna.

Tabel 2.8 Dimensi Kualitas Informasi (*information quality*)

No	Deskripsi Indikator
1	<i>Website</i> menyajikan informasi yang akurat.
2	Informasi yang disajikan <i>website</i> dapat dipercaya.
3	Informasi yang disajikan tepat waktu atau <i>up to date</i> .
4	Informasi yang disajikan relevan.
5	Informasi yang disajikan mudah dipahami.
6	Informasi yang disajikan sangat detail.
7	Informasi disajikan dalam format yang sesuai.

Tabel 2.9 Dimensi Kualitas Interaksi (*interaction quality*)

No	Deskripsi Instansi
1	<i>Website</i> memiliki reputasi yang baik.
2	Pengguna merasa aman untuk melakukan transaksi atau interaksi dengan <i>website</i> ..
3	<i>Website</i> menjaga informasi pribadi pengguna.
4	<i>Website</i> memberi ruang untuk personalisasi.
5	<i>Website</i> memberi ruang untuk komunitas.
6	<i>Website</i> memudahkan pengguna untuk berkomunikasi dengan organisasi.
7	Pengguna merasa yakin dengan layanan/informasi yang disediakan karena sesuai dengan yang dijanjikan.

Beberapa tahapan yang harus dilakukan untuk melakukan pengujian *webqual* antara lain:

2.4.1 Uji Instrumen

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

2.4.1.1 Uji Validitas

Menurut (Arikunto, 2010) validitas adalah suatu ukuran yang menunjukkan tingkat kehandalan dan kesahihan suatu instrumen. Instrumen yang valid berarti alat ukur yang digunakan untuk mendapatkan data itu valid. Menurut (Sugiyono, 2010) valid berarti instrumen tersebut dapat digunakan untuk mengukur apa yang hendak diukur. Suatu kuesioner dinyatakan valid apabila $r_{hitung} > r_{tabel}$, sebaliknya kuesioner dapat dinyatakan tidak valid apabila $r_{hitung} < r_{tabel}$. Pengujian validitas ini menggunakan batasan r_{tabel} dengan signifikansi 0,05 dan uji 2 sisi.

2.4.1.2 Uji Reliabilitas

Menurut (Arikunto, 2010) reliabilitas adalah sesuatu instrumen yang dapat dipercaya untuk digunakan sebagai alat pengumpul data karena instrumen tersebut sudah baik. Instrumen yang reliabel atau dapat dipercaya akan menghasilkan data

yang dapat dipercaya, instrumen yang reliabel mengandung arti bahwa instrumen tersebut harus baik sehingga mampu mengungkap data yang bisa dipercaya.

Berdasarkan (Iman, 2012) terdapat aturan praktis yang dapat diterapkan terkait dengan nilai alpha, jika $\alpha > 0,9$ berarti reliabilitas model sangat bagus, $\alpha > 0,8$ berarti reliabilitas model bagus, $\alpha > 0,7$ artinya reliabilitas model bisa diterima, $\alpha > 0,6$ berarti reliabilitas model layak, $\alpha > 0,5$ berarti reliabilitas model kurang bagus, dan $\alpha < 0,5$ berarti reliabilitas model tidak dapat diterima.

2.4.2 Uji Asumsi Klasik

Menurut (Yoedo, Susilowati, & Julianto, 2016), uji asumsi klasik digunakan sebelum uji regresi liliier berganda, ada data yang harus terpenuhi agar kesimpulan dari regresi bisa menjadi kuat, antara lain uji normalitas, uji multikolinearitas, uji heteroskedastisitas dan uji autokorelasi.

2.4.2.1 Uji Normalitas

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

Salah satu cara untuk mendeteksi apakah residual berdistribusi normal atau tidak yaitu dengan uji statistik one-sample kolmogorov-smirnov. Dasar pengambilan keputusan dari one-sample kolmogorov-smirnov adalah :

1. Jika hasil one-sample Kolmogorov-Smirnov diatas tingkat signifikansi 0,05 menunjukkan pola distribusi normal, maka model regresi tersebut memenuhi asumsi normalitas.
2. Jika hasil one-sample Kolmogorov-Smirnov dibawah tingkat signifikansi 0,05 tidak menunjukkan pola distribusi normal, maka modal regresi tersebut tidak memenuhi asumsi normalitas.

2.4.2.2 Uji Multikolinearitas

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

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

1. Jika nilai *tolerance* $> 0,10$ dan $VIF < 10$, maka dapat diartikan bahwa tidak terdapat multikolinearitas pada penelitian tersebut.
2. Jika nilai *tolerance* $< 0,10$ dan $VIF > 10$, maka dapat diartikan bahwa terjadi gangguan multikolinearitas pada penelitian tersebut.

2.4.2.3 Uji Heteroskedastisitas

Menurut (Prakosad, 2017), pengujian ini bertujuan untuk menguji apakah dalam model regresi terjadi ketidaksamaan varians dari residual satu pengamatan ke pengamatan yang lain. Jika variansi dari residual satu pengamatan ke pengamatan lain tetap, maka disebut homoskedastisitas dan jika berbeda disebut heteroskedastisitas. Untuk mengetahui adanya heteroskedastisitas adalah dengan melihat ada/tidaknya pola tertentu pada grafik Scatter Plot dengan ketentuan:

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

2.4.3 Analisis Regresi Linier Berganda

Menurut (Prakosad, 2017), metode analisis regresi linear berganda berfungsi untuk mengetahui pengaruh/hubungan variabel *independent* (bebas) dengan variabel *dependent* (terikat). Pengujian regresi linear berganda ini

dilakukan dengan melakukan uji koefisien determinasi, uji parsial (Uji T) dan uji simultan (Uji F)

2.4.3.1 Uji F (Uji Simultan)

Menurut (Ghozali, 2011), uji F digunakan untuk menunjukkan apakah semua variabel independent 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 H_0 ditolak dan H_a diterima.

2.4.3.2 Uji T (Uji Parsial)

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

1. Jika nilai $sig < 0,05$ atau $t_{hitung} > t_{tabel}$ maka terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terikat).
2. Jika nilai $sig > 0,05$ atau $t_{hitung} < t_{tabel}$ maka tidak terdapat pengaruh variabel X (variabel bebas) terhadap variabel Y (variabel terikat).

2.4.3.3 Koefisien Determinasi (R^2)

Menurut (Prakosad, 2017), koefisien determinasi (R^2) pada intinya mengukur seberapa jauh kemampuan model menerangkan variasi variabel *dependent*. Range nilainya antara 0-1, apabila nilai R^2 kecil berarti kemampuan variabel-variabel *independent* dalam menjelaskan variasi variabel *dependent* sangat terbatas, dan sebaliknya apabila R^2 besar berarti kemampuan variabel-variabel *independent* dalam menjelaskan variasi variabel *dependent* besar.