

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

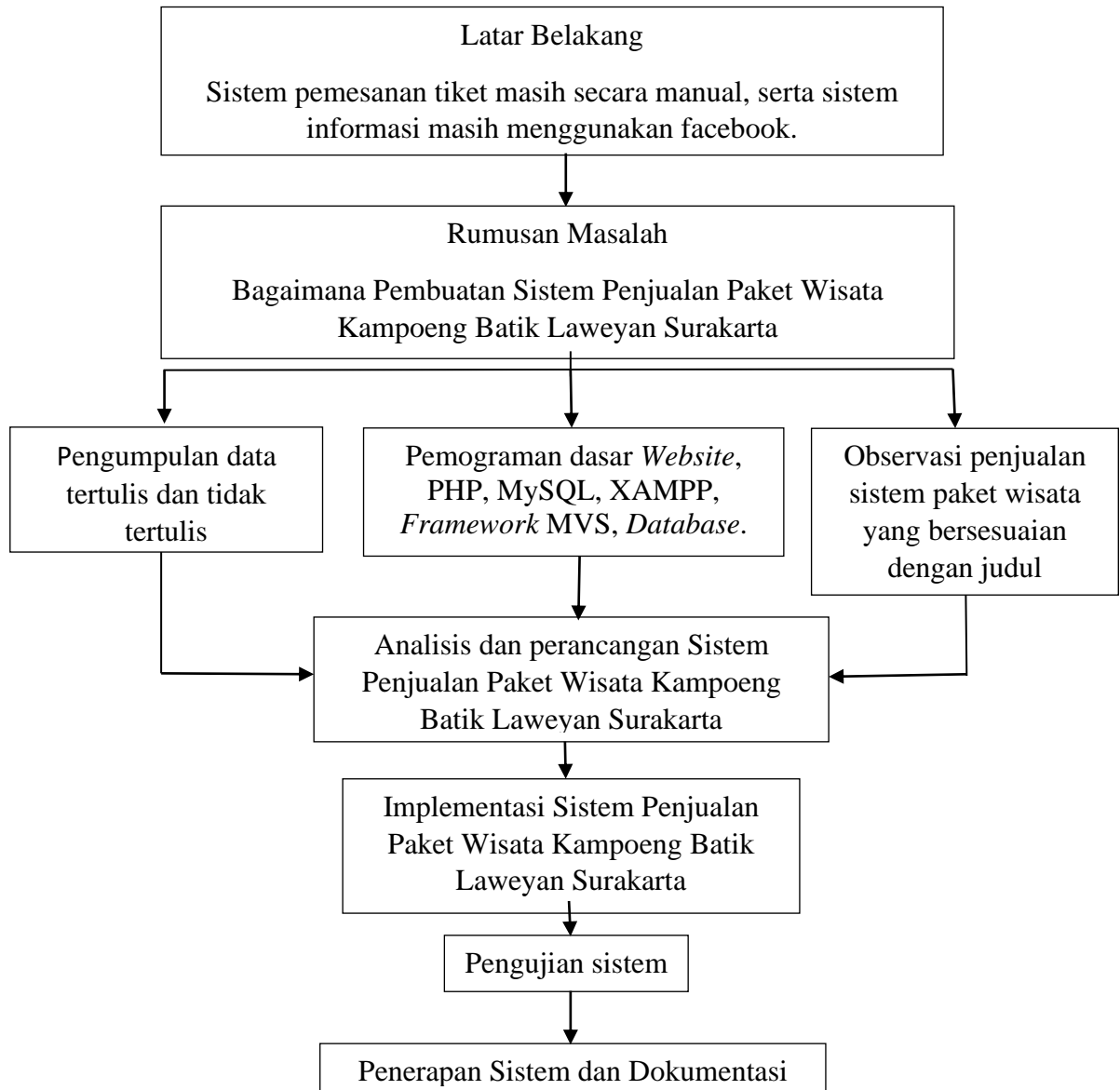
Penelitian yang dilakukan oleh Dewi Kumalasari, Dwi Retnoningsih, dan Dahlan Susilo (2014) yang berjudul “Aplikasi Penjualan Tiket Di Travel Jaya Sakti Surakarta” akan membuat aplikasi yang diharapkan dapat membantu fungsi sistem pelayanan tiket khususnya pada Travel Jaya Sakti untuk mengatasi berbagai persoalan yang timbul akibat penggunaan sistem pencatatan manual.

Penelitian yang dilakukan Abdulghani (2017) yang berjudul “Pembuatan Sistem Informasi Tour and Travel Berbasis *Website*” membuat sistem informasi tour and travel berbasis *website*. Sistem informasi tersebut diharapkan dapat mempermudah bagian marketing Marissa Holiday dalam mempromosikan, menyampaikan informasi tour dan travel dan transaksi pemesanan tiket ataupun transaksi *booking* tempat wisata. Salah satu fitur yang nantinya terdapat dalam sistem informasi ini di antaranya seperti informasi harga travel, tour, promosi pemesanan tiket tour dan travel secara *online* serta informasi pelayanan yang disediakan oleh pihak Marissa Holiday.

Penelitian yang dilakukan oleh Effendri (2016) yang berjudul “Sistem Informasi Pemesanan Paket Wisata Berbasis *Website* Di Citra Madinah Tour and Travel” akan membuat *website* yang dapat mempermudah para wisatawan domestik membeli tiket secara *online* serta sebagai media alternatif untuk meningkatkan jumlah penjualan tiket dengan biaya yang relatif lebih efisien yaitu dengan memanfaatkan sarana internet.

2.2 Kerangka Berfikir

Berikut ini adalah tahapan kerangka berfikir dalam membuat sistem penjualan paket wisata Kampoeng Batik Laweyan Surakarta.



Gambar 2.1 Kerangka Berfikir

Kerangka pemikiran pada Gambar 2.1 menjelaskan alur proses pembuatan sistem Penjualan Paket Wisata Kampoeng Batik Laweyan Surakarta:

1. Latar Belakang

Sistem pemesanan tiket masih secara manual, serta sistem informasi masih menggunakan facebook.

2. Rumusan Masalah
Bagaimana Pembuatan Sistem Penjualan Paket Wisata Kampoeng Batik Laweyan Surakarta?
3. Pengumpulan data tertulis dan tidak tertulis.
Pengumpulan data dilakukan baik dengan tanya jawab (*interview*) dengan pengelola wisata (Alpha Febela Priyatmono), observasi, maupun studi literatur di Kampoeng Batik Laweyan, Surakarta.
4. Pemograman Dasar
Tahap untuk mempelajari dasar-dasar *Website, Php* , *MySQL*, *XAMPP*, *Framework MVS*, dan *Database*. agar lebih menguasai program-program yang akan digunakan untuk membangun sistem.
5. Observasi Sistem Penjualan Paket Wisata Kampoeng Batik Laweyan Surakarta
Merupakan tahap pengamatan sampel-sampel *website* yang telah ada, jurnal, buku, maupun karya ilmiah untuk kajian yang dapat dijadikan referensi untuk pembangunan sistem.
6. Analisis dan perancangan Sistem Penjualan Paket Wisata Kampoeng Batik Laweyan Surakarta menggunakan metode berbasis objek.
7. Implementasi pembuatan sistem penjualan paket wisata Kampoeng Batik Laweyan.
Implementasi sistem adalah langkah-langkah atau prosedur-prosedur yang dilakukan dalam menyelesaikan desain aplikasi yang telah disetujui, untuk menguji dan memulai sistem baru atau sistem yang diperbaiki. Lingkungan implementasi aplikasi ini meliputi kebutuhan perangkat lunak, perangkat keras, *form* program yang sesuai, *script* yang digunakan, pemrograman, pengujian program dan pengujian aplikasi yang telah dirancang sesuai dengan kebutuhan dalam pembuatan sistem penjualan paket wisata Kampoeng Batik Laweyan.
8. Pengujian Sistem
Pengujian sistem adalah prosedur yang dilakukan untuk menyelesaikan rancangan sistem yang telah disetujui, menguji sistem, menginstal serta

memulai penggunaan sistem baru atau sistem yang telah diperbaiki. Dalam pengujian sistem ini menggunakan metode *webqual*.

9. Dokumentasi disini berarti pembuatan laporan mengenai apa yang telah diperoleh pada saat membuat sistem penjualan paket wisata Kampoeng Batik Laweyan.

2.3 Teori Pendukung

2.3.1 Website

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Awalnya suatu *web* hanya dapat menyajikan informasi, saat ini suatu *web* telah dapat berinteraksi dengan pengguna melalui pengisian form validasi *input* atau transaksi *online* (Mayadewi, dkk : 2016).

2.3.2 Pariwisata

Pariwisata adalah keseluruhan rangkaian kegiatan yang berhubungan dengan gerakan manusia yang melakukan perjalanan atau persinggahan sementara dari tempat tinggalnya, ke suatu atau beberapa tempat tujuan di luar lingkungan tempat tinggal yang didorong oleh beberapa keperluan tanpa bermaksud mencari nafkah (Zakaria, 2014).

2.3.3 Internet

Menurut Ahmadi dan Hermawan (2013), internet adalah komunikasi global yang menghubungkan seluruh komputer di dunia meskipun berbeda sistem operasi dan mesin.

Sedangkan menurut Sibero (2013), internet (*interconnected network*) adalah jaringan komputer yang menghubungkan antar jaringan secara global, internet dapat juga disebut jaringan dalam suatu jaringan yang luas.

Berdasarkan kesimpulan di atas, internet adalah sistem jaringan komunikasi secara global yang menghubungkan seluruh komputer di dunia dengan waktu dan wilayah tak terbatas sehingga semua orang dapat selalu terhubung di waktu siang maupun malam dengan *platform* apapun.

2.3.4 Word Wide Web (WWW)

Menurut Kadir (2003) *World wide web* atau yang biasa disebut dengan *web* merupakan salah satu sumber daya internet yang berkembang pesat. Internet sangat berkaitan satu sama lain. Internet adalah suatu jaringan computer global, sedangkan WWW bukan sekedar jaringan tetapi memungkinkan suatu teks pendek menjadi acuan untuk membuka dokumen yang lain.

2.3.5 MYSQL

Rudianto (2011), MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data 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. Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja mengoptimalkan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai server basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak server basisdata kompetitor lainnya.

2.3.6 XAMPP

Menurut Riyanto (2010), XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. XAMPP mengombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket. Memanfaatkan XAMPP sebagai *database* karena XAMPP menyediakan aplikasi *database* MySQL dengan *interface* lebih mudah dalam pengoperasiannya, *tool-tool* yang disediakan cukup lengkap dan memenuhi kebutuhan perancangan *database* selain itu XAMPP aplikasi gratis.

2.3.7 Framework MVC (Model View Control)

Menurut penelitian Maulana (2013), *Framework* adalah kerangka kerja yang digunakan *web programmer* dalam menuliskan kode program supaya lebih terstruktur. *MVC (Model View Control)* adalah sebuah konsep pemrograman yang memisahkan pemrograman *logic* aplikasi dengan presentasinya. Terdapat 3 jenis komponen yang membangun suatu *MVC pattern* dalam suatu aplikasi yaitu :

1. Modul

Modul merupakan struktur data. Secara spesifik *Class* model akan mengandung fungsi kode yang akan membantu dalam segala proses yang berhubungan dengan *database* seperti memasukkan, meng-*edit*, mendapatkan dan menghapus sebuah *database*.

2. View

View merupakan informasi yang disampaikan ke pengguna. Yang mengandung keseluruhan detail dari implementasi *user interface* dengan melibatkan komponen grafis yang menyediakan representasi proses internal aplikasi dan menuntun alur interaksi *user* terhadap aplikasi.

3. Controller

Controller merupakan sebuah perantara antara Model dan View dan semua sumber yang dibutuhkan untuk memproses permintaan HTTP. Bertanggungjawab akan penampungan event yang dibuat oleh *user* dari *view* dan melakukan *update* terhadap komponen model menggunakan data dari *user*.

2.3.8 PHP

Menurut Maimunah (2017), *PHP* singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman web *server-side* yang bersifat *open source*. *PHP* merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embeded scripting*). *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru *uptodate*. Semua *script* dieksekusi pada *server* dimana *script* tersebut dijalankan.

2.3.9 Database

Menurut Maulana (2013), basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis dapat diartikan sebagai markas atau gudang, tempat berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

2.3.10 Analisis Berorientasi Objek

Menurut Gushelmi dan Kamda (2012), Analisis berorientasi objek (OOA-*Object Oriented Analysis*) adalah tahapan perangkat lunak dengan menentukan spesifikasi sistem, sering orang menyebutnya sebagai SRS (*System Requirement Specification*) dan mengidentifikasi kelas-kelas serta hubungannya satu terhadap yang lain. Agar dapat memahami spesifikasi sistem kita perlu mengidentifikasi para pengguna atau yang sering disebut sebagai aktor-aktor. Jacobson (1995), memperkenalkan konsep *use case* sebagai skenario untuk menjelaskan interaksi

pengguna dengan sistem. Model *use case* menggambarkan pandangan pengguna atau kebutuhan pengguna.

2.3.11 Perancangan Berorientasi Objek

Menurut Gushelmi dan Kamda (2012), 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. Aktivitas dan fokus dari analisis berorientasi objek dan perancangan berbasis objek saling bekerja sama, saling melengkapi.

2.3.12 Object Oriented Programming (OOP)

Menurut Fridayanthie dan Charter (2016), pemrograman berorientasi objek sekarang ini merupakan teknik pemrograman yang paling populer dan banyak digunakan oleh programmer untuk menggantikan teknik pemrograman berbasis prosedur. “OOP (*Object Oriented Programming*) adalah sebuah pendekatan untuk pengembangan suatu *software*, dimana dalam struktur *software* tersebut didasarkan kepada interaksi objek dalam penyelesaian suatu proses atau tugas”. “Pemrograman berorientasi objek atau *Object Oriented Programming* (OOP) adalah suatu cara baru dalam berfikir serta berlogika untuk menghadapi masalah-masalah yang akan dicoba batasi dengan bantuan komputer”.

2.3.13 Unified Modeling Language (UML)

Menurut Rossa dan Shalahuddin (2013), *Unified Modeling Language* adalah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek. UML hanya berfungsi untuk melakukan pemodelan. Jadi, penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

2.3.14 Sejarah UML

Menurut Akil (2016), ada beberapa usaha untuk menyatukan metode-metode berorientasi objek yang bermunculan. Contohnya Fusion oleh Coleman dan kawan-kawan (Coleman-94), yang memasukan konsep dari OMT (*Object Modeling Technique*) (Rumbaugh-91), Booch (Booch-91), dan CRC (Wirfs-Brock-90). Karena hal ini tidak melibatkan penulis asli, maka dianggap sebagai metode baru dan bukan menggantikan berbagai metode-metode yang sudah ada. Usaha penggabungan yang sukses pertama kali dan mengganti metode yang ada adalah ketika Rumbaugh bergabung dengan Booch pada perusahaan Rational Software tahun 1994. Mereka mulai mengkombinasikan konsep OMT dan metode Booch, yang menghasilkan proposal pertama tahun 1995. Pada waktu itu, Jacobson juga bergabung di Rational dan mulai berkerjasama dengan Booch dan Rumbaugh. Kerjasama mereka disebut *Unified Modeling Language* (UML). Mereka merevisi metode masing-masing untuk menghasilkan suatu metode lengkap yang harmonis. Pada tahun 1996 *Object Management Group* mengeluarkan permintaan untuk proposal-proposal untuk pendekatan standar pemodelan berorientasi objek. *Unified Modeling Language* diadopsi oleh anggota OMG sebagai standar pada November 1997, OMG bertanggung jawab untuk pengembangan lebih lanjut dari standar UML.

Menurut Havaluddin (2011), ada 4 (empat) prinsip dasar dari pemrograman berorientasi obyek yang menjadi dasar kemunculan UML, yaitu abstraksi, enkapsulasi, modularitas dan hirarki. Berikut dijelaskan satu persatu secara singkat:

1. Abstraksi memfokuskan perhatian pada karakteristik obyek yang paling penting dan paling dominan yang bisa digunakan untuk membedakan obyek tersebut dari obyek lainnya.
2. Enkapsulasi menyembunyikan banyak hal yang terdapat dalam obyek yang tidak perlu diketahui oleh obyek lain. Dalam praktek pemrograman, enkapsulasi diwujudkan dengan membuat suatu kelas *interface* yang akan

dipanggil oleh obyek lain, sementara di dalam obyek yang dipanggil terdapat kelas lain yang mengimplementasikan apa yang terdapat dalam kelas *interface*.

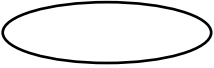
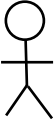

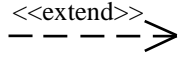
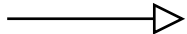
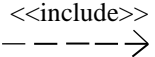
3. Modularitas membagi sistem yang rumit menjadi bagian-bagian yang lebih kecil yang bisa mempermudah *developer* memahami dan mengelola obyek tersebut.
4. Hirarki berhubungan dengan abstraksi dan modularitas, yaitu pembagian berdasarkan urutan dan pengelompokkan tertentu. Misalnya untuk menentukan obyek mana yang berada pada kelompok yang sama, obyek mana yang merupakan komponen dari obyek yang memiliki hirarki lebih tinggi. Semakin rendah hirarki obyek berarti semakin jauh abstraksi.

Dalam membangun perancangan sistem dengan alat bantu perancangan *Unified Modeling Language* (UML) ada beberapa tahapan yang akan dilakukan, yaitu sebagai berikut :

2.3.15 Use Case Diagram

Menurut Yusmiarti (2016), *Use Case Diagram* 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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. *Use-case diagram* dapat digunakan selama proses analisis untuk menangkap *requirement system* dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use-case diagram* berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. *Usecase diagram* juga memiliki komponen seperti. Aktor sistem lain yang berinteraksi dengan sistem yang akan dibuat, *usecase* sistem sebagai unit bertukar pesan antar unit maupun aktor, dan relasi sistem baik aktor maupun antara *usecase* dengan *usecase* dan aktor. Simbol-simbol yang digunakan *Use Case Diagram* disajikan pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case*




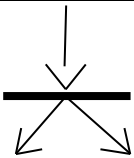
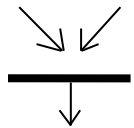
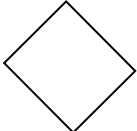
NO	SIMBOL	NAMA	KETERANGAN
1		<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama <i>use case</i> .
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan 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.

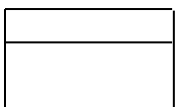
2.3.16 Activity Diagram

Menurut Yusmiarti (2016), *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *work flow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status.

Menguntungkan untuk membuat *activity diagram* pada awal pemodelan proses untuk membantu memahami keseluruhan proses. Simbol-simbol yang digunakan *Activiy Diagram* disajikan pada Tabel 2.2.

Tabel 2.2 Simbol *Activity Diagram*

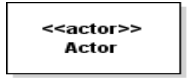


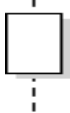
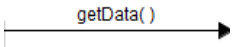
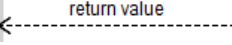
No	Gambar	Nama	Keterangan
1.		<i>Start Point</i>	<i>Start Point</i> diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
2.		<i>End Point</i>	<i>End Point</i> merupakan akhir aktivitas.
3.		<i>Activities</i>	<i>Activities</i> menggambarkan suatu proses/ kegiatan bisnis.
4.		<i>Fork</i>	<i>Fork/</i> percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5.		<i>Join</i>	<i>Join /</i> penggabungan atau rake digunakan untuk menunjukkan adanya dekomposisi
6.		<i>Decision Point</i>	<i>Decision Point</i> menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i> .

7.		<i>Swimlane</i>	<i>Swimlane</i> merupakan pembagian <i>activity diagram</i> untuk menunjukkan siapa dan melakukan apa.
----	---	-----------------	--

2.3.17 Sequence Diagram

Sukanto dan Shalahudin (2014), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. *Sequence diagram* mempunyai simbol-simbol yang ditunjukkan pada Tabel 2.3.



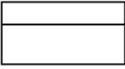


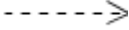

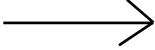
Tabel 2.3 Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	<i>Actor</i> merupakan eksternal sistem berupa orang atau sistem lain yang mendapatkan manfaat dari <i>use-case</i> .
2.		<i>Object</i>	<i>Object</i> merupakan sesuatu yang ikut dalam rangkaian kegiatan dengan mengirim atau menerima pesan.
3.		<i>Lifeline</i>	<i>Lifeline</i> menunjukan waktu hidup suatu objek dalam rangkaian kegiatan.
4.		<i>Executio Occurrence</i>	<i>Execution Occurrence</i> menandakan ketika suatu objek mengirim atau menerima pesan.
5.		<i>Request Message</i>	<i>Request Message</i> Merupakan permintaan informasi dari satu objek ke objek lain.
6.		<i>Return Message</i>	<i>Return Message</i> merupakan permintaan informasi yang dikirim oleh suatu objek.

2.3.18 Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Simbol-simbol yang digunakan *Class Diagram* disajikan pada tabel 2.4.

Tabel 2.4 Simbol *Class Diagram*.

NO	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
8.		<i>Directed Association</i>	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Menurut Yusmiarti (2016), *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), relasi *Assosiation*, atribut (*attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*. Simbol-simbol yang digunakan *Multiplicity* pada *Class Diagram* disajikan pada Tabel 2.5.

Tabel 2.5 Simbol *Multiplicity* pada *Class Diagram*

No	<i>Multiplicity</i>	Penjelasan
1.	1	Satu dan hanya satu.
2.	0..*	Boleh tidak ada atau 1 atau lebih.
3.	1..*	1 atau lebih.
4.	0..1	Boleh tidak ada, maksimal 1.
5.	n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4.

2.3.19 *Component Diagram*

Menurut Qossam dan Izudin (2015), diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan *source code* program perangkat lunak dan komponen *executable* yang dilepas ke *user*. Di dalam komponen diagram, terdapat 3 tipe model, yaitu sebagai berikut :

1. *Deployment Component*

Merupakan tipe model yang menjadi basis dari *executable system*. Contohnya DLL (*Dynamic Library Link*), Active X Control, Java Bean dan lain sebagainya.

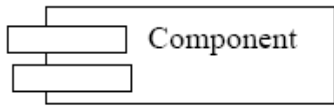
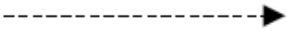
2. *Work Product Component*

Merupakan salah satu tipe model yang berisi atas file-file yang dibutuhkan untuk pembuatan *deployment component*. Contohnya file data, file *source code*, dan file penting lainnya.

3. *Execution Component*

Merupakan tipe terakhir dari komponen diagram. Dimana model ini dibuat sebagai hasil dari sistem yang akan dijalankan. Simbol-simbol yang digunakan *Component Diagram* disajikan pada Tabel 2.6.

Tabel 2.6 Simbol *Component Diagram*


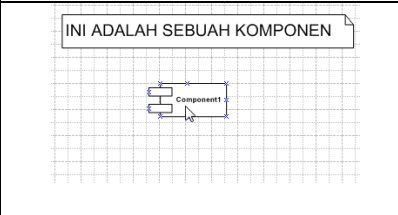
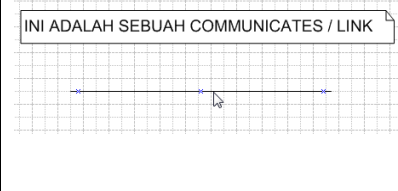
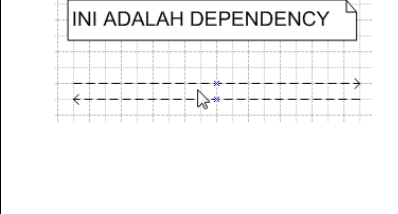
NO	Gambar	Nama	Keterangan
1		Component	Sebuah komponen melambangkan sebuah entitas <i>software</i> dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel di sebelah kirinya.
2		Depedency	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.

2.3.20 *Deployment Diagram*

Menurut Andhika (2015), *Deployment diagram* adalah salah satu jenis alat atau bahasa (*UML*) yang digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak

berbasis *Object Oriented* yang akan dibangun. Simbol-simbol yang digunakan *Deployment Diagram* disajikan pada Tabel 2.7.

Tabel 2.7 Simbol *Deployment Diagram*

NO	Gambar	Nama	Keterangan
1.		Node	Node digunakan untuk menggambarkan infrastruktur apa saja yang terdapat pada sistem. Biasanya node digambarkan sebagai <i>server</i> , <i>pc</i> , dan lain-lain.
2		Component	Komponen digunakan untuk menggambarkan elemen-elemen apa saja yang terdapat pada suatu node.
3		Link	Communicates digunakan untuk menghubungkan antar node yang saling berinteraksi.
4		Dependency	Dependency digunakan untuk menggambarkan hubungan ketergantungan antar node atau komponen yang saling ketergantungan.

2.3.21 Metode *Webqual*

WebQual merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna. *WebQual* sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa interaksi dalam penyusunan dimensi dan butir pertanyaan (Pratama, 2015). *Webqual* merupakan alat untuk

mengukur *usability* (kegunaan), *information quality* (kualitas informasi), *interactionquality* (kualitas interaksi) dan *customer satisfaction* (kepuasan pengguna). Instrumen kualitas penggunaan (*Usability Quality*) disajikan pada Table 2.8.

Tabel 2.8 Kualitas Penggunaan (*Usability Quality*)

KUALITAS	DESKRIPSI
<i>Usability Quality</i> (Kualitas Penggunaan)	
1	<i>I find the site easy to learn to operate</i> /website mudah untuk dioperasikan
2	<i>My interaction with the site is clear and understandable</i> /interaksi dengan website sangat mudah dimengerti dan tidak membingungkan
3	<i>I find the site easy to navigate</i> /mudah menemukan link-link yang diinginkan
4	<i>I find the site easy to use</i> /website mudah digunakan
5	<i>The site has an attractive appearance</i> /website memiliki tampilan yang menarik
6	<i>The design is appropriate to the type of site</i> /desain website sesuai dengan tipe website
7	<i>The site conveys a sense of competency</i> /website dapat menyampaikan suatu rasa dari sebuah kemampuan → menunjukkan kompetensi
8	<i>The site creates a positive experience for me</i> /website dapat memberikan pengaruh/pengalaman positif bagi saya

Instrumen kualitas informasi (*Information Quality*) disajikan pada Tabel 2.9.

Tabel 2.9 Kualitas Informasi (*Information Quality*)

<i>Information Quality</i>(Kualitas Informasi)	
1	<i>Provides accurate information</i> /informasi yang disajikan akurat
2	<i>Provides believable information</i> /informasi yang disajikan dapat dipercaya.
3	<i>Provides timely information</i> /informasi yang disajikan tepat waktu dengan apa yang saya butuhkan.
4	<i>Provides relevant information</i> /Informasi yang disajikan relevan dengan apa yang saya inginkan.
5	<i>Provides easy to understand information</i> /informasi yang disediakan mudah untuk dipahami.

6	<i>Provides information at the right level of detail/informasi yang disajikan sangat detail.</i>
7	<i>Present the information in an appropriate format/informasi yang disajikan dalam format yang sesuai.</i>

Instrumen kualitas interaksi (*Interaction Quality*) disajikan pada Tabel 2.10.

Tabel 2.10. Kualitas Interaksi (*Interaction Quality*)

<i>Interaction Quality</i>(Kualitas Interaksi)	
1	<i>Has a good reputation/website memiliki reputasi yang bagus.</i>
2	<i>It feels safe to complete transactions/saya merasa aman jika saya melakukan transaksi/interaksi dengan website.</i>
3	<i>My personal information feels secure/website sangat menjaga informasi pribadi saya.</i>
4	<i>Creates a sense of personalization/sense personalisasi sangat diperhatikan dalam website</i>
5	<i>Conveys a sense of community/Sense masyarakat (komunitas) diperhatikan oleh website</i>
6	<i>Makes it easy to communicate with the organization/website memudahkan saya berkomunikasi dengan organisasi</i>
7	<i>I feel confident that goods/services will be delivered as promised/saya merasa yakin dengan layanan atau informasi yang disediakan, karena sesuai dengan yang dijanjikan</i>

Instrumen kepuasan pelanggan (*Customer Satisfaction*) disajikan pada Tabel 2.11.

Tabel 2.11. *Customer Satisfaction* (Kepuasan Pelanggan)

<i>Customer Satisfaction</i> (Kepuasan Pelanggan)	
1	<i>I like the look of the website www.kampoengbatik.com/saya menyukai tampilan website www.kampoengbatik.com</i>
2	<i>I like the service available on the website/saya menyukai pelayanan yang tersedia pada website</i>
3	<i>I like interacting with this website/saya senang berinteraksi dengan website ini</i>
4	<i>I didn't wait long when I entered the website/saya tidak menunggu lama ketika masuk ke website</i>

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

a) Uji Instrumen

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

b) 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 mampu mengungkap data tentang karakteristik gejala yang diteliti secara tepat (Prakosad, 2017).

c) 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. Dengan demikian reliabel adalah 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 (Prakosad, 2017).

d) Uji Asumsi Klasik

Menurut Prakosad (2017), 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.

e) Uji Normalitas

Pengujian ini bertujuan untuk mengetahui apakah data terdistribusi secara normal/tidak. Pengujian normalitas normalitas dapat dilihat dari titik-titik yang menyebar mengikuti garis diagonal pada *Normal PP-Plot Regression*, di mana jika titik-titik tersebut mengikuti garis diagonal maka dikatakan bahwa data yang digunakan adalah berdistribusi normal (Prakosad, 2017).

f) Uji Multikolinearitas

Pengujian ini bertujuan untuk menguji apakah dalam model regresi ditemukan adanya korelasi variabel independen. Model regresi yang baik seharusnya tidak terjadi korelasi variabel independen. Pendeteksiannya dilakukan dengan menggunakan *tolerance value* dan *Variance Inflation Factor* (VIF). Jika nilai *tolerance value* > 0.10 dan $VIF < 10$, maka tidak terjadi multikolinearitas, sebaliknya jika nilai *tolerance value* < 0.10 dan $VIF > 10$ maka terjadi multikolinearitas (Prakosad, 2017).

g) Uji Heterokedastisitas

Pengujian ini bertujuan untuk menguji apakah dalam model regresi terjadi ketidaksamaan *variance* dari *residual* satu pengamatan ke pengamatan yang lain. Jika *variance* dari residual satu pengamatan ke pengamatan yang lain tetap maka disebut Homokedastisitas dan jika berbeda maka disebut Heterokedastisitas. Model yang baik adalah yang homokedastisitas atau tidak terjadi heterokedastisitas. Pendeteksiannya dilakukan dengan melihat grafik plot antara nilai prediksi variabel terikat yaitu ZPRED (Z predictor) dengan residualnya SRESID (*standardized residual*). Deteksi terjadinya heterokedastisitas dapat

dilakukan dengan melihat penyebaran titik-titik pada sumbu Y. Jika data tersebut baik di atas maupun di bawah sumbu Y, maka dapat disimpulkan bahwa tidak terjadi gejala heterokedastisitas (Prakosad, 2017).

h) Uji Autokorelasi

Uji Autokorelasi merupakan pengujian yang dilakukan untuk menguji ada tidaknya pengaruh antara variabel pengganggu dalam masing-masing variabel bebas. Uji autokorelasi bisa menggunakan tes Durbin Watson dengan ketentuan sebagai berikut :

$dW < dL$: berarti ada autokorelasi positif (+).

$dL < dW < 4-dL$: tidak dapat disimpulkan.

$dU < dW < 4-dU$: berarti tidak terjadi autokorelasi.

$4-dU < dW < 4-dL$: tidak dapat disimpulkan.

$dW > 4-dL$: berarti ada autokorelasi negatif (-).

Menurut Prakosad (2017) menyatakan bahwa uji autokorelasi merupakan pengujian asumsi dalam regresi dimana variabel dependen tidak berkorelasi dengan dirinya sendiri. Maksud korelasi dengan diri sendiri adalah bahwa nilai dari variabel dependen tidak berhubungan dengan nilai variabel itu sendiri, baik nilai variabel sebelumnya atau nilai periode sesudahnya. Dasar pengambilan keputusan adalah sebagai berikut :

Angka D-W di bawah -2 : berarti ada autokorelasi.

positif Angka D-W diantara -2 sampai +2 : berarti tidak ada.

autokorelasi Angka D-W di atas +2 : berarti ada autokorelasi negatif.

i) Analisis Regresi Linier Berganda

Menurut Prakosad (2017), mengemukakan analisis regresi linier berganda digunakan untuk melakukan prediksi, bagaimana perubahan nilai *variable*

dependen bila nilai variabel independen dinaikan atau diturunkan nilainya. Analisis ini digunakan dengan melibatkan dua atau lebih variabel bebas antara variabel dependen (Y) dan variabel independen (X1, X2, dan X3), cara ini digunakan untuk mengetahui kuatnya hubungan antara beberapa variabel bebas secara serentak terhadap variabel terkait dan dinyatakan dengan rumus.

Teknik analisis data yang digunakan dalam penelitian ini adalah teknik analisis kuantitatif dengan menggunakan model Regresi Linier Berganda (*Multiple Linier Regresion Method*) yang akan diolah dengan program *SPSS for Windows* versi 20. Metode ini digunakan untuk mengetahui pengaruh variabel bebas terhadap variabel tergantung. Rumuskan analisis regresi linier berganda sebagai berikut :

$$Y' = a + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \quad \text{persamaan (2.1)}$$

Keterangan :

Y' = Kepuasan Pengguna, yang merupakan variabel dependen.

a = Konstanta, merupakan nilai terkait yang dalam hal ini adalah Y' pada saat variabel independennya adalah 0 ($X_1, X_2, X_3=0$).

β_1 = Koefisien regresi berganda variabel independen X_1 terhadap variabel Y' , bila variabel X_2 dan X_3 dianggap konstan.

β_2 = Koefisien regresi berganda variabel independen X_2 terhadap variabel Y' , bila variabel X_1 dan X_3 dianggap konstan.

β_3 = Koefisien regresi berganda variabel independen X_3 terhadap variabel Y' , bila variabel X_1 dan X_2 dianggap konstan.

X_1 = Kualitas Penggunaan yang merupakan variabel independen ke-1.

X_2 = Kualitas Informasi yang merupakan variabel independen ke-2.

X_3 = Kualitas Interaksi yang merupakan variabel independen ke-3.

j) Pengujian Hipotesis

Menurut Prakosad (2017), uji hipotesis dilakukan untuk menganalisis dan menarik kesimpulan terhadap permasalahan yang diteliti. Pengujian ini dimaksudkan untuk mengetahui apakah terdapat pengaruh variabel bebas terhadap

variabel tidak bebas. Uji hipotesis yang digunakan adalah uji F dan uji t. Berdasarkan program SPSS, jika probabilitas hasil (*p value*) dari nilai $F < 5\%$ maka hipotesis yang diajukan diterima; dan jika probabilitas hasil (*p value*) dari nilai $F > 5\%$ maka hipotesis yang diajukan ditolak. Demikian juga apabila probabilitas hasil (*p value*) dari nilai $t < 5\%$ maka hipotesis yang diajukan diterima; dan jika probabilitas hasil (*p value*) dari nilai $t > 5\%$ maka hipotesis yang diajukan ditolak.

k) Uji F (Uji Simultan)

Menurut Prakosad (2017), uji F dimaksud untuk menguji hipotesis apakah variabel-variabel bebas secara bersama-sama mempunyai pengaruh terhadap variabel tergantung. Tahap-tahap uji F adalah :

1. Merumuskan hipotesis

$$H_0 : b_1 = b_2 = b_3 = 0$$

Artinya variabel bebas secara bersama-sama tidak mempunyai pengaruh terhadap variabel tergantung.

$$H_a : b_1 = b_2 = b_3 \neq 0$$

Artinya variabel bebas secara bersama-sama mempunyai pengaruh terhadap variabel tergantung.

2. Menentukan tingkat signifikan

Tingkat signifikan yang diharapkan (α) adalah 5% dengan *degree of freedom* (df) = (k-1) dan (n-k) guna menetapkan nilai F tabel, di mana n= jumlah responden dan k = jumlah variabel bebas

3. Menentukan nilai F hitung, dengan rumus :

$$R^2 / (k - 1)$$

$$F_{hit} = \frac{R^2}{(1 - R^2) / (n - k)}$$

4. Membandingkan nilai F hitung dengan F tabel, bila:

$F_{hit} < F_{tab}$ = berarti H_0 diterima dan H_1 ditolak

$F_{hit} > F_{tab}$ = berarti H_1 diterima dan H_0 ditolak.

l) Uji T (Uji Parsial)

Menurut Prakosad (2017), uji t dimaksudkan untuk menguji hipotesis apakah variabel bebas secara parsial mempunyai pengaruh terhadap variabel tergantung.

Tahap-tahap uji t adalah :

Merumuskan hipotesis

$H_0: b_1 = 0$

Artinya variabel bebas secara parsial tidak mempunyai pengaruh terhadap variabel tergantung.

$H_a: b_1 \neq 0$

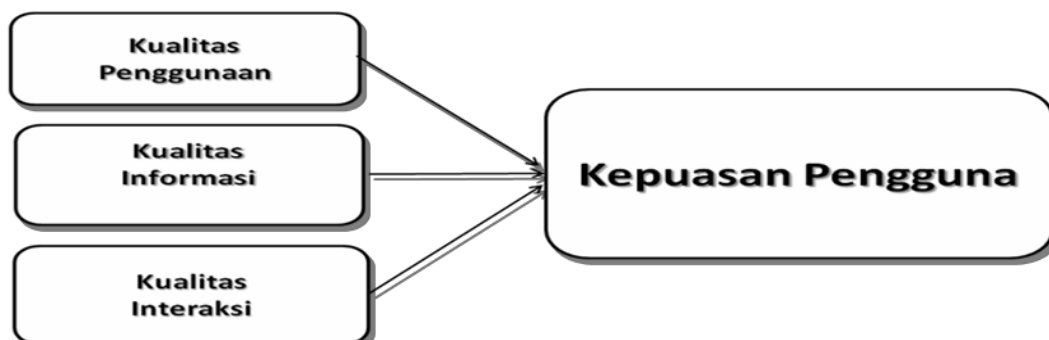
Artinya variabel bebas secara parsial mempunyai pengaruh terhadap variabel tergantung.

m) Uji Koefisien Determinasi Berganda

Koefisien determinasi merupakan koefisien yang nilainya dimaksudkan untuk mengetahui seberapa besar variasi perubahan dalam satu variabel independen. Koefisien korelasi digunakan untuk menentukan koefisien determinasi. Dalam konteks ini, koefisien determinasi merupakan kuadrat dari koefisien korelasi yang dinotasikan dengan R^2 Prakosad (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.

n) Kerangka Konseptual

Kerangka Konseptual disajikan pada Gambar 2.2.



Gambar 2.2 Kerangka Konseptual

Berdasarkan kerangka konseptual tersebut maka akan dilakukan analisis data untuk mengetahui pengaruh variabel kualitas penggunaan, kualitas informasi, dan kualitas interaksi terhadap kepuasan pengguna.