

BAB II

LANDASAN TEORI

2.1 Kajian Pustaka

a. Aplikasi Penjualan Berbasis Web (*E-Commerce*) menggunakan Joomla pada Mutiara Fashion.

Penelitian mahasiswa Universitas Widyatama melakukan pengembangan aplikasi penjualan berbasis *web* menggunakan *Content Management System* (CMS) Joomla. Penelitian tersebut bertujuan untuk memberikan kemudahan kepada konsumen dalam mendapatkan informasi maupun dalam membeli produk yang dijual oleh Mutiara *Fashion*, membantu Mutiara *Fashion* dalam mempromosikan dan menjual produk kepada konsumen, dan memberikan kemudahan dalam pengolahan data transaksi serta pembuatan laporan penjualan. Penelitian tersebut adalah penelitian pengembangan atau *Research and Development* (R&D). Instrumen pengumpulan data terdiri dari lembar penilaian kualitas *website* menggunakan metode *blackbox* (Ginting, 2013).

b. Perancangan dan Implementasi Toko *Online* Perlengkapan Bayi dan Anak Berbasis *Web* dan *Mobile*.

Penelitian ini merupakan penelitian mahasiswa Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta pada tahun 2013. Dalam penelitian tersebut dilatarbelakangi kurang berkembangnya toko *Kids Corner* karena belum memiliki *website* toko *online*, sehingga dibuatlah toko *online* perlengkapan bayi dan anak berbasis *web* dengan menggunakan *framework* CakePHP. Sistem tersebut merupakan suatu sistem yang berbasis *web* dan *mobile* android yang dapat mengelola data penjualan dari sebuah toko perlengkapan bayi dan anak *Kids Corner* Yogyakarta. Sistem tersebut telah melalui tahap pengujian dengan melibatkan 12 orang penguji. Dari hasil pengujian didapatkan 100% jawaban ya dan 0% jawaban tidak untuk pengujian fungsionalitas, serta hasil pengujian antarmuka dan pengaksesan didapatkan 43% jawaban sangat setuju, 53% setuju, 3% tidak setuju, dan 0% sangat tidak setuju (Nucifera, 2013).

c. Perancangan Sistem Informasi Penjualan *E-Commerce* Berbasis Web pada Toko Fara Collection.

Penelitian ini merupakan penelitian mahasiswa Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Gici Batam pada tahun 2017. Pada penelitian tersebut, perancangan sistem informasi penjualan *e-commerce* berbasis *web* menggunakan CMS Prestashop. Pengembangan media penjualan *online* ini dilakukan dalam beberapa tahap, yaitu mengumpulkan data, perancangan sistem, pembuatan sistem, pengujian sistem, dan implementasi sistem (Sitanggang, 2017).

Berdasarkan uraian dari tiga penelitian tersebut dapat disimpulkan bahwa terdapat kesamaan tujuan penelitian yaitu membuat toko *online* berbasis *web*, tetapi menggunakan *platform* yang berbeda-beda. Tabel 2.1 menunjukkan perbandingan ketiga penelitian tersebut.

2.2. Kerangka Pemikiran

Kerangka pemikiran dari penelitian tugas akhir ini telah tercantum pada Gambar 2.1. Berdasarkan Gambar 2.1 maka kerangka pemikiran dapat diuraikan sebagai berikut:

1. Latar belakang masalah

Permasalahan yang mendasari perlunya dibangun toko *online* PunPun Shop berbasis *web*.

2. Perumusan masalah

Inti permasalahan dan jalan keluar untuk menyelesaikan permasalahan penelitian.

3. Pengumpulan data tertulis dan tidak tertulis

Pengumpulan data penelitian ini menggunakan metode observasi, dokumentasi, dan wawancara.

4. Penguasaan dasar

Pada penelitian dilakukan percobaan membuat sistem agar lebih menguasai *platform* yang digunakan untuk membangun sistem dan kinerja sistem.

5. Observasi sistem

Pada penelitian dilakukan pengamatan pada sistem yang sebelumnya digunakan agar dapat menjadi referensi dalam membangun aplikasi ini.

6. Analisis dan perancangan sistem

Pada penelitian dilakukan analisis dan perancangan bagaimana sistem nantinya akan dibuat untuk membantu memecahkan permasalahan yang ada.

7. Implementasi sistem

Pada penelitian dilakukan implementasi apa yang sudah dirancang untuk membangun toko *online PunPun Shop* berbasis *web*.

8. Pengujian sistem

Pada penelitian dilakukan uji coba aplikasi apakah masih terjadi kesalahan ataupun kekurangan pada sistem.

9. Analisis kualitas sistem

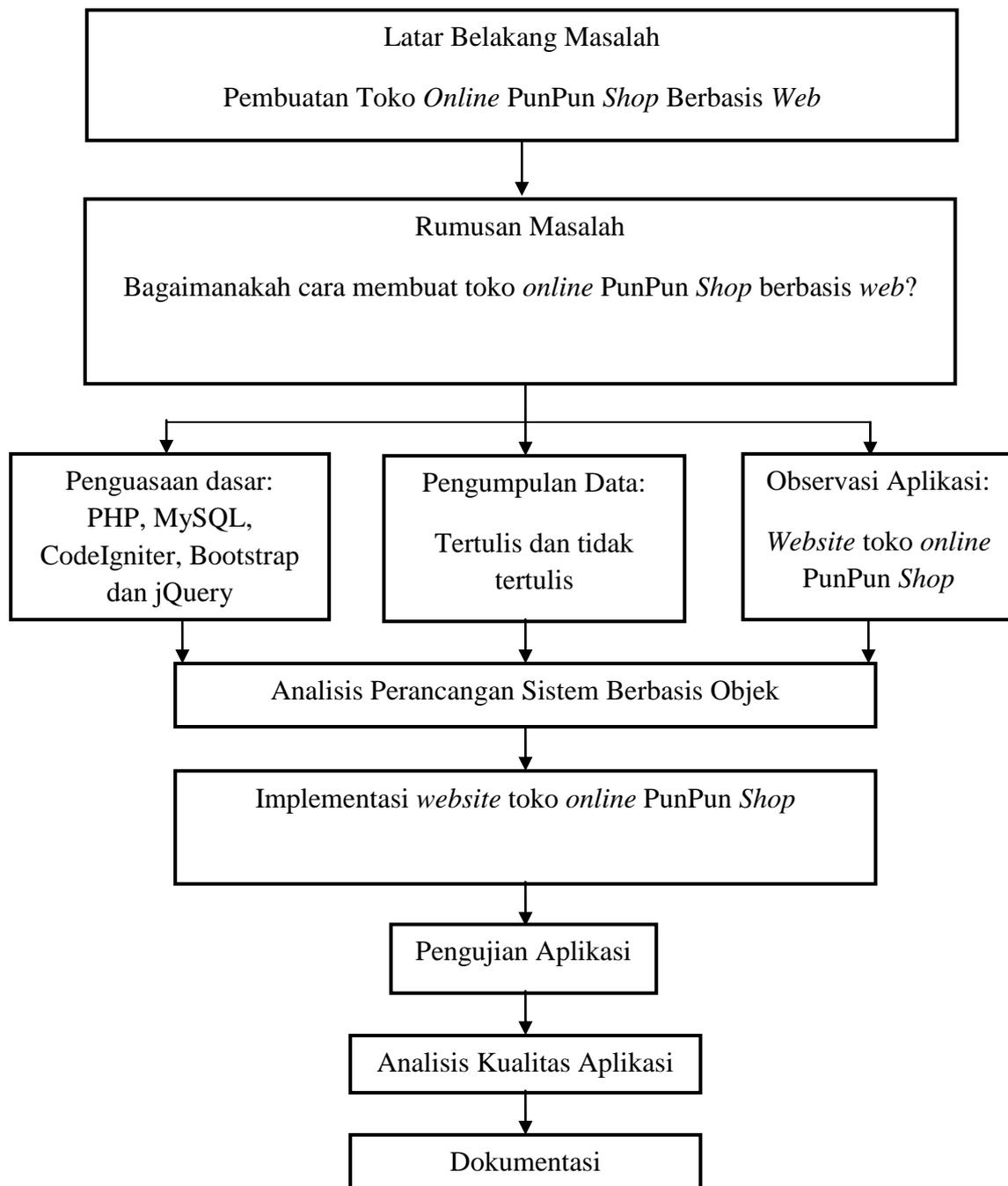
Pada penelitian dilakukan analisis kualitas *website* sesuai standar WebQual.

10. Dokumentasi

Sistem yang sudah diimplementasikan, diuji coba dan dianalisis kemudian diterapkan pada *PunPun Shop* dan setelah itu dibuat dokumentasi dari keseluruhan kegiatan penyusunan Tugas Akhir.

Tabel 2.1. Perbandingan Penelitian

No	Judul	Tujuan	Metode	Hasil
1	Aplikasi Penjualan Berbasis Web (<i>E-Commerce</i>) menggunakan Joomla pada Mutiara Fashion	Memberikan kemudahan dalam mendapatkan informasi maupun dalam membeli produk. Membantu mempromosikan dan menjual produk kepada konsumen. Memberikan kemudahan dalam pengolahan data transaksi serta pembuatan laporan penjualan.	<i>Research and Development</i> (R&D), metode <i>Waterfall</i> , CMS Joomla, <i>Blackbox testing</i> .	Toko <i>online</i> Mutiara Fashion berjalan dengan baik.
2	Perancangan dan Implementasi Toko <i>Online</i> Perlengkapan Bayi dan Anak Berbasis Web dan <i>Mobile</i>	Memudahkan penjualan dan pengelolaan data penjualan.	<i>Research and Development</i> (R&D), metode <i>Waterfall</i> , <i>Framework</i> CakePHP, Kuesioner fungsionalitas, antarmuka dan pengaksesan.	Toko <i>online</i> perlengkapan bayi dan anak berbasis <i>web</i> dan <i>mobile</i>
3	Perancangan Sistem Informasi Penjualan <i>E-Commerce</i> Berbasis Web pada Toko Fara Collection	Memudahkan pengelolaan data transaksi penjualan	<i>Research and Development</i> (R&D), metode <i>Waterfall</i> , CMS Prestashop, <i>Blackbox testing</i>	Toko <i>online</i> Fara Collection dapat berjalan dengan baik.



Gambar 2.1. Diagram Kerangka Pemikiran

2.3 Teori Pendukung

Penyusunan Tugas Akhir memerlukan suatu referensi pendukung yang digunakan sebagai landasan teori agar penelitian dapat berjalan dengan benar dan tidak menyimpang dari kaedah ilmu pengetahuan yang ada. Landasan teori

diperoleh dari berbagai sumber dan literatur yang mempublikasikan pendapat beberapa ilmuwan yang digunakan sebagai pendukung pembahasan masalah dalam penelitian Tugas Akhir. Berikut ini beberapa diantaranya:

2.3.1 Toko Online (E-Commerce)

Sebagian orang berpikir bahwa *e-commerce* adalah *shopping online*, belanja, atau membeli sesuatu di internet. Transaksi *online* hanyalah sebagian kecil dari tujuan *e-commerce*. Sebagian besar dari fungsi *e-commerce* adalah proses hubungan dagang bisnis ke bisnis, untuk memudahkan pembelian atau penjualan antar perusahaan-perusahaan. Proses pembayarannya biasanya menggunakan *online processing credit card*, *money transfer*, atau *cash on delivery* jika alamat konsumen satu kota dengan penjual (Sutisna, 2007).

2.3.2 PunPun Shop

PunPun Shop merupakan toko mainan anak yang beralamat di Jalan Lompo Batang Raya, Mojosoongo, Surakarta. Toko tersebut menjual berbagai mainan anak-anak yaitu *slime*, boneka *squishy* dan biskuit karakter (*cookies*). Hingga sekarang proses jual beli dilakukan melalui tiga cara yaitu melalui sosial media, toko di Mojosoongo dan di *car free day* Jalan Slamet Riyadi Surakarta setiap hari Minggu.

2.3.3 Web Server

Web server adalah *software* yang memberikan layanan data yang mempunyai fungsi untuk menerima permintaan HTTP (*Hypertext Transfer Protocol*) atau HTTPS yang dikirim oleh klien melalui *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman *web* yang umumnya berbentuk HTML (*Hypertext Markup Language*). *Web server* berguna sebagai tempat aplikasi *web* dan sebagai penerima *request* dari *client* (Indra Warman, 2013).

2.3.4 Web Library (Pustaka Web)

Pustaka atau *library* adalah kumpulan program atau fungsi yang telah ada pada *compiler* atau *intepreter* untuk memudahkan pemrogram membuat program dan tidak perlu mengakses langsung sistem komputer untuk memprogram. Pustaka yang disediakan bergantung pada bahasa pemrograman yang digunakan maupun *compiler* atau *intepreter* yang digunakan, untuk mengetahui pustaka apa saja yang ada perlu dilakukan eksplorasi terhadap bahasa pemrograman yang digunakan. Setiap pustaka menyediakan fungsi-fungsi tertentu yang dapat mendukung pemrograman (Indra Warman, 2013).

2.3.5 XAMPP

XAMPP adalah paket program *web* lengkap yang dapat dipakai untuk belajar pemrograman web khususnya PHP dan *MySQL* (Nugroho, 2013). XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Bagian penting dari XAMPP yang biasa digunakan adalah :

- 1) *htdocs* adalah folder tempat meletakkan basis data *MySQL* yang ada seperti berkas PHP, HTML dan *script* lain.
- 2) *phpMyAdmin* merupakan bagian untuk mengelola basis data *MySQL* yang ada di komputer. Untuk membukanya, buka browser lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman *phpMyAdmin*.
- 3) *control panel* yang berfungsi untuk mengelola layanan (*service*) XAMPP seperti menghentikan (*stop*) layanan atau memulai (*start*).

XAMPP adalah singkatan yang masing-masing hurufnya :

- 1) X : Program ini dapat dijalankan di banyak sistem operasi, seperti *Windows*, *Linux*, *Mac OS* dan *Solaris*.
- 2) A : *Apache*, merupakan aplikasi *web server*. Tugas utama *apache* adalah menghasilkan halaman *web* yang benar kepada *user* berdasarkan kode PHP yang dituliskan oleh pembuat halaman *web*.

- 3) M : *MySQL*, merupakan aplikasi *database server*. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah *database*.
- 4) P : PHP, bahasa pemrograman *web*. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat *web* yang bersifat *server-side scripting*.
- 5) P : Perl, bahasa pemrograman.

2.3.6 *MySQL*

MySQL adalah salah satu perangkat lunak sistem manajemen basis data atau *database SQL* atau sering disebut dengan *Database Management System* (DBMS). Berbeda dengan basis data konvensional seperti *.dat*, *.dbf*, *.mdb*, *MySQL* memiliki kelebihan yaitu bersifat multithread, dan multi *user* serta mendukung sistem jaringan. *MySQL* didistribusikan secara gratis dibawah lisensi GNU *General Public License* (GPL), namun ada juga versi komersial bagi kalangan tertentu yang menginginkannya (Virgi dan Hirin, 2011).

Ada beberapa kelebihan yang dimiliki *MySQL* sehingga dapat menarik banyak pengguna. Kelebihan tersebut yaitu :

1. Fleksibilitas

Saat ini, *MySQL* telah dioptimasi untuk duabelas platform seperti *HP-UX*, *Linux*, *Mac OS X*, *Novell Netware*, *OpenBSD*, *Solaris*, *Microsoft Windows* dan lain-lain. *MySQL* juga menyediakan *source code* yang dapat diunduh secara gratis, sehingga pengguna dapat mengkompilasi sendiri sesuai platform yang digunakan. Selain itu, *MySQL* juga dapat dikustomisasi sesuai keinginan penggunanya, misalnya mengganti bahasa yang digunakan pada antarmukanya.

2. Performansi

Sejak rilis pertama, pengembang *MySQL* fokus kepada performa. Hal ini masih tetap dipertahankan hingga sekarang dengan terus meningkatkan fiturnya.

3. Lisensi

MySQL menawarkan berbagai pilihan lisensi kepada penggunanya. Lisensi *open source* yang ditawarkan yaitu lisensi GNU *General Public License* dan *Free/Libre and Open Source Software (FLOSS) License Exception*. Selain itu

ditawarkan juga lisensi komersil berbayar yang memiliki fasilitas dukungan teknis.

2.3.7 PHP

PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka layout *web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, *web* akan sangat mudah di-*maintenance*. PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya *web server*(Saputra, 2011).

PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada *web server apache* dan sebagai *binary* yang dapat berjalan sebagai CGI.

2.3.8 CodeIgniter

CodeIgniter merupakan *framework* PHP yang diklaim memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya. *CodeIgniter* bersifat *open source* dan menggunakan model kasus MVC (*Model View Controller*) yang merupakan model konsep modern *framework* yang digunakan saat ini (Saputra, 2011). Kelebihan *framework CodeIgniter*:

- 1) Mendukung PHP 4 dan PHP5,
- 2) Berukuran kecil dan cepat,
- 3) Menggunakan konsep MVC.

CodeIgniter juga memiliki implementasi *Active Record* yang berfungsi untuk memudahkan dalam penulisan *query* SQL yang kompleks dan membuat kode aplikasi lebih mudah dibaca. Fitur ini juga memungkinkan mengganti *driver* basis data dengan mudah, misalkan menulis program dengan basis data MySQL

maka dapat dengan mudah diganti dengan basis data Oracle tanpa harus menulis ulang deklarasi kode-kode program dalam aplikasi. *CodeIgniter* juga dilengkapidenganjumlah *library* yang sangat berguna dan beberapa perangkat lainnya yang memudahkan dalam membangun aplikasi (Griffiths, 2010).

2.3.9 Bootstrap

Bootstrap merupakan sebuah *framework* CSS (*Cascading Style Sheets*) yang memudahkan pengembang untuk membangun *website* yang menarik dan responsif. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk mengembangkan dan pemeliharaannya. Bootstrap adalah CSS tetapi dibentuk dengan Less, sebuah pre-prosesor yang memberi fleksibilitas dari CSS biasa. Bootstrap memberikan solusi rapi dan seragam terhadap solusi yang umum, tugas *interface* yang setiap pengembang hadapi. Bootstrap dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan *design* yang dibutuhkan.

Keunggulan dalam menggunakan Bootstrap adalah semua bagian untuk antarmuka pengguna menggunakan *style* CSS, Bootstrap dapat menggunakan Less *preprocessor* sebuah teknologi yang mengurangi dan mengefisienkan penulisan kode CSS. Bootstrap dapat diintegrasikan dengan JavaScript untuk menjadikan lebih menarik dengan efek-efek yang dapat diberikan dengan JavaScript.

Kelemahan dalam menggunakan Bootstrap adalah dengan adanya penggunaan Bootstrap menjadi tidak adanya keunikan di dalam *website* karena akan samanya tampilan yang diberikan. Terdapat juga laporan bahwa sistem *grid* pada Bootstrap tidak responsif (Oliver, 2015).

2.3.10 JQuery

JQuery adalah *add-on library* javascript. Keunggulan menggunakan jQuery dibandingkan dengan javascript, yaitu menyederhanakan kode javascript dengan cara memanggil fungsi-fungsi yang disediakan oleh jQuery. Javascript sendiri merupakan bahasa *scripting* yang bekerja di sisi *client/browser* sehingga *website*

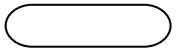
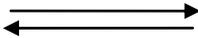
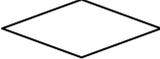
menjadi lebih interaktif. JQuery pertama kali dirilis tahun 2006 oleh John Resig. JQuery menjadi sangat populer sehingga telah digunakan pada banyak *website-website* kelas dunia seperti Google, Amazon, Twitter, ESPN, dan lain-lain (Beighley, 2010).

JQuery menyediakan fasilitas *plugins* yang beragam. JQuery dapat menyesuaikan *CSS style* dalam semua *web browser*, sehingga pengunjung dapat menikmati tampilan *web* yang bagus dari semua *browser*. JQuery juga menyediakan fasilitas untuk tampilan animasi seperti pada pembuatan animasi menggunakan *flash*. Selain itu JQuery didukung oleh komunitas dan pengembang JQuery yang tersebar di seluruh dunia. JQuery merupakan *open source*, sehingga bebas dan gratis digunakan oleh siapa saja (Sanjaya, 2012).

2.3.11 Flowchart

Flowchart merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis. Pembuatan *flowchart* tidak terdapat patokan yang bersifat mutlak karena *flowchart* merupakan gambaran hasil pemikiran dalam menganalisis suatu masalah dengan komputer (Eka Iswandy, 2014). *Flowchart* memiliki simbol-simbol yang digunakan untuk menggambarkan urutan program seperti pada Tabel 2.2.

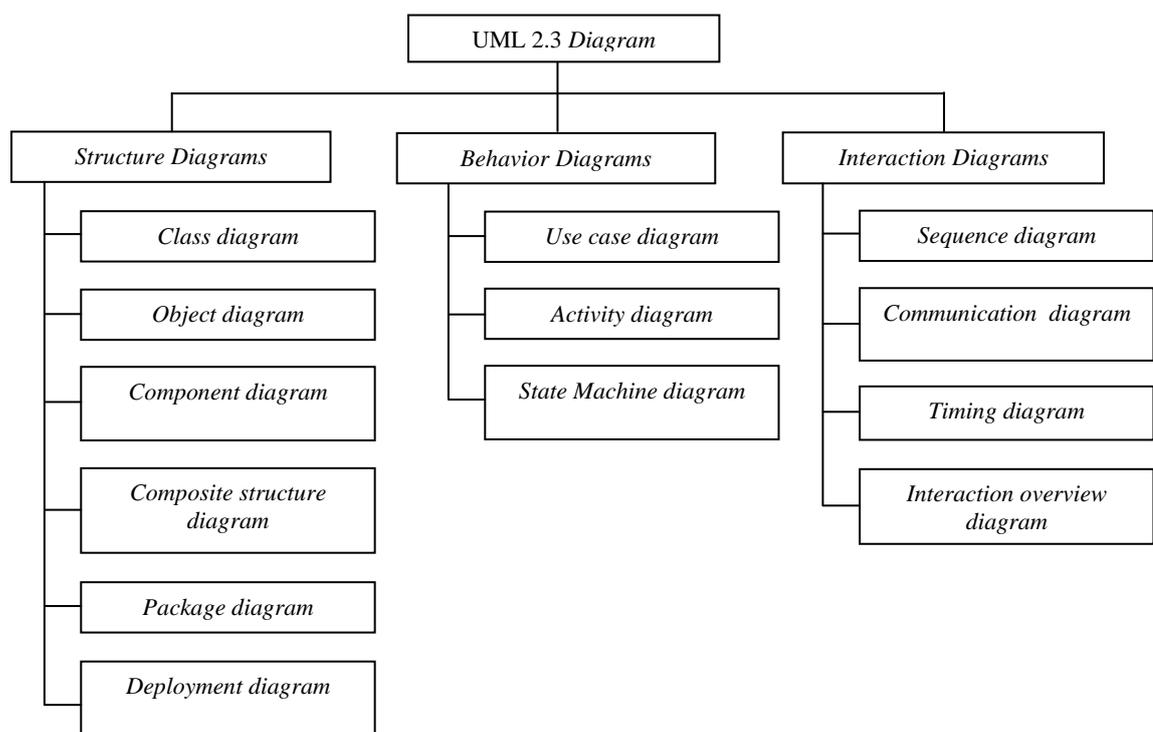
Tabel 2.2. Simbol-Simbol *Flowchart*

No.	Simbol	Deskripsi
1.		Simbol titik terminal digunakan untuk menunjukkan awal dan akhir suatu proses.
2.		Simbol arah data atau arus data atau aliran data
3.		Digunakan untuk mewakili suatu proses misalnya komputer melakukan proses pengolahan data pembeli.
4.		Simbol keputusan digunakan untuk suatu penyelesaian kondisi di dalam program.
5.		Simbol penyimpanan (<i>storage</i>) pada komputer, misalnya menyimpan <i>database</i> .

2.3.12 Unified Modeling Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa permodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul 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 (Rosa Ariani Sukamto, 2016).

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dapat dilihat pada Gambar 2.2.



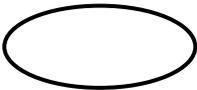
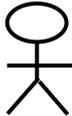
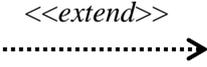
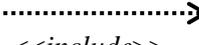
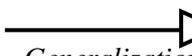
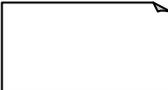
Gambar 2.2 Diagram UML (Rosa Ariani Sukamto, 2016)

2.3.12.1 Use case Diagram

Use case atau diagram *Use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah

sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa Ariani Sukamto, 2016). Simbol-simbol diagram *use case* ditunjukkan pada Tabel 2.3.

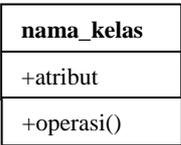
Tabel 2.3. Simbol-Simbol pada Diagram *Use case*

No.	Simbol	Deskripsi
1.	 <i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>Use case</i> .
2.	 <i>Aktor/Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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, mirip dengan prinsip inheritance pada pemrograman berorientasi objek.
5.	 <i><<include>></i>	Relasi <i>Use case</i> tambahan ke sebuah <i>Use case</i> dimana <i>Use case</i> yang ditambahkan memerlukan <i>Use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use case</i> ini.
6.	 <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.
7.	 <i>Realization</i>	Penjelasan use case dari sisi physical yaitu basis datanya.
8.	 <i>Note</i>	Memberikan informasi method apa yang ada didalamnya dan nama database.

2.3.12.2 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 atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa Ariani Sukamto, 2016). Simbol-simbol *class diagram* ditunjukkan pada Tabel 2.4.

Tabel 2.4. Simbol-Simbol pada *Class Diagram*

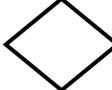
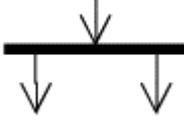
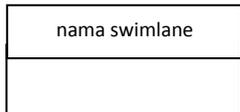
No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	generalisasi 	Relasi antar kelas dengan menggunakan makna generalisasi-spesialisasi (umum-khusus).
6.	kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)
8.	<i>multiplicity</i> 1...*	Satu atau lebih objek sebuah kelas yang berelasi dengan sebuah objek lain dari kelas lain yang berasosiasi dengan kelas tersebut.

2.3.12.3 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada

pada perangkat lunak. Diagram aktivitas juga banyak digunakan untuk mendefinisikan rancangan proses bisnis dimana ssetiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan, urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan, rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya, rancangan menu yang ditampilkan pada perangkat lunak (Rosa Ariani Sukamto, 2016). Simbol-simbol *activity diagram* ditunjukkan pada Tabel 2.5.

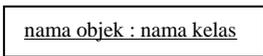
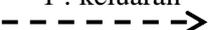
Tabel 2.5. Simbol-Simbol pada *ActivityDiagram*

No.	Simbol	Deskripsi
1.	status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.3.12.4 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case* (Rosa Ariani Sukanto, 2016). Simbol-simbol *sequence diagram* ditunjukkan pada Tabel 2.6.

Tabel 2.6. Simbol-Simbol pada *Sequence Diagram*

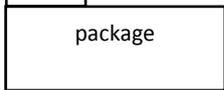
No.	Simbol	Deskripsi
1.	objek 	Menyatakan objek yang berinteraksi pesan.
2.	garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
3.	waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi.
4.	pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
5.	pesan tipe call 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
6.	pesan tipe send 1 : masukan 	Menyatakan suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
7.	pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

2.3.12.5 Component Diagram

Diagram komponen atau *component* diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah komponen *user interface* yang menangani tampilan, komponen *business processing* yang

menangani fungsi-fungsi proses bisnis, komponen *data* yang menangani manipulasi data, komponen *security* yang menangani keamanan sistem (Rosa Ariani Sukamto, 2016). Simbol-simbol *component diagram* ditunjukkan pada Tabel 2.7.

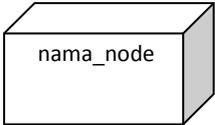
Tabel 2.7. Simbol-Simbol pada *Component Diagram*

No.	Simbol	Deskripsi
1.	<p><i>Package</i></p>  <p>package</p>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.
2.	<p>komponen</p> 	Komponen sistem.
3.	<p>kebergantungan / <i>dependency</i></p> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
4.	<p>antar muka / <i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
5.	<p><i>link</i></p> 	Relasi antar komponen.

2.3.12.6 Deployment Diagram

Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut : (1) Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*, (2) Sistem *client/server*, (3) Sistem terdistribusi murni, (4) Rekayasa ulang aplikasi (Rosa Ariani Sukamto, 2016). Simbol-simbol *deployment diagram* ditunjukkan pada Tabel 2.8.

Tabel 2.8. Simbol-Simbol pada *Deployment Diagram*

No.	Simbol	Deskripsi
1.	<i>Constraint</i> 	<i>Constraint</i> adalah mekanisme perpanjangan yang memungkinkan untuk menyempurnakan semantik elemen model UML.
2.	<i>Node</i> 	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
3.	kebergantungan / <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> .

2.3.13 Metode WebQual

WebQual merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari SERQUAL yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. Instrument penelitian pada WebQual tersebut dikembangkan dengan metode *Quality Function Development* (QFD).

WebQual sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa interaksi dalam penyusunan dimensi dan 14 butir pertanyaan. WebQual4.0 disusun berdasarkan penelitian pada tiga area yaitu *information quality*, *interaction quality* dan *usability quality* (Barnes, 2000).

2.3.14 Dimensi WebQual

Dimensi - dimensi pada WebQual terdiri dari tiga yaitu:

1. Kualitas Informasi (*Information Quality*)

Kualitas Informasi meliputi hal - hal seperti informasi yang akurat, informasi yang bisa dipercaya, informasi yang *up to date* terbaru, informasi yang sesuai dengan topik bahasan, informasi yang mudah di mengerti, informasi yang sangat detail, dan informasi yang disajikan dalam format desain yang sesuai (Barnes, 2003).

2. Kualitas Interaksi (*Interaction Quality*)

Meliputi kemampuan memberi rasa aman saat transaksi, memiliki reputasi yang bagus, memudahkan komunikasi, menciptakan perasaan emosional yang lebih personal, memiliki kepercayaan dalam menyimpan informasi pribadi pengguna, mampu menciptakan komunitas yang lebih spesifik, mampu memberi keyakinan bahwa janji yang disampaikan akan ditepati.

3. Kualitas Penggunaan (*Usability Quality*)

Meliputi kemudahan untuk dipelajari, kemudahan untuk dimengerti, kemudahan untuk ditelusuri, kemudahan untuk digunakan, sangat menarik, menampilkan bentuk visual yang menyenangkan, memiliki kompetensi yang baik, memberikan pengalaman baru yang menyenangkan.

Persepsi pengguna tersebut terdiri dari dua bagian, yaitu persepsi tentang mutu layanan yang dirasakan (aktual) dengan tingkat harapan (ideal). Barnes dan Vidgin melakukan penelitian dengan menggunakan WebQual untuk mengukur kualitas *website* yang dikelola oleh OECD (*Organization for Economic Cooperation and Development*). *Website* yang bermutu dari perspektif penggunadapat dilihat dari tingkat persepsi layanan aktual yang tinggi dan kesenjangan persepsi aktual dengan ideal (gap) yang rendah (Barnes, 2003).

Adapun instrumen - instrumen penelitian yang digunakan dalam WebQual4.0 mengacu pada Tabel 2.9.

Tabel 2.9. Instrumen WebQual 4.0 Menurut Barnes dan Vidgin

Kategori	Variabel <i>Webqual 4.0</i>
Kegunaan (<i>Usability Quality</i>)	<i>Website</i> mudah dipelajari dan dioperasikan
	Interaksi dalam menggunakan <i>website</i> tersebut jelas dan mudah dipahami
	Menu navigasi <i>website</i> mudah dipahami
	<i>Website</i> mudah digunakan
	Penampilan <i>website</i> menarik
	Desain tampilan <i>website</i> proporsional dan tepat dengan jenis <i>website</i> tersebut
	<i>Website</i> tersebut terlihat kompeten
	<i>Website</i> tersebut memberikan pengalaman positif bagi saya
Kualitas Informasi (<i>Information Quality</i>)	<i>Website</i> tersebut menampilkan informasi yang akurat
	<i>Website</i> tersebut menampilkan informasi yang dapat dipercaya
	<i>Website</i> tersebut memberikan informasi yang aktual
	<i>Website</i> tersebut menampilkan informasi yang relevan
	<i>Website</i> tersebut menampilkan informasi yang mudah dimengerti
	<i>Website</i> tersebut menampilkan detail informasi yang sesuai dengan keperluan pengguna
	<i>Website</i> tersebut menampilkan informasi dengan format yang tepat
Interaksi Layanan (<i>Service Interaction</i>)	Sistem memiliki reputasi yang baik
	Privasi data saya aman saat menggunakan <i>website</i> tersebut
	Saya merasa informasi pribadi saya aman pada <i>website</i> tersebut
	Penggunaan <i>website</i> terasa personal
	<i>Website</i> memberikan rasa kebersamaan dengan pengguna lain
	<i>Website</i> mempermudah komunikasi dalam organisasi/institusi
	Kepercayaan bahwa <i>website</i> tersebut membantu tercapainya kualitas pelayanan organisasi/institusi

	menjadi lebih baik
Keseluruhan	Tampilan keseluruhan <i>website</i>