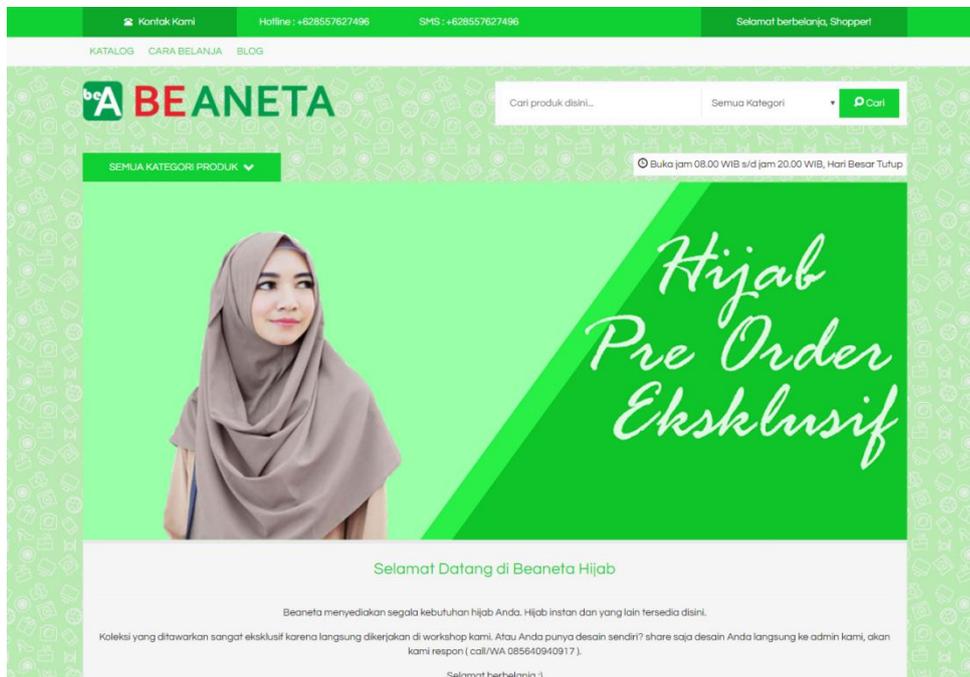


BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan dalam penelitian ini adalah karya ilmiah yang berjudul “Pengembangan *Website* Beaneta Hijab Dengan Pengujian Metode Webqual” Sidiq Eko Setiawan (2019). Jurnal ini membahas tentang pengembangan *website online shop* pada toko Beaneta hijab dengan pengujian webqual. Pembuatan *website* Beaneta hijab yaitu menggunakan *Content Management System* (CMS) sebagai pendukung sistemnya dengan tema Lapax theme dan kemudian diuji dengan menggunakan metode webqual sebagai pengukur kualitas *website* berdasarkan persepsi dari pengguna *website*. Adapun produknya adalah hijab dengan kategori hijab instan. Sistem pembeliannya memesan produk yang diinginkan melalui kontak yang ditampilkan dalam *website*. Salah satu contoh tampilan/ *interface website* beaneta hijab dapat disajikan pada Gambar 2.1.



Gambar 2.1 *Website* Beaneta Hijab

Sumber: <http://beaneta.com/>

Penelitian kedua dilakukan oleh Ibnu Dwi Lesmono (2018) dalam jurnal Swabumi yang berjudul “Rancang Bangun Sistem Informasi Penjualan Sepatu Berbasis *Website* dengan Metode *Waterfall*”. Pada penelitian ini membahas tentang perancangan sistem informasi penjualan *online e-commerce*”, perkembangan *e-commerce* dapat memicu timbulnya usaha dibidang apapun pada dunia maya. Perancangan sistem informasi penjualan sepatu berbasis *web* untuk memberikan pelayanan yang lebih baik lagi bagi pelanggan dan mempermudah dalam membeli produk sepatu. Pada pembuatan aplikasi *web* penjualan menggunakan model pengembangan perangkat lunak *waterfall* dan rancangan aplikasi menggunakan metode struktural *Entity Relationship Diagram (ERD)*. Dalam pembuatan aplikasi menggunakan PHP sebagai bahasa *script* dan XAMPP sebagai *database* tempat penyimpanan data. Program diuji dengan menggunakan metode *blackbox* dengan memasukkan data-data yang telah ada, teknik pengumpulan data yang digunakan dalam penulisan ini adalah dokumentasi, wawancara, dan studi pustaka. Salah satu contoh tampilan sistem informasi penjualan sepatu dapat disajikan pada Gambar 2.2.



Gambar 2.2 Tampilan Sistem Informasi Penjualan Sepatu

Sumber: Ibnu Dwi Lesmono (2018)

Penelitian ketiga dilakukan oleh Andri Prasetyo, Rahel Susanti (2016) dalam jurnal Ilmiah Teknologi dan Informasia ASIA (JITIKA) yang berjudul “Sistem Informasi Penjualan Berbasis Web Pada PT. Cahaya Sejahtera Sentosa Blitar”. Pada penelitian ini membahas bagaimana membangun sistem informasi penjualan *online* tiwul instan pada PT. Cahaya Sejahtera Sentosa Blitar. Kegiatan jual beli yang masih manual dan pemasaran serta promosi yang masih terbatas untuk itu diperlukan sitem informasi yang dapat memperluas pemasaran dan mempermudah transaksi penjualan sehingga lebih hemat, cepat dan mudah tanpa terhalang oleh jarak dan waktu serta dapat memudahkan pelanggan dalam pencarian informasi tiwul instan dengan pemesanan produk yang cukup mudah dengan bergabung menjadi member didalam sistem informasi ini dan pembayaran dilakukan melalui transfer bank. Berikut contoh tampilan sistem informasi penjualan pada PT.Cahaya Sejahtera Sentosa Blitar disajikan pada Gambar 2.3.

Screenshot of the PT. Cahaya Sejahtera Sentosa Blitar web-based sales information system interface. The page shows a date "Selasa, 3 Juli 2012" and a "Daftar produk" section. On the left, there is a "MENU" with links like BERITA, PROFIL, PROSES PRODUKSI, PRODUK, TIPS LAIN, and BANTUAN. Below the menu is a "PELANGGAN" section with a login form (User Name, Password, Login, Cancel) and links for "Pendaftaran Baru" and "Lupa Password". The main content area displays three products: Sari Tiwul, Sri Kandi, and Tepung. Each product listing includes an image, name, price, taste, benefits, weight, and stock.

Product Name	Price	Taste	Benefits	Weight	Stock
Sari Tiwul	Rp. 6000	Tawar, dan bisa diperkaya dengan rasa asin maupun manis	Penuh dengan nutrisi, berserat, lebih praktis, dan lebih kenyang.	250 gr	13
Sri Kandi	Rp. 8000	Tawar, dan bisa diperkaya dengan rasa asin maupun manis Dengan aroma khas daerah	Penuh dengan gizi, nutrisi lebih lengkap, berserat, lebih praktis, dan lebih kenyang	250gr	16
tepung	Rp. 2500	manis	kenyang	500gr	80

Gambar 2.3 Sistem Informasi PT. Cahaya Sejahtera Sentosa Blitar

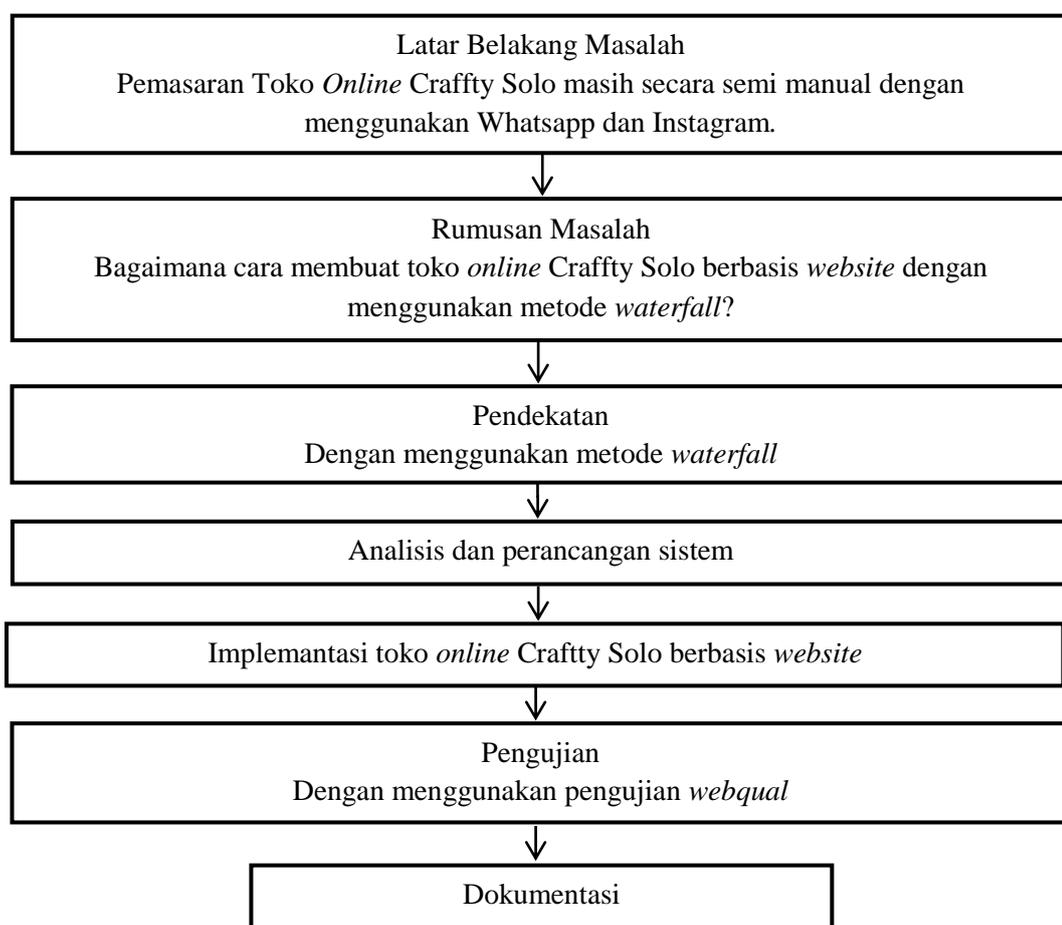
Sumber: Andri Prasetyo, Rahel Susanti (2016)

Berdasarkan tinjauan pustaka diatas dapat disimpulkan bahwa terdapat kesamaan tujuan penelitian yaitu membuat toko *online* berbasis *web* untuk memberikan pelayanan yang lebih baik lagi bagi pelanggan dan mempermudah dalam proses jual beli serta pencarian produk. Pada penelitian diatas dapat

disimpulkan bahwa ketiga jurnal penelitian diatas belum menggunakan *framework* dalam pembuatan sistem, padahal dalam pembuatan sistem diperlukan adanya penggunaan *framework* untuk mempermudah *maintenance* agar saat terjadi perubahan tidak merubah seluruh tatanan kode program.

2.2. Kerangka Pemikiran

Kerangka pemikiran penelitian tugas akhir ini ditunjukkan pada Gambar 2.4



Gambar 2.4 Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah :

1. Latar belakang masalah

Pokok permasalahan yang mendasari perlunya dibangun toko *online* Craftty Solo berbasis *website*. Selama ini pemasaran yang dilakukan adalah dengan melalui sosial media seperti Instagram dan Whattshap serta dengan bertatap muka

langsung antar penjual dan pembeli atau sering dikenal dengan COD (*Cash On Delivery*). Penjualan melalui sosial media tidak dapat dilakukan kontrol bisnis, pemilik bisnis tidak dapat menentukan sendiri tema desain situsnya serta tidak dapat melakukan SEO (*Search Engine Optimization*), gangguan lain yang menimpa sosial media juga akan mempengaruhi keberlangsungan bisnis dan kemungkinan apabila sosial media ditutup oleh pemilik situsnya.

Craftty Solo berkeinginan untuk memperluas pemasaran, oleh karena itu dibutuhkan sebuah sistem seperti pembuatan *website* untuk meningkatkan integritas dari *brand* Craftty Solo, membangun kepercayaan pelanggan dan juga untuk memperluas pemasaran tanpa dibatasi.

2. Perumusan masalah

Perumusan masalah merupakan inti permasalahan dan jalan keluar menyelesaikan permasalahan, bagaimana cara membuat toko *online* buket bunga wisuda Craftty Solo berbasis *website* menggunakan metode *waterfall* ?

3. Pendekatan

Pada penelitian ini menggunakan pendekatan metode *waterfall*. Metode ini digunakan untuk mengembangkan sistem dengan memiliki alur secara terurut yang dimulai dari analisa, desain, pengkodean dan pengujian.

4. Analisis dan perancangan sistem

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

5. Implementasi sistem

Pada penelitian dilakukan implementasi apa yang sudah dirancang untuk membangun toko *online* Craftty Solo berbasis *website*.

6. Pengujian sistem

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

7. Dokumentasi

Sistem yang sudah diimplementasikan, diuji coba dan dianalisis kemudian diterapkan pada Craftty Solo dan setelah itu dibuat dokumentasi dari keseluruhan kegiatan penyusunan tugas akhir ini.

2.3. Landasan Teori

2.3.1. Rancang Bangun

Menurut Fajriyah (2017) Pengertian rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan.

2.3.2. Media sosial

Menurut Celeb T. Carr & Rebecca A. Hayes (2015) Media sosial adalah media berbasis *Internet* yang memungkinkan pengguna berkesempatan untuk berinteraksi dan mempresentasikan diri, baik secara seketika ataupun tertunda, dengan khalayak luas maupun tidak yang mendorong nilai dari *user-generated content* dan persepsi interaksi dengan orang lain.

2.3.3. Penjualan *Online*

Menurut Sidiq Eko Setiawan (2019) dalam jurnalnya Penjualan *online* melakukan aktivitas penjualan dari mencari calon pembeli sampai menawarkan produk atau barang dengan memanfaatkan jaringan *internet* yang didukung dengan seperangkat alat elektronik sebagai penghubung dengan jaringan *internet*.

2.3.4. Toko *Online*

Toko *online* atau *Online Shop* adalah tempat pembelian barang dan jasa melalui media Internet, merupakan salah satu bentuk perdagangan elektronik (*e-commerce*) yang digunakan untuk kegiatan transaksi penjual ke penjual ataupun penjual ke konsumen. Toko *Online* di Indonesia semakin hari semakin menunjukkan perkembangan yang signifikan, belanja secara *online* tidak hanya dimonopoli belanja barang, namun juga layanan jasa seperti perbankan yang memperkenalkan teknik *e-banking*. Melalui teknik *e-banking* pelanggan dapat melakukan kegiatan seperti transfer uang, membayar tagihan listrik, air, telepon, Internet, pembelian pulsa, pembayaran uang kuliah dan lain sebagainya. Toko *Online* di Indonesia untuk pembelian suatu barang mengalami perkembangan yang cukup pesat. Mulai dari situs jualan handphone, gitar, butik, toko buku,

makanan, fashion bahkan hingga ke alat elektronik pun mulai dirambah oleh layanan belanja *online* (Dedy Ansari Harahap, Dita Amanah, 2018).

2.3.5. Website

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 (Fritz, 2014).

2.3.6. E-commerce

E-commerce atau Perdagangan elektronik (bahasa Inggris: *electronic commerce* atau *e-commerce*) adalah penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atau televisi, www, atau jaringan komputer lainnya. *E-commerce* dapat melibatkan transfer dana elektronik, pertukaran data elektronik, sistem manajemen inventori otomatis, dan sistem pengumpulan data otomatis (Fauzhi Eko Nugroho, 2016).

2.4. Software Pendukung

2.4.1. PHP (*PHP Hypertext Preprocessor*)

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* (Hakim, 2014).

2.4.2. Database

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* (Rosa dan M.Shalahudin, 2015).

2.4.3. MySQL

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 (Edy dan Zaki, 2014).

2.4.4. HTML (*Hypertext Markup Language*)

HTML (*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)

2.4.5. XAMPP

XAMP merupakan *tool* yang menyediakan paket perangkat lunak kedalam satu buah paket. Dengan meng-*instal* xampp maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server* Apache, PHP dan MySQL secara manual (Hendra, 2018). XAMPP merupakan paket PHP berbasis *open source* yang

dikembangkan oleh sebuah komunitas *open source*. Dengan menggunakan XAMPP tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah disediakan oleh XAMPP. Beberapa paket yang telah disediakan adalah Apache, MySQL, PHP, *Filezila* dan *phpmyadmin* (Hendra,2018).

2.4.6. 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.

2.4.7. Codeigniter

Codeigniter merupakan salah satu *framework* PHP yang tangguh dan populer. *Codeigniter* memudahkan *developer* untuk membuat aplikasi *web* dengan cepat mudah dibandingkan dengan membuatnya dari awal. Fungsi penggunaan *codeigniter* pertama adalah *codeigniter* akan menghasilkan suatu struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file PHPnya dikarenakan *codeigniter* dibangun berbasis MVC (*Model, View, Controller*) yang memisahkan antara tampilan dan logika aplikasi. Model adalah bagian yang bertanggung jawab terhadap operasi *database*, baik itu *create, read, update* dan *delete*, model berupa fungsi-fungsi operasional *database* yang dapat dipanggilkan oleh *controller*. *View* adalah bagian menangani tampilan, yang bertugas mempresentasikan data kepada *user*, *view* berbentuk struktur HTML yang berisikan variable data yang dikirimkan oleh *controller*. *Controller* adalah bagian yang mengatur hubungan antar *Model* dan *View*, *Controller* adalah otak

dari kinerja aplikasi dan terdiri dari fungsi-fungsi yang bersifat operasional dan logika (Nur, H. A., 2017).

2.5. Analisis Sistem

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 (Muhamad & Oktafianto, 2016).

2.6. Perancangan Sistem Berorientasi Objek

Menurut Rosa dan M.Shalahudin (2015) 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, pemrograman 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.

Banyak orang berfikir bahwa pemrograman berorientasi objek identik dengan bahasa Java. 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.

2.7. UML (*Unified Modelling Language*)

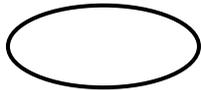
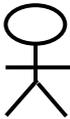
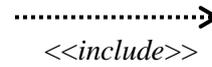
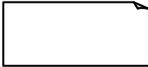
Menurut Sukamto dan Shalahuddin (2013) dijelaskan *Unified Modelling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman *Smalltalk* pada awal 1980-an, pada 1996 *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan september 1997 *Unified Modelling Language* (UML) diakomodasi oleh *Object Management Group* (OMG) sehingga sampai saat ini *Unified Modelling Language* (UML) telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.

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).

Berikut merupakan penjelasan mengenai symbol *use case diagram* yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

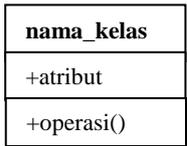
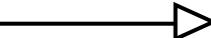
No.	Simbol	Deskripsi
1.	 <i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>Use case</i> .
2.	 Aktor/ <i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi 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. 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).

Berikut ini merupakan tabel penjelasan mengenai simbol *class diagram* ditunjukkan pada Tabel 2.2.

Tabel 2.2 Simbol-simbol *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.

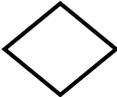
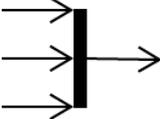
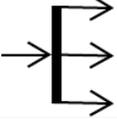
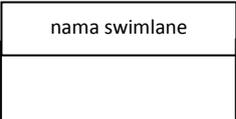
3. Activity Diagram

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 setiap 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).

Berikut merupakan penjelasan mengenai simbol *activity diagram* ditunjukkan pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Activity Diagram*

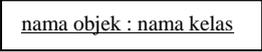
No.	Simbol	Deskripsi
1.	status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas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

4. Sequence Diagram

Sequence diagram 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 Sukamto, 2016). Simbol *sequence diagram* ditunjukkan pada Tabel 2.4.

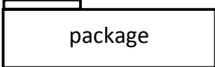
Tabel 2.4 Tabel *Sequence Diagram*

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.

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 *bussiness processing* yang menangani fungsi-fungsi proses bisnis, komponen *data* yang menangani manipulasi data, komponen *security* yang menangani keamanan sistem (Rosa Ariani Sukamto, 2016). Simbol *component diagram* ditunjukkan pada Tabel 2.5.

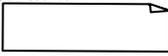
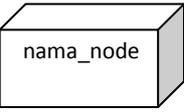
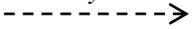
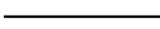
Tabel 2.5 Simbol *Component Diagram*.

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

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 Sukanto, 2016). Berikut merupakan penjelasan mengenai Simbol *deployment diagram* ditunjukkan pada Tabel 2.6.

Tabel 2.6 Simbol-simbol *Deployment*

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.	<i>dependency</i> 	Kebergantungan / <i>dependency</i> antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.	<i>link</i> 	Relasi antar <i>node</i> .

2.8. Metode Pengujian Sistem

2.8.1. Metode *Webqual*

Menurut Rosa dan M. Shalahudin (2015) 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 Sanjaya (2012) dalam jurnal Penelitian IPTEK-KOM, *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 (*informationquality*) dan interaksi layanan (*service interaction*). Masing-masing dimensi terdiri dari beberapa pernyataan yang ditunjukkan pada Tabel 2.7, Tabel 2.8 dan Tabel 2.9.

Tabel 2.7 Dimensi Kegunaan (*usability*)

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>uptodate</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.