

BAB II

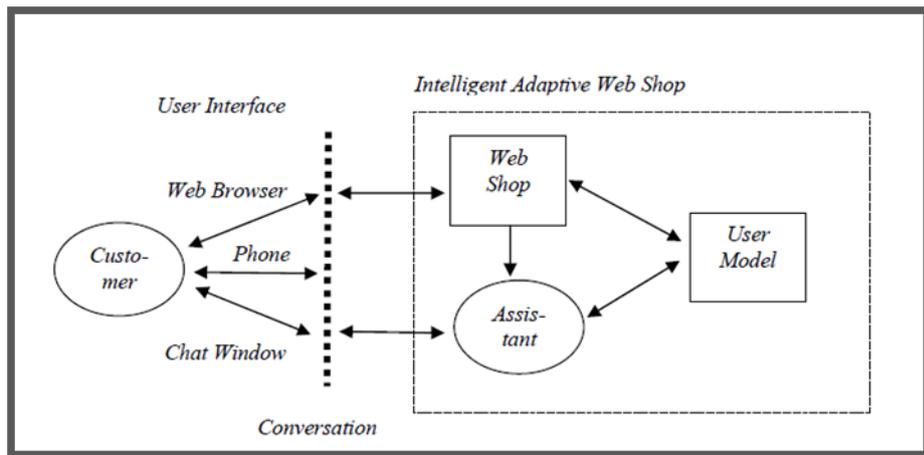
LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 E-Commerce

2.1.1.1 Apa Itu E-Commerce

E-Commerce (Electronic Commerce) adalah pembelian, penjualan dan pemasaran barang serta jasa melalui sistem elektronik, seperti radio, televisi dan jaringan komputer atau internet (Jony Wong, 2010). *E-Commerce* dapat dilakukan oleh siapa saja dengan mitra bisnisnya, tanpa dibatasi ruang dan waktu. Dalam aktivitas *e-commerce* sesungguhnya mengandung makna adanya hubungan antara penjual dan pembeli, transaksi antar pelaku bisnis, dan proses internal yang mendukung transaksi dengan perusahaan. Aktivitas *e-commerce* saat ini biasanya dilakukan menggunakan media berbasis website ataupun mobile seperti smartphone atau tablet. Oleh karena itu, perbaikan terus menerus mengenai pelayanan yang disediakan oleh sistem web *e-commerce* akan mempunyai pengaruh yang besar pada tingkat penjualan dan kepuasan konsumen. Adapun struktur sistem *e-commerce* berbasis web sebagaimana disajikan pada gambar berikut:



Gambar 1. Struktur Sistem E-Commerce Berbasis Web

Berdasarkan gambar diatas dapat penulis simpulkan bahwa konsumen dapat melakukan interaksi kepada pengusaha penyedia layanan e-commerce melalui tiga jalur. Yang pertama melalui web browser, kemudian melalui perangkat mobile seperti smartphone atau tablet dan yang terakhir melalui fasilitas chatting (chat window). Untuk mengenai informasi yang berkaitan dengan konsumen, maka akan diolah pada user model yang perusahaan akan gunakan sebagai database profil konsumen. Faktor-faktor yang menjadi pendorong dan penghambat implementasi e-commerce meliputi:

No	Faktor Pendukung	Faktor Penghambat
1	Globalisasi dan liberalisasi perdagangan	Faktor investasi

2	Kompetisi yang semakin tajam	Faktor teknis
3	Perkembangan teknologi	Faktor organisasi
4	Pengurangan tujuan secara fisik	Faktor jaringan
5	Publisitas	

Table 1. Faktor Pendukung dan Penghambat

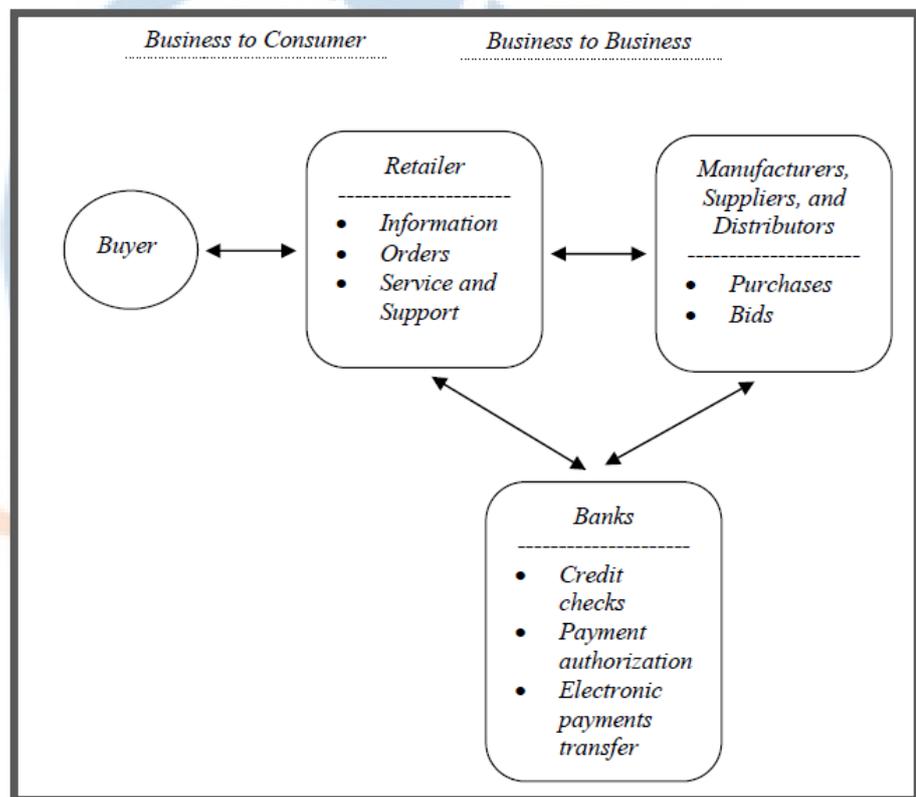
2.1.1.2 Jenis-Jenis E-Commerce

Dalam praktiknya, *e-commerce* dikelompokkan menjadi dua segmen, yaitu *business to business* (B2B) dan *business to consumer* (B2C). B2B *e-commerce* merupakan bentuk transaksi perdagangan melalui internet yang dilakukan oleh dua atau lebih perusahaan, sedangkan B2C *e-commerce* merupakan transaksi jual beli melalui internet antara penjual dengan konsumen (*end user*). Transaksi B2B melibatkan relatif lebih sedikit orang. Orang yang terlibat dalam transaksi B2B biasanya orang yang terlatih dalam mempergunakan sistem informasi dan telah terbiasa dengan proses bisnis yang dipengaruhi oleh transaksi. Jumlah transaksi lebih kecil tetapi memiliki nilai transaksi yang tinggi (McLeod dan Schell, 2004). Transaksi yang terjadi pada B2B dilakukan dalam bentuk *electronic data interchange (EDI)*, dan transaksi ini biasanya dilakukan dengan *supplier* atau *vendor*. Secara

fundamental transaksi B2C memiliki desain yang berbeda dengan. Konsumen yang dihadapi dalam transaksi B2C mungkin memiliki

atau tidak memiliki kemampuan dalam mempergunakan teknologi informasi. Oleh karena itu, di dalam web *e-commerce* untuk keperluan B2C mutlak harus dipasang panduan atau bantuan bagi konsumen yang mengalami kesulitan (McLeod dan Schell, 2004). Jika dibandingkan dengan B2B, jumlah transaksi B2C lebih besar, tetapi nilai transaksinya lebih kecil.

Di dalam proses transaksi *e-commerce*, baik itu B2B maupun B2C, terkadang melibatkan lembaga perbankan sebagai institusi yang menangani transfer pembayaran transaksi. Arus informasi pada transaksi *e-commerce* sebagaimana disajikan pada gambar berikut ini.



Gambar 2. Alur Informasi E-Commerce

2.1.1.3 Komponen E-Commerce

Dalam proses *e-commerce*, perusahaan membutuhkan beberapa komponen utama agar operasi dan manajemen aktivitas *e-commerce* berjalan dengan baik (Dewi Irmawati, 2011). Komponen-komponen pokok yang memiliki peran penting dalam proses *e-commerce* dunia usaha tampak seperti pada gambar berikut:

1. Pengendalian akses dan keamanan

Situs *e-commerce* harus memberikan rasa percaya dan akses yang aman untuk berbagai pihak dalam transaksi *e-commerce*, misalkan dengan adanya kata kunci (*password*), kunci enkripsi, sertifikasi, atau tanda tangan digital. Kemudian ada otorisasi akses yang hanya ke bagian tertentu saja sehingga hanya para pelanggan yang terdaftar saja yang dapat mengakses informasi dan aplikasi yang ada. Pengendalian akses dan keamanan ini perlu dilakukan untuk melindungi sumber daya situs *e-commerce* dari berbagai ancaman seperti peretas (*hacker*), pencurian *password* atau nomor kartu kredit, atau menghindari kegagalan sistem.

2. Membuat profil dan personalisasi

Proses pembuatan profil dan personalisasi menggunakan alat pembuat profil seperti pendaftaran, file cookie, software penelusur perilaku dalam situs web dan respon pemakai. Profil ini digunakan untuk mengenali kita sebagai pemakai individual, memberikan tampilan personalisasi, saran atas produk dan iklan web. Tujuan proses pembuatan profil ini untuk tujuan manajemen rekening, pembayaran, mengumpulkan data mengenai manajemen hubungan pelanggan, perencanaan pemasaran, dan untuk manajemen situs web itu sendiri.

3. Manajemen pencarian

Software *e-commerce* harus meliputi komponen mesin pencari situs web untuk dapat membantu para pelanggannya dalam menemukan produk dan jasa tertentu yang mereka inginkan untuk dievaluasi atau dibeli.

4. Manajemen isi dan katalog

Isi *e-commerce* sebagian besar berbentuk katalog multimedia yang memuat informasi produk sehingga membuat dan mengelola katalog merupakan rangkaian utama dari manajemen isi. Software manajemen isi tersebut bekerja dengan alat pembuat profile yang sudah disebutkan sebelumnya. Software manajemen isi akan membantu perusahaan *e-commerce* untuk mengembangkan, menghasilkan, mengirimkan, memperbaharui, dan menyimpan data teks serta informasi multimedia di situs web *e-commerce*. Selanjutnya manajemen isi dan katalog dapat diperluas untuk memasukkan proses konfigurasi produk yang akan mendukung layanan mandiri berbasis web dan penyesuaian massal atas berbagai produk perusahaan.

5. Manajemen arus kerja

Sistem arus kerja *e-business* digunakan untuk membantu para karyawan secara elektronik bekerja sama untuk menyelesaikan tugas pekerjaan dengan menggunakan mesin software arus kerja (*workflow software engine*). Sistem ini memastikan bahwa transaksi, keputusan, dan aktivitas kerja yang tepat dilakukan, serta data dan dokumen yang benar telah dikirimkan ke para karyawan, pelanggan, pemasok, dan pihak stakeholder.

6. Pemberitahuan kegiatan

Proses pemberitahuan kegiatan (*event notification*) memainkan peranan penting dalam sistem *e-commerce* karena sistem ini digunakan untuk memonitor semua proses *e-commerce* dan mencatat semua kegiatan yang relevan, termasuk perubahan mendadak atau ketika dalam masalah. Sistem ini akan memberitahukan kepada para pelanggan, pemasok, dan pegawai serta stakeholder mengenai semua kegiatan transaksi yang berkaitan dengan status mereka dengan melalui pesan elektronik seperti e-mail, newsgroup, penyeranta (pager), atau fax.

7. Kerjasama dan perdagangan

Tujuan utama *e-commerce* adalah untuk mendukung kesepakatan kerjasama dan layanan perdagangan yang dibutuhkan oleh para pelanggan, pemasok, dan stakeholder lainnya. Seperti halnya dalam *e-business*, sistem *e-commerce* juga fokus menumbuhkan komunitas berkepentingan online untuk meningkatkan layanan pelanggan dan membangun loyalitasnya.

8. Proses pembayaran elektronik

Pembayaran sebagai proses nyata dan penting dalam transaksi *e-commerce*. Sekarang ini sebagian besar sistem *e-commerce* yang terlibat dalam web dan bisnis B2C menggunakan proses pembayarannya dengan kartu kredit.

STT-NF

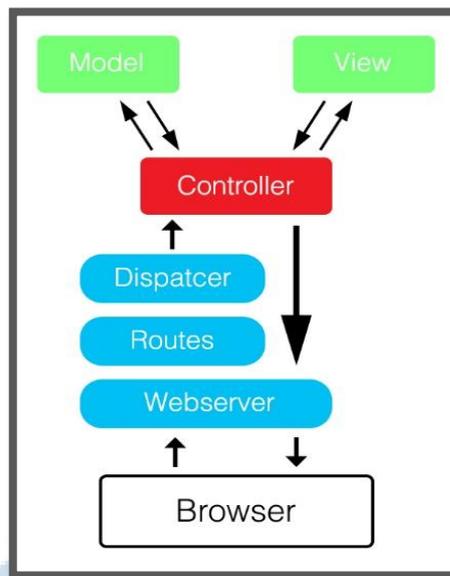
2.1.2 Web Framework

2.1.2.1 Framework

Framework adalah kumpulan dari file-file pustaka atau class-class yang terdapat dalam suatu kerangka kerja yang mendukung dalam pengembangan aplikasi secara terstruktur dan independent terhadap aplikasi. Menurut Rosa dan Shalahuddin (2011), framework merupakan kerangka kerja yang bertujuan untuk memudahkan dalam membuat sebuah aplikasi agar dapat dilakukan perubahan dengan cepat dan dapat digunakan kembali dengan aplikasi lainnya yang sejenis.

2.1.2.2 MVC

Model-View-Controller (MVC) adalah model pembuatan program yang menerapkan arsitektur aplikasi menjadi tiga bagian yaitu memisahkan antara proses, tampilan dan bagian yang menghubungkan dengan database. MVC bertujuan untuk memisahkan proses bisnis dari pertimbangan antarmuka user agar para pengembang bisa lebih mudah mengembangkan salah satu bagian dari aplikasi sehingga tidak memengaruhi bagian yang lain (Badiyanto, 2013). Dalam MVC model menggambarkan informasi (data) dan proses bisnis. View (tampilan) berisi elemen antarmuka seperti text, gambar, ataupun form masukan, sementara controller mengatur komunikasi antara view dan model. Jika dipetakan alur kerja sebuah MVC akan tampak seperti gambar 2.3.



Gambar 3. Konsep MVC

Badiyanto (2013) menjelaskan tentang Model-View-Controller sebagai berikut:

A. Model

Model merupakan kelas yang mendasari logika proses dalam aplikasi perangkat lunak dan kelas yang terkait dengannya. Model adalah suatu objek yang tidak mengandung informasi tentang user interface. Model juga merupakan suatu kelas yang berisi metode atau fungsi dan digunakan untuk menyimpan data dan aturan bisnis yang relevan.

B. View

View merupakan kumpulan dari kelas yang mewakili unsur-unsur dalam antarmuka, dalam view terdapat nama yang dipakai untuk mengidentifikasi *file script* tampilan saat dipanggil lewat fungsi render. Nama view sama seperti nama *file script* view-nya.

C. Controller

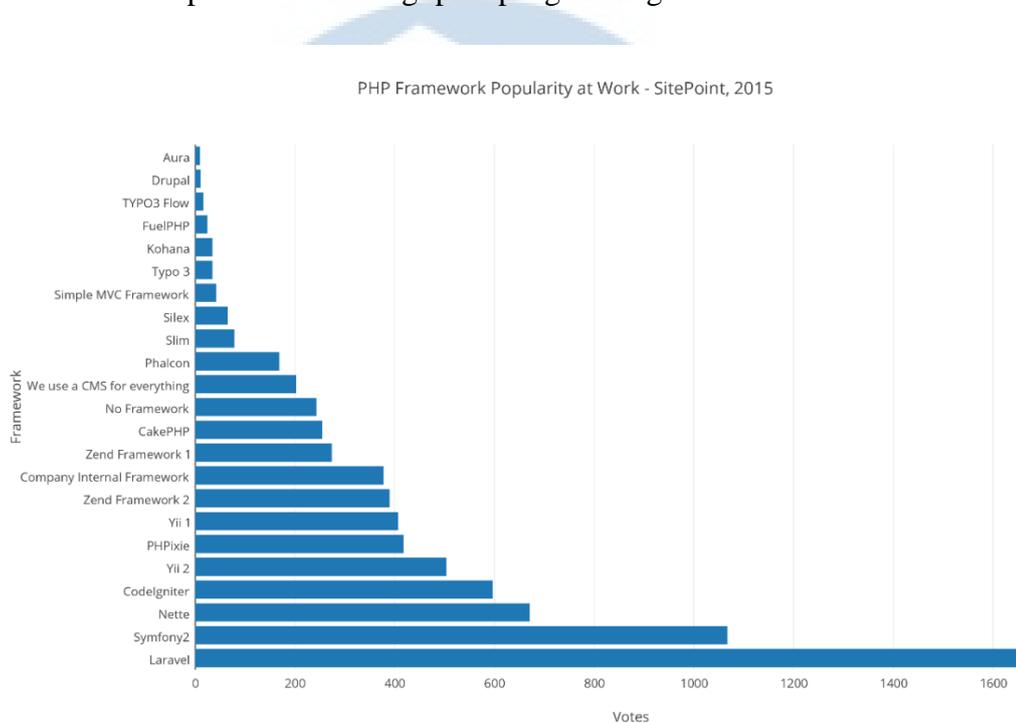
Controller merupakan kelas yang menghubungkan model dan view, digunakan untuk berkomunikasi antara kelas dalam model dan view. Controller mempunyai action standar. Ketika permintaan user tidak menetapkan action mana yang dijalankan, program akan menjalankan action standar.

2.1.2.3 Laravel

Laravel adalah salah satu framework MVC yang dibuat oleh Taylor Otwell pada tahun 2011. Laravel merupakan pengembangan web yang ditulis menggunakan bahasa pemrograman PHP. Walaupun termasuk framework baru, namun pengguna laravel sudah berkembang pesat dan mampu menjadi alternatif utama dari sejumlah framework besar seperti CodeIgniter & Yii. Laravel adalah sebuah MVC web development framework untuk sebuah pengembangan aplikasi yang diharapkan dapat meningkatkan kualitas aplikasi yang dihasilkan, dengan mengurangi biaya pengembangan dan perbaikan serta menghasilkan source code yang rapih dan fungsional yang dapat mengefisiensikan untuk implementasinya (Widodo dan Purnomo, 2016). Laravel merupakan framework PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya.

Laravel membuat proses development yang menyenangkan bagi pengembang tanpa mengurangi fungsionalitas aplikasi. Laravel juga merupakan framework yang mudah diakses, powerful dan menyediakan tools yang diperlukan untuk skala aplikasi besar.

Laravel adalah salah satu framework PHP yang up-to-date, karena laravel mengisyaratkan penggunaan PHP versi 5.3 keatas. Laravel dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di Github. Berdasarkan data dari situs *Trends Builwith* pada tanggal 12 April 2018 pengguna laravel saat ini mencapai 951.097 situs web di seluruh dunia. Dan dalam survei yang dilakukan oleh SitePoint, laravel adalah framework PHP terbaik untuk tahun 2015 mengalahkan Symfony2, Nette dan Codeigniter seperti pada gambar 2.4. Jadi dapat disimpulkan bahwa laravel saat ini menjadi pilihan utama bagi para pengembang web di seluruh dunia.



Gambar 4. Survei Pengguna Framework PHP Pada 2015

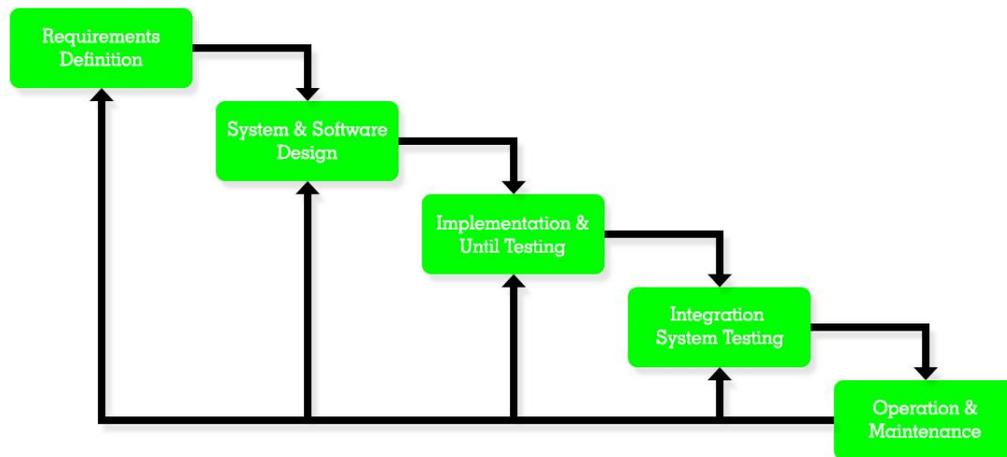
2.1.2.4 MySQL

Menurut Kadir (2013), MySQL (My Structure Query Language) adalah nama sebuah database server yang menangani akses database yang selalu dalam bentuk pernyataan SQL (Structured Query Language) yaitu suatu bahasa yang digunakan untuk mengakses database relasional. Menurut Aditya Alan Nur (2010) MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public Licensi). Data-data yang ada pada MySQL dikelola dalam sebuah database yang akan diletakkan pada beberapa tabel yang terpisah, sehingga manipulasi data akan menjadi jauh lebih cepat. Beberapa pertimbangan programmer memilih MySQL dalam mengolah database yaitu kecepatan, kemudahan, open source, dan lintas platform.

2.1.3 Model Pengembangan Software

2.1.3.1 Waterfall

Metode *Waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial dengan tujuan yang berbeda untuk setiap fase pengembangannya (Pressman, 2002). Metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara terstruktur dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (Rosa dan Shalahuddin, 2013). Berikut adalah ilustrasi metode *waterfall* pada gambar 2. 5.



Gambar 5. Alur Tahapan Pada Metode Waterfall

Penjelasan mengenai tahapan-tahapan pada metode *waterfall* yang ada pada gambar:

1. *Requirements Definition*

Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan user sistem. Persyaratan ini kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and Software Design*

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras dan perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya.

3. *Implementation and Until Testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

4. *Integration System Testing*

Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirimkan kepada pelanggan.

5. *Operation and Maintenance*

Ini adalah tahapan terakhir dalam metode *waterfall* dan biasanya (walau tidak selalu), ini merupakan fase siklus hidup yang paling lama. Sistem diinstal dan selanjutnya digunakan oleh user. Pemeliharaan mencakup koreksi dari berbagai error yang tidak ditemukan pada tahap-tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan (Ian Sommerville, 2003).

2.1.3.2 UML

Unified Modelling Language (UML) adalah satu set ketentuan Modelling yang digunakan untuk menspesifikan atau mendeskripsikan sebuah sistem peranti lunak dalam suatu kondisi dari objek. Beberapa model *tools* diagram yang digunakan sebagai dasar menggunakan UML antara lain (H. Bangun, 2009) :

1. Diagram *Use Case*

Diagram yang menggambarkan aktifitas *actors* dan *use case* yang dilakukan oleh sistem dari sudut pandang pengamatan seseorang. Model *use case* menggambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk teks

yang terdiri dari input user dan respon-respon sistem bukan bagaimana sistem itu bekerja.

2. Diagram *Class*

Diagram yang menggambarkan hubungan antara *class-class* yang didalamnya terdapat atribut dan fungsi dari suatu objek. Diagram *class* mempunyai 3 relasi dalam penggunaannya yaitu : *Assosiation* (hubungan interkasi antar *class*), *Generalization* (hubungan antar *class* dari khusus ke umum), *Constraint* (hubungan antar *class* yang dibatasi oleh sistem).

3. Diagram *Sequence*

Diagram yang menggambarkan *interaction* bagaimana suatu operasi dilakukan, apa yang dikirim dan kapan pelaksanaannya berdasarkan waktu. *Sequence* diagram membantu untuk mengidentifikasi setiap data masuk dan keluar dari sebuah sistem.

4. Diagram *Collaboration*

Diagram yang menggambarkan *interaction* sama halnya dengan diagram *sequence* akan tetapi lebih memusatkan atau memfokuskan pada kegiatan objek dari waktu pesan itu dikirimkan. Diagram *sequence* mengacu terhadap konteks objek sedangkan diagram *collaboration* mengacu terhadap waktu.

5. Diagram *State*

Diagram yang menggambarkan keadaan dari satu *state* ke *state* lainnya dalam masa transisi dari suatu objek pada sistem yang mengalami perubahan akibat dari respon yang diterima. Pada umumnya *stateclass* tidak dapat digambarkan untuk semua *class* dan satu *class* dapat memiliki lebih dari satu *stateclass*.

6. Diagram *Activity*

Diagram yang menggambarkan berbagai aliran dari aktivitas dalam sistem yang dirancang. Diagram *activity* digunakan untuk mendeskripsikan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih.

7. Diagram *Deployment*

Diagram yang menggambarkan konfigurasi fisik perangkat lunak dan perangkat keras pada sistem serta hubungan antara *node-node*. Tujuan atau fungsi dari *deployment* diagram yaitu untuk menggambarkan secara umum proses yang terjadi pada suatu sistem atau software.

2.1.3.3 BlackBox Testing

Black-box testing merupakan pengujian yang berpusat pada kebutuhan fungsional perangkat lunak dimana memungkinkan untuk memperoleh sekumpulan kondisi input yang secara penuh memeriksa fungsional dari sebuah aplikasi. *Black-box testing* berusaha menemukan kesalahan-kesalahan seperti kesalahan fungsi dan kesalahan tampilan aplikasi. *Black-box testing* dapat digunakan untuk menguji aplikasi konvensional dan aplikasi yang berorientasi objek (Pressman, 2010). Menurut Janner Simarmata (2010) dalam bukunya Rekayasa Perangkat Lunak klasifikasi *black box testing* adalah sebagai berikut:

1. Pengujian Fungsional (*functional testing*)

Pengujian dilakukan dalam bentuk tertulis untuk memeriksa apakah aplikasi berjalan seperti yang diharapkan. Pengujian fungsional meliputi seberapa baik sistem melaksanakan fungsinya,

termasuk perintah-perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar, dan integrasi.

2. Pengujian Tegangan (*stress testing*)

Pengujian tegangan berkaitan dengan kualitas aplikasi didalam lingkungan.

3. Pengujian Beban (*load testing*)

Pada pengujian beban, aplikasi akan diuji dengan beban berat atau masukan, seperti yang terjadi pada pengujian situs web, untuk mengetahui apakah aplikasi/situs gagal atau kinerjanya menurun.

4. Pengujian Khusus (*ad-hoc testing*)

Jenis pengujian ini dilakukan tanpa penciptaan rencana pengujian atau kasus pengujian. Salah satu penggunaan terbaik dari pengujian khusus adalah untuk penemuan. Pengujian ini membaca persyaratan atau spesifikasi (jika ada) jarang memberikan panduan yang jelas mengenai bagaimana sebuah program benar-benar bertindak, bahkan dokumentasi pengguna tidak menangkap “*look and feel*” dari sebuah program.

5. Pengujian Penyelidikan (*exploratory testing*)

Pengujian penyelidikan mirip dengan pengujian khusus dan dilakukan untuk mempelajari/mencari aplikasi.

6. Pengujian Usabilitas (*usability testing*)

Pengujian usabilitas adalah proses yang bekerja dengan pengguna akhir secara langsung maupun tidak langsung untuk menilai bagaimana pengguna merasakan paket perangkat lunak dan bagaimana mereka berinteraksi dengannya.

7. Pengujian Asap (*smoke testing*)

Pengujian ini dilakukan untuk memeriksa apakah aplikasi tersebut sudah siap untuk pengujian yang lebih besar dan bekerja dengan baik tanpa cela sampai tingkat yang paling diharapkan.

8. Pengujian Pemulihan (*recovery testing*)

Pengujian Pemulihan pada dasarnya dilakukan untuk memeriksa seberapa cepat dan baiknya aplikasi bisa pulih terhadap semua jenis crash atau kegagalan hardware, masalah bencana, dan lain-lain.

9. Pengujian Volume (*volume testing*)

Pengujian volume adalah pengujian sebuah sistem (baik perangkat keras dan perangkat lunak) untuk serangkaian pengujian dengan volume data yang diproses adalah subjek dari pengujian.

10. Pengujian Domain (*domain testing*)

Pengujian domain merupakan penjelasan yang paling sering menjelaskan teknik pengujian.

11. Pengujian Skenario (*scenario testing*)

Pengujian skenario adalah pengujian yang realistis, kredibel dan memotivasi stakeholder, tantangan untuk program dan mempermudah pengujian untuk melakukan evaluasi.

12. Pengujian Regresi (*regression testing*)

Pengujian regresi adalah gaya pengujian yang berfokus pada pengujian ulang setelah ada perubahan. Pada pengujian regresi berorientasi risiko.

13. Penerimaan Pengguna (*user acceptance*)

Pada jenis pengujian ini, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan.

14. Pengujian Alfa (*alpha testing*)

Pada jenis pengujian ini, pengguna akan diundang ke pusat pengembangan. Pengguna akan menggunakan aplikasi dan pengembang mencatat setiap masukan atau tindakan yang dilakukan oleh pengguna.

15. Pengujian Beta (*beta testing*)

Pada jenis pengujian ini, perangkat lunak didistribusikan sebagai sebuah versi beta dengan pengguna yang menguji aplikasi disitus mereka. Pengecualian atau cacat yang terjadi akan dilaporkan kepada pengembang.

2.1.3.4 UAT (User Acceptance Test)

Menurut Perry (2006), *User Acceptance Testing* merupakan pengujian yang dilakukan oleh *end-user* dimana user tersebut adalah staff atau karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya. Setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem software memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

2.1.3.5 Scala Likert

Skala likert menurut Sugiyono (2010) digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang

tentang fenomena sosial. *Skala Likert* adalah skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang suatu gejala atau fenomena pendidikan. Secara umum *skala likert* adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei.

2.2 Penelitian Terkait

2.2.1 Tabel Penelitian

No	Judul	Penulis	Tahun	Framework	Kesimpulan
1	Analisa Dan Perancangan Website <i>E-Commerce</i> Berbasis Framework Codeigniter Pada Denayu Bag Bantul	Ahmad, Anggit Dwi Hartanto	2013	Codeigniter	Aplikasi web pada Denayu Bag Bantul dikembangkan dengan menggunakan framework PHP Codeigniter. Penggunaan framework Codeigniter membantu memudahkan dalam mengembangkan aplikasi ini.
2	Implementasi Teknologi Framework Yii Di	Gisma Gerry	2018	Yii	Aplikasi Toko Passion ini berbasis website, dan dikembangkan menggunakan

	Toko Passion Yogyakarta				framework Yii dalam perancangan sistem toko seperti katalog produk, pembelian dan pembayaran dan pengelolaan data user.
3	Implementasi Framework CakePHP Pada Website Penjualan Online (Studi Kasus Unit Produksi Karawi SMKN 3 Kota Bima)	Nurhaidah	2014	CakePHP	Penelitian ini membahas mengenai pembuatan website untuk toko unit produksi karawi dengan menggunakan framework cakePHP. Dan menggunakan fitur <i>SEO (Search Engine Optimization)</i> dalam memaksimalkan penjualannya
4	Rancang Bangun Pembuatan Aplikasi Website Alcatech dan Akademik Informasi Sistem (AIS) Menggunakan Framework Laravel	Muhammad Rosul	2019	Laravel	Penelitian ini adalah membuat rancang bangun pada aplikasi web <i>e-commerce</i> untuk STT Terpadu Nurul Fikri. Pengembangan aplikasinya menggunakan framework laravel, karena saat ini laravel

					menjadi framework yang paling banyak digunakan.
--	--	--	--	--	---

Table 2. Perbandingan Penelitian

Berdasarkan keempat penelitian diatas dapat disimpulkan bahwa saat ini pengembangan aplikasi berbasis web masih banyak menggunakan framework PHP seperti codeigniter, laravel dan Yii. Karena menggunakan framework akan mempermudah, dan mempercepat pengembangan aplikasi web. Ketiga framework ini masing-masing memiliki kekurangan dan kelebihan, jadi penggunaannya tergantung kebutuhan dalam pengembangan aplikasi web. Dan framework PHP yang paling banyak digunakan saat ini di seluruh dunia adalah framework laravel.

