

BAB II

Tinjauan Pustaka

2.1 Tinjauan Pustaka

Aplikasi *web* adalah sebuah program yang dikirim melalui internet yang disimpan dalam server dan dapat diakses melalui antar muka *web browser*. Aplikasi *web* juga dapat diartikan sebagai suatu perangkat lunak komputer yang dikodekan dalam bahasa pemrograman yang didukung oleh perangkat lunak seperti *html*, *javascript*, *ruby*, *python*, *php*, *java*, *ajax* dan bahasa pemrograman lainnya. Tinjauan pustaka bertujuan sebagai referensi terhadap hasil penelitian sebelumnya.

Carana (2012) Dalam penelitiannya yang berjudul “aplikasi sistem informasi dan pemesanan makanan dan minuman berbasis *web* pada heritage cafe menggunakan *jquery*” dimana dalam proses penjualannya masih menggunakannya sistem konvensional, yaitu masih dilakukan dengan cara lama dan belum memaksimalkan penggunaan teknologi modern. Para konsumen harus datang ataupun menelepon untuk melakukan pemesanan. Fokus pada tahap analisis dan perancangan dalam pembuatan *web* ini berupa data *flow diagram* (dfd) dan normalisasi. Lalu tahap analisis kebutuhan sistem dan analisis kelayakan juga sangat membantu dalam pengambilan keputusan sesuatu sistem apakah layak atau tidak layak untuk dikembangkan. Kolaborasi *html*, *php*, *css*, *jquery* memberi dampak yang menarik pada tampilan *web* serta memberi fungsi yang sederhana dan mudah saat digunakan maupun dikembangkan.

Istiawan (2015) Dalam penelitiannya yang berjudul “Sistem Informasi Penjualan & Pemesanan Makanan Secara *Online* di Dapur Catering Bu Ti Kediri Berbasis *Web*” mengemukakan bahwa catering bu ti dimana transaksi penjualan di dapur catering Bu Ti semuanya masih dilakukan secara manual yaitu pelanggan datang langsung ke toko tersebut untuk membeli atau memesan makanan yang diinginkan. Metode pengembangan sistem yang penulis gunakan adalah *sdic* (*system development life cycle*) dengan model proses *waterfall*. Metode ini bisa juga disebut dengan *linier sukuensial model* dimana menggunakan pendekatan sistematis dan sekuensial dalam pengembangan aplikasi yang dimulai melalui proses analisis, desain, pengkodean uji coba dan pemeliharaan.

Sasongko (2010) Dalam penelitiannya yang berjudul “Aplikasi Pemesanan Makanan Dan Minuman Pada Rumah Makan” mengemukakan bahwa penggunaan media alat tulis dan kertas untuk pemesanan makanan dan minuman di rumah makan masih digunakan sampai saat ini. Perancangan aplikasi pemesanan ini terdiri dari perancangan untuk *client*, untuk kasir dan

untuk dapur. proses perancangan untuk tiap perancangan bagian terdiri dari struktur navigasi, *flowchart*, *entity relationship diagram*, normalisasi, *database* dan perancangan tampilan. aplikasi pemesanan pada *client* terdiri dari perancangan info lokasi, daftar menu, tentang dan pemesanan. pada pemesanan terdiri dari pesan, batal dan *history*. Aplikasi untuk kasir terdiri atas perancangan halaman makanan, minuman, tambah menu, transaksi dan pembayaran. Sedangkan aplikasi untuk dapur terdiri dari perancangan untuk pemesanan dan status.

Widarda Dan Hakim (2014) Dalam penelitiannya yang berjudul “Perancangan Sistem Informasi Pemesanan Menu Berbasis *Web* (Studi Kasus: Pada Coffe Toffe Tembalang Semarang)” mengemukakan bahwa pengalihan sistem manual menjadi sistem komputerisasi bisa menjadi cara alternatif untuk mengintegrasikan satu sama lain. peneliti menggunakan wawancara, observasi, dan sastra belajar sebagai koleksi data. desain sistem dalam penelitian ini adalah sistemik dan berturut-turut. Mengenai pengamatan, gerai coffee toffee tembalang masih menggunakan sistem manual. Hasil dari penelitian ini memungkinkan pelanggan untuk melihat dan menu *order by themselfe*. Pelanggan dapat memesan langsung tanpa melalui pelayan dan secara otomatis memberikan pemberitahuan kepada barista mencetak formulir pemesanan. supervisor dapat menggunakan fitur laporan ketika pemilik meminta laporan penjualan berdasarkan periode tertentu sehingga kinerja manajemen untuk menjadi efisien.

Inayati, Hidayatulloh Dan Kamisutara (2015) Dalam penelitiannya yang berjudul “Aplikasi Pemesanan Makanan Berbasis *Web* (Studi Kasus Rm Lesehan Berkah Ilaahi Gresik)” memaparkan proses pembangunan sistem pemesanan berbasis *web* dengan menekankan pada tahap analisa, desain, dan implementasi. Analisis kondisi lapangan dilakukan dengan cara observasi terhadap sistem lama, wawancara dan kuesioner pelanggan. Hasil analisa akan digambarkan menggunakan notasi uml (*unified modeling language*) untuk selanjutnya diimplementasikan dalam sebuah aplikasi e-crm menggunakan bahasa pemrograman php serta basis data postgresql. Metode pengembangan yang digunakan adalah *object oriented* dengan memanfaatkan *yii framework* yang merupakan *framework php* berbasis *model view controller* (mvc). Selain itu digunakan pula *bootstrap framework* dari sisi desain aplikasi untuk memberikan fleksibilitas aplikasi ketika diakses dengan *device* yang resolusinya lebih kecil seperti telepon genggam.

Sistem pemesanan menu berbasis *web* dapat menjadi salah satu solusi yang dapat digunakan untuk mempermudah pelanggan melihat menu-menu yang ditawarkan dan dapat menghindari kesalahan pencatatan manual. dalam kelima penelitian sebelumnya, aplikasi yang

dilakukan menunjukkan adanya fungsionalitas pemesanan pada sistem yang dikembangkan menonjolkan promosi dan penawaran, harus memasukkan kode makanan yang dipesan sehingga memakan waktu, namun penelitian ini hanya menonjolkan sistem pemesanan makanan. oleh karena itu, penelitian *website* untuk pemesanan pastel ondrowino di legend coffee menjadi solusi untuk mempermudah dalam konfirmasi pembayaran, pencarian data, pengolahan data transaksi penjualan dan pengiriman. Metode pengembangan sistem yang digunakan adalah *waterfall* dalam *website* ini menggunakan fitur *bootstrap*. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman php, html, *javascript*, css dan mysql sebagai *databasenya*.

2.2 Landasan Teori

Untuk mendukung hasil penelitian diperlukan suatu konsep dalam merumuskan definisi-definisi yang menunjang kegiatan penelitian baik teori dasar maupun teori umum.

2.2.1 Sistem Informasi

Definisi sistem informasi menurut Rommey (1997:16) adalah cara untuk mengumpulkan, memasukan, mengolah, dan menyimpan data dan terorganisasi cara untuk menyimpan, mengelola, mengenc informasi dengan cara yang suatu organisasi dapat mencapai tujuan

Edhy Sutanta (2009: 4) men umum sebagai kumpulan hal atau elemen yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem dan sasaran sistem.

Dengan demikian sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai suatu tujuan dan sasaran dalam ruang lingkup yang sempit.

2.2.2 Pengembangan Sistem Berbasis Web

Simarmata (2009) Pengembangan sistem berbasis *web* adalah aplikasi yang sejak awal dirancang untuk dieksekusi di lingkungan berbasis *web*. Definisi ini mengungkapkan dua aspek penting dari aplikasi ini sebagai berikut:

1. Suatu aplikasi *web* dirancang dapat berjalan di dalam lingkungan berbasis *web*. artinya aspek-aspek hipermedia dalam kaitannya dengan *hiperteks* dan multimedia di dalam kombinasi dengan kelola aplikasi tradisional harus diperhitungkan di seluruh hidup aplikasi.
2. Aplikasi *web* adalah suatu aplikasi yang tidak hanya berupa sekumpulan halaman-halaman *web*.

2.2.3 Javascript

Yeni Kustuyahningsih Dan Devi Rosa Anamisa (2011) *javascript* adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen html. *javascript* adalah bahasa yang “*case sensitive*” artinya menandakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil.

2.2.4 Css (*Cascading Style Sheet*)

Menurut Prasetio (2011) *Cascading Style Sheet* adalah teknologi yang digunakan untuk memperindah halaman *web* (*Cascading Style Sheet* adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. *css* bekerja sebagai pelengkap pada elemen html yang kesemuanya itu dapat dikendalikan dengan menggunakan sebuah bahasa *script* *css*. *Css* mempunyai 2 bagian utama yaitu *selectors* dan deklarasi. yang dimaksud *selectors* biasanya elemen html yang ingin dirubah, sedangkan deklarasi biasanya terdiri dari properti dan nilai. properti adalah atribut *style* yang di ingin diubah dan setiap properti memiliki nilai. menurut (bunafit nugroho, 2014: 1)

2.2.5 Bootstrap

Menurut Otto (2011) *Bootstrap* merupakan sebuah *framework* *css* yang memudahkan pengembang untuk membangun *web* yang menarik dan responsif. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk mengembangkan dan pemeliharaannya. *bootstrap* adalah *css* tetapi dibentuk dengan *less*, sebuah *pre-processor* yang memberi fleksibilitas dari *css* biasa. *bootstrap* memberikan solusi rapi dan seragam terhadap solusi yang umum dan tugas *interface* yang setiap pengembang hadapi. *Bootstrap* dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan *design* yang dibutuhkan.

2.2.6 Xampp

Menurut Wicaksono (2008:7) menjelaskan bahwa *xampp* adalah sebuah *software* yang berfungsi untuk menjalankan *web* berbasis *php* dan menggunakan pengolah data *mysql* di komputer lokal. *Xampp* merupakan paket *php* dan *mysql* berbasis *open source*, yang dapat

digunakan sebagai tool pembantu pengembangan aplikasi berbasis *web*. Xampp berperan sebagai *server web* pada komputer lokal. xampp juga dapat disebut sebuah *cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *web* tanpa harus *online* atau terakses dengan *internet*. Dikutip dalam Choliviana, Triyono & Sukadi (2012)

2.2.7 Mysql

Menurut Anhar (2010:B5) mengatakan bahwa “Mysql (*My Structured Query Language*) adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dbms (*database management system*), sifat dari dbms ini adalah *open source*.”

2.2.8 Alat Bantu Perancangan Sistem

Alat bantu digunakan untuk mendukung kesuksesan dalam dalam tahapan yang dilalui agar pemakaian alat-alat tersebut sesuai dengan kegunaannya maka diperlukan teori sebagai alat bantu untuk perancangan sistem yang nantinya digunakan adalah sebagai berikut :

1. *Flowchart*

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. dengan demikian setiap simbol menggambarkan proses tertentu.

2. *Site Map*

Site Map adalah daftar isi dari sebuah situs yang berbasis file xml, *site map* sering disebut juga dengan peta situs. dengan adanya *site map* memudahkan pengenalan halaman-halaman atau konten-konten situs anda kepada *search engine*. *Site map* tidak berfungsi hanya untuk *blogspot* saja, tetapi semua situs atau *website* membutuhkan *site map* seperti *wordpress*, *joomla*, *blogdetik*, dan lain sebagainya (Upan Blog, Selasa, 21 Maret 2017).

Site map juga merupakan sebuah cara sederhana bagi para *webmaster* atau *wapmaster* untuk menginformasikan kepada *search engine* mengenai halaman-halaman pada situs mereka yang bisa di *crawl*. Bentuk paling sederhana dari *sitemap* adalah berupa *file xml* yang berisi daftar url sebuah situs beserta metadata-nya masing-masing.

3. *Unified Modeling Language* (UML)

Unified Modeling Language (Uml) menurut (Nugroho, 2010:6) Adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). *Unified Modeling Language* (Uml) adalah bahasa yang telah menjadi standar untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak. Sedangkan *unified modeling language* (uml) adalah keluarga notasi grafis yang didukung oleh meta-model 28 tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. Jenis diagram itu antara lain:

a. Diagram Kelas (*Class Diagram*)

Bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antar muka-antar muka, kolaborasi-kolaborasi, serta relasi-relasi. diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

b. Diagram Paket (*Package Diagram*)

Bersifat statis. diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

c. Diagram Use-Case (*Usecase Diagram*)

Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

d. Diagram Interaksi dan *Sequence* (*Sequence Diagram*)

Bersifat dinamis. Diagram urutan adalah iterasiksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

e. Diagram Komunikasi (*Communication Diagram*)

Bersifat dinamis. Diagram komunikasi uml yang menekankan organisasi objek yang menerima serta mengirim pesan.

- f. Diagram *Statechart* (*Statechart Diagram*)
Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktivitas.
- g. Diagram Aktivitas (*Activity Diagram*)
Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.
- h. Diagram Komponen (*Component Diagram*)
Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya.
- i. Diagram *Deployment* (*Deployment Diagram*)
Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada uml dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data *flow diagram*, *entity relationship diagram*, dan sebagainya.

4. *Entity Relationship Diagram* (Erd)

Menurut (Brady Dan Loonam, 2010) *entity relationship diagram* (erd) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *system analysts* dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari pengembangan. erd bersama-sama dengan detail pendukung mereka pada gilirannya digunakan sebagai spesifikasi untuk *database*. erd untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. erd untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

a. Entitas

Pengertian menurut Brady Dan Loonam (2010), entitas adalah objek yang menarik di bidang organisasi yang dimodelkan. Pada *post* sebelumnya mengenai basis data telah dijelaskan sedikit tentang pengertian entity (entitas) yaitu suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data.

Contoh : Mahasiswa, kartu anggota perpustakaan (kap), dan buku.

b. Hubungan (*Relasi Atau Relationship*)

Suatu hubungan adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.

Contoh : Mahasiswa mendaftar sebagai anggota perpustakaan (kap), relasinya adalah mendaftar.

c. Atribut

Atribut memberikan informasi lebih rinci tentang jenis entitas. atribut memiliki struktur internal berupa tipe data.

Jenis-jenis atribut:

- Atribut *Key*

Atribut *key* adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*record*) dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan *key* tidak boleh ada baris data dengan nilai yang sama.

Contoh : Nomor Pendaftaran (Nomor) Nomor Induk Mahasiswa (Nim) dan nomor pokok lain

- Atribut *Simple*

Atribut yang bernilai tunggal, tidak dapat dipecah atau dipilah lagi.

Contoh: Alamat, Penerbit, Tahun Terbit, Judul Buku.

- Atribut *Multivalued*

Nilai dari suatu atribut yang mempunyai lebih dari satu (*multivalued*) nilai dari atribut yang bersangkutan.

Contoh: Dari sebuah buku, yaitu terdapat beberapa pengarang.

- Atribut *Composite*

Atribut *composite* adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisa dipecah lagi atau mempunyai sub atribut.

Contoh: Dari entitas nama yaitu nama depan, nama tengah, dan nama belakang.

- Atribut *Derivatif*

Atribut yang tidak harus disimpan dalam *database* ex. total atau atribut yang dihasilkan dari atribut lain atau dari suatu *relationship*. Atribut ini dilambangkan dengan bentuk oval yang bergaris putus-putus.

2.2.9 Metode Pengembangan Sistem

Menurut Rosa dan Shalahuddin (2011), SDLC atau *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model dan metodologi yang digunakan orang untuk mengembangkan sistem sebelumnya.

2.2.10 Waterfall

Menurut Roger S. Pressman (2010: 257) dalam Widarda dan Hakim (2014) model air terjun (*waterfall*) kadang dinamakan siklus hidup klasik (*classic life cycle*) dimana hal ini menyuarakan pendekatan sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak yang dimulai dengan spesifikasi kebutuhan pengguna sampai dengan penyerahan sistem ke pelanggan. Tahapan utama pengembangan sistem terdiri dari tahapan perencanaan sistem (*system planning*), konstruksi (*construction*), dan penyerahan sistem ke para pelanggan atau pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak lengkap yang dihasilkan.

Langkah-langkah dalam tahap perancangan adalah (Roger S. Pressman, 2010:17) :

1. Komunikasi, proses tahapan ini untuk memahami tujuan-tujuan *stakeholder* atas proyek perangkat lunak yang sedang dikembangkan dan mengumpulkan kebutuhan-kebutuhan yang akan membantu mengartikan fitur-fitur perangkat lunak beserta fungsi-fungsinya.
2. Perencanaan, rencana proyek perangkat lunak mengartikan kerja rekayasa perangkat lunak dengan menggambarkan tugas-tugas teknis yang harus dilakukan, resiko-resiko yang mungkin muncul, sumber daya yang akan dibutuhkan, produk-produk kerja yang harus dilakukan dan jadwal-jadwal kerja.
3. Pemodelan, proses tahapan ini untuk membuat sketsa-sketsa menjadi lebih terperinci supaya dapat memahami masalah sesungguhnya dan lebih memahami bagaimana sistem akan memecahkan masalah tersebut. Ini lebih kearah membuat model-model untuk memahami kebutuhan perangkat lunak maupun rancangan-rancangan yang akan memenuhi kebutuhan tersebut.

4. Kontruksi, kegiatan ini menggabungkan pembentukan kode (*codegeneration*) dan pengujian yang sangat dibutuhkan untuk menemukan kekeliruan atau kesalahan dalam kode program komputer yang akan dihasilkan sebelumnya.
5. *Deployment*, perangkat lunak disajikan kepada pelanggan yang kemudian akan mengevaluasi produk yang disajikan dan akan memberikan umpan balik berdasarkan evaluasi tersebut.

2.2.11 Prototype

Rifani (2015) dalam penelitiannya menggunakan metode pengembangan sistem *prototype*. *prototype* di definisikan sebagai pendekatan ke desain sistem yang mengembangkan model kerja yang disederhanakan dari sistem *Prototype* atau rancangan awal ini dapat dengan cepat dan murah untuk dibangun oleh para pemakai atau diuji. Berikut merupakan gambar alur proses yang menunjukkan bahwa *Prototype-prototype* dibuat untuk memuaskan kebutuhan klien dan untuk memahami kebutuhan klien lebih baik.

Berikut langkah-langkah yang digunakan di dalam metode pengembangan sistem *prototype* :

1. Mengidentifikasi Kebutuhan Pemakai

Pada tahap ini, analisis sistem akan melakukan studi kelayakan dan studi terhadap kebutuhan pemakai baik meliputi model *interface*, teknik prosedural maupun dalam teknologi yang akan digunakan.

2. Mengembangkan kebutuhan pemakai analisis sistem kerja sama dengan pemrograman mengembangkan *prototyping* sistem untuk memperlihatkan kepada klien pemodelan sistem yang akan dibangun.

3. Menentukan *Prototyping*

Analisis sistem pada tahap ini akan mengidentifikasi sejauh mana pemodelan yang dibuatkannya dapat diterima oleh pemesan atau bahkan harus merombak secara keseluruhan.

4. Penggunaan *Prototyping*

Pada tahap ini analisis sistem akan mengimplementasikan pemodelan yang dibuatnya menjadi suatu sistem.

2.2.12 Metode Pengujian Sistem

Pengujian sistem menyajikan anomali yang menarik bagi rekayasa perangkat lunak pada proses perangkat lunak, perancang berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

2.2.13 Pengujian Fungsional

Pengujian fungsional adalah teknik pengujian yang digunakan untuk menguji fitur atau fungsi dari sistem atau perangkat lunak, harus mencakup semua skenario termasuk jalur kegagalan dan kasus batas.