

BAB II

LANDASAN TEORI

2.1. Konsep Dasar

2.1.1. Sistem Informasi

Menurut Mulyanto dalam (Kuswara & Kusmana, 2017:18) “sistem informasi adalah suatu sistem yang terdiri dari kumpulan komponen sistem, yaitu software, hardware dan brainware yang memproses informasi menjadi sebuah output yang berguna untuk mencapai suatu tujuan tertentu dalam suatu organisasi”.

2.1.2. Sistem Informasi Pemasaran

Menurut (Sugesti, 2018:22) Sistem Informasi Pemasaran merupakan “sebuah sistem berbasis komputer yang bekerja sama dengan sistem informasi fungsional lain untuk mendukung manajemen bank dalam menyelesaikan masalah yang berhubungan dengan pemasaran produk”.

2.1.3. Website

Menurut (Bekti, 2015:35) menyimpulkan bahwa:

Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

1. Internet

Menurut (Priyono, 2019:136) mengemukakan bahwa:

Internet adalah kumpulan jaringan dari jaringan- jaringan komputer dunia yang terdiri dari jutaan unit- unit kecil, seperti jaringan pendidikan, jaringan bisnis, jaringan pemerintahan dan lain-lain, yang secara bersama menyediakan layanan informasi seperti e-mail, online chat, transfer file dan saling keterhubungan (linked) antara satu halaman web dengan sumber halaman web yang lainnya.

2. *E-Commerce*

Menurut (Ferdika & Kuswara, 2017:177) menyimpulkan bahwa “E-commerce adalah dimana dalam satu website menyediakan atau dapat melakukan Transaksi secara online atau juga bisa merupakan suatu cara berbelanja atau berdagang secara online atau direct selling yang memanfaatkan fasilitas Internet dimana terdapat website yang dapat menyediakan layanan *get and deliver*”.

3. **Web Server**

Menurut (Bekti, 2015:28) “Web Server merupakan software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML”.

4. **Web Browser**

Menurut Setyaji dalam (Kuswara & Kusmana, 2017:18) menyimpulkan bahwa “Web browser atau sering juga disebut internet browser yang berfungsi sebagai jembatan bagi pengguna komputer dalam menjelajah dunia maya”.

2.1.4. Bahasa Pemrograman

1. **HTML**

Menurut (Bekti, 2015:8) “HTML merupakan dasar untuk pembuatan desain web. File HTML berisi suatu instruksi tertentu yang dapat memberikan sebuah Format pada dokumen yang akan ditampilkan pada WWW (World Wide Web)”.

2. **PHP (*Personal Home Page*)**

Menurut (Bekti, 2015:8) “PHP (HyperText Preprocessor) adalah sebuah bahasa pemrograman yang umum digunakan untuk scripting Server-side”.

3. JQuery

Menurut (Bekti, 2015:59) “jQuery merupakan suatu librari JavaScript yang memungkinkan Anda untuk membuat program web pada suatu halaman web tanpa harus menambahkan *event* atau pun *property* pada halaman web tersebut”.

4. CSS

Menurut Kadir dalam (Supriyanta & Masturah, 2019:10) “Cascading Style Sheet (CSS) adalah skrip yang ditujukan secara khusus untuk mengatur tampilan aplikasi web”.

2.1.5. Basis Data

Menurut Winarno dan Utomo dalam (Prayitno & Safitri, 2015:2) “Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel-tabel yang saling berhubungan satu sama lain, dengan menggunakan *field*/kolom pada tiap tabel yang ada”.

1. MySQL

Menurut Arief dalam (Kuswara & Kusmana, 2017:18) “MySQL (My Structure Query Language) adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya”.

2. XAMPP

Menurut Arief dalam (Kuswara & Kusmana, 2017:18) “XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi”.

3. PhpMyAdmin

Menurut Rahman dalam (Masturah, 2019) “*PHP Myadmin* adalah sebuah *software* berbasis pemrograman PHP yang dipergunakan sebagai administrator MySQL melalui *browser (web)* yang digunakan untuk *management database*”.

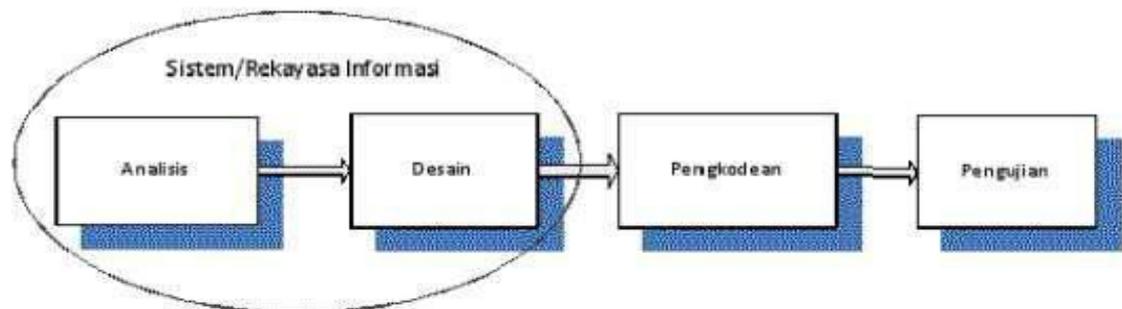
2.1.6. Text Editor

1. Sublime Text Editor

Menurut Putra dalam (Pradiatiningtyas & Suparwanto, 2017:4) “Sublime text adalah *text* editor berbasis *Python*, sebuah *text* editor yang elegan, kaya fitur, *cross platform*, mudah dan *simple* yang cukup terkenal dikalangan *developer* (pengembang) dan *desainer*”.

2.1.7. Model Pengembangan Perangkat Lunak

Menurut Sukanto dan Shalahuddin dalam (Susanto & Astuti, 2017:77) “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)”. Berikut adalah gambar model air terjun:



Sumber: Rosa dan Shalahuddin dalam (Susanto & Astuti, 2017:77)

Gambar II.1
Ilustrasi Model *Waterfall*

1. Analisa kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara *intensif* untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*), dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

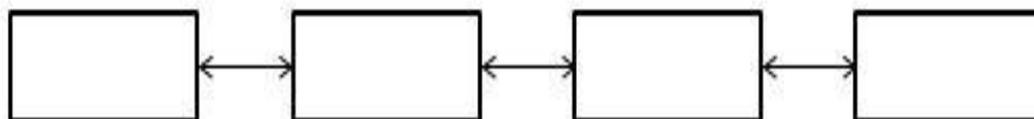
2.2. Peralatan Pendukung

2.2.1. Struktur Navigasi

(Nurdin & Darwati, 2017:75) mengemukakan “Struktur navigasi adalah susunan menu atau *hirarki* dari suatu situs yang menggambarkan isi dari setiap halaman dan *link* atau navigasi halaman pada suatu situs *web*”. Ada empat macam bentuk dasar navigasi, yaitu :

1. Struktur Navigasi Berurut (*Linier*)

Struktur navigasi *linier* hanya mempunyai satu rangkaian cerita yang berurut yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya, pengguna akan melakukan navigasi secara berurutan, dalam *frame* atau *byte* informasi satu ke yang lainnya.



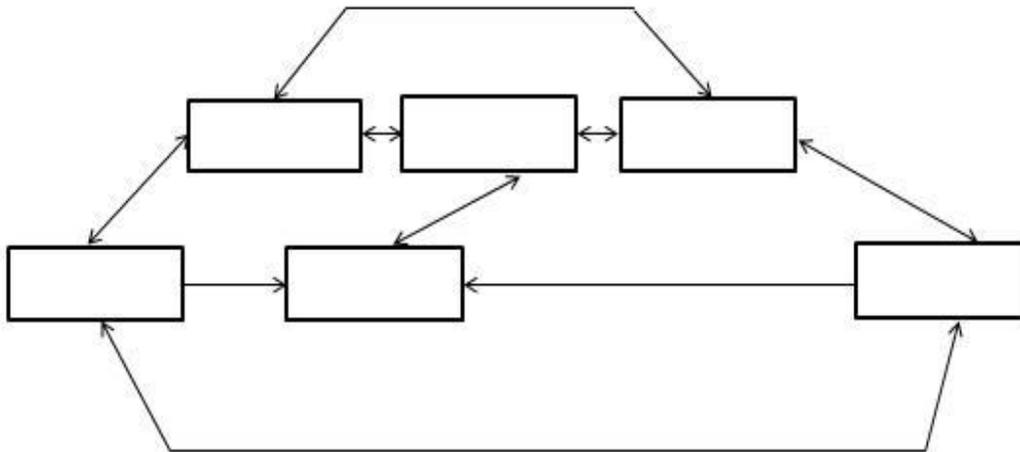
Sumber : (R, 2018:57)

Gambar II.2
Struktur Navigasi Berurut (*Linier*)

2. Struktur Navigasi Tidak Berurut (*Non Linear*)

Struktur navigasi *non-linier* merupakan pengembangan diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur *non-linier* ini berbeda dengan percabangan non-linier ini walupun terdapat percabangan tetap tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan

Slave Page, pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terkait dengan jalur yang sudah ditentukan sebelumnya.

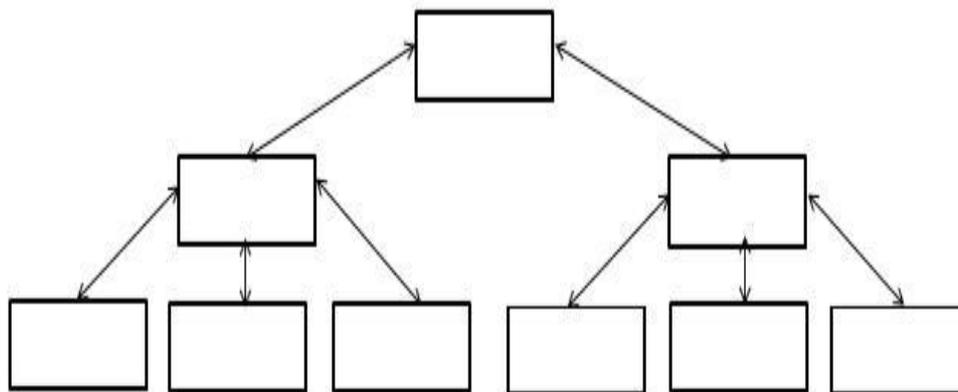


Sumber: (R, 2018:58)

Gambar II.3
Struktur Navigasi Tidak Berurut (*Non-Linier*)

3. Struktur Navigasi Hierarki (*Hierarchi*)

Struktur dasar ini disebut juga struktur *linier* dengan percabangan karena pengguna melakukan navigasi disepanjang cabang pohon struktur yang terbentuk oleh logika isi.

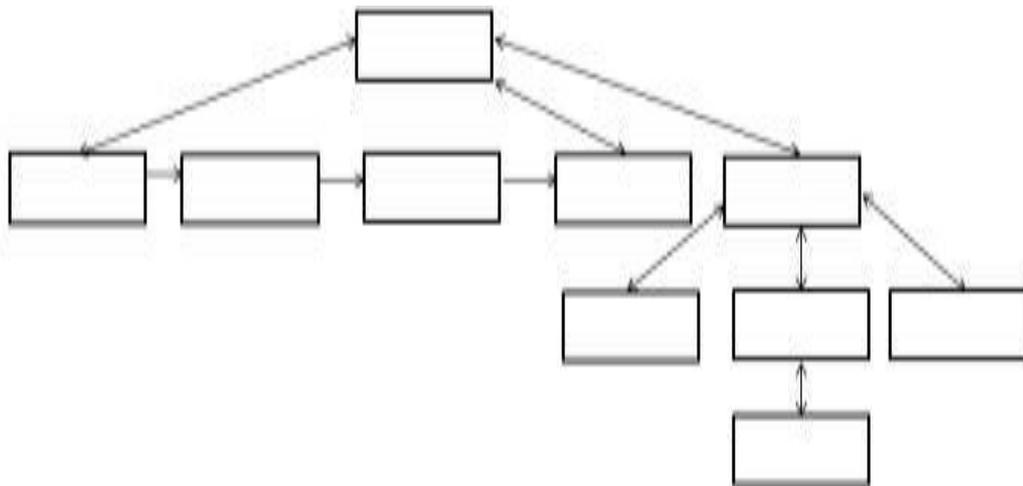


Sumber : (R, 2018:58)

Gambar II.4
Struktur Navigasi Hirarki (*Hierarchi*)

4. Struktur Navigasi Campuran (*Composite*)

Struktur navigasi jenis ini pengguna akan melakukan navigasi dengan bebas (secara *non-linier*). tetapi terkadang dibatasi presentasi *linier* film atau informasi penting dan pada data yang paling terorganisasi secara logis pada suatu hirarki.



Sumber: (R, 2018:58)

Gambar II.5
Struktur Navigasi Campuran (*Composite*)

2.2.2. UML (*Unified Modeling Language*)

Menurut Yasin dalam (Destiana & Fajrin, 2014:34) “Unified Modeling Language (UML) sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak”. Adapun beberapa jenis diagram pada UML yang dapat membantu perancangan sistem adalah sebagai berikut :

1. *Use Case Diagram*

Use Case Diagram merupakan suatu bentuk diagram yang digunakan menggambarkan fungsi-fungsi yang diharapkan dari sebuah sistem yang dirancang. Dalam use case diagram penekanannya adalah “apa” yang diperbuat oleh sistem, dan

bukan “bagaimana”. Sebuah use case akan merepresentasikan sebuah interaksi antara pelaku atau aktor dengan sistem.

2. *Activity Diagram*

Sebuah *Activity Diagram* menunjukkan suatu alur kegiatan secara berurutan dan digunakan untuk menggambarkan kegiatan-kegiatan dalam sebuah operasi. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* tidak menggambarkan sifat internal dari sebuah sistem dan interaksi antara beberapa sub sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

3. *Sequence Diagram*

Sequence Diagram merupakan diagram yang menggambarkan kolaborasi yang dinamis antara obyek satu dengan yang lain. Kolaborasi ini ditunjukkan dengan adanya interaksi antar obyek didalam dan disekitar sistem yang berupa pesan atau instruksi yang berurutan. *Sequence diagram* umumnya digunakan untuk menggambarkan suatu scenario atau urutan langkah-langkah yang dilakukan baik oleh aktor maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau *output*.

4. *Deployment Diagram*

Deployment Diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server dan hal-hal lain yang bersifat fisikal.

2.2.3. ERD (*Entity Relationship Diagram*)

Menurut Rosa dan Shalahuddin dalam (Imaniawan & Wati, 2017:4) “EntityRelationship Diagram (ERD) merupakan model yang banyak digunakan dalam pemodelan basis data, ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika”.

1. *Entity (Entitas)*

Entitas (*entity*) menunjukkan obyek-obyek dasar yang terkait di dalam sistem. Obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data. Untuk menggambarkan entitas dilakukan dengan mengikuti aturan-aturan sebagai berikut:

- a. Entitas dinyatakan dengan simbol persegi panjang.
- b. Nama entitas berupa kata benda tunggal.
- c. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan menyatakan maknanya dengan jelas.

2. *Attribute (Atribut)*

Atribut sering juga disebut sebagai properti (*property*), merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data. Atribut berfungsi sebagai penjelas sebuah entitas untuk menggambarkan atribut yang dilakukan dengan mengikut aturan sebagai berikut:

- a. Atribut dinyatakan dengan simbol *ellips*.
- b. Nama atribut dituliskan dalam simbol *ellips*.
- c. Nama atribut berupa kata benda tunggal.
- d. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan menyatakan maknanya dengan jelas.

- e. Atribut dihubungkan dengan entitas yang bersesuaian dengan menggunakan garis.

3. **Relation (Relasi)**

Relasi atau hubungan adalah kejadian atau transaksi yang terjadi di antara dua entitas yang keterangannya perlu disimpan dalam basis data. Aturan penggambaran relasi antara *entity* adalah:

- a. Relasi dinyatakan dengan simbol belah ketupat.
- b. Nama relasi dituliskan dalam simbol belah ketupat.
- c. Relasi menghubungkan dua entitas.
- d. Nama relasi menggunakan kata kerja aktif (diawali awalan me) tunggal.
- e. Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

4. **Derajat Relasi (Kardinalitas)**

Model relasi ini berdasarkan persepsi dunia nyata diantaranya himpunan objek dasar dan relasi antara entitas. Entitas dapat diartikan sebagai objek dan diidentifikasi secara unik, dan objeknya dapat berbentuk orang, barang, dan sebagainya. Kardinalitas relasi menunjukkan maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi di antara dua himpunan entitas (misalkan A dan B) dapat berupa satu ke satu (*one to one*), satu ke banyak (*one to many*), banyak ke satu (*many to one*) dan banyak ke banyak (*many to many*).

a. **Satu ke satu (*One to one*)**

Berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya

setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas himpunan entitas A.

b. Satu ke banyak (*One to many*)

Berarti setiap entitas pada himpunan entitas A dan dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

c. Banyak ke satu (*Many to one*)

Berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan B, tetapi tidak sebaliknya, di mana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

d. Banyak ke banyak (*Many to many*)

Berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

2.2.4. Pengujian Web

1. *Black Box Testing*

Menurut (Janti, 2017:107) “*Black box testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya”.

Pengujian black box juga merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode white box. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian

kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah.

