**BAB II  
LANDASAN TEORI**

1. **Aplikasi Penjualan *(E-commerce)***

Menurut (Pratama et al., 2018) dalam jurnal Apriandi dan Saputra dapat mendefinisikan e-commerce merupakan salah satu akses yang dimiliki dan dapat bertemunya antara penjual dan pembeli yang dapat melakukan proses transaksi atau jual beli dengan melalui media online atau internet. Fokus penjualan melalui *internet* yaitu konsumenkonsumen individual karena lebih efektif dan efisien.

Demikian penjelasan mengenai klasifikasi *e-commerce* berdasarkan pendapat (Pratama et al., 2018), antara lain:

1. *Business to Business* (B2B)

*E-Commerce* tipe ini meliput transaksi antar organisasi yang dilakukan di *electronic market*.

1. *Business to Consumers* (B2C)

*Business to Consumer* penjualan adalah suatu organisasi dan pembeli adalah individu.

1. *Consumer to Business* (C2B)

Dalam *Consumer to Business* memberitahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen.

1. *Costumer to Costumer* (C2C)

Sedangkan *Costumer to Costumer* konsumen menjual secara langsung ke konsumen lain atau mengiklankan jasa pribadi di *internet*.

1. *Nobusiness e-commerce*

Lembaga non bisnis seperti akademis, organisasi, organisasi keagamaan, organisasi sosial, dan lembaga pemerintahan yang menggunakan berbagai tipe *e-commerce* untuk mengurangi biaya guna meningkatkan operasi dan layanan publik.

1. *Intrabusiness e-commerce*

Semua aktivitas intern organisasi, biasanya dijalankan di *internet* yang melibatkan pertukaran barang, jasa atau informasi.

1. **Sistem**

Menurut Fauzi (2017:2) menyatakan bahwa “sistem adalah suatu kesatuan yang terdiri dari interaksi subsistem untuk mencapai tujuan yang sama”. Sedangkan menurut Hutahaean (2015:2) menyatakan bahwa “sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu”. Berdasarkan kutipan dari para ahli di atas, dapat disimpulkan bahwa sistem merupakan kumpulan dari suatu komponen, elemen, atau prosedur yang saling berhubungan, berinteraksi dan bekerja sama alam mencapai tujuan atau sasaran yang telah ditentukan sebelumnya.

1. Karakteristik Sistem

Sistem memiliki karakteristik atau ciri-ciri agar dikategorikan sebagai suatu sistem yang baik. Karakteristik dari sistem (Fauzi, 2017:2) diuraikan sebagai berikut:

1. Komponen sistem

Suatu sistem terjadi dikarenakan adanya sejumlah komponen yang melakukan interaksi. Suatu sistem yang sekecil apapun akan selalu mengandung komponen-komponen.

1. Batas sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

1. Lingkungan luar sistem

Lingkungan luar dari suatu sistem adalah daerah di luar batas dari suatu sistem yang mempengaruhi operasi sistem.

1. Penghubung sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumbersumber daya mengalir dari suatu subsistem ke subsistem yang lainnya. Keluaran dari sistem menjadi masukan untuk subsistem lainnya.

1. Masukan sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem.

1. Keluaran sistem

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisi pembuangan.

1. Pengolah sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan mengubah masukan mejadi keluaran.

1. Sasaran sistem Suatu sistem pasti mempunyai tujuan *(goal)* atau sasaran *(objective)*. Kalau suatu sistem tidak mempunyai sasaran maka sistem tersebut tidak berguna.
2. Klasifikasi Sistem

Sistem dapat diklasifikasikan dapat beberapa sudut pandang. Klasifikasi sistem (Hutahaean, 2015:6), terdiri dari:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem telogi. Sedangkan sistem fisik diartikan sebagai sistem yang nampak secara fisik sehingga setiap mahluk dapat melihatnya, misalnya sistem komputer.

1. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain. Sedangkan sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

1. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem probabilistik merupakan sistem yang kondisi masa depanya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

1. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih sepesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan beriteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia. Sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa danya campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benarbenar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

1. **Informasi**

Menurut Robert J. Verzello/John Reuter III dalam buku Fauzi (2017:10) menyatakan bahwa “informasi adalah kumpulan data yang relevan dan mempunyai arti yang menggambarkan suatu kejadian-kejadian atau kegiatankegiatan”. Sedangkan, menurut Hutahaean (2015:9) mengemukakan bahwa “informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Berdasarkan pendapat dari para ahli yang sudah diuraikan penulis diatas, maka dapat disimpulkan bahwa informasi merupakan sekumpulan data yang relevan dan mempunyai arti dan diolah menjadi bentuk yang lebih berguna bagi pemakainya. Relevansi informasi setiap pengguna informasi akan berbeda-beda tergantung dari kebutuhan penggunaan informasi. Adapun kriteria yang harus dimiliki oleh informasi (Fauzi, 2017:11), terdiri dari:

1. Akurat

Informasi yang didapatkan harus bebas dari kesalahan-kesalahan dan dapat menyesatkan. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan yang dapat mengubah atau merusak informasi tersebut.

1. Tepat Pada Waktunya

Informasi yang diterima pengguna informasi tidak boleh terlambat. Informasi yang diterima sudah tidak dapat digunakan, maka tidak akan mempunya nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Mengingat mahalnya nilai informasi maka diperlukan teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

1. Lengkap

Informasi itu harus lengkap sehingga tidak kehilangan aspek-aspek yang penting dari kejadian yang merupakan dasar aktivitas yang diukurnya.

1. Rangkuman

Informasi harus di *filter* agar sesuai dengan kebutuhan pemakai. Manajer tingkat lebih rendah cenderung memerlukan informasi yang sangat rinci. Semakin aliran informasi mengarah ke atas maka informasi akan semakin mengerucut.

1. Dapat diverifikasi

Informasi harus dapat diverifikasi jika diperoleh dari dua orang yang berbeda dari suatu sistem yang saling berinteraksi. Dimana hasil informasi dari kedua orang tersebut adalah sama.

1. **Sistem Informasi**

Menurut John F. Nash dan Martin B. Roberts dalam Fauzi (2017:18) berpendapat bahwa suatu sistem informasi adalah suatu kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan sesuatu dasar untuk pengambilan keputusan.

Sedangkan, menurut Hutahaean (2015:13), “sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”.

Dapat disimpulkan bahwa sistem informasi merupakan kumpulan dari orang, perangkat lunak, perangkat keras, dan prosedur yang saling berinteraksi, bekerja sama dalam menyelesaikan sesuatu untuk menghasilkan informasi yang dapat dijadikan sebagai dasar pengambilan keputusan. Komponen-komponen dari sistem disebut dengan blok bangunan *(building block)*. Penjelasan blok bangunan (Hutahaean, 2015:13) diuraikan sebagai berikut:

1. Blok masukkan *(Input Block)*

Blok masukan merupakan blok yang bertugas dalam input data agar masuk ke dalam sistem informasi. Blok masukan bertugas dalam merekam data yang akan dimasukkan, biasanya berupa dokumen-dokumen dasar.

1. Blok model *(Model Block)*

Blok model terbentuk dari kombinasi prosedur, logika dan model matematik yang memproses data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

1. Blok keluaran *(Output Block)*

Sistem informasi menghasilkan keluaran *(output)* yaitu informasi yang berkualitas dan berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

1. Blok teknologi *(Technology Block)*

Teknologi digunakan merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran berupa informasi dan membantu pengendalian dari sistem secara menyeluruh. Blok teknologi perangkat lunak *(software)* dan perangkat keras *(hardware)* yang dioperasikan oleh teknisi *(brainware)*.

1. Blok basis data *(Database Block)*

Basis data *(database)* merupakan media untuk menyimpan data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan dapat dipergunakan kembali, diperlukan perangkat lunak untuk memanipulasinya.

1. Blok kendali *(Control Block)*

Sistem informasi memiliki kontrol kendali untuk menanggulangi gangguan gangguan terhadap sistem apabila terlanjur terjadi kesalahan maka dapat langsung diatasi.

1. **Konsep Dasar *Web***

Konsep dasar *web* meliputi dari berbagai aspek dan sudut pandang yang berbeda-beda sesuai dengan keterangan dalam hal-hal yang berkaitan dengan *web*.

1. *Internet*

Menurut Hidayatullah dan Kawistara (2015:1) “*Internet* adalah jaringan global yang menghubungkan komputer-komputer di seluruh dunia”.

1. *Website*

Menurut Pohan 2013:1 dalam jurnal Tina (2017) dapat mengemukakan bahwa “*World wide web* atau yang biasa disingkat www merupakan kumpulan situs *web* yang dapat diakses di *internet* dimana di dalamnya berisikan semua informasi yang dibutuhkan *internet*”. Pada awalnya, *web* merupakan ruang informasi dalam *internet*, dengan menggunakan teknologi *hypertext*, pemakai dituntut untuk menemukan informasi dengan mengikuti *link* yang disediakan didalam dokumen *web* yang ditampilkan dalam *browser web*.

1. Aplikasi Berbasis *Website*

Aplikasi berbasis *web* terbagi menjadi *web server* dan *web browser*.

1. *Web server*

Menurut Sibero dalam Fandi, dkk (2017) “*Web Server* adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak”. Secara bentuk fisik perangkat keras *web server* tidak berbeda dengan komputer. yang membedakan adalah kapasitas dan kapabilitas.

1. *Web browser*

*Web browser* menampilkan halaman halaman *web* dengan cara membaca instruksi-instruksi yang biasa disebut dengan *tag-tag* HTML (Hidayatullah dan Kawistara, 2015:4).

1. **Bahasa Pemrograman**

Untuk membuat *website* yang dinamis dan mudah di *upgrade* setiap saat dari *browser*, dibutuhkan sebuah program yang mampu mengolah data dari komputer *client* atau dari komputer *server* itu sendiri sehingga mudah dan nyaman untuk disajikan di *browser*, untuk melakukan semua itu bisa menggunakan *HyperText Markup Language* (HTML), pemrograman *Hypertext Preprocessor* (PHP, *Cascading Style Sheet* (CSS), *JavaScript* dan *Jquery*.

1. Bahasa *Markup* HTML (*Hyper Text Markup Language*)

Menurut Hidayatullah dan Kawistara (2015:13) menyimpulkan bahwa: *Hyper Text Markup Languange (HTML)* adalah bahasa standar yang digunakan untuk menampilkan halaman *web* yang bisa dilakukan dengan HTML yaitu mengatur tampilan dari halaman *web* dan isinya, membuat tabel dalam halaman *web*, mempublikasikan halaman *web* secara *outline*, membuat *form* yang bisa digunakan untuk menangani *registrasi* dan transaksi melalui *web*, menambahkan objek-objek seperti citra, audio, video, animasi, java aplet dalam web, serta menampilkan area gambar *(canvas)* di *browser*.

1. Bahasa Pemrograman PHP *(Hypertext Processor)*

Menurut Hidayatullah dan Kawistara (2015:231) menyimpulkan bahwa: PHP singkatan dari *Perl Hypertext Proccesor* yaitu bahasa pemprograman *web server-side* yang bersifat *open source.* PHP merupakan *script* yang berintegrasi dengan HTML dan berada pada *server* (*server side* HTML *embeded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *web* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client.* Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *update*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

1. Bahasa CSS *(Cascading Style Sheets)*

Menurut Sibero dalam Fandi, dkk (2017) “*CSS* memiliki arti gaya menata halamn bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak yang telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya”.

1. *JavaScript*

Menurut Abdulloh dalam Wijianto,dkk (2018) “memaparkan tentang *JavaScript* dalam membuat website adalah memberikan efek animasi yang menarik dan interaktif dalam penanganan *event* yang dilakukan oleh pengguna *website*”.

1. *Jquery*

Menurut Abdulloh dalam Wijianto,dkk (2018) “menjelaskan jika *jquery* merupakan salah satu *javascript library,* yaitu kumpulan fungsi *javascript* siap pakai*,* sehingga mempermudah dan mempercepat kita dalam membuat kode *javascript”.*

1. **Basis Data**

Suatu kumpulan data yang saling terhubung antara satu dengan yang lainnya sehingga membentuk suatu basis data. Dimana sekumpulan *table* tersebut saling terhubung satu dengan yang lainnya.

1. Basis Data

Menurut Priyadi (2014:2) “basis data adalah sekumpulan representasi tabel yang saling berhubungan dalam media penyimpanan *digital”*.

1. Aplikasi Basis Data
2. PhpMyadmin

PhpMyAdmin adalah *tool open source* yang ditulis dalam bahasa PHP untuk menangani administrasi MySQL berbasis *World Wide Web*. (Hidayatullah dan Kawistara, 2015:184).

1. MySQL

Suatu aplikasi komputer yang dapat merujuk pada konsep *Relational Database Management System* (RDBMS), terdapat struktur bahasa yang sudah standar untuk membangun basis datanya adalah *Structured Query Language* (Priyadi, 2014:82). Untuk lebih fokus pada proses pembangunan aplikasi basis data tersebut, perintah SQL dibagi menjadi tiga jenis (Priyadi, 2014:82), yaitu:

1. DDL (*Data Definition Language*)

*Data Definition Language* (DDL) merupakan perintah SQL, yang digunakan untuk melakukan definisi awal suatu basis data dan tabel pada konsep RDBMS. Penulisan perintah SQL pada kelompok ini terdiri dari *create*, *alter* dan *drop*. Berikut pembahasan untuk perintah SQL tersebut (Priyadi, 2014:83):

1. *Create*

Perintah SQL ini digunakan untuk membuat suatu basis data dan tabel pendukung didalam pembangunan basis data tersebut.

1. *Alter*

Perintah SQL ini digunakan untuk mengubah struktur tabel yang terdapat didalam basis data.

1. *Drop*

Perintah SQL ini digunakan untuk menghapus tabel yang terdapat didalam basis data.

1. DML (*Data Manipulation Language*)

*Data Manipulation Language* (DML) merupakan perintah SQL yang digunakan untuk melakukan pengolahan *record* atau manipulasi *record* pada tabel dalam suatu basis data Berikut ini adalah deskripsi mengenai kelompok perintah DML tersebut (Priyadi, 2014:90):

1. *Insert*

Perintah SQL ini, digunakan untuk melakukan *entry* atau penambahan suatu *record* pada tabel dalam basis data.

1. *Select*

Perintah ini SQL ini digunakan untuk memilih *record* yang akan di tampilkan berdasarkan data pada tabel dalam basis data.

1. *Update*

Perintah SQL untuk mengubah data dalam suatu tabel pada *field* tertentu, dengan *record* baru berdasarkan suatu *field* sebagai kriteria pengubahan *record*-nya.

1. *Delete*

Perintah SQL ini digunakan untuk menghapus data dalam suatu tabel, berdasarkan suatu *field* sebagai kriteria penghapusan *record*-nya.

1. DCL (*Data Control Language*)

*Data Control Language* (DCL) merupakan perintah SQL yang digunakan untuk melakukan pengaturan hak akses suatu onjek data para pengguna dalam basis data (Priyadi, 2014:94). Penulisan perintah SQL pada kelompok ini terdiri dari :

1. *Grant*

Perintah SQL ini digunakan oleh seorang *administrator* basis data untuk memberikan hak aksesnya kepada pengguna tertentu agar dapat mengakses suatu tabel dalam basis data.

1. *Revoke*

Perintah SQL ini digunakan oleh seorang *administrator* basis data, untuk membatalkan/menghentikan hak akses yang telah diberikan kepada pengguna tertentu, agar tidak dapat mengakses tabel dalam basis data.

1. **Metode Pengembangan Perangkat Lunak**

Model untuk pengembangan perangkat lunak dilakukan oleh seorang *programmer* dalam rangka meningkatkan untuk meningkatkan kualitas program yang sudah ada. Menurut Sukamto dan Shalahuddin (2014:28) mengatakan bahwa Model SDLC air terjun (*waterfall*) sering juga disebut “model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Untuk lebih jelasnya metode *waterfall* dapat digambarkan sebagai berikut

Analisis

Desain

Pengodean

Pengujian

Gambar 2.1. Model *Waterfall*

Sumber: Sukamto dan Shalahuddin ( 2014:28)

1. Analisa Kebutuhan Perangkat Lunak

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

1. 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 pengkodean. 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 didokumentasikan.

1. Pembuatan Kode Program

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

1. Pengujian

Pengujian fokus pada perangkat lunak 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.

1. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisaterjadi 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.

1. **Struktur Navigasi**

Menurut ASFA dalam Fandi, dkk (2017) “Struktur Navigasi dapat diartikan sebagai alur dari suatu program yang menggambarkan rancangan hubungan antara area yang berbeda sehingga memudahkan proses pengorganisasian seluruh elemen-elemen *website*”.

Menurut Widayati (2014) ada beberapa cara yang digunakan dalam mendesain aliran aplikasi multimedia diantaranya adalah sebagai berikut:

1. Struktur Navigasi *Linier*

Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang ditampilkan pada sruktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

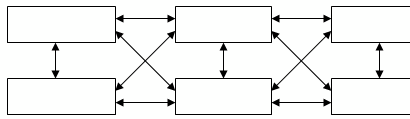


Gambar 2.2. Struktur Navigasi *Linier*

Sumber: Widayati (2014)

1. Struktur Navigasi *Non-Linier*

Struktur navigasi *non-linier* atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur *non linier* ini berbeda dengan percabangan pada struktur *hirarki*, karena pada percabangan *nonlinier* ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *master page* dan *slave page*.

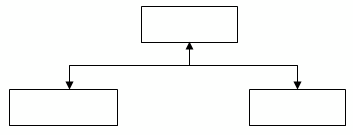


Gambar 2.3. Struktur Navigasi *Non-Linier*

Sumber: Widayati (2014)

1. Struktur Navigasi *Hirarki*

Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *master page* (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut *slave page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *master page* (halaman utama kedua), dan seterusnya. Pada struktur navigasi ini tidak diperkenankan adanya tampilan secara linier.

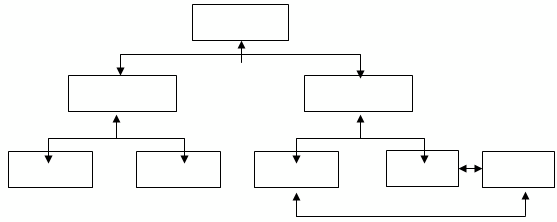


Gambar 2.4. Struktur Navigasi Hirarki

Sumber: Widayati (2014)

1. Struktur Navigasi Campuran

Struktur navigasi campuran merupakan gabungan dari ketiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan *website* karena struktur ini dapat digunakan dalam pembuatan *website* sehingga dapat memberikan keinteraksian yang lebih tinggi.



Gambar 2.5. Struktur Navigasi Campuran

Sumber: Widayati (2014)

1. ***Entity Relationship Diagram***

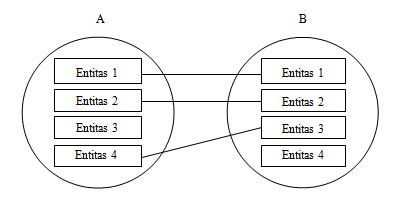
Menurut Sarita, dkk (2017) “*Entity Relationship Diagram* adalah alat pemodelan data utama dan akan membantu mengorganisasikan data dalam suatu proyek ke dalam entitas-entitas dan hubungan antar entitas”. ERD (*Entity Relationship Diagram*) digunakan untuk pemodelan basis data dan relasional model jaringan (Sukamto dan Shalahuddin, 2014:50).

1. Derajat Relasi (*Relationsip Degree*)

Menurut Priyadi (2014:5) mengemukakan bahwa derajat *relationship* atau kardinalitas adalaha” batasan dari banyaknya hubungan yang dapat dilakukan oleh suatu himpunan”. Adapun beberapa macam Kardinalitas atau Derajat *Relationship* menurut Fathansyah (2015:79) adalah sebagai berikut:

1. Satu ke satu (*one two one*)

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

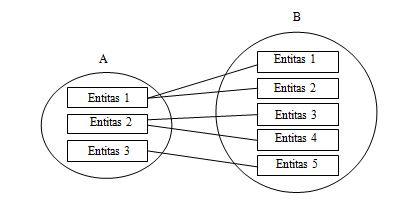


Gambar 2.6. Kardinalitas Relasi *One To One*

Sumber : Fathansyah (2015:79)

1. Satu ke banyak (*one to many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. Namun satu *record* pada tabel B hanya boleh berelasi dengan satu satu *record* saja pada tabel.

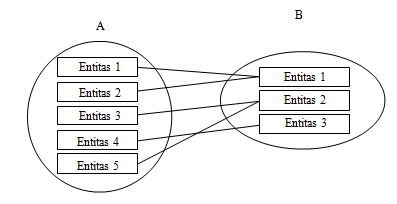


**Gambar 2.7**. **Kardinalitas Relasi *One To Many***

Sumber : Fathansyah (2015:80)

1. Banyak ke satu (*many to one*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan bengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi sebaliknya himpunan entitas B dapat banyak berhubungan dengan himpunan entitas A.

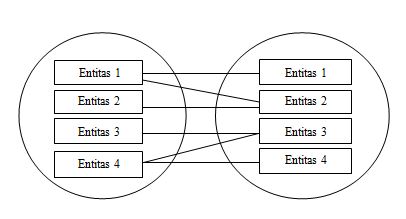


Gambar 2.8. Kardinalitas Relasi *Many To One*

Sumber : Fathansyah (2015:80)

1. Banyak ke banyak (*many to many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebalinya himpunan entitas B dapat berhubungan dengan banyak himpunan entitas A.



Gambar 2.9. Kardinalitas Relasi *Many To Many*

Sumber: Fathansyah (2015:81)

1. Simbol-Simbol ERD *(Entity Relationship Diagram)*

Menurut Husda, N. E dan Wangdra, Y (2016:112), menjelaskan bahwa ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi didalam dunia nyata. Diagram hubungan entitas digunakan untuk mengkonstruksikan model data konseptual, memodelkan struktur data dan hubungan antar data dan mengimplementasikan basis data secara logika maupun secara fisik dengan DBMS (*Database Management System*).

Tabel 2.1. Simbol-Simbol ERD (*Entity Relationship Diagram*)

Sumber: Husda, N. E dan Wangdra, Y (2016:113)

|  |  |
| --- | --- |
| **Simbol** | **Nama** |
|  | Entitas, adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai. |
|  | Relasi, menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda. |
|  | Atribut, berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai *key* diberi garis bawah). |
|  | Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut. |

1. ***Logical Record Structured***

Menurut Simarmata dan Paryudi dalam Fridayanthie, W. E dan Mahdiati, T (2016:7), “*logical record structured* (LRS) adalah representasi dari struktur *record- record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas”. Berikut adalah cara membentuk skema *database* atau *logical record structured* berdasarkan *entity relationship diagram*:

1. Jika relasinya satu-ke-satu, maka *foreign key* diletakkan pada salah satu dari dua entitas yang ada atau menyatukan kedua entitas tersebut.
2. Jika relasinya satu-ke-banyak, maka *foreign key* diletakkan pada entitas *many*.
3. Jika relasinya banyak-ke-banyak, maka dibuat *file* konektor yang berisi dua *foreign key* yang berasal dari kedua entitas.
4. ***Unified Modelling Language***

Menurut Sukamto, R. A dan Shalahuddin, M (2016:133), “UML *(Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement,* membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML mempunyai beberapa atau sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Diagram tersebut akan menggambarkan atau mendokumentasikan beberapa aspek dari sebuah sistem. Abstraksi konsep dasar UML terdiri dari *structure diagram, behavior diagram*, dan *interaction diagram*. Beberapa diagram UML menurut Sukamto, R. A dan Shalahuddin, M (2016:141), adalah sebagai berikut:

1. Definisi *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. *Activity Diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2. Simbol-Simbol *Activity Diagram*

Sumber: Sukamto, R. A dan Shalahuddin, M (2016:162)

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Status Awal  . | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan/*Decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan/*Join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

1. Definisi *Use Case Diagram*

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) 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 didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.3. Simbol-Simbol *Use Case Diagram*

Sumber: Sukamto, R. A dan Shalahuddin, M (2016:156)

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| *Use Case*  .  nama *use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal *frase* nama *use case*. |
| Aktor/*actor* | Orang, proses, atau sistem lain yang  berinteraksi dengan sistem informasi  yang akan dibuat diluar 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. |
| Asosiasi/*association* | Komunikasi antara aktor dan *use case*  yang berpartisipasi pada use case atau  use case memiliki interaksi dengan aktor. |
| Ekstensi/*Extend*  <<*Extend*>> | Relasi *use case* tambahan ke sebuah  *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan. |
| Generalisasi/*generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi lebih umum dari lainnya, misalnya:  111  arah panah mengarah pada *use case* yang menjadi generalisasinya (umum) |
| Menggunakan / *include /uses*  <<*include*>>  <<*uses*>> | Relasi *use case* tambahan ke sebuah  *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini.  Ada dua sudut pandang yang cukup  besar mengenai *include* di *use case*:   1. Pertama *include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut:222 2. *Include* berarti *use case* yang tambahan akan selalu melakukan   pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:  333  Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan. |

1. Definisi *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Susunan struktur kelas yang baik pada diagram kelas memiliki jenis-jenis kelas sebagai berikut:

1. Kelas *main*
2. Kelas yang menangani tampilan sistem *(view)*
3. Kelas yang diambil dari pendefinisian use case *(controller)*
4. Kelas yang diambil dari pendefinisian data *(model)*

Tabel 2.4. Simbol-Simbol *Class Diagram*

Sumber : Sukamto, R. A dan Shalahuddin, M (2016:146)

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Kelas  444 | Kelas pada struktur sistem. |
| Antarmuka/*Interface*  nama\_interface | Sama dengan konsep interface dalam pemrograman berorientasi objek. |
| Asosiasi/*association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity.* |
| Asosiasi/*association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.* |
| Generaliasi | Relasi antar kelas dengan makna generalisasi-spesialisasi. |
| Kebergantungan/*depedency* | Relasi antar kelas dengan makna kebergantungan antar kelas. |
| Agregasi/*aggregation* | Relasi antar kelas dengan makna semua-bagian *(whole-part).* |

1. Definisi *Sequence Diagram*

Menurut Hendini, A (2016:110), “sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek”.

Tabel 2.5. Simbol-Simbol *Sequence Diagram*

Sumber : Hendini, A (2016:110)

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Aktor/*actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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. |
| *Entity Class* | Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |
| *Boundary Class* | Kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak |
| *Control Class* | Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
| *Message* | Simbol mengirim pesan antar *class*. |
| *Recursive*  *124* | Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri. |
| *Activation* | Mewakili sebuah eksekusi operasi dari  objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi. |
| Garis hidup/*lifeline*  123 | *Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat *activation.* |

1. Definisi *Statechart Diagram*

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

1. **Survei dan Penelitian Terdahulu**

Adapun survei yang dilakukan oleh penulis untuk mempermudah berjalannya survei dan penelitian yang dilaksanakan, karena penelitian ini dilaksanakan sebagai dasar. Kemudian, ada survei dan penelitian yang telah dilaksanakan sebelumnya dan dapat digunakan oleh peneliti sebagai dasar penelitian, sebagai berikut:

1. Berdasarkan jurnal penelitian dengan judul Sistem Penjualan Berbasis *E-Commerce* Menggunakan Metode Objek *Oriented* pada Distro Dlapak *Street Wear* yang dilakukan oleh Susandi dan Sukisno (2017) yang membahas mengenai pemasaran Distro Dlapak Street Wear yang tidak maksimum, tidak ada media yang membantu memudahkan pelanggan dari luar kota untuk transaksi, dan tidak ada yang dapat membantu dalam memasarkan produk perangkat komputer untuk memasarkan produknya setiap hari maka dikembangkan suatu aplikasi penjualan online pada Distro Dlapak Street Wear. Penjualan *online* atau *e-commerce* adalah suatu aplikasi dan proses bisnis yang menghubungkan toko dan konsumen melalui transaksi elektronik dan dapat membantu pihak toko dalam pemasaran produk yang maksimal. Maka diwujudkan sebuah aplikasi dengan melakukan penelitian yang bertujuan membantu Distro Dlapak *Street Wear* dalam memperluas cakupan promosi, membantu pelanggan yang berasal dari luar kota agar tetap dapat berbelanja dan memaksimalkan dalam pemasaran produk. Penelitian pengembangan sistem yang dilakukan merupakan jenis Penelitian Terapan *(Applied Research)*. Dalam penelitian ini akan menerapkan teori analisis sistem dengan pendekatan berorientasi obyek. Pengujian dilakukan dengan metode ISO 9126. Hasil penelitian dapat langsung diterapkan untuk memecahkan permasalahan pada Distro Dlapak *Street Wear*.
2. Berdasarkan jurnal penelitian dengan judul Sistem Penjualan Berbasis *Web* *(E-Commerce)* Pada Tata Distro Kabupaten Pacitan yang dilakukan oleh Rulia Puji Hastanti, Indah Uly Wardati dan Bambang Eka Purnama (2015) yang membahas mengenai pembuatan sebuah media promosi dan penjualan *online* berbasis *website*, perancangan dan pembuatan ini dimaksudkan untuk memudahkan pengelolaan, penjualan dan promosi, juga mempermudah pembeli. Agar konsumen lebih mudah memilih model–model fashion, konsumen tanpa harus datang ke toko. Mempromosikan suatu produk harus disertai,dengan konsep yang mudah dan menarik agar diminati oleh konsumen dan masyarakat. Dengan dilakukan penelitian ini diharapkan mampu menciptakan sarana pemasaran yang lebih efektif daripada sistem sebelumnya pada Tata Distro kelurahan Ploso kabupaten Pacitan sehingga meningkatkan omset penjualan pada Tata Distro.
3. Berdasarkan jurnal penelitian dengan judul Implementasi E-Commerce Dalam Meningkatkan Penjualan Kue Di Toko Kemuning yang dilakukan oleh Yati Nurhayati (2017) membahas mengenai proses bisnis yang terjadi di dalam Toko Kemuning. Toko Kemuning adalah salah satu *home industy* dimana mekanisme penjualannya masih dilakukan secara tradisional (pelanggan membeli produk di toko). Mekanisme penjualan tradisional mengakibatkan jangkauan pemasaran yang sempit. Oleh karena itu diperlukan suatu sistem yang dapat mengelola transaksi penjualan secara elektronik. Sistem yang digunakan dalam penelitian ini adalah *e-commerce*. Dimana, sistem akan dianalisa terlebih dahulu untuk mengetahui kebutuhan masing-masing pengguna. Kemudian sistem dirancang menggunakan *Data Flow Diagram* (DFD). Hasil perancangan, diimplementasikan ke dalam Bahasa Pemrograman PHP dan MySQL. Dan hasil dari pengujian adalah sistem *e-commerce* telah berhasil dirancang dan diimplementasikan sehingga dapat meningkatkan penjualan produk di Toko Kemuning.
4. Berdasarkan jurnal penelitian dengan judul Perancangan Sistem Informasi Penjualan Berbasis Web Dalam Memasarkan Mobil Bekas yang dilakukan oleh Sandy Kosasi (2015), penjualan yang dilaksanakan merupakan penjualan melalui website untuk meningkatkan penjualan mobil dan penjualan mobil yang berada pada showroom (offline) menggalami penurunan, maka peneliti yang dilaksanakan oleh Sandy Kosasi dengan penjualan online dapat meninggkan penjualan dan melalui pendekatan ICDM (Internet Commerce Development Management).
5. Berdasarkan Jurnal pada Aplikasi Penjualan Berbasis Web Pada Toko Velg Q yang dilaksanakan oleh Joseph Bernadi merupakan penjualan yang dilaksanakan menggunakan Bahasa pemrograman PHP dan MYSQL untuk penyelesaian dalam pembuatan program dan sistem yang dilaksanakan untuk mempermudah dalam berjalannya sistem penjualan menggunakan metode fast methodology.