

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

A. Model Pembelajaran Berbasis *Web*

“Pembelajaran berbasis *web* dapat didefinisikan sebagai pembelajaran yang pelaksanaannya didukung oleh bantuan teknologi *internet*” (Batubara, 2018). Kelebihan pembelajaran berbasis *web* yaitu “dapat menyediakan mesin pencari (contoh: *google chrome, mozilla firefox, opera browser*) yang dapat digunakan untuk mencari informasi yang mereka butuhkan” (Kuswanto & Radiansah, 2018). Sedangkan fungsi pembelajaran berbasis *web* yaitu “dapat mengatasi hambatan waktu, tempat, dan jarak” (Ahmad Khoirun Nasir, 2019).

B. Sistem

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan yang berinteraksi untuk mencapai suatu tujuan, sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar. Sedangkan informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat (Taufik, 2019).

Anggraini mengemukakan bahwa sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi

dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan (Taufik, 2019).

C. *Website/Program*

Internet adalah sebuah jaringan *world wide*, bentuk jaringan bebas menggunakan jaringan *line* telepon, dari jutaan pengguna diseluruh dunia yang membuat koneksi melalui modem dapat berkomunikasi satu sama lain (Irawan et al., 2019).

Abdulloh mengemukakan bahwa *website* dapat didefinisikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan dari semua yang disajikan melalui jalur koneksi *internet* sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia (Sintawati, 2018).

Web Browser dapat didefinisikan sebagai aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*. Sumber informasi *web* diidentifikasi dengan *Uniform Resource Identifier (URL)* yang dapat terdiri dari halaman *web*, video, gambar, ataupun konten lainnya (Pakpahan & Halawa, 2020).

Yasin mengemukakan bahwa *web server* adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui *browser* kemudian memberikan tanggapan permintaan dalam bentuk halaman situs *web* atau lebih umumnya dalam dokumen *HTML*. Jika merujuk pada *hardware*, *web server* digunakan untuk menyimpan semua data seperti *HTML*, dokumen, gambar, *file CSS stylesheets*, dan *file javascript*. Sedangkan pada sisi *software*, fungsi *web server* adalah sebagai

pusat kontrol untuk memproses permintaan yang diterima dari *browser*. Jadi sebenarnya semua yang berhubungan dengan *website* biasanya juga berhubungan dengan *web server*, karena tugas *web server* adalah mengatur semua komunikasi yang terjadi antara *browser* dengan *server* untuk memproses sebuah *website* (Hayati, 2021).

D. Basis Data

Basis data terdiri atas dua kata, yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sehingga dapat didefinisikan basis data adalah kumpulan file / tabel / arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik (Pakpahan & Halawa, 2020).

E. Model Pengembangan Perangkat Lunak

RAD (Rapid Application Development) adalah strategi pengembangan sistem ekstensif yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang dalam konstruksi, cepat, berulang dan bertambah serangkaian prototype-prototypenya yang bekerja dalam sebuah sistem yang pada akhirnya berkembang kedalam sistem final (Sintawati, 2018). Metode *RAD* tersebut memiliki tahapan yang dilakukan sebagai berikut:



Gambar II.1 Siklus *RAD*, (Hamzah et al., 2019)

1. Analisis Kebutuhan

Disini peneliti melakukan pengumpulan data secara lengkap kemudian melakukan analisis dan didefinisikan kebutuhan-kebutuhan yang akan digunakan untuk perancangan sistem yang baru. Hal ini merupakan bagian yang sangat penting karena sistem yang akan dibuat akan berinteraksi langsung dengan elemen-elemen penting dalam komputer seperti *hardware* dan *software*.

2. *Design*

Dalam tahap ini peneliti menggambarkan desain dari sistem yang akan dibangun sesuai dengan rencana kebutuhan yang dilakukan pada tahap sebelumnya. Dalam memodelkan sistem, peneliti menggunakan konsep *procedural*. Perancangan sistem dilakukan pembuatan perancangan sistem seperti membuat *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Database*, dan *Interface* berdasarkan analisa permasalahan dalam sistem informasi penjualan produk pada PT Sinar Rasa Kencana.

3. Implementasi / *Implementation*

Merupakan tahap akhir dari metode *RAD* yaitu implementasi, disini peneliti mengembangkan sebuah sistem informasi penjualan dari tahapan-tahapan sebelumnya menjadi suatu program atau sistem. Setelah sistem selesai secara keseluruhan, maka dilakukan proses pengujian terhadap program tersebut apakah terdapat kesalahan

(*error*) atau tidak sebelum diaplikasikan. Jika proses tersebut telah dilakukan maka akan dihasilkan sistem yang lengkap sesuai dengan desain awal.

2.2 Teori Pendukung

Teori pendukung atau peralatan pendukung merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem, dimana simbol, lambang dan diagram menunjukkan secara tepat arti fisiknya (Taufik, 2019).

A. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram adalah sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut entitas dan hubungan antara entitas-entitas tersebut digambarkan dalam suatu model atau diagram (Taufik, 2019). Contoh *ERD* dapat dilihat pada gambar II.2.

“*ERD* memiliki beberapa komponen yang terdiri sebagai berikut” (Palasara et al., 2020):

a. Entitas

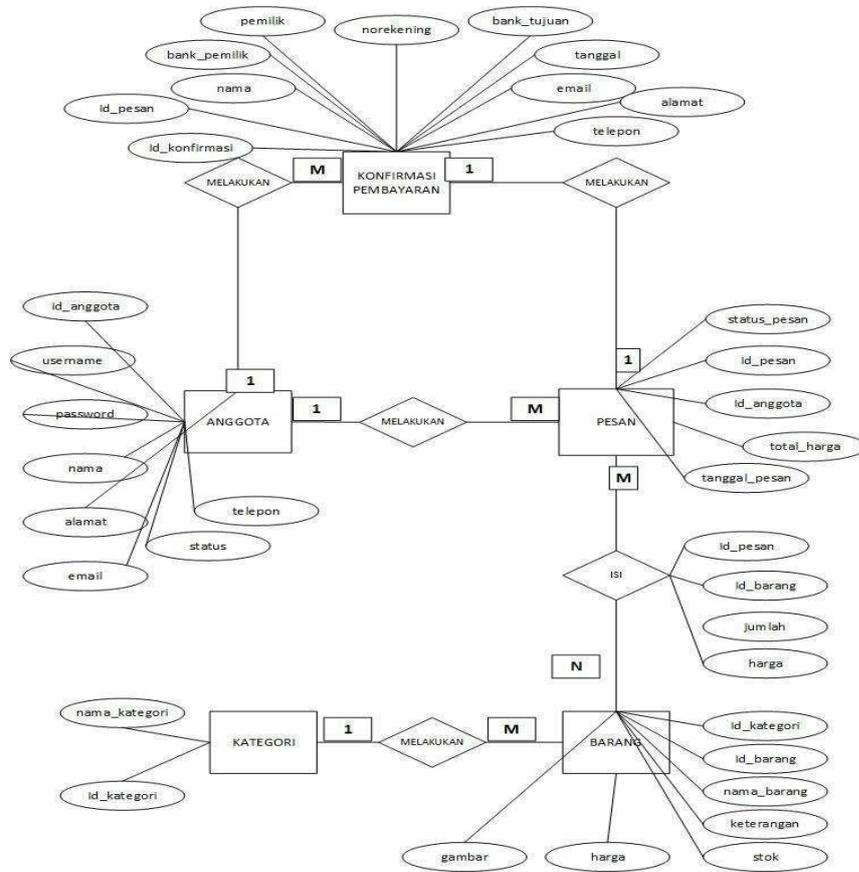
Entitas adalah objek yang dapat dibedakan dengan yang lain dalam dunia nyata.

b. Atribut

Atribut adalah karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut.

c. *Relationship*

Relationship adalah hubungan yang terjadi antara satu atau lebih *entity* / entitas.



Gambar II.2 Contoh ERD, (Susandi & Sukisno, 2017)

B. Logical Record Structure (LRS)

Logical Record Structure (LRS) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas, menentukan kardinalitas, jumlah tabel, dan *foreign key* (Lestari dkk,2018).

“Nugraha dan Octavia mengemukakan bahwa *Logical Record Structure (LRS)* merupakan struktur *record* pada tabel yang terbentuk dari hasil antara himpunan entitas, memiliki aturan pokok yang sangat dipengaruhi oleh elemen yang menjadi titik perhatian utama” (Taufik, 2019).

Sedangkan menurut Kusri mengemukakan bahwa *LRS* (*Logical Record Structure*) merupakan representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas pada diagram E-R (Taufik, 2019). Contoh *LRS* dapat dilihat pada gambar II.3.



Gambar II.3 Contoh *Logical Record Structure*, (Taufik, 2019)

C. *Unified Modelling Language (UML)*

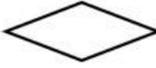
Definisi *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah tujuan umum, perkembangan, bahasa pemodelan dibidang rekayasa perangkat lunak yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem (Taufik, 2019).

“Sukanto dan Shalahudin mengemukakan bahwa UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks pendukung” (Taufik, 2019).

Activity Diagram

“Purwati dan Hasan mengemukakan bahwa *usecase* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *usecase* atau interaksi” (Dewi & Suminten, 2019).

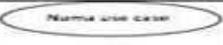
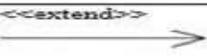
Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar II.4 Contoh Simbol *Activity Diagram*, (Taufik, 2019)

Use Case Diagram

Use Case adalah abstraksi dari interaksi antara sistem dan actor. *Use Case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat dimata

user. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis *client* (Dewi & Suminten, 2019).

Nama Simbol	Simbol	Deskripsi
<i>Use case</i>		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 <i>use case</i>
Aktor / <i>actor</i>	 nama aktor	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 diawali frase nama aktor
Asosiasi / <i>association</i>		Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i>		<i>Case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Arah panah mengarah pada <i>use case</i> yang ditambahkan
Generalisasi / <i>generalization</i>		Hubungan generalisasi dengan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
Menggunakan <i>include / uses</i>		Fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i>

Gambar II.5 Contoh Simbol *Use Case*, (Kuswanto & Radiansah, 2018)

Class Diagram

Class adalah deksripsi kelompok obyek-obyek dengan properti, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*, *class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem (Dewi & Suminten, 2019).

SIMBOL CLASS DIAGRAM			
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	<u>Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor</u>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	<u>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri</u>
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar II.6 Contoh Simbol *Class Diagram*, (Dewi & Suminten, 2019)

Sequence Diagram

Sequence Diagram adalah gambaran perilaku pada sebuah *scenario*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar *object* juga interaksi antar *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem (Dewi & Suminten, 2019).

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

Gambar II.7 Contoh Simbol *Sequence Diagram*, (Dewi & Suminten, 2019)

D. *HTML (Hyper Teks Markup Language)*

HTML atau (*Hyper Teks Markup Language*) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Dokumen ini dikenal dengan sebagai *web page*, dokumen *HTML* merupakan dokumen yang disajikan pada *web browser* (Sintawati, 2018).

E. *Page Hyperteks Preprocessor (PHP)*

“Arief mengemukakan bahwa *Page Hyperteks Preprocessor (PHP)* adalah bahasa *serve-side scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis”(Sintawati, 2018).

F. *MySQL*

“Rosa dan Shalahuddin mengemukakan bahwa *SQL (Structural Query Language)* adalah bahasa yang digunakan untuk mengelola data pada RDBMS. *SQL* awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus” (Sintawati, 2018).