

## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Dasar

Sistem dapat didefinisikan dengan mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu. Seperti sistem lainnya, sebuah sistem informasi terdiri atas input (data, instruksi) dan output (laporan, kalkulasi).

Sistem informasi adalah suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan.

##### 2.1.1. Klasifikasi Sistem

Menurut (Jeperson Hutahean, 2014) “Sistem diklasifikasi dalam beberapa sudut pandang:

1. Klasifikasi sistem sebagai :

- a. Sistem abstrak (*abstract system*)

Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik.

- b. Sistem fisik (*physical system*)

Sistem fisik adalah sistem yang ada secara fisik.

2. Sistem diklasifikasi sebagai :

- a. Sistem alamiah (*natural system*)

Sistem alamiyah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi.

b. Sistem buatan manusia (*human made system*)

Sistem buatan manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*human machine system*).

### 2.1.2. Pengertian Sistem informasi

Menurut Jerry FitzGerald dalam (Suryanto, 2018) “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.

Menurut Wing dalam (Krismiaji, 2015) "Sistem Informasi Akuntansi adalah komponen organisasi yang dirancang untuk mengolah data keuangan menjadi informasi atau laporan keuangan yang ditujukan kepada pihak internal maupun eksternal perusahaan. Data dapat diolah menjadi informasi dengan cara manual maupun dengan bantuan komputer".

### 2.1.3. Pengertian Perancangan Sistem

Menurut (Sukadi, 2012) Perancangan sistem merupakan tahap lanjutan setelah tahap analisis sistem, yang bertujuan untuk mendefinisikan kebutuhan-kebutuhan fungsional, mempersiapkan pembuatan implementasi sistem baru, menggambarkan sistem baru, mengatur dan merencanakan elemen-elemen yang terpisah kedalam satu kesatuan yang utuh, serta mengkonfigurasi perangkat-perangkat yang dibutuhkan dalam perancangan sistem baru.

## 2.2. Peralatan Pendukung

Peralatan pendukung merupakan alat yang digunakan untuk menggambar logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, ataupun diagram-diagram yang menunjukkan secara tepat arti dan fungsinya. Fungsi dari peralatan pendukung adalah untuk menjelaskan kepada user bagaimana fungsi dari sistem informasi dapat bekerja dengan suatu bentuk logika model dan phsyscal model.

### 2.2.1. PHP (*Hypertext Preprocessor*)

Menurut (Abdussomad, 2016) mengemukakan bahwa “*PHP* adalah bahasa pemrograman yang memungkinkan anda menggenerate kode HTML secara dinamis, artinya adalah bisa membuat tampilan halaman *web* yang dinamis, bisa berubah-ubah sesuai dengan keinginan programmernya”.

### 2.2.2. XAMPP

Menurut (Fridayanthie & Tias, 2016) menjelaskan bahwa “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal”.

### 2.2.3. MySQL

Menurut (Abdussomad, 2016) mengemukakan bahwa “My SQL merupakan salah satu database kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. *MySQL* bekerja menggunakan bahasa SQL (*Structured Query Language*) yang merupakan bahasa standar yang digunakan untuk manipulasi database.

Menurut Sukamto dan Shalahuddin (2014:46) dalam (Candra, Siregar, Rukiastiandari, Cut, & No, n.d.) mengemukakan bahwa “SQL (Structured Query Language) adalah bahasa yang digunakan untuk mengelola data pada RDBMS”.

#### **2.2.4. Entity Relationship Diagram (ERD)**

Menurut Sukamto & Shalahuddin (2018) dalam (Suryanto, 2018) “Entity Relationship Diagram (ERD) adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional.” Sebelum kita membahas lebih jauh tentang bagaimana Diagram E-R tersebut dapat kita gambarkan, maka yang harus lebih dulu diketahui adalah komponen-komponen *Model Entity-Relationship*. Sesuai namanya, ada 2 (dua) komponen utama pembentuk *Model Entity-Relationship*, yaitu Entitas (*Entity*) dan Relasi (*Relation*). Kedua komponen ini didedikasikan jauh melalui sejumlah Atribut atau property.

#### **2.2.5. Logical Record Structure (LRS)**

Menurut Andriansyah (2016:53) dalam (Candra et al., n.d.) memberikan batasan bahwa LRS adalah “sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola atau aturan permodelan tertentu dalam kaitannya dengan konvensi ke LRS”.

#### **2.2.6. Unified modelling language (UML)**

Menurut Urva dan Siregar (2015) dalam (Suryanto, 2018) “Unified Modelling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak”. UML.

Menurut (Akil, 2018) mengemukakan bahwa “*UML* adalah bahasa pemodelan visual yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak”.

### 2.2.7. *Activity Diagram*

Menurut (Sukanto dan Shalahuddin, 2016) mengatakan bahwa “Diagram aktivitas atau *activity diagram* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut menurut (Sukanto dan Shalahuddin, 2016), adalah:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

### 2.2.8. *Sequence Diagram*

Menurut (Sukanto dan Shalahuddin, 2016) menjelaskan bahwa, “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

### 2.2.9. Deployment Diagram

Menurut (Tohari, 2014) menjelaskan bahwa, “Diagram *deployment* menunjukkan tata letak sebuah sistem secara fisik. Diagram ini akan menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan kethubungan antara komponen-komponen *hardware* tersebut”.

Untuk menggambar diagram *deployment*, ada beberapa hal yang harus diidentifikasi terlebih dahulu menurut (Tohari, 2014), yaitu:

1. Menentukan *node*.
2. Hubungan antar *node*.

Penentuan *node*, diantaranya dapat dilakukan dengan beberapa pedoman menurut (Tohari, 2014) sebagai berikut:

1. *Node* merupakan elemen fisik yang sudah tersedia dalam sistem atau aplikasi.

Amatilah dan jadikan elemen fisik sebagai *node* di diagram *deployment*.

2. *Node* mewakili sumber-sumber komputasi.
3. *Node* dilambangkan dengan kubus.

Hubungan antara *node* dan komponen adalah komponen merupakan hal terkait dan ada dalam eksekusi sistem, sebagai *node* merupakan hal yang mengeksekusi komponen.

#### 2.2.10. Struktur Navigasi

Menurut (Rahman, 2014) “Struktur Navigasi merupakan hubungan antar *scene* sehingga terbentuk alur atau kegiatan dari suatu aplikasi”.

Menurut Prihatna (2005:51) “Struktur navigasi adalah struktur atau alur suatu program yang merupakan rancangan hubungan dan rantai kerja dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan website”.

Menentukan struktur navigasi merupakan halaman yang sebaiknya dilakukan sebelum membuat suatu website. Ada empat macam bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan website, yaitu:

##### 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 dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

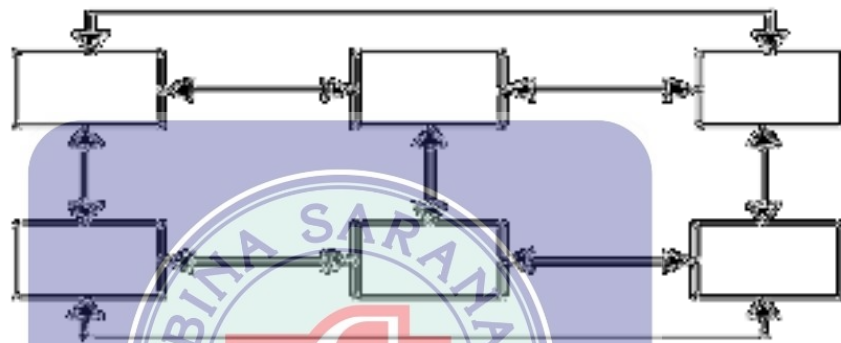


##### 2. Struktur Navigasi Hirarki

Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan

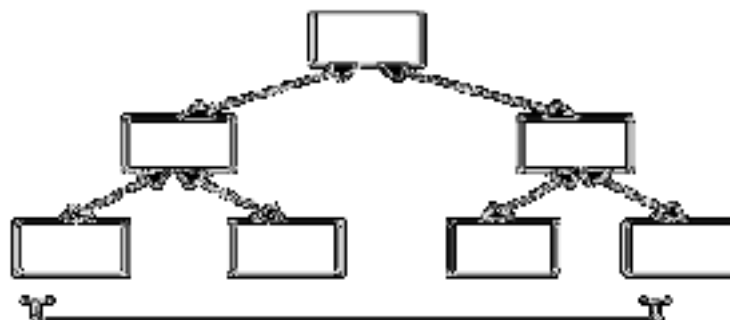


kriteria tertentu. Tampilan pada menu satu akan disebut sebagai Master Page (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut Slave Page (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama Master Page (halaman utama kedua) dan seterusnya. Pada navigasi ini tidak diperkenalkan adanya tampilan secara linier.



### 3. Struktur Navigasi Non-Linier

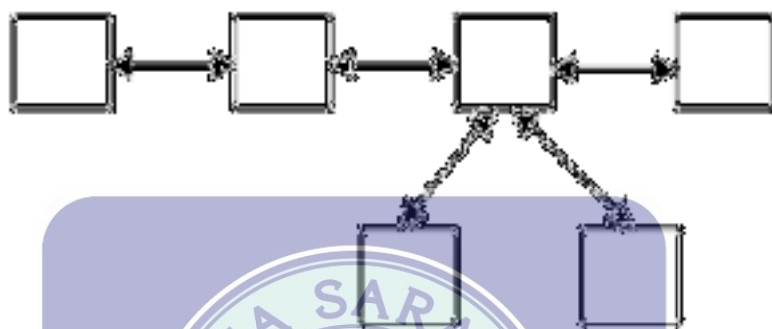
Struktur navigasi Non-linier merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan pada struktur non linier ini berbeda dengan percabangan pada struktur hirarki. Karena pada percabangan ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada Master Page dan Slave Page.





#### 4. Struktur Navigasi Composite (Campuran)

Struktur navigasi composite (campuran) disebut juga struktur navigasi bebas yang merupakan gabungan dari ketiga struktur yang ada. Struktur navigasi ini biasa digunakan dalam pembuatan multimedia karena dapat memberikan keinteraksian yang lebih tinggi.



##### 2.2.11. *Framework*

Framework adalah suatu library yang telah diorganisasikan pada sebuah rancangan arsitektur untuk memberikan kecepatan, ketetapan, kemudahan, dan konsisten dalam pengembangan suatu aplikasi. Untuk memudahkan para programmer untuk membuat sebuah aplikasi web yang di dalamnya ada berbagai fungsi diantaranya plugin, dan konsep untuk membentuk suatu sistem tertentu agar tersusun dan bersetruktur dengan rapih.

##### 2.2.12. *Framework CodeIgniter*

(Ruli dkk, 2017) *CodeIgniter* adalah sebuah web application framework yang digunakan untuk membangun aplikasi PHP dinamis yang dibangun menggunakan konsep model *view controller development pattern*. *CodeIgniter* menyediakan berbagai macam library yang dapat mempermudah dalam

pengembangan dan termasuk *framework* tercepat dibandingkan dengan *framework* lainnya.

### 2.2.13. OOP (*Object Oriented Programming*)

OOP adalah paradigm pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus mengikat.

