

## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Dasar Web

Menurut A. Kadir dalam (Mochamad Rendy Riskianto Widodo, M. Roziq Zainuddin, 2016:87), mengemukakan bahwa: *Web* berasal dari kata Bahasa Inggris yang bila diterjemahkan dalam Bahasa Indonesia berarti Jaring Laba-Laba. Hampir sama dengan arti dari kata *web* itu sendiri, *web* telah membentang ke seluruh penjuru dunia. Tidak hanya terbatas pada lembaga-lembaga penelitian yang ingin mempublikasikan hasil riset, tetapi juga telah banyak digunakan oleh perusahaan bisnis yang ingin mengiklankan produk atau untuk melakukan transaksi bisnisnya.

##### 2.1.1. Website

Menurut (Herliana & Rasyid, 2016:43), *Website* merupakan istilah yang sudah tidak asing lagi dewasa ini. Secara umum *website* dapat diartikan sebagai sebuah halaman yang tersedia dalam sebuah server yang dapat diakses menggunakan jaringan internet dimana didalamnya berisi bermacam-macam informasi dari suatu konten tertentu. Sebuah halaman web yang tampil pada jejaring, umumnya dibuat melalui serangkaian *plain text* yang dikenal dengan istilah HTML (*Hyper Text Markup Language*) atau XHTML (*Extensible HyperText Markup Language*).

##### 2.1.2. Internet

Menurut Shalahuddin & Sukanto dalam (Suparwanto & Pradiatiningtyas, 2017:2), mengemukakan bahwa: “*Internet* adalah sekumpulan jaringan berbeda yang saling bersama sebagai kesatuan dengan menggunakan berbagai macam *protocol*, salah satunya adalah protocol TCP/IP (*Transmission Control Protocol/Internet Protocol*)”.

##### 2.1.3. Web Server

Menurut Sovia & Febio dalam (Suparwanto & Pradiatiningtyas, 2017:2), mengemukakan bahwa: *Web Server* adalah sebuah program atau aplikasi yang memberikan layanan yang berisikan sekelompok kode-kode program berbasis teks yang sederhana ataupun teks yang rumit yang bersifat

universal yang sering disebut *html*, kode-kode *html* tersebut yang sifatnya universal akan diterjemahkan oleh komputer *user* dengan bentuk tampilan yang sama dalam bentuk teks, grafik atau multimedia.

#### 2.1.4. Web Browser

Menurut Sovia & Febio dalam (Suparwanto & Pradiatiningtyas, 2017:2), mengemukakan bahwa: “*Web browser* merupakan suatu program komputer yang dirancang untuk menerjemahkan kode-kode perintah *html* menjadi sebuah tampilan *web* di komputer yang berupa teks, gambar dan multimedia yang bisa dilihat langsung dan dinikmati secara langsung, dengan tampilan yang dihasilkan dapat dibaca dan dimengerti oleh orang awam sekalipun”.

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

Menurut Wahyono dalam (Ferdika & Kuswara, 2017) mengemukakan bahwa: *PHP* atau *Hypertext Preprocessor* merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. Artinya, dapat membentuk suatu tampilan berdasarkan permintaan terkini. *PHP* merupakan teknologi open source yang bebas diperoleh dan diatur untuk memenuhi kebutuhan perusahaan, pendidikan, atau profesional yang didukung oleh komunitas pengguna dan pengembangnya. Untuk menjalankan *PHP* dibutuhkan suatu web server.

#### 2.1.6. HTML (*Hypertext Markup Language*)

Menurut (Zaenal A Rozi & SmithDev Community, 2016:17), berpendapat bahwa: “*HTML* merupakan singkatan dari *Hypertext Markup Language*”. *HTML* adalah bahasa pengkodean yang digunakan untuk membuat halaman web agar bisa ditampilkan melalui web browser”.

Dari definisi tersebut, kita dapat mengetahui tugas dasar teknologi *HTML* yang dikembangkan, yaitu sebagai berikut:

1. *HTTP (Hypertext Transfer Protocol)*: Protokol yang menjadi pondasi bagi pertukaran informasi di layanan web.

2. *URL (Uniform Resource Locator)*: Pengidentifikasian sebuah sumber informasi di web melalui sederet karakter tertentu. Saat ini lazim dikenal dengan sebutan “alamat web”.
3. *HTML (Hypertext Markup Language)*: Sebuah bahasa untuk mempublikasikan informasi di web.

### 2.1.7. Bahasa Pemrograman

Menurut (Litalia, 2018), mengatakan bahwa: “*Bahasa pemrograman* adalah sebuah instruksi untuk memerintah komputer menjalankan fungsi tertentu, namun hanya instruksi standar saja. bahasa pemrograman ini berupa sebuah himpunan aturan sintaks dan semantic yang tugasnya untuk mendefinisikan program komputer”.

### 2.1.8. Sistem

Menurut Kadir dalam (Trimahardhika & Sutinah, 2017:250), mengemukakan bahwa: “Sistem merupakan sekumpulan elemen yang saling terintegrasi dan dimaksudkan untuk mencapai suatu tujuan”.

Sedangkan Menurut (Andi, 2017:3), menyatakan bahwa: *Sistem* adalah dua atau lebih komponen yang saling berkaitan yang berinteraksi untuk mencapai suatu tujuan.

Sebagian besar sistem terbentuk dari beberapa subsistem yang lebih kecil yang mendukung sistem yang lebih besar. Setiap subsistem didesain untuk mencapai satu atau lebih tujuan organisasi. Perubahan dalam subsistem tidak dapat dilaksanakan tanpa mempertimbangkan efeknya terhadap subsistem lainnya atau terhadap sistem keseluruhan. Semakin besar suatu organisasi, maka sistemnya semakin kompleks, sehingga semakin sulit untuk menyelaraskan tujuan dari semua subsistem agar searah dengan tujuan sistem secara keseluruhan.

### 2.1.9. Karakteristik Sistem

Untuk membedakan dan mengenal antara satu sistem dengan sistem lainnya, maka dapat dilakukan dengan pendekatan karakteristiknya yang melekat pada suatu sistem tersebut.

Karakteristik sistem Menurut (Anggraeni, 2017:24) yaitu:

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari jumlah komponen yang saling berinteraksi, yang saling bekerja sama membentuk suatu komponen sistem.

2. Batasan Sistem (*Boundary*)

Daerah membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

3. Subsistem (*Sub System*)

Bagian-bagian dari sistem yang berinteraksi dan berinteraksi sama lain untuk mencapai tujuan dengan sasarannya masing-masing.

4. Lingkungan luar sistem (*Environment*)

Suatu sistem yang ada diluar dari batas sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*interface*)

Media penghubung antara suatu sub sistem dengan sub sistem lain. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu sub sistem ke sub sistem lainnya.

6. Masukan Sistem (*input*)

Energi yang masuk kedalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat berinteraksi.

7. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasi menjadi keluaran yang berguna dan sisa pembuangan.

8. Pengelolaan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan mengubah masukan menjadi keluaran.

9. Sasaran Sistem (*Object*)

Tujuan yang ingin dicapai oleh sistem akan dikatakan berhasil apabila mengenai sasaran atau tujuan.

### 2.1.10. Klasifikasi Sistem

Selain mempunyai karakteristik sistem, sistem juga mempunyai beberapa klasifikasi menurut (Anggraeni, 2017:26), yang terdiri dari beberapa kelompok diantaranya sebagai berikut:

1. Sistem Abstrak dan Sistem Fisik
  - a. Sistem Abstrak Sistem yang berisi gagasan atau konsep.
  - b. Sistem Fisik Sistem yang secara fisik dapat dilihat.
2. Sistem Deterministik dan Sistem Probabilitik
  - a. Sistem Deterministik: Suatu sistem yang operasinya dapat diprediksi secara tepat.
  - b. Sistem Probabilistik: Sistem yang tak dapat diramal dengan pasti karena mengansung unsur probabilitas.
3. Sistem Tertutup dan Sistem Terbuka
  - a. Sistem Tertutup: Sistem yang tidak bertukar materi, informasi, atau energi lingkungan. Tedapat pula sistem yang relatif tertutup dengan ciri-ciri adalah sistem hanya mempunyai masukan dan keluaran yang tertentu, terkendali, dan gejolak dari luar sistem/lingkungan tidak mempengaruhinya.

- b. Sistem Terbuka: Sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan. Ciri-cirinya, sistem menerima masukan yang diketahui, yang diketahui, yang bersifat acak, maupun gangguan lain. Umumnya sistem melakukan adaptasi terhadap lingkungan.
4. Sistem Alamiah dan Sistem Buatan Manusia
    - a. Sistem Alamiah: Sistem yang terjadi karena alam (tidak dibuat oleh manusia)
    - b. Sistem Buatan Manusia: Sistem yang dibuat oleh manusia.
  5. Sistem Sederhana dan Sistem Kompleks
    - a. Sistem yang sederhana misalnya sepeda
    - b. Sistem yang kompleks misalnya otak manusia.

#### 2.1.11. Informasi

Menurut (Andi, 2017:4), mengatakan bahwa: *Informasi* adalah data yang telah diorganisir dan diproses sehingga bermanfaat bagi proses pengambilan suatu keputusan.” Semakin banyak dan semakin berkualitas informasi yang tersedia, maka pengambilan keputusan menjadi semakin baik. Namun, jika informasi terlalu banyak sehingga melebihi kemampuan otak untuk menyerap dan memprosesnya, maka hanya akan menurunkan kualitas pengambilan keputusan dan meningkatkan biaya penyediaan informasi tersebut.

Menurut (Andi, 2017:4) Ada tujuh karakteristik suatu informasi dikatakan bermanfaat, yaitu:

1. Relevan, yaitu dapat mengurangi ketidakpastian, meningkatkan kualitas pengambilan keputusan, serta mengkonfirmasi atau mengkoreksi ekspektasi awal.
2. Andal, yaitu bebas dari kesalahan atau bias.
3. Lengkap, yaitu informasi dikatakan lengkap jika tidak menghilangkan aspek penting dari suatu kejadian atau aktivitas yang diukur.

4. Tepat waktu, yaitu tersedia saat diperlukan untuk mengambil suatu keputusan.
5. Dapat dipahami. Disajikan dalam format yang mudah dipahami dan bermanfaat.
6. Dapat diverifikasi. Jika informasi tersebut dibaca oleh dua orang berbeda yang berpengetahuan memadai akan menghasilkan informasi yang sama.

#### **2.1.12. Sistem Informasi**

Menurut Reynolds dalam (Wiguna, 2017:189), mengemukakan bahwa “*Sistem informasi (SI)* adalah seperangkat komponen yang saling terkait dalam mengumpulkan, memproses, menyimpan, serta menyebarkan data dan informasi. Suatu sistem informasi menyediakan mekanisme umpan balik dalam memantau dan mengendalikan operasinya untuk memastikan agar terus memenuhi tujuan-sasarannya”.

#### **2.1.13. Sistem Informasi Akuntansi**

Menurut (Fauzi, 2017:25), mengatakan bahwa: “*Sistem Informasi Akuntansi* adalah suatu komponen organisasi yang mengumpulkan, mengklasifikasikan, memproses, menganalisis, mengomunikasikan informasi pengambilan keputusan dengan orientasi finansial yang relevan bagi pihak-pihak luar dan pihak-pihak dalam perusahaan”.

#### **2.1.14. Penggajian**

Menurut Susanto dan Sumarsono dalam (Hidayatun, 2016:3), mengemukakan bahwa: “Gaji adalah pendapatan/ penghasilan yang dimasukkan kedalam daftar gaji karyawan yang setiap bulannya dibayarkan dimuka”.

Sedangkan Penggajian adalah suatu penerimaan sebagai imbalan dari pengusaha kepada karyawan untuk suatu pekerjaan atau jasa yang telah atau dilakukan dan dinyatakan atau dinilai dalam bentuk uang yang ditetapkan atas dasar suatu persetujuan atau peraturan perundang-undangan serta dibayarkan atas dasar

suatu perjanjian kerja antara pengusaha dengan karyawan termasuk tunjangan, baik untuk karyawan itu sendiri maupun untuk keluarga.

Dari paparan diatas dapat disimpulkan bahwa: gaji merupakan suatu kompensasi yang dibayarkan oleh perusahaan kepada pegawai sebagai balas jasa atas kinerjanya yang telah diberikan kepada perusahaan. Kompensasi tersebut biasanya diberikan rutin bulanan kepada pegawai. Adapun pencatatan proses akuntansi penggajian ke dalam jurnal sebagai berikut :

1. Pada saat gaji yang belum dibayarkan

|             |     |
|-------------|-----|
| Biaya Gaji  | xxx |
| Hutang Gaji | xxx |

2. Pada saat pembayaran gaji

|            |     |
|------------|-----|
| Biaya Gaji | xxx |
| Kas        | xxx |

#### 2.1.15. Basis Data (*Database*)

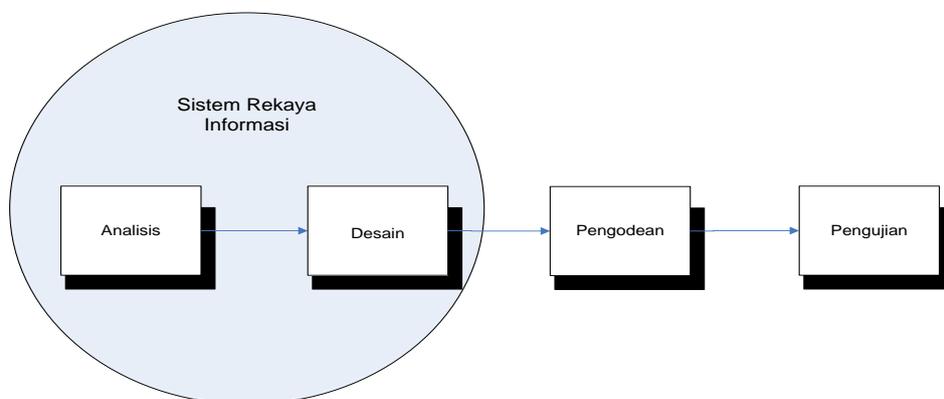
(Herliana & Rasyid, 2016:46), mengatakan bahwa: “Sistem basis data atau *Database* merupakan Kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut”.

#### 2.1.16. Model Pengembangan Perangkat Lunak

Menurut (Rosa A.S dan M. Shalahuddin, 2015:28), mengemukakan bahwa: “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 ini adalah gambar model air terjun:



Sumber: Rosa A.S dan M. Shalahuddin (2015:29)

**Gambar II.1. Ilustrasi Model *Waterfall***

1. Analisis kebutuhan perangkat lunak

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

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 didokumentasikan.

3. 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.

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.

**2.1.17. Netbeans**

Menurut (Nofriadi, 2018:4), *Netbeans* merupakan sebuah aplikasi *Integrated Development Environment (IDE)* yang berbasiskan Java dari Sun Microsystems yang berjalan diatas swing dan banyak digunakan sekarang sebagai editor untuk berbagai bahasa pemrograman, pada Netbeans kita dapat membuat bahasa pemrograman Java, JavaScript, PHP, Python, Ruby, Groovy, C++, Scala, Clojure.

### 2.1.18. *Servlet*

Menurut (Supardi, 2015:3), mengatakan bahwa: “Servlet merupakan class yang berjalan dalam *Java enabled Web Server* dan melakukan generating dokumen HTML yang kemudian dikirimkan ke browser klien untuk ditampilkan.”

### 2.1.19. *JSP*

Menurut (Supardi, 2015:3), mengatakan bahwa: “JSP (*Java Server Page*) merupakan salah satu bagian dari J2EE (*Java 2 Enterprise Edition*). JSP dibangun diatas Servlet dan bertujuan untuk menambah efisiensi. *Java Server Pages* (JSP) merupakan *script* atau program yang diletakkan di server, yang dapat dijalankan memakai *browser* tertentu”.

### 2.1.20. *PhpMyAdmin*

Menurut Winoto dalam (Suparwanto & Pradiatiningtyas, 2017:4), mengemukakan bahwa: “*PhpMyAdmin* adalah suatu program open source yang berbasis web yang dibuat menggunakan aplikasi PHP. Program ini digunakan untuk mengakses database MySQL”.

### 2.1.21. *MySQL*

Menurut Nugroho dalam (Ferdika & Kuswara, 2017:177), mengatakan bahwa: *MySQL* merupakan salah satu perangkat lunak sistem manajemen basis data (*database management system*) atau DBMS yang menggunakan perintah standart SQL (*structured Query Language*). Dimana MySQL mampu untuk melakukan banyak eksekusi perintah query dalam satu permintaan, baik itu menerima dan mengirimkan data. MySQL juga multi-user dalam arti dapat dipergunakan oleh banyak pengguna dalam waktu bersamaan. MySQL tersedia tersedia dalam perangkat lunak gratis dibawah lisensi GNU General Public Lidence (GPL) dan juga menjual dalam lisensi komersial untuk keperluan jika pengguna tidak cocok menggunakan lisensi General Public Lidence (GPL). Pengguna MySQL yang merupakan sebuah database server sekaligus dapat sebagai client, dan dapat berjalan di multi-OS (Operating System) memiliki keunggulan lainnya seperti Open Source sehingga penggunaanya tidak perlu membayar lisensi kepada pembuatnya. Bahasa pemrograman yang dapat digunakan untuk mengakses MySQL diantaranya adalah dengan C, C++, Java, Perl, PHP, Phytion, dan APIs.

### 2.1.22. CSS (*Cascading Style Sheets*)

Menurut (Zaenal A.Rozi & SmitDev Community, 2016:69), mengatakan bahwa: “CSS adalah singkatan dari *Cascading Style Sheets*. CSS adalah bahasa pengkodean yang digunakan untuk menata gaya tampilan halaman web agar lebih cantik dan indah saat ditampilkan di web browser”.

### 2.1.23. Xampp

Menurut Bunafit Nugroho dalam (Mochamad Rendy Riskianto Widodo, M. Roziq Zainuddin, 2016:89), mengemukakan bahwa Xampp merupakan paket PHP berbasis open source yang berfungsi sebagai server yang berdiri sendiri (localhost). Beberapa paket yang telah disediakan adalah Apache, MySQL, PHP, Filezila dan PhpMyAdmin. Program ini tersedia dalam GNU General Public License dan bebas, merupakan tampilan halaman web dinamis.

### 2.1.24. iReport

Menurut Kurniawan dalam (Bukit & Sofica, 2018:44), “*iReport* adalah *report designer visual* yang dibangun pada *JasperReport*. *iReport* bersifat intuitif dan mudah digunakan pembangunan laporan *visual* atau *desainer* untuk *JasperReport* dan tertulis dalam kitab *Java*.

Sebagai alternatif, terdapat *tools iReport* (dengan *library JasperReport* yang dapat membantu dalam pembuatan laporan. *Library JasperReport* sendiri merupakan *Java Library* (JAR) yang bersifat *open* dan dirancang untuk menambahkan kemampuan pelaporan (*reporting capabilities*) pada aplikasi *Java*.

## 2.2. Peralatan Pendukung (*Tools Systems*)

Dalam membangun sebuah sistem informasi memerlukan adanya peralatan pendukung atau tools untuk mempermudah dalam proses pembangunan sistem itu sendiri. Selain itu peralatan pendukung juga sebagai alat yang tepat digunakan untuk menggambarkan bentuk logika model, simbol-simbol, lambang-lambang dan

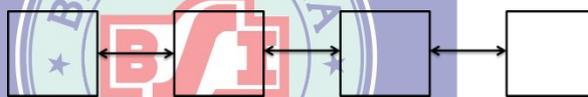
diagram dari suatu program sistem informasi yang dapat menambah kemudahan dalam rangka merencanakan prosedur-prosedur yang akan dipakai pada sistem informasi tersebut.

### 2.2.1. Struktur Navigasi

Dalam pembuatan website, hal yang harus diperhatikan sebelum merancang tampilan web adalah pembuatan struktur navigasi. Ada empat struktur dasar yang digunakan Menurut Binanto dalam (Eka & Arviana, 2018:179), yaitu: linear, hierarkis, nonlinear, dan komposit.

#### a. Linear

Pengguna akan melakukan navigasi secara berurutan dari frame atau byte informasi yang satu ke yang lainnya.

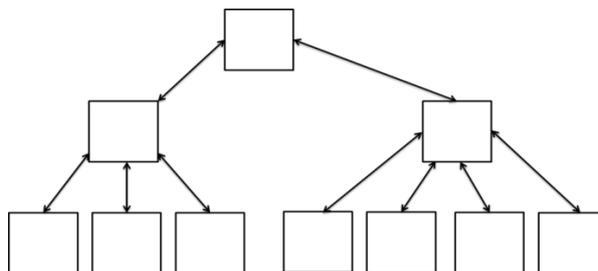


Sumber: Menurut Binanto dalam (Eka & Arviana, 2018:179)

**Gambar II.2. Struktur Navigasi Linear**

#### b. Hierarkis

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

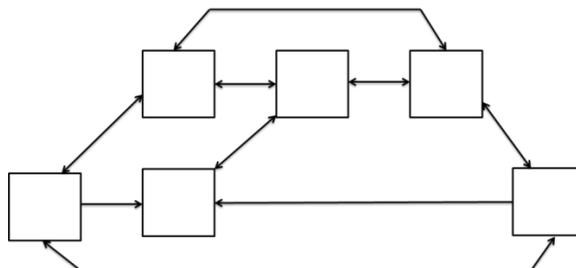


Sumber: Menurut Binanto dalam (Eka & Arviana, 2018:179)

**Gambar II.3. Struktur Navigasi Hierarkis**

c. Nonlinear

Pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terkait dengan jalur yang sudah ditentukan sebelumnya.

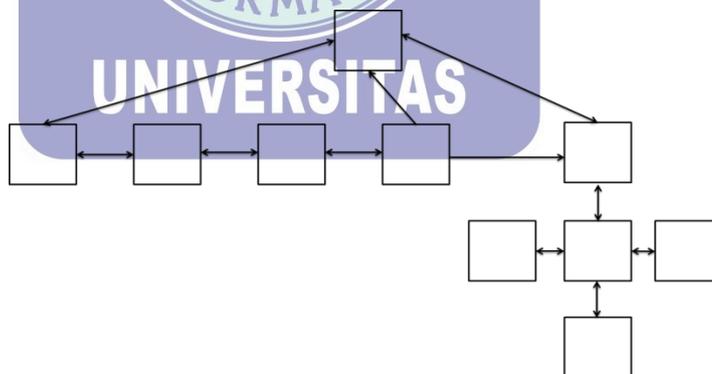


Sumber: Menurut Binanto dalam (Eka & Arviana, 2018:179)

**Gambar II.4. Struktur Navigasi Nonlinear**

d. Komposit

Pengguna akan melakukan navigasi dengan bebas (secara nonlinear), tetapi terkadang dibatasi presentasi linear film atau informasi penting dan atau pada data yang paling terorganisasi secara logis pada suatu hierarki.



Sumber: Menurut Binanto dalam (Eka & Arviana, 2018:179)

**Gambar II.5. Struktur Navigasi Komposit**

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

Menurut Menurut (Rosa A.S dan M. Shalahuddin, 2015:50), mengemukakan bahwa: “*Entity Relationship Diagram (ERD)* dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional”. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

1. Entitas/*entity*

Entitas merupakan data inti yang akan di simpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat di akses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

2. Atribut

*Field* atau kolom data yang butuh disimpan dalam suatu entitas.

3. Atribut kunci primer

*Field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan biasanya berupa id, kunci primer dapat lebih dari satu kolom asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4. Atribut multinilai/*multivalue*

*Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

5. Relasi

Relasi yang menghubungkan antar entitas biasanya diawali dengan kata kerja.

6. Asosiasi/*association*

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakainya atau kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan, ada kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dengan entitas B.

**2.2.3. Logical Record Structure (LRS)**

Menurut Wulandari dalam (Cahyani & Nurmalasari, 2016:69), *Logical Record Structure* dibentuk dengan nomor tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Berikut tahapan transformasi ERD ke LRS:

1. Konversi ERD ke LRS, *Entity Relationship Diagram* harus diubah ke bentuk LRS (*struktur record secara logic*). Dari bentuk LRS inilah yang nantinya dapat ditransformasikan ke bentuk relasi tabel.
2. Konversi ERD ke LRS sebuah model sistem yang digambarkan dengan sebuah model sistem yang digambarkan dengan sebuah ERD akan mengikuti pola pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS.

**2.2.4. UML (*Unified Modeling Language*)**

Menurut Nugroho dalam (Ferdika & Kuswara, 2017:177), mengemukakan bahwa: UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan

permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

### 2.2.5. *Use Case Diagram*

Menurut Martin Fowler dalam (Ferdika & Kuswara, 2017:178), menyimpulkan bahwa: “*Use case* adalah teknik untuk merekam persyaratan fungsional sebuah sistem”. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use case* diagram menampilkan aktor mana yang menggunakan *use case* mana, *Use case* mana yang memasukkan *use case* lain dan hubungan antara aktor dan *use case*. *Use case* diagram menggambarkan efek fungsionalitas yang telah diharapkan oleh sistem. *Use case* diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan sebuah rancangan aplikasi dengan konsumen, serta merancang test case untuk semua feature yang ada pada sistem. Aturannya, sebuah *Use case* dapat dimasukan lebih dari *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsional yang common”.

### 2.2.6. *Activity Diagram*

Menurut (Rosa A.S dan M. Shalahuddin, 2015:161) mengatakan bahwa: Diagram Aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

### 2.2.7. *Sequence Diagram*

Menurut Munawar dalam (Ferdika & Kuswara, 2017:178), mengemukakan bahwa: “*Sequence Diagram* merupakan interaksi antara sejumlah object dalam urutan waktu. Kegunaan untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antar object yang terjadi pada titik tertentu dalam eksekusi system”.

### 2.2.8. *Deployment Diagram*

Menurut (Rosa A.S dan M. Shalahuddin, 2015:154) mengatakan bahwa: "Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi".

Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
2. Sistem *client server*.
3. Sistem terdistribusi murni
4. Rekayasa ulang aplikasi.

### 2.2.9. *User Interface*

Menurut (T.Suratno, 2016:64), mengatakan bahwa: "*User Interface* atau disebut sebagai antar muka merupakan komponen penting dari sebuah perangkat lunak yang menjadi perantara antara mesin dengan manusia". Antar muka yang tidak tepat akan menimbulkan beberapa kerugian seperti kesenjangan interaksi antara perangkat lunak dengan manusia, hilangnya informasi yang disajikan, *stressing* pengguna, bahkan berdampak terhadap penolakan oleh pengguna.

### 2.2.10. *Black-Box Testing*

Menurut Sukamto dan Shalahuddin dalam (Cahyani & Nurmalasari, 2016:69), mengemukakan bahwa: "*Black-box testing* yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program". Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah.

### 2.2.11. Spesifikasi Perangkat *Hardware* dan *Software*

Menurut Ambarita (2016:2), menjelaskan bahwa: “perangkat keras komputer (*hardware*) adalah: salah satu komponen dari komputer yang bisa dilihat dan diraba secara langsung serta berfungsi untuk mendukung proses komputerisasi”.

Menurut Ambarita (2016:17), menjelaskan bahwa: “perangkat lunak komputer (*software*) terdiri dari perangkat lunak sistem (operasi sistem *software*) dan perangkat lunak program aplikasi (*program application software*)”.

