

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Tinjauan Teori Umum**

##### **2.1.1. Sistem Informasi**

Menurut (Wibowo & Sismoro, 2012: 5), sistem adalah suatu kumpulan komponen yang membentuk suatu jaringan kerja yang saling terhubung untuk melakukan suatu kegiatan guna mencapai sasaran tertentu. Dalam buku (Husda, 2012) yang berjudul *Pengantar Teknologi Informasi*, sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki *item-item* penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada di negara tersebut.

Sedangkan menurut (Tukino & Amrizal, 2017: 200), Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Elemen Sistem adalah suatu sistem terdiri dari sejumlah elemen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan.

Pendapat lain (Mayasari, 2015: 278), adapun syarat-syarat sistem adalah sebagai berikut :

1. Sistem harus dibentuk untuk menyelesaikan masalah.

2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

Dalam buku (Husda, 2012: 111) yang berjudul *Pengantar Teknologi Informasi*, sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Menurut beberapa ahli pengertian sistem dalam bukunya (Husda, 2012: 111-112):

1. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
2. Sistem merupakan seperangkat unsur yang saling terkait dalam suatu unsur relasi diantara unsur-unsur tersebut dengan lingkungan.
3. Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.
4. Sistem merupakan prosedur logis dan rasional untuk merancang suatu rangkaian komponen yang berhubungan satu dengan yang lainnya dengan bermaksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan yang telah ditentukan.

Sehingga dapat disimpulkan bahwa sistem adalah kumpulan komponen-komponen yang terdiri dari sub-sub sistem yang saling berinteraksi dan

bekerjasama untuk menghasilkan *output* yang diinginkan.

Adapun menurut (Husda, 2012: 112-115), sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu seperti:

1. Komponen Sistem (*Component System*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Suatu sistem juga dapat mempunyai suatu sistem yang besar yang disebut *supra system*.

2. Batasan Sistem (*Boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan suatu sistem menunjukkan ruang lingkup (*Scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan, dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-

sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. *Output* dari suatu subsistem akan menjadi *input* dari subsistem lainnya.

#### 5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan pemeliharaan (*Maintenance Input*) dan masukan sinyal (*Signal Input*). Sebagai contoh, didalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah *signal input* yang akan diolah menjadi informasi.

#### 6. Keluaran Sistem (*Output*)

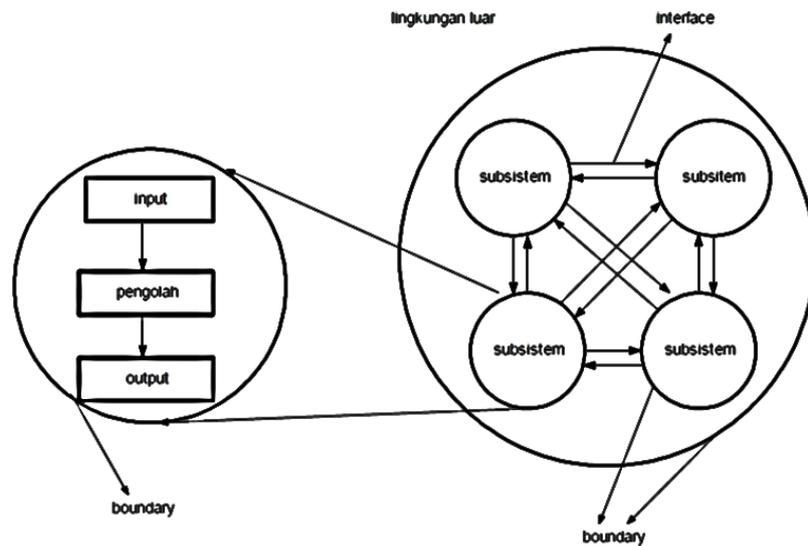
Merupakan hasil dari energi yang diolah oleh sistem. Misalnya, keluaran yang berguna dan keluaran yang tidak berguna.

#### 7. Pengolahan Sistem (*Proses*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

#### 8. Tujuan Sistem (*Goal*)

Setiap sistem mempunyai tujuan ataupun sasaran yang mempengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau mengoperasikan sistem itu mengenai sasaran atau tujuannya.



**Sumber:** (Husda, 2012)

**Gambar 2.1** Karakteristik Sistem

Sistem dapat diklasifikasikan dari berbagai sudut pandang dalam buku (Husda, 2012: 115-116), diantaranya sebagai berikut:

1. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya, Sistem *Teologia* yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan Manusia.

2. Sistem Fisik (*Physical System*)

Merupakan sistem yang ada secara fisik sehingga setiap makhluk hidup dapat melihatnya. Misalnya, Sistem Komputer, Sistem Akuntansi, sistem Produksi dan sebagainya.

3. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam dalam artian tidak dibuat oleh manusia. Misalnya, Sistem Tata Surya, Sistem *Galaxi*, dan lain sebagainya.

4. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *Human Machine System*. Contohnya, Sistem Informasi.

5. Sistem tertutup (*Close System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-bener tertutup).

6. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. lebih spesifik dikenal dengan sistem otomasi yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan control oleh satu arah atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern.

Menurut (Mayasari, 2015: 278), Infomasi adalah data yang diolah menjadi bentuk yang lebih berarti bagi penerimanya dan bermanfaat dalam mengambil

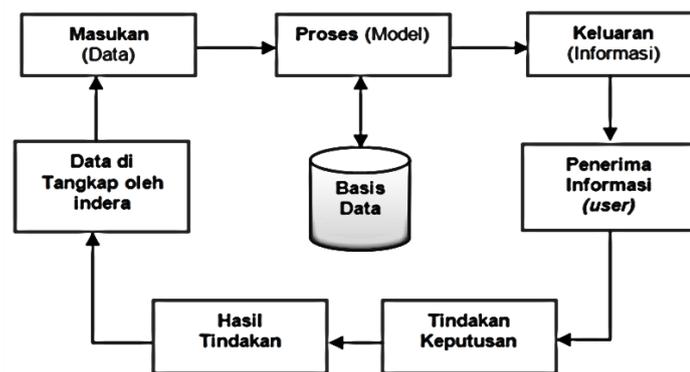
keputusan. Istilah informasi mengarah pada penggunaan teknologi komputer di dalam organisasi untuk menyajikan informasi kepada pemakai. Informasi juga merupakan hal yang sangat penting bagi manajemen didalam pengambilan keputusan.

Sedangkan menurut (Afrizal & Fitriani, 2017: 25), Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data, dimana data itu sendiri merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*events*) adalah sesuatu yang terjadi pada saat yang tertentu.

Pendapat lain dalam buku (Husda, 2012: 117), informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang. Sedangkan menurut Raymond Mcleod, informasi merupakan data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Kesimpulan dari buku (Husda, 2012: 117), secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti

menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan dianggap *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi dapat digambarkan sebagai berikut:



**Sumber:** (Husda, 2012: 118)

**Gambar 2.2** Siklus Informasi

Adapun kualitas Informasi menurut (Husda, 2012: 118-119) adalah sebagai berikut:

1. Akurat

Informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya.

Pendapat (Sutanta, 2011: 14), informasi yang diperoleh dari pengolahan data dapat dinilai berdasarkan sifatnya. Sifat informasi yang menentukan nilai informasi adalah:

1. Kemudahan dalam perolehannya.
2. Sifat luas dan kelengkapannya.
3. Ketelitian (*accuracy*).
4. Kecocokan dengan pengguna (*relevancy*).
5. Ketepatan waktu.
6. Kejelasan (*clarity*).
7. Fleksibilitas/keluwesannya.
8. Dapat dibuktikan.
9. Tidak ada prasangka.
10. Dapat diukur.

•  
Menurut (Tukino, 2016: 70), Sistem informasi merupakan suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. Dalam hal ini, TI hanya merupakan salah satu komponen dalam perusahaan. Komponen-komponen lainnya adalah prosedur, struktur organisasi, sumber daya manusia, produk, pelanggan, rekanan dan sebagainya. Keandalan suatu sistem informasi dalam organisasi terletak pada keterkaitan antar komponen yang ada, sehingga dapat dihasilkan dan dialirkan suatu informasi yang berguna (akurat, terpercaya, detail, cepat, relevan, dan sebagainya) untuk lembaga yang bersangkutan.

Adapun menurut (Gunawan & Agustian, 2014: 5), bahwa “sistem informasi

(*information system*) merupakan kombinasi teratur dari orang-orang, perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi”.

Sedangkan menurut (Mayasari, 2015: 278), Sistem Infomasi adalah “Sebagai satuan yang saling berhubungan yang mengumpulkan (mendapatkan komponen kembali), memproses, menyimpan dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi.

Pendapat dalam buku (Husda, 2012: 120-122), komponen sistem informasi dapat diuraikan sebagai berikut:

1. Blok Masukan (*Input Block*)

*Input* merupakan data yang masuk ke dalam sistem informasi.

2. Blok Model (*Model Block*)

Kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technologi Block*)

Teknologi merupakan kotak alat (*Tool Box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.

5. Blok Basis Data (*Database Block*)

Merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*)

Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.

### **2.1.2. SDLC (*System Development Life Cycle*)**

Dalam jurnal (Hermawan, Hidayat, & Utomo, 2016: 3), Metode *System Development Life Cycle* atau sering disingkat dengan SDLC merupakan pengembangan yang berfungsi sebagai sebuah mekanisme untuk mengidentifikasi perangkat lunak. Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut di terapkan, dioperasikan, dan dipelihara. Daur atau siklus hidup dari pengembangan sistem merupakan suatu

bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah di dalam tahapan tersebut dalam proses pengembangannya.

Menurut (A.S & Shalahuddin, 2013: 26-39) dalam bukunya yang berjudul *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, SDLC memiliki beberapa model dalam penerapan tahapan prosesnya. Hal terpenting adalah mengenai tipe pelanggan (*customer*) dan memilih menggunakan model SDLC yang sesuai dengan karakter pelanggan (*customer*) dan sesuai dengan karakter pengembang. Model-model SDLC sebagai berikut:

1. Model *Waterfall*

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linear (*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, pengkodean, pengujian, dan tahap pendukung (*support*).

2. Model *Prototype*

Model *prototype* banyak digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.

3. Model *Rapid Application Development (RAD)*

Model *Rapid Application Development (RAD)* adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak.

#### 4. Model *Iterative*

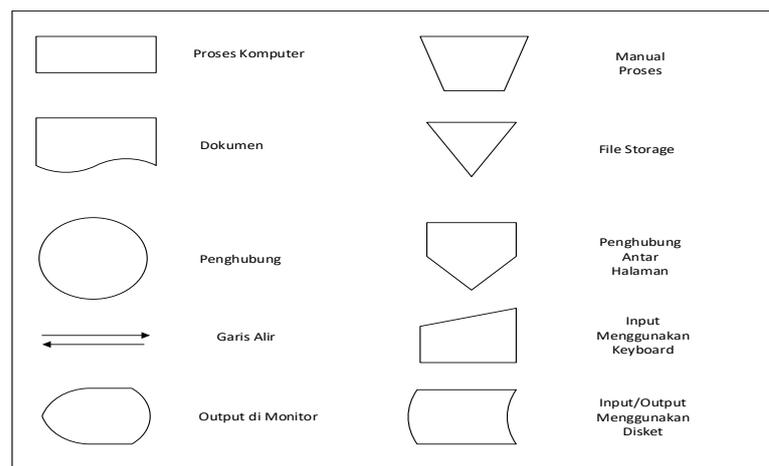
Model Iteratif mengkombinasikan proses-proses pada model air terjun dan Iteratif model prototipe.

#### 5. Model *Spiral*

Model *Spiral* memasang iteratif pada model prototipe dengan kontrol dan aspek sistematis yang diambil dari model air terjun.

### 2.1.3. Aliran Sistem Informasi

Menurut (Ismael, 2017: 149), Aliran Sistem Informasi(ASI) merupakan bagan yang menunjukkan arus pekerjaan serta keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada dalam sistem Adapun simbol-simbol yang dapat digunakan pada Aliran Sistem Informasi(ASI) ini adalah sebagai berikut:



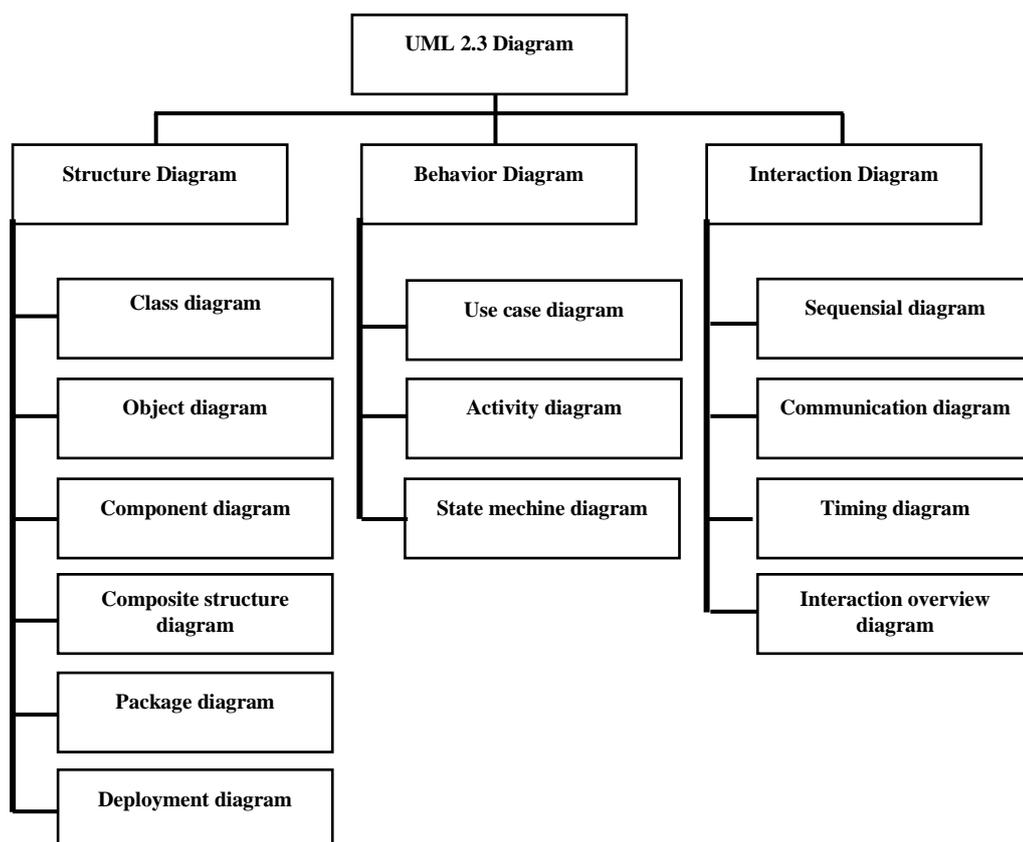
**Sumber:** (Ismael, 2017: 149)

**Gambar 2.3** Simbol-Simbol Aliran Sistem informasi

### 2.1.4. UML (*Unified Modelling Language*)

Menurut (Sanjani, Hartati, & Sudarmaningtyas, 2011: 88), UML adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan sesungguhnya digunakan untuk penyederhanaan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut (A.S & Shalahuddin, 2013: 140), pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar berikut:



**Sumber:** (A.S & Shalahuddin, 2013: 140)

**Gambar 2.4** Diagram UML

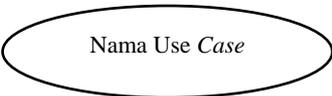
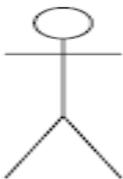
Ada beberapa diagram umum pemodelan UML yang akan digunakan dalam penelitian ini yaitu:

1. *Use Case Diagram*

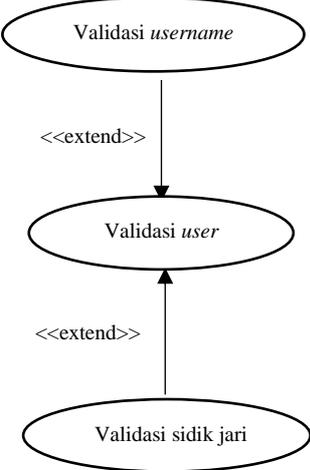
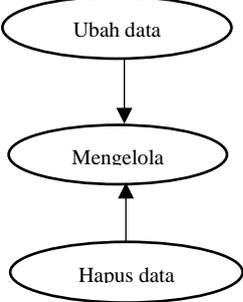
Menurut (Isa & Hartawan, 2017: 141), *Use Case* diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). Sehingga pembuatan *use case diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem.

Berikut simbol-simbol yang ada pada diagram *Use Case Diagram*:

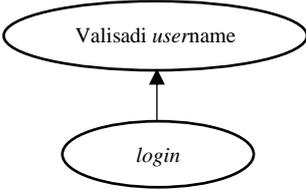
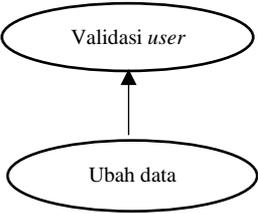
**Tabel 2.1** Simbol Use Case Diagram

No.	Simbol	Deskripsi
1.		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> .
2.	 nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor itu sendiri adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.		Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

**Tabel 2.1** Lanjutan

<p>4.</p>	<p>Ekstensi / <i>extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>5.</p>	<p>Generalisasi / <i>generalization</i></p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p> 

Tabel 2.1 Lanjutan

		arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum).
6.	Menggunakan / <i>include/ uses</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <p>1) <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT     login([login]) --&gt; validasi_username([Validasi username])   </pre> <p>2) <i>Include</i> berarti <i>use case</i> tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT     ubah_data([Ubah data]) --&gt; validasi_user([Validasi user])   </pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: (A.S & Shalahuddin, 2013: 156-158)

## 2. Class Diagram

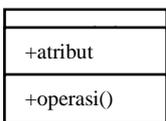
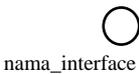
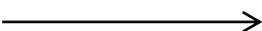
Menurut (Isa & Hartawan, 2017: 141-142), *Class* adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi).

Kelas memiliki tiga area pokok:

- 1) Nama (*Class Name*)
- 2) Atribut
- 3) Metode (*Operation*)

Berikut simbol-simbol yang ada pada diagram *Class Diagram*:

**Tabel 2.2** Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas pada struktu sistem.
2.	<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

**Tabel 2.2** Lanjutan

5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antarkelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).

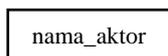
Sumber: (A.S & Shalahuddin, 2013: 146-147)

### 3. *Sequence Diagram*

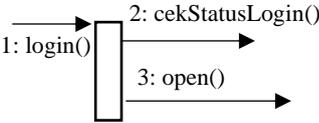
Menurut (Isa & Hartawan, 2017: 141), *Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

Berikut simbol-simbol yang ada pada diagram *Sequence Diagram*:

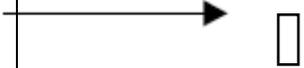
**Tabel 2.3** Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	Aktor  nama aktor atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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.
2.	Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.

Tabel 2.3 Lanjutan

3.	<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">       nama objek: nama_kelas     </div>	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> <div style="border: 1px solid black; width: 20px; height: 40px; margin: 10px auto;"></div>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <pre> sequenceDiagram     actor Actor     activate Actor     Actor-&gt;&gt; Actor: 1: login()     Actor-&gt;&gt; Actor: 2: cekStatusLogin()     Actor-&gt;&gt; Actor: 3: open()     deactivate Actor   </pre> <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login()</p> <p>Aktor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe create</p> <p style="text-align: center;">&lt;&lt;create&gt;&gt;</p> <div style="text-align: center;">  </div>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	<p>Pesan tipe call</p> <p style="text-align: center;">1: nama_metode()</p> <div style="text-align: center;">  </div>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p> <div style="text-align: center;">  <p style="margin-left: 40px;">1: nama_metode()</p> </div> <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>

**Tabel 2.3** Lanjutan

7.	Pesan tipe <i>send</i>  l: masukan  	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i>  l: keluaran  	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i>  <<destroy>>  	Menyatakan suatu objek mengakhiri hidup yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

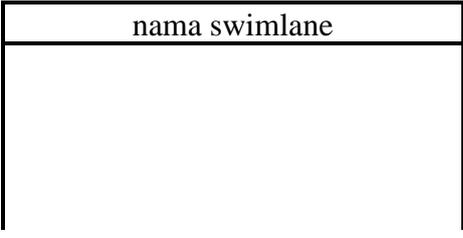
**Sumber:** (A.S & Shalahuddin, 2013: 165-167)

#### 4. *Activity Diagram*

Menurut (Isa & Hartawan, 2017: 141-142), *Activity diagram* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas ke aktifitas yang lainnya, atau dari aktifitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*.

Berikut simbol-simbol yang ada pada diagram *Activity Diagram*:

**Tabel 2.4** Simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

**Tabel 2.4** Lanjutan

	atau	
	nama swimlane	

**Sumber:** (A.S & Shalahuddin, 2013: 162-163)

## 2.2. Tinjauan Teori Khusus

### 2.2.1. *Feedback*

Menurut (Afrizal & Fitriani, 2017: 25), *Feedback* atau umpan balik adalah segala informasi baik yang menyangkut output maupun transformasi. *Feedback* diperlukan untuk memperbaiki input maupun transformasi. *Input* diartikan sebagai pelanggan yang baru membeli suatu produk. *Output* adalah petugas yang melayani pelanggan melalui suatu proses transaksi, sedangkan transformasi adalah pengolah *input* dan *output* dalam hal transaksi tersebut. *Feedback* atau umpan balik adalah respon yang diberikan oleh penerima pesan kepada pengirim sebagai tanggapan atas informasi yang dikirimkan.

Menurut (Saweduling, 2013) Steers (1988:159) menyatakan pekerjaan yang memberikan lebih banyak motivasi menunjukkan lebih banyak variasi, otonomi, tanggung jawab, umpan balik dan identitas tugas. Selanjutnya dikatakan bahwa

atribut tugas semacam itu menciptakan situasi yang lebih menantang pekerjaan dan menawarkan kepuasan intrinsik pada mereka karena mengerjakan sesuatu yang lebih penting dan berguna. Menurut Gitosudarmo (2001:191) karakteristik pekerjaan merupakan sifat dari pelaksanaan tugas karyawan serta meliputi wewenang, tanggung jawab serta bentuk tugas yang diembannya dan juga tingkat kepuasan yang individu peroleh dari karakteristik pekerjaan yang bersangkutan. Sebaliknya suatu perusahaan yang tidak memberikan fasilitas yang cukup bagi karyawannya tentu saja semangat kerja dan motivasi kerjanya juga akan rendah. Robbins (2003:165) menjelaskan karakteristik pekerjaan yang berhubungan dengan motivasi kerja, kinerja dan kepuasan seperti Hackman dan Oldham's (1975) menjelaskan inti dari model karakteristik pekerjaan sebagai berikut:

1. Keragaman ketrampilan adalah lingkup dimana pekerjaan memerlukan seorang individu yang mampu melakukan berbagai tugas yang mengharuskan menggunakan ketrampilan dan kemampuan yang berbeda. Pekerjaan yang lebih banyak motivasi menunjukkan lebih banyak variasi agar dapat terlaksana dengan baik.
2. Identitas tugas adalah lingkup di mana pekerjaan mengharuskan seorang individu untuk melaksanakan seluruh pekerjaan secara lengkap yang dapat diidentifikasi. Dengan kata lain, tingginya identitas tugas tampak pada saat seseorang mengerjakan sesuatu produk atau suatu proyek sejak awal hingga akhir dan membuat hasil yang nyata. Bila pekerjaan tidak mempunyai identitas, para karyawan tidak akan atau kurang merasa bertanggung jawab dan mungkin kurang bangga dengan hasil-hasilnya.

3. Signifikansi tugas adalah lingkup dimana pekerjaan mempengaruhi kehidupan orang lain didalam atau diluar organisasi atau tingkatan suatu pekerjaan, pengaruhnya terhadap pekerjaan orang lain atau pengaruhnya terhadap pegawai lain dalam organisasi.
4. Otonomi adalah lingkup dimana pekerjaan memungkinkan seorang individu untuk mendapatkan kebebasan, kemerdekaan dan keleluasaan baik dalam penjadwalan maupun dalam menentukan prosedur yang digunakan dalam menyelesaikan pekerjaan.
5. Umpan balik adalah lingkup dimana seorang individu menerima informasi yang langsung dan jelas mengenali seberapa efektif melaksanakan pekerjaan atau derajat sejauh mana pekerja memperoleh informasi tentang penilaian prestasi dari pelaksanaan tugasnya.

Umpan balik atau *feedback* merupakan salah satu bagian terpenting dalam sebuah perusahaan, karena dengan adanya *feedback* maka karyawan dan pihak perusahaan bisa mengevaluasi apakah gerak yang dilakukan sudah sesuai dengan yang diharapkan.

### **2.2.2. Web**

Menurut (Isa & Hartawan, 2017: 142), *Website* adalah kumpulan kumpulan halaman *web* yang di dalamnya terdapat sebuah domain mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman *web* yang saling berhubungan. Jadi bisa dikatakan bahwa pengertian *website* adalah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau

gerak, animasi, suara, atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan melalui jaringan-jaringan halaman. Hubungan antara satu halaman *website* dengan halaman *website* lainnya disebut *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*.

### 2.2.3. HTML 5

Menurut (Saputra, 2012: 1), HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*. (Saputra, 2012), saat ini html versi 5 yang paling marak dibicarakan didunia maya. Html 5 layaknya sebuah html biasa yang sering kita gunakan dalam membangun aplikasi *web*, hanya saja html 5 ini memiliki keunggulan dibanding versi terdahulunya. Html 5 mampu menyederhanakan kode-kode html terdahulu menjadi lebih ringkas.

Menurut (Saputra, 2012: 17-18), berikut adalah fitur-fitur terbaru dalam html 5 yang wajib anda ketahui:

1. Unsur *canvas* untuk gambar.
2. Bentuk kontrol *form* seperti kalender, tanggal, waktu, *email*, url, dan *search*.
3. Elemen konten yang lebih spesifik, seperti artikel, *footer*, *header*, navigasi, dan *section*.
4. Dukungan yang lebih baik untuk menyimpan secara *offline*.

5. Dan tentunya juga dukungan untuk pemutaran video dan audio.

#### 2.2.4. CSS3

Menurut (Saputra, 2012: 27), CSS yang merupakan singkatan dari *Cascading Style Sheet* merupakan bahasa pemrograman *web* yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* lebih rapih, terstruktur, dan seragam. Tujuan utama dari CSS adalah untuk memisahkan konten utama dengan tampilan dokumen lainnya (html dan sejenisnya).

(Saputra, 2012), CSS3 merupakan pengembangan dari versi sebelumnya. Peningkatan yang paling mencolok pada versi 3 ini adalah peningkatan fitur yang mengarah pada efek animasi, seperti *crop*, *slide images*, dll.

Menurut (Saputra, 2012: 59-60), fitur-fitur baru dalam CSS3 yang dapat digunakan adalah:

1. *Borders*
  - 1) *Border-color*
  - 2) *Border-image*
  - 3) *Border-radius*
  - 4) *Box-shadow*
2. *Backgrounds*
  - 1) *Background-origin and Backgroun-clip*
  - 2) *Background-size*
  - 3) *Multiple Background*

3. *Color*
  - 1) *HSL colors*
  - 2) *HSLA colors*
  - 3) *Opacity*
  - 4) *RGBA colors*
4. *Text Effects*
  - 1) *Text-shadow*
  - 2) *Text-overflow*
  - 3) *Word-wrap*
5. *User Interface selectors, attribute selector*
  - 1) *Box-sizing*
  - 2) *Resize*
  - 3) *Outline*
  - 4) *Nav-top, nav-right, nav-bottom, nav-left*
6. *Selectors, attribute selectors*
7. *Basic Box Model, overflow-x, overflow-y*
8. *Generated Content, content*
9. *Other Modules*
  - 1) *Media queries*
  - 2) *Multi-column layout*
10. *Web Fonts, speech*

### 2.2.5. PHP

Menurut (Isa & Hartawan, 2017: 142), PHP adalah bahasa *server-side* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke *browser* dengan format HTML.

Menurut (Ramadhani, 2014: 623), PHP merupakan bahasa pemrograman *web* yang bersifat *server-side HTML=embedded scripting*, di mana *script*-nya menyatu dengan HTML dan berada di *server*. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di *server* tetapi disertakan HTML biasa. Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan sistem *database* di dalam *web*.

Sistem *database* yang dapat didukung oleh PHP adalah :

1. *Oracle*
2. *MySQL*
3. *Sybase*

### 2.2.6. JavaScript

Menurut (Andi & Komputer, 2016: 2) dalam bukunya yang berjudul *Webmaster Series JavaScript*, *JavaScript* adalah bahasa yang digunakan untuk membuat program yang digunakan supaya dokumen HTML yang ditampilkan

dalam *browser* menjadi lebih interaktif, tidak sekedar indah saja. Umumnya, program *JavaScript* adalah program yang ditanamkan (disisipkan) ke dalam halaman *web* sehingga halaman (dokumen) *web* menjadi sebuah aplikasi yang berjalan di dalam *browser web*.

### 2.2.7. Basis Data

Menurut (Ramadhani, 2014: 623), pada dasarnya basis data bukanlah sistem yang selalu terkait dengan komputer. Adapun beberapa penjelasan terkait dengan basis data adalah pengertian data, operasi dasar basis data, dan pengertian sistem informasi itu sendiri sistem manajemen basis data. Basis data terdiri dari 2 kata, yaitu basis dan data, basis dapat diartikan sebagai maskas atau gudang tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli dan lain-lain), barang hewan, peristiwa, konsep keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya *Unified Modeling Language (UML)*.

### 2.2.8. MySQL

Menurut (Isa & Hartawan, 2017: 142), MySQL (*My Structure Query Language*) atau yang biasa dibaca mai-sekuel adalah sebuah program pembuat basis data yang bersifat *open source*, artinya siapa saja boleh menggunakannya, MySQL sebenarnya produk yang berjalan pada *platform Linux*, Karena sifatnya

yang *open source*, sehingga kita bisa menggunakannya secara gratis. Bahasa pemrograman PHP juga sangat *support* dengan basis data MySQL. Kelebihan lain dari MySQL adalah menggunakan bahasa *query* standar yang dimiliki SQL (*Structure Query Language*).

Menurut (Saputra, 2012: 77), perintah yang sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data.

### **2.2.9. Dreamweaver CS6**

Menurut (Andi, 2013: 2) dalam bukunya, *Adobe Dreamweaver CS6* adalah versi terbaru dari *Adobe Creative Suite 6*. *Adobe Dreamweaver* sendiri merupakan aplikasi yang digunakan sebagai *HTML editor* profesional untuk mendesain *web* secara visual.



**Sumber :** (Andi, 2013: 2)

**Gambar 2.5** Tampilan awal *Adobe Dreamweaver CS6*

Tampilan awal *Adobe Dreamweaver CS6* diatas menampilkan beberapa pilihan seperti:

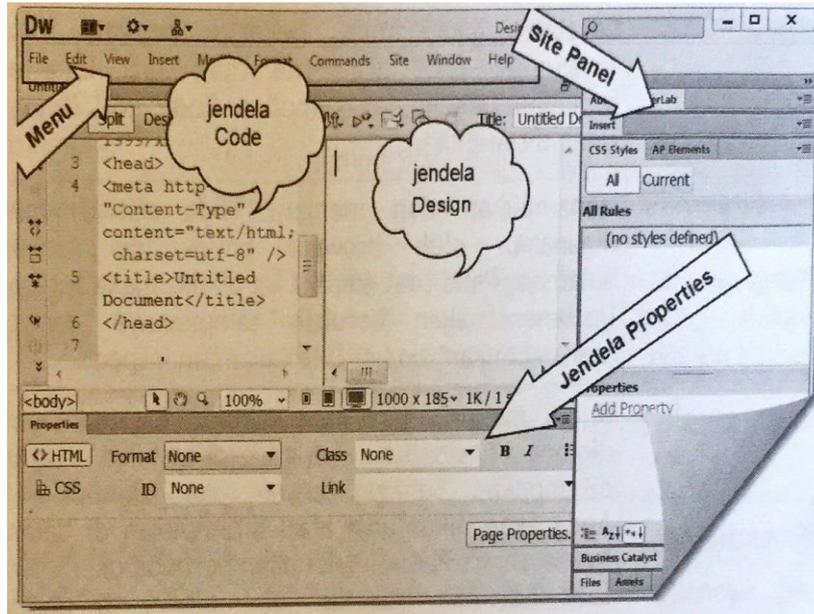
- 1) *Open a Recent Item*, untuk membuka *file* yang pernah dibuka atau dibuat sebelumnya.
- 2) *Create New*, untuk membuat *file* baru.
- 3) *Top Feature*, menampilkan fitur-fitur terbaru dan terpopuler dari *Adobe Dreamweaver CS6*.
- 4) *Getting Started*, *New Features*, untuk melihat fitur-fitur baru, atau tuntutan penggunaan *Adobe Dreamweaver CS6*.

Berikut ini adalah alur kerja pembuatan sebuah *website* dengan menggunakan *Adobe Dreamweaver CS6*:

1. Membuat rancangan dan menentukan lokasi *site*.
2. Melakukan organisasi dan manajemen *file site*.
3. Membuat *layout* halaman *web*.

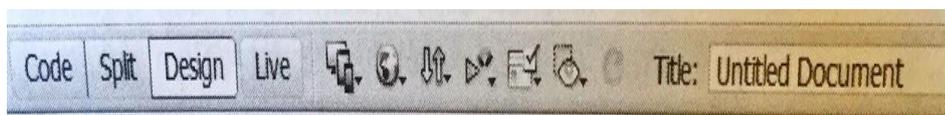
4. Membuat halaman dengan kode program.
5. Mengatur *web* menjadi aplikasi dengan isi yang dinamis.

Setelah *Adobe Dreamweaver CS6* berhasil dijalankan, pilih file baru (*create new*) dengan tipe HTML di dalam kotak dialog *New Document* yang tampil.



Sumber: (Andi, 2013: 5)

**Gambar 2.6** Tampilan ruang kerja *Adobe Dreamweaver CS6* dan elemen-elemennya



Sumber: (Andi, 2013: 5)

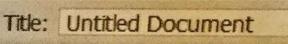
**Gambar 2.7** Tampilan *toolbar Document*

Berikut keterangan fungsi tombol dalam *toolbar document*:

**Tabel 2.5** Keterangan Fungsi Tombol dalam *toolbar Document*

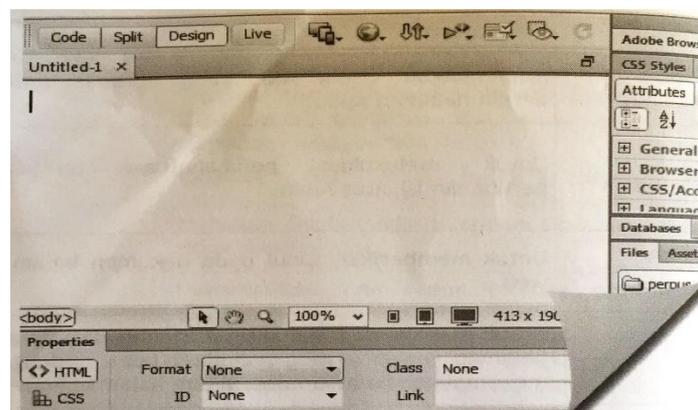
No.	Tombol	Keterangan
1.	 <i>Code</i>	Untuk mengubah tampilan jendela dokumen dalam mode pengetikan kode HTML.
2.	 <i>Split</i>	Untuk mengubah tampilan jendela dokumen menjadi kombinasi antara tampilan <i>Code</i> dan <i>Design</i> .
3.	 <i>Design</i>	Untuk mengubah tampilan jendela dokumen menjadi tampilan <i>layout</i> desain. Apa yang anda lihat pada tampilan ini akan sama seperti tampilan pada jendela <i>browser</i> . <b>Catatan:</b> jika Anda bekerja dengan XML, <i>JavaScript</i> , CSS, atau <i>file</i> tipe kode lainnya, tampilan <i>Design</i> dan tampilan <i>Code and Design</i> tidak ditampilkan.
4.	 <i>Live</i>	Menampilkan dan menguji kode aktual yang digunakan oleh <i>browser</i> untuk menjalankan halaman. Pada saat tombol diklik, maka <i>toolbar Document</i> akan berubah tampilannya dengan menambahkan <i>Live Code</i> dan <i>Inspect</i> .
5.	 <i>Multiscreen</i>	Untuk memilih jenis-jenis ukuran <i>web</i> halaman dimana tersebut akan dikunjungi atau ditampilkan. Sebagai contoh, jika halaman <i>web</i> Anda akan dikunjungi atau ditampilkan di dalam media <i>Smart Phone</i> maka akan berbeda ukurannya saat halaman tersebut ditampilkan di media <i>Tablet</i> atau <i>dekstop</i> .
6.	 <i>Preview/Debug in Browser</i>	Untuk melihat hasil halaman Anda dalam <i>browser</i> yang dikenali oleh <i>Dreamweaver</i> dan digunakan untuk mencari kesalahan <i>script</i> dalam dokumen.

**Tabel 2.5** Lanjutan

		Sebagai contoh <i>Firefox</i> , <i>Chrome</i> , <i>Iexplore</i> , dan <i>Adobe Browser Lab</i> .
7.	 <i>File Management</i>	Digunakan untuk manajemen <i>file</i> baik dalam <i>site</i> lokal, <i>testing server</i> , maupun <i>remote server</i> , diantaranya adalah proses <i>upload</i> dan <i>download</i> .
8.	 <i>W3C Validation</i>	Digunakan untuk memvalidasi dokumen <i>web</i> baik HTML maupun XHTML yang Anda buat menggunakan layanan dari W3C ( <i>World Wide Web Consortium</i> ).
9.	 <i>Check Browser Compatibility</i>	Untuk melakukan pengecekan kesalahan pada dokumen <i>web</i> .
10.	 <i>Visual Aids</i>	Menampilkan <i>visual Aids</i> yang berbeda untuk desain halaman Anda.
11.	 <i>Refresh Design View</i>	Untuk melakukan pembaharuan terhadap perubahan isi situs Anda.
12.	 <i>Title: Untitled Document</i> Document Title	Untuk memberikan judul pada dokumen halaman <i>web</i> .

**Sumber:** (Andi, 2013: 5-7)

- Jendela Dokumen**, adalah lembar kerja tempat Anda membuat dan mengedit desain halaman *web*.



**Sumber:** (Andi, 2013: 8)

**Gambar 2.8** Jendela Dokumen

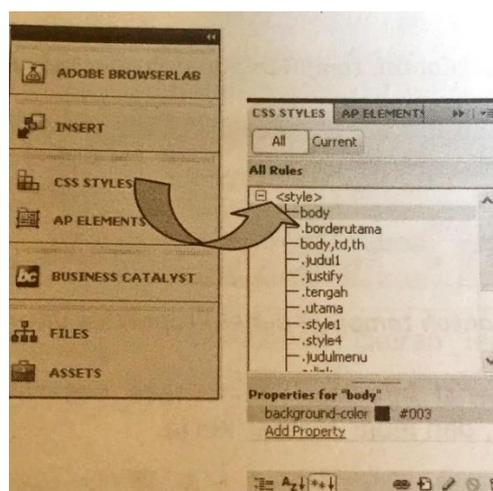
2. **Workspace Switcher**, digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Adobe Dreamweaver CS6*. Sebagai contoh mengubah tampilan menjadi tampilan *classic*, yaitu tampilan ruang kerja *Adobe Dreamweaver* versi sebelumnya.



Sumber: (Andi, 2013: 8)

**Gambar 2.9** Tombol perintah untuk memilih jenis tampilan ruang kerja

3. **Panel Groups** adalah kumpulan *panel* yang saling berkaitan, *panel-panel* ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. *Panel* ini digunakan untuk memonitor dan memodifikasi pekerjaan.



Sumber: (Andi, 2013: 9)

**Gambar 2.10** Contoh tampilan *Group Panel* dengan *CSS Styles* yang terbuka

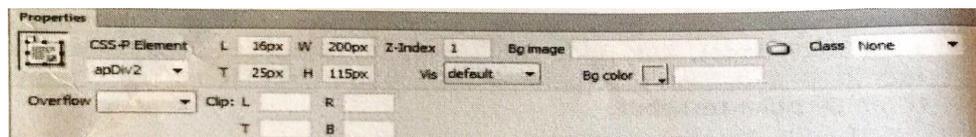
4. **Tag Selector**, berada di pojok kiri bawah jendela dokumen, satu baris dengan *Status Bar*. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut.



Sumber: (Andi, 2013: 9)

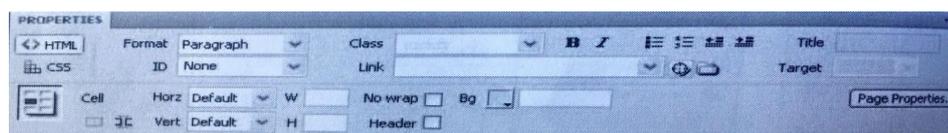
**Gambar 2.11** Contoh tampilan *Tag Selector*

5. **Property Inspector**, biasa juga disebut dengan *panel properties*. Digunakan untuk melihat dan mengubah berbagai objek atau teks pada jendela *design*.



Sumber: (Andi, 2013: 10)

**Gambar 2.12** Contoh tampilan *panel Properties* untuk AP Div



Sumber: (Andi, 2013: 10)

**Gambar 2.13** Contoh tampilan *panel Properties* untuk sel dan isinya

6. **Status Bar**, berisi informasi persentase besarnya tampilan, jenis *pointer mouse*, dan lebar lembar kerja.



Sumber: (Andi, 2013: 10)

**Gambar 2.14** Tampilan *Status Bar*

Berikut keterangan tombol dalam *Status Bar*:

**Tabel 2.6** Keterangan tombol dalam *Status Bar*

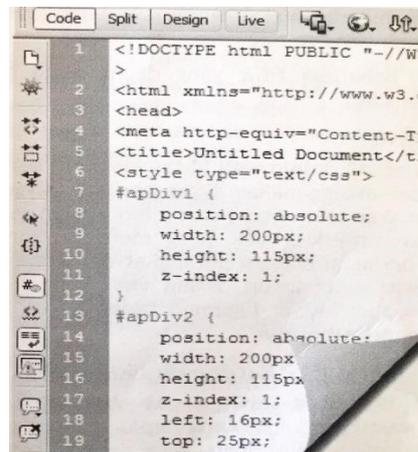
No.	Tombol	Keterangan
1.	<b>[A]</b> <i>Tag Selector</i>	Menunjukkan hirarki ruang lingkup dari pekerjaan yang sedang terpilih. Klik salah satu hirarki maka semua isinya akan terpilih. Sebagai contoh, klik tag <b>&lt;body&gt;</b> maka seluruh isi halaman <i>web</i> akan terpilih.
2.	<b>[B]</b> <i>Select Tool</i>	Untuk mengaktifkan dan menonaktifkan <i>Hand tool</i> . Fungsi lain adalah mengaktifkan <i>pointer mouse</i> untuk seleksi objek.
3.	<b>[C]</b> <i>Hand Tool</i>	Untuk menggeser atau menggulung jendela dokumen, dengan cara klik tahan geser menggunakan <i>mouse</i> .
4.	<b>[D]</b> <i>Zoom Tool</i>	Untuk memperbesar ukuran tampilan jendela dokumen. Gunakan kombinasi <b>Alt</b> + klik untuk mengurangnya.
5.	<b>[E]</b> <i>Set Magnification</i>	Untuk mengisikan nilai persentase besarnya tampilan jendela dokumen.
6.	<b>[F]</b> <i>Mobile Size</i>	Menunjukkan ukuran lembar kerja dalam ukuran <i>mobile</i> .
7.	<b>[G]</b> <i>Tablet Size</i>	Menunjukkan ukuran lembar kerja dalam ukuran <i>tablet</i> .

**Tabel 2.6** Lanjutan

8.	<b>[H]</b> <i>Dekstop Size</i>	Menunjukkan ukuran lembar kerja dalam ukuran dekstop.
9.	<b>[I]</b> <i>Window Size</i>	Menunjukkan ukuran lembar kerja yang sekarang aktif.

**Sumber:** (Andi, 2013: 10-11)

7. **Toolbar Coding**, berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. *Toolbar* ini hanya tampil pada jendela **Code**.



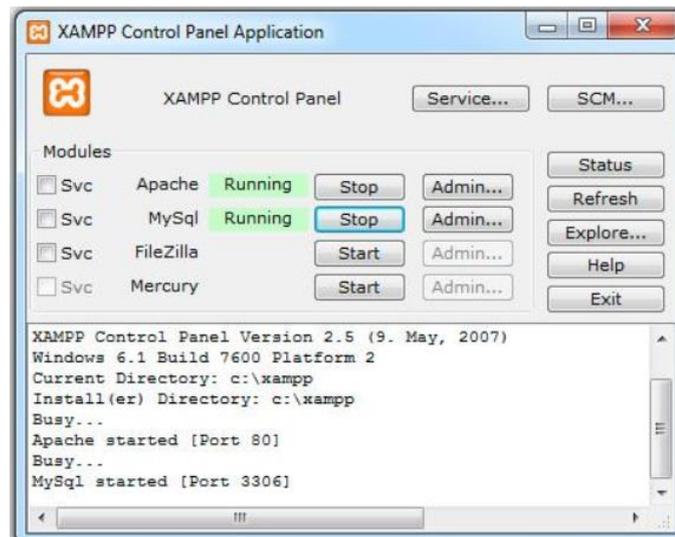
**Sumber:** (Andi, 2013: 11)

**Gambar 2.15** Tampilan *toolbar Coding*

### 2.2.10. XAMPP

Menurut (Fadly, Triyanto, & Ruslianto, 2015: 14), XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi dari XAMPP adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP*

dan Perl. Nama XAMPP merupakan singkatan dari X (sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.



Sumber: (Fadly et al., 2015: 15)

**Gambar 2.16** Tampilan XAMPP Control Panel

### 2.2.11. Penelitian Terdahulu

Penelitian terdahulu dari penelitian ini adalah sebagai berikut:

**Tabel 2.7** Daftar Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Penelitian
1.	Hendra Gunawan dan Irwan Agustian	Aplikasi Pelayanan Gangguan Listrik Berbasis web di PLN Rayon Banjaran	Waterfall	User dapat melihat dan mengupdate pelayanan gangguan listrik secara online.

**Tabel 2.7** Lanjutan

2.	Tukino dan Amrizal	Perancangan Sistem Informasi Pelaporan Transaksi Berbasis Web Pada PT Pos Indonesia Batam	Pendekatan pengembangan sistem (studi kelayakan dan rancangan sistem)	Sistem informasi pelaporan transaksi dengan menggunakan Pemrograman berbasis web pada PT Pos Indonesia (Persero) Batam 29400 dimana sistem ini dapat lebih mudah diterapkan dan lebih mudah dipahami oleh user sehingga dapat menghasilkan informasi yang cepat, tepat dan akurat.
3.	Indra Griha Tofik Isa dan George Pri Hartawan	Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis Web (Studi Kasus Koperasi Mitra Setia)	Berbasis objek	Aplikasi koperasi berbasis web diharapkan dapat membantu para anggotanya untuk bertransaksi dan mempermudah dalam melakukan pencatatan keuangan.
4.	Syarifudin Ramadhani	Sistem Informasi Pelayanan (BPJS) Berbasis Web	Berbasis objek	Untuk mengetahui bagaimana Sistem Informasi Pelayanan Askes di Rumah Sakit Umum Daerah (RSUD) Kabupaten Lamongan.

**Tabel 2.7** Lanjutan

5.	Tukino	Perancangan Sistem Informasi Manajemen Proyek Pengaksesan Dokumen Perakitan PCBA di PT Surya Teknologi Batam Berbasis Web	Waterfall	Sistem informasi pengelolaan dokumen di PT. Surya Teknologi Batam berbasis web telah sesuai dengan kebutuhan fungsional sistem.
6.	Lukman Arif Sanjani, Sulis Janu Hartati, dan Pantjawati Sudarmaningtyas	Rancang Bangun Sistem Informasi penggajian Pegawai dan Remunerasi Jasa Medis Pada Rumah Sakit Bedah Surabaya	Waterfall	Meningkatkan efisiensi kinerja departemen HRD dalam melakukan proses penghitungan gaji.
7.	Afrizal dan Fitriani	Pengembangan Sistem Informasi Feedback Pelanggan Hosting Pada CV. E-Padi Corporation Berbasis Web	Waterfall	Merancang dan membangun suatu sistem informasi feedback pelanggan hosting berbasis komputer yang sistematis, terstruktur dan terarah pada CV.
8.	Mansur dan Kasmawi	Pengembangan Sistem Database Terpadu Berbasis Web Untuk Penyediaan Layanan Informasi Website Desa	Pendekatan prototype	Prototype database terpadu berbasis web untuk penyediaan informasi website desa.

**Tabel 2.7** Lanjutan

9.	Fábio Paulo Basso, Raquel Mainardi Pillat, Toacy Cavalcante Oliveira, Fabricia Roos-Frantz, Rafael Z. Frantz	Automated design of multi-layered web information systems	Prototypes	<i>Assist the production of models by means of transformation tasks and refinements.</i> Artinya: Membantu produksi model dengan cara melakukan transformasi tugas dan penyempurnaan.
10.	Rudi Hermawan, Arief Hidayat, dan Victor Gayuh Utomo	Sistem Informasi Penjadwalan Kegiatan Belajar Mengajar Berbasis Web (Studi Kasus : Yayasan Ganesha Operation Semarang)	SDLC (System Development Life Cycle)	Sistem informasi penjadwalan kegiatan belajar mengajar berbasis web yang diharapkan dapat membantu bagian akademik Yayasan Ganesha Operation Semarang dalam mengolah data jadwal secara lebih akurat sebagai pedoman atau panduan pengajar dan siswa.