

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rancang Bangun**

Menurut Zulfiandi (2014), kata “rancang” merupakan kata sifat dari “perancangan” yakni merupakan serangkaian prosedur untuk menerjemahkan hasil analisis dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. proses menyiapkan spesifikasi yang terperinci untuk mengembangkan sistem yang baru. Kata “bangun” merupakan kata sifat dari “pembangunan” adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

#### **2.2 Media Promosi**

Zulyadi (2017) menjelaskan, promosi adalah sejenis komunikasi yang memberi penjelasan yang meyakinkan calon konsumen tentang barang dan jasa. Tujuan utama promosi untuk memberi informasi, menarik perhatian dan selanjutnya memberi pengaruh peningkatan penjualan dengan cara meyakinkan calon konsumen.

#### **2.3 Virtual Tour**

Yuliana (2017) menjelaskan, Virtual Tour (atau disebut juga panoramic tour) adalah sebuah simulasi dari suatu tempat yang benar-benar ada, biasanya terdiri dari kumpulan foto-360 foto panorama, kumpulan gambar yang terhubung oleh hyperlink, ataupun video, Bisa juga menggunakan unsur-unsur multimedia lainnya seperti efek suara, musik, narasi, dan tulisan. Berbeda dengan *tour* sebenarnya, *Virtual Tour* biasanya diakses melalui komputer *desktop*, Kios informasi atau media elektronik lainnya. Istilah "*panoramic tour*" dan "*virtual tour*" sering digunakan untuk menggambarkan berbagai macam video dan media

berbasis fotografi. Kata "panorama" mengindikasikan sebuah pandangan yang tidak terputus, karena panorama bisa berupa sekumpulan foto memanjang ataupun hasil pengambilan video yang kameranya berputar/bergeser.

Menurut Daud (2016), *virtual tour* merupakan teknologi yang menempatkan user di dalam gambar dan memungkinkan user untuk meningkatkan kesadaran situasional serta meningkatkan daya lihat, tangkap dan menganalisa data *virtual* secara signifikan. *Virtual Tour* merupakan sebuah simulasi dari sebuah lokasi yang terdiri dari rentetan. Rentetan gambar tersebut akan digabungkan (*stitch*) untuk menghasilkan foto panorama 360 derajat. *Virtual Tour* sendiri biasanya digunakan untuk memberi pengalaman 'pernah berada' di suatu tempat hanya dengan melihat layar monitor. Penyajian virtual tour dapat dilakukan dengan cara memanfaatkan gambar ataupun video, selain itu dapat menggunakan model 3dimensi. Untuk penyajian dengan menggunakan gambar, dapat digunakan foto panorama. Pemilihan jenis foto panorama juga mempengaruhi hasil virtual tour yang dihasilkan. Untuk panorama jenis *cylindrical*, bagian vertikalnya hanya dapat menangkap tidak lebih dari 180.

#### **2.4 HTML5**

Menurut Chumairoh (2014) *HTML5* adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi dari *WWW (world wide web)* sebuah teknologi inti dari internet. *HTML5* adalah revisi kelima dari *HTML* yang pertama kali diciptakan pada tahun 1990. *HTML5* merupakan salah satu karya *World Wide Web Consortium (W3C)* untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara *HTML* ataupun *XHTML*. Salah satu kelebihan dari *HTML5* adalah mendukung untuk penyimpanan secara offline.

Tanuwijaya (2017) menjelaskan, HTML 5 merupakan revisi kelima dari HTML yang telah dikembangkan untuk memperbaiki teknologi pada HTML agar mendukung teknologi multimedia dan grafis pada web browser tanpa tergantung pada jenis browser tertentu.

## 2.5 Javascript

Rohim (2015) menjelaskan *JavaScript* adalah bahasa pemrograman *website* yang bersifat *CSPL* atau *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti *Google Chrome* dan *Mozilla Firefox*. Jenis bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman *Server Side* seperti *PHP*, dimana untuk *server side* seluruh kode program dijalankan di sisi *server*. Untuk menjalankan *JavaScript*, kita hanya membutuhkan aplikasi text editor, dan web browser. *JavaScript* memiliki fitur: *high-level programming language*, *client-side*, *loosely typed*, dan berorientasi objek. *JavaScript* pada awal perkembangannya berfungsi untuk membuat interaksi antara user dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di *web server*. Sebelum *javascript*, setiap interaksi dari user harus diproses oleh *web server*. Bayangkan ketika kita mengisi form registrasi untuk pendaftaran sebuah situs, lalu men-klik tombol submit, menunggu sekitar 20 detik untuk *website* memproses isian form tersebut, dan mendapati halaman yang menyatakan bahwa terdapat kolom form yang masih belum diisi. Untuk keperluan seperti inilah *JavaScript* dikembangkan. Pemrosesan untuk mengecek apakah seluruh form telah terisi atau tidak, bisa dipindahkan dari *web server* ke dalam *web browser*. Dalam perkembangan selanjutnya, *JavaScript* tidak hanya berguna untuk validasi *form*, namun untuk berbagai keperluan yang lebih *modern*. Berbagai animasi untuk mempercantik halaman *web*, fitur *chatting*, efek-efek modern, games, semuanya bisa dibuat menggunakan *JavaScript*.

## 2.6 CSS

Nugroho(2017) menjelaskan *Cascading Style Sheet (CSS)* adalah salah satu bahasa untuk mengatur tampilan dalam *web* yang berfungsi memisahkan antara desain dengan *content*. *CSS* dapat ditulis untuk mendefinisikan ulang tag-tag *HTML* ataupun dengan menambah atribut *id* dan *class* pada tag *HTML*.

## 2.7 *Unified Modeling Language (UML)*

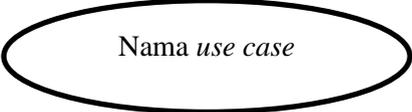
Menurut Rosa (2013), *Unified Modelling Language* atau biasa disingkat *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung. *UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek. Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan *syntax*/semantik. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

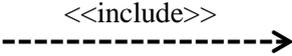
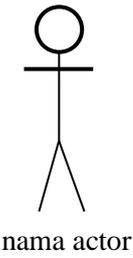
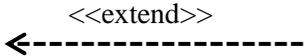
### 2.7.1 *Use Case Diagram*

*Diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Rosa dan M. Shalahuddin (2013).

Simbol-simbol yang ada pada *use case* diagram dapat dilihat pada tabel 2.1

**Tabel 2.1** Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya di nyatakan dengan menggunakan kata kerja di awal frase <i>name use case</i>.</p>
<p>Asosiasi/<i>assosiation</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

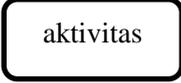
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan <i>include</i></p> 	<p>Relasi <i>use case</i> tambahkan ke subbuah <i>use case</i> dimana <i>use case</i> ditambahkan memerlukan <i>use case</i> ini menjalankan fungsinya atau syarat di jalankan <i>use case</i> ini.</p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu menggunakan orang; biasanya di nyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Ektensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang di tambahkan misal arah panah mengarah pada <i>use case</i> yang ditambahkan</p>

### 2.7.2. Activity Diagram

Menurut Rosa A.S dan M. Shalahuddin (2013), Diagram aktivitas menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem sistem.

Simbol-simbol yang ada pada *activity* diagram dapat dilihat pada tabel 2.2

**Tabel 2.2** Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memilih sebuah status awal.
Aktivitas 	Aktivitas yang di lakukan sistem aktivitas biasanya diawali dengan kata kerja
Percabangan/decision 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari
Penggabungan/join 	Asosiasi penggabungan dimana lebih satu aktivitas di gabung menjadi satu
Status akhir 	Status akhir yang di lakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir

## 2.8 Metode Pengumpulan Data

Tamagola (2017) menjelaskan, metode pengumpulan data ini dilakukan untuk memudahkan dalam mendapatkan data yang diperlukan. Berikut beberapa metode pengumpulan data yang digunakan :

### a. Observasi

Menurut Tamagola (2018) , observasi (Pengamatan Langsung) merupakan salah satu teknik pengumpulan data yang tidak hanya mengukur sikap dari responden (wawancara dan angket) namun juga dapat digunakan untuk merekam berbagai fenomena yang terjadi (situasi, kondisi).

### b. Wawancara

Amnah (2016) menjelaskan, teknik wawancara merupakan teknik pengumpulan data atau fakta yang dilakukan dengan cara menanyakan langsung kepada bagian yang terkait sesuai yang dibutuhkan dalam proses penelitian skripsi.

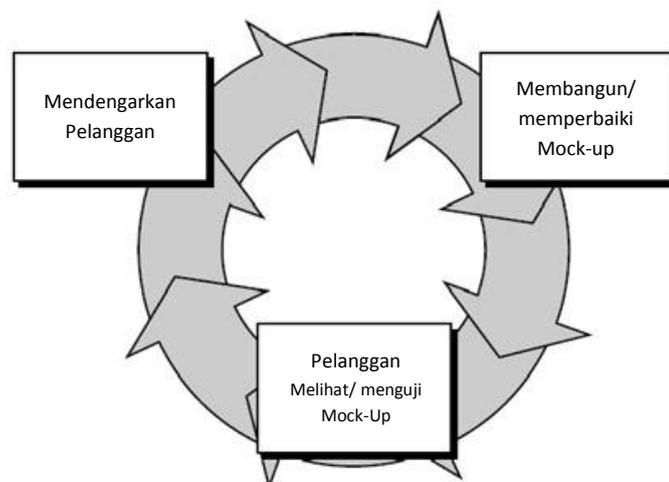
### c. Studi Pustaka

Amnah (2016) menjelaskan, studi pustaka mempelajari buku-buku serta literatur-literatur yang ada pada perpustakaan, mempelajari beberapa alur karya ilmiah yang berkaitan dengan judul yang diangkat, dan mempelajari bentuk-bentuk data pengolahan data sebagai dasar informasi.

## 2.9 Metode Pengembangan Perangkat Lunak

### 2.9.1 Prototipe

Menurut Rosa A.S. dan M. Shalahuddin(2013), model prototipe (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini dievaluasi oleh pelanggan atau *user*. Model prototipe dapat dilihat pada gambar 2.1



**Gambar 2.1** Ilustrasi Model Prototipe (Sumber:Rosa A.S. dan M. Shalahuddin(2013))

*Mock-up* adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak. Iterasi terjadi pada pembuatan prototipe sampai sesuai dengan keinginan pelanggan (*customer*) atau *user*.

Proses pada *model prototyping* dapat dijelaskan sebagai berikut:

- a. Mendengarkan Pelanggan, Tahap pertama dari metode ini adalah mendengarkan pelanggan yaitu mengumpulkan data – data mengenai kebutuhan sistem yang akan dibuat.
- b. Membangun/memperbaiki mock-up, setelah mendapatkan kebutuhan yang dibutuhkan untuk membuat prototipe, tahap berikutnya yang dilakukan adalah membangun/memperbaiki prototipe tahap ini dilakukan dengan pembuatan sistem secara keseluruhan hingga selesai berdasarkan pada analisa kebutuhan yang sudah dilakukan sebelumnya.
- c. Pelanggan menguji coba prototipe, tahap akhir dari model prototipe ini adalah tahap pengujian. Pengujian dilakukan oleh pengguna dari sistem. Tahap pengujian dilakukan untuk mendapatkan tanggapan atas sistem yang telah dibuat.

Menurut Purnomo (2017), Prototipe dapat diterapkan pada pengembangan sistem kecil maupun besar dengan harapan agar proses pengembangan dapat berjalan dengan baik, tertata serta dapat selesai tepat waktu. Keterlibatan pengguna secara penuh ketika protipe terbentuk akan menguntungkan seluruh pihak yang terlibat, bagi pimpinan, pengguna sendiri serta pengembang sistem. Manfaat lainnya dari penggunaan prototyping adalah :

- a. Mewujudkan sistem sesungguhnya dalam sebuah replika sistem yang akan berjalan, menampung masukan dari pengguna untuk kesempurnaan sistem.
- b. Pengguna akan lebih siap menerima setiap perubahan sistem yang berkembang sesuai dengan berjalannya prototipe sampai dengan hasil akhir pengembangan yang akan berjalan nantinya.

- c. Prototype dapat ditambah maupun dikurangi sesuai berjalannya proses pengembangan. Kemajuan tahap demi tahap dapat diikuti langsung oleh pengguna.
- d. Penghematan sumberdaya dan waktu dalam menghasilkan produk yang lebih baik dan tepat guna bagi pengguna.

## **2.10 Metode Pengujian Perangkat Lunak**

### **2.10.1 Blackbox Testing**

Menurut Purwati (2018) *black box testing* adalah pengujian 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. Pengujian *black box* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.