

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Perancangan**

Definisi perancangan menurut jacob nielsen “pada dasarnya ada 2 pendekatan yang bersifat mendasar pada perancangan perangkat lunak :idealisme *artistic* yang mengekspresikan diri anda sendiri dan idealisme rekayasa yang bertujuan untuk menyelesaikan permasalahan-permasalahan yang ada di arah pelanggan.” (Pressman, 2012)

Dengan demikian pengertian perancangan merupakan kegiatan yang dilakukan untuk menyelesaikan permasalahan-permasalahan yang ada sesuai kebutuhan pengguna.

#### **2.2 E-service**

Layanan elektronik (*e-service*) adalah istilah yang sangat generik, biasanya mengacu pada "penyediaan layanan melalui internet (awalan berdiri 'e elektronik', seperti dalam penggunaan lain), sehingga layanan elektronik bisa juga termasuk perdagangan internet, mungkin juga termasuk layanan non-komersial (online), yang biasanya disediakan oleh pemerintah.

Menurut Yuliana pratiwi (2013) dalam jurnalnya, *e-service* adalah perbuatan, usaha atau pertunjukan yang pengiriman di mediasi oleh teknologi informasi. Layanan elektronik tersebut meliputi unsur layanan *e-tailing*, dukungan pelanggan, dan pelayanan.

*E-service* merupakan gabungan antara pelayanan dan elektronik. *E-service* adalah langkah yang tepat untuk mengatasi kesulitan dari pelayanan tradisional yang kendalanya sering ada di terbatasnya fasilitas dari perusahaan itu sendiri. Dari sisi keefiesianan waktu, *e-service* bisa diakses lebih fleksibel bagi pelanggan dimanapun dan waktu respon dari perusahaan yang tergolong cepat.(yuliana pratiwi, 2013: 206-214)

### 2.3 Terintegrasi

Integrasi adalah adanya saling keterkaitan antar sub sistem sehingga data dari satu sistem secara rutin dapat melintas, menuju atau diambil oleh satu atau lebih sistem yang lain.

Sistem terintegrasi (*integrated system*) merupakan sebuah rangkaian proses untuk menghubungkan beberapa sistem-sistem komputerisasi dan software aplikasi baik secara fisik maupun secara fungsional. Sistem terintegrasi akan menggabungkan komponen sub-sub sistem ke dalam satu sistem dan menjamin fungsi-fungsi dari sub sistem tersebut sebagai satu kesatuan sistem.

Integrasi adalah menyatukan sistem informasi yang terpisah satu sama lain untuk dijadikan satu agar inter unit dan antar unit bisa saling terhubung serta dapat mengakses data secara terpusat (*server*) untuk dapat memenuhi kebutuhannya. Integrasi mengurangi duplikasi. Salah satu faktor penting untuk integrasi, terutama antar bisnis. (handoyo widi nugroho, 2011).

### 2.4 Algoritma *Brute Force*

Algoritma *brute force* adalah salahsatu cara penyelesaian masalah khususnya di dalam dunia komputer. *Brute force* adalah sebuah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan.

Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*). Algoritma yang secara jelas langsung ke pusat permasalahan. Algoritma ini biasanya tidak memerlukan teori khusus untuk mengimplementasikannya. Algoritma ini sering juga disebut algoritma sapu jagad karena hampir semua persoalan pemrograman bisa diselesaikan dengan algoritma ini.

Algoritma *brute force* sering disebut algoritma tidak cerdas karena tidak memerlukan pemikiran secara kompleks, hanya dengan beberapa baris semua permasalahan dapat dipecahkan dengan mudahnya.

Kelemahan algoritma *brute force* yaitu lemah dalam sisi penggunaan memori pada komputer jika data yang diinginkan berada dalam titik tengah ke atas. Yaitu titik dimana ketika data yang akan dicari berjumlah 10.000 dan *brute force* melakukan perulangan pencarian data lebih dari  $10.000/2 = 5.000$  kali. *Brute force* menjadi sangat efektif ketika data yang akan dicari berada pada titik tengah ke bawah. (Affandi & Wahyuningsih, 2017)

## **2.5 Web Mobile**

*Mobile Web* merupakan platform yang paling mudah untuk dipelajari, paling murah untuk diproduksi, terstandarisasi, yang paling tersedia, dan paling mudah untuk didistribusikan; sesuai dengan prinsip *Ubiquity*. Mobile web juga satu-satunya platform yang tersedia dan mampu berjalan pada semua perangkat mobile, menggunakan satu set standar dan protokol yang sama dengan desktop web. (Indriasari, Adi, & Sidhi, 2011)

## **2.6 Perangkat Lunak Yang Digunakan**

Untuk membangun *website* diperlukan beberapa perangkat lunak yang digunakan dalam membangun *website* tersebut. Beberapa perangkat lunak yang digunakan adalah sebagai berikut :

### **2.6.1 Sublime Text 3**

Sublime text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi python. terciptanya aplikasi ini terinspirasi dari aplikasi vim, aplikasi ini sangatlah fleksibel dan powerful. sublime text bukanlah aplikasi opensource dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

Sublime text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; *c, c++, c#, css, d, dylan, erlang, html, groovy, haskell, java, javascript, latex, lisp, lua, markdown, matlab, ocaml, perl, php, python, r, ruby, sql, tcl, textile and xml*. Biasanya bagi bahasa pemrograman yang didukung ataupun belum didukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user. (Faridl, 2015)

### **2.6.2 Framework Bootstrap**

Menurut (Alatas, 2013) *bootstrap* merupakan *framework* ataupun *tools* untuk membuat aplikasi *website* ataupun situs *web responsive* secara cepat, mudah dan gratis, karena *website* yang dibangun oleh peneliti merupakan *website* yang dapat diakses dalam perangkat *mobile* ataupun *personal computer*. *Bootstrap* terdiri dari *css* dan *html* untuk menghasilkan *grid, layout, typography, table, form, navigation*, dan lain lain. Di dalam *bootstrap* juga sudah terdapat *jquery plugins* untuk menghasilkan komponen ui yang cantik seperti *transitions, modal, dropdown, scrollspy, tooltip, tab, popever, alert, button, carousel*, dan lain lain.

### **2.6.3 Xampp**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya 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 XAMP merupakan singkatan dari X (empat system 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. (Priyanti, 2013)

## **2.7 Basis Data**

### **2.7.1 Definisi Basis Bata**

Menurut (Rosa & Shalahudin, 2015) mengatakan basis data (*database*) adalah system terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.pada intinya basis data adalah media untuk menyimpan data agara dafapat diakses dengan mudah dan cepat.

### **2.7.2 Tujuan basis data**

#### **1. Kecepatan dan kemudahan (*speed*)**

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.

#### **2. Efisiensi ruang penyimpanan(*space*)**

Karena keterkaitan erat anantara kelompok dalam basis data, maka redudansi data pasti selalu ada. Dengan basis data,efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan karena kita dapat melakukan penekanan jumlah redudansi data, baik menerapkan sejumlah pengkodean atau membuat relasi-relasi anatar kelompok data yang saling berhubungan.

#### **3. Keakuratan (*accuracy*)**

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data,domain data,keunikan data dan sebagainya yng secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan/penyimpanan data.

#### **4. Ketersediaan ( *availability*)**

Pertumbuhan data sejalan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data selalu kita gunakan/butuhkan. Karena itu kita dapat melakukan pemilihandata, sehingga data yang sudah jarang kita gunakan dapat kita pindahkan kedalam media penyimpanan offline.

### 5. Kelengkapan (*completeness*)

Untuk mengkomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam penambahan objek baru (tabel) atau dengan penambahan *field-field* baru pada suatu tabel.

### 6. Keamanan (*security*)

Ada sejumlah sistem pengolahan basis data yang tidak menerapkan aspek keamanan dalam sebuah basis data.

### 7. Kebersamaan pemakain (*sharebility*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja atau di satu lokasi saja oleh satu sistem aplikasi.

## 2.8 Penelitian Terdahulu

Berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan *eletronik service*

**Table 2.1** penelitian terdahulu

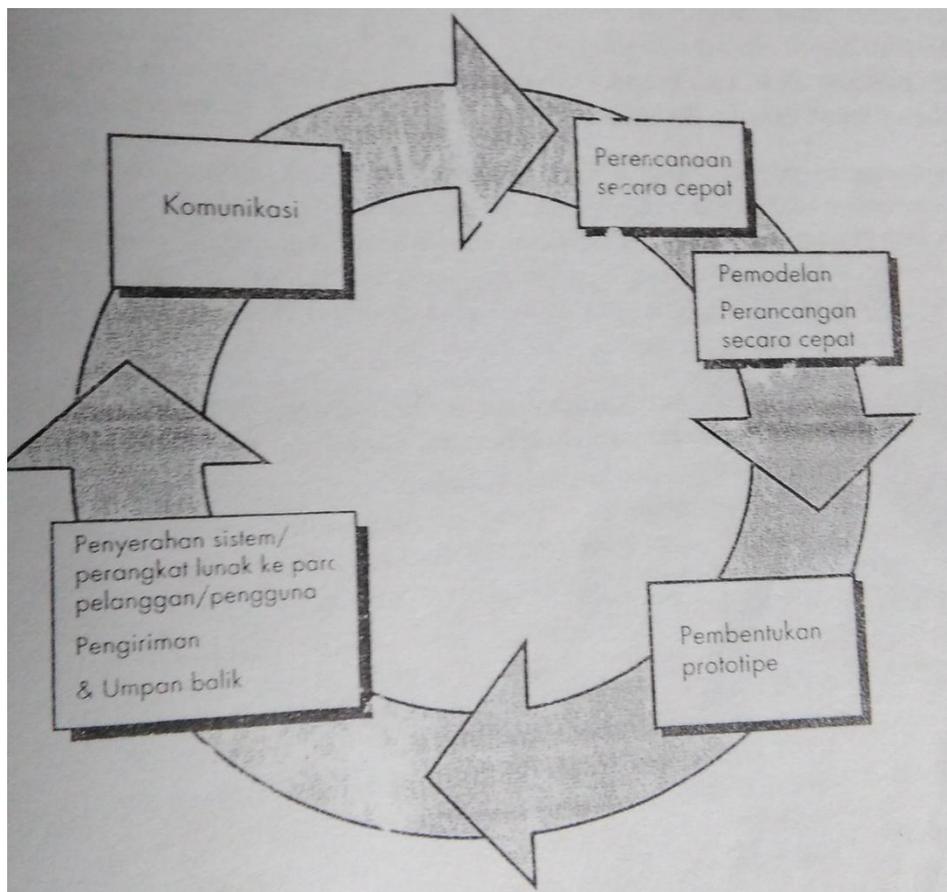
<b>Nama</b>	<b>Judul</b>	<b>Terbit / tahun</b>	<b>Keterangan</b>
Budi santosa, dessyanto boedi prasetyo ,yunita pungi	Integrasi toko online menggunakan teknologi <i>webservice</i>	<i>E-commerce</i> , <i>webservice</i> , php,2011	Saat ini banyak terdapat toko <i>online</i> untuk memudahkan penjual mempromosikan produk-produknya sekaligus memperluas ruang lingkup kerjanya. Amazon, ebay, dan commission junction merupakan contoh <i>webstore</i> .

			<p>Kadang pembeli agak repot jika harus mengakses halaman mereka satu persatu. Untuk itu dibuatlah suatu aplikasi penjualan yang berbasis ti dan internet (<i>e-commerce</i>) dengan menggunakan teknologi <i>webservice</i> sehingga produk yang ada pada situs penjualan ini terdiri atas beberapa produk dari <i>webstore</i> amazon, ebay, dan <i>commission junction</i>, sehingga akan mengurangi masalah integrasi data dan calon pembeli dapat dengan mudah membandingkan berbagai macam produk dan harga yang ditawarkan dalam sebuah segmen pasar.</p>
--	--	--	--

## 2.9 Metode Pengembangan Perangkat Lunak

### 2.9.1 Metode *Prototype*

Pembuatan *prototype* menurut Pressman (2012: 50-52) dalam bukunya seringkali pelanggan mendefinisikan sejumlah sasaran perangkat lunak secara umum, tetapi tidak bisa mengidentifikasi spesifikasi kebutuhan yang rinci untuk fungsi-fungsi dan fitur-fitur yang nantinya yang akan dimiliki perangkat lunak yang akan dikembangkan. Tahap *prototype* dapat dilihat pada gambar 2.1 berikut ini:



Gambar 2.1 prototype

Pembuatan prototype dimulai dengan dilakukannya komunikasi antara tim pengembang perangkat lunak dengan para pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan para stakeholder untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan,

mengidentifikasi spesifikasi kebutuhan yang saat ini diketahui menggambarkan area-area dimana definisi lebih jauh pada iterasi selanjutnya merupakan keharusan. meski demikian, pembuatan prototipe bisa saja menimbulkan masalah untuk berbagai alasan berikut ini.

1. Para stakeholder melihat tampilan dari perangkat lunak yang akan mereka pakai kelak, tidak peduli bahwa sesungguhnya prototipe pada umumnya tidak dirancang secara seksama tidak peduli pada masalah bagaimana sistem itu kelak akan bekerja dengan baik karena pengembang pada umumnya belum mempertimbangkan aspek pemeliharaan sistem dalam jangka panjang.
2. Sebagai rekayasawan perangkat lunak, kita sering membuat implementasi sedemikian rupa sehingga kita bisa mendapatkan prototipe jadi dengan cepat.

## **2.10 Unified Modeling Language (UML)**

Verdi yasin (2012) mendefinisikan *Unified Modelling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, uml menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan penggunaan uml yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

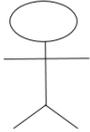
Menurut verdi yasin (2012) tipe-tipe diagram uml adalah sebagai berikut :

### *1) Use case diagram*

*Use case diagram* adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mepresentasikan sebuah interaksi antara actor dan sistem.

Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *case relationship*. Penjelasan simbol pada tabel 2.2.

**Tabel 2.2** Simbol Use Case

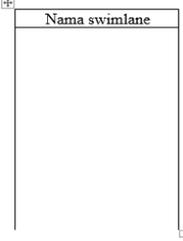
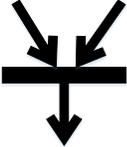
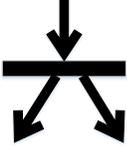
Simbol	Keterangan
	Aktor : seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<i>Use case</i> : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.
	<i>Association</i> : adalah relasi antara actor dan <i>use case</i> .
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.

## 2) *Activity diagram*

*Activity diagram* menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity diagram* berupa *flow chart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut :

**Tabel 2.3** Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Activity</i> : memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial node</i> : bagaimana objek dibentuk atau diawali

	<i>Activity final node</i> : bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	<i>Swimlane</i> : memisalkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	<i>Join</i> : digunakan untuk menunjukkan kegiatan yang digabungkan.
	<i>Fork</i> : digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel

### 3) *Sequence diagram*

*Sequence diagram* menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.

Dalam *sequence diagram* terdapat 2 simbol yaitu :

- a. *Actor*, untuk menggambarkan pengguna sistem.
- b. *Lifeline*, untuk menggambarkan kelas dan objek.

### 4) *Class diagram*

*Class diagram* menggambarkan dstruktur data dan desripsi *class*, *package*, dan objek beserta hubungan satu sama lain. *Class diagram* berfungsi untuk menjelaskan tipe

dari objek sistem dan hubungannya dengan objek yang lain. *Class* memiliki 3 area pokok yaitu nama, atribut dan metode.

### **2.11 Black Box Testing**

Menurut (Rosa & Shalahudin, 2015) *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. Kasus uji yang dibuat untuk melakukan pengujian *black box* harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

1. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tetapi kata sandi salah, atau sebaliknya, atau keduanya salah.