



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Pernikahan

Terdapat beragam pendapat dari para ahli yang menjelaskan tentang pengertian pernikahan. Duvall & Miller (1985) mendefinisikan pernikahan sebagai berikut :

*“Marriage is a socially recognized relationship between a man and a woman that provides for sexual relation, legitimized childbearing and establishing a division of labour between spouses”*

Pernikahan dikenali sebagai hubungan antara pria dan wanita yang memberikan hubungan seksual, keturunan, membagi peran antara suami-istri.

Dalam dalam Undang-Undang No. 1 tahun 1974 Bab I pasal 1, perkawinan diartikan sebagai: “Perkawinan adalah ikatan batin antara laki-laki dan perempuan sebagai suami-istri dengan tujuan membentuk keluarga (rumah tangga) yang bahagia dan kekal berdasarkan ketuhanan Yang Maha Esa.”

Beberapa sumber lain menjelaskan bahwa pernikahan adalah ikatan atau komitmen emosional dan legal antara seorang pria dengan seorang wanita yang terjalin dalam waktu yang panjang dan melibatkan aspek ekonomi, sosial, tanggungjawab pasangan, kedekatan fisik, serta hubungan seksual. (Regan, 2003; Olson & DeFrain, 2006; Seccombe & Warner, 2004)

Berdasarkan beberapa pengertian diatas, peneliti membatasi pengertian pernikahan sebagai ikatan yang bersifat kontrol sosial antara pria dan wanita yang didalamnya diatur mengenai hak dan kewajiban, kebersamaan emosional, juga aktivitas seksual, ekonomi dengan tujuan untuk membentuk keluarga serta mendapatkan kebahagiaan dan kasih berdasarkan ketuhanan Yang Maha Esa.



## 2.2 Teori Judul

### 2.2.1 Pengertian Sistem Pendukung Keputusan

Menurut Bonczek, dkk. (1980) dalam buku *Decision Support Systems and Intelligent System* (Turban, 2005: 137) mendefinisikan sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan). Karakteristik dari sistem pendukung keputusan yaitu:

- a. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
- b. Adanya *interface* manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
- c. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung keputusan yang saling berinteraksi.
- d. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan keputusan.
- e. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
- f. Memiliki dua komponen utama yaitu data dan model.

Menurut Simon ada tiga fase dalam proses pengambilan keputusan diantaranya sebagai berikut:

#### 1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendekteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.



## 2. Design

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan solusi.

## 3. Choice

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplmentasikan dalam proses pengambilan keputusan.

### 2.2.2 Pengertian Jasa

Banyak para pakar pemasaran jasa yang telah mendefinisikan pengertian jasa. Adapun pengertian jasa menurut para pakar sebagai berikut:

Menurut Kotler dalam Lupiyoadi (2014:7) “Jasa adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh satu pihak kepada pihak lain, pada dasarnya tidak berwujud dan tidak mengakibatkan perpindahan kepemilikan apapun. Produksi jasa mungkin berkaitan dengan produk fisik atau tidak.”

Selanjutnya, (Zethaml dan Bitner : 1996) dalam Lupiyoadi (2014:7) memberikan batasan tentang jasa sebagai berikut “*Service is all economic activities whose output is not a physical product or construction is generally consumed at that time it is produced, and provides added value in forms (such as convenience, amusement, comfort or health)*.” “Jasa merupakan semua aktivitas ekonomi yang hasilnya bukan berbentuk produk fisik atau konstruksi, yang umumnya dihasilkan dan dikonsumsi secara bersamaan serta memberikan nilai tambah (misalnya kenyamanan, hiburan, kesenangan, atau kesehatan) konsumen.”

Menurut Mursid (1993:116), “Jasa adalah kegiatan yang dapat diidentifikasi secara tersendiri, pada hakikatnya bersifat tidak teraba, untuk memenuhi kebutuhan dan tidak harus terikat pada penjualan produk atau jasa lain.

Beberapa pengertian tersebut memberikan kesimpulan bahwa jasa adalah kegiatan ekonomi dengan hasil keluaran yang tidak berwujud yang ditawarkan dari penyedia jasa yaitu perusahaan kepada pengguna jasa atau konsumen.



### 2.2.3 Pengertian Pelayanan

Pelayanan menurut Kamus Besar Bahasa Indonesia (KBBI) adalah sebagai suatu usaha untuk membantu menyiapkan atau mengurus apa yang diperlukan orang lain.

Sementara itu, menurut Moenir (2010:26) pelayanan adalah kegiatan yang dilakukan oleh seseorang atau sekelompok orang dengan landasan faktor materi melalui sistem, prosedur dan metode tertentu dalam rangka usaha memenuhi kepentingan orang lain sesuai dengan haknya. Pelayanan hakikatnya adalah serangkaian kegiatan, karena itu pelayanan merupakan sebuah proses. Sebagai proses, pelayanan berlangsung secara rutin dan berkesinambungan, meliputi seluruh kehidupan orang dalam masyarakat.

Sedangkan menurut Groomros (1990:27) dalam Ratminto dan Atik (2005:2) pelayanan adalah suatu aktivitas atau serangkaian aktivitas yang bersifat tidak kasat mata (tidak dapat diraba) yang terjadi sebagai akibat adanya interaksi antara konsumen dengan karyawan atau hal-hal lain yang di sediakan oleh perusahaan pemberi pelayanan yang dimaksudkan untuk memecahkan permasalahan konsumen atau pelanggan.

Berbeda dengan Supranto (2006:227) mengatakan bahwa pelayanan atau jasa merupakan suatu kinerja penampilan, tidak terwujud dan cepat hilang, lebih dapat dirasakan dari pada dimiliki, serta pelanggan lebih dapat berpartisipasi aktif dalam proses mengonsumsi jasa tersebut.

### 2.2.4 Pengertian *Wedding Organizer*

Menurut Rahmat Hidayat (2016:5), *wedding organizer* adalah suatu jasa khusus yang secara pribadi membantu calon pengantin dan keluarga dalam perencanaan dan supervise pelaksanaan rangkaian acara pesta pernikahan sesuai dengan jadwal yang telah ditetapkan, antara lain:

1. Memberikan informasi mengenai berbagai macam hal yang berhubungan acara pernikahan.
2. Membantu merumuskan konsep pernikahan.



3. Fasilitas, negosiasi dan koordinasi dengan pihak gedung/hotel dan *supplier/vendor* seperti : catering, dekorasi, fotografer, perias, grup musik, dll.
4. Membantu memilih jenis makanan, kuantitas dan *food tasting*.
5. Menghubungi setiap rekanan untuk *follow up* yang diperlukan.
6. Mengikuti rapat koordinasi dengan keluarga, juga memimpin rapat teknis dengan para rekanan.
7. *Technical meeting* di tempat resepsi.
8. Membantu menyusun buku acara.
9. Menurunkan minimal 5 (lima) orang *staff*
10. Mengkoordinasikan acara dan seluruh rekanan pada hari H, memonitoring proses dan detail acara serta membantu mengingatkan calon pengantin atau anggota keluarga yang berkepentingan mengenai hal-hal yang berkaitan dengan kelancaran penyelenggaraan acara.

### 2.2.5 Metode *Simple Multi Attribute Rating Technique* (SMART)

Metode SMART merupakan metode pengambilan keputusan multi kriteria yang dikembangkan oleh Edward pada tahun 1977. SMART merupakan teknik pengambilan keputusan multi kriteria ini didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting ia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik.

SMART menggunakan *linear additive* model untuk meramal nilai setiap alternatif. SMART merupakan metode pengambilan keputusan yang fleksibel. SMART lebih banyak digunakan karena kesederhanaanya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisa yang terlibat adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan.

Model fungsi *utility linear* yang digunakan oleh SMART adalah seperti berikut (Edward, 1977):



$$u(a_i) = \sum_{j=1}^k w_j u_{ij}$$

Dimana:

- 1)  $w_j$  adalah nilai pembobotan kriteria ke- $j$  dari  $k$  kriteria.
- 2)  $k$  adalah jumlah kriteria
- 3)  $u_{ij}$  adalah nilai utility alternatif  $i$  pada kriteria  $j$ .
- 4) Pemilihan keputusan adalah mengidentifikasi mana dari  $n$  alternatif yang mempunyai nilai fungsi terbesar.
- 5) Nilai fungsi ini juga dapat digunakan untuk meranking  $n$  alternatif.

Urutan dalam penggunaan metode SMART (Goodwin dan Wright, 2004) adalah sebagai berikut:

- a. Menentukan banyaknya kriteria digunakan.
- b. Menentukan bobot kriteria pada masing- masing kriteria dengan menggunakan interval 1-100 untuk masing-masing kriteria dengan prioritas terpenting.
- c. Hitung normalisasi dari setiap kriteria dengan membandingkan nilai bobot kriteria dengan jumlah bobot kriteria. Menggunakan rumus:

$$nw_j = \frac{w_j}{\sum w_j}$$

Dimana:

- 1)  $nw_j$  adalah normalisasi bobot kriteria ke- $j$
- 2)  $w_j$  adalah nilai bobot kriteria ke- $j$
- d. Memberikan nilai parameter kriteria pada setiap kriteria untuk setiap alternatif.
- e. Menentukan nilai utiliti dengan mengkonversikan nilai kriteria pada masing-masing kriteria menjadi nilai kriteria data baku. Nilai utility diperoleh dengan menggunakan persamaan:

$$u_i(a_i) = \frac{C_{out} - C_{min}}{C_{max} - C_{min}}$$

Dimana:

- 1)  $u_i(a_i)$  adalah nilai utiliti kriteria ke-1 untuk kriteria ke- $I$



- 2)  $C_{max}$  adalah nilai kriteria maksimal
- 3)  $C_{min}$  adalah nilai kriteria minimal
- 4)  $C_{out}^i$  adalah nilai kriteria ke- i

Maka didapat nilai tersebut adalah:

$$C_{out}^i = u_i(a_i), 1 = 0 ; 2 = 0,5 : 3 = 1$$

- f. Menentukan nilai akhir dari masing- masing kriteria dengan mengalihkan nilai yang didapat dari normalisasi nilai kriteria data baku dengan nilai normalisasi bobot kriteria. Kemudian jumlahkan nilai dari perkalian tersebut.

$$u(a_i) = \sum_{j=i}^m w_j u_i(a_i)$$

Dimana;

- 1)  $u(a_i)$  adalah nilai total alternatif
- 2)  $w_j$  adalah hasil dari normalisasi bobot kriteria
- 3)  $u_i(a_i)$  adalah hasil penentuan nilai utiliti

## 2.3 Teori Program

### 2.3.1 Basis Data (*Database*)

Menurut Indrajani (2015:70), "Basisdata adalah sekumpulan data terintegrasi yang secara logika saling berhubungan."

Menurut Sutanta (2011:35), "Basisdata adalah kumpulan dari berbagai macam tipe record yang memiliki hubungan antar-record dan rincian data terhadap objek tertentu."

Menurut Madcoms (2010:13), "*Database* adalah wadah atau tempat yang digunakan untuk mengumpulkan atau mengorganisasikan data atau informasi."

Dari ketiga definisi diatas dapat disimpulkan bahwa basisdata adalah kumpulan data yang disimpan di dalam wadah tertentu secara sistematis untuk memperoleh informasi dari basis data tersebut.



### 2.3.2 MySQL

Menurut Saputra et.al (2012:7), "MySQL merupakan *database* yang bekerja menggunakan *Structur Query Language* (SQL)."

Menurut Sidik (2014:333), "MySQL merupakan software database yang termasuk pling populer di lingkungan *Linux*, kepopuleran ini karena ditunjang performasi query dari databsnya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah."

Dari kedua definisi diatas dapat disimpulkan bahwa MySQL merupakan salah satu sistem manajemen *database* (DBMS) berbasis SQL yang bersifat *open source* dan paling populer saat ini.

### 2.3.3 PHP Hypertext Preprocessor (PHP)

#### 2.3.3.1 Pengertian PHP Hypertext Preprocessor (PHP)

Menurut Sidik (2014:4), "PHP secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*."

Menurut Raharjo (2015:315),"PHP adalah bahasa pemrograman yang dibuat oleh Rasmus Lerdorfdan dirilis pada tahun 1995, yang umumnya digunakan untuk mengembangkan aplikasi berbasis web."

Dari definisi diatas dapat disimpulkan bahwa PHP *Hypertext Preprocessor* dikenal sebagai bahasa pemrograman *server side* yang membuat dokumen HTML secara *on the fly* yang dieksekusi dari server web yang difungsikan untuk membangun suatu web dinamis atau aplikasi berbasis web.

#### 2.3.3.2 *Script Hypertext Preprocessor* (PHP)

Kadir, Abdul (2008:3), skrip Hypertext Processor berkedudukan sebagai tag dalam Bahasa HTML (*Hypertext Markup Language*) adalah bahasa standar untuk membuat halaman-halaman web. Kode PHP *Hypertext Processor* diawali



dengan tag `<?php` dan diakhiri dengan tag `?>`. Baris kode yang berada diantara tag kode PHP tersebutlah yang akan dibaca oleh server dan kemudian diproses.

### 2.3.3.3 Konsep Kerja PHP

Menurut Kadir (2008: 4) menjelaskan tentang skema kerja PHP sebagai berikut: Model kerja HTML (*Hypertext Markup Language*) diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resource Localator*) atau dikenal dengan sebutan alamat internet, *browser* merupakan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari file yang diminta dan memberikan isinya ke *web browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan menampilkan hasilnya ke layar pemakai.

Sedangkan model kerja PHP pada prinsipnya serupa dengan kode HTML. Hanya saja ketika berkas PHP yang diminta berhasil didapatkan *web server*, isinya segera dikirim ke mesin PHP. Mesin inilah yang kemudian memproses dan memberikan hasil prosesnya kembali *web server*. Selanjutnya, *web server* menyampaikannya ke klien.

### 2.3.3.4 Cascading Style Sheet (CSS)

Koesheryatin (2014:101), "*Cascading Style Sheet* adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu website, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk memformat halaman web yang ditulis dengan HTML.

Madcoms (2012:88), "*Cascading Style Sheet* adalah suatu kumpulan kode-kode untuk memformat dan mengendalikan tampilan isi dalam suatu halaman web."

Dari kedua definisi diatas dapat disimpulkan bahwa CSS merupakan kode-kode untuk mengatur tampilan web dan memformat halaman web yang



ditulisd dengan HTML baik tata letaknya, jenis huruf, warna dan semua yang berhubungan dengan tampilan.

### 2.3.4 XAMPP

Raharjo (2015:25) menjelaskan tentang XAMPP sebagai berikut: XAMPP adalah kompilasi software yang membungkus Apache HTTP *Server*, MySQL, PHP dan *Pearl*. Dengan menggunakan instalasi paket XAMPP, instalasi paket *software* yang dibutuhkan untuk proses pengembangan web (Apache HTTP *Server*, MySQL, PHP) dapat dilakukan dengan sangat mudah tanpa harus dilakukan secara terpisah.



**Gambar 2.1** Logo XAMPP

### 2.3.5 *Sublime Text*

*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi Python API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*.

*Sublime Text* bukanlah aplikasi *open source* 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 linsensi aplikasi gratis.

*Sublime Text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrogramman 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 di-*download* sesuai kebutuhan user.



**Gambar 2.2** Logo *Sublime Text*

Berikut beberapa fitur yang diunggulkan dari aplikasi *Sublime Text*:

a. *Goto Anything*

Fitur yang sangat membantu dalam membuka file ataupun menjelajahi isi dari file hanya dengan beberapa keystrokes.

b. *Multiple Selections*

Fitur ini memungkinkan user untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya.

c. *Command Palette*

Dengan hanya beberapa keystrokes, user dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

d. *Distraction Free Mode*

Bila user memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu user dengan memberikan tampilan layar penuh.

e. *Split Editing*

Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan editing perpecahan. Mengedit sisi file dengan sisi, atau mengedit dua lokasi di



satu file. Anda dapat mengedit dengan banyak baris dan kolom yang user inginkan.

f. *Instant Project Switch*

Menangkap semua file yang dimasukkan kedalam project pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam *project* lainnya dengan cepat.

g. *Plugin API*

Dilengkapi dengan plugin API berbasis Python sehingga membuat aplikasi ini sangat tangguh.

h. *Customize Anything*

Aplikasi ini memberikan user fleksibilitas dalam hal pengaturan fungsional.

i. *Cross Platform*

Aplikasi ini dapat berjalan hampir disemua *operating system modern* seperti *Windows, OS X, dan Linux based operating system*.

### 2.3.6 CodeIgniter

Menurut Hakim (2010:8) *CodeIgniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal.



**Gambar 2.3** Logo *CodeIgniter*

Sumber: Hakim (2010:9) Membangun Web Berbasis PHP dengan *Framework CodeIgniter*

*CodeIgniter* pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc., sebuah perusahaan yang memproduksi CMS (*Content Management System*)



yang cukup handal, yaitu *Expression Engine*. Saat ini, *CodeIgniter* dikembangkan dan di-*maintain* oleh *Expression Engine Development Team*. Adapun beberapa keuntungan menggunakan *CodeIgniter*, diantaranya:

1. Gratis

*CodeIgniter* berlisensi dibawah Apache/BSD *open source*.

2. Ditulis Menggunakan PHP 4

Meskipun *CodeIgniter* dapat berjalan di PHP 5, namun sampai saat ini kode program *CodeIgniter* masih dibuat dengan menggunakan PHP 4.

3. Berukuran Kecil

Ukuran *CodeIgniter* yang kecil merupakan keunggulan tersendiri. Dibanding dengan *framework* lain yang berukuran besar.

4. Menggunakan Konsep MVC

*CodeIgniter* menggunakan konsep MVC yang memungkinkan pemisahan *layer application-logic* dan *presentation*.

5. URL yang Sederhana

Secara default, URL yang dihasilkan *CodeIgniter* sangat bersih dan *Search Engine Friendly* (SEF).

6. Memiliki Paket *Library* yang Lengkap

*CodeIgniter* mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses *database*, mengirim *email*, memvalidasi *form*, menangani *session* dan sebagainya.

7. *Extensible*

Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.

8. Tidak Memerlukan *Template Engine*

Meskipun *CodeIgniter* dilengkapi dengan *template* parser sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

9. Dokumentasi Lengkap dan Jelas

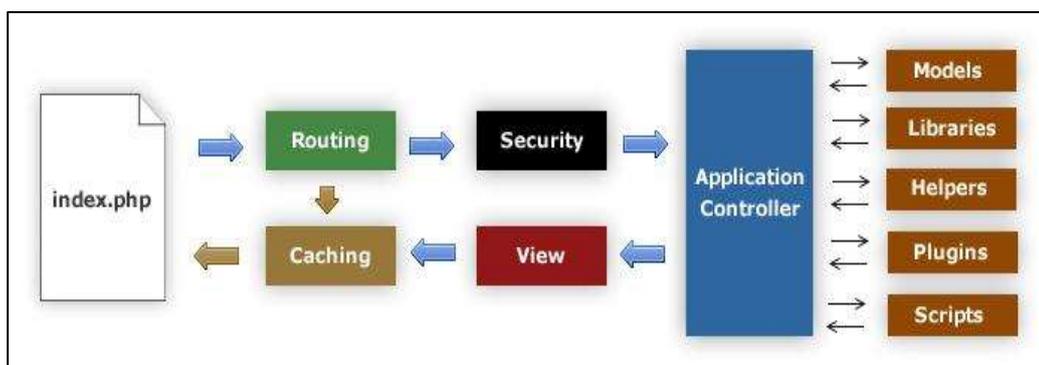


Dari sekian banyak *framework*, *CodeIgniter* adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas.

#### 10. Komunitas

Komunitas *CodeIgniter* saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>).

Proses aliran data aplikasi pada sistem dapat diilustrasikan seperti terlihat pada berikut:



**Gambar 2.4** *Application Flowchart*

Sumber: Hakim (2010:12) Membangun Web Berbasis PHP dengan *Framework* *CodeIgniter*

Keterangan:

1. *Index.php* berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan *CodeIgniter*.
2. Router memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan dengannya.
3. Jika *Cache* aktif, maka hasilnya akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
4. *Security*. Sebelum *Controller* dimuat, *HTTP request* dan data yang dikirimkan *user* akan difilter untuk keamanan.
5. *Controller* memuat *model*, *core libraries*, *plugins*, *helpers* dan semua *resource* yang diperlukan untuk memproses *request*.



6. Akhirnya *View* yang dihasilkan akan dikirimkan ke *browser*. Jika *Cache* aktif, maka *View* akan disimpan sebagai *Cache* dahulu, sehingga pada *request* berikutnya langsung dapat ditampilkan.

## 2.4 Teori Khusus

### 2.4.1 Unified Modeling Language (UML)

Menurut Mustakim et.al (2016:4), “UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.”

Menurut Nugroho (2005:16), ”UML adalah notasi-notasi yang digunakan sebagai pemodelan visual untuk sistem informasi atau perangkat lunak. Pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati”.

Beberapa diagram dalam UML yang akan digunakan dalam pembuatan laporan kerja praktik ini antara lain sebagai berikut:

#### 2.4.1.1 Usecase Diagram

Menurut Indrajani (2015:45), “*Usecase diagram* adalah suatu diagram yang berisi *usecase*, *actor* serta *relationship* yang merupakan titik awal dalam menganalisis kebutuhan sistem pada saat perancangan.”

Menurut Nugroho (2015:19), "*Usecase diagram* bersifat statis yang memperlihatkan himpunan *usecase* dan *actor* untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan dan diharapkan pengguna."

Diagram *usecase* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*, sedangkan *usecase* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata pengguna dan aktor mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. *Usecase diagram* dapat digunakan untuk menentukan



kebutuhan apa saja yang diperlukan suatu sistem, jadi dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan sesuatu. Berikut merupakan simbol-simbol diagram *usecase*:

**Tabel 2.1** Simbol-simbol Diagram *Usecase*

No	Gambar	Nama	Keterangan
1.		Aktor/ Actor	Aktor menggambarkan pengguna sistem, dapat berupa manusia atau sistem <i>terotomatisasi</i> lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
2.		Use Case	Simbol ini menggambarkan interaksi antara actor dengan software aplikasi tersebut.
3.		System Boundary	Menggambarkan batasan antara sistem dengan actor.
4.		Asosiasi/ Asosiation	Menggambarkan hubungan antar actor dan <i>use case</i> .

(Sumber : Indrajani 2015:46)

#### 2.4.1.2 *Activity Diagram*

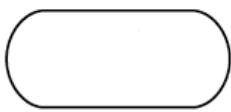
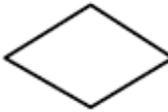
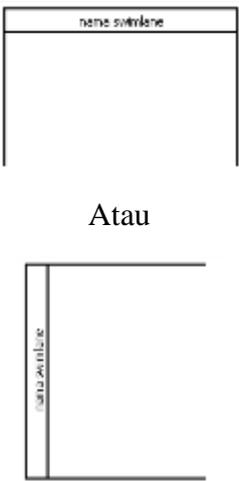
Menurut Nugroho (2015:61), "*Activity Diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *Usecase*. *Activity diagram* digunakan untuk memodelkan aspek dinamis dari sistem. *Activity diagram* secara esensial mirip diagram alir (*flowchat*), memperlihatkan aliran kendali dari suatu aktivitas ke aktivitas yang lainnya."

Menurut Rosa dan Salahudin (2014:161), "Diagram aktivitas (*activity diagram*) menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi



aktivitas yang dapat dilakukan oleh sistem." Berikut merupakan simbol-simbol yang digunakan dalam Diagram Aktivitas yang dapat dilihat pada tabel berikut:

**Tabel 2.2** Simbol-simbol Diagram dengan Aktivitas

No	Simbol	Nama	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.		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5.	 <p>Atau</p>	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Rosa dan Salahudin, 2014:161)

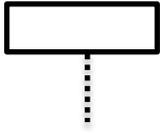


### 2.4.1.3 Sequence Diagram

Diagram sekuensial adalah interaction diagram yang memperlihatkan event-event yang berurutan sepanjang berjalannya waktu, dibaca dari atas ke bawah. Diagram sekuen akan menggambarkan aliran-aliran pada suatu usecase.

Indrajani (2015:50), menjelaskan tentang sequence diagram merupakan suatu diagram yang menggambarkan bagaimana objek-objek berpartisipasi dalam bagian interaksi dan pesan yang ditukar dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem waktu atau urutan gunakan Sequence Diagram. Berikut simbol-simbolnya:

**Table 2.3** Simbol-simbol Diagram Sekuensial

No	Simbol	Nama	Keterangan
1.		Lifeline	<i>Object entity</i> , yang saling berinteraksi.
2.		Aktor	Menggambarkan <i>actor</i> yang terlibat.
3.		Message (call)	Menggambarkan alur <i>message</i> yang merupakan kejadian objek pengirim <i>lifeline</i> ke objek penerima <i>lifeline</i> .
4.		Message (return)	Menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.
5.		Activation	Menggambarkan hubungan antara <i>object</i> dengan <i>message</i> .

(Sumber: Indrajani 2015:51)

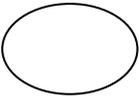
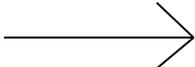
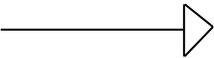


#### 2.4.1.4 Class Diagram

Menurut Nugroho (2005:110), "Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang dikembangkan. Dalam class diagram memberi gambaran diagram statis tentang sistem atau perangkat lunak dengan relasi-relasi yang ada didalamnya.

Menurut Indrajani (2014:49): "Class diagram adalah diagram yang menggambarkan perbedaan antara class-class hubungan antar class dan dinamakan sub sistem class, pada class diagram terdapat nama class, atribut dan operation. Berikut merupakan simbol Diagram Kelas:

**Tabel 2.4** Simbol-simbol Diagram Kelas

No	Simbol	Nama	Keterangan			
1.	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">Class</td> </tr> <tr> <td style="text-align: center;">Attribute</td> </tr> <tr> <td style="text-align: center;">Operation</td> </tr> </table>	Class	Attribute	Operation	Kelas	Kelas pada struktur sistem
Class						
Attribute						
Operation						
2.		Antarmuka/ Interface	Sama dengan konsep interface dalam pemrograman berorientasi obyek.			
3.		Asosiasi/ Association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.			
4.		Asosiasi berarah/directed	Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multiplicity.			
5.		Spesialisasi	Asosiasi antar kelas dengan makna generalisasi spesialisasi (umum – khusus)			



---

6.		Kebergantungan/ dependency	Ketergantungan antarkelas.
7.		Agregasi	Relasi antar kelas dengan makna umum semua bagian (whole-part)

(Sumber: Rosa dan Salahudin, 2014:141)