

BAB II

LANDASAN TEORI

Landasan Teori merupakan sebuah aktivitas untuk meninjau atau mengkaji kembali berbagai literatur yang telah dipublikasikan oleh akademisi atau peneliti lain sebelumnya terkait topik yang akan diteliti (Taylor & Procter 2010). Dalam hal ini bab tinjauan pustaka akan menjelaskan mengenai teori-teori relevan yang akan di gunakan pada penelitian ini.

2.1 Web

World Wide Web atau WWW atau jugadikenaldengan WEB adalahsalahsatulayanan yang didapatolehpemakai computer yang terhubungke internet. Web inimenyediakaninformasibagipemakai computer yang terhubungke internet darisekedarinformasi “sampah” atauinformasi yang tidakbergunamasekalisampaiinformasi yang serius; dariinformasi yang gratisansampaiinformasi yang komersial. Website atausitusdapatdiartikansebagaikumpulanhalaman-halaman yang digunakanuntukmenampilkaninformasitekst, gambardiamataugerak, animasi, suara, danataugabungandarisemuanyaitubaik yang bersifatstatismaupundinamis yang membentuksaturangkaiانبangunan yang salingterkaitdimanamasing-masingdihubungkandenganjaringan-jaringanhalaman (hyperlink).

Untuk menghubungkan beberapa komputer sehingga menjadi sebuah kelompok jaringan, perangkat yang akan dipantaumembutuhkan suatu media penghubung yang bernama TCP/IP, yaitu sebuah protokol yang mengidentifikasi sebuah komputer yang terhubung di dalam jaringan. TCP/IP memiliki teknik

mengidentifikasi dengan menggunakan penomoran yang dinamakan *Internet Protokol Address* atau biasa disebut nomor IP/IP address. Dengan menggunakan nomor ini sebuah komputer dapat terhubung dengan komputer lain dalam sebuah jaringan atau dalam jaringan global yang disebut internet.

Word Wide Web (Jaringan Dunia Luas) atau yang sering disebut WWW adalah sebuah bagian dari internet yang sangat dikenal dalam dunia internet, dengan adanya WWW seorang pengguna dapat menampilkan sebuah halaman virtual yang disebut dengan Website.

Web adalah perangkat lunak yang berbasis teknologi dan standar dari *World Wide Web Consortium* (W3C) yang menyediakan sumber daya yang khusus untuk Web seperti konten dan layanan melalui tampilan antar muka, peramban web (Kappel, et al., 2006).

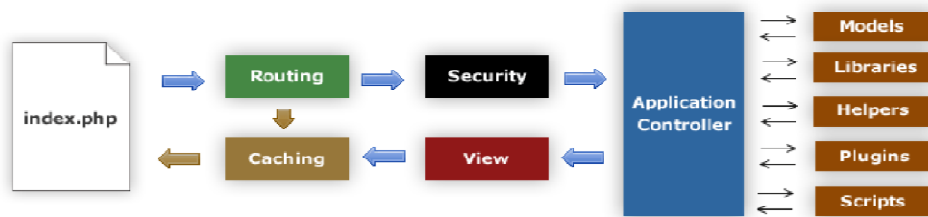
Perkembangan web saat ini sangat pesat. Jumlah pengembangan yang sangat banyak membuat perawatan web yang dibutuhkan menjadi banyak pula. Hal ini menyebabkan memburuknya kualitas dari web. Masalah yang timbul sangat banyak, mulai dari informasi yang sudah ketinggalan jaman atau kurang relevan, respon web yang lambat, gangguan saat mengakses web, sampai masalah keamanan (Suh, 2005).

2.2 Codeigniter

CodeIgniter adalah suatu *Application Development Framework* untuk pembangunan suatu website berbasis PHP. Tujuannya adalah agar pembangunan projek bisa lebih cepat dibandingkan dengan membangunnya dari awal, dengan menyediakan library yang umum digunakan untuk pembangunan suatu *website*,

dengan *interface* dan struktur logika yang sederhana untuk mengaksesnya. *CodeIgniter* akan membuat developer semakin kreatif.

Tujuan dari pembuatan *frameworkCodeIgniter* adalah untuk menghasilkan *framework* yang akan dapat digunakan untuk pengembangan proyek pembuatan website secara lebih cepat dibandingkan dengan pembuatan website dengan cara



Gambar 1.1 Diagram aplikasi menggunakan Codeigniter
Sumber :www.codeigniter.com

koding secara manual, dengan menyediakan banyak sekali pustaka yang dibutuhkan dalam pembuatan website, dengan antarmuka yang sederhana dan struktur logika untuk mengakses pustaka yang dibutuhkan.

CodeIgniter membiarkan *user* untuk memfokuskan diri pada pembuatan website dengan meminimalkan pembuatan kode untuk berbagai tujuan pembuatan website.

CodeIgniter menggunakan pola *Model-View-Controller* (MVC). MVC menggunakan pendekatan dimana terjadi pemisahan antara logika aplikasi dan tampilan, akan dijelaskan secara terpisah dibawah ini:

- a. *Model*, merepresentasikan struktur data. Umumnya class model mengandung fungsi untuk menampilkan, menambah, mengubah informasi pada basis data

- b. *View*, merupakan informasi yang dipresentasikan kepada user. *View* biasanya merupakan sebuah halaman web, tapi pada CodeIgniter ini juga bisa merupakan *fragment*, seperti *header* atau *footer*. Atau bisa juga merupakan sebuah *rsspage* ataupun jenis halaman lainnya.
- c. *Controller*, berfungsi sebagai penghubung antara *Model* dan *View* dan resource lainnya yang dibutuhkan untuk memproses HTTP request dan menghasilkan sebuah halaman web.

Walaupun begitu, konsep MVC pada CodeIgniter tidak begitu ketat, karena *Model* bisa tidak digunakan. Apabila pada saat pembuatan website tidak diperlukan pemisahan, atau ketika user memandang bahwa penggunaan *model* hanya akan menambah kompleksitas, maka cukup digunakan hanya *Controller* dan *View*. CodeIgniter juga mengizinkan *programmer* untuk menggabungkan *script* yang telah dibuat sebelumnya walaupun tidak menggunakan kaidah/aturan yang digunakan oleh CodeIgniter, ataupun membuat *library* baru, yang memungkinkan programmer untuk bekerja menggunakan caranya sendiri.

2.3 HTML

HTML adalah singkatan dari *HyperText Markup Language* yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah web browser. HTML dapat juga digunakan sebagai link-link antara file-file dalam situs atau dalam komputer dengan menggunakan localhost, atau link yang menghubungkan antar situs dalam dunia internet. Supaya dapat menghasilkan tampilan wujud yang terintegrasi Pemformatan hiperteks

sederhana ditulis dalam berkas format ASCII sehingga menjadi halaman web dengan perintah-perintah HTML.

2.4 Javascript

JavaScript adalah bahasa pemrograman yang bisa disisipkan ke HTML seperti halnya PHP akan tetapi javascript berjalan di sisi Client.JavaScript pertama kali dikembangkan oleh Brendan Eich dari Netscape dibawah nama Mocha, yang nantinya namanya diganti menjadi LiveScript, dan akhirnya menjadi JavaScript.JavaScript bisa digunakan untuk banyak tujuan, misalnya untuk membuat efek rollover baik di gambar maupun teks, dan yang penting juga adalah untuk membuat AJAX. JavaScript adalah bahasa yang digunakan untuk AJAX.

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*.

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*.

Akan tetapi karena sifatnya yang dijalankan di sisi client yakni di dalam web browser yang digunakan oleh pengunjung situs, user sepenuhnya dapat mengontrol eksekusi JavaScript. Hampir semua web browser menyediakan fasilitas untuk mematikan JavaScript, atau bahkan mengubah kode JavaScript yang ada. Sehingga tidak bisa bergantung sepenuhnya kepada JavaScript.

2.5 Database MYSQL

Kadir (2008) menyatakan bahwa, basis data merupakan sekumpulan data yang disimpan keseluruhan dalam satu media. Sebelum disimpan, data-data ini disusun dalam bentuk tabel yang saling berkaitan, maupun berdiri sendiri-sendiri.

Data-data yang akan disimpan dalam basis data haruslah memperhatikan beberapa hal sebagai berikut:

- a. Kerangkapan data
- b. Inkonsistensi data
- c. Data terisolasi yang disebabkan oleh pemakaian beberapa file basis data. Program aplikasi tidak dapat mengakses file tertentu dalam sistem basis data tersebut, kecuali program aplikasi dirubah atau ditambah sehingga seolah-olah ada file yang terpisah atau terisolasi terhadap file yang lain,
- d. Keamanan data. Hal ini berkaitan dengan akses yang terbatas untuk beberapa orang tertentu yang bertanggung jawab.
- e. Integrasi data yang berkaitan dengan kinerja sistem agar dapat melakukan kontrol pada semua bagian sistem, sehingga sistem tersebut selalu berada dalam kendali.

MySQL adalah sebuah perangkat lunak suatu sistem manajemen basis data yang multithread dan multi-user. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL merupakan susunan salah satu konsep utama dalam basis data sejak lama, yaitu *Structured Query Language*(SQL). Kendala dari suatu DBMS dapat diketahui dari cara kerja optimalisasinya dalam melakukan perintahperintah SQL yang dibuat user ataupun program-program aplikasinya. Sebagai basis data server, MySQL dapat dikatakan lebih unggul dibanding basis data server lainnya dalam query data.

Suatu basis data relasional menyimpan data dalam tabel-tabel terpisah. Hal ini memungkinkan kecepatan dan fleksibilitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasikan data dari beberapa tabel pada suatu permintaan. Untuk menambahkan, mengakses, dan memproses data yang tersimpan pada suatu basis data komputer maka diperlukan sistem manajemen basis data seperti MySQL. Karena komputer sangat unggul dalam menangani sejumlah besar data, sistem manajemen basis data memainkan suatu peranan yang penting dalam komputasi, baik sebagai *utilitystand-alone* maupun bagian dari aplikasi lainnya.

2.6 Algoritma Apriori

Algoritma Apriori adalah algoritma paling terkenal untuk menemukan pola frekuensi tinggi. Pola frekuensi tinggi adalah pola-pola item di dalam suatu database yang memiliki frekuensi atau support di atas ambang batas tertentu yang disebut dengan istilah minimum support. Pola frekuensi tinggi ini digunakan untuk menyusun aturan asosiatif dan juga beberapa teknik data mining lainnya.

Walaupun akhir-akhir ini dikembangkan banyak algoritma yang lebih efisien dari Apriori seperti FP-growth, LCM dsb, tetapi Apriori tetap menjadi algoritma yang paling banyak diimplementasikan dalam produk komersial untuk data mining karena dianggap algoritma yang paling mapan.

Algoritma Apriori dibagi menjadi beberapa tahap yang disebut iterasi atau pass. Tiap iterasi menghasilkan pola frekuensi tingg dengan panjang yang sama dimulai dari pass pertama yang menghasilkan pola frekuensi tingg dengan panjang satu. Di iterasi pertamanya, support dari setiap item dihitung dengan men-scan database.

Setelah support dari setiap item didapat, item yang memiliki support di atas minimum support dipilih sebagai pola frekuensi tingg dengan panjang 1 atau sering disingkat 1-itemset. Singkatan k-itemset berarti suatu set yang terdiri dari k item.

Iterasi kedua menghasilkan 2-itemset yang tiap set-nya memiliki dua item. Pertama dibuat kandidat 2-itemset dari kombinasi semua 1-itemset. Lalu untuk tiap kandidat 2-itemset ini dihitung support-nya dengan men-scan database. Support disini artinya jumlah transaksi dalam database yang mengandung kedua item dalam kandidat 2-itemset. Setelah support dari semua kandidat 2-itemset didapatkan, kandidat 2-itemset yang memenuhi syarat minimum support dapat ditetapkan sebagai 2-itemset yang juga merupakan pola frekuensi tingg dengan panjang 2.

Untuk selanjutnya pada iterasi ke-k dapat dibagi lagi menjadi beberapa bagian :

1. Pembentukan kandidat itemset, Kandidat k -itemset dibentuk dari kombinasi $(k-1)$ -itemset yang didapat dari iterasi sebelumnya. Satu ciri dari algoritma Apriori adalah adanya pemangkasan kandidat k -itemset yang subset-nya yang berisi $k-1$ item tidak termasuk dalam polafrekuensi tinggi dengan panjang $k-1$.
2. Penghitungan support dari tiap kandidat k -itemset. Support dari tiap kandidat k -itemset didapat dengan men-scan database untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat k -itemset sb. Ini adalah juga ciri dari algoritma Apriori dimana diperlukan penghitungan dengan scan seluruh database sebanyak k -itemset terpanjang.
3. Tetapkan polafrekuensi tinggi. Polafrekuensi tinggi yang memuat k item atau k -itemset ditetapkan dari kandidat k -itemset yang support-nya lebih besar dari minimum support.
4. Bila tidak didapat polafrekuensi tinggi baru maka seluruh proses dihentikan. Bila tidak, maka k ditambah satu dan kembali ke bagian 1.

Pseudocode dari algoritma Apriori dapat dilihat di Gambar berikut :

```

 $L_1 := \{ \text{large 1-itemsets} \};$ 
 $k := 2;$  //  $k$  represents the pass number
while ( $L_{k-1} \neq \emptyset$ ) do
begin
   $C_k :=$  New candidates of size  $k$  generated from  $L_{k-1}$ ; (apriori-gen)
  forall transactions  $t \in \mathcal{D}$  do
    Increment the count of all candidates in  $C_k$  that are contained in  $t$ ;
   $L_k :=$  All candidates in  $C_k$  with minimum support;
   $k := k + 1$ ;
end
Answer :=  $\bigcup_k L_k$ ;

```

Sedangkan pseudocode dari pembentukan kandidat itemset bersamapemangkannya di berikan di Gambar berikut :

```

- (1) Join Step
insert into candidate  $k$ -itemset
select  $p.item_1, p.item_2, \dots, p.item_{k-1}$ 
from large  $(k-1)$ -itemset  $p$ , large  $(k-1)$ -itemset  $q$ 
where  $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$ ;

(2) Prune Step
forall itemsets  $c \in$  candidate  $k$ -itemset do
  forall  $(k-1)$ -subsets  $s$  of  $c$  do
    if ( $s \notin$  large  $(k-1)$ -itemset) then
      delete  $c$  from candidate  $k$ -itemset;

```

Satu contoh dari penerapan algoritma Apriori diilustrasikan pada Gambar berikut:

Transaction D			C_1	L_1
TID	Items		Itemset	Count
100	1,3,4	Scan D \Rightarrow	{1}	2
200	2,3,5		{2}	3
300	1,2,3,5		{3}	3
400	2,5		{4}	1
			{5}	3
C_2			C_2	L_2
Itemset			Itemset	Count
{1,2}		Scan D \Rightarrow	{1,2}	1
{1,3}			{1,3}	2
{1,5}			{1,5}	1
{2,3}			{2,3}	2
{2,5}			{2,5}	3
{3,5}			{3,5}	2
C_3			C_3	L_3
Itemset			Itemset	Count
{2,3,5}		Scan D \Rightarrow	{2,3,5}	2

Di sini minimum support adalah 50% atau minimal support-nya adalah 2. Pada iterasi pertama, item yang support-nya atau count-nya di bawah 2 dieliminasi dari 1-itemset L_1 . Kemudian kandidat 2-itemset C_2

dari iterasi kedua dibentuk dari cross product item-item yang ada di L1. Setelah kandidat 2-itemset itu dihitung dari database, ditetapkan 2-itemset L2. Proses serupa berulang di iterasi ketiga, tetapi perhatikan bahwa selain {2,3,5} yang menjadikan kandidat 3-itemset C3 sebenarnya ada juga itemset {1,2,3} dan {1,3,5} yang dapat diperoleh dari kombinasi item-item di L2, tetapi kedua itemset itu dipangkas karena {2,3} dan {1,5} tidak ada di L2. Proses ini berulang sampai tidak ada lagi kandidat baru yang dapat dihasilkan di iterasi ke 4.

Dalam contoh ini bisa dilihat bahwa Apriori dapat mengurangi jumlah kandidat yang harus dihitung support-nya dengan pemangkasan. Misalnya kandidat 3-itemset dapat dikurangi dari 3 menjadi 1 saja. Pengurangan jumlah kandidat ini merupakan sebab utama peningkatan performa Apriori.

Tetapi di lain pihak Apriori memiliki kelemahan karena harus melakukan scan database setiap kali iterasi, sehingga waktu yang diperlukan bertambah dengan makin banyak iterasi.