

BAB II

LANDASAN TEORI

A. Konsep Dasar Web

Pengertian *web* sangatlah luas, sebuah website sangat diperlukan dalam melakukan suatu kegiatan kerja menggunakan sistem yang sudah terkomputerisasi. Penggunaan *web* ini memungkinkan terciptanya kinerja untuk menghasilkan informasi yang cepat, dan efisien (Arief:2011).

B. Website

Menurut Hidayat (2010:2) “*Web* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”. Hubungan antara satu halaman *web* yang lainnya disebut *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*.

1. Internet

Menurut Simarmata (2010:47) mengemukakan bahwa “*Internet* adalah kelompok atau kumpulan dari jutaan komputer”. *Internet* merupakan jaringan global yang menghubungkan suatu jaringan dengan jaringan yang lain diseluruh dunia.

Internet pertama kali digunakan sebagai proyek penelitian yang ditemukan oleh *Advanced Research Project Agency* (ARPA) *Departement Of Defense* (DOD) di Amerika Serikat. Pada dasarnya, *internet* digunakan untuk menghubungkan komputer. Versi yang pertama disebut *Advanced Research Projects Agency Network* (ARPANET). Pada tahun 1972, ARPA berubah menjadi *Defence Advanced Research Project Agency* (DARPA) dengan tetap mempromosikan ARPANET. Pengembangan *internet* dengan jenis peralatan yang berbeda, namun bisa saling berhubungan satu sama lain merupakan tantangan yang besar pada saat itu. Pada tahun 1973-1974, peneliti merancang sebuah *Transmission Control Protocol* (TCP) dan *Internet Protocol* (IP).

Pada awal tahun 1980-an ARPANET dipecah menjadi dua bagian, yaitu MILNET dan ARPANET karena pertimbangan keamanan. Pihak militer berjalan terus dengan MILNET, sedangkan penelitian tetap menggunakan ARPANET. Pada pertengahan tahun 1980-an, *National Science Foundation* (NSF) di Washington, D.C mendistribusikan teknologi *internet* kepada beberapa universitas seperti Berkeley, MIT, Stanford, dan UCLA. Dan pada akhirnya *internet* pun mulai menyebar di dunia.

2. Web Browser

Menurut Sibero (2013:12) menyatakan bahwa: “ *Web browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*”. Sumber informasi web diidentifikasi dengan *Uniform*

Resource Identifier (URI) yang terdiri dari halaman *web*, video, gambar ataupun konten lainnya.

Sejarah *web browser* dimulai pada tahun 1991 saat Tim Berners-Lee membuat aplikasi *web browser* pertama pada komputer NeXT dengan nama *World Wide Web Browser*, kemudian di tahun 1993 NCSA (*National Center Supercomputing Application*) mengembangkan *web browser* grafis bernama NCSA Mosaic yang kemudian dilanjutkan pada tahun 1994 merilis Netscape Navigator dan pada tahun 1998 berubah menjadi *Mozilla Firefox*, *Microsoft corp*, ikut meramaikan perkembangan *web browser* dengan merilis *Internet Explorer* pada tahun 1995, dan diikuti oleh Opera di tahun 1996. Tidak ketinggalan *Apple inc*, juga meramaikan pasar *web browser* dengan merilis Safari pada tahun 2003. Dan di tahun 2008 *Google Inc*, merilis *web browser* dengan nama *Google Chrome*.

3. Web Server

Menurut Fathansyah, (2007:202) “HTTP menyediakan fasilitas yang bagus, disamping transfer dokumen yang mudah juga melakukan enkripsi pada dokumen demi alasan keamanan”. Bagian nama dokumen dalam URL dapat mengidentifikasi sebuah program *executable*, yang ketika dijalankan, akan menghasilkan dokumen HTML.

Ketika sebuah *server* HTTP menerima permintaan untuk pengambilan dokumen tertentu, *server* mengeksekusi program tersebut dan mengirimkan dokumen HTML yang dihasilkan program tersebut. Lebih jauh lagi, *client web*

dapat mengirimkan argumen-argumen tambahan yang akan digunakan program pada saat dieksekusi. Dengan begitu, pembentukan dokumen dapat menjadi bervariasi karena ditentukan oleh argumen-argumen yang disertakan ke *server* tersebut.

4. *Personal Home Page (PHP)*

Menurut Arief (2011:43) PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server-side scripting* 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 dikirimkan ke *browser* dalam format HTML.

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

- a. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. *Web Server* yang mendukung PHP dapat ditemukan di mana - mana dari mulai *apache*, *IIS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya *milis - milis* dan *developer* yang siap membantu dalam pengembangan.
- d. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
- e. *Personal Home Page (PHP)* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux*, *Unix*, *Macintosh*, *Windows*) dan

dapat dijalankan secara runtime melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

5. Hypertext Markup Language (HTML)

Menurut Sibero (2013:19) “*HyperText Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*”. *File* jenis ini biasanya disimpan dengan ekstensi atau akhiran HTM atau HTML. HTML menekankan penggambaran komponen-komponen struktur dan format di dalam halaman *web*. Adapun untuk menentukan penampilan adalah tugas CSS. Dengan adanya HTML, maka komputer menjadi memiliki bahasa komunikasi yang standar. Jadi dokumen HTML ini bisa dilihat dengan tampilan yang sama, baik di komputer *Windows*, *MAC*, *Linux*, dan lain sebagainya.

5. MYSQL

Menurut Arief (2011:151) “*Mysql* adalah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dalam pengelolaan datanya”. Kepopuleran MYSQL antara lain karena MYSQL menggunakan SQL sebagai bahan dasar untuk mengakses *databasenya* sehingga mudah untuk digunakan, kinerja *query* cepat dan mencukupi untuk kebutuhan-kebutuhan *database* perusahaan skala menengah kecil.

6. Xampp

Menurut Aditya (2011:16), Xampp adalah “perangkat lunak bebas, yang mendukung banyak sistem 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 XAMPP merupakan singkatan dari X (empat sistem operasi apapun). Apache, MySQL, PHP dan Perl. Program ini tersedia dalam *General Public License* (GNU) dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

7. Codeigniter

Codeigniter adalah *framework* web untuk bahasa pemrograman PHP, yang di buat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab (www.ellislab.com). EllisLab adalah suatu tim kerja yang berdiri pada tahun 2002 dan bergerak di bidang pembuatan software dan *tool* untuk para pengembangan web. Sejak tahun 2014 sampai sekarang, EllisLab telah menyerahkan hak kepemilikan *Codeigniter* ke *British Columbia Institute of Technology* (BCIT) untuk proses pengembangan lebih lanjut. Saat ini, situs resmi dari *Codeigniter* telah berubah dari www.ellislab.com ke www.codeigniter.com. (Budi Raharjo 2015).

Codeigniter memiliki banyak fitur (fasilitas) yang membantu para pengembang (*developer*) PHP untuk dapat membuat aplikasi *web* secara mudah

dan cepat. Dibandingkan dengan *framework* PHP lainnya, bersifat fleksibel (tidak kaku). *Codeigniter* mengizinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. Ini berarti *Codeigniter* masih memberi kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional (tanpa *framework*).

a. Beberapa Keunggulan *Codeigniter*

Codeigniter merupakan sebuah *toolkit* yang ditujukan untuk orang yang ingin membangun aplikasi *web* dalam bahasa pemrograman PHP. Beberapa keunggulan yang ditawarkan oleh *Codeigniter* adalah sebagai berikut :

- 1) *Codeigniter* adalah *framework* yang bersifat *free* dan *open-source*.
- 2) *Codegniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain.
- 3) Aplikasi yang dibuat menggunakan *Codeigniter* bisa berjalan cepat.
- 4) *Codeigniter* menggunakan pola desain *Model-View-Controller* (MVC) sehingga satu *file* tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami dan dipelihara di kemudian hari.
- 5) *Codeigniter* dapat diperluas sesuai dengan kebutuhan.
- 6) *Codeigniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *Codeigniter* dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.

b. Versi Codeigniter

Codeigniter memiliki dua versi yang dikembangkan secara silmutan (bersamaan), yaitu versi 2.2 dan 3.0. *Codeigniter* 2.2 merupakan versi yang ditetapkan sebagai versi yang stabil, sedangkan versi 3.0 merupakan perbaikan dari versi 2.2 yang digunakan untuk keperluan di masa mendatang. *Codeigniter* 3.0 juga menambahkan beberapa fitur baru yang sebelumnya tidak dimiliki oleh versi 2.2.

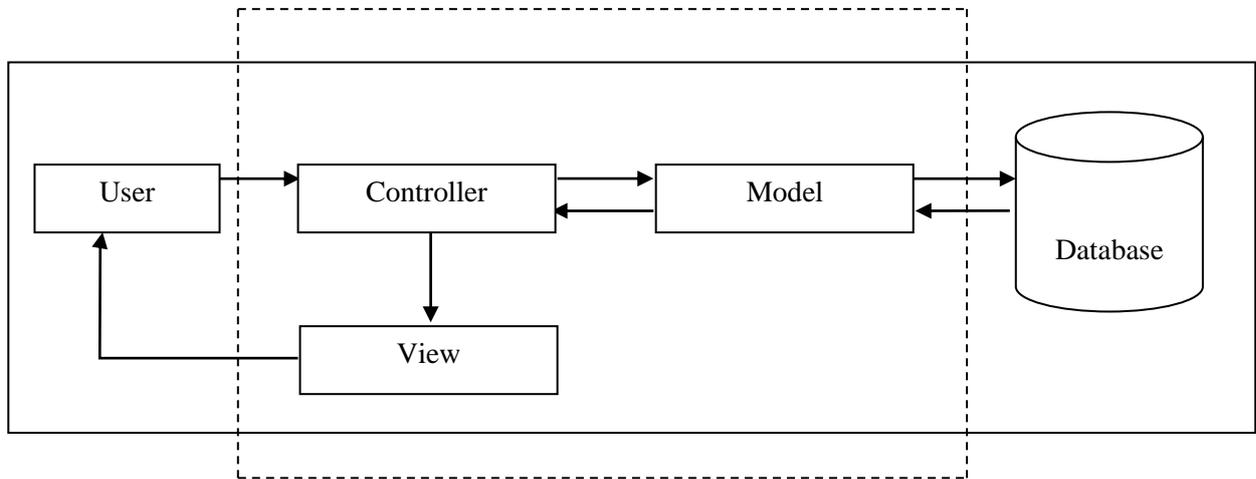
c. Mengenal pola Desain MVC

Dalam teknik pemrograman berorientasi objek, *Model-View-Controller* (MVC) adalah nama dari suatu, metodologi atau pola desain (*design pattern*) yang digunakan untuk merelasikan data dan user interface aplikasi secara *efisien*. Hampir semua *framework* untuk pengembangan aplikasi *web* mengimplementasikan arsitektur MVC. Bukan hanya *framework* PHP, tapi *framework* untuk bahasa pemrograman lain juga menerapkan konsep dasar yang sama.

Dalam pola MVC, komponen aplikasi dibagi menjadi tiga bagian, yaitu :

- 1) **Model**, yang mempresentasikan struktur data.
- 2) **View**, yang merupakan representasi keluaran (*output*) dari suatu model.
- 3) **Controller**, yaitu komponen yang bertugas mengambil masukan (*input*) dari user dan menggubahnya menjadi perintah untuk *model* dan atau *view*.

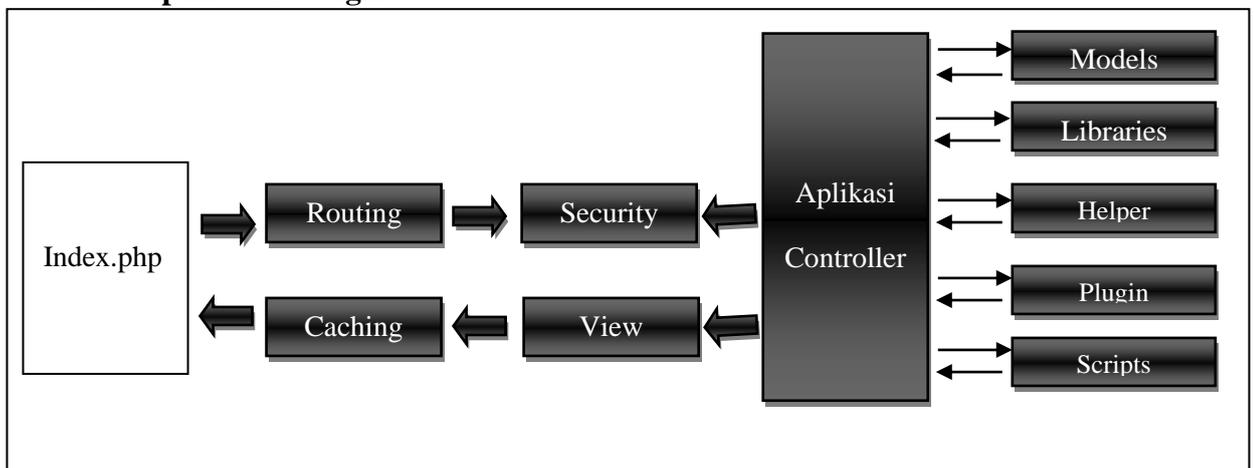
Secara umum, pola MVC dapat digambarkan sebagai berikut:



Gambar 2.1 Aplikasi dengan arsitektur *Model View Controller* (MVC)

Sama seperti *framework* PHP lainnya, CodeIgniter juga mengimplementasikan pola desain *Model View Controller* (MVC). Meskipun demikian, implementasi MVC untuk masing-masing *framework* tentu akan berbeda. Beberapa *framework* tentu bisa saja menambahkan komponen lain ke dalam arsitektur di atas untuk meningkatkan kerja aplikasi.

d. Alur Aplikasi CodeIgniter



Gambar 2.2 Alur Aplikasi menggunakan Codeigniter

Berikut ini penjelasan dari gambar diatas:

- 1) File *index.php* (kadang disebut *entry script* berperan sebagai *controller* depan, yang akan menginisialisasi daftar file yang dibutuhkan untuk menjalankan *CodeIgniter*. Untuk melakukan permintaan aplikasi server web melalui file *index.php*, dengan format *Unified Resource Identification* (URL) seperti berikut:

http://namahost/index.php/kelas-controller/metode-controller

Catatan :

Dalam beberapa frame work lai, metode yang didefinisikan di dalam kelas controller terkadang disebut juga sebagai aksi.

- 2) Permintaan yang dikirim oleh *user* (dalam bentuk URL) akan ditangkap oleh *router*, dan *router* akan menentukan *controller* dan metode mana yang harus dipanggil.
- 3) Jika ternyata halaman yang diminta oleh *user* sudah ditembolokkan (*cached*), halaman tersebut akan diambil dari *cache* dan langsung disajikan ke *web browser*.

- 4) Sebelum *controller* yang diminta oleh *user* dimuat, permintaan HTTP dan semua data yang dikirim akan disaring terlebih dahulu untuk keperluan pengamanan.
- 5) *Controller* memuat *model*, *library*, *helper*, dan *file-file* pendukung lain yang diperlukan untuk memroses permintaan *user*.
- 6) *Controller* memuat *view* untuk disajikan ke *web browser*. Jika mode penembolokan (*caching*) diaktifkan, maka *view* akan ditembolokkan terlebih dahulu sebelum ditampilkan. Dengan demikian jika nantinya ada permintaan yang sama, maka halaman yang diminta hanya akan diambil dari *cache*.

8. *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 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 fungsional (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

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

A. Basis Data

Menurut Kustiyaningsih (2011:146) “Database adalah struktur penyimpanan data. Untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah komputer, diperlukan sistem manajemen *database* seperti *MYSQL server*”.

Intinya database dalam dunia *programming* difungsikan untuk menyimpan seluruh data atau informasi yang dihasilkan dari proses interaksi didalam aplikasi, baik itu desktop ataupun *internet programming*. Dan biasanya pembangunan database akan dilakukan terlebih dahulu sebelum membangun aplikasi.

B. Model Pengembangan Perangkat Lunak

1. Model *Waterfall*

Menurut Sukamto dan Shalahuddin (2013:28) Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic lifecycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisa, desain, pengkodean, pengujian, dan tahap pendukung (*support*).

a. Analisa kebutuhan perangkat lunak

Proses Pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasi.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisa kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

d. Pembuatan Kode Program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

C. Pengujian Web

1. *Logical Relational Structure (LRS)*

Menurut Ladjamudin (2005:3) “LRS dibentuk dengan nomor tipe *record*”. Beberapa tipe *record* digambarkan dengan kotak empat persegi panjang dengan nama yang unik. LRS juga terdiri dari hubungan diantara tipe *record*. Salah satu metode pembuatan LRS yaitu dimulai dengan membuat ERD kemudian dikonversi ke dalam LRS.

2. *Black Box Testing*

Menurut Rizky (2011:264), *Black Box Testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja *internalnya*. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses *testing* di bagian luar.

Jenis *testing* ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Sebagai contoh, jika terdapat sebuah perangkat lunak yang merupakan sebuah sistem informasi *inventory* di sebuah perusahaan. Maka pada jenis *white box testing*,

perangkat lunak tersebut akan berusaha dibongkar *listing* programnya untuk kemudian dites menggunakan teknik-teknik yang telah dijelaskan sebelumnya. Sedangkan pada jenis *black box testing*, perangkat lunak tersebut akan dieksekusi kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar *listing* programnya.