

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Pengertian BUMIMAS KATONG BESARI

- BUMIMAS KATONG BESARI didirikan di Ponorogo pada tahun 2005 dengan ijin dari departemen tenaga kerja dan transmigrasi Republik Indonesia melalui : No.SIPP : KEP.549/Men/2006. BUMIMAS KATONG BESARI dikelola dan didukung oleh Tenaga pengajar yang telah ahli dan terlatih di bidang ketenagakerjaan sehingga dapat memberikan pelayanan yang memuaskan.

2.2. Pengertian calon Tenaga Kerja Indonesia (TKI)

Menurut Pasal 1 yang bagian (2) Undang-Undang Nomor 39 Tahun 2004 tentang Penempatan dan Perlindungan Tenaga Kerja Indonesia di Luar Negeri, calon Tenaga Kerja Indonesia adalah setiap warga negara Indonesia yang memenuhi syarat sebagai pencari kerja yang akan bekerja di luar negeri dan terdaftar di instansi pemerintahan kabupaten/kota yang bertanggung jawab di bidang ketenagakerjaan.

2.4 Pengertian Sistem

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsure atau variabel – variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain. Murdick dan Ross (1993) mendefinisikan sistem sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Sementara, definisi sistem dalam kamus *Webster's Unbringed* adalah elemen – elemen yang saling berhubungan dan membentuk satu kesatuan atau organisasi. Menurut Scott (1996), sistem terdiri dari unsure – unsure seperti masukan (*input*), pengolahan (*proccessing*), serta keluaran (*output*). Ciri pokok sistem menurut Gaspert ada empat, yaitu sistem itu beroperasi dalam suatu lingkungan, terdiri atas unsur –

unsur, ditandai dengan saling berhubungan, dan mempunyai suatu fungsi atau tujuan yang utama.

2.3. Sistem Informasi

Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan.

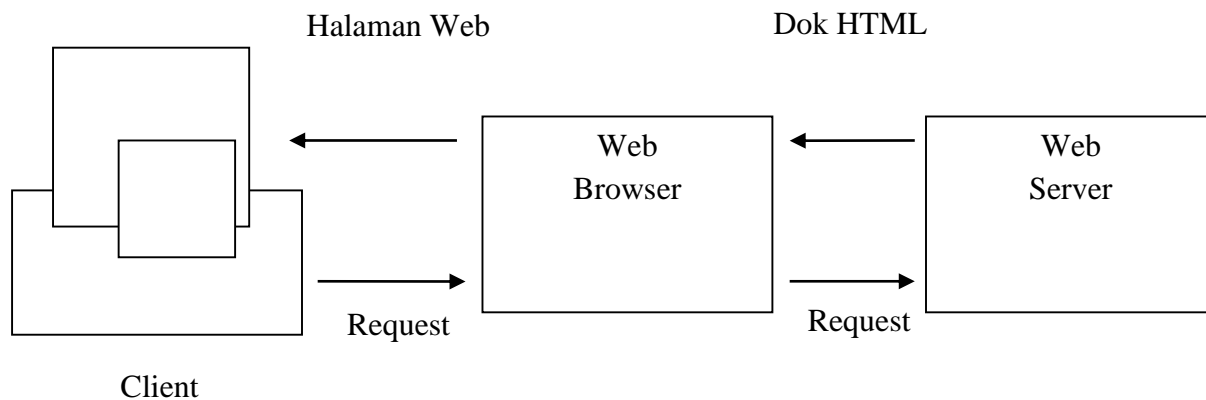
2.4. Konsep Arsitektur Sistem

- a. Menyiapkan perangkat yang dibutuhkan seperti notepad ++, adobe , web browser, xampp.
- b. Membuat rancangan database menggunakan xampp yang dipanggil melalui web browser.
- c. Mendesain tampilan menggunakan adobe dan notepad +.
- d. Membuat program untuk dapat menjalankan web melalui notepad ++ dan adobe dreamweaver.
- e. Melihat hasil web menggunakan web browser seperti google crome atau mozila.

2.5. Client Side Scripting

Client side scripting adalah salah satu jenis pemrograman web yang proses pengolahannya (baca: diterjemahkan) dilakukan disisi *client*. Porses pengolahan *client side scripting* dilakukan oleh web *browser* sebagai *client*-nya. Didalam *web browser* terdapat *library* yang mampu menerjemahkan semua perintah didalam halaman *web* yang menggunakan *client side scripting*. *Library* ini secara teknis disebut *web engine*. [3]

Dibawah ini adalah cara kerja dari *Client Side Scripting*.



Gambar 2.1 Proses Kerja Client Side Scripting

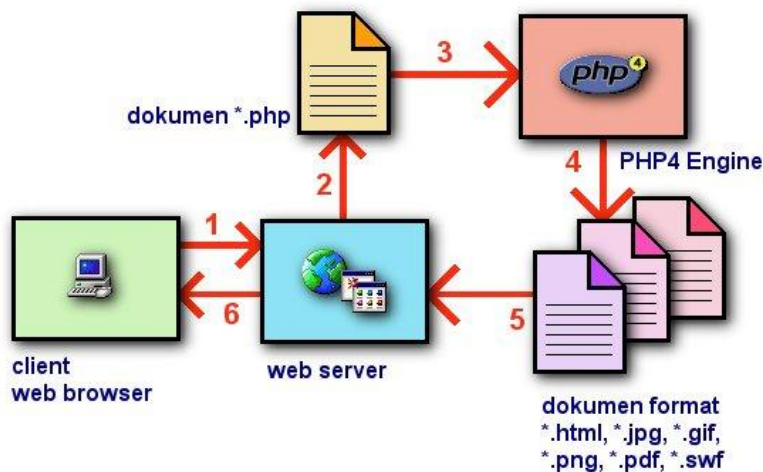
Keterangan :

- a) User meminta halaman web yang berisi *client side scripting* melalui browser.
- b) Browser mendapatkan alamat web dari web *server* dan mengidentifikasi alamat yang diminta.
- c) *Web server* mencarikan berkas yang diminta dan memberikan isinya ke browser.
- d) Browser mendapatkan isinya segera melakukan proses penerjemahan ke bentuk HTML.
- e) Halaman web diterima *client* melalui browser.

2.6. Server Side Scripting

Server Side Scripting adalah bahasa pemrograman web yang pengolahannya dilakukan disisi *server*. Maksud *server* disini adalah *web server* yang didalamnya telah diintegrasikan komponen *web engine*. Tugas *engine* adalah memproses semua *script* yang termasuk kategori *client side scripting* didalam dokumen web.

Dibawah ini adalah cara kerja dari *Server Side Scripting* :



Gambar 2.2 Proses Kerja Server Side Scripting

Keterangan :

- a) User me-*request* web dibrowser.
- b) Browser mengirimkan *request* ke *web server*
- c) Berkas PHP yang diterima web, dikirim ke mesin PHP untuk diproses.
- d) PHP *engine* menerjemahkan berkas PHP menjadi kode HTML.
- e) Setelah melalui proses, berkas kembali dikirim ke *web server*.

2.7.HTML

HTML (*Hyper Text Markup Language*) adalah sekumpulan simbol – simbol atau tag – tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman pada web browser. Tag – tag HTML selalu diawali dengan <x> dan diakhiri dengan </x> dimana x tag HTML itu seperti b, I, dan u.

Dokumen HTML disusun oleh elemen – elemen pembentuknya, ini adalah dasar dari semua materi HTML. Contoh elemen html adalah : head, body, title, html, dan table.[4]

Contoh dokumen HTML :

```

<html>

    <head> <title> Judul Website </title>

    </head>

    <body>

    Konten dari HTML

    </body>

</html>

```

2.8.CSS

Cascade Sheet Style adalah sebuah set aturan yang memberikan kontrol lengkap tampilan halaman *website* dan tampilan isinya (Jamsa, 2002). Menurut Neiderst (2001) CSS bermanfaat untuk :

1. *Greater Typhography and page layout control* (mengontrol *typhography* dan tampilan halaman dengan baik).
2. *Style is separate from structure* (*style* terpisah dari struktur halaman).

3. *Potentially smaller documents* (berpotensi membuat dokumen lebih kecil).

2.9.Bootstrap

Bootstrap adalah sebuah produk *open source* yang digunakan untuk mendesain tampilan website yang dibuat oleh Mark Otto dan Jacob Thornton, dimana pada saat itu mereka adalah karyawan dari Twitter.

Bootstrap pertama kali diperkenalkan atau diluncurkan pada Agustus 2011 pada *event Hackweek*. Dari situ popularitas Bootstrap mulai meningkat, tidak hanya menyediakan tampilan yang menarik Bootstrap juga menyediakan sejumlah *plugin* javascript untuk membuat tampilan website lebih menarik dan atraktif. Yang paling menarik dari Bootstrap itu sendiri adalah terdapat fasilitas untuk membuat website yang *responsive*. Maksudnya adalah kita dapat membuat sebuah website dengan tampilan yang multi *platform*, artinya website yang dibuat dapat dibuka diberbagai ukuran layar *desktop* maupun *gadget*. [6]

Bootstrap mempunyai struktur *file* seperti dibawah ini :

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   └── bootstrap.min.css
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
├── img/
│   ├── glyphs-halflings.png
│   └── glyphs-halflings-white.png
└── README.md
```

Gambar 2.4 Struktur File Bootstrap

2.10. Apache

Apache adalah salah satu aplikasi *web server*, apache bertugas menerjemahkan *Uniform Resource Locator* (URL) menuju file, kemudian mengirimkan file tersebut melalui internet, atau ke program yang kemudian dijalankan oleh program tersebut dan mengirimkan hasilnya (Laurie, B and Laurie, P, 2002). Kelebihan-kelebihan Apache menurut Kabir (2002) adalah:

1. *Apache is highly configurable Web Server with a modular design* (didesain dengan sistem modular dan dapat dikonfigurasi).
2. *Apache is free, open source technology* (gratis, dan *source code* diberikan secara bebas).
3. *Apache works great with Perl, PHP, and other scripting language* (dapat bekerja dengan Perl, PHP, dan bahasa scripting lain).
4. *Apache runs on Linux and other Unix systems* (dapat berjalan pada Linux dan sistem Unix lainnya).
5. *Apache also runs on Windows* (dapat berjalan pada Microsoft Windows).

Web Server berfungsi untuk melayani permintaan pemanggilan alamat dari pengguna melalui web browser, dimana *web server* mengirim kembali informasi yang diminta tersebut melalui HTTP (*Hypertext Transfer Protocol*).[7]

Cara kerja Apache : [8]

- a) Browser meminta sebuah halaman `index.php`.
- b) Permintaan diterima oleh *web server* (*server* yang melayani permintaan halaman web).

- c) *Web server* menerima dan memeriksa halaman `index.php` serta memeriksa apakah halaman `index.php` menggunakan *database*.
- d) Jika menggunakan *database*, *web server* akan mengambil ke dalam *database*.
- e) Kemudian mengirim halaman `index.php`.
- f) Browser pada klien akan menampilkan halaman `index.php`.

2.11. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, 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.

2.12. XAMPP

XAMPP merupakan salah satu paket instalasi apache, PHP dan MYSQL instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut. Selain paket instalasi instant XAMPP juga memberikan fasilitas pilihan penggunaan PHP4 dan PHP5. untuk berpindah versi PHP yang ingin digunakan dilakukan dengan bantuan *PHP-Switch* yang telah disertakan oleh XAMPP, dan yang terpenting bersifat *free* atau gratis untuk digunakan (Gregorious Agung.1995).

XAMPP adalah paket *software* yang didalamnya sudah terkandung *Web Server*, Apache, *Database MySQL*, dan *PHP Interpreter*. [9]

2.13. UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan system piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. Dengan menggunakan UML model dapat dibuat untuk semua jenis aplikasi piranti lunak, dimana aplikasi dapat berjalan pada piranti keras, system operasi dan jaringan apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebu cocok untuk penulisan piranti lunak dalam berbagai bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi procedural dalam VB atau C. [10]


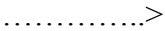

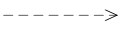
Seperti bahasa lainnya, UML mendefinisikan notasi dan syntax/semantic. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu dan UML syntax mendefinisikan bagaimana bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada

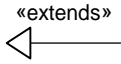

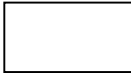

sebelumnya seperti Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*) dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). UML mendefinisika diagram sebagai berikut:

1. Use Case Diagram

Menggambarkan fungsionalitas dari sebuah sistem (apa fungsinya), yang mempresentasikan sebuah interaksi antara aktor dengan sistem (sebuah pekerjaan). Misalnya menambah data / membuat laporan. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan – pekerjaan.

Tabel 2.1 Notasi Use Case Diagram


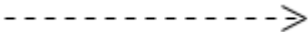
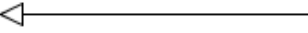
NO	GAMBAR	NAMA	KETERANGAN
1.		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan usecase.
2.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3.		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		Include	Menspesifikasi bahwa <i>use case</i> sumber secara <i>explicit</i> .

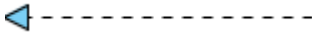

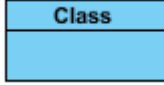

5.		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		Association	Apa yang menghubungkan antara objek satu dengan objek lain.
7.		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

2. Class Diagram

Class Diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain.

Tabel 2.2 Notasi Class Diagram

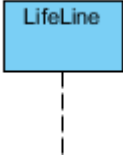
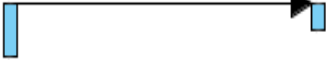
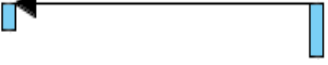
NO	GAMBAR	NAMA	KETERANGAN
1.		Association	Apa yang menghubungkan antara objek satu dengan yang lain
2.		Dependency	Hubungan dimana perubahan terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3.		Generalization	Hubungan dimana objek anak

			(<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
4.		Realization	Operasi yang benar – benar dilakukan oleh suatu objek
5.		Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari dua objek
6.		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
7.		Collaboration	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>

3. Sequence Diagram

Diagram ini menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari sebuah aktivitas tertentu, kemudian berproses mengikuti urutan tertentu, yang bisa terlihat melalui *message* antar objeknya.






Tabel 2.3 Notasi Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		Lifeline	Apa yang menghubungkan antara objek satu dengan yang lain
2.		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3.		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

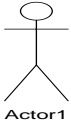


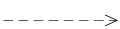
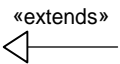

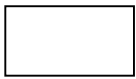
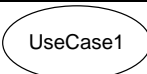
4. Activity Diagram

Diagram ini menggambarkan berbagai aktivitas dalam sistem yang sedang dirancang, mulai dari titik awal, melalui kondisi (*decision*) yang mungkin terjadi, kemudian sampai pada titik akhir.

Tabel 2.4 Notasi Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Activity</i>	Memperlihatkan bagaimana masing–masing kelas antarmuka saling berinteraksi satu sama lain
2.		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali
4.		<i>Actifity</i> <i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran


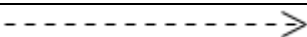



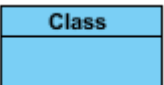

Tabel 2.1 Notasi Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan usecase.
2.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3.		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		Include	Menspesifikasi bahwa <i>use case</i> sumber secara <i>explicit</i> .
5.		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		Association	Apa yang menghubungkan antara objek satu dengan objek lain.
7.		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

1. Class Diagram

Class Diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain.

Tabel 2.2 Notasi Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		Association	Apa yang menghubungkan antara objek satu dengan yang lain
2.		Dependency	Hubungan dimana perubahan terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3.		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
4.		Realization	Operasi yang benar – benar dilakukan oleh suatu objek
5.		Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari dua objek
6.		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
7.		Collaboration	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>

2.14. Framework CodeIgniter

Framework – sebagaimana arti dalam Bahasa Indonesia yaitu kerangka kerja – dapat diartikan sebagai kumpulan dari *library (class)* yang bisa diturunkan, atau bisa langsung dipakai fungsinya oleh modul – modul atau fungsi yang akan di kembangkan.

Sedangkan pengertian *CodeIgniter* adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun website dinamis. Dengan menggunakan PHP CodeIgniter akan memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuat dari awal