

PERANCANGAN *FRAMEWORK* APLIKASI WEB WiPHP

Andri¹⁾, Arifin²⁾, Wilson³⁾

¹Program Studi Manajemen Informatika STMIK Mikroskil

²Program Studi Teknik Informatika STMIK Mikroskil

Jl. Thamrin No. 124, 140 dan 112 Medan 20212

Telp : (061) 4573767, Fax : (061) 4567789

E-mail : andri@mikroskil.ac.id¹⁾, arifins2c@yahoo.com²⁾, 081110133@students.mikroskil.ac.id³⁾

Abstrak

Pada umumnya dalam mengembangkan sistem berbasis web yang cukup besar membutuhkan banyak pemrogram tetapi para pemrogram memiliki cara kerja, style, manajemen script yang berbeda-beda sehingga dalam penyelesaian masalah dapat memakan waktu yang lama dan menimbulkan masalah terhadap kelangsungan siklus sistem bila terjadi perubahan bisnis proses di dalam sistem atau bila pemrogram bersangkutan berhalangan hadir atau telah mengundurkan diri. Teknik pengembangan sistem demikian dapat menyebabkan kerumitan dalam hal pengembangan dan pemeliharaan. Oleh karena itu perlu dibuat sebuah *framework* untuk pengembangan aplikasi web berbasis PHP dengan konsep MVC (Model, View, Controller) yang diberi nama WiPHP. *Framework* yang dibuat diharapkan dapat meminimalkan ketergantungan terhadap seorang pemrogram, menghemat proses pembuatan sistem, memperpanjang siklus hidup suatu sistem dan menjadi acuan bagi akademisi dalam pembuatan *framework* aplikasi web sejenis. Untuk membuktikan *framework* yang dirancang dapat berfungsi maka dibuat suatu studi kasus berupa aplikasi berbasis web untuk manajemen data karyawan.

Kata kunci: PHP, *framework*, WiPHP, MVC

Abstract

It is common that large web based system development requires many programmers, however, each programmer has different way of working, style, and script management which eventually resulting in a prolonged problem solving time and problems in system lifecycle due to the changing of business process; or the absence or resignation of involved programmer. This kind of system development technique may cause further complexity in term of development and maintenance. Therefore to solve it, an MVC (Model, View, Controller) *framework* for web based application development that used PHP language called WiPHP is invented. This *framework* is expected to minimize the dependence of certain programmer, shorten system development time, lengthen system lifecycle and as guidance for academics in inventing similar web application *framework*. To prove that the developed *framework* can function well, this research took a case study in developing web based application for employee data management.

1. PENDAHULUAN

Dewasa ini perkembangan perusahaan telah mendorong pemanfaatan teknologi informasi untuk mengembangkan bisnis, dimana dibutuhkan tenaga pemrogram untuk merancang suatu sistem berbasis web, dan kebanyakan pemrogram memiliki cara kerja, style, manajemen script yang berbeda-beda. Sehingga dalam proses pengembangan sistem akan memakan waktu yang lama, karena perlu memulai dari awal, seperti perancangan basis data, perancangan desain web, dan validasi bisnis proses. Hal ini tentu menimbulkan masalah terhadap kelangsungan siklus sistem bila terjadi perubahan bisnis proses di dalam sistem atau bila pemrogram bersangkutan berhalangan hadir atau telah mengundurkan diri. Ada tiga pilihan penyelesaian terhadap masalah diuraikan sebelumnya, yaitu membiarkan kondisi saat ini dan menerima semua resiko, migrasi ke bahasa pemrograman lain yang lebih terstruktur, menggunakan *framework* untuk menciptakan standar pemrograman. Dari beberapa opsi tersebut, opsi menggunakan *framework* cukup menarik seperti *framework* CodeIgniter, CakePHP, Symfony, Zend atau Akelos. Di sini muncul keinginan untuk menghasilkan *framework* yang berfokus pada aplikasi bersifat transaksi dan diharapkan dapat meningkatkan efisiensi pengembangan program dan memperpanjang siklus hidup program sehingga dirancang suatu *framework* pengembangan aplikasi web bernama WiPHP. Beberapa fitur dari WiPHP: a) *Framework* memakai konsep MVC (Model, View, Controller) dengan berfokus pada *back end*. b) Khusus untuk aplikasi yang bersifat transaksi disediakan pustaka dalam bentuk *class* untuk *form* dan *view* untuk mempermudah pembuatan tampilan. *Class-class* ini telah baku dan tidak ada fasilitas *custom*. c) *Framework* menerapkan *javascript* untuk menampilkan LOV (List of Value) dari suatu master data dengan fasilitas *filter* dan pengambilan informasi dari suatu kode master. d) *Framework*

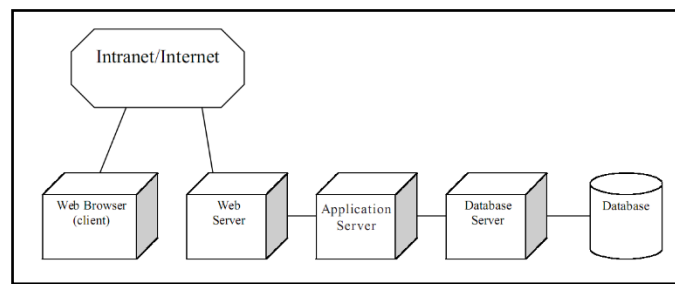
mendukung *Web Service* dengan SOAP (*Simple Object Access Protocol*) untuk menampilkan data secara *custom*. e) *Framework* berbasis OOP. f) Penerapan *framework* dalam bentuk *controller form entry* data berupa *add, edit, delete, view*, dan *filter, controller view, controller report*. g) Standar menu dalam *framework* ini telah baku, tidak ada fasilitas *custom*. h) *Language* dalam *framework* ini telah baku sesuai tulisan yang ditulis dalam label *Framework* pada pembuatan *controller*.

2. KAJIAN PUSTAKA

Kajian pustaka terkait dengan perancangan *framework* aplikasi web WiPHP meliputi arsitektur aplikasi berbasis web, *web application framework* dan konsep *Model, View, Controller* (MVC).

2.1 Arsitektur Aplikasi Berbasis Web

Untuk menjalankan fungsinya, sebuah aplikasi berbasis web harus menjalankan serangkaian urutan pekerjaan yang melibatkan beberapa komponen perangkat keras, sebagaimana ditampilkan pada gambar 1. Pertama pengguna berinteraksi menggunakan peramban dengan *script* pemrograman sisi klien seperti HTML dan JavaScript. Kemudian melalui jaringan internet atau intranet permintaan yang dikirimkan klien diterima oleh *web server* untuk kemudian diolah oleh *application server*. Jika membutuhkan pengambilan data dari basis data, *Application Server* memberikan perintah pada *Database Management System (DBMS)* untuk mengubah atau sekedar membaca data. Terakhir dari basis data proses berbalik kembali ke antarmuka pengguna yang menggunakan peramban internet [1].



Gambar 1. Komponen Aplikasi Berbasis Web [1]

2.2 Web Application Framework

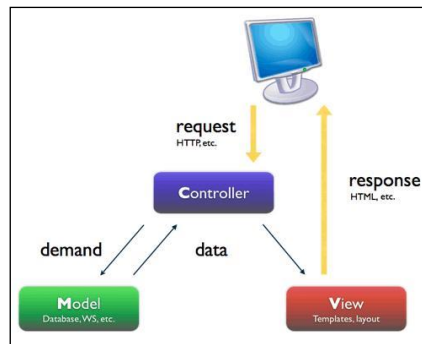
Kata *framework* sering muncul dalam literatur ilmu komputer, terutama dalam domain rekayasa perangkat lunak dan sistem terdistribusi. Namun, masing-masing penulis dan peneliti memiliki pengertian sendiri yang spesifik mengenai *framework*. Hal inilah yang menyulitkan pembaca untuk mengidentifikasi *framework* manakah yang sedang dibicarakan oleh si penulis. Satu hal yang sama dari pengertian-pengertian *framework* tersebut adalah *framework* lebih menitikberatkan kepada struktur bukan isi [2].

Web application framework adalah rancangan umum yang mungkin diterapkan pada arsitektur aplikasi berbasis web, mencakup konseptual, navigasional, dan juga aspek antarmuka. *Web application framework* biasanya menyediakan fungsi-fungsi yang berhubungan dengan manajemen sesi, penyimpanan data, pembuatan *template* tampilan dan lain sebagainya. Hampir semua *framework* paling tidak mengatur beberapa aspek dari alur kerja perangkat lunak yang dibangun, sebagaimana *framework* untuk aplikasi berbasis web mengatur rute dari *uniform resource locator* (URL) [3].

2.3 Konsep Model, View, Controller (MVC)

Konsep *Model-View-Controller* atau MVC adalah sebuah metode untuk membuat sebuah aplikasi web dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memrosesnya (*Controller*). Dalam implementasinya kebanyakan *framework* dalam aplikasi *website* adalah berbasis arsitektur MVC seperti terlihat pada gambar 2. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web [3].

- *Model* –mengandung fungsi-fungsi untuk berkomunikasi dengan basis data, seperti: mengambil data produk dengan id tertentu, hapus produk, menyimpan komentar.
- *View* –kode yang mengeluarkan HTML yang akan dilihat oleh pengguna.
- *Controller* –kode yang memeriksa input dari pengguna dan memilih apa yang harus dilakukan.



Gambar 2. Konsep MVC

3. ANALISIS DAN PERANCANGAN *FRAMEWORK* SERTA HASIL IMPLEMENTASI

Berikut ini diuraikan dasar-dasar analisis kebutuhan *framework* WiPHP dan perancangan apa saja yang dilakukan serta implementasi *framework* ini pada studi kasus.

3.1 Analisis

Untuk melakukan perancangan WiPHP dilakukan beberapa analisis yaitu analisis 2 (dua) *framework* PHP populer (*CodeIgniter* dan *Symfony*) dan analisis aplikasi bisnis. Dari hasil analisis kedua *framework* PHP populer tersebut, dapat diambil beberapa dasar pemikiran pembuatan *framework* yaitu membagi beberapa kode web ke dalam beberapa *folder* dengan alasan untuk memberi kemudahan dalam hal perawatan kode program dan peletakan kode menjadi lebih terstruktur. Untuk menganalisis aplikasi bisnis, dilakukan pengamatan aplikasi *desktop* yang dikembangkan dengan bahasa Visual Basic 6.0 yang digunakan secara internal oleh PT. Capella Dinamik Nusantara untuk manajemen data perusahaan secara internal. Dari hasil pengamatan terhadap aplikasi tersebut, ada beberapa fitur yang diadopsikan ke dalam *framework* WiPHP. Berdasarkan hasil analisis tersebut, maka didefinisikan daftar kebutuhan untuk menunjang fitur *framework* WiPHP yang dapat dilihat pada tabel 1 yaitu:

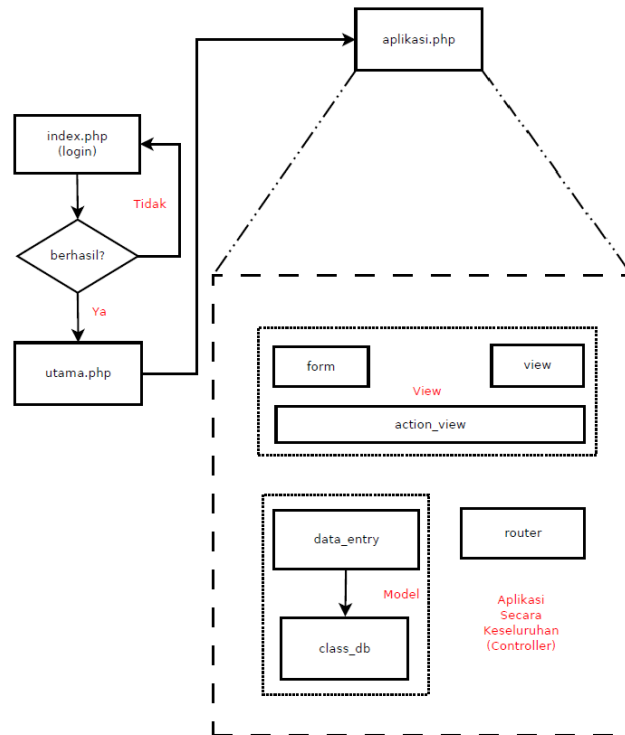
Tabel 1. Daftar Analisis Kebutuhan Framework WiPHP

No	Daftar Kebutuhan	Dasar Pemikiran
1	<i>Framework</i> menggunakan <i>javascript</i> untuk : <ul style="list-style-type: none"> Menampilkan <i>List of Value</i> (LOV) dari suatu master data dengan fasilitas <i>filter</i>. Untuk pengambilan informasi dari suatu kode master. 	<ul style="list-style-type: none"> Fitur yang berdasarkan fungsi <i>HelpTable</i>. Fitur mengembalikan nilai fungsi <i>HelpTable</i>.
2	Manajemen file diatur sesuai kebijakan standar yang dikehendaki dengan sedikit modifikasi dalam pengaturan yang disimpan dalam basis data.	Hasil analisis <i>framework</i> dimana file bisa diatur lebih terstruktur.
3	Validasi sisi klien dengan <i>javascript</i> yang bisa dikembangkan / dimodifikasi secara <i>custom</i> sesuai kebutuhan.	Fitur yang berdasarkan fungsi <i>DataNavigator</i> .
4	Tersedia standar untuk <i>filter</i> data sehingga menghemat <i>coding</i> untuk keperluan <i>view</i> data.	Fitur yang berdasarkan fungsi <i>DataViewer</i> .
5	Tersedia prosedur standar yang akan dilewati untuk validasi data sebelum data disimpan ke basis data. Prosedur standar untuk masing-masing proses yaitu tambah data baru, edit data lama dan hapus data.	Fitur yang berdasarkan fungsi <i>DataNavigator</i> .
6	<i>Framework</i> mendukung <i>Web Service</i> dengan SOAP (<i>Simple Object Access Protocol</i>) untuk menampilkan data.	Fitur yang berdasarkan fungsi <i>CrystalReport</i> .
7	Tersedia fasilitas <i>script</i> untuk <i>printer friendly</i> (untuk <i>controller view</i> untuk <i>report</i>).	Fitur yang berdasarkan fungsi <i>CrystalReport</i> .
8	Tersedia fasilitas <i>pagination</i> (untuk <i>controller view</i>) untuk membagi data ke dalam beberapa halaman.	Fitur yang berdasarkan fungsi <i>ListingTable</i> dan <i>DataViewer</i> .

3.2 Perancangan

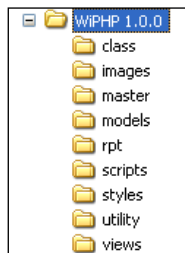
Perancangan *framework* WiPHP meliputi:

- Arsitektur WiPHP dapat dilihat pada gambar 3.



Gambar 3. Arsitektur WiPHP

- b. Struktur *folder* dapat dilihat pada gambar 4.



Gambar 4. Struktur Folder WiPHP

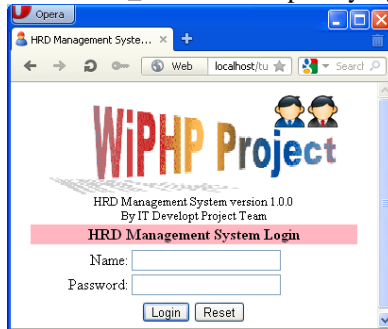
- c. Perancangan standar meliputi standar akses, standar kolom untuk setiap tabel, standar aksi untuk setiap form dan standar tabel pada basis data. Selain itu dilakukan perancangan fungsi utama terdapat pada file index.php yang berisikan file HTML untuk login dan pengecekan validasi login (sesi pengguna dibuat jika login berhasil), library.php yang berisikan fungsi-fungsi utama untuk kebutuhan *framework*, logout.php untuk menghapus sesi pengguna dan utama.php yang merupakan *controller* untuk menu-menu yang berada dalam *framework*. Kemudian dilakukan perancangan parameter diletakkan pada file var.php yang berisikan parameter untuk pengaturan URL, lokasi server basis data, *username*, *password* dan nama basis data, nama perusahaan dan pengaturan warna.
- d. Perancangan *class* dan fungsi meliputi *class* data_entry yang berfungsi sebagai prosedur standar untuk pengecekan validasi data dan beberapa fungsi tambahan untuk mendukung koneksi ke basis data, *class* def_class sebagai *template* untuk menampilkan pesan kesalahan apabila terjadi kesalahan pada suatu *class*, *class* db_class sebagai *class* untuk mengakses basis data seperti melakukan *query select*, *insert*, *update* atau *delete*, *class* find untuk melakukan pemeriksaan dan pencarian kode pada suatu tabel, *class* form sebagai *template* untuk menghasilkan kode HTML untuk mempercepat pembuatan tampilan berbentuk form, *class* view sebagai *template* untuk menghasilkan kode HTML agar mempercepat pembuatan tampilan daftar data berbentuk tabel dengan tombol ubah dan hapus untuk setiap baris data dan tambahan fasilitas pembatasan hak akses, *class* action_view sebagai *template* untuk mengatur *class* form dan *view* agar memiliki tampilan yang standar, *class* List of Value (LoV) sebagai *template* untuk menghasilkan kode HTML dalam menampilkan LoV dari suatu tabel, *class* printer friendly sebagai penghasil kode HTML yang bisa langsung dicetak ke *printer*, fungsi untuk fasilitas **SOAP** sebagai *web service*, perancangan kode untuk fasilitas **AJAX** yang berfungsi untuk mengembalikan nilai yang diminta oleh pengguna, *class* router yang berfungsi sebagai sifat dasar dari *controller*, perancangan kode standar untuk login dan pengatur aplikasi.

- e. Perancangan instalasi. Untuk melakukan instalasi, cukup ekstraksi file sumber WiPHP, yaitu file WiPHP_1.0.0.zip, kemudian letakan *folder* hasil ekstraksi di *DocumentRoot* web server.
- f. Perancangan konfigurasi WiPHP yaitu dengan mengubah nama *folder* WiPHP_1.0.0 menjadi nama *project*, misalnya *dasar_tutorial* maka *website* bisa diakses lewat http://localhost/dasar_tutorial, tapi sebelumnya jangan lupa melakukan sedikit modifikasi pada bagian konfigurasi WiPHP. Buka file *var.php*. Yang perlu diubah untuk instalasi awal ini adalah pada bagian *MyRootWebDir* seperti yang terlihat pada gambar 5:

```
$TEXT['MyRootWebDir'] = "http://localhost/dasar_tutorial/";
```

Gambar 5. Konfigurasi Root Directory pada *var.php*

Setelah itu buka alamat http://localhost/dasar_tutorial/. Tampilan yang dihasilkan seperti gambar 6.



Gambar 6. Tampilan Standar WiPHP

- g. Perancangan konfigurasi basis data dilakukan pada file *var.php* seperti gambar 7 berikut:

```
$TEXT['DB_SERVER'] = "localhost"; //lokasi server
$TEXT['DB_USER'] = "root"; //user login ke basis data
$TEXT['DB_PASS'] = ""; //password login ke basis data
$TEXT['DB_NAME'] = "wiphp"; //nama database yang digunakan
```

Gambar 7. Konfigurasi Basis Data pada *var.php*

- h. Konfigurasi *routing* digunakan untuk memetakan permintaan ke *class controller website* yang dibuat. Misalnya jika dibuka alamat <http://www.nama-website.com> (permintaan tersebut tidak menyertakan nama *controller* apa yang ingin dibuka) secara *default framework* mengarahkan ke *controller* utama yaitu *utama.php*. Adapun pola alamat URL yang digunakan untuk memanggil *class* dan juga parameter (atribut) seperti terlihat pada gambar 8.

[http://wson.com/utama.php/data_group?aksi=...\[&id=...\]](http://wson.com/utama.php/data_group?aksi=...[&id=...])

■ wson.com: alamat host	■ data_group: nama class controller
■ utama.php: berkas router	■ daftar parameter

Gambar 8. Konfigurasi Routing

- i. Perancangan *library* atau pustaka untuk menyediakan fungsi-fungsi yang bisa digunakan dalam pembuatan *website*. Misalnya saja pustaka untuk akses ke basis data, pustaka untuk *generator* form, pustaka untuk *generator* listing tabel, pustaka untuk LoV dan lain-lain.

3.3 Hasil Implementasi

Untuk memastikan *framework* WiPHP berfungsi dengan baik, maka dibuat sebuah web pengelolaan sumber daya manusia yang digunakan oleh bagian personalia dengan mengimplementasikan *framework* ini dengan fitur-fitur yaitu menginput perekrutan karyawan, perubahan jabatan, data karyawan berhenti, pemberian memo kepada karyawan, melihat laporan (profil karyawan, daftar karyawan, perpanjangan premi, penambahan premi, pengurangan premi, perekrutan karyawan dan pemberhentian karyawan). Salah satu tampilan dari hasil implementasi *framework* ini dapat dilihat pada gambar 9.

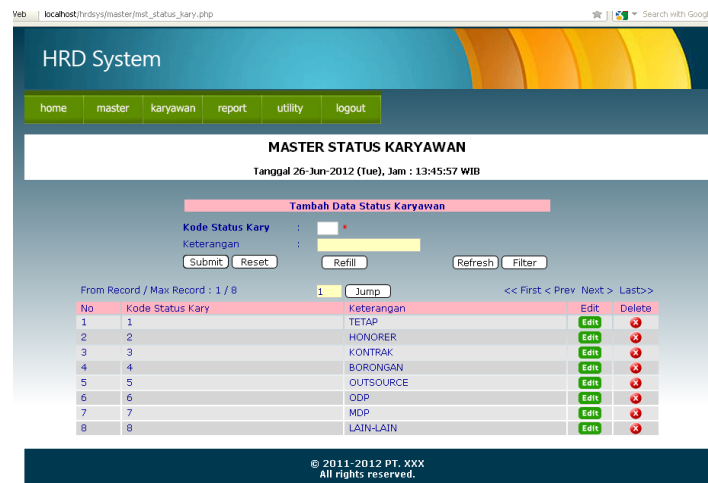
4. SIMPULAN DAN SARAN

Berdasarkan hasil analisis dan perancangan serta uji coba WiPHP dalam penyelesaian studi kasus, maka dapat diambil beberapa simpulan dan saran.

4.1 Simpulan

Ada beberapa simpulan yang dapat diambil terkait dengan *framework* WiPHP:

1. *Framework* WiPHP cocok diterapkan pada pengembangan aplikasi yang sederhana seperti aplikasi dengan satu tampilan form untuk satu tabel dalam basis data.
2. Penggunaan *List of Value* (LOV) pada *framework* WiPHP membuat aplikasi berbasis web menjadi dinamis seperti aplikasi berbasis *desktop*.
3. Penggunaan AJAX untuk mengambil informasi dari suatu kode master pada *framework* WiPHP memberikan suatu tambahan informasi kepada pengguna aplikasi.
4. Penggunaan variabel *session* secara langsung menyebabkan penggunaan sesi dibagi pakai oleh beberapa proyek meskipun proyek berada dalam *folder* yang berbeda dalam satu komputer.



Gambar 9. Implementasi *Framework* WiPHP pada Web Manajemen Sumber Daya Manusia

4.2 Saran

Berikut ini beberapa saran yang diharapkan dapat menjadi bahan pertimbangan bagi para pengembang web:

1. Untuk penelitian selanjutnya, dapat dilakukan penyederhanaan *class* atau fungsi yang sering dipakai / dibutuhkan dalam *framework* dan pengembangan agar bisa mendukung transaksi untuk *master detail* (satu tabel induk dan beberapa tabel anak).
2. Sering pemrogram mempunyai pustaka sendiri selain pustaka yang telah disediakan. Untuk dapat digunakan pada *framework* WiPHP terpaksa harus mengubah kode inti, padahal salah satu syarat *framework* yang baik adalah dapat dikembangkan dengan mudah (*extensible*). Untuk itu diperlukan pembaharuan kode untuk mendukung penggunaan pustaka-pustaka lain tanpa harus mengubah kode inti (*hacking the core*).
3. Pemrogram dapat menggunakan *framework* WiPHP sebagai perangkat alternatif pengembangan perangkat lunak berbasis web, untuk perangkat lunak yang harus dibangun dengan cepat, dan juga perangkat lunak yang dibangun bersama-sama oleh banyak pemrogram.
4. Pustaka yang disediakan pada *framework* WiPHP masih kurang lengkap untuk dapat digunakan pada pengembangan perangkat lunak yang lebih kompleks. Untuk itu perlu dilengkapi pustaka yang berhubungan dengan web seperti pengaturan tampilan pesan kesalahan, *versioning*, *database abstraction*, manajemen sesi, *testing unit*, *data sanitization* dan juga XML-RPC.
5. *Framework* adalah perangkat bantu pemrograman yang dibangun dengan abstraksi di atas bahasa pemrograman. Untuk dapat menggunakannya secara optimal pengguna (pemrogram) disarankan menguasai bahasa pemrograman (dalam konteks ini adalah PHP) terlebih dahulu.

5. DAFTAR RUJUKAN

- [1] Kurniawan, F., 2007. *Kebutuhan TI dan Aplikasi Berbasis Web*. Yogyakarta: Teknologi Informasi.
- [2] Priadi, F. M., 2011. *Framework Mobile Payment Menggunakan Barcode 2d Sebagai Media Transfer Informasi Pengguna*. Fakultas Matematika Dan Ilmu Pengetahuan Alam, Thesis: Universitas Gadjah Mada, Yogyakarta
- [3] Bari, A. dan Syam, A., 2008. *CakePHP Application Development: Step-by-step introduction to rapid web development using the open-source MVC CakePHP framework*. Birmingham: Packt Publishing.