

## 1. Pendahuluan

Perkembangan teknologi *web* membuat proses pengolahan dan penyimpanan data beralih dari yang semula *manual* ke berbasis desktop, kemudian sekarang telah berbasis *server*. Teknologi *web*, semua data disimpan ke dalam *server* untuk kemudian disediakan antarmuka yang dapat menerima, mengambil, melihat data-data tersebut kembali dari tempat yang tidak terbatas dengan syarat terkoneksi internet.

Fenomena tersebut membuat pengguna merasa lebih nyaman di dalam melakukan operasi data. Tidak terbatas waktu, tidak terbatas tempat juga tidak terbatas teknologi. Khusus untuk teknologi *web*, dibutuhkan perangkat lunak browser untuk mengaksesnya.

Keuntungan-keuntungan tersebut membuat perkembangan teknologi *web* berjalan dengan pesat. Maka muncul banyak sekali perangkat lunak, pustaka dan alat bantu untuk mengembangkan *web* yang baik, cepat dan sesuai dengan standard.

Zend *Framework* merupakan jawaban dari perkembangan tersebut. Teknologi PHP memegang peranan penting di dalam proses pengembangan *web*. Zend, pembuat PHP telah membuat *framework* yang dapat menghasilkan *web* dengan standard. Di dalam *framework* tersebut telah tersedia *library* yang lengkap untuk membangun *website* yang baik sesuai dengan standard pemrograman PHP.

Pada penelitian ini, akan mengangkat *framework* tersebut sebagai teknologi yang akan digunakan untuk membangun *website*. Adapun studi kasus yang dipakai adalah sistem pendaftaran siswa baru di MTS Negeri Salatiga. Di MTS N Salatiga belum ada sistem pendaftaran berbasis *online*, yang awalnya bersifat *konvensional* atau masih menggunakan cara manual seperti mengisi formulir pendaftaran secara tertulis. Pendaftaran siswa berbasis *online* yang sudah ada di sekolah lain belum ada tes *online* nya.

Sistem pendaftaran merupakan studi kasus yang dapat dijadikan implementasi penerapan Zend *Framework* dan menerapkan konsep MVC di dalamnya. Zend merupakan *framework* dengan *library* yang lengkap, kelemahannya adalah *website* menjadi berat. Walaupun berat, tapi ada kelebihan menggunakan Zend *Framework*. Zend *Framework* akan mendukung *Web* di masa Depan yaitu *web* dengan konsep 2.0.

## 2. Tinjauan Pustaka

Terdapat dua penelitian terdahulu yang berhubungan dengan penelitian yang dilakukan. Adapun penelitian terdahulu digunakan untuk memperlihatkan andil yang dilakukan sehingga tidak terjadi penelitian yang sama dan diharapkan ada perbedaan sehingga penelitian yang dilakukan penulis dapat bermanfaat.

Penelitian yang pertama adalah penelitian yang berjudul "Rancang Bangun Sistem Informasi Penerimaan Siswa Baru *Online*" oleh Pusparani Sholikhah Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Insitut Teknologi Sepuluh Nopember Tahun 2008[1].

Pada penelitian tersebut, dirancang dan diimplementasikan sebuah sistem yang bernama PSB (Penerimaan Siswa Baru) *Online*. Di dalam sistem tersebut, terdapat beberapa modul seperti: Pengelolaan Sistem, Pengelolaan Akun Pengguna, Pengelolaan Rekomendasi, Pengelolaan Data, Pengelolaan Registrasi, Pengelolaan Pengunduran Diri.

Modul pengelolaan sistem adalah modul yang digunakan untuk menangani proses *login* dan *logout* pengguna sistem. Modul pengelolaan akun pengguna yaitu modul yang menangani penambahan, pengubahan atau penghapusan akun yang dapat menggunakan sistem. Modul pengelolaan rekomendasi, data dan registrasi adalah modul yang digunakan untuk melakukan manajemen pendaftaran. Modul pengunduran diri untuk menangani bagi calon siswa yang mengundurkan diri.

Sistem PSB *Online* tersebut diimplementasikan menggunakan bahasa pemrograman PHP dengan menggunakan *CodeIgniter Framework*. Sedangkan *database* yang digunakan adalah MySQL. Untuk pengujian, sistem diuji menggunakan metode *Test Case*.

Kesimpulan dalam penelitian tersebut adalah terdapat empat macam pengguna yang ada pada sistem yaitu super admin, admin, operator dan calon siswa. Spesifikasi kebutuhan modul seperti yang telah dijelaskan sebelumnya. Sistem informasi PSB *Online* tidak memiliki kebutuhan non fungsional, maka pada saran pengembangan disebutkan bahwa diperlukan adanya wawancara untuk mendapatkan kebutuhan non fungsional.

Penelitian yang kedua adalah penelitian yang berjudul "Analisis dan Perancangan Modul Aplikasi Penerimaan Mahasiswa Baru Berbasis SOA" oleh Toninetti, Binus Jakarta Tahun 2009. Pada penelitian tersebut dikembangkan sebuah sistem dengan nama PSB (Penerimaan Siswa Baru) yang berbasis SOA (*Services Oriented Arsitek*) dengan memiliki tiga modul utama yaitu modul penjualan formulir, modul registrasi dan modul penerimaan mahasiswa[2].

Sistem PSB dibangun menggunakan teknologi SOA dan dikembangkan menggunakan konsep *Agile Software Development*. Database yang digunakan adalah *database* yang telah ada pada sistem pendaftaran secara reguler di Binus.

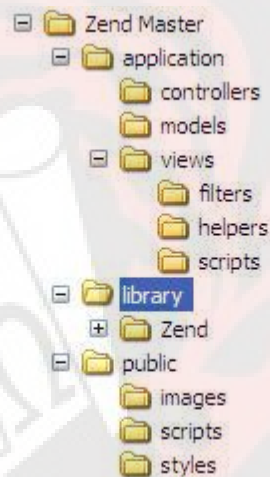
Kesimpulan yang dapat diambil dari penelitian tersebut adalah metodologi baru yaitu *Agile Software Development* dapat mempercepat proses pengembangan sistem informasi. Sistem yang dikembangkan dengan menggunakan konsep SOA, memiliki tingkat *usability* yang tinggi. Database yang telah dirancang memiliki service sistem yang akurat.

Pada dua penelitian sebelumnya implementasi dilakukan menggunakan teknologi yang berbeda-beda. Pada penelitian ini akan menggunakan *Zend Framework* untuk melakukan implementasi dan menggunakan MySQL sebagai *database* sistemnya. Yang dapat diambil dari kedua penelitian sebelumnya adalah modul-modul yang akan dibuat pada penelitian ini.

*Framework* adalah sekumpulan perintah/fungsi dasar yang dapat membantu dalam menyelesaikan proses-proses yang lebih kompleks. Mengapa harus *Framework* ? Sebenarnya bukan keharusan untuk menggunakan *framework*. Namun, seperti pengertian diatas, *framework* dapat membantu menyelesaikan

pekerjaan yang lebih kompleks. Dari banyaknya library yang telah disediakan oleh *ZendFramework* yang dipakai adalah *Zend\_Controller*, *Zend\_Model*, *Zend\_Session*, *Zend\_Db*, *Zend\_File*, *Zend\_Paginator* dan *Zend\_Form*.

Teknologi *Zend Framework* adalah merupakan salah satu *framework* yang mengimplementasikan *MVC pattern* untuk pengembangan aplikasi *web* di mana *Zend Framework* memisahkan bagian yang menjadi *Controller*, *Model* dan *View*[4]. *Zend Framework* yang populer tersebut dibuat oleh Zend sendiri yang merasa perlu untuk membuat suatu *framework* serta *library* yang dibangun di atas teknologi *php5* dengan mendayagunakan kelebihan pemrograman berorientasi objek yang telah disediakan serta menghasilkan *library* yang lebih maju dan lengkap untuk mendukung kebutuhan developer *php* pada saat ini yang mulai beralih ke pembuatan aplikasi *web 2.0*[5].



**Gambar 1** Directory Struktur *Zend Framework*

Pada Gambar 1 merupakan directory struktur *zend framework* yang berada di folder *Zend Master*:

- /Application
  - Folder Controller* berisi tempat meletakkan *controller* ( logika )
  - Folder Model* berisi tempat meletakkan *model*
  - Folder View/script* berisi tempat meletakkan *view*( *template* )
- /Library -> tempat menyimpan library zend
- /public -> tempat untuk meletakkan css

*Zend Framework* memiliki keunggulan-keunggulan seperti berikut[6]:

- Dibuat oleh Zend, pengembang engine PHP
- Dibangun dengan teknologi *php 5*
- Lisensi *Open source* berbasis BSD yang fleksibel baik untuk pengembangan aplikasi *open source* maupun komersial
- Dokumentasi lengkap (lebih dari 500 halaman)

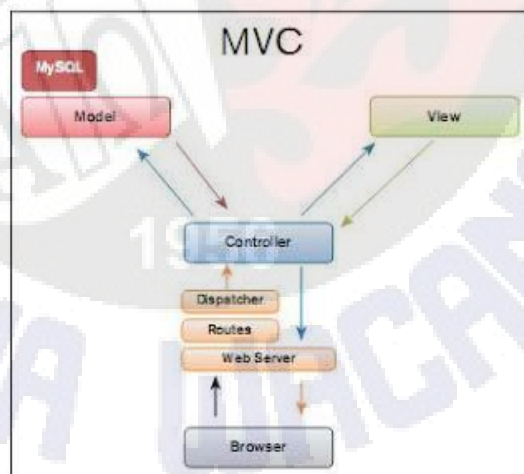
- Proses pengembangan *Zend Framework* menyertakan *unit test* yang lengkap, mencakup 84% kode yang ada
- *Library* yang menyediakan fungsi yang lengkap sehingga hampir semua kebutuhan pengembangan aplikasi *web* dapat terpenuhi
- Mendukung pengembangan aplikasi MVC
- URL yang bagus dan *search engine friendly*

Macam-macam library yang disediakan oleh *Zend Framework*[7]:

- Authentication & Authorization
- Core Infrastructure
- Documentation
- Internationalization (i18n) & Localization (l10n)
- Mail, Formats, & Search
- *Model-View-Controller* (MVC)
- *Web & Web Services*[8]

Jadi hampir semua yang diperlukan sebagai pengembang aplikasi *web* masa depan (*Web 2.0*) telah disediakan oleh PHP[9].

MVC atau *Model-View-Controller* adalah konsep pemisahan kode program menjadi tiga bagian utama. Dengan konsep ini akan menyebabkan halaman *web* dituliskan oleh programmer dengan lebih sedikit kode program dibandingkan tanpa menggunakan konsep MVC. Hal tersebut didapatkan karena telah dipisahkan antara struktur tampilan, logika dan operasi ke basis data.



**Gambar 2** Skema MVC

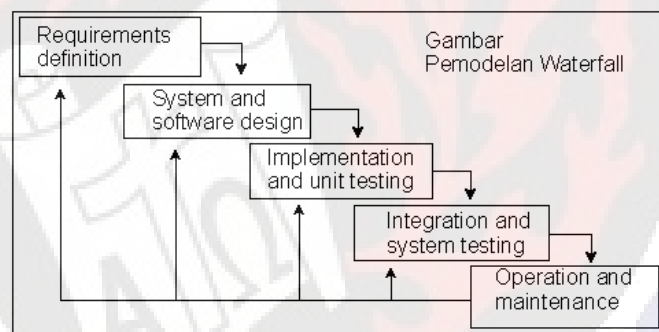
Pada Gambar 2 merupakan Skema dari MVC, terdapat beberapa komponen yang terdapat di dalam arsitektur MVC yaitu *Client* yang merupakan pengguna saat meminta request pada URL *Web Browser*. *Web Browser*, berfungsi untuk memeriksa URL yang diminta serta menangani permintaan yang sesuai dengan *controller*. *Controller* merupakan pengendali yang menghubungkan antara model dan juga view. Karena semua permintaan yang masuk, akan ditangani oleh *controller* terlebih dahulu. Dalam *controller* juga memerlukan spesifikasi logika.

Model merupakan tempat untuk representasi data atau bagian yang berhubungan dengan basis data. *View* merupakan tempat untuk memberi informasi atau menampilkan data yang diminta pengguna ke dalam tampilan layar. Secara default, hal yang terdapat di dalam *view* merupakan hal yang keliatan di halaman *web*.

### 3. Metode Penelitian

Metode Waterfall terdiri dari lima proses yang saling berurutan yaitu *Requirements Definition*, *System and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing* dan *Operation and Maintenance*. Sesuai dengan namanya waterfall (air terjun) maka tahapan dalam model ini disusun bertingkat, setiap tahap dalam metoder ini dilakukan berurutan, satu sebelum yang lainnya (lihat tanda anak panah). Selain itu dari satu tahap kita dapat kembali ke tahap sebelumnya.

#### Metode Waterfall

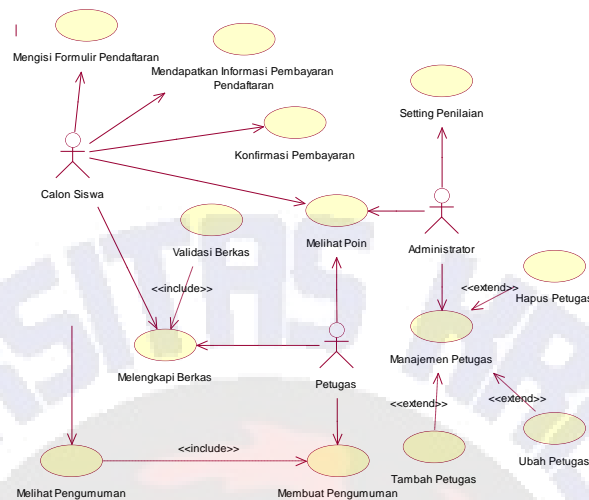


Gambar 3 Metode Waterfall [11]

1. Tahap Pertama : *Requirements Definitions*. Analisis kebutuhan pengguna, melakukan wawancara terhadap guru-guru di MTS Negeri Salatiga agar mengetahui kebutuhan sistem berdasarkan definisi mereka. yang awalnya bersifat *konvensional* atau masih menggunakan cara manual seperti mengisi formulir pendaftaran secara tertulis.
2. Tahap Kedua : *System and Software Design*. Dari hasil analisis kebutuhan, dibuatlah rancangan dengan menggunakan UML diagram yang terdiri dari *use case diagram*, *activity diagram* dan *class diagram*.
3. Tahap Ketiga : *Implementation and Unit Testing*. Proses Implementation di sini artinya adalah proses pengkodean program dari desain UML ke dalam kode program *Zend Framework*.
4. Tahap Keempat : *Integration and System Testing*. Hasil implementasi yang sudah jadi, diujikan kepada guru-guru di MTS Negeri Salatiga.
5. Tahap Kelima: *Operation and Maintenance*. Hasil pengujian akan memperlihatkan kekurangan yang ada kemudian diperbaiki pada tahap ini.



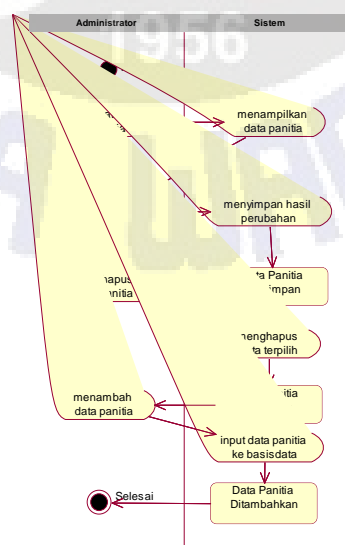
## Use Case Diagram



**Gambar 4** Diagram Use Case

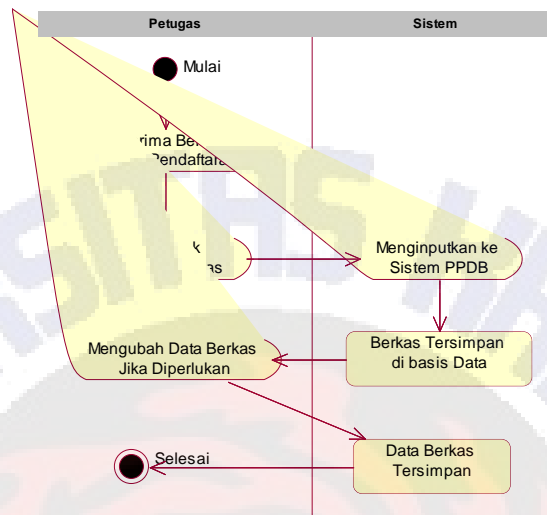
Pada Gambar 4 merupakan Perancangan *use case diagram*. terdapat tiga aktor yang dapat mengakses sistem yaitu Aktor Calon Siswa, Aktor Petugas dan Aktor Administrator. Aktor Calon Siswa dapat mengisi formulir, kemudian mendapatkan informasi pembayaran, konfirmasi pembayaran, melihat pengumuman, melengkapi berkas dan melihat poin pendaftaran. Sedangkan Aktor Petugas dapat melakukan validasi berkas, membuat pengumuman dan melihat poin dari pendaftaran Aktor Calon Siswa. Untuk Aktor Administrator dapat melakukan *setting* dan manajemen petugas berupa tambah, ubah dan hapus petugas.

## Activity Diagram



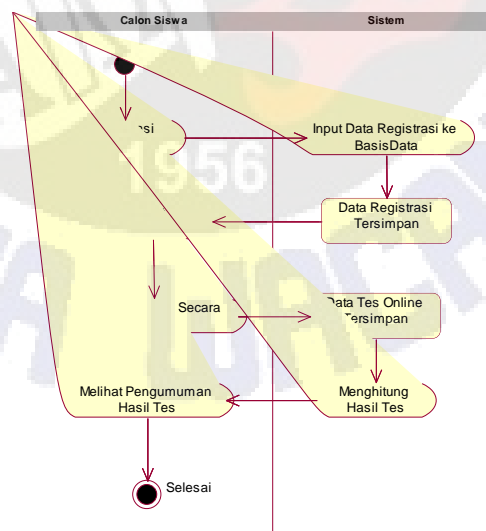
**Gambar 5** Activity Diagram Administrator

Pada Gambar 5 Aktivitas administrator di dalam sistem adalah untuk melakukan manajemen data seperti melihat semua data, mengubah data, menghapus data, dan juga menambah data pantia.



**Gambar 6** Activity Diagram Petugas

Pada Gambar 6 Perancangan *activity diagram* petugas. Petugas memiliki aktifitas di dalam sistem yaitu menerima berkas manual kemudian menginputkannya ke dalam sistem. Aktivitas berikutnya adalah Petugas melakukan perubahan Setting pada sistem yang tentunya sistem akan menyimpan setting tersebut untuk digunakan sesuai dengan keperluan.

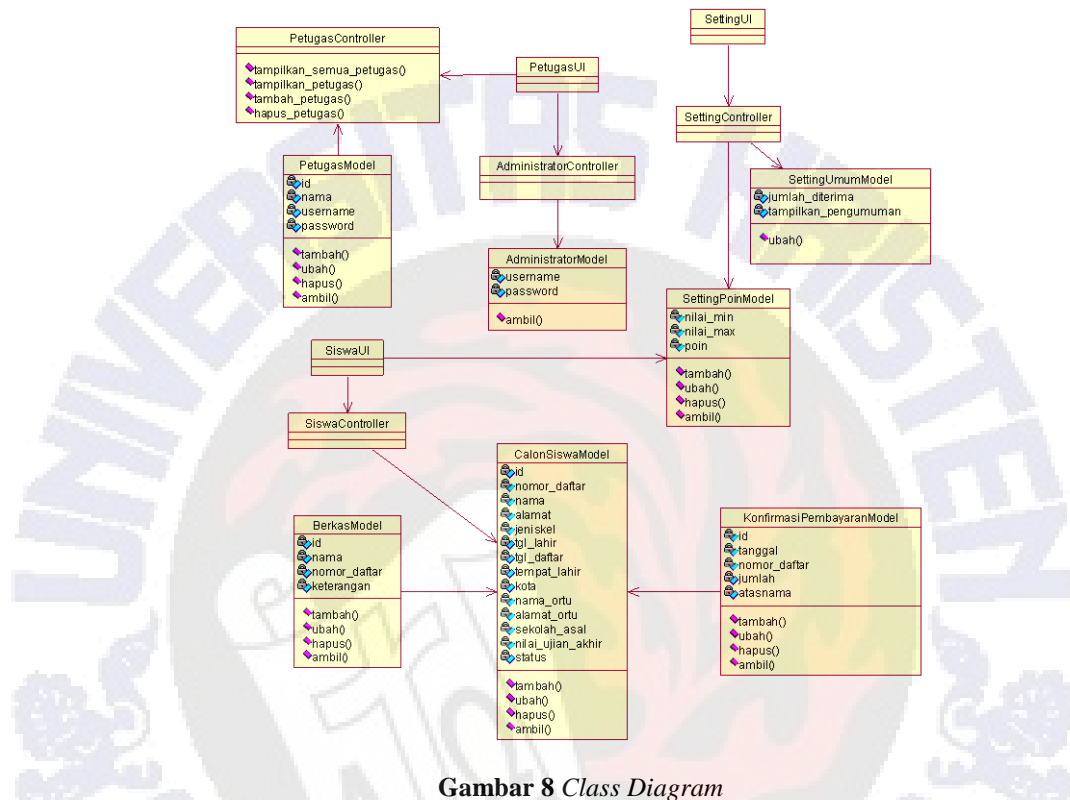


**Gambar 7** Activiy Calon Siswa

Pada Gambar 7 Perancangan *activity diagram* calon siswa dapat dilihat pada. calon siswa memiliki aktivitas dengan urutan melakukan registrasi, kemudian membayar uang registrasi dan dikonfirmasi kepada sistem.

Kemudian menunggu pengumuman untuk melihat hasil diterima atau tidak oleh hak sekolah sebagai siswa baru.

### Class Diagram



Gambar 8 Class Diagram

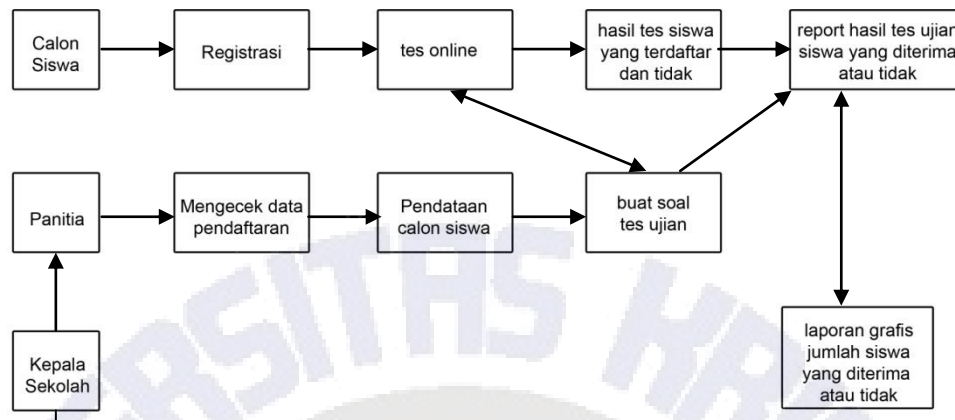
Pada Gambar 8 Perancangan *class diagram* dapat dilihat. terdapat tujuh *class model* yang terdiri dari **PetugasModel**, **SettingPoinModel**, **SettingUmumModel**, **AdminModel**, **BerkasModel**, **CalonSiswaModel** dan **KonfirmasiPembayaranModel**.

Sedangkan *Controller* yang dibutuhkan adalah **PetugasController**, **SettingController**, **AdminController**, **CalonSiswaController**, **BerkasController** dan **KonfirmasiPembayaranController**. *Controller* berguna untuk mengatur tampilan yang akan dipanggil serta menggunakan *model* jika dibutuhkan operasi yang berkaitan dengan basis data.

Setiap *Controller* akan berhubungan dengan model dan ui. Misalnya **PetugasController** akan berhubungan dengan **PetugasUI** dan **PetugasModel**. Di mana **PetugasUI** merupakan tampilan yang dapat dilihat user, sedangkan **PetugasModel** bertugas untuk manajemen yang berhubungan dengan basis data.



## Proses bisnis



**Gambar 9** Proses Bisnis

Gambar 9 merupakan proses bisnis dari sistem penerimaan dan pendataan siswa yang dilakukan di tempat studi kasus MTs Negeri Salatiga. Siswa melakukan aktivitas dengan registrasi sehingga terdaftar, lalu calon siswa menunggu informasi tes ujian masuk dan mengikuti tes online. Panitia mengecek data pendaftaran calon siswa dan mendata, lalu guru membuat soal tes ujian, sehingga didapatkan hasil Inputan berupa report hasil tes ujian siswa yang diterima atau tidak dan laporan *grafis* jumlah siswa yang diterima atau tidak diterima.

## 4. Hasil dan Pembahasan

### Hasil Implementasi Administrator

The screenshot shows a login form titled ":: Login ::". It contains three input fields: "Username :", "Password :", and "Login Sebagai :". The "Login Sebagai :" field is a dropdown menu with the text "-pilih-". Below the input fields is a "Submit" button.

**Gambar 10** Tampilkan Form Login

Pada Gambar 10, ditampilkan ditampilkan proses login yang dibentuk oleh View di dalam Zend Framework melalui kode html. *Form login* seperti pada

Gambar 7 merupakan *form* yang dipanggil dari *Controller* user dan dengan nama login.phtml. Sedangkan proses login ada pada Kode Program 1.

#### Kode Program 1. Proses Login

```
6. if($akun['pilihan']=='administrator'){
7. $admin = new Application_Model_DbTable_Admin();
8. $login = array('username'=>$akun['username'],
9. 'password'=>$akun['password']);
10. $infoakun = $admin->checklogin($login);
11. if(empty($infoakun))
12. throw new Exception();
13. else{
14. $this->_infoakun->username = $infoakun['username'];
15. $this->_infoakun->as = $infoakun['as'];
16. }
17. }
```

NO	NAMA	JABATAN	USERNAME	PROSES
1	Mohamad Idris	penanggung jawab	idrezz	
2	Kiftiyatul Deroiyati	wakil	kifty3	
3	Margiyono	bendahara	margiono	
4	Anekar Ibrahim S	bendahara	anwar	
5	Sugiman	seksi kebersihan	giman	

Gambar 11 Tampilkan Semua Data Panitia

Pada Gambar 11, hasil eksekusi menampilkan semua data panitia. Untuk menampilkan semua data panitia maka harus dipanggil model DBTable\_Panitia dari PanitiaController untuk kemudian ditampilkan di bagian views panitia/daftar.phtml. Controller dapat dilihat pada Kode Program 2.

#### Kode Program 2. Menampilkan Semua Data

```
1. public function daftarAction()
2. {
3. // action body
4. $this->active['a'] = 'class="active"';
5. $this->view->active = $this->active;
6. $this->view->title="Daftar Panitia PPD Th ".$this->tahunppd;
7. $panitia = new Application_Model_DbTable_Panitia();
8. $this->view->panitia = $panitia->semua($this->tahunppd);
9. }
```



**Gambar 12** Ubah Konten Sambutan

Pada Gambar 12, hasil eksekusi untuk melakukan perubahan konten halaman sambutan. Form ubah konten dan kontak dapat dibuat dengan memanggil data dari controller kemudian menampilkannya. Ketika dilakukan post, maka akan dipanggil fungsi ubah(\$konten).

### Kode Program 3. Ubah Konten Sambutan dan Kontak

```
1. $this->view->title="Konten Web PPD Th.".$this->tahunppd;
2. $konten = array();
3. $konten['sambutan'] = $this->ppd->sambutan();
4. $konten['kontak'] = $this->ppd->kontak();
5. $this->view->konten = $konten;
6. if($this->getRequest()->isPost()){
7. $konten = $this->getRequest()->getPost();
8. $this->ppd->ubah($konten);
9. $this->flash->addMessage('Data berhasil diubah');
10. $this->_helper->redirector('konten');
11. }
```

### Hasil Pengujian

Hasil pengujian menggunakan metode *blackbox* testing. *Black-box testing* adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian *white-box*). pengetahuan khusus dari kode aplikasi/struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau *non-fungsional*, meskipun biasanya fungsional. Perancang uji memilih input yang *valid* dan tidak *valid* dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: *unit*, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga.

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari software. Karna itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox*.

Ujicoba *blackbox* untuk menemukan kesalahan dalam beberapa kategori, diantaranya : a). Fungsi-fungsi yang salah atau hilang, b). Kesalahan *interface*, c). Kesalahan dalam struktur data atau akses *database eksternal*, d). Kesalahan performa, e). kesalahan inisialisasi dan terminasi

## 5. Simpulan

Berdasarkan pada proses penelitian, proses implementasi dan juga proses pengujian maka didapatkan hasil penelitian sebagai berikut: (1) Perangkat lunak yang telah didesain telah dapat diimplementasikan dengan baik dengan melihat hasil pengujian baik secara internal maupun pengujian secara eksternal. (2) *Model-view-controller* diimplementasikan dalam Zend dengan cara membuat folder *Model*, *View*, dan *Controller* di dalam struktur yang telah ditentukan Zend. Implementasi tersebut membuat pemrograman sistem menjadi lebih terfokus pada ketiga bagian tersebut. Setting koneksi ke basisdata juga mudah cukup menuliskan di berkas *application.ini*.

## 6. Daftar Pustaka

- [1] Sholikhah, Pusparani. 2008. *Rancang Bangun Sistem Informasi Penerimaan Siswa Baru Online*. Surabaya: Sistem Informasi, Fakultas Teknologi Informasi, Insitut Teknologi Sepuluh Nopember.
- [2] Toninetti. 2009. *Analisis dan Perancangan Modul Aplikasi Penerimaan Mahasiswa Baru Berbasis SOA*. Jakarta: Binus.
- [3] Nixon, Robin. 2009. *Learning PHP, MySQL & JavaScript*. United States of America: O'Reilly.
- [4] Siregar. 2010. *Pemrograman Web Berbasis MVC dengan Zend Framework*. Yogyakarta: Andi Publisher.
- [5] Lyman, Forrest. 2009. *Pro Zend Framework Techniques*. United States of America: Apress.
- [6] Padilla, Armando. 2009. *Beginning Zend Framework*. United States of America: Apress.

- [7] Pope, Keith. 2009. *Zend Framework 1.8 Web Application Development*. United States of America: Apress.
- [8] Vaswani, Vikram. 2010. *Zend Framework, A Beginner's Guide*. United States of America: McGraw-Hill.
- [9] Mariën, Jonas. 2011. *Zend Framework Web Services*. United States of America: Marco Tabini & Associates, Inc.
- [10] Cabral & Murphy. 2009. *MySQL Administrator Bible*. United States of America: Willey.
- [11] Jalote, Pankaj. 2002. *Software Project Management in Practice*. United States of America: Addison Wisley.
- [12] Beizer, Boris. 2005. *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. United States of America: Wiley.

