

Pembangunan *Framework* untuk Aplikasi berbasis *Web* Politeknik Negeri Batam

TUGAS AKHIR

Oleh :

Febriend Roni S 3311001035

Ristomi Novantyas 3311001066

Disusun untuk memenuhi syarat kelulusan Program Diploma III



PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2012

LEMBAR PENGESAHAN

Batam, 7 November 2012

Pembimbing,

Tandhy Bintang Parlindungan S, S.Kom

NIK. 111085

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3311001035

Nama : Febriend Roni Sianipar

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

Pembangunan *Framework* untuk Aplikasi berbasis *Web* Politeknik Negeri Batam

disusun dengan :

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 8 Juli 2013

Febriend Roni Sianipar
3311001035

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3311001066

Nama : Ristomi Novantyas

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

Pembangunan *Framework* untuk Aplikasi berbasis *Web* Politeknik Negeri Batam

disusun dengan :

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 8 Juli 2013

Ristomi Novantyas
3311001066

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, atas berkat, rahmat dan karunia-Nya, Laporan Tugas Akhir dengan judul ” *Pembangunan Framework untuk Aplikasi berbasis Web Politeknik Negeri Batam*” untuk memenuhi tugas matakuliah Tugas Akhir Program Studi Informatika Politeknik Negeri Batam Tahun Ajaran 2012-2013 telah berhasil diselesaikan tepat pada waktunya.

Dalam kesempatan ini pula penulis mengucapkan terima kasih kepada :

1. Tuhan Yang Maha Esa.
2. Orang tua saya tercinta yang selalu memberikan dukungan dan doa.
3. Bapak Tandhy Bintang Parlindungan S, S.Kom, selaku Dosen yang selalu sabar membimbing kami.
4. Ibu Hilda Widyastuti, MT, selaku Ketua Jurusan Informatika Politeknik Negeri Batam.
5. Bapak Andy Triwinarko, MT, selaku Ketua Program Studi Teknik Informatika.
6. Seluruh dosen dan instruktur jurusan Informatika yang selalu membantu.
7. Kepada teman – teman seperjuangan yang selalu memotivasi dan membantu dalam pengerjaan Laporan Tugas Akhir ini.
8. Kepada pihak – pihak lain yang telah direpotkan dan tidak dapat disebutkan satu – persatu.

Penulis menyadari dalam pembuatan Laporan Tugas Akhir ini masih banyak kekurangan yang harus diperbaiki, maka penulis mengharapkan kritik dan saran yang membangun untuk perbaikan dan pengembangan di masa – masa mendatang

Batam, 8 Juli 2013

Penulis

ABSTRAK

Pembangunan *Framework* untuk Aplikasi berbasis *Web*

Politeknik Negeri Batam

Pnb framework merupakan *PHP framework* berorientasi objek yang mengimplementasikan pola desain *Model-View-Controller* (MVC) yang bertujuan untuk memisahkan logika bisnis dari pertimbangan antarmuka pengguna agar pengembang bisa lebih mudah mengubah setiap bagian tanpa mempengaruhi yang lainnya.

Pnb framework ini dapat memudahkan Politeknik Negeri Batam untuk melakukan pengembangan terhadap aplikasi berbasis *web*nya seperti *e-learning* dan *simpol* yang selalu melakukan perubahan yang dinamis demi kemajuan pendidikan serta fasilitas pendidikannya karena memiliki kemampuan *reusability* maksimum dalam pemrograman *web*. *Pnb framework* ini memiliki sangat ringan dan memiliki kemampuan mekanisme *caching* yang bagus sehingga sangat cocok untuk pengembangan aplikasi dengan lalu lintas yang tinggi.

Pnb framework ini dibuat secara serius untuk pengembangan aplikasi *web* sehingga memiliki kemampuan lebih dalam hal efisiensi, kekayaan fitur serta kejelasan dokumentasi juga memiliki kinerja yang lebih cepat karena menggunakan teknik *lazy loading ekstensif*. Misalnya tidak menyertakan *file* kelas sampai kelas digunakan untuk pertama kalinya, dan juga tidak menciptakan objek sampai objek tersebut diakses untuk pertama kalinya.

Kata Kunci : *Framework, Lazy loading ekstensif, Model-View-Controller, PHP, e-learning.*

ABSTRACT

Pembangunan *Framework* untuk Aplikasi berbasis *Web*

Politeknik Negeri Batam

Pnb framework is object oriented PHP *framework* that implements a design pattern Model-View-Controller (MVC), which aims to separate the business logic from *user* interface considerations that developers can more easily change each part without affecting the other.

This *Pnb framework* can facilitate Batam Polytechnic to undertake the development of its *web*-based applications such as e-learning and *simpol* that always changes the dynamic for the advancement of education and educational facilities because it has the ability to maximum *reusability* in *web* programming. This *Pnb framework* has very light and has a great ability *caching* mechanism so it is suitable for developing applications with high traffic.

This *Pnb framework* was seriously created for the development of *web* applications that have more capability in terms of efficiency, feature richness and clarity of documentation also has a faster performance because it uses lazy loading extensively technique. For example, do not include the *file* class until the class is used for the first time, and also does not create the object until the object is accessed for the first time.

Keywords: *Framework*, Lazy loading extensively, Model-View-Controller, PHP, e-learning.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
LEMBAR PERNYATAAN	iv
KATA PENGANTAR	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
Bab I Pendahuluan.....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	2
I.3 Batasan Masalah	2
I.4 Tujuan	3
I.5 Sistematika Penulisan	3
Bab II Landasan Teori	4
II.1 <i>Framework</i>	4
II.2 <i>Yii Frameworks</i>	7
II.3 Codeigniter	12
II.4 PHP	14
II.5 MySQL	15
Bab III Analisis dan Perancangan	18
III.1 Analisis Pnb <i>Framework</i>	18
III.1.1 Analisa Batasan Sistem	18
III.1.2 Analisa Cara Kerja Pnb <i>Framework</i>	19
III.2 Diagram Kelas Pnb <i>Framework</i>	20
III.2.1 Diagram Kelas	20
III.2.2 Diagram Kelas Rinci	21

Bab IV Bab-Bab Hasil dan Pembahasan.....	37
IV.1 Implementasi.....	37
IV.1.1 Spesifikasi Hardware.....	37
IV.1.2 Spesifikasi Software.....	37
IV.2 Pengujian.....	37
IV.2.1 Skenario Pengujian.....	37
IV.2.2 Hasil Pengujian.....	42
Bab V Kesimpulan dan Saran.....	43
V.1 Kesimpulan.....	43
V.2 Saran.....	43

DAFTAR PUSTAKA

Lampiran I

Lampiran II

DAFTAR GAMBAR

Gambar 1 Konsep MVC.....	4
Gambar 2 Alur Kerja umum aplikasi pnb	19
Gambar 3 Rancangan Pemetaan Kelas Dasar <i>Framework</i>	20
Gambar 4 Skenario pengujian pada kelas CBehavior	40
Gambar 5 Skenario pengujian pada kelas CComponent.....	40
Gambar 6 Skenario pengujian pada kelas CModel	40
Gambar 7 Skenario pengujian pada kelas CModule	41
Gambar 8 Skenario pengujian pada kelas CStatePersister.....	41
Gambar 9 Skenario pengujian pada kelas CApplicationComponent	41
Gambar 10 Skenario pengujian pada kelas CSecurityManager	41

DAFTAR TABEL

Tabel 1 Perbandingan antara Yii <i>framework</i> dengan CI <i>framework</i>	6
Tabel 2 Batasan sistem pada Pnb <i>Framework</i>	18
Tabel 3 Diagram Kelas Rinci	36
Tabel 4 Deskripsi Kelas-kelas Tambahan.....	40

Bab I Pendahuluan

I.1 Latar Belakang

Politeknik Negeri Batam merupakan salah satu Perguruan Tinggi Negeri yang telah memanfaatkan aplikasi-aplikasi berbasis *web* dalam proses pembelajarannya. Contoh pemanfaatan aplikasi berbasis *web* yang telah diterapkan dalam proses pembelajarannya adalah *e-learning*, *simpol*, registrasi, beasiswa dan juga *website* *polibatam*. Pemanfaatan Aplikasi berbasis *web* tersebut memberikan kontribusi yang besar bagi pengembangan sistem belajar mengajar di Politeknik Negeri Batam karena dapat memperbaharui informasi dengan cepat, menghemat biaya komunikasi dan administrasi dan mudah di *maintenance*, serta bisa diakses dimana saja dan kapan saja.

Dalam melakukan perubahan yang dinamis demi kemajuan pendidikan dan fasilitas pendidikannya, Politeknik Negeri Batam telah banyak memanfaatkan aplikasi berbasis web seperti *simpol* yang merupakan website akademik, *e-learning* dalam sistem pendidikannya, *digilib* yang merupakan website perpustakaan digital, beasiswa yang merupakan website untuk mahasiswa yang ingin melakukan pendaftaran beasiswa secara online dan registrasi yang merupakan website untuk melakukan pendaftaran secara online bagi yang ingin mendaftar untuk kuliah di Politeknik Negeri Batam.

Dalam melakukan pengembangan aplikasi berbasis *web* di Politeknik Negeri Batam tersebut, metode yang digunakan berbeda-beda. Misalnya pada *e-learning*, pembangunannya menggunakan *Content Management System Moodle*, *simpol* masih memakai bahasa pemrograman *hyper text markup language (HTML)* biasa dan PHP, dan aplikasi-aplikasi berbasis *web* lainnya menggunakan *template web* yang sudah tersedia atau dengan *framework* juga. Karena belum seragamnya cara yang digunakan dalam membangun *web*, maka proses pengembangan aplikasi *web* tersebut dilakukan satu-persatu.

Oleh karena itu, diperlukan kerangka dasar untuk membangun semua aplikasi *web* yang ada di kampus Politeknik Negeri Batam, agar pembangunan *web* lebih berseragam untuk memudahkan perawatan dan pengembangan aplikasi-aplikasi *web* dan juga untuk mempermudah dan mempercepat dalam pembangunan *web* yang baru. Kerangka dasar tersebut bisa dengan membuat PHP *framework* untuk membangun *web-web* yang ada di kampus poteknik negeri batam. Kampus Politeknik Negeri Batam belum mempunyai *framework* sendiri untuk membangun aplikasi *web*. Dengan mempunyai *framework* sendiri maka akan mempermudah dalam pengembangan kedepan dari sisi *frameworknya* maupun pengembangan dari sisi aplikasi *webnya*.

I.2 Rumusan Masalah

Bagaimana cara membangun *framework* yang nantinya dapat digunakan untuk aplikasi berbasis *web* di Politeknik Negeri Batam?

I.3 Batasan Masalah

Berikut ini adalah batasan masalah pada Tugas Akhir:

- 1) Membuat dasar dan perancangan *framework* Politeknik Negeri Batam dengan mengacu pada YII *framework*.
- 2) Tidak melakukan pembangunan *web* dengan menggunakan *framework* ini karena *framework* yang dibangun belum selesai dan masih perlu pengembangan lebih lanjut.
- 3) Membuat sebagian komponen dasar dari *framework* berdasarkan acuannya, yaitu sebagai berikut:
 - a. Paket system terdiri dari kelas Pnb dan kelas PnbBase
 - b. Paket system.base yang terdiri dari kelas CApplication, CApplicationComponent, CBehavior, CComponent, CEnumerable, CErrorEvent, CErrorHandler, CEvent, CException, CExceptionEvent, CHttpException, CModel, CModelBehavior, CModelEvent, CModule,

CSecurityManager, CStatePersister, IAction, IApplicationComponent, IAuthManager, IBehavior, IFilter, IStatePersister, IUserIdentity, IViewRenderer, IWebServiceProvider.

I.4 Tujuan

Tujuan dari penelitian ini adalah membuat rancangan awal *framework* dan menyelesaikan satu bagian dari *framework* tersebut, yang nantinya dapat digunakan dalam pengembangan aplikasi berbasis *web* di Politeknik Negeri Batam.

I.5 Sistematika Penulisan

- a) Bab I Pendahuluan

Berisi mengenai latar belakang, rumusan masalah, batasan masalah dan tujuan serta sistematika penulisan.
- b) Bab II Tinjauan Pustaka

Berisi mengenai landasan teori yang berhubungan dengan penelitian tugas akhir yaitu mengenai *framework* , Yii *Framework*, Codeigniter, PHP dan MySQL.
- c) Bab III Analisis dan Perancangan

Berisi mengenai analisis Pnb *framework* serta perancangan diagram kelas Pnb *framework*.
- d) Bab IV Implementasi

Berisi tentang implementasi dan pengujian kelas.
- e) Bab V Kesimpulan dan Saran

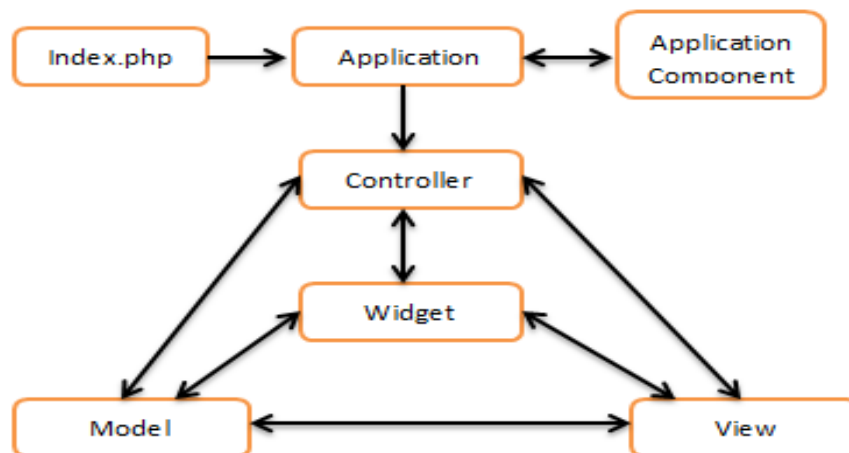
Berisi tentang kesimpulan dan saran.

Bab II Landasan Teori

II.1 Framework

Framework adalah sekumpulan *library* yang diorganisasikan pada sebuah rancangan arsitektur untuk memberikan kecepatan, ketepatan, kemudahan dan konsistensi di dalam pengembangan aplikasi. Dari definisi tersebut, *framework* mengandung unsur-unsur seperti *library*, arsitektur dan metodologi. *Framework* menyediakan *library* yang lengkap dan terintegrasi. Unsur inilah yang paling banyak memberikan keuntungan bagi pengembang aplikasi. Sebagai contoh, *framework* aplikasi *web* menyediakan *library* dasar berkaitan layanan *web*, seperti *authentication*, *database access*, *HTML generation*, *template engine*, *session management*, *caching*, *searching*, *email*, *http*, *pdf*, *zip* dan *ajax*. *Framework* juga menyediakan sarana *extension* untuk *library* yang belum tersedia atau *library* yang perlu ditingkatkan kemampuannya.

Sebuah aplikasi membutuhkan rancangan bagaimana sistem dibangun, yaitu bagaimana elemen - elemen di dalam aplikasi disusun dan diintegrasikan, seperti pengorganisasian *file* dan *library*, dan bagaimana *library* digunakan.



Gambar 1 Konsep MVC

Framework menyediakan struktur aplikasi dan bagaimana interaksi di dalam struktur tersebut. MVC (Model View Controller) adalah salah satu pola rancangan (*design pattern*) yang banyak digunakan dalam arsitektur aplikasi saat ini seperti pada gambar 1 diatas.

PHP *framework* menggunakan *method* pengembangan berbasis MVC. MVC merupakan suatu *method* untuk memisahkan bagian-bagian dari suatu *web* aplikasi. MVC adalah kependekan dari Model View Controller. MVC terdiri dari tiga bagian yaitu:

1. Model

Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu kita dalam pengelolaan *database* seperti memasukkan data ke *database*, *update* data dan lain-lain.

2. View

View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman *web*.

3. Controller

Controller merupakan bagian yang menjembatani model dan *view*. *Controller* berisi skrip *PHP* yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman *web*.

Disinilah letak perbedaan utama antara *framework* dengan CMS atau PHP konvensional. Dengan *method* MVC, bagian tampilan, logika serta *query database* diletakkan secara terpisah namun tetap sinkron sehingga pembuatan aplikasi menjadi lebih terstruktur dan sederhana. Model digunakan dalam menuliskan skrip *database*, *Controller* untuk mengembangkan logika pemrogramannya, sedangkan *View* berfungsi dalam menampilkan *layout* dari aplikasi yang kita buat.

Analisis sederhana tentang *framework* yang sudah ada dan yang paling sering digunakan oleh *programmer* yang juga dijadikan sebagai acuan untuk memudahkan *programmer* tersebut dalam proses perancangan, maka digunakan

cara yaitu dengan cara membandingkan antara *framework* terbaik untuk saat ini menurut *website* <http://www.PHPFrameworks.com/top-10-PHP-Frameworks/>. Pada *website* tersebut diberitahukan bahwa *framework* yang menduduki peringkat pertama adalah Yii (Yes It Is) *framework* dan yang menduduki peringkat kedua CI (Codeigniter) *framework*. Parameter perbandingannya hanya mencakup hal yang sangat signifikan dan umum yaitu:

1. Dokumentasi

Dari segi dokumentasi CI lebih lengkap, jelas dan mudah dipahami dari pada Yii.

2. Kemampuan

Perbandingan kemampuan antara CI dan Yii bisa dilihat dari tabel berikut:

PHP Framework	CI	Yii
PHP4	✓	-
PHP5	✓	✓
MVC	✓	✓
<i>Multiple DB's</i>	✓	✓
ORM	-	✓
<i>DB Objects</i>	✓	✓
<i>Templates</i>	✓	✓
<i>Caching</i>	✓	✓
<i>Validation</i>	✓	✓
Ajax	-	✓
<i>Auth Module</i>	-	✓
<i>Modules</i>	-	✓
EDP	-	✓

Tabel 1 Perbandingan antara *Yii framework* dengan *CI framework*

3. *Performance*

Dalam hal ini CI memilih untuk tidak menggunakan seluruh kemampuan *Object Oriented Programming* (OOP) dengan alasan *performance*, namun Yii mematahkan anggapan bahwa OOP pasti akan mengorbankan *performance*. Tetapi bila dibandingkan dengan *structural programming* besar kemungkinan OOP tertinggal, namun dengan kemudahan pengembangan aplikasi menjadi nilai lebih bagi pengembangan dengan OOP.

Dengan ini dapat disimpulkan bahwa Yii merupakan *framework* yang terbaik saat yang menggunakan bahasa pemrograman PHP dan memiliki konsep berorientasi objek serta menggunakan pola model *view controller*, sehingga *framework* ini menjadi acuan dalam pembuatan Tugas Akhir ini.

II.2 Yii Frameworks

Yii adalah *framework* (kerangka kerja) PHP berbasis-komponen, berkinerja tinggi untuk pengembangan aplikasi *Web* berskala-besar. Yii menyediakan *reusability* maksimum dalam pemrograman *Web* dan mampu meningkatkan kecepatan pengembangan aplikasi secara signifikan. Nama Yii (dieja sebagai /i:/) singkatan dari "Yes It Is!". Yii pertama kali dikembangkan oleh seorang master bernama Qiang Xue pada Januari 2008. Pada bulan Oktober 2008 setelah sepuluh bulan pembangunan, versi alpha Yii pertama dirilis. Pada tanggal 3 Desember 2008, Yii 1.0 secara resmi dirilis ke publik.

Untuk menjalankan aplikasi *Web* berbasis *Yii Framework*, pengguna memerlukan *server Web* yang mendukung PHP 5.1.0. Bagi para pengembang yang ingin menggunakan *Yii*, memahami pemrograman berorientasi-objek (OOP) akan sangat membantu, karena *Yii* merupakan *framework* OOP murni. *Yii* adalah *framework* pemrograman umum *Web* yang bisa dipakai untuk mengembangkan semua jenis aplikasi *Web*. Dikarenakan sangat ringan dan dilengkapi dengan mekanisme *caching* yang canggih, *Yii* sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, forum, sistem manajemen konten (*CMS, Content Management System*), sistem *e-commerce*, dan lain-lain.

Seperti kebanyakan PHP *framework*, *Yii* adalah Model View Controller (MVC) *framework*. *Yii* melampaui *framework* PHP lain dalam hal efisiensi, kekayaanfitur, dan kejelasan dokumentasi. *Yii* didesain dengan hati-hati dari awal agar sesuai untuk pengembangan aplikasi *Web* secara serius. *Yii* bukan berasal dari produk pada beberapa proyek maupun konglomerasi pekerjaan pihak-ketiga. *Yii* adalah

hasil dari pengalaman kaya para pembuat pada pengembangan aplikasi *Web* dan investigasi *framework* pemrograman *Web* dan aplikasi yang paling populer.

Performa kinerja *Yii Framework* jauh lebih cepat karena menggunakan teknik *lazy loading ekstensif*. Misalnya, tidak menyertakan *file* kelas sampai kelas digunakan untuk pertama kalinya, dan tidak menciptakan objek sampai objek tersebut diakses untuk pertama kalinya. *Framework* lainnya memiliki kinerja berat karena memungkinkan fungsi (misalnya koneksi database, sesi pengguna) terus digunakan walaupun tidak selama permintaan.

Berikut fitur-fitur utama yang terdapat pada *Yii Framework* :

1. Design pola *Model-View-Controller* (MVC). *Yii* mengadopsi arsitektur MVC, yang memungkinkan untuk pemisahan perhatian.
2. *Database Access Objects* (DAO), *Query Builder*, *Active Record*, *DB Migration*. *Yii* memungkinkan pengembang untuk melakukan model data database dalam hal objek dan menghindari kebosanan dan kompleksitas menulis pernyataan SQL berulang.
3. *Form input* dan validasi. *Yii* mengumpulkan *form input* sangat mudah dan aman. Muncul dengan satu *set validator* serta *helper methods* dan *widget* untuk menyederhanakan tugas untuk *input* bentuk dan validasi.
4. *AJAX-enabled widget*. Terintegrasi dengan *jQuery*, *Yii* datang dengan satu set *AJAX-enabled widget*, seperti *input field*, *treeview*, *data grid*, yang memungkinkan menulis antarmuka pengguna yang sangat efisien, serbaguna dan sangat sederhana.
5. Otentikasi dan otorisasi. *Yii* memiliki *built-in* dukungan otentikasi. Ini mendukung otorisasi melalui hirarkis *role-based* akses kontrol.
6. Layanan *Web*. *Yii* mendukung generasi otomatis layanan spesifikasi kompleks WSDL dan manajemen penanganan permintaan layanan *Web*.
7. Internasionalisasi (I18N) dan lokalisasi (L10N). *Yii* mendukung terjemahan pesan, tanggal dan waktu *format*, *format* angka, dan lokalisasi antarmuka.

8. *Layered caching* skema. *Yii* mendukung *caching* data, *cache* halaman, *cache* fragmen dan konten dinamis. Media penyimpanan *caching* dapat diubah dengan mudah tanpa menyentuh kode aplikasi.
9. Penanganan dan penebangan kesalahan. Kesalahan akan ditangani dan disajikan lebih baik, dan pesan log dapat dikategorikan, disaring dan disalurkan ke berbagai tujuan.
10. Keamanan. *Yii* dilengkapi dengan langkah-langkah keamanan untuk membantu mencegah aplikasi *Web* Anda dari serangan seperti *SQL injection*, *cross-site scripting* (XSS), *cross-site* pemalsuan permintaan (CSRF), dan *cookie* gangguan.
11. *Unit* dan pengujian fungsionalitas. *Yii* menyediakan dukungan untuk menulis dan menjalankan *unit* test serta tes fungsi, berdasarkan PHP *Unit* dan Selenium.
12. Otomatis menggenerasi kode. *Yii* menyediakan satu set alat kode intuitif dan *extensible code generation* yang dapat membantu pengguna lebih cepat menghasilkan kode yang dibutuhkan untuk fitur seperti *form input*, CRUD.
13. Kepatuhan ke XHTML. Kode HTML yang dihasilkan oleh komponen *Yii* dan generator kode sesuai dengan standar XHTML.
14. Murni berorientasi objek. *Yii framework* menggunakan paradigma OOP. *Yii* tidak mendefinisikan fungsi global atau variabel. Dan hirarki kelas yang didefinisikan memungkinkan usabilitas maksimum dan kustomisasi.
15. Ramah dengan kode pihak ketiga. *Yii* dirancang untuk bekerja dengan baik dengan kode pihak ketiga. Sebagai contoh, pengguna dapat menggunakan kode dari PEAR atau *Zend Framework* dalam aplikasi *Yii*.
16. Modular dan mudah ditambahkan dengan dukungan ekstensi dan komponen tambahan sehingga beragam fitur tambahan dengan mudah dimasukkan.

Banyak orang awam beranggapan bahwa penggunaan *Yii Framework* masih sulit digunakan karena dokumentasinya belum lengkap dan masih terbatas. Dari situlah banyak orang awam pesimis bagaimana cara membuat *website* *yii* yang

berkualitas. Meskipun dokumentasi yang dimiliki yii belum lengkap dan masih terbatas, namun sudah banyak *website* berkualitas yang dihasilkan oleh yii *framework* ini. Berikut adalah *website-website* yang berkualitas dalam menggunakan atau mengimplementasi yii *frameworks* :

1. *Website* Dikti

Website milik pemerintah di bidang beasiswa pendidikan untuk jenjang sekolah menengah menuju bangku kuliah ini sudah berbasis yii *Framework*. Untuk laman resminya bisa anda kunjungi di alamat: <http://daftar.bidikmisi.dikti.go.id>

2. *Website* EduSec

Website ERP ini sangatlah mudah di gunakan menurut *review* dari pembuatnya. Untuk demonya bisa anda lihat di laman: <http://www.edusec.rudrasoftech.com> (username: admin@rudrasoftech.com | password: admin).

3. *Website* Rajawali Parama Konstruksi

Website ini menawarkan alat-alat di bidang konstruksi. Bisa anda kunjungi di laman: <http://www.rajawaliparama.com>.

4. *Website* UrbanIndo

Website ini bergerak di bidang jual beli rumah. Untuk lamannya bisa anda kunjungi di alamat: <http://www.urbanindo.com>

5. *Website* Ngomik

Website tentang komik ini bisa anda kunjungi di laman <http://www.ngomik.com>. Anda bisa secara gratis membaca berbagai komik yang di tawarkan *website* ini

6. *Website* montirGw (DEMO)

Website yang cukup unik dan di gemari oleh para master yii *Framework* Indonesia ini bergerak di bidang jasa yaitu menawarkan montir panggilan di daerah Jakarta dan sekitarnya. Untuk lamannya bisa anda kunjungi di <http://www.montirgw.com>

7. *Website* pusat BeNiceKids Bandung (DEMO)

Website ini merupakan *website* jual beli (e-commerce) barang-barang anak. Bisa anda jadikan referensi untuk pembuatan *website* jual beli / e-commerce anda. Bisa anda kunjungi di laman : <http://pusatbenicekidsbandung.com>

8. *Website* ebizu

Website ini menawarkan jual beli makanan tertentu berbasis android tidak hanya sekedar makanan tetapi ebizu mencoba bekerjasama dengan banyak perusahaan, dan juga memberikan penawaran, diskon, dari tiap-tiap perusahaan ke membeinya atau ke orang lain. Untuk lamannya dapat anda kunjungi di laman: <https://beta.ebizu.com>

9. *Website* Fakultas Kedokteran Trisakti

Website milik salah satu Universitas swasta di Jakarta ini sudah menggunakan Yii *Framework*. Dapat anda lihat di laman: <http://cpdfkusakti.com> | <http://fkusaktitracerstudy.org>

10. *Website* Genmoodbreakers

Website yang bergerak di bidang jual beli bola ini menawarkan barang-barang dengan kualitas baik yang di kemas dalam *website* berbasi Yii *Framework*. Bisa anda kunjungi di laman <http://www.genmoodbreakers.com>

11. Bursa Tenaga Kerja

Website yang bergerak di bidang bursa tenaga kerja. Sebagai usaha nyata Institusi untuk mewujudkansemboyannya, yaitu “Lulusan Mudah Bekerja”, LPKIA sejak tahun 1998telah mengaktifkan Bursa Tenaga Kerja (BTK), yang berfungsi untuk“menjembatani” kebutuhan perusahaan akan tenaga kerja terampil siap pakai dankebutuhan alumni untuk memperoleh pekerjaan yang tepat.

12. Agung Podomoro

PT. Agung Podomoro Group adalah perusahaan pengembang terbesar di sektor properti di Indonesia. PT Agung Podomoro membangun hunian di daerah Sunter dan Kelapa Gading. Selain itu, perusahaan ini juga ikut menyediakan perumahan bagi masyarakat miskin di luar Jabodetabek, seperti Subang, Lebak dan Kresek.

13. ERP Indonesia

Website ini dibangun oleh Admin grup Yii PHP *Framework* Indonesia yang bergerak di bidang ERP (*Enterprise Resource Planning*). Dapat anda kunjungi demonya di laman <http://www.erp-indonesia.tk> dengan Username: peter dan Password: kambey.

Yii *framework* memiliki keuntungan dan juga kerugian. Keuntungan yang dimiliki oleh yii *framework* adalah sebagai berikut :

1. Yii adalah salah satu *framework* yang sangat ringan dan dilengkapi dengan solusi *caching* yang memuaskan.
2. Yii sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, forum, sistem manajemen konten (CMS), sistem e-commerce, dll.
3. Yii didokumentasikan dengan jelas, efisien, dan kaya-fitur.

Tidak banyak kekurangan yang ditemui pada penggunaan Yii *framework*. Yii *framework* mencakup segala macam fitur-fitur yang diperlukan dalam membangun sebuah *web*. Namun, bukan berarti Yii tidak mempunyai kekurangan. Salah satu kekurangan Yii yang cukup signifikan adalah tidak compatible nya Yii dengan PHP 4.

II.3 Codeigniter

Codeigniter merupakan *framework* PHP yang mempunyai eksekusi tercepat dibandingkan dengan *framework* lain. Codeigniter bersifat Open Source Dan memakai model basis MVC (Model View Controller), yang merupakan model konsep modern *framework* yang digunakan saat ini.

Codeigniter menawarkan banyak sekali library yang bisa kita pakai. Codeigniter dibuat pertama kali oleh Rick Ellis yang merupakan CEO dari Ellislab. Situsnya bias dikunjungi di ellislab.com . Perlu kita ketahui, Ellislab merupakan sebuah perusahaan yang memproduksi CMS-CMS yang handal. Dan saat ini Codeigniter dikembangkan oleh Exspression Engine Development Team. *Framework* memiliki kelebihan dan kekurangan masing-masing, begitu pula dengan *framework* Codeigniter.

Kelebihan *framework* Codeigniter adalah sebagai berikut :

1. Gratis

Codeigniter dilisensikan dibawah lisensi Apache/BSD style open source. Jadi ini berarti Codeigniter dapat dipakai secara bebas.

2. Support PHP4 dan PHP5

Walau saat ini PHP telah mencapai versi 5 bahkan versi 6, namun masih banyak juga orang – orang yang memakai PHP4. Oleh sebab itu, pengembang *Framework* Codeigniter memperhatikan betul setiap pengguna, oleh karenanya *Framework* Codeigniter dikembangkan agar mampu berjalan baik PHP4 Maupun PHP5.

3. Ukuran *file* kecil dan cepat

Dibandingkan dengan *Framework* lain, semisal CakePHP, Codeigniter adalah alternative bagi kamu yang menginginkan akses yang cepat. Kenapa bisa cepat? Karena Codeigniter hanya me-load fungsi atau library yang digunakan saja, berbeda dengan *Framework* lainnya yang menggunakan seluruh library walaupun library tersebut tidak digunakan. Alasan inilah yang menjadikan Codeigniter dengan akses tercepat dan ringan.

4. Dokumentasi

Framework yang baik pastinya dilengkapi dengan dokumentasi yang lengkap dan mendukung, agar bisa mudah dipahami oleh penggunanya. Didukung oleh User Guide yang mudah dimengerti, dari mulai install sampai pada fungsi-fungsinya semua ada.

5. Memakai konsep MVC

Codeigniter memakai konsep MVC (*Model View Controller*), konsep modern yang banyak dipakai oleh *framework* PHP lainnya. Dengan adanya MVC, pengerjaan antara logika dengan *layout* telah dipisahkan, sehingga antara si programmer dan designer dapat santai melakukan tugasnya.

6. Komunitas

Framework Codeigniter memiliki sebuah komunitas yang cukup besar dan bisa bergabung pada situs codeigniter.com/forums

Disamping kelebihan atau keuntungan, *framework* codeigniter ini juga memiliki beberapa kelemahan atau kekurangan diantaranya adalah :

1. Codeigniter dikembangkan oleh Ellislab dan bukan oleh suatu komunitas, yang menyebabkan *update* core engine-nya tidak secepat *framework* lain.
2. Tidak ditujukan untuk pembuatan *web* dengan skala besar (enterprise) walaupun tersedia banyak library.
3. Masih banyak kelonggaran dalam hal *coding*, misalnya bebas dalam menambah *file*.
4. Tidak mencerminkan MVC yang sesungguhnya, misalnya penulisan echo masih dapat dilakukan pada *file controller*.

II.4 PHP

PHP (akronim dari PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman yang berfungsi untuk membuat *website* dinamis maupun aplikasi *web*. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file* dan *folder*, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website*. *Blog*, *Toko Online*, *CMS*, *Forum*, dan *Website Social Networking* adalah contoh aplikasi *web* yang bisa dibuat oleh PHP. PHP adalah bahasa *scripting*, bukan bahasa *tag-based* seperti HTML. PHP termasuk bahasa yang *cross-platform*, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). *Program* PHP ditulis dalam *file plain text* (teks biasa) dan mempunyai akhiran “.PHP”. PHP ditulis (diciptakan) oleh Rasmus Lerdorf, seorang *software engineer* asal Greenland sekitar tahun 1995. Pada awalnya PHP digunakan Rasmus hanya sebagai pencatat jumlah pengunjung pada *website* pribadi beliau. Karena itu bahasa tersebut dinamakan *Personal Home Page (PHP) Tools*. Tetapi karena perkembangannya yang cukup disukai oleh komunitasnya, maka beliau pun merilis bahasa PHP tersebut ke publik dengan lisensi *open-source*. Saat ini, PHP adalah *server-side scripting* yang paling banyak digunakan di *website-website* di seluruh dunia, dengan *versi* sudah mencapai *versi 5* dan statistiknya terus bertambah.

II.5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. 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.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language).

SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

Keistimewaan yang dimiliki oleh MySQL adalah sebagai berikut :

1. Portabilitas

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi

2. Open Source

MySQL didistribusikan secara open source, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. Multiuser

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik

4. Performance tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu

5. Jenis Kolom

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan Fungsi

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).

7. Keamanan

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi

8. Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT)

10. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar Muka

MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. Klien dan Peralatan

MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online

13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Bab III Analisis dan Perancangan

III.1 Analisis Pnb Framework

III.1.1 Analisa Batasan Sistem

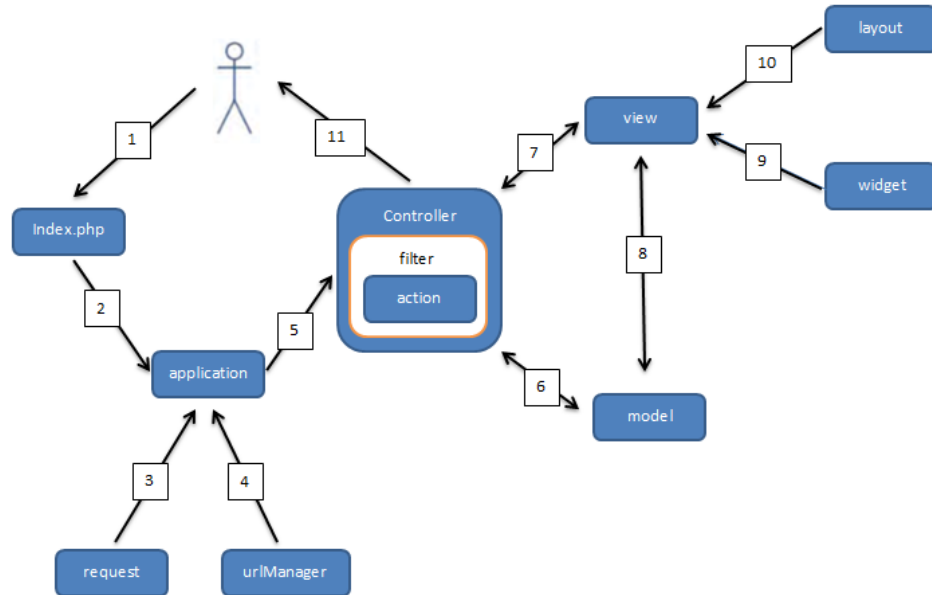
Batasan sistem yang dimiliki oleh pnb *framework* ini dapat dilihat pada tabel dibawah ini :

Komponen	Spesifikasi	Pnb Framework	
		Ya	Tidak
RAM	0 - 128 MB		✓
	256 MB	✓	
	Lebih dari 512 MB	✓	
Sistem Operasi	Windows	✓	
	Macintosh	✓	
	Linux	✓	
Core Proccesor	128 MHz – 512 MHz		✓
	1 GHz – 2,2 GHz	✓	
	Lebih dari 2,2 GHz	✓	
Web Server	Apache HTTP Web Server	✓	
	WampServer	✓	
Database	MySQL	✓	
	Oracle	✓	
	PostgreSQL	✓	
	SQLite	✓	
Harddisk	40 GB	✓	
	80-320 GB	✓	
	Lebih dari 500 GB	✓	

Tabel 2 Batasan sistem pada Pnb Framework

III.1.2 Analisa Cara Kerja Pnb Framework

Diagram berikut memperlihatkan alur kerja umum aplikasi yang menggunakan pnb *framework* saat menangani permintaan dari pengguna.



Gambar 2 Alur Kerja umum aplikasi pnb

Deskripsi dari gambar di atas, alur kerja umum aplikasi pnb akan dipaparkan sebagai berikut :

1. Pengguna membuat permintaan dengan menggunakan URL `http://www.example.com/index.php?` dan *server Web* akan menangani permintaan dengan menjalankan skrip *bootstrap index.php*.
2. Skrip *bootstrap* di `index.php` akan membuat sebuah instan Aplikasi dan langsung menjalankannya.
3. Aplikasi akan mendapatkan rincian informasi permintaan pengguna dari komponen yang terdapat pada aplikasi yang bernama *request*.
4. Aplikasi akan menentukan *controller* dan aksi yang diminta dengan bantuan komponen lain yang dimiliki aplikasi yang bernama *urlManager*. Dalam contoh ini, *controller* adalah `post` yang merujuk pada kelas *PostController*; dan aksi adalah *show* yang arti sebenarnya ditentukan oleh *controller*.
5. Aplikasi membuat *instance controller* yang diminta untuk selanjutnya menangani permintaan pengguna. *Controller* menentukan aksi *show* merujuk

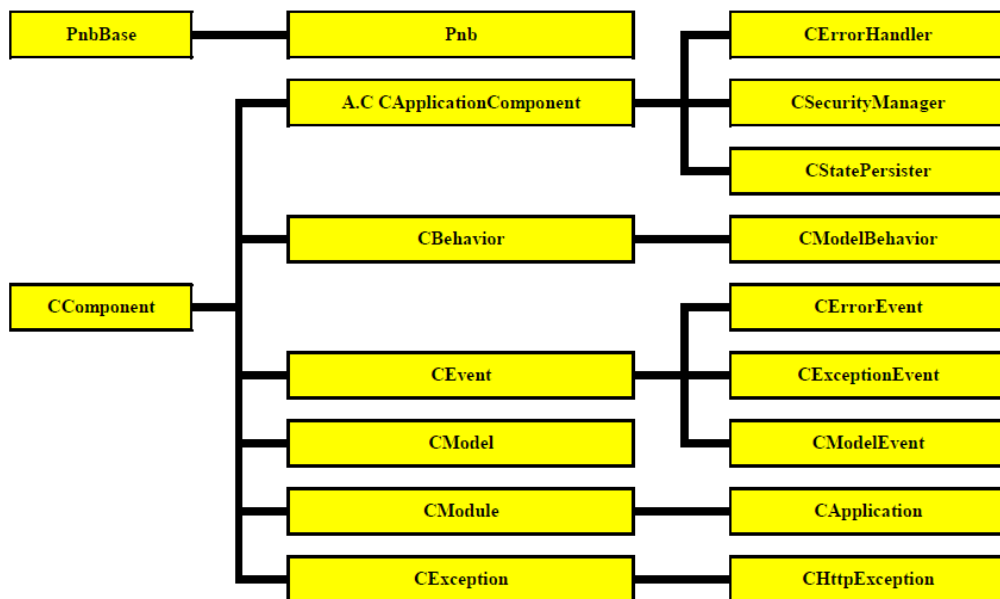
pada sebuah metode bernama *actionShow* dalam kelas *controller*. Kemudian membuat dan menjalankan filter (contoh kontrol akses, pengukuran) terkait dengan aksi ini. Aksi dijalankan jika diijinkan oleh filter.

6. Aksi membaca Post model di mana ID adalah 1 dari database.
7. Aksi menyiapkan *view* (tampilan) bernama show dengan model Post.
8. *View* membaca dan menampilkan atribut model Post.
9. *View* menjalankan beberapa widget.
10. *View* menyiapkan hasil yang dipasang dalam *layout* (tata letak).
11. Aksi mengakhiri pembuatan *view* dan menampilkan hasil akhir kepada pengguna.

III.2 Diagram Kelas Pnb Framework

III.2.1 Diagram Kelas

Gambar di bawah ini merupakan rancangan pemetaan kelas pada *framework* yang akan dibangun. Untuk pemetaan kelas secara keseluruhan dapat dilihat di bagian Lampiran II.



Gambar 3 Rancangan Pemetaan Kelas Dasar Framework

III.2.2 Diagram Kelas Rinci

Tabel berikut merupakan diagram kelas rinci dalam pembangunan *framework* ini.

Untuk detail perancangan dapat dilihat pada Lampiran I.

NO	Nama kelas	Atribut		Metod	
		Nama	Tipe	Nama	Jenis
1	CApplication	basePath	string	__call()	Publik
		baseUrl	string	__construct()	Publik
		behaviors	array	__get()	Publik
		cache	CCache	__isset()	Publik
		charset	string	__set()	Publik
		components	array	__unset()	Publik
		<i>controller</i>	CController	asa()	Publik
		coreMessages	CPhpMessage Source	attachBehavior()	Publik
		dateFormatter	CDateFormatter	attachBehaviors()	Publik
		db	CDbConnection	attachEventHandler()	Publik
		errorHandler	CErrorHandler	canGetProperty()	Publik
		extensionPath	string	canSetProperty()	Publik
		homeUrl	string	clearGlobalState()	Publik
		id	string	configure()	Publik
		language	string	createAbsoluteUrl()	Publik
		locale	CLocale	createUrl()	Publik
		localeDataPath	string	detachBehavior()	Publik
		messages	CMessageSource	detachBehaviors()	Publik
		modulePath	string	detachEventHandler()	Publik
		modules	array	disableBehavior()	Publik
		name	string	disableBehaviors()	Publik
		numberFormatter	CNumberFormatter	displayError()	Publik
		params	CAttributeCollection	displayException()	Publik
		parentModule	CModule	enableBehavior()	Publik
preload	array	enableBehaviors()	Publik		
request	CHttpRequest	end()	Publik		

	runtimePath	string	evaluateExpression()	Publik
	securityManager	CSecurityManager	findLocalizedFile()	Publik
	sourceLanguage	string	getBasePath()	Publik
	statePersister	CStatePersister	getBaseUrl()	Publik
	timeZone	string	getCache()	Publik
	urlManager	CUrlManager	getComponent()	Publik
			getComponents()	Publik
			getController()	Publik
			getCoreMessages()	Publik
			getDateFormatter()	Publik
			getDb()	Publik
			getErrorHandler()	Publik
			getEventHandlers()	Publik
			getExtensionPath()	Publik
			getGlobalState()	Publik
			getHomeUrl()	Publik
			getId()	Publik
			getLanguage()	Publik
			getLocale()	Publik
			getLocaleDataPath()	Publik
			getMessages()	Publik
			getModule()	Publik
			getModulePath()	Publik
			getModules()	Publik
			getNumberFormatter()	Publik
			getParams()	Publik
			getParentModule()	Publik
			getRequest()	Publik
			getRuntimePath()	Publik
			getSecurityManager()	Publik
			getStatePersister()	Publik
			getTimeZone()	Publik
			getUrlManager()	Publik
			handleError()	Publik
			handleException()	Publik
			hasComponent()	Publik

			hasEvent()	Publik
			hasEventHandler()	Publik
			hasModule()	Publik
			hasProperty()	Publik
			loadGlobalState()	Publik
			onBeginRequest()	Publik
			onEndRequest()	Publik
			onError()	Publik
			onException()	Publik
			processRequest()	Publik
			raiseEvent()	Publik
			run()	Publik
			saveGlobalState()	Publik
			setAliases()	Publik
			setBasePath()	Publik
			setComponent()	Publik
			setComponents()	Publik
			setExtensionPath()	Publik
			setGlobalState()	Publik
			setHomeUrl()	Publik
			setId()	Publik
			setImport()	Publik
			setLanguage()	Publik
			setLocaleDataPath()	Publik
			setModulePath()	Publik
			setModules()	Publik
			setParams()	Publik
			setRuntimePath()	Publik
			setTimeZone()	Publik
			init()	Privat
			initSystemHandlers()	Privat
			preinit()	Privat
			preloadComponents()	Privat
			registerCoreCompon ents()	Privat
			onBeginRequest	Event
			onEndRequest	Event
			onException	Event

				onError	Event
2	CApplication Component	behaviors	array	__call()	publik
		isInitialized	Boolean	__get()	publik
				__isset()	publik
				__set()	publik
				__unset()	publik
				asa()	publik
				attachBehavior()	publik
				attachBehaviors()	publik
				attachEventHandler()	publik
				canGetProperty()	publik
				canSetProperty()	publik
				detachBehavior()	publik
				detachBehaviors()	publik
				detachEventHandler()	publik
				disableBehavior()	publik
				disableBehaviors()	publik
				enableBehavior()	publik
				enableBehaviors()	publik
				evaluateExpression()	publik
				getEventHandlers()	publik
		getIsInitialized()	publik		
		hasEvent()	publik		
		hasEventHandler()	publik		
		hasProperty()	publik		
		init()	publik		
		raiseEvent()	publik		
3	CBehavior	enabled	boolean	__call()	publik
		owner	CComponent	__get()	publik
				__isset()	publik
				__set()	publik
				__unset()	publik
				asa()	publik
				attach()	publik
				attachBehavior()	publik
				attachBehaviors()	publik
		attachEventHandler()	publik		

			canGetProperty()	publik
			canSetProperty()	publik
			detach()	publik
			detachBehavior()	publik
			detachBehaviors()	publik
			detachEventHandler()	publik
			disableBehavior()	publik
			disableBehaviors()	publik
			enableBehavior()	publik
			enableBehaviors()	publik
			evaluateExpression()	publik
			events()	publik
			getEnabled()	publik
			getEventHandlers()	publik
			getOwner()	publik
			hasEvent()	publik
			hasEventHandler()	publik
			hasProperty()	publik
			raiseEvent()	publik
			setEnabled()	publik
4	CComponent		__call()	publik
			__get()	publik
			__isset()	publik
			__set()	publik
			__unset()	publik
			asa()	publik
			attachBehavior()	publik
			attachBehaviors()	publik
			attachEventHandler()	publik
			canGetProperty()	publik
			canSetProperty()	publik
			detachBehavior()	publik
			detachBehaviors()	publik
			detachEventHandler()	publik
			disableBehavior()	publik
			disableBehaviors()	publik

				enableBehavior()	publik
				enableBehaviors()	publik
				evaluateExpression()	publik
				getEventHandlers()	publik
				hasEvent()	publik
				hasEventHandler()	publik
				hasProperty()	publik
				raiseEvent()	publik
5	CEnumerable				
6	CErrorEvent	code	string	__call()	publik
		<i>file</i>	string	__construct()	publik
		handled	boolean	__get()	publik
		line	string	__isset()	publik
		message	string	__set()	publik
		params	mixed	__unset()	publik
		sender	object	asa()	publik
				attachBehavior()	publik
				attachBehaviors()	publik
				attachEventHandler() ()	publik
				canGetProperty()	publik
				canSetProperty()	publik
				detachBehavior()	publik
				detachBehaviors()	publik
				detachEventHandler() ()	publik
				disableBehavior()	publik
				disableBehaviors()	publik
				enableBehavior()	publik
				enableBehaviors()	publik
				evaluateExpression()	publik
		getEventHandlers()	publik		
		hasEvent()	publik		
		hasEventHandler()	publik		
		hasProperty()	publik		
		raiseEvent()	publik		
7	CErrorHandler	adminInfo	string	__call()	publik
		behaviors	array	__get()	publik
		discardOutput	boolean	__isset()	publik

	error	array	__set()	publik
	errorAction	string	__unset()	publik
	isInitialized	boolean	asa()	publik
	maxSourceLines	integer	attachBehavior()	publik
	maxTraceSourceLines	integer	attachBehaviors()	publik
	versionInfo	string	attachEventHandler())	publik
			canGetProperty()	publik
			canSetProperty()	publik
			detachBehavior()	publik
			detachBehaviors()	publik
			detachEventHandler())	publik
			disableBehavior()	publik
			disableBehaviors()	publik
			enableBehavior()	publik
			enableBehaviors()	publik
			evaluateExpression()	publik
			getError()	publik
			getEventHandlers()	publik
			getIsInitialized()	publik
			handle()	publik
			hasEvent()	publik
			hasEventHandler()	publik
			hasProperty()	publik
			init()	publik
			raiseEvent()	publik
			argumentsToString()	Privat
			getExactTrace()	Privat
			getHTTPHeader()	Privat
			getVersionInfo()	Privat
			getViewFile()	Privat
			getViewFileInternal())	Privat
			handleError()	Privat
			handleException()	Privat
			isAjaxRequest()	Privat
			isCoreCode()	Privat

				render()	Privat
				renderSourceCode()	Privat
8	CEvent	handled	boolean	__call()	Publik
		params	mixed	__construct()	Publik
		sender	object	__get()	Publik
				__isset()	Publik
				__set()	Publik
				__unset()	Publik
				asa()	Publik
				attachBehavior()	Publik
				attachBehaviors()	Publik
				attachEventHandler() ()	Publik
				canGetProperty()	Publik
				canSetProperty()	Publik
				detachBehavior()	Publik
				detachBehaviors()	Publik
				detachEventHandler() ()	Publik
				disableBehavior()	Publik
				disableBehaviors()	Publik
				enableBehavior()	Publik
				enableBehaviors()	Publik
				evaluateExpression()	Publik
		getEventHandlers()	Publik		
		hasEvent()	Publik		
		hasEventHandler()	Publik		
		hasProperty()	Publik		
		raiseEvent()	Publik		
9	CException	code		__construct()	Publik
		<i>file</i>		__toString()	Publik
		line		getCode()	Publik
		message		getFile()	Publik
				getLine()	Publik
				getMessage()	Publik
				getPrevious()	Publik
				getTrace()	Publik
		getTraceAsString()	Publik		
10	CExceptionE	exception	CException	__call()	Publik

	vent	handled	boolean	__construct()	Publik
		params	mixed	__get()	Publik
		sender	object	__isset()	Publik
				__set()	Publik
				__unset()	Publik
				asa()	Publik
				attachBehavior()	Publik
				attachBehaviors()	Publik
				attachEventHandler()	Publik
				canGetProperty()	Publik
				canSetProperty()	Publik
				detachBehavior()	Publik
				detachBehaviors()	Publik
				detachEventHandler()	Publik
				disableBehavior()	Publik
				disableBehaviors()	Publik
				enableBehavior()	Publik
				enableBehaviors()	Publik
				evaluateExpression()	Publik
				getEventHandlers()	Publik
				hasEvent()	Publik
				hasEventHandler()	Publik
				hasProperty()	Publik
				raiseEvent()	Publik
11	CModel	attributes	array	__call()	Publik
		errors	array	__get()	Publik
		iterator	CMapIterator	__isset()	Publik
		safeAttributeNames	array	__set()	Publik
		scenario	string	__unset()	Publik
		validatorList	CList	addError()	Publik
		validators	array	addErrorors()	Publik
				asa()	Publik
				attachBehavior()	Publik
				attachBehaviors()	Publik
				attachEventHandler()	Publik
				attributeLabels()	Publik

			attributeNames()	Publik
			behaviors()	Publik
			canGetProperty()	Publik
			canSetProperty()	Publik
			clearErrors()	Publik
			createValidators()	Publik
			detachBehavior()	Publik
			detachBehaviors()	Publik
			detachEventHandler()	Publik
			disableBehavior()	Publik
			disableBehaviors()	Publik
			enableBehavior()	Publik
			enableBehaviors()	Publik
			evaluateExpression()	Publik
			generateAttributeLabel()	Publik
			getAttributeLabel()	Publik
			getAttributes()	Publik
			getError()	Publik
			getErrors()	Publik
			getEventHandlers()	Publik
			getIterator()	Publik
			getSafeAttributeNames()	Publik
			getScenario()	Publik
			getValidatorList()	Publik
			getValidators()	Publik
			hasErrors()	Publik
			hasEvent()	Publik
			hasEventHandler()	Publik
			hasProperty()	Publik
			isAttributeRequired()	Publik
			isAttributeSafe()	Publik
			offsetExists()	Publik
			offsetGet()	Publik
			offsetSet()	Publik
			offsetUnset()	Publik
			onAfterConstruct()	Publik

				onAfterValidate()	Publik
				onBeforeValidate()	Publik
				onUnsafeAttribute()	Publik
				raiseEvent()	Publik
				rules()	Publik
				setAttributes()	Publik
				setScenario()	Publik
				unsetAttributes()	Publik
				validate()	Publik
				afterConstruct()	Privat
				afterValidate()	Privat
				beforeValidate()	Privat
				onAfterConstruct	Event
				onBeforeValidate	Event
				onAfterValidate	Event
				onUnsafeAttribute	Event
12	CModelBehavior	enabled	boolean	__call()	Publik
		owner	CComponent	__get()	Publik
				__isset()	Publik
				__set()	Publik
				__unset()	Publik
				asa()	Publik
				attach()	Publik
				attachBehavior()	Publik
				attachBehaviors()	Publik
				attachEventHandler()	Publik
				canGetProperty()	Publik
				canSetProperty()	Publik
				detach()	Publik
				detachBehavior()	Publik
				detachBehaviors()	Publik
				detachEventHandler()	Publik
				disableBehavior()	Publik
				disableBehaviors()	Publik
				enableBehavior()	Publik
				enableBehaviors()	Publik
		evaluateExpression()	Publik		

				events()	Publik
				getEnabled()	Publik
				getEventHandlers()	Publik
				getOwner()	Publik
				hasEvent()	Publik
				hasEventHandler()	Publik
				hasProperty()	Publik
				raiseEvent()	Publik
				setEnabled()	Publik
				afterConstruct()	Privat
				afterValidate()	Privat
				beforeValidate()	Privat
				Method Details	Privat
13	CModelEvent	handled	boolean	__call()	Publik
		isValid	boolean	__construct()	Publik
		params	mixed	__get()	Publik
		sender	object	__isset()	Publik
				__set()	Publik
				__unset()	Publik
				asa()	Publik
				attachBehavior()	Publik
				attachBehaviors()	Publik
				attachEventHandler() ()	Publik
				canGetProperty()	Publik
				canSetProperty()	Publik
				detachBehavior()	Publik
				detachBehaviors()	Publik
				detachEventHandler() ()	Publik
				disableBehavior()	Publik
				disableBehaviors()	Publik
				enableBehavior()	Publik
				enableBehaviors()	Publik
				evaluateExpression()	Publik
				getEventHandlers()	Publik
				hasEvent()	Publik
				hasEventHandler()	Publik
				hasProperty()	Publik

14	CModule			raiseEvent()	Publik
		basePath	string	__call()	Publik
		behaviors	array	__construct()	Publik
		components	array	__get()	Publik
		id	string	__isset()	Publik
		modulePath	string	__set()	Publik
		modules	array	__unset()	Publik
		params	CAttributeCollection	asa()	Publik
		parentModule	CModule	attachBehavior()	Publik
		preload	array	attachBehaviors()	Publik
				attachEventHandler() ()	Publik
				canGetProperty()	Publik
				canSetProperty()	Publik
				configure()	Publik
				detachBehavior()	Publik
				detachBehaviors()	Publik
				detachEventHandler() ()	Publik
				disableBehavior()	Publik
				disableBehaviors()	Publik
				enableBehavior()	Publik
				enableBehaviors()	Publik
				evaluateExpression()	Publik
				getBasePath()	Publik
				getComponent()	Publik
				getComponents()	Publik
				getEventHandlers()	Publik
				getId()	Publik
				getModule()	Publik
				getModulePath()	Publik
				getModules()	Publik
				getParams()	Publik
		getParentModule()	Publik		
		hasComponent()	Publik		
		hasEvent()	Publik		
		hasEventHandler()	Publik		
		hasModule()	Publik		
		hasProperty()	Publik		

				raiseEvent()	Publik
				setAliases()	Publik
				setBasePath()	Publik
				setComponent()	Publik
				setComponents()	Publik
				setId()	Publik
				setImport()	Publik
				setModulePath()	Publik
				setModules()	Publik
				setParams()	Publik
				init()	Privat
				preinit()	Privat
				preloadComponents()	Privat
15	CSecurityMa nager	behaviors	array	__call()	Publik
		cryptAlgorith m	mixed	__get()	Publik
		encryptionKe y	string	__isset()	Publik
		hashAlgorith m	string	__set()	Publik
		isInitialized	boolean	__unset()	Publik
		validation	string	asa()	Publik
		validationKe y	string	attachBehavior()	Publik
				attachBehaviors()	Publik
				attachEventHandler()	Publik
				canGetProperty()	Publik
				canSetProperty()	Publik
				decrypt()	Publik
				detachBehavior()	Publik
				detachBehaviors()	Publik
				detachEventHandler()	Publik
				disableBehavior()	Publik
				disableBehaviors()	Publik
				enableBehavior()	Publik
				enableBehaviors()	Publik
				encrypt()	Publik

				evaluateExpression()	Publik
				getEncryptionKey()	Publik
				getEventHandlers()	Publik
				getIsInitialized()	Publik
				getValidation()	Publik
				getValidationKey()	Publik
				hasEvent()	Publik
				hasEventHandler()	Publik
				hasProperty()	Publik
				hashData()	Publik
				init()	Publik
				raiseEvent()	Publik
				setEncryptionKey()	Publik
				setValidation()	Publik
				setValidationKey()	Publik
				validateData()	Publik
				computeHMAC()	Privat
				generateRandomKey() ()	Privat
				openCryptModule()	Privat
16	CStatePersister	behaviors	array	__call()	Publik
		cacheID	string	__get()	Publik
		isInitialized	boolean	__isset()	Publik
		stateFile	string	__set()	Publik
				__unset()	Publik
				asa()	Publik
				attachBehavior()	Publik
				attachBehaviors()	Publik
				attachEventHandler() ()	Publik
				canGetProperty()	Publik
				canSetProperty()	Publik
				detachBehavior()	Publik
				detachBehaviors()	Publik
				detachEventHandler() ()	Publik
				disableBehavior()	Publik
				disableBehaviors()	Publik
				enableBehavior()	Publik

				enableBehaviors()	Publik
				evaluateExpression()	Publik
				getEventHandlers()	Publik
				getIsInitialized()	Publik
				hasEvent()	Publik
				hasEventHandler()	Publik
				hasProperty()	Publik
				init()	Publik
				load()	Publik
				raiseEvent()	Publik
				save()	Publik

Tabel 3 Diagram Kelas Rinci

Bab IV Bab-Bab Hasil dan Pembahasan

Dalam bab ini perlu dilakukan pengujian untuk melihat apakah sintaks program yang telah dibuat bisa berjalan dengan baik dan tidak mengalami error.

IV.1 Implementasi

IV.1.1 Spesifikasi Hardware

Perangkat keras yang digunakan untuk menjalankan *framework* yang telah dibuat adalah sebagai berikut :

1. Laptop atau PC i5
2. 1 GB RAM
3. Harddisk 500 GB

IV.1.2 Spesifikasi Software

Selain membutuhkan perangkat keras, juga dibutuhkan perangkat lunak dalam implementasi yang dilakukan. Perangkat lunak yang digunakan untuk menjalankan *framework* yang telah dibuat adalah sebagai berikut :

1. *Text Editor* PHP Designer 8
2. *Text Editor* Notepad++
3. *Database* MySQL
4. *Browser* Mozilla Firefox
5. *Web Server* Xampp

IV.2 Pengujian

IV.2.1 Skenario Pengujian

Dalam melakukan pengujian, maka dibentuklah kelas tambahan. Kelas-kelas tambahan ini akan digunakan untuk memanggil kelas kelas dasar pada paket *system.base*. Keterangan lebih lanjut mengenai kelas tambahan ini ada pada tabel dibawah ini.

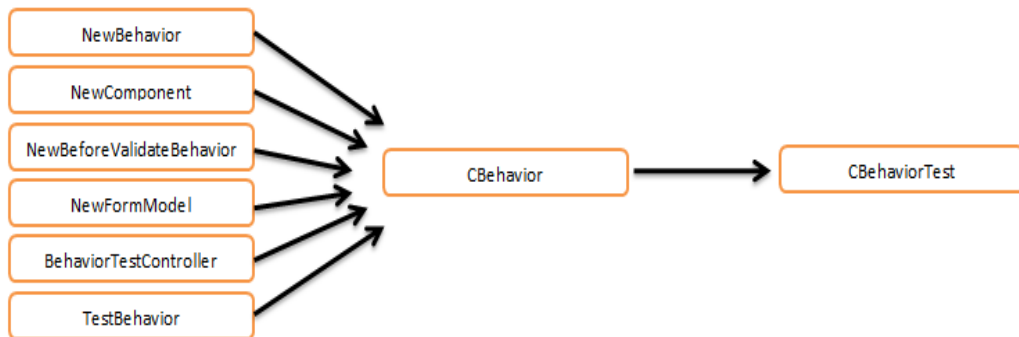
No	Nama Kelas	Nama <i>File Fisik</i>	Keterangan
1	AnotherNewApplicationComponent	AnotherNewApplicationComponent.php	kelas yang membantu dalam menguji kelas CModule. Kelas ini merupakan perluasan dari kelas CApplication
2	BehaviorTestController	BehaviorTestController.php	kelas yang merupakan perluasan dari kelas Ccontroller yang digunakan untuk membantu menangani pegujian kelas CBehavior
3	CApplicationComponentTest	CApplicationComponentTest.php	kelas yang merupakan perluasan kelas CTestCase. Kelas ini untuk menguji kelas CApplicationComponent
4	CBehaviorTest	CBehaviorTest.php	Kelas untuk menguji kelas CBehavior dan merupakan perluasan dari kelas CTestCase
5	CcomponentTest	CComponentTest.php	Kelas untuk menguji kelas CComponent dan merupakan perluasan dari kelas CTestCase
9	CModelTest	CModelTest.php	Kelas untuk menguji kelas CModel dan merupakan perluasan dari kelas CTestCase
10	CModuleTest	CModuleTest.php	Kelas untuk menguji kelas CModule dan merupakan perluasan dari kelas CTestCase
13	CsecurityManagerTest	CsecurityManagerTest.php	Kelas untuk menguji kelas CSecurityManager dan merupakan perluasan dari kelas CTestCase

14	CstatePersisterTest	CstatePersisterTest.php	Kelas untuk menguji kelas CStatePersister dan merupakan perluasan dari kelas CTestCase
15	NewApplicationComponent	NewApplicationComponent.php	Kelas untuk membantu pengujian pada CModule dan merupakan perluasan kelas CApplicationComponent
16	NewBeforeValidateBehavior	NewBeforeValidateBehavior.php	Kelas untuk membantu menangani validasi pada kelas CBehavior dan merupakan perluasan kelas CBehavior
17	NewBeforeValidateBehaviorException	NewBeforeValidateBehaviorException.php	Kelas pengecualian yang digunakan pada pengujian kelas Cbehavior dan merupakan perluasan CException
18	NewBehavior	NewBehavior.php	Kelas yang digunakan untuk menguji kelas CBehavior. Kelas ini merupakan perluasan kelas CBehavior
19	NewComponent	NewComponent.php	Kelas yang membantu pengujian pada kelas CComponent yang merupakan perluasan kelas CComponent
20	NewFormModel	NewFormModel.php	Kelas ini untuk membantu pengujian pada kelas CBehavior. Kelas ini merupakan perluasan CformModel
21	NewModel	NewModel.php	Kelas untuk membantu pengujian CModel, kelas ini merupakan perluasan dari kelas

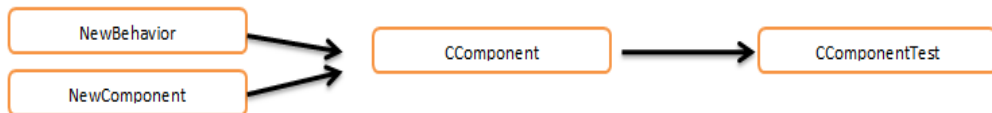
			CModel
22	NewModule	NewModule.php	Kelas untuk membantu pengujian CModule, kelas ini merupakan perluasan dari kelas CModule
23	TestBehavior	TestBehavior.php	Kelas yang menangani event pada saat pengujian terhadap kelas CBehavior.

Tabel 4 Deskripsi Kelas-kelas Tambahan

Untuk skenario pengujian pada kelas-kelas dasar *framework* dapat dilihat seperti gambar dibawah ini



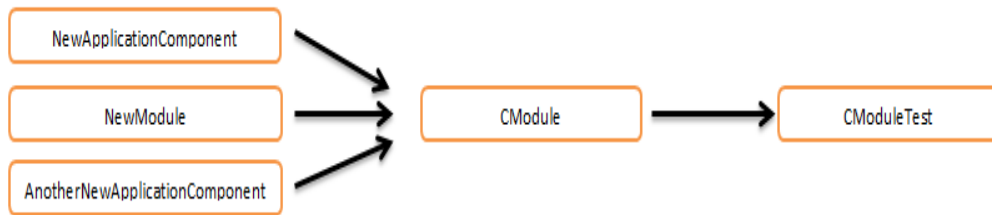
Gambar 4 Skenario pengujian pada kelas CBehavior



Gambar 5 Skenario pengujian pada kelas CComponent



Gambar 6 Skenario pengujian pada kelas CModel



Gambar 7 Skenario pengujian pada kelas CModule



Gambar 8 Skenario pengujian pada kelas CStatePersister



Gambar 9 Skenario pengujian pada kelas CApplicationComponent



Gambar 10 Skenario pengujian pada kelas CSecurityManager

Semua kelas tambahan merupakan perluasan dari kelas CTestCase pada paket test, namun ada juga kelas tambahan yang membutuhkan objek lain yang tidak disediakan oleh kelas penguji CTestCase. Oleh karena itu, skenario pengujiannya dibedakan menjadi dua berdasarkan objek yang akan digunakan dalam proses pengujian yaitu skenario dimana kelas penguji membutuhkan objek tambahan dari kelas tambahan yang di buat sebelum diuji oleh kelas penguji. Contoh skenario ini bisa dilihat pada gambar yang ada diatas yaitu pengujian yang terjadi pada kelas CBehavior, CModel, CComponent serta pada kelas CModule yang membutuhkan objek dari kelas lain dalam proses pengujiannya. Sedangkan untuk skenario lainnya, tidak membutuhkan objek tambahan lagi dan bisa langsung diuji oleh kelas penguji seperti halnya yang terjadi pada pengujian kelas CStatePersister, CApplicationComponent dan kelas CSecurityManager.

IV.2.2 Hasil Pengujian

No.	Nama Kelas	Nama Kelas Penguji	Data Hasil Uji	Target	Hasil Pengujian	
					Sesuai	Tidak Sesuai
1	CApplicationComponent	CApplicationComponentTest	Berhasil Dijalankan dan tidak <i>error</i>	Berhasil Dijalankan dan tidak <i>error</i>	✓	-
2	CBehavior	CBehaviorTest	Berhasil Dijalankan dan tidak <i>error</i>	Berhasil Dijalankan dan tidak <i>error</i>	✓	-
3	CComponent	CComponentTest	Berhasil Dijalankan dan tidak <i>error</i>	Berhasil Dijalankan dan tidak <i>error</i>	✓	-
4	CModel	CModelTest	Berhasil Dijalankan dan tidak <i>error</i>	Berhasil Dijalankan dan tidak <i>error</i>	✓	-
5	CModule	CModuleTest	Berhasil Dijalankan dan tidak <i>error</i>	Berhasil Dijalankan dan tidak <i>error</i>	✓	-
6	CSecurityManager	CsecurityManagerTest	Berhasil Dijalankan dan tidak <i>error</i>	Berhasil Dijalankan dan tidak <i>error</i>	✓	-
7	CStatePersister	CStatePersister	Berhasil Dijalankan dan tidak <i>error</i>	Berhasil Dijalankan dan tidak <i>error</i>	✓	-

Bab V Kesimpulan dan Saran

V.1 Kesimpulan

1. Rancangan awal *framework* ini dapat disusun dengan baik yaitu dalam bentuk diagram kelas secara keseluruhan yang berisi tentang paket - paket, kelas, dan telah tersusun secara teratur.
2. Dengan dapat dibangun dua paket awal dan sudah teruji maka bisa dilanjutkan pengembangan untuk paket dan kelas selanjutnya.
3. Melihat dari rumusan dan batasan masalah pada proyek ini, maka *framework* ini belum bisa digunakan untuk membangun aplikasi berbasis *web*, tetapi bisa dikembangkan dan dilengkapi lagi agar bisa digunakan untuk membangun aplikasi *web* yang diharapkan.

V.2 Saran

Dengan telah disusunnya Pnb *framework* untuk kampus Politeknik Negeri Batam ini, maka diharapkan dapat segera digunakan untuk membangun aplikasi berbasis *web*. Namun proyek ini masih belum lengkap dan perlu banyak penambahan dan perbaikan.

Proyek ini sudah dirancang dengan teliti dan sudah dilakukan pengerjaan sebagian paket untuk menjadi acuan dalam proses melengkapi dan mengembangkan. Untuk itu bagi pengembang selanjutnya disarankan untuk membangun paket dan kelas selanjutnya.

Dalam pengembangannya sangat disarankan untuk mempelajari *Framework* YII sebagai acuan agar mempercepat pengembangan. Sangat disarankan dalam melengkapi *framework* ini untuk mengikuti proses yang sudah berjalan

sebelumnya yaitu pengembangan bertahap dengan mengikuti paket atau pengembangannya dilakukan perpaket. Tahap pengerjaan pengembangannya secara berurutan dari paket dasar atau paket sistem terlebih dahulu agar pengerjaannya terstruktur dan mempermudah dalam proses pengujian

DAFTAR PUSTAKA

Hanya berisi daftar karya orang lain yang diacu dalam dokumen TA

1. *Phpframeworks*. “Top 10 Ranking PHP Frameworks”. <http://www.phpframeworks.com/top-10-php-frameworks/> (diakses pada tanggal 25 Mey 2013).
2. Uinjkt. “Analisis Perbandingan *framework* php berdasarkan moose ck dan properti kualitas disain menggunakan metode analytic hierarchy process (AHP)”. <http://repository.uinjkt.ac.id/dspace/handle/123456789/2259> (diakses pada tanggal 28 Mey 2013).
3. Wikipedia. “PHP”. <http://id.wikipedia.org/wiki/PHP> (diakses pada tanggal 9 November 2012).
4. Wikipedia. “Pembelajaran Elektronik”. <http://id.wikipedia.org/wiki/E-learning> (diakses pada tanggal 9 November 2012).
5. Wikipedia. “Moodle”. <http://id.wikipedia.org/wiki/Moodle> (diakses pada tanggal 9 November 2012).
6. Wikipedia. “Sistem Manajemen Konten”. <http://id.wikipedia.org/wiki/CMS> (diakses pada tanggal 9 November 2012).
7. Wikipedia. “Kerangka Kerja”. <http://id.wikipedia.org/wiki/Framework> (diakses pada tanggal 9 November 2012).
8. Maniacms. “Mengenal *Framework* Codeigniter” <http://www.maniacms.web.id/2012/10/mengenal-framework-codeigniter.html> (diakses pada tanggal 26 Juli 2013)
9. Wikipedia. “MySQL”. <http://id.wikipedia.org/wiki/MySQL> (diakses pada tanggal 26 Juli 2013).
10. *Yiiframework*. “Tutorial Yii Framework”. <http://www.yiiframework.com/forum/index.php/topic/378-tutorial-yii-framework/> (diakses pada tanggal 9 November 2012).

11. Github. “Yii PHP *Framework*”. <https://github.com/yiisoft/yii> (diakses pada tanggal 15 Juni 2013).
12. Septian, Wahyu Rifa’i Dwi. Analisis Perbandingan *Framework* PHP Berdasarkan Moose CK dan Properties Kualitas Disain Menggunakan Metode Analytic Hierarchy Process (AHP). Jakarta : Universitas Islam Negeri Syarif Hidayatullah. 2010.
13. Mutiara, A.B., A.Muslim, T. Oswari & R. A. Miharja. Pengembangan Open E-health Gunadarma Information System. Depok : Penerbit Gunadarma.2012.
14. Joomla. “Pengenalan Model *View Controller*”. <http://www.id-joomla.com/tutorial/182-pengenalan-model-view-controller> (diakses pada tanggal 21 November 2012).
15. Diskusiweb. “Fungsi Kelas Interface”. <http://www.diskusiweb.com/discussion/3173/fungsi-kelas-interface/p1> (diakses pada tanggal 18 Februari 2013).
16. Teknojurnal. “Tutorial Pemrograman PHP Berbasis MVC Menggunakan *Framework* CodeIgniter dan Doctrine”. <http://www.teknojurnal.com/2010/02/21/tutorial-pemrograman-php-berbasis-mvc-menggunakan-framework-codeigniter-dan-doctrine-bagian-1-setup-dan-instalasi/> (diakses pada tanggal 21 November 2012).
17. Teknojurnal. “*Framework* MVC – Apa Pengaruhnya Bagi Para Programmer?”. <http://www.teknojurnal.com/2010/02/24/framework-mvc-apa-pengaruhnya-bagi-para-programmer/> (diakses pada tanggal 21 November 2012).
18. Winesett, Jeffrey. Agile *Web* Application Development with Yii 1.1 and PHP5. Birmingham : Packt Publishing Ltd.2010.
- 19.

Lampiran I

Lampiran II