

# PERANCANGAN SISTEM INFORMASI BEASISWA UNIVERSITAS DIPONEGORO BERBASIS WEB

Bayu Ardiyanto  
Maman Somantri, ST.,MT.  
Ir. Kodrat Iman Satoto, MT.

## ABSTRACT

*Along with advances in information technology, the use of web-based information system is in great demand. The ability to be accessed anytime and anywhere is one reason such systems. Diponegoro University is one of the state university is quite well known in the community, certainly a lot of parties that want to offer a scholarship at the University of Diponegoro in order to help smooth the student studies but less capable of achieving its economic circumstances. Therefore necessary to develop a web-based information systems scholarship that is expected later delivery scholarships to more appropriate and well targeted.*

*This final project includes the design and development of information systems scholarships web-based Diponegoro University. Software design in this thesis using the Unified Modeling Language (UML) and using Model View Controller for designing the components. System with the user interface is set and controlled by a web page designed using Joomla! as its framework and PHP programming language that runs on the Apache service. Last is the implementation and testing system that includes the process of implementation of data base, black box testing and white box testing.*

*The test results showed that the application can work with in accordance with the mechanism of scholarship applications that have been designed. This application has three levels of users are Administrators, Registered Users, and General User. This application can handle with good process management and Diponegoro University scholarship application.*

**Keywords:** *scholarship, web-based information systems, UML, MVC.*

## I. PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan kemajuan teknologi informasi, penggunaan sistem informasi berbasis *web* sangat diminati. Kemampuan untuk diakses kapan saja dan dimana saja merupakan salah satu alasan sistem seperti ini. Adanya Sistem Informasi Akademik (SIA) Universitas Diponegoro memudahkan mahasiswa dan dosen dalam proses administrasi seperti pengisian kartu rencana studi, penyajian transkrip nilai, pencarian data mahasiswa dan lain sebagainya.

Universitas Diponegoro merupakan salah satu universitas negeri yang cukup dikenal luas di masyarakat, tentunya banyak sekali pihak-pihak yang ingin menawarkan beasiswa pada Universitas Diponegoro guna membantu kelancaran studi mahasiswanya yang berprestasi namun kurang mampu keadaan ekonominya.

Sebagai rintisan universitas riset, ada baiknya kita memanfaatkan kecanggihan dunia maya untuk memudahkan para mahasiswa mengakses berbagai informasi beasiswa yang ada di Universitas Diponegoro. Oleh karena itu Penulis merasa perlu mengembangkan sistem informasi beasiswa yang diharapkan nantinya penyampaian beasiswa dapat lebih tepat guna dan tepat sasaran.

### 1.2 Tujuan Tugas Akhir

Tujuan tugas akhir ini yaitu untuk merancang dan membuat sebuah aplikasi sistem informasi beasiswa berbasis *web* di Universitas Diponegoro.

### 1.3 Batasan Masalah

Pada perancangan ini, masalah dibatasi pada ruang lingkup sebagai berikut.

1. Antarmuka dirancang dengan bahasa pemrograman PHP
2. Basis Data yang digunakan adalah MySQL
3. Sistem ini berjalan di atas layanan *server* Apache
4. Hal teknis mengenai prosedur standar dari pengajuan beasiswa secara konvensional diabaikan
5. Perancangan ini hanya merupakan rancang bangun sebuah sistem yang belum diaplikasikan.
6. Penggunaan Joomla! hanya sebagai *framework* untuk penanaman komponen yang dibuat.

## II. DASAR TEORI

### 2.1 Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien.

Sistem informasi berbasis komputer (*Computer Based Information System*) terdiri dari komponen-komponen sebagai berikut :

- a. Perangkat keras (*hardware*)  
Merupakan komponen untuk melengkapi kegiatan memasukkan data, memproses data dan keluaran data.
- b. Perangkat lunak (*software*)  
Adalah program dan instruksi yang diberikan ke komputer untuk menjalankan sistem.

- c. Basis data (*database*)  
Merupakan sekumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.
- d. Telekomunikasi  
Adalah sebuah sistem yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.
- e. Manusia (*human*)  
Yaitu personel dari sistem informasi, meliputi manajer, analis, *programmer*, dan *operator*, serta bertanggung jawab terhadap perawatan sistem.
- f. Prosedur  
Yaitu tata cara yang meliputi strategi, kebijakan, metode, dan peraturan-peraturan dalam menggunakan sistem informasi.

## 2.2 UML

UML adalah keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).

### 2.2.1 Diagram dalam UML

#### *Use Case Diagram*

Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Secara garis besar, *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.

#### *Class Diagram*

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

#### *Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

#### *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas

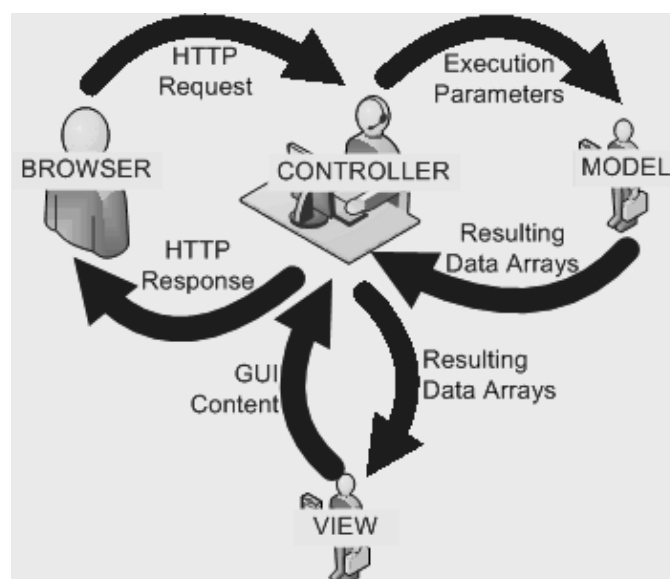
dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

## 2.3 Model View Controller (MVC)

MVC adalah sebuah pola (*pattern*) yang mengorganisasikan dan memisahkan *software* menjadi tiga bagian yang terpisah :

- *Model*, bagian yang mengorganisasikan *business logic*, *application data* dan *application flow* dari *software*.
- *View*, berperan untuk menampilkan data dari bagian Model.
- *Controller*, berfungsi untuk mengarahkan aliran aplikasi, menerima input, dan menerjemahkan untuk *Model* dan *View*.

### 2.3.1 Cara kerja MVC



Gambar 1 Pola Model View Controller

Dari gambar di atas terlihat aliran kerja dari MVC. Pengguna mengakses *website* dengan *browser*, *browser* mengirimkan permintaan kepada *controller* yang kemudian meresponnya dengan berkomunikasi dengan *model*. Komunikasi ini dapat berupa akses data atau mengubah data yang tersimpan pada *model*. Kemudian *controller* akan berkomunikasi dengan *view* untuk kemudian akan ditampilkan sebagai respon balik dari permintaan pengguna tersebut.

Secara garis besar, *Model View Controller* atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memrosesnya (*Controller*), seperti yang sudah dijelaskan di atas.

## III. PERANCANGAN PERANGKAT LUNAK

Analisis kebutuhan telah dilakukan baik melalui wawancara dengan pihak-pihak terkait dan observasi dari lingkungan dimana sistem akan dibangun. Dapat disimpulkan kebutuhan yang diinginkan sebagai berikut :

- **Kebutuhan Fungsional**
  - a. Adanya fasilitas untuk menampilkan pengumuman maupun informasi beasiswa.
  - b. Adanya fasilitas pendaftaran untuk beasiswa di Universitas Diponegoro.
  - c. Adanya fasilitas untuk melihat daftar mahasiswa yang mendaftar beasiswa.
  - d. Adanya fasilitas untuk melihat mahasiswa yang menerima beasiswa.
  - e. Adanya pembagian pengguna menjadi tiga kelompok sebagai berikut :
    - *User* Umum (tanpa *login*)
    - *User* Terdaftar (dengan *login*)
    - Administrator (*admin*)
  - f. *User* Terdaftar dapat menikmati semua fasilitas yang disediakan.
  - g. *User* umum hanya dapat menikmati fasilitas informasi saja tanpa dapat mendaftar beasiswa.
  - h. Adanya administrator untuk *me-maintenance website*.

- **Kebutuhan Non-Fungsional**

- a. Operasional
  - Dapat diakses client dari berbagai *browser* yang mendukung *javascript*.
  - Adanya perbedaan halaman *user* dan *admin*.
- b. Keamanan
  - Adanya penggunaan *password* dalam *form login* untuk membedakan tipe *user* termasuk hak akses masing-masing.
  - Pengguna umum hanya dapat melihat halaman utama saja tanpa dapat melakukan pendaftaran.
- c. Informasi
  - Adanya informasi tata cara registrasi, pendaftaran dan peraturan dalam melaksanakan prosedur pendaftaran beasiswa.
  - Informasi yang diberikan selalu *up to date*.

### 3.2 Pemodelan Perangkat Lunak

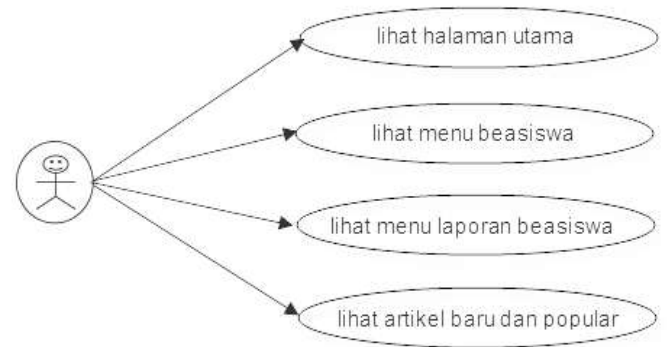
Tahapan pemodelan menggunakan pendekatan berorientasi objek dengan menggunakan metode UML. yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak terdiri atas *functional modeling*, *structural modeling*, dan *behavioral modeling*.

#### 3.2.1 Functional Modeling

*Functional modeling* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, bukan “bagaimana” sistem bekerja. Yang termasuk dalam *functional modeling* adalah *activity diagram* dan *use-case diagram*.

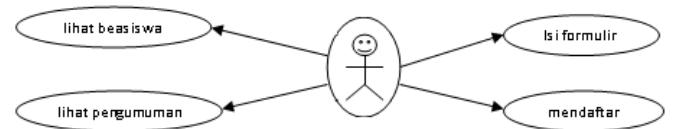
- **Use Case Diagram**

Ada tiga buah *use-case* yang dipakai pada perancangan ini, masing-masing *use-case* mewakili tiap pengguna pada sistem informasi ini.



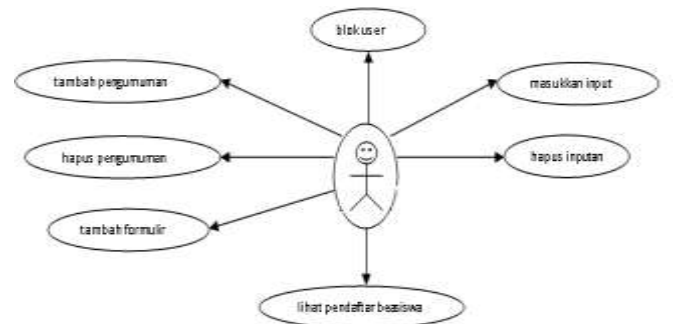
**Gambar 2** Use-case user umum

*Use-case* di atas adalah *use-case* untuk pengguna umum pada sistem informasi ini. *User* umum tidak mempunyai akun pada sistem informasi ini. Hak yang diberi pada *user* umum ini adalah melihat halaman beasiswa, menu beasiswa yg ditawarkan, menu laporan beasiswa dan melihat artikel.



**Gambar 3** Use-case user terdaftar

*Use-case* di atas adalah *use-case* untuk pengguna terdaftar. Hak yang didapat sama dengan *user* umum tetapi mendapat keistimewaan pada saat setelah *login user* terdaftar tersebut dapat mendaftar beasiswa.

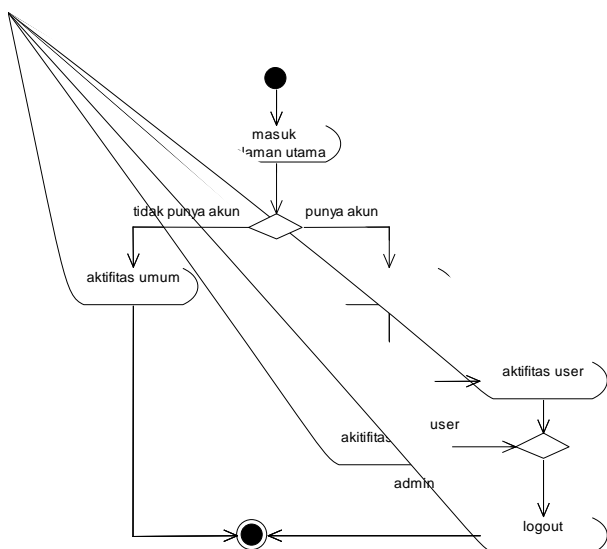


**Gambar 4** Use-case admin

*Use-case* ini adalah *use-case* untuk admin yang memiliki hak dalam *maintenance website*. Hak tersebut antara lain untuk menambah dan menghapus pengumuman, membuat formulir beasiswa, melihat mahasiswa yang mendaftar beasiswa dan menggeblok *user*.

- **Activity Diagram**

Dari *use-case* yang telah dibuat akan diperjelas dengan aliran aktifitas dalam sistem mulai dari awal sampai akhir yang disajikan dalam berbagai *level* pengguna dan diawali diagram aktivitas sistem secara umum sebagai berikut.

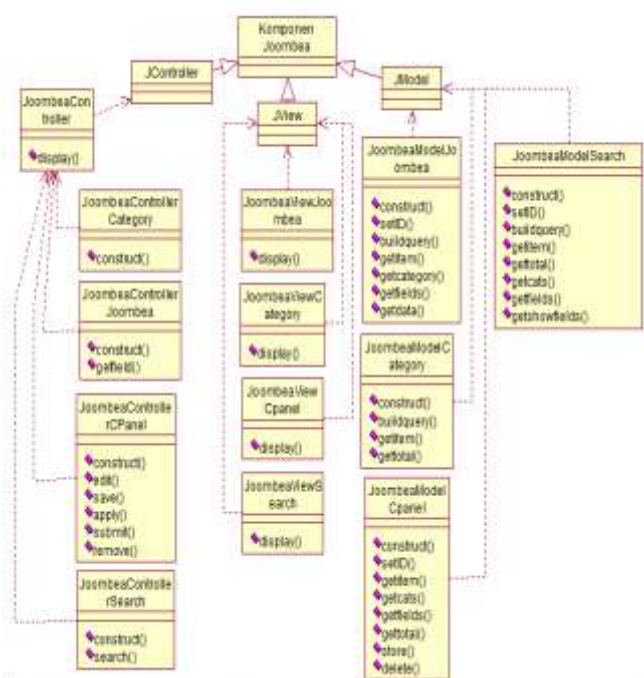


Gambar 5 Activity diagram sistem secara umum

Diagram di atas merupakan *activity diagram* sistem secara umum. Terdapat perbedaan hak dari ketiga *user* yang ada.

### 3.2.2 Structural Modeling

*Structural modeling* atau biasa disebut juga dengan *conceptual model*, mendeskripsikan tentang struktur dari data yang mendukung kinerja dari suatu organisasi. Yang termasuk dalam *structural modeling* adalah *class diagram*. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam perangkat lunak yang sedang kita kembangkan. *Class diagram* menggambarkan struktur dan deskripsi kelas beserta hubungan kelas satu dengan yang lain.

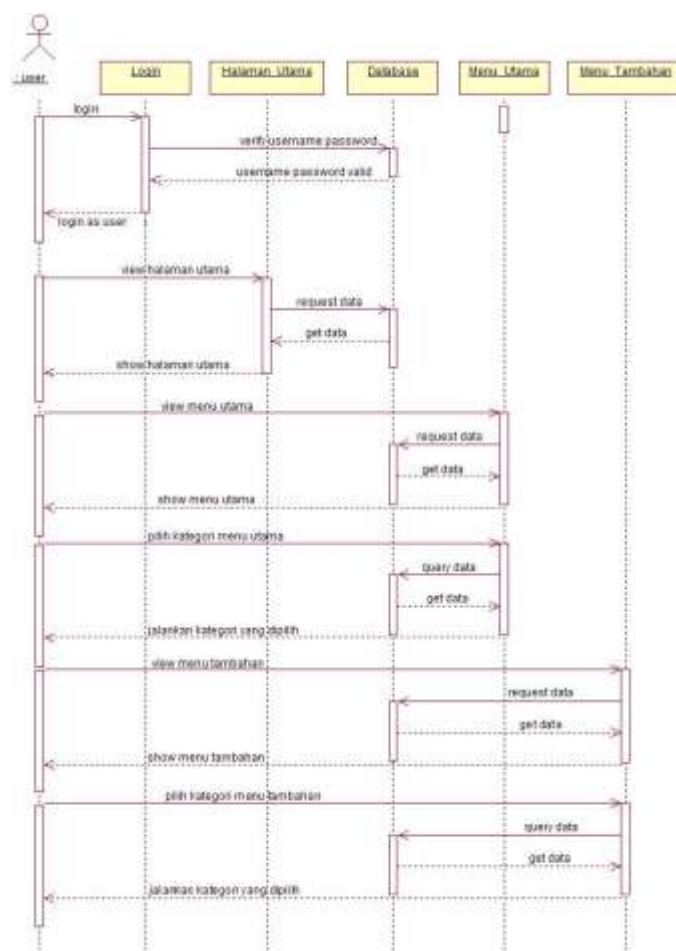


Gambar 6 Class Diagram komponen

### 3.2.3 Behavioral Modeling

*Behavioral modeling* mendeskripsikan aspek internal yang bersifat dinamis dari sebuah sistem informasi yang mendukung kinerja dari suatu organisasi. Yang tergolong dalam *behavioral modeling* adalah *sequence diagram*. *Sequence diagram* digunakan untuk menunjukkan interaksi antar *user* dengan objek-objek lain yang terkait. Diagram ini lebih menitikberatkan pada urutan pesan yang dikirimkan antar *user* dan obyek berdasarkan waktu. Biasanya diagram ini digunakan untuk menggambarkan skenario dalam suatu *use-case* yang ingin dijelaskan secara detail atau juga dapat digunakan untuk menggambarkan sebuah proses dalam *use-case*.

Gambar 7 Sequence diagram user terdaftar



Gambar 7 di atas merupakan *sequence diagram* untuk level pengguna terdaftar. Hal pertama adalah melakukan *view* tampilan awal program. setelah itu melakukan *login*. Program akan melakukan proses verifikasi dan validasi ke basis data. Jika sudah *valid* maka *login* sukses, dan *user* terdaftar tersebut dapat mengakses menu yang ada sesuai dengan hak yang diberikan.

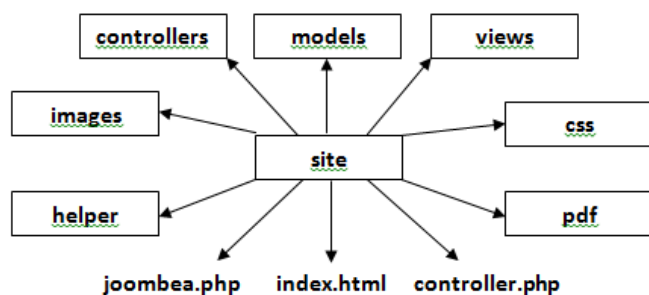
### 3.3 Perancangan Komponen dengan Pola Model View Controller

Pada tahapan ini, perancangan komponen yang dipakai pada sistem informasi ini untuk ditanamkan pada *framework* Joomla! menggunakan

pola *Model View Controller*. Komponen yang dipakai yaitu *com\_joombea* untuk menampilkan jenis beasiswa.

### 3.3.1 Perancangan *com\_joombea*

- Perancangan pada *front end*

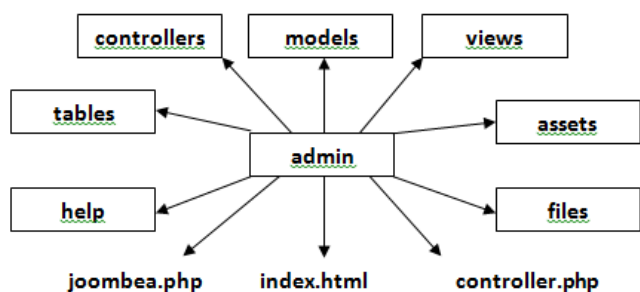


Gambar 8 Pola *front end* *com\_joombea*

Diagram di atas menunjukkan beberapa *folder* dan *file* yang dibutuhkan pada bagian *front end* Joomla!. *Folder controller* mempunyai fungsi mengarahkan aliran aplikasi, menerima input, dan menerjemahkan untuk *Model* dan *View*, *folder model* berfungsi mengorganisasikan *business logic*, *application data* dan *application flow*. *Folder views* berfungsi menampilkan data dari bagian *Model*. *Folder images* berfungsi untuk menampung data *images* yang digunakan pada komponen ini, *folder css* berfungsi untuk mengatur *preference* dari tampilan pada *front end*, *folder pdf* berisi segala pengaturan yang terkait dengan *file* berekstensi *.pdf*.

Pada waktu kita memanggil komponen, Joomla! akan mengeksekusi *joombea.php* untuk pertama kalinya. *Joombea.php* itu akan menerima *HTTP request* dan meneruskannya ke *controller*. *Controller* inilah yang memegang kendali aliran aplikasi. Ia akan melakukan permintaan data dengan melakukan *getmodel()* dan juga memilih *template / presentation layer* untuk memformat tampilan melalui *getview()*. *Presentation layer* tersebut akan mengikat data dari *model* dan ditampilkan kembali ke layar.

- Perancangan pada *back end*



Gambar 9 Pola *back end* *com\_joombea*

Diagram pola di atas menunjukkan beberapa *folder* dan *file* yang dibutuhkan pada bagian *back end* Joomla!. Terdiri atas *folder controller*, *models*, *views*, *assets*, *files*, *tables*, dan *help*. Semua fungsi *folder* pada pola di *back end* ini sama dengan pola bagian *front end*.

Pada waktu kita memanggil komponen, Joomla! akan mengeksekusi *joombea.php* untuk pertama kalinya. *File joombea.php* itu akan menerima *HTTP request* dan meneruskannya ke *controller*. *Controller* ini yang memegang kendali aliran aplikasi. Ia akan melakukan permintaan data dengan melakukan *getmodel()* dan juga memilih *template/presentation layer* untuk memformat tampilan melalui *getview()*. *Presentation layer* tersebut akan mengikat data dari *model* dan ditampilkan kembali ke layar.

## IV. IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas tentang implementasi dan pengujian Sistem Informasi Beasiswa Universitas Diponegoro. Implementasi yang dilakukan ini adalah merupakan cara bagaimana mewujudkan hasil dari perancangan sistem yang sudah dilakukan sehingga menghasilkan suatu aplikasi yang dapat bekerja sebagaimana mestinya. Pengujian sistem dilakukan dengan menggunakan Mozilla firefox sebagai *web browser*, mesin basis data yang digunakan adalah MySQL yang bekerja pada *web server* Apache.

### 4.1 Implementasi Model View Controller

Setelah perancangan sudah terbangun, tahapan berikutnya adalah implementasi komponen tersebut ke dalam *framework* Joomla!.

#### 4.1.1 Implementasi *com\_joombea*

Hal yang dilakukan pertama kali adalah membuat *file install.joombea.php*. Berikut cuplikan senarai programnya :

```

/* membuat folder directory untuk
menampung berkas dokumen yang akan
diunggah */
if(jfolder::create(JPATH_SITE.DS.'image
s'.DS.'directory'))
jfile::copy(JPATH_SITE.DS.'components'.
DS.'com_joomrepo'.DS.'index.html',
JPATH_SITE.DS.'images'.DS.'directory'.D
S.'index.html');
  
```

```

/* instalasi modul latest entry untuk
menampilkan berkas yang paling akhir
diunggah */
  
```

```

if(jfolder::move(JPATH_ADMINISTRATOR.DS
.'components'.DS.'com_joomrepo'.DS.'fil
es'.DS.'mod_latest_entry',
JPATH_SITE.DS.'modules'.DS.'mod_latest_
entry')) {
else {
echo 'Latest Entry modules could not be
installed. Please move the folder of
Latest entry module from
''.JPATH_ADMINISTRATOR.DS.'components'.
DS.'com_joomrepo'.DS.'files'.DS.'mod_la
test_entry" to
''.JPATH_SITE.DS.'modules'.DS.'mod_late
st_entry"';
$flag = false;
}
  
```

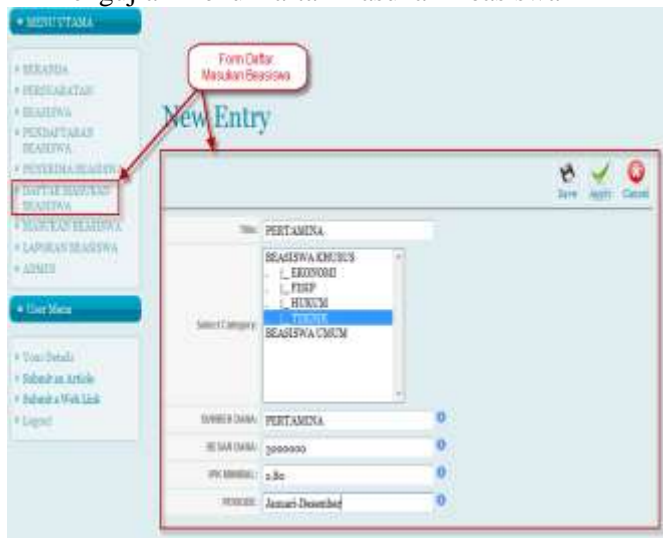
\$error = false;

## 4.2 Pengujian Sistem

Pengujian dilakukan dengan menggunakan metode *white box* dan *black box*.

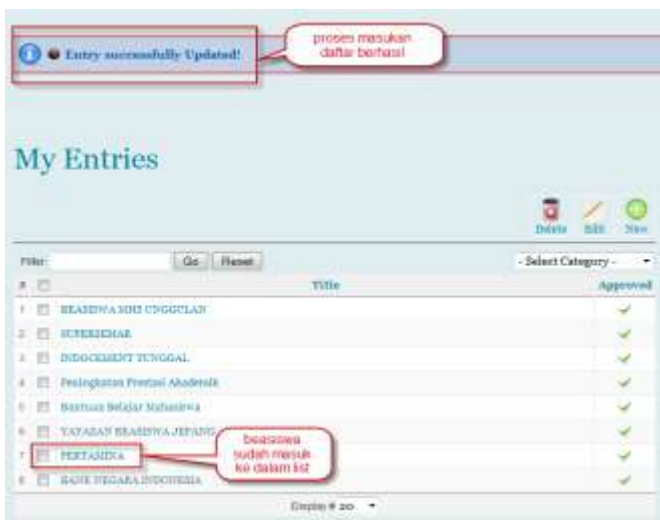
### 4.2.1 Pengujian Black Box

- Pengujian menu Daftar Masukan Beasiswa



Gambar 12 Menu daftar masukan beasiswa

Menu Daftar Masukan beasiswa di atas adalah hasil dari *com\_joombea* yang telah dibuat. *Maintenance* menu hanya dilakukan oleh admin. Selain admin, *user* lain tidak dapat mengakses menu tersebut. Pada menu tersebut diisikan informasi beasiswa secara lengkap oleh admin.



Gambar 13 Hasil masukan beasiswa

Gambar di atas adalah hasil masukan dari menu daftar masukan beasiswa. Dua gambar di atas tersebut hanya bisa dilihat oleh admin karena admin mempunyai hak untuk melakukan *maintenance*. Untuk *user* selain admin dapat melihat pada menu beasiswa yang ada di beranda sistem informasi beasiswa.

Tipe	SALURAN DANA	BESAR DANA	PH BUNYAL	PERIODE
Beasiswa Belajar Mahasiswa	Bantuan Belajar Mahasiswa	3000000	2.75	Januari-Desember
Peningkatan Prestasi Akademik	Peningkatan Prestasi Akademik	3000000	3.00	Januari-Desember
SUPERSEMAR	SUPERSEMAR	1200000	2.50	Januari-Desember
BANK INDONESIA	BANK INDONESIA	3000000	3.00	Januari-Desember

Gambar 14 Menu beasiswa/beasiswa umum

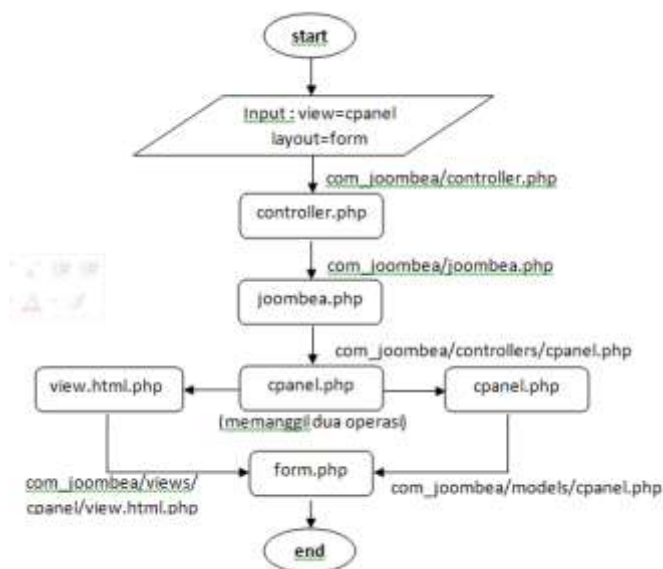
Gambar di atas adalah menu beasiswa/beasiswa umum yang dapat dilihat oleh *user* umum maupun terdaftar.

### 4.2.2 Pengujian White Box

Pada pengujian *white box* ini dicontohkan dengan menu **Daftar Masukan Beasiswa** dengan alamat *URL* :

[http://localhost/beasiswa/index.php?option=com\\_joombea&view=cpanel&layout=t=form](http://localhost/beasiswa/index.php?option=com_joombea&view=cpanel&layout=t=form)

Dari *URL* di atas akan dijelaskan alur logika *com\_joombea* dengan masukan **view=cpanel** dan **layout=form**. Alur tersebut seperti terlihat pada *flowchart* di bawah ini.



Gambar 15 Alur kerja pengujian white box

Pada gambar di atas diketahui dengan masukan **view=cpanel** dan **layout=form**. Lalu dilakukan pemanggilan *file controller.php*, oleh *file* tersebut lalu diteruskan kepada *file joombea.php*. oleh *file joombea.php* tersebut kemudian diteruskan ke *file cpanel.php* yang ada pada *folder com\_joombea/controller*. Oleh *file cpanel.php* tersebut dipanggil dua operasi yang dilakukan oleh *file view.html.php* di dalam *folder com\_joombea/views/cpanel* dan *file cpanel.php* pada *folder com\_joombea/models*. Setelah operasi tersebut selesai maka didapatkan keluarannya yaitu *form.php*.

## PENUTUP

### Kesimpulan

Dari hasil analisa dan pembahasan dapat disimpulkan bahwa :

- 1) Telah dirancang dan dibangun sebuah Sistem Informasi Beasiswa Universitas Diponegoro dengan menggunakan Joomla! hanya sebagai *framework* saja untuk aplikasi penanaman komponennya, bahasa pemrograman PHP dan basis data MySQL.
- 2) Sistem yang dibangun merupakan aplikasi berbasis web yang dapat dijalankan dan diakses melalui *web browser* yang mendukung *javascript*.
- 3) Hasil pengujian *Black Box* menunjukkan bahwa sistem dapat bekerja dengan baik dalam proses pengolahan dan pelacakan data-data yang terkait dengan beasiswa Undip. Aplikasi dapat menangani dengan baik data-data yang dimasukkan ke dalam sistem, baik data dengan format yang benar maupun data dengan format yang tidak sesuai.
- 4) Hasil pengujian *White Box* menunjukkan bahwa aplikasi telah sesuai dengan spesifikasi sistem yang diinginkan. Aplikasi dapat menangani dan memproses dengan baik pernyataan-pernyataan kondisional yang terdapat dalam senarai program.

### Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah :

- 1) Aplikasi Sistem Informasi Beasiswa Universitas Diponegoro ini dapat diterapkan secara *online* pada jaringan internet sehingga bisa diakses dan dijalankan pengguna dari jarak jauh.
- 2) Dapat dikembangkan lagi dengan pengintegrasian Sistem Informasi Beasiswa ini dengan Sistem Informasi Akademik Universitas Diponegoro dengan akses masuk menggunakan akun SIA.
- 3) Dapat dikembangkan aplikasi Sistem Informasi Beasiswa Universitas Diponegoro dengan menggunakan bahasa pemrograman selain PHP dan mesin basis data selain MySQL agar dapat dibandingkan kinerjanya.

### DAFTAR PUSTAKA

- [1] Ladjamudin, Al Bahra Bin., *Analisis dan Desain Sistem Informasi*, Penerbit Graha Ilmu, Yogyakarta, 2007
- [2] Al Fatta, Hanif. 2007. *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta: CV. Andi Offset
- [3] Kadir, Abdul, *Dasar Pemrograman Web Dinamis Menggunakan PHP*, Penerbit Andi, Yogyakarta, 2008

- [4] Kadir, Abdul. 2009. *Mastering AJAX dan PHP*. Yogyakarta : CV. Andi Offset
- [5] Yakub. 2008. *Sistem Basis Data ; Tutorial Konseptual*. Yogyakarta : Graha Ilmu
- [6] Madcoms, *Aplikasi Program PHP dan MySql*, Penerbit Andi, Yogyakarta, 2004.
- [7] Dennis, Alan, 2005. *System Analysis and Design with UML Version 2.0*, WILEY, New Jersey
- [8] Fowler, Martin. 2004. *UML Distilled 3<sup>th</sup> Ed. A Brief Guide to the Standard Object Modelling Language*. Pearson Education, Inc.
- [9] Munawar, *Pemodelan Visual dengan UML*, Penerbit Graha Ilmu, 2005
- [10] Siswoutomo, Wiwit. 2008. *Step by Step Joomla! Programming*. Jakarta: PT. Elex Media Computindo.
- [11] <http://www.doc.joomla.org/MVC>

### BIODATA



BAYU ARDIYANTO  
(L2F005520) Dilahirkan di Semarang, 28 Juni 1987. Menempuh pendidikan dasar sampai sekolah menengah atas di Semarang. Dan semenjak tahun 2005 hingga kini sedang menyelesaikan studi Strata- 1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang, Konsentrasi Informatika dan Komputer.

Menyetujui,  
Dosen Pembimbing I

Maman Somantri, S.T., M.T.  
NIP. 197406271999031002

Dosen Pembimbing II

Ir. Kodrat Iman Satoto M.T.  
NIP. 196310281993031002