

PERANCANGAN SISTEM INFORMASI PUSAT RISET UNIVERSITAS DIPONEGORO BERBASIS WEB

Kumboro Bimantyo Kurniatama

Maman Somantri, ST.,MT.

Ir. Kodrat Iman Satoto, MT.

ABSTRACT

Along with advances in information technology, the use of web-based information systems are very popular because of its ability to be accessed anytime and anywhere. Given one Undip vision is "a University of Diponegoro University Research Excellence in 2020" certainly needed a container to accommodate the works of students in the form of research, scientific papers, thesis and other matters related to the research. This has challenged the author to write task finally with the title "Design of Information Systems Web-Based Research Center at the University of Diponegoro".

This final project includes the design and development of information systems Research Center, University of Diponegoro (Diponegoro University Research Center) is web based. Software design in this thesis using the Unified Modeling Language (UML). System with the user interface is set and controlled by a web page that is designed to instill a component with MVC pattern (Model View Controller) in the framework Joomla! and the 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.

This application has four levels of users are Administrator, General User, User Diponegoro University, and Public Users. The results show that applications can work well in the process of managing research documents, conducting research agenda, and communication among students at the University of Diponegoro.

Keyword : *research, web-based information system, UML, MVC*

I. PENDAHULUAN

1.1 Latar Belakang

Seiring dengan kemajuan teknologi informasi, penggunaan sistem informasi berbasis web sangat diminati karena kemampuannya untuk diakses kapan saja dan dimana saja.

Mengingat salah satu visi Undip yaitu "Universitas Diponegoro menjadi Universitas Riset yang Unggul pada Tahun 2020" tentunya diperlukan suatu wadah untuk menampung karya-karya mahasiswa baik berupa penelitian, karya ilmiah, tugas akhir dan berbagai hal yang terkait dengan riset. Hal inilah yang menantang penulis untuk menulis tugas akhirnya dengan judul "Perancangan Sistem Informasi Pusat Riset Berbasis Web di Universitas Diponegoro".

Diharapkan nantinya dengan adanya sistem ini menjadikan sebuah inspirasi dan pemacu motivasi bagi para mahasiswa untuk mengembangkan kemampuannya dalam menghasilkan karya-karya yang sangat berguna bagi Undip di masa kini dan masa yang akan datang.

1.2 Tujuan

Tujuan tugas akhir ini yaitu untuk merancang dan membuat sebuah aplikasi sistem informasi berbasis *web* sebagai pusat riset dalam lingkup Universitas Diponegoro.

1.3 Batasan Masalah

Agar tidak menyimpang dari pokok pembahasan, maka permasalahan dari tugas akhir ini dibatasi pada:

1. Aplikasi yang dibuat masih berupa *prototype* sistem informasi berbasis web
2. Aplikasi dibangun dengan *framework* Joomla! 1.5.20, bahasa pemrograman PHP 5.3.0, basis data MySQL 5.1.36, dan *web server* Apache 2.2.11.
3. Tidak membahas keamanan sistem serta tindakan pengamanan yang berhubungan dengan konfigurasi jaringan (*firewall* dan lain-lain)

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 seperti

- Perangkat keras (*hardware*) Yaitu perangkat keras komponen untuk melengkapi kegiatan memasukan data, memproses data dan keluaran data.
- Perangkat lunak (*software*) Yaitu program dan instruksi yang diberikan ke komputer untuk menjalankan sistem.
- Database* Yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.
- Telekomunikasi Yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.
- Manusia Yaitu personel dari sistem informasi, meliputi manajer, analis, programmer, dan operator, serta bertanggung jawab terhadap perawatan sistem.
- Prosedur Yaitu tata cara yang meliputi strategi , kebijakan, metode, dan peraturan-peraturan dalam menggunakan sistem informasi.

2.2 UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*).

Use case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. 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.

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/property) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Activity Diagram

Activity diagrams 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.

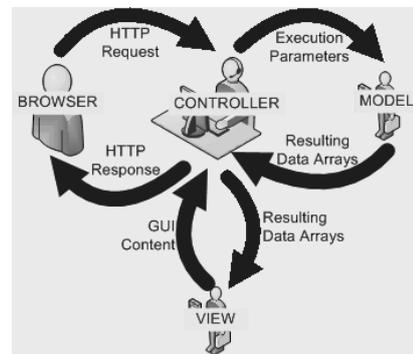
Sequence Diagram

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

2.3 MVC (*Model View Controller*)

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*.



Gambar 1 Diagram cara kerja MVC

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 memprosesnya (*Controller*), seperti yang sudah dijelaskan di atas.

III. PERANCANGAN PERANGKAT LUNAK

Perancangan sistem ini meliputi analisis kebutuhan yang kemudian diikuti pemodelan dengan menggunakan *Unified Modeling Language* (UML) untuk perancangan perangkat lunak.

3.1 Analisis Kebutuhan

Analisis kebutuhan telah dilakukan baik melalui wawancara dengan pihak-pihak terkait dan observasi dari lingkungan dimana sistem akan dibangun. Oleh karena itu penulis menyimpulkan kebutuhan rinci sebagai berikut:

- **Kebutuhan Fungsional**
 - a. Adanya fasilitas untuk menampilkan pengumuman maupun informasi riset.
 - b. Adanya fasilitas pendaftaran untuk mengikuti kegiatan riset di Undip.
 - c. Adanya fasilitas pengunggahan dokumen yang terkait dengan riset.
 - d. Adanya fasilitas untuk melihat dokumen yang terkait dengan riset.
 - e. Dokumen yang disajikan tidak dapat disalin maupun diunduh.
 - f. Adanya fasilitas forum dan chatting untuk berkomunikasi antar mahasiswa.
 - g. Adanya pembagian pengguna menjadi tiga kelompok sebagai berikut :
 - *User* Umum (tanpa *login*)
 - *User* Undip (dengan *login*) dan *User* non-Undip (dengan *login*)
 - Administrator (*admin*)
 - h. Adanya fasilitas registrasi untuk pihak dari luar Undip untuk menjadi *user* non-Undip.
 - i. *User* umum hanya dapat melihat-lihat informasi tanpa dapat berpartisipasi.
 - j. *User* Undip dapat menikmati semua fasilitas yang disediakan.
 - k. *User* non-Undip dapat menikmati fasilitas yang diberikan kecuali mengunggah dokumen.
 - l. Adanya administrator untuk *maintenence website*.
- **Kebutuhan Non-Fungsional**
 - a. Operasional
 - Dapat diakses *client* dari berbagai *browser* yang mendukung *javascript*.
 - Adanya pembedaan halaman *user* dan *admin*.
 - b. Keamanan
 - Adanya penggunaan *password* dalam *form login* untuk membedakan tipe *user* berikut hak akses masing-masing.
 - Pengguna umum hanya dapat melakukan registrasi untuk menjadi *user* non-Undip dengan hak akses yang telah ditetapkan.
 - Semua materi hanya dapat dilihat tanpa

bisa di-*download* maupun *copy-paste*.

- c. Informasi
 - Adanya informasi tata cara registrasi, *upload* materi dan peraturan forum diskusi.
 - Informasi yang diberikan selalu *up to date*.
- d. Kinerja
 - Antarmuka yang atraktif, dinamis, dan *user friendly*.
 - Batasan *upload* maksimal 10Mb per *user* dalam satu hari.

3.2 Pemodelan Perangkat Lunak

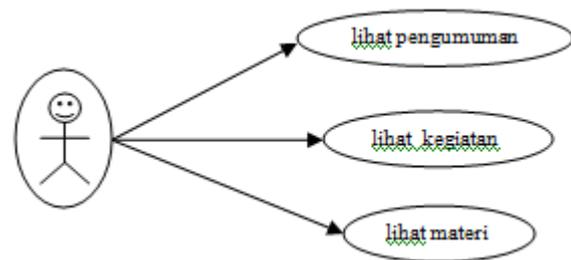
Dalam perancangan perangkat lunak ini, digunakan salah satu bahasa pemodelan yaitu UML (*Unified Modeling Language*) yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak terdiri atas *functional modeling*, *structural modeling*, dan *behavioral modeling*.

3.2.1 Functional Modeling

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

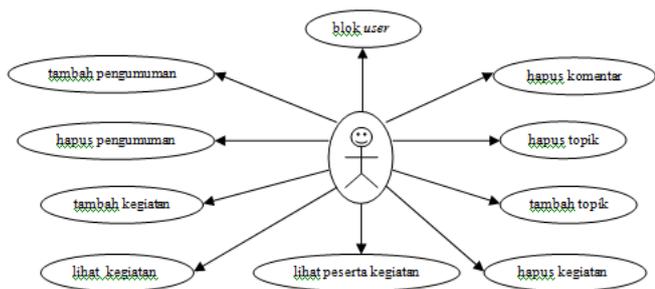
- **Use case Diagram**

Ada empat buah *use case* yang dipakai pada perancangan ini. Tiap-tiap *use case* menjelaskan apa saja yang dapat dilakukan oleh pengguna sesuai hak aksesnya. Ada empat aktor dengan hak akses berbeda seperti yang dijelaskan berikut ini :



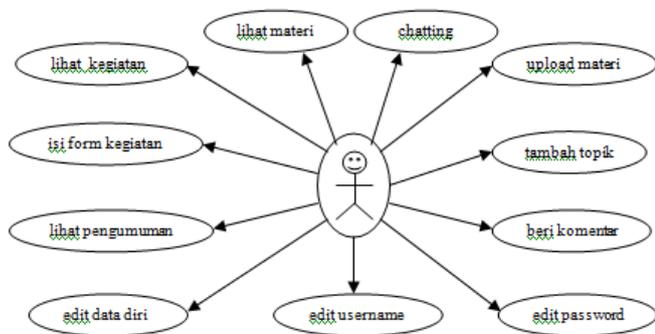
Gambar 2 *use case* user umum

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 pengumuman, melihat kegiatan, dan melihat materi yang terkait dengan riset saja.



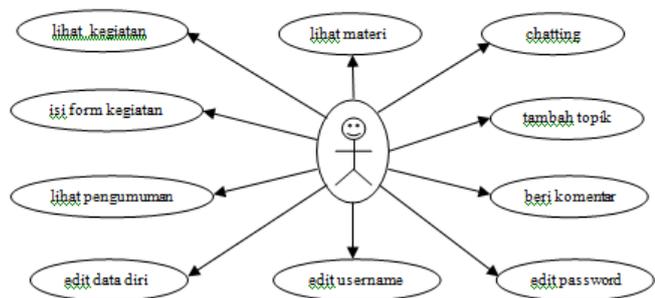
Gambar 3 use case admin

Gambar di atas adalah use case untuk admin. Admin memiliki halaman khusus untuk *maintenance website*. Admin diberi hak untuk mengatur *repository* dokumen riset, agenda kegiatan, pengumuman, forum dan berbagai hal dari sisi *back end*.



Gambar 4 use case user Undip

Gambar di atas adalah use case untuk pengguna dari Undip yang telah melakukan *login*. *User Undip* memiliki akses ke segala hal seperti melihat materi, meng-*upload* materi, berpartisipasi dalam kegiatan riset di Undip, dapat berkomunikasi dan berdiskusi dalam forum maupun *chatting*.

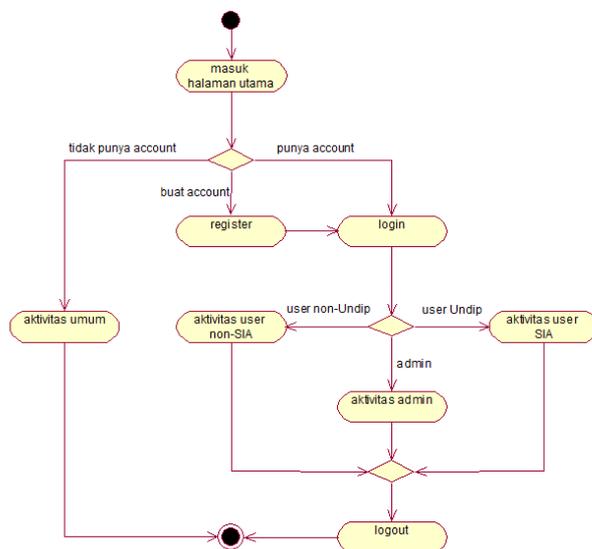


Gambar 5 use case user non-Undip

Gambar di atas adalah use case untuk *user non-Undip* yang telah melakukan *login*. *User* ini memiliki hak akses hampir sama dengan *user Undip* namun tidak diperkenankan untuk meng-*upload* materi. Pengguna umum dapat menjadi *user non-Undip* setelah melakukan registrasi dengan persyaratan tertentu yang telah ditetapkan.

• **Activity Diagram**

Dari *usecase* 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 6 activity diagram sistem secara umum

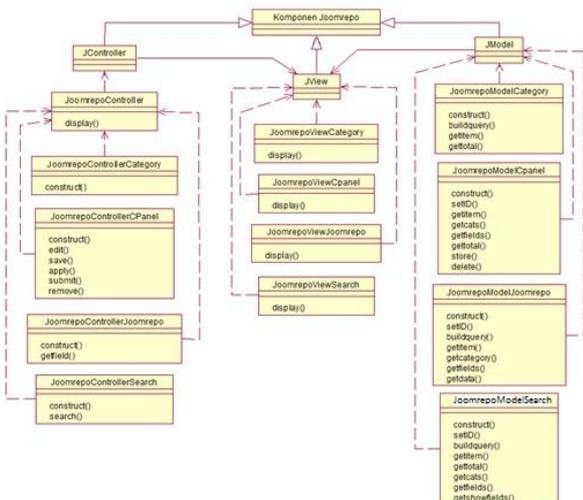
Diagram di atas merupakan *activity diagram* sistem secara umum. Terlihat adanya aliran aktivitas mulai awal masuk halaman utama hingga selesai. Pembagian aktivitas sesuai dengan *level user* seperti *user* umum, *user Undip* dan non-Undip maupun administrator.

3.2.1 **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.

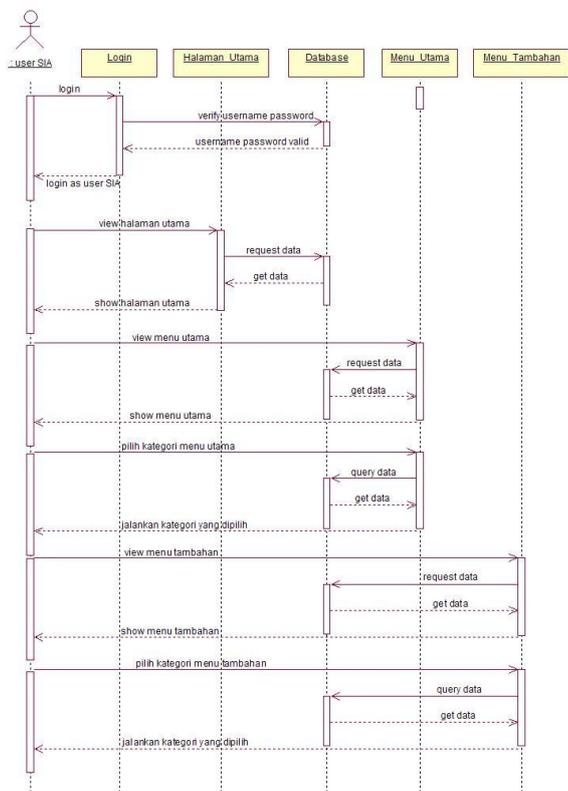
Di bawah ini akan ditunjukkan *class-class* apa yang membentuk *com_joomrepo*. Komponen inilah yang berfungsi dalam pengelolaan materi yang terkait dengan riset di Undip.



Gambar 7 diagram kelas *com_joomla*

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



Gambar 8 *Sequence diagram user Undip*

Gambar 8 merupakan contoh *sequence diagram* untuk *user Undip*. Pengguna disuguhkan halaman utama pada awal mengakses aplikasi. Setelah melakukan *login()*, akan dilakukan proses verifikasi dan validasi ke basis data. Jika sudah valid maka *login* sukses, dan *user Undip* tersebut dapat mengakses menu yang ada sesuai dengan hak yang diberikan.

3.3 Perancangan Komponen dengan Pola Model View Controller

Dalam pembuatan Sistem Informasi Pusat Riset Undip Berbasis Web akan ditanamkan beberapa komponen dalam *framework Joomla!*. Komponen tersebut diantaranya adalah :

1. Agenda

Untuk memenuhi kebutuhan ini digunakanlah *com_seminar* yang dapat menampilkan berbagai agenda riset serta menangani fasilitas untuk pendaftarannya.

2. Materi

Untuk memenuhi kebutuhan ini digunakanlah *com_joomla* yang dapat menampilkan berbagai dokumen yang terkait dengan riset berikut fasilitas untuk pengunggahan dokumennya.

3. Forum

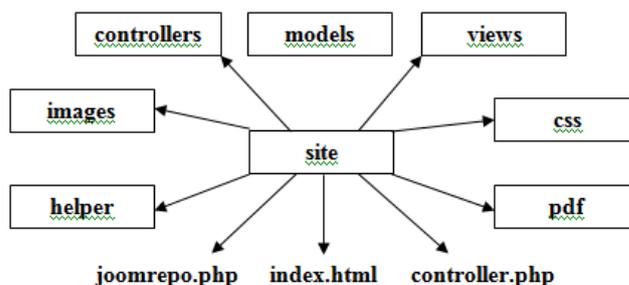
Untuk memenuhi kebutuhan ini digunakanlah *com_ccboard* yang khusus menangani hal-hal yang terkait dengan forum.

4. Chatting

Untuk memenuhi kebutuhan ini digunakanlah *com_obrolan* yang berfungsi sebagai media bercakap-cakap antar *user* yang mengakses sistem informasi tersebut.

Pada sub bab berikutnya akan dijelaskan lebih rinci akan penanaman komponen pada *joomla!* yang menggunakan pola MVC (*Model View Controller*). Penulis memberikan contoh penerapan pola MVC dalam pembuatan komponen *com_joomla* dan *com_obrolan* yang terbagi pada bagian *front end* dan *back end*.

• Perancangan pada front end

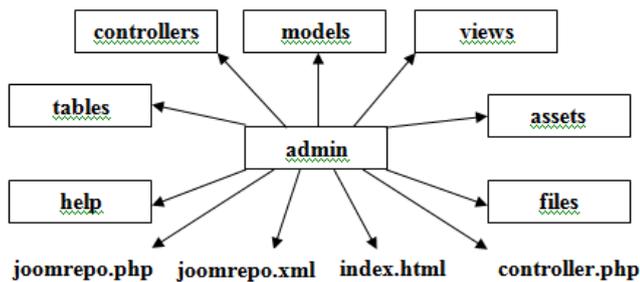


Gambar 9 Pola *front end com_joomla*

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 *joomrepo.php* untuk pertama kalinya. *Joomrepo.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 10 Pola back end *com_joomrepo*

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 *joomrepo.php* untuk pertama kalinya. *File joomrepo.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 Pusat Riset 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 sitem 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_joomrepo*

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

```

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

```

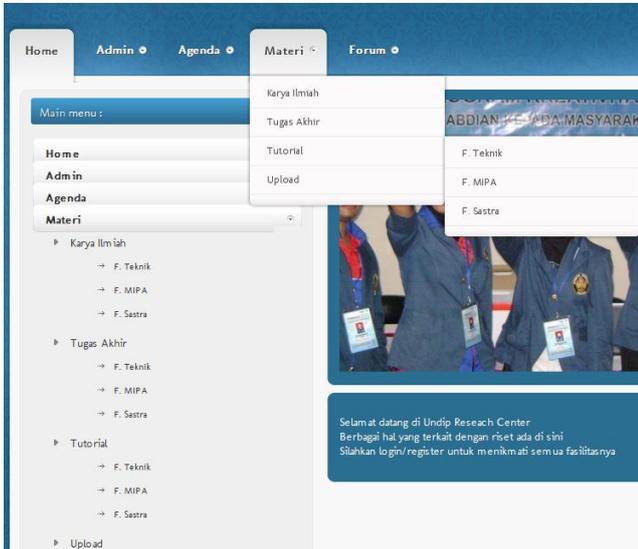
/* instalasi modul latest entry untuk
menampilkan berkas yang paling akhir
diunggah */
if(jfolder::move(JPATH_ADMINISTRATOR.D
S.'components'.DS.'com_joomrepo'.DS.'f
iles'.DS.'mod_latest_entry',
JPATH_SITE.DS.'modules'.DS.'mod_latest
_entry')) {
else {
echo 'Latest Entry modules could not
be installed. Please move the folder
of Latest entry module from
''.JPATH_ADMINISTRATOR.DS.'components'.
.DS.'com_joomrepo'.DS.'files'.DS.'mod_
latest_entry" to
''.JPATH_SITE.DS.'modules'.DS.'mod_lat
est_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 Materi



Gambar 11 tampilan menu Materi

Menu Materi di atas adalah hasil dari `com_joomrepo` yang telah dibuat. Terdapat dua cara untuk mengakses menu materi, yaitu melalui top menu dan drop down menu di bagian kiri atas.

Penulis mencontohkan saat mengakses menu Materi >> Tutorial >> F. Teknik sehingga didapat tampilan seperti di bawah ini.



Gambar 12 daftar Materi Tutorial F. Teknik

Untuk melihat isi dokumen dapat dilakukan dengan meng-klik *link view* pada kolom enam. Berikut ini tampilan saat dokumen ditampilkan.



Gambar 13 tampilan salah satu dokumen

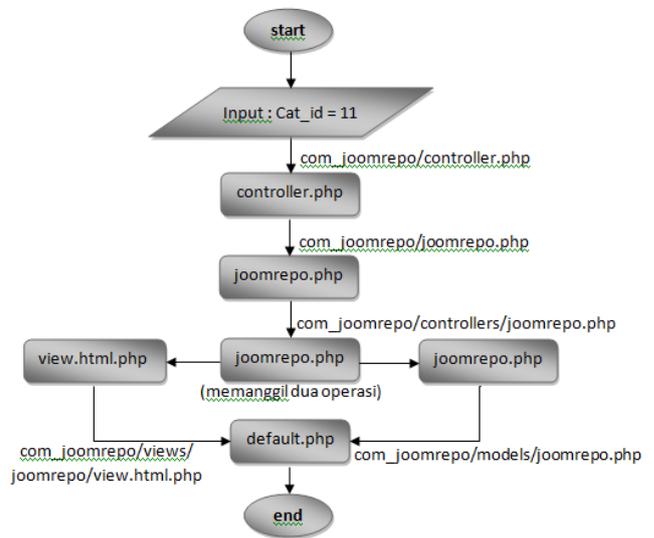
Gambar di atas adalah tampilan salah satu dokumen tutorial Fakultas Teknik yang terkoneksi dengan *googledoc*.

4.2.2 Pengujian White Box

Pada pengujian white box ini dicontohkan dengan menu **Materi** dengan alamat *URL* :

http://localhost/URC/index.php?option=com_joomrepo&view=cpanel&layout=form

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



Gambar 13 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 `joomrepo.php`. oleh file `joomrepo.php` tersebut kemudian diteruskan ke file `cpanel.php` yang ada pada folder `com_joomrepo/controller`. Oleh file `cpanel.php` tersebut dipanggil dua operasi yang dilakukan oleh file `view.html.php` di dalam folder `com_joomrepo/views/cpanel` dan file `cpanel.php` pada folder `com_joomrepo/models`. Setelah operasi tersebut selesai maka didapatkan keluarannya yaitu `form.php`.

IV. PENUTUP

5.1 Kesimpulan

- 1) Telah dirancang dan dibangun sebuah Sistem Informasi Pusat RisetUndip (*Undip Research Center*) dengan menggunakan *framework* Joomla!, 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 menampilkan data-data yang terkait dengan dunia riset 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 memroses dengan baik semua alur logika program.

5.2 Saran

- 1) Aplikasi Sistem Informasi Pusat Riset Undip (*Undip Research Center*) ini dapat diterapkan secara online pada jaringan internet dan terintegrasi dengan Sistem Informasi Akademik (SIA) maupun Kuliah Online (Kulon) agar dapat diakses kapanpun dan di mana pun.
- 2) Dapat dikembangkan aplikasi Sistem Informasi Pusat Riset Undip (*Undip Research Center*) dengan menggunakan bahasa pemrograman selain PHP dan mesin basis data selain MySQL agar dapat dibandingkan kinerjanya.

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

BIODATA PENULIS



Kumboro Bimantyo Kurnia tama, lahir di Semarang tanggal 2 Agustus 1987. Menempuh pendidikan dasar di SD Negeri VI Ungaran. Melanjutkan ke SMPN 1 Ungaran, kemudian menempuh pendidikan tingkat atas di SMUN 4 Semarang lulus tahun 2005. Dari tahun 2005 sampai saat

ini masih menyelesaikan studi Strata-1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang, konsentrasi Komputer dan Informatika.

Menyetujui,
Dosen Pembimbing I

DAFTAR PUSTAKA

- [1] Ladjamudin, Al Bahra Bin. 2007. *Analisis dan Desain Sistem Informasi*. Yogyakarta : Penerbit Graha Ilmu
- [2] Al Fatta, Hanif. 2007. *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta : CV. Andi Offset
- [3] Kadir, Abdul. 2008. *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta : Penerbit Andi
- [4] Kadir, Abdul. 2009. *Mastering AJAX dan PHP*. Yogyakarta : CV. Andi Offset

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

Dosen Pembimbing II

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