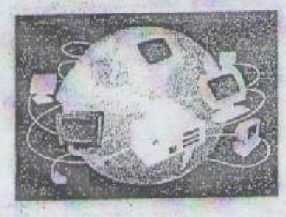
BADAN PENERBIT UNIVERSITAS DIPONEGORO

# PROSIDING

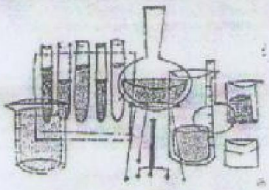
## Seminar Internasional Hasil-hasil Penelitan Tahun 2009

Purwokerto, 3 Mei 2009



## KLASTER EKSAKTA

1-2



Tema :
Pembangunan berkelanjutan berbasis Ilmu Pengetahuan, Teknologi dan Seni (Ipteks)

KLASTER
PENDIDIKAN
SOSIAL DAN HUMANIORA
EKSAKTA

Penyunting :
Eko Sri Israhayu, Dini Siswani Mulia, Tumisem,
Edi Joko Setyadi, Ratna Kartikawati, Hengky Widhiandono,
Ragil Setiyabudi, Hindayati Mustafidah, Dumasari, Kusno,
Sony Irianto, Sugeng Priyadi, Suwarno

Lembaga Penelitian dan Pengabdian pada Masyarakat
Universitas Muhammadiyah Purwokerto
Jl. Raya Dukuh Waluh PO. Box. 202 Purwokerto 53182
Telp. (0281) 636751, 630463 psw. 231-232
Fax. (0281) 637239, email : lppmump@yahoo.com

# PENGEMBANGAN PERANGKAT LUNAK SISTEM UJIAN ON-L' BERBASIS KONSEP ARSITEKTUR MODEL, VIEW DAN CONTROLE

**Ermatita, Huda Ubaya, Dwi Rosa Indah**
**Fakultas Ilmu Komputer, Universitas Sriwijaya**

**Abstract**

Online examination system is a system service in the process of the test subjects in the lectures through the website, which is used by students and lecturers for the evaluation of the course materials that have been conducted for the software online exam system is web based and GUI, so it requires the handling of powerful to handle the events done by the user. Handling System of the events done by the user can be in the layers. Layers will have a different responsibility in accordance with their respective domain. Architecture that is suitable for this Model-View-Controller (MVC), where the development of software in a Tier Model for the MVC pattern that represents the data used by applications as business processes associated it. With divide as separate parts, such as gathering data, persistence, and manipulation of the process, separate from other parts of the application. View layer that interacts directly with the user, controller responsible for gathering events made by the user from the View and update component of the model using data entered by the user. Development of software with the online exam system to apply the concept of MVC architecture will facilitate the development of software for evolution.

Keywords : *Software, online examination system, GUI, web, Model-View-Controller*

**Pendahuluan**

Fakultas Ilmu Komputer Universitas Sriwijaya merupakan institusi yang bergerak didalam bidang pendidikan. Sebagai universitas yang selalu berkembang mengikuti perkembangan teknologi, Universitas Sriwijaya selalu berusaha memberikan pelayanan yang terbaik kepada mahasiswa. Kemajuan teknologi informasi telah membawa dampak kepada proses belajar mengajar dalam dunia pendidikan demikian juga dengan Universitas Sriwijaya. Dengan bantuan teknologi informasi semua aktifitas dalam proses belajar mengajar tidak lagi terbatas pada tempat, ruang dan waktu. Kegiatan Ujian sebagai salah satu aktivitas dalam belajar mengajar dapat dilakukan di mana saja dengan bantuan teknologi informasi. Kegiatan ini disebut dengan ujian *online*.

Sistem ujian *online* adalah suatu sistem pelayanan dalam proses ujian mata kuliah dalam perkuliahan secara *online* melalui website, yang digunakan oleh mahasiswa dan dosen untuk proses evaluasi terhadap materi kuliah yang telah dilaksanakan.Sistem Ujian *Online* yang diterapkan oleh Fakultas Ilmu Komputer Universitas Sriwijaya harus memberikan kemudahan bagi para mahasiswa dan dosen, .yang tidak terbatas oleh tempat dan waktu.

Pada Sistem Ujian *Online* ini akan dikembangkan perangkat lunak sistem yang berbasis Web dan GUI. Pengembangan Perangkat lunak ini merupakan tugas yang kompleks. Proses pengembangan Perangkat Lunak, mulai dari konsep sampai dengan implementasinya dikenal dengan istilah *System Development Life Cycle* (SDLC) yang meliputi tahapan-tahapan berupa analisis kebutuhan, desain, generasi kode, Implementasi, pengujian dan pemeliharaan (Pressman,2002;37-38). Selain itu, dalam pengembangan Perangkat Lunak perlu digunakan suatu metodologi yang dapat digunakan sebagai pedoman bagaimana dan apa yang harus dikerjakan selama pengembangan. Metodologi yang dapat digunakan terdiri dari metodologi yang berorientasi pada keluaran, proses (biasa digunakan), data dan objek (Sutabri, 2004: 68-69). Metodologi ini dapat menjadi acuan dalam melakukan pengembangan perangkat lunak. Dalam pengembangan perangkat lunak juga membutuhkan arsitektur yang dapat membantu dalam pengembangan perangkat lunak

Dalam kegiatan ujian yang dilaksanakan secara on-line , kebutuhan akan banyaknya variasi soal dan persediaan soal sangat penting . Efektivitas dan tingkat keamanan yang baik dibutuhkan dalam mendokumentasikan soal-soal ujian tersebut Selain itu juga dibutuhkan penampilan yang baik dan handal dalam menampilkan soal-soal ujian. Untuk itu dibutuhkan suatu konsep pengembangan dan arsitektur yang baik dalam pengembangan perangkat lunak pendukung ujian on-line ini.

Untuk pengembangan perangkat lunak pada sistem ujian Online ini akan diterapkan konsep dengan arsitektur *Model View Controler*. Konsep pengembangan perangkat lunak dengan arsitektur model, view dan controller ini, sangat membantu dalam pengembangan perangkat lunak ujian on-line. Hal ini dikarenakan pengembangan perangkat lunak dengan konsep model view dan controller ini dapat membantu membuat tampilan yang menarik dan membuat interaksi user dengan perangkat lunak lebih

menarik dan lebih mudah. Karena di rancang dengan mengutamakan user interface, untuk [illegible] interaksi antara user dan perangkat lunak.

Dengan konsep arsitektur ini pengembangan Perangkat Lunak pada Sistem Ujian *Online* yang berbasis user interface akan lebih mudah dikembangkan lebih lanjut. Arsitektur ini sangat cocok untuk pengembangan perangkat lunak yang berbasis interaksi dengan GUI.

Perangkat Lunak Sistem Ujian *Online* ini diharapkan akan meningkatkan kualitas pada proses pembelajaran pada fakultas Ilmu Komputer Universitas Sriwijaya O.leh karena itu, perlu dadakan suatu penelitian yang dapat mengembangkan Perangkat lunak pada Sistem Ujian *Online* dengan menerapkan konsep arsitektur Model View Controller.

Penelitian ini difokuskan pada Konsep Pengembangan Perangkat lunak yang berbasis GUI dengan menerapkan pola Model-view-Controller Pola ini akan diterapkan pada pengembangan perangkat lunak Ujian On-line pada Fakultas Ilmu Komputer Universitas Sriwijaya berkaitan dengan pertimbangan konsep tersebut dalam hal kualitas, fleksibilitas serta kemudahan perawatan (perluasan atau peningkatan) seperti yang dikemukakan oleh banyak literatur.

Dari latar belakang yang telah dijelaskan di atas maka akan dilakukan penelitian dengan judul "Konsep Model, View dan Controler (MVC) Penerapannya pada Pemecahan pengembangan Perangkat Lunak ujian on-line"

Penelitian ini bertujuan:

1. Melakukan studi terhadap pendekatan MVC
2. Mengembangkan perangkat lunak ujian on-line dengan konsep MVC agar dapat digunakan dalam proses belajar mengajar terutama untuk ujian-ujian tertentu pada Fakultas Ilmu Komputer Universitas Sriwijaya.

**Metode Penelitian**

Pada penelitian ini digunakan metode Unified (Fowler, 2004). Langkah-langkah yang akan dilakukan dalam penelitian ini adalah :

1. Spesifikasi kebutuhan pemakai sistem dalam merancang model, view dan controller
2. Mengidentifikasi skenario pemakaian atau *use-case*
3. Membuat Class diagram.
4. Membuat Activity Diagram
5. Membuat Sequence diagram
6. Mengimplementasikan objek kedalam perangkat lunak ujian on-line

**1 Sistem**

Definisi sistem menurut O'Brien (2003, p18) adalah sekumpulan komponen-komponen terkait yang bekerja sama untuk mencapai suatu tujuan dengan menerima masukan dan menghasilkan keluaran dalam suatu proses transformasi yang terorganisir.

Sistem mempunyai tiga komponen dasar yang saling berinteraksi, yaitu :

1. *Input* (masukan), meliputi menangkap dan mengumpulkan elemen yang memasuki sistem untuk dapat diproses. *Input* yang dimaksud yaitu melalui sistem informasi KRS *Online* mahasiswa dapat memasukkan peminatan, matakuliah dan jadwal kuliah yang dipilih dengan mudah.
2. *Processing* (proses), meliputi proses perubahan yang merubah *input* menjadi *output*. Proses yang dimaksud adalah mahasiswa tidak mengalami gangguan dalam melakukan registrasi dan dalam merevisi hasil registrasi perkuliahan melalui sistem informasi KRS *Online*.
3. *Output* (keluaran atau hasil), meliputi pengiriman elemen yang telah dihasilkan dalam proses perubahan ke dalam tujuan akhirnya. *Output* yang dimaksud adalah mahasiswa dapat mengetahui hasil registrasi peminatan, mata kuliah dan jadwal kuliah melalui sistem informasi KRS *Online*.

Jadi dapat disimpulkan bahwa sistem adalah sekumpulan komponen-komponen terkait yang bekerja sama untuk mencapai suatu tujuan dengan menerima masukan (*input*) dan menghasilkan keluaran (*Output*) dalam suatu proses (*Process*) transformasi yang terorganisir.

**2 Pengertian Unified Modelling Language (UML)**

Menurut Jones dan Rama (2006, p60), *UML is a language used for specifying, visualizing, constructing, and documenting an information system.*

Artinya *UML* adalah suatu bahasa yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi

***Pengertian Use Case Diagram***

Menurut Jones dan Rama (2006, p267), *"Use case diagram is a list of use cases that occur in an application and that indicate the actor responsible for each use case"*.

Artinya *use case diagram* adalah daftar dari use case yang terjadi dalam suatu aplikasi dan itu menandakan bahwa aktor tersebut bertanggungjawab untuk masing-masing *use case*.

***Pengertian Class Diagram***

Menurut Jones dan Rama (2006, p158), *"UML class diagram is a diagram that can be used to document (a) tables in an Accounting Information System, (b) relationships between tables, and (c) attributes of tables"*.

Artinya *UML class diagram* adalah suatu diagram yang dapat digunakan untuk dokumen (a) tabel dalam Sistem Informasi Akuntansi, (b) hubungan antar tabel, dan (c) atribut dari tabel.

***Pengertian Activity Diagram***

Menurut Jones dan Rama (2006, p60), *"UML activity diagram is a diagram that shows the sequence of activities in a process"*.

Artinya *UML activity diagram* adalah suatu diagram yang menunjukkan langkah-langkah aktivitas dalam sebuah proses.

**3 Model-View-Controller Architecture**

Model View and Controller adalah salah satu arsitektur yang dapat dipergunakan dalam pengembangan perangkat lunak. Arsitektur ini sangat cocok untuk pengembangan perangkat lunak yang berbasis interaksi dengan GUI. Hal ini sesuai dengan pernyataan Ballangan;2007 : *"MVC pattern is one of common architecture especially in the development of rich user interactions GUI Aplication. Its main idea is to decouple the model. The interaction between the view and the model is managed by the controller"*.

Selain itu pernyataan di bawah ini juga menyatakan hal yang sama dengan di atas;

> *The Model-View-Controller architecture is a widely-used architectural approach for interactive applications. It divides functionality among objects involved in maintaining and presenting data to minimize the degree of coupling between the objects. The architecture maps traditional application tasks--input, processing, and output--to the graphical user interaction model. They also map into the domain of multitier Web-based enterprise applications.*

**4 Ujian**

Ujian adalah salah satu aktivitas dalam proses belajar mengajar. Ujian dilakukan untuk mengevaluasi hasil belajar kepada peserta didik. Untuk melakukan Ujian dibutuhkan soal-soal yang yang dapat mengukur hasil belajar peserta didik. Soal-soal ini harus dikerjakan oleh peserta didik yang dapat mengukur kemampuan tentang materi yang telah diberikan.

***5 Online***

Menurut O'Brien (2003, p226), *online* adalah sistem pemrosesan transaksi, dimana data diproses dengan segera setelah sebuah transaksi terjadi. Segera yang dimaksud adalah tidak memakan waktu yang lama.

Dengan melakukan registrasi perkuliahan secara *online* maka proses pemilihan peminatan, matakuliah dan jadwal kuliah melalui sistem informasi KRS *Online* dapat dilakukan dengan segera.

Menurut http://www.mercubuana.ac.id/sistem.php, *online* adalah konsep penggunaan data yang selalu dapat diakses dari manapun dan di manapun.

Dengan sistem Ujian yang dilakukan secara *online* maka mahasiswa dapat melaksanakan ujian *Online* dimanapun dan kapanpun.

Jadi dapat disimpulkan bahwa *online* adalah sistem pemrosesan transaksi yang dapat dilakukan dimana saja dan kapan saja dengan menggunakan komputer dimana data diproses dengan segera setelah sebuah teransaksi terjadi.

**6 Ujian *Online***

Ujian yang dilaksanakan secara jarak jauh, tanpa memerlukan kehadiran fisik peserta ujian dan penguji. Jadi tidak memerlukan tatap muka langsung antara penguji dan peserta uji namun tetap dapat berkomunikasi melalui suatu perangkat lunak yang didesain untuk melakukan ujian online.

Arsitektur Model-View-Controller adalah sebuah pola yang terbukti membangun proyek secara lebih efektif. Hal itu dilakukan dengan memilah komponen antara Model, View dan Controller pada bagian – bagian dalam proyek.

Selain itu kode program yang telah ditulis bisa dimanfaatkan lagi untuk aplikasi lain dengan menggubah sebagian saja. Pada dasarnya MVC yakni untuk memisahkan pemroses data yang disebut sebagai *Model* dan penyaji antarmuka (*UI*) atau *HTML template* dalam hal ini disebut View. Sedangkan *Controller* sebagai pengikat Model â€" View untuk menangani permintaan pengguna sehingga data ditampilkan dengan benar.

Perangkat lunak/sistem yang dikembangkan adalah untuk kebutuhan pengguna, harus mengetahui apa sesungguhnya dibutuhkan pengguna. Ada beberapa upaya yang dapat dilakukan, yaitu :

a. Wawancara. Teknik ini adalah teknik yang dapat digunakan untuk mengidentifikasi kebutuhan pengguna.
b. Mempelajari sistem yang sedang berjalan. Dengan cara ; menelaah dokumen yang ada, mengenali struktur organisasi perusahaan, melihat deskripsi kerja setiap personal dalam struktur organisasi.
c. Membuat daftar pertanyaan serta kuisioner
d. Pengamatan langsung ke lapangan.

Model-View-Controller membantu merancang perangkat lunak dengan pemilahan kebutuhan yang sejenis. Model View dan Controller adalah membawa kelas-kelas dari tahapan analisis ke tahap perancangan. (Nugroho,2002: 188)

Beberapa penelitian dalam bidang MVC telah dilakukan antara lain :

Sauer dalam penelitiannya yang *berjudul "MVC-Based Modeling Support for Embedded Real-Time Systems" menyatakan bahwa "Effects of real-time requirements on model, view,controller and communication components* telah dapat diidentifikasi. Efek kebutuhan secara real time pada model view controller dapat di identifikasi sehingga dapat di tangani dengan cepat .

Lebih lanjut Didtante,dkk dalam penelitiannya yang *berjudul Model-Driven Development of Web Applications with UWA, MVC and JavaServer Faces menyatakan "The proposed approach, by defining the UML-MVC logical model, makes it possible to establish a correspondence between di UWA conceptual design of a web application and its implementation.* Pemeliharaan dan evolusi pada perangkat lunak akan menjadi lebih mudah dengan menggunakan arsitektur MVC. Selain itu perangkat lunak akan mudah ditelusuri ketika terjadi kesalahan..

Pengembangan dengan pendekatan MVC, pemeliharaan dan evolusi terhadap sistem menjadi lebih mudah dilakukan, dengan adanya pemilahan pada lapisan-lapisan dalamdengan adanya pemilahan pada lapisan-lapisan dalam perangkat lunak.

Ogata dalam penelitiannya *"An event Driven MVC based on active object"* mengungkapkan bahwa *"The MVC (Model-view-Controller) frame work [3] offers a basic function for building a GUI ( Graphical User Interface) in smalltalk[2]. It threats input evens by using a polling mechanism. As only a single process is assigned to a single window make in Smaltalk, Smaltalk does not make the best use of the advantages of a multiwindow which it adopts*

*An interface process –a process assigned to window interacting with a user-checks the device table one by one in order to know whether an expected event has occurred. Checking a device state table is done by executing .*

Ogata dalam makalah ini menggambarkan bahwa design pada kerangka Model-View-Controller disebut Event-driven-Active MVC yang berdasar padaobjek yang aktif. Pada model ini memperlakukan setiap input dengan menggunakan mekanisme Event-driven dan menempatkan proses dari objek yang aktif pada beberapa tampilan. Model ini sangat cocok digunakan untuk aplikasi yang focus pada GUI. Model ini telah di imlementasikan pada Smalltalk.

**Analisis Perangkat Lunak Ujian Online Dengan Konsep Model Mvc**

Arsitektur *Model-View-Controller* merupakan pola yang dapat membantu membangun proyek secara lebih efektif. Pada pola ini pengembangan dilakukan dengan memilah komponen antara *Model, View* dan *Controller* pada bagian – bagian dalam pengembangan perangkat lunak. Pemilahan ini berguna untuk memisahkan bagian-bagian dalam aplikasi sehingga mempermudah dalam pengembangannya dan pemeliharaan aplikasi.

Pada sebuah sistem yang dilengkapi dengan perangkat lunak, bagian yang sering mengalami perubahan adalah bagian *user interface. User interface* adalah bagian yang berhadapan langsung dengan user dan bagaimana ia berinteraksi dengan aplikasi, membuatnyamenjadi titik fokus pengubahan berdasar kemudahan penggunaan.

*Business-logic* yang rumit pada user-interface membuat pengubahan pada user interface menjadi lebih kompleks dan mudah terjadi kesalahan. Perubahan pada satu bagian memiliki potensi keterkaitan dengan keseluruhan perangkat lunak.

MVC dapat memberikan solusi terhadap permasalahan tersebut dengan membagi pola menjadi bagian – bagian tersendiri, *Model, View* dan *Controller*, memisahkan antar bagian tersebut dan membuat tata interaksi diantaranya.

**1. Model**

Lapisan Model pada pola MVC yang merepresentasikan data yang digunakan oleh aplikasi sebagaimana proses bisnis yang diasosiasikan terhadapnya. Dengan memilahnya sebagai bagian terpisah, seperti penampungan data, persistence, serta proses manipulasi, terpisah dari bagian lain aplikasi.

Terdapat beberapa kelebihan dalam pendekatan ini. Pertama, membuat detail dari data dan operasinya dapat ditempatkan pada area yang ditentukan (Model) dibanding tersebar dalam keseluruhan lingkup aplikasi. Hal ini memberikan keuntungan dalam proses maintenance aplikasi.

Kedua, dengan pemisahan total antara data dengan implementasi interface, komponen model dapat digunakan kembali oleh aplikasi lain yang memiliki kegunaan yang hampir sama.

Pada perangkat lunak ujian Online ini data data yang ada akan ditangani oleh lapisan model yang di buat dengan menggunakan PHP 5. layer ini akan memberikan solusi perminaan aksi dari controller. Dalam Model ini database di buat dengan menggunakan mySQL yang dapat memberikan data yang dibutuhkan oleh Model.

**2. View**

Layer ini mengandung keseluruhan detail dari implementasi user interface. Disini, komponen grafis menyediakan representasi proses internal aplikasi dan menuntun alur interaksi user terhadap aplikasi. Tidak ada layer lain yang berinteraksi dengan user, hanya View.

Penggunaan layer View memiliki beberapa kelebihan : Pertama, memudahkan pengabungan divisi desain dalam pengembangan team. Divisi desain dapat berkonsentrasi pada style, look & feel, dan sebagainya, dalam aplikasi tanpa harus memperhatikan lebih pada detail yang lain.

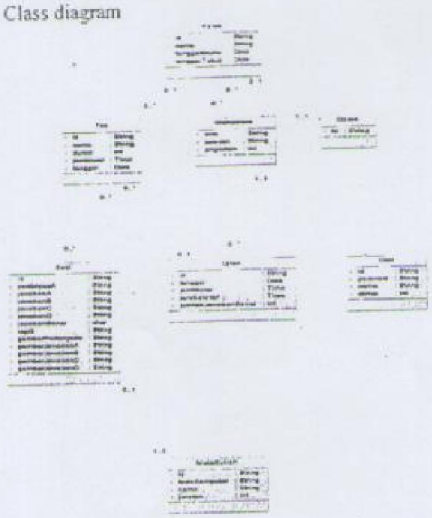
Dan juga, memiliki layer View yang terpisah memungkinkan ketersediaan multiple interface dalam aplikasi. Jika inti dari aplikasi terletak pada bagian lain (dalam Model), multiple interfaces dapat dibuat (Swing, Web, Console), secara keseluruhan memiliki tampilan yang berbeda namun mengeksekusi komponen Model sesuai fungsionalitas yang diharapkan.

Lapisan View dalam perangakat lunak sistem ujian Online ini menggunakan dreamweaper untuk tampilan pada user. Lapisan view ini yang akan berinteraksi dengan controller.

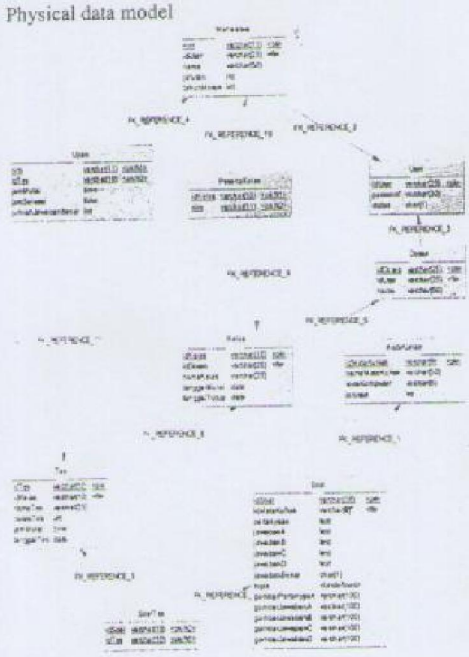
Pada perangkat lunak Sistem Ujian Online yang sedang dikembangkan, lapisan view terdiri dari file-file yang berisi tampilan bagi user yang sebagian besar terdiri dari skrip HTML.

| | |
|---|---|
| | dosen.. halaman kelas, halaman ujian dan halaman hasil |

Class diagram



Physical data model



1. **Lapisan Model**

Lapisan model pada pola MVC yang merepresentasikan data yang digunakan oleh aplikasi sebagaimana proses bisnis yang diasosiasikan terhadapnya.

| | Deskripsi |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |

**. Controller**

Lapisan Controller pada MVC menyediakan detail alur program dan transisi setiap lapisan, ertanggungjawab akan penampungan events yang dibuat oleh user dari View dan melakukan update rhadap komponen Model menggunakan data yang dimasukkan oleh user.

Penggunaan lapisan Controller secara terpisah mempunyai keuntungan:

. Pengembangan dengan menggunakan komponen terpisah dapat menampung detail dari transisi lapisan, omponen view dapat didesain tanpa harus memperhatikan bagian lain secara berlebih. Hal ini emudahkan team pengembang multiple interface bekerja secara terpisah dari yang lain secara simultan. teraksi antar komponen View terabstraksi dalam Controller.

. Dapat melakukan update terhadap komponen Model, detail tersebut dihapus dari lapisan view. Lapisan iew kembali pada fungsi utamanya untuk menampilkan data kepada user. Detail tentang bagaimana data ari user mengubah ketetapan aplikasi disembunyikan oleh Controller. Hal ini memisahkan dengan jelas tara *presentation logic dengan business logic*.

Perangkat lunak Sistem Ujian Online ini sangat cocok dikembangkan dengan konsep arsitektur *odel-View-Controller* hal ini dikarenakan system ujian online ini merupakan suatu sistem yang embutuhkan perangkat lunak yang berbasis GUI dan dapat berinteraksi dengan pengguna. Dengan stem Model –View-Controller ini perangkat lunak dapat menangani event-event yang dilakukan oleh er dengan baik, karena setiap aksi yang dilakukan oleh user akan di tangani oleh lapisan-lapisan yang ah mempunyai tanggung jawab masing-masing.

View atau tampilan adalah lapisan yang menangani setiap even/permintaan yang dilakukan oleh er, kemudian bagian view akan meneruskan ke bagian controller untuk menangani dan memverifikasi si atau permintaan user. Setelah itu Controller akan meminta kepada Model untuk mengecek data atau emberikan data dari bagian database.

**sil Perancangan Sistem**

***nctional Requirement***

*uirement* utama ada 3 menu:

**Proses Login**

**bel 1. Requirement Proses Login**

| | |
|---|---|
| | *User* memasukkan data :<br>1. *username*<br>2. *password* |
| | Sistem Membaca *username* dan *password* apakah sesuai dengan *database* atau tidak (verfikasi dan validasi) |
| | Halaman berikutnya yang tampil jika *username* dan *Password* benar adalah halaman utama |
| | Jika *username* dan *password* salah maka tetap pada halaman login |

R2 Halaman Utama

Tabel 2. Requirement Halaman Utama

| | |
|---|---|
| *Input* | *User* memilih menu pilihan<br>• Dosen<br>• Mahasiswa |
| Proses | Sistem akan menampilkan halaman dosen jika *user* memilih menu dosen. *User* dapat malakukan *input*, *edit* dan hapus data mata kuliah. |
| Output | Halaman berikutnya yang tampil jika *user* memilih data dosen adalah halaman dosen |
| Catatan | Jika *user* memilih menu dosen maka halaman berikutnya yang tampil adalah halaman dosen dan prosesnya pun meliputi proses data dosen. Begitu juga dengan menu jadwal dan report. |

R3 Menu Dosen

Tabel 3. Requirement Menu Mata kuliah

| | |
|---|---|
| *Input* | *User* memilih menu mengelola kelas yang tersedia pada menu utama |
| Proses | Sistem dapat menerima *input* kelas baru, meng*edit* dan mengapus data kelas |
| Output | Halaman berikutnya yang tampil jika *user* memilih data mengelola kelas adalah halaman membuat kelas baru |
| Catatan | Dalam halaman ini, *user* dapat melakukan *entry* data, *edit* data, dan hapus data kelas sesuai dengan kebutuhan. |

R4 Menu Mahasiswa

Tabel 4. Requirement Menu Mahasiswa

| | |
|---|---|
| *Input* | *User* memilih menu Mahasiswa |
| Proses | Sistem dapat menerima *input* data mahasiswa, meng*edit* dan menhapus data mahasiswa. |
| Output | Halaman berikutnya yang tampil jika *user* memilih data mahasiswa adalah halaman ujian |
| Catatan | Dalam halaman ini, *user* dapat melihat data kelas dan |