



Sistem Pengajuan Judul Tugas Akhir Di Universitas Ekasakti Arsitektur Model View Controller

Harry Setya Hadi¹, Wahyuni Yahyan²

^{1,2}D3 Manajemen Informatika , Fakultas Ekonomi, Universitas Ekasakti
xmoensen@gmail.com

Abstract

The final project is a requirement for a student to get a bachelor's degree at any State Higher Education (PTN) or Private Higher Education (PTS) in Indonesia. In the preparation of the final project, the initial stage that must be done is submitting a title. Submission of the title of the final project at Ekasakti University, especially in the D3 Informatics and Computer Management study program, still uses a manual process. The technology that supports this title submission system uses the web as the basis and the architecture uses the Model View Controller, with web technology accessible to all people on campus, especially students, because the system is accessed online. The implementation of the Model View Controller which cannot process data (models) with a user interface design (view) which then controls the data that depends on the architecture (Controller) which facilitates system development by programmers. The results to be achieved in making this application system is the creation of a final assignment submission system.

Keywords: Submission of Title, MVC, Information System, DFD, Website

Abstrak

Tugas akhir menjadi persyaratan bagi seorang mahasiswa/i untuk mendapatkan gelar sarjana di setiap Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta (PTS) yang ada di Indonesia. Pada penyusunan tugas akhir, tahap awal yang harus dilakukan adalah pengajuan judul. Pengajuan judul tugas akhir di Universitas Ekasakti terutama pada prodi D3 Manajemen Informatika dan Komputer masih menggunakan proses manual. Teknologi yang mendukung dalam sistem pengajuan judul ini menggunakan web sebagai basisnya dan arsitektur menggunakan Model View Controller, dengan teknologi web dapat diakses oleh semua kalangan yang ada di kampus terutama mahasiswa, dosen karena sistem diakses secara online. Penerapan arsitektur Model View Controller yang memisahkan pengolahan data (model) dengan desain interface pengguna (view) yang kemudian mengontrol data yang diproses untuk ditampilkan (Controller) yang memudahkan dalam pembangunan sistem oleh programmer. Hasil yang akan dicapai pada pembuatan sistem aplikasi ini adalah terciptanya sebuah sistem pengajuan tugas akhir. Kata kunci: Pengajuan Judul, MVC, Sistem Informasi, DFD, Website

1. Pendahuluan

Universitas Ekasakti atau disingkat UNES adalah salah satu universitas swasta di Kota Padang. Universitas Ekasakti mewajibkan mahasiswanya untuk menyusun tugas akhir sebagai syarat kelulusan.

Tugas akhir menjadi persyaratan bagi seorang mahasiswa/i untuk mendapatkan gelar sarjana di setiap Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta (PTS) yang ada di Indonesia. Pada penyusunan tugas akhir, tahap awal yang harus dilakukan adalah pengajuan judul. Pengajuan judul tugas akhir di Universitas Ekasakti terutama pada prodi D3 Manajemen Informatika dan Komputer masih menggunakan proses manual.

Namun di Universitas Ekasakti masih belum efektif dan efisien dalam memberikan pelayanan kepada mahasiswa tingkat akhir terutama dalam pengajuan tugas akhir yang mana masih menggunakan cara manual. Mahasiswa melakukan pendaftaran pengajuan judul tugas akhir dengan cara mengambil formulir pengajuan judul pada prodi masing-masing dan mengisi data judul tugas akhir yang ingin di jadikan Tugas akhir,

saat semua data terpenuhi maka mahasiswa akan memberikan kepada program studi. Program Studi akan melakukan pengecekan secara manual terhadap judul mahasiswa tersebut, pada pengecekan ini akan terjadi kesalahan atau *human error* yang mana mahasiswa dapat mengajukan judul yang sudah pernah di ajukan oleh mahasiswa sebelumnya, sehingga terjadi *duplicate* judul.

Teknologi yang mendukung dalam sistem pengajuan judul ini menggunakan web sebagai basisnya dan arsitektur menggunakan *Model View Controller*, dengan teknologi web dapat diakses oleh semua kalangan yang ada di kampus terutama mahasiswa, dosen karena sistem diakses secara online. Penerapan arsitektur *Model View Controller* yang memisahkan pengolahan data (*model*) dengan desain interface pengguna (*view*) yang kemudian mengontrol data yang diproses untuk ditampilkan (*Controller*) yang memudahkan dalam pembangunan sistem oleh programmer.

2. Metode Penelitian

Model-View-Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu Smalltalk

(Trygve Reenskaug) untuk meng-*enkapsulasi* data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface* (Deacon, 2009).

MVC mengikuti pendekatan yang paling umum dari *Layering*. *Layering* hanyalah sebuah logika yang membagi kode kita ke dalam fungsi di kelas yang berbeda. Pendekatan ini mudah dikenal dan yang paling banyak diterima. Keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode. (Satish, 2004)

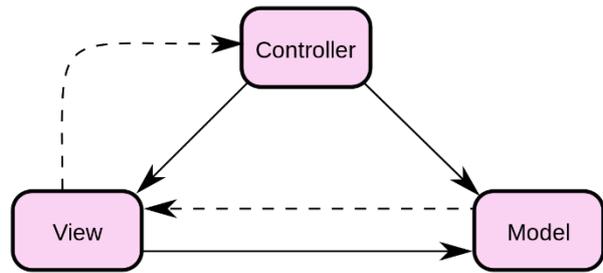
Definisi teknis dari arsitektur MVC dibagi menjadi tiga lapisan (Burbeck, 1992), yaitu :

a) *Model*, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi. Hanya model yang mengandung data dan fungsi yang berhubungan dengan pemrosesan data. Sebuah model meringkas lebih dari sekedar data dan fungsi yang beroperasi di dalamnya. Pendekatan model yang digunakan untuk komputer model atau *abstraksi* dari beberapa proses dunia nyata. Hal ini tidak hanya menangkap keadaan proses atau sistem, tetapi bagaimana sistem bekerja. Sebagai contoh, programmer dapat menentukan model yang menjembatani komputasi *back-end* dengan *front-end* GUI (*graphical user interface*).

b) *View*, bertanggung jawab untuk pemetaan *grafis* ke sebuah perangkat. View biasanya memiliki hubungan 1-1 dengan sebuah permukaan layar dan tahu bagaimana untuk membuatnya. View melekat pada model dan me-render isinya ke permukaan layar. Selain itu, ketika model berubah, view secara otomatis menggambar ulang bagian layar yang terkena perubahan untuk menunjukkan perubahan tersebut. Terdapat kemungkinan beberapa view pada model yang sama dan masing-masing *view* tersebut dapat merender isi model untuk permukaan tampilan yang berbeda.

c) *Controller*, menerima input dari pengguna dan menginstruksikan model dan *view* untuk melakukan aksi berdasarkan masukan tersebut. Sehingga, *controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi. Sebagai contoh, ketika pengguna mengklik tombol atau memilih item menu controller bertanggung jawab untuk menentukan bagaimana aplikasi seharusnya merespon.

Model, *view* dan *controller* sangat erat terkait, oleh karena itu, mereka harus merujuk satu sama lain.



Gambar 1. mengilustrasikan hubungan dasar Model-View-Controller.

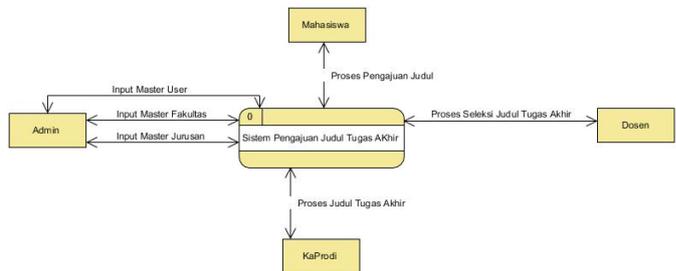
Arsitektur MVC memiliki manfaat yaitu pemisahan antara *model* dan *view* memungkinkan beberapa *view* menggunakan *model* yang sama. Akibatnya, komponen *model* sebuah aplikasi lebih mudah untuk diterapkan, diuji, dan dipelihara, karena semua akses ke *model* berjalan melalui komponen ini. (Balani:2002)

2.1 Perancangan

Perancangan sistem meliputi perancangan (*Data Flow Diagram*) perancangan *database* dan perancangan *user interface*. Perancangan sistem ini dilakukan dengan menggunakan bantuan perangkat lunak Visual Paradigm Enterprise

2.2 Data Flow Diagram level 0

Pada *Data flow Diagram level 0* ini menjelaskan tentang entiti yang digunakan dalam sistem aplikasi pengajuan judul tugas akhir. Adapun entitas yang digunakan yaitu : admin, KaProdi, Dosen, dan Mahasiswa

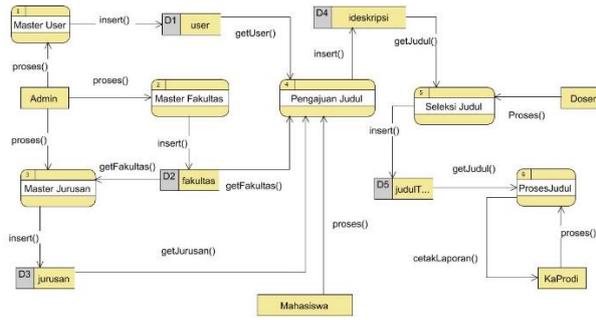


Gambar 2 Data Flow Diagram Level 0

Pada DFD level 0 ini terlihat beberapa proses yang dilakukan oleh entitas terhadap sistem pengajuan judul tugas akhir. Setiap proses kesistem akan dikembangkan pada DFD Level 1.

2.3 Data Flow Diagram level 1

Pada *Data flow Diagram level 1* ini akan melakukan pengembangan rancangan dari DFD level 0. Pada setiap proses akan dikembangkan lagi untuk mendapatkan setiap proses ke *data store*.



Gambar 3 Data Flow Diagram level 1

Pada Data Flow Diagram level 1 terlihat proses sudah dapat melakukan penyimpanan ke *data store*. Mahasiswa melakukan proses pengajuan judul dan dilakukan proses seleksi oleh entitas Dosen, hasil dari proses seleksi dosen akan diterima oleh KaProdi untuk dapat diketahui oleh mahasiswa untuk dapat melanjutkan tugas akhirnya. Penolakan judul akan diseleksi oleh dosen dan diberitahu ke mahasiswa bahwa judul yang di ajukan tersebut ditolak, sehingga mahasiswa dapat melakukan pergantian judul.

3. Hasil dan Pembahasan

Pengujian

Berikut ini hasil pengujian pada sistem pengajuan judul tugas akhir :

login

Gambar 4 Halaman login sistem pengajuan judul tugas akhir

Pada Gambar 4 di atas merupakan kondisi awal saat melakukan akses ke sistem. Ketika pengguna melakukan akses login dengan menggunakan username dan *password* yang sudah ditentukan, maka pengguna dapat masuk kedalam sistem, jika *username* dan *password* yang di inputkan salah maka pengguna tidak dapat mengakses sistem lebih lanjut.



Gambar 5 Proses Input data mahasiswa



Gambar 6 Pemberitahuan Kepada Mahasiswa

Saat mahasiswa belum di verifikasi oleh KaPordi maka mahasiswa belum bisa melakukan proses pengajuan judul tugas akhir.

Gambar 7 Form pengajuan judul tugas akhir

Pada gambar 7 mahasiswa dapat menuliskan deskripsi tugas akhirnya dan melakukan proses submit ke sistem sampai menunggu hasil seleksi dari dosen.

4. Kesimpulan

Setelah melakukan pembahasan dan pengujian pada sistem aplikasi pengajuan judul tugas akhir ini, maka dapat diambil kesimpulan yaitu :

1. Aplikasi pengajuan judul tugas akhir ini dapat dilakukan secara online.
2. Aplikasi ini sangat membantu mahasiswa yang jaraknya jauh dari kampus sehingga dapat menghemat waktu.
3. Membantu pihak Program studi dalam pengelola pengajuan judul tugas akhir
4. Setiap dosen, KaProdi, Mahasiswa akan mendapatkan username dan password dari admin.
5. Mahasiswa dapat melakukan pengajuan judul dengan menuliskan deskripsi dari tugas akhirnya

Daftar Rujukan

- [1] A. Hidayat and B. Surarso, "Penerapan Arsitektur Model View Controller (MVC) Dalam Rancang Bangun Sistem Kuis Online Adaptif," *Semin. Nas. Teknol. Inf. dan Komun.*, 2012.
- [2] D. Putra, A. Azhar, and A. Fata, "Rancang Bangun Aplikasi Pengajuan Proposal Judul Tugas Akhir Berbasis Web Dan Sms Gateway Menggunakan Konsep Model View Control," *J. Infomedia*, 2016, doi: 10.30811/v1i2.329.
- [3] R. A. Krisdiawan, "Rancang Bangun Sistem Informasi Judul Dan Pembimbing Skripsi /Tugas Akhir Fkom Uniku," *Buffer Inform.*, 2018, doi: 10.25134/buffer.v4i1.1135.

- [4] A. M. Rudianto, *Penrograman Web Dinamis menggunakan PHP dan MySQL*. 2011.
- [5] Abdul Kadir, "Pengenalan Sistem Informasi Edisi Revisi," *Edisi Revisi*. 2014.
- [6] Havaluddin, "Memahami Penggunaan UML (Unified Modelling Language)," *Memahami Pengguna. UML (Unified Model. Lang.*, 2011.
- [7] B. Sidik, "Framework Codeigniter," *Bandung: Informatika*. 2012.
- [8] M. Destiningrum and Q. J. Adrian, "Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)," *J. Teknoinfo*, 2017, doi: 10.33365/jti.v11i2.24.
- [9] B. Raharjo, "Belajar Otodidak Framework CodeIgniter," *Informatika Bandung*, 201
-