

PENERAPAN ARSITEKTUR MODEL VIEW CONTROLLER (MVC) PADA SISTEM INFORMASI E-SKRIPSI STMIK ROYAL

Rolly Yesputra¹, Nasrun Marpaung²

¹Sistem Komputer, Sekolah Tinggi Manajemen Informatika dan Komputer Royal
Jln. Prof. H. M. Yamin. No. 173, Kabupaten Asahan, Sumatera Utara 21222
Email: ¹rollyyp@royal.ac.id

²Sistem Informasi, Sekolah Tinggi Manajemen Informatika dan Komputer Royal
Jln. Prof. H. M. Yamin. No. 173, Kabupaten Asahan, Sumatera Utara 21222
Email: ²nasrunmrp@royal.ac.id

Abstrak

Penelitian ilmiah dalam bentuk skripsi merupakan mata kuliah yang wajib bagi mahasiswa yang harus diselesaikan sebagai salah satu syarat dalam proses penentuan kelulusan. Perancangan Aplikasi E-Skripsi yang menerapkan arsitektur Model View Controller ini dibuat menggunakan bahasa pemrograman PHP dan database MySQL. Aplikasi dengan konsep MVC yang memisahkan pengolahan data (Model) dengan antar muka yang ditampilkan oleh aplikasi (View) dan bagaimana mengatur data untuk ditampilkan (Controller) dapat memudahkan programmer dalam pengembangan aplikasi dengan menyesuaikan kebutuhan perguruan tinggi. Sehingga pada penelitian ini akan menghasilkan sebuah aplikasi E-Skripsi dengan arsitektur MVC berbasis web yang memberi kemudahan bagi mahasiswa dan pihak perguruan tinggi dalam proses pengajuan skripsi sesuai dengan yang dibutuhkan perguruan tinggi.

Kata Kunci : Arsitektur MVC, E-Skripsi, Skripsi

I.PENDAHULUAN

Pemanfaatan teknologi informasi diberbagai kalangan memiliki pengaruh yang sangat besar dalam pengelolaan setiap kegiatan untuk mendukung efektifitas dan kualitas kinerja yang dihasilkan. Begitu juga pada proses pendidikan yang berjalan pada lingkungan perguruan tinggi, dengan memanfaatkan teknologi. informasi tentunya akan berpengaruh terhadap kualitas pelayanan perguruan tinggi.

Dengan adanya peranan sistem informasi yang dibangun pada perguruan tinggi akan memudahkan setiap kegiatan dilingkungan perguruan tinggi tersebut, sistem informasi pengelolaan skripsi mahasiswa (e-skripsi) adalah salah satu contoh

pemanfaatan teknologi informasi yang memudahkan pihak perguruan tinggi untuk mengelola pengajuan judul skripsi mahasiswa, penentuan dosen pembimbing, maupun administrasi serta surat menyurat yang terintegrasi secara baik sesuai dengan kebutuhan perguruan tinggi dengan hasil yang efektif dan efisien.

Dalam mengelola judul skripsi mahasiswa yang diajukan tentunya membutuhkan sistem informasi pengelolaan skripsi (e-skripsi). Sistem Informasi E-Skripsi dirancang menggunakan bahasa pemrograman PHP dan MySQL sebagai database dengan penerapan arsitektur MVC yang memisahkan pengolahan data (*Model*) dengan desain interface pengguna (*View*) yang kemudian mengontrol data yang diproses untuk ditampilkan (*Controller*) yang nantinya juga memudahkan programmer dalam melakukan pengembangan sistem sesuai dengan kebutuhan perguruan tinggi untuk proses akademik yang lebih baik.

II.METODE PENELITIAN

Sistem Informasi E-Skripsi STMIK Royal yang dirancang dengan konsep *Model View Controller* ini menggunakan metodologi SDLC (*System Development Life Cycle*).



Gambar 1. System Development Life Cycle (Paramartha, Darmawiguna, & Kertiasih, 2017)

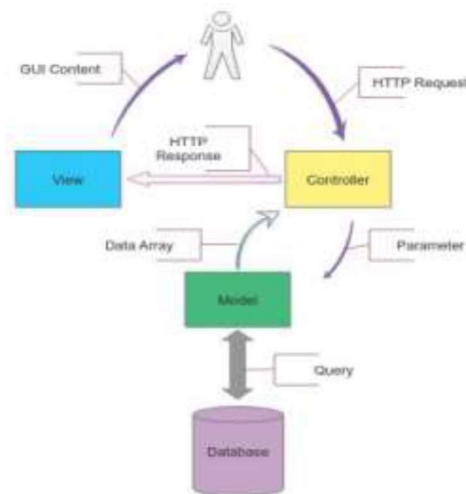
Berikut ini keterangan tahapan-tahapan waterfall menurut gambar diatas

1. *Requirements Analysis* , mengumpulkan kebutuhan secara komprehensif, lalu dilakukan analisa dan didefinisikan kebutuhan dari beberapa data yang dikumpulkan untuk diolah sehingga menjadi bagian penting yang harus dipenuhi oleh program nanti.

2. **Design**, setelah mendapatkan data yang sudah dikumpulkan sesuai dengan analisa kebutuhan, maka dilakukan perancangan sistem.
3. **Implementation**, pada tahapan ini dilakukan *coding* program yang akan dibuat sesuai dengan rancangan sistem pada tahap desain, dengan menggunakan bahasa pemrograman PHP. Setelah *coding* sudah dibuat, maka dilakukan pengujian tiap-tiap modulnya. Untuk pengkodean diimplementasikan sturur MVC didalamnya.
4. **Testing**, pada tahap ini dilakukan uji coba setelah program selesai dibuat.
5. **Evolution**, pada tahap ini dilakukan perubahan-perubahan pada sistem setelah dilakukan *testing*, sesuai dengan kebutuhan pengguna jika diperlukan.

Konsep Dasar *Model View Controller* (MVC)

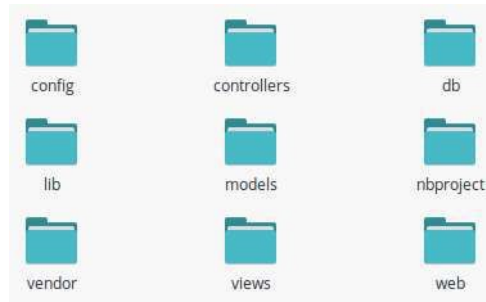
Menurut (Istiono, Hijrah, & Sutarya, 2016) *Model View Controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *Small Talk*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user *interface*, dan bagian yang menjadi kontrol aplikasi.



Gambar 2. Model MVC (Riana, Sanjaya, & Kalsoem, 2018)

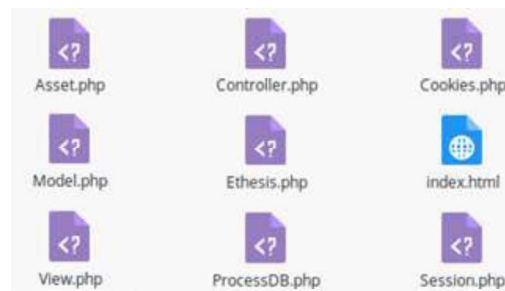
III.HASIL PEMBAHASAN

Implementasi arsitektur Model-View-Controller (MVC) pada aplikasi E-Skripsi dirancang dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai database. Struktur folder yang dibuat seperti gambar 2 berikut ini:



Gambar 3. Struktur Folder Aplikasi E-Skripsi

Program utama ditempatkan di dalam folder lib, dimana didalam folder ini terdapat beberapa program utama yang menerapkan konsep MVC tersebut. Isi dari folder tersebut sebagai berikut:



Gambar 4. File didalam folder lib

A. File Ethesis.php

Source code pada ethsis sebagai berikut:

```

1. <?php
2. class Ethesis {
3.     private $page;
4.     private $aksi;
5.     public static $pesan="";
6.     public function __construct() {
7.         $this->page = filter_input(INPUT_GET,
8.             "p", FILTER_SANITIZE_STRING);
9.         $this->aksi = filter_input(INPUT_GET,
10.            "x", FILTER_SANITIZE_STRING);
11.         if (empty($this->page) || $this->page
12.            == "index" || $this->page == "index")
13.            {
14.                require
15.                "../controllers/HomeController.php";
16.                $controller = new HomeController();
17.                if (isset($this->aksi) &&
18.                    method_exists($controller, "action" .
19.                    $this->aksi) {
20.                    $x = "action" . $this->aksi;
21.                    $controller->$x();
22.                } else {
23.                    $controller->index();
24.                }
25.            } else {
26.                // Jika File Tidak Ditemukan
27.                kembalikan ke index
28.                require
29.                "../controllers/HomeController.php";
30.                $controller = new HomeController();
31.                $controller->index(); } } }

```

Source Code 1. Pengontrolan utama aplikasi

B. File Model.php

Model merupakan source code yang akan digunakan untuk proses ke basisdata.

```

1. <?php
2. abstract class Model {
3.     public $crud;
4.     public function __construct() {
5.         $this->crud = new ProcessDB();
6.     }
7.     abstract public function tabel();
8.     abstract public function find();
9.     abstract public function
   findOne($rows=null, $where = null,
   $order = null, $limit = null);
10.    abstract public function save($rows =
   null, $page = null);
11.    abstract public function edit($field
   = null, $where = null, $page = null);
12.    abstract public function
   delete($where = null, $page = null);
13. }

```

Source Code 2. Model

Pada source code ini menggunakan class abstract dimana ada 6 (enam) fungsi abstract yang ada dalam class model ini. Berikut ini source code controller.php

```

1. <?php
2. class Controller{
3.     public function __construct() {
4.         $this->view=new View();
5.     }
6.     public function loadModel($nama){
7.         $path="../../models/".$nama.".php";
8.         if(file_exists($path)){
9.             require_once
   '../../models/'.$nama.'.php';
10.            $namamodel=$nama;
11.            $this->model=new $namamodel;
12.        }
13.    }
14.    public function post($text){
15.        return filter_input(INPUT_POST,
   $text);
16.    }
17.    public function get($text){
18.        return filter_input(INPUT_GET,
   $text);
19.    }

```

Source Code 3. Controller

Pada source code 3 di atas bisa dilihat bahwa *Controller* digunakan untuk mengontrol antara View dan Model serta begitu juga sebaliknya. Berikut kode program untuk view.

```

1. <?php
2. class View{
3. public function __construct() {
4. }
5. public function render($view,
6. $noinclude=false, $data=null, $data2=null, $fact=null){
7. if(file_exists("../views/" . $view . ".php")){
8.     if($noinclude){
9.         if($fact=="report"){
10.             require_once '../layout/header-report.php';
11.             require_once "../views/" . $view . ".php";
12.             require_once '../layout/footer.php';
13.         }else{
14.             require_once '../layout/header2.php';
15.             require_once "../views/" . $view . ".php";
16.             require_once '../layout/footer2.php';
17.         }
18.         //include header
19.         require_once '../layout/header2.php';
20.         //include contents
21.         require_once '../layout/contents2.php';
22.         require_once "../views/" . $view . ".php";
23.         //include footer
24.         require_once '../layout/footer2.php';
25.     }
26. }else{
27.     include_once '../layout/header2.php';
28.     //include contents
29.     require_once '../layout/contents2.php';
30.     require_once "../views/error/error.php";
31.     include_once '../layout/footer2.php';
32. }
33. }
34. }

```

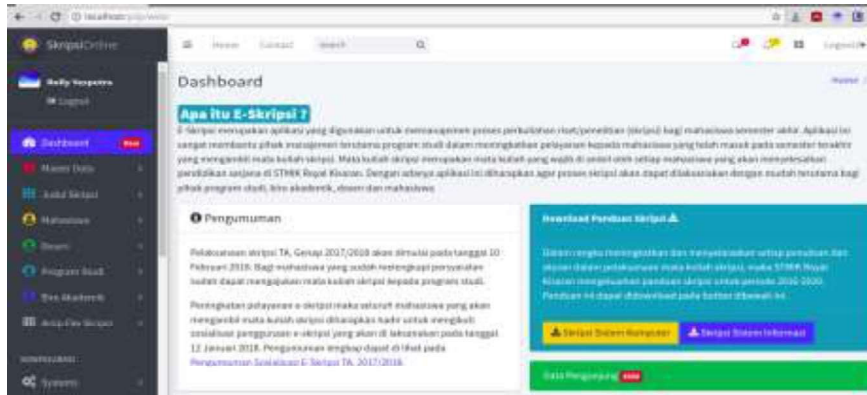
Source Code 4. Implementasi View.

Setelah dibuatkan program utama seperti source code diatas, maka bisa dilakukan implementasi untuk aplikasi E-Skripsi. Standar penamaan file harus mengikuti ketentuan yang sudah dibuat sesuai dengan program utama diatas. Untuk model menggunakan nama Judul.php, untuk *controller* menggunakan nama JudulController, sedangkan untuk view menggunakan nama folder dosen dan didalam folder tersebut ada beberapa file diantaranya, index.php, _form.php, detail.php dan edit.php.

Hasil dari program diatas sebagai berikut:

1. Tampilan Utama Aplikasi E-Skripsi.

Merupakan tampilan utama yang akan diakses dengan menu yang disediakan sesuai dengan kebutuhan pengguna (*user*).



Gambar 5. Tampilan E-Skripsi

2. Tampilan Data Judul Skripsi

Tampilan data judul yang menampilkan proses pengajuan judul dengan status diterima atau ditolak.

No	NIM	Prodi	Judul	Bahasa	Status	Aksi
1	14221386	SI	KEAMANAN MANAJEMEN SURAT MASUK DAN KELUAR MENGGUNAKAN TEKNOLOGI QR CODE PADA KANTOR DESA SEI ALIN HASAK	Visual Basic .Net	SUDAH ACC	[✓] [✗]
2	14221384	SI	ANALISIS PERAMALAN PERSEDIAAN BAHAN BAKU RAYU DADA CV. BINTANG ABADI	Visual Basic .Net	SUDAH ACC	[✓] [✗]
3	14221318	SI	Sistem Pembidangan Kapasitas Perencanaan Kinerja Sales Untuk Promosi Supervisor III PT WINA SENTOSA Dengan Metode SAW	Java	SUDAH ACC	[✓] [✗]

Gambar 6. Data judul skripsi

IV. PENUTUP

A. KESIMPULAN

Dengan adanya Sistem Informasi Skripsi *Online* (E-Skripsi) yang dibuat dengan arsitektur MVC, tentu akan memudahkan proses pengelolaan skripsi bagi civitas akademik, baik bagi mahasiswa maupun struktural akademik yang terkait pada sistem. Fitur-fitur yang dibuat pada e-skripsi disesuaikan dengan kebutuhan *user*, sehingga proses pengelolaan data menjadi lebih mudah dan cepat.

PHP versi 7 sangat mendukung object oriented programming, sehingga sangat mendukung dalam pengembangan aplikasi yang berarsitektur MVC. Dengan MVC ini penggunaan kode ulang (reusable) dapat dilakukan dengan mudah. Source code yang dibuat menjadi lebih rapi, dan mudah dibaca serta dipahami. Pengembangan aplikasi E-skripsi yang berarsitektur MVC sangat mudah, karena kode yang digunakan sudah dipisahkan sesuai dengan fungsinya. Ketika merubah view maka tidak akan mempengaruhi controller.

V. UCAPAN TERIMA KASIH

Sebagai rasa syukur dan terima kasih penulis yang telah menyelesaikan penelitian ini dengan baik, penulis mengucapkan terima kasih kepada Kementerian Riset Teknologi dan Pendidikan Tinggi atas bantuan berupa dukungan, kesempatan dan finansial terhadap penelitian ini untuk tahun anggaran 2018. Penulis juga mengucapkan terima kasih kepada Sekolah Tinggi Manajemen Informatika dan Komputer Royal yang telah membantu, memberikan dukungan sepenuhnya, memberikan ruang gerak yang sangat besar dan juga membantu dalam penyediaan saran dan prasarana dalam proses penelitian.

DAFTAR PUSTAKA

- Destiningrum, M., & Adrian, Q. J. (2017). Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus : Rumah Sakit Yukum Medical Centre). *Jurnal TEKNOINFO*, 11(2), 6–13. Retrieved from <http://ejournal.teknokrat.ac.id/index.php/teknoinfo/article/view/24>
- Ferdiansyah, D. (2018). Penerapan Konsep Model View Controller Pada Rancang Bangun Sistem Informasi Klinik Kesehatan Berbasis Web, 18(2), 195–205.
- Hidayat, A., & Surarso, B. (2012). Penerapan Arsitektur Model View Controller (MVC) Dalam Rancang Bangun Sistem Kuis Online Adaptif. *Seminar Nasional Teknologi Informasi Dan Komunikasi, 2012*(Sentika), 57–64. Retrieved from <https://fti.uajy.ac.id/sentika/publikasi/makalah/2012/2012-5.pdf>
- Istiono, W., Hijrah, & Sutarya. (2016). Pengembangan Sistem Aplikasi Penilaian dengan Pendekatan MVC dan Menggunakan Bahasa PHP dengan Framework Codeigniter dan Database MYSQL pada Paho College Indonesia. *Jurnal TICOM*, 5(1), 53–59. Retrieved from <https://media.neliti.com/media/publications/93757-ID-pengembangan-sistem-aplikasi-penilaian-d.pdf>
- Jogiyanto, H.M., 2005. Analisis dan Desain Sistem Informasi, Andi Offset, Yogyakarta.
- Paramartha, A. A. G. Y., Darmawiguna, I. G. M., & Kertiasih, N. K. (2017). Sistem Informasi Pembimbingan Skripsi Online Berbasis Web (Studi Kasus : Ftk , Undiksha).
- Ramadhan, M. R., Nugroho, L. E., & Sulisty, S. (2017). Perancangan Sistem Informasi Monitoring Skripsi, 290–295.
- Riana, D., Sanjaya, R., & Kalsoem, O. (2018). Sistem Informasi Manajemen Laboratorium Patologi Anatomi Menggunakan Model MVC Berbasis Laravel Framework. *Konferensi Nasional Sistem Informasi*, 237–242. Retrieved from <http://knsi2018.info/prosiding/>