

## ANALISA DAN PENERAPAN HMVC PADA APLIKASI E – LEARNING

Haryo Adi Nugroho  
1103080023

[rvodbluedragon@gmail.com](mailto:rvodbluedragon@gmail.com)

Pembimbing 1

Dana S. Kusumo, ST., MT., PhD  
02780291 – 1

[dskusumo.itt@gmail.com](mailto:dskusumo.itt@gmail.com)

Pembimbing 2

Ade Romadhony, ST., MT.  
06840337 – 1

[ade.romadhony@gmail.com](mailto:ade.romadhony@gmail.com)

### Abstrak

Aplikasi *e – learning* PT. Javan IT Services menerapkan pola *Model – View – Controller* (MVC) pada rancangannya. Namun pola ini memiliki kelemahan, yaitu keterkaitan yang erat antar komponen (*model*, *view*, dan *controller*) pada pola tersebut. Hal ini menyebabkan pengembangan aplikasi yang menerapkan pola MVC, termasuk aplikasi *e – learning* PT. Javan IT Services, menjadi lebih sulit. *Hierarchical – Model – View – Controller* (HMVC) merupakan pola variasi dari MVC. HMVC dikembangkan untuk mengatasi kekurangan yang dimiliki oleh MVC. *Cohesion* dan *coupling* merupakan atribut aplikasi yang dapat menunjukkan kualitas aplikasi tersebut. Perawatan, pengujian, dan pengembangan aplikasi akan lebih mudah dilakukan bila aplikasi tersebut memiliki *cohesion* dan *coupling* yang baik.

Pada tugas akhir ini, dua rancangan aplikasi *e – learning* PT. Javan IT Services dibuat. Salah satu rancangan menerapkan pola HMVC sedangkan rancangan lainnya menerapkan pola MVC. *Cohesion* dan *coupling* dari kedua rancangan kemudian dianalisa. Hasil analisa menunjukkan bahwa rancangan yang menerapkan pola HMVC memiliki *cohesion* dan *coupling* yang lebih baik dibandingkan dengan rancangan yang menerapkan pola MVC. Diharapkan melalui tugas akhir ini pengembang aplikasi web khususnya pengembang *e – learning* PT. Javan IT Services dapat melihat HMVC sebagai pola alternatif dalam mengembangkan aplikasinya.

**Kata kunci** : aplikasi web, *cohesion*, *coupling*, HMVC, MVC.

### Abstract

PT. Javan IT Services's *e – learning* web application use Model – View – Controller (MVC) in its design. However, this pattern has a weakness. There is a close connection between each component (*model*, *view*, and *controller*). This makes development of the application, including PT. Javan IT Services's *e – learning* which using it, become harder. *Hierarchical - Model - View - Controller* (HMVC) is variation of the MVC pattern. HMVC developed to overcome the shortcomings that are owned by the MVC. *Cohesion* and *coupling* are the attributes of an application that can indicate the quality of the application. Maintenance, testing, and application development will be easier to do if the application has good cohesion and coupling.

Two design of the web application has been made. One of it use HMVC pattern and the other use MVC pattern. Then coupling and cohesion of each pattern analyzed. The result shows that the design which use HMVC is better than the other. Hopefully web application developers, especially PT. Javan IT Services's *e – learning* developers, seen HMVC as alternative pattern in developing web applications.

**Keyword** : cohesion, coupling, HMVC, MVC, web application.

### 1. Pendahuluan

Bila dibandingkan dengan saat World Wide Web (WWW) berdiri, aplikasi web saat ini sudah berkembang jauh. Fungsi aplikasi web tidak lagi hanya sebagai media informasi. Berbagai macam layanan bisa disajikan dalam aplikasi web, contohnya layanan transaksi, media pembelajaran, dan permainan. Salah satu contoh aplikasi web adalah aplikasi *e – learning* yang dibuat oleh PT. Javan IT Services. Aplikasi ini memfasilitasi berbagai kegiatan yang terkait dengan pembelajaran, misalnya penyampaian materi, pengerjaan tugas, dan evaluasi tugas. Tujuan pembuatan aplikasi ini yaitu untuk membuat proses pembelajaran menjadi lebih menarik, mudah, dan penyampaian materi yang lebih baik.

*E – learning* PT. Javan IT Services menerapkan pola *Model – View – Controller* (MVC) pada rancangannya. MVC membagi sebuah sistem menjadi tiga komponen utama : Model, Controller, dan View. Meskipun MVC merupakan pola yang umum digunakan [1], namun MVC memiliki kekurangan, yaitu keterkaitan yang erat antar ketiga komponen (Model, View, dan Controller) [2]. Hal ini menyebabkan pengembangan aplikasi yang menerapkan pola MVC, termasuk aplikasi *e – learning* PT. Javan IT Services, menjadi lebih sulit [2].

Salah satu alternatif pola arsitektur yang ada yaitu *Hierarchical – Model – View – Controller* (HMVC). HMVC merupakan variasi dari MVC [1]. HMVC membagi aplikasi web menjadi komponen

– komponen yang disebut MVC *triad*. MVC *triad* adalah komponen yang menerapkan pola MVC. Setiap MVC *triad* saling terkait melalui Controller dari masing – masing MVC *triad* tersebut.

Kemudahan pengembangan sebuah aplikasi dapat dilihat dari *coupling* dan *cohesion* aplikasi tersebut. *Coupling* merupakan sebuah ukuran untuk melihat tingkat keterikatan antar komponen pada suatu sistem. *Cohesion* adalah ukuran keselarasan antar elemen dalam sebuah komponen di dalam sebuah sistem [3].

Berdasarkan paparan di atas maka permasalahan yang diangkat pada tugas akhir ini adalah “Apakah rancangan aplikasi *e – learning* PT. Javan IT Services yang menerapkan pola HMVC memiliki *cohesion* dan *coupling* yang lebih baik bila dibandingkan dengan rancangan serupa yang menerapkan pola MVC?”

Adapun batasan masalah pada tugas akhir ini yaitu :

- Requirement* aplikasi pada tugas akhir ini didefinisikan berdasarkan sumber acuan setelah dilakukan beberapa perubahan sesuai kebutuhan.
- Data terkait *coupling* dan *cohesion* diambil setelah tahap perancangan aplikasi, bukan setelah tahap implementasi.
- Analisa dilakukan pada tingkat *package* dan *class* dengan mengabaikan atribut dan operasi yang ada di dalamnya .
- Metrik yang digunakan adalah *relational cohesion*, *afferent coupling*, dan *efferent coupling*.

## 2. Cohesion dan Coupling

*Coupling* dan *cohesion* merupakan atribut internal suatu sistem yang digunakan untuk mengukur kualitas sistem tersebut. *Coupling* merupakan sebuah ukuran untuk melihat tingkat keterikatan sebuah komponen sistem dengan komponen lainnya. Pada sebuah sistem, *coupling* yang baik adalah *coupling* yang rendah. *Cohesion* adalah ukuran keselarasan antar elemen dalam sebuah komponen (*package*, *class*, dll.). Komponen pada sebuah sistem diharapkan memiliki *cohesion* yang tinggi.

Metrik yang digunakan untuk mengukur *cohesion* dan *coupling* pada tugas akhir ini adalah *relational cohesion*, *afferent coupling*, dan *efferent coupling*. *Relational cohesion* adalah *cohesion* yang dihitung dengan melihat relasi antar *class* dalam sebuah *package*. *Relational Cohesion* adalah angka rata – rata relasi internal yang ada di dalam sebuah *package*. *Relational cohesion* dihitung dari jumlah relasi yang ada di dalam *package* tersebut ditambah satu lalu dibagi jumlah *class* yang ada di dalam *package* tersebut. [4]

*Afferent coupling* dan *efferent coupling* adalah *coupling* yang dihitung dengan melihat relasi antara *class* dalam sebuah *package* dengan *class* lain di luar *package* tersebut. *Afferent coupling* merupakan *coupling export*. *Afferent coupling* adalah jumlah *class* di luar sebuah *package* yang membutuhkan *class* di dalam *package* tersebut [4] [5]. *Efferent coupling* merupakan *coupling import*. *Efferent coupling* adalah jumlah *class* di luar sebuah *package* yang dibutuhkan oleh *class* di dalam *package* tersebut [4] [5].

## 3. HMVC dan MVC

*Hierarchical Model – View – Controller* (HMVC) merupakan pola arsitektur variasi dari *Model – View – Controller* (MVC). Pada MVC, sebuah sistem dibagi menjadi tiga komponen utama : Model, Controller, dan View [6] [7]. Model merupakan komponen yang menangani data pada sistem. Model memiliki fungsi untuk memanipulasi data dan dapat mengakses basisdata. View adalah komponen yang menangani tampilan visual sistem. Controller berisi alur proses sistem dan menangani *input* dari pengguna sistem. Controller dapat memanggil Model dan View dalam menjalankan fungsinya.

Pada HMVC, sistem dibagi menjadi komponen – komponen yang disebut MVC *triad*. Setiap MVC *triad* adalah komponen yang menggunakan pola MVC. Setiap MVC *triad* saling terkait melalui Controller dari masing – masing MVC *triad* tersebut [8].

## 4. Rancangan Aplikasi

Pada tugas akhir ini aplikasi yang dirancang mengacu pada aplikasi web *e – learning* yang dibuat oleh PT. Javan IT Services. Rancangan aplikasi dibuat dua buah. Salah satu rancangan menerapkan pola HMVC sedangkan rancangan lainnya menerapkan pola MVC.

Pada rancangan yang menerapkan pola HMVC, aplikasi dibagi menjadi komponen – komponen yang disebut MVC *triad*. MVC *triad* pada aplikasi dibagi menjadi dua kelompok besar, yaitu MVC *triad* halaman dan MVC *triad* komponen.

MVC *triad* halaman adalah MVC *triad* yang bertanggung jawab untuk menangani halaman web secara keseluruhan. MVC *triad* ini dibuat berdasarkan halaman aplikasi. *Controller* pada MVC *triad* ini memiliki tanggung jawab untuk menerima *request* dari pengguna dan memprosesnya sedangkan *view* berisi rangka tampilan halaman yang diperlukan oleh *request* tersebut.

MVC *triad* komponen merupakan MVC *triad* yang bertanggung jawab untuk menangani komponen suatu halaman. MVC *triad* ini memiliki tanggung jawab dalam hal fungsionalitas dan basisdata

aplikasi. *Model* pada MVC *triad* ini dibuat berdasarkan data aplikasi. *Controller* bertugas menerima *request* dari MVC *triad* halaman dan memprosesnya. *View* pada komponen ini dapat berupa rangka tampilan atau *form* untuk menambah data baru.

Berbeda dengan rancangan pertama yang menerapkan pola HMVC, pada rancangan kedua aplikasi dirancang dengan menerapkan pola MVC. Pada rancangan ini, *request* dari pengguna akan ditangani oleh *controller* pada *package* Controller. *Controller* yang menangani *request* tersebut sesuai dengan halaman web yang di – *request*. Misalnya, jika *request* yang diterima berkaitan dengan halaman pelatihan, maka *class* Pelatihan yang akan menangani *request* tersebut. *Controller* tersebut kemudian akan memanggil *view* dan *model* yang dibutuhkan dari masing – masing *package* kemudian memprosesnya untuk memenuhi *request* dari pengguna.

## 5. Hasil analisa

Berdasarkan data yang diambil dari kedua rancangan aplikasi, terdapat beberapa kesimpulan yang bisa diambil. Yang pertama, *relational cohesion* pada rancangan yang menerapkan HMVC lebih tinggi bila dibandingkan dengan rancangan yang menerapkan MVC. Pada rancangan yang menerapkan HMVC, *relational cohesion* pada setiap *package* memiliki nilai satu atau lebih, sedangkan pada rancangan yang menerapkan pola MVC setiap *package* memiliki nilai 0,1 atau lebih rendah. Kesimpulan kedua, *afferent coupling* (Ca), dan *efferent coupling* (Ce) pada rancangan yang menerapkan pola HMVC memiliki nilai tertinggi yang rendah, yaitu hanya empat, sedangkan rancangan yang menerapkan pola MVC memiliki nilai tiga puluh tujuh.

## 6. Penutup

Pada tugas akhir ini telah dipaparkan perbandingan *cohesion* dan *coupling* antara rancangan aplikasi *e – learning* PT. Javan IT Services yang menerapkan pola HMVC dengan rancangan serupa yang menerapkan pola MVC. Berdasarkan hasil analisa, dapat disimpulkan bahwa rancangan yang menggunakan HMVC memiliki *cohesion* dan *coupling* yang lebih baik bila dibandingkan dengan rancangan yang menggunakan pola MVC.

*Cohesion* dan *coupling* yang baik pada rancangan tersebut berperan pada kualitas aplikasi yang lebih baik. Perawatan, pengujian, dan pengembangan aplikasi akan lebih mudah dilakukan bila rancangan aplikasi tersebut menggunakan pola HMVC dibandingkan dengan menggunakan pola MVC.

Selain *coupling* dan *cohesion*, pengujian terhadap atribut aplikasi lain perlu dilakukan. Pengujian pun perlu dilakukan pada tahap yang berbeda.

Contohnya, pengujian dapat dilakukan pada aplikasi hasil implementasi.

## Daftar Pustaka

- [1] S. d. Freyssinet, “Scaling Web Applications with HMVC,” 22 februari 2010. [Online]. Available: <http://techportal.inviqa.com/2010/02/22/scaling-web-applications-with-hmvc/>. [Diakses 19 september 2013].
- [2] S. A. M. Rizvi dan S. I. Hassan, “Achieving Loose Coupling Between Different Components of Model-View-Controller For Web Based Application,” New Delhi, 2009.
- [3] J. Eder, G. Kappel dan M. Schrefl, *Coupling and Cohesion in Object - Oriented Systems*, 1994.
- [4] J. Wüst, “SDMetric User Manual,” 2002. [Online]. Available: <http://www.sdmetrics.com/manual/index.html>. [Diakses 22 3 2014].
- [5] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*, Prentice Hall PTR, 2003.
- [6] C. Heller, D. Streitferdt dan I. Philippow, *A new Pattern Systematics*, 2005.
- [7] A. Karagkasidis, “Developing GUI Applications: Architectural Patterns Revisited,” EuroPLoP, 2008.
- [8] J. Cai, R. Kapila dan G. Pal, “HMVC: The layered pattern for developing strong client tiers,” juli 2000. [Online]. Available: <http://www.javaworld.com/jw-07-2000/jw-0721-hmvc.html>. [Diakses 19 september 2013].
- [9] G. Kappel, B. Proll, R. Siegfried dan W. Retschitzegger, *Web engineering: The Discipline of Systematic Development of Web Applications.*, John Wiley & Sons, 2006.
- [10] M. Hitz dan B. Montazeri, *Measuring Coupling and Cohesion in Object - Oriented Systems*, 1995.
- [11] Wescrow, “Hierarchical Model-View-Controller (HMVC): Planning for the Future,” 9 Januari 2012. [Online]. Available: <http://somethingstatic.com/hierarchical-model-view-controller-planning-future/>. [Diakses 18 Maret 2014].
- [12] G. Gui dan P. D. Scott, *Coupling and Cohesion Measures for Evaluation of Component Reusability*.