

EFEKTIVITAS ARSITEKTUR APLIKASI JAVA GUI SWING DENGAN METODE MODEL-VIEW-CONTROLLER

Arochman⁽¹⁾, Tory Arianto⁽²⁾

Program Studi Teknik Informatika, STMIK Widya Pratama
Jl. Patriot 25 Pekalongan Telp (0285) 427816
email : arochman.aryanta@gmail.com

ABSTRAK

Java merupakan bahasa pemrograman yang matang. Java menyediakan fitur komponen Swing yang memuat semua kelas-kelas yang dibutuhkan untuk membuat aplikasi berbasis GUI. Penggunaan Swing pada pengembangan aplikasi berbasis GUI memberikan kemudahan. Akan tetapi, dalam arsitektur aplikasi memiliki kode yang sangat “kotor” dan kurang efektif. Di mana kode yang berisi pengendalian terhadap event komponen Swing bercampur dengan kode yang berisi aturan bisnis dan kode yang berisi manipulasi terhadap data (Bima, 2011). Pada tahun 1970-an Trygve Reenskaug memunculkan ide untuk memisahkan proses logika aplikasi dengan antarmuka pada aplikasi web. Dari ide tersebut munculah sebuah pattern Model-View-Controller atau yang dikenal dengan metode MVC. Sebuah metode yang mampu memudahkan dalam pengembangan dan perubahan antar muka aplikasi web. Berdasarkan permasalahan yang serupa pada arsitektur Java GUI Swing akan digunakan metode MVC untuk meningkatkan efektifitas arsitektur Java GUI Swing.

Kata Kunci: metode Model-View-Controller, efektifitas arsitektur aplikasi, java GUI Swing

1. PENDAHULUAN

1.1 Latar Belakang

Java merupakan bahasa pemrograman yang matang. Beragam fasilitas yang terus dikembangkan untuk menyelesaikan persoalan-persoalan di bahasa pemrograman Java. Dalam memenuhi kebutuhan pengembangan aplikasi berbasis Graphical User Interface (GUI) Java menyediakan Java Foundation Class (JFC) (Hariyanto, 2014). Pada JFC tersebut Java menyediakan fitur komponen Swing yang memuat semua kelas-kelas yang dibutuhkan untuk membuat aplikasi berbasis GUI.

Swing dikembangkan untuk menyediakan komponen-komponen GUI yang lebih canggih dari komponen-komponen sebelumnya yang bernama AWT (Abstract Windows Toolkit). Secara arsitektur, Swing dibangun diatas arsitektur AWT yang masih memiliki banyak kekurangan. Kehadiran Swing tidak lain untuk memperbaiki kekurangan dari AWT (Hermawan, 2004).

Penggunaan Swing pada pengembangan aplikasi berbasis GUI memberikan kemudahan. Akan tetapi, dalam arsitektur aplikasi memiliki kode yang sangat “kotor” dan kurang efektif. Di mana kode yang berisi pengendalian terhadap event komponen Swing bercampur dengan kode yang berisi aturan bisnis dan kode yang berisi manipulasi terhadap data (Bima, 2011).

Permasalahan pencampuran kode yang berisi aturan bisnis dan kode yang berisi manipulasi terhadap data merupakan permasalahan yang sama

pada pengembangan aplikasi berbasis web. Hal tersebut terlihat pada tahun 1970-an Trygve Reenskaug memunculkan ide untuk memisahkan proses logika aplikasi dengan antarmuka. Dari ide tersebut munculah sebuah *pattern Model-View-Controller* atau yang dikenal dengan metode MVC. Sebuah metode yang mampu memudahkan dalam pengembangan dan perubahan antar muka aplikasi web.

Metode MVC telah diterapkan di beberapa pengembangan aplikasi diantaranya pada: sistem informasi keuangan (Hartomo, 2009), pendaftaran mahasiswa baru online (Fathiah, 2013), kuis online adaptif (Hidayat, 2012). Hasil menunjukkan bahwa penerapan metode MVC pada aplikasi berbasis web lebih baik dari pada pengembangan aplikasi secara konvensional.

Hasil yang baik telah ditunjukkan oleh metode MVC untuk memisahkan proses logika aplikasi dengan antarmuka pada pengembangan aplikasi berbasis web. Berdasarkan permasalahan yang serupa pada arsitektur Java GUI Swing akan digunakan metode MVC untuk meningkatkan efektifitas arsitektur Java GUI Swing.

1.1.1 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah disebutkan dalam uraian sebelumnya maka dapat dirumuskan permasalahannya adalah Perlu pemisahan kode yang berisi aturan bisnis dan kode yang berisi manipulasi terhadap data pada arsitektur aplikasi Java GUI Swing sehingga lebih efektif.

Pertanyaan penelitian (*research question*) yang dipilih:

1. Bagaimana menerapkan metode MVC untuk meningkatkan efektifitas pada arsitektur aplikasi Java GUI Swing?
2. Seberapa efektif penerapan metode MVC pada arsitektur aplikasi Java GUI Swing?

1.1.2 Tujuan

Sesuai dengan latar belakang dan permasalahan, penelitian ini dilaksanakan dengan tujuan menerapkan metode MVC untuk meningkatkan efektifitas pada arsitektur aplikasi Java GUI Swing.

1.1.3 Manfaat

1. Memberikan kontribusi keilmuan pada pada bidang penelitian *software engineering*.
2. Memberikan kontribusi dalam penggunaan Java GUI Swing yang lebih efektif.

1.2 Landasan Teori

1.2.1 Tinjauan Studi

Terdapat penelitian–penelitian sebelumnya yang mendasari dan menjadi tinjau dalam penelitian ini. Diantaranya penelitian yang dilakukan oleh Hartomo pada tahun 2009. Penerapan MVC pada pengembangan aplikasi CMS Sistem Informasi Keuangan, dan disimpulkan bahwa penulisan kode program dengan menggunakan arsitektur MVC lebih terorganisasi dengan baik dan aplikasi lebih mudah dikembangkan (Hartomo, 2009). Selain itu Fathiah dan Hidayat juga telah menerapkan MVC pada penelitiannya dan didapat bahwa penerapan MVC memudahkan pada pengembangan aplikasi web karena memisahkan antara *logic* bisnis, *database*, dan *view* (web design) serta dapat meningkatkan modularitas dan reusabilitas dari sistem (Fathiah, 2013) (Hidayat, 2012).

Berdasarkan tinjauan studi diatas, menjadikan dasar usulan penelitian ini bahwa akan diterapkan metode MVC untuk peningkatan efektifitas pada arsitektur aplikasi Java GUI Swing.

1.2.2 Tinjauan Pustaka

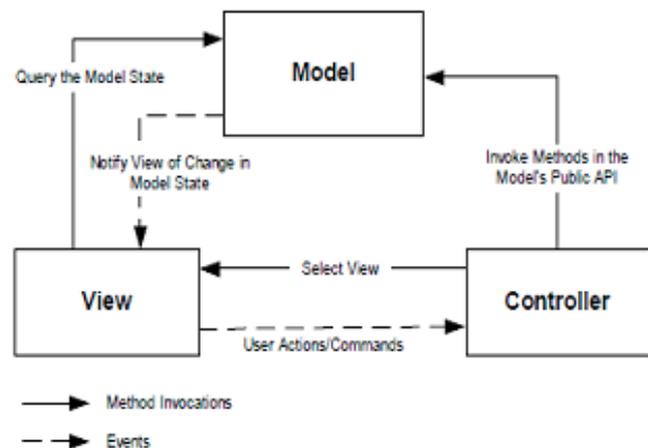
1.2.2.1 Design-pattern MVC (Model-View-Controller)

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 tampilan muka (Deacon, 2009). MVC mengikuti pendekatan yang paling umum dari *layering*. *Layering* hanyalah sebuah logika yang membagi kode kedalam fungsi di kelas yang berbeda. Keuntungan utama dalam

pendekatan ini adalah penggunaan ulang (*reusability*) kode (Satish, 2004).

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

1. *Model*, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi. Model mengandung data dan fungsi yang berhubungan dengan pemrosesan data.
2. *View*, bertanggung jawab untuk pemetaan grafis sebuah perangkat.
3. *Controller*, menerima input dari pengguna dan mendistribusikan *model* dan *view* untuk melakukan aksi berdasarkan masukan tersebut. Sehingga *controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi.
4. *Model*, *View*, dan *Controller* ketiganya berkaitan erat, oleh karena itu ketiganya harus merujuk satu sama lain. Ilustrasi hubungan dasar *Model-View-Controller* dapat digambarkan sebagai berikut.

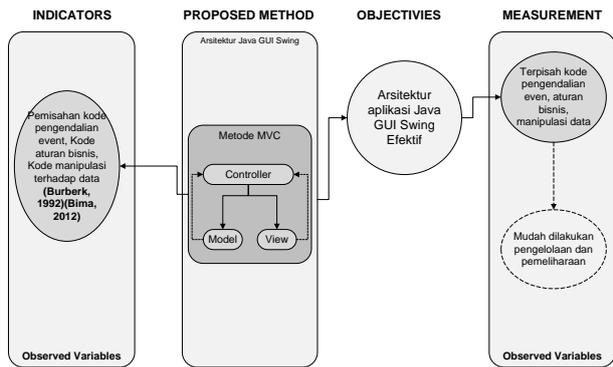


Gambar 1 Hubungan antara model, view dan controller (Gulzar, 2002)

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).

1.2.3 Kerangka Teoritis

Kerangka teoritis dalam penerapan Metode MVC pada arsitektur aplikasi Java GUI Swing dapat digambarkan sebagai berikut.



Gambar 2 Kerangka teoritis penerapan Metode MVC pada arsitektur aplikasi Java GUI Swing

2. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah metode eksperimen dengan tahapan pengumpulan data, pengolahan data, eksperimen, Evaluasi dan validasi.

2.1 Metode Pengumpulan Data

1. Sumber data

Pada tahap persiapan dan penelitian awal, pelaksanaan pengumpulan data menggunakan jenis data sekunder. Data sekunder dikumpulkan dari beberapa buku, jurnal ilmiah dan website. Pengumpulan data sekunder dilakukan berdasarkan pemahaman yang lebih lengkap mengenai masalah efektifitas arsitektur aplikasi Java GUI Swing.

Pada tahap ini diharapkan diidentifikasi berbagai teori atau data sekunder yang lebih lengkap hingga dapat digunakan untuk memodelkan permasalahan dan melakukan analisis solusi.

2. Sampel Penelitian

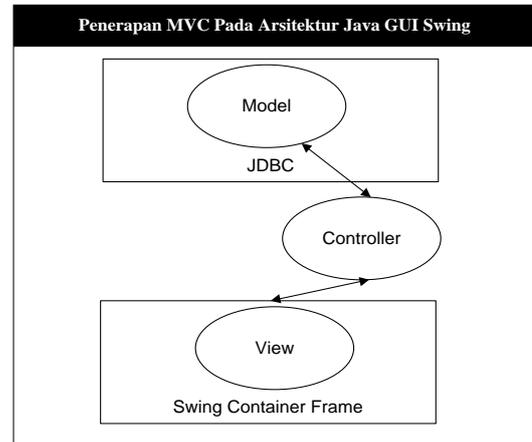
Penerapan metode MVC untuk efektifitas arsitektur aplikasi Java GUI Swing yang akan dibahas sebagai studi kasus adalah aplikasi database dengan penggunaan komponen Swing dan database Mysql.

2.2 Metode Pengolahan Awal Data

Setelah data-data sekunder terkumpul maka langkah berikutnya akan dilakukan pengolahan awal data. Metode yang akan digunakan adalah metode *verification* dengan tujuan pada mulanya asumsi awal atau hipotesa, yang kemudian akan diverifikasi dengan data.

2.3 Metode yang Diusulkan

Penerapan metode MVC dalam penelitian ini didasarkan pada studi literatur tentang penerapan metode MVC dalam pengembangan aplikasi (Burbeck, 1992). Metode MVC dipilih dan diterapkan karena dipandang mampu meningkatkan efektifitas dalam arsitektur aplikasi Java GUI Swing.



Gambar 3 Model yang diusulkan

Model yang menjadi usulan akan dilakukan pengukuran pencapaian tingkat efektifitasnya.

2.4 Eksperimen dan Pengujian Model/Metode

Desain eksperimen dalam penelitian ini adalah *one-shot case study*, dengan tujuan pemberian perlakuan dan kemudian menganalisa hasil setelah adanya perlakuan. Eksperimen bertujuan untuk meningkatkan efektifitas arsitektur aplikasi Java GUI Swing.

Penerapan MVC pada java GUI Swing akan dilakukan pada Aplikasi pengelolaan data karyawan. Dengan aplikasi yang sederhana ini diharapkan mampu dilakukan evaluasi dan validasi penerapan MVC pada java GUI Swing secara mendalam.

2.5 Evaluasi dan validasi hasil

Untuk mengetahui seberapa efektif metode MVC yang diterapkan dalam arsitektur aplikasi Java GUI Swing. Beberapa upaya akan dilakukan untuk mengukur keefektifan metode MVC pada aplikasi Java GUI Swing diantaranya percobaan untuk melakukan pengelolaan dan perubahan tampilan, query dan logika proses.

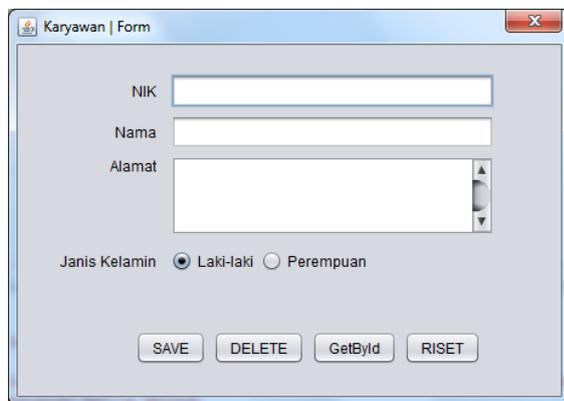
3. HASIL DAN PEMBAHASAN

3.1 Penerapan metode MVC pada arsitektur Java GUI Swing

Penerapan model yang menjadi usulan dalam penerapan metode MVC pada arsitektur Java GUI Swing digunakan sebuah sampel aplikasi pengelolaan data. Hal tersebut diharapkan metode MVC mampu diterapkan sepenuhnya pada arsitektur Java GUI Swing.

3.2 Sampel aplikasi

Sampel aplikasi yang akan digunakan dalam penelitian ini adalah aplikasi pengelolaan data karyawan. Aplikasi tersebut menggunakan DBMS MYSQL dan dengan bahasa pemrograman java. Tabel karyawan memiliki empat field yaitu nik, nama, alamat, jenis kelamin. Field-field tersebut dalam kelas model akan menjadi atribut. Sehingga kelas model yang akan dihasilkan harus mengimplementasikan tabel karyawan dengan tepat.



Gambar 4 Tampilan muka aplikasi pengelolaan data karyawan

Pada tampilan muka pengelolaan data karyawan terdiri dari lima tombol yang diharapkan mampu untuk menangani pengelolaan data yaitu tambah, ubah, hapus, dan tampil (Cread-Read-Update-Delete).

3.2.1 Model

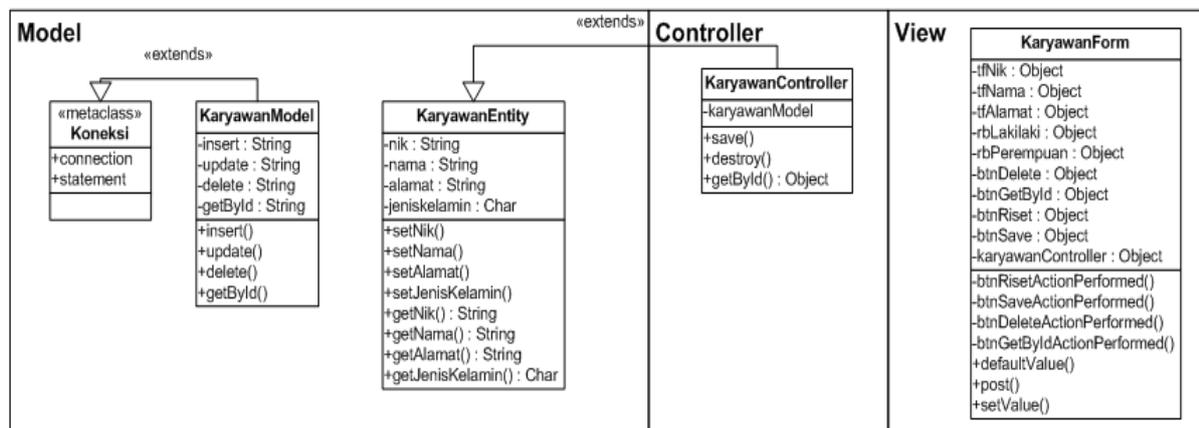
Model mengandung data dan fungsi yang berhubungan dengan pemrosesan data. Dalam aplikasi pengelolaan data karyawan model berbentuk sebuah kelas karyawan. Kelas karyawan mengandung atribut karyawan, enkapsulasi (setter dan getter) atribut dan metode insert, update, delete, select. Koneksi ke database menggunakan driver MYSQL JDBC Driver dan metode penulisan query menggunakan PreparedStatement.

3.2.2 View

View diterjemahkan sebagai tampilan muka dari aplikasi, maka dalam pengelolaan data karyawan perwujudan dari view adalah form karyawan. Pembuatan form karyawan dilakukan secara visual dengan menggunakan IDE Netbeans.

3.2.3 Controller

Controller berfungsi untuk menerima input dari pengguna dan mendistribusikan *model* dan *view* untuk melakukan aksi berdasarkan masukan tersebut. Pada aplikasi pengelolaan data karyawan controller akan dieksekusi melalui event-event yang ada pada view.



Gambar 5 Penerapan metode MVC pada arsitektur Java GUI Swing

3.3 Efektifitas penerapan metode MVC pada arsitektur Java GUI Swing

Metode MVC berhasil diterapkan pada aplikasi berarsitektur java GUI swing. Pemisahan kode program logika proses dan antarmuka pada aplikasi pengelolaan data karyawan dapat dilakukan dengan baik. Pemisahan tersebut dapat terlihat pada package model, controller dan view. File-file class yang terdapat dalam package tersebut dapat mengimplementasikan fungsi-fungsi sebagaimana yang ada pada metode MVC.

Pengembangan aplikasi berarsitektur java GUI lebih efektif dengan penerapan metode MVC. Karena adanya pemisahan kode program logika proses dan antarmuka. Dengan pemisahan kode program tersebut perawatan dan pengembangan aplikasi dinilai lebih efektif. Misal dalam pengembangan aplikasi diperlukan perubahan tampilan muka, maka hanya pada class view yang dikelola untuk dilakukan perubahan. Pencarian file akan lebih terfokus pada file view terkait sehingga ketika akan dilakukan perubahan tidak perlu mencari kesemua file dalam penelusurannya. Pemisahan kode program query mampu memberikan kontribusi yang positif ketika pengembangan dan perubahan harus dilakukan pada pengelolaan data.

4. KESIMPULAN

Dari tahapan-tahapan yang telah diselesaikan pada penelitian ini berjalan dengan baik. Dari penelitian ini telah membuktikan bahwa,

1. Metode MVC mampu diterapkan pada arsitektur java GUI swing.
2. Penerapan metode MVC mampu meningkatkan efektifitas dalam pengembangan aplikasi berarsitektur java GUI swing.

Selain hal tersebut diatas penerapan metode MVC akan sangat membantu dalam perawatan dan pengembangan aplikasi berarsitektur java GUI swing pada waktu mendatang. Penerapan metode MVC pada arsitektur java GUI swing masih dilakukan pada lingkup pengembangan aplikasi yang sederhana. Untuk itu perlu dilakukan penelitian lebih lanjut penerapan metode MVC pada arsitektur java GUI swing pada lingkup aplikasi yang lebih kompleks. Sehingga akan diketahui sejauh mana efektifitas penerapan metode MVC pada arsitektur java GUI swing.

Pada penelitian berikutnya perlu adanya pengembangan metode MVC untuk meningkatkan kinerja model tersebut dalam pengembangan aplikasi berarsitektur java GUI swing. Hal ini akan

memberikan kontribusi dalam bidang *software engineering* khususnya dalam pengembangan aplikasi berarsitektur java GUI swing lainnya.

5. DAFTAR PUSTAKA

Balani, N., 2002. *Web services architecture using MVC style*. [Online] Available at: [http://www.webifysolutions.com?subject=Web services architecture using MVC style](http://www.webifysolutions.com?subject=Web%20services%20architecture%20using%20MVC%20style) [Accessed 18 October 2014].

Bima, I., 2011. *Java Desktop - Aplikasi POS Berarsitektur Three Tier Menggunakan Swing, Hibernate, dan Spring*.

Burbeck, S., 1992. *Application Programmings in Smaltalk's 80™: How To Use MVC*. [Online] Available at: <http://stwww.cs.illinois.edu/users/smarch/stdocs/mvc.html> [Accessed 14 October 2014].

Deacon, J., 2009. *Model-View-Controller*. [Online] Available at: <http://www.jdl.co.uk/briefings/index.html/#mvc> [Accessed 18 October 2014].

Fathiah, 2013. Studi Dan Penerapan Framework Mvc Dan Ajax Pada Pengembangan Sistem Pendaftaran Mahasiswa Baru Online. *Jurnal Ilmiah STMIK U'Budiyah*, pp.84-70.

Gulzar, N., 2002. *Fast track to struts: what it does and how*. [Online] Available at: <http://media.techtarget.com/tss/static/articles/content/StrutsFastTrack/StrutsFastTrack.pdf> [Accessed 20 October 2014].

Hariyanto, B., 2014. *Esensi-esensi Bahasa Pemrograman Java*. Bandung: Informatika.

Hartomo, K.D., 2009. Implementasi Model View Controller dan Object Relational Mapping pada Content Management System Sistem Informasi Keuangan. *Jurnal Teknologi Informasi-Aiti*, pp.31-44.

Hermawan, B., 2004. *Menguasai Java 2 & Object Oriented Programming*. Yogyakarta: Andi.

Hidayat, A., 2012. Penerapan Arsitektur Model View Controller (MVC) Dalam Rancang Bangun Sistem Kuis Online adaptif. In *Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012)*. Yogyakarta, 2012.

