

## **ABSTRAK**

Perkembangan teknologi pada pengembangan aplikasi berbasis web saat ini berkembang pesat dengan adanya arsitektur Model-View-Controller(MVC). Microsoft mengembangkan sebuah framework yang mengadaptasi arsitektur MVC. Framework ini dapat digunakan untuk mengembangkan sebuah aplikasi web yang bersifat modular. Pada framework tersebut terdapat kekurangan yang dapat mengakibatkan kurang optimalnya sebuah aplikasi web pada saat pengembangan seperti adanya fat controller dan tidak ada dependency injection container bawaan. Oleh karena itu diperlukan solusi untuk mengatasi kekurangan-kekurangan tersebut dan akan memudahkan pengembangan ke tingkat lebih lanjut dari sebuah aplikasi web apabila terjadi perubahan-perubahan proses bisnis atau perubahan pada logika pengolahan data. Solusi yang akan dikemukakan pada penelitian ini adalah penggunaan repository pattern dan penambahan dependency injection container dalam proses pengembangan aplikasi web yang berbasis ASP.NET MVC Framework. Hasil akhir dari penelitian ini adalah pembuktian optimalisasi yang dilakukan dengan cara penggunaan repository pattern dan penambahan dependency injection container dalam proses pengembangan sebuah aplikasi web berbasis ASP.NET MVC Framework.

Kata kunci: repository pattern, dependency injection, asp net mvc

## **ABSTRACT**

*Technological developments in web-based application development is currently booming with the Model-View-Controller (MVC). Microsoft developed a framework that adapts the MVC architecture. This framework can be used to develop a web application that is modular. On the framework contained deficiencies that can result in less optimal a web application at the time of development such as the fat controller and no dependency injection container default. Therefore we need a solution to overcome these shortcomings and will facilitate further development to the level of a web application in case of changes in business processes or changes to the data processing logic. The solution will be presented in this study is the use of the repository pattern and adding dependency injection container in the process of web application development based ASP.NET MVC Framework. The end result of this research is proving the optimization is done by the use of the repository pattern and adding dependency injection container in the process of developing a web application based on ASP.NET MVC Framework.*

*Keywords: repository pattern, dependency injection, asp net mvc*

## DAFTAR ISI

Lembar Pengesahan .....	i
Pernyataan Orisinalitas Laporan Penelitian .....	ii
Pernyataan Publikasi Laporan Penelitian.....	iii
Prakata .....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Tujuan Pembahasan .....	3
1.4. Ruang Lingkup Kajian .....	3
1.5. Sistematika Penyajian .....	4
BAB II TEORI .....	6
2.1. Permasalahan-permasalahan Pada ASP.NET MVC Framework.....	6
2.2. Contoh <i>Framework</i> yang memiliki <i>native IoC Container</i> .....	8
2.2.1. <i>Spring Framework</i> .....	8
2.2.2. <i>Symfony2 PHP Framework</i> .....	9
2.2.3. <i>Use Case Diagram</i> .....	18
2.2.4. <i>Class Diagram</i> .....	19
2.2.5. <i>Activity Diagram</i> .....	21
2.3. <i>Dependency Injection</i> .....	11
2.3.1. <i>Constructor Injection</i> .....	11
2.3.2. <i>Property Injection</i> .....	11
2.3.3. <i>Method Injection</i> .....	11
2.3.5. <i>Dependency Injection</i> Pada ASP.NET MVC Menurut Buku “ASP.NET MVC 4 in Action” .....	14
2.4. <i>Repository Pattern</i> .....	16
2.5. <i>Unified Modelling Language (UML)</i> .....	17
2.6. Entity Relationship Diagram .....	22
2.7. Microsoft Visual C# .....	22
2.8. Microsoft SQL Server (Transact SQL).....	24
2.9. Pengertian ASP.NET.....	24
2.10. Arsitektur Model-View-Controller .....	25

2.11.	ASP.NET Model-View-Controller 5 <i>Framework</i> .....	25
2.12.	<i>Entity Framework</i> .....	26
2.13.	<i>Object Oriented Application Framework</i> .....	26
2.14.	<i>StructureMap</i> .....	26
2.15.	<i>AutoMapper</i> .....	27
2.16.	<i>JSON.NET</i> .....	27
2.17.	Microsoft Unity .....	27
BAB III ANALISIS DAN RANCANGAN SISTEM .....		29
3.1.	Application Framework.....	29
3.1.1.	Penggunaan <i>IoC Container</i> .....	29
3.1.2.	Penggunaan <i>Automatic Mapping</i> .....	30
3.1.3.	Menghapus Penggunaan <i>String</i> sebagai Parameter Pada <i>Method RedirectToAction</i> .....	31
3.1.4.	Penggunaan <i>Repository Pattern</i> .....	31
3.2.	Tools Yang Digunakan Pada Application Framework .....	32
3.3.	Entity Relationship Diagram .....	34
BAB IV PENGEMBANGAN PERANGKAT LUNAK .....		38
4.1.	Solusi Menurut Hanselman .....	38
4.1.1.	Penggunaan <i>Repository Pattern</i> .....	38
4.1.2.	Penggunaan <i>Dependency Injection</i> .....	39
4.2.	Solusi Menurut Adam Freeman dan Steven Sanderson .....	41
4.3.	Aplikasi <i>Unit Of Work</i> .....	42
4.4.	Implementasi <i>Infrastructure</i> .....	44
4.4.1.	<i>FreakCommerceDependencyResolver</i> .....	44
4.4.2.	<i>FreakCommerceServiceLocator</i> .....	44
4.4.3.	<i>CommonRegistration</i> .....	45
4.4.4.	<i>ControllerConvention</i> .....	45
4.4.5.	<i>ControllerRegistration</i> .....	46
4.4.6.	<i>WebRegistration</i> .....	46
4.4.7.	<i>ActionFilterRegistration</i> .....	47
4.4.8.	<i>IRunAfterRequest</i> .....	47
4.4.9.	<i>IRunAtInit</i> .....	47
4.4.10.	<i>IRunAtStartup</i> .....	48
4.4.11.	<i>IRunAtError</i> .....	48

4.4.12. IRunOnRequest .....	48
4.4.13. JobRegistration .....	48
4.4.14. Contoh Penggunaan <i>Unit of Work</i> .....	49
4.5. DbInfrastructure .....	49
4.5.1. IFreakCommerceContext.....	49
4.5.2. FreakCommerceContext.....	50
4.5.3. Bank .....	51
4.5.4. Banner .....	51
4.5.5. Category .....	52
4.5.6. City .....	52
4.5.7. Confirmation .....	52
4.5.8. Invoice .....	53
4.5.9. InvoiceDetail .....	54
4.5.10. Menu .....	55
4.5.11. MenuRole .....	55
4.5.12. Message .....	56
BAB V TESTING DAN EVALUASI SISTEM.....	57
5.1. Perbandingan Antara Solusi Yang Ada .....	57
5.2. Perbandingan Dengan Menggunakan Dependency Injection .....	58
5.3. Penggunaan Unity Container .....	62
BAB VI KESIMPULAN DAN SARAN.....	65
6.1. Kesimpulan .....	65
6.2. Saran .....	65

## DAFTAR GAMBAR

Gambar 1.	<i>Error</i> pada proses dependency injection pada ASP.NET MVC Framework.....	7
Gambar 2.	<i>The dependencies for a Dependency Injector</i> (Fowler, Inversion of Control Containers and the Dependency Injection pattern, 2004) .....	12
Gambar 3.	Class Dependency and Model Complexity.....	13
Gambar 4.	Dependency Injection – Golf Analogy.....	14
Gambar 5.	Contoh <i>Class Diagram</i> .....	20
Gambar 6.	Contoh <i>Activity Diagram</i> .....	21
Gambar 7.	Rancangan Penggunaan <i>IoC Container</i> .....	29
Gambar 8.	Rancangan Penggunaan <i>Automatic Mapping</i> .....	30
Gambar 9.	Rancangan <i>Repository Pattern</i> .....	31
Gambar 10.	Tools yang akan digunakan pada <i>application framework</i> .....	32
Gambar 11.	<i>Entity Relationship Diagram</i> .....	34

## DAFTAR TABEL

Tabel I.	Notasi Simbol Pada UML.....	18
Tabel II.	Deksripsi <i>ER-Diagram</i> .....	34
Tabel III.	Tabel Perbandingan Antar Solusi .....	57