

BAB I PENDAHULUAN

Pada BAB I dibahas mengenai latar belakang dipilihnya topik ini, rumusan masalah, tujuan pembahasan, ruang lingkup kajian dan sistematika penyajian.

1.1. Latar Belakang

Saat ini perkembangan aplikasi *e-commerce* dengan *scope B2C (Business to Consumer)* berbasis web sangatlah pesat. Dengan banyaknya pilihan bahasa pemrograman saat ini pengembang dimudahkan untuk menentukan dengan apa suatu aplikasi web *e-commerce* akan dikembangkan dan dengan arsitektur apakah aplikasi tersebut dikembangkan.

Pada saat ini pengembang banyak diperkenalkan kepada arsitektur *MVC (Model-View-Controller)*. Arsitektur *MVC* merupakan sebuah solusi agar pengembang dapat membuat sebuah aplikasi yang ramah dengan kerja tim, perubahan cara kerja aplikasi pada saat proses pengembangan (*agile*) dan efisiensi waktu pengembangan.

Karena arsitektur *MVC* yang sudah sangat dikenal baik oleh para pengembang maka *Microsoft* mengembangkan sebuah *framework* dengan arsitektur *MVC* untuk *ASP.NET*.

Oleh karena teknologi pendukung untuk mengembangkan sebuah aplikasi *e-commerce* berbasis web sudah sangat baik maka pengembang dapat mengembangkan aplikasi lebih baik. Akan tetapi aplikasi web *e-commerce B2C* banyak memiliki kesamaan cara berpikir antara satu dengan yang lainnya. Maka diperlukan sebuah *framework* untuk mengakomodir kesamaan-kesamaan yang ada pada sebuah aplikasi web *e-commerce B2C*.

Pada *Microsoft ASP.NET MVC Framework* juga terdapat beberapa kekurangan yang dianggap oleh developer sangat mengganggu yaitu adalah:

1. Tidak ada *Inversion of Control (IoC) Container* yang mengakibatkan sulitnya mengembangkan aplikasi yang modular tanpa bantuan sebuah *IoC Container*.

2. Kurangnya solusi untuk mengatasi hal-hal yang bersifat umum seperti *casting* data dari entitas bisnis ke *view-model*.
3. Pada fungsi *redirect to action* ASP.NET MVC Framework menggunakan *string* sebagai parameter yang mengakibatkan seringkali terjadi kesalahan penulisan parameter oleh pengembang dan tidak terdeteksi oleh *visual studio* bahwa pengembang merujuk ke nama *controller* atau nama *view* yang salah
4. *Developer* sering membuat *controller* yang “berat” (*fat controller*) karena banyak contoh yang dibuat oleh Microsoft dituangkan pada *controller*. (Honeycutt, 2014)

Permasalahan yang sangat penting pada *native* ASP.NET MVC Framework adalah tidak adanya *native IoC(Inversion of Control)* yang mengakibatkan pengembang sulit untuk *decoupling* antar kelas dan sangat sulit untuk dilakukan *testing* pada *unit test* karena semua kelas yang dependent merupakan *concrete class* dan tidak dapat dibuat *fake* untuk dilakukan *unit test*.

Karena adanya permasalahan-permasalahan di atas maka diperlukan sebuah *application framework* yang dapat menangani *e-commerce b2c* dengan lebih *developer oriented* yang dapat mengarahkan pengembang untuk mengembangkan aplikasi yang lebih moduler dan sangat ramah dengan perkembangan kebutuhan. *Application framework* harus mencakup beberapa hal yaitu :

1. Memiliki *Inversion of Control (IoC)* yang dapat menangani *dependency* untuk setiap kelas.
2. Memiliki alat untuk melakukan *mapping* data otomatis dari model ke *view-model*.
3. Memiliki cara yang lebih baik untuk melakukan *redirect to action* yang lebih baik untuk mengurangi kesalahan pegetikan pada saat proses pengembangan.

4. Memiliki konsep *repository* pada model yang dapat mengakomodir proses data dimana proses tersebut tidak berada pada *controller*.

1.2. Rumusan Masalah

Berikut ini adalah rumusan masalah dalam tugas akhir ini berdasarkan dari latar belakang pada sub bab 1.1 :

1. Bagaimana membangun sebuah *application framework* untuk aplikasi web *e-commerce B2C*?
2. Bagaimana membangun sebuah *application framework* yang moduler?
3. Bagaimana meminimalkan penggunaan *string* pada proses pengembangan?

1.3. Tujuan Pembahasan

Berikut ini adalah tujuan pembahasan dalam tugas akhir ini berdasarkan dari rumusan masalah pada sub bab 1.2:

1. Membuat sebuah *application framework* yang dapat mengakomodir hal-hal yang *general* pada sebuah aplikasi web *e-commerce B2C*
2. Menambahkan *IoC Container* pada *application framework* agar dapat mengembangkan sebuah aplikasi yang moduler lebih mudah
3. Membuat kelas yang mengakomodir penggunaan *string-based tools* menjadi *strongly typed base*.

1.4. Ruang Lingkup Kajian

Berikut ini adalah ruang lingkup kajian dari pembahasan yang dibuat pada *application framework* ini:

1. Batasan Perangkat Keras
 - a. Prosesor Intel Pentium 4 2.0 GHz
 - b. *Memory* 1 GB

- c. *Hard drive* 20 GB
- 2. Batasan Perangkat Lunak
 - a. Sistem operasi Microsoft Windows XP Professional Service Pack 3
 - b. Sistem basis data Microsoft SQL Server
 - c. Bahasa pemrograman Microsoft Visual C#
 - d. *Editor* Microsoft Visual Studio
- 3. Batasan Aplikasi

Berikut adalah ruang lingkup kajian pada aplikasi.

 - a. Membuat sebuah *framework* bukan sebuah *content management system*.
 - b. *Template default* hanya sebuah studi kasus.
 - c. Membuat sebuah *application framework* untuk aplikasi web.
 - d. Membuat *application framework* yang mendukung *modularity*.

1.5. Sistematika Penyajian

Berikut ini adalah sistematika penyajian dari laporan “Pengembangan application framework untuk aplikasi web *e-commerce* B2C dengan menggunakan *ASP.NET MVC Framework* sebagai *framework* dasar”.

BAB I Pendahuluan

Pada BAB I pendahuluan membahas latar belakang masalah yang memuat hal-hal yang menyebabkan munculnya masalah, mengapa masalah itu penting untuk dibahas; rumusan masalah yang merupakan pokok-pokok persoalan yang ditelaah, dikupas dan dipecahkan dalam bab pembahasan, tujuan pembahasan membahas organ-organ yang memuat garis-garis besar hasil yang ingin dicapai setelah membahas masalah dalam rumusannya, ruang lingkup kajian berfungsi memerinci prinsip-prinsip teori sebagai tolak ukur dalam pembahasan dan pemecahan masalah dan sistematika penyajian.

BAB II Kajian Teori

Pada BAB II kajian teori membahas tentang landasan – landasan teori yang digunakan untuk pembuatan aplikasi ini.

BAB III Analisis dan Rancangan Sistem

Pada BAB III analisis dan rancangan sistem membahas mengenai perancangan dan analisis yang digunakan dalam pembuatan aplikasi.

BAB IV Hasil Penelitian

Pada BAB IV hasil penelitian membahas mengenai hasil yang tercapai dalam pembuatan aplikasi.

BAB V Pembahasan dan Uji Coba Hasil Penelitian

Pada BAB V pembahasan dan uji coba hasil penelitian membahas mengenai hasil testing dan evaluasi dalam pembuatan aplikasi.

BAB VI Simpulan dan Saran

Pada BAB VI simpulan dan saran membahas mengenai rangkuman hasil-hasil pembahasan dan menarik kesimpulan secara keseluruhan yang merupakan jawaban atas persoalan yang dikemukakan dalam rumusan masalah serta pendapat penyusun yang menganjurkan tindakan atas masalah yang terjadi.