

ANALISIS DAN IMPLEMENTASI TEMPLATE GENERATOR WEBSITE MENGUNAKAN METODE GENERATIVE PROGRAMMING

Kama Raditya¹, Dana Sulistyio Kusumo², Arie Ardiyanti Suryani³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Pengkodean yang berulang-ulang merupakan suatu masalah dalam pengembangan perangkat lunak, dalam konteks pembangunan perangkat lunak berbasis web, perintah-perintah yang berhubungan dengan database access layer adalah salah satu masalah yang dihadapi, dimana programmer seringkali melakukan pengkodean yang berulang-ulang. Generative programming adalah satu paradigma pemrograman yang bertujuan untuk mengurangi tingkat kesulitan dalam mengubah bahasa model menjadi kode. Tujuan dari konsep generative programming dapat dijalankan dengan jalan membangun sebuah code generator yang bertujuan untuk menghasilkan kode berdasarkan spesifikasi pengembang itu sendiri. Tugas akhir ini akan membangun sebuah active code generator sebagai implementasi dari generative programming dengan menggunakan PHP, HTML dan MySQL. Code generator akan dibangun dengan PHP dengan MySQL sebagai aplikasi database access layer.

Kata Kunci : Generative programming, code generator. Database access layer

Abstract

Repetitive coding is a problem in software development, in the context of web based application, repetitive coding is oftenly used to interact with database access layer, where programmer often used repetitive coding. Generative programming is a programming paradigm that have a couse to help reducing the gap between the model and the code. Therefor, we must build a code generator to produce codes based on its own developer spesific requirements. This final project will be developing active code generator as an implementation of generative programming using PHP, HTML and MySQL. Code generator will be developed with PHP and MySQL as its database access layer application.

Keywords : Generative programming, code generator. Database access layer



Telkom
University

BAB I

PENDAHULUAN

1.1 Latar Belakang

Masalah yang timbul dalam pengembangan perangkat lunak berbasis web belakangan ini adalah pengembang perangkat lunak menghabiskan banyak waktu pengembangan perangkat lunak untuk melakukan pengkodean yang berulang-ulang.

Generative Programming (GP) adalah sebuah paradigma pemrograman yang menghasilkan *source code* berdasarkan permintaan atau *requirements* yang spesifik. Yang tujuannya adalah mengurangi *gap* antara bahasa kode program dan konsep dari domain [5]. Yang didalamnya terdapat pemodelan, *generator* dan *output* dari sistem.

Salah satu hal yang penting dalam *generative programming* adalah pembangunan *code generator*, sebagai *tool* untuk menghasilkan *script* yang diinginkan, pembangunan *active code generator* sebagai implementasi pembangunan aplikasi dengan sudut pandang *generative programming* ingin dianalisa dan diimplementasikan dalam tugas akhir ini dengan menggunakan *template-based code generation* berbasis PHP.

Tugas akhir ini bertujuan menganalisa perangkat lunak *code generator* yang menerapkan teknik *template-based code generation* berdasarkan *input* berupa *database access layer* sebagai *metadata* yang didapat dari bahasa *Domain Specific Language* (DSL) SQL sehingga pengembang perangkat lunak yang dalam tugas akhir ini dibatasi sebagai pengembang perangkat lunak dapat menghasilkan *template website* secara sebagian maupun keseluruhan sehingga pengembang tidak perlu lagi untuk menulis banyak *source code* berulang – ulang untuk membuat suatu aplikasi berbasis web.

Adapun bahasa model yang digunakan dalam tugas akhir ini adalah SQL yang sudah banyak dikenal sebagai DSL (*Domain Specific Language*) yang spesifik untuk pemodelan *database* [6] dan menggunakan PHP sebagai bahasa dari *code generator* yang menghasilkan *template website* secara sebagian atau keseluruhan.

1.2 Perumusan Masalah

Dalam Tugas Akhir ini penulis merumuskan beberapa masalah yang timbul dari latar belakang masalah yang dipaparkan diatas, yaitu antara lain:

1. Bagaimana membuat sebuah perangkat lunak *active code generator* menggunakan metode *template-based code generation* sebagai salah satu teknik dalam *generative programming* yang dalam tugas akhir ini diimplementasikan sebagai *code generator* untuk implementasi CRUD dari website.
2. Bagaimana dan seperti apa kode yang baik yang dapat dihasilkan *generator* untuk implementasi web.
3. Bagaimana membangun dan memilih suatu *interface* pemodelan yang dapat merepresentasikan abstraksi dan semantik domain dengan baik.
4. Bagaimana memilih dan menerapkan teknik generatif yang tepat untuk *code generator*.

Untuk Menghindari meluasnya materi pembahasan tugas akhir ini maka penulis membatasi permasalahan dalam tugas akhir ini hanya mencakup hal-hal berikut

1. Dengan asumsi website kebanyakan menggunakan perintah CRUD (*Create, Retrieve, Update, Delete*) maka *generator* PHP hanya dapat meng-generate template yang menggunakan perintah-perintah tersebut.
2. *Database* yang digunakan adalah MySQL dengan *input* terbatas pada pembuatan tabel.
3. Pemodelan *Database* untuk *website generator* berasal dari *Database Access Layer* yang diimplementasikan oleh tabel-tabel dalam DBMS MySQL.
4. Analisa dan implementasi *generative programming* akan lebih terfokus kepada *active code generator* sebagai salah satu sudut pandang teknik *generative programming*.

1.3 Tujuan

Tujuan Tugas Akhir ini adalah :

1. Membangun *code generator* dengan PHP dan MYSQL yang dapat menghasilkan template kelas dalam PHP untuk pembangunan aplikasi berbasis web.
2. Membangun sebuah aplikasi yang dapat membangun sebuah template situs berdasarkan pemodelan basis data.
3. Menganalisa apakah perangkat lunak *code generator* merupakan *active code generator* sebagai implementasi *generative programming*.
4. Menganalisa apakah *generated code* yang dihasilkan dapat diterapkan untuk pembangunan *website* .

1.4 Metodologi Penyelesaian Masalah

Metode penyelesaian masalah dalam penelitian ini adalah sebagai berikut :

1. Studi Literatur
Mempelajari buku, makalah, dan referensi lain yang mendukung penyusunan tugas akhir. Studi literatur ini meliputi studi literatur tentang *generative programming*, studi literatur tentang teknik *code generation*, *object-oriented programming*, dan studi literatur tentang metode analisis *code generator*.
2. Analisis dan Perancangan
Melakukan analisis *active code generator* sebagai salah satu metode implementasi *code generator* yang menerapkan *generative programming* kemudian melakukan perancangan model dari *code generator*.
3. Implementasi dan Pengujian
Mengimplementasikan aplikasi perangkat lunak dengan menggunakan PHP. Sedangkan pengujiannya dilakukan dengan memeriksa apakah *code generator* yang dihasilkan memenuhi hal-hal yang harus dipenuhi dalam pembangunan *active code generator*.
4. Pengambilan Kesimpulan
Pengambilan kesimpulan dilakukan berdasarkan kesimpulan hasil pengujian.
5. Penyusunan Buku Laporan
Pembuatan laporan Tugas Akhir yang mendokumentasikan tahap-tahap kegiatan dan hasil dalam Tugas Akhir ini.

BAB V

PENUTUP

A. KESIMPULAN

Kesimpulan yang dapat diambil dari Tugas Akhir judul “**Analisis dan Implementasi Template generator Website menggunakan Metode Generative Programming**” adalah :

1. *Template engine*, yang dalam perangkat lunak ini diaplikasikan sebagai *code generator* dengan PHP tujuannya adalah memisahkan *business logic* dengan *presentation logic*, tidak dapat memisahkan secara total antara kode PHP dengan kode HTML.
2. Untuk tercapainya *mapping* kardinalitas secara sempurna antara *database tabel* dan kode objek yang dihasilkan *code generator* maka harus dilakukan perubahan salah satunya, sehingga dalam aplikasi web belum dapat menerapkan tujuan *active code generator* yaitu tidak mengubah kode yang ada tanpa merubah *database* ataupun kelas *model*.
3. *Generated code* yang menerapkan MVC (*Model View Controller*) dapat menerapkan sebagian tujuan dari *active code generator* dengan memisahkan antara *model* (bagian yang tidak perlu dimodifikasi user) dan *view* (bagian yang dimodifikasi user untuk tampilan).
4. *Design pattern* MVC memiliki kompleksitas yang rendah, kecuali untuk *controller* (yang mempunyai tugas untuk menangani semua *request* HTTP), memiliki kompleksitas sedang.

B. SARAN

Saran yang dapat diajukan untuk pengembangan Tugas Akhir judul “**Analisis dan Implementasi Template generator Website menggunakan Metode Generative Programming**” adalah :

1. *Metadata* yang dimaksudkan sebagai *input* dalam *active code generator* tidak hanya dapat diimplementasikan dengan *database*, tetapi untuk lebih lanjut dapat juga diimplementasikan dengan format lain (misal XML), sehingga tercipta jembatan untuk mengembangkan WTG di *platform* yang berbeda, ataupun dengan bahasa lain, keuntungan lain adalah dapat XML *schema* dapat mengidentifikasi kardinalitas antar tabel sehingga *mapping* yang sempurna dapat dicapai dengan konfigurasi user.
2. Dapat pula dikembangkan penggunaan algoritma *filtering*, untuk menyaring *template* yang ada, sehingga *file* PHP tidak tercampur dengan HTML, seperti *metaprogramming* dalam C++ ataupun penggunaan algoritma *slicing*.
3. *Mapping* dari *database* ke objek dapat pula digabungkan dengan *active record ADOdb* pada library PHP, sehingga dapat mengimplementasikan DBMS lain selain MySQL.

Daftar Pustaka

- [1] Braam, Paul A.J. 2004. Design Patterns applied to Web Programming in PHP.
- [2] Czarnecki, Krzysztof. 2002. Brief Overview of Generative Programming. DaimlerChrysler Research and Technology Ulm, Germany
- [3] DuBois, Paul. 2003. MySQL Second Edition. Sams Publisher.
- [4] Furtado, André W. B and Santos, André L. de M. 2002. Tutorial: Applying Domain-Specific Modeling to Game Development with the Microsoft DSL Tools, Microsoft Innovation Center at Recife / Informatics Center (CIn).
- [5] van Emde Boas, Ghica. 2005. Generative and Model-Driven Approaches for PHP. Bronstee.com, the Netherlands
- [6] Heering, Jan and Mernik, Marjan. 2002. Domain Specific Languages for software engineering. Proceedings of the 35th Hawaii International Conference on System Sciences.
- [7] J. McCabe, Thomas. 1976. IEEE Transactions On software Engineering, vol. SE-2, NO.4.
- [8] Krahn, Holger and Rumpe, Bernhard. 2003. Techniques Enabling Generator Refactoring. Institute for Software Systems Engineering Technische Universität Braunschweig, Braunschweig, Germany
- [9] Perugini, Saverio and Ramakrishnan, Naren. 2005. A Generative Programming Approach to Interactive Information Retrieval: Insights and Experiences. Department of Computer Science, University of Dayton,
- [10] Pohjonen, Risto and Tolvanen, Juha-Pekka. 2005. Product Derivation through Domain-Specific Modeling: Collected Experiences.
- [11] Sverdlov, Alexander and Kopec, Danny. 2004. Applied Software Synthesis. The Graduate Center, Brooklyn College.
- [12] Voelter, Markus. 2003. A Catalog of Patterns for Program Generation. Voelter - Ingenieurbüro Für Softwaretechnologie.
- [13] Herrington, Jack. Database Access. 2005. <http://freshmeat.net/articles/view/843/>. Didownload pada tanggal 27 November 2007.
- [14] Massassi, Brian. Template Engines. http://www.massassi.com/php/articles/template_engines/. Didownload pada tanggal 27 november 2007.
- [15] Pillai, AnandB. Introducing code complexity metrics. 2004. http://spikesource.com/Measuring_Code_Complexity_Metrics.htm. Didownload pada tanggal 28 November 2007.
- [16] _____. 2005. http://www.informit.com/content/images/0131472208/samplechapter/0131472208_ch06.pdf Didownload pada tanggal 3 Desember 2007.
- [17] _____. 2007. <http://en.wikipedia.org/wiki/generativeprogramming>. Didownload pada tanggal 3 Desember 2007.
- [18] _____. 2002. <http://Intensoft/technology/glossary.html>. Didownload pada tanggal 3 Desember 2007.