

GENERATOR KODE PROGRAM PHP DARI DIAGRAM UML (UNIFIED MODELING LANGUAGE) BERBASIS XML (EXTENSIBLE MARKUP LANGUAGE)

Ayu Dwi Rahman Harnani¹, Dana Sulistyo Kusumo², Retno Novi Dayawati³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Sebelum mengimplementasikan sistem berorientasi objek ke dalam tahap pengkodean biasanya dilakukan pemodelan atau **perancangan sistem terlebih dahulu untuk memudahkan** penggambaran dan representasi sistem. Pemodelan terhadap sistem berorientasi objek yang sering digunakan adalah UML (Unified Modeling Language). Saat ini terdapat banyak tool diagram UML yang dapat digunakan antara lain Metamill, Rational Rose, dan lain-lain. Tools ini sendiri menyediakan fasilitas **pengubahan class diagram menjadi beberapa bahasa pemrograman** seperti C++, Visual C++, Java, dan Visual Basic.

Tugas akhir yang berjudul "Generator Kode Kerangka Program PHP dari Diagram UML (Unified Modeling Language) Berbasis XML (eXtensible Markup Language)" ini dibangun untuk membangkitkan kode program PHP dari suatu class diagram. Karena PHP merupakan bahasa pemrograman untuk web yang perkembangannya sangat pesat dan saat ini tidak hanya mendukung pemrograman yang sifatnya prosedural, tapi juga berorientasi objek. Disamping itu, tools diagram UML yang tersedia belum ada yang mendukung pengubahan class diagram menjadi bahasa pemrograman PHP.

Output dari aplikasi dapat memudahkan programmer dalam mengimplementasikan sistem agar sesuai dengan perancangan yang telah dibuat oleh perancang sistem dengan tool diagram UML berbasis XML yang berbeda-beda. Aplikasi ini dibangun menggunakan analisa dan desain berorientasi objek dengan menggunakan bahasa pemrograman Java 1.5 dan Visual Paradigm for UML yang berbasis XML sebagai tool pemodelan sistem. Tugas Akhir ini menganalisa proses parsing (dengan metode Recursive Descent Parser) dan menganalisa data input aplikasi (dari diagram UML berbasis XML yang berbeda-beda), membangkitkan kode program serta menganalisa kebenaran kode program dan relationship antar kelas yang dihasilkan aplikasi, serta menguji performansi aplikasi dalam membangkitkan kode program berdasar jumlah kelas.

Kata Kunci : Kata kunci : UML, Class Diagram, XML, PHP

Abstract

Before we implements object oriented system into a program, we usually have to design the model of system representation more simple. Object oriented modeling design that usually used is UML (Unified Modeling Language). Now there is a lot of tools of UML digram that can used such as Metamill, Rational Rose, etc. It provides the facilities to convert class diagram into language programming such as C++, Visual C++, Java, and Visual Basic.

The project of "Generator for PHP Source Code from UML Diagram (Unified Modeling Language) Based on XML (Extensible MarkUp Language)" was build to generate PHP code from the class diagram. PHP is a programming language for web that more increase every day. Because It supports procedural and object oriented programming. Beside of that, tools of UML Diagram not support to generate PHP source code.

Output of the application can help programmer to make the system that suit with design of UML Diagram based on XML. This Application was build by object oriented analyze and design with Java 1.5 and Visual Paradigm for UML (based on XML) as a tool of modeling system. This project analyze the parsing process (with Recursive Descent Parser method) and analyze input file from different UML diagram based XML, generate PHP code from input file of application, build php code and analyze the code and relationship between class that is generated by the application, and to analyze performance of application.

Keywords : Key Word : UML, Class Diagram, XML, PHP

1. Pendahuluan

1.1 Latar Belakang

Sebelum mengimplementasikan sistem berorientasi objek ke dalam tahap pengkodean biasanya dilakukan pemodelan atau perancangan sistem terlebih dahulu untuk memudahkan penggambaran dan representasi sistem. Pemodelan terhadap sistem berorientasi objek yang sering digunakan adalah UML (*Unified Modeling Language*). Tahap yang dilakukan selanjutnya adalah pembuatan program sesuai dengan perancangan sistem yang dibuat. Namun seringkali terjadi ketidakkonsistenan antara aplikasi program dengan desain yang telah dibuat sebelumnya. Disamping itu, waktu yang digunakan untuk membuat kode program seringkali lebih banyak sehingga hanya sedikit waktu untuk desain dan menganalisa sistem yang dibangun.

Tugas akhir yang berjudul "Generator Kode Program PHP dari Diagram UML (*Unified Modeling Language*) Berbasis XML (*eXtensible Markup Language*)" ini dibangun untuk membangkitkan kode program PHP berdasarkan diagram UML berbasis XML, sehingga dapat memudahkan programmer dalam membuat kode program yang sesuai dan konsisten dengan desain yang telah dibuat sebelumnya. Selain itu, waktu pengerjaan program juga menjadi lebih singkat sehingga lebih banyak waktu yang tersedia untuk menganalisa dan mendesain sistem. Hal ini berguna untuk mencegah terjadinya kesalahan perancangan yang menyebabkan pengerjaan ulang sistem.

Tool diagram UML yang digunakan untuk memodelkan sistem adalah diagram UML yang berbasis XML yang dapat *men-generate file XMI (XML Metadata Interchange)*. *File XMI* merupakan format standar yang dibuat oleh OMG (*Object Management Group*) dan digunakan oleh berbagai *tools* diagram UML untuk mendeskripsikan model-model UML menggunakan bahasa XML [5]. Dengan demikian, aplikasi tugas akhir ini dapat membangkitkan kode program PHP dari tool diagram UML berbasis XML yang berbeda-beda.

Kode program yang dibangkitkan oleh aplikasi adalah kode program PHP, karena bahasa pemrograman web khususnya PHP saat ini berkembang pesat dan merupakan salah satu *item* dalam teknologi *object oriented*. Disamping itu, karena belum adanya *tool* diagram UML yang dapat membangkitkan kerangka kode program PHP.

Aplikasi tugas akhir ini dibangun menggunakan analisa dan desain berorientasi objek dengan menggunakan bahasa pemrograman Java 1.5, Metamill 4.1 yang berbasis XML sebagai *tool* diagram UML yang memodelkan sistem. Kode program yang *di-generate* menggunakan bahasa pemrograman PHP 5 yang mendukung *object oriented programming*, dan Zend 5.0 untuk pengecekan kebenaran kode program PHP yang dibangkitkan.

1.2 Perumusan Masalah

Dengan mengacu pada latar belakang di atas, maka permasalahan yang dibahas dan diteliti adalah sebagai berikut :

1. Melakukan dan menganalisa kemampuan aplikasi dalam melakukan parsing terhadap *file* XMI yang dihasilkan oleh berbagai macam *tool* perancangan model berorientasi objek yang mendukung *generate* file XMI. *File* XMI merupakan format standar yang dibuat oleh OMG (*Object Management Group*) dan digunakan oleh berbagai *tools* diagram UML untuk mendeskripsikan model-model UML menggunakan bahasa XML.
2. Mengecek dan menganalisa kebenaran format *file* inputan aplikasi dengan melakukan analisa leksikal, sintak, maupun semantik terhadap *file* inputan. *File* inputan aplikasi dengan format XMI memiliki ekstensi *.xmi*, *.xmmd*, *.xmmt* yang sesuai dengan standar.
3. Membangun dan menganalisa hubungan antar *class diagram* yang terbentuk pada kode program, baik berupa *association relationship*, *aggregation relationship*, *composition relationship*, *generalization relationship*, *realization relationship* dan *dependency relationship*.
4. Mengecek dan menganalisa kebenaran semantik kode program maupun *relationship* yang terbentuk pada kode program yang dibangkitkan oleh aplikasi, meliputi kelas, atribut, operasi, maupun kode dalam aplikasi.
5. Menganalisa performansi proses pembangkitan kode program yang dapat dilakukan oleh aplikasi. Yaitu berapa banyak jumlah *class diagram* yang dapat dibangkitkan kode programnya oleh aplikasi.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan dari tugas akhir ini antara lain :

1. Menerapkan metode *Recursive Descent Parser* sebagai metode parsing terhadap file XMI dari suatu model UML yang dihasilkan oleh berbagai macam *tool* perancangan model berorientasi objek, serta menganalisa proses dan performansi *parsing* terhadap *file* inputan aplikasi.
2. Menghasilkan perangkat lunak yang dapat membangun kode program berbasis PHP dari *class diagram* yang telah dibuat.
3. Membangun dan menganalisa hubungan antar *class* pada kode program, baik berupa *association relationship*, *aggregation relationship*, *composition relationship*, *generalization relationship*, *realization relationship* dan *dependency relationship*.
4. Mengecek dan menganalisa kebenaran kode program yang dibangkitkan oleh aplikasi, meliputi kelas, atribut, operasi, maupun kode dalam operasi.
5. Menganalisa performansi pembangkitan kode program yang dapat dilakukan oleh aplikasi. Performansi ini mencakup seberapa banyak jumlah diagram kelas yang dapat dibangkitkan oleh aplikasi.

1.4 Batasan Masalah

Batasan penelitian pada tugas akhir ini adalah sebagai berikut :

1. Data masukan perangkat lunak ini berupa *file* XMI. *File* ini hanya dihasilkan menggunakan *tools* diagram UML yang berbasis XML dan mendukung *generate* file XMI.
2. Aplikasi generator kode PHP ini hanya men-*generate* satu file .php untuk satu file *class diagram*.
3. Kode program PHP yang dihasilkan berdasar pada deskripsi *class diagram* yang dibuat oleh *user* pada *tools* yang mendukung *generate* file XMI, seperti *class-class*, relasi antar *class*, atribut-atribut, dan operasi-operasinya.
4. Aplikasi generator kode pada tugas akhir ini tidak melakukan perbaikan terhadap kesalahan pendefinisian atribut maupun operasi pada *class diagram*. Aplikasi generator kode akan tetap membangkitkan atribut maupun operasi yang telah dideskripsikan pada *class diagram*.
5. Penambahan ataupun pengubahan yang dilakukan terhadap kode program yang dibangkitkan tidak akan mengubah desain *class diagram* secara otomatis (pengubahan *class diagram* hanya dilakukan secara manual).

1.5 Metodologi Penyelesaian Masalah

Untuk mencapai tujuan yang dimaksud, maka metodologi yang digunakan dalam penelitian penyusunan tugas akhir ini adalah :

1. Studi Literatur
Tahap ini bertujuan untuk mencari dan mengumpulkan jurnal ilmiah yang berkaitan dengan penelitian sebagai referensi, mempelajari teori-teori dasar tentang diagram UML khususnya *class diagram*, metode parsing XML untuk file XMI (.xmi), serta cara mengkonversikan *class diagram* ke script PHP.
2. Pendalaman Materi
Mempelajari konsep *parsing* pada algoritma *Recursive Descent Parser*, UML diagram khususnya *class diagram* dengan XMI format, dan generator kode sehingga dapat menentukan tujuan yang ingin dicapai berdasarkan parameter-parameter inputan.
3. Perancangan dan Implementasi
Melakukan perancangan berorientasi objek pada perangkat lunak generator kode program PHP dari Diagram UML Berbasis XML (*Extensible Markup Language*), dan mengimplementasikan hasil perancangan menggunakan bahasa pemrograman Java 1.5. Aplikasi ini dijalankan pada computer PC.
4. Pengujian dan Analisa Hasil
 - a. Melakukan berbagai macam percobaan dengan berbagai macam format inputan. Proses parsing akan mengecek kebenaran format isi maupun ekstensi file inputan aplikasi.
 - b. Melakukan berbagai macam percobaan dengan berbagai macam bentuk hubungan, jumlah *class*, dan operasi-operasi yang dihasilkan dari perangkat lunak dengan *class diagram*.
 - c. Melakukan berbagai macam percobaan dan membandingkan hasil perangkat lunak untuk *tools* yang berbeda-beda.

5. Penyusunan Laporan Tugas Akhir

Melakukan penyusunan laporan dan penarikan kesimpulan terhadap perangkat lunak yang dibuat serta pemberian saran terhadap pengembangan perangkat lunak ini kedepannya dalam bentuk tertulis sebagai laporan penelitian.

1.6 Sistematika Penulisan

Tugas akhir ini disusun berdasarkan sistematika sebagai berikut :

BAB 1: Pendahuluan

Dalam bab ini akan dimuat latar belakang masalah, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penulisan dan sistematika penulisan

BAB 2: Dasar Teori

Pada bab ini akan dimuat beberapa teori yang berkaitan dengan bahasa pemodelan yang menggunakan UML, konsep object oriented, teknik kompilasi dan object oriented programming PHP.

BAB 3: Analisa dan Perancangan

Pada bab ini akan dimuat tentang analisa dan perancangan system perangkat lunak dengan menggunakan metode object oriented.

BAB 4: Implementasi dan Pengujian

Pada bab ini akan dimuat tentang implementasi sistem perangkat lunak meliputi lingkungan implementasi, deskripsi tahapan proses serta memuat pengujian sistem dan memeriksa hasil kebenaran pengujian.

BAB 5: Kesimpulan dan Saran

Pada bab ini akan dimuat tentang kesimpulan secara keseluruhan dan saran untuk pengembangan sistem selanjutnya.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari analisa sistem perangkat lunak yang dibangun adalah:

1. Perangkat lunak bisa membangkitkan kode program PHP dari hasil translasi suatu *class diagram* beserta relasinya (*association relationship*, *inheritance relationship (IS-A)*, *aggregation relationship (HAS-A)*, *realization relationship*, *composition relationship* dan *dependency relationship*) yang dibuat pada aplikasi perancangan diagram UML berbasis XML.
2. Dengan menggunakan metode *Recursive Descent Parser*, aplikasi dapat digunakan untuk *memparsing* file input yang berupa file XMI.
3. Hasil file php yang digenerate berupa sebuah file untuk semua *class* yang dirancang.
4. Perangkat lunak ini dapat membangkitkan kode program dari *class diagram* yang dibuat dengan tool perancangan diagram UML berbasis XML yang berbeda-beda.
5. Perangkat lunak bisa menangani kebenaran sintak dari file XMI yang diinputkan.
6. Perangkat lunak sudah diuji coba mentranslasikan 300 *class*.

5.2 Saran

Beberapa saran yang dapat dikembangkan dari tugas akhir ini diantaranya:

1. *Class diagram* juga dapat diubah menjadi bahasa pemrograman web selain php seperti ASP dan lainnya.
2. Perangkat lunak dapat membangkitkan kode program berupa file .php untuk masing-masing *class*.
3. Perangkat lunak dapat digunakan tidak hanya untuk *memparsing* file XMI yang mendeskripsikan *class diagram*, tapi juga diagram lainnya.

Telkom
University

Daftar Pustaka

- [1] Suciadi, James. *Studi Analisis Metode-Metode Parsing dan Interpretasi Semantik pada Natural Language Processing*.
<http://www.petra.ac.id/~puslit/journals/journals.php?ID=INF060702>, 2001.
Didownload tanggal 12 Agustus 2007.
- [2] Rachmat, Antonius. *Teknik Kompiler*.
<http://lecturer.ukdw.ac.id/anton/kompiler.php> , 2004. Didownload tanggal 10 September 2007.
- [3] Umrigar, Zerkis D. Algoritma recursive descent parsing.
http://www.codecadex.com/wiki/index.php?title=Recursive_descent_parsing, 1997. Didownload tanggal 10 September 2007.
- [4] Benjamin Davison, Tom Ruckle, Shawn Swanson, Brad Winkelhake. *Automated Code Generators*. <http://www-users.cs.umn.edu/~dliang/5802reports/06/Davison/codegen.pdf>, 2006. Didownload tanggal 12 Agustus 2007.
- [5] Marchal, Benoit. *Working XML: UML, XMI, and code generation*.
<http://www.ibm.com/developerworks/xml/library/x-wxxm23/>, 2004. Didownload tanggal 12 Agustus 2007.
- [6] Fowler, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition*. Addison Wesley, 2003.
- [7] Dimas, Mengenal UML (Unified Modeling Language)
<http://staffsite.gunadarma.ac.id/wsilfi/index.php?stateid=download&id=1034&part=files>, 2007. Didownload tanggal 10 September 2007.
- [8] Herrington, Jack. *Code Generation in Action*. Manning Publications Co., 2003.
- [9] Hunter, Jeffrey M. *Java Programming*.
http://www.idevelopment.info/data/Programming/java/PROGRAMMING_Java_Programming.shtml, 2007. Didownload tanggal 30 Januari 2008.
- [10] Kak, Avinash C. *Programming with Objects: A Comparative Presentation of Object-Oriented Programming with C++ and Java*. John Wiley & Sons, 2003.
- [11] Hariyanto, Bambang. *Esensi-esensi Bahasa Pemrograman Java*. Informatika Bandung, 2003.
- [12] Siswoutomo, Wiwit. *PHP Undercover Mengungkap Rahasia Pemrograman PHP*. PT.Elex Media Komputindo, 2005.
- [13] Norvell, Theodore S. *The JavaCC FAQ*.
<http://www.engr.mun.ca/~theo/JavaCC-FAQ>, 2002. Didownload tanggal 30 Januari 2008.
- [14] Nugroho, Adi, ST., MMSI. *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Object*. Informatika Bandung, 2005.
- [15] Trachtenberg, Adam. *Upgrading to PHP 5*. O'Reilly, 2004.

- [16] Utdirartatmo, Frrar. *Teknik Kompilasi*. Graha Ilmu, 2005.
- [17] Ambler, Scott W. *The Elements of UML Style*. Cambridge University Press, 2003.
- [18] Azis, Farid M,Ir,M.Kom. *Object Oriented Programming dengan PHP5*. PT.Elex Media Komputindo, 2005.
- [19] [SiAlhir](#), Sinan. *Learning UML*. O'Reilly,2003.

