**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : Divi Galih Prasetyo Putri**

**NRP : 5110100118**

**DOSEN WALI : Bilqis Amaliah, S.Kom, M.Kom**

**DOSEN PEMBIMBING : 1. Daniel Oranova Siahaan, S.Kom, M.Sc, P.D.Eng  
 2. Rizky Januar Akbar, S.Kom, M.Eng**

# JUDUL TUGAS AKHIR

“Pembuatan Kakas Pendeteksi *Unused Methods* pada Kode Program PHP dengan Framework CodeIgniter menggunakan *Call Graph*”

# LATAR BELAKANG

Sebuah aplikasi web sering mengalami perubahan, baik karena perubahan *user requirement*, peningkatan performa, penambahan fungsionalitas, atau adaptasi terhadap teknologi baru. Pada proses *maintenance* ini banyak *programmer* tidak lagi berpatokan pada desain sistem. Banyak *programmer* yang sekedar menambahkan fungsi baru untuk menggantikan fungsi lama tanpa menghapuskan fungsi yang lama. Fungsi-fungsi yang tidak terpakai tersebut akan menjadi *unused method. Unused method* akan sangat merugikan pengembang sistem. Dari sisi *maintenance*, akan sulit dilakukan pembenahan sistem karena banyak kode program yang tidak jelas kegunaannya namun masih berada dalam sistem. *Programmer* juga akan kesulitan dalam memahami kode program. Identifikasi dan penghapusan *unused method* juga akan mengurangi kompleksitas dan ukuran sistem.

Untuk menganalisis sebuah program dapat menggunakan metode *dynamic analysis* dan *static analysis*. Metode *dynamic analysis* dilakukan dengan mengeksekusi sebuah program sedangkan pada metode *static analysis* identifikasi bagian dari program yang menunjukkan *behavior* yang tidak diinginkan dapat dilakukan tanpaperlu melakukan eksekusi terhadap program. Metode *static analysis* dapat dilakukan dengan memanfaatkan beberapa teknik antara lain *model checking*, *data-flow analysis*, *abstract interpretation*, dan lain sebagainya.

Pada tugas akhir ini akan dilakukan sebuah proses analisis mengenai adanya *unused method* pada aplikasi web dengan bahasa pemrograman PHP dan *framework* CodeIgniter menggunakan *static analysis.* Metode ini akan diimplementasikan dengan memanfaatkan *call graph* antar *method* dan antar *layer. Call graph* dibangun dengan menelusuri *method-method* yang terdapat pada *controller* terleih dahulu. Setelah itu, dilihat hubungannya dengan *method-method* pada *view* dan *model*.Dari *call graph* yang terbentuk akan dapat ditelusuri dan ditampilkan *method-method* mana saja yang tidak digunakan.

# RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana membangun sebuah *call graph* antar *method* dan antar *layer* dari sebuah sistem web berbahasa pemrograman PHP dengan framework CodeIgniter?
2. Bagaimana melakukan analisis terhadap sebuah program untuk menemukan *unused method* dengan memanfaatkan *call graph* yang telah dibangun?

# BATASAN MASALAH

Permasalahan yang dibahas di dalam tugas akhir ini memiliki beberapa batasan masalah, yaitu:

1. Bahasa pemrograman yang akan digunakan untuk membangun kakas adalah Java.
2. Sistem yang akan dianalisis adalah aplikasi web dengan bahasa pemrograman PHP dan dibangun menggunakan framework CodeIgniter*.*
3. Kakas hanya melakukan deteksi terhadap *unused code* pada *class-class* MVC aplikasi web bukan pada *library* CodeIgniter.
4. Kakas dapat menampilkan call graph dan blok *method* yang dideteksi sebagai *unused method*.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tugas akhir ini dikerjakan dengan beberapa tujuan, yaitu:

1. Merancang dan membangun perangkat lunak yang dapat mendeteksi adanya *unused code* pada sebuah program menggunakan *call graph*

# MANFAAT TUGAS AKHIR

Pengerjaan tugas akhir ini dilakukan dengan harapan dapat memberikan kontribusi terhadap *programmer* dalam proses *code analysis*. Dengan adanya perangkat lunak ini diharapkan dapat membantu proses *maintenance* yang dilakukan pada sebuah sistem.

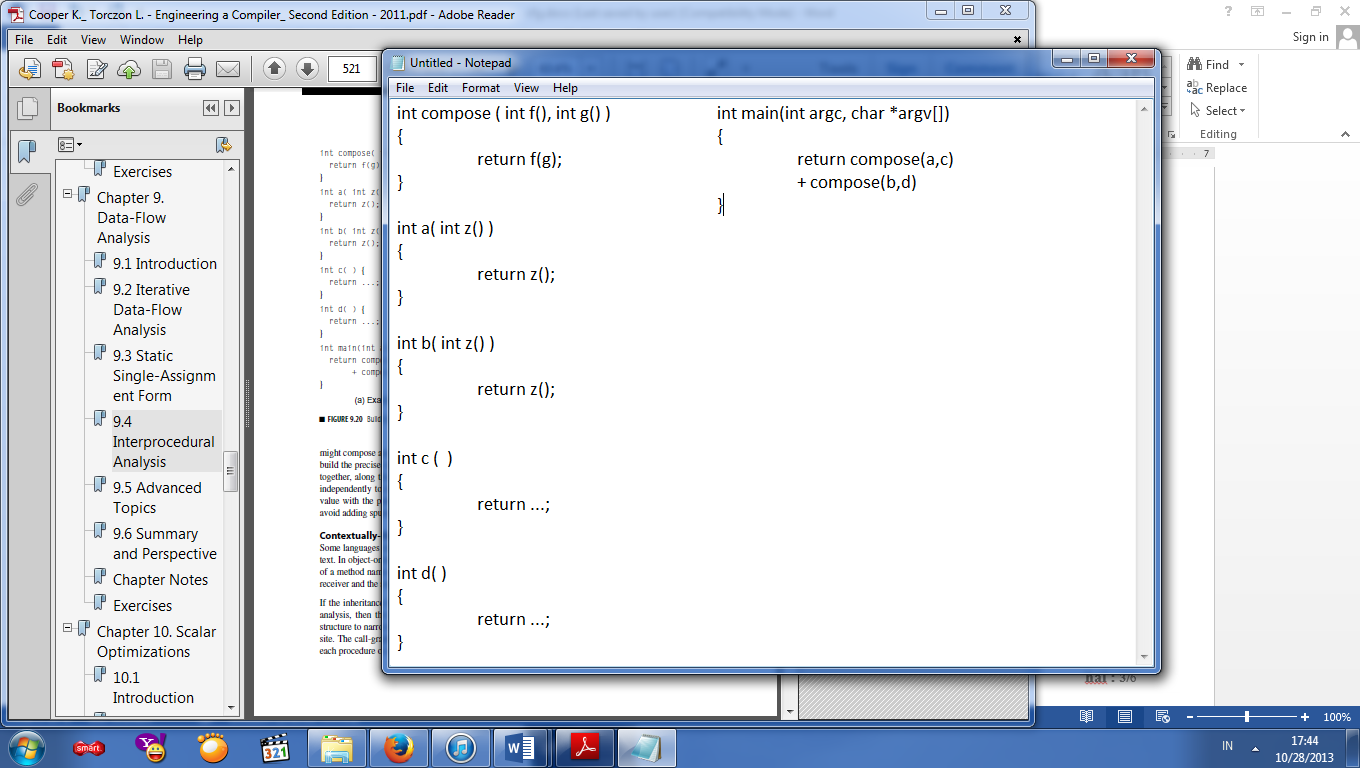
# TINJAUAN PUSTAKA

* + 1. ***Static Analysis***

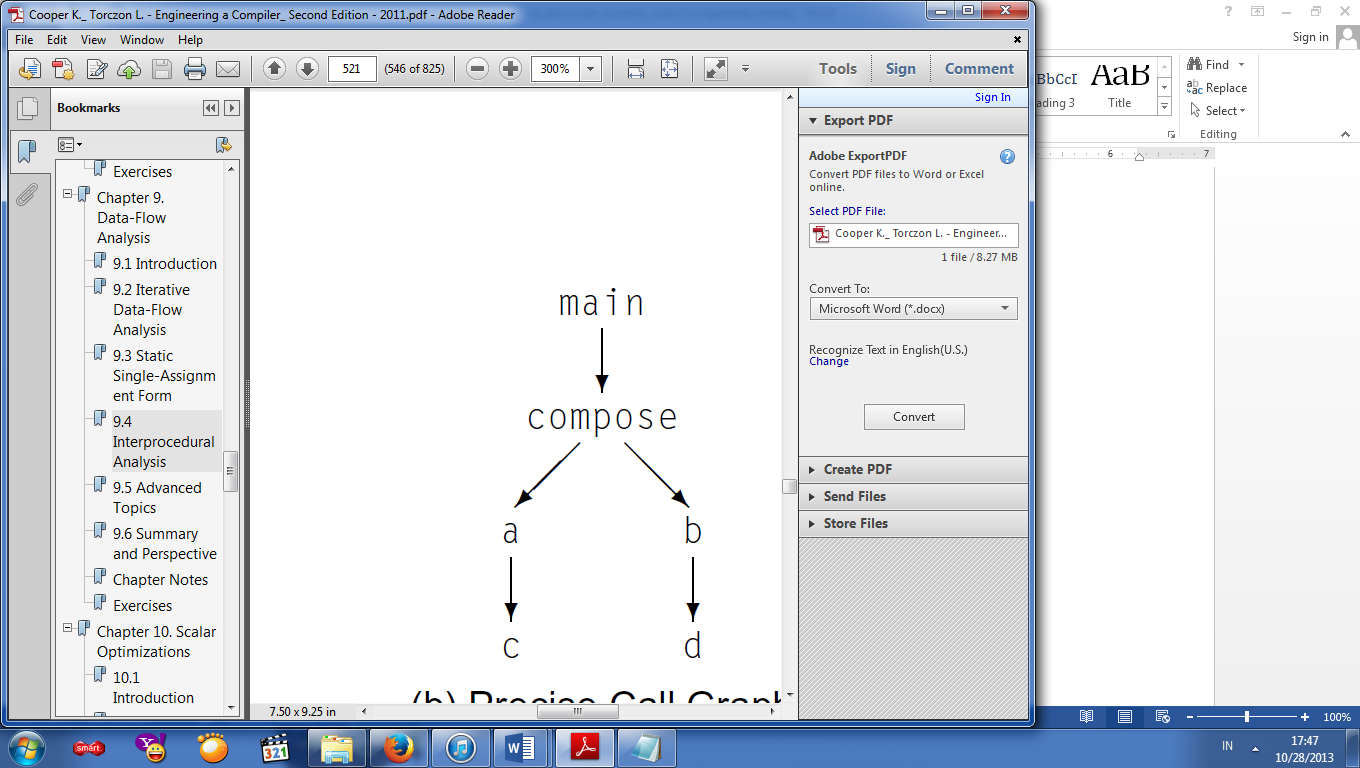
*Static analysis* adalah sebuah teknik analisis untuk mendapatkan informasi tentang jalur yang mungkin dilalui dalam sebuah eksekusi program tanpa menjalankan program tersebut. teknik ini menelusuri perilaku dari sebuah program dengan menggunakan semua *input* dan *output* yang mungkin didapatkan. [1]

* 1. ***Call Graph***

*Call graph* merepresentasikan transfer kontrol antar prosedur. Setiap *call graph* memiliki *node* sebagai prosedur dan *edge* untuk setiap pemanggilan prosedur. Setiap bahasa pemrograman memiliki tantangan tersendiri dalam pembangunan *call graph.* [2]



Gambar 1. Contoh program dalam bahasa C



Gambar 2. *Call* *Graph* dari contoh program pada Gambar 1

Gambar 1 menunjukkan contoh program dalam bahasa pemrograman C. Gambar 2 menunjukkan *call graph* dari pemanggilan prosedur yang dilakukan pada contoh program. Prinsip pembangunan model *call graph* sangat sesuai diimplemantasikan pada tugas akhir ini, karena model yang akan dibangun untuk CodeIgniter PHP web sistem karena yang akan dimodelkan adalah *method-method* pada setiap kelas.

* 1. ***Unused Code***

*Unused code* adalah kode program yang tidak pernah dieksekusi secara langsung oleh *programmer*. Penghapusan *unused code* dapat mengurangi ukuran program, mempercepat kompilasi dan eksekusi program. [2]

* 1. **CodeIgniter**

CodeIgniter adalah salah satu framework yang dapat digunakan untuk mengembangkan web dengan menggunkaan bahasa pemrograman PHP. CodeIgniter menggunakan *pattern* MVC dalam pengembangan web. *Pattern* MVC ini yang akan dijadikan dasar untuk pembuatan model *call graph* dalam tugas akhir ini. Program akan ditelusuri dari controller untuk melihat hubunganantar *method.*

# RINGKASAN ISI TUGAS AKHIR

Topik tugas akhir yang akan dikerjakan berkaitan dengan pendeteksian *unused method* pada CodeIgniter PHP web sistem dengan memanfaatkan *call graph*.



Gambar 3. Diagram Alir

Pada gambar 4 ditunjukkan proses pengolahan dari *source code* sampai dapat menghasilkan deteksi dari adanya *unused method* dalam sistem tersebut. Tahap parsing kode merupakan tahap menganalisis kode dan membangun *call graph* berdasarkan hasil analisa tersebut. Informasi yang didapat dari *source code* berupa daftar *method-method* akan dibentuk menjadi *node-node*. *Edge* dibangun berdasarkan pemanggilan prosedur antar *method*.

Tahap berikutnya adalah *traverse graph* yang dilakukan terhadap *call graph* yang dihasilkan dari tahap sebelumnya. Pada tahap ini dianalisis *node-node* yang tidak memiliki hubungan dengan dengan node lain. Node tersebut yang dimasukkan kedalam kategori *unused method*.



Gambar 4. Diagram kasus penggunaan

Penjelasan fitur berdasarkan diagram diatas adalah sebagai berikut:

1. Melihat hasil deteksi *unused* dari *source code* yang dimasukkan

Merupakan fitur yang dapat menampilkan hasil deteksi kepada pengguna berupa *method* yang dideteksi sebagai *unused*.

# METODOLOGI

## Penyusunan proposal tugas akhir

Pada proposal ini akan dibahas mengenai rencana pengembangan perangkat lunak yang akan membantu proses *maintenance* sistem dalam proses *code analysis.* Perangkat lunak yang akan dibangun akan dapat mendeteksi adanya *unused method* pada kode program. Dari kode program yang dimasukkan akan dilakukan *parsing* kedalam bentuk *call graph*. *Call graph* yang dihasilkan tersebut akan ditelusuri untuk mencari *method* yang *unused.*

## Studi literatur

Studi literatur yang digunakan untuk menunjang pengerjaan tugas akhir ini antara lain:

1. Cara membangun *Call Graph* dari kode program berbahasa PHP dengan framework CodeIgniter
2. Cara melakukan penelusuran pada *call graph*
3. Cara menganalisa *call graph* untuk mendeteksi *node* yang *unused*

## Analisis dan desain perangkat lunak

Aktor dari sistem ini adalah pengembang perangkat lunak yang sedang melakukan *code analysis* pada sistem. Kebutuhan fungsional dari sistem yang dibangun ini antara lain:

1. Memasukkan *source code* yang ingin dianalisis
2. Melihat hasil analisis deteksi *unused method*

## Implementasi perangkat lunak

Proses pembangunan perangkat lunak deteksi *unused method* ini membutuhkan beberapa kakas bantu antara lain:

1. IDE menggunakan *Eclipse kepler.*
2. *Java Development Kit* 7.0.
3. *Java Runtime Environment*.

## Pengujian dan evaluasi

Pengujian akan dilakukan dengan metode blackbox. Sistem yag dikembangkan akan diuji coba dengan memasukkan data uji berupa beberapa sistem web berbahasa pemrograman PHP dan dilakukan pengecekan apakah hasil deteksi yang dilakukan sudah sesuai.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | Tahun I | | | | | | | | | | | | | | | |
| Bulan I | | | | Bulan II | | | | Bulan III | | | | Bulan IV | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | T. Reps, M. Sagiv and R. Wilhelm, "Static Program Analysis via 3-Valued Logic," in *9th International Symposium, SAS*, Madrid, Spain, 2002. |
| [2] | K. D. Cooper and L. Torezon, Engineering a Compiler, Houston: Morgan Kaufmann, 2012. |