**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : Luluk Eko Mawati**

**NRP : 5110100150**

**DOSEN WALI : Ir. FX. Arunanto, M.Sc**

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

# JUDUL TUGAS AKHIR

“Pembuatan Kakas Pendeteksi *Unused methods* pada Kode Program *Ruby* Menggunakan *Control Flow Graph*”

# LATAR BELAKANG

Perangkat lunak memiliki tahapan pengembangan dan *maintenance*. Salah satu aktivitas yang dilakukan pada saat *maintenance* adalah menganalisis kode program, melihat *bugs* dan *error*, dan aktivitas analisis kode program lainnya. Proses ini merupakan salah satu proses yang membutuhkan waktu yang lama. Salah satu contoh kesalahan yang ada dalam kode program adalah *unused code*. Kode program ini ada, padahal tidak pernah dipanggil dalam eksekusi. Hal ini dapat mengurangi waktu komputasi pada program.

Salah satu bentuk visualisasi dalam *static analysis* adalah *Control Flow Graph* (CFG). CFG ini memodelkan program ke dalam graph. Dari graph tersebut dapat diketahui bagian dari kode program yang tidak terhubung ke node yang lain, yang berarti kode tersebut tidak digunakan.

Oleh karena itu dalam tugas akhir ini penulis menggagas untuk membuat kakas pendeteksi *unused code* pada kode program *ruby* menggunakan *Control Flow Graph* (CFG). Harapannya dengan kakas pendeteksi ini dapat membantu *programmer* untuk mengetahui *unused methods* yang ada dalam kode program *ruby* sehingga dapat mempermudah dalam proses analisis program.

# RUMUSAN MASALAH

Permasalahan yang diangkat dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. bagaimana cara memodelkan kode program *ruby* ke dalam bentuk *control flow graph*?
2. bagaimana cara mendeteksi *unused methods* menggunakan  *control flow graph*?

# BATASAN MASALAH

Penyusunan tugas akhir ini memiliki batasan-batasan masalah sebagaimana berikut:

1. kakashanya digunakan untuk menganalisis program yang menggunakan bahasa pemrograman *ruby*,
2. kakas hanya mendeteksi *unused methods* pada program masukan,
3. kakashanya memberikan visualisasi *control flow graph* dan menunjukkan bagian program yang merupakan *unused methods*

# TUJUAN PEMBUATAN TUGAS AKHIR

Adapun tujuan pembuatan tugas akhir ini adalah membuat kakas pendeteksi *unused methods* yang dapat membantu pengembang dalam memahami perangkat lunak yang menggunakan bahasa pemrograman *ruby*.

# MANFAAT TUGAS AKHIR

Manfaat pembuatan tugas akhir ini adalah membantu pengembang untuk mengetahui *unused methods* dalam sebuah proyek perangkat lunak yang menggunakan bahasa pemrograman *ruby*.

# TINJAUAN PUSTAKA

## Rancang Bangun Perangkat Lunak

Rancang bangun merupakan tahapan-tahapan untuk mendesain dan membangun suatu perangkat lunak tertentu berdasarkan kebutuhan-kebutuhan yang harus terpenuhi oleh sistem tersebut.

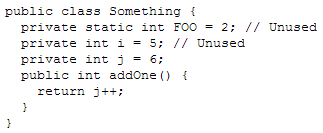
Rancang bangun perangkat lunak diperlukan untuk menentukan konsep, strategi, dan praktik yang baik diterapkan untuk menciptakan perangkat lunak yang berkualitas tinggi, sesuai anggaran biaya, mudah dalam pemeliharaannya, serta tidak membutuhkan waktu yang lama dalam pembangunannya [1].

## *Static Analysis*

*Static analysis* merupakan teknik untuk mendapatkan informasi bahwa program berjalan selama eksekusi, tanpa menjalankan program dengan masukan tertentu [2]. Static analysis bertujuan untuk mengumpulkan informasi yang dibutuhkan untuk mengalokasikan kesempatan untuk optimisasi dan menentukan optimisasi tersebut. Static analysis digunakan untuk menentukan dimana transformasi optimasi dapat diimplementasikan dengan aman dan menguntungkan. Static analysis adalah teknik dimana compiler membuktikan keamana dan keuntungan dari trasnformasi yang potensial [3].

## *Unused Code*

Dalam kode program terdapat kode-kode yang sebenarnya tidak dipakai dalam proses eksekusi. Kode-kode seperti ini mengambil alokasi memori, padahal dari segi penggunaan kode tidak ada. *Unused code* merupakan kode program yang tidak pernah dijalankan karena kode tersebut tidak berpengaruh dan tidak digunakan dalam logika program. Contoh *unused code* dapat dilihat pada Gambar 1. Baris 2 dan dan baris 3 tidak digunakan dalam eksekusi program.

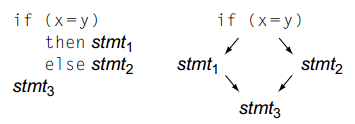
****

Gambar 1. *Unused code* pada kode program c#

## Control Flow Graph (CFG)

Unit paling sederhana pada aliran kontrol adalah blok dasar. Blok dasar adalah urutan operasi yang selalu mengeksekusi bersama-sama, meskipun operasi yang menyebabkan eksepsi. Kontrol selalu memasuki blok dasar pada operasi pertama dan keluar pada operasi terakhir.

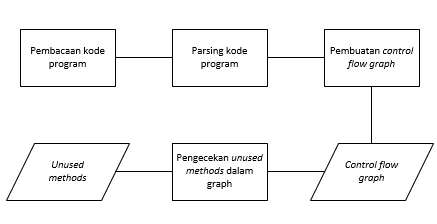
*CFG* memodelkan aliran kontrol dari blok-blok dasar program. CFG adalah graph berarah, *G=(N,E)*. Masing-masing *node* berkorespondensi pada blok dasar, *n ϵ N*. Masing-masing *edge* *e=(ni,nj) ϵ E* berkorespondensi pada kemungkinan perpindahan kontrol dari blok ni ke blok nj [4]. Contoh CFG dapat dilihat pada Gambar 2.



Gambar 2. *Asyclic* *Control Flow Graph*

# RINGKASAN ISI TUGAS AKHIR

Dalam tugas akhir ini akan dibuat kakas untuk mendeteksi *unused methods* pada kode program *Ruby* menggunakan *control flow graph*. Berikut diagram alir kakas pendeteksi *unused methods* dapat dilihat pada Gambar 3.



Gambar 3. Diagram alir kakas pendeteksi

Adapun deskripsi kakas adalahsebagai berikut.

1. Pengguna memasukkan kode program *ruby*

Pengguna memasukkan kode program *ruby* ke dalam kakas untuk dianalisis *unused methods*

1. Kakas memodelkan kode program *ruby* dalam bentuk CFG

Setelah kakas mempeeroleh masukan kode program dari pengguna, selanjutnya kode tersebut dimodelkan ke dalam bentuk CFG untuk dianalisis. Dari mode CFG yang dihasilkan dianalisis mana-mana bagian *unused methods* yang ada pada program. CFG ini kemudian ditampilkan pada antarmuka pengguna.

1. Kakasmenampilkan hasil analisis berbentuk CFG dan menunjukkan bagian *unused methods*

CFG hasil pemodelan ditampilkan kepada pengguna, sehingga pengguna dapat mengetahui model CFG dari program yang dimasukkan.

1. Pengguna melihat *unused methods* hasil analisis oleh kakas

Berikut diagram kasus penggunaan kakas dapat dilihat pada Gambar 4.



Gambar 4. Diagram kasus penggunaan kakas

# METODOLOGI

## Penyusunan proposal tugas akhir

Pada tahapan ini penulis menyusun proposal tugas akhir sebagai langkah pertama dalam penyusunan tugas akhir. Dalam hal ini, penulis menggagas tugas akhir untuk membuat kakaspendeteksi *unused methods* pada kode program *ruby* menggunakan *control flow graph*.

## Studi literatur

Literatur yang digunakan dalam pengerjaan tugas akhir ini antara lain:

1. buku,
2. jurnal,
3. literatur pendukung

Adapun studi literatur ini digunakan untuk mendapatkan informasi sebagai berikut:

1. pengertian dan contoh *unused methods,*
2. teori *control flow graph*,
3. bahasa pemrograman *ruby*,
4. proses pembacaan kode program *ruby* menjadi bentuk *control flow graph*,
5. implementasi *control flow graph* pada pembangunan kakas bantu pendeteksi *unused methods*

## Analisis dan desain perangkat lunak

Pada tahapan ini, penulis melakukan analisis dan desain kakas. Analisis kebutuhan dan perancangan kakas, dilakukan untuk merumuskan spesifikasi kebutuhan kakas, mendapatkan kebutuhan-kebutuhan dalam pemodelan *control flow graph.* Adapun tahap desain dilakukan untuk memodelkan hasil analisis dalam bentuk desain arsitektur, diagram kasus penggunaan, diagram sekuensial, kelas-kelas, desain antarmuka dan desain-desain pendukung lain dalam pembangunan kakas.

## Implementasi perangkat lunak

Pada tahap implementasi ini, analisis dan desain diimplementasikan dalam sumber kode. Adapun kakas bantu yang dipakai adalah:

1. IDE *Eclipse*.
2. *Java Development Kit* 7.0
3. *Java Environment*
   1. **Pengujian dan evaluasi**

Tahapan ini digunakan untuk melakukan pengujian dan evaluasi pada kakas. Tahapan ini bertujuan untuk mengetahui kesalahan-kesalahan yang ada pada kakas sehingga dapat dilakukan perbaikan terhadap kakas itu sendiri. Pengujian menggunakan metode *white box* dengan uji kasus beberapa kode program web dengan bahasa pemrograman *ruby*.

## 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 1 | | | | | | | | | | | | | | | | |
| Bulan 1 | | | | Bulan 2 | | | | Bulan 3 | | | | Bulan 4 | | | | |
| Studi literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | I. Sommerville, Software Engineering Ninth Edition, United States of America: Pearson Education Inc., 2011. |
| [2] | M. S. R. W. Thomas Reps, "Static Program Analysis via 3-Valued Logic". |
| [3] | M. I. S. Anders Moller, Static Program Analysis, Denmark: Department of Computer Science Aarhus University, 2012. |
| [4] | L. T. Keith D. Cooper, Engineering a Compiler Second Edition, United States of America: Elsevier, Inc., 2012. |