



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil implementasi dan uji coba yang telah dilakukan, simpulan dari penelitian ini adalah sebagai berikut.

1. Algoritma XTA telah berhasil diimplementasikan dalam sebuah aplikasi plugin Eclipse, yang dapat *generate* sebuah *static call graph* dari *source code* berbahasa pemrograman JAVA dan dapat menunjukkan relasi antar fungsi, sehingga dapat memberikan manfaat bagi penggunanya.
2. Berdasarkan hasil respon dari uji *user experience*, aplikasi *plug in* ini dinilai membantu dalam memberikan kemudahan kepada para penggunanya dalam membaca alur program. Selain itu aplikasi ini juga dinilai membantu dalam pengembangan aplikasi JAVA di IDE Eclipse.
3. Algoritma XTA memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan algoritma RA dalam menghasilkan *call graph*, dilihat dari *node* dan *edge* yang dihasilkan. Terutama dalam *project* dengan *source code* yang berukuran besar, algoritma XTA pada hasil uji cobanya mampu mempertahankan tingkat akurasinya.
4. Algoritma XTA membutuhkan waktu dan mengkonsumsi memori yang cukup besar dalam melakukan proses *generate call graph*, terutama pada *source code* yang berukuran besar.

5. Algoritma RA lebih baik dalam penggunaan memori dan waktu pemrosesan. Namun, algoritma ini memiliki akurasi yang buruk karena bersifat *name-based resolution*. Ini membuat algoritma RA cocok digunakan untuk menganalisa *source code* yang tidak banyak menggunakan nama yang sama pada setiap *method*-nya.

5.2 Saran

Beberapa saran yang dapat dijadikan pedoman untuk penelitian selanjutnya yaitu sebagai berikut.

1. Meskipun algoritma XTA dirancang sebagai algoritma *generate call graph* yang memiliki unsur skalabilitas, konsumsi memori dari algoritma ini tergolong cukup tinggi. Hal ini bisa dijadikan pokok pembahasan dalam penelitian selanjutnya.
2. Dikarenakan keterbatasan akses dari *AST node* yang tidak bisa mengakses *library (.jar)*, akurasi dari *callgraph* pun menurun. Keterbatasan akses ini dapat dijadikan topik pembahasan dalam penelitian selanjutnya.
3. Untuk penelitian selanjutnya, bisa dilakukan peningkatan interaksi dan antarmuka dari aplikasi ini. *Call graph* bisa menjadi antarmuka utama dalam IDE, dimana pembuatan, pengembangan *source code* dilakukan berdasarkan *call graph*. Juga *call graph* bisa digenerate secara otomatis, sepanjang pengembangan *source code* (tidak berupa laporan). Sehingga bisa membuat gaya baru dalam dunia pemrograman.
4. Aplikasi *plug in* ini terbatas pada *JAVA project*, dikarenakan dibutuhkannya sebuah *entry point* dalam algoritma XTA. Dalam penelitian

selanjutnya, bisa dipelajari *behaviour* dari berbagai jenis *project* lainnya, sehingga algoritma XTA ini bisa diadaptasi ke *project* jenis lain.

