pISSN: 2087-8893
eISSN: 2527-3671

Kakas bantu perhitungan nilai kopling menggunakan *conceptual coupling metrics*

Laras Husna Aulia[1)], Fajar Pradana[2)] dan Bayu Priyambadha[3)]
[1,2,3)]Program Studi Informatika, Fakultas Ilmu Komputer Universitas Brawijaya
Jalan Veteran No 8 Malang, Jawa Timur, Indonesia 65145
e-mail: husnaaulialaras@yahoo.co.id[1)], fajar.p@ub.ac.id[2)], bayu_priyambadha@ub.ac.id[3)]

Info Artikel
Diserahkan 2 November 2016, direvisi 21 Februari 2017, diterima 20 Maret 2017, tersedia *online* 12 Desember 2017

ABSTRAK

Kualitas perangkat lunak dapat diukur dengan nilai kopling pada fase perancangan. Tingkat kopling pada perancangan mengindikasikan seberapa besar hubungan antar komponen dalam sebuah sistem. Untuk menentukan nilai kopling dapat menggunakan empat parameter metrik kopling konseptual antara lain *Conceptual Similarity be-tween Methods* (CSM), *Conceptual Similarity between a Method and a Class* (CSMC), *Conceptual Similarity between two Classes* (CSBC) dan *Conceptual Coupling of Class* (CoCC). Perhitungan nilai kopling dapat dilakukan dengan cara mengambil *method-method* pada klas kemudian dihitung nilai kecocokan antar dokumen kode sumber dengan menggunakan *Latent Semantic Indexing* (LSI). Metode LSI digunakan untuk menghitung kecocokan antar dokumen kode sumber berdasarkan kesamaan kata maupun kesamaan makna kata, kemudian nilai tersebut digunakan untuk menghitung nilai metrik kopling konseptual. Pengujian akurasi pada kakas bantu ini dilakukan dengan membandingkan dengan kakas bantu lain sebagai pembanding. Pengujian dikelompokan ke dalam 3 (tiga) nilai toleransi yang berbeda yaitu 0,05; 0,1 dan 0,20. Penggunaan nilai toleransi yang berbeda disebabkan karena terdapat selisih antara kakas bantu yang dikembangan dengan kakas bantu pembanding. Hasil pengujian akurasi dengan nilai toleransi sebesar 0,05; didapatkan nilai akurasi sistem sebesar 24,83%. Jika nilai toleransi sebesar 0,1 maka nilai akurasi sistem sebesar 43,41%. Sedangkan, jika nilai toleransi sebesar 0,20 maka nilai akurasi yang diperoleh diatas 50% yaitu 69,78%.

Kata kunci: kopling, kualitas, *Latent Semantic Indexing* (LSI), metrik kopling konseptual *parsing method*, perangkat lunak.

ABSTRACT

Coupling is one of the parameter of quality of software. Coupling is a paramater for measuring how much the relationship between components of a system. To determine the value of coupling can use four of the parameters of conceptual coupling metrics among other Conceptual Similarity between Methods (CSM), Conceptual Similarity between a Method and a Class (CSMC), Conceptual Similarity between two Classes (CSBC) and Conceptual Coupling of Class (CoCC). Before counting the value of the coupling, parsing method carried out using spoon library and computation macth between source code documents to be done by using a method of information retrieval that is Latent Semantic Indexing (LSI). The use of LSI method because this research requires the calculation of a match between the document's source code based on common words or similarity of meaning of the word. After that, the LSI value is used to calculate the Conceptual Coupling Metric Testing accuracy of these tools is done by comparing with other systems as a comparison.. Testing is classified into 3(three) different tolerance values are 0.05, 0.1 and 0.20. The use tolerance values differ, because there is a difference between systems developed and the comparator system. If tolerance value is 0.05 then the accuracy value is 24,83%. If tolerance value is 0.1 then the accuracy value is 43,41%. Meanwhile, If tolerance value is 0.2 then the accuracy value is up to 50%, it is 69,78%.

Keywords: *Conceptual coupling metrics, coupling, Latent Semantic Indexing,* LSI, *parsing method.*

I. PENDAHULUAN

KOPLING adalah suatu parameter untuk mengukur seberapa besar hubungan antar komponen dalam sebuah sistem [1]. Kopling juga dikaitkan dengan parameter untuk pemeliharaan perangkat lunak. Pengukuran kopling dapat dilakukan dengan berbagai cara, salah satunya adalah dengan menggunakan satu set langkah perhitungan nilai kopling yang disebut dengan kopling konseptual. Di mana pada konsep ini, pengukuran dilakukan dengan cara mengambil informasi semantik yang dibagi antar elemen pada kode sumber [1].

Salah satu hal yang melatarbelakangi penelitian ini adalah tidak adanya pengukuran kopling yang jelas dalam mengembangkan sebuah perangkat lunak. Pengukuran tingkat kopling yang rendah sering membuat ambigu ketika para pengembang ingin merancang suatu sistem. Selama ini, besaran nilai kopling masih diukur dengan cara perkiraan saja sehingga tidak ada besaran nilai kopling yang jelas.

Terdapat teknik dalam menentukan besaran nilai kopling pada rancangan perangkat lunak agar memudahkan penggunaan kembali sistem tersebut menggunakan komponen Java yang diambil dari mesin pencari [2]. Langkah-langkah dalam menentukan sejauh mana tingkat penggunaan ulang komponen Java

tersebut digunakan adalah dengan cara mengidentifikasi komponen-komponen fungsionalitas yang diperlukan. Kemudian langkah berikutnya dengan menilai kedua *class* tersebut apakah terdapat ketidakcocokan dan *class* tersebut mudah diadaptasi untuk sistem yang lebih kompleks [2]. Suatu *class* dianggap berhubungan ke *class* lain jika kelas tersebut mengakses satu atau lebih variabel dari kelas lain atau setidaknya memanggil salah satu method dari kelas lain [2].

Teknik pengelompokan data dapat diterapkan untuk mengetahui tingkat ketergantungan *class* pada perancangan *class* diagram [3]. Komponen perangkat lunak yang memiliki karakteristik yang sama dan dipadukan dengan pola kopling yang sering terjadi. Pengelompokkan klas menggunakan *k-mean clustering* dan *cosine similarity*. Pada akhirnya, hasil klusterisasi tersebut diberi peringkat untuk mengetahui komponen-komponen yang dapat digunakan kembali [3]. Selain itu, juga terdapat metode pengukuran nilai kopling dengan menggunakan 4 (empat) parameter *conceptual coupling metrics* antara lain antara lain *Conceptual Similarity between Methods* (CSM), *Conceptual Similarity between a Method and a Class* (CSMC), *Conceptual Similarity between two Classes* (CSBC) dan *Conceptual Coupling of Class* (CoCC) [1].

Dari penelitian yang telah dilakukan sebelumnya, belum terdapat sebuah kakas bantu yang dapat membantu dalam menghitung nilai kopling secara otomatis. Perhitungan nilai kopling secara manual membutuhkan waktu yang lama apabila jumlah klas yang dihitung lebih dari 1, karena banyak hal yang perlu dipertimbangkan. untuk menentukan nilai kopling dapat menggunakan empat parameter Untuk menghitung nilai-nilai tersebut diperlukan *parsing* pada kode sumber yang selanjutnya akan diterapkan metode temu kembali informasi. Metode temu kembali informasi yang digunakan adalah LSI (*Latent Semantic Indexing*), LSI digunakan untuk memodelkan dan menganilisis informasi semantik antara kode sumber yang diuji. Penelitian ini akan mengembangkan sebuah aplikasi yang akan membantu pengembang perangkat lunak dalam mengukur nilai kopling secara otomatis dengan menggunakan metrik kopling konseptual. Pengujian akurasi perhitungan pada kakas bantu dilakukan dengan membandingkan dengan kakas bantu lain.

II. LANDASAN KEPUSTAKAAN

A. Kopling

Kopling adalah sebuah pengukuran yang berdasarkan dari kekuatan hubungan dari komponen satu ke komponen lainnya [4] [5]. Komponen dalam konsep *object oriented* disebut *class diagram*. *Class diagram* yang nantinya bisa dijadikan objek dikatakan memiliki hubungan antar komponen jika memiliki kriteria sebagai berikut [4]

1. Ada sebuah *message* yang dilewati oleh objek maka objek ini dikatakan ada keterkaitan.
2. Jika *class-class* tersebut mendeklarasikan dan menggunakan *method* atau atribut dari *class* lain.
3. Jika *class-class* tersebut memiliki relasi *inheritance* yang secara signifikan kuat antara *superclass* dengan *subclass*-nya. *Inheritance* adalah hubungan atau relasi antara dua *class* atau lebih di mana kedua *class* berbagi perilaku yang sama.

Kopling merupakan salah satu parameter kualitas perancangan perangkat lunak untuk memudahkan *developer* perangkat lunak dalam melakukan perawatan perangkat lunak. Perancangan perangkat lunak yang baik adalah yang memiliki kopling serendah mungkin. Hal ini dikarenakan jika kopling rendah maka pada saat perubahan tidak banyak komponen lain ikut berubah [1].

B. *Conceptual Coupling*

Conceptual coupling atau kopling konseptual adalah sebuah dimensi pengukuran kopling yang baru untuk melengkapi pengukuran kopling yang sudah ada sebelumnya. Kopling konseptual menghitung tingkat kopling antar komponen berdasarkan informasi semantik yang dibagi antar elemen pada kode sumber program. Kopling konseptual juga digunakan untuk memperbanyak ragam pengukuran kopling [1]. Kopling konseptual terdiri dari empat parameter yang saling berkaitan dalam proses perhitungannya. Keempat parameter tersebut antara lain sebagai berikut [1]:

1) *Conceptual Similarity between Methods (CSM)*
Parameter *Conceptual Similarity between Methods* (CSM) menghitung nilai kemiripan antara *method* satu dengan *method* lainnya maupun *method* itu sendiri. Pengukurannya berdasarkan nilai vektor yang dimiliki masing-masing *method*. Nilai CSM sama dengan nilai *cosine similarity* yang dihasilkan pada perhitungan LSI. Perhitungan CSM dirumuskan pada Rumus 1,

$$CSM\ (m_k, m_j) = \frac{vm_k^T vm_j}{|vm_k|_2\ x\ |vm_j|_2} \quad (1)$$

2) *Conceptual Similarity between a Methods and a Class (CSMC)*
Setelah mendapatkan hasil perhitungan CSM maka langkah selanjutnya adalah menggunakan hasil CSM ke dalam perhitungan parameter *Conceptual Similarity between a Methods and a Class* (CSMC).

Parameter CSMC menghitung nilai kemiripan antara *method* dengan *class*. Perhitungan CSMC dirumuskan pada Rumus 2,

$$CSMC(m_k, c_j) = \frac{\sum_{q=1}^{t} CSM^1(m_j, m_{jq})}{t} \quad (2)$$

3) *Conceptual Similarity between Two Classes (CSBC)*
Parameter *Conceptual Similarity between Two Classes* (CSBC) menghitung nilai kemiripan antara dua *class*. Perhitungan CSBC dirumuskan pada Rumus 3,

$$CSBC(c_k, c_j) = \frac{\sum_{l=1}^{r} CSMC(m_{kl}, c_j)}{r} \quad (3)$$

4) *Conceptual Coupling of a Class (CoCC)*
Parameter *Conceptual Coupling of a Class* (CoCC) menghitung nilai kopling konseptual. Perhitungan CoCC dirumuskan pada Rumus (4),

$$CoCC(c) = \frac{\sum_{i=1}^{n} CSBC(c, d_i)}{n-1} \quad (4)$$

C. *Information Retrieval*

Information retrieval digunakan sebagai metode bantuan pencarian informasi. Pencarian informasi dilakukan dengan cara mencari informasi yang juga memuat *query*. *Query* adalah kalimat atau kata yang menjadi acuan untuk pencarian informasi [6]. Informasi yang relevan dengan *query* akan ditampilkan sebagai hasil cari informasi. Kategori informasi yang relevan tersebut terdiri dari dua syarat sebagai berikut:

1. Memuat kata atau kalimat yang sama dengan *query*, atau,
2. Memuat kata atau kalimat yang bermakna sama dengan *query*

Makna kata sama dalam hal ini dilihat dari dua pandangan, yaitu sinonim dan polisemi. Sinonim adalah istilah untuk kata yang memiliki makna serupa atau sama. Sedangkan polisemi adalah istilah bagi kata yang sama namun memiliki makna berbeda. Salah satu contoh kata yang memiliki polisemi adalah "membajak". Membajak dalam kalimat "membajak sawah" berarti mengolah tanah di sawah agar dapat segera ditanami tumbuhan. Berbeda halnya dengan kalimat "membajak pesawat" di mana pada kalimat kata "membajak" berarti mengambil alih [6].

Query bersifat tidak terstruktur karena *query* merupakan kalimat yang diekspresikan oleh pengguna. Dokumen adalah salah satu contoh informasi yang tidak terstruktur [6] [7].

D. *Latent Semantic Indexing* (LSI)

Salah satu contoh metode pencarian informasi dari segi kecocokan makna antara *query* dengan informasi adalah metode LSI. Metode LSI bekerja dengan cara mencari dan menemukan informasi dari sebuah dokumen tidak hanya memperhatikan kesamaan kata namun kesamaan makna kata per kata [6].

Langkah-langkah perhitungan LSI secara umum diawali dengan menghitung *transpose* matriks, menghitung determinan dari matriks yang sudah di *transpose*, mencari nilai *eigen value*, menghitung nilai *singular value*, menghitung *eigen vektor*, normalisasi *eigen vektor*, membuat matriks *V*, membuat matriks *U*, mereduksi dimensi SVD dengan mengurangi ukuran dimensi sebanyak *k* (dalam penelitian ini $k = 5$), menghitung *cosine similarity* (tingkat kemiripan antar dokumen), memasukkan *query* oleh pengguna, *query* dicocokkan dengan dokumen *method* yang ada dan langkah terakhir adalah perankingan kemiripan antara *query* dengan dokumen berdasarkan nilai *cosine similarity*. Perangkingan diurutkan mulai dari nilai *cosine similarity* tertinggi hingga terendah [6].

Penelitian ini menganalisis dan menghitung nilai kopling berdasarkan kesamaan kata maupun kesamaan makna kata dalam dokumen *source code*. Oleh sebab itu, penggunaan metode LSI untuk pencarian dalam *souce code* pada penelitian ini dianggap cocok karena metode LSI menemukan dan mencari kembali informasi dari sebuah dokumen dengan cara memperhatikan kesamaan kata maupun kesamaan makna kata.

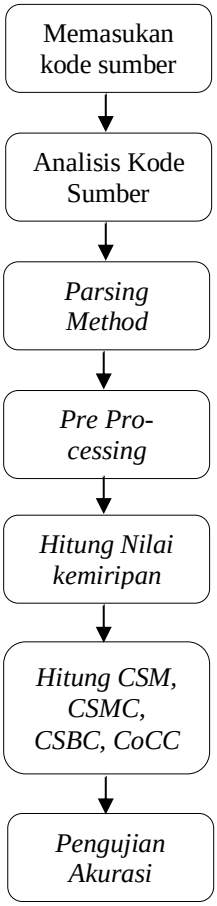
Terkait dengan studi literatur telah dilakukan sebuah penelitian dengan objek yang berbeda, metode yang dimodifikasi dan analisis hasil yang berbeda. Alur penelitian akan dijelaskan pada bagian metode dan hasilnya akan dipaparkan dalam bagian hasil dan diskusi.

III. METODE PENELITIAN

Pada Gambar 1 dijelaskan bahwa alur kerja sistem dimulai dari memasukkan kode sumber. Kode sumber yang akan dianalisis adalah kode sumber berisi beberapa *class* uji. Setelah itu kode sumber tersebut akan dilakukan *parsing* untuk mendapatkan nama *class*, *method*, dan *body method*.

Setelah didapatkan komponen-komponen tersebut maka langkah berikutnya adalah dilakukan *pre-processing*. Pada proses ini akan dihapus tanda baca, *symbol*, kata imbuhan, dan kata penghubung, kemudian akan disimpan didalam *file* berekstensi .txt. Perhitungan kemiripan dokumen menggunakan LSI dan diakhiri dengan perhitungan *coupling metric* menggunakan empat parameter CSM, CSMC, CSBC dan CoCC.

Pengujian akurasi perhitungan pada penelitian ini adalah dengan cara melakukan analisis perbandingan antara hasil perhitungan implementasi kakas bantu dan hasil perhitungan dari sebuah aplikasi pembanding perhitungan LSI yaitu *lsa.colorado.edu*.



Gambar 1. Metodologi Penelitian

Setelah didapatkan komponen-komponen tersebut maka langkah berikutnya adalah dilakukan *pre-processing*. Pada proses ini akan dihapus tanda baca, *symbol*, kata imbuhan, dan kata penghubung, kemudian akan disimpan didalam *file* berekstensi .txt. Perhitungan kemiripan dokumen menggunakan LSI dan diakhiri dengan perhitungan *coupling metric* menggunakan empat parameter CSM, CSMC, CSBC dan CoCC. Pengujian akurasi perhitungan pada penelitian ini adalah dengan cara melakukan analisis perbandingan antara hasil perhitungan implementasi kakas bantu dan hasil perhitungan dari sebuah aplikasi pembanding perhitungan LSI yaitu *lsa.colorado.edu*.

IV. HASIL DAN PEMBAHASAN

Pengujian akurasi perhitungan dilakukan untuk mengetahui tingkat akurasi kakas bantu yang telah dibangun. Konsep pengujian akurasi perhitungan pada penelitian ini adalah dengan cara melakukan analisis perbandingan antara hasil perhitungan kakas bantu dan hasil perhitungan dari sebuah kakas bantu pembanding dalam hal menghitung LSI yaitu *lsa.colorado.edu.*

TABEL I
HASIL PERHITUNGAN LSI DAN CSM

Document	inOrder	insertRoot	isEmpty	isiCari	Pencarian	PostOrder	preOrder	*print*
main	0.13	0.10	0.08	0.04	0.29	0.13	0.13	0.12
cari	0.48	0.70	0.91	0.51	0.61	0.48	0.48	0.50
cariParent	0.43	0.65	0.82	0.48	0.62	0.43	0.43	0.46
cari parentInternal	0.45	0.07	0.22	0.06	0.78	0.45	0.45	0.34
cekAda	0.43	0.64	0.84	0.48	0.63	0.43	0.43	0.45
getChild	0.44	0.16	0.27	0.14	0.83	0.44	0.44	0.33
getOrtu	0.55	0.62	0.78	0.58	0.72	0.55	0.55	0.48
InOrder	1.00	0.32	0.50	0.35	0.62	1.00	1.00	0.88
insertRoot	0.32	1.00	0.61	0.84	0.38	0.32	0.32	0.36

Skenario pengujian yang dilakukan adalah dengan cara menangkap nilai kemiripan antar dokumen yang dihitung menggunakan metode *LSI*, di mana nilai kemiripan dokumen dengan LSI ini juga menjadi nilai CSM lalu dibandingkan dan dicari selisih antar kedua nilai dari sistem yang berbeda tersebut. Setelah mengetahui

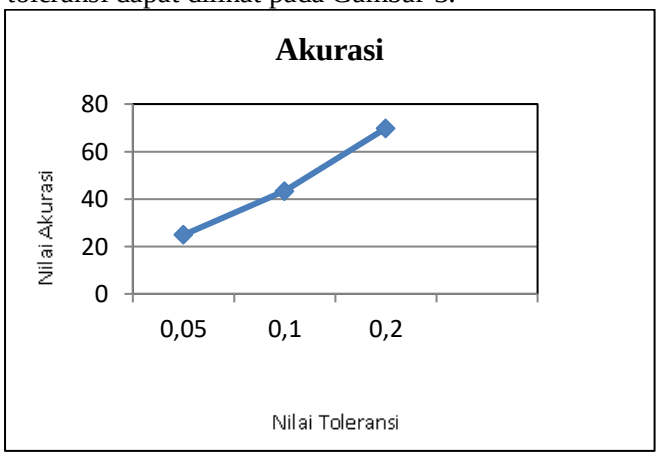
selisihnya, lalu nilai toleransi diberikan untuk mengetahui akurasi sistem. Pemberian nilai toleransi tersebut hanya berupa sebuah percobaan saja. Nilai toleransi tersebut merupakan nilai yang dapat dan perlu diteliti lebih lanjut. Pemberian nilai toleransi ini dikarenakan nilai yang ditangkap dari kedua sistem tersebut hanya sedikit yang sama. Pengujian dilakukan dengan cara mengambil sebanyak 576 *dataset* didapatkan dari 24 *method* dan 3 *class*. *Dataset* sebanyak 576 didapatkan dari matriks kemiripan antar dokumen dengan ordo matriks sebesar 24 kali 24 sehingga menghasilkan total data uji sebanyak 576 data uji. Tabel 1 adalah hasil perhitungan LSI atau nilai CSM pada aplikasi pembanding *lsa.colorado.edu*.

Setelah mendapatkan hasil perhitungan dari kakas bantu pembanding, langkah selanjutnya adalah dengan membandingkan dengan hasil perhitungan LSI atau nilai CSM pada kakas bantu perhitungan nilai kopling yang diusulkan pada penelitian ini. Tabel 2 adalah hasil perhitungan LSI atau nilai CSM pada sistem kakas bantu dan selisih antara hasil pada sistem kakas bantu dengan aplikasi pembanding.

TABEL II
HASIL PERHITUNGAN NILAI CSM PADA KAKAS BANTU

Document	*Main*	Selisih	Cari	Selisih	cariParent	Selisih	cariParentInternal
main	0.99	0.01	0.15	-0.03	0.12	0.10	0.07
cari	0.15	-0.03	0.98	0.02	0.57	-0.35	0.18
cariParent	0.16	0.05	0.55	0.36	0.98	-0.71	0.45
cari parentInternal	0.11	0.11	0.22	0.00	0.47	0.53	0.99
cekAda	0.12	0.10	0.57	0.34	0.59	-0.33	0.16
getChild	0.25	0.08	0.36	-0.06	0.35	0.55	0.44
getOrtu	0.15	0.00	0.36	0.52	0.36	-0.04	0.28
InOrder	0.16	-0.03	0.32	0.16	0.33	0.12	0.38
insertRoot	0.17	-0.07	0.48	0.22	0.41	-0.34	0.27

Berdasarkan selisih hasil perbandingan perhitungan implementasi program dan perhitungan aplikasi pembanding seperti yang tertera pada Tabel 1 dan Tabel 2, didapatkan bahwa dari total data uji sebanyak 24 *method* dalam tiga *class* semuanya memberikan nilai akurasi berbeda. Percobaan dilakukan dengan cara memberikan tiga nilai toleransi yang berbeda yaitu 0,05; 0,1 dan 0,20. Penentuan nilai toleransi tersebut masih *open-issue*, memungkinkan untuk dilakukan penelitian lebih dalam. Pemberian nilai toleransi sebesar 0,05; 0,1 dan 0,20 dirasakan memiliki *range* yang cukup untuk dijadikan sebagai percobaan. Pemberian toleransi tersebut berguna untuk memberikan toleransi akurasi yang artinya adalah jika nilai selisih kurang dari nilai toleransi maka data tersebut masih bisa di toleransi keakuratan datanya. Pada saat memberikan ketiga nilai toleransi tersebut, hasil yang didapatkan jika nilai toleransi sebesar 0,05 maka nilai akurasi sistem sebesar 24,83%. Jika nilai toleransi sebesar 0,1 maka nilai akurasi sistem sebesar 43,41%. Sedangkan, jika nilai toleransi sebesar 0,20 maka nilai akurasi yang diperoleh di atas 50% yaitu 69,78%. Grafik perbedaan nilai akurasi pada tiap-tiap nilai toleransi dapat dilihat pada Gambar 3.



Gambar 2. Grafik perbedaan nilai akurasi pada tiap-tiap nilai toleransi

Nilai akurasi didapat dari Rumus 5,

$$Akurasi\ sistem = \frac{jumlah\ data\ valid}{jumlah\ keseluruhan\ data} \text{ x } 100\% \quad (5)$$

Keterangan :

Jumlah data *valid* adalah jumlah data yang hasil selisihnya kurang dari nilai toleransi. Jumlah keseluruhan data adalah jumlah keseluruhan data uji yaitu 576 data. Jumlah ini didapatkan dari 24 *method* yang saling dibandingkan satu sama lain sehingga total data uji sebenarnya adalah 576 data uji.

Dari perhitungan akurasi seperti pada Gambar 4 yang dilakukan dengan metode perhitungan perbedaan

selisih antara perhitungan implementasi program dan perhitungan aplikasi pembanding maka dapat dilakukan analisis bahwa terdapat perbedaan hasil akurasi. Hal ini disebabkan oleh perbedaan hasil perhitungan antara perhitungan pada saat implementasi program dengan hasil perhitungan pada aplikasi pembanding. Perbedaan ini disebabkan oleh kamus yang digunakan pada aplikasi pembanding dengan program berbeda.

Pada saat analisis data atau pencarian informasi menggunakan metode LSI kamus yang digunakan dapat berbeda-beda sesuai dengan ruang lingkup bahasan informasi yang ingin dicari. Pada kakas bantu pembanding, pengguna dapat memilih topik bahasan informasi yang ingin dicocokkan dan tentu saja aplikasi akan memberikan kamus yang sesuai dengan topik yang dibutuhkan. Contohnya adalah ruang lingkup dibedakan antara tema psikologi, energi, kesehatan, dan lain-lain. Berbeda halnya dengan implementasi program pada penelitian ini yang hanya menggunakan satu kamus umum saja.

Metode perhitungan kopling secara konseptual memiliki kelebihan yaitu dapat mengukur nilai kopling antar *class* dengan akurat. Dengan adanya kakas bantu ini, dapat mempercepat proses perhitungan kopling. Namun, metode perhitungan kopling ini memiliki kelemahan dari sisi metode LSI yang digunakan. LSI bekerja dengan menghitung frekuensi kemunculan kata antara kode sumber yang diuji. Sehingga apabila terdapat kasus dimana terdapat 2 kata yang memiliki makna yang sama namun secara sintaks berbeda seperti misalnya: kapal terbang dan pesawat, maka dianggap tidak sama secara makna. Perlu dianalis lebih dalam lagi untuk penelitian selanjutnya agar bisa mengakomodir kasus tersebut.

V. KESIMPULAN

Kesimpulan yang diperoleh dalam penelitian Kakas Bantu Perhitungan Nilai Kopling Menggunakan *Conceptual Coupling Metrics*, antara lain:

1. Pengidentifikasian *class* yang saling berkaitan satu sama lain dapat diukur dengan pertama-tama melakukan *parsing* terhadap *method-method* yang dimiliki tiap-tiap class. Lalu, *method-method* tersebut diuraikan menjadi masing-masing dokumen agar memudahkan pada saat dicocokkan satu sama lain. Proses pencocokan dokumen yang berisi *body method* tersebut menggunakan sebuah metode temu kembali informasi yaitu *LSI*. Terakhir, perhitungan kopling dilakukan dengan menggunakan kopling metrik yaitu CSM, nilai CSMC, CSBC dan CoCC.
2. Cara menentukan *conceptual coupling metrics* berdasarkan pada method dalam sebuah kumpulan dokumen kode sumber adalah dengan cara mengambil nilai *relevance* tiap-tiap *method* dan nilai *relevance* tersebut dimasukkan ke dalam tabel CSM. Nilai *relevance* dalam tabel CSM tersebut menjadi nilai CSM. Lalu, nilai CSM tersebut menjadi dasar bagi perhitungan parameter *CSMC*. Nilai CSMC menjadi dasar bagi perhitungan nilai *CSBC* dan nilai CSBC menjadi dasar bagi perhitungan nilai CoCC.
3. Tingkat akurasi dari sistem berbanding lurus dengan nilai toleransi yang diberikan. Pada saat memberikan ketiga nilai toleransi tersebut, hasilnya jika nilai toleransi sebesar 0,05 maka nilai akurasi sistem sebesar 24,83%. Jika nilai toleransi sebesar 0,1 maka nilai akurasi sistem sebesar 43,41%. Sedangkan, jika nilai toleransi sebesar 0,20 maka nilai akurasi yang diperoleh di atas 50% yaitu 69,78%. Semakin tinggi nilai toleransi maka semakin tinggi pula nilai akurasi sistem.

VI. DAFTAR PUSTAKA

[1] D. Poshyvanyk and A. Marcus, "The Conceptual Coupling Metrics for Object-Oriented Systems," in *22nd IEEE International Conference on Software Maintenance*, Philadelphia, 2006.

[2] G. Gui and P. D. Scott, "Ranking reusability of software components using coupling metrics," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1450-1459, 2007.

[3] A. Parashar and J. K. Chhabra, "Clustering Dynamic Class Coupling Data to Measure Class Reusability Pattern," in *Communications in Computer and Information Science*, High Performance Architecture and Grid Computing ed., vol. 169, Heidelberg, Springer, 2011, pp. 126-130.

[4] A. Dennis, B. H. Wixom and R. M. Roth, Systems Analysis and Design, 6 ed., Hoboken: Wiley, 2014.

[5] A. Aloysius and L. Arockiam, "Coupling Complexity Metric: A Cognitive Approach," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 4, no. 9, pp. 29-35, 2012.

[6] H. Bunyamin, "Information Retrieval System Dengan Metode Latent Semantic Indexing," Institut Teknologi Bandung, Bandung, 2005.

[7] M. Riley, E. Heinen and J. Ghosh, "A Text Retrieval Approach To Content-Based Audio Retrieval," in *ISMIR 2008: Proceedings of the 9th International Conference of Music Information Retrieval*, Philadelphia, 2008.