

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang didapat dari penelitian ini adalah:

1. Dari 9 buah jenis *coupling* yang telah disebutkan pada bagian dasar teori, beberapa relevan dengan metrik yang dicetuskan oleh Chidamber-Kemerer. Jenis-jenis *coupling* yang relevan dengan metrik Chidamber-Kemerer adalah: *Common coupling* (relevan dengan CBO), *stamp coupling* (relevan dengan CBO), *Data coupling* (relevan dengan CBO dan RFC), *routine call* (relevan dengan CBO dan RFC), dan *type use coupling* (relevan dengan CBO dan RFC).
2. Pengukuran *coupling* dilakukan dengan cara mengimplementasikan definisi metrik Chidamber-Kemerer. Metrik-metrik tersebut adalah DIT, NOC, CBO, RFC, dan LCOM.
3. Pengukuran *coupling* diimplementasikan dengan membuat algoritma dari definisi metrik Chidamber-Kemerer. Algoritma tersebut bisa berbeda-beda sehingga implementasinya pun berbeda pula. Ini terlihat dari pengujian terhadap perangkat lunak sejenis. Nilai yang dihasilkan perangkat lunak yang dikembangkan dan perangkat lunak sejenis tidak sama. Hal ini dikarenakan implementasi yang digunakan oleh tiap perangkat lunak berbeda.
4. Kuantitas perangkat lunak yang dihasilkan diukur dengan cara melakukan pengujian. Telah dilakukan pengujian berupa *unit testing* dan didapat bahwa hasilnya sesuai dengan implementasi. Selain itu dilakukan pengujian terhadap perangkat sejenis dan didapatkan bahwa perilaku perangkat lunak yang dikembangkan serupa dengan perilaku perangkat lunak sejenis.
5. Berdasarkan pengujian eksperimental, didapat bahwa nilai RFC merupakan nilai yang paling berpengaruh dalam pengukuran *coupling*.

6.2 Saran

Saran yang bisa diberikan sebagai ide untuk penelitian sejenis adalah:

1. Membuat perangkat lunak yang dapat mengukur tingkat *coupling* pada perangkat lunak yang menggunakan bahasa pemrograman selain Java.
2. Melihat kemungkinan implementasi metrik-metrik lain, misalnya *Afferent coupling* (Ca) atau *Number of Public Methods for a class* (NPM).

DAFTAR REFERENSI

- [1] Lethbridge, T. C. dan Laganriere, R. (2005) *Object-oriented software engineering*. McGraw-Hill New York.
- [2] Gosling, J., Joy, B., Steele, G. L., Jr., Bracha, G., dan Buckley, A. (2013) The java language specification, java se 7 edition. URL <http://docs.oracle.com/javase/specs/jls/se7/jls7.pdf>. (Zitiert auf Seite 10), 1.
- [3] Rosen, K. H. (2006) Discrete mathematics and its applications. *AMC*, 6.
- [4] Munir, R. (2012) *Matematika Diskrit*. Penerbit Informatika, Bandung.
- [5] Aho, A. V., Lam, M. S., Sethi, R., dan Ullman, J. D. (2006) *Compilers: Principles, Techniques, and Tools (2Nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [6] Parr, T. (2007) *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Bookshelf.
- [7] Sommerville, I. (2010) *Software Engineering*, 9th edition. Addison-Wesley Publishing Company, USA.
- [8] Stevens, W. P., Myers, G. J., dan Constantine, L. L. (1974) Structured design. *IBM Systems Journal*, 13, 115–139.
- [9] Chidamber, S. R. dan Kemerer, C. F. (1994) A metrics suite for object oriented design. *IEEE Transactions on software engineering*, 20, 476–493.
- [10] Sanjaya, V. (2016) Perangkat lunak pengkonversi java ke struktur diagram kelas. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [11] 14977 (1996) *ISO/IEC 14977:1996 Information Technology - Syntactic Metalanguage - Extended BNF*. International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). Geneva, Switzerland.
- [12] Myers, G. J., Sandler, C., dan Badgett, T. (2011) *The art of software testing*. John Wiley & Sons.