DAFTAR PUSTAKA

[1] A. Finandhita, “Development of Software Quality Assessment Model for Mobile-based Elderly Fall Detection Software,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Aug. 2020. doi: 10.1088/1757-899X/879/1/012088.

[2] B. Latte, S. Henning, and M. Wojcieszak, “*Clean code*: On the Use of Practices and Tools to Produce Maintainable Code for Long-Living Software.” [Online]. Available: http://gitlab.org

[3] D. Galin, *Software Quality Assurance From theory to implementation*. 2004. [Online]. Available: www.pearsoned.co.uk

[4] A. Mukharil Bachtiar, D. Dharmayanti, and M. K. Sabariah, “ANALISIS KUALITAS PERANGKAT LUNAK TERHADAP SISTEM INFORMASI UNIKOM,” *Majalah Ilmiah UNIKOM*, vol. 11, no. 2, pp. 224–233, 2007.

[5] Suman and M. Wadhwa, “A Comparative Study of Software Quality Models,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, 2014, [Online]. Available: www.ijcsit.com

[6] C. R. Kothari, *Research Methodology: Methods and Techniques*. New Age International (P) Limited, 2004.

[7] A. Ganpati, A. Kalia, and H. Singh, “A Comparative Study of *Maintainability Index* of Open Source Software,” 2012. [Online]. Available: www.ijetae.com

[8] T. Kuipers and J. Visser, “*Maintainability Index* Revisited-position paper,” *in Special session on system quality and maintainability (SQM 2007) of the 11th European conference on software maintenance and reengineering (CSMR 2007)*, 2007.

[9] D. Coleman, D. Ash, B. Lowther, and P. Oman, “Using Metrics to Evaluate Software System Maintainability,” *Computer (Long Beach Calif)*, vol. 27, no. 8, pp. 44–49, 1994, doi: 10.1109/2.303623.

[10] Zain Naboulsi, “Code Metrics – *Maintainability Index*,” 2010.

[11] T. J. Mccabe, “A Complexity Measure,” *IEEE Trans. Softw. Eng*, vol. SE-2, no. 4, pp. 308–320, 1976.

[12] M. M. Suleman Sarwar, S. Shahzad, and I. Ahmad, “Cyclomatic complexity: The nesting problem,” in *8th International Conference on Digital Information Management, ICDIM 2013*, 2013, pp. 274–279. doi: 10.1109/ICDIM.2013.6693981.

[13] Verifysoft, “Measurement of Halstead Metrics with Testwell CMT++ and CMTJava,” 2017.

[14] R. C. Martin, *Clean code: A Handbook of Agile Software Craftsmanship*. Pearson Education, 2008.

[15] H. Singh and S. Imtiyaz Hassan, “Effect of SOLID Design Principles on Quality of Software: An Empirical Assessment,” *Int J Sci Eng Res*, vol. 6, no. 4, 2015, [Online]. Available: http://www.ijser.org

[16] M. Hills and P. Klint, “PHP AiR: Analyzing PHP Systems with Rascal,” *Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, pp. 454–457, 2014, doi: 10.1109/CSMR-WCRE.2014.6747217.

[17] Jean-François Lépine, “PHPMetrics Beautiful and understandable static analysis tool for PHP,” 2013.

[18] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns – Elements of Reusable Object-Oriented Software*. 2002.

[19] Bella Chintya Neyfa and Dony Tamara, “PERANCANGAN APLIKASI E-CANTEEN BERBASIS ANDROID DENGAN MENGGUNAKAN METODE OBJECT ORIENTED ANALYSIS & DESIGN (OOAD),” *Jurnal Penelitian Komunikasi dan Opini Publik*, vol. 20, no. 1, pp. 83–91, 2016.

[20] Jogiyanto H.M, *Analisis Dan Desain*. Yogyakarta: Andi, 2005.

[21] Rosa A.S and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Beriorientasi Objek*. Bandung: Informatika, 2019.

[22] B. R. Reddy and A. Ojha, “Performance of *Maintainability Index* prediction models: a feature selection based study,” *Evolving Systems*,

vol. 10, no. 2, pp. 179–204, Jun. 2019, doi: 10.1007/s12530-017-9201-0.

[23] A. Madi, O. Kassem Zein, and S. Kadry, “On the Improvement of Cyclomatic Complexity Metric,” 2013.

[24] T. J. Mccabe, “A Complexi-ty Measure,” 1976.

[25] Pasquale Ceres, “McCabe’s Cyclomatic Complexity,” 2016. https://ars.altervista.org/lint_php/lint_php.php (accessed Jun. 08, 2023).

[26] S. McConnell, *Code Complete 2nd Edition*. 2004.

[27] H. Zhang, “An investigation of the relationships between lines of code and defects,” in *IEEE International Conference on Software Maintenance, ICSM*, 2009, pp. 274–283. doi: 10.1109/ICSM.2009.5306304.

[28] A. Molnar and S. Motogna, “Discovering Maintainability Changes in Large Software Systems,” 2017. [Online]. Available: https://doi.org/10.475/123_4

[29] M. G. Spiegel, J. E. Gaffney, H. Joseph. Highland, and ACM-Sigmetrics., “A SOFTWARE STUDY USING HALSTEAD METRICS,” p. 197, 1981.

[30] R. Anggrainingsih, B. O. P. Johannanda, A. P. Kuswara, D. Wahyuningsih, and T. Rejekiningsih, “Comparison of maintainability and flexibility on open source LMS,” in *Proceedings - 2016 International Seminar on Application of Technology for Information and Communication, ISEMANTIC 2016*, Institute of Electrical and Electronics Engineers Inc., Mar. 2017, pp. 273–277. doi: 10.1109/ISEMANTIC.2016.7873850.

[31] R. C. Martin, “Design Principles and Design Patterns,” 2000. [Online]. Available: www.objectmentor.com

[32] M. Prajapati, “International Journal of All Research Writings 23 ASP.NET MVC-GENERIC REPOSITORY PATTERN AND UNIT OF WORK,” vol. 1, no. 1, 2019, [Online]. Available: www.ijarw.com