

DAFTAR PUSTAKA

- [1] “Support Library | Android Developers.” <https://developer.android.com/topic/libraries/support-library> (accessed Jul. 16, 2020).
- [2] R. S. Pressman, *Software Engineering: A Practitioner’s Approach*, Seventh Ed. London: McGraw-Hill Education, 2010.
- [3] M. Majthoub, M. H. Qutqui, and Y. Odeh, “Software Re-engineering: An Overview,” in *2018 8th International Conference on Computer Science and Information Technology, CSIT 2018*, Oct. 2018, pp. 266–270, doi: 10.1109/CSIT.2018.8486173.
- [4] “Android Developers Blog: Google I/O 2019: Empowering developers to build the best experiences on Android + Play.” <https://android-developers.googleblog.com/2019/05/google-io-2019-empowering-developers-to-build-experiences-on-Android-Play.html> (accessed Feb. 06, 2020).
- [5] “Android Developers Blog: Android’s commitment to Kotlin.” <https://android-developers.googleblog.com/2019/12/androids-commitment-to-kotlin.html> (accessed Jul. 16, 2020).
- [6] Kumar and Ranjit, *RESEARCH METHODOLOGY a step-by-step guide for beginners*. London: SAGE Publications, 2009.
- [7] A. C. Graciamary, “Enhanced Re-Engineering Mechanism to Improve the Efficiency of Software Re-Engineering,” 2016. Accessed: Dec. 26, 2019. [Online]. Available: www.ijacsa.thesai.org.
- [8] A. C. Graciamary and M. Chidambaram, “EESRM: An Effective Approach to Improve the Performance of Software Re-Engineering,” 2018. Accessed: Jan. 07, 2020. [Online]. Available: <http://www.ripublication.com>.
- [9] I. Jovanovikj, G. Engels, A. Anjorin, and S. Sauer, “Model-driven test case migration: The test case reengineering horseshoe model,” in *Lecture Notes in Business Information Processing*, 2018, vol. 317, pp. 133–147, doi: 10.1007/978-3-319-92901-9_13.

- [10] A. Shatnawi, A. D. Seriai, H. Sahraoui, and Z. Alshara, "Reverse engineering reusable software components from object-oriented APIs," *J. Syst. Softw.*, vol. 131, pp. 442–460, Sep. 2017, doi: 10.1016/j.jss.2016.06.101.
- [11] B. Dorninger, M. Moser, and J. Pichler, "Multi-language re-documentation to support a COBOL to Java migration project," in *SANER 2017 - 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering*, Mar. 2017, pp. 536–540, doi: 10.1109/SANER.2017.7884669.
- [12] T. G. S. Ducasse and J.-M. Favre., *Modeling software evolution by treating history as a first class entity, in on Software Evolution Through Transformation*. 2004.
- [13] Y. Moghaddas and H. Rashidi, "A novel approach for replacing legacy systems," *J. Appl. Sci.*, vol. 9, no. 22, pp. 4086–4090, 2009, doi: 10.3923/jas.2009.4086.4090.
- [14] B. Joshi and B. Joshi, "Overview of SOLID Principles and Design Patterns," in *Beginning SOLID Principles and Design Patterns for ASP.NET Developers*, Apress, 2016, pp. 1–44.
- [15] A. M. Bachtiar, D. Dharmayanti, and M. K. Sabariah, "Analisis Kualitas Perangkat Lunak Terhadap Sistem Informasi UNIKOM," *Maj. Ilm. UNIKOM*, vol. 11, no. 2, pp. 224–233, 2007.
- [16] B. R. Reddy and A. Ojha, "Performance of Maintainability Index prediction models: a feature selection based study," *Evol. Syst.*, vol. 10, no. 2, pp. 179–204, Jun. 2019, doi: 10.1007/s12530-017-9201-0.
- [17] A. A. Saifan and A. Al-Rabadi, "Evaluating maintainability of android applications," in *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, Oct. 2017, pp. 518–523, doi: 10.1109/ICITECH.2017.8080052.
- [18] "Pembangunan Kakas Bantu Untuk Mengukur Maintainability Index Pada Perangkat Lunak Berdasarkan Nilai Halstead Metrics dan McCabe's Cyclomatic Complexity | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer." <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4625>

(accessed Jul. 11, 2020).

- [19] S. Counsell, S. Swift, and J. Crampton, “The interpretation and utility of three cohesion metrics for object-oriented design,” *ACM Trans. Softw. Eng. Methodol.*, vol. 15, no. 2, pp. 123–149, 2006, doi: 10.1145/1131421.1131422.
- [20] J. M. Bieman and B.-K. Kang, “Cohesion and reuse in an object-oriented system,” in *Proceedings of the 1995 Symposium on Software reusability - SSR '95*, 1995, pp. 259–262, doi: 10.1145/211782.211856.
- [21] “Project Metrics Help - Cohesion metrics.” <https://www.aivosto.com/project/help/pm-oo-cohesion.html> (accessed Jun. 12, 2020).
- [22] M. Flower, “Principles in Refactoring,” in *Refactoring: Improving the Design of Existing Code*, 2nd editio., New Jersey: Pearson Education (US), 2019, p. 46.
- [23] C. Chen, R. Alfayez, K. Srisopha, L. Shi, and B. Boehm, “Evaluating human-assessed software maintainability metrics,” in *Communications in Computer and Information Science*, 2016, vol. 675, pp. 120–132, doi: 10.1007/978-981-10-3482-4_9.
- [24] “PENGERTIAN ANDROID: Sejarah, Kelebihan & Versi Sistem Operasi Android | Salamadian.” <https://salamadian.com/pengertian-android/> (accessed Jan. 09, 2020).
- [25] “Apa itu Android? Penjelasan Super Lengkap Android Ada Disini! - Windowsku.” <https://windowsku.com/apa-itu-android-adalah/> (accessed Dec. 10, 2019).
- [26] R. JR, “Android versions: A living history from 1.0 to 10 | Computerworld.” <https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html> (accessed Jan. 09, 2020).
- [27] “Mobile Operating System Market Share Indonesia | StatCounter Global Stats.” <https://gs.statcounter.com/os-market-share/mobile/indonesia/2017> (accessed Jan. 09, 2020).
- [28] T. Lou, “A comparison of Android Native App Architecture MVC, MVP and

MVVM.”

- [29] A. Daoudi, N. Moha, G. ElBoussaidi, and S. Kpodjedo, “An exploratory study of MVC-based architectural patterns in android apps,” in *Proceedings of the ACM Symposium on Applied Computing*, 2019, vol. Part F147772, pp. 1711–1720, doi: 10.1145/3297280.3297447.
- [30] R. C. Martin, “Design Principles,” in *Clean Architecture : A Craftman’s Guide to Software Structure and Design*, Upper Saddle River: Pearson Education (US), 2017, pp. 72–103.
- [31] R. C. Martin, *Clean Code: : A Handbook of Agile Software Craftsmanship*. Upper Saddle River: Pearson Education (US), 2009.
- [32] T. J. McCabe, “A Complexity Measure,” *IEEE Trans. Softw. Eng.*, vol. SE-2, no. 4, pp. 308–320, 1976, doi: 10.1109/TSE.1976.233837.
- [33] “Cyclomatic Complexity Defined | | FREE Demo | | Video Explanation.” <https://www.castsoftware.com/glossary/cyclomatic-complexity> (accessed Apr. 26, 2020).
- [34] “Samsung Health | SAMSUNG Developers.” <https://developer.samsung.com/health> (accessed Jan. 12, 2020).
- [35] J. H.M, “Analisis Dan Desain,” Yogyakarta, 2005.
- [36] Jogiyanto H.M, *Analisis Dan Desain*. Yogyakarta: J&J Learning, 2005.
- [37] S. S. Alhir, *Learning UML*, Sebastopol., no. 0. California: O’Reilly, 2003.
- [38] J. S. Carpenter and M. A. Andrykowski, “Psychometric evaluation of the Pittsburgh Sleep Quality Index,” *J. Psychosom. Res.*, vol. 45, no. 1, pp. 5–13, Jul. 1998, doi: 10.1016/S0022-3999(97)00298-5.
- [39] “Home | Spotify for Developers.” <https://developer.spotify.com/> (accessed Jan. 12, 2020).
- [40] “Mengenal Android Studio | Android Developers.” <https://developer.android.com/studio/intro?hl=ID> (accessed Dec. 10, 2019).
- [41] “Android Jetpack | Android Developers.” <https://developer.android.com/jetpack> (accessed Jan. 13, 2020).
- [42] “Mobile & Tablet Android Version Market Share Indonesia | StatCounter Global Stats.” <https://gs.statcounter.com/android-version-market->

share/mobile-tablet/indonesia (accessed Apr. 27, 2020).

- [43] “Clean Code: Explanation, Benefits, and Examples - DZone Agile.” <https://dzone.com/articles/clean-code-explanation-benefits-amp-examples> (accessed Apr. 29, 2020).
- [44] G. McCulloch, “Coding Is for Everyone—as Long as You Speak English | WIRED.” <https://www.wired.com/story/coding-is-for-everyone-as-long-as-you-speak-english/> (accessed Apr. 29, 2020).