

DAFTAR PUSTAKA

- Amato, G., Meo, M. C., & Scozzari, F. (2016). Exploiting Linearity in Sharing Analysis of Object-oriented Programs. *Electronic Notes in Theoretical Computer Science*, 322, 3–18. <https://doi.org/10.1016/j.entcs.2016.03.002>
- Bakliwal, C., Kishor, I., & Verma, V. (2020). Dependency Injection in OOP. *International Journal of Psychosocial Rehabilitation*, 31295–31298. <https://doi.org/10.61841/V24I4/400342>
- Chiba, S., & Ishikawa, R. (2005). Aspect-Oriented Programming Beyond Dependency Injection. In *Lecture Notes in Computer Science* (Vol. 3586, pp. 121–143). https://doi.org/10.1007/11531142_6
- Crasso Marco, Mateo Cristian, Zunino Alejandro, & Campo Marcelo. (2010). Empirically assessing the impact of dependency injection on the development of web service applications. *Journal of Web Engineering*, 66–94.
- Duggan, D. , & P. D. (2019). Issue Information. *Journal of Software: Evolution and Process*, 31(5). <https://doi.org/10.1002/smr.1970>
- Ekaterina Razina, & David Janzen. (2007). *EFFECTS OF DEPENDENCY INJECTION ON MAINTAINABILITY* .
- Eniyati, S. (2006). *Pengembangan Berorientasi Objek Metode Fusion*.
- Fóthi, Á., Nyéky-Gaizler, J., & Porkoláb, Z. (2003). The structured complexity of object-oriented programs. *Mathematical and Computer Modelling*, 38(7–9), 815–827. [https://doi.org/10.1016/S0895-7177\(03\)90066-5](https://doi.org/10.1016/S0895-7177(03)90066-5)
- Hudli, S. R., & Hudli, R. V. (2013a). A Verification Strategy for Dependency Injection. *Lecture Notes on Software Engineering*, 71–74. <https://doi.org/10.7763/lmse.2013.v1.16>
- Hudli, S. R., & Hudli, R. V. (2013b). A Verification Strategy for Dependency Injection. *Lecture Notes on Software Engineering*, 71–74. <https://doi.org/10.7763/LNSE.2013.V1.16>
- Jasmin, G. A., Galih, D., & Putri, P. (2024). Komparasi Metode Automasi dan Hybrid pada Pengujian Aplikasi Mobile Webrtc Menggunakan Appium. *Journal of Internet and Software Engineering*, 5(1).
- Kartarina, K., & Apriliansyah, A. (2022). Analisis Dependency Injection dan Model-View-Presenter Pada Aplikasi Berbasis Android. *Progresif: Jurnal Ilmiah Komputer*, 18(1), 23. <https://doi.org/10.35889/progresif.v18i1.781>
- Kocsis, Z. A., & Swan, J. (2017). *Dependency Injection for Programming by Optimization*.

- Laigner, R., Mendonça, D., Garcia, A., & Kalinowski, M. (2022). Cataloging dependency injection anti-patterns in software systems. *Journal of Systems and Software*, 184, 111125. <https://doi.org/10.1016/j.jss.2021.111125>
- Ma'ayan, D. D. (2018). The quality of junit tests. *Proceedings of the 1st International Workshop on Software Qualities and Their Dependencies*, 33–36. <https://doi.org/10.1145/3194095.3194102>
- Medromi, H., Moussaid, L., & Fal, L. (2018). Analysis of the allocation of classes, threads and CPU used in embedded systems for Java applications. *Procedia Computer Science*, 134, 334–339. <https://doi.org/10.1016/j.procs.2018.07.181>
- Melton, H., & Tempero, E. (2007). An empirical study of cycles among classes in Java. *Empirical Software Engineering*, 12(4), 389–415. <https://doi.org/10.1007/s10664-006-9033-1>
- Meyer, B. (2000). *Software Development*. www.sdmagazine.com/documents/s=746/sdm0003d/0003d.htm?temp=N3TyuRoNcQ
- Nidhra, S. (2012). Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications*, 2(2), 29–50. <https://doi.org/10.5121/ijesa.2012.2204>
- RDAV Thennakoon, D. B. H. (n.d.). *A STUDY ON OBJECT-ORIENTED DESIGN PRINCIPLES AND PATTERNS*.
- Roemers, A., Hatun, K., & Bockisch, C. (2013). An adapter-aware, non-intrusive dependency injection framework for Java. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/2500828.2500834>
- Setiyanto Rudi, Nurmaesah Nunung, & Nyai Sri Astuti Rahayu. (2019). *Perancangan Sistem Informasi Persediaan Barang Studi Kasus di Vahncollections*.
- Sumanda, N., Munawar, G., & Wisnuadhi, B. (2018). *Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin*.
- Sun, P., Kim, D. K., Ming, H., & Lu, L. (2022). Measuring Impact of Dependency Injection on Software Maintainability. *Computers*, 11(9). <https://doi.org/10.3390/computers11090141>
- Taelman, R., Van Herwegen, J., Vander Sande, M., & Verborgh, R. (2022). Components.js: Semantic dependency injection. *Semantic Web*, 14(1), 135–153. <https://doi.org/10.3233/SW-222945>