

DAFTAR PUSTAKA

- [1] C. Deming, M. A. Khair, S. R. Mallipeddi, and A. Varghese, "Software Testing in the Era of AI: Leveraging Machine Learning and Automation for Efficient Quality Assurance," *Asian J. Appl. Sci. Eng.*, vol. 10, no. 1, pp. 66–76, 2021, doi: 10.18034/ajase.v10i1.88.
- [2] G. Singh, "A Study on Software Testing Life Cycle in Software Engineering," *Int. J. Soft Comput. Eng.*, vol. 9, no. 2, pp. 1–5, 2018.
- [3] O. J. Amman Paul, *Introduction to Software Testing*. Cambridge Press, 2017. [Online]. Available: https://books.google.co.id/books?hl=id&lr=&id=bQtQDQAAQBAJ&oi=fnd&pg=PR9&dq=Introduction+to+Software+Testing&ots=fA6P213_pQ&sig=vTKZfwWNJMsYUzib1KSgQ-TjRDI&redir_esc=y#v=onepage&q&f=false
- [4] V. H. S. Durelli *et al.*, "Machine learning applied to software testing: A systematic mapping study," *IEEE Trans. Reliab.*, vol. 68, no. 3, pp. 1189–1212, 2019, doi: 10.1109/TR.2019.2892517.
- [5] U. Subbiah, M. Ramachandran, and Z. Mahmood, "Software engineering approach to bug prediction models using machine learning as a service (MLaaS)," *ICSOFTE 2018 - Proc. 13th Int. Conf. Softw. Technol.*, no. Icsft, pp. 879–887, 2019, doi: 10.5220/0006926308790887.
- [6] R. R. Saputra, E. Setiawan, and A. Ambarwati, "Manajemen Risiko Teknologi Informasi Menggunakan Metode OCTAVE Allegro pada PT . Hakiki Donarta Surabaya," vol. 17, no. 1, pp. 1–10, 2019.
- [7] A. Pandey, S. Maddula, G. P. Kumar, S. K. Shailendra, and K. Mudaliar, "A Comprehensive Analysis of Ensemble-based Fault Prediction Models Using Product , Process , and Object-Oriented Metrics in Software Engineering," no. December 2023, 2024, doi: 10.5281/zenodo.10464708.
- [8] A. O. Balogun, A. O. Bajeh, V. A. Orie, and A. W. Yusuf-Asaju, "Software Defect Prediction Using Ensemble Learning: An ANP Based Evaluation Method," *FUOYE J. Eng. Technol.*, vol. 3, no. 2, 2018, doi: 10.46792/fuoyejet.v3i2.200.
- [9] R. Malhotra, S. Chawla, and A. Sharma, *Software defect prediction using hybrid techniques: a systematic literature review*, vol. 27, no. 12. Springer Berlin Heidelberg, 2023. doi: 10.1007/s00500-022-07738-w.
- [10] D. Gray, "Why Does Java Remain So Popular?," *blogs.oracle.com*, 2019. <https://blogs.oracle.com/oracleuniversity/post/why-does-java-remain-so-popular>
- [11] M. Crouse, "TIOBE Index for November 2024: Top 10 Most Popular Programming Languages," 2024. <https://www.techrepublic.com/article/tiobe-index-language-rankings/> (accessed Nov. 25, 2024).
- [12] A. Alazba and H. Aljamaan, "Software Defect Prediction Using Stacking Generalization of Optimized Tree-Based Ensembles," *Appl. Sci.*, vol. 12, no. 9, 2022, doi: 10.3390/app12094577.
- [13] Y. Peng, G. Kou, G. Wang, W. Wu, and Y. Shi, "Ensemble of software defect predictors: An AHP-based evaluation method," *Int. J. Inf. Technol. Decis. Mak.*, vol. 10, no. 1, pp. 187–206, 2011, doi: 10.1142/S0219622011004282.
- [14] L. qiong Chen, C. Wang, and S. long Song, "Software defect prediction based on nested-stacking and heterogeneous feature selection," *Complex Intell. Syst.*,

- vol. 8, no. 4, pp. 3333–3348, 2022, doi: 10.1007/s40747-022-00676-y.
- [15] M. A. Ihsan Aquil, “Predicting Software Defects using Machine Learning Techniques,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 6609–6616, 2020, doi: 10.30534/ijatcse/2020/352942020.
 - [16] Y. Yao, Z. Xiao, B. Wang, B. Viswanath, H. Zheng, and B. Y. Zhao, “Complexity vs. performance,” no. 119, pp. 384–397, 2017, doi: 10.1145/3131365.3131372.
 - [17] G. S. I. & C. Raimund Moser, Witold Pedrycz, “A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction,” *Proc. 30th Int. Conf. Softw. Eng.*, p. Pages 181-190, 2008, doi: <https://doi.org/10.1145/1368088.136811>.
 - [18] M. Ali *et al.*, “Software Defect Prediction Using an Intelligent Ensemble-Based Model,” *IEEE Access*, vol. 12, pp. 20376–20395, 2024, doi: 10.1109/ACCESS.2024.3358201.
 - [19] G. Pandey, V. J. Pugazhenthiraj, and A. Murugan, “Advances in Software Testing in 2024 : Experimental Insights , Frameworks , and Future Directions,” no. November, 2024, doi: 10.17148/IJARCC.2024.131103.
 - [20] M. Alenezi and M. Akour, “AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions,” *Appl. Sci.*, vol. 15, no. 3, pp. 1–26, 2025, doi: 10.3390/app15031344.
 - [21] K. Phung, E. Ogunshile, and M. Aydin, “A Novel Software Fault Prediction Approach to Predict Error-type Proneness in the Java Programs Using Stream X-Machine and Machine Learning,” *Proc. - 2021 9th Int. Conf. Softw. Eng. Res. Innov. CONISOFT 2021*, no. August, pp. 168–179, 2021, doi: 10.1109/CONISOFT52520.2021.00032.
 - [22] S. S. Rathore and S. Kumar, “A study on software fault prediction techniques,” *Artif. Intell. Rev.*, vol. 51, no. 2, pp. 255–327, 2019, doi: 10.1007/s10462-017-9563-5.
 - [23] J. Jiarpakdee, C. K. Tantithamthavorn, H. K. Dam, and J. Grundy, “An Empirical Study of Model-Agnostic Techniques for Defect Prediction Models,” *IEEE Trans. Softw. Eng.*, vol. 48, no. 1, pp. 166–185, 2022, doi: 10.1109/TSE.2020.2982385.
 - [24] G. Esteves, E. Figueiredo, A. Veloso, M. Viggiano, and N. Ziviani, “Understanding machine learning software defect predictions,” *Autom. Softw. Eng.*, vol. 27, no. 3–4, pp. 369–392, 2020, doi: 10.1007/s10515-020-00277-4.
 - [25] T. Beranič, V. Podgorelec, and M. Heričko, “Towards a reliable identification of deficient code with a combination of software metrics,” *Appl. Sci.*, vol. 8, no. 10, 2018, doi: 10.3390/app8101902.
 - [26] K. A. M. Ferreira, M. A. S. Bigonha, R. S. Bigonha, L. F. O. Mendes, and H. C. Almeida, “Identifying thresholds for object-oriented software metrics,” *J. Syst. Softw.*, vol. 85, no. 2, pp. 244–257, 2012, doi: 10.1016/j.jss.2011.05.044.
 - [27] C. K. Tantithamthavorn and J. Jiarpakdee, “Explainable AI for Software Engineering,” *Proc. - 2021 36th IEEE/ACM Int. Conf. Autom. Softw. Eng. ASE 2021*, pp. 1–2, 2021, doi: 10.1109/ASE51524.2021.9678580.
 - [28] F. Zhang, A. E. Hassan, S. McIntosh, and Y. Zou, “The use of summation to aggregate software metrics hinders the performance of defect prediction models,” *IEEE Trans. Softw. Eng.*, vol. 43, no. 5, pp. 476–491, 2017, doi: 10.1109/TSE.2016.2599161.
 - [29] I. Q. U. Fatwa Ramdani, *Pengantar Data Science*. Jakarta: Bumi Aksara, 2022.
 - [30] R. Li, L. Zhou, S. Zhang, H. Liu, X. Huang, and Z. Sun, “Software Defect

- Prediction Based on Ensemble Learning,” *ACM Int. Conf. Proceeding Ser.*, pp. 1–6, 2019, doi: 10.1145/3352411.3352412.
- [31] E. Ronchieri, M. Canaparo, and M. Belgiovine, *Software Defect Prediction on Unlabelled Datasets: A Comparative Study*, vol. 12250 LNCS. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58802-1_25.
 - [32] A. A. Ibrahim, R. L. Ridwan, M. M. Muhammed, R. O. Abdulaziz, and G. A. Saheed, “Comparison of the CatBoost Classifier with other Machine Learning Methods,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, pp. 738–748, 2020, doi: 10.14569/IJACSA.2020.0111190.
 - [33] Guryanov, “Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees,” *Anal. Images, Soc. Networks Texts*, vol. 11832, pp. 39–50, 2019, doi: https://doi.org/10.1007/978-3-030-37334-4_4.
 - [34] G. Santos, E. Figueiredo, A. Veloso, M. Viggiano, and N. Ziviani, “Predicting Software Defects with Explainable Machine Learning,” *ACM Int. Conf. Proceeding Ser.*, 2020, doi: 10.1145/3439961.3439979.
 - [35] Z. Faska, L. Khriisi, K. Haddouch, and N. El Akkad, “A robust and consistent stack generalized ensemble-learning framework for image segmentation,” *J. Eng. Appl. Sci.*, vol. 70, no. 1, pp. 1–20, 2023, doi: 10.1186/s44147-023-00226-4.
 - [36] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Aug, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.
 - [37] A. K. A. Nandan Mullakara, *Robotic Process Automation Projects: Build real-world RPA solutions using UiPath and Automation Anywhere*, 1st ed. Birmingham: Packt Publishing, Ltd., 2020. [Online]. Available: https://books.google.co.id/books?hl=id&lr=&id=jNPnDwAAQBAJ&oi=fnd&pg=PP1&dq=reframework+uipath&ots=uZEB6dSGKD&sig=enkirj8BY2BDWKtxEYCHia85wvM&redir_esc=y#v=onepage&q=reframeworkuipath&f=false
 - [38] R. J. Verzello, *Data processing, systems and concepts*. New York : McGraw-Hill, 1982.
 - [39] G. Gay and R. Just, “Defects4J as a Challenge Case for the Search-Based Software Engineering Community,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12420 LNCS, no. Section 2, pp. 255–261, 2020, doi: 10.1007/978-3-030-59762-7_19.
 - [40] Maurício Aniche, “Java Code Metrics Calculator (CK Metrics),” 2015.
 - [41] A. Daza *et al.*, “Stacking ensemble based hyperparameters to diagnosing of heart disease: Future works,” *Results Eng.*, vol. 21, no. January, 2024, doi: 10.1016/j.rineng.2024.101894.
 - [42] S. K. Palaniswamy and R. Venkatesan, “Hyperparameters tuning of ensemble model for software effort estimation,” *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 6, pp. 6579–6589, 2021, doi: 10.1007/s12652-020-02277-4.
 - [43] D. Rajapaksha, C. Tantithamthavorn, J. Jiarapakdee, C. Bergmeir, J. Grundy, and W. Buntine, “SQAPLanner: Generating Data-Informed Software Quality Improvement Plans,” *IEEE Trans. Softw. Eng.*, vol. 48, no. 8, pp. 2814–2835, 2022, doi: 10.1109/TSE.2021.3070559.