

DAFTAR PUSTAKA

- [1] A. M. Nuris, A. Maharani, and R. N. Rachmadita, “Analisis Risiko Proyek Pengembangan Perangkat Lunak Menggunakan Kerangka Kerja ISO 31000,” *J. METRIS*, vol. 22, no. 02, pp. 73–81, Feb. 2022, doi: 10.25170/metris.v22i02.2800.
- [2] A. N. Hasibuan and T. Dirgahayu, “Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna”.
- [3] H. Liu, L. Liu, C. Yue, Y. Wang, and B. Deng, “Autotestgpt: A System for the Automated Generation of Software Test Cases Based on Chatgpt,” 2023, *SSRN*. doi: 10.2139/ssrn.4584792.
- [4] Dudekula Mohammad Rafi, Katam Reddy Kiran Moses, K. Petersen, and M. V. Mantyla, “Benefits and limitations of automated software testing: Systematic literature review and practitioner survey,” in *2012 7th International Workshop on Automation of Software Test (AST)*, Zurich, Switzerland: IEEE, Jun. 2012, pp. 36–42. doi: 10.1109/IWAST.2012.6228988.
- [5] G. Candea and P. Godefroid, “Automated Software Test Generation: Some Challenges, Solutions, and Recent Advances,” in *Computing and Software Science*, vol. 10000, B. Steffen and G. Woeginger, Eds., in *Lecture Notes in Computer Science*, vol. 10000. , Cham: Springer International Publishing, 2019, pp. 505–531. doi: 10.1007/978-3-319-91908-9_24.
- [6] M. Olsthoorn, “More effective test case generation with multiple tribes of AI,” in *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*, Pittsburgh Pennsylvania: ACM, May 2022, pp. 286–290. doi: 10.1145/3510454.3517066.
- [7] S. Bhatia, T. Gandhi, D. Kumar, and P. Jalote, “Unit Test Generation using Generative AI : A Comparative Performance Analysis of Autogeneration Tools,” in *Proceedings of the 1st International Workshop on Large Language Models for Code*, Lisbon Portugal: ACM, Apr. 2024, pp. 54–61. doi: 10.1145/3643795.3648396.
- [8] G. Yi, Z. Chen, Z. Chen, E. Wong, and N. C, “Exploring the Capability of ChatGPT in Test Case Generation,” in *2023 IEEE 23rd International*

- Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, Chiang Mai, Thailand: IEEE, 2023.
- [9] Ç. Yıldız, N. K. Ravichandran, N. Sharma, M. Bethge, and B. Ermis, “Investigating Continual Pretraining in Large Language Models: Insights and Implications,” Feb. 12, 2025, *arXiv*: arXiv:2402.17400. doi: 10.48550/arXiv.2402.17400.
 - [10] A. Cossu, A. Carta, L. Passaro, V. Lomonaco, T. Tuytelaars, and D. Bacciu, “Continual pre-training mitigates forgetting in language and vision,” *Neural Netw.*, vol. 179, p. 106492, Nov. 2024, doi: 10.1016/j.neunet.2024.106492.
 - [11] A. Yadav, A. Patel, and M. Shah, “A comprehensive review on resolving ambiguities in natural language processing,” *AI Open*, vol. 2, pp. 85–92, 2021, doi: 10.1016/j.aiopen.2021.05.001.
 - [12] A. Ansari, M. B. Shagufta, A. Sadaf Fatima, and S. Tehreem, “Constructing Test cases using Natural Language Processing,” in *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, Chennai, India: IEEE, Feb. 2017, pp. 95–99. doi: 10.1109/AEEICB.2017.7972390.
 - [13] R. Gröpler, V. Sudhi, E. J. C. García, and A. Bergmann, “NLP-Based Requirements Formalization for Automatic Test Case Generation”.
 - [14] Swamy Prasad Rao Velaga, “AUTOMATED TESTING FRAMEWORKS: ENSURING SOFTWARE QUALITY AND REDUCING MANUAL TESTING EFFORTS,” *Int. J. Innov. Eng. Res. Technol.*, vol. 5, no. 2, pp. 78–85, Feb. 2018, doi: 10.26662/ijiert.v5i2.pp78-85.
 - [15] S. S. Rahayu, D. A. Firmansyah, S. Susanti, and U. A. R. Sanjaya, “Analisis Penggunaan Tools Automation Testing pada Aplikasi : Systematic Literature Review,” vol. 8, 2024.
 - [16] L. S. Chan, “Software testing life cycle and its efficiency”.
 - [17] T. R. Devi, “Importance of Testing in Software Development Life Cycle,” 2012.
 - [18] S. Jalil, S. Rafi, T. D. LaToza, K. Moran, and W. Lam, “ChatGPT and Software Testing Education: Promises & Perils,” in *2023 IEEE International*

- Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Apr. 2023, pp. 4130–4137. doi: 10.1109/ICSTW58534.2023.00078.
- [19] Z. Yuan *et al.*, “No More Manual Tests? Evaluating and Improving ChatGPT for Unit Test Generation,” May 19, 2024, *arXiv*: arXiv:2305.04207. doi: 10.48550/arXiv.2305.04207.
- [20] Z. Ke, Y. Shao, H. Lin, T. Konishi, G. Kim, and B. Liu, “Continual Pre-training of Language Models,” Apr. 12, 2023, *arXiv*: arXiv:2302.03241. doi: 10.48550/arXiv.2302.03241.
- [21] X. Jin *et al.*, “Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora,” Jul. 19, 2022, *arXiv*: arXiv:2110.08534. doi: 10.48550/arXiv.2110.08534.
- [22] J. White *et al.*, “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT,” Feb. 21, 2023, *arXiv*: arXiv:2302.11382. doi: 10.48550/arXiv.2302.11382.
- [23] S. Schulhoff *et al.*, “The Prompt Report: A Systematic Survey of Prompt Engineering Techniques,” Feb. 26, 2025, *arXiv*: arXiv:2406.06608. doi: 10.48550/arXiv.2406.06608.
- [24] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” Mar. 16, 2025, *arXiv*: arXiv:2402.07927. doi: 10.48550/arXiv.2402.07927.
- [25] I. Lauriola, A. Lavelli, and F. Aiolli, “An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools,” *Neurocomputing*, vol. 470, pp. 443–456, Jan. 2022, doi: 10.1016/j.neucom.2021.05.103.
- [26] S. Utomo, I. M. I. Subroto, and A. Riansyah, “Deteksi Plagiat Tugas Akhir dengan Metode Jaccard Similarity,” vol. 4, no. 2, 2022.
- [27] M. Staats, M. W. Whalen, M. P. E. Heimdahl, and A. Rajan, “Coverage Metrics for Requirements-Based Testing: Evaluation of Effectiveness”.