BAB VI. PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa aplikasi web deteksi plagiarisme *source code* Java berhasil dikembangkan. Sistem ini mengintegrasikan model *machine learning* berbasis enam indikator kemiripan ke dalam sebuah aplikasi web berbasis *Streamlit*, sehingga dapat digunakan secara interaktif oleh dosen dan asisten dosen tanpa memerlukan pengetahuan teknis mengenai proses pelatihan model.

Model deteksi yang digunakan dibangun dengan pendekatan klasifikasi biner menggunakan enam fitur kemiripan, yaitu kemiripan baris logika, kemiripan teks ternormalisasi, rasio kemiripan urutan (LCS), kemiripan komentar, kemiripan penamaan variabel, dan kemiripan *token non-boilerplate*. Hasil evaluasi pada Bab IV menunjukkan bahwa model linear SVM memberikan keseimbangan terbaik pada performa, stabilitas generalisasi, dan efisiensi komputasi, sehingga layak diintegrasikan ke dalam sistem aplikasi web.

Sistem ini tidak dimaksudkan untuk menggantikan penilaian dosen atau asisten dosen dalam menilai tugas mahasiswa terkait plagiarisme *source code*. Sistem ini hanya sebagai alat pendukung yang menyediakan skor kemiripan, ringkasan fitur kemiripan, dan *highligth* bagian kode yang perlu diverifikasi secara manual.

B. Saran

Berdasarkan hasil penelitian dan pengembangan sistem yang telah dilakukan, terdapat beberapa saran untuk pengembangan lebih lanjut. Pertama, penelitian selanjutnya dapat memperluas cakupan bahasa pemrograman yang didukung, tidak hanya terbatas pada bahasa Java, sehingga sistem dapat digunakan pada mata kuliah pemrograman dengan bahasa lain seperti Python, PHP, atau bahasa C.

Kedua, fitur deteksi dapat dikembangkan untuk mendukung analisis masal terhadap banyak berkas tugas mahasiswa secara sekaligus, misal melalui mekanisme unggah folder atau integrasi dengan LMS. Pengembangan ini akan meningkatkan efisiensi penggunaan sistem pada kelas dengan jumlah mahasiswa yang besar.

Ketiga, penelitian lanjutan dapat mengeksplorasi pendekatan berbasis fitur kemiripan yang lebih kompleks agar tahan terhadap transformasi plagiarisme yang lebih kompleks, dengan tetap memperhatikan aspek *explainability*.

DAFTAR PUSTAKA

[1] M. Ashraf, A. Sayed, S. E. Elbohy, and E. Eliwa, "Automated Detection Approaches for Source Code Plagiarism in Students' Submissions," 2023.

[2] H. Cheers, Y. Lin, and W. Yan, "Identifying Plagiarised Programming Assignments with Detection Tool Consensus," *Informatics in Education*, vol. 22, no. 1, pp. 1–19, 2023, doi: 10.15388/infedu.2023.05.

[3] B. Devore-Mcdonald and E. D. Berger, "Mossad: Defeating software plagiarism detection," *Proceedings of the ACM on Programming Languages*, vol. 4, no. OOPSLA, Nov. 2020, doi: 10.1145/3428206.

[4] N. Gandhi, K. Gopalan, and P. Prasad, "A Support Vector Machine based approach for plagiarism detection in Python code submissions in undergraduate settings," *Front. Comput. Sci.*, vol. 6, 2024, doi: 10.3389/fcomp.2024.1393723.

[5] N. Awale, M. Pandey, A. Dulal, and B. Timsina, "Plagiarism Detection in Programming Assignments using Machine Learning," *Journal of Artificial Intelligence and Capsule Networks*, vol. 2, no. 3, pp. 177–184, Jul. 2020, doi: 10.36548/jaicn.2020.3.005.

[6] E. Hosam, M. Hadhoud, A. Atiya, and M. Fayek, "Classification feature sets for source code plagiarism detection in Java," *Journal of Engineering and Applied Science*, vol. 69, no. 1, Dec. 2022, doi: 10.1186/s44147-022-00155-8.

[7] R. S. Mehsen and H. D. Joshi, "Detection of Source Code Plagiarism Utilizing an Approach Based on Machine Learning," *International Journal of Computing*, vol. 23, no. 1, pp. 78–84, 2024, doi: 10.47839/ijc.23.1.3438.

[8] F. Ebrahim and M. Joy, "Source Code Plagiarism Detection with Pre-Trained Model Embeddings and Automated Machine Learning," in *International Conference Recent Advances in Natural Language*

Processing, RANLP, Incoma Ltd, 2023, pp. 301–309. doi: 10.26615/978-954-452-092-2_034.

[9] A. Eppa and A. Murali, "Source Code Plagiarism Detection: A Machine Intelligence Approach," in *Proceedings of IEEE 2022 4th International Conference on Advances in Electronics, Computers and Communications, ICAECC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICAECC54045.2022.9716671.

[10] M. A. Fokam and R. Ajoodha, "Influence of Contrastive Learning on Source Code Plagiarism Detection through Recursive Neural Networks," in *2021 3rd International Multidisciplinary Information Technology and Engineering Conference, IMITEC 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/IMITEC52926.2021.9714688.

[11] P. Koukaras and C. Tjortjis, "Data Preprocessing and Feature Engineering for Data Mining: Techniques, Tools, and Best Practices," Oct. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/ai6100257.

[12] E. Slobodkin and A. Sadovnikov, "Towards a Dataset of Programming Contest Plagiarism in Java," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.10763

[13] L. Xiao, H. Zhong, J. Liu, K. Zhang, Q. Xu, and L. Chang, "A Novel Source Code Representation Approach Based on Multi-Head Attention," *Electronics (Switzerland)*, vol. 13, no. 11, Jun. 2024, doi: 10.3390/electronics13112111.

[14] J. Martinez-Gil, "Source Code Clone Detection Using Unsupervised Similarity Measures," 2024, pp. 21–37. doi: 10.1007/978-3-031-56281-5_2.

[15] H. Cheers, Y. Lin, and S. P. Smith, "Academic source code plagiarism detection by measuring program behavioral similarity," *IEEE Access*, vol. 9, pp. 50391–50412, 2021, doi:

10.1109/ACCESS.2021.3069367.

[16] B. T. Sutrisno SP and A. Arief Setyawan, “Menggunakan Binary Classification untuk Mendeteksi SPAM pada SMS dengan Metode Logistic Regression,” *CONTEN : Computer and Network Technology*, vol. 4, no. 1, 2024, [Online]. Available: http://jurnal.bsi.ac.id/index.php/conten

[17] M. Humayoun, M. A. Hashmi, and A. H. Khan, “Measuring Plagiarism in Introductory Programming Course Assignments,” 2022. [Online]. Available: http://bnfc.digitalgrammars.com/LBNF-report.pdf

[18] M. A. Saputra and T. Sugihartono, “Evaluasi Kinerja Model LSTM Untuk Prediksi Risiko Penyakit Jantung Menggunakan Dataset,” *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 7, pp. 1823–1833, Jul. 2025, doi: 10.52436/1.jpti.821.

[19] S. Arti and E. Suherlan, “Evaluasi Kinerja Machine Learning dalam Memprediksi Kemampuan Adaptasi Mahasiswa pada Lingkungan Pembelajaran Daring,” *Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence)*, vol. 5, no. 1, pp. 50–57, Apr. 2025, doi: 10.55382/jurnalpustakaai.v5i1.901.

[20] C. Lafado Yesa, N. A. Putri, and D. S. Noer, “243 | Designing an Academic Website User Interface Using the Design Thinking Method (Caesareano Lafado Yesa) Designing an Academic Website User Interface Using the Design Thinking Method,” *Jurnal Ilmu Sosial dan Pendidikan (JISIP)*, vol. 8, no. 2, pp. 2598–9944, 2024, doi: 10.58258/jisip.v7i1.6726/http.

[21] M. Wildan Yasykur and D. A. Efrilianda, “Evaluation of User Interface and User Experience on Solo Destination App using System Usability Scale and Human-centered Design Methods ARTICLE HISTORY,” *Journal of Advances in Information Systems and Technology*, vol. 6, no. 2, pp. 183–199, 2024.

[22] A. R. Gunawan, R. N. Alif, A. L. Muhajirin, N. I. Iryani, and M. F.

Ramadhan, “Pentingnya Pemanfaatan Framework Dalam Membangun Sebuah Aplikasi Perangkat Lunak,” *JRIIN :Jurnal Riset Informatika dan Inovasi*, vol. 2, no. 1, pp. 38–43, Jun. 2024, Accessed: Jan. 10, 2026. [Online]. Available: https://jurnalmahasiswa.com/index.php/jriin/article/view/1035

[23] S. Tewari, *Deploying Machine Learning Models: A Comparative Study of Flask, FastAPI, and Streamlit*. 2025. doi: 10.13140/RG.2.2.24413.06880.

[24] R. Akbar Wibowo and D. indrayana, “Implementasi Metode Waterfall dalam Pengembangan Aplikasi Toko Online Bebasis Web di Cv Aishastore.Id,” 2025.