



Analisis Kemiripan *Source Code Project* Menggunakan Metode *CodeBERT* dan *Winnowing Algorithm*

INFO PENPULIS	INFO ARTIKEL
Fauzan Azhari Rahman Universitas Muhammadiyah Makassar 105841109622@student.unismuh.ac.id	ISSN: 3026-3603 Vol. 4, No. 1 April 2026 http://jurnal.ardenjaya.com/index.php/ajst
Rizki Yusliana Bakti Universitas Muhammadiyah Makassar rizkiyusliana@unismuh.ac.id	
Muhyiddin A.M Hayat Universitas Muhammadiyah Makassar muhyiddin@unismuh.ac.id	

© 2026 Arden Jaya Publisher All rights reserved

Saran Penulisan Referensi:

Rahman, F. A., Bakti, R. Y., & Hayat, M. A. M., (2026). Analisis Kemiripan Source Code Project Menggunakan Metode CodeBERT dan Winnowing Algorithm. *Arus Jurnal Sains dan Teknologi*, 4 (1),111-117.

Abstrak

Pemeriksaan keaslian kode pada tugas *Capstone Project* umumnya masih dilakukan secara manual, sehingga tidak efisien dan berisiko melewatkan kasus plagiarisme yang disamarkan melalui *refactoring* atau penggantian nama variabel. Penelitian ini bertujuan menerapkan kombinasi metode *CodeBERT* dan *Winnowing Algorithm* untuk mendeteksi kemiripan kode sumber secara semantik dan tekstual, serta mengintegrasikan fitur tersebut ke dalam sistem pengumpulan tugas *Capstone Project* mahasiswa Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar. *CodeBERT* digunakan untuk menganalisis kemiripan semantik, sedangkan *Winnowing Algorithm* digunakan untuk mendeteksi kemiripan tekstual berbasis fingerprint k-gram. Hasil dari kedua metode digabungkan untuk menghasilkan penilaian kemiripan yang lebih komprehensif. Pengujian dilakukan terhadap 17 proyek valid yang menghasilkan 136 pasangan unik. Hasil analisis menunjukkan 16 pasangan termasuk kategori Plagiarisme Kuat, 30 pasangan Mirip Semantik, 30 pasangan Mirip Tekstual, dan 60 pasangan Normal. Selain itu, seluruh 11 skenario *black box testing* berhasil dijalankan dengan tingkat keberhasilan 100%. Hasil ini menunjukkan bahwa kombinasi *CodeBERT* dan *Winnowing Algorithm* efektif diterapkan untuk mendukung analisis kemiripan kode pada lingkungan akademik.

Kata Kunci: Deteksi Plagiarisme, Kode Sumber, *CodeBERT*, *Winnowing Algorithm*, *Capstone Project*

Abstract

Source code originality checking in Capstone Project assignments is generally still conducted manually, making the process inefficient and prone to overlooking plagiarism cases disguised through refactoring or variable renaming. This study aims to implement a combination of CodeBERT and Winnowing Algorithm to detect source code similarity both semantically and textually, and to integrate this feature into the Capstone Project submission system for students of the Informatics Study Program, Faculty of Engineering, Universitas Muhammadiyah Makassar. CodeBERT is used to analyze semantic similarity, while the Winnowing Algorithm is used to detect textual similarity based on k-gram fingerprints. The outputs of both methods are combined to provide a more comprehensive similarity assessment. Testing was conducted on 17 valid projects resulting in 136 unique pairs. The results showed that 16 pairs were categorized as Strong Plagiarism, 30 pairs as Semantic Similarity, 30 pairs as Textual Similarity, and 60 pairs as Normal. In addition, all 11 black box testing scenarios passed successfully with a 100% success rate. These findings indicate that the combination of CodeBERT and Winnowing Algorithm is effective for supporting source code similarity analysis in academic environments.

Keywords: Plagiarism Detection, Source Code, CodeBERT, Winnowing Algorithm, Capstone Project

A. Pendahuluan

Perkembangan platform repositori terbuka seperti *GitHub*, *GitLab*, dan *Stack Overflow* telah mempermudah akses terhadap berbagai sumber kode yang dapat dipelajari, digunakan, dan dimodifikasi oleh pengembang perangkat lunak (Zakeri-Nasrabadi et al., 2023). Di satu sisi, keterbukaan ini mendukung proses pembelajaran dan inovasi. Namun, di sisi lain, kondisi tersebut juga meningkatkan potensi terjadinya plagiarisme kode sumber, baik melalui penyalinan langsung maupun melalui penyamaran seperti perubahan nama variabel, pengaturan ulang struktur kode, dan *refactoring* (Zakeri-Nasrabadi et al., 2023).

Dalam konteks akademik, plagiarisme kode sumber menjadi persoalan penting karena dapat memengaruhi objektivitas penilaian tugas pemrograman dan *Capstone Project*. Pemeriksaan secara manual sering kali tidak efektif ketika jumlah proyek yang dinilai cukup banyak dan setiap proyek memiliki banyak berkas kode. Selain itu, pendekatan manual juga sulit mendeteksi kemiripan logika program yang tidak tampak secara langsung pada struktur teks permukaan.

Metode berbasis *fingerprinting* seperti *Winnowing Algorithm* telah banyak digunakan untuk mendeteksi kemiripan tekstual pada dokumen maupun kode sumber. Metode ini efektif dalam mengenali pola penyalinan literal karena bekerja dengan membandingkan *fingerprint* hasil pembentukan k-gram (Schleimer et al., 2003). Namun, pendekatan ini memiliki keterbatasan ketika kode telah mengalami modifikasi struktur, perubahan nama variabel, atau penyusunan ulang blok program. Sebaliknya, model berbasis *Transformer* seperti *CodeBERT* mampu memahami representasi semantik kode sumber sehingga lebih baik dalam mendeteksi kemiripan logika program meskipun bentuk teksnya telah berubah (Feng et al., 2020).

Beberapa penelitian menunjukkan bahwa penggabungan pendekatan tekstual dan semantik memberikan hasil yang lebih baik dibandingkan penggunaan metode tunggal (Hidayat et al., 2020; Mangundap et al., 2022). Akan tetapi, sebagian besar penelitian sebelumnya masih menerapkan kedua pendekatan tersebut secara terpisah atau belum mengintegrasikannya langsung ke dalam sistem pengumpulan tugas berbasis web.

Berdasarkan kondisi tersebut, penelitian ini dilakukan untuk menerapkan kombinasi *CodeBERT* dan *Winnowing Algorithm* dalam mendeteksi kemiripan *source code project* mahasiswa, sekaligus mengintegrasikan fitur tersebut ke dalam sistem pengumpulan tugas *Capstone Project* pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar. Penelitian ini diharapkan dapat membantu dosen dalam mengidentifikasi indikasi kemiripan kode secara lebih cepat, sistematis, dan objektif.

B. Metodologi

1. Pengumpulan Data

Data penelitian berupa proyek *Capstone Project* mahasiswa Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar yang tersimpan dalam repositori dan dapat diakses melalui sistem. Setelah proses seleksi, diperoleh 17 proyek valid yang digunakan sebagai data analisis. Dari seluruh proyek tersebut dibentuk 136 pasangan unik untuk dibandingkan satu sama lain.

2. Preprocessing

Pada tahap ini, kode sumber dari setiap proyek dibersihkan terlebih dahulu dengan menghapus komentar, *whitespace* berlebih, dan karakter yang tidak relevan. Tahap ini dilakukan agar proses analisis menjadi lebih konsisten dan fokus pada isi kode yang benar-benar merepresentasikan logika program.

3. Analisis Kemiripan Semantik

berikutnya adalah analisis kemiripan semantik menggunakan *CodeBERT*. Setiap potongan kode diubah menjadi representasi *embedding* sehingga sistem dapat mengenali kemiripan makna dan logika program (Devlin et al., 2019; Feng et al., 2020; Akbar & Ningrum, 2025).

4. Analisis Kemiripan Tekstual

Secara paralel, kode sumber juga dianalisis menggunakan *Winnowing Algorithm*. Pada tahap ini, sistem membentuk *k-gram*, menghasilkan *hash*, menyusun *window*, dan memilih *fingerprint* untuk setiap berkas kode. Hasilnya digunakan untuk menghitung tingkat kemiripan tekstual antar proyek berdasarkan kecocokan pola karakter dan struktur literal kode dengan membandingkan *fingerprint* hasil pembentukan *k-gram* (Schleimer et al., 2003; Ramli et al., 2021; Mangundap et al., 2022; Hidayat et al., 2020).

5. Penggabungan Hasil Analisis

Setelah skor kemiripan semantik dan tekstual diperoleh, kedua hasil tersebut digabungkan untuk menghasilkan skor akhir yang lebih komprehensif. Pendekatan ini digunakan agar sistem tidak hanya sensitif terhadap penyalinan literal, tetapi juga mampu mengenali kemiripan logika program yang telah dimodifikasi.

6. Klasifikasi Kategori

Berdasarkan hasil analisis, setiap pasangan proyek diklasifikasikan ke dalam empat kategori, yaitu Plagiarisme Kuat, Mirip Semantik, Mirip Tekstual, dan Normal. Klasifikasi ini digunakan untuk membantu interpretasi hasil oleh dosen dan memudahkan proses peninjauan lanjut terhadap pasangan proyek yang terindikasi memiliki kemiripan tinggi.

7. Integrasi Sistem

Modul analisis yang telah dibangun kemudian diintegrasikan ke dalam sistem pengumpulan tugas *Capstone Project* berbasis *web* (Putra, 2022; Setiawan et al., 2020). Hasil analisis ditampilkan pada *dashboard* dosen dalam bentuk skor kemiripan, kategori indikasi, serta potongan kode yang teridentifikasi mirip. Pada sisi mahasiswa, sistem menampilkan ringkasan hasil analisis untuk proyek yang diunggah.

8. Pengujian Sistem

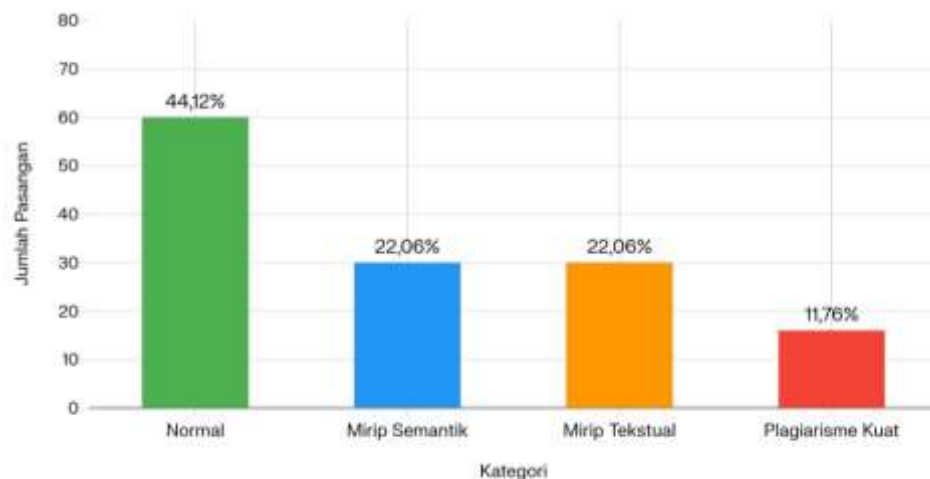
Pengujian dilakukan menggunakan metode *black box testing* untuk memastikan seluruh fungsi berjalan sesuai kebutuhan. Pengujian meliputi layanan analisis kemiripan, analisis langsung pada pasangan kode, analisis *batch* seluruh proyek, serta integrasi tampilan antarmuka pada *dashboard* dosen dan mahasiswa. Total terdapat 11 skenario pengujian yang digunakan dalam penelitian ini.



Bagan 1 Alur Metodologi Penelitian

C. Hasil dan Pembahasan

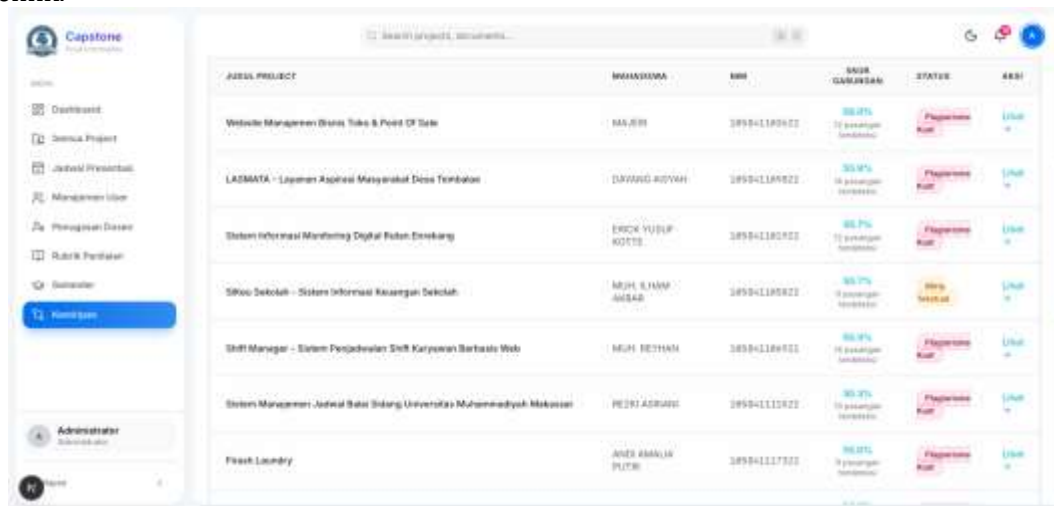
Hasil analisis terhadap 17 proyek valid menghasilkan 136 pasangan unik yang selanjutnya diklasifikasikan ke dalam empat kategori kemiripan. Dari seluruh pasangan tersebut, sebanyak 16 pasangan termasuk kategori Plagiarisme Kuat, 30 pasangan termasuk Mirip Semantik, 30 pasangan termasuk Mirip Tekstual, dan 60 pasangan termasuk kategori Normal. Hasil ini menunjukkan bahwa sistem mampu membedakan variasi tingkat kemiripan kode secara lebih rinci, tidak hanya menjadi dua kelas serupa dan tidak serupa.



Gambar 1 Distribusi Kategori Indikasi Kemiripan

Distribusi hasil tersebut menunjukkan bahwa kombinasi *CodeBERT* dan *Winnowing Algorithm* memberikan kontribusi yang saling melengkapi. Kategori Mirip Semantik menunjukkan bahwa terdapat pasangan proyek yang memiliki kesamaan logika program meskipun tidak terlalu mirip secara tekstual. Sebaliknya, kategori Mirip Tekstual menunjukkan adanya kesamaan literal pada struktur kode meskipun tidak selalu identik secara semantik. Dengan demikian, pendekatan gabungan ini lebih informatif dibandingkan metode tunggal.

Pada implementasinya, sistem berhasil menampilkan hasil analisis secara lengkap pada *dashboard* dosen. Informasi yang ditampilkan meliputi skor kemiripan semantik, skor kemiripan tekstual, skor gabungan, kategori indikasi, serta potongan kode yang terdeteksi mirip. Fitur ini memudahkan dosen dalam melakukan peninjauan awal terhadap proyek yang dicurigai memiliki kemiripan tinggi. Di sisi lain, mahasiswa dapat melihat ringkasan hasil analisis pada proyek miliknya, sehingga sistem juga berfungsi sebagai sarana umpan balik akademik.



ASAS PROJECT	USERNAME	EMAIL	SKOR GABUNGAN	STATUS	Aksi
Website Manajemen Bisnis Toko & Point Of Sale	RAJIB	RAJIB@CIIIRZZ	88.47%	Pengiriman Full	Ubat
LAMATA - Layanan Aplikasi Manajemen Dosa Tambatan	DUNYU ADYAN	DUNYU@CIIIRZZ	88.97%	Pengiriman Full	Ubat
Sistem Informasi Monitoring Digital Robot Berakal	ERICK YUSUF KOTIS	ERICK@CIIIRZZ	88.7%	Pengiriman Full	Ubat
Silau Sekolah - Sistem Informasi Keuangan Sekolah	MEHA S HANN AGAS	MEHA@CIIIRZZ	88.7%	Pengiriman Full	Ubat
Shift Manager - Sistem Penjadwalan Shift Karyawan Berbasis Web	MEHA DEYHAN	MEHA@CIIIRZZ	88.9%	Pengiriman Full	Ubat
Sistem Manajemen Jadwal Bekerja Siswa Universitas Muhammadiyah Makassar	REDA ADRIAN	REDA@CIIIRZZ	88.9%	Pengiriman Full	Ubat
Franchise Laundry	ANAS AMALIA PUTRI	ANAS@CIIIRZZ	88.8%	Pengiriman Full	Ubat

Gambar 2 Tampilan *Dashboard* Deteksi Kemiripan
Tabel 1 Ringkasan Hasil Pengujian *Black Box Testing*

No	Area Pengujian	Skenario	Berhasil	Keberhasilan
1	Layanan analisis kemiripan	1	1	100%
2	Analisis kemiripan langsung	4	4	100%
3	Analisis <i>batch</i> dan konsistensi data	4	4	100%
4	Integrasi antarmuka	2	2	100%
Total		11	11	100%

Hasil pengujian menunjukkan bahwa seluruh 11 skenario *black box testing* dinyatakan berhasil. Keberhasilan ini mencakup pengujian layanan analisis kemiripan, analisis pasangan kode secara langsung, pemrosesan *batch* terhadap seluruh proyek, dan integrasi tampilan antarmuka sistem. Temuan ini menunjukkan bahwa modul analisis yang dibangun tidak hanya bekerja secara konseptual, tetapi juga berjalan dengan baik ketika diterapkan pada sistem pengumpulan tugas yang sesungguhnya.

Secara keseluruhan, hasil penelitian memperlihatkan bahwa kombinasi *CodeBERT* dan *Winnowing Algorithm* mampu meningkatkan kualitas analisis kemiripan kode sumber. Sejalan dengan temuan Virginia dan Alamsyah (2026) dan Zidan & Setiawan (2025) yang menunjukkan keunggulan pendekatan keduanya. *CodeBERT* unggul dalam mengenali kesamaan logika program, sedangkan *Winnowing* efektif dalam mendeteksi kesamaan literal pada potongan kode. Ketika keduanya digabungkan, sistem menghasilkan penilaian yang lebih seimbang dan dapat membantu proses evaluasi keaslian proyek secara lebih objektif di lingkungan akademik.

D. Kesimpulan

1. Kombinasi metode *CodeBERT* dan *Winnowing Algorithm* berhasil diterapkan untuk mendeteksi kemiripan *source code project* mahasiswa Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar secara semantik dan tekstual. Hasil analisis terhadap 136 pasangan proyek menunjukkan bahwa sistem mampu mengelompokkan pasangan proyek ke dalam kategori Plagiarisme Kuat, Mirip Semantik, Mirip Tekstual, dan Normal secara jelas dan informatif.

2. Fitur analisis kemiripan kode berbasis *CodeBERT* dan *Winnowing Algorithm* berhasil diintegrasikan ke dalam sistem pengumpulan tugas *Capstone Project*. Hasil pengujian membuktikan bahwa seluruh fungsi sistem berjalan dengan baik dan hasil analisis dapat ditampilkan secara lengkap pada *dashboard* dosen maupun mahasiswa.

E. Referensi

- Akbar, M. I., & Ningrum, N. K. (2025). Identification of source code plagiarism using a natural language processing (NLP) approach based on code writing style analysis. *Journal of Applied Informatics and Computing (JAIC)*, 9(6), 1–10. <https://jurnal.polibatam.ac.id/index.php/JAIC>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)* (Vol. 1, pp. 4171–4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
- Ebrahim, F., & Joy, M. (2023). Source code plagiarism detection with pre-trained model embeddings and automated machine learning. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2023)* (pp. 301–309). INCOMA Ltd. https://doi.org/10.26615/978-954-452-092-2_034
- Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., & Zhou, M. (2020). CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 1536–1547). Association for Computational Linguistics. <https://doi.org/10.48550/arXiv.2002.08155>
- Hidayat, W., Utami, E., & Hartanto, A. D. (2020). Pemilihan parameter terbaik pada algoritma winnowing dalam mendeteksi tingkat kesamaan dokumen bahasa Indonesia. *Citec Journal*, 7(2), 256–265. <https://citec.amikom.ac.id/main/index.php/citec/article/view/256>
- Kurniawan, A., Erlangga, E., & Tanjung, T. (2022). Analisis kinerja algoritma winnowing pada pendeteksian plagiarisme. *Explore: Jurnal Sistem Informasi dan Telematika*, 13(2), 125–132. <https://www.researchgate.net/publication/366662136>
- Mangundap, G. H. O., Sujaini, H., & Pratiwi, H. S. (2022). Implementasi algoritma winnowing pada aplikasi pendeteksi kemiripan dokumen. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 8(1), 47–54. <https://doi.org/10.26418/jp.v8i1.47822>
- NU Editorial Contributors. (2023, Juni 16). *What is a capstone project?* National University. <https://www.nu.edu/blog/what-is-a-capstone-project/>
- Putra, Y. D. E. (2022). Deteksi plagiarisme tugas akhir mahasiswa dengan menggunakan metode cosine similiarity Tugas akhir sarjana, Universitas Islam Sultan Agung Semarang. *Repository UNISSULA*. <https://repository.unissula.ac.id/29641/>
- Ramli, M. S., Cokrowibowo, S., & Rustan, M. F. (2021). Uji plagiarisme pada tugas mahasiswa menggunakan algoritma winnowing. *Journal of Applied Computer Science and Technology (JACOST)*, 2(2), 108–112. <https://doi.org/10.52158/jacost.v2i2.177>
- Schleimer, S., Wilkerson, D. S., & Aiken, A. (2003). Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD '03)* (pp. 76–85). Association for Computing Machinery. <https://doi.org/10.1145/872757.872770>
- Setiawan, R., Kurniawan, A., & Sari, D. P. (2020). Analisis dan perancangan software pengukuran kemiripan kode program. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 2(3), 285–296. <https://doi.org/10.28926/ilkomnika.v2i2.64>
- Sugiono, S., Herwin, H., Hamdani, H., & Erlin, E. (2018). Aplikasi pendeteksi tingkat kesamaan dokumen teks: Algoritma Rabin-Karp vs. winnowing. *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, 9(1), 82–93. <https://doi.org/10.31849/digitalzone.v9i1.1242>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)* (pp. 5998–6008). Curran Associates. <https://doi.org/10.48550/arXiv.1706.03762>
- Virginia, C., & Alamsyah, D. (2026). Plagiarism detection in English academic documents using a lexical semantic hybrid and support vector machine. *INOVTEK Polbeng – Seri Informatika*, 11 (1), 96–107. <https://doi.org/10.35314/2zz12581>

- Zakeri-Nasrabadi, M., Parsa, S., Ramezani, M., Roy, C., & Ekhtiarzadeh, M. (2023). A systematic literature review on source code similarity measurement and clone detection: Techniques, applications, and challenges. *Journal of Systems and Software*, 204, 111796. <https://doi.org/10.1016/j.jss.2023.111796>
- Zidan, M., & Setiawan, R. (2025). Implementasi algoritma Rabin-Karp dalam pendeteksian plagiarisme pada dokumen makalah mahasiswa. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi (JIMIK)*, 6(1), 45-54. <https://journal.stmiki.ac.id/index.php/jimik/article/view/1191>