

DAFTAR PUSTAKA

- [1] D. Kenian Irsya and T. Faturrohman, "The Demographical Analysis Of Indonesian Buy-Now-Pay-Later Users Financial Wellbeing," *Ekombis Review: Jurnal Ilmiah Ekonomi dan Bisnis*, vol. 12, no. 4, pp. 3861–3872, 2024, doi: 10.37676/ekombis.v12i4.
- [2] M. A. Hidayat and P. Rudito, "The Analysis of the 'Buy Now, Pay Later' Use Intention in Indonesia," *International Journal of Current Science Research and Review*, vol. 5, no. 7, pp. 2384–2388, 2022, doi: 10.47191/ijcsrr/V5-i7-19.
- [3] T. Berg, V. Burg, J. Keil, and M. Puri, "The Economics of 'Buy Now, Pay Later': A Merchant's Perspective," Cambridge, 33152, Nov. 2024.
- [4] G. Ardiansyah and C. Kusuma Dewi, "Pengaruh Perceived Benefit, Social Influence, Perceived Risk Dan Habit Terhadap Behavioural Intention Dan Kaitannya Dengan Usage Behaviour Layanan Buy-Now, Pay-Later (BNPL) Pada Gen-Z Dan Milenial Di Indonesia," 2024, pp. 5458–5468.
- [5] S. Dea Anisa and C. Saepuloh, "Pengaruh Penggunaan Paylater Terhadap Perilaku Konsumtif Diera Gen Z (Studi Kasus Pada Masyarakat Kelurahan Cipatik)," in *Festival Riset Ilmiah Manajemen & Akuntansi "FRIMA"*, 2024, pp. 501–509.
- [6] "[LIVE] Konferensi Pers Asesmen Sektor Jasa Keuangan dan Kebijakan OJK Hasil RDKB Maret 2025," Otoritas Jasa Keuangan. Accessed: Mar. 13, 2025. [Online]. Available: <https://www.youtube.com/watch?v=zXcuJi2Y828&t=1807s>
- [7] L. Parwati and T. Dwi Cahyono, "Pengaruh Kemudahan Transaksi, Keamanan Transaksi, Gaya Hidup Terhadap Perilaku Pembelian Tidak Terduga (Studi Kasus Pengguna Shopee Paylater di Kec.Sumbawa)," *SEMINAR NASIONAL MANAJEMEN INOVASI*, vol. 2, no. 1, pp. 241–250, 2024, [Online]. Available: <https://conference.uts.ac.id/index.php/Student>
- [8] S. I. Purwati and F. Setiawan, "Pengaruh Kemudahan, Manfaat, Risiko dan Religiusitas terhadap Keputusan Penggunaan Spaylater," *Journal of Management, Accounting, and Administration*, vol. 1, no. 3, pp. 181–189, Mar. 2025, doi: 10.52620/jomaa.v1i3.114.
- [9] V. Juita, V. Pujani, R. Rahim, and R. Rahayu, "Understanding Impulsive Buying Behaviour Among Buy Now Pay Later (BNPL) Users and Its Implication for Overconsumption and The Environment," *Management Analysis Journal*, vol. 12, no. 4, pp. 433–440, 2023, [Online]. Available: <http://maj.unnes.ac.id>
- [10] A. Nurajizah and F. Indriani, "Analisis Pengaruh Persepsi Kemudahan Penggunaan dan Persepsi Risiko terhadap Minat Beli Ulang Konsumen dengan Kepercayaan sebagai Variabel Perantara (Studi Pada Layanan Shopee Paylater)," *DIPONEGORO JOURNAL OF MANAGEMENT*, vol. 12, no. 1, pp. 1–14, 2023.

- [11] H. Fajrussalam, Ihsanudin, T. Luthfi, I. Sallsabila, and R. Puspita Sari, "Persepsi Masyarakat Terhadap Paylater Dalam Online Shopping," *Jurnal Pendidikan Tambusa*, vol. 6, no. 2, pp. 8886–8893, 2022.
- [12] Waluyo, Y. Arif Nurohman, and R. Sari Qurniawati, "Buy Now, Pay Later: Apakah Paylater Memengaruhi Pembelian Impulsif Generasi Muda Muslim?," *Among Makarti*, vol. 15, no. 3, pp. 319–338, 2022.
- [13] D. Ayu, Z. Lia, and S. Lu'ay Natswa, "Buy-Now-Pay-Later (BNPL): Generation Z's Dilemma on Impulsive Buying and Overconsumption Intention," 2021.
- [14] E. P. Adamansyah and A. Yudhistira, "Evaluasi Opini Publik di Media Sosial X terhadap Kebijakan Pajak Pertambahan Nilai 12% di Indonesia Menggunakan Naive Bayes dan Decision Tree," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 3, pp. 831–843, Mar. 2025, doi: 10.52436/1.jpti.710.
- [15] D. K. Setyadi, "Peran Twitter dalam Digital Customer Relation Management di Industri Perbankan," *Journal Communication Spectrum*, vol. 9, no. 2, pp. 110–124, Feb. 2020, doi: 10.36782/jcs.v9i2.1952.
- [16] T. Srikanth, S. Bhanu Sri, S. Dedeepya, and P. Deepthi, "Unveiling Insights with Twitter Data: Exploring Trends, Sentiments, and Predictions Through Social Media Mining," *Journal for Educators, Teachers and Trainers JETT*, vol. 15, no. 5, pp. 355–365, 2024, doi: 10.47750/jett.2024.15.05.35.
- [17] S. Saepudin, S. Widiastuti, and C. Irawan, "Sentiment Analysis of Social Media Platform Reviews Using the Naïve Bayes Classifier Algorithm," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 2, pp. 236–243, Jul. 2023, doi: 10.32736/sisfokom.v12i2.1650.
- [18] F. Fajris and Imelda, "Analisis Sentimen Terhadap Komentar Video Short YouTube Mengenai Paylater Menggunakan Algoritma Bert," *Jurnal TICOM: Technology of Information and Communication*, vol. 13, no. 2, pp. 79–84, 2025.
- [19] R. Rizaldi and R. Aryanti, "Analisis Sentimen Pengguna Terhadap Aplikasi Indodana Di Google Play Store Menggunakan Metode Naive Bayes Classifier," *Journal of Informatics Management and Information Technology*, vol. 3, no. 4, pp. 98–105, 2024, doi: 10.47065/jimat.v4i3.400.
- [20] F. Junian Wahidna and P. Nerisafitra, "Analisis Sentimen Pengguna Sistem Pay Later Menggunakan Support Vector Machine Metode Pembobotan Lexicon," *Journal of Informatics and Computer Science*, vol. 04, 2023.
- [21] A. Safira and F. N. Hasan, "Analisis Sentimen Masyarakat terhadap Paylater Menggunakan Metode Naive Bayes Classifier," *ZONAsi: Jurnal Sistem Informasi*, vol. 5, no. 1, pp. 59–70, Jan. 2023, doi: 10.31849/zn.v5i1.12856.

- [22] I. Mahayani, D. A. R., and M. E. Supriyadi, "Analisis Sentimen Twitter Terhadap Pembayaran ShopeePayLater Pada Aplikasi Belanja Online (Shopee) Menggunakan Metode Lexicon Based Dan Naive Bayes Classifier," *Jurnal Ilmiah Komputasi*, vol. 19, no. 4, pp. 545–558, Dec. 2020, doi: 10.32409/jikstik.19.4.293.
- [23] A. D. Widianoro and B. Harnadi, "Analysis of User Comments Based on Topic Modeling using LDA on OVO E-Wallet," *SISFORMA: Journal of Information Systems*, vol. 9, no. 2, pp. 99–107, Jan. 2022, doi: 10.24167/sisforma.v9i2.5343.
- [24] J. Setiawan, A. Milenia, and A. Faza, "An Integrated Approach for Sentiment Analysis and Topic Modeling of a Digital Bank in Indonesia using Naïve Bayes and Latent Dirichlet Allocation Algorithms on Social Media Data," in *2023 4th International Conference on Big Data Analytics and Practices, IBDAP 2023*, 2023. doi: 10.1109/IBDAP58581.2023.10271956.
- [25] F. Furqan, C. H. Pangaribuan, S. Tjhin, and C. Luhur, "Buy now, pray later? Demystifying unrestrained spending in the age of BNPL," *Jurnal Siasat Bisnis*, pp. 15–26, Jan. 2025, doi: 10.20885/jsb.vol29.iss1.art2.
- [26] "Investigating Drivers of Continued BNPL Service Usage and Subjective Well-being Impacts Among China's MZ Generation," *Journal of System and Management Sciences*, Jun. 2024, doi: 10.33168/jsms.2024.0826.
- [27] A. Raj, "A Review of the Literature on Natural Language Processing (NLP) to Promote Industry Advancement," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 08, no. 05, pp. 1–5, May 2024, doi: 10.55041/IJSREM33210.
- [28] M. R. Sudrajat and M. Zakariyah, "Penerapan Natural Language Processing dan Machine Learning untuk Prediksi Stres Siswa SMA Berdasarkan Analisis Teks," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 3, Dec. 2024, doi: 10.47065/bits.v6i3.6180.
- [29] D. N. Yagamurthy, "Advancements in Natural Language Processing (NLP) and Its Applications in Voice Assistants and Chatbots," *Journal of Artificial Intelligence & Cloud Computing*, pp. 1–6, Dec. 2023, doi: 10.47363/JAICC/2023(2)326.
- [30] W. Hardi, W. A. Kusuma, and S. Basuki, "Pengelompokan Topik Dokumen Berbasis Text Mining dengan Algoritma K-Means: Studi Kasus pada Dokumen Kedutaan Besar Australia Jakarta," *VISI PUSTAKA*, vol. 21, no. 1, pp. 67–76, Apr. 2019.
- [31] Nofiyani, Wulandari, and Y. P. Dewi, "Meningkatkan Efektivitas dan Efisiensi Implementasi Electronic Data Processing Pembobotan Kata pada Klasifikasi Text Minging," *JIKI (JURNAL ILMU KOMPUTER DAN INFORMATIKA)*, vol. 5, no. 1, pp. 48–56, Jul. 2024.
- [32] A. Arham, E. R. Swedia, M. Cahyanti, and M. R. D. Septian, "Implementasi Sentiment Analysis pada Opini Masyarakat Indonesia di Twitter terhadap Virus

- COVID-19 Varian Omicron dengan Algoritma Naive Bayes, Decision Tree, dan Support Vector Machine,” *Sebatik*, vol. 26, no. 2, pp. 565–572, Dec. 2022, doi: 10.46984/sebatik.v26i2.1961.
- [33] L. Ardiani, H. Sujaini, and T. Tursina, “Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak,” *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 2, p. 183, Apr. 2020, doi: 10.26418/justin.v8i2.36776.
- [34] C. H. Hill, J. E. Fresneda, and M. Anandarajan, “The wisdom of the lexicon crowds: leveraging on decades of lexicon-based sentiment analysis for improved results,” *J Big Data*, vol. 12, no. 1, p. 129, May 2025, doi: 10.1186/s40537-025-01186-7.
- [35] T. Islam *et al.*, “Lexicon and Deep Learning-Based Approaches in Sentiment Analysis on Short Texts,” *Journal of Computer and Communications*, vol. 12, no. 01, pp. 11–34, 2024, doi: 10.4236/jcc.2024.121002.
- [36] N. A. Helal, A. Hassan, N. L. Badr, and Y. M. Afify, “A contextual-based approach for sarcasm detection,” *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-65217-8.
- [37] Md. T. H. K. Tusar and Md. T. Islam, “A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data,” Oct. 2021, [Online]. Available: <http://arxiv.org/abs/2110.00859>
- [38] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “Sentiment analysis based on deep learning: A comparative study,” *Electronics (Switzerland)*, vol. 9, no. 3, Mar. 2020, doi: 10.3390/electronics9030483.
- [39] A. S. Talaat, “Sentiment analysis classification system using hybrid BERT models,” *J Big Data*, vol. 10, no. 1, Dec. 2023, doi: 10.1186/s40537-023-00781-w.
- [40] A. T. Mahmood, S. S. Kamaruddin, R. K. Naser, and M. M. Nadzir, “A combination of lexicon and machine learning approaches for sentiment analysis on facebook,” *Journal of System and Management Sciences*, vol. 10, no. 3, pp. 140–150, 2020, doi: 10.33168/JSMS.2020.0310.
- [41] J. Camacho-Collados and M. T. Pilehvar, “On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis,” Jul. 2017, [Online]. Available: <http://arxiv.org/abs/1707.01780>
- [42] M. Guo *et al.*, “Normal Workflow and Key Strategies for Data Cleaning Toward Real-World Data: Viewpoint,” *Interact J Med Res*, vol. 12, p. e44310, Sep. 2023, doi: 10.2196/44310.
- [43] T. Agustiranti, A. Khalfani Izzati Kurdiana, B. Al Ghiffari, E. Dwi Juniar, and D. Gita Purnama, “Penerapan Naive Bayes Terhadap Sentimen Analisis Media Sosial

- Twitter Pengguna Kereta Cepat Jakarta-Bandung (Whoosh)," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 297–305, 2024.
- [44] B. Hakim, "Analisa Sentimen Data Text Preprocessing Pada Data Mining Dengan Menggunakan Machine Learning," *JBASE - Journal of Business and Audit Information Systems*, vol. 4, no. 2, Aug. 2021, doi: 10.30813/jbase.v4i2.3000.
 - [45] S. M. Mohammad, "Sentiment Analysis: Automatically Detecting Valence, Emotions, and Other Affectual States from Text," May 2020, [Online]. Available: <http://arxiv.org/abs/2005.11882>
 - [46] T. A. Nurlanuly, "Sentiment Analysis of Texts from Social Networks Based on Machine Learning Methods for Monitoring Public Sentiment," 2024.
 - [47] A. Ethan and D. K. Chikwari, "Real-Time Sentiment Analysis in Social Media Using Deep Learning International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence Real-Time Sentiment Analysis in Social Media Using Deep Learning," *International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence*, vol. 14, pp. 1–16, Dec. 2023, [Online]. Available: <https://www.researchgate.net/publication/391624442>
 - [48] S. S. Shanto, Z. Ahmed, N. Hossain, A. Roy, and A. I. Jony, "Binary vs. Multiclass Sentiment Classification for Bangla E-commerce Product Reviews: A Comparative Analysis of Machine Learning Models," *International Journal of Information Engineering and Electronic Business*, vol. 15, no. 6, pp. 48–63, Dec. 2023, doi: 10.5815/ijieeb.2023.06.04.
 - [49] Y. Xiao, C. Li, M. Thüerer, Y. Liu, and T. Qu, "Towards Lean Automation: Fine-Grained sentiment analysis for customer value identification," *Comput Ind Eng*, vol. 169, Jul. 2022, doi: 10.1016/j.cie.2022.108186.
 - [50] K. Ahmed *et al.*, "Breaking down linguistic complexities: A structured approach to aspect-based sentiment analysis," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, Sep. 2023, doi: 10.1016/j.jksuci.2023.101651.
 - [51] M. I. Syafaah and L. Lestandy, "Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) And LSTM (Long Short-Term Memory)," 2022. [Online]. Available: https://atapdata.ai/dataset/192/HIMPUNAN_DATA_E
 - [52] Chingmuankim and R. Jindal, "A comparative study of Naive Bayes Classifiers with improved technique on Text Classification," pp. 1–8, 2022.
 - [53] P. Saigal and V. Khanna, "Multi-category news classification using Support Vector Machine based classifiers," *SN Appl Sci*, vol. 2, no. 3, Mar. 2020, doi: 10.1007/s42452-020-2266-6.

- [54] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [55] Y. Chen, P. Liu, and C. P. Teo, "Regularised Text Logistic Regression: Key Word Detection and Sentiment Classification for Online Reviews," Sep. 2020, [Online]. Available: <http://arxiv.org/abs/2009.04591>
- [56] A. Singh *et al.*, "Evolving Long Short-Term Memory Network-Based Text Classification," *Comput Intell Neurosci*, vol. 2022, p. 1, Feb. 2022, doi: 10.1155/2022/4725639.
- [57] B. Grici^{ut} e, L. Han, and G. Nenadic, "Topic Modelling of Swedish Newspaper Articles about Coronavirus: a Case Study using Latent Dirichlet Allocation Method," Aug. 2023.
- [58] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," Mar. 2022.
- [59] M. Yousef and D. Voskergian, "TextNetTopics: Text Classification Based Word Grouping as Topics and Topics' Scoring," *Front Genet*, vol. 13, Jun. 2022, doi: 10.3389/fgene.2022.893378.
- [60] Y. N. Bisoumi *et al.*, "Papua dalam Perspektif Komentar Youtube: Studi Pemodelan Topik dan Analisis Sentimen dengan Pendekatan Text Mining," *Seminar Nasional Sains Data*, vol. 2024, [Online]. Available: <https://www.youtube.co>
- [61] V. K. Garbhapu, "A comparative analysis of Latent Semantic analysis and Latent Dirichlet allocation topic modeling methods using Bible data," *Indian J Sci Technol*, vol. 13, no. 44, pp. 4474–4482, Nov. 2020, doi: 10.17485/IJST/v13i44.1479.
- [62] D. Angelov, "Top2Vec: Distributed Representations of Topics," Aug. 2020.
- [63] T. Shaik, X. Tao, C. Dann, H. Xie, Y. Li, and L. Galligan, "Sentiment analysis and opinion mining on educational data: A survey," Feb. 2023, doi: 10.1016/j.nlp.2022.100003.
- [64] A. Rufaida, A. Permanasari, and N. Setiawan, "Lexicon-Based Sentiment Analysis Using Inset Dictionary: A Systematic Literature Review," *European Alliance for Innovation n.o.*, Jul. 2023. doi: 10.4108/eai.5-10-2022.2327474.
- [65] R. Suryanti and T. Prasetyaningrum, "Perbandingan Metode TF-IDF dan Bag of Words dalam Analisis Sentimen Diet Kopi Americano di Media Sosial Twitter Menggunakan Naïve Bayes," *Technology and Science (BITS)*, vol. 7, no. 1, 2025, doi: 10.47065/bits.v7i1.7244.

- [66] A. F. AlShammari, "Implementation of Keyword Extraction using Term Frequency-Inverse Document Frequency (TF-IDF) in Python," *Int J Comput Appl*, vol. 185, no. 35, pp. 9–14, Sep. 2023.
- [67] A. Addiga and S. Bagui, "Sentiment Analysis on Twitter Data Using Term Frequency-Inverse Document Frequency," *Journal of Computer and Communications*, vol. 10, no. 08, pp. 117–128, 2022, doi: 10.4236/jcc.2022.108008.
- [68] H. Liu, X. Chen, and X. Liu, "A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis," *IEEE Access*, vol. 10, pp. 32280–32289, 2022, doi: 10.1109/ACCESS.2022.3160172.
- [69] P. S. Ghatore, S. E. Hosseini, S. Pervez, M. J. Iqbal, and N. Shaukat, "Sentiment Analysis of Product Reviews Using Machine Learning and Pre-Trained LLM," *Big Data and Cognitive Computing*, vol. 8, no. 12, Dec. 2024, doi: 10.3390/bdcc8120199.
- [70] S. Kumar, P. P. Roy, D. P. Dogra, and B.-G. Kim, "A Comprehensive Review on Sentiment Analysis: Tasks, Approaches and Applications," pp. 1–29, Nov. 2023.
- [71] D. Ananda and R. R. Suryono, "Analisis Sentimen Publik Terhadap Pengungsi Rohingya di Indonesia dengan Metode Support Vector Machine dan Naïve Bayes," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 2, p. 748, Apr. 2024, doi: 10.30865/mib.v8i2.7517.
- [72] M. Zahirma, "Sistemasi: Jurnal Sistem Informasi Analisis Sentimen terhadap Review Penggunaan Produk Sunscreen dengan menggunakan Algoritma Naive Bayes Classifier Sentiment Analysis of Sunscreen Product Reviews using Naive Bayes Classifier Algorithm," 2025. [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [73] C. Yovita, K. Nasution, and T. Haramaini, "Penerapan Algoritma Naive Bayes dan Selection Sort pada Penilaian Kuis di Aplikasi Pembelajaran Pemrograman Java dan PHP," *Hello World Jurnal Ilmu Komputer*, vol. 2, no. 3, pp. 120–128, Aug. 2023, doi: 10.56211/helloworld.v2i3.278.
- [74] F. Syarifuddin and D. Kusumaningsih, "Analisis Perbandingan Algoritma Naive Bayes dan Support Vector Machine dengan Pendekatan TF-IDF Sebagai Klasifikasi Perintah Suara," *Technology and Science (BITS)*, vol. 7, no. 1, 2025, doi: 10.47065/bits.v7i1.7160.
- [75] P. D. P. Silitonga, D. E. R. Purba, and A. Rikki, "Perbandingan Peramalan Penerimaan Calon Mahasiswa Baru menggunakan Naïve Bayes dan Support Vector Machine," *KAKIFIKOM (Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer)*, vol. 06, no. 02, Oct. 2024.

- [76] T. Titiana and D. H. Bangkalang, "Analisis dan Penerapan Topic Modeling pada Judul Tugas Akhir Mahasiswa Menggunakan Metode Latent Dirichlet Allocation (LDA)," *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 4, pp. 1275–1287, Nov. 2023, doi: 10.29100/jipi.v8i4.4254.
- [77] D. L. C. Pardede and M. A. I. Waskita, "Analisis Pemodelan Topik untuk Ulasantentang Peduli Lindungi," *Jurnal Ilmiah Informatika Komputer*, vol. 28, no. 1, pp. 17–26, 2023, doi: 10.35760/ik.2023.v28i1.7925.
- [78] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [79] D. M. Blei, "Probabilistic topic models," in *Communications of the ACM*, Apr. 2012, pp. 77–84. doi: 10.1145/2133806.2133826.
- [80] S. García-Méndez, F. de Arriba-Pérez, A. Barros-Vila, F. J. González-Castaño, and E. Costa-Montenegro, "Automatic detection of relevant information, predictions and forecasts in financial news through topic modelling with Latent Dirichlet Allocation," *Applied Intelligence*, vol. 53, no. 16, pp. 19610–19628, Aug. 2023, doi: 10.1007/s10489-023-04452-4.
- [81] H. Satria, "Tweet Harvest (Twitter Crawler)." Accessed: May 11, 2025. [Online]. Available: <https://github.com/helmisatria/tweet-harvest>
- [82] G. Kusuma Wijaya, A. Venie Diniar, S. Alwan Jalaluddin, and I. Kharisudin, "PRISMA, Prosiding Seminar Nasional Matematika Dinamika Sentimen Publik dalam Sukses Pemerintahan Indonesia berdasarkan Analisis Data Media Sosial," *PRISMA*, vol. 8, pp. 407–413, 2025, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/>
- [83] fornigulo, "kamuskatabaku," Kaggle. Accessed: May 15, 2025. [Online]. Available: <https://www.kaggle.com/datasets/fornigulo/kamuskatabaku>
- [84] R. Prabowo, H. Sujaini, and T. Rismawan, "Analisis Sentimen Pengguna Twitter Terhadap Kasus COVID-19 di Indonesia Menggunakan Metode Regresi Logistik Multinomial," *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 11, no. 2, p. 366, Jul. 2023, doi: 10.26418/justin.v11i2.57449.
- [85] evanmartua34, "Twitter-COVID19-Indonesia-Sentiment-Analysis---Lexicon-Based." Accessed: May 27, 2025. [Online]. Available: https://github.com/evanmartua34/Twitter-COVID19-Indonesia-Sentiment-Analysis---Lexicon-Based/blob/master/lexicon/modified_full_lexicon.csv
- [86] M. Shenify, "Sentiment analysis of Saudi e-commerce using naïve bayes algorithm and support vector machine," *International Journal of Data and Network Science*, vol. 8, no. 3, pp. 1607–1612, Jun. 2024, doi: 10.5267/j.ijdns.2024.3.006.

- [87] A. P. Ratnasari and R. Nur'aini, "Performance of Random Oversampling, Random Undersampling, and SMOTE-NC Methods in Handling Imbalanced Class in Classification Models," *International Journal of Scientific Research and Management (IJSRM)*, vol. 12, no. 04, pp. 494–501, Apr. 2024, doi: 10.18535/ijssrm/v12i04.m03.
- [88] M. Widad and T. Asi, "Analisis Praktik Pinjaman Online dengan Sistem Buy Now Pay Later Ditinjau dari Hukum Bisnis Syari'ah," *Jurnal Ilmiah Nusantara (JINU)*, vol. 1, no. 5, pp. 232–239, Sep. 2024, doi: 10.61722/jinu.v1i5.2527.



LAMPIRAN

Lampiran 1. Kode Instalasi Library Tweet-Harvest

```
# Import required Python package
!pip install pandas

# Install Node.js (because tweet-harvest built using
Node.js)
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-
repo.gpg.key | sudo gpg --dearmor -o
/etc/apt/keyrings/nodesource.gpg

!NODE_MAJOR=20 && echo "deb [signed-
by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_${NODE_MAJOR}.x nodistro
main" | sudo tee /etc/apt/sources.list.d/nodesource.list

!sudo apt-get update
!sudo apt-get install nodejs -y

!node -v
```

Lampiran 2. Kode Pengumpulan Data

```
#Crawl data keyword Buy Now Pay Later

# Twitter Auth Token
twitter_auth_token = '*****'

filename = 'scraping_BuyNowPayLater.csv'
keyword = 'Buy Now Pay Later since:2025-01-01 until:2025-
05-10 lang:id'
limit = 5000

!npx --yes tweet-harvest@2.6.1 -o "{filename}" -s
"{keyword}" --tab "LATEST" -l {limit} --token
{twitter_auth_token}

#Crawl data keyword BNPL
filename = 'scraping_BNPL.csv'
keyword = 'BNPL since:2025-01-01 until:2025-05-10
lang:id'
limit = 5000
```

```

!npx --yes tweet-harvest@2.6.1 -o "{filename}" -s
"{keyword}" --tab "LATEST" -l {limit} --token
{twitter_auth_token}

#Crawl data keyword pay later
filename = 'scraping_pay_later.csv'
keyword = 'pay later since:2025-01-01 until:2025-05-10
lang:id'
limit = 5000

!npx --yes tweet-harvest@2.6.1 -o "{filename}" -s
"{keyword}" --tab "LATEST" -l {limit} --token
{twitter_auth_token}

#Crawl data keyword paylater
filename = 'scraping_paylater.csv'
keyword = 'paylater since:2025-01-01 until:2025-05-10
lang:id'
limit = 5000

!npx --yes tweet-harvest@2.6.1 -o "{filename}" -s
"{keyword}" --tab "LATEST" -l {limit} --token
{twitter_auth_token}

```

Lampiran 3. Kode Menggabungkan *Dataset*

```

import pandas as pd

all_dfs = []

for path in file_paths:
    df = pd.read_csv(path)
    all_dfs.append(df)
    print(f"Loaded file: {path}, shape: {df.shape}")

# gabungkan semua DataFrame jadi satu
combined_df = pd.concat(all_dfs, ignore_index=True)
combined_df.to_csv('/content/drive/MyDrive/Colab
Notebooks/scraping.csv', index=False)

display(combined_df)

```

Lampiran 4. Kode Eliminasi Bahasa

```

#filter bahasa 2
!pip install langdetect
from langdetect import detect
import pandas as pd

# Baca file hasil crawling

```

```

df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/scraping.csv')

# Fungsi untuk deteksi bahasa
def detect_language(text):
    try:
        return detect(str(text))
    except:
        return 'error'

df['detected_lang'] =
df['full_text'].apply(detect_language)

# Fungsi
df_final = df[df['detected_lang'] == 'id']

# Simpan
df_final.to_csv('/content/drive/MyDrive/Colab
Notebooks/scraping_1.csv', index=False)

print(f>Data awal: {len(df)} baris")
print(f>Data setelah filter bahasa Indonesia:
{len(df_final)} baris")
display(df_final)

```

Lampiran 5. Kode Eliminasi Lokasi

```

import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/scraping_1.csv')

# Daftar kata yang ingin dikecualikan
keywords = ['batu kawan', 'melaka', 'india', 'kilang
kopi', 'malaysia', 'africa', 'puncak bestari',
'selangor', 'pahang', 'johor', 'putrajaya', 'petaling',
'penang', 'kuala lumpur']

# Gabungkan kata kunci menjadi pola regex (pencocokan
tidak case-sensitive)
pattern = '|'.join(keywords)

# Filter: ambil data yang TIDAK mengandung kata kunci di
kolom 'location'
df_final =
df[~df['location'].str.lower().str.contains(pattern,
na=False)]

# Simpan
df_final.to_csv('/content/drive/MyDrive/Colab
Notebooks/scraping_2.csv', index=False)

```

```
# Tampilkan hasil
print(f"Data setelah location: {len(df_final)} baris")
display(df_final)
```

Lampiran 6. Kode Eliminasi *Tweet Gestun*

```
import pandas as pd

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/scraping_manual.csv')

# Hapus baris yang mengandung kata 'gestun' (case-
insensitive) di kolom full_text
df = df[~df['full_text'].str.contains(r'\bgestun\b',
case=False, na=False)]

# Simpan
df_final.to_csv('/content/drive/MyDrive/Colab
Notebooks/scraping_2.csv', index=False)

# Tampilkan jumlah data setelah penghapusan
print(f"Data setelah eliminasi location: {len(df)}")
display(df)
```

Lampiran 7. Kode Hapus Data Duplikat dan *Missing Value*

```
# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/scraping_final.csv')

#hapus missing value
df = df.dropna()

#hapus data duplikat
df = df.drop_duplicates(subset = "full_text")

df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')

print(f"Jumlah data setelah dibersihkan: {len(df)}
baris")
display(df)
```

Lampiran 8. Kode *Cleaning*

```
# cleaning data
import re
import string

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')

# hapus emoji
def remove_emoji(text):
    emoji_pattern = re.compile(
        "["u"\U0001F600-\U0001F64F" # emotikon wajah
        u"\U0001F300-\U0001F5FF" # simbol &
        pictographs
        u"\U0001F680-\U0001F6FF" # transport & simbol
        u"\U0001F700-\U0001F77F" # alchemical symbol
        u"\U0001F780-\U0001F7FF" # geometric shape
        extended
        u"\U0001F800-\U0001F8FF" # supplemental
        arrows-C
        u"\U0001F900-\U0001F9FF" # supplemental symbol
        & pictographs
        u"\U0001F1E0-\U0001F1FF" # bendera
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001FA00-\U0001FA6F"
        u"\U0001FA70-\U0001FAFF"
        u"\U0001F004-\U0001F0CF"
        "]" +, flags=re.UNICODE)
    return emoji_pattern.sub(r'', text)

# Cleaning text
def clean_text(text):
    text = re.sub(r'@w+', '', text) #
    hapus mention
    text = re.sub(r'#w+', '', text) #
    hapus hashtag
    text = re.sub(r'http\S+|www\S+|https\S+', '', text,
    flags=re.MULTILINE) # hapus URL
    text = re.sub(r'\d+', '', text) #
    hapus angka
    text = text.translate(str.maketrans('', '',
    string.punctuation)) # hapus tanda baca
    text = remove_emoji(text) #
    hapus emoji
    text = re.sub(r'\s+', ' ', text).strip() #
    hapus spasi ganda
    return text

# Fungsi
```

```

df['cleaning'] = df['full_text'].apply(clean_text)

# Simpan
df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv', index=False)

# Tampil
display(df[['full_text', 'cleaning']])

```

Lampiran 9. Kode *Case Folding*

```

# case folding

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')

# Fungsi
df['case_folding'] =
df['cleaning'].astype(str).str.lower()

# Simpan
df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv', index=False)

# Tampil
display(df[['full_text', 'cleaning', 'case_folding']])

```

Lampiran 10. Kode *Normalization*

```

# normalization

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')

# Kamus normalization
kamus_kata_baku =
pd.read_excel('/content/drive/MyDrive/Colab
Notebooks/kamuskatabaku.xlsx')
kamus_normalization =
dict(zip(kamus_kata_baku['tidak_baku'],
kamus_kata_baku['kata_baku']))

# Fungsi
def normalize_text(text):
    words = str(text).split()
    normalized_words = [kamus_normalization.get(word,
word) for word in words]
    return ' '.join(normalized_words)

```

```

df['normalization'] =
df['case_folding'].apply(normalize_text)

# Simpan
df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv', index=False)

# Tampil
display(df[['full_text', 'cleaning', 'case_folding',
'normalization']])

```

Lampiran 11. Kode *Tokenizing*

```

!pip install nltk
import nltk
nltk.download('punkt_tab')
from nltk.tokenize import word_tokenize

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')

# Fungsi
def word_tokenize_wrapper(text):
    return word_tokenize(text)

df['tokenizing'] =
df['normalization'].apply(word_tokenize_wrapper)

# Simpan
df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv', index=False)

# Tampil
display(df[['full_text', 'cleaning', 'case_folding',
'normalization', 'tokenizing']])

```

Lampiran 12. Kode *Stopword Removal*

```

# stopwords removal

from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory

# Inisialisasi stopwords remover Sastrawi
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')

```

```

# Pastikan kolom tokenizing berupa list
df['tokenizing'] = df['tokenizing'].apply(lambda x:
ast.literal_eval(x) if isinstance(x, str) else x)

# Fungsi
def remove_stopwords_sastrawi(tokens):
    # Gabungkan list token menjadi string
    text = ' '.join(tokens)
    # Hapus stopwords dengan sastrawi
    cleaned_text = stopwords.remove(text)
    # Split lagi jadi list token bersih
    return cleaned_text.split()

df['stopword_removal'] =
df['tokenizing'].apply(remove_stopwords_sastrawi)

# Simpan
df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv', index=False)

# Tampil
display(df[['full_text', 'cleaning', 'case_folding',
'normalization', 'tokenizing', 'stopword_removal']])

```

Lampiran 13. Kode *Stemming*

```

# Stemming
!pip install Sastrawi
from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')
df['stopword_removal'] =
df['stopword_removal'].apply(ast.literal_eval)

# Fungsi
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(token):
    return [stemmer.stem(word) for word in token]

df['stemming'] = df['stopword_removal'].apply(stem_text)

# Simpan
df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv', index=False)

# Tampil

```

```
display(df[['full_text', 'cleaning', 'case_folding',
'normalization', 'tokenizing', 'stopword_removal',
'stemming']])
```

Lampiran 14. Kode Pelabelan Sentimen

```
#pelabelan sentimen dengan lexicon-based (INSET)

# Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv')

# Baca kamus lexicon
lexicon_df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/lexicon_evanmartua34.csv')

lexicon_dict = dict(zip(lexicon_df['word'],
lexicon_df['weight']))

# Pastikan kolom 'stemming' adalah list
df['stemming'] = df['stemming'].apply(lambda x:
ast.literal_eval(x) if isinstance(x, str) else x)

# Fungsi
def sentiment_determine(tokens):
    sentiment_score = sum(lexicon_dict.get(word, 0) for
word in tokens)

    if sentiment_score > 0:
        sentiment = "positif"
    elif sentiment_score < 0:
        sentiment = "negatif"
    else:
        sentiment = "netral"
    return sentiment_score, sentiment

results = df['stemming'].apply(lambda x:
pd.Series(sentiment_determine(x),
index=['sentiment_score', 'sentiment_label']))
df = pd.concat([df, results], axis=1)

# Simpan
df.to_csv('/content/drive/MyDrive/Colab
Notebooks/dataset.csv', index=False)

# Tampil
display(df[['full_text', 'cleaning', 'case_folding',
'normalization', 'tokenizing', 'stopword_removal',
'stemming', 'sentiment_score', 'sentiment_label']])
```

Lampiran 15. Kode Ekstraksi Fitur

```
from sklearn.feature_extraction.text import
TfidfVectorizer

df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/dataset_final.csv')
df['stemming'] = df['stemming'].apply(ast.literal_eval)
data_joined = df['stemming'].apply(lambda tokens: '
'.join(tokens))

vectorizer = TfidfVectorizer(
    max_features=10000,
    ngram_range=(1,2),          # bigram + trigram
    min_df=5,                   # token muncul ≥2
    dokumen
    max_df=0.85,                # token muncul ≤90 %
    dokumen
)

tfidf_matrix = vectorizer.fit_transform(data_joined)

tfidf_df = pd.DataFrame(tfidf_matrix.toarray(),
    columns=vectorizer.get_feature_names_out())
tfidf_df['label'] = df['sentiment_label']

# Simpan
tfidf_df.to_csv('/content/drive/MyDrive/Colab
Notebooks/TF-IDF.csv', index=False)

# Tampil
display(tfidf_df)
```

Lampiran 16. Kode Grafik Top 20 Kata dengan Nilai TF-IDF Tertinggi

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd

# Fungsi
tfidf_scores = tfidf_matrix.toarray().sum(axis=0)

# DataFrame untuk visualisasi tabel
df_visual = pd.DataFrame({
    'Kata': vectorizer.get_feature_names_out(),
    'Nilai TF-IDF': tfidf_scores
})

# Urutkan dari TF-IDF tertinggi
df_visual = df_visual.sort_values(by='Nilai TF-IDF',
    ascending=False).reset_index(drop=True)
```

```

# Top 20 kata
top_n = 20
df_top = df_visual.head(top_n)

# Plot Barchart
plt.figure(figsize=(12, 6))
colors = sns.color_palette("Blues", n_colors=top_n)[::-1]
sns.barpplot(
    x='Nilai TF-IDF',
    y='Kata',
    data=df_top,
    hue='Kata',          # tambahkan hue dengan
    # variabel yang sama seperti 'y'
    palette=colors,
    legend=False        # agar tidak muncul legend
    # tambahan
)
plt.title(f'Top {top_n} Kata dengan Nilai TF-IDF
Tertinggi', fontsize=14)
plt.xlabel('Nilai TF-IDF')
plt.ylabel('Kata')
plt.tight_layout()
plt.show()

```

Lampiran 17. Kode *Splitting Data*

```

# Rasio 90:10
from sklearn.model_selection import train_test_split

X = tfidf_df.drop(columns=['label'])
y = tfidf_df['label']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.1, random_state=42, stratify=y
)

# Rasio 80:20
from sklearn.model_selection import train_test_split

X = tfidf_df.drop(columns=['label'])
y = tfidf_df['label']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Rasio 70:30
from sklearn.model_selection import train_test_split

X = tfidf_df.drop(columns=['label'])
y = tfidf_df['label']

```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)
```

Lampiran 18. Kode untuk *Import Metode*

```
from imblearn.over_sampling import SMOTE,
RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler

resampling_methods = {
    'Baseline': None,
    'SMOTE': SMOTE(random_state=42),
    'Oversampling': RandomOverSampler(random_state=42),
    'Undersampling': RandomUnderSampler(random_state=42)
}
```

Lampiran 19. Kode Pelatihan dan Pengujian Model

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report,
accuracy_score

for method_name, resampler in resampling_methods.items():
    if resampler is not None:
        X_resampled, y_resampled =
resampler.fit_resample(X_train, y_train)
    else:
        X_resampled, y_resampled = X_train, y_train

    model = MultinomialNB()
    model.fit(X_resampled, y_resampled)
    y_pred = model.predict(X_test)

    acc = accuracy_score(y_test, y_pred) * 100

    print(f"Akurasi: {acc:.4f}%")
    print(f"==== {method_name} =====")
    print(classification_report(y_test, y_pred,
zero_division=0))
```

Lampiran 20. Kode Evaluasi Model

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

cm = confusion_matrix(y_test, y_pred, labels=['negatif',
'netral', 'positif'])

plt.figure(figsize=(8, 6))
```

```

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Negatif', 'Netral', 'Positif'],
            yticklabels=['Negatif', 'Netral', 'Positif'])

plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix - Oversampling (90:10)')
plt.show()

```

Lampiran 21. Kode Visualisasi Barchart Distribusi Prediksi Sentimen

```

# Hitung distribusi prediksi
pred_counts =
pd.Series(y_pred).value_counts().reindex(['positif',
'netral', 'negatif'], fill_value=0).reset_index()
pred_counts.columns = ['Sentimen', 'Jumlah']

plt.figure(figsize=(8,6))
bars = sns.barplot(x='Sentimen', y='Jumlah',
data=pred_counts, hue='Sentimen', palette='viridis',
legend=False)

# Tambahkan label jumlah di atas masing-masing bar
for bar in bars.patches:
    plt.text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 2,
format(int(bar.get_height()))), ha='center',
fontsize=10)

plt.title('Distribusi Hasil Prediksi Sentimen')
plt.xlabel('Sentimen')
plt.ylabel('Jumlah Prediksi')
plt.show()

```

Lampiran 22. Kode Visualisasi WordCloud

```

from wordcloud import WordCloud
import matplotlib.pyplot as plt
import pandas as pd

# Gabungkan list token menjadi string panjang untuk
masing-masing sentimen
positif_text = " ".join(df[df['sentiment_label'] ==
'positif']['stemming'].sum())
netral_text = " ".join(df[df['sentiment_label'] ==
'netral']['stemming'].sum())
negatif_text = " ".join(df[df['sentiment_label'] ==
'negatif']['stemming'].sum())

# Fungsi untuk menampilkan WordCloud
def plot_wordcloud(text, title, color='black'):
    wordcloud = WordCloud(

```

```

        width=800,
        height=400,
        background_color='black',
        colormap=color,
        max_words=100
    ).generate(text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title(title, fontsize=18)
plt.show()

# Tampilkan WordCloud per sentimen
plot_wordcloud(positif_text, "WordCloud Sentimen Positif", color='Greens')
plot_wordcloud(netral_text, "WordCloud Sentimen Netral", color='Blues')
plot_wordcloud(negatif_text, "WordCloud Sentimen Negatif", color='Reds')

```

Lampiran 23. Kode Instalasi *Library* Pemodelan Topik

```
!pip install numpy==1.24.4 scipy==1.11.4 gensim==4.3.3
```

Lampiran 24. Kode Hapus *Keyword* Tertentu untuk Pemodelan Topik

```

#menghapus kata "paylater", "pay", "later", "bnpl"

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset_final.csv')
df['stemming'] = df['stemming'].apply(ast.literal_eval)
kata_dihapus = ['paylater', 'pay', 'later', 'bnpl']

# Fungsi
def hapus_kata(tokens):
    return [token for token in tokens if token not in kata_dihapus]

df['hapus_keyword'] = df['stemming'].apply(hapus_kata)
df.to_csv('/content/drive/MyDrive/Colab Notebooks/dataset_final.csv', index=False)
display(df[['stemming', 'hapus_keyword']])

```

Lampiran 25. Kode Pemisahan Data Berdasarkan Label Sentimen

```

#pemisahan data teks berdasarkan label sentimen
from ast import literal_eval

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset_final.csv')

```

```

df['tokens'] = df['hapus_keyword'].apply(literal_eval)

# Memisahkan berdasarkan label sentimen
positif_tokens = df[df['sentiment_label'] ==
'positif']['tokens'].tolist()
netral_tokens = df[df['sentiment_label'] ==
'netral']['tokens'].tolist()
negatif_tokens = df[df['sentiment_label'] ==
'negatif']['tokens'].tolist()

# Lihat jumlah data per label
print(len(positif_tokens), len(netral_tokens),
len(negatif_tokens))

```

Lampiran 26. Kode Penyusunan *Corpus* dan *Dictionary*

```

#penyusunan corpus dan dictionary
from gensim import corpora

# Dictionary untuk masing-masing sentimen
positif_dict = corpora.Dictionary(positif_tokens)
netral_dict = corpora.Dictionary(netral_tokens)
negatif_dict = corpora.Dictionary(negatif_tokens)

# Corpus untuk masing-masing sentimen
positif_corpus = [positif_dict.doc2bow(text) for text in
positif_tokens]
netral_corpus = [netral_dict.doc2bow(text) for text in
netral_tokens]
negatif_corpus = [negatif_dict.doc2bow(text) for text in
negatif_tokens]

```

Lampiran 27. Kode Penentuan Jumlah Topik Optimal

```

#penentuan jumlah topik optimal sentimen positif
from gensim.models import LdaModel, CoherenceModel
import matplotlib.pyplot as plt

# Fungsi
def compute_coherence_values(dictionary, corpus, texts,
limit, start=2, step=1):
    coherence_values = []
    for num_topics in range(start, limit, step):
        model = LdaModel(corpus=corpus,
id2word=dictionary, num_topics=num_topics,
random_state=42)
        coherence_model = CoherenceModel(model=model,
texts=texts, dictionary=dictionary, coherence='c_v')

    coherence_values.append(coherence_model.get_coherence())
    return coherence_values

```

```

coherence_vals = compute_coherence_values(positif_dict,
positif_corpus, positif_tokens, limit=11)

# Visualisasi
plt.plot(range(2, 11), coherence_vals)
plt.xlabel("Num Topics")
plt.ylabel("Coherence Score")
plt.title("Optimal Number of Topics (Positif)")
plt.xticks(range(2, 11))
plt.show()

max_coherence = max(coherence_vals)
print(f"Koherensi tertinggi: {max_coherence}")

#penentuan jumlah topik optimal sentimen netral
# Fungsi
def compute_coherence_values(dictionary, corpus, texts,
limit, start=2, step=1):
    coherence_values = []
    for num_topics in range(start, limit, step):
        model = LdaModel(corpus=corpus,
id2word=dictionary, num_topics=num_topics,
random_state=42)
        coherence_model = CoherenceModel(model=model,
texts=texts, dictionary=dictionary, coherence='c_v')
    coherence_values.append(coherence_model.get_coherence())
    return coherence_values

coherence_vals = compute_coherence_values(netral_dict,
netral_corpus, netral_tokens, limit=11)

# Visualisasi
plt.plot(range(2, 11), coherence_vals)
plt.xlabel("Num Topics")
plt.ylabel("Coherence Score")
plt.title("Optimal Number of Topics (Netral)")
plt.xticks(range(2, 11))
plt.show()

max_coherence = max(coherence_vals)
print(f"Koherensi tertinggi: {max_coherence}")

#penentuan jumlah topik optimal sentimen negatif
# Fungsi
def compute_coherence_values(dictionary, corpus, texts,
limit, start=2, step=1):
    coherence_values = []
    for num_topics in range(start, limit, step):
        model = LdaModel(corpus=corpus,
id2word=dictionary, num_topics=num_topics,
random_state=42)

```

```

        coherence_model = CoherenceModel(model=model,
        texts=texts, dictionary=dictionary, coherence='c_v')

        coherence_values.append(coherence_model.get_coherence())
        return coherence_values

    coherence_vals = compute_coherence_values(negatif_dict,
    negatif_corpus, negatif_tokens, limit=11)

    # Visualisasi
    plt.plot(range(2, 11), coherence_vals)
    plt.xlabel("Num Topics")
    plt.ylabel("Coherence Score")
    plt.title("Optimal Number of Topics (Negatif)")
    plt.xticks(range(2, 11))
    plt.show()

    max_coherence = max(coherence_vals)
    optimal_topics = range(2,
    11)[coherence_vals.index(max_coherence)]
    print(f"Koherensi tertinggi: {max_coherence}")
    print(f"Jumlah topik optimal: {optimal_topics}")

```

Lampiran 28. Kode Pelatihan Model LDA

```

#pelatihan model LDA
lda_model_positif = LdaModel(corpus=positif_corpus,
id2word=positif_dict, num_topics=8, random_state=42)
lda_model_netral = LdaModel(corpus=netral_corpus,
id2word=netral_dict, num_topics=8, random_state=42)
lda_model_negatif = LdaModel(corpus=negatif_corpus,
id2word=negatif_dict, num_topics=8, random_state=42)

# Keyword sentimen positif
print("Topik positif:")
for idx, topic in lda_model_positif.print_topics(-1):
    print(f"Topik {idx+1}: {topic}")

# Keyword sentimen positif netral
print("Topik netral:")
for idx, topic in lda_model_netral.print_topics(-1):
    print(f"Topik {idx+1}: {topic}")

# Keyword sentimen positif
print("Topik negatif:")
for idx, topic in lda_model_negatif.print_topics(-1):
    print(f"Topik {idx+1}: {topic}")

```

Lampiran 29. Tabel Revisi

No	Komentar Penguji *)	Tanggapan **)
1.	Tambahkan alasan menyasar <i>platform X</i> dibanding <i>platform</i> lain	Alasan menyasar <i>platform X</i> dan pertimbangannya dengan <i>platform</i> lain telah ditambahkan pada halaman 3.
2.	Pertanyaan penelitian 1 disesuaikan dengan tujuan penelitian 1	Pertanyaan penelitian 1 telah disesuaikan dengan tujuan penelitian 1 pada halaman 5.
3.	Disesuaikan manfaat penelitiannya untuk regulator/pemangku kepentingan	Manfaat penelitian 2 telah disesuaikan untuk regulator keuangan (OJK) pada halaman 5.
4.	Tabel 2.1. Tabel Perbandingan Penelitian, dibuat <i>repeat header</i>	Tabel 2.1. telah dibuat <i>repeat header</i> pada halaman 11-12.
5.	Kenapa menggunakan Metode Naïve Bayes Classifier Dan Latent Dirichlet Allocation? Apakah 1 metode tidak cukup?	Alasan menggunakan dua metode telah ditambahkan pada halaman 26-27.
6.	Tambahkan perbandingan LDA dengan metode lain	Perbandingan metode LDA dengan metode lainnya telah ditambahkan pada halaman 30.
7.	Alasan menggunakan 2 tahap, klasifikasi dan <i>topic modelling</i> , mengapa tidak langsung menggunakan <i>Aspect-Based Sentiment Analysis (ABSA)</i>	Alasan menggunakan pendekatan klasifikasi sentimen dan <i>topic modelling</i> dibandingkan ABSA telah ditambahkan pada halaman 27.
8.	Tambahkan implikasi manajerial pada bagian akhir	Implikasi manajerial telah ditambahkan pada halaman 89.
9.	Pada bagian saran untuk penelitian selanjutnya tambahkan: untuk menggunakan metode <i>Aspect-Based Sentiment Analysis (ABSA)</i> , penyesuaian pengaplikasian <i>preprocessing</i> agar hasil sentimen lebih tergambar, dan memberikan n-gram minimal 2 agar wordcloud dapat membaca/menampilkan sentimen asli	Tambahan pada bagian saran untuk penelitian selanjutnya telah ditambahkan pada halaman 91-92.
10.	Kesimpulan dibuat lebih ringkas	Kesimpulan telah dibuat lebih ringkas pada halaman 91.
11.	Kesimpulan belum menjawab pertanyaan penelitian karena menggunakan <i>keyword</i> yang berbeda	Kesimpulan telah diubah pada halaman 91—menggunakan <i>keyword</i> yang sama dengan pertanyaan penelitian.
12.	Perbaiki <i>typo</i>	Perbaiki <i>typo</i> telah dilakukan pada keseluruhan dokumen.