Daftar Pustaka

[1] Shahid, Dr Muhammad & Ibrahim, Suhaimi. (2013). A New Model For Requirements to Code Traceability to Support Code Coverage Analysis. Asian Academic Research Journal of Multidisciplinary (AARJMD). 1. 159-172.

[2] J. H. Husen and R. R. Riskiana, "Alat Pengukur Keatomikan Kebutuhan Perangkat Lunak Berbasis Kemajemukan Kalimat," Techno.Com, vol. 18, no. 3, pp. 203–213, Aug. 2019, doi: 10.33633/tc.v18i3.2383.

[3] Cleland-Huang, Jane, Carl K. Chang,and Mark J. Christensen. "Event-Based Traceability for ManagingEvolutionary Change." IEEE Trans-actions on Software Engineering 29.9(2003): 796-810.

[4] Kannenberg, A & Saiedian, Hossein. (2009). Why Software Requirements Traceability Remains a Challenge. The Journal of Defense Software Engineering. 22. 14-19.

[5] Yadla, S., Hayes, J.H., and Dekhtyar, A., "Tracing Require-ments to Defect Reports: An Application of Information Re-trieval Techniques", *Innovations Systems Software Engineer-ing: A NASA Journal*, vol.1, no.2, pp.116-121, September2005.

[6] Amrizal, Victor. (2018). Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk mengetahui Syarah Hadits berbasis WEB (STUDI KASUS: HADITS SHAHIH BUKHARI-MUSLIM). Jurnal Teknik Informatika. 11. 149-164. 10.15408/jti.v11i2.8623.

[7] Moreira, Jose Roberto. "Redirect Notice." *Research Gate*, Jose Roberto Moreira , 2000

[8] Jane Cleland-Huang, Carl K. Chang, Mark J. Christensen: Event-Based Traceability for Managing Evolutionary Change. IEEE Trans. Software Eng. 29(9): 796-810 (2003)

[9] K. Wiegers and J. Beatty, Software Requirements Third Edition, Washington: Microsoft Press, 2013.

[10] IEEE Standards Collection : Software Engineering, IEEE Standard 610.12-1990, IEEE, 1993.

[11] Dorfman, M. 8c Thayer, R.H. (1990). Standards, Guidelines, and Examples on System and Software Requirements Engineering, IEEE Computer Society Press Tutorial

[12] EE. (1991). Tools and Techniques for Maintaining Traceability During Design, IEE Colloquium, Computing and Control Division, Profegpiional Group C1, Digest No.: 1991/180.

[13] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," *Proceedings of the first instructional conference on machine learning,* vol. 242, 2003.

[14] P. G. Sapna and H. Mohanty, Clustering test cases to achieve effective test selection, in Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India(ACM, New York, USA, 2010).

[15] Note Narciso, Everton & Delamaro, Márcio & Nunes, Fátima. (2014). Test Case Selection: A Systematic Literature Review. International Journal of Software Engineering and Knowledge Engineering. 24. 653-676. 10.1142/S0218194014500259.

[16] Standard for Software Test Documentation. ANSI/IEEE STD 829- 1983

[17] Cem Kaner, J.D., "What Is a Good Test Case?", Florida Institute of Technology Department of Computer Sciences, May-2003.

[18] A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, and S. Panichella, "Using IR methods for labeling source code artifacts: Is it worthwhile?," in IEEE International Conference on Program Comprehension, 2012, pp. 193–202, doi: 10.1109/icpc.2012.6240488.

[19] Z. Zhang, Y. Lei, J. Xu, X. Mao, and X. Chang, "TFIDF-FL: Localizing faults using term frequency-9 inverse document frequency and deep learning," IEICE Trans. Inf. Syst., vol. E102D, no. 9, pp. 1860– 1864, Sep. 2019, doi: 10.1587/transinf.2018EDL8237.

[20] G. Scanniello and A. Marcus, "Clustering support for static concept location in source code," in IEEE International Conference on Program Comprehension, 2011, pp. 1–10, doi: 10.1109/ICPC.2011.13.

[21] Rahutomo, Faisal & Ririd, Ariadi. (2019). Evaluasi Daftar Stopword Bahasa Indonesia. Jurnal Teknologi Informasi dan Ilmu Komputer. 6. 41. 10.25126/jtiik.2019611226.

[22] Jabbar, Abdul & Iqbal, Sajid & Tamimy, Manzoor & Hussain, Shafiq & Akhunzada, Adnan. (2020). Empirical evaluation and study of text stemming algorithms. Artificial Intelligence Review. 53. 10.1007/s10462-020-09828-3.

[23] Pooja. (2018). Natural Language Processing using Python.

[24] H. Christian, M. P. Agus, and D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)," ComTech Comput. Math. Eng. Appl., vol. 7, no. 4, p. 285, Dec. 2016, doi: 10.21512/comtech.v7i4.3746.

[25] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," ACM Trans. Inf. Syst., vol. 26, no. 3, pp. 1–37, Jun. 2008, doi: 10.1145/1361684.1361686.

[26] S. Albitar, S. Fournier, and B. Espinasse, "An effective TF/IDF-based text-to-text semantic similarity measure for text classification," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8786, pp. 105–114, Oct. 2014, doi: 10.1007/978-3-319-11749-2_8.

[27] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” Int. J. Comput. Appl., vol. 181, no. 1, pp. 25–29, 2018, doi: 10.5120/ijca2018917395.
[28] R. Kohavi and F. Provost, “Glossary of Terms: Special Issue on Applictions of Macine Learning and the Knowledge Discovery Process,” Mach. Learn., vol. 30, pp. 271–274, 1998.