

**PERBANDINGAN KINERJA *LARGE LANGUAGE MODEL* (LLM)
BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN
PENULISAN KODE JAVASCRIPT**

SKRIPSI

*diajukan untuk memenuhi sebagian syarat untuk memperoleh gelar Sarjana
Komputer pada Program Studi Rekayasa Perangkat Lunak*



Oleh
Iman Nurohman
2005086

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK
UNIVERSITAS PENDIDIKAN INDONESIA
KAMPUS CIBIRU
2024**

**PERBANDINGAN KINERJA *LARGE LANGUAGE MODEL* (LLM)
BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN
PENULISAN KODE JAVASCRIPT**

Oleh
Iman Nurohman

Sebuah skripsi yang diajukan untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Pendidikan pada Fakultas Pendidikan Bahasa dan Seni

© Iman Nurohman 2024
Universitas Pendidikan Indonesia
Juli 2024

Hak Cipta dilindungi undang-undang.
Skripsi ini tidak boleh diperbanyak seluruhnya atau sebagian,
dengan dicetak ulang, difoto kopi, atau cara lainnya tanpa ijin dari penulis.

HALAMAN PENGESAHAN

Iman Nurohman

**PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS
LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE
JAVASCRIPT**

Disetujui dan disahkan oleh pembimbing:

Pembimbing I



Dian Angeraini, S.ST., M.T.

NIP 920190219930526201

Pembimbing II

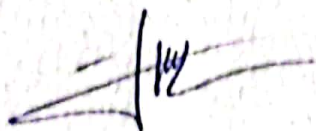


Raditya Muhammad, S.T., M.T.

NIP 920190219920507101

Mengetahui,

Ketua Program Studi Rekayasa Perangkat Lunak



Mochamad Iqbal Ardimansyah, S.T., M.Kom.

NIP 920190219910328101

**PERNYATAAN
KEASLIAN SKRIPSI DAN BEBAS PLAGIARISME**

Dengan ini saya menyatakan bahwa skripsi dengan judul "*Perbandingan Kinerja Large Language Model (LLM) Berbasis Llama2 Dalam Memperbaiki Kesalahan Penulisan Kode Javascript*" ini beserta seluruh isinya adalah benar-benar karya saya sendiri. Saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika ilmu yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung risikol sanksi apabila di kemudian har ditemukan adanya pelanggaran etika keilmuan atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, Juli 2024



Iman Nurohman

NIM 2005086

HALAMAN UCAPAN TERIMA KASIH

Puji syukur atas kehadiran Allah *Subhanahu wata'la* yang telah memberikan nikmat dan rizki melimpah yang begitu besar, sehingga penulis dapat menyelesaikan penelitian ini yang berjudul “PERBANDINGAN KINERJA *LARGE LANGUAGE MODEL* (LLM) BERBASIS *LLAMA2* DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT” untuk memperoleh gelar Sarjana Komputer pada program studi S1 Rekayasa Perangkat Lunak di Universitas Pendidikan Indonesia.

Penulis Menyadari bahwa keberhasilannya penelitian dan penyusunan skripsi ini berkat bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada :

- 1 Ayah, Ibu, Kakak dan keluarga besar penulis atas segala doa dan dukungan yang telah diberikan selama masa perkuliahan baik secara materil maupun moril sehingga dapat menyelesaikannya dengan baik.
- 2 Bapak Mochammad Iqbal Ardimansyah, S.T., M.Kom., selaku dosen Ketua Program Studi Rekayasa Perangkat Lunak
- 3 Ibu Dian Anggraini, S.ST., M.T., selaku dosen pembimbing pertama dalam penelitian penyusunan skripsi ini.
- 4 Bapak Raditya Muhammad M.T., selaku dosen pembimbing kedua dalam penelitian ini dan penyusunan skripsi ini.
- 5 Seluruh dosen dan civitas akademika Universitas Pendidikan Indonesia Kampus UPI di Cibiru atas Segala kebaikan dan jasa selama masa perkuliahan penulis.
- 6 Seluruh rekan-rekan Angkatan 2020 yang telah kebersamai dalam proses perkuliahan bersama dengan penulis.

Bandung, 13 Juni 2024

Penulis

ABSTRAK

ESLint merupakan alat analisa statis yang digunakan untuk mendeteksi kesalahan penulisan logika dari sintaks kode JavaScript. Akan tetapi, tidak semua kesalahan sintaks kode dapat diperbaiki oleh *ESLint*. Untuk mengatasi hal tersebut pendekatan dengan *Large Language Model* adalah salah satu cara yang dapat diterapkan. *Large Language Model* (LLM) merupakan bagian dari pengembangan *Deep Learning* pada penugasan pemrosesan bahasa alami yang telah menarik perhatian belakangan ini. Kemampuannya telah terbukti dapat memahami dan membantu untuk menyelesaikan permasalahan saat menulis sebuah kode program. Penelitian ini mengusulkan penggunaan LLM yang berbasis *Llama2* dan melatih model tersebut pada tugas khusus untuk melakukan perbaikan pada kesalahan penulisan sintaks kode khususnya pada kode JavaScript. Pendekatan ini memanfaatkan kemampuan *Llama2* dalam memahami sintaksis dan semantik kode program serta menghasilkan perbaikan yang akurat dan efektif. Hasil pengujian menunjukkan bahwa model-model berbasis *Llama2* mampu mencapai akurasi perbaikan kesalahan khususnya kode JavaScript dengan rata-rata skor akurasi 93%. Kualitas kode yang dihasilkan model mendapatkan hasil yang baik dengan rata-rata skor 57% pada metrik CodeBLEU, dan dapat melampaui hasil model sebelumnya. Temuan ini menunjukkan potensi besar dalam penggunaan LLM khususnya model berbasis *Llama2* untuk melakukan tugas *Automated Program Repair* (APR) dan dapat membuka jalan untuk pengembangan sistem perbaikan kode program yang lebih canggih dan efisien.

Kata Kunci : *ESLint*, JavaScript, *Automated Program Repair*, *Large Language Model* dan *Llama2*

ABSTRACT

ESLint is a static analysis tool used to detect logic errors and syntax issues in JavaScript code. However, not all syntax errors can be fixed by ESLint. To address this, one approach is to leverage Large Language Models (LLMs). Large Language Models are a part of the Deep Learning development in natural language processing tasks that has gained recent attention. Their ability has proven effective in understanding and aiding in resolving issues while writing code. This research proposes the use of an LLM based on Llama2, trained specifically to correct syntax errors, particularly in JavaScript code. This approach harnesses Llama2's capabilities in understanding both the syntax and semantics of programming code, producing accurate and effective corrections. Testing results indicate that Llama2-based models can achieve an average accuracy score of 93% in correcting JavaScript syntax errors. The quality of code produced by the model also shows promising results, with an average CodeBLEU score of 57%, surpassing previous model outcomes. These findings highlight significant potential in using LLMs, especially Llama2-based models, for Automated Program Repair (APR) tasks, paving the way for the development of more advanced and efficient code repair systems.

Kata Kunci : *ESLint, JavaScript, Automated Program Repair, Large Language Model dan Llama2*

DAFTAR ISI

HALAMAN UCAPAN TERIMA KASIH	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang Penelitian	1
1.2 Rumusan Masalah Penelitian	4
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah.....	6
1.6 Struktur Organisasi Skripsi	6
BAB II KAJIAN PUSTAKA	8
2.1 Penelitian terkait.....	8
2.2 Machine Learning	14
2.3 <i>Deep Learning</i>	15
2.4 <i>Large Language Models (LLMs)</i>	16
2.4.1 <i>Llama2</i>	17
2.4.2 <i>Code Llama</i>	20
2.5 Optimasi Model.....	22
2.5.1 <i>Parameter-Efficient Fine-Tuning (PEFT)</i>	22
2.5.2 <i>QLoRA</i>	24
2.6 JavaScript	26
2.7 <i>Static Analysis Tools</i>	26
2.7.1 <i>ESLint</i>	27
2.5 Skor Akurasi.....	28

2.6	<i>BLEU</i>	28
2.6.1	<i>CodeBLEU</i>	30
2.7	Model Referensi	31
BAB III METODE PENELITIAN.....		33
3.1	Desain Penelitian.....	33
3.1.1	Klarifikasi Penelitian.....	34
3.1.2	Studi Deskriptif I.....	34
3.1.3	Studi Perspektif	34
3.1.3.1	Model Dampak.....	34
3.1.3.2	Pengembangan Model Perbaikan Kesalahan Kode Program....	35
3.1.3.3	Pengumpulan Data	36
3.1.3.4	Pra-Pemrosesan Data	37
3.1.3.5	Format dan Tokenisasi Perintah (<i>Prompt</i>).....	38
3.1.3.6	Perbaikan Kode dengan <i>Large Language Model</i> (LLM).....	38
3.1.4	Studi Deskriptif II	39
3.1.4.1	Evaluasi Akurasi Model.....	39
3.1.4.2	Evaluasi Akurasi Model Berdasarkan Tipe Kesalahan.....	40
3.1.4.3	Evaluasi BLEU dan CodeBLEU	40
3.2	Instrumen Penelitian.....	42
3.2.1	Aturan-aturan Pada <i>ESLint</i>	42
3.2.2	Perangkat Keras	43
3.2.3	<i>Library Python</i>	43
3.3	Analisis Data	44
BAB IV HASIL DAN PEMBAHASAN		45
4.1	Pengembangan Model Perbaikan Kode Program	45
4.1.1	Persiapan Dataset	45
4.1.2	Pra – Pemrosesan Data.....	48
4.1.1.1	Format Perintah (Prompting)	48
4.1.1.2	Tokenisasi Perintah	49
4.1.2	Pra-Pelatihan Model.....	51

4.1.3	Pelatihan <i>Large Language Model</i> (LLM) Berbasis <i>Llama2</i>	52
4.2	Evaluasi Kinerja <i>Large Language Model</i> (LLM) Berbasis <i>Llma2</i>	55
4.2.1	Akurasi Perbaikan Kesalahan Kode.....	55
4.2.2	Akurasi Perbaikan Berdasarkan Tipe Kesalahan.....	56
4.2.3	Uji Evaluasi dengan BLEU dan <i>CodeBLEU</i>	57
BAB V PENUTUP.....		59
5.1	Kesimpulan	59
5.2	Saran.....	60
DAFTAR PUSTAKA		61
LAMPIRAN.....		70

DAFTAR TABEL

Tabel 2.1 Pratinjau Penelitian Terkait.....	12
Tabel 3.1 Deskripsi Dataset yang Digunakan.....	37
Tabel 3.2 Interpretasi Skor BLEU dan CodeBLEU.....	41
Tabel 3.3 Daftar Aturan pada <i>ESlint</i>	42
Tabel 3.4 Spesifikasi Perangkat Keras.....	43
Tabel 3.5 Daftar <i>Library Python</i>	44
Tabel 4.1 Rincian Struktur Data.....	46
Tabel 4.2 Pembagian dataset.....	46
Tabel 4.3 Format Prompt Untuk Melakukan Perbaikan Kode.....	48
Tabel 4.4 Contoh Perubahan Tokenisasi pada <i>Prompt</i>	50
Tabel 4.5 Konfigurasi Pelatihan Model.....	51
Tabel 4.6 Hasil Pengujian Akurasi Model.....	56
Tabel 4.7 Hasil Pengujian Berdasarkan Tipe Kesalahan.....	57
Tabel 4.8 Hasil Pengujian dengan <i>CodeBLEU</i>	57

DAFTAR GAMBAR

Gambar 2.1 Alur kerja Machine Learning Menurut Wei, dkk., (2019).....	15
Gambar 2.2 Alur Proses Fine-Tuning Llama-2-Chat (Touvron dkk., 2023)	18
Gambar 2.3 Hasil Perbandingan Evaluasi <i>Llama2</i> dengan Model Open-Source Lain (Touvron dkk., 2023).	20
Gambar 2.4 Alur Pembuatan <i>Code Llama</i> (Roziere dkk., 2023).....	21
Gambar 2.5 Penambahan <i>Adapter Layer</i> pada <i>Transformer Layer</i> (Houlsby dkk., 2019)	23
Gambar 2.6 Perbedaan metode fine tuning biasa dengan QLoRA (Dettmers dkk., 2024)	25
Gambar 2.7 Tahapan Perhitungan Metrik Evaluasi CodeBLEU (Ren dkk., 2020)	31
Gambar 2.8 Model Referensi	32
Gambar 3.1 Desain Penelitian.....	33
Gambar 3.2 Model Dampak.....	35
Gambar 3.3 Diagram Alur Pengembangan Model Prediksi Perbaikan Kesalahan Penulisan Sintaks Kode JavaScript	36
Gambar 4.1 Distribusi Pembagian Data berdasarkan 52 Tipe Error <i>ESLint</i>	47
Gambar 4.2 Perbandingan Hasil <i>Train Loss</i> model berbasis <i>Llama2</i>	53
Gambar 4.3 Perbandingan Hasil <i>Evaluation Loss</i> model berbasis <i>Llama2</i>	54

DAFTAR LAMPIRAN

Lampiran 1 Rincian Kolom pada Dataset	70
Lampiran 2 Rincian Aturan pada <i>ESlint</i>	72
Lampiran 3 Rincian Hasil Uji Akurasi Berdasarkan Tipe Kesalahan.....	82
Lampiran 4 Rincian Model Berbasis <i>Llama2</i> setelah menerapkan <i>Parameter Efficient Fine-Tuning (PEFT) dan QLoRA</i>	84
Lampiran 5 Hasil Perbaikan <i>Llama2</i> Chat	87
Lampiran 6 Hasil Perbaikan dengan <i>Code Llama</i>	91
Lampiran 7 Hasil Perbaikan dengan <i>Code Llama</i> Instruct.....	94
Lampiran 8 Dokumentasi model pada website <i>Hugging Face</i>	98

DAFTAR PUSTAKA

- Aghajanyan, A., Huang, B., Ross, C., Karpukhin, V., Xu, H., Goyal, N., ... Lewis, M. (2022). Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*.
- Bavarian, M., Jun, H., Tezak, N., Schulman, J., McLeavey, C., Tworek, J., dan Chen, M. (2022). Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*.
- Berabi, B., He, J., Raychev, V., dan Vechev, M. (2021). TFix: Learning to Fix Coding Errors with a Text-to-Text Transformer. Dalam M. Meila dan T. Zhang (Ed.), *Proceedings of the 38th International Conference on Machine Learning* (hlm. 780–791). PMLR. Diambil dari <https://proceedings.mlr.press/v139/berabi21a.html>
- Bi, Q., Goodman, K. E., Kaminsky, J., dan Lessler, J. (2019). What is Machine Learning? A Primer for the Epidemiologist. *American Journal of Epidemiology*, 188(12), 2222–2239. <https://doi.org/10.1093/aje/kwz189>
- Blessing, dan Chakrabarti. (2009). *DRM, a design research methodology*. Springer Dordrecht.
- Câmara, V., Mendonca-Neto, R., Silva, A., dan Cordovil, L. (2024). A Large Language Model approach to SQL-to-Text Generation. *2024 IEEE International Conference on Consumer Electronics (ICCE)*, 1–4. <https://doi.org/10.1109/ICCE59016.2024.10444148>
- Cerf, V. G. (2023). Large Language Models. *Commun. ACM*, 66(8), 7. <https://doi.org/10.1145/3606337>
- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... Wang, Y. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3), 1–45.

- Chatzikoumi, E. (2020). How to evaluate machine translation: A review of automated and human metrics. *Natural Language Engineering*, 26(2), 137–161.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., ... Brockman, G. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Chen, Z., Kommrusch, S., Tufano, M., Pouchet, L.-N., Poshyvanyk, D., dan Monperrus, M. (2021). SequenceR: Sequence-to-Sequence Learning for End-to-End Program Repair. *IEEE Transactions on Software Engineering*, 47(9), 1943–1959. <https://doi.org/10.1109/TSE.2019.2940179>
- Dettmers, T., Pagnoni, A., Holtzman, A., dan Zettlemoyer, L. (2024). Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.
- Dinella, E., Dai, H., Li, Z., Naik, M., Song, L., dan Wang, K. (2020). Hoppity: Learning graph transformations to detect and fix bugs in programs. *International Conference on Learning Representations (ICLR)*.
- Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., ... Sun, M. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3), 220–235. <https://doi.org/10.1038/s42256-023-00626-4>
- Do, L. N. Q., Wright, J. R., dan Ali, K. (2022). Why Do Software Developers Use Static Analysis Tools? A User-Centered Study of Developer Needs and Motivations. *IEEE Transactions on Software Engineering*, 48(3), 835–847. <https://doi.org/10.1109/TSE.2020.3004525>
- Doeraene, S. J. R. (2018). *Cross-Platform Language Design*. EPFL.
- Ehrlinger, L., dan Wöß, W. (2022). A survey of data quality measurement and monitoring tools. *Frontiers in big data*, 5, 850611.
- Farzat, F. de A., Barros, M. de O., dan Travassos, G. H. (2021). Evolving JavaScript Code to Reduce Load Time. *IEEE Transactions on Software Engineering*, 47(8), 1544–1558. <https://doi.org/10.1109/TSE.2019.2928293>

Iman Nurohman, 2024

PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

- Ferguson, R. (2019). Introduction to JavaScript. Dalam *Beginning JavaScript: The Ultimate Guide to Modern JavaScript Development* (hlm. 1–10). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-4395-4_1
- Fried, D., Aghajanyan, A., Lin, J., Wang, S. I., Wallace, E., Shi, F., ... Lewis, M. (2022a). InCoder: A Generative Model for Code Infilling and Synthesis. *ArXiv, abs/2204.05999*. Diambil dari <https://api.semanticscholar.org/CorpusID:248157108>
- Fried, D., Aghajanyan, A., Lin, J., Wang, S., Wallace, E., Shi, F., ... Lewis, M. (2022b). InCoder: A generative model for code infilling and synthesis. *arXiv preprint arXiv:2204.05999*.
- Fu, Z., Yang, H., So, A. M.-C., Lam, W., Bing, L., dan Collier, N. (2023). On the Effectiveness of Parameter-Efficient Fine-Tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11), 12799–12807. <https://doi.org/10.1609/aaai.v37i11.26505>
- Gillioz, A., Casas, J., Mugellini, E., dan Khaled, O. A. (2020). Overview of the Transformer-based Models for NLP Tasks. *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, 179–183. <https://doi.org/10.15439/2020F20>
- Hadi, M. U., tashi, qasem al, Qureshi, R., Shah, A., muneer, amgad, Irfan, M., ... Mirjalili, S. (2023). *Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects*. <https://doi.org/10.36227/techrxiv.23589741.v4>
- He, J., Beurer-Kellner, L., dan Vechev, M. (2022). On Distribution Shift in Learning-based Bug Detectors. Dalam K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, dan S. Sabato (Ed.), *Proceedings of the 39th International Conference on Machine Learning* (hlm. 8559–8580). PMLR. Diambil dari <https://proceedings.mlr.press/v162/he22a.html>
- He, J., dan Vechev, M. (2023). Large Language Models for Code: Security Hardening and Adversarial Testing. *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 1865–1879. New

Iman Nurohman, 2024

PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

- York, NY, USA: Association for Computing Machinery.
<https://doi.org/10.1145/3576915.3623175>
- Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP. Dalam K. Chaudhuri dan R. Salakhutdinov (Ed.), *Proceedings of the 36th International Conference on Machine Learning* (hlm. 2790–2799). PMLR. Diambil dari <https://proceedings.mlr.press/v97/houlsby19a.html>
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Hussain, F., Hussain, R., Hassan, S. A., dan Hossain, E. (2020). Machine learning in IoT security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, 22(3), 1686–1721.
- Lee, S., Lee, J., Moon, H., Park, C., Seo, J., Eo, S., ... Lim, H. (2023). A survey on evaluation metrics for machine translation. *Mathematics*, 11(4), 1006.
- Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., ... Vinyals, O. (2022). Competition-level code generation with AlphaCode. *Science*, 378(6624), 1092–1097. <https://doi.org/10.1126/science.abq1158>
- Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., ... Shmueli, Y. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637), 1123–1130.
- Liu, J., Sha, C., dan Peng, X. (2023). An Empirical Study of Parameter-Efficient Fine-Tuning Methods for Pre-Trained Code Models. *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 397–408. <https://doi.org/10.1109/ASE56229.2023.00125>
- Liu, Y., Li, T., Wang, Z., Zhu, G., Zhang, Y., dan Zou, Q. (2023). Exploring Parameter-Efficient Fine-Tuning of a Large-Scale Pre-Trained Model for scRNA-seq Cell Type Annotation. *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 580–585.
<https://doi.org/10.1109/BIBM58861.2023.10385599>

Iman Nurohman, 2024

PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

- Lu, J., Yu, L., Li, X., Yang, L., dan Zuo, C. (2023). LLaMA-Reviewer: Advancing Code Review Automation with Large Language Models through Parameter-Efficient Fine-Tuning. *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, 647–658.
<https://doi.org/10.1109/ISSRE59848.2023.00026>
- Lutellier, T., Pham, H. V., Pang, L., Li, Y., Wei, M., dan Tan, L. (2020). CoCoNuT: combining context-aware neural translation models using ensemble for program repair. *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 101–114. New York, NY, USA: Association for Computing Machinery.
<https://doi.org/10.1145/3395363.3397369>
- Lv Zheqi and Wang, F. and Z. S. and Z. W. and K. K. and W. F. (2024). Parameters Efficient Fine-Tuning for Long-Tailed Sequential Recommendation. Dalam J. and Z. G. and W. R. Fang Lu and Pei (Ed.), *Artificial Intelligence* (hlm. 442–459). Singapore: Springer Nature Singapore.
- Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9(1), 381–386.
- Mathew Amitha and Amudha, P. and S. S. (2021). Deep Learning Techniques: An Overview. Dalam R. and D. A. Hassanien Aboul Ella and Bhatnagar (Ed.), *Advanced Machine Learning Technologies and Applications* (hlm. 599–608). Singapore: Springer Singapore.
- Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
- Ozkaya, I. (2023). Application of Large Language Models to Software Engineering Tasks: Opportunities, Risks, and Implications. *IEEE Software*, 40(3), 4–8. <https://doi.org/10.1109/MS.2023.3248401>
- Paramshetti, P., dan Phalke, D. A. (2014). *Survey on Software Defect Prediction Using Machine Learning Techniques*.

Iman Nurohman, 2024

PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

- Patil Tejashri and Pandey, S. and V. K. (2021). A Review on Basic Deep Learning Technologies and Applications. Dalam V. and S. H. N. and P. R. Kotecha Ketan and Piuri (Ed.), *Data Science and Intelligent Applications* (hlm. 565–573). Singapore: Springer Singapore.
- Pradel, M., dan Sen, K. (2018). DeepBugs: A Learning Approach to Name-Based Bug Detection. *Proc. ACM Program. Lang.*, 2(OOPSLA).
<https://doi.org/10.1145/3276517>
- Pujar, S., Buratti, L., Guo, X., Dupuis, N., Lewis, B., Suneja, S., ... Puri, R. (2023). Invited: Automated Code generation for Information Technology Tasks in YAML through Large Language Models. *2023 60th ACM/IEEE Design Automation Conference (DAC)*, 1–4.
<https://doi.org/10.1109/DAC56929.2023.10247987>
- Purba, M., Ghosh, A., Radford, B. J., dan Chu, B. (2023). Software Vulnerability Detection using Large Language Models. *2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 112–119. Los Alamitos, CA, USA: IEEE Computer Society.
<https://doi.org/10.1109/ISSREW60843.2023.00058>
- Purnawansyah, Ali, Z., Darwis, H., Ilmawan, L. B., Jabir, S. R., dan Manga, A. R. (2024). Memory Efficient with Parameter Efficient Fine-Tuning for Code Generation Using Quantization. *2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 1–6.
<https://doi.org/10.1109/IMCOM60618.2024.10418267>
- Rafnsson Willard and Giustolisi, R. and K. M. and H. M. (2020). Fixing Vulnerabilities Automatically with Linters. Dalam J. and C. C. Kutyłowski Mirosław and Zhang (Ed.), *Network and System Security* (hlm. 224–244). Cham: Springer International Publishing.
- Ren, S., Guo, D., Lu, S., Zhou, L., Liu, S., Tang, D., ... Ma, S. (2020). Codebleu: a method for automatic evaluation of code synthesis. *arXiv preprint arXiv:2009.10297*.
- Rodríguez-Pérez, G., Robles, G., Serebrenik, A., Zaidman, A., Germán, D. M., dan Gonzalez-Barahona, J. M. (2020). How bugs are born: a model to
 Iman Nurohman, 2024
PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT
 Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

- identify how bugs are introduced in software components. *Empirical Software Engineering*, 25(2), 1294–1340. <https://doi.org/10.1007/s10664-019-09781-y>
- Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., ... Rapin, J. (2023). Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.
- Sarsa, S., Denny, P., Hellas, A., dan Leinonen, J. (2022). Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, 27–43. New York, NY, USA: Association for Computing Machinery.
<https://doi.org/10.1145/3501385.3543957>
- Sharma, T., Kechagia, M., Georgiou, S., Tiwari, R., Vats, I., Moazen, H., dan Sarro, F. (2024). A survey on machine learning techniques applied to source code. *Journal of Systems and Software*, 209, 111934.
<https://doi.org/https://doi.org/10.1016/j.jss.2023.111934>
- Shen, Z., dan Chen, S. (2020). A Survey of Automatic Software Vulnerability Detection, Program Repair, and Defect Prediction Techniques. *Security and Communication Networks*, 2020, 8858010.
<https://doi.org/10.1155/2020/8858010>
- Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., dan Liu, Y. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 127063.
- Tehrani, F. S., Calvello, M., Liu, Z., Zhang, L., dan Lacasse, S. (2022). Machine learning and landslide studies: recent advances and applications. *Natural Hazards*, 114(2), 1197–1245.
- Teubner, T., Flath, C. M., Weinhardt, C., van der Aalst, W., dan Hinz, O. (2023). Welcome to the Era of ChatGPT et al. *Business & Information Systems Engineering*, 65(2), 95–101. <https://doi.org/10.1007/s12599-023-00795-x>
- Tomar, R., dan Dangi, S. (2021). *JavaScript: Syntax and Practices*. Chapman and Hall/CRC.

Iman Nurohman, 2024

PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

- Tómasdóttir, K. F., Aniche, M., dan van Deursen, A. (2017). Why and how JavaScript developers use linters. *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 578–589.
<https://doi.org/10.1109/ASE.2017.8115668>
- Tómasdóttir, K. F., Aniche, M., dan Van Deursen, A. (2020). The Adoption of JavaScript Linters in Practice: A Case Study on ESLint. *IEEE Transactions on Software Engineering*, 46(8), 863–891.
<https://doi.org/10.1109/TSE.2018.2871058>
- Tonja, A. L., Woldeyohannis, M. M., dan Yigezu, M. G. (2021). A parallel corpora for bi-directional neural machine translation for low resourced ethiopian languages. *2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, 71–76. IEEE.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... Bhosale, S. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Uc-Castillo, J. L., Marín-Celestino, A. E., Martínez-Cruz, D. A., Tuxpan-Vargas, J., dan Ramos-Leal, J. A. (2023). A systematic review and meta-analysis of groundwater level forecasting with machine learning techniques: Current status and future directions. *Environmental Modelling & Software*, 105788.
- Vaithilingam, P., Zhang, T., dan Glassman, E. L. (2022). Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery.
<https://doi.org/10.1145/3491101.3519665>
- Vasic, M., Kanade, A., Maniatis, P., Bieber, D., dan Singh, R. (2019). Neural Program Repair by Jointly Learning to Localize and Repair. *CoRR*, *abs/1904.01720*. Diambil dari <http://arxiv.org/abs/1904.01720>
- Vujović, Ž. (2021). Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6), 599–606.

Iman Nurohman, 2024

PERBANDINGAN KINERJA LARGE LANGUAGE MODEL (LLM) BERBASIS LLAMA2 DALAM MEMPERBAIKI KESALAHAN PENULISAN KODE JAVASCRIPT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

- Wang, Y., Ji, S., dan Feng, J. (2020). *Selection of Programming Language in the Development of Computer Application Software*. Diambil dari <https://api.semanticscholar.org/CorpusID:230650891>
- Wei, J., Chu, X., Sun, X., Xu, K., Deng, H., Chen, J., ... Lei, M. (2019). Machine learning in materials science. *InfoMat*, 1(3), 338–358.
- Xin, J., Tang, R., Yu, Y., dan Lin, J. (2021). BERxiT: Early Exiting for BERT with Better Fine-Tuning and Extension to Regression. Dalam P. Merlo, J. Tiedemann, dan R. Tsarfaty (Ed.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (hlm. 91–104). Online: Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.eacl-main.8>
- Zhang, C., Chen, J., Li, J., Peng, Y., dan Mao, Z. (2023). Large language models for human–robot interaction: A review. *Biomimetic Intelligence and Robotics*, 3(4), 100131. <https://doi.org/https://doi.org/10.1016/j.birob.2023.100131>
- Zheng, Y., Pujar, S., Lewis, B., Buratti, L., Epstein, E., Yang, B., ... Su, Z. (2021). D2a: A dataset built for ai-based vulnerability detection methods using differential analysis. *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 111–120. IEEE.