

IndoWebGen: *Language Model* untuk Pembuatan *Website* secara Otomatis Berdasarkan Instruksi Bahasa Indonesia melalui Metode LoRA

Alim Tegar Wicaksono, Handaru Jati
Universitas Negeri Yogyakarta
Email: alimtegar.2020@student.uny.ac.id

ABSTRAK

Di tengah pentingnya peran website di era saat ini, pembuatan *website* masih memerlukan waktu, tenaga, dan biaya yang tidak sedikit. Terlebih, pembuatan *website* juga kurang inklusif bagi mereka yang tidak memiliki pengetahuan dalam pengembangan web. Oleh karena itu, penelitian ini dilaksanakan demi memberikan solusi terhadap permasalahan tersebut melalui IndoWebGen: *Language Model* (LM) untuk pembuatan *website* secara otomatis berdasarkan instruksi Bahasa Indonesia melalui metode LoRA. Penelitian ini menggunakan metode R&D dengan mengadopsi model pengembangan CRISP-DM, yang terdiri dari 6 tahap: (1) pemahaman masalah dan tujuan, (2) pengumpulan dan pemahaman data, (3) penyiapan data, (4) pemodelan, (5) penyebaran, dan (6) pengujian. IndoWebGen dikembangkan dengan melakukan melakukan proses *finetuning* melalui metode LoRA pada model basis, Code Llama 7B, dengan menggunakan 500 data sintesis yang di-*generate* melalui metode self-instruct. IndoWebGen yang telah dikembangkan kemudian diuji kualitasnya melalui *human evaluation* untuk aspek *relevance*, *usability*, dan *visual design*. Penelitian ini menghasilkan IndoWebGen: LM untuk pembuatan *website* secara otomatis berdasarkan instruksi Bahasa Indonesia dengan hasil pengujian untuk aspek *relevance*, *usability*, dan *visual design* sebesar 5,933; 5,889; dan 6,089, yang berarti semuanya berkualitas "Baik". Dengan demikian, IndoWebGen dapat memberikan solusi pembuatan *website* yang lebih inklusif dan efisien dari segi waktu, tenaga, serta biaya, terutama bagi masyarakat Indonesia.

Kata kunci: *language model*, *website*, metode LoRA

ABSTRACT

In the midst of the important role of websites in the current era, creating a website still requires a lot of time, energy and costs. Moreover, website creation is also less inclusive for those who do not have knowledge in web development. Therefore, this research was carried out to provide a solution to this problem through IndoWebGen: Language Model (LM) for automated website creation based on Indonesian instructions using the LoRA method. This research uses the R&D method by adopting the CRISP-DM development model, which consists of 6 stages: (1) business understanding, (2) data collection and understanding, (3) data preparation, (4) modeling, (5) deployment, and (6) evaluation. IndoWebGen was developed by carrying out a finetuning process using the LoRA method on the base model, Code Llama 7B, using 500 synthetic data generated via the self-instruct method. The IndoWebGen that has been developed is then evaluated through human evaluation for aspects of relevance, usability and visual design. This research produced IndoWebGen: LM for automated website creation based on Indonesian language instructions with evaluation results for aspects of relevance, usability and visual design of 5.933, 5.889, and 6.089, which means they are all "Good" quality. In this way, IndoWebGen can provide website creation solutions that are more inclusive and efficient in terms of time, energy and costs, especially for the Indonesian people.

Keywords: *language model*, *website*, LoRA method

PENDAHULUAN

Ragam manfaat yang ditawarkan oleh *website* membuat penggunaannya semakin meningkat dalam beberapa tahun terakhir, terutama di Indonesia. Hal ini tercermin dalam

laporan Indonesia Website Awards 2020 yang diselenggarakan oleh Exabytes Indonesia, di mana tercatat ada 1057 *website* yang terdaftar sepanjang tahun 2020. Angka tersebut mengalami peningkatan sebesar 61,6% dibandingkan laporan tahun

sebelumnya [1]. Meskipun demikian, di tengah pentingnya peran *website* di era saat ini, pembuatan *website* masih memerlukan waktu, tenaga, dan biaya yang tidak sedikit. Selain itu, pembuatan *website* juga kurang bersifat inklusif. Bagi mereka yang ingin membuat *website* secara mandiri, mereka perlu memiliki pengetahuan dalam pengembangan web.

Memasuki era Revolusi Industri 5.0, teknologi termutakhir seperti *Artificial Intelligence* (AI) mulai dimanfaatkan. Salah satu teknologi AI tersebut adalah *Language Model* (LM) [2]. LM sendiri dapat dimanfaatkan sebagai solusi untuk mengoptimalkan proses pembuatan *website*. Hal itu dapat dicapai dengan memanfaatkan LM untuk meng-*generate* kode HTML/CSS untuk pembuatan *website* secara otomatis. Dengan demikian, pembuatan *website* akan menjadi lebih efisien dan inklusif.

Sebelumnya, telah dilakukan sejumlah penelitian mengenai pengembangan LM dalam hal pengkodean. Beberapa contoh di antaranya adalah Codex oleh OpenAI [3], CodeGen oleh Salesforce [4], StarCoder oleh Big Code [5], dan yang terbaru adalah Code Llama oleh Meta AI [6]. Namun, meskipun model-model tersebut menunjukkan kinerja yang baik dalam memahami kode, model-model tersebut tidak dikembangkan atau dilatih secara khusus untuk meng-*generate* kode HTML/CSS untuk pembuatan *website*. Selain itu, model-model tersebut juga dilatih dengan *dataset* yang sebagian besarnya berbahasa Inggris, sehingga pemahamannya terhadap Bahasa Indonesia kurang terlalu baik. Untuk membuat model mampu melakukan tugas tertentu, seperti meng-*generate* kode HTML/CSS, dan membuatnya mampu lebih memahami Bahasa Indonesia, model tersebut dapat dilatih kembali dengan *dataset* berisi instruksi untuk meng-*generate* kode HTML/CSS dalam Bahasa Indonesia beserta *output*-nya. Proses ini disebut

finetuning, yaitu pendekatan yang menginisialisasi parameter model untuk tugas yang diinginkan dari parameter yang telah dilatih sebelumnya pada tugas yang lain [7]. Adapun untuk melakukan *finetuning* pada model dengan jumlah parameter yang banyak, seperti LM dengan arsitektur Transformer, metode LoRA (*Low-Rank Adaptation*) merupakan opsi yang cocok untuk dipilih. Hal ini karena metode LoRA dapat mengurangi jumlah parameter yang perlu dilatih hingga 1000 kali lipat dan kebutuhan GPU hingga 3 kali lipat [8].

Berdasarkan penjelasan di atas, peneliti terdorong untuk mengembangkan atau melatih IndoWebGen: LM yang mampu meng-*generate* kode HTML/CSS untuk pembuatan *website* secara otomatis berdasarkan instruksi Bahasa Indonesia melalui metode LoRA. Diharapkan teknologi ini dapat memberikan solusi pembuatan *website* lebih inklusif dan efisien dari segi waktu, tenaga, serta biaya, terutama bagi masyarakat Indonesia.

METODE

Penelitian ini menggunakan metode R&D (*Research & Development*) [9] dengan mengadopsi model pengembangan CRISP-DM (*CRoss-Industry Standard Process for Data Mining*) [10], yang terdiri dari 6 tahap: (1) pemahaman masalah dan tujuan, (2) pengumpulan dan pemahaman data, (3) penyiapan data, (4) pemodelan, (5) penyebaran, dan (6) pengujian. Pada tahap pemodelan, IndoWebGen dikembangkan dengan melakukan proses *finetuning* melalui metode LoRA pada model basis, Code Llama 7B, dengan menggunakan 500 data sintesis yang di-*generate* melalui metode *self-instruct* [11].

Sedangkan pada tahap pengujian, IndoWebGen diuji kualitasnya melalui *human evaluation* [12] untuk aspek *relevance*, *usability*, dan *visual design*.

Kualitas IndoWebGen dalam aspek *relevance* mencakup kualitas IndoWebGen dalam meng-*generate website* secara akurat, benar, dan logis, sesuai instruksi dalam Bahasa Indonesia. Sedangkan kualitas IndoWebGen dalam aspek *usability* mencakup kualitas IndoWebGen dari segi kegunaan; keefisienan waktu, tenaga, dan biaya; serta inklusivitas bagi individu yang tidak memiliki pengetahuan dalam pengembangan web. Dan kualitas IndoWebGen dalam aspek *visual design* mencakup kualitas *website* yang di-*generate* oleh IndoWebGen dari segi tata letak, tipografi, dan warna.

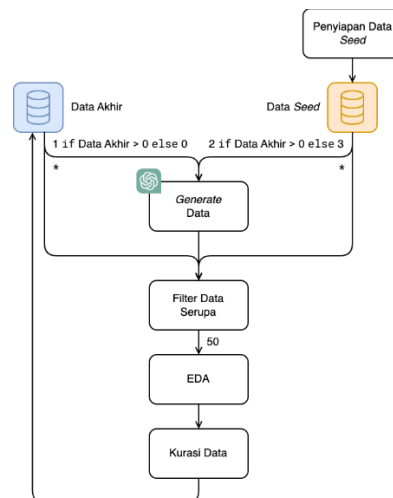
Teknik pengumpulan data yang digunakan untuk melakukan *human evaluation* adalah *kuesioner* yang terdiri dari 3 pertanyaan untuk masing-masing aspek yang diuji. Data dikumpulkan dari subjek penelitian yang terdiri dari 30 pengujian yang berasal dari kalangan masyarakat umum, yang tidak memiliki pengetahuan mendalam dalam pengembangan web. Teknik analisis data yang digunakan untuk menganalisis data *kuesioner* adalah analisis metrik skala Likert 7 poin [13]. Pada skala Likert 7 poin ini, tiap pertanyaan dalam *kuesioner*

memiliki skala penilaian yang terdiri dari 7 tingkat untuk menilai kualitas dari IndoWebGen. Skala penilaian kualitas ini ditetapkan pada tabel berikut:

Tabel 1. Skala Penilaian Kualitas

Nilai	Kualitas
1	“Sangat Buruk”
2	“Buruk”
3	“Agak Buruk”
4	“Biasa”
5	“Agak Baik”
6	“Baik”
7	“Sangat Baik”

Data *kuesioner* yang diperoleh kemudian dihitung rata-rata poin atau skor yang diberikan oleh setiap pengujian, untuk masing-masing aspek *human evaluation: relevance*, *usability*, dan *visual design*. Setelah himpunan skor rata-rata aspek *relevance*, *usability*, dan *visual design* dari seluruh pengujian diketahui, dicari skor *relevance*, *usability*, dan *visual design* melalui nilai rata-rata dari masing-masing himpunan sebagai ukuran pemusatan data (*central tendency*) yang mewakili keseluruhan penilaian kualitas IndoWebGen dalam aspek *relevance*, *usability*, dan *visual design* [14].



Gambar 1. Alur *Self-Instruct*

HASIL DAN PEMBAHASAN

Pemahaman Masalah dan Tujuan

Tahap ini melibatkan penjelasan lebih lanjut mengenai tujuan penelitian ini berdasarkan permasalahan yang teridentifikasi serta cara untuk mencapainya. Hal tersebut meliputi:

Pembuatan *website* masih kurang inklusif dan efisien dari segi waktu, tenaga, serta biaya. Oleh karena itu, fokus penelitian ini adalah untuk mengembangkan IndoWebGen: *Language Model* (LM) untuk pembuatan *website* secara otomatis berdasarkan instruksi Bahasa Indonesia, guna memberikan solusi pembuatan *website* yang lebih efisien dari segi waktu, tenaga, dan biaya, serta lebih inklusif, terutama bagi masyarakat Indonesia.

LM yang dikembangkan saat ini belum dikembangkan secara khusus untuk pembuatan *website* dengan pemahaman instruksi Bahasa Indonesia dengan baik. Oleh karena itu, fokus penelitian ini adalah untuk mengembangkan IndoWebGen melalui metode LoRA dan juga menguji kualitasnya dalam membuat *website* sesuai instruksi Bahasa Indonesia melalui *human evaluation* untuk aspek *usability*, *relevance*, dan *visual design*.

Pengumpulan dan Pemahaman Data

Tahap ini mencakup *self-instruct* untuk pengumpulan data awal dan pendeskripsian data secara otomatis, EDA untuk eksplorasi data, dan kurasi data untuk verifikasi kualitas data.

1. *Self-Instruct*

Dalam penelitian ini, metode *self-instruct* dilakukan dengan memanfaatkan model GPT-3.5 untuk meng-*generate* 500 data sintesis berdasarkan 7 data *seed* atau data contoh, untuk digunakan dalam pemodelan IndoWebGen. Alur dari *self-instruct* ini dapat dilihat pada **Gambar 1**.

2. EDA (*Exploratory Data Analysis*)

EDA [15] dimulai dengan pemaparan ikhtisar data, diikuti oleh analisis data hilang, data duplikat, instruksi dan *output*, *prompt*, *token*, dan jenis *website*. Hasil dari setiap analisis tersebut dijabarkan sebagai berikut:

a. Ikhtisar Data

Diketahui bahwa jumlah data adalah 500 dan terdapat 4 fitur data, yaitu *instruction*, *output*, *most_similar_instructions*, dan *avg_similarity_score*.

b. Analisis Data Hilang

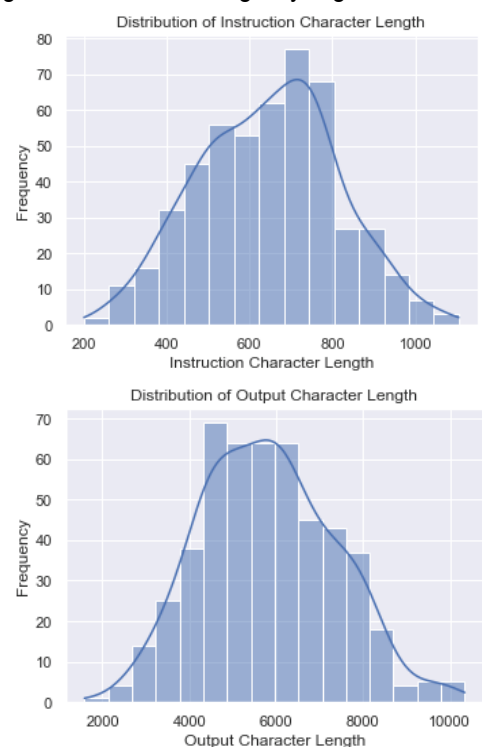
Diketahui bahwa tidak ada data hilang yang ditemukan.

c. Analisis Data Duplikat

Diketahui bahwa tidak ada data duplikat yang ditemukan.

d. Analisis Instruksi dan *Output*

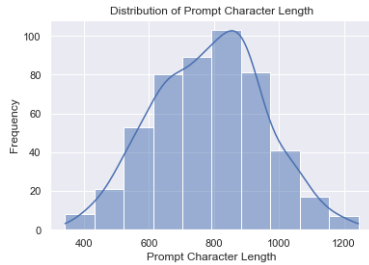
Diketahui panjang instruksi dan *output* memiliki distribusi yang tampak simetris, mengindikasikan kemiringan yang minimal.



Gambar 2. Distribusi Panjang Instruksi dan *Output*

e. Analisis Prompt

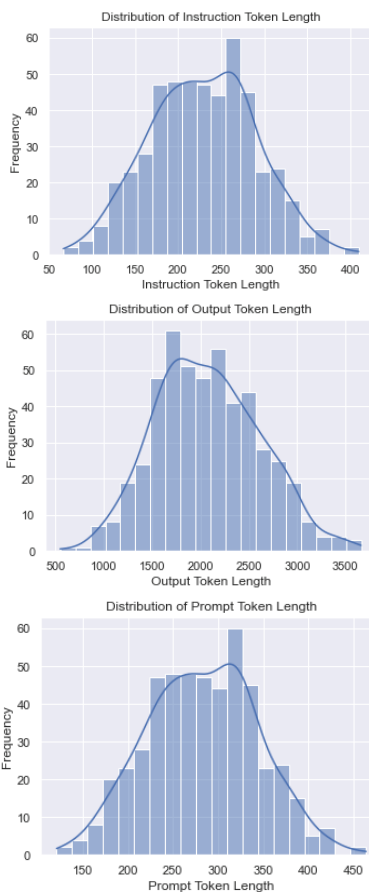
Diketahui panjang *prompt* memiliki distribusi yang tampak simetris, mengindikasikan kemiringan yang minimal.



Gambar 3. Distribusi Panjang Prompt

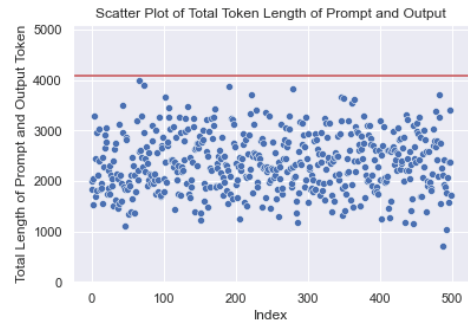
f. Analisis Token

Diketahui panjang *token* instruksi, *output*, dan *prompt* memiliki distribusi yang tampak simetris, mengindikasikan kemiringan yang minimal.



Gambar 4. Distribusi Panjang Token Instruksi, Output, dan Prompt

Selain itu, diketahui juga bahwa tidak ada data dengan total jumlah *token prompt* dan *output* yang melebihi 4096 *token*.



Gambar 5. Distribusi Panjang Instruksi dan Output

g. Analisis Jenis Website

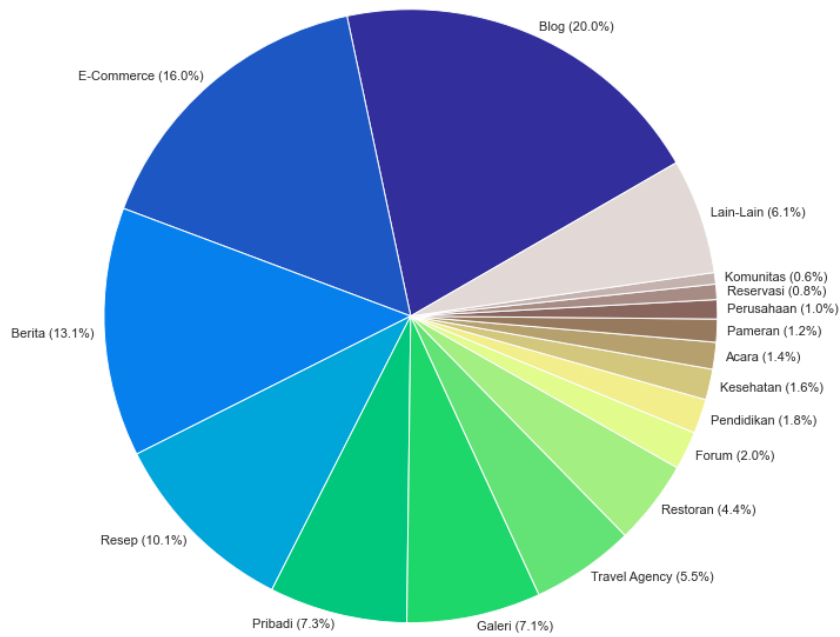
Diketahui bahwa jenis *website* yang ada pada instruksi cukup bervariasi, dengan yang paling umum adalah jenis *website blog*, diikuti oleh jenis *website e-commerce*, berita, resep, pribadi, galeri, *travel agency*, dan restoran. Hal ini ditunjukkan pada **Gambar 6**.

3. Kurasi Data

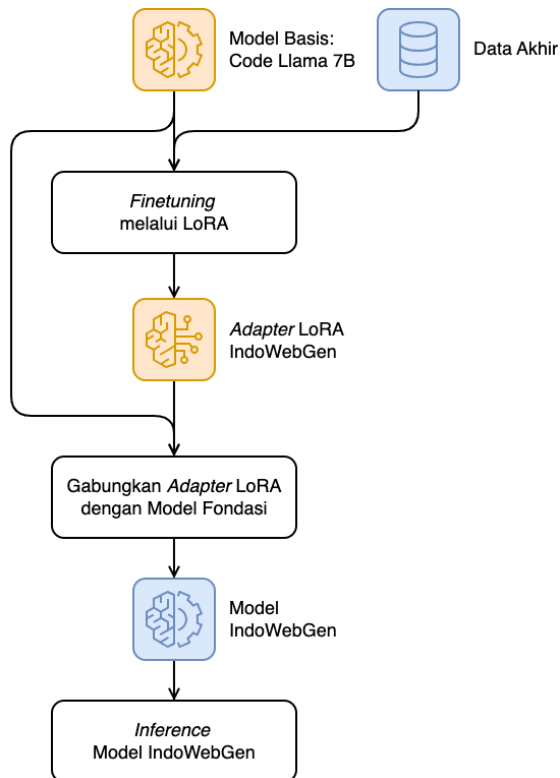
Kurasi data dilakukan menggunakan *tool* yang dikembangkan sendiri bernama IndoWebGen Curator. IndoWebGen Curator memiliki berbagai fitur, seperti melihat, menavigasi, memperbarui, menghapus, menyimpan, dan mengekspor ke JSON untuk data yang berupa pasangan *instruksi-output* kode *website*. Selain itu, IndoWebGen Curator juga dilengkapi fitur untuk menampilkan *website* dari data *output* kode, sehingga desain visual dari *website* dapat dikurasi juga.

Penyiapan Data

Pada tahap ini, data yang telah dikumpulkan perlu melalui proses seleksi untuk memilih fitur data yang relevan. Fitur data relevan yang diseleksi antara lain adalah fitur *instruksi* dan *output*. Setelah melalui proses seleksi, selanjutnya data diunggah ke repositori *dataset* pada platform Hugging Face untuk memudahkan pemuatan data saat digunakan dalam pemodelan.



Gambar 6. Persentase Jenis Website



Gambar 7. Alur Pemodelan

Pemodelan

Proses pemodelan IndoWebGen melibatkan *finetuning* melalui LoRA, penggabungan *adapter* LoRA dengan model basis, dan *inference* model IndoWebGen. Alur dari proses ini ditunjukkan pada **Gambar 7**.

1. *Finetuning* melalui LoRA

Finetuning melalui LoRA dilakukan pada model basis, Code Llama 7B, menggunakan data akhir yang telah disiapkan sebelumnya. Dan seperti yang telah dijelaskan sebelumnya, *finetuning* melalui LoRA dapat mengurangi kebutuhan sumber daya GPU dengan mengurangi jumlah parameter yang perlu dilatih. Diketahui jumlah parameter yang perlu dilatih, yang dikurangi melalui metode LoRA, adalah 4.194.304, yakni **0,622%** dari total parameter model yang berjumlah 6.742.740.992.

Saat proses *finetuning* telah selesai, hasil *finetuning* kemudian disimpan ke dalam direktori *output*. Hasil *finetuning* ini mencakup *adapter* LoRA dan laporan perubahan metrik selama proses *finetuning*. Terlebih, dari keseluruhan proses *finetuning* yang telah dilakukan, diketahui bahwa model mampu memperoleh *training loss* sebesar **0,1235** dan *evaluation loss* sebesar **0,1346**.

2. Penggabungan *Adapter* LoRA dengan Model Basis

Adapter LoRA yang dihasilkan pada proses *finetuning* sebelumnya perlu *digabungkan* dengan model basis, Code Llama 7B, untuk menghasilkan model IndoWebGen yang dapat disimpan dan digunakan kembali.

3. *Inference* Model IndoWebGen

Inference model IndoWebGen bertujuan untuk menerapkan model IndoWebGen yang

telah dihasilkan untuk meng-*generate website* berdasarkan instruksi dalam Bahasa Indonesia. Model IndoWebGen sendiri diaplikasikan pada sebuah *website* yang dapat menyajikan antarmuka untuk menggunakan model IndoWebGen. *Website inference* ini ditunjukkan pada **Gambar 8**.

Pengujian

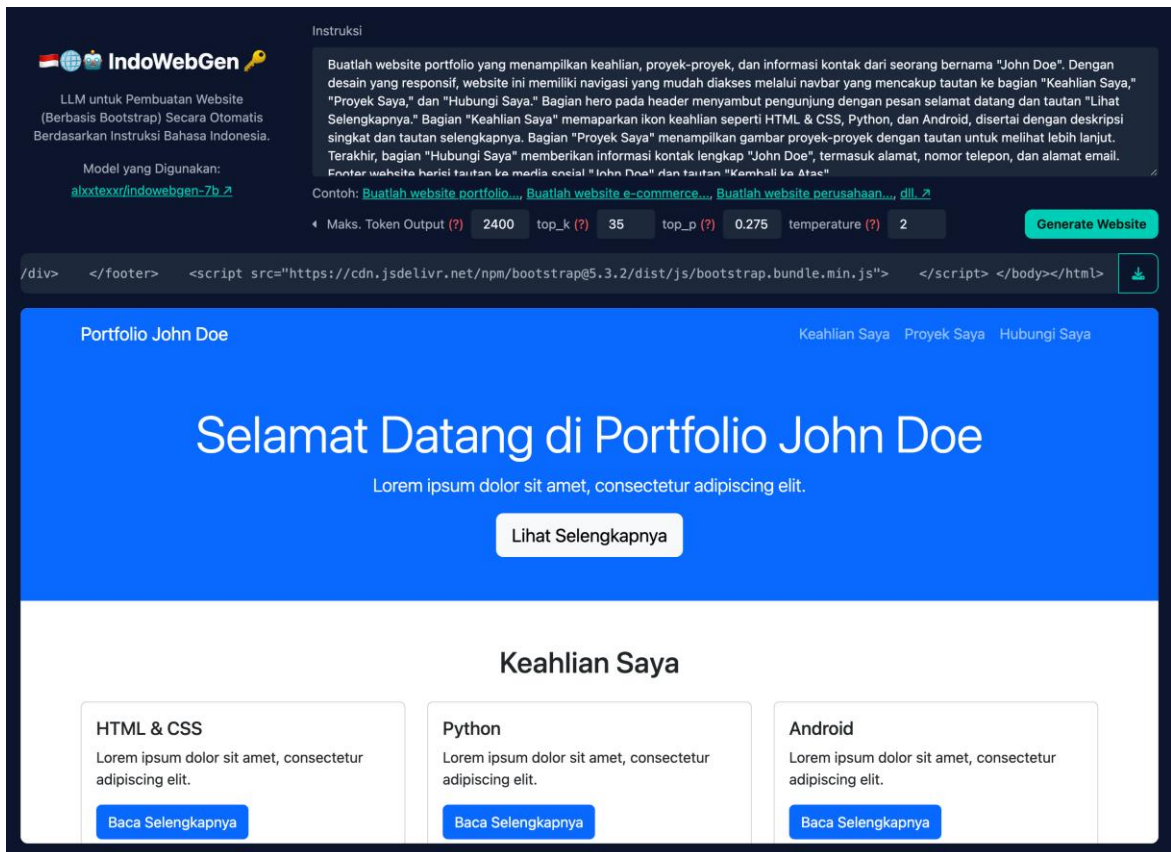
Sebelum melakukan pengujian, para penguji diwajibkan untuk mencoba menggunakan IndoWebGen yang telah diaplikasikan pada *website inference*. Setelah mencobanya, para penguji kemudian diarahkan untuk mengisi kuesioner guna menilai kualitas *output* dari IndoWebGen. Data kuesioner yang diperoleh kemudian dianalisis untuk mengetahui hasil pengujian kualitas IndoWebGen dalam aspek *relevance*, *usability*, dan *visual design*. Hasil pengujian kualitas IndoWebGen untuk setiap aspeknya dijabarkan pada tabel di bawah ini:

Tabel 2. Hasil Pengujian

Aspek	Kualitas
<i>Relevance</i>	5,933 (“Baik”)
<i>Usability</i>	5,889 (“Baik”)
<i>Visual Design</i>	6,089 (“Baik”)

Penyebarluasan

Penyebarluasan IndoWebGen dilakukan dengan menyusun laporan penelitian berupa laporan skripsi dan jurnal, serta membagikan *file* model IndoWebGen melalui platform Hugging Face.



Gambar 8. Website Inference Model IndoWebGen

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Berdasarkan pengembangan yang telah dilakukan melalui metode LoRA, telah dihasilkan IndoWebGen: *language model* untuk pembuatan *website* secara otomatis berdasarkan instruksi Bahasa Indonesia. Melalui IndoWebGen, pembuatan *website* dapat menjadi lebih inklusif dan efisien dari segi waktu, tenaga, serta biaya, terutama bagi masyarakat Indonesia.
2. Berdasarkan pengujian yang telah dilakukan melalui *human evaluation*, IndoWebGen memperoleh skor *relevance*, *usability*, dan *visual design* sebesar **5,933**; **5,889**; dan **6,089**. Ini mengindikasikan bahwa kualitas IndoWebGen dalam aspek *relevance*, mencakup kualitasnya dalam *generate website* secara akurat,

benar, dan logis, sesuai instruksi dalam Bahasa Indonesia, rata-rata dinilai mendekati **"Baik"**. Selain itu, kualitas IndoWebGen dalam aspek *usability*, yang mencakup kegunaan, keefisienan waktu, tenaga, dan biaya, serta inklusivitas bagi orang awam, juga rata-rata dinilai mendekati **"Baik"**. Terakhir, kualitas IndoWebGen dalam aspek *visual design*, yang mencakup tata letak, tipografi, dan warna, kualitas *website* yang di-*generate*, rata-rata dinilai **"Baik"**.

DAFTAR RUJUKAN

- [1] Kurniawan F, "Pengguna Website di Indonesia Naik 61,6% Sepanjang 2020," *SINDOnews*, Apr. 07, 2021. Accessed: Oct. 02, 2023. [Online]. Available: <https://tekno.sindonews.com/read/389902/207/pengguna-website-di->

[indonesia-naik-616-sepanjang-2020-1617800664](https://doi.org/10.1617800664)

- [2] D. Jurafsky and J. H. Martin, "Speech and Language Processing," (3rd ed. draft)., 2023. Accessed: Oct. 11, 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3>
- [3] M. Chen *et al.*, "Evaluating Large Language Models Trained on Code," Jul. 2021.
- [4] E. Nijkamp *et al.*, "CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis," Mar. 2022.
- [5] R. Li *et al.*, "StarCoder: may the source be with you!," May 2023.
- [6] B. Rozière *et al.*, "Code Llama: Open Foundation Models for Code," Aug. 2023.
- [7] W. Ouyang, X. Wang, C. Zhang, and X. Yang, "Factors in finetuning deep model for object detection with long-tail distribution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 864–873.
- [8] E. J. Hu *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.
- [9] D. Sugiyono, "Metode penelitian pendidikan pendekatan kuantitatif, kualitatif dan R&D," 2013.
- [10] C. Shearer, "The CRISP-DM model: the new blueprint for data mining," *Journal of data warehousing*, vol. 5, no. 4, pp. 13–22, 2000.
- [11] Y. Wang *et al.*, "Self-Instruct: Aligning Language Models with Self-Generated Instructions," Dec. 2022.
- [12] H. Schuff, L. Vanderlyn, H. Adel, and N. T. Vu, "How to do human evaluation: A brief introduction to user studies in NLP," *Nat Lang Eng*, vol. 29, no. 5, pp. 1199–1222, 2023, doi: 10.1017/S1351324922000535.
- [13] C. van der Lee, A. Gatt, E. van Miltenburg, S. Wubben, and E. Kraemer, "Best practices for the human evaluation of automatically generated text," in *Proceedings of the 12th International Conference on Natural Language Generation*, K. van Deemter, C. Lin, and H. Takamura, Eds., Tokyo, Japan: Association for Computational Linguistics, Oct. 2019, pp. 355–368. doi: 10.18653/v1/W19-8643.
- [14] G. M. Sullivan and A. R. Artino Jr, "Analyzing and interpreting data from Likert-type scales," *J Grad Med Educ*, vol. 5, no. 4, pp. 541–542, 2013.
- [15] J. Fox, J. W. Tukey, D. R. McNeil, B. H. Erickson, and T. A. Nosanchuk, "Exploratory Data Analysis.," 1978. [Online]. Available: <https://api.semanticscholar.org/CorpusID:42027475>