

Implementasi *Fine-Tuning* Untuk Prediksi Solusi Dari Kalimat Masalah Pada Artikel Ilmiah Menggunakan *Large Language Models* (LLM)

Alfiyatu Nurinayah, Sam Farisa Chaerul Haviana

Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung

Correspondence Author: alfiyatu@std.unissula.ac.id

Abstract

Artikel ilmiah merupakan sumber utama pengembangan ilmu pengetahuan, yang berperan penting dalam menyampaikan hasil penelitian, identifikasi masalah, dan solusi yang diusulkan. Namun, di dalam artikel ilmiah tidak semua menyajikan solusi secara eksplisit untuk setiap permasalahan yang dibahas. Penelitian ini bertujuan untuk mengembangkan model berbasis *fine-tuning* menggunakan LLaMA 3.2 1B Instruct untuk memprediksi solusi berdasarkan pernyataan masalah. Model dilatih menggunakan 99 pasangan kalimat masalah-solusi untuk mengenali pola hubungan antara keduanya. Evaluasi model dilakukan dengan menggunakan metrik ROUGE, yang menghasilkan skor ROUGE-1 sebesar 0,611, ROUGE-2 sebesar 0,496, dan ROUGE-L sebesar 0,576. Hasil evaluasi tersebut menunjukkan bahwa model mencapai skor yang cukup baik. Selama proses pelatihan, nilai kerugian menurun dari 0,4394 pada epoch pertama menjadi 0,0286 pada epoch kelima, yang menunjukkan peningkatan kinerja model dalam memahami hubungan antara pernyataan masalah dan solusinya. Namun, model tersebut masih menghadapi kendala dalam memprediksi solusi untuk masalah yang tidak termasuk dalam data pelatihan, dan hasil prediksinya terkadang kurang relevan ketika diterapkan pada aplikasi nyata. Salah satu faktor yang dapat memengaruhi keterbatasan ini adalah keterbatasan sumber daya komputasi. Oleh karena itu, diperlukan penelitian lebih lanjut untuk meningkatkan akurasi dan stabilitas model, terutama pada sistem dengan keterbatasan komputasi.

Keyword: *Fine-tuning*, Llama3.2 1B Instruct, Prediksi Solusi, Artikel Ilmiah, NLP.

1. PENDAHULUAN

Artikel ilmiah merupakan sumber utama pengembangan ilmu pengetahuan, yang berperan penting dalam menyampaikan hasil penelitian, identifikasi masalah, dan solusi yang diusulkan. Namun, di dalam artikel ilmiah tidak semua menyajikan solusi secara eksplisit untuk setiap permasalahan yang dibahas. Banyak artikel ilmiah hanya menggambarkan masalah atau analisis tanpa memberikan alternatif penyelesaian, sehingga pembaca, khususnya akademisi dan peneliti, harus berupaya lebih keras untuk menemukan solusi yang relevan.

Seiring dengan peningkatan jumlah artikel ilmiah yang diterbitkan setiap tahun, masalah ini menjadi semakin kompleks. Berdasarkan data Database Jurnal Ilmiah Indonesia (ISJD), pada tahun 2017 terdapat 23.980 jurnal yang diunggah, dengan 23.910 jurnal yang dibaca dan 23.912 jurnal yang diunduh. Pada tahun 2018, jumlah jurnal yang diunggah meningkat signifikan menjadi 34.551, dengan 34.231 jurnal dibaca dan 34.215 jurnal diunduh (Fatmalasari dan Lumbanraja 2022) [1]. Peningkatan jumlah artikel ini mencerminkan kemajuan dalam produksi ilmu pengetahuan, namun juga menimbulkan tantangan besar dalam menavigasi informasi yang tersebar luas, terutama untuk mencari solusi yang mungkin tidak secara eksplisit disebutkan dalam teks artikel.

Ollama merupakan inovasi di bidang pengolahan bahasa alami yang telah memberikan kontribusi besar dalam memahami dan menganalisis teks. Ollama merupakan framework yang digunakan untuk menjalankan model bahasa besar secara lokal [2]. Model-model yang dikembangkan oleh Ollama, seperti Llama-3.2, adalah contoh dari LLM yang memiliki kemampuan luar biasa dalam menangani berbagai tugas NLP. Llama-3.2 dirancang dengan arsitektur transformator yang sangat efisien untuk berbagai tugas pemrosesan bahasa alami, termasuk prediksi solusi dari kalimat permasalahan.

Llama-3.2 menggunakan pendekatan *auto-regresif*, di mana input teks diproses untuk menghasilkan output yang relevan dan berkualitas tinggi. Model ini mampu menangani tugas-tugas kompleks seperti penulisan ulang, ringkasan teks, dan bahkan menjawab pertanyaan berdasarkan teks yang diberikan. Meskipun

LLM dalam bentuk generik memiliki kemampuan yang luas, sering kali perlu dioptimalkan untuk tugas-tugas tertentu, seperti identifikasi kalimat solusi dalam artikel ilmiah.

Llama-3.2-1B-Instruct adalah varian yang dioptimalkan dari Llama-3.2 untuk tugas-tugas instruksional dan kontekstual melalui proses *fine-tuning*. *Fine-tuning* adalah teknik pengoptimalan model pembelajaran mesin yang bekerja dengan menyesuaikan parameter-parameter seperti bobot dan bias pada model yang telah dilatih sebelumnya untuk disesuaikan dengan tugas baru [3]. *Fine-tuning* memungkinkan model untuk belajar dari *dataset* khusus yang lebih kecil dan spesifik untuk tugas prediksi kalimat solusi, sehingga meningkatkan performa dan akurasi model.

Penelitian ini bertujuan untuk mengimplementasikan *fine-tuning* pada model Llama-3.2-1B-Instruct guna memprediksi kalimat solusi dari masalah yang disajikan dalam artikel ilmiah. *Fine-tuning* menjadi sangat relevan karena memungkinkan adaptasi model LLM yang telah dilatih dengan *dataset* yang luas untuk dioptimalkan pada tugas yang lebih spesifik. Dengan penelitian ini, diharapkan pembaca akan lebih mudah mendapatkan kalimat solusi, sehingga memudahkan peneliti, akademisi, atau pembaca lainnya dalam mengakses solusi dari berbagai masalah yang dibahas dalam artikel ilmiah

Natural Language Processing (NLP)

NLP merupakan kemampuan suatu komputer mengenali dan memproses bahasa alami manusia, baik dalam bentuk lisan maupun tulisan, sebagaimana digunakan dalam komunikasi sehari-hari. NLP secara sederhana bertujuan untuk memungkinkan komputer memahami instruksi yang ditulis dalam bahasa manusia [4]. Untuk konteks penelitian ini, NLP digunakan untuk menganalisis teks pada artikel ilmiah, yang bertujuan untuk mendeteksi kalimat yang memuat solusi dari permasalahan yang dibahas. Dengan menggunakan NLP, penelitian ini berfokus pada pengembangan sistem berbasis model besar untuk mengidentifikasi solusi secara otomatis, sehingga mempermudah proses pemahaman artikel ilmiah oleh akademisi dan peneliti.

Large Language Models (LLM)

LLM merupakan model bahasa skala besar yang dilatih dengan banyak data teks untuk memahami serta menghasilkan bahasa alami. Model ini diterapkan dalam berbagai bidang, seperti penerjemahan, pembuatan konten, dan chatbot untuk dukungan emosional [5]. LLM seperti Llama3.2 1B Instruct digunakan untuk menangkap pola hubungan dalam teks artikel ilmiah, termasuk pola hubungan antara kalimat masalah dan solusi. LLM menawarkan keunggulan berupa fleksibilitas untuk menyelesaikan berbagai tugas NLP hanya dengan memodifikasi format *input* dan *output*. Contoh implementasi LLM dalam penelitian ini adalah menghasilkan kalimat solusi dari kalimat masalah melalui pendekatan *text-to-text*. Model ini dilatih secara khusus untuk memahami konteks dalam artikel ilmiah dan menghasilkan kalimat solusi yang relevan, dengan fokus pada kategori masalah-solusi, tantangan-jawaban, peluang-jawaban, dan kelemahan-peningkatan.

Fine Tuning

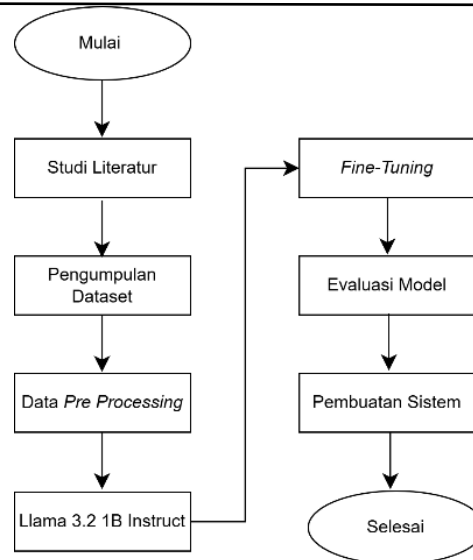
Fine-Tuning merupakan teknik pembelajaran mesin yang umum digunakan untuk menyesuaikan model bahasa yang telah dilatih sebelumnya agar sesuai dengan tugas-tugas lanjutan tertentu. Strategi ini memperbarui bobot model dengan melatihnya menggunakan *dataset* berlabel yang relevan dengan tugas yang ditargetkan [6]. Proses penyempurnaan diawali dengan penggunaan model yang sebelumnya telah dilatih pada kumpulan data besar sebagai model dasar atau model terlatih. Proses *fine-tuning* memungkinkan model yang awalnya bersifat umum untuk mengkhususkan diri pada tugas tertentu, seperti prediksi kalimat solusi. *Fine-Tuning* diterapkan pada model Llama 3.2 1B Instruct yang telah dilatih sebelumnya menggunakan *dataset* generik. Proses *fine-tuning* melibatkan pelatihan ulang model dengan *dataset* berupa pasangan kalimat masalah-solusi yang diambil dari artikel ilmiah. Tahapan ini bertujuan agar model dapat mengenali pola hubungan antara masalah dan solusi dalam teks. Dengan *fine-tuning*, model Llama 3.2 yang awalnya bersifat umum dioptimalkan untuk tugas prediksi kalimat solusi, menghasilkan prediksi yang lebih relevan dengan konteks artikel ilmiah.

Ollama

Ollama merupakan kerangka kerja yang dirancang untuk menjalankan model bahasa besar (LLM) langsung di perangkat pengguna. Dengan menggunakan Ollama, model kecerdasan buatan dapat dijalankan secara lokal tanpa memerlukan ketergantungan pada layanan cloud, sehingga memberikan fleksibilitas dan efisiensi dalam penggunaannya [2]. Ollama mempunyai banyak model di dalamnya, salah satunya yaitu Llama 3.2 1B Instruct yang merupakan koleksi model generative yang telah dilatih sebelumnya dan disesuaikan dengan instruksi dalam ukuran 1B dan 3B. Llama juga merupakan model bahasa *auto-regresif* yang menggunakan arsitektur transformator yang dioptimalkan. Model Llama dirancang dengan pendekatan terstruktur yang efisien dan optimal, memprioritaskan tugas generatif melalui model khusus *Decoder*. Pendekatan ini menyederhanakan proses pelatihan dan meningkatkan kecepatan eksekusi [7].

2. METODE PENELITIAN

Penelitian ini menggunakan metode *Fine-Tuning* menggunakan LLM. Dalam penelitian ini, LLM akan di training untuk memprediksi solusi dari kalimat masalah yang terdapat dalam artikel ilmiah. Tahapan yang perlu dilakukan dalam penelitian ini adalah sebagai berikut :



Gambar 1. Alur Penelitian

1. Studi Literatur

Peneliti akan meninjau berbagai sumber literatur, termasuk *e-book*, artikel, jurnal, penelitian sebelumnya seperti tesis dan skripsi, serta informasi dari berbagai situs web yang relevan dengan topik penelitian. Studi literatur ini berfokus pada pemahaman teori mengenai *fine-tuning*, LLM, metode prediksi berbasis teks, serta penerapan model *machine learning* untuk menganalisis solusi, tantangan, peluang, dan kelemahan yang diidentifikasi dalam artikel ilmiah.

2. Pengumpulan data

Dataset pada penelitian ini dikumpulkan secara manual karena belum tersedianya *dataset* yang sesuai. Data diperoleh dari 100 artikel di Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK) dengan memilih kalimat yang mengandung masalah dan solusi. Kalimat-kalimat tersebut dikelompokkan menjadi empat kategori utama: Masalah-Solusi, Tantangan-Jawaban, Peluang-Jawab, dan Kelemahan-Peningkatan. Jika suatu artikel mempunyai lebih dari satu kalimat masalah, kalimat tersebut dicatat, dan jika solusi tidak tersedia, asumsi dibuat berdasarkan konteks yang relevan dan konsultasi dengan pembimbing. Setelah mereview artikel, data terkumpul sebanyak 99 untuk *training*. Setelah data dikumpulkan, semuanya divalidasi untuk memastikan kesesuaian untuk penelitian.

3. Preprocessing data

Preprocessing data merupakan tahapan krusial dalam analisis data yang dilakukan untuk membersihkan, menyesuaikan format, dan menyiapkan data agar lebih siap dan akurat saat dianalisis [8]. Ada beberapa tahap yang dilakukan dalam data preprocessing, yaitu:

- Case folding*, *case folding* atau normalisasi huruf merupakan salah satu tahapan dalam *text preprocessing* yang bertujuan untuk menyamakan format huruf dengan mengubah seluruh teks menjadi huruf kecil atau huruf besar. Proses ini dilakukan agar teks lebih mudah dibandingkan dan dianalisis tanpa terpengaruh oleh perbedaan penggunaan huruf besar. *Case folding* yang dilakukan yaitu *text.lower* untuk mengubah teks menjadi huruf kecil semua. Mengubah semua karakter menjadi huruf kecil membantu meningkatkan konsistensi data, membuat analisis dan pemrosesan lebih lanjut menjadi lebih mudah [9].
- Normalisasi spasi merupakan salah satu langkah dalam pemrosesan awal teks yang bertujuan untuk merapikan teks dengan menghilangkan spasi berlebih. Dengan cara ini, teks menjadi lebih mudah diproses tanpa terpengaruh oleh jumlah spasi yang tidak seragam.

4. Pemilihan Konfigurasi Model

Setelah data siap, langkah selanjutnya adalah memilih dan mengatur model Llama 3.2 1B Instruct untuk proses *fine-tuning*. Model ini dipilih karena kemampuannya dalam memahami serta menghasilkan teks sesuai dengan instruksi yang diberikan. Dalam tahap konfigurasi, beberapa hyperparameter seperti *batch size*, *learning rate*, dan jumlah *epoch* ditetapkan agar proses pelatihan berjalan dengan optimal serta menghindari overfitting. Selain itu, format *input output* juga disesuaikan, dengan merancang *prompt* tertentu agar model dapat lebih memahami tugasnya dalam memprediksi solusi dari kalimat masalah.

Prompt yang digunakan dalam penelitian ini dirancang untuk memberikan konteks yang jelas kepada model, dengan format seperti "Question: {input_question}\nAnswer: The output not be more than 2 sentences"

Dengan format ini, model dapat memahami bahwa input berupa sebuah pertanyaan atau masalah yang membutuhkan solusi sebagai *output* nya. Selain itu, *output* juga tidak lebih dari 2 kalimat.

5. Fine-Tuning Model

Fine-tuning merupakan proses penyesuaian model yang telah melalui tahapan sebelumnya (*pre-trained* model) agar lebih efektif dalam menyelesaikan tugas tertentu atau menangani data tertentu. Model pra-terlatih umumnya dilatih dengan kumpulan data yang luas dan bervariasi, sehingga mampu memahami berbagai jenis data secara umum [10].

Pada tahap pelatihan, model Llama 3.2 1B instruct dilatih menggunakan *dataset* yang telah diproses sebelumnya. Data pelatihan ada sebanyak 99 untuk memastikan model mampu mengenali pola hubungan antara kalimat masalah dan solusi dengan baik.

Fine tuning model Llama 3.2 1B Instruct pada penelitian ini dilakukan untuk menyesuaikan model agar lebih optimal dalam memprediksi solusi berdasarkan kalimat permasalahan pada artikel ilmiah. Proses ini menggunakan pendekatan pembelajaran *sequence to sequence*, dimana model dilatih untuk memahami hubungan antara *input* berupa kalimat masalah dan *output* berupa solusi yang sesuai dengan konteks.

Pelatihan dilakukan dengan strategi evaluasi secara berkala untuk memastikan model dapat beradaptasi dengan baik terhadap data yang digunakan. Jumlah *epoch* ditentukan agar model memiliki waktu yang cukup untuk mempelajari pola hubungan antara masalah dan solusi, sedangkan ukuran *batch* disesuaikan agar pelatihan berjalan lebih efisien. Selain itu, penyesuaian kecepatan pembelajaran dilakukan untuk menjaga keseimbangan antara stabilitas pelatihan dan kecepatan konvergensi model.

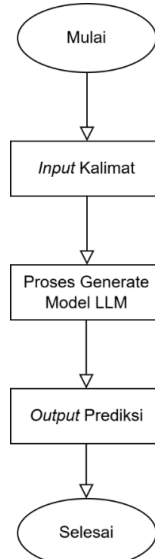
6. Evaluasi

Setelah model selesai dilatih, dilakukan validasi menggunakan metrik evaluasi seperti ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*). Metrik ini mengukur kesamaan antara solusi yang diprediksi oleh model dengan solusi yang ada dalam *dataset*. Validasi bertujuan untuk mengukur relevansi, akurasi, dan kualitas prediksi model.

ROUGE merupakan pendekatan penilaian yang banyak digunakan dalam NLP, khususnya untuk menilai kualitas ringkasan otomatis dan terjemahan mesin. Evaluasi ini dilakukan dengan mengevaluasi perbandingan hasil prediksi solusi yang diperoleh dari sistem dan dibandingkan dengan data acuan, yang disebut *ground truth summarization*. ROUGE bekerja menggunakan pendekatan *intrinsic*, yakni menganalisis tingkat kesamaan antara hasil prediksi dan referensi yang diharapkan.

7. Analisis Sistem

Dalam penelitian ini, penulis akan mengembangkan sistem prediksi solusi berbasis *web* yang dirancang untuk menawarkan sistem prediksi solusi bagi akademisi, peneliti, dan banyak orang. Alur sistem akan digambarkan menggunakan *flowchart* pada Gambar 2



Gambar 2. Alur Sistem

Berikut adalah tahapan alur kerja sistem :

a. Mulai

Pengguna terlebih dahulu mengakses halaman utama yang menampilkan proses kerja dari aplikasi sistem prediksi solusi.

b. *Input* Kalimat

Kemudian pengguna memasukkan kalimat permasalahan yang terdapat dalam artikel, setelah itu tekan tombol “Kirim” yang terdapat di tengah *input* teks.

c. Permasalahan diproses

Selanjutnya sistem akan mulai memproses permasalahan yang diberikan oleh user dengan mencocokkan data jawaban yang sudah dilatih.

d. *Output* Prediksi

Setelah permasalahan diproses, model akan memberikan *output* prediksi, yaitu kalimat solusi yang relevan dengan masalah yang diberikan. Solusi ini dihasilkan berdasarkan pola-pola yang telah dipelajari oleh model selama proses *fine-tuning*.

3. HASIL DAN ANALISA

3.1 Hasil *Training*

Hasil *training* menunjukkan nilai *loss* mengalami penurunan yang cukup signifikan pada setiap *epoch*. Pada *epoch* pertama nilai rata-rata kerugiannya masih tergolong tinggi yaitu 0,4394. Namun seiring bertambahnya jumlah *epoch*, nilai *loss* terus menurun hingga mencapai 0,0286 pada *epoch* ke-5. Penurunan ini menunjukkan bahwa model semakin terlatih dan mampu menyesuaikan parameter untuk mengurangi kesalahan dalam melakukan prediksi. Berikut tabel dari hasil *fine-tuning*.

Tabel 1. Hasil *Training*

<i>Epoch</i>	<i>Average Loss</i>	<i>Best Model Saved (Loss)</i>
1	0.4394	0.4394
2	0.1860	0.1860
3	0.0810	0.0810
4	0.0457	0.0457
5	0.0286	0.0286

3.2 Hasil Evaluasi

Untuk memastikan bahwa model yang digunakan mampu memberikan prediksi yang tepat, maka dilakukan evaluasi menggunakan *ROUGEscore*. Berikut akan ditampilkan tabel perbandingan antara pernyataan prediksi, dan juga tabel hasil evaluasi menggunakan *ROUGEscore*.

Tabel 2. Hasil Evaluasi

<i>Input</i>	Hasil yang diharapkan	<i>Output</i>	Kesimpulan
Pedagang keliling saat ini mengalami kesulitan mendapatkan pelanggan atau pembeli akibat persaingan dengan penyedia jasa pemesanan makanan secara online	Mereka dapat memanfaatkan media sosial dan platform online untuk mempromosikan produk mereka serta menawarkan layanan pesan antar yang lebih fleksibel	Mereka dapat memanfaatkan media sosial dan platform online untuk mempromosikan produk mereka serta menawarkan layanan pesan antar yang lebih fleksibel.	Hasil Relevan
Dalam menghadapi tantangan dari era digital dan konektivitas yang cepat, Pusdatik harus memiliki kesiapan untuk mengatasi gangguan layanan dan operasional yang	Salah satu langkah untuk meminimalkan dampak dari gangguan tersebut dengan melalui penyusunan Disaster Recovery Plan (DRP)	Puspatik perlu berinovasi untuk mendukung pelaksanaan proses bisnis di tengah-tengah lingkungan yang cepitubah dengan adanya teknologi baru.	Hasil Tidak Relevan

<i>Input</i>	Hasil yang diharapkan	<i>Output</i>	Kesimpulan
disebabkan oleh bencana dan ancaman.			
Akurasi model prediksi burnout hanya mencapai 65% pada tahap awal, menunjukkan bahwa beberapa fitur tidak memberikan kontribusi signifikan	Fitur yang tidak relevan menambah noise dalam data dan mengurangi efektivitas model, sehingga feature selection harus diterapkan untuk meningkatkan kinerja model secara optimal	Fitur yang kurang berperan dalam prediksi burnout membuka peluang untuk mengintegrasikan teknik pengenalan pola secara lebih aman dan efisien dengan <i>dataset</i> yang lebih komprehensif guna meningkatkan akurasi pergerakan tinggi	Hasil Tidak relevan

Dapat diamati dalam tabel diatas bahwa 3 *sample* kali hasil evaluasi yang sudah dilakukan, terdapat satu hasil yang memprediksi secara relevan dengan *input* yang dimasukkan, namun dua hasil evaluasi lainnya menghasilkan prediksi solusi yang kurang relevan dengan *input* yang diberikan oleh pengguna. Pengujian tersebut menginputkan 3 kalimat masalah dari 20 data evaluasi yang termasuk dalam data model yang sudah dilatih, sehingga dapat disimpulkan bahwa setiap kalimat masalah yang di *inputkan* belum tentu menghasilkan prediksi solusi yang *relevan*.

Tabel 3. Hasil Evaluasi ROUGE

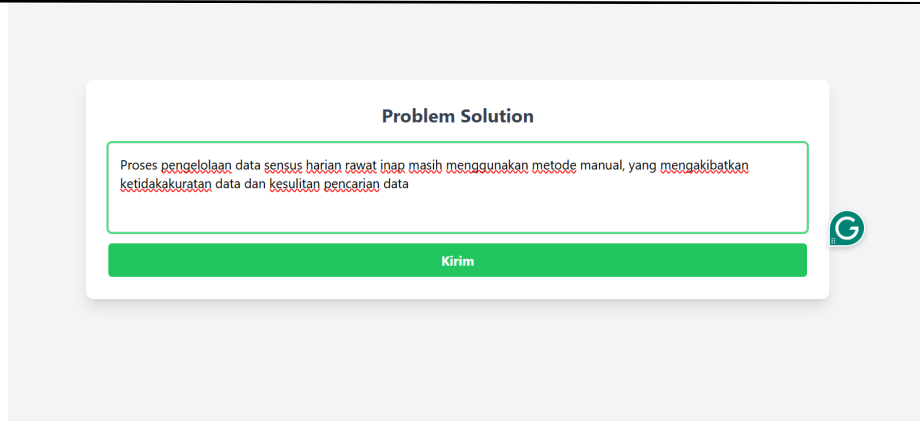
ROUGE-1	ROUGE-2	ROUGE-L
0.611	0.496	0.576

Berdasarkan hasil evaluasi menggunakan ROUGE terlihat model mempunyai tingkat kesesuaian yang cukup baik dengan acuan. Nilai ROUGE-1 yang diperoleh sebesar 12,236 dengan rata-rata sebesar 0,611 yang menunjukkan bahwa sekitar 61,1% unigram pada hasil prediksi telah sesuai dengan acuan. Sedangkan ROUGE-2 memiliki total skor 9.924 dengan rata-rata 0.496 yang berarti sekitar 49.6% bigram yang diprediksi sesuai dengan referensi. Nilai ini memang lebih rendah dibandingkan ROUGE-1 karena konformitas bigram lebih sulit dicapai. Sedangkan untuk ROUGE-L, model memperoleh skor total 11,525 dengan rata-rata 0,576 yang menunjukkan bahwa urutan kata dalam prediksi memiliki kesesuaian sebesar 57,6% dengan referensi. Dari hasil tersebut dapat disimpulkan bahwa model mampu menghasilkan prediksi yang cukup relevan, meskipun masih ada ruang untuk perbaikan terutama dalam memahami hubungan antar kata yang lebih kompleks.

3.3 Implementasi Sistem

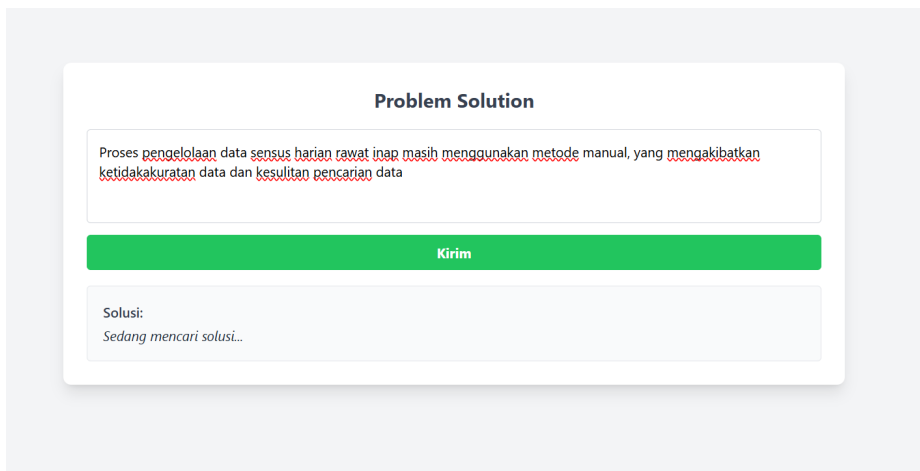
Setelah model berhasil diunduh, langkah selanjutnya adalah membuat Aplikasi agar lebih mudah digunakan untuk mendapatkan prediksi solusi dari masalah yang dimasukkan. Dalam proyek ini, saya menggunakan *Flask*, kerangka kerja berbasis *Python* yang ringan dan fleksibel, untuk membangun aplikasi *web* yang dapat berinteraksi langsung dengan model yang telah saya sesuaikan.

Aplikasi *web* ini dirancang agar pengguna cukup memasukkan kalimat permasalahan pada kolom masukan, kemudian sistem akan memprosesnya dan menampilkan hasil prediksi solusi pada halaman keluaran. Dengan memanfaatkan *Flask*, aplikasi ini dapat dijalankan secara lokal atau diunggah ke *server* untuk memudahkan akses pengguna lain.



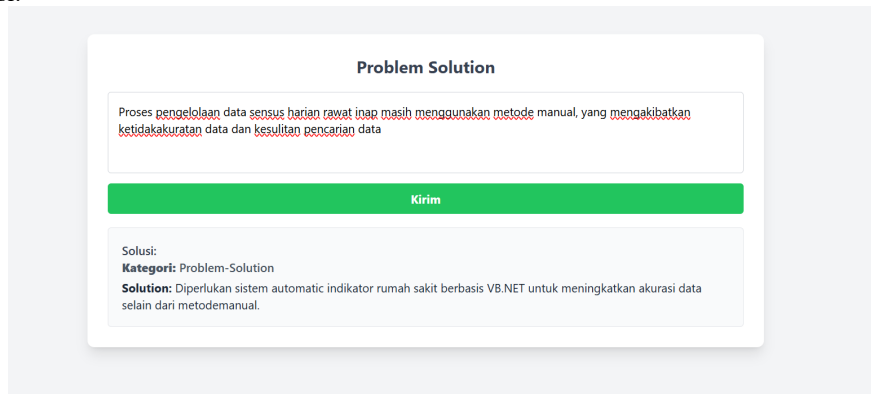
The screenshot shows a white card titled "Problem Solution" on a light gray background. Inside the card, there is a text input field containing the text: "Proses pengelolaan data sensus harian rawat inap masih menggunakan metode manual, yang mengakibatkan ketidakakuratan data dan kesulitan pencarian data". Below the input field is a green button labeled "Kirim". To the right of the input field, there is a small green circular icon with a white 'G' inside.

Gambar 3. Tampilan saat *Input* kalimat masalah
Gambar 3 menampilkan saat pengguna *input* kalimat masalah untuk mendapatkan prediksi.



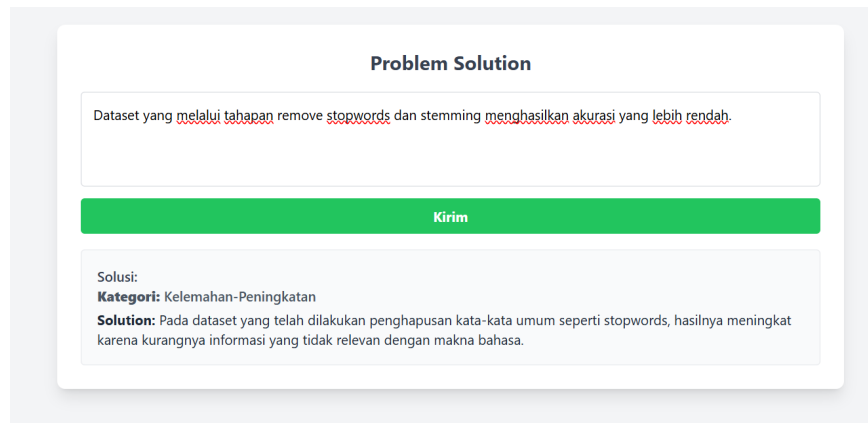
The screenshot shows the same "Problem Solution" card. The text input field now contains a loading message: "Solusi: Sedang mencari solusi...". The green "Kirim" button is still visible above the message.

Gambar 4. Tampilan saat *generate* model
Gambar 4 menampilkan saat sistem *generate* model yang *diinput* oleh pengguna untuk mendapatkan prediksi solusi.



The screenshot shows the "Problem Solution" card with the final prediction result. The text input field contains the same problem text as in Gambar 3. Below the "Kirim" button, the solution is displayed: "Solusi: Kategori: Problem-Solution Solution: Diperlukan sistem automatic indikator rumah sakit berbasis VB.NET untuk meningkatkan akurasi data selain dari metodemanual."

Gambar 5. Tampilan hasil prediksi yang *relevan*
Hasil prediksi dapat dilihat pada Gambar 5 dari *problem* yang telah *diinput* oleh pengguna, sistem memberikan prediksi solusi yang dinilai *relevan* dengan *input* masalah yang membahas Teknologi. Sehingga, dari percobaan demo aplikasi, sistem tersebut bekerja dalam memproses dan memberikan prediksi solusi yang *relevan* dengan *input* yang dilakukan oleh pengguna.



Gambar 6. Tampilan hasil prediksi yang tidak *relevan*

Hasil prediksi dapat dilihat pada Gambar 6 dari *problem* yang telah diinput oleh pengguna. namun, sistem memberikan prediksi solusi yang kurang *relevan* dengan input masalah yang membahas efek penghapusan *stopwords* dan *stemming*.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan *fine-tuning* model LLaMA-3.2 1B Instruct dalam memprediksi solusi kalimat permasalahan pada artikel ilmiah memberikan hasil yang cukup baik. Evaluasi menggunakan metrik ROUGE menghasilkan skor ROUGE-1 sebesar 0,611, ROUGE-2 sebesar 0,496, dan ROUGE-L sebesar 0,576 yang menunjukkan kesesuaian model dalam menghasilkan solusi yang tepat. Selama proses pelatihan, nilai *loss* mengalami penurunan yang cukup signifikan yaitu dari 0.4394 pada epoch pertama menjadi 0.0286 pada *epoch* kelima yang menunjukkan bahwa model semakin memahami pola yang ada pada data.

Namun, model masih menghadapi tantangan dalam memprediksi solusi terhadap beberapa masalah, terutama jika masalah tersebut berada di luar cakupan data pelatihan. Keterbatasan data yaitu 99 pasangan kalimat menjadi salah satu faktor yang mempengaruhi kinerja model. Selain itu, keterbatasan spesifikasi perangkat juga berdampak pada kecepatan dan akurasi prediksi. Untuk meningkatkan performa model perlu dilakukan optimasi agar lebih efisien atau menggunakan perangkat dengan spesifikasi lebih tinggi.

DAFTAR PUSTAKA

- [1] D. Fatmalasari dan F. R. Lumbanraja, "Peringkasan Teks Artikel Ilmiah Berbahasa Indonesia dengan Metode Pembobotan Kalimat," *J. Pepadun*, vol. 3, no. 3, hal. 314–322, 2022.
- [2] I. P. A. Peña dan J. C. Ortega-Castro, "Implementation and Evaluation of an Anti-Fraud Prototype Based on Generative Artificial Intelligence for the Ecuadorian Financial Sector," *Rev. Gestão Soc. e Ambient.*, vol. 18, no. 9, hal. e08601, 2024.
- [3] A. G. Pradana, D. R. I. M. Setiadi, dan A. R. Muslikh, "Fine tuning model Convolutional Neural Network EfficientNet-B4 dengan augmentasi data untuk klasifikasi penyakit kakao," *J. Inf. Syst. Appl. Dev.*, vol. 2, no. 1, hal. 01–11, 2024.
- [4] S. R. Y. Salim, dan M. Hasnawi, "Konversi Bahasa Indonesia ke Perintah Data Manipulation Language pada Structured Query Language menggunakan Natural Language Processing," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 3, no. 3, hal. 181–187, 2022.
- [5] M. R. J. K. VM, H. Warriar, dan Y. Gupta, "Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations," 2024.
- [6] M. Mosbach, T. Pimentel, S. Ravfogel, D. Klakow, dan Y. Elazar, "Few-shot Fine-tuning vs. In-context Learning: A Fair Comparison and Evaluation," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, no. 2022, hal. 12284–12314, 2023.
- [7] S. Chen, H. Li, Z. Zhang, Z. Sun, dan Y. Cheng, "OMAI: A Specialized Large Language Model for Operational Maintenance in Institute of High Energy Physics," *Proc. Sci.*, vol. 458, hal. 24–29, 2024.
- [8] A. Agung, A. Daniswara, I. Kadek, dan D. Nuryana, "Data Preprocessing Pola Pada Penilaian Mahasiswa Program Profesi Guru," *J. Informatics Comput. Sci.*, vol. 05, hal. 97–100, 2023.
- [9] R. R. Salam, M. F. Jamil, Y. Ibrahim, R. Rahmadden, S. Soni, dan H. Herianto, "Analisis Sentimen Terhadap Bantuan Langsung Tunai (BLT) Bahan Bakar Minyak (BBM) Menggunakan Support Vector Machine," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, hal. 27–35, 2023.
- [10] A. Susilo, V. Christanti, dan M. D. Lauro, "Penerjemah Bahasa Gaul menggunakan LoRA dan QLoRA Fine-Tuning LLaMA-2-Chat untuk ChatBot," vol. 9, no. 2, hal. 248–260, 2024.