Techno.Com, Vol. 25 No. 3, August 2026: 972 - 981

Implementasi Arsitektur Transformer GPT-2 untuk Prediksi Kata Selanjutnya pada Teks Resep Masakan Indonesia

Implementation of Transformer GPT-2 Architecture for Next Word Prediction in Indonesian Cooking Recipe Texts

Ahmad Farhan Ramadhansyah*[1] , Jeny Fajarwati[2] , Rezky Aulia[3], Adha Mashur Sajiah[4]
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kota Kendari, Indonesia
E-mail : ramadhansyahfarhan96@gmail.com,[1], jenyfajarwati@gmail.com[2] , rezkyaulia.1275@gmail.com[3], adha.m.sajiah@uho.ac.id[4]*
**Corresponding author*
Received 4 July 2026; Revised 4 August 2026; Accepted 14 August 2026

Abstrak - Pemrosesan bahasa alami untuk domain spesifik memerlukan model yang mampu memahami struktur kalimat yang unik dan kosakata informal. Penelitian ini mengimplementasikan arsitektur Transformer bergaya GPT-2 yang dibangun dari awal untuk tugas prediksi kata selanjutnya (*next-word prediction*) pada teks resep masakan Indonesia. Dataset yang digunakan terdiri dari 14.945 resep masakan, yang ditokenisasi menggunakan metode *SentencePiece Byte Pair Encoding* (BPE) dengan *vocabulary size* sebanyak 8.000 token. Model *causal language* yang diusulkan memiliki 4 blok transformer, 4 *attention heads*, dan dimensi embedding sebesar 256, dengan total sekitar 5,26 juta parameter yang dapat dilatih. Proses pelatihan dilakukan menggunakan *optimizer AdamW* selama 30 *epoch*. Evaluasi pada data Data uji (*test set*) menghasilkan nilai *cross-entropy loss* sebesar 2,8471 dan *perplexity* sebesar 17,24. Selain itu, model ini berhasil mencapai akurasi Top-1 sebesar 40,07% dan akurasi Top-5 sebesar 67,72%. Pengujian decoding interaktif menggunakan strategi *greedy*, top-k, dan *nucleus sampling* menunjukkan kemampuan model yang baik dalam menghasilkan instruksi langkah memasak yang relevan secara kontekstual. Hasil ini menunjukkan bahwa arsitektur GPT-2 berskala kecil sangat efektif untuk tugas pembuatan teks yang spesifik pada domain tertentu.

Kata kunci - Prediksi Kata Selanjutnya, GPT-2, Transformer, Byte Pair Encoding, Resep Masakan

Abstract *– Natural language processing for specific domains requires models capable of understanding unique sentence structures and informal vocabulary. This study implements a Transformer architecture in the style of GPT-2, built from scratch, for the task of next-word prediction in Indonesian cooking recipe texts. The dataset consists of 14,945 recipes, tokenized using the SentencePiece Byte Pair Encoding (BPE) method with a vocabulary size of 8,000 tokens. The proposed causal language model comprises 4 transformer blocks, 4 attention heads, and an embedding dimension of 256, totaling approximately 5.26 million trainable parameters. Training was conducted using the AdamW optimizer for 30 epochs. Evaluation on the test set yielded a cross-entropy loss of 2.8471 and a perplexity of 17.24. In addition, the model achieved a Top-1 accuracy of 40.07% and a Top-5 accuracy of 67.72%. Interactive decoding tests using greedy, top-k, and nucleus sampling strategies demonstrated the model's strong ability to generate contextually relevant cooking instructions. These results indicate that a small-scale GPT-2 architecture is highly effective for domain-specific text generation tasks.*

Keywords *– Next-Word Prediction, GPT-2, Transformer, Byte Pair Encoding, Cooking Recipes*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

1. PENDAHULUAN

Perkembangan teknologi *Natural Language Processing* (NLP) telah membawa perubahan signifikan dalam cara manusia berinteraksi dengan sistem komputasi, khususnya melalui implementasi pemodelan *Text Generation* (pembuatan teks otomatis). Kemampuan mesin untuk menghasilkan teks yang koheren dan relevan secara kontekstual kini mulai diadaptasi secara luas ke dalam berbagai domain spesifik, termasuk dalam industri dan literatur kuliner lokal [1]. Dalam konteks ini, pemodelan bahasa tingkat lanjut dapat digunakan untuk mendokumentasikan, menyusun, dan memprediksi instruksi langkah-langkah memasak secara otomatis, sehingga memfasilitasi interaksi pengguna dengan basis data resep yang masif.

Terobosan terbesar dalam NLP modern dimulai dengan diperkenalkannya arsitektur *Transformer* yang mengandalkan mekanisme *self-attention* tanpa menggunakan komputasi rekuren yang lambat [2]. Berdasarkan arsitektur fundamental tersebut, *Generative Pre-trained Transformer* (GPT-2) muncul sebagai salah satu model *causal language* yang paling andal dalam mengeksekusi tugas memprediksi kata selanjutnya berdasarkan sejarah konteks [3]. Beberapa penelitian sebelumnya telah sukses membuktikan efektivitas GPT-2 pada teks bahasa Indonesia, seperti pada pembuatan puisi tradisional atau pantun secara otomatis, yang menunjukkan bahwa model *Transformer* mampu mempelajari struktur dan pola bahasa lokal dengan sangat akurat [4].

Beberapa penelitian terkini telah menyoroti pentingnya lokalisasi dan efisiensi pemodelan bahasa untuk pemrosesan teks pada domain spesifik [5], [6]. Studi-studi tersebut menunjukkan bahwa melatih model bahasa berparameter kecil (*Small Language Models*) secara khusus pada korpus lokal dapat memberikan efisiensi komputasi sekaligus mempertahankan akurasi kontekstual [7], [8]. Lebih lanjut, eksplorasi arsitektur Transformer untuk tugas pembuatan teks instruksional dan ekstraksi informasi pada literatur kuliner juga telah terbukti sangat menjanjikan [9], [10].

Meskipun model bahasa skala besar (LLM) berbasis cloud saat ini memiliki kemampuan generalisasi yang tinggi, model-model tersebut sering kali mengalami penurunan performa, inkonsistensi, atau halusinasi ketika dihadapkan pada bahasa domain spesifik yang sangat terlokalisasi seperti resep masakan Indonesia. Teks resep memiliki hierarki struktur yang unik (terdiri dari judul, daftar bahan, dan langkah-langkah) serta sarat dengan kosakata informal, istilah serapan, angka terikat, dan singkatan khusus (seperti "sdm", "siung", "secukupnya") [11], yang mana sering kali menjadi tantangan dalam pemrosesan bahasa alami [12], [13]. Selain itu, LLM konvensional membutuhkan sumber daya infrastruktur komputasi yang masif untuk dijalankan, menjadikannya kurang efisien jika diimplementasikan pada ekosistem rekayasa perangkat lunak (*software engineering*) berskala kecil atau aplikasi dengan memori terbatas [14].

Untuk mengatasi kesenjangan tersebut, penelitian ini mengusulkan implementasi arsitektur *Transformer* GPT-2 berskala kecil (*small-scale*) yang dibangun dari awal (*from scratch*) khusus untuk domain prediksi teks resep masakan Indonesia. Model ini dilatih secara mandiri menggunakan korpus data yang divalidasi, terdiri dari 14.945 resep masakan Nusantara. Dengan konfigurasi dimensi *embedding* sebesar 256, 4 *attention heads*, dan 4 *transformer blocks*, model ini didesain agar sangat ringan, hanya memiliki sekitar 5,26 juta parameter yang dapat dilatih—namun tetap mempertahankan tingkat efisiensi yang tinggi dalam menghasilkan teks instruksi kuliner yang logis secara algoritmik.

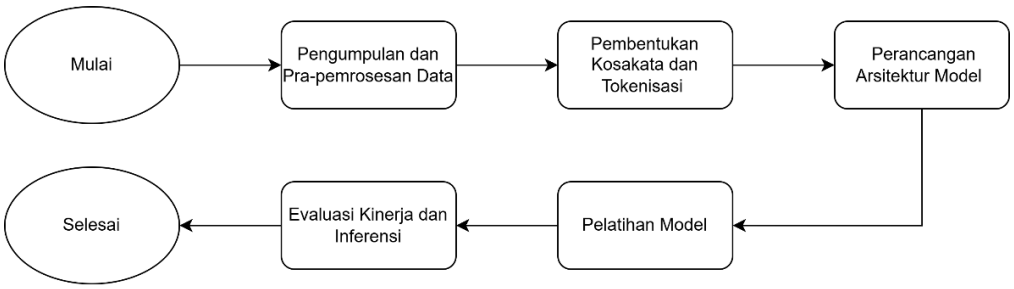
Salah satu kontribusi utama dalam penelitian ini adalah optimasi pada fase pra-pemrosesan melalui pendekatan tokenisasi *SentencePiece Byte Pair Encoding* (BPE) dengan membatasi ukuran kosakata pada 8.000 token [15]. Metode BPE ini terbukti sangat optimal dalam menangani morfologi bahasa Indonesia yang kaya akan afiksasi(imbuhan) serta mampu menekan masalah kata *Out-of-Vocabulary* (OOV) tanpa harus merusak konteks istilah memasak [15]. Melalui implementasi komprehensif ini, penelitian bertujuan untuk mengevaluasi kemampuan model *causal* berskala kecil melalui metrik *cross-entropy loss* dan *perplexity*, serta membuktikan

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

kelayakan implementasi strategi *decoding* dinamis (*greedy, top-k,* dan *nucleus sampling*) untuk menghasilkan kalimat resep yang natural.

2. METODE PENELITIAN

Penelitian ini dilakukan melalui serangkaian tahapan komputasional yang sistematis untuk membangun dan melatih model *causal language* secara mandiri (*from scratch*). Alur kerja utama dalam metodologi ini dibagi menjadi lima tahapan inti, yaitu: (1) pengumpulan, pra-pemrosesan, dan pembagian dataset teks resep; (2) pembentukan kosakata dan pembuatan sekuens menggunakan tokenisasi *SentencePiece Byte Pair Encoding* (BPE); (3) perancangan arsitektur *Transformer* bergaya GPT-2; (4) pelaksanaan skenario pelatihan (*training*) model untuk tugas *next-word prediction*; serta (5) evaluasi kinerja kuantitatif dan pengujian inferensi generasi teks. Seluruh tahapan ini dirancang secara berurutan agar model dapat mengekstraksi karakteristik leksikal dan sintaksis dari instruksi kuliner berbahasa Indonesia secara optimal, hingga akhirnya mampu menghasilkan teks yang koheren.



Gambar 1. Alur tahapan penelitian

2.1. Pengumpulan dan Pra-pemrosesan Data

Data yang digunakan dalam penelitian ini bersumber dari dataset resep masakan Indonesia yang diperoleh melalui platform repositori Kaggle. Korpus data ini mencakup total 14.945 baris teks resep masakan Nusantara yang memuat informasi terstruktur berupa judul resep, daftar bahan-bahan, serta tahapan instruksi pembuatan. Ketiga bagian informasi tersebut kemudian diekstraksi dan digabungkan menjadi satu kesatuan teks untuk setiap resep, dengan memisahkan antar bagian menggunakan tanda baris baru agar model dapat mempelajari struktur hierarki resep secara alami. Mengingat dokumen resep masakan yang dikumpulkan dari internet sering kali ditulis dalam format yang tidak baku oleh pengguna, maka tahapan *text preprocessing* menjadi tahapan krusial untuk membuang elemen non-kontekstual sebelum teks tersebut dialirkan ke dalam model Transformer.

Proses pembersihan teks dilakukan melalui beberapa tahapan sistematis secara berurutan. Pertama, seluruh teks diubah menjadi huruf kecil untuk menyamakan representasi kata terlepas dari posisi kapitalisasinya. Kedua, dilakukan penghapusan nomor urut penulisan langkah-langkah memasak, seperti "1)", "2)", dan seterusnya, serta penyaringan dan pembuangan *seluruh Uniform Resource Locator* (URL) yang tidak memiliki korelasi dengan instruksi kuliner. Ketiga, pembersihan karakter khusus diterapkan dengan hanya mempertahankan huruf alfabet, angka, spasi, serta tanda baca dasar (titik, koma, tanda seru, dan tanda tanya).

Keempat, dilakukan penghapusan *noise* untuk mengeliminasi angka yang mengaburkan pola probabilitas kata selanjutnya. Pendekatan ini diterapkan sedemikian rupa sehingga angka yang menempel pada sebuah kata seperti takaran ukur "200ml" atau "150g" tetap dipertahankan agar tidak merusak konteks instruksi bahan. Terakhir, dataset yang telah bersih dan divalidasi dibagi menjadi tiga bagian, yaitu 80% data latih, 10% data validasi, dan 10% data uji. Contoh perbandingan bentuk teks resep masakan sebelum dan setelah melewati tahapan pra-pemrosesan disajikan pada Tabel 1.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

Tabel 1. Perbandingan Teks Resep Sebelum dan Setelah Pra-pemrosesan

Teks Sebelum Pembersihan	Teks Setelah Pembersihan
1. Panaskan minyak sebanyak 2 sdm di atas wajan.	panaskan minyak sebanyak sdm di atas wajan.
2. Cuci bersih daging ayam, lihat tips di https://resepkuliner.id	cuci bersih daging ayam lihat tips di
3. Tumis bumbu halus hingga matang dan harum.	tumis bumbu halus hingga matang dan harum.

2.2. Pembentukan Kosakata dan Tokenisasi

Setelah teks melalui tahap pra-pemrosesan, tahapan selanjutnya adalah mengonversi teks mentah menjadi representasi numerik yang dapat diproses oleh model. Penelitian ini menggunakan metode tokenisasi *SentencePiece Byte Pair Encoding* (BPE) [6]. Metode BPE dipilih karena keunggulannya dalam menangani morfologi bahasa Indonesia yang kaya akan afiksasi seperti imbuhan me-, di-, -kan, serta kemampuannya secara alami dalam menangani kata-kata *Out-of-Vocabulary* (OOV) menjadi *subword*. Di samping itu, sebuah token pemisah khusus (</s> atau <|endoftext|>) disisipkan pada bagian akhir setiap resep sebagai sinyal bagi model untuk mengenali batas akhir dari satu dokumen teks masakan secara mandiri.

Pada proses pelatihannya, *vocabulary size* dibatasi sebanyak 8.000 token untuk menjaga efisiensi memori. Korpus teks yang telah ditokenisasi kemudian dipotong menjadi *sequences* menggunakan metode *sliding window* dengan *context length* sebesar 128 token. Secara spesifik, pada data latih, proses pemotongan ini menerapkan nilai *stride* sebesar 64 token untuk menciptakan segmen data yang *overlapping*, sehingga memaksimalkan penangkapan variasi konteks pembelajaran.

Sementara itu, untuk data validasi dan data uji, ukuran *stride* ditetapkan sebesar 128 token tanpa *overlap*. Format pemotongan ini menghasilkan segmen berukuran 129 token, di mana 128 token pertama berfungsi sebagai rentetan *input*, dan 128 token terakhir (bergeser satu posisi ke kanan) berfungsi sebagai target atau label. Melalui perancangan ini, setiap posisi ke-i pada rangkaian *input* ditugaskan untuk memprediksi token target pada posisi ke-(i+1), sesuai dengan prinsip komputasi dasar dari tugas *next word prediction*.

2.3. Perancangan Arsitektur Model

Model komputasi utama yang dibangun dalam penelitian ini adalah varian dari arsitektur Transformer *decoder-only* yang mengadopsi mekanisme spesifik dari *Generative Pre-trained Transformer* (GPT-2). Karakteristik utama dari arsitektur ini adalah penggunaan *Causal* (*Masked*) *Multi-Head Self-Attention*, di mana matriks atensi dimanipulasi agar pemrosesan probabilitas token ke-i murni hanya didasarkan pada konteks token-token sebelumnya, tanpa dapat melihat token masa depan. Berbeda dengan arsitektur Transformer orisinal yang menggunakan fungsi *sinusoidal* statis untuk menyimpan informasi urutan kalimat, model ini mengimplementasikan *Learned Positional Embedding*. Pendekatan ini memungkinkan *positional vectors* dilatih dan diperbarui secara dinamis menyesuaikan karakteristik bahasa domain target selama proses *training*.

Tabel 2. Spesifikasi Hyperparameter GPT-2

Parameter	Nilai / Deskripsi
Jumlah Layer (n_layer)	4 Blok Transformer
Attention Heads (n_head)	4 Heads
Dimensi Embedding (d_model)	256

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

Dimensi Feed-Forward (d_ff)	1.024
Panjang Konteks (context_len)	128 Token
Ukuran Kosakata (vocab_size)	8.000 Token
Dropout Rate	0.3
Total Parameter Trainable	5.268.992

Untuk memastikan model tetap ringan (*lightweight*) namun representatif, konfigurasi *hyperparameter* diatur pada skala kecil (*small config*). Model ini dibangun menggunakan 4 blok Transformer (n_layer = 4), 4 *attention heads* (n_head = 4), dimensi *embedding* sebesar 256 (d_model = 256), dan dimensi jaringan *feed-forward* sebesar 1.024 (d_ff = 1024). Sebagai tambahan, arsitektur ini menerapkan fungsi aktivasi GELU (*Gaussian Error Linear Unit*) yang terbukti lebih presisi dalam pemodelan bahasa, serta teknik regularisasi *Dropout* dengan rasio 0,3 untuk mencegah *overfitting*. Untuk mengoptimalkan arsitektur, diimplementasikan pula teknik *weight tying*, yaitu penggunaan bobot parameter yang sama antara lapisan *token embedding* awal dan *linear head* lapisan keluaran. Integrasi konfigurasi tersebut secara kolektif menghasilkan model kausal yang efisien, dengan total keseluruhan parameter sebesar 5.268.992 yang seluruhnya dapat dilatih *trainable*.

2.4. Pelatihan Model

Skenario pelatihan dijalankan menggunakan *optimizer AdamW*. Untuk mencegah masalah *overfitting* secara optimal, diterapkan strategi pemisahan *parameter splitting* pada pengaturan *weight decay*. Nilai *weight decay* sebesar 0,1 hanya diaplikasikan secara eksklusif pada *weights* yang memiliki dimensi tensor dua atau lebih (>= 2).

Sementara itu, parameter dengan dimensi tensor kurang dari dua, seperti *bias* dan bobot pada *Layer Normalization*, tidak dikenakan penalti *weight decay* (diatur pada nilai 0,0) guna menjaga stabilitas konvergensi model. Selain itu, *batch size* ditetapkan sebesar 64 observasi per iterasi, dan diterapkan teknik *gradient clipping* pada ambang batas 1,0 untuk menstabilkan pembaruan bobot serta mencegah fenomena *exploding gradient*.

Penyesuaian *learning rate* diatur secara dinamis menggunakan penjadwalan *Cosine Annealing* yang dikombinasikan dengan *Linear Warmup* selama 200 iterasi awal. Pendekatan ini memungkinkan *learning rate* untuk naik secara bertahap dari 0 hingga menyentuh puncaknya di 3×10^{-4}, lalu secara perlahan meluruh mengikuti kurva cosinus hingga nilai minimum 3×10^{-5} pada fase akhir pelatihan. Model dilatih dengan mengoptimasi *Cross-Entropy Loss* selama batas maksimal 30 *epoch*. Di samping itu, diterapkan pula mekanisme *early stopping* dengan *patience* sebanyak 5 *epoch* guna memantau dan menyimpan *best checkpoint* yang divalidasi dengan tingkat *loss validation* terendah.

2.5. Evaluasi Kinerja dan Inferensi

Tahapan akhir dalam metodologi penelitian ini adalah pengukuran performa model yang telah dilatih menggunakan data uji yang belum pernah dilihat oleh model sebelumnya. Evaluasi kinerja dilakukan melalui dua pendekatan, yaitu evaluasi kuantitatif dan inferensi kualitatif. Pada evaluasi kuantitatif, performa pemodelan diukur menggunakan empat metrik utama: *Cross-Entropy Loss*, *Perplexity* (PPL), Akurasi Top-1, dan Akurasi Top-5. Penghitungan *loss* dan akurasi dilakukan secara ketat dengan mengabaikan token *padding* (*masking PAD tokens*) agar evaluasi murni hanya menilai token instruksi bahasa. Akurasi Top-1 mengukur seberapa sering model menebak token target secara presisi pada urutan pertama dengan probabilitas mutlak tertinggi, sedangkan Akurasi Top-5 mengukur persentase keberhasilan token target berada di dalam himpunan lima tebakan teratas model.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

Selanjutnya, pengujian inferensi dilakukan secara kualitatif untuk mendemonstrasikan kapabilitas model dalam menghasilkan teks lanjutan dari sebuah *prompt* yang diberikan. Untuk melihat ragam struktur bahasa yang dihasilkan, eksperimen ini membandingkan tiga strategi penulisan teks (*decoding strategies*) secara interaktif. Pertama, *Greedy Search*, yaitu pendekatan deterministik yang selalu memilih kata dengan probabilitas tertinggi pada setiap langkah waktu. Kedua, *Top-K Sampling*, yaitu metode pengambilan sampel kata secara acak dari kumpulan *K* token dengan probabilitas tertinggi (ditetapkan $K = 10$). Ketiga, *Nucleus* atau *Top-p Sampling*, yaitu pengambilan sampel dinamis dari kumpulan token yang total akumulasi probabilitasnya mencapai ambang batas *p* (ditetapkan $p = 0{,}90$). Pada strategi *sampling* (Top-K dan *Nucleus*), *temperature* diatur pada 0,8 untuk menyeimbangkan antara tingkat kreativitas dan koherensi tata bahasa dalam teks resep yang dihasilkan.

3. HASIL DAN PEMBAHASAN

Pada bagian ini dipaparkan hasil evaluasi kinerja serta pembahasan mendalam mengenai model *causal language* berbasis arsitektur *Transformer* GPT-2 berskala kecil yang telah melalui proses pelatihan. Pembahasan hasil pengujian diuraikan ke dalam dua aspek utama, yaitu analisis kuantitatif dan analisis kualitatif. Analisis kuantitatif berfokus pada penilaian performa objektif model berdasarkan metrik evaluasi standar yang meliputi kurva konvergensi *Cross-Entropy Loss*, *Perplexity*, serta nilai akurasi prediksi pada pengujian *Top-1* dan *Top-5*. Sementara itu, analisis kualitatif dilakukan untuk menguji fungsionalitas model secara interaktif dalam melanjutkan teks masakan menggunakan tiga strategi *decoding* yang berbeda, yaitu *greedy search*, *top-k sampling*, dan *nucleus sampling*. Melalui kedua pendekatan evaluasi ini, efektivitas arsitektur model dalam memahami karakteristik leksikal teks kuliner Nusantara dapat diukur secara komprehensif.

3.1. Hasil Evaluasi Kinerja (Metrik Kuantitatif)

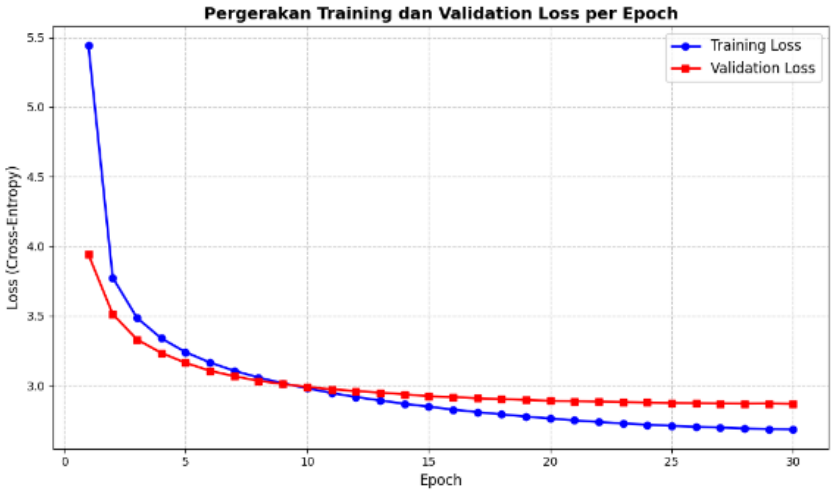
Evaluasi kinerja model dilakukan secara komprehensif setelah proses pelatihan diselesaikan pada *epoch* 30. Penilaian performa kuantitatif ini didasarkan pada empat parameter pengujian utama yang diterapkan pada data uji , yaitu *Cross-Entropy Loss*, *Perplexity*, Akurasi *Top-1*, dan Akurasi *Top-5*. Rangkuman dari keseluruhan hasil evaluasi tersebut disajikan pada Tabel 3.

Tabel 3. Rangkuman Metrik Evaluasi Model (Epoch 30)

Metrik Evaluasi	**Nilai Akhir**
Cross-Entropy Loss	2,8471
Perplexity	17,24
Akurasi Top-1	40,07%
Akurasi Top-5	67,72%

Berdasarkan hasil pengujian, model berhasil mencapai nilai *Cross-Entropy Loss* sebesar 2,8471. *Loss value* ini merepresentasikan seberapa jauh penyimpangan antara distribusi probabilitas prediksi model dengan label target yang sebenarnya. Seperti yang diilustrasikan pada Gambar 1, kurva *loss* menunjukkan tren penurunan yang stabil sejak iterasi pertama hingga *epoch* ke-30. Hal ini mengindikasikan bahwa model mampu belajar dan melakukan konvergensi dengan baik tanpa mengalami indikasi *overfitting* yang drastis.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163



Gambar 2. Kurva Konvergensi Loss selama Proses Pelatihan

Lebih lanjut, performa model dikonfirmasi melalui nilai *Perplexity* yang menyentuh angka 17,24. *Perplexity* merupakan fungsi eksponensial dari *cross-entropy loss* yang mengukur tingkat "kebingungan" model; semakin rendah nilainya, semakin presisi model dalam menebak kata selanjutnya. Nilai 17,24 tergolong sangat kompetitif dan efisien untuk sebuah model *Transformer* berskala kecil yang dibangun *from scratch*.

Dalam aspek keakuratan prediksi, kemampuan model dievaluasi menggunakan metrik akurasi terhadap total 208.896 token validasi pada data uji. Tingginya jumlah token ini terjadi karena penerapan metode tokenisasi *SentencePiece Byte Pair Encoding* (BPE). Meskipun data uji yang digunakan hanya sebanyak 1.495 baris resep dengan rata-rata 113 kata per resep, BPE secara alami memecah kata-kata berafiksasi dalam bahasa Indonesia menjadi beberapa sub-kata (*subwords*). Pemecahan ini membuat rata-rata panjang teks meningkat menjadi sekitar 139,7 token per baris resep, sehingga total keseluruhan mencapai 208.896 token validasi. Model mencatatkan Akurasi Top-1 sebesar 40,07%, yang berarti jika model diinstruksikan untuk memprediksi satu token mutlak dengan probabilitas tertinggi, tebakan tersebut benar secara presisi pada 83.705 token dari total keseluruhan pengujian.

3.2. Pengujian Prediksi Kata Interaktif (*Decoding Strategies*)

Selain evaluasi berdasarkan metrik kuantitatif, kapabilitas model juga divalidasi secara kualitatif melalui pengujian prediksi kata interaktif. Pengujian ini bertujuan untuk mengukur sejauh mana model dapat melanjutkan sebuah kalimat awalan (*prompt*) menjadi instruksi resep masakan yang koheren, relevan secara kontekstual, dan memiliki tata bahasa yang natural. Untuk menghasilkan variasi teks keluaran, eksperimen ini menerapkan tiga strategi *decoding* yang berbeda, yaitu *Greedy Search*, *Top-K Sampling*, dan *Nucleus (Top-p) Sampling*.

Pendekatan pertama, *Greedy Search*, beroperasi dengan cara selalu memilih token kata yang memiliki nilai probabilitas tertinggi pada setiap langkah waktu (*time step*). Meskipun metode ini secara matematis paling aman, hasil prediksinya sering kali mengalami masalah repetisi atau pengulangan frasa yang sama secara terus-menerus, sehingga teks kehilangan makna pragmatisnya. Pendekatan kedua, *Top-K Sampling*, mengatasi masalah repetisi ini dengan cara mendistribusikan probabilitas pada *K* token teratas (misalnya 50 token dengan probabilitas tertinggi) dan memilih kata berikutnya secara acak dari kumpulan tersebut. Metode ini memberikan hasil yang lebih variatif, namun terkadang dapat memunculkan kata yang sedikit keluar dari konteks instruksi memasak.

Pendekatan ketiga adalah *Nucleus Sampling* atau *Top-p Sampling*. Alih-alih menggunakan jumlah *K* yang statis, metode ini secara dinamis memilih token dari himpunan kata yang total probabilitas kumulatifnya melebihi ambang batas *p* (misalnya 0,90). Berdasarkan hasil

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

eksperimen, *Nucleus Sampling* terbukti menjadi metode yang paling optimal. Strategi ini berhasil menyeimbangkan antara koherensi kalimat dan keragaman kosakata, sehingga teks resep yang dihasilkan terdengar paling masuk akal, mengalir secara natural, dan terhindar dari siklus repetisi. Perbandingan nyata dari teks yang dihasilkan oleh ketiga strategi *decoding* tersebut terhadap kalimat pancingan yang sama dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan Hasil Generasi Teks berdasarkan Strategi Decoding

Strategi Decoding	**Teks Awalan (Prompt)**	**Hasil Teks Lanjutan (Generated Text)**
Greedy Search	panaskan minyak, lalu tumis	bawang merah dan bawang putih hingga harum. lalu masukkan bawang merah dan bawang putih hingga matang. lalu masukkan bawang merah... (mengalami repetisi)
Top-K Sampling	panaskan minyak, lalu tumis	bawang putih, irisan cabai, dan potongan tomat hijau. aduk perlahan sampai sedikit layu dan angkat.
Nucleus (Top-p) Sampling	panaskan minyak, lalu tumis	bumbu halus hingga harum dan matang. masukkan potongan daging ayam, aduk hingga berubah warna lalu tuangkan sedikit air.

Melalui pembuktian pada Tabel 4, dapat disimpulkan bahwa arsitektur GPT-2 skala kecil yang dibangun pada penelitian ini tidak hanya berhasil mengonvergensi nilai *loss* secara matematis, tetapi juga berfungsi dengan sangat baik secara aplikatif sebagai sistem generator teks instruksi kuliner berbahasa Indonesia.

3.3. Perbandingan Kualitatif dengan Model LLM Umum *(Zero-Shot*)

Untuk memvalidasi klaim bahwa arsitektur GPT-2 skala kecil (5,26 juta parameter) lebih efisien dan tangguh dalam mengenali struktur bahasa resep lokal dibandingkan model bahasa skala besar (LLM) pada umumnya, dilakukan pengujian komparatif secara kualitatif. Pengujian ini membandingkan keluaran model usulan dengan model GPT-3.5 menggunakan metode *zero-shot prompting*. Sebagai representasi model usulan, diterapkan strategi *Nucleus Sampling* yang telah terbukti memberikan hasil paling optimal pada tahap pengujian sebelumnya. Kedua model diberikan *prompt* yang identik, yaitu *"panaskan minyak, lalu tumis"*.

Tabel 5. Perbandingan Kualitatif Hasil Generasi Teks Model Usulan dan GPT-3.5

Model	**Teks Awalan (*Prompt*)**	**Hasil Teks Lanjutan (*Generated Text*)**
Generasi *GPT-3.5*	panaskan minyak, lalu tumis	panaskan minyak, lalu tumis bawang merah dan bawang putih hingga harum.
	panaskan minyak, lalu tumis	bumbu halus hingga harum dan matang. masukkan potongan daging ayam, aduk

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

Model Usulan (*Nucleus Sampling*)		hingga berubah warna lalu tuangkan sedikit air.

Berdasarkan perbandingan pada tabel di atas, model GPT-2 usulan mampu secara deterministik langsung menghasilkan kalimat lanjutan yang padat dan sangat relevan dengan format penulisan resep masakan Nusantara. Sebaliknya, GPT-3.5 cenderung mengulang kembali teks awalan secara penuh menjadi satu kalimat standar, alih-alih langsung melanjutkan potongan instruksi berikutnya. Hal ini mengonfirmasi bahwa pemodelan berskala kecil yang dilatih khusus (*from scratch*) pada korpus domain target terbukti lebih efisien, presisi, dan siap diimplementasikan secara berkesinambungan untuk tugas pembuatan teks spesifik.

4. KESIMPULAN

Berdasarkan keseluruhan tahapan perancangan, pelatihan, dan evaluasi yang telah dilakukan, dapat disimpulkan bahwa arsitektur *Transformer* bergaya GPT-2 berskala kecil berhasil diimplementasikan secara efektif untuk tugas memprediksi kata selanjutnya (*next-word prediction*) pada domain teks resep masakan Indonesia. Penggunaan metode tokenisasi *Byte Pair Encoding* (BPE) dengan batasan 8.000 kosakata terbukti optimal dalam merepresentasikan karakteristik morfologi bahasa Nusantara serta berbagai istilah informal pada instruksi kuliner. Model yang beroperasi dengan bobot parameter efisien sebesar 5,26 juta ini mampu melakukan konvergensi dengan baik selama 30 *epoch*, mencatatkan performa evaluasi yang kompetitif dengan nilai *cross-entropy loss* 2,8471 dan *perplexity* sebesar 17,24. Dari segi akurasi, model mampu mencapai Akurasi *Top-1* sebesar 40,07% dan Akurasi *Top-5* sebesar 67,72%.

Lebih lanjut, pengujian fungsionalitas model secara interaktif membuktikan bahwa pemilihan strategi *decoding* sangat menentukan kualitas teks yang dihasilkan. Dibandingkan dengan *Greedy Search* yang rentan terhadap repetisi dan *Top-K Sampling* yang kurang terkalibrasi secara dinamis, metode *Nucleus (Top-p) Sampling* terbukti sebagai pendekatan yang paling ideal. Pendekatan ini secara konsisten menghasilkan lanjutan instruksi memasak yang mengalir secara natural, logis secara tata bahasa, dan relevan dengan konteks kalimat pancingan. Secara keseluruhan, penelitian ini menunjukkan bahwa pemodelan bahasa *causal* yang ringan (*lightweight*) sangat mumpuni untuk diaplikasikan pada ekosistem rekayasa perangkat lunak sebagai asisten penulis teks kuliner berbasis kecerdasan buatan.

DAFTAR PUSTAKA

[1] Y. Yanfi, Y. Heryadi, L. Lukas, W. Suparta, dan Y. Arifin, "Sentiment Analysis of User Review on Indonesian Food and Beverage Group using Machine Learning Techniques," *2022 IEEE Creative Communication and Innovative Technology (ICCIT)*, 2022, pp. 1-5.

[2] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998-6008, 2017.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, dan I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[4] F. N. Andita, A. F. Hidayatullah, dan S. Sumpeno, "Poetry Generation for Indonesian Pantun Using SeqGAN and GPT-2," *Jurnal Ilmu Komputer dan Informasi*, vol. 16, no. 1, pp. 59-67, 2023.

[5] S. Cahyawijaya *et al.*, "IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 8875-8898.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163

[6] F. Koto, A. Rahimi, J. H. Lau, dan T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 757-770.
[7] R. Eldan dan Y. Li, "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?," *arXiv preprint arXiv:2305.07759*, 2023.
[8] S. Gunasekar *et al.*, "Textbooks Are All You Need," *arXiv preprint arXiv:2306.11644*, 2023.
[9] H. Lee, K. Shu, P. Bhatia, Y. Zhang, dan S. R. Mudambi, "RecipeGPT: Generative Pre-training Based Recipe Generation and Evaluation System," *Companion Proceedings of the Web Conference 2020*, 2020, pp. 181-184.
[10] B. P. Majumder, S. Li, J. Ni, dan J. McAuley, "Generating Personalized Recipes from Historical User Preferences," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 5976-5982.
[11] E. R. Luthfi dan L. M. Alif, "Pengenalan Entitas Bernama pada Teks Resep Makanan Berbahasa Indonesia menggunakan Model Hidden Markov," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 8, no. 4, pp. 332-337, 2019.
[12] H. A. Wibowo *et al.*, "Semi-Supervised Low-Resource Style Transfer for Indonesian Informal to Formal Language," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 3167-3183.
[13] A. F. Aji *et al.*, "One Country, 700+ Languages: NLP Challenges for Underrepresented Languages and Dialects in Indonesia," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 7226-7249.
[14] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.
[15] R. Sennrich, B. Haddow, dan A. Birch, "Neural machine translation of rare words with subword units," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 1715-1725.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i3.17163