

**GENERATOR MAKNA LIRIK LAGU MENGGUNAKAN
TRANSFORMER (STUDI KASUS : LIRIK LAGU DRAKE)**

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



Disusun Oleh :

Nama : Tubagus Alwasi'i

NIM : 32602100121

Program Studi : Teknik Informatika

**FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2025

FINAL PROJECT
SONG LYRIC MEANING GENERATOR USING TRANSFORMER
(CASE STUDY: DRAKE'S SONG LYRICS)

*Proposed to complete the requirement to obtain a bachelor's degree (S1) at
Informatics Engineering Department of Industrial Technology Faculty Sultan
Agung Islamic University*



Disusun Oleh :

Nama : Tubagus Alwasi'i

NIM : 32602100121

Program Studi : Teknik Informatika

FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025

LEMBAR PENGESAHAN
TUGAS AKHIR

GENERATOR MAKNA LIRIK LAGU MENGGUNAKAN *TRANSFORMER*
(STUDI KASUS : LIRIK LAGU DRAKE)

TUBAGUS ALWASI'I
NIM 32602100121

Telah dipertahankan di depan tim penguji sidang tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 12 / 11 / 2025

TIM PENGUJI TUGAS AKHIR:

Moch Taufik, ST.,MIT.
NIK. 210604034
(Ketua Penguji)

12 / 11
2025

Dedy Kurniadi, M.Kom
NIK. 210615048
(Anggota Penguji)

12 / 11
2025

Sam Farisa Chaerul Haviana,
ST.,M.Kom
NIK. 210615046
(Pembimbing)

13 / 11
2025

Semarang, 13 November 2025
Mengetahui,
Kaprodin Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Tubagus Alwasi'i

NIM : 32602100121

Judul Tugas Akhir : Generator Makna Lirik Lagu Menggunakan *Transformer*
(Studi Kasus : Lirik Lagu Drake)

Bahwa dengan ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, Agustus 2025

Yang menyatakan,



Tubagus Alwasi'i

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Tubagus Alwasi'i

NIM : 32602100121

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Semarang

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Generator Makna Lirik Lagu Menggunakan Transformer (Studi Kasus : Lirik Lagu Drake). Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, Agustus 2025

Yang menyatakan,



Tubagus Alwasi'i

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Generator Makna Lirik Lagu Menggunakan Transformer (Studi Kasus : Lirik Lagu Drake).” ini dengan baik. Dengan penuh rasa hormat, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T
3. Dosen pembimbing penulis Bapak Sam Farisa Chaerul Haviana, ST.,M.Kom yang telah memberikan arahan, bimbingan, dan saran yang berarti dalam penyelesaian tugas akhir ini.
4. Seluruh dosen Program Studi Teknik Informatika, Fakultas Teknologi Industri UNISSULA yang telah memberikan ilmunya kepada penulis.
5. Orang tua penulis, Bapak Tugiman dan Ibu Eni Nilawati yang selalu memberikan segala doa, dukungan, dan motivasi dengan penuh limpahan kasih sayang sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.
6. Kakak penulis, Fifi Egiseptifera, Maya Zulfida, Sari Susanto dan adik penulis Kaffah An nas, yang selalu memberikan dukungan moril sehingga penulis mampu menyelesaikan studinya.
7. Untuk Diri Sendiri yang tidak pernah menyerah dan berhenti mencoba sesulit apapun rintangan kuliah dan selama proses penyusunan laporan ini.
8. Rekan seperjuangan, yang telah memberikan dukungan moral, motivasi, serta semangat dalam proses penyelesaian tugas akhir ini.
9. Dan kepada seluruh pihak yang tidak dapat saya sebutkan satu persatu.

Semarang, 27 Agustus 2025

Tubagus Alwasi'i

DAFTAR ISI

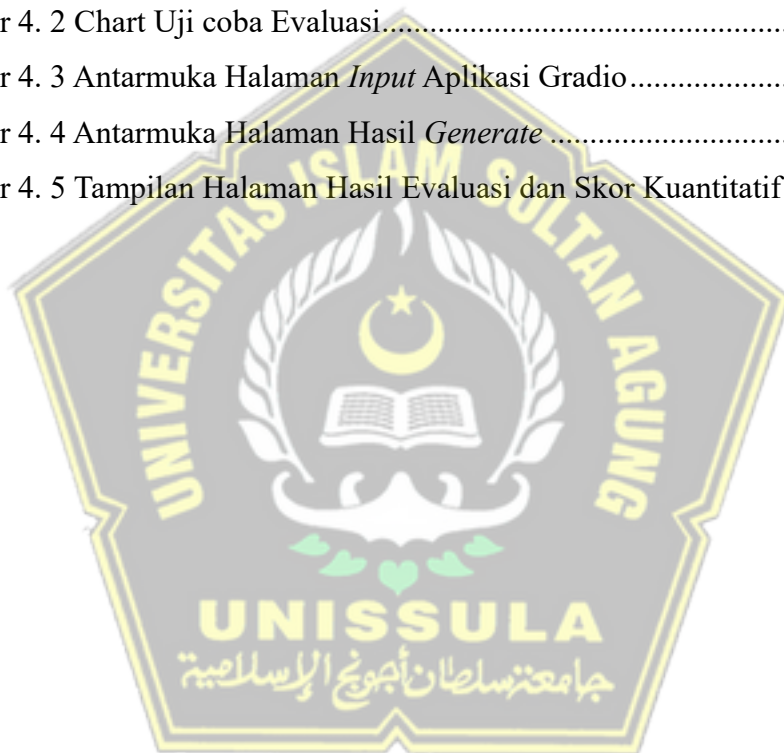
HALAMAN JUDUL	1
LEMBAR PENGESAHAN TUGAS AKHIR	ii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
ABSTRAK	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori	10
2.2.1 Pemrosesan Bahasa Alami (<i>Natural Language Processing</i> - NLP)	10
2.2.2 Model <i>Transformer</i> dan LLaMA 3.....	12
2.2.3 Interpretasi Makna Lirik Lagu	19
2.2.4 Drake	21
BAB III METODE PENELITIAN	18
3.1 Alur Penelitian.....	18
3.2 Rancang Alur Penggunaan Aplikasi	21
3.3 Perancangan <i>User Interface</i>	23
3.3.1 Halaman Awal Sistem	23
3.3.2 Tampilan Hasil Sistem	24
3.4 Perancangan Prompt Sistem	24
BAB IV HASIL DAN ANALISIS PENELITIAN.....	26
4.1 Hasil Penelitian.....	26
4.1.1 Pengumpulan Data	26
4.2 Implementasi sistem Deteksi.....	28
4.2.1 Pra-pemrosesan Data.....	28
4.2.2 Arsitektur Model	31
4.2.3 Pengembangan dan <i>Fine-tuning</i> Model LLaMA 3 + LoRA	32

4.2.4	Pengembangan Aplikasi Web Generator Makna Lirik Lagu	35
4.2.5	Evaluasi Model.....	36
4.3	Implementasi Antar Muka Menggunakan Gradio di Google Colab.....	39
4.3.1	Tujuan Penggunaan Gradio di Google Colab	39
4.3.2	Fitur Utama Aplikasi Gradio di Google Colab	40
4.3.3	Cara Akses Gradio di Google Colab	41
4.3.4	Hasil Implementasi.....	41
4.4	Pembahasan Implementasi	46
BAB V KESIMPULAN DAN SARAN		49
5.1	Kesimpulan.....	49
5.2	Saran	49
DAFTAR PUSTAKA.....		1



DAFTAR GAMBAR

Gambar 2. 1 Foto Drake	21
Gambar 3. 1 <i>Flowchart</i> Rancangan Alur Penelitian	18
Gambar 3. 2 <i>Flowchart</i> Alur Penggunaan Aplikasi	22
Gambar 3. 3 Tampilan Rancangan Halaman <i>Input</i> Pengguna	23
Gambar 3. 4 Tampilan Rancangan <i>Output</i> Hasil.....	24
Gambar 4. 1 Diagram Arsitektur Model	31
Gambar 4. 2 Chart Uji coba Evaluasi.....	38
Gambar 4. 3 Antarmuka Halaman <i>Input</i> Aplikasi Gradio.....	42
Gambar 4. 4 Antarmuka Halaman Hasil <i>Generate</i>	43
Gambar 4. 5 Tampilan Halaman Hasil Evaluasi dan Skor Kuantitatif <i>Real-time</i> .	45



DAFTAR TABEL

Tabel 2. 1 Tabel Tinjauan Pustaka.....	7
Tabel 4. 1 <i>Dataset</i> Lirik Lagu dan Makna Drake.....	26
Tabel 4. 2 Tampilan Sebelum dan Sesudah Data Cleaning.....	28
Tabel 4. 3 Tampilan Sebelum dan Sesudah Diubah ke JSONL	29
Tabel 4. 4 Tabel Konfigurasi <i>Fine Tuning</i>	32
Tabel 4. 5 Tabel Konfigurasi LoRA	33



ABSTRAK

Terbatasnya sumber yang menyediakan interpretasi makna lirik lagu Drake menjadi kendala bagi pendengar untuk memahami pesan lagu secara utuh. Penelitian ini bertujuan membangun aplikasi web generator makna lirik lagu Drake secara otomatis untuk mengatasi masalah tersebut. Sistem ini dikembangkan dengan melakukan *fine-tuning* pada model bahasa LLaMA 3 menggunakan metode *Low-Rank Adaptation* (LoRA) yang efisien. Model yang telah dilatih kemudian diintegrasikan ke dalam aplikasi web interaktif berbasis Gradio di Google Colab dengan arsitektur *Retrieval-Augmented Generation* (RAG). Hasil penelitian menunjukkan sistem berhasil menghasilkan interpretasi lirik yang relevan, divalidasi melalui pendekatan evaluasi ganda: skor kesamaan semantik (*semantic similarity*) yang tinggi untuk akurasi makna dan metrik ROUGE-L untuk keselarasan struktural dengan data referensi. Penelitian ini berhasil menyajikan prototipe fungsional yang membuktikan kelayakan penggunaan LLM dengan metode *fine-tuning* efisien untuk analisis teks musikal yang kompleks.

Kata Kunci: Generator Makna Lirik, Transformer, LLaMA 3, Drake, Natural Language Processing, Fine-tuning, LoRA, ROUGE-L.

ABSTRACT

The limited availability of sources providing interpretations of Drake's song lyrics hinders listeners from fully understanding the messages within his music. This research aims to address this issue by building a web-based song lyric meaning generator capable of automatically interpreting Drake's lyrics. The system was developed by fine-tuning the LLaMA 3 large language model using the efficient Low-Rank Adaptation (LoRA) method. The trained model was then integrated into an interactive web application built with Gradio on Google Colab, featuring a Retrieval-Augmented Generation (RAG) architecture. The results show that the system successfully generates relevant lyric interpretations, validated through a dual-evaluation approach: a high semantic similarity score for contextual accuracy and the ROUGE-L metric for structural alignment with reference data. This study successfully presents a functional prototype, proving the feasibility of using LLMs with efficient fine-tuning methods for the analysis of complex musical texts.

Keywords: Song Lyric Meaning Generator, Transformer, LLaMA 3, Drake, Natural Language Processing, Fine-tuning, LoRA, ROUGE-L.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan kecerdasan buatan (AI) dan pemrosesan bahasa alami (NLP) telah mengalami kemajuan pesat dalam beberapa tahun terakhir. Salah satu inovasi yang banyak digunakan dalam NLP (*Natural Language Processing*) adalah model *Transformer*, yang memiliki kemampuan unggul dalam memahami konteks panjang dan hubungan antar kata dalam suatu teks. Teknologi ini telah diterapkan dalam berbagai bidang, termasuk analisis lirik lagu, yang sering kali memiliki struktur kompleks dan makna tersirat. Penerapan *Transformer* dalam tugas ini memungkinkan analisis teks yang lebih akurat dan efisien (Hu, 2024).

Hip-hop merupakan salah satu *genre* musik yang dikenal memiliki lirik dengan makna mendalam, penuh metafora, permainan kata, dan referensi budaya yang kaya. Salah satu artis hip-hop terkenal, Drake, sering menggunakan teknik penulisan yang kompleks dalam lirik lagunya. Interpretasi makna lirik dalam lagu-lagu hip-hop menjadi tantangan tersendiri karena adanya penggunaan bahasa kiasan dan konteks sosial yang khas. Oleh karena itu, penggunaan model *Transformer* dalam analisis lirik hip-hop dapat membantu mengidentifikasi pola makna dan memahami pesan yang ingin disampaikan oleh musisi (Agrawal dkk, 2021).

Beberapa penelitian telah menunjukkan bahwa model *Transformer* mampu menghasilkan dan menganalisis lirik lagu dengan tingkat akurasi yang tinggi. Misalnya, model T5 *Transformer* telah digunakan untuk menghasilkan lirik yang sesuai dengan pola rima dan ritme tertentu, serta memahami keterkaitan antara teks dan aspek musikalitas lainnya. Selain itu, penelitian lain menunjukkan bahwa *Transformer* dapat menghubungkan lirik dengan struktur melodi dan pola nada dalam musik, sehingga membantu dalam interpretasi makna lagu secara lebih menyeluruh (Duan dkk, 2022; Ram dkk, 2021).

Pemilihan Drake sebagai studi kasus utama dalam penelitian ini didasarkan pada dua alasan fundamental: signifikansi budaya *global* dan kompleksitas lirik karyanya. Dari sisi signifikansi, Drake adalah salah satu musisi paling berpengaruh di era kontemporer, dibuktikan dengan statusnya sebagai artis dengan *streaming*

terbanyak dalam sejarah Spotify. Popularitas kuantitatif ini, yang mencakup lebih dari 80,76 juta pendengar aktif bulanan per November 2025, menunjukkan adanya basis audiens global yang masif dan permintaan yang tinggi untuk memahami makna di balik karyanya.

Secara lirik, karya Drake dikenal dengan gaya penulisan yang emosional, reflektif, dan sarat metafora. Tidak seperti lirik yang lugas, interpretasi makna lagu Drake sering kali tidak eksplisit dan sangat bergantung pada pemahaman konteks sosial, sejarah pribadi, atau referensi budaya populer yang kompleks. Kombinasi antara popularitas global yang masif dan kedalaman liris yang menantang inilah yang menjadikan karya Drake sebagai studi kasus yang mendesak dan relevan untuk pengembangan generator makna berbasis AI.

Penelitian ini bertujuan untuk mengembangkan generator berbasis *Transformer* yang mampu menganalisis dan menginterpretasikan makna lirik lagu-lagu Drake dari album *More Life* hingga *Dark Lane Demo Tapes*. Dengan memanfaatkan kecanggihan NLP, penelitian ini diharapkan dapat memberikan wawasan lebih dalam tentang bagaimana AI dapat memahami makna lirik lagu dalam musik hip-hop, khususnya dalam karya Drake.

Urgensi dari penelitian ini terletak pada masih terbatasnya sumber yang menyediakan interpretasi makna lirik lagu Drake secara lengkap dan akurat. Informasi makna yang tersedia di berbagai platform sering kali belum mencakup seluruh lagu atau hanya memberikan penjelasan singkat tanpa konteks mendalam. Keterbatasan ini membuat kebutuhan akan sebuah sistem otomatis yang mampu menghasilkan penafsiran makna lirik secara cepat, relevan, dan kontekstual menjadi semakin penting. Oleh karena itu, penelitian ini mengembangkan *generator* makna lirik lagu berbasis AI yang dapat memberikan interpretasi lirik Drake secara menyeluruh, sehingga penikmat musik dapat memperoleh pemahaman yang lebih lengkap terhadap karya tersebut.

1.2 Rumusan Masalah

Terbatasnya sumber yang menyediakan interpretasi makna lirik lagu Drake menyebabkan penikmat musik tidak memperoleh pemahaman yang utuh terhadap pesan dalam lirik tersebut. Kondisi ini menimbulkan permasalahan mengenai

bagaimana menghadirkan interpretasi makna lirik yang lengkap guna memenuhi kebutuhan tersebut?

1.3 Batasan Masalah

1. Analisis lirik lagu akan dibatasi pada album *More Life* hingga *Dark Lane Demo Tapes* milik Drake, tanpa mencakup lagu-lagu di luar album tersebut. Pemilihan album ini didasarkan pada evolusi gaya lirik Drake yang terlihat dalam periode tersebut, serta variasi tema yang kaya dari liriknya. Selain itu dataset lirik dari album – album ini tersedia dalam format yang memudahkan proses analisis dan pelatihan model.
2. Fokus penelitian adalah pada pemahaman makna lirik berdasarkan struktur bahasa dan penggunaan kata dalam konteks lagu, tanpa mempertimbangkan aspek musikal seperti melodi atau harmoni.
3. Target pengguna sistem ini adalah *Team Drizzy* atau penggemar musik Drake seperti komunitas yang tergabung di Reddit <https://www.reddit.com/r/Drizzy/>, yang ingin memahami makna lirik lagu.

1.4 Tujuan Penelitian

Membangun aplikasi web generator makna lirik lagu yang dapat menginterpretasikan makna lirik lagu Drake secara otomatis.

1.5 Manfaat

1. Membantu pengguna umum dan penikmat musik dalam memahami makna tersembunyi atau simbolik dari lirik lagu, khususnya dalam genre hip-hop terutama penyanyi Drake yang kaya metafora dan kompleks.
2. Menyediakan alat bantu otomatis berupa aplikasi web yang dapat digunakan untuk menghasilkan interpretasi makna lirik secara cepat dan kontekstual.

1.6 Sistematika Penulisan

BAB I : PENDAHULUAN

Pada bab 1, penulis mengutarakan urgensi dari penelitian yang diangkat, mulai dari penulisan latar belakang, membuat rumusan masalah, membatasi permasalahan yang dibahas, serta tujuan dan manfaat yang diperoleh, dan sistematika penulisan

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada Bab 2, penulis memuat dasar teori yang digunakan, serta rujukan dari penelitian terdahulu yang akan digunakan dalam perancangan sistem, dan membantu penulis untuk memahami teori yang berhubungan dengan NLP, Transformers dan model Llama selama proses penelitian.

BAB III : METODE PENELITIAN

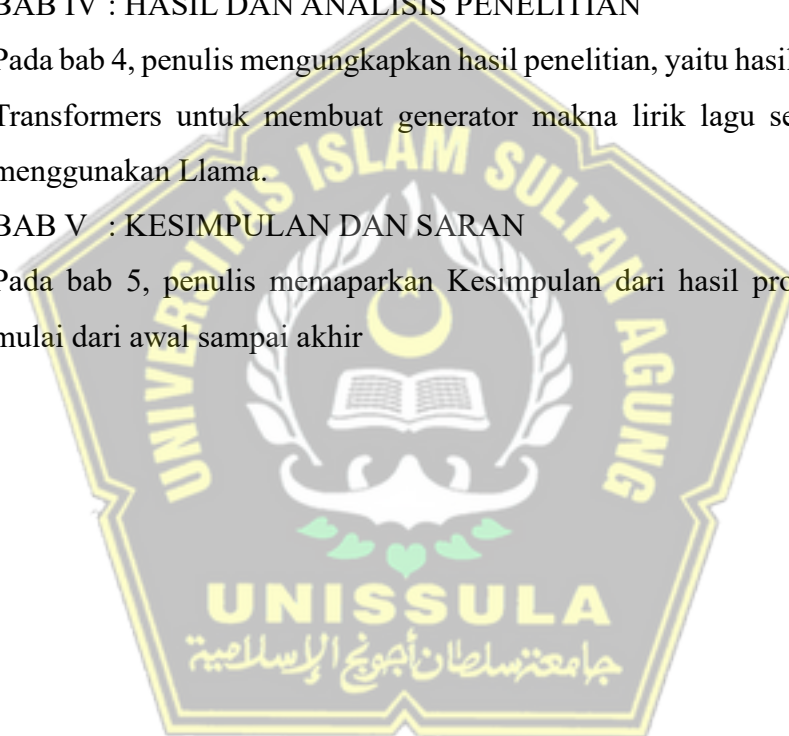
Pada bab 3, penulis mengungkapkan proses dan tahapan penelitian yang dimulai dari mendapatkan dataset hingga proses pemodelan topik data yang ada.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada bab 4, penulis mengungkapkan hasil penelitian, yaitu hasil implementasi Transformers untuk membuat generator makna lirik lagu secara otomatis menggunakan Llama.

BAB V : KESIMPULAN DAN SARAN

Pada bab 5, penulis memaparkan Kesimpulan dari hasil proses penelitian mulai dari awal sampai akhir



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian oleh Agrawal dkk. (2021) mengimplementasikan arsitektur XLNet (*Extended Transformer-Based Language Model*) untuk analisis emosi dalam lirik lagu dan memperoleh akurasi tinggi, membuktikan bahwa pendekatan ini lebih unggul dibandingkan metode tradisional dalam menangkap makna emosional yang terkandung dalam teks lagu. Keunggulan tersebut terutama terletak pada kemampuan *Transformer* dalam mempertahankan konteks kata dan memahami hubungan semantik antarbaris lirik, sehingga hasil analisis menjadi lebih akurat (Agrawal dkk, 2021). Sejalan dengan itu, pada penelitian oleh Naseri dkk. (2022) mengembangkan model *Transformer* untuk menghubungkan lirik dengan kategori emosi tertentu secara otomatis dan akurat. Pendekatan ini tidak hanya mengklasifikasikan emosi, tetapi juga memberikan interpretasi yang lebih kontekstual dibandingkan metode analisis sentimen berbasis kata kunci (Naseri dkk, 2022). Pada penelitian oleh Ventura dan Toker (2022) turut memperluas penerapan *Transformer* dengan mengembangkan TRBLLmaker (*Transformer Research for Behavior, Language, and Lyrics Maker*) model GPT-2 yang dirancang khusus untuk lirik hip-hop—yang mampu menafsirkan makna tersirat dalam teks lagu. Model ini terbukti unggul dalam memahami makna kompleks dan permainan kata yang lazim digunakan dalam genre hip-hop, menunjukkan relevansi tinggi untuk penelitian yang berfokus pada lirik artis seperti Drake (Ventura dan Toker, 2022).

Pengembangan model LLaMA juga menjadi fokus beberapa penelitian yang relevan dengan topik ini. Pada penelitian oleh Zhang dkk. (2023) memperkenalkan LLaMA-Adapter, yaitu metode *fine-tuning* efisien dengan teknik *zero-init attention* yang memungkinkan adaptasi cepat terhadap instruksi baru tanpa mengganggu pengetahuan awal model. Keunggulan metode ini terletak pada efisiensi pelatihan ulang yang membuatnya cocok untuk penyesuaian terhadap domain khusus, seperti analisis lirik lagu (Zhang dkk, 2023). Sementara itu, pada penelitian Wu dkk. (2024) mengembangkan LLaMA Pro, sebuah pengembangan dari LLaMA-2 melalui

ekspansi blok *Transformer* secara progresif. Hasilnya menunjukkan performa tinggi dalam berbagai tugas generalisasi bahasa, termasuk pemrosesan teks yang kompleks secara sintaksis maupun semantic (Wu dkk, 2024). Penelitian oleh Dennis Heraldi & Zakhralativa Ruskanda (2024) membuktikan efektivitas LLaMA-2 dalam mendeteksi makna implisit seperti sarkasme, yang merupakan aspek penting dalam menafsirkan lirik dengan bahasa kiasan (Dennis Heraldi dan Zakhralativa Ruskanda, 2024). Pada penelitian Lv dkk. (2025) melalui T-LLaMA menunjukkan bahwa model ini dapat diadaptasi untuk bahasa dengan struktur yang rumit, menandakan potensi besar untuk memahami nuansa linguistik yang khas dalam karya musik tertentu (Lv dkk, 2025).

Selain itu, sejumlah studi terkini menunjukkan kemajuan signifikan dalam analisis lirik lagu dari berbagai perspektif. Pada penelitian oleh Preniqi dkk (2023) menemukan bahwa lirik dan fitur audio tidak hanya berperan sebagai elemen estetika, tetapi juga mencerminkan preferensi musik serta nilai moral pendengar. Penelitian ini menegaskan bahwa integrasi analisis lirik dengan fitur audio dapat meningkatkan akurasi sistem rekomendasi musik berbasis konteks, terutama dalam memahami keterkaitan antara emosi, pesan, dan karakteristik musik yang disukai (Preniqi dkk, 2023). Pada penelitian oleh Huang dan Benetos (2024) memperkenalkan 4MuLA, sebuah dataset multitugas, multimodal, dan multibahasa yang menggabungkan lirik lagu dan fitur audio. Dataset ini dirancang untuk mendukung pengembangan model analisis musik yang mampu memahami keterkaitan antara bahasa, melodi, dan emosi, serta memfasilitasi penelitian lintas bahasa dan lintas genre music (Huang dan Benetos, 2024). Pada penelitian oleh Choi (2023) melalui studi *Computational Thematic Analysis of Poetry via Bimodal Large Language Models* mengembangkan metode untuk mengidentifikasi dan menganalisis tema dalam puisi menggunakan model bahasa besar bimodal. Meskipun fokus awalnya pada karya sastra puisi, pendekatan ini sangat relevan untuk mengungkap pola tematik dan naratif dalam lirik lagu, terutama yang memiliki kompleksitas bahasa tinggi (Kahyun dan Choi, n.d.). Sementara itu, pada penelitian oleh Chu dkk (2022) dalam penelitian HIT at SemEval-2022 *Task 2* menunjukkan bahwa model bahasa *pre-trained* dapat mendeteksi ungkapan

idiomatik dan literal dalam konteks teks secara akurat. Kemampuan ini sangat bermanfaat dalam menganalisis lirik genre hip-hop dan R&B yang kerap menggunakan bahasa kiasan, permainan kata, dan metafora kompleks untuk menyampaikan pesan artistic (Chu dkk, 2022).

Selain penelitian utama yang telah diuraikan sebelumnya, terdapat pula sejumlah studi relevan lainnya yang memperkuat landasan pengembangan generator makna lirik serta pemanfaatan LLaMA dalam konteks pemahaman makna lirik lagu. Penelitian-penelitian ini dirangkum dalam tabel berikut untuk menyajikan gambaran ringkas mengenai metode, hasil, dan kontribusinya.

Tabel 2. 1 Tabel Tinjauan Pustaka

Judul Penelitian	Metode	Dataset	Hasil
<i>Effective Intended Sarcasm Detection Using Fine-tuned LLaMA 2</i> (Dennis Heraldi dan Zakhrativa Ruskanda, 2024)	<i>Fine-tuning</i> model LLaMA-2 menggunakan <i>Parameter Efficient Fine-tuning</i> (PEFT) dan QLoRA untuk mendeteksi sarkasme dalam teks	iSarcasm dataset	Model ini menunjukkan peningkatan signifikan dalam mendeteksi sarkasme dengan F1-score sebesar 0.6867. Sistem juga mampu mengidentifikasi kategori sarkasme dan pasangan pernyataan sarkastik secara akurat, meskipun menggunakan perangkat dengan sumber daya terbatas. Hal ini membuktikan bahwa LLaMA-2 dapat digunakan secara efisien untuk memahami makna implisit seperti ironi atau sindiran, yang relevan untuk analisis makna dalam lirik music.
T-LLaMA: Tibetan <i>Large Language Model Based on LLaMA2</i> (Lv dkk., 2025)	<i>Pre-training</i> ulang LLaMA-2 dengan penambahan korpus dan kosa kata bahasa Tibet menggunakan <i>SentencePiece</i>	Tibetan Web Corpus, Wikipedia Tibet, dan berita <i>online</i> lokal berbahasa Tibet	Model ini berhasil mencapai akurasi 79.8% pada klasifikasi berita berbahasa Tibet, serta menghasilkan teks ringkasan dan teks baru yang dinilai baik oleh penilai manusia. Studi

			ini menunjukkan efektivitas LLaMA dalam menangani bahasa dengan struktur kompleks, yang dapat diadaptasi untuk menganalisis lirik lagu dengan konteks budaya dan linguistik khusus seperti hip-hop.
Scaling Open Foundation Models for Long-Form Music Generation (Yuan dkk, 2025)	Foundation model berbasis LLaMA2 untuk menghasilkan lagu dari lirik (lyrics-to-song).	Token audio dan teks skala besar.	Mampu menjaga keselarasan lirik dan musik dengan gaya beragam, cocok untuk eksplorasi multimodal generatif.
<i>Radiology-LLaMA2: Domain-Specific LLM for Radiology</i> (Liu dkk., 2023)	<i>Fine-tuning</i> LLaMA-2 dengan korpus khusus radiologi menggunakan pendekatan <i>instruction tuning</i>	MIMIC-CXR dan <i>PadChest</i> (dataset berisi laporan radiologi)	Model ini berhasil mencapai performa terbaik dalam menghasilkan ringkasan laporan radiologi dengan skor ROUGE tertinggi pada dua dataset medis. Evaluasi oleh ahli juga menunjukkan bahwa hasil keluaran model sangat relevan, ringkas, dan koheren. Hal ini membuktikan efektivitas LLaMA-2 ketika disesuaikan pada <i>domain</i> tertentu, seperti halnya jika digunakan pada <i>domain</i> lirik musik hip-hop yang khas.
<i>A Strategy to Combine IstGen Transformers and Open LLMs for Text Classification</i>	Kombinasi model transformer generasi pertama (BERT, RoBERTa) dan LLaMA-2 dengan strategi klasifikasi	AG News, IMDb, dan Yelp Reviews	Studi ini menyarankan strategi efisien dengan menggunakan model ringan untuk kasus prediksi yakin, dan LLaMA-2 untuk kasus ambigu. Hasilnya menunjukkan bahwa

(de Andrade dkk, 2024)	berbasis tingkat kepercayaan		metode ini hampir menyamai performa model LLaMA-2 yang di-fine-tune sepenuhnya, namun dengan biaya komputasi yang jauh lebih rendah. Ini menunjukkan bahwa LLaMA dapat dimanfaatkan secara selektif dalam tugas-tugas kompleks, termasuk klasifikasi makna lirik yang ambigu atau berlapis.
<i>Transformer-Based Approach Towards Music Emotion Recognition From Lyrics</i> (Agrawal dkk, 2021)	<i>Fine-tuning</i> XLNet untuk klasifikasi emosi berbasis fitur teks	Dataset emosi lagu	Model menunjukkan performa tinggi dalam mengklasifikasikan emosi dalam lirik, lebih baik dibanding metode konvensional
<i>The Contribution of Lyrics and Acoustics to Collaborative Understanding of Mood</i> (Naseri dkk, 2022)	<i>Fine-tuning</i> model <i>Transformer</i> untuk klasifikasi multi-label emosi/mood berdasarkan lirik lagu.	Dataset lirik lagu dengan label emosi	<i>Transformer</i> menghasilkan prediksi suasana hati yang akurat dan interpretatif, melebihi metode berbasis kata kunci
TRBLLmaker -- <i>Transformer Reads Between Lyrics Lines maker</i> (Ventura dan Toker, 2022)	Model GPT-2 <i>decoder</i> untuk generasi makna lirik hip-hop	Kumpulan lirik hip-hop	TRBLLmaker dapat menginterpretasi makna tersirat dari lirik hip-hop dengan akurasi tinggi dalam konteks budaya dan Bahasa
LLaMA-Adapter: <i>Parameter-Efficient Fine-Tuning of LLaMA for Instruction</i>	<i>Zero-init attention</i> untuk <i>fine-tuning</i> efisien pada LLaMA	<i>Instruction</i> dataset	LLaMA-Adapter memungkinkan adaptasi cepat terhadap instruksi baru tanpa kehilangan pengetahuan awal

<i>Tuning</i> (Zhang dkk, 2023)			
LLaMA-Pro: <i>Progressive Expansion of Transformer for Code and Math Tasks</i> (Wu dkk, 2024)	Ekspansi progresif blok <i>Transformer</i> dalam LLaMA-2	Korpus umum termasuk kode dan teks teknis	Model dapat menangani berbagai tugas bahasa kompleks termasuk kode dan teks bermakna tinggi, cocok untuk lirik kompleks

Dengan demikian, penelitian ini menawarkan pendekatan yang lebih spesifik dengan membangun generator makna lirik dari nol yang dirancang khusus untuk memahami makna lirik dalam musik Drake, dengan mempertimbangkan struktur bahasa dan konteks budaya yang khas. Hal ini diharapkan dapat meningkatkan akurasi pemahaman makna dibandingkan dengan model umum yang sudah ada.

2.2 Dasar Teori

2.2.1 Pemrosesan Bahasa Alami (*Natural Language Processing* - NLP)

Pemrosesan Bahasa Alami (*Natural Language Processing*/NLP) merupakan cabang dari kecerdasan buatan yang memungkinkan mesin untuk memahami, memproses, dan menghasilkan bahasa manusia secara alami. Dalam konteks analisis lirik lagu, NLP (*Natural Language Processing*) digunakan untuk memahami pola linguistik, struktur kalimat, dan hubungan semantik antar kata yang membentuk makna secara keseluruhan. Dengan pendekatan ini, sistem dapat menangkap nuansa bahasa seperti metafora, permainan kata, dan ekspresi emosional dalam teks lagu. Fondasi dari pemahaman mesin ini terletak pada dua tingkat analisis fundamental. Pertama adalah analisis sintaktik, yang berfokus pada struktur gramatikal kalimat melalui proses seperti tokenisasi untuk memecah teks menjadi unit kata dan *Part-of-Speech* (POS) *Tagging* untuk melabeli setiap kata dengan kelas katanya (misalnya, kata benda, kata kerja). Kedua adalah analisis semantik, yang bertujuan untuk menafsirkan makna eksplisit dan implisit dalam teks, seperti dalam tugas *Word Sense Disambiguation* yang menentukan makna kata yang tepat berdasarkan konteks kalimat di sekitarnya (Weir dkk, 2020).

Di atas fondasi analisis tersebut, NLP (*Natural Language Processing*) menjalankan serangkaian tugas yang lebih kompleks dan berorientasi pada aplikasi spesifik. Tugas-tugas ini memanfaatkan pemahaman sintaksis dan semantik untuk melakukan fungsi yang lebih canggih. Beberapa di antaranya adalah klasifikasi teks, yang digunakan untuk mengkategorikan sebuah tulisan ke dalam kelas yang telah ditentukan, seperti analisis sentimen atau deteksi spam. Tugas penting lainnya adalah pengenalan entitas bernama (NER), yang berfungsi untuk mengekstrak informasi kunci seperti nama orang, organisasi, atau lokasi dari dalam teks. Selain itu, terdapat penerjemahan mesin yang secara otomatis mengalihbahasakan teks, peringkasan teks untuk menghasilkan rangkuman dari dokumen panjang, serta generasi bahasa alami (NLG), di mana mesin menghasilkan teks naratif yang koheren dari data terstruktur, seperti yang dilakukan dalam pembuatan laporan atau respons *chatbot* (Mao dkk, 2025).

Penerapan dari berbagai tugas NLP (*Natural Language Processing*) ini telah terintegrasi secara luas dalam teknologi yang kita gunakan sehari-hari. Asisten virtual seperti Siri dan Google Assistant, bersama dengan *chatbot* layanan pelanggan, mengandalkan NLP untuk memahami perintah dan pertanyaan pengguna lalu memberikan respons yang relevan. Mesin pencari modern juga sangat bergantung pada NLP untuk menafsirkan maksud (*intent*) di balik kueri pencarian, sehingga mampu menyajikan hasil yang lebih akurat daripada sekadar pencocokan kata kunci. Dalam aktivitas digital lainnya, NLP digunakan oleh layanan email untuk menyaring pesan spam secara efektif melalui klasifikasi teks dan oleh aplikasi pemeriksa tata bahasa seperti *Grammarly* untuk memberikan koreksi dan saran penulisan secara *real-time*. Di dunia bisnis, NLP (*Natural Language Processing*) menjadi alat vital untuk analisis sentimen, yang memungkinkan perusahaan memantau opini publik dan umpan balik pelanggan dari jutaan sumber data tidak terstruktur seperti media sosial (Tong dkk, 2022).

Meskipun memiliki kemampuan yang kuat, teknologi NLP (*Natural Language Processing*) juga dihadapkan pada kelebihan dan tantangan yang fundamental. Keunggulan utamanya terletak pada efisiensi dan skalabilitas dalam memproses data tekstual bervolume besar secara cepat dan objektif, serta

kemampuannya untuk mengotomatiskan tugas-tugas berulang. Namun, tantangan terbesar NLP (*Natural Language Processing*) adalah kesulitan dalam memahami ambiguitas dan konteks bahasa manusia yang kompleks, seperti menafsirkan sarkasme, ironi, atau metafora. Masalah kritis lainnya adalah potensi bias dalam model, di mana prasangka yang terdapat dalam data pelatihan dapat diserap dan diperkuat oleh model, yang menimbulkan isu etis. Selain itu, performa model NLP (*Natural Language Processing*) canggih sangat bergantung pada ketersediaan dataset pelatihan yang besar dan berkualitas tinggi, dan teknologi ini masih kesulitan menangani variasi bahasa seperti dialek, bahasa gaul, atau bahasa dengan sumber daya digital yang terbatas (Bender dkk, 2021).

2.2.2 Model *Transformer* dan LLaMA 3

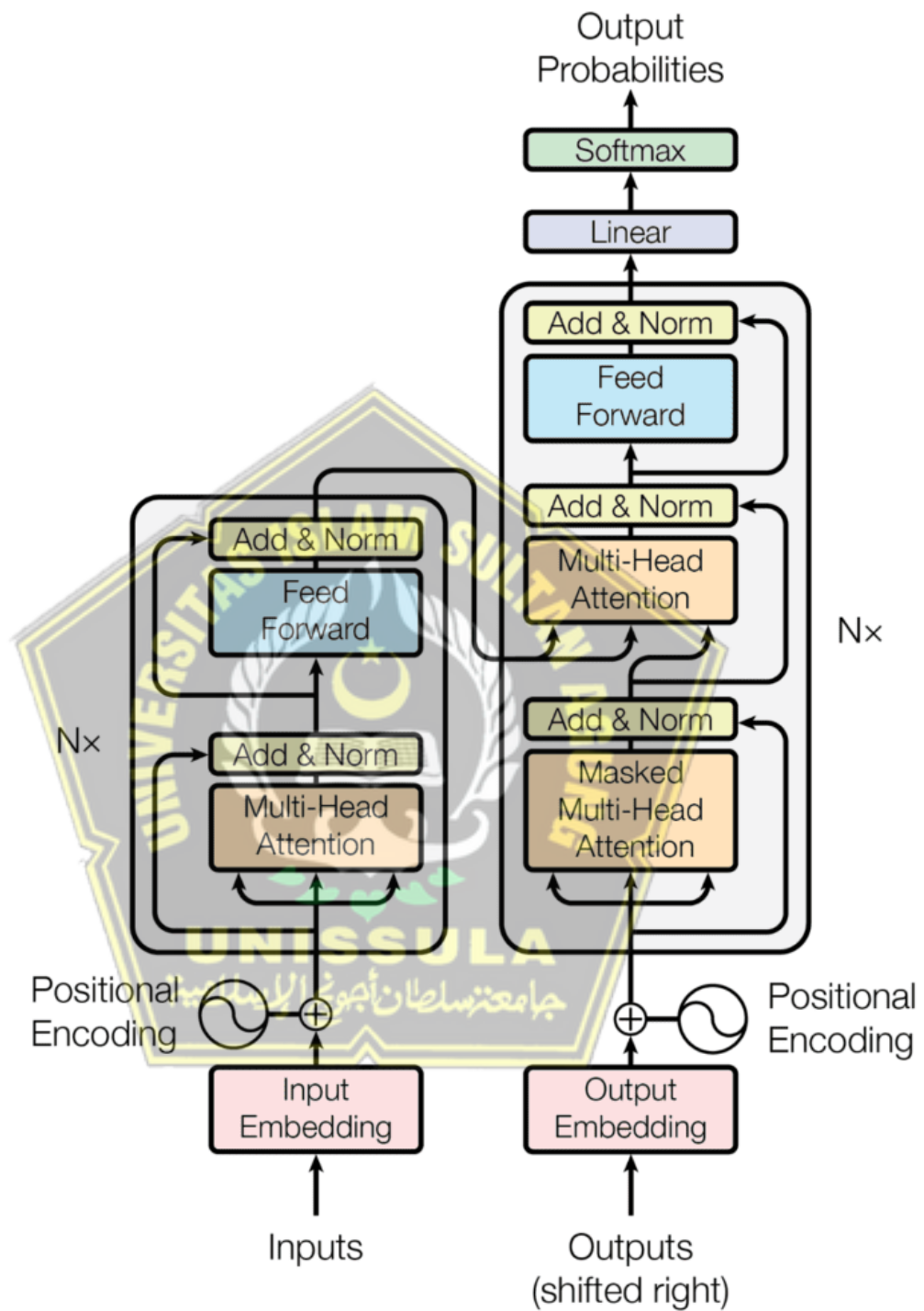
Transformer merupakan arsitektur jaringan saraf dalam yang diperkenalkan oleh Vaswani dkk. (2017) melalui makalah *Attention is All You Need*. Arsitektur ini merevolusi bidang pemrosesan bahasa alami (*Natural Language Processing*/NLP) karena berhasil mengatasi keterbatasan pendekatan berbasis sekuensial seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM). Kedua model terdahulu tersebut bekerja dengan memproses kata secara berurutan, sehingga menghadapi kesulitan dalam menangani dependensi jarak jauh serta cenderung lambat karena keterbatasan paralelisasi. *Transformer* hadir dengan pendekatan berbeda, yakni mengolah seluruh data masukan secara paralel melalui mekanisme inti yang disebut *self-attention*. Mekanisme ini memungkinkan setiap kata dalam sebuah kalimat untuk dipertimbangkan relevansinya dengan semua kata lain dalam kalimat tersebut secara simultan, terlepas dari jarak posisinya (Lin dkk, 2022; Vaswani dkk, 2017).

Secara umum, arsitektur *Transformer* terdiri dari dua komponen utama yaitu *encoder* dan *decoder*. *Encoder* berfungsi mengubah teks masukan menjadi representasi vektor yang kaya konteks, sedangkan *decoder* bertugas menghasilkan keluaran berdasarkan representasi tersebut. Setiap lapisan *encoder* dibangun dari dua komponen inti, yaitu *multi-head self-attention* yang memungkinkan model memperhatikan berbagai hubungan antar token secara paralel, dan *feed-forward neural network* yang memperkuat representasi hasil proses atensi. *Decoder*

memiliki struktur serupa dengan encoder, namun dilengkapi dengan *masked self-attention* untuk mencegah kebocoran informasi dari token di masa depan, serta *encoder-decoder attention* yang memungkinkan *decoder* memanfaatkan representasi dari *encoder* untuk menghasilkan keluaran yang koheren. Baik *encoder* maupun *decoder* menggunakan *residual connection* dan *layer normalization* agar pelatihan lebih stabil, serta *positional encoding* untuk menjaga informasi urutan kata karena mekanisme atensi tidak memiliki kesadaran posisi secara alami (Khan dkk, 2022; Vaswani dkk, 2017).

Gambar berikut memperlihatkan arsitektur dasar *Transformer* dengan alur informasi dari masukan hingga keluaran melalui lapisan *encoder* dan *decoder*.





Gambar 2. 1 Gambar Arsitektur *Transformers*

Gambar 2.1 menunjukkan arsitektur Transformer yang terdiri dari dua komponen utama, yaitu encoder dan decoder, di mana keduanya bekerja sama untuk memproses input dan menghasilkan output secara autoregresif. Model ini

diperkenalkan oleh Vaswani (2017) sebagai pendekatan baru yang sepenuhnya berbasis pada mekanisme attention. Bagian encoder menerima input berupa representasi vektor dari kata melalui proses embedding, kemudian ditambahkan informasi urutan melalui positional encoding. Setiap lapisan encoder terdiri dari mekanisme multi-head self-attention yang memungkinkan kata dalam suatu kalimat saling memperhatikan satu sama lain untuk menangkap konteks, dilanjutkan dengan feed forward network untuk memperkuat representasi, serta add & norm yang berfungsi menjaga kestabilan pembelajaran melalui residual connection dan normalisasi. Hasil dari encoder berupa representasi kontekstual yang akan digunakan oleh decoder (Vaswani dkk, 2017).

Decoder bekerja dengan prinsip serupa namun memiliki perbedaan penting. Pertama, terdapat masked multi-head self-attention yang membatasi perhatian hanya pada token sebelumnya agar generasi teks bersifat autoregresif. Selanjutnya, terdapat mekanisme encoder-decoder attention yang menghubungkan representasi input dari encoder dengan proses generasi output. Sama seperti encoder, setiap lapisan decoder juga dilengkapi dengan feed forward network dan add & norm. Setelah melalui lapisan-lapisan tersebut, keluaran decoder diproyeksikan ke ruang kosakata melalui lapisan linear, lalu diproses dengan softmax untuk menghasilkan distribusi probabilitas kata berikutnya. Dengan cara ini, Transformer dapat membangun keluaran teks secara bertahap berdasarkan konteks input dan token yang telah dihasilkan sebelumnya (Vaswani dkk, 2017).

Keunggulan utama *Transformer* terletak pada fleksibilitas dan skalabilitasnya. Berbeda dengan RNN (*Recurrent Neural Network*) yang kesulitan dalam mempelajari konteks panjang, *Transformer* mampu menangkap dependensi jarak jauh dengan lebih efektif. Selain itu, karena seluruh token dapat diproses secara paralel, pelatihan *Transformer* jauh lebih cepat dibandingkan model sekuensial. *Transformer* juga sangat mudah diskalakan ke dalam model dengan jumlah parameter yang sangat besar, sehingga mendukung lahirnya model bahasa berskala besar (*Large Language Models/LLM*) yang saat ini mendominasi bidang NLP (*Neuro Linguistic Programming*) (Qiu dkk, 2020). Namun demikian, arsitektur ini memiliki kekurangan berupa kebutuhan komputasi yang sangat tinggi,

baik dari sisi memori maupun energi. Selain itu, *Transformer* memerlukan jumlah data pelatihan yang masif agar dapat mencapai performa optimal, serta berpotensi menyerap bias dari data yang digunakan dalam proses pelatihan (Raffel dkk., 2020).

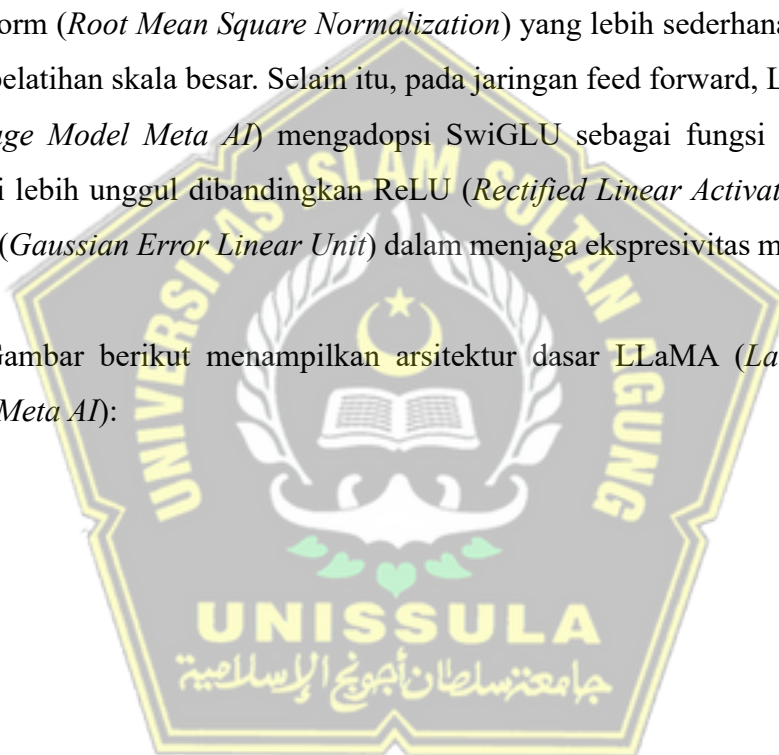
Transformer kini telah digunakan secara luas dalam berbagai tugas NLP (Neuro Linguistic Programming). Model encoder seperti BERT (*Bidirectional Encoder Representations*) terbukti efektif untuk klasifikasi teks, ekstraksi entitas, dan pemahaman bahasa. Model decoder seperti GPT (*Generative Pre-trained Transformer*) digunakan untuk generasi teks dan percakapan otomatis, sedangkan model encoder-decoder seperti T5 berhasil mencapai performa tinggi pada penerjemahan mesin, peringkasan teks, hingga sistem tanya jawab. Lebih jauh, adaptasi Transformer juga telah berkembang di luar NLP (*Neuro Linguistic Programming*), misalnya dalam bidang visi komputer melalui *Vision Transformer* (ViT) (Dosovitskiy dkk., 2021), bioinformatika, dan pemrosesan audio. Dengan fleksibilitas dan efektivitasnya, *Transformer* telah menjadi *state-of-the-art* dalam pembelajaran mesin modern.

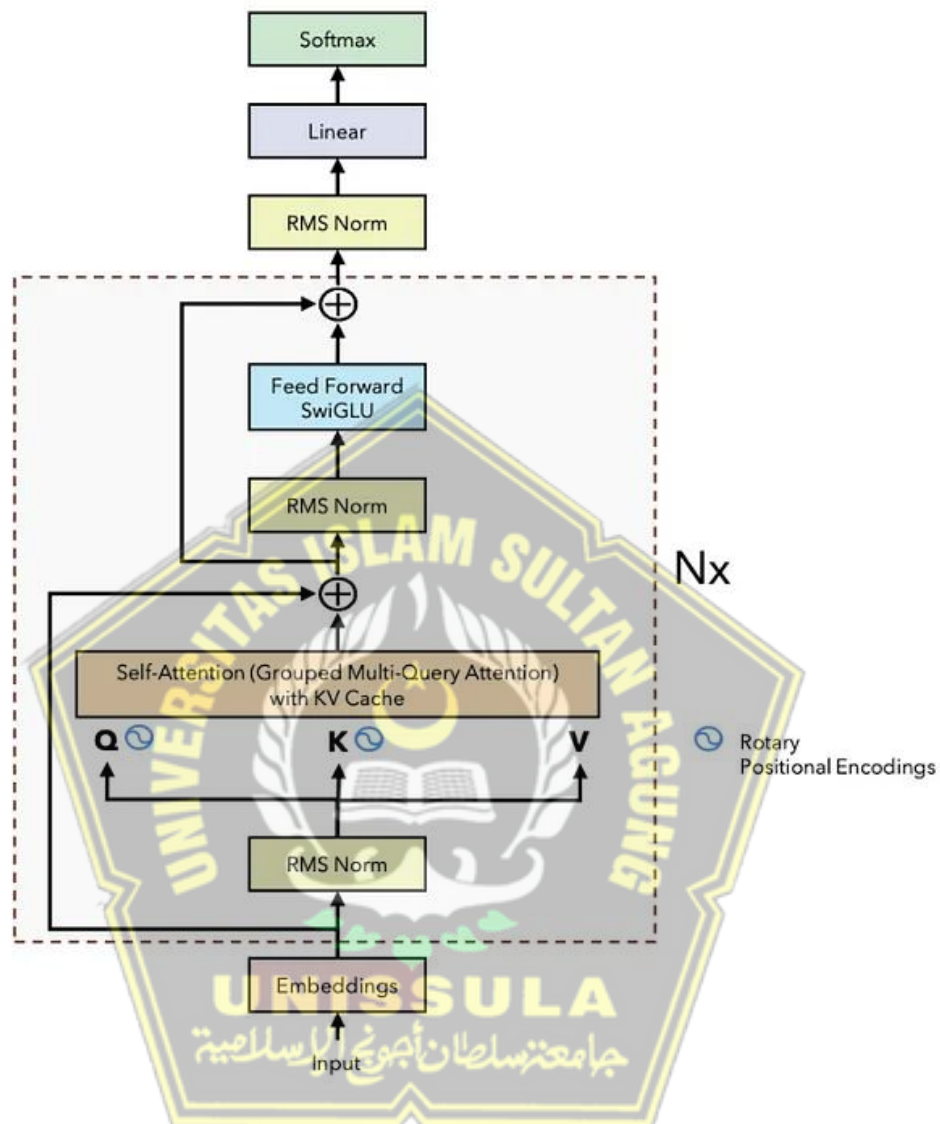
LLaMA (*Large Language Model Meta AI*) adalah sebuah model bahasa berskala besar yang dikembangkan oleh Meta AI (*Artificial Intelligence*) sebagai salah satu alternatif terbuka untuk penelitian kecerdasan buatan generatif. LLaMA (*Large Language Model Meta AI*) pertama kali diperkenalkan pada tahun 2023 dengan tujuan menghadirkan model bahasa yang lebih efisien, hemat sumber daya, tetapi tetap memiliki performa setara bahkan melampaui model besar lain seperti GPT-3 (Touvron dkk., 2023). Berbeda dengan model tertutup komersial, LLaMA (*Large Language Model Meta AI*) dirancang agar lebih terjangkau bagi komunitas akademik dan peneliti, karena dapat dijalankan di perangkat dengan GPU tunggal atau lingkungan komputasi yang relatif terbatas, sehingga memperluas akses terhadap penelitian LLM (*Large Language Model*).

Dari sisi arsitektur, LLaMA (*Large Language Model Meta AI*) merupakan model *decoder-only Transformer*, artinya hanya terdiri dari lapisan-lapisan *decoder* yang bekerja secara autoregresif untuk menghasilkan token berikutnya dalam sebuah urutan teks. Meskipun berbasis pada desain *Transformer* klasik (Vaswani dkk., 2017), LLaMA (*Large Language Model Meta AI*) memperkenalkan sejumlah

modifikasi penting yang meningkatkan efisiensi sekaligus kualitas representasi bahasa. Pertama, LLaMA (*Large Language Model Meta AI*) menggunakan *Rotary Positional Encodings* (RoPE) yang lebih fleksibel dalam mengatasi dependensi jarak jauh dibandingkan positional encoding tradisional. Kedua, pada mekanisme atensinya, LLaMA (*Large Language Model Meta AI*) menerapkan *Grouped-Query Attention* (GQA) yang memungkinkan beberapa *query* berbagi *key* dan *value*, sehingga mempercepat inferensi sekaligus mengurangi kebutuhan memori. Ketiga, LLaMA (*Large Language Model Meta AI*) mengganti *Layer Normalization* dengan RMSNorm (*Root Mean Square Normalization*) yang lebih sederhana namun stabil untuk pelatihan skala besar. Selain itu, pada jaringan feed forward, LLaMA (*Large Language Model Meta AI*) mengadopsi SwiGLU sebagai fungsi aktivasi, yang terbukti lebih unggul dibandingkan ReLU (*Rectified Linear Activation Unit*) atau GELU (*Gaussian Error Linear Unit*) dalam menjaga ekspresivitas model (Shazeer, 2020).

Gambar berikut menampilkan arsitektur dasar LLaMA (*Large Language Model Meta AI*):





Gambar 2. 2 Gambar Arsitektur LLAMA

LLaMA (*Large Language Model Meta AI*) memiliki berbagai kegunaan terutama dalam pemahaman bahasa alami (NLP) dan generasi teks. Beberapa aplikasi utamanya meliputi menjawab pertanyaan, meringkas dokumen, penerjemahan bahasa, penyusunan konten kreatif, serta analisis teks yang memerlukan sensitivitas konteks budaya maupun linguistik. Karena efisiensi komputasinya, LLaMA (*Large Language Model Meta AI*) juga sangat relevan untuk digunakan dalam riset dan aplikasi industri yang tidak memiliki infrastruktur superkomputer.

Kelebihan utama LLaMA (*Large Language Model Meta AI*) adalah efisiensi parameter yang tinggi, sehingga model dengan jumlah parameter lebih sedikit mampu menyaingi atau melampaui performa model lain yang lebih besar. Selain itu, desainnya yang terbuka memungkinkan para peneliti untuk melakukan fine-tuning dengan metode hemat sumber daya, seperti Low-Rank Adaptation (LoRA), tanpa harus melatih ulang seluruh model. Namun, kekurangannya adalah keterbatasan dalam menangani multimodalitas (misalnya teks + gambar), karena LLaMA (*Large Language Model Meta AI*) pada dasarnya hanya dilatih untuk pemrosesan teks. Selain itu, meskipun relatif lebih ringan dibandingkan GPT-3 atau PaLM, model LLaMA (*Large Language Model Meta AI*) terbesar tetap memerlukan sumber daya komputasi signifikan untuk pelatihan dari awal (Mao dkk, 2025).

Dalam hal tugas, LLaMA (*Large Language Model Meta AI*) terbukti kompetitif pada berbagai benchmark NLP. Model ini mampu menangani tugas generasi teks (seperti menulis esai, artikel, atau lirik), tugas pemahaman bahasa (seperti menjawab pertanyaan berbasis konteks, inferensi logis, atau analisis sentimen), serta tugas ekstraksi informasi dari teks panjang. Bahkan, dalam penelitian terbaru, LLaMA 3 menunjukkan peningkatan dalam *reasoning* (penalaran), *coding assistance*, serta analisis linguistik kompleks, menjadikannya sebagai kandidat kuat untuk riset yang berfokus pada interpretasi makna, termasuk dalam ranah budaya dan musik.

2.2.3 Interpretasi Makna Lirik Lagu

Interpretasi makna lirik lagu adalah sebuah proses analisis yang bertujuan untuk memahami pesan, baik yang disampaikan secara eksplisit maupun tersirat, oleh seorang penulis lagu. Proses ini tidak hanya terbatas pada pemahaman arti literal dari kata-kata, tetapi juga mencakup penggalian lapisan makna yang lebih dalam, seperti tema naratif, nuansa emosional, dan penggunaan gaya bahasa. Lirik, terutama dalam genre yang puitis, sering kali berfungsi sebagai karya sastra yang padat akan makna. Oleh karena itu, interpretasi yang komprehensif menuntut kemampuan untuk mengenali dan menafsirkan elemen-elemen artistik seperti citraan (*imagery*), alusi, dan struktur penceritaan yang dibangun oleh sang artis di sepanjang lagu (Kahyun dan Choi, n.d.).

Tantangan dalam interpretasi makna menjadi semakin signifikan ketika berhadapan dengan genre musik yang kompleks seperti hip-hop. Lirik dalam genre ini sering kali penuh dengan metafora, permainan kata (*wordplay*) yang cerdas, dan referensi budaya yang kaya. Musisi hip-hop kerap menggunakan bahasa kiasan, *double entendres* (kata atau frasa dengan dua makna), dan sosiolek (bahasa khas kelompok sosial tertentu) yang mungkin tidak dapat dipahami tanpa pengetahuan konteks sosial, historis, atau budaya yang relevan. Oleh karena itu, untuk dapat menafsirkan lirik hip-hop secara akurat, dibutuhkan sebuah sistem yang tidak hanya mampu memproses bahasa, tetapi juga mampu menghubungkan kata-kata dengan konteks sosial dan simbolik secara mendalam (Chu dkk, 2022).

Secara komputasional, pendekatan awal untuk menganalisis lirik sering kali terbatas pada metode berbasis kata kunci atau analisis sentimen sederhana yang gagal menangkap kedalaman makna. Namun, seiring berkembangnya model bahasa, pendekatan yang lebih canggih mulai muncul. Salah satu contoh yang relevan adalah penelitian TRBLLmaker (*Transformer Reads Between Lyrics Lines maker*), yang menggunakan model berbasis *Transformer* seperti GPT-2 untuk mengungkap makna tersembunyi dalam lirik hip-hop. Model semacam ini bekerja melalui mekanisme *decoding* dan analisis kontekstual untuk "membaca di antara baris lirik", yang merupakan sebuah lompatan besar dari metode-metode sebelumnya. Kemampuan ini menunjukkan potensi AI untuk tidak hanya memproses teks, tetapi juga menafsirkan niat artistik di baliknya (Ventura dan Toker, 2022).

Di era *Large Language Models* (LLM) modern seperti LLaMA 3, kapabilitas untuk interpretasi lirik telah mencapai tingkat yang lebih tinggi. Model-model ini, yang dilatih pada data dalam skala masif, memiliki pemahaman yang lebih luas tentang referensi budaya, ekspresi idiomatik, dan pola bahasa yang kompleks. Dengan menerapkan teknik *fine-tuning* yang efisien seperti LoRA (*Low-Rank Adaptation*), sensitivitas model terhadap elemen-elemen khas genre hip-hop dapat ditingkatkan lebih lanjut. Selain itu, LLM (*Large Language Models*) modern tidak hanya menafsirkan baris per baris, tetapi juga mampu mengidentifikasi tema utama yang berulang dalam sebuah lagu atau bahkan album. Kemampuan ini

memungkinkan pemetaan makna lirik terhadap konteks sosial dan budaya yang lebih luas, sehingga menghasilkan interpretasi yang lebih holistik dan mendalam (Agrawal dkk, 2021).

2.2.4 Drake



Gambar 2. 3 Foto Drake

Drake adalah seorang musisi dan penulis lagu asal Kanada yang telah menjadi salah satu tokoh paling berpengaruh dalam industri musik hip-hop dan R&B kontemporer. Ia dikenal karena gaya penulisan lirik yang emosional, reflektif, dan penuh metafora, yang sering mencerminkan pengalaman pribadi, dinamika hubungan sosial, dan tekanan dari ketenaran.

Sejak debut albumnya *Thank Me Later* (2010), Drake terus menorehkan pencapaian komersial dan kritik yang mengesankan. Album *Take Care* (2011) memenangkan *Grammy Awards* untuk kategori *Best Rap Album*. Beberapa albumnya seperti *Views* (2016), *Scorpion* (2018), dan *Certified Lover Boy* (2021) memuncaki tangga lagu *Billboard 200* dan menghasilkan banyak *single* yang menempati posisi nomor satu di *Billboard Hot 100*. Hingga tahun 2024, Drake tercatat sebagai artis dengan jumlah entri terbanyak dalam sejarah *Billboard Hot 100*, serta *streaming* terbanyak di Spotify secara global.

Dalam penelitian ini, digunakan kumpulan lirik dari beberapa album utama Drake yang mencerminkan perkembangan gaya penulisannya dari tahun ke tahun. Album *Take Care* (2011) menjadi tonggak awal dari citra introspektif Drake, dengan lirik-lirik yang membahas refleksi diri, cinta, dan kesepian. *Nothing Was the Same* (2013) memperlihatkan kedewasaan emosional dan fokus pada identitas diri serta dinamika hubungan sosial. *Views* (2016) membawa nuansa kota asalnya,

Toronto, ke dalam lirik yang banyak menggunakan simbol lokal dan nada melankolis. Pada *Scorpion* (2018), Drake menyampaikan isu-isu personal seperti menjadi ayah, relasi publik, dan dualitas antara kehidupan pribadi dan citra publik, lewat format album ganda. *Certified Lover Boy* (2021) menghadirkan eksplorasi tema cinta, ego, dan kompleksitas hubungan modern dengan penggunaan bahasa yang lebih metaforis dan simbolik.

Album *Honestly, Nevermind* (2022) memperlihatkan pergeseran gaya menuju pengaruh *dance* dan *house*, namun tetap memuat lirik-lirik kontemplatif yang khas. Disusul oleh *Her Loss* (2022) yang merupakan kolaborasi dengan 21 Savage, Drake menampilkan sisi liris yang lebih tajam, sarkastik, dan penuh permainan kata. Album *For All the Dogs* (2023) membawa nuansa emosional kembali, menyajikan lirik-lirik yang lebih personal, sentimental, dan penuh kode simbolik. Album terbaru, *Some Sexy Songs 4 U* (2025), menunjukkan fase baru dalam kariernya dengan fokus pada tema romantis dewasa dan emosi halus, serta penggunaan bahasa yang lebih intim dan reflektif.

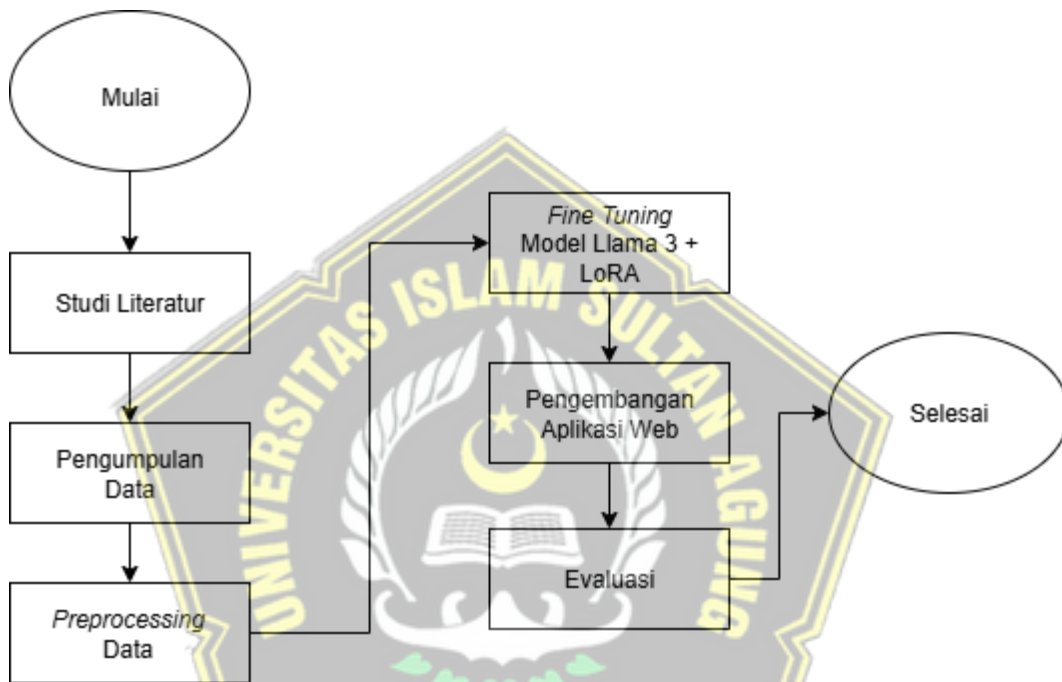
Pendekatan interpretasi makna lirik dalam konteks musik Drake menjadi tantangan tersendiri, karena banyak liriknya yang tidak eksplisit secara semantik, tetapi sangat bergantung pada konteks sosial, sejarah pribadi, atau referensi budaya populer. Oleh karena itu, dibutuhkan model bahasa yang tidak hanya mampu memahami teks secara literal, tetapi juga mampu menangkap nuansa, gaya bahasa, dan simbolisme yang digunakan dalam hip-hop.

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Pada penelitian ini terdapat beberapa tahapan yang akan dilaksanakan untuk dapat membuat generator makna lirik lagu secara otomatis. Alur tahapan penelitian dapat dilihat pada gambar dibawah ini



Gambar 3. 1 *Flowchart* Rancangan Alur Penelitian

a. Studi Literatur

Peneliti melakukan studi literatur untuk memahami konsep dasar pemrosesan bahasa alami (NLP), model *Transformer*, arsitektur LLaMA (*Large Language Model Meta AI*), serta teknik *fine-tuning* seperti LoRA (*Low-Rank Adaptation*). Literatur ini juga mencakup penelitian sebelumnya terkait interpretasi makna lirik lagu, terutama pada genre hip-hop yang dikenal kaya akan metafora, permainan kata, dan referensi budaya. Studi ini menjadi dasar dalam merancang sistem yang mampu memahami makna dalam lirik lagu Drake.

b. Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa lirik lagu dari artis hip-hop Drake, beserta judul lagu, nama album, interpretasi makna lirik, dan

sumber referensi makna. Sumber data dikumpulkan dari berbagai situs terpercaya seperti Genius.com, Pitchfork, wawancara resmi dengan Drake, serta media musik lainnya. Dataset ini mencakup lagu-lagu dari berbagai album Drake, mulai dari *More Life* hingga *Dark Lane Demo Tapes*.

c. *Preprocessing Data*

Sebelum digunakan dalam proses pelatihan model, data lirik lagu terlebih dahulu melalui tahapan *preprocessing* untuk memastikan formatnya sesuai dengan kebutuhan pemrosesan bahasa alami (NLP) dan model pembelajaran mesin. Proses ini mencakup beberapa langkah penting, yaitu normalisasi teks untuk mengubah huruf kapital menjadi huruf kecil serta menghapus spasi berlebih, kemudian tokenisasi untuk memecah teks lirik menjadi unit kata atau frasa agar lebih mudah dianalisis oleh model. Selain itu, dilakukan juga penghapusan karakter khusus, seperti simbol, angka, dan tanda baca yang tidak memiliki nilai semantik dalam konteks pemahaman makna lirik. Tahapan-tahapan ini penting untuk menghasilkan input yang bersih, seragam, dan dapat diproses secara optimal oleh model LLaMA 3 yang akan dilatih.

d. *Fine-tuning Model LLaMA 3 + LoRA*

Model LLaMA 3 yang telah dilatih sebelumnya (*pre-trained*) disesuaikan (*fine-tuned*) menggunakan teknik LoRA (*Low-Rank Adaptation*) agar efisien dalam penggunaan sumber daya. Pelatihan dilakukan menggunakan dataset yang telah diproses untuk menghubungkan lirik lagu dengan maknanya.

e. Pengembangan Aplikasi Web Menggunakan Gradio di Google Colab

Tahap pengembangan aplikasi web difokuskan pada implementasi antarmuka yang responsif dan efisien. Aplikasi ini dikembangkan menggunakan *framework* Gradio, yang menyediakan antarmuka pengguna (UI) yang sederhana dan mendukung integrasi cepat. Untuk mengatasi kendala komputasi dan memastikan performa inferensi yang optimal, aplikasi ini dijalankan di lingkungan *cloud* Google Colab yang menyediakan akses akselerasi GPU. Pendekatan ini berhasil mereduksi waktu pemrosesan makna

lirik menjadi kurang dari 60 detik. Sistem ini dirancang dengan arsitektur *Retrieval-Augmented Generation* (RAG). Alur kerjanya dimulai saat pengguna memasukkan lirik; sistem kemudian menjalankan fase *retrieval* untuk mengambil konteks lagu yang relevan, yang kemudian digunakan oleh model LLaMA 3 yang telah di-*fine-tune* untuk menghasilkan interpretasi makna secara otomatis.

f. Evaluasi Sistem

Evaluasi dilakukan untuk mengukur sejauh mana model LLaMA 3 yang telah di-*fine-tune* mampu menghasilkan interpretasi makna lirik lagu yang memiliki kemiripan dengan makna referensi yang tersedia dalam dataset. Karena sistem ini bersifat generatif dan *output* yang dihasilkan berupa teks naratif bebas struktur (tidak berbentuk label klasifikasi), maka metrik evaluasi yang digunakan harus mampu menilai kesamaan makna, bukan hanya kemiripan struktur kata.

Metrik pertama berfokus pada metode evaluasi berbasis *semantic similarity*, yaitu pengukuran kesamaan makna antar dua teks dengan pendekatan vektor. Evaluasi dilakukan menggunakan model *pre-trained* dari *library sentence-transformers*, yaitu all-MiniLM-L6-v2, yang mampu mengubah teks (kalimat atau paragraf) menjadi representasi vektor berdimensi tinggi (*embedding*). Dua teks yang memiliki makna yang sama atau mirip akan menghasilkan vektor yang saling mendekati dalam ruang vektor.

Perhitungan similarity dilakukan dengan rumus *Cosine Similarity*, yang secara matematis dituliskan sebagai berikut:

$$\text{Similarity}(A, B) = \frac{A \cdot B}{||A|| \times ||B||}$$

Keterangan:

- A dan B adalah vektor *embedding* dari dua teks yang dibandingkan, yaitu makna hasil prediksi model dan makna referensi dari dataset.
- $A \cdot B$ adalah dot *product* antara kedua vector.
- $||A||$ dan $||B||$ adalah panjang (norma) dari masing-masing vektor.

Nilai *cosine similarity* berada dalam rentang 0 hingga 1, di mana:

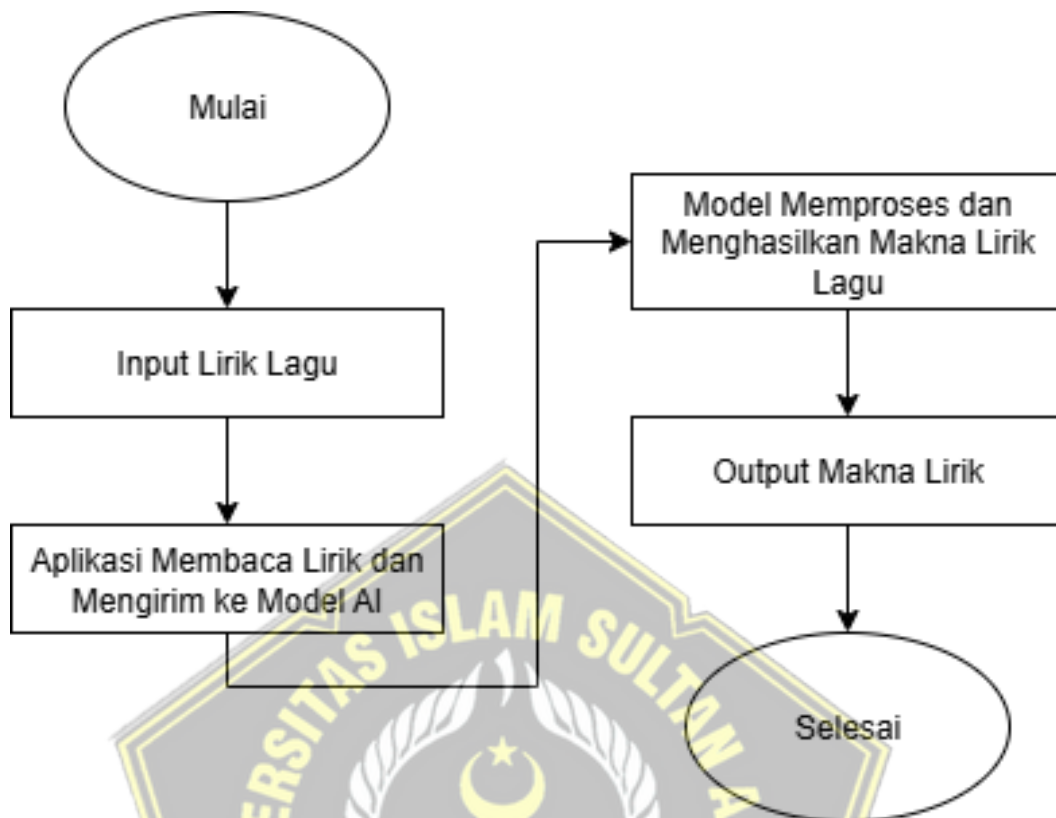
- Nilai 0 menunjukkan bahwa dua teks tidak memiliki kesamaan makna.
- Nilai 1 menunjukkan bahwa dua teks sangat mirip atau identik secara semantik.

Sebagai pelengkap, metrik kedua yaitu ROUGE-L (*Longest Common Subsequence*) digunakan untuk mengevaluasi kesamaan struktural dan leksikal. ROUGE-L menilai kualitas generasi teks dengan mengukur panjang urutan kata terpanjang (LCS) yang sama antara teks yang dihasilkan (kandidat) dan teks referensi. Skor ini membantu memastikan bahwa model tidak hanya menangkap makna yang benar, tetapi juga mampu menyusun kalimat dengan pilihan kata dan struktur yang mirip dengan referensi.

Dengan demikian, penggunaan kombinasi *semantic similarity* dan ROUGE-L ini akan mencerminkan seberapa baik model dapat memahami, menafsirkan, dan menyajikan kembali lirik lagu dengan konteks yang mendekati makna aslinya.

3.2 Rancang Alur Penggunaan Aplikasi

Tahapan perancangan model dalam penelitian ini mengikuti alur yang sistematis untuk memastikan model dapat menghasilkan interpretasi makna lirik lagu secara optimal. Adapun tahapan yang dilakukan meliputi:



Gambar 3. 2 *Flowchart* Alur Penggunaan Aplikasi

Alur penggunaan aplikasi ini dirancang agar sederhana dan intuitif bagi pengguna. Proses dimulai ketika pengguna memasukkan potongan lirik lagu yang ingin diketahui maknanya ke dalam kolom *input* yang tersedia pada antarmuka aplikasi. Setelah lirik dimasukkan, aplikasi akan membaca teks tersebut dan secara otomatis mengirimkannya ke model AI yang menjadi inti dari sistem ini. Di sinilah proses analisis berlangsung. Model AI kemudian akan memproses dan menganalisis lirik yang diterima untuk menghasilkan interpretasi makna yang relevan dan kontekstual. Terakhir, hasil dari analisis model tersebut akan menjadi *output* berupa penjelasan makna lirik, yang kemudian ditampilkan kembali kepada pengguna melalui antarmuka aplikasi. Dengan tampilnya makna lirik tersebut, alur proses pun selesai, dan pengguna berhasil mendapatkan pemahaman yang lebih dalam mengenai lagu yang mereka cari.

3.3 Perancangan *User Interface*

3.3.1 Halaman Awal Sistem

Gambar di bawah ini menunjukkan *wireframe* dari antarmuka pengguna untuk aplikasi generator makna lirik lagu Drake. Desain ini berfokus pada halaman awal yang intuitif, yang bertujuan untuk memberikan kemudahan bagi pengguna saat memasukkan lirik sebagai langkah pertama interaksi dengan sistem.

Generator Makna Lirik Drake

Masukkan Lirik Lagu di Sini



Gambar 3. 3 Tampilan Rancangan Halaman *Input* Pengguna

Rancangan tampilan *input* pengguna, seperti yang diilustrasikan pada gambar 3.3, dirancang sebagai halaman utama yang menjadi titik interaksi pertama bagi pengguna. Desain ini sengaja dibuat minimalis untuk memfokuskan pengguna pada fungsi utamanya. Di bagian atas, judul aplikasi "Generator Makna Lirik Drake" memberikan identitas yang jelas, yang kemudian diikuti oleh label instruksional "Masukkan Lirik Lagu di Sini" untuk memandu pengguna. Komponen sentralnya adalah sebuah area teks yang luas untuk memberikan ruang yang cukup saat memasukkan lirik, dan tepat di bawahnya terdapat tombol aksi "Hasilkan Makna" yang berfungsi sebagai pemicu utama untuk memulai proses analisis. Tata letak ini secara keseluruhan bertujuan untuk menciptakan proses input yang sederhana dan terfokus.

3.3.2 Tampilan Hasil Sistem

Hasil Analisis

Judul Lagu :

Album :

Sumber :

Makna :

Hasil analisis makna

Evaluasi Semantik

Skor :

Makna Refrensi (Dataset) :

Makna dari Dataset

Gambar 3. 4 Tampilan Rancangan *Output* Hasil

Setelah pengguna menekan tombol aksi, sistem akan menampilkan hasil analisis yang rancangannya diilustrasikan pada gambar 3.4. Tampilan *output* ini sengaja dibagi menjadi dua bagian logis untuk meningkatkan kejelasan dan keterbacaan informasi. Bagian pertama, "Hasil Analisis", menyajikan konteks lagu seperti Judul Lagu, Album, dan Sumber, serta Makna yang merupakan hasil interpretasi dari model AI. Sementara itu, bagian kedua, "Evaluasi Semantik", berfungsi untuk tujuan transparansi dan validasi, di mana sistem menampilkan skor kemiripan semantik beserta Makna Referensi dari dataset agar pengguna dapat secara objektif menilai kualitas keluaran model.

3.4 Perancangan Prompt Sistem

Perancangan *prompt* (*prompt engineering*) adalah tahap krusial dalam pengembangan sistem ini, karena berfungsi sebagai jembatan instruksi antara data yang relevan dan *Large Language Model* (LLM). Kualitas dan konsistensi keluaran

model sangat bergantung pada seberapa baik *prompt* dirancang. Tujuannya adalah untuk membuat serangkaian instruksi yang tidak hanya jelas tetapi juga mampu mengarahkan model untuk menghasilkan analisis lirik yang mendalam dan terstruktur.

Desain *prompt* untuk sistem ini didasarkan pada beberapa prinsip strategis:

1. Kejelasan Tugas (*Task Clarity*): Secara eksplisit mendefinisikan peran yang harus diambil oleh model dan tugas yang harus diselesaikan.
2. Penyediaan Konteks (*Context Provisioning*): Menyuntikkan informasi dinamis yang relevan (diperoleh dari tahap *retrieval*) untuk membantu model menghasilkan analisis yang akurat.
3. Instruksi Format (*Output Formatting*): Memberikan panduan yang jelas tentang struktur dan gaya bahasa dari teks yang diharapkan.
4. Pemberian Contoh (*One-Shot Prompting*): Menyertakan satu contoh konkret untuk "mengajari" model secara langsung bagaimana cara melakukan tugas sesuai format yang diinginkan.

Berdasarkan prinsip-prinsip tersebut, *prompt* disusun dengan beberapa komponen utama yang digabungkan secara dinamis menggunakan F-string pada Python. Dekomposisi struktur dan fungsi setiap komponen dapat dilihat pada tabel berikut.

Tabel 3. 1Komponen dan Fungsi dalam Desain *Prompt*

Definisi Tugas	Tugas: Analisis makna dari lirik lagu Drake berikut ini.	Secara langsung memerintahkan model untuk berperan sebagai analis lirik. Ini menghilangkan ambiguitas dan memfokuskan model pada tujuan utama.
Konteks Dinamis	- Judul: {title} - Album: {album} - Lirik: "{lyrics}"	Menyuntikkan data yang diambil dari tahap sebelumnya. Judul dan album memberikan konteks artistik, sementara kutipan lirik menjadi subjek utama analisis.

Instruksi Format & Contoh	Berikan penjelasan mendalam dengan cara mengutip langsung... Contoh: "Pada lirik '[kutipan lirik]', Drake..."	Memberikan instruksi eksplisit tentang format keluaran dan menyertakan sebuah contoh <i>one-shot</i> untuk melatih model <i>on-the-fly</i> agar mengikuti gaya analisis yang diinginkan.
Titik Awal Generasi	Mulai analisis Anda di bawah. Makna:	Berfungsi sebagai <i>anchor</i> atau pemicu bagi model untuk memulai generasi, sekaligus memudahkan proses <i>parsing</i> untuk memisahkan hasil dari <i>template</i> .

Pada tabel 3.1 merupakan tampilan dari *prompt* yang dirancang sebagai serangkaian instruksi strategis untuk memandu model AI secara efektif. Awalnya, *prompt* secara eksplisit mendefinisikan apa tugasnya, yaitu untuk menganalisis makna lirik. Kemudian, *prompt* menyediakan konteks yang diperlukan untuk analisis, berupa judul lagu, album, dan potongan lirik yang relevan. Model juga diberi tahu bagaimana format jawaban yang benar melalui instruksi untuk mengutip lirik dan sebuah contoh konkret yang harus diikuti. Terakhir, sebuah pemicu yang jelas diberikan untuk memberitahu model di mana harus mulai menjawab, yaitu tepat setelah kata kunci "Makna:". Kombinasi instruksi yang lengkap ini memastikan model bekerja secara terarah untuk menghasilkan analisis yang terstruktur dan sesuai harapan.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Penelitian

4.1.1 Pengumpulan Data

Pada tahap pengumpulan data, penelitian ini menggunakan data lirik lagu dari artis hip-hop Drake, yang mencakup judul lagu, nama album, lirik lagu, interpretasi makna lirik, dan sumber referensi makna. Data dikumpulkan dari berbagai situs terpercaya seperti Genius.com, Pitchfork, wawancara resmi dengan Drake, serta media musik lainnya. Dataset ini mencakup lagu-lagu dari album Drake, mulai dari *More Life* hingga *Dark Lane Demo Tapes*. Total data yang terkumpul adalah 514 baris data, dengan setiap baris merepresentasikan bait lirik beserta maknanya. Berikut data dari dataset yang terdapat pada tabel 4.1

Tabel 4. 1 *Dataset* Lirik Lagu dan Makna Drake

album	song_title	lyric_bait	meaning_bait	source
<i>Dark Lane Demo Tapes</i>	<i>Deep Pockets</i>	<i>Dressed in fatigues, I rep the East with my trustees Smokin' on crushed leave ...</i>	lirik ini menggambarkan perjalanan Drake dari masa kecil yang penuh tantangan hingga menjadi sosok ...	LyricsMeanings.com
<i>Dark Lane Demo Tapes</i>	<i>Deep Pockets</i>	<i>Look Back when Big Apple sold dreams, I stuck to my own thing Back when the house</i>	Lirik ini menggambarkan perjalanan Drake dari masa-masa awal kariernya di Toronto, menekankan bagaimana ia	<i>Drake Interview</i>

		<i>that I own now was ...</i>	memilih jalannya ...	
<i>Dark Lane Demo Tapes</i>	<i>When To Say When</i>	<i>Yeah Somebody, um Somebody hit me today and they were like "Nah, just make sure, ..."</i>	Bait ini mencerminkan sikap dermawan dan keterbukaan Drake terhadap orang-orang di sekitarnya. Ia menekankan bahwa berkat yang ...	Genius.com
<i>Dark Lane Demo Tapes</i>	<i>When To Say When</i>	<i>Ayy Thirty-three years, I gave that to the game Thirty-three mil', ...</i>	Bait ini menampilkan refleksi mendalam Drake tentang perjalanan kariernya ...	Pitchfork

Pada tabel 4.1 merupakan tampilan dari dataset yang telah terkumpul diorganisir dalam format tabel. Tabel ini menampilkan struktur dataset yang digunakan, yang memuat informasi mengenai nama album (*album*), judul lagu (*song_title*), potongan lirik atau frasa kunci yang akan diinterpretasikan maknanya (*lyric_bait*), interpretasi makna dari potongan lirik tersebut (*meaning_bait*), dan sumber asli dari lirik atau interpretasi makna (*source*). Sebagai contoh, pada baris pertama terlihat lirik "*Dressed in fatigues, I rep the East with my trust*" dari album "*Dark Lane Demo Tapes*" dan lagu "*Deep Pockets*", yang memiliki interpretasi makna "Drake ingin menggambarkan perjalanan" dengan sumber dari "LyricsMeanings.com". Contoh lain menunjukkan lirik dari lagu "*When To Say When*" dan "*Chicago Freestyle (Ft. Giveon)*", yang juga disertai

dengan interpretasi makna dan sumbernya, seperti Genius.com. Total lirik/frasa yang ditampilkan dalam contoh dataset ini adalah 5 baris, namun dataset tersebut sebenarnya berisi 514 baris.

4.2 Implementasi sistem Deteksi

4.2.1 Pra-pemrosesan Data

Setelah data terkumpul dalam format tabel, proses pra-pemrosesan dilakukan untuk memastikan bahwa data siap digunakan dalam pelatihan model *Natural Language Processing* (NLP). Langkah ini penting untuk menghasilkan *input* yang bersih, konsisten, dan dapat diproses dengan optimal oleh model LLaMA 3 yang digunakan dalam penelitian ini. Pra-pemrosesan dimulai dengan membaca *dataset* dari *file* CSV menggunakan pustaka *Pandas*. Data yang berisi potongan lirik lagu Drake, makna lirik, judul lagu, nama album, dan sumber referensi kemudian diformat ulang menjadi pasangan *prompt* dan *response* yang sesuai dengan format instruksi pada model bahasa besar (LLM).

Dalam proses ini, dilakukan pembersihan dasar seperti penghapusan spasi berlebih pada awal dan akhir teks dengan menggunakan metode `strip()`. Setiap baris data diproses untuk mengambil nilai dari kolom `lyric_bait`, `meaning_bait`, `song_title`, `album`, dan `source`. Nilai-nilai tersebut kemudian disusun dalam struktur yang menggabungkan bagian lirik, makna, serta informasi tambahan seperti judul lagu dan nama album ke dalam bagian *prompt*, sedangkan bagian *response* berisi makna dari lirik tersebut. Penyusunan *prompt* yang kaya konteks ini bertujuan untuk membantu model memahami hubungan antara lirik dan maknanya secara lebih baik.

Tabel 4. 2 Tampilan Sebelum dan Sesudah Data *Cleaning*

Sebelum	Sesudah
<i>First time in a long time</i>	<i>first time in a long time</i>
<i>Hurtin' deeply inside</i>	<i>hurtin deeply inside</i>
<i>First time in a long time</i>	<i>first time in a long time</i>
<i>Hurtin' deeply</i>	<i>hurtin deeply</i>

Pada tabel 2.4 merupakan tampilan sebelum dan sesudah dari salah satu proses *preprocessing* yakni pada pembersihan data, yang menghapus karater

husus atau *punctuation removal* dan mengubah huruf kapital menjadi huruf kecil atau proses *lowercase*.

Setelah proses pemformatan selesai, data disimpan dalam format JSONL (*JavaScript Object Notation Lines*) dengan nama file `drake_formatted_dataset_from_csv_new.jsonl` yang ditempatkan di direktori `/content/`. Format JSONL (*JavaScript Object Notation Lines*) dipilih karena mendukung efisiensi dalam pembacaan data oleh pustaka *datasets* dari *Hugging Face* yang akan digunakan pada tahap pelatihan. Proses penyimpanan ini berhasil menghasilkan total 513 entri data yang telah diformat dan siap untuk digunakan.

Tabel 4. 3 Tampilan Sebelum dan Sesudah Diubah ke JSONL

Sebelum					Sesudah
album	song_title	lyric_bait	meaning_bait	Source	JSONL (<i>JavaScript Object Notation Lines</i>)
Dark Lane Demo Tapes	Not Too (Ft. Chris Brown)	first time in a long time hurtin' deeply inside first time in a long time hurtin' deeply	Bait ini menggambarkan perasaan sakit hati yang mendalam dan mengejutkan setelah sekian lama tidak merasakannya, menandakan bahwa pengkhianatan atau kehilangan dalam hubungan tersebut sangat membekas secara emosional bagi Drake.	Genius.com	{"lyrics": "First time in a long time\nHurtin' deeply inside\nFirst time in a long time\nHurtin' deeply", "meaning": "Bait ini menggambarkan perasaan sakit hati yang mendalam dan mengejutkan setelah sekian lama tidak merasakannya, menandakan bahwa pengkhianatan atau kehilangan

					dalam hubungan tersebut sangat membekas secara emosional bagi Drake.", "title": "Not You Too (Ft. Chris Brown)", "album": "Dark Lane Demo Tapes", "source": "genius.com"}
--	--	--	--	--	---

Pada tabel 4.3 menampilkan sebelum dan sesudah proses mengubah format dari csv ke JSONL. Yang dimana pada dataset tersebut sebelumnya pada *file* csv tampilannya untuk bagian album, *song_title*, *lyric_bait*, *meaning_bait*, *meaning* terpisah dalam setiap kolom tabel namun ketika sudah diubah menjadi JSONL maka isinya berubah menjadi satu baris yang berisikan lirik lagu, makna, judul lagu, album, dan makna. Tujuan utama konversi data ke format JSONL adalah untuk menstrukturkan dataset menjadi format pasangan *prompt-response* yang esensial untuk metode pelatihan *instruction fine-tuning*. Selain itu, format ini dipilih karena sangat efisien untuk diproses oleh pustaka modern seperti *Hugging Face datasets*, sehingga mempercepat persiapan data sebelum pelatihan model.

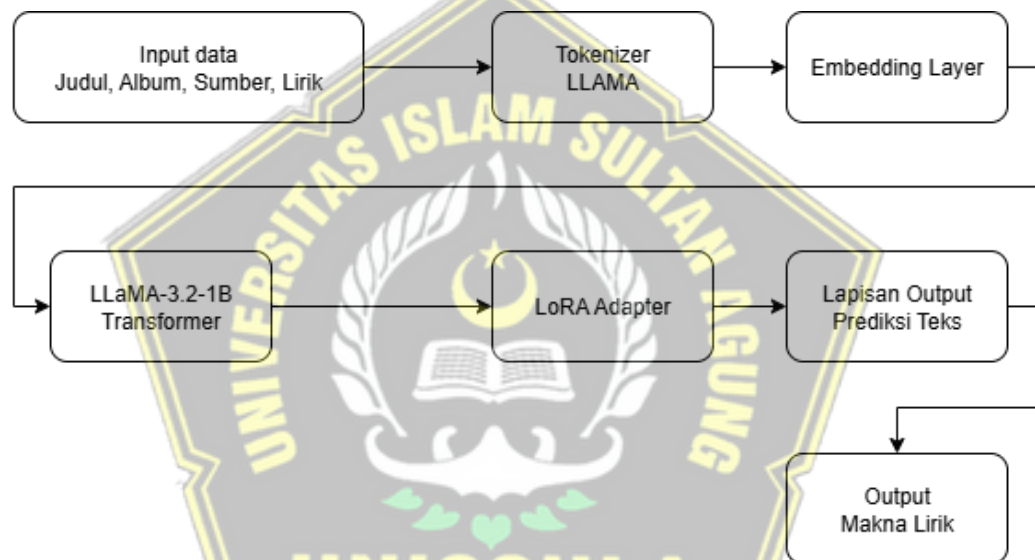
Dataset yang telah diformat dalam JSONL (*JavaScript Object Notation Lines*) kemudian dimuat kembali sebagai objek `datasets.Dataset` dan dilanjutkan ke tahap tokenisasi. Tokenisasi adalah proses penting di mana teks diubah menjadi token atau representasi numerik (*input_ids*) agar dapat diproses oleh model *Transformer*. Pada tahap ini, bagian *prompt* dan *response* dari masing-masing entri digabung menjadi satu kolom teks, lalu diproses menggunakan *AutoTokenizer* dari model `meta-llama/Llama-3.2-1B-Instruct`. *Tokenizer* ini secara otomatis menangani normalisasi teks seperti pengubahan huruf kapital menjadi huruf kecil dan penghapusan karakter khusus yang tidak diperlukan.

Tokenisasi dilakukan dengan parameter `truncation=True` untuk memotong teks yang terlalu panjang, serta `padding="max_length"` untuk memastikan bahwa semua *input* memiliki panjang tetap, yaitu maksimal 512 token. Selain

menghasilkan *input_ids*, proses ini juga membuat salinan sebagai label untuk keperluan pelatihan model. Hasil akhir dari tahap ini adalah *dataset* yang telah ditokenisasi sepenuhnya, terdiri dari 513 entri yang siap digunakan dalam proses *fine-tuning* model LLaMA.

4.2.2 Arsitektur Model

Arsitektur model yang digunakan dalam penelitian ini ditunjukkan pada Gambar 4.1. Diagram ini menggambarkan bagaimana data masukan diproses oleh tokenizer, embedding, hingga melalui LLaMA yang telah dilengkapi dengan LoRA adapter untuk menghasilkan keluaran berupa makna lirik.



Gambar 4. 1 Diagram Arsitektur Model

Model yang diterapkan pada penelitian ini menggunakan LLaMA-3.2-1B-*Instruct* sebagai pondasi utama. Data masukan berupa judul lagu, album, sumber, dan lirik terlebih dahulu diproses oleh *tokenizer*, yang mengubah teks menjadi deretan token. Token-token tersebut kemudian dikonversi menjadi representasi vektor melalui *embedding layer* sebelum diproses lebih lanjut oleh mekanisme transformer LLaMA. Agar proses pelatihan menjadi lebih efisien, ditambahkan LoRA Adapter pada bagian *self-attention*, khususnya di proyeksi *query* dan *value*. Dengan pendekatan ini, hanya sejumlah kecil parameter yang dilatih sehingga memungkinkan *fine-tuning* berjalan lebih hemat sumber daya namun tetap menghasilkan model yang adaptif. Keluaran dari transformer yang telah diperkuat dengan LoRA kemudian diarahkan ke lapisan *output*, yang berfungsi

menerjemahkan representasi internal model menjadi prediksi token berbentuk teks. Tahap inilah yang akhirnya menghasilkan keluaran berupa interpretasi makna lirik sesuai dengan data latih yang tersedia.

4.2.3 Pengembangan dan *Fine-tuning* Model LLaMA 3 + LoRA

Model inti yang digunakan dalam penelitian ini adalah LLaMA-3.2-1B-*Instruct* yang dikembangkan oleh Meta, yang merupakan salah satu varian terbaru dari arsitektur *Transformer* dengan kapabilitas instruksional. Model ini dipilih karena kemampuannya dalam memahami dan menghasilkan teks secara kontekstual, yang sesuai dengan kebutuhan tugas interpretasi makna lirik lagu. Untuk mengoptimalkan penggunaan sumber daya dan memungkinkan pelatihan dilakukan pada perangkat keras dengan kapasitas memori terbatas, model dimuat menggunakan teknik kuantisasi dari modul *BitsAndBytesConfig*.

Tabel 4. 4 Tabel Konfigurasi Fine Tuning

<i>Hyperparameter</i>	Nilai (<i>Value</i>)	Deskripsi Singkat
num_train_epochs	3	Jumlah total siklus pelatihan pada dataset.
per_device_train_batch_size	1	Jumlah data yang diproses per perangkat dalam satu langkah.
gradient_accumulation_steps	2	Jumlah langkah untuk mengakumulasi gradien (simulasi <i>batch size</i> 2).
optim	paged_adamw_8bit	Optimizer yang hemat memori.
learning_rate	5e-5	Tingkat laju pembelajaran untuk optimizer.
bf16	True	Mengaktifkan <i>mixed-precision training</i> untuk efisiensi.

weight_decay	0.01	Nilai regularisasi untuk mencegah <i>overfitting</i> .
--------------	------	--

Pada Tabel 4.4 ditampilkan konfigurasi *hyperparameter* ini dirancang dengan cermat untuk mengatur inti dari proses pembelajaran model. Pemilihan `num_train_epochs` sebanyak 3 siklus merupakan langkah fundamental untuk memastikan model memiliki waktu yang cukup untuk mempelajari pola-pola unik dalam lirik Drake, tanpa risiko *overfitting* atau menghafal data latihan. Proses pembelajaran ini dikendalikan oleh `learning_rate` sebesar $5e-5$, sebuah nilai kecil yang krusial dalam konteks *fine-tuning*. Laju yang konservatif ini memungkinkan model untuk secara hati-hati mengadaptasi pengetahuannya yang sudah ada, membuat penyesuaian yang presisi tanpa merusak pemahaman bahasa yang kompleks yang telah dimilikinya. Untuk menjaga agar model tidak menjadi terlalu spesifik dan kehilangan kemampuan generalisasinya, diterapkan `weight_decay` sebesar 0.01. Teknik regularisasi ini memberikan penalti pada bobot model yang terlalu besar, mendorongnya untuk menemukan solusi yang lebih sederhana dan efektif saat dihadapkan pada lirik yang belum pernah dilihat sebelumnya.

Tantangan terbesar dalam *fine-tuning* model skala besar adalah keterbatasan sumber daya, terutama memori GPU. Konfigurasi ini mengatasi masalah tersebut melalui serangkaian teknik efisiensi yang cerdas. Penggunaan `per_device_train_batch_size` sebesar 1 adalah langkah yang diambil untuk meminimalkan jejak memori, meskipun berisiko menyebabkan pelatihan yang tidak stabil. Risiko ini dimitigasi secara efektif oleh `gradient_accumulation_steps` sebanyak 2 langkah, sebuah metode yang mensimulasikan *batch size* yang lebih besar dengan mengumpulkan gradien sebelum melakukan pembaruan bobot, sehingga menstabilkan pelatihan tanpa membebani VRAM. Efisiensi memori ini diperkuat lebih lanjut oleh *optimizer* `paged_adamw_8bit`, varian canggih yang secara dramatis mengurangi penggunaan memori dengan menyimpan statusnya dalam format 8-bit yang lebih ringkas.

Untuk melengkapi strategi efisiensi, kecepatan komputasi ditingkatkan secara signifikan dengan mengaktifkan `bf16=True`. Penggunaan *mixed precision*

ini memungkinkan sebagian besar operasi matematis yang berat dijalankan dalam format 16-bit yang lebih cepat, hampir menggandakan kecepatan pelatihan sambil tetap menjaga stabilitas melalui penggunaan presisi 32-bit pada komponen-komponen kritis. Secara keseluruhan, kombinasi dari durasi belajar yang terkontrol, penyesuaian yang hati-hati, dan serangkaian teknik efisiensi memori serta komputasi ini membentuk sebuah pendekatan holistik yang memungkinkan adaptasi model bahasa canggih untuk tugas analisis lirik yang spesifik, bahkan dengan sumber daya perangkat keras yang terbatas.

Tabel 4. 5 Tabel Konfigurasi LoRA

Konfigurasi LoRA		
<i>Hyperparameter</i>	Nilai (Value)	Deskripsi Singkat
R	32	Ukuran (rank) dari matriks adaptasi LoRA yang bisa dilatih.
lora_alpha	32	Faktor penskalaan untuk bobot LoRA.
target_modules	["q_proj", "k_proj", "v_proj", "o_proj"]	Lapisan spesifik pada model yang menjadi target adaptasi LoRA.
lora_dropout	0.15	Nilai <i>dropout</i> pada lapisan LoRA untuk mencegah <i>overfitting</i> .

Konfigurasi LoRA (*Low-Rank Adaptation*) yang disajikan pada Tabel 4.5 merupakan inti dari strategi *fine-tuning* yang efisien. LoRA bekerja dengan "membekukan" jutaan parameter asli dari model LLaMA dan hanya melatih beberapa matriks kecil ("adapter") yang disisipkan ke dalamnya. Ukuran dan pengaruh adapter ini dikendalikan oleh `rank (r)` dan `lora_alpha`. Nilai `r=32` yang dipilih memberikan kapasitas atau "ruang belajar" yang cukup besar bagi adapter untuk menangkap pola-pola linguistik yang kompleks, metafora, dan gaya puitis yang khas dalam lirik Drake. Ini adalah peningkatan signifikan yang memungkinkan adaptasi yang lebih mendalam. Sementara itu, `lora_alpha=32` berfungsi sebagai faktor skala. Dengan menjaga `lora_alpha` seimbang dengan `r`, kita memastikan bahwa kontribusi dari adapter baru ini cukup kuat untuk membuat

perbedaan, namun tidak terlalu dominan sehingga mengganggu pengetahuan dasar model yang sudah ada.

Kunci efektivitas LoRA terletak pada penempatan adapter di lokasi yang strategis. Pengaturan `target_modules` yang mencakup `q_proj`, `k_proj`, `v_proj`, dan `o_proj` adalah keputusan krusial yang menargetkan jantung dari arsitektur Transformer: mekanisme *self-attention*. Keempat modul ini adalah lapisan proyeksi untuk *Query* (apa yang dicari), *Key* (apa yang ditawarkan), *Value* (informasi yang dibawa), dan *Output* (hasil kombinasi). Dengan menyisipkan adapter ke semua komponen ini, kita memberikan kemampuan pada model untuk memodifikasi setiap aspek dari cara ia menimbang dan menghubungkan kata-kata dalam sebuah kalimat. Ini adalah pendekatan komprehensif yang memungkinkan adaptasi yang jauh lebih holistik dan efektif dibandingkan hanya menargetkan satu atau dua komponen saja, seumpama melakukan *tuning* pada seluruh bagian mesin, bukan hanya satu komponen.

Terakhir, untuk memastikan proses adaptasi ini kuat dan sesuai dengan tujuan, dua parameter final ditetapkan. `lora_dropout=0.15` adalah mekanisme regularisasi yang penting. Selama pelatihan, ia secara acak menonaktifkan 15% koneksi di dalam lapisan LoRA. Tindakan ini memaksa model untuk tidak terlalu bergantung pada jalur pembelajaran tertentu, sehingga mencegah *overfitting* dan meningkatkan kemampuannya untuk melakukan generalisasi pada lirik baru. Selain itu, `task_type` secara eksplisit didefinisikan sebagai `CAUSAL_LM` (*Causal Language Model*). Ini adalah instruksi teknis yang menegaskan bahwa tujuan akhir model adalah untuk memprediksi token berikutnya dalam sebuah urutan. Pengaturan ini memastikan bahwa seluruh arsitektur dan proses *fine-tuning* selaras dengan tugas utamanya, yaitu menghasilkan teks interpretasi makna yang koheren dan kontekstual.

4.2.4 Pengembangan Aplikasi Web Generator Makna Lirik Lagu

Model LLaMA 3 yang telah di-*fine-tune* dalam penelitian ini kemudian diintegrasikan ke dalam sebuah aplikasi web yang berfungsi sebagai generator makna lirik lagu. Aplikasi ini dirancang untuk dapat di-*deploy* dan diakses melalui

browser, dengan inti sistem yang berfokus pada eksekusi model dan proses *retrieval* data.

Inti dari *backend* aplikasi ini terletak pada dua fungsi utama, yaitu `load_llm_model()` dan `setup_retriever()`. Fungsi `load_llm_model()` bertugas untuk memuat model LLaMA 3 dasar dari repositori "meta-llama/Llama-3.2-1B-Instruct" beserta adapter LoRA yang telah dilatih sebelumnya dari *path* `"/content/drive/MyDrive/Generator-Makna-Lirik-Lagu-Drake-main/drake-finetuned-model-final"`. Pemanggilan model dilakukan dengan menggunakan konfigurasi kuantisasi `BitsAndBytesConfig` untuk memastikan efisiensi dalam penggunaan memori selama proses inferensi.

Sementara itu, fungsi `setup_retriever()` dirancang untuk mempersiapkan sistem pencarian lirik berbasis metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Sistem ini mengambil referensi dari file dataset `drake songs and meaning datasets2.csv`. Tujuan utama dari *retriever* ini adalah memberikan konteks tambahan kepada model berdasarkan kemiripan input pengguna dengan lirik-lirik lagu yang telah tersedia di *database*.

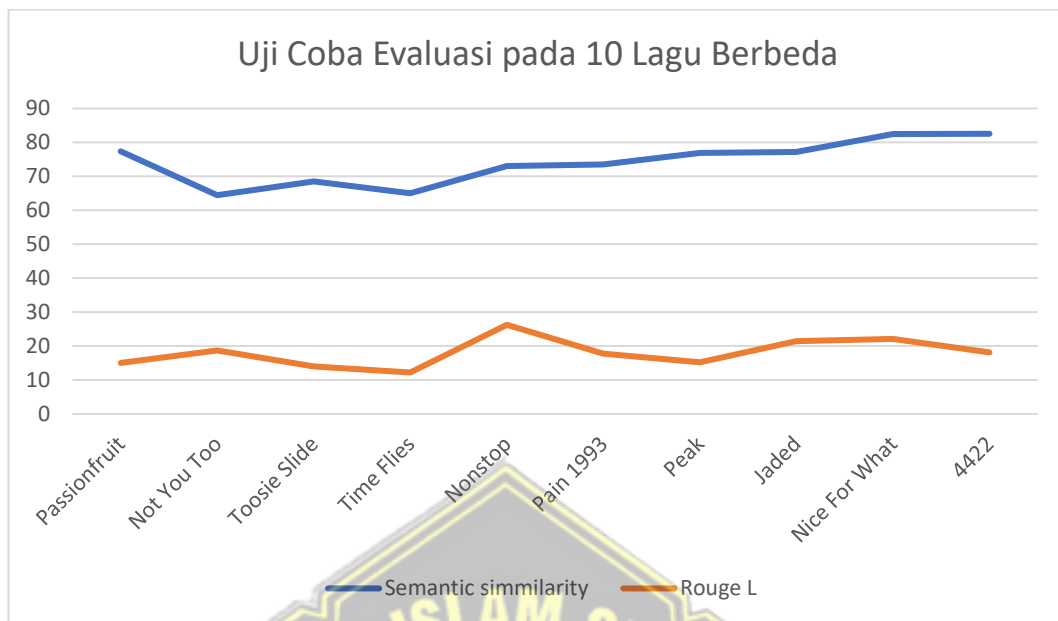
Antarmuka pengguna dari aplikasi ini dirancang secara fungsional. Terdapat elemen utama berupa *text area* tempat pengguna dapat memasukkan lirik lagu dan sebuah tombol untuk memulai analisis. Setelah pengguna menekan tombol, aplikasi akan menjalankan proses yang disebut *Retrieval-Augmented Generation* (RAG), yang dieksekusi oleh fungsi utama `generate_and_evaluate()`. Pada tahap awal (*retrieval*), sistem akan mencari entri lirik yang paling mirip dari *database* untuk dijadikan konteks. Kemudian, pada tahap selanjutnya (*generation*), model LLaMA yang telah di-*fine-tune* akan menghasilkan interpretasi makna berdasarkan lirik input dan konteks tersebut. Hasil akhir berupa makna lirik akan ditampilkan secara langsung di antarmuka web.

4.2.5 Evaluasi Model

Evaluasi dilakukan untuk mengukur sejauh mana model LLaMA 3 yang telah di-*fine-tune* mampu menghasilkan makna lirik lagu Drake yang relevan dan akurat. Untuk mendapatkan penilaian yang komprehensif, pendekatan evaluasi ini menggunakan dua metrik yang saling melengkapi: kesamaan semantik (*semantic*

similarity) dan ROUGE-L (*Longest Common Subsequence*). Pendekatan pertama berfokus pada kesamaan semantik. Metode ini menghitung tingkat kemiripan makna antara teks yang dihasilkan model dengan teks referensi dalam *dataset*. Proses ini menggunakan model *Sentence Transformer*, yaitu all-MiniLM-L6-v2, untuk mengubah kedua teks menjadi representasi vektor. Selanjutnya, *cosine similarity* digunakan untuk mengukur kemiripan antar vektor tersebut. Nilai yang mendekati 1 menandakan tingkat kesamaan makna yang sangat tinggi.

Sebagai pelengkap, metrik kedua yaitu ROUGE-L (*Longest Common Subsequence*) digunakan untuk mengevaluasi kesamaan struktural dan leksikal. ROUGE-L (*Longest Common Subsequence*) menilai kualitas generasi teks dengan mengukur panjang urutan kata terpanjang yang sama antara teks yang dihasilkan dan referensi. Skor ini membantu memastikan bahwa model tidak hanya menangkap makna yang benar, tetapi juga mampu menyusun kalimat dengan pilihan kata dan struktur yang mirip dengan referensi. Dalam implementasinya, sistem secara otomatis menghitung kedua skor ini setiap kali pengguna meminta analisis lirik. Nilai kemiripan semantik dan ROUGE-L (*Longest Common Subsequence*) ditampilkan pada antarmuka aplikasi, bersama dengan makna referensi sebagai pembanding. Hal ini memungkinkan pengguna menilai performa model dari dua sisi: seberapa dekat maknanya (*Cosine Similarity*) dan seberapa mirip cara penyampaiannya (ROUGE-L).



Gambar 4. 2 Chart Uji coba Evaluasi

Pada gambar 4.2 menyajikan grafik visualisasi terperinci mengenai kinerja model melalui serangkaian 10 uji coba yang berbeda. Evaluasi ini bertujuan untuk mengukur performa model secara komprehensif dengan menggunakan dua metrik utama: *Cosine Similarity* yang ditandai dengan garis biru untuk mengukur kesamaan makna (semantik), dan ROUGE-L yang diwakili oleh garis oranye untuk menilai kesamaan struktur kalimat (leksikal). Tujuan utama dari visualisasi ini adalah untuk menunjukkan bagaimana efektivitas model dapat bervariasi secara signifikan tergantung pada lirik spesifik yang dianalisis.

Adanya fluktuasi skor yang terlihat naik-turun pada setiap uji coba merupakan hasil yang dapat diantisipasi dan wajar dalam pengujian model bahasa. Variabilitas ini terutama disebabkan oleh tingkat kompleksitas yang berbeda pada setiap lirik yang diuji; lirik yang kaya akan metafora, slang, atau ambiguitas makna secara inheren lebih menantang untuk diinterpretasikan dibandingkan dengan lirik yang lebih lugas. Selain itu, perbedaan dalam kualitas dan gaya penulisan pada data referensi yang digunakan sebagai pembandingan, serta kemampuan kontekstual model yang mungkin lebih terlatih pada tema-tema tertentu, juga turut menjadi faktor penyebab variasi skor tersebut.

Analisis yang lebih mendalam pada hubungan antara kedua metrik memberikan wawasan yang lebih kaya. Sebagai contoh, pada uji coba 5, skor

Cosine Similarity yang tinggi disertai dengan skor ROUGE-L yang rendah mengindikasikan sebuah keberhasilan interpretasi. Hal ini menunjukkan bahwa model mampu menangkap makna inti lirik dengan sangat akurat, namun menyampaikannya kembali menggunakan kalimat dan pilihan kata yang sepenuhnya berbeda dari teks referensi, yang menyoroti kemampuan parafrase yang baik. Sebaliknya, skor yang sama-sama tinggi pada kedua metrik, seperti yang terlihat pada uji coba 3, menandakan hasil yang ideal di mana makna dan cara penyampaiannya sangat selaras dengan data referensi.

Secara keseluruhan, grafik ini menegaskan bahwa kinerja model bersifat kontekstual dan tidak dapat dianggap seragam untuk semua kasus. Meskipun model menunjukkan pemahaman makna yang cukup solid secara umum, yang tecermin dari skor *Cosine Similarity* yang relatif stabil, kemampuannya untuk meniru struktur kalimat dan pilihan kata dari referensi menunjukkan variasi yang lebih besar. Oleh karena itu, penggunaan dua metrik ini terbukti krusial untuk memberikan gambaran evaluasi yang holistik, yang mampu membedakan antara pemahaman makna sejati dan sekadar kemampuan replikasi leksikal.

4.3 Implementasi Antar Muka Menggunakan Gradio di Google Colab

4.3.1 Tujuan Penggunaan Gradio di Google Colab

Tujuan utama dari penggunaan Gradio dalam penelitian ini adalah untuk menyediakan antarmuka pengguna (*user interface*) yang sederhana, responsif, dan mudah digunakan. Hal ini memungkinkan model LLaMA 3 yang telah di-*fine-tune* dapat diakses dan diuji oleh pengguna non-teknis tanpa harus berinteraksi langsung dengan kode program. Dengan Gradio, proses implementasi antarmuka menjadi lebih efisien karena *framework* ini mendukung integrasi langsung di dalam *notebook* Google Colab, memungkinkan pemanggilan model, pemrosesan input, serta penyajian output dilakukan dalam satu alur kerja yang ringkas.

Selain itu, Gradio memungkinkan visualisasi hasil model secara langsung dan *real-time*, sehingga sangat sesuai untuk tugas yang bersifat interaktif seperti generasi makna lirik lagu. Melalui fitur-fitur seperti `gr.Textbox`, `gr.Button`, dan `gr.Tabs`, pengguna dapat dengan mudah memasukkan potongan lirik, menekan tombol untuk memprosesnya, dan segera melihat hasil interpretasi makna yang

diberikan oleh model. Tujuan lainnya adalah untuk mendemonstrasikan proses *Retrieval-Augmented Generation* (RAG) secara transparan dalam konteks yang mudah dipahami oleh pengguna akhir

Dengan demikian, penggunaan Gradio yang dijalankan di lingkungan Google Colab berperan penting dalam menjembatani teknologi model bahasa besar dengan pengguna. Lingkungan Google Colab menyediakan akses GPU yang krusial untuk mengatasi masalah performa, sementara Gradio menyediakan antarmuka intuitif. Kombinasi ini tidak hanya mendukung proses eksplorasi hasil model, tetapi juga proses validasi, dengan kemampuan untuk menampilkan skor evaluasi (*Cosine Similarity* dan ROUGE-L) secara langsung kepada pengguna.

4.3.2 Fitur Utama Aplikasi Gradio di Google Colab

Aplikasi web yang diimplementasikan ulang menggunakan Gradio ini dirancang sebagai antarmuka satu halaman yang interaktif dan terstruktur, dibangun menggunakan `gr.Blocks`. Fungsionalitas aplikasi secara jelas dibagi menjadi dua panel utama: panel *input* di sisi kiri dan panel *output* di sisi kanan. Panel *input* menyediakan area teks (`gr.Textbox`) bagi pengguna untuk memasukkan potongan lirik, tombol aksi utama "Hasilkan Makna" (`gr.Button`), serta beberapa contoh lirik (`gr.Examples`) untuk mempermudah pengujian.

Panel *output* dirancang secara strategis menggunakan `gr.Tabs` untuk memisahkan hasil analisis dan validasi, sehingga menyajikan informasi secara bersih dan tidak tumpang tindih. Tab pertama adalah "Hasil *Generate*", yang menampilkan hasil utama dari proses *Retrieval-Augmented Generation* (RAG). Setelah pengguna menekan tombol, tab ini akan diisi dengan informasi kontekstual yang berhasil diambil oleh *retriever* (Judul Lagu, Album, dan Sumber) serta "Makna yang Dihasilkan AI" yang merupakan *output* dari model LLaMA 3.

Tab kedua adalah "Hasil Evaluasi", yang berfungsi sebagai fitur validasi transparan dan merupakan peningkatan signifikan dari implementasi sebelumnya. Tab ini secara otomatis menampilkan "Makna Referensi" (standar emas) dari dataset yang sesuai dengan lirik yang di-*input*. Selain itu, sistem juga menghitung dan menampilkan dua skor evaluasi kuantitatif secara *real-time*: "Kemiripan Semantik (*Cosine*)" dan "Kecocokan Teks (ROUGE-L)".

Dengan struktur ini, aplikasi tidak hanya menjalankan fungsi utamanya sebagai generator makna lirik, tetapi juga secara instan menyediakan bukti kuantitatif atas performa model untuk setiap input yang diberikan. Ini sangat mendukung proses analisis dan validasi penelitian.

4.3.3 Cara Akses Gradio di Google Colab

Aplikasi web dalam penelitian ini dikembangkan dan dijalankan pada lingkungan *cloud* Google Colab, yang berbeda dari implementasi awal yang dijalankan secara lokal. Pendekatan ini dipilih untuk memanfaatkan akselerasi GPU yang esensial untuk mengatasi masalah performa. Akses ke aplikasi difasilitasi oleh server web yang terintegrasi dalam *framework* Gradio, yang dijalankan di dalam *runtime* Google Colab.

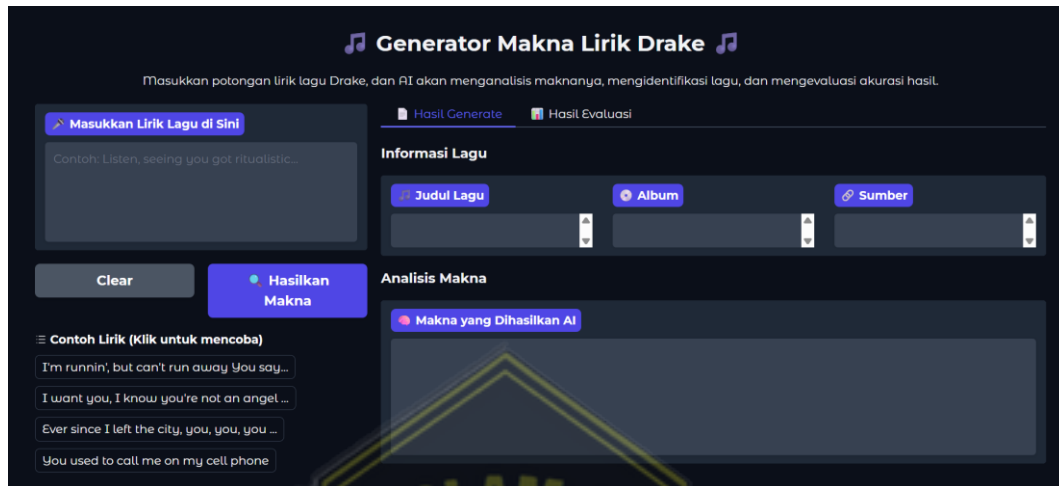
Proses untuk mengakses aplikasi dimulai dengan membuka *notebook* `gradio.ipynb` dan menjalankan sel-sel kode secara berurutan. Langkah pertama adalah menyiapkan lingkungan, di mana semua dependensi yang diperlukan—seperti `gradio`, `torch`, `transformers`, `peft`, dan `bitsandbytes`—diinstal melalui sel kode pertama. Setelah lingkungan siap, sel-sel berikutnya dieksekusi untuk memuat model (fungsi `load_llm_model`) dan menyiapkan *retriever* (fungsi `setup_retriever`).

Terakhir, sel kode yang berisi perintah `app.launch(debug=True, share=True)` dieksekusi. Perintah ini secara otomatis akan memulai server Gradio. Berkat parameter `share=True`, Gradio akan menghasilkan sebuah URL publik (biasanya berakhir dengan `.gradio.live`) yang ditampilkan di *output* sel. URL ini dapat diakses melalui *browser* dari perangkat apa pun, yang sangat memudahkan proses demonstrasi dan validasi hasil penelitian.

4.3.4 Hasil Implementasi

Setelah proses migrasi ke Google Colab dan implementasi ulang antarmuka menggunakan Gradio selesai dilakukan, aplikasi berhasil dijalankan dan dapat digunakan untuk menghasilkan makna dari potongan lirik lagu Drake. Antarmuka aplikasi ditampilkan secara bersih, sederhana, dan mudah digunakan, yang memungkinkan interaksi langsung antara pengguna dan sistem secara *real-time*.

Implementasi ini menunjukkan bahwa hasil *fine-tuning* model dapat digunakan secara praktis dalam bentuk aplikasi berbasis web.



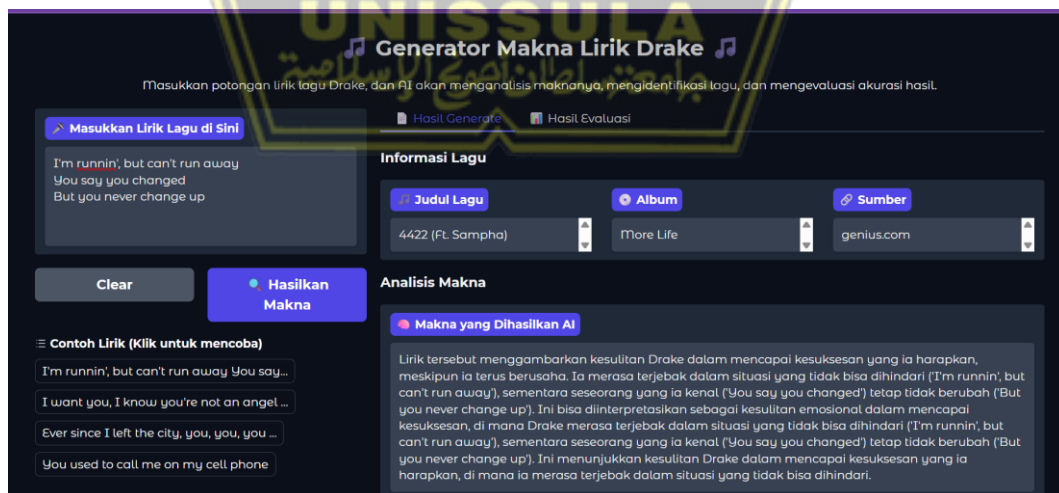
Gambar 4. 3 Antarmuka Halaman Input Aplikasi Gradio

Gambar 4.3 menyajikan antarmuka pengguna (UI) utama dari aplikasi "Generator Makna Lirik Drake", yang secara efektif berfungsi sebagai "wajah" sistem dan titik interaksi pertama bagi pengguna. Desain ini secara teknis diimplementasikan menggunakan `gr.Blocks` dari *framework* Gradio, yang memungkinkan fleksibilitas penuh dalam menyusun tata letak. Pilihan tata letak dua kolom yang bersih ini bersifat fundamental; secara kognitif, ini memisahkan dengan jelas antara 'aksi' (*input*) di sebelah kiri dan 'reaksi' (*output*) di sebelah kanan. Prinsip desain ini sangat intuitif karena mengurangi beban kognitif pengguna. Alih-alih harus menavigasi beberapa halaman, pengguna langsung disajikan dengan seluruh alur kerja dalam satu pandangan, yang secara inheren berorientasi pada tugas (*task-oriented*).

Panel input fungsional di sisi kiri dirancang untuk memandu pengguna secara mulus menuju aksi utama. Area teks utama ("Masukkan Lirik Lagu di Sini") dilengkapi dengan teks *placeholder* yang tidak hanya memberi instruksi, tetapi juga memberi contoh format *input* yang diharapkan. Tombol aksi utama, "Hasilkan Makna", secara visual ditekankan menggunakan atribut `variant="primary"` (menampilkannya sebagai tombol berwarna solid) untuk menarik perhatian pengguna sebagai langkah logis berikutnya. Penambahan tombol "Clear" adalah fitur kualitas hidup (*quality-of-life*) sederhana yang sangat penting untuk pengujian

berulang, memungkinkan pengguna untuk dengan cepat membersihkan *input* sebelumnya. Lebih lanjut, fitur "Contoh Lirik" (gr.*Examples*) yang terletak di bawahnya memiliki nilai strategis ganda: bagi pengguna umum, ini menghilangkan hambatan untuk mencoba aplikasi, dan bagi peneliti atau penguji, ini menyediakan satu set data validasi standar yang konsisten, memastikan fungsionalitas sistem dapat diverifikasi dengan mudah tanpa perlu mencari lirik secara manual.

Di sisi kanan, panel *output* yang awalnya kosong secara visual mengkomunikasikan kesiapan sistem untuk menerima permintaan sebuah kanvas kosong yang menunggu untuk diisi. Penggunaan strategis gr.*Tabs* adalah keputusan desain krusial. Daripada menampilkan semua informasi sekaligus yang dapat membingungkan dan membebani pengguna, panel ini secara rapi memisahkannya menjadi dua konteks yang berbeda: "Hasil *Generate*" (ditujukan untuk interpretasi kualitatif yang dihasilkan AI) dan "Hasil Evaluasi" (ditujukan untuk validasi kuantitatif dan perbandingan dengan data referensi). Pemisahan ini sangat penting untuk transparansi penelitian, memungkinkan pengguna untuk memilih tingkat kedalaman informasi yang mereka inginkan. Pada saat pemuatan awal (seperti yang ditunjukkan gambar), keadaan kosong ini berfungsi sebagai pemicu visual yang jelas bahwa sistem sedang dalam mode 'mendengarkan' dan menunggu *input* pengguna.



Gambar 4. 4 Antarmuka Halaman Hasil *Generate*

Gambar 4.4 secara komprehensif mendemonstrasikan alur kerja penuh dan fungsional dari arsitektur *Retrieval-Augmented Generation* (RAG). Gambar ini

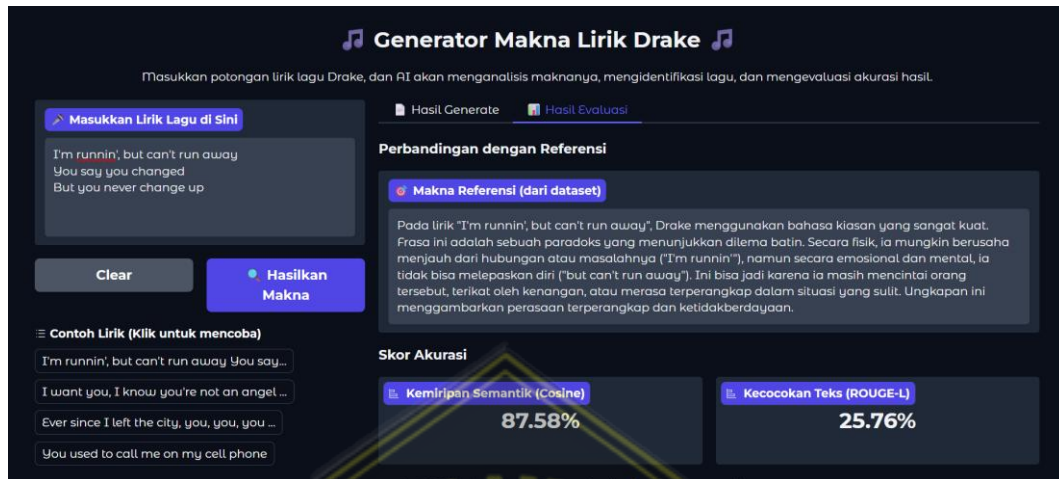
menangkap momen krusial tepat setelah pengguna memasukkan potongan lirik dalam contoh ini, "*I'm runnin', but can't run away...*" dan menekan tombol "Hasilkan Makna". Alur kerja sistem yang cerdas langsung terlihat. Alih-alih langsung mengirimkan lirik mentah ke model AI untuk diinterpretasi, yang berisiko menghasilkan jawaban generik dan tidak akurat, sistem terlebih dahulu menjalankan proses penarikan informasi kontekstual.

Keberhasilan langkah awal ini ditunjukkan dengan jelas pada bagian "Informasi Lagu". *Retriever* TF-IDF yang telah dilatih pada *database* lirik berhasil memproses input pengguna. Ia dengan akurat mencocokkan potongan lirik yang relatif singkat tersebut dengan entri yang paling relevan secara statistik di dalam korpus data. Sistem dengan tepat mengidentifikasi konteks spesifik dari lirik tersebut, yaitu lagu "4422 (Ft. Sampha)", dari album "*More Life*", dengan sumber referensi "genius.com". Penampilan informasi kontekstual ini sangat penting; ini berfungsi sebagai mekanisme transparansi yang membuktikan kepada pengguna bahwa sistem telah "mengenali" lagu yang tepat sebelum mencoba menganalisisnya lebih dalam.

Hanya setelah konteks yang relevan ini berhasil ditarik dan dikonfirmasi, sistem melanjutkan ke proses inti: generasi makna. Informasi yang baru saja diambil (judul, album) kemudian "disuntikkan" bersama dengan lirik input asli ke dalam *prompt* yang dikirim ke model LLaMA 3 yang telah di-*fine-tune*. Oleh karena itu, output yang ditampilkan di bawah "Makna yang Dihasilkan AI" bukanlah sekadar tebakan, melainkan hasil dari generasi yang telah *augmented* atau diperkaya oleh konteks spesifik lagu tersebut.

Teks penjelasan yang muncul ("Lirik tersebut menggambarkan kesulitan Drake...") adalah bukti utama dari kinerja dan keberhasilan proses *fine-tuning* model. Tampilan ini menunjukkan kemampuan model untuk melakukan tugas-tugas kognitif yang kompleks; ia tidak hanya mengulang kata kunci, tetapi juga berhasil *memahami* ide-ide abstrak seperti paradoks ("*runnin', but can't run away*") dan menerjemahkannya menjadi konsep emosional ("kesulitan emosional," "terjebak dalam situasi"). Kemampuan untuk mengubah lirik puitis yang seringkali sulit dimengerti menjadi penjelasan naratif yang koheren, jelas, dan masuk akal

secara langsung memvalidasi bahwa model LLaMA 3 telah berhasil dilatih untuk berpikir dan menginterpretasi sesuai dengan tugas spesifik yang diberikan.



Gambar 4. 5 Tampilan Halaman Hasil Evaluasi dan Skor Kuantitatif *Real-time*

Gambar 4.5 menunjukkan fungsionalitas validasi kuantitatif transparan yang tersedia pada tab "Hasil Evaluasi". Fitur ini merupakan komponen krusial dalam arsitektur penelitian ini, karena berfungsi untuk memindahkan penilaian kinerja model dari ranah subjektif ("terlihat bagus") ke ranah objektif dan terukur untuk setiap *input* yang diberikan. Antarmuka ini dirancang untuk transparansi penuh. Saat pengguna menguji lirik, sistem secara otomatis mengambil dan menampilkan "Makna Referensi (dari dataset)". Teks referensi ini berfungsi sebagai standar emas (*ground truth*) pembandingan, yang menyediakan jangkar faktual untuk menilai kualitas keluaran yang dihasilkan oleh AI.

Di bawah perbandingan kualitatif *side-by-side* ini, sistem menghitung dan menyajikan dua skor metrik evaluasi yang berbeda secara *real-time*. Skor pertama, "Kemiripan Semantik (*Cosine*)", yang dalam contoh ini mencapai 87.58%, adalah metrik yang paling penting untuk tujuan penelitian ini. Skor yang sangat tinggi ini menandakan bahwa makna, esensi inti, dan niat konseptual dari teks yang dihasilkan AI secara semantik sangat mirip dengan makna referensi. Ini membuktikan bahwa model telah berhasil *memahami* konsep abstrak dan emosi di balik lirik, terlepas dari perbedaan pilihan kata.

Sementara itu, skor kedua, "Kecocokan Teks (ROUGE-L)", tercatat hanya sebesar 25.76%. Metrik ini mengukur kesamaan leksikal, atau tumpang tindih kata

dan frasa dalam urutan yang sama. Kombinasi yang tampak kontras antara skor semantik yang tinggi dan skor leksikal (ROUGE-L) yang rendah ini bukanlah sebuah kegagalan; sebaliknya, ini adalah validasi kunci dan hasil ideal dari proses *fine-tuning*. Hal ini secara kuantitatif membuktikan bahwa model tidak sekadar *overfitting* atau "menghafal" dan menyalin frasa secara mentah-mentah dari data latihannya. Jika skor ROUGE-L juga tinggi, itu akan menunjukkan bahwa model tersebut gagal melakukan generalisasi.

Sebaliknya, hasil ini membuktikan bahwa model telah berhasil mengembangkan kemampuan abstraksi dan parafrase yang canggih. Model ini berhasil memahami makna inti (seperti yang ditunjukkan oleh skor *Cosine* yang tinggi), lalu mengungkapkannya kembali secara mandiri menggunakan struktur kalimat dan pilihan katanya sendiri (seperti yang ditunjukkan oleh skor ROUGE-L yang rendah). Fenomena ini secara kuat memvalidasi bahwa proses *fine-tuning* telah berhasil: model ini tidak bertindak sebagai mesin pencari yang meniru, melainkan sebagai mesin interpretasi sejati yang mampu *berpikir* dan *mensintesis* penjelasan baru berdasarkan pemahaman kontekstual.

4.4 Pembahasan Implementasi

Implementasi sistem Generator Makna Lirik Lagu dalam penelitian ini berhasil membuktikan kelayakan teknis dari arsitektur yang diusulkan. Setelah melalui tahap revisi untuk optimasi performa, orkestrasi antara model bahasa skala besar (LLaMA 3) dengan kerangka kerja aplikasi interaktif Gradio yang dijalankan di Google Colab terbukti sangat efektif. Keseluruhan alur kerja, mulai dari fase preparasi data, penerapan metode *fine-tuning* yang efisien (LoRA), hingga *deployment* menjadi sebuah aplikasi fungsional yang responsif, telah berhasil dieksekusi secara terstruktur dan koheren.

Secara fungsional, sistem yang dibangun mengadopsi arsitektur *Retrieval-Augmented Generation* (RAG) yang terbukti andal. Pada praktiknya, pengguna cukup menyediakan *input* berupa sekuens lirik, dan sistem secara otonom menjalankan dua fase krusial. Fase *retrieval*, yang memanfaatkan kalkulasi TF-IDF, menunjukkan efektivitas signifikan dalam mengidentifikasi lagu dengan konteks paling relevan dari dalam korpus data. Keberhasilan tahap ini (seperti terlihat pada

Gambar 4.4) menjadi fondasi penting bagi tahap *generation*, di mana model LLaMA 3 yang telah di-*fine-tune* dapat menghasilkan interpretasi makna yang diperkaya dengan konteks spesifik.

Dari perspektif antarmuka pengguna, revisi implementasi dari Streamlit lokal ke Gradio di Google Colab merupakan respons strategis untuk mengatasi kendala performa. Pilihan Gradio terbukti tepat karena kemudahannya dalam integrasi dengan *notebook* Colab dan kemampuannya membuat tautan publik untuk demonstrasi. Lebih penting lagi, antarmuka baru ini (seperti terlihat pada Gambar 4.5) secara langsung meningkatkan transparansi penelitian. Dengan menggunakan `gr.Tabs`, aplikasi kini tidak hanya menyajikan makna yang dihasilkan, tetapi juga secara *real-time* menampilkan "Makna Referensi" dan skor kuantitatif ("Kemiripan Semantik" dan "ROUGE-L"). Fitur ini berkontribusi pada pengalaman pengguna (UX) yang lebih informatif dan mendukung validasi model secara instan.

Validasi terhadap kualitas *output* model dilakukan melalui pendekatan kuantitatif multi-metrik untuk mendapatkan gambaran yang holistik. Metrik pertama, kesamaan semantik (*semantic similarity*), secara objektif mengukur konvergensi makna antara teks yang dihasilkan model dengan data referensi (*ground truth*). Metrik kedua, ROUGE-L, ditambahkan untuk mengevaluasi kesamaan pada level struktural dan leksikal dengan mengukur tumpang tindih urutan kata. Penggunaan dua metrik ini memungkinkan penilaian dari dua sisi: seberapa akurat makna yang ditangkap dan seberapa mirip cara penyampaiannya.

Hasil pengujian secara konsisten menunjukkan bahwa mayoritas *output* berada dalam spektrum kesamaan semantik moderat hingga tinggi, namun dengan skor ROUGE-L yang lebih bervariasi dan cenderung lebih rendah. Seperti yang ditunjukkan pada Gambar 4.5, skor 87.58% (Semantik) dan 25.76% (ROUGE-L) adalah contoh nyata dari pola ini. Fenomena ini tidak mengindikasikan kegagalan, melainkan memberikan wawasan mendalam mengenai kapabilitas model. Skor *semantic similarity* yang tinggi membuktikan kemampuan model untuk melakukan abstraksi dan pemahaman konteks, sementara skor ROUGE-L yang lebih rendah menunjukkan bahwa model cenderung memparafrasekan makna daripada meniru struktur kalimat referensi. Fenomena ini justru memvalidasi keberhasilan proses

fine-tuning, yang telah mengajarkan model untuk menginterpretasi dan meregenerasi makna secara mandiri.

Salah satu inovasi teknis yang paling berdampak dalam penelitian ini adalah penerapan LoRA yang dikombinasikan dengan kuantisasi. Strategi ini berhasil secara drastis mereduksi jumlah parameter yang dapat dilatih hingga hanya berkisar 0.13% dari total parameter asli model. Efisiensi ini memungkinkan proses *fine-tuning* yang kompleks dapat dilaksanakan pada infrastruktur komputasi dengan sumber daya terbatas (Google Colab), namun tetap mampu menghasilkan tingkat performa yang kompeten untuk tugas spesifik interpretasi lirik.

Meskipun demikian, sebagai bagian dari analisis kritis, perlu diakui beberapa limitasi. Kualitas *output* sistem menunjukkan ketergantungan yang tinggi pada kualitas dan kekayaan data pelatihan. Terdapat tendensi bagi model untuk menghasilkan interpretasi yang bersifat lebih generik ketika dihadapkan pada *input* lirik yang sangat singkat atau tidak memiliki padanan kontekstual yang kuat dalam *database*. Lebih lanjut, pemahaman terhadap metafora yang kompleks atau referensi budaya yang sangat spesifik masih menjadi tantangan. Oleh karena itu, kurasi dan diversifikasi data menjadi variabel krusial yang menentukan plafon performa sistem di masa depan.

Secara konklusif, implementasi sistem ini telah berhasil mencapai seluruh objektif yang digariskan pada awal penelitian. Proyek ini tidak hanya sukses membangun sebuah sistem *end-to-end* yang mampu menginterpretasikan makna lirik secara otomatis, tetapi juga memvalidasi efektivitas dari gabungan arsitektur RAG dengan teknik *fine-tuning* yang efisien. Keberhasilan dalam mengintegrasikan seluruh komponen menjadi satu alur kerja yang solid dan responsif (setelah optimasi) membuktikan bahwa pendekatan yang digunakan merupakan fondasi yang valid dan menjanjikan untuk pengembangan aplikasi serupa.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil mencapai tujuannya untuk membangun sebuah aplikasi web generator makna lirik yang dapat menginterpretasikan lirik lagu Drake secara otomatis. Keberhasilan ini dicapai melalui tahapan terstruktur, mulai dari pengumpulan data dan melakukan *fine-tuning* pada model LLaMA 3 dengan pendekatan *Low-Rank Adaptation* (LoRA). Pada tahap implementasi akhir, dilakukan revisi signifikan dengan mengganti *framework* antarmuka dari Streamlit (yang dijalankan secara lokal) menjadi Gradio yang dieksekusi di lingkungan Google Colab. Revisi ini terbukti krusial untuk mengatasi kendala performa, dengan mereduksi waktu inferensi dari ± 5 menit menjadi kurang dari 60 detik.

Aplikasi yang dihasilkan secara efektif mengimplementasikan arsitektur *Retrieval-Augmented Generation* (RAG), yang memungkinkan sistem mengambil konteks lagu relevan sebelum menghasilkan interpretasi. Kinerja sistem divalidasi secara kuantitatif melalui evaluasi ganda yang juga diintegrasikan secara transparan ke dalam antarmuka aplikasi. Metrik kesamaan semantik (*semantic similarity*) menunjukkan kesamaan makna yang tinggi antara hasil model dengan data referensi. Sebagai pelengkap, metrik ROUGE-L digunakan untuk mengukur kesamaan struktural dan leksikal. Secara keseluruhan, penelitian ini menyajikan prototipe fungsional yang membuktikan kelayakan integrasi model bahasa canggih dengan teknik *fine-tuning* efisien untuk aplikasi analisis teks yang responsif dan tervalidasi.

5.2 Saran

Untuk pengembangan di masa depan, beberapa peningkatan teknis pada inti sistem sangat disarankan. Prioritas utama adalah memperluas dan mendiversifikasi dataset dengan lirik dari berbagai artis serta genre untuk meningkatkan kemampuan generalisasi model. Seiring dengan itu, mekanisme *retrieval* yang saat ini menggunakan TF-IDF dapat ditingkatkan menjadi model *embedding* semantik seperti Sentence-BERT untuk pencarian konteks yang lebih akurat.

Khusus untuk mengatasi skor ROUGE-L yang rendah (seperti yang teridentifikasi dalam hasil evaluasi), dapat dilakukan tahap *fine-tuning* kedua yang berfokus pada gaya penulisan. Dengan menggunakan dataset yang dirancang untuk meniru struktur kalimat referensi, model dapat dilatih tidak hanya untuk memahami makna, tetapi juga untuk menyajikannya dengan gaya yang lebih selaras.

Dari sisi aplikasi dan pengalaman pengguna, sistem dapat diperkaya dengan fitur-fitur interaktif seperti histori pencarian, dukungan multibahasa, dan mekanisme umpan balik. Fitur umpan balik ini akan menciptakan siklus perbaikan berkelanjutan (*human-in-the-loop*), di mana data dari pengguna dapat dimanfaatkan untuk penyempurnaan model di masa depan. Lebih jauh lagi, penelitian ini diharapkan dapat menjadi fondasi yang kokoh bagi pengembangan sistem serupa di domain lain yang memerlukan interpretasi teks secara kontekstual



DAFTAR PUSTAKA

- Agrawal, Y., Shanker, R. G. R., & Alluri, V. (2021). *Transformer-based approach towards music emotion recognition from lyrics*. https://doi.org/10.1007/978-3-030-72240-1_12
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. <https://doi.org/10.1145/3442188.3445922>
- Chu, Z., Yang, Z., Cui, Y., Chen, Z., & Liu, M. (2022). *HIT at SemEval-2022 Task 2: Pre-trained Language Model for Idioms Detection*. <http://arxiv.org/abs/2204.06145>
- de Andrade, C. M. V., Cunha, W., Reis, D., Pagano, A. S., Rocha, L., & Gonçalves, M. A. (2024). *A Strategy to Combine 1stGen Transformers and Open LLMs for Automatic Text Classification*. <http://arxiv.org/abs/2408.09629>
- Dennis Heraldi, F., & Zakhralativa Ruskanda, F. (2024). *EasyChair Preprint Effective Intended Sarcasm Detection Using Fine-Tuned Llama 2 Large Language Models Effective Intended Sarcasm Detection Using Fine-tuned Llama 2 Large Language Models*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). an Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale. *ICLR 2021 - 9th International Conference on Learning Representations*.
- Duan, W., Zhang, Z., Yu, Y., & Oyama, K. (2022). *Interpretable Melody Generation from Lyrics with Discrete-Valued Adversarial Training*. <http://arxiv.org/abs/2206.15027>
- Han, A., & Park, R. (n.d.). *Deep-Hop Rapper: Using LSTM and Transformer for Rap Lyric Generation Stanford CS224N {Custom, Default} Project*. <https://deepbeat.org/>.
- Hu, Y. (2024). Performance exploration of Generative Pre-trained Transformer-2 for lyrics generation. *Applied and Computational Engineering*, 48(1), 53–60.

<https://doi.org/10.54254/2755-2721/48/20241154>

- Huang, J., & Benetos, E. (2024). *Towards Building an End-to-End Multilingual Automatic Lyrics Transcription Model*. <http://arxiv.org/abs/2406.17618>
- Kahyun, & Choi. (n.d.). *Computational Thematic Analysis of Poetry via Bimodal Large Language Models*.
- Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision: A Survey. *ACM Computing Surveys*, 54(10), 1–30. <https://doi.org/10.1145/3505244>
- Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, 3(1), 111–132. <https://doi.org/10.1016/j.aiopen.2022.10.001>
- Liu, Z., Li, Y., Shu, P., Zhong, A., Yang, L., Ju, C., Wu, Z., Ma, C., Luo, J., Chen, C., Kim, S., Hu, J., Dai, H., Zhao, L., Zhu, D., Liu, J., Liu, W., Shen, D., Liu, T., ... Li, X. (2023). *Radiology-Llama2: Best-in-Class Large Language Model for Radiology*. <http://arxiv.org/abs/2309.06419>
- Lv, H., Pu, C., Duo, L., Li, Y., Zhou, Q., & Shen, J. (2025). T-LLaMA: a Tibetan large language model based on LLaMA2. *Complex and Intelligent Systems*, 11(1). <https://doi.org/10.1007/s40747-024-01641-7>
- Mao, Y., Ge, Y., Fan, Y., Xu, W., Mi, Y., Hu, Z., & Gao, Y. (2025). A survey on LoRA of large language models. *Frontiers of Computer Science*, 19(7), 1–144. <https://doi.org/10.1007/s11704-024-40663-9>
- Naseri, S., Reddy, S., Correia, J., Karlgren, J., & Jones, R. (2022). *The Contribution of Lyrics and Acoustics to Collaborative Understanding of Mood*. <http://arxiv.org/abs/2207.05680>
- Prentiqi, V., Kalimeri, K., & Saitis, C. (2023). Soundscapes of morality: Linking music preferences and moral values through lyrics and audio. *PLoS ONE*, 18(11 November). <https://doi.org/10.1371/journal.pone.0294402>
- Qiu, X. P., Sun, T. X., Xu, Y. G., Shao, Y. F., Dai, N., & Huang, X. J. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10), 1872–1897. <https://doi.org/10.1007/s11431-020-1647-3>
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li,

- W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21, 1–67.
- Ram, N., Gummadi, T., Bhethanabotla, R., Savery, R. J., & Weinberg, G. (2021). Say What? Collaborative Pop Lyric Generation Using Multitask Transfer Learning. *HAI 2021 - Proceedings of the 9th International User Modeling, Adaptation and Personalization Human-Agent Interaction*, 165–173. <https://doi.org/10.1145/3472307.3484175>
- Shazeer, N. (2020). *GLU Variants Improve Transformer*. 4, 1–5. <http://arxiv.org/abs/2002.05202>
- Tong, M., Xu, B., Wang, S., Han, M., Cao, Y., Zhu, J., Chen, S., Hou, L., & Li, J. (2022). DocEE: A Large-Scale and Fine-grained Benchmark for Document-level Event Extraction. *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 3970–3982. <https://doi.org/10.18653/v1/2022.naacl-main.291>
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., ... Scialom, T. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. <http://arxiv.org/abs/2307.09288>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. <http://arxiv.org/abs/1706.03762>
- Ventura, M., & Toker, M. (2022). *TRBLLmaker -- Transformer Reads Between Lyrics Lines maker*. <http://arxiv.org/abs/2212.04917>
- Weir, N., Poliak, A., & Van Durme, B. (2020). Probing Neural Language Models for Human Tacit Assumptions. *Proceedings for the 42nd Annual Meeting of the Cognitive Science Society: Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020*, 377–383.
- Wu, C., Gan, Y., Ge, Y., Lu, Z., Wang, J., Feng, Y., Shan, Y., & Luo, P. (2024). *LLaMA Pro: Progressive LLaMA with Block Expansion*.

<http://arxiv.org/abs/2401.02415>

Yuan, R., Lin, H., Guo, S., Zhang, G., Pan, J., Zang, Y., Liu, H., Liang, Y., Ma, W., Du, X., Du, X., Ye, Z., Zheng, T., Ma, Y., Liu, M., Tian, Z., Zhou, Z., Xue, L., Qu, X., ... Guo, Y. (2025). *YuE: Scaling Open Foundation Models for Long-Form Music Generation*. <http://arxiv.org/abs/2503.08638>

Zhang, R., Han, J., Liu, C., Gao, P., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., & Qiao, Y. (2023). *LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention*. <http://arxiv.org/abs/2303.16199>

