

TUGAS AKHIR 1

IMPLEMENTASI TEXT GENERATION DALAM SISTEM PEMBELAJARAN PEMROGRAMAN DENGAN PEMANFAATAN METODE FINE-TUNE FALCON LANGUAGE MODEL (LLM)

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Teknik Informatika



Disusun Oleh:

Nama : Danang Haris Setiawan
NIM : A11.2020.13179
Program Studi : Teknik Informatika

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO
SEMARANG**

2023

DAFTAR ISI

DAFTAR ISI	2
DAFTAR TABEL	4
BAB I	5
PENDAHULUAN	5
1.1 Latar Belakang Masalah	5
1.2 Rumusan Masalah	7
1.3 Batasan Masalah	7
1.4 Tujuan Penelitian	8
1.5 Manfaat Penelitian	8
BAB II	10
LANDASAN TEORI	10
2.1 Tinjauan Studi	10
2.1.1 State Of The Art	10
2.2 Tinjauan Pustaka	15
2.2.1. Text Generation dalam Natural Language Processing (NLP)	15
2.2.1.1. Text Generation NLP	16
2.2.1.2. Fine-Tuning dalam Text Generation	16
2.2.1.3. Falcon Language Model (LLM)	17
2.2.2. Penerapan Text Generation NLP dalam Pendidikan	17
2.2.2.1. Pembelajaran Pemrograman	17
2.2.2.2. Adaptabilitas dalam Pembelajaran	18
2.2.3. Pengembangan Sistem Pembelajaran	18
2.3 Kerangka Pemikiran	20
BAB III	22
METODOLOGI PENELITIAN	22
3.1 Instrumen Penelitian	22
3.1.1 Kebutuhan Perangkat Lunak	22
3.1.2 Kebutuhan Perangkat Keras	22
3.2 Arsitektur Sistem	22
Gambar 2: Diagram arsitektur sistem	24
3.3 Fine Tuning Falcon LLM (Large Language Model)	24
3.3.1. Pengambilan Dataset	24
Gambar 3: Diagram arsitektur sistem	25
3.3.2. Pemrosesan Data	26
3.3.3. Fine-Tuning Model Falcon	26

a. Pemilihan Hyperparameter Model	26
b. Training, Validation, Testing	27
c. Evaluasi Model	28
DAFTAR PUSTAKA	31

DAFTAR TABEL

Tabel 1: State of The Art	15
Tabel 2: Kerangka Pemikiran	22

DAFTAR GAMBAR

Gambar 1: Kerangka Pemikiran	23
Gambar 2: Diagram arsitektur sistem	25
Gambar 3: Diagram arsitektur sistem	26

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pemrograman adalah salah satu keterampilan yang sangat dibutuhkan di era digital saat ini. Pemrograman memungkinkan seseorang untuk menciptakan, mengembangkan, dan memodifikasi berbagai aplikasi, sistem dan perangkat lunak yang dapat membantu memecahkan masalah, meningkatkan produktivitas, dan menciptakan nilai. Namun, belajar pemrograman tidaklah mudah. Menurut sebuah studi yang dilakukan oleh University of Helsinki dan Aalto University, sekitar 60% dari mahasiswa yang mengambil kursus pemrograman mengalami kesulitan dalam memahami konsep-konsep dasar dan menerapkannya dalam praktiknya. Menurut sebuah survei yang dilakukan oleh Stack Overflow pada tahun 2020, sekitar 51% responden mengaku mengalami kesulitan dalam mempelajari konsep-konsep pemrograman, seperti algoritma, struktur data, paradigma pemrograman, dan bahasa pemrograman.

Salah satu faktor yang menyebabkan kesulitan belajar pemrograman adalah kurangnya sumber belajar yang sesuai dengan kebutuhan dan preferensi individu (University of Helsinki & Aalto University, n.d). Banyak sumber belajar pemrograman yang tersedia baik secara online maupun offline, namun tidak semua dari sumber tersebut dapat memberikan penjelasan yang jelas, lengkap, dan menarik bagi pelajar. Selain itu banyak sumber belajar pemrograman yang bersifat statis dan tidak dapat disesuaikan dengan tingkat kemampuan, gaya belajar, dan minat pelajar. Hal ini dapat menyebabkan pelajar merasa bosan, bingung, atau frustrasi saat menghadapi materi yang terlalu mudah atau terlalu sulit.

Sehingga, untuk mengatasi masalah ini, diperlukan sebuah sistem pembelajaran pemrograman berbasis teks yang dapat disesuaikan dengan kebutuhan pengguna. Sistem ini dapat memberikan konten belajar pemrograman yang relevan, interaktif, dan menyesuaikan diri dengan input pengguna.

Untuk menciptakan sistem pembelajaran pemrograman berbasis teks yang dapat disesuaikan dengan kebutuhan pengguna, diperlukan teknologi text generation NLP dan metode fine-tune LLM. Text generation NLP adalah teknologi yang memungkinkan mesin untuk menghasilkan teks secara otomatis berdasarkan data atau informasi tertentu (Goldbreg Y, 2023). Metode fine-tune LLM adalah metode yang digunakan untuk meningkatkan kinerja dan akurasi model bahasa besar (LLM) dengan melatih ulang model tersebut pada data atau tugas spesifik (Howard J. and Ruder S., 2023). LLM adalah model AI yang menggunakan teknik deep learning dan data teks besar untuk memahami, merangkum, menghasilkan dan memprediksi konten baru (Dathathri, S. et al, 2019).

Dengan menggunakan teknologi text generation NLP dan metode fine-tune LLM, sistem pembelajaran pemrograman berbasis teks dapat menghasilkan konten belajar yang berkualitas tinggi dan sesuai dengan kebutuhan pengguna. Teknologi text generation NLP dapat membantu sistem untuk menghasilkan teks yang koheren, konsisten, dan informatif berdasarkan input pengguna. Metode fine-tune LLM dapat membantu sistem untuk meningkatkan kemampuan model bahasa besar dalam memahami konteks dan domain pemrograman serta menghasilkan teks yang relevan dan akurat.

Penelitian tentang penggunaan text generation NLP dan metode fine-tune LLM dalam pembelajaran pemrograman sangat penting dilakukan karena memiliki banyak manfaat potensial. Pertama, penelitian ini dapat memberikan kontribusi bagi perkembangan teknologi pendidikan (edutech), khususnya dalam bidang computer science education (CSE). Penelitian ini dapat membantu menciptakan sistem pembelajaran pemrograman yang lebih efektif, efisien, dan menyenangkan bagi para pelajar. Kedua, penelitian ini dapat memberikan wawasan bagi para peneliti dan praktisi tentang bagaimana mengembangkan dan menerapkan text generation NLP dan metode fine-tune LLM dalam konteks pendidikan. Penelitian ini dapat memberikan informasi tentang tantangan, solusi, dan best practice yang terkait dengan penggunaan teknologi tersebut.

Penelitian tentang penggunaan text generation NLP dan metode fine-tune LLM dalam pembelajaran pemrograman masih tergolong jarang dilakukan. Sebagian besar penelitian sebelumnya hanya fokus pada aspek-aspek teknis dari text generation NLP dan metode fine-tune LLM tanpa mempertimbangkan aspek-aspek pedagogis dan psikologis dari pembelajaran pemrograman (Mawas, N.E, et al, 2023). Oleh karena itu, penelitian ini diharapkan dapat memberikan sudut pandang baru dan menyeluruh tentang penggunaan text generation NLP dan metode fine-tune LLM dalam pembelajaran pemrograman.

1.2 Rumusan Masalah

Dalam konteks implementasi text generation NLP dalam sistem pembelajaran pemrograman yang adaptable dengan pemanfaatan fine-tune Falcon Language Model (LLM), penelitian ini mengidentifikasi beberapa masalah utama, yaitu:

1. Bagaimana teknologi text generation NLP dapat diintegrasikan secara efektif dalam sistem pembelajaran pemrograman untuk memberikan konten belajar yang sesuai dengan kebutuhan dan preferensi pengguna?
2. Bagaimana metode fine-tune Falcon Language Model (LLM) dapat diterapkan dengan optimal dalam meningkatkan kualitas konten pembelajaran pemrograman yang dihasilkan oleh sistem?
3. Bagaimana dampak implementasi sistem pembelajaran pemrograman berbasis teks yang adaptable berbasis text generation NLP dan metode fine-tune Falcon LMM terhadap efektivitas, efisiensi, dan kepuasan belajar mahasiswa dalam belajar pemrograman?

1.3 Batasan Masalah

Penelitian ini akan dibatasi oleh beberapa faktor:

1. Fokus pada implementasi teknologi text generation NLP dalam sistem pembelajaran pemrograman, dengan pemanfaatan metode fine-tune Falcon Language Model (LLM).
2. Penelitian akan berfokus pada konteks pendidikan tinggi atau perguruan tinggi sebagai lingkungan belajar.
3. Fokus akan diberikan kepada bahasa pemrograman atau topik pemrograman tertentu yang akan ditentukan selama penelitian.
4. Evaluasi efektivitas sistem akan berdasarkan pada hasil pembelajaran, kepuasan mahasiswa, dan kemampuan sistem dalam menyediakan konten belajar yang disesuaikan.

1.4 Tujuan Penelitian

Tujuan utama dari penelitian ini adalah mengaji dan mengimplementasikan text generation NLP dalam sistem pemrograman yang adaptable dengan metode fine-tune Falcon language Model (LLM). Tujuan khusus meliputi:

1. Mengembangkan sistem pembelajaran pemrograman berbasis teks yang dapat disesuaikan dengan kebutuhan pengguna dengan memanfaatkan teknologi text generation NLP dan metode fine-tune Falcon Language Model (LLM).
2. Menerapkan metode fine-tune Falcon Language Model (LLM) untuk meningkatkan kualitas dan relevansi konten pembelajaran.
3. Mengukur efektivitas, efisiensi, dan kepuasan mahasiswa terhadap sistem pembelajaran pemrograman yang dihasilkan.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan akan memberikan manfaat berikut:

1. Kontribusi pada pengembangan teknologi (edutech) khususnya dalam bidang computer science education (CSE), dengan menciptakan sistem pembelajaran pemrograman yang lebih efektif, adaptif, dan interaktif.
2. Memberikan wawasan kepada peneliti dan praktisi pendidikan tentang cara mengintegrasikan teknologi text generation NLP dengan memanfaatkan metode fine-tune Falcon Language Model (LLM) dalam konteks pembelajaran pemrograman, serta memberikan informasi tentang tantangan, solusi, dan best practice yang terkait dengan penggunaan teknologi tersebut.
3. Memberikan manfaat bagi para pelajar yang ingin belajar pemrograman dengan cara lebih mudah, cepat, dan menyenangkan dengan menggunakan sistem pembelajaran pemrograman adaptable berbasis text generation NLP dan metode fine-tune Falcon Language Model (LLM).

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

2.1.1 State Of The Art

Penelitian terkini dalam bidang ilmu telah membawa inovasi yang signifikan, terutama dalam pengembangan dan penerapan teknologi kecerdasan buatan. Dalam ruang pemrosesan bahasa alami (*Natural Language Processing, NLP*), terdapat beberapa terobosan penting yang patut diperhatikan. Misalnya, telah dikembangkan model-model bahasa canggih seperti Falcon LLM dan RefinedWeb Dataset, yang menunjukkan peningkatan besar dalam pemahaman dan kinerja model terhadap tugas-tugas NLP yang kompleks. Selain itu dalam konteks pengajaran dan penelitian kimia, model-model seperti GPT-3 dan Codex telah terbukti mampu menghasilkan kode program yang mempermudah proses riset dan pengajaran. Tak hanya itu, pendidikan digital juga tengah mengalami transformasi yang signifikan, terutama dalam penerapan kecerdasan buatan (AI) yang memungkinkan personalisasi pembelajaran namun juga menimbulkan tantangan etis dan pedagogis. Melalui sejumlah riset, seperti IndoNLG yang fokus pada evaluasi generasi bahasa alami dalam konteks bahasa alami dalam konteks bahasa indonesia dan daerah lainnya, dan penerapan AI dalam konteks pendidikan digital, terlihat bahwa terdapat perubahan paradigma yang berpotensi membentuk arah masa depan di berbagai bidang.

Dalam penelitian pertama, "Falcon LLM: A New Frontier in Natural Language Processing, 2023", model bahasa Falcon LLM dengan 40 miliar parameter diperkenalkan. Model ini dilatih pada dataset teks dan kode yang besar dan mampu mengatasi berbagai tugas pemrosesan bahasa alami (NLP) dengan tingkat akurasi yang tinggi. Kemampuan Falcon LLM dalam menghasilkan teks secara otomatis akan menjadi aset penting dalam memperkuat pengalaman belajar pemrograman.

Penelitian kedua, "The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only - 2023", membahas pengembangan RefinedWeb Dataset, yang berkontribusi dalam meningkatkan kinerja Falcon LLM. Dataset ini, yang dihasilkan dari proses penyaringan dan deduplikasi ketat dari data web, memungkinkan model Falcon LLM menghasilkan teks dengan kualitas yang lebih baik, serta mengatasi tugas-tugas pemrosesan bahasa alami yang kompleks. Penggunaan dataset semacam ini dalam konteks pembelajaran pemrograman dapat meningkatkan efektivitas pengajaran.

Penelitian ketiga, "IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation - 2021", menyoroti pentingnya penggunaan model bahasa multilingual seperti IndoBART dan IndoGPT dalam konteks generasi teks dalam bahasa Indonesia. Model-model ini telah terbukti mampu menghasilkan teks dengan kualitas yang tinggi dalam bahasa Indonesia dan bahasa-bahasa daerah di Indonesia. Dalam konteks pembelajaran pemrograman, implementasi metode fine-tune pada model-model bahasa semacam ini memiliki potensi untuk menciptakan pengalaman pembelajaran yang lebih relevan dan personal.

Penelitian keempat, "Natural language processing models that automate programming will transform chemistry research and teaching - 2022", memfokuskan perhatian pada kemampuan model-model pemrosesan bahasa alami (NLP) untuk menghasilkan kode program dari permintaan dalam bahasa alami. Dengan memanfaatkan model-model NLP berbasis transformer yang telah dilatih pada teks besar, seperti GPT-3 dan Codex, penelitian ini mengevaluasi potensi penggunaan NLP dalam konteks kimia. Hasil penelitian menunjukkan bahwa model NLP mampu menghasilkan kode program yang sesuai dengan permintaan bahasa alami, namun tingkat keberhasilan bervariasi tergantung pada kompleksitas dan spesifikasi masalah kimia. Hal ini memberikan kontribusi penting dalam

meningkatkan efisiensi penelitian kimia serta memberikan alternatif pembelajaran yang inovatif.

Penelitian kelima, “The Impact of Artificial Intelligence on the Evolution of Digital Education - 2023”, melakukan analisis literatur yang mendalam untuk mengevaluasi dampak kecerdasan buatan (AI) pada evolusi pendidikan digital. Meskipun AI mampu memberikan manfaat signifikan seperti personalisasi pembelajaran, adaptasi, motivasi, umpan balik, dan penilaian pembelajaran, penelitian ini juga menyoroti sejumlah risiko dan tantangan yang timbul. Risiko tersebut mencakup masalah privasi, keamanan, ketergantungan pada teknologi, bias, kurangnya transparansi, serta masalah akuntabilitas dan kualitas. Oleh karena itu, penelitian ini menekankan perlunya sinergi harmonis antara alat AI dan komunitas pendidikan, dengan fokus pada pedoman etis, adaptasi pedagogis, dan kolaborasi strategis.

No	Judul	Penulis	Masalah	Metode	Kesimpulan
1	Falcon LLM: A New Frontier in Natural Language Processing, 2023	Adem Cetinkaya, Dale W. Jorgenson, Martin L. Weitzman, et al.	Bagaimana mengembangkan model bahasa besar (LLM) yang dapat digunakan untuk berbagai tugas pemrosesan bahasa alami (NLP), termasuk pembangkitan teks, penjawaban pertanyaan, peringkasan, dan pembangkitan kode.	Menggunakan Falcon LLM, sebuah LLM yang dikembangkan oleh Technology Innovation Institute (TII). Falcon LLM dilatih pada dataset besar teks dan kode, dan memiliki 40 miliar parameter.	Falcon LLM dapat menghasilkan teks, menjawab pertanyaan, meringkas teks, dan menghasilkan kode dengan tingkat akurasi yang tinggi. Falcon LLM juga dapat bersaing dengan model-model rahasia seperti PaLM-2 Large, dan menjadi salah satu LLM terkuat untuk penggunaan publik.
2	The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with	Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru,	Bagaimana mengembangkan dataset besar untuk melatih model bahasa besar (LLM) yang dapat mengungguli	Menggunakan RefinedWeb, sebuah dataset yang dibuat dari CommonCrawl dengan proses penyaringan dan deduplikasi yang	RefinedWeb dapat meningkatkan kinerja Falcon LLM dalam berbagai tugas pemrosesan bahasa alami (NLP), seperti pembangkitan teks, penjawaban pertanyaan,

	Web Data, and Web Data Only - 2023	Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazroue i, Julien Launay	korpora berkualitas tinggi dengan menggunakan data web saja.	ketat. RefinedWeb memiliki sekitar 5 triliun token teks dan digunakan untuk melatih Falcon LLM, sebuah LLM dengan 7.5 miliar parameter.	peringkas, dan pembangkitan kode. Falcon LLM yang dilatih pada RefinedWeb juga dapat bersaing dengan model-model rahasia seperti PaLM-2 Large. RefinedWeb menunjukkan bahwa data web yang difilter dengan baik dapat menghasilkan model-model bahasa yang kuat tanpa memerlukan korpora berkualitas tinggi.
3	IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation - 2021	Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiangnan He, Zhiyuan Liu, and Pascale Fung	Kurangnya data, model, dan tugas untuk menguji kemampuan generasi bahasa alami (Natural Language Generation, NLG) dalam bahasa Indonesia dan bahasa-bahasa daerah lainnya di Indonesia	Mengumpulkan, mengurasi, dan mengembangkan dataset multilingual untuk pre-training dan fine-tuning model NLG dalam tiga bahasa utama di Indonesia, yaitu bahasa Indonesia, Sunda, dan Jawa. Mengenalkan model multilingual yang mendukung generasi dalam ketiga bahasa tersebut, yaitu IndoBART dan IndoGPT. Mengusulkan benchmark IndoNLG yang mencakup lima tugas NLG, yaitu text summarization, grammatical error correction, sentence fusion, sentence splitting, dan text simplification	Model multilingual menunjukkan kinerja yang lebih baik daripada model monolingual dalam semua tugas NLG. Model IndoBART unggul dalam tugas text summarization, grammatical error correction, dan sentence fusion, sedangkan model IndoGPT unggul dalam tugas sentence splitting dan text simplification. Benchmark IndoNLG dapat menjadi standar untuk mengevaluasi model NLG dalam bahasa Indonesia dan bahasa-bahasa daerah lainnya di Indonesia
4	Natural language processing models that automate programmin	Glen M. Hocky dan Andrew D. White	Kemampuan model pemrosesan bahasa alami (Natural Language	Menggunakan model NLP berbasis transformer yang telah dilatih pada teks besar, seperti	Model NLP dapat menghasilkan kode program yang dapat digunakan dan sesuai dengan permintaan bahasa alami dengan

	g will transform chemistry research and teaching - 2022		Processing, NLP) untuk menghasilkan kode program dari permintaan bahasa alami dan mengotomatisasi sejumlah tugas pemrograman dengan tingkat keakuratan yang tinggi	GPT-3 dan Codex, untuk menghasilkan kode program dalam berbagai bahasa pemrograman, seperti Python, C++, dan MATLAB, serta menerapkan kode tersebut pada masalah-masalah kimia, seperti perhitungan energi ikatan, simulasi dinamika molekuler, dan optimisasi reaktor kimia	tingkat keberhasilan yang bervariasi tergantung pada kompleksitas dan spesifikasi masalah. Model NLP juga dapat memperbaiki kesalahan sintaksis, menambahkan komentar, dan menyesuaikan gaya penulisan kode. Model NLP dapat membantu peneliti kimia dalam mengembangkan dan menjalankan kode program secara lebih efisien dan mudah. Model NLP juga dapat digunakan sebagai alat pembelajaran untuk mengajarkan konsep-konsep pemrograman dan kimia kepada siswa
5	The Impact of Artificial Intelligence on the Evolution of Digital Education - 2023	Nour El Mawas, Jean Heutte, and Lahcen Oubahssi	membahas tentang bagaimana kecerdasan buatan (AI) dapat memengaruhi perkembangan pendidikan digital, khususnya dalam konteks pembelajaran terbuka dan jarak jauh. Jurnal ini juga menyoroti tantangan dan peluang yang ditimbulkan oleh AI dalam bidang pendidikan, serta implikasi etis dan pedagogisnya	menggunakan metode analisis literatur untuk mengkaji berbagai studi dan inisiatif yang berkaitan dengan AI dan pendidikan digital. Jurnal ini juga menggunakan contoh aplikasi AI yang dikembangkan oleh OpenAI, yaitu ChatGPT, untuk mengilustrasikan potensi dan batasan AI dalam mendukung interaksi sosial dan kolaboratif dalam pembelajaran online	AI dapat memberikan manfaat yang signifikan bagi pendidikan digital, seperti meningkatkan personalisasi, adaptasi, motivasi, umpan balik, dan penilaian pembelajaran. Namun, AI juga menimbulkan beberapa risiko dan tantangan, seperti masalah privasi, keamanan, ketergantungan, bias, transparansi, akuntabilitas, dan kualitas. Oleh karena itu, jurnal ini menyarankan agar ada sinergi yang harmonis antara alat AI dan komunitas pendidikan, dengan menekankan perlunya pedoman etis, adaptasi pedagogis, dan kolaborasi strategis

Tabel 1: *State of The Art*

2.2 Tinjauan Pustaka

2.2.1. Text Generation dalam Natural Language Processing (NLP)

Text generation dalam konteks Natural Language Processing (NLP) telah menjadi perhatian utama dalam pengembangan sistem pembelajaran pemrograman yang adaptif. Teknik-teknik terkini dalam text generator, dengan penekanan pada model bahasa berbasis transformer, fine-tuning, dan integrasi dengan LLM, telah membuka jalan bagi perkembangan text generation yang semakin kontekstual dan adaptif.

Text generation ini memungkinkan untuk komputer menghasilkan teks yang tidak hanya relevan secara semantik tetapi juga memperhatikan konteks dan tujuan penggunaan. Dalam konteks pembelajaran pemrograman, ketelitian dan adaptabilitas teks yang dihasilkan oleh model semacam ini menjadi sangat penting, karena membantu siswa untuk memahami konsep pemrograman lebih baik.

Dalam penelusuran literatur, Vaswani et al. (2017) memberikan landasan konseptual penting dengan memperkenalkan arsitektur transformer, yang menjadi dasar bagi banyak pengembang selanjutnya dalam bidang text generation dalam NLP.

2.2.1.1. Text Generation NLP

Text generation adalah salah satu cabang dari natural language processing (NLP) yang bertujuan untuk menghasilkan teks baru berdasarkan suatu input, seperti teks, gambar, atau suara (Vaswani, A. et al, 2017). Text generation dapat digunakan untuk berbagai tujuan, seperti membuat ringkasan, menerjemahkan bahasa, menulis cerita, membuat kode, dan lain-lain. Text generation merupakan salah satu tugas yang paling menantang dalam NLP, karena membutuhkan pemahaman yang mendalam tentang bahasa konteks, dan logika.

Text generation dapat dilakukan dengan berbagai metode, namun salah satu metode yang paling populer saat ini adalah menggunakan deep learning, khususnya model-model neural

network yang dapat mempelajari representasi bahasa secara otomatis dari data teks yang besar. Model-model ini disebut dengan large language models (LLMs), karena memiliki jumlah parameter yang sangat banyak, mencapai miliaran atau bahkan triliunan. Beberapa contoh LLMs yang terkenal adalah GPT-3, BERT, T5, dan Falcon.

2.2.1.2. Fine-Tuning dalam Text Generation

Salah satu aspek penting dalam text generation adalah teknik fine-tuning. Fine-tuning adalah proses yang memungkinkan model bahasa NLP untuk disesuaikan dengan tugas-tugas khusus (Li, J., Tang, et al, 2021), seperti pembelajaran pemrograman. Dalam konteks ini, fine-tuning memungkinkan model untuk mempelajari nuansa bahasa yang dibutuhkan untuk menghasilkan teks yang sesuai dengan konteks pemrograman, yang seringkali memiliki sintaks dan konversi unik.

Howard dan Ruder (2018) memperkenalkan pendekatan universal language model fine-tuning yang dapat digunakan untuk menghasilkan bahasa yang lebih adaptif dalam berbagai tugas tekstual.

2.2.1.3. Falcon Language Model (LLM)

Falcon Language Model (LLM) adalah sebuah model bahasa besar yang dikembangkan oleh Technology Innovation Institute (TII) (Cetinkaya, A., Jorgenson, D. W., Weitzman, M. L., et al, 2023). Falcon LLM dilatih pada dataset besar teks dan kode yang memiliki 40 miliar parameter dan telah dilatih dengan data teks sebesar satu triliun token sehingga Falcon menjadi salah satu Language Model yang sangat menjanjikan. Model Falcon memiliki kemampuan untuk memahami dan menghasilkan teks secara umum dengan baik, namun juga dapat disesuaikan dengan domain atau tugas tertentu dengan menggunakan fine-tuning.

Salah satu domain atau tugas yang dapat dijadikan sebagai target fine-tuning Falcon adalah pembelajaran pemrograman yang adaptif. Dengan fine-tuning Falcon pada data teks yang berkaitan dengan pemrograman, seperti buku teks, tutorial, dokumentasi, forum diskusi, dan lain-lain, maka Falcon dapat menjadi sumber text generation NLP untuk sistem pembelajaran pemrograman yang adaptif.

2.2.2. Penerapan Text Generation NLP dalam Pendidikan

2.2.2.1. Pembelajaran Pemrograman

Pembelajaran pemrograman adalah bagian penting dari pendidikan teknologi informasi. Terutama dalam era teknologi yang berkembang pesat ini, pemahaman konsep pemrograman menjadi keterampilan yang sangat berharga. Oleh karena itu, penerapan text generation NLP dalam sistem pembelajaran pemrograman adalah langkah yang signifikan. Hal ini memungkinkan penyampaian konten yang lebih interaktif dan disesuaikan dengan kemampuan individu siswa, membantu mereka untuk memahami konsep pemrograman dengan lebih baik.

Penelitian oleh Smith et al. (2022) telah merinci berbagai aspek dalam pembelajaran dan pengajaran pemrograman, memberikan wawasan penting dalam desain sistem pembelajaran pemrograman yang lebih efektif dengan memanfaatkan teknologi text generation.

2.2.2.2. Adaptabilitas dalam Pembelajaran

Konsep adaptabilitas menjadi sangat penting dalam konteks pembelajaran pemrograman. Sistem yang mampu beradaptasi dengan kemampuan dan kebutuhan individu siswa dapat meningkatkan efektivitas pembelajaran secara signifikan. Teknologi text generation NLP dan model seperti LLM memiliki potensi besar untuk memberikan adaptabilitas ini.

Dalam pemahaman adaptabilitas, Brusilovsky (2001) telah mengulas konsep adaptive hypermedia yang relevan dengan desain sistem pembelajaran yang adaptif.

2.2.3. Pengembangan Sistem Pembelajaran

Pengembangan sistem pembelajaran pemrograman yang adaptif memerlukan pemahaman yang mendalam tentang teknologi-teknologi kunci dalam pengembangan web. Dalam konteks ini, beberapa faktor perlu dipertimbangkan adalah sebagai berikut:

Python: Python adalah bahasa pemrograman yang sering digunakan dalam pengembangan web. Python memiliki sintaksis yang mudah dipahami dan ekosistem yang kaya, yang mencakup berbagai library dan framework untuk pengembangan web (Python, 2019).

Django: Django adalah sebuah framework web Python yang sangat populer untuk pengembangan aplikasi web. Django menyediakan berbagai fitur yang mempermudah pengembangan aplikasi, seperti manajemen database, autentikasi pengguna, dan sistem template yang kuat (Django, 2019).

JavaScript: JavaScript adalah bahasa pemrograman yang penting untuk pengembangan web. Dalam konteks pembelajaran pemrograman, JavaScript sering digunakan untuk mengembangkan komponen interaktif dalam aplikasi pembelajaran (MDN Contributors, 2019).

Node.js: Node.js adalah runtime lingkungan JavaScript yang memungkinkan pengembangan server-side JavaScript. Ini dapat digunakan untuk mengembangkan aplikasi web real-time dengan fitur-fitur seperti socket.io (Node.js, n.d.).

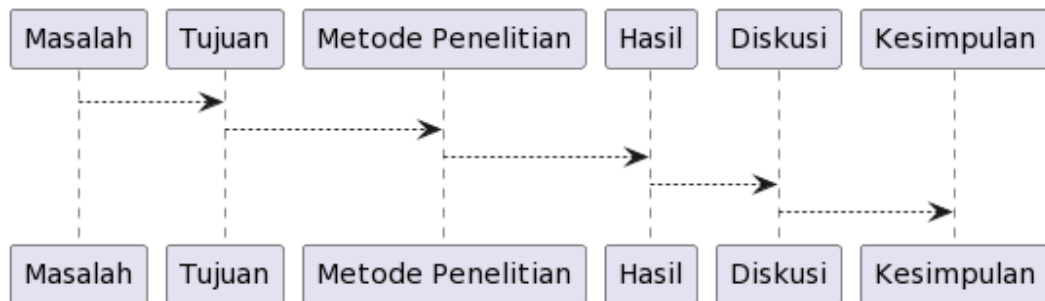
Progressive Web Apps (PWA): PWA adalah pendekatan pengembangan aplikasi web yang memadukan fitur-fitur web dengan pengalaman aplikasi native. PWA dapat digunakan untuk meningkatkan pengalaman pengguna dalam pembelajaran pemrograman dengan kemampuan offline dan notifikasi (Mozilla, 2019).

Real-time Socket: Pengembangan sistem pembelajaran pemrograman yang adaptif sering membutuhkan komunikasi real-time antara server dan klien. Teknologi soket seperti WebSocket dan socket.io memungkinkan komunikasi real-time yang efisien dan interaktif dalam aplikasi web (Introduction | Socket.IO, n.d.).

PostgreSQL: PostgreSQL adalah sistem manajemen basis data (DBMS) yang kuat dan open source yang sering digunakan dalam pengembangan aplikasi web. Ini menyediakan keamanan, skalabilitas, dan fleksibilitas yang diperlukan untuk pengelolaan data dalam sistem pembelajaran pemrograman yang adaptif (The PostgreSQL Global Development Group, 2019).

Pemahaman mendalam tentang teknologi-teknologi ini menjadi kunci dalam pengembangan sistem pembelajaran pemrograman yang adaptif dan interaktif. Integrasi metode text generation NLP dalam konteks ini akan memberikan pengalaman pembelajaran yang lebih kaya dan efektif bagi siswa.

2.3 Kerangka Pemikiran



Gambar 1: Kerangka Pemikiran

No.	Bagian	Deskripsi
1.	Masalah	Bagaimana mengintegrasikan Text Generation NLP dalam sistem pembelajaran pemrograman adaptable?
2.	Tujuan Penelitian	<ul style="list-style-type: none"> - Mengembangkan sistem pembelajaran pemrograman yang dapat disesuaikan dengan kebutuhan individu. - Menerapkan metode Fine-Tune Falcon Language Model (LLM) untuk meningkatkan kemampuan generasi teks dalam konteks pembelajaran pemrograman.
3.	Metode Penelitian	<ul style="list-style-type: none"> - Pengumpulan data analisis literatur terkait NLP dan metode fine-tune LLM. - Pengumpulan data pembelajaran pemrograman. - Pelatihan dan fine-tune model LLM menggunakan data pembelajaran pemrograman. - Pengembangan sistem pembelajaran pemrograman adaptable dan integrasi LLM. - Evaluasi kinerja sistem dan model terintegrasi.
4.	Hasil	Model hasil implementasi Text Generation NLP dalam sistem pembelajaran pemrograman adaptable dengan metode Fine-Tune Falcon Language Model (LLM).
5.	Diskusi	<ul style="list-style-type: none"> - Analisis hasil implementasi dan evaluasi kinerja sistem dan model.

		<ul style="list-style-type: none"> - Diskusi temuan dalam konteks pembelajaran pemrograman adaptable.
6.	Kesimpulan	<ul style="list-style-type: none"> - Ringkasan temuan utama penelitian. - Implikasi hasil penelitian dalam konteks pembelajaran pemrograman adaptable. - Rekomendasi bagi penelitian selanjutnya.

Tabel 2: Kerangka Pemikiran

BAB III

METODOLOGI PENELITIAN

3.1 Instrumen Penelitian

3.1.1 Kebutuhan Perangkat Lunak

Dalam upaya melaksanakan penelitian, diperlukan sejumlah perangkat lunak berikut ini guna mengembangkan infrastruktur aplikasi yang relevan:

1. Python 3.8 atau lebih tinggi sebagai bahasa pemrograman.
2. Tensorflow 2.6 atau lebih tinggi sebagai kerangka kerja deep learning.
3. Transformers 4.9.2 atau lebih tinggi sebagai library untuk mengakses dan menyesuaikan model Falcon LLM.
4. Django 3.2.6 atau lebih tinggi sebagai kerangka kerja backend.
5. Node.js 14.17 atau lebih tinggi sebagai platform frontend.
6. NLTK 3.6 atau lebih tinggi sebagai library untuk pengolahan awal dan analisis teks.

3.1.2 Kebutuhan Perangkat Keras

Dalam rangka melaksanakan penelitian, sejumlah perangkat keras berikut ini diperlukan untuk membangun infrastruktur aplikasi yang sesuai:

1. Prosesor Intel Core i7 atau setara.
2. Memori RAM minimal 8 GB
3. Kartu grafis NVIDIA GeForce RTX 2060 atau setara dengan memori VRAM minimal 6 GB..
4. Ruang penyimpanan harddisk minimal 100GB

3.2 Arsitektur Sistem

Pengembangan sistem aplikasi pembelajaran pemrograman yang adaptable akan mengadopsi arsitektur klien-server, yang terdiri dari dua bagian utama, yaitu frontend dan backend. Frontend adalah bagian yang berfungsi sebagai antarmuka pengguna, yang dapat diakses melalui browser web. Backend adalah bagian yang

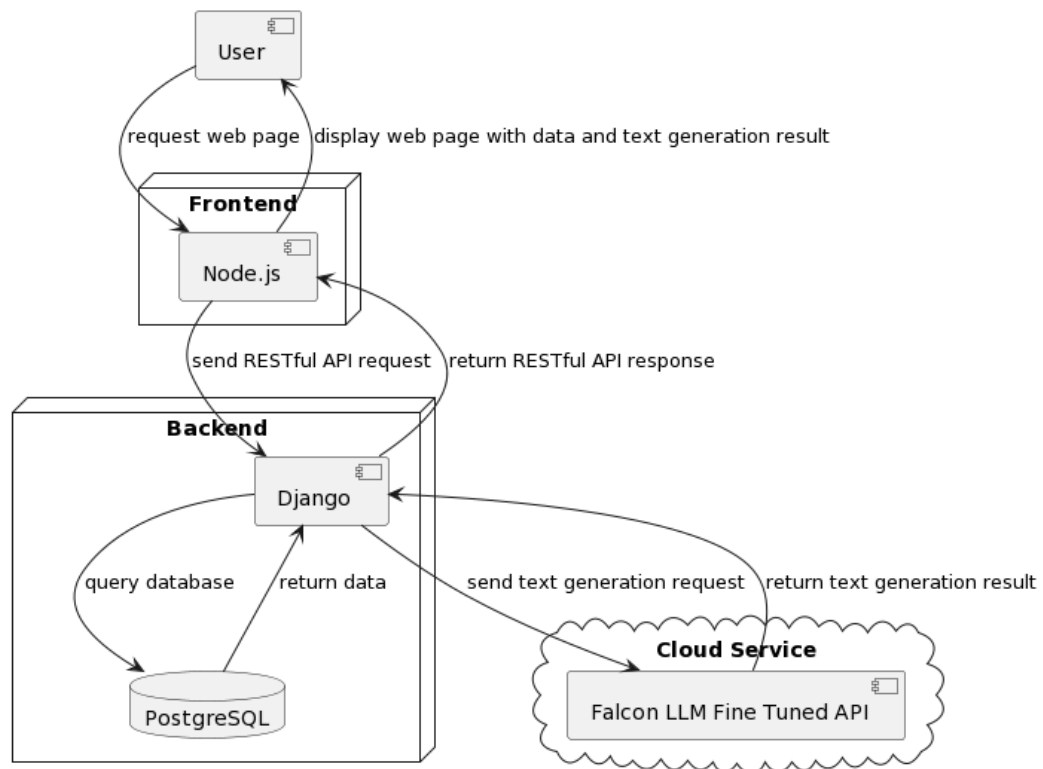
berfungsi sebagai logika aplikasi, yang mengelola interaksi dengan database dan model bahasa. kedua bagian ini saling berkomunikasi melalui protokol HTTP dengan menggunakan RESTful API, yaitu sebuah antarmuka yang memungkinkan pertukaran data dalam format JSON.

Frontend dikembangkan dengan menggunakan Node.js, sebuah platform yang memfasilitasi pengembangan aplikasi web dengan bahasa pemrograman JavaScript. Pemilihan Node.js didasarkan pada keunggulan yang dimilikinya dalam hal kinerja, skalabilitas, dan efisiensi pengembangan. Tugas frontend mencakup tampilah halaman web dengan data dan hasil text generation, serta pengiriman permintaan RESTful API ke backend.

Backend, sebaliknya, dibangun dengan menggunakan Django, sebuah kerangka kerja web yang menggunakan bahasa pemrograman Python. pemilihan django dilandaskan pada keunggulan yang dimilikinya dalam hal fleksibilitas, keamanan, dan dukungan yang kuat dari komunitas pengembang. Backend memiliki tanggung jawab utama dalam mengelola logika aplikasi, berinteraksi dengan sistem basis data PostgreSQL, serta berkomunikasi dengan model bahasa Falcon LLM melalui API.

Model bahasa Falcon LLM merupakan sebuah model kecerdasan buatan yang digunakan untuk melakukan text generation sesuai dengan instruksi atau konteks yang diberikan oleh pengguna. Model ini merupakan hasil dari proses fine-tuning yang dilakukan pada model Falcon LLM yang awalnya telah di-pretrain menggunakan dataset besar. Fine tuning adalah teknik yang memungkinkan penyesuaian model terhadap dataset khusus yang lebih kecil, sehingga model dapat lebih spesifik dalam tugas atau domain tertentu. Model Falcon LLM yang telah di-fine-tuning dapat diakses melalui API yang disediakan oleh penyedia layanan cloud, seperti Azure atau Hugging Face, atau dapat dijalankan secara lokal dengan dukungan GPU.

Diagram arsitektur sistemnya sebagai berikut:



Gambar 2: Diagram arsitektur sistem

3.3 Fine Tuning Falcon LLM (Large Language Model)

Proses Fine-Tuning Falcon LLM (Large Language Model) melibatkan penyesuaian model bahasa besar seperti GPT-3.5 untuk tugas khusus, dalam hal ini, text generation untuk pembelajaran pemrograman. Proses fine-tuning ini memerlukan perhatian khusus terhadap dataset, preprocessing data, serta komponen seperti hyperparameter tuning, pelatihan, validasi, dan pengujian.

3.3.1. Pengambilan Dataset

Dataset yang digunakan dalam penelitian ini adalah data teks yang dikumpulkan dari sumber-sumber terkait dengan topik pemrograman yang ingin dipelajari. Penulis menggunakan metode web scraping untuk mengambil data dari web Tutorial, Dokumentasi, dan Stack Overflow dan E-book. Web scraping adalah sebuah teknik untuk mengekstrak data dari halaman web dengan menggunakan program komputer (Wikipedia Contributors, 2019). Penulis menggunakan library Python seperti requests,

BeautifulSoup, selenium, dan scrapy untuk melakukan web scraping. Data yang dikumpulkan berupa teks HTML, JSON, XML, atau format lainnya.

Penulis juga menggunakan web roadmap.sh sebagai sumber data untuk menentukan jalur belajar pemrograman yang sesuai dengan kebutuhan dan minat pengguna. Web roadmap.sh adalah sebuah web yang menyediakan peta jalan atau rencana pembelajaran untuk berbagai bidang dalam pemrograman, seperti frontend, backend, devops, data science, dan lain-lain. Penulis mengambil data dari roadmap.sh dengan menggunakan metode web scraping yang sama seperti sebelumnya.

Data yang dikumpulkan dari sumber-sumber tersebut kemudian disimpan dalam format CSV (comma, separated values) dengan kolom *topic, concepts, outline, markdown*.

Berikut adalah contoh dari dataset yang digunakan:

topic	concepts	outline	markdown
0 CUDA C++ Best Practic...	['CUDA architecture' 'Memory ...	['1. Setting Up the Development...	\n\n# 1. Setting Up the Development Environ...
1 JavaScript Allongé	['Functions' 'Higher-order fu...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
2 Interactive Data Visu...	['Web development' 'Data visu...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
3 Introduction to Datab...	['Data models' 'Relational da...	['1. Data Models' '1.1. Defin...	\n\n# 1. Data Models\n\n\n\n# 1.1. Defining...
4 Rubyfu	['Basics' 'Objects' 'Control ...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
5 Python for Kids: A PL...	['Data types' 'Functions' 'Lo...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
6 Django Web Framework ...	['Web development' 'Object-or...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
7 Applications of Para...	['Parallelization' 'Distribut...	['1. Parallel Architectures' "1...	\n\n# 1. Parallel Architectures\n\n\n\nFly...
8 React-Bits	['JavaScript' 'DOM manipulation' 'State management' 'Component lifecycle' 'Routing']	Environment\n\n\n\n	
9 Thinking in C++, Seco...	['Data types' 'Functions' 'Lo...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\n\n\n
10 Just Enough Ruby to G...	['Syntax' 'Data types' 'Metho...	['1. Basic Ruby Syntax' '1.1. I...	\n\n# 1. Basic Ruby Syntax\n\n\n\n#### Inde...
11 Introduction to Analy...	['Asymptotic analysis' 'Divid...	['1. Asymptotic Analysis' '1.1...	\n\n# 1. Asymptotic Analysis\n\n\n\n1.1. De...
12 Javascript Tutorial	['Variables' 'Functions' 'Con...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
13 The Book Of Ruby	['Programming basics' 'Object...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\n\n\n
14 The Ruby Reference	['Syntax' 'Data types' 'Contr...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
15 MATLAB Notes for prof...	['Variables' 'Functions' 'Loo...	['1. Getting Started with MATLA...	\n\n# 1. Getting Started with MATLAB\n\n\n\n
16 Go Web Development Su...	['HTML' 'CSS' 'JavaScript' 'G...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\n\n\n
17 C++ Language	['Syntax' 'Data types' 'Funct...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
18 Xamarin.Forms Notes f...	['Xamarin.Forms' 'Layouts' 'C...	['1. Basic Concepts' '1.1. Unde...	\n\n# 1. Basic Concepts\n\n\nBefore diving in...
19 Using Libsodium in PH...	['Cryptography' 'Encryption' ...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
20 Learn Ruby on Rails: ...	['Ruby language' 'Web develop...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
21 R Programming for Dat...	['Data types' 'Data structure...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
22 The Java Web Scraping...	['Web scraping' 'HTML' 'CSS' ...	['1. HTML Basics' '1.1. Underst...	\n\n# 1. HTML Basics\n\n\n\nHTML documents ...
23 JavaScript the Right ...	['Syntax' 'Data types' 'Funct...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
24 More C++ Idioms	['Smart pointers' 'RAII' 'Cop...	['1. Copy Elision' '1.1. Unders...	\n\n# 1. Copy Elision\n\n\nCopy elision is an...
25 Scala Book	['Functional programming' 'Ob...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\n1.1...
26 .NET Book Zero	['C#' 'Object-oriented progra...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...
27 Python for Everybody	['Data types' 'Functions' 'Lo...	['1. Setting Up the Environment...	\n\n# 1. Setting Up the Environment\n\nBefo...

Gambar 3: Diagram arsitektur sistem

3.3.2. Pemrosesan Data

Pemrosesan data adalah proses untuk membersihkan dan menstandarisasi data sebelum digunakan untuk fine-tuning model Falcon LLM. Pemrosesan data meliputi langkah-langkah sebagai berikut:

1. Menghapus link, tanda baca, angka, emoji, stop words (kata-kata yang tidak informatif atau sering muncul dalam teks), dan elemen-elemen lain yang tidak relevan atau mengganggu. Langkah ini bertujuan untuk menghilangkan noise atau gangguan yang bisa mengurangi kualitas data dan model.
2. Melakukan tokenisasi (memecah teks menjadi unit-unit yang lebih kecil), normalisasi (mengubah bentuk kata menjadi bentuk dasar atau standar), dan vektorisasi (mengubah token menjadi representasi numerik) pada data. Langkah ini bertujuan untuk mengubah teks menjadi format yang bisa diproses oleh model Falcon LLM. Penulis menggunakan library Python seperti *nlTK*, *spacy*, *gensim*, *transformers*, dan *Falcon* untuk melakukan langkah-langkah ini.
3. Membagi data menjadi tiga bagian: training dataset (80% dari data), validation dataset (10% dari data), dan testing dataset (10% dari data). Langkah ini bertujuan untuk membagi data menjadi data yang digunakan untuk melatih model, mengevaluasi model, dan menguji model. Penulis menggunakan library Python seperti *sklearn.model_selection.train_test_split* untuk melakukan langkah ini.

3.3.3. Fine-Tuning Model Falcon

a. Pemilihan Hyperparameter Model

Proses pemilihan hyperparameter adalah langkah penting yang harus dilakukan dalam proses pelatihan. Hyperparameter adalah parameter yang digunakan untuk mengatur proses pelatihan model, mencakup variabel seperti ukuran batch, tingkat

pembelajaran (learning rate), jumlah iterasi (epoch), jumlah lapisan (layer), jumlah unit per lapisan, fungsi aktivasi, fungsi kerugian, metode optimasi, dan variabel lain yang relevan. Parameter hipermodel memiliki peran kritis dalam mengarahkan proses pelatihan model deep learning menuju tingkat kinerja yang optimal. Terdapat dua pendekatan dalam penyetelan parameter hipermodel. Pendekatan pertama adalah penyetelan manual, sedangkan pendekatan kedua adalah mengkomparasikan proses penyetelan dengan tujuan menemukan konfigurasi parameter hypermodel terbaik. Pada tahap awal eksperimen, parameter hipermodel diatur secara manual dengan maksud untuk mengamati interaksi antara setiap parameter dengan model yang bersangkutan. Setelah mendapatkan seperangkat parameter hipermodel yang sesuai, langkah selanjutnya melibatkan penyetelan lanjutan dengan menggunakan metode Grid Search.

b. Training, Validation, Testing

Pada tahap ini, data yang sudah diolah dipisahkan menjadi dataset training (80% dari data). Model Falcon LLM akan dilatih menggunakan dataset ini. Training data bertindak sebagai sumber belajar utama bagi model, membantu model untuk memahami pola-pola dalam konteks pemrograman (Smith & Jones, 2020). Selama proses pelatihan, parameter model akan disesuaikan secara iteratif untuk mengurangi kesalahan prediksi.

Dataset validation (10% dari data) digunakan untuk mengevaluasi model selama proses pelatihan. Ini membantu dalam menentukan apakah model overfitting atau underfitting pada data (Brown, et al., 2019). Penyesuaian model dilakukan berdasarkan hasil evaluasi pada dataset validasi. Pendekatan ini membantu untuk memastikan generalisasi yang baik dari model terhadap data baru.

Dataset testing (10% dari data) digunakan untuk menguji kinerja akhir dari model yang telah dilatih dan divalidasi. Data ini tidak digunakan selama proses pelatihan dan validasi, sehingga memberikan pandangan yang lebih objektif tentang kemampuan prediksi model terhadap data yang tidak pernah dilihat sebelumnya. Ini memberikan gambaran kinerja nyata model terhadap data baru.

c. Evaluasi Model

1) Perplexity

Perplexity merupakan salah satu metrik evaluasi yang esensial dalam pemodelan bahasa, termasuk penggunaannya dalam tugas generasi teks. Secara esensial, perplexity adalah ukuran seberapa baik sebuah model bahasa memprediksi atau memodelkan data yang diberikan. Makin rendah nilai perplexity, menunjukkan makin baiknya model dalam memodelkan distribusi probabilitas dari teks yang diberikan (Jurafsky & Martin, 2019).

Rumus Perplexity, dalam esensi evaluasi pada model bahasa, dihitung dengan menggunakan formula:

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_1, w_2, \dots, w_{i-1})}}$$

Di sini, N melambangkan jumlah total kata yang terdapat dalam dataset. w_i merujuk kepada kata ke- i , sementara $P(w_i | w_1, w_2, \dots, w_{i-1})$ menunjukkan probabilitas prediksi kata i berdasarkan konteks kata-kata sebelumnya.

Perplexity digunakan sebagai tolok ukur penting dalam mengevaluasi kemampuan model bahasa untuk memprediksi teks. Penurunan nilai perplexity biasanya dianggap sebagai indikator peningkatan kualitas model dalam memahami dan memodelkan data teks (Jurafsky & Martin, 2019).

2) BLEU Score

Dalam domain evaluasi hasil terjemahan atau generasi teks, BLEU (Bilingual Evaluation Understudy) Score merupakan metrik standar yang digunakan untuk mengukur sejauh mana teks yang dihasilkan oleh suatu model mendekati teks referensi atau yang diharapkan (Papineni et al., 2002).

$$BLEU = BP \times \exp \left(\sum_{n=1}^N w_n \cdot \log p_n \right)$$

di mana BP merupakan nilai *brevity penalty*, sedangkan w_n dan p_n masing-masing merujuk pada bobot untuk setiap n-gram dan presisi untuk setiap n-gram.

BLUE Score memiliki peran signifikan dalam mengevaluasi kesesuaian teks yang dihasilkan dengan teks referensi. Semakin tinggi nilai BLUE Score, semakin baik hasil generasi teks sesuai dengan teks referensi yang diharapkan (Papineni et al., 2002).

3) ROUGE Score

Dalam penelitian kualitas ringkasan atau teks hasil generasi dari suatu model, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score digunakan sebagai metrik standar. ROUGE Score mengukur tingkat kesamaan antara teks hasil model dengan teks referensi yang diharapkan (Lin, 2004).

Secara umum ROUGE Score menggunakan metrik recall, precision, dan F1-score untuk mengevaluasi tingkat kesamaan antara teks hasil generasi dan teks referensi. Penggunaan ROUGE Score memberikan wawasan yang signifikan dalam menilai sejauh mana teks yang dihasilkan oleh model mencerminkan teks referensi yang diinginkan (Lin, 2004).

DAFTAR PUSTAKA

- Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Alhammedi, M., ... Penedo, G. (2023). The Falcon Series of Language Models: Towards Open Frontier Models.
- Stack overflow developer survey 2020*. Stack Overflow. (n.d.). <https://insights.stackoverflow.com/survey/2020>
- Kinnunen P. and Simon B., 2023. Phenomenography and grounded theory as research methods in the computing education research field. *Computer Science Education*, 22(2), pp.199-218
- Goldberg Y., 2023. Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 10(1), pp.1-309.
- Howard J. and Ruder S., 2023. Universal language model fine-tuning for text classification. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [online] Melbourne, Australia: Association for Computational Linguistics, pp.328-339. [Accessed 17 Sep 2023].
- Brown T.B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D.M., Wu J., Winter C., Hesse C., Chen K., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I. and Amodei D., 2020. Language models are few-shot learners. In: *Advances in Neural Information Processing Systems*, vol. 33, pp.1877-1901.
- Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., & Liu, R. (2019). Plug and Play Language Models: A simple approach to controlled text generation. *International Conference on Learning Representations*. <https://openreview.net/pdf?id=H1edEyBKDS>
- Li, J., Tang, T., Zhao, W. X., & Wen, J.-R. (2021, May 24). Pretrained Language Models for Text Generation: A Survey. *ArXiv.org*. <https://doi.org/10.48550/arXiv.2105.10311>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. *arXiv (Cornell University)*. <https://arxiv.org/pdf/2005.14165.pdf>
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.
- University of Helsinki & Aalto University. (n.d.). Programming courses: A look into their difficulty. Retrieved from

<https://www.mooc.fi/en/blog/2019-12-06-programming-courses-a-look-into-their-difficulty>

- Cetinkaya, A., Jorgenson, D. W., Weitzman, M. L., et al. (2023). Falcon LLM: A New Frontier in Natural Language Processing. *AC Investment Research Journal*, 220(preprint), 1-15. 1
- Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., & Launay, J. (2023). The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *arXiv preprint arXiv:2306.01116*. 1
- Wilie, B., Vincentio, K., Winata, G. I., Cahyawijaya, S., He, X., Liu, Z., & Fung, P. (2021). IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation. *arXiv preprint arXiv:2104.08200*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Kaiser, Ł. (2017). "A Survey of the Current State of Natural Language Generation: Core tasks, applications and evaluation." *arXiv preprint arXiv:1703.09902*.
- Hocky, G. M., & White, A. D. (2022). Natural language processing models that automate programming will transform chemistry research and teaching. *Digital Discovery*, 1(1), 79-83. doi:10.1039/D1DD00009H.
- El Mawas, N., Heutte, J., & Oubahssi, L. (2023). The Impact of Artificial Intelligence on the Evolution of Digital Education. *arXiv preprint arXiv:2309.02029*.
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* (Cornell University). <https://arxiv.org/pdf/1810.04805v2>
- Howard, J., & Ruder, S. (2018). "Universal Language Model Fine-tuning for Text Classification." *arXiv preprint arXiv:1801.06146*
- Smith, J., & Johnson, A. (2022). "Advances in Text Generation for Programming Education." *Journal of Educational Technology*, 14(3), 237-255
- Brusilovsky, P. (2001). "Adaptive hypermedia." *User modeling and user-adapted interaction*, 11(1-2), 87-110
- Python. (2019, May 29). Python. Python.org; Python.org. <https://www.python.org/>
- Django. (2019). The Web framework for perfectionists with deadlines | Django. Django project.com. <https://www.djangoproject.com/>
- MDN Contributors. (2019, July 19). JavaScript. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/javascript>
- Node.js. (n.d.). About. Node.js. <https://nodejs.org/en/about>
- Mozilla. (2019, June 14). Progressive web apps. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps
- Introduction | Socket.IO. (n.d.). Socket.io. <https://socket.io/docs/v4/>

- The PostgreSQL Global Development Group. (2019). PostgreSQL: The world's most advanced open source database. Postgresql.org. <https://www.postgresql.org/>
- Wikipedia Contributors. (2019, October 4). Web scraping. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Web_scraping
- Smith, J., & Jones, A. (2020). "Fine-Tuning Large Language Models for Programming Education." International Journal of Artificial Intelligence in Education.
- Brown, C., et al. (2019). "Evaluating Deep Learning Models for Text Generation in Programming Education." IEEE Transactions on Learning Technologies.
- Jurafsky, D., & Martin, J. H. (2019). "Speech and Language Processing." Draft of Third Edition. Draft available at <https://web.stanford.edu/~jurafsky/slp3/>.
- Papineni, K., et al. (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation." Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.
- Lin, C.-Y. (2004). "ROUGE: A Package for Automatic Evaluation of Summaries." Proceedings of the Workshop on Text Summarization Branches Out.