

Presentasi Sidang Skripsi

Penalaran Logika Berbasis Framework Translation-Decomposition-Search-Resolve pada Open-Weight Small Language Model dengan Dataset Berbahasa Indonesia

Dipresentasikan oleh :
MIKHAEL DEO BARLI
NPM 1906350572
Jurusan Ilmu Komputer

GARIS BESAR

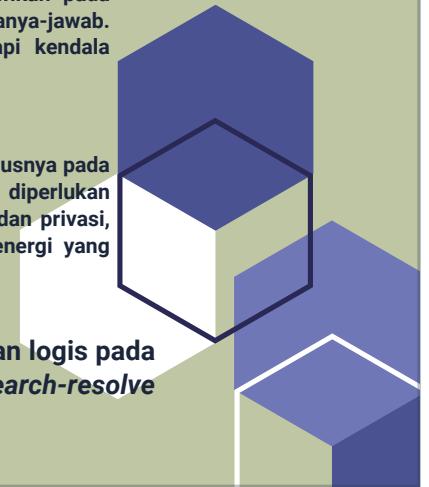
Latar Belakang	01	Decomposition to CNF	06
Rumusan masalah	02	Search and Resolve	07
Landasan Teori	03	Desain Eksperimen	08
Framework Aristotle	04	Hasil dan Analisis	09
Translation to FOL	05	Penutup	10



LATAR BELAKANG

1. Perkembangan *Large Language Model* (LLM) telah mendorong kemajuan signifikan pada berbagai tugas pemrosesan bahasa natural seperti penerjemahan, ringkasan, dan tanya-jawab. Namun, kemampuan LLM untuk melakukan penalaran logis masih menghadapi kendala fundamental.
2. Adanya *shift* dari penggunaan LLM ke SLM karena keterbatasan sumber daya, khususnya pada negara berkembang seperti negara Indonesia, sehingga penggunaan SLM sangat diperlukan agar dapat menjalankan model secara lokal yang menjaminkan kedaulatan data dan privasi, serta menguntungkan secara ekonomi karena dapat mengurangi penggunaan energi yang signifikan jika SLM dibandingkan dengan LLM

Dari 2 poin tersebut, muncullah ide apakah SLM dapat melakukan penalaran logis pada dataset berbahasa Indonesia dengan *framework translation-decompose-search-resolve* pada penelitian ini



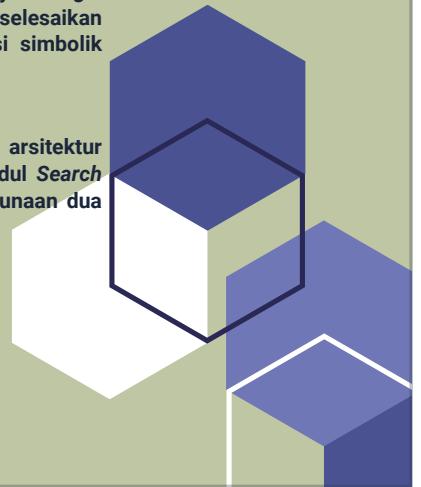
EVOLUSI METODE PENALARAN LOGIS

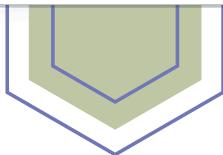
1. **Naive Prompting (Implicit Reasoning)**: Pendekatan paling dasar di mana model diminta langsung menjawab kesimpulan dari premis yang diberikan (misalnya, Zero-shot). Kelemahannya adalah "Curse of Complexity", di mana akurasi model menurun secara eksponensial seiring bertambahnya langkah logika karena model tidak memiliki memori kerja eksternal untuk menyimpan status inferensi perantara
2. **Chain-of-Thought (CoT)**: CoT mendorong model untuk menghasilkan serangkaian langkah penalaran perantara sebelum memberikan jawaban akhir. Metode ini terbukti meningkatkan performa pada tugas aritmatika dan simbolik secara signifikan dengan mengubah pemetaan Input-Output menjadi Input-Reason1-Reason2-Output. Namun, CoT rentan terhadap propagasi kesalahan, jika satu langkah penalaran salah (halusinasi), seluruh kesimpulan akan salah karena tidak ada mekanisme verifikasi eksternal
3. **Tree-of-Thoughts (ToT)**: dikembangkan dari konsep CoT menjadi struktur pohon, memungkinkan model untuk mengeksplorasi berbagai jalur penalaran, melakukan *lookahead*, dan *backtracking* saat menemui jalan buntu. Meskipun lebih kuat, ToT sangat mahal secara komputasi dan masih bergantung pada intuisi probabilistik model itu sendiri untuk mengevaluasi validitas setiap cabang pemikiran.



EVOLUSI METODE PENALARAN LOGIS

4. **Neuro-Symbolic Approaches(Logic-LM & SymbCoT)**: Untuk mencapai ketepatan logika yang strict, pendekatan Neuro-Symbolic mulai diadopsi. *Logic-LM* menggunakan LLM hanya sebagai penerjemah masalah ke dalam kode simbolik (seperti Prolog) , yang kemudian diselesaikan oleh solver deterministik. *SymbCoT* akan mencoba mengintegrasikan verifikasi simbolik langsung ke dalam rantai pemikiran LLM.
5. **Aristotle Framework**: Menyempurnakan pendekatan Neuro-Symbolic dengan arsitektur *Translation-Decompose-Search-Resolve*. Keunggulan utamanya adalah adanya modul *Search Router* yang memangkas ruang pencarian premis yang tidak relevan, serta penggunaan dua jalur pembuktian untuk meminimalkan halusinasi.





RUMUSAN MASALAH

Masalah Pertama

01

Bagaimana penalaran logis pada dataset berbahasa Indonesia dilakukan dengan menggunakan *framework translation-decompose-search-resolve*?

Masalah Kedua

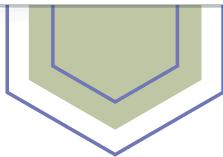
02

Bagaimana perbandingan performa *framework translation-decompose-search-resolve* untuk penalaran logis berbahasa Indonesia antar *open-weight LLM* berparameter rendah atau SLM?

Masalah Ketiga

03

Bagaimana perbandingan performa *framework translation-decompose-search-resolve* dibandingkan *naive prompting* untuk penalaran logis berbahasa Indonesia pada *open-weight LLM* berparameter rendah atau SLM?



TUJUAN PENELITIAN

Tujuan Pertama

01

mengimplementasikan penalaran logis pada dataset berbahasa Indonesia dilakukan dengan menggunakan *framework translation-decompose-search-resolve*?

Tujuan Kedua

02

Mengevaluasi perbandingan performa *framework translation-decompose-search-resolve* untuk penalaran logis berbahasa Indonesia antar *open-weight LLM* berparameter rendah atau SLM?

Tujuan Ketiga

03

Mengevaluasi perbandingan performa *framework translation-decompose-search-resolve* dibandingkan naive prompting untuk penalaran logis berbahasa Indonesia pada *open-weight LLM* berparameter rendah atau SLM?



Sintaks FOL dibangun dari komponen-komponen berikut menurut Brachman and Levesque (2004):

01

Simbol Logis

- Konektif: \neg (Negasi), \wedge (Konjungsi), \vee (Disjungsi), \rightarrow (Implikasi), \leftrightarrow (Bikondisional).
- Kuantor: \forall (Universal), \exists (Eksistensial).
- Variabel: x, y, z, \dots

02

Simbol Non-Logis

- Konstanta: Simbol yang merepresentasikan objek spesifik (misalnya, "Alice", "42").
- Fungsi ($f(x_1, \dots, x_n)$): Memetakan objek ke objek lain. Contoh: AyahDari(Budi).
- Predikat ($P(x_1, \dots, x_n)$): Fungsi yang memetakan tuple objek ke nilai kebenaran (True/False). Contoh: Ayah(Budi, True).



First-Order Logic (FOL), atau Kalkulus Predikat, adalah logika formal yang memperluas logika proposisi dengan memperkenalkan variabel, fungsi, dan kuantor untuk merepresentasikan objek dan relasi di dunia nyata.

Logika proposisi adalah cabang logika yang mempelajari hubungan antara pernyataan-pernyataan yang dapat dinilai benar atau salah

- Simbol Predikat: Menunjukkan relasi atau properti. Contoh: Person, OlderThan, Dog. Ditulis dengan huruf kapital.
- Simbol Fungsi: Menunjukkan fungsi yang memetakan objek ke objek lain. Contoh: fatherOf, successor. Ditulis dengan huruf kecil.



INTERPRETASI DAN KEBENARAN

Interpretasi memetakan:

- Konstanta: Setiap simbol konstanta c dipetakan ke elemen spesifik di D.
- Predikat: Setiap simbol predikat n-ary dipetakan ke himpunan relasi n-ary di D (objek-objek mana yang memiliki sifat tersebut).
- Fungsi: Setiap simbol fungsi f dipetakan ke operasi konkret pada D. Misalnya, jika domainnya adalah bilangan bulat, simbol fungsi plus(x, y) dapat diinterpretasikan sebagai operasi penjumlahan $x + y$.



Mengapa membutuhkan entailment?

Karena kita ingin menarik kesimpulan dari informasi yang sudah tersedia

Kebenaran sebuah kalimat FOL ditentukan oleh sebuah Interpretasi (I) atas Domain (D) yang tidak kosong.



KONSEKUENSI LOGIS (ENTAILMENT)

Dalam sistem logika, kumpulan fakta dan aturan yang kita yakini kebenarannya disebut sebagai Knowledge Base (KB). Menurut Huth and Ryan (2004) $KB \models \alpha$ jika dan hanya jika untuk setiap interpretasi I di mana semua kalimat dalam KB bernilai Benar, maka α juga pasti bernilai Benar. Namun, memeriksa setiap interpretasi secara komputasi adalah mustahil karena jumlahnya bisa tak terbatas. Oleh karena itu, kita menggunakan algoritma inferensi sintaksis (seperti Resolusi) untuk membuktikan validitas tersebut secara otomatis.

Mengapa membutuhkan entailment?

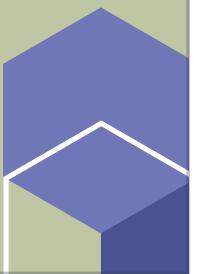
Karena kita ingin menarik kesimpulan dari informasi yang sudah tersedia

Artinya, tidak mungkin ada situasi di mana premis-premis kita benar tetapi kesimpulannya salah.

RESOLUSI

Resolusi bekerja dengan prinsip Proof by Contradiction. Untuk membuktikan bahwa $KB \models a$, kita dapat menunjukkan bahwa himpunan $KB \cup \{\neg a\}$ adalah Unsatisfiable (tidak ada interpretasi yang memenuhi), maka terbukti secara logis bahwa a haruslah benar.

Tanda terjadinya kontradiksi adalah ketika algoritma berhasil menurunkan Klausa Kosong(\square atau False) (Brachman and Levesque (2004)).





KONVERSI KE CONJUNCTIVE NORMAL FORM (CNF)

Agar aturan resolusi dapat diterapkan, formula logika harus diubah ke bentuk standar yang disebut Conjunctive Normal Form (CNF). Menurut Fitting (1996) merinci langkah-langkah algoritma konversi ini sebagai berikut:

1. Eliminasi Implikasi: Ubah $A \rightarrow B$ menjadi $\neg A \vee B$.

2. Geser Negasi ke Dalam: Gunakan hukum De Morgan dan aturan $\neg \forall x(P) \equiv \exists x(\neg P)$.

3. Standardisasi Variabel: Ubah nama variabel agar unik untuk setiap kuantor (misal: $\forall x(P(x)) \vee \forall x(Q(x))$ menjadi $\forall x(P(x)) \vee \forall y(Q(y))$).



KONVERSI KE CONJUNCTIVE NORMAL FORM (CNF)

4. Prenex Normal Form: Pindahkan semua kuantor ke depan formula.

5. Skolemisasi: Menghilangkan kuantor eksistensial (\exists). Variabel y yang terikat oleh \exists diganti dengan Fungsi Skolem ($f(x)$) yang bergantung pada variabel universal sebelumnya. Contoh: $\forall x \exists y (\text{Parent}(x, y))$ menjadi $\forall x (\text{Parent}(x, f(x)))$.

6. Distribusi & CNF: Gunakan aturan distributif untuk mendapatkan bentuk konjungsi dari klausa (AND of ORs).

ATURAN UNIFIKASI RESOLUSI DAN

Aturan resolusi untuk logika predikat adalah:

$$\frac{C_1 \vee L_1, \quad C_2 \vee L_2}{C_1 \vee C_2 \theta}$$

Di mana L_1 dan L_2 adalah literal yang saling berlawanan (misal $P(x)$ dan $\neg P(a)$). Agar mereka bisa saling menghilangkan (cancel out), argumen di dalamnya harus disamakan terlebih dahulu.

Proses penyamaan argumen ini disebut Unifikasi. Unifikasi mencari substitusi θ (disebut Most General Unifier / MGU) yang membuat dua atom menjadi identik secara sintaksis. Contoh: Unifikasi $P(x)$ dan $P(Alex)$ menghasilkan $\theta = \{x / Alex\}$. Tanpa unifikasi, resolusi pada FOL tidak mungkin dilakukan karena variabel pada premis yang berbeda harus diselaraskan.





DATASET PRONTOQA



Data tidak diambil dari internet, melainkan dibangkitkan dari aturan logika formal yang kemudian diterjemahkan ke bahasa alami karena sifatnya yang generatif dan sintetis.

Keunggulan: Kemampuannya mengisolasi kemampuan deduksi dari pengetahuan dunia (world knowledge) dengan menggunakan istilah fiktif (seperti "Wumpus" atau "Tumpus") mencegah model menjawab benar hanya karena hafal fakta (misalnya "Burung bisa terbang").

01

Sistem membuat graf ontologi fiktif

Wumpus → Jompus, Jompus → Tumpus

02

Sistem menurunkan rantai deduksi berdasarkan graf tersebut.

Setiap Wumpus adalah Jompus dan Setiap Jompus adalah Tumpus

03

Simbol-simbol logika diganti dengan kata-kata fiktif atau nyata untuk membentuk kalimat bahasa alami.

Setiap Wumpus adalah Merah. Wumpuses adalah Jompus. Jompus bersifat transparan. Semua Jompus adalah Tumpus

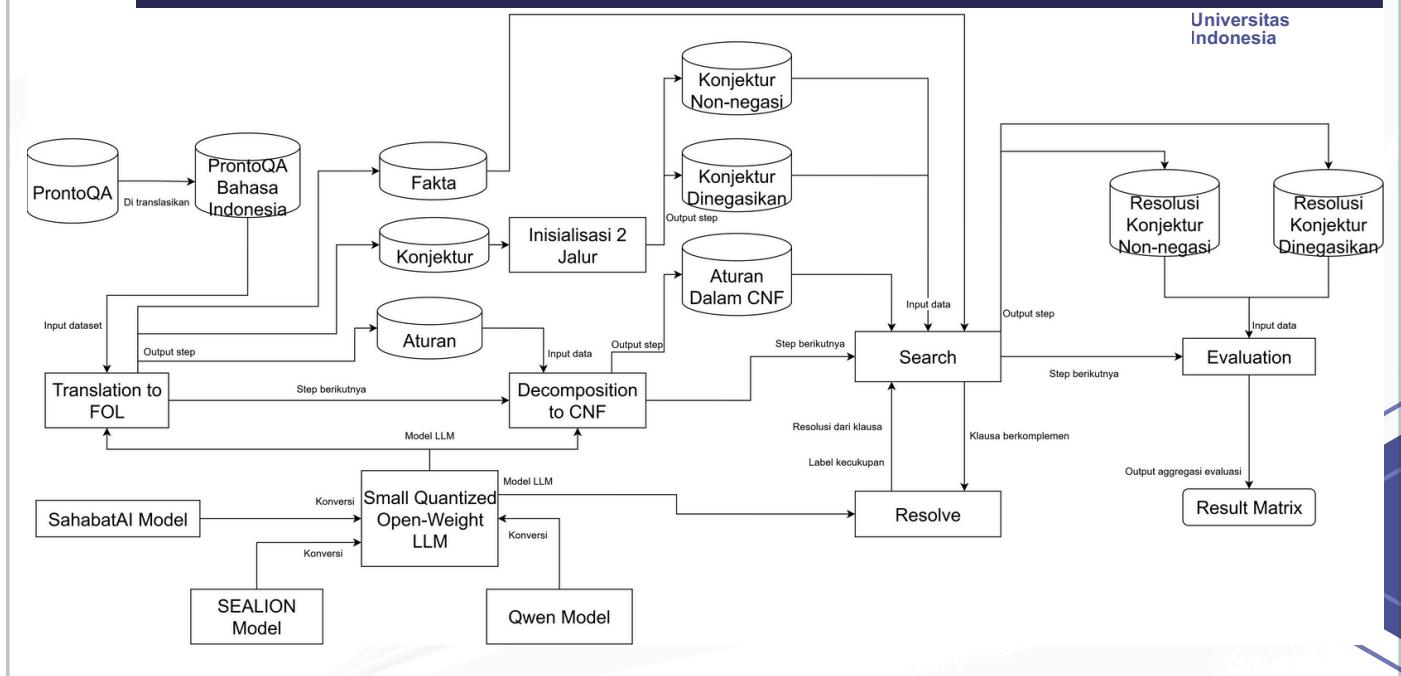
ProntoQA yang dikembangkan oleh Saparov and He (2023).

Model dipaksa untuk melakukan penalaran ("Jika Wumpus adalah Burung, maka Wumpus bisa terbang") berdasarkan aturan yang diberikan di dalam prompt, bukan dari memori pelatihannya.

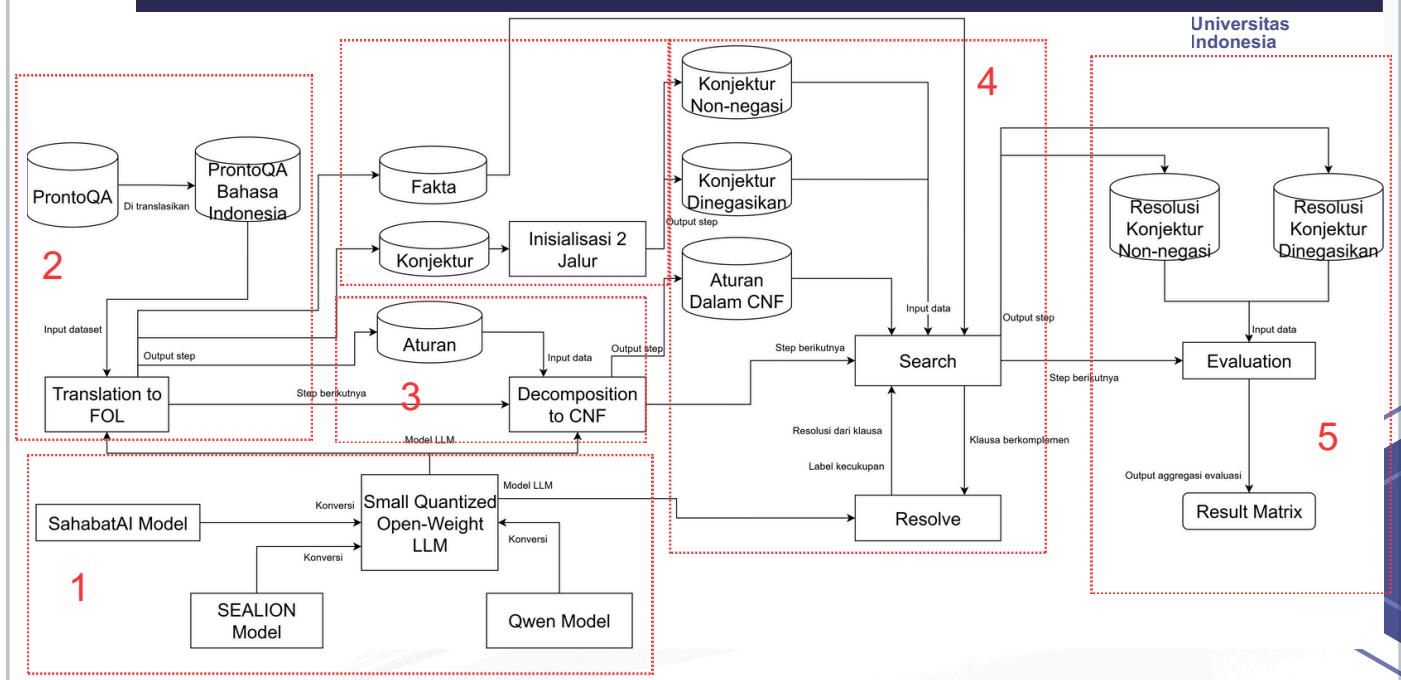
Karena sifatnya generatif dan sintetis, maka dataset ini juga memungkinkan kontrol terhadap "kedalaman penalaran" (reasoning hops), yaitu berapa langkah logika yang diperlukan untuk mencapai kesimpulan (misal: 1-hop, 3-hop, hingga 5-hop).

FRAMEWORK ARISTOTLE

Universitas
Indonesia



FRAMEWORK ARISTOTLE





PEMILIHAN MODEL



Parameter untuk setiap model dibuat sama dengan ketentuan berikut:

- **Temperature:** Ditetapkan ke **0.0** sehingga model hanya memilih token dengan probabilitas tertinggi.
- **Max Tokens:** Adapun token yang dibutuhkan adalah **2500 tokens** untuk mengatasi overhead atau halusinasi jika model tidak secara strict mengikuti format.
- **n_ctx:** Ukuran konteks diatur untuk menentukan berapa banyak token sebelumnya yang dapat dipertimbangkan model saat menghasilkan respons. **n_ctx di set ke 0**
- **n_gpu_layers:** Menentukan jumlah layer model yang dijalankan di GPU untuk akselerasi. **Nilai di set ke -1**

01

Qwen2.5-7B-IT-GGUF

Model ini dipilih untuk karena keunggulannya dalam benchmark logika matematika global. Model ini merepresentasikan kemampuan model global dalam dataset lokal ini

02

SEALIONv3-Llama-8B-IT-GGUF

Model ini dipilih karena sudah melalui CPT(continued pre-training) dalam bahasa di Asia Tenggara. Model ini merepresentasikan kemampuan model regional dalam dataset lokal ini

03

SahabatAlv1-Llama-8B-IT-GGUF

Model ini dipilih karena sudah melalui CPT(continued pre-training) dalam bahasa Indonesia. Model ini merepresentasikan kemampuan model lokal dalam dataset lokal ini

Temperature set: Hal ini memastikan setiap prompt menghasilkan output yang identik.

Max Token set: Dibatasi sesuai dengan panjang output yang wajar untuk setiap tahap dalam pipeline, mencegah generasi yang berlebihan. Proses translasi ke FOL memerlukan output setidaknya 400 token, proses dekomposisi setidaknya memerlukan output 700 token, dan proses pencarian bukti setidaknya memerlukan 1000 token

n_ctx set: Nilai yang lebih besar memungkinkan model memahami konteks yang lebih panjang, namun meningkatkan penggunaan memori. Di set ke 0, sehingga ukuran konteks akan sesuai dengan kemampuan model masing-masing.

n_gpu_layers set: Nilai yang lebih tinggi mempercepat inferensi tetapi memerlukan VRAM yang lebih besar. Nilai di set ke -1 berarti semua komputasi dilakukan di GPU.

SahabatAI dilatih dari SEALIONv2-Llama-8B-IT



TRANSLATION TO FOL



Menerjemahkan narasi bahasa alami menjadi representasi First-Order Logic (FOL) yang terdiri dari Fakta, Aturan Implikasi, dan Konjektur.

Model diberikan few-shot prompt yang berisi contoh pasangan kalimat Indonesia dan format logika target.

01

Fakta

Format: $P(\text{Subjek}, \text{NilaiKebenaran})$.

Contoh: $\text{Wumpus}(\text{Max}, \text{True})$

02

Aturan

Format: $P(\text{Subjek}, \text{NilaiKebenaran}) \ggg P(\text{Objek}, \text{NilaiKebenaran})$.

Contoh: $\text{Wumpus}(\$x, \text{True}) \ggg \text{Manis}(\$x, \text{False})$

03

Konjektur

Format: $P(\text{Subjek}, \text{NilaiKebenaran})$.

Contoh: $\text{Manis}(\text{Max}, \text{True})$

DECOMPOSITION TO CNF



Model diinstruksikan untuk melakukan konversi ke Prenex Normal Form(PNF) dan kemudian ke Conjunctive Normal Form (CNF).

Proses penting di sini adalah Skolemisasi, yaitu penghapusan kuantor eksistensial (\exists) yang sering menjadi sumber ambiguitas bagi model bahasa.

Hasil Decomposition to CNF:

Contoh:

Input: $\text{Wumpus}(\$x, \text{True}) \ggg \text{Manis}(\$x, \text{False})$

Output: $\backslash\text{left}(\text{Wumpus}(x, \text{False}) \backslash\text{lor}$

$\text{Manis}(x, \text{False}) \backslash\text{right}$

01

Prenex Normal Form (PNF)

Konversi FOL ke bentuk PNF, di mana semua kuantor diletakkan di awal formula, diikuti oleh matriks bebas kuantor.

02

Skolemisasi

Penghapusan kuantor eksistensial (\exists), dengan mengganti variabel eksistensial dengan fungsi Skolem.

03

Conjunctive Normal Form (CNF)

Transformasi akhir ke CNF, di mana formula diekspresikan dalam bentuk konjungsi (AND) dari disjungsi (OR) literal.

Tetapi proses skolemisasi tidak berlaku untuk dataset ProntoQA karena dataset ini tidak memerlukan proses skolemisasi karena tidak ada premis yang menggunakan kata seperti "sebagian"

Step-by-step:

Ubah implikasi ke bentuk OR

Input: $\text{Wumpus}(\$x, \text{True}) \ggg \text{Manis}(\$x, \text{False})$

Output: $\text{Wumpus}(\$x, \text{False}) \backslash\text{lor} \text{Manis}(\$x, \text{False})$

Sudah PNF karena tidak membutuhkan standardisasi variabel

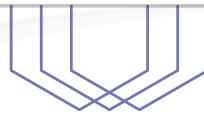
Cek skolemisasi; tidak ada, maka juga sudah berbentuk CNF

Input: $\text{Wumpus}(\$x, \text{True}) \ggg \text{Manis}(\$x, \text{False})$

Output: $\backslash\text{left}(\text{Wumpus}(x, \text{False}) \backslash\text{lor} \text{Manis}(x, \text{False}) \backslash\text{right}$



SEARCH AND RESOLVE



Universitas
Indonesia

Search Resolve merupakan sebuah modul dari *framework* dengan dua fungsi

Logical Search Router

Memilih klausa mana yang relevan untuk diproses dan juga maksimum ronde pencarian atau ruang pencarian

01

Logical Resolver

Menerapkan aturan *resolusi* secara iteratif hingga ditemukan kontradiksi (untuk pembuktian salah) atau hingga ruang pencarian habis.

02



Search Router akan mencari semua klausa yang berkomplemen dan akan dikirimkan ke SLM untuk di cari resolusinya.

Hasil dari SLM akan dikembalikan berupa klausa baru beserta apakah ditemukan kontradiksi dengan label kecukupan

Logical Search Router: Hal ini mencegah proses komputasi yang memerlukan resource terlalu banyak dan yang lama dalam pencarian bukti.



SEARCH AND RESOLVE



Tabel 3.1: Matriks Keputusan untuk Penentuan Label Jawaban (Kesimpulan)

Kondisi (Skenario)	Jalur A: Buktikan S (Cari Kontradiksi di $\neg S$)	Jalur B: Buktikan $\neg S$ (Cari Kontradiksi di S)	Kesimpulan
1	Berhasil (Ada Kontradiksi)	Gagal	TRUE
2	Gagal	Berhasil (Ada Kontradiksi)	FALSE
3	Gagal	Gagal	UNKNOWN
4	Berhasil	Berhasil	SELF-CONTRADICTORY

01

Jalur A: Konjektur Dinegasikan
Asumsikan $\neg S$ (Negasi S), jika ditemukan kontradiksi (Klausa Kosong), maka S terbukti Benar.

02

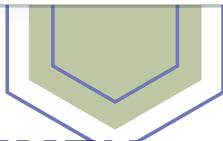
Jalur B: Konjektur non-negasi
Asumsikan S , jika ditemukan kontradiksi, maka $\neg S$ terbukti Benar.

03

Jawaban Akhir
Apabila Jalur A berhasil ditemukan kontradiksi dan Jalur B tidak ditemukan kontradiksi, maka jawaban dari Konjektur tersebut adalah True dan akan dibandingkan dengan ground_truth / jawaban asli untuk data point tersebut

Keterangan:

- Berhasil (Ada Kontradiksi): Resolver menemukan klausa kosong, yang berarti asumsi awal salah, sehingga kebalikannya pasti benar.
- TRUE: Premis mendukung hipotesis S .
- FALSE: Premis menyangkal hipotesis S .
- UNKNOWN: Informasi pada premis tidak cukup untuk membuktikan atau menyangkal S .
- SELF-CONTRADICTORY: Premis itu sendiri tidak konsisten (mengandung konflik internal).



DESAIN EKSPERIMENT

Variabel dalam eksperimen ini:

01

Variabel Kontrol

1. Parameter setiap model dibuat sama dan dataset yang digunakan juga sama.
2. Setiap model terkuantisasi (4-bit GGUF)

03

Variabel Terikat

1. Akurasi (Accuracy): Ketepatan hasil akhir (Benar/Salah) yang diverifikasi terhadap ground truth.
2. Kepatuhan Format (Parsability): Kemampuan model menghasilkan sintaks logika (FOL/CNF) yang valid secara komputasi. Kegagalan menghasilkan format yang benar dianggap sebagai kegagalan penalaran.

02

Variabel Bebas

1. Model: Tiga kategori model yang mewakili spektrum pemahaman bahasa: Global (Qwen), Regional (SEA-LION), dan Nasional (SahabatAI).
2. Metode Prompting: Naive vs Aristotle Framework.



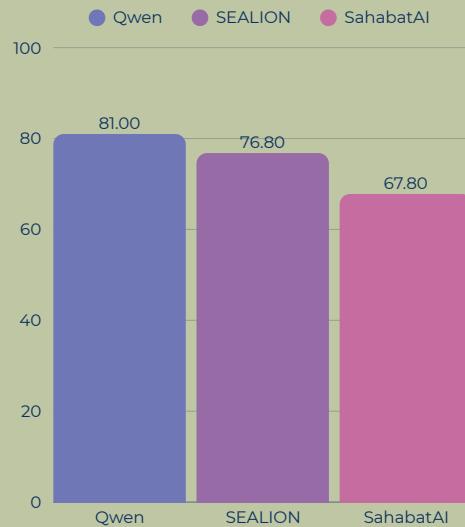
HASIL EKSPERIMENT



Universitas
Indonesia



Naive Prompting Explanation Before Answer



Naive Prompting Explanation After Answer





HASIL EKSPERIMENT



Universitas
Indonesia

Ada beberapa poin yang bisa didapatkan dari hasil eksperimen tersebut:

1. Keunggulan Qwen pada CoT: Qwen2.5 menunjukkan lonjakan performa tertinggi (81%) ketika menggunakan metode Before Answer. Ini konsisten dengan literatur yang menyatakan bahwa model yang dilatih pada korpus kode/matematika besar (seperti Qwen) memiliki kemampuan Chain-of-Thought internal yang kuat (Wei et al. (2022)).

2. Kekuatan Semantik Sahabat-AI: Pada mode After Answer (yang lebih mengandalkan intuisi bahasa langsung), Sahabat-AI unggul (61.4%). Ini mengindikasikan bahwa model ini memiliki pemahaman Bahasa Indonesia yang lebih natural, sehingga intuisi "tebakan"-nya lebih akurat dibandingkan model global.

3. Naive Prompting dengan metode Explanation Before Answer menunjukkan hasil yang lebih baik dibanding Explanation Before Answer karena pada dasarnya SLM merupakan transformer yang menggunakan probabilitas untuk menentukan token selanjutnya ($P(w_t|w_{t-1}, \dots)$), sehingga dengan penjelasan sebelum jawaban akhir merupakan salah satu cara model melakukan Chain-of-Thought untuk menjelaskan bagaimana cara model mendapatkan jawaban tersebut

3. ...+ maka dari itu, Explanation After Answer tidak menambah akurasi karena model cenderung mengjustifikasi jawaban/tebakan yang salah dengan penjelasan yang salah. Walaupun mendapatkan jawaban yang penuh, dalam artian, jawaban tidak unknown karena infinite generation/halusinasi dari model tersebut



HASIL EKSPERIMENT



Universitas
Indonesia

Ada beberapa penemuan menarik dari hasil eksperimen:

1. Anomali Qwen (14%): Meskipun unggul di Naive Prompting, Qwen gagal total dalam framework ini. Analisis log menunjukkan kegagalan ini bukan pada logika, melainkan pada kepatuhan format sintaks (parsability). Qwen sering menghasilkan output yang terlalu mengikuti format pada tahap search & resolve tanpa mengandalkan kemampuan penalaran logikanya, sehingga jawaban pada tahap tersebut menjadi tidak valid.

2. Dominasi SEA-LION (81.6%): SEA-LION v3 menunjukkan performa terbaik. Ini membuktikan hipotesis bahwa model regional yang dilatih dengan instruksi spesifik bahasa lokal (dan mungkin data multilingual alignment yang lebih baik) mampu menjadi "Penerjemah Logika" yang lebih patuh pada tahapan-tahapan framework Aristotle pada dataset berbahasa Indonesia.

3. Stabilitas Sahabat-AI: Sahabat-AI menunjukkan performa yang seimbang. Meskipun tidak setinggi SEA-LION, akurasinya (61.2%) sebanding dengan performa Naivenya, menunjukkan konsistensi pemahaman.



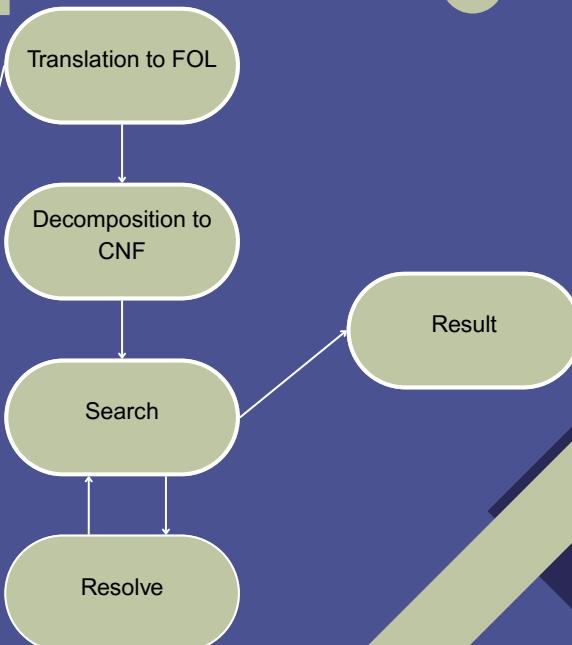
Framework Aristotle



Kesimpulan

Mekanisme penalaran logis dengan open-weight SLM pada Bahasa Indonesia

Translasi Dataset ke
Bahasa Indonesia → Framework Aristotle



Mekanisme Penalaran Logis pada Bahasa Indonesia: Penalaran logis pada konteks Bahasa Indonesia dalam penelitian ini dilakukan melalui pembuatan sumber daya data dan eksekusi arsitektur Neuro-Symbolic. Proses ini diawali dengan adaptasi dataset, di mana dataset sintetis ProntoQA diterjemahkan ke dalam Bahasa Indonesia dengan mempertahankan struktur ontologi logika formal. Selanjutnya, mekanisme penalaran logis dijalankan melalui empat tahapan dalam framework Aristotle:

- (a) Logical Translation, di mana SLM melakukan translation narasi alamiah Indonesia ke dalam sintaks First-Order Logic (FOL)
- (b) Decomposition, yang mengonversi aturan dalam format FOL menjadi Conjunctive Normal Form (CNF)
- (c) Search Router, yang memilah klausa berkomplemen untuk disiapkan pada tahap resolusi
- (d) Logical Resolver, yang melakukan deduksi deterministik menggunakan prinsip resolusi untuk menghasilkan jawaban yang valid secara simbolik dan mengembalikan ke Search Router untuk mencari klausa baru

Kesimpulan

Perbandingan Performa Antar Model Open-Weight:

Best: SEALION v3
(Regional)

Worst: Qwen2.5
(Global)

Middle: SahabatAIv1
(Nasional)



Perbandingan Performa Antar Model Open-Weight:

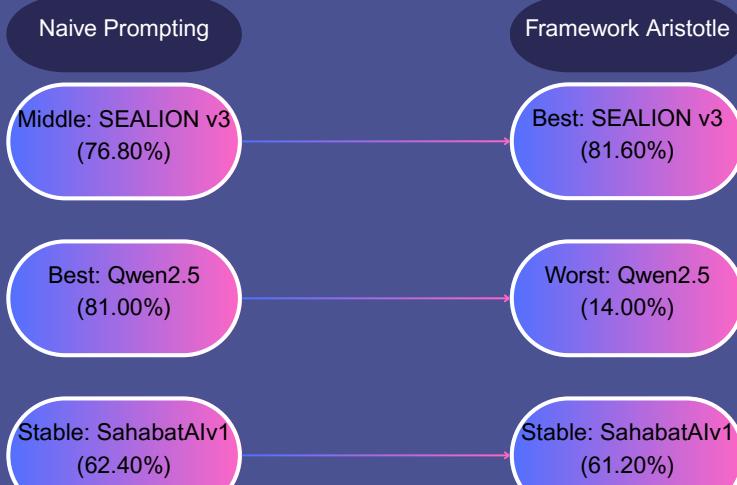
SEA-LION v3 (Regional) mencatat performa terbaik dengan akurasi 81.60%. Model ini menunjukkan keseimbangan terbaik antara pemahaman instruksi dalam Bahasa Indonesia dan kepatuhan terhadap format logika formal.

Sahabat-AI v1 (Nasional) menunjukkan performa yang stabil (61.20%), mengungguli model global dalam memahami nuansa semantik lokal, meskipun masih dibawah SEA-LION dalam konsistensi sintaks kompleks.

Qwen2.5 (Global), meskipun dikenal unggul dalam code generation, mengalami kegagalan katastropik (14.00%) pada metode ini. Kegagalan ini disebabkan oleh ketidakmampuan model untuk menjawab sesuai penalaran logis ketika dipaksa mengikuti format ketat, menandakan bahwa kapabilitas instruction following tidak selalu berbanding lurus dengan kemampuan penalaran logis dalam konteks bahasa tertentu.

Kesimpulan

Efektivitas Aristotle vs Naive Prompting (Explanation Before Answer):



Efektivitas Aristotle vs Naive Prompting:

Penerapan framework Aristotle terbukti meningkatkan reliabilitas jika dan hanya jika model memiliki kapabilitas instruction following yang kuat dalam bahasa target.

Pada model SEA-LION, penggunaan framework Aristotle meningkatkan akurasi dari 76.80% (Naive) menjadi 81.60%. Ini membuktikan bahwa framework Aristotle dapat memanfaatkan kemampuan model untuk menghasilkan logika yang lebih tepat ketika diarahkan dengan benar.

Sebaliknya, pada model Qwen2.5, performa justru turun drastis dari 81.00% (Naive) menjadi 14.00%. Hal ini menunjukkan bahwa model yang tidak dapat mematuhi atau terlalu patuh pada format prompt template tidak akan menghasilkan penalaran logis yang baik, berbeda dengan Naive Prompting yang lebih toleran terhadap kesalahan parsial.

Framework Aristotle memiliki sifat brittle, kesalahan kecil pada satu tahap akan merusak keseluruhan hasil atau error propagation. Hal ini bisa dilihat pada penurunan performa Sahabat-AI dari 62.40% (Naive) menjadi 61.20% ketika menggunakan framework Aristotle karena kesalahan pada tahap awal (seperti translation) menyebabkan kegagalan pada tahap selanjutnya.

Sekian & Terima Kasih

Dipresentasikan oleh :
MIKHAEL DEO BARLI
NIM 1906350572
Jurusan Ilmu Komputer

INSPIRASI



BERANDA PROFIL PUBLIKASI MAPANB JOB PPID BerAKHLAK

Pemerintah Apresiasi Pengembangan Sahabat-AI sebagai LLM dengan Konteks Keindonesiaan

03 Jun 2021 | Dilihat: 4976



[pengembangan-sahabat-ai-](#)

gto

Beranda Terbang Kartu Tela Kartu Hubungan Internasional Komunikasi Kard Produk Buang Berkita Kartu Hubung Kartu

Ruang Berita

Detail Peta Pengaruh Media

Sahabat-AI Semakin Canggih: Indosat dan GoTo Luncurkan Model 70 Miliar Parameter dengan Layanan Chat Multibahasa



INDONESIA.GO.ID

BERANDA PROFIL NATIONALE RAGAM LATARAN BERITA GALERI LAPOR

Universitas
Indonesia

Beranda / Sosial & Budaya / Wamenkomdig: LLM Lokal Sahabat AI Sesuai Peta Jalan yang Sedang Dibuat Pemerintah

Administrator | Rabu, 4 Juni 2021 | 02:34 wib



<https://www.menpan.go.id/site/berita-sebagai-llm-dengan-konteks-ke>

<https://www.menpan.go.id/site/berita-sebagai-llm-dengan-konteks-ke>

ABLASI

Universitas
Indonesia

