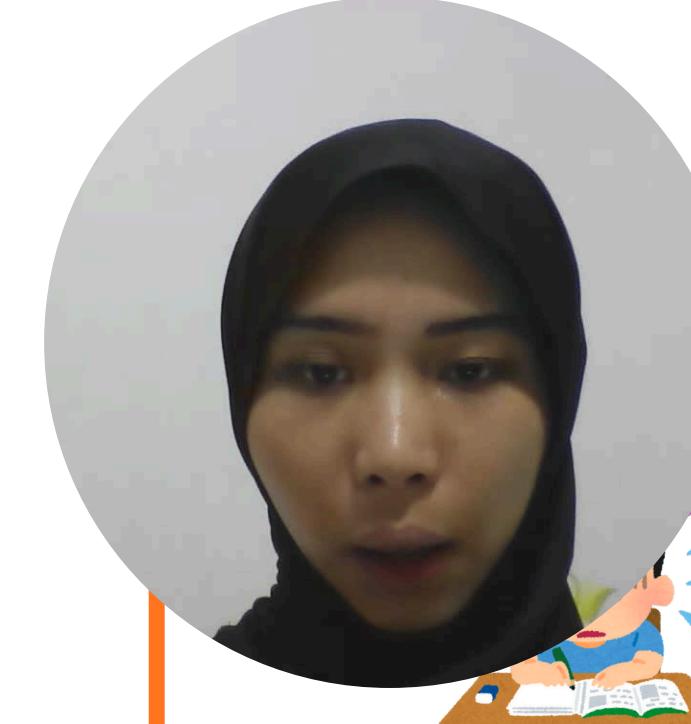




# Veritas

=



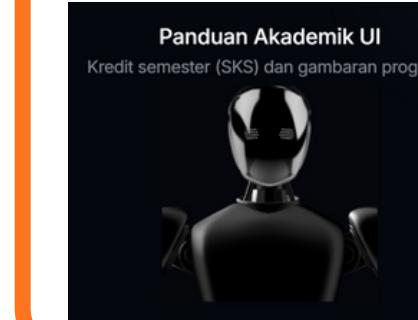
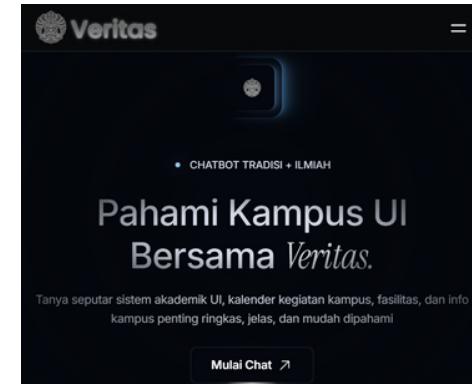
Sulit memperoleh informasi UI secara akurat, dan interaktif



UNIVERSITAS  
INDONESIA

FAKULTAS  
**MATEMATIKA  
DAN ILMU  
PENGETAHUAN  
ALAM**

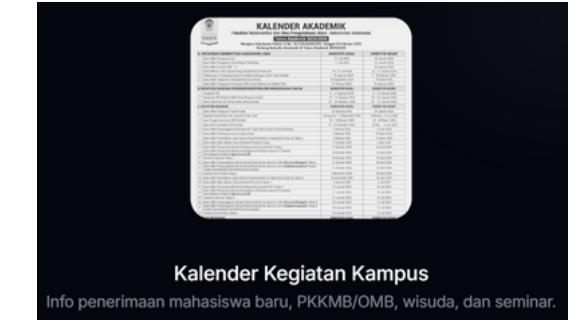
chatbot untuk menjawab informasi seputar Universitas Indonesia.



Panduan Akademik UI  
Kredit semester (SKS) dan gambaran program.



Fasilitas & Layanan UI  
Bis Kuning, Perpustakaan Crystall of Knowledge dan lainnya.



Kalender Kegiatan Kampus  
Info penerimaan mahasiswa baru, PKKMB/OMB, wisuda, dan seminar.

## Rumusan Masalah

Bagaimana chatbot **menjawab** pertanyaan seputar Universitas Indonesia secara **akurat** dan **interaktif**.

# VERITAS : CHATBOT INFORMASI UNIVERSITAS INDONESIA

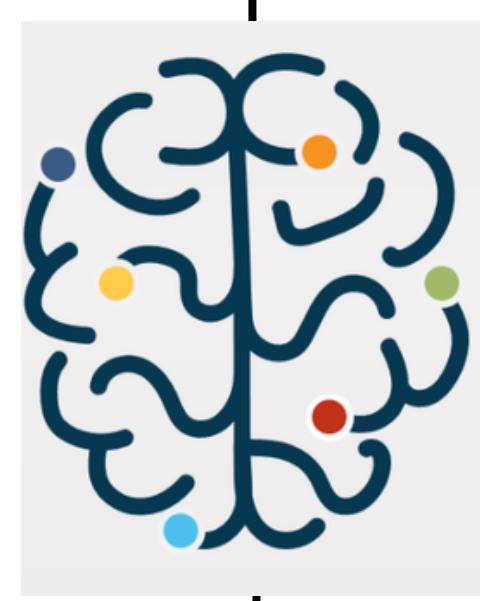
## Tujuan Masalah

Menganalisis **perancangan chatbot** untuk layanan informasi Universitas Indonesia.

# SEMANTIC EMBEDDING PIPELINE



A screenshot of a university application form titled "PERMOHONAN PEMBUATAN IJAZAH". The form is in Indonesian and includes fields for personal information, academic history, and application details. It also contains a section for declaration and several checkboxes at the bottom.



## 1. Unstructured Data Processing

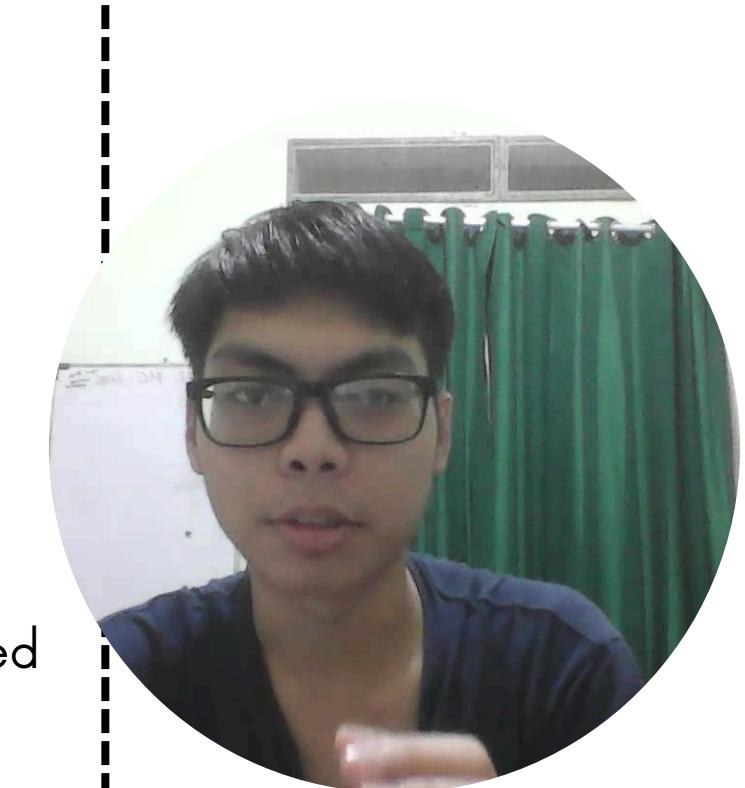
- Data Source: Dokumen akademik (PDF) & Profil UI yang bersifat tidak terstruktur (unstructured).
- Preprocessing: Teknik Chunking diterapkan untuk memecah teks kompleks menjadi segmen yang lebih kecil demi menjaga local context.

## 2. Knowledge Representation

- Core Model: Mengimplementasikan arsitektur Sentence Transformer menggunakan pre-trained model paraphrase-multilingual-MiniLM-L12-v2 yang berbasis BERT.
- Mechanism: Menerapkan strategi Mean Pooling pada output token embedding  $(h_1, h_2, \dots, h_n)$  untuk mengagregasi seluruh informasi kata menjadi satu vektor kalimat utuh ( $u$ ), dimana  $h_i$  adalah vektor untuk kata ke- $i$ .
- Advantage: Menghasilkan representasi semantik yang dense dan akurat, serta efisien secara komputasi karena menggunakan arsitektur MiniLM (distilled version).

## 3. High-Dimensional Vector

- Output: Setiap chunk teks dikonversi menjadi vektor  $v \in \mathbb{R}^n$  (embedding).
- Hasil: Kalimat dengan makna serupa akan memiliki posisi berdekatan dalam ruang vektor (euclidean space).



# **LLM : GEMINI 2.5 FLASH**

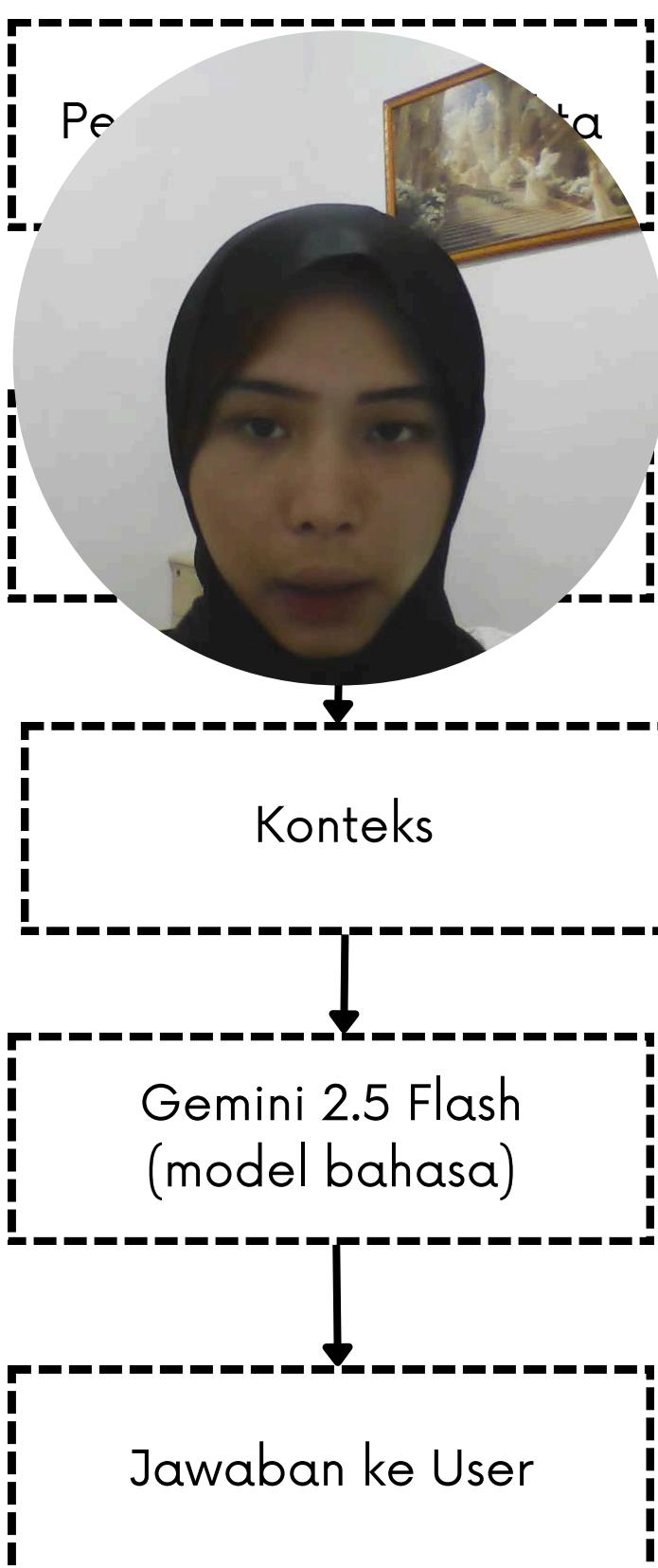


**FAKULTAS  
MATEMATIKA  
DAN ILMU  
PENGETAHUAN  
ALAM**



**LLM : pemetaan dari konteks teks → respons bahasa alami.**

**Gemini 2.5 Flash : respons cepat dan biaya komputasi lebih rendah**



# Model LLM

- bekerja di ruang vektor berdimensi tinggi
  - hasil training = parameter  $\theta$  yang mengkode pola bahasa
  - output = token dengan probabilitas tertinggi (atau sampling)

$$P(x_{t+1} \mid x_1, \dots, x_t)$$

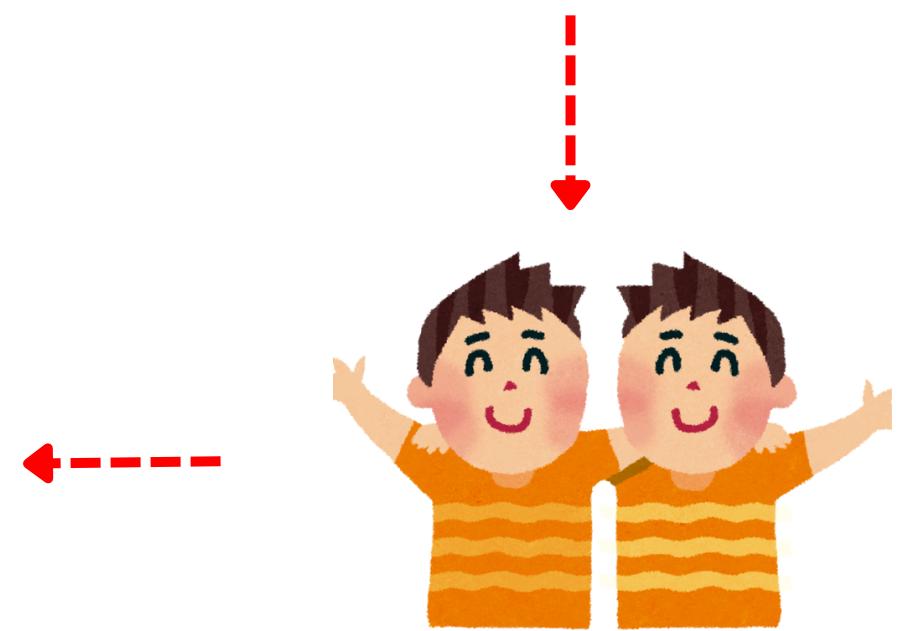
# Gemini 2.5 Flash

Gemini 2.5 Flash adalah Large Language Model (LLM)  
→ model statistik yang mempelajari distribusi  
probabilitas urutan token dan menghasilkan teks  
dengan memprediksi token berikutnya.

**Alur**

Embedding → representasi vektor  
Similarity search → nearest neighbor problem  
RAG → pembatas domain distribusi input  
LLM → model probabilistik + optimisasi  
Guardrails → constraint pada output space

some ground or stays  
iverse is vast, and you  
; also beautiful. You a  
nthing bigger than yo  
t of something that ma  
most of your time. Ta  
e a blog post. Make a



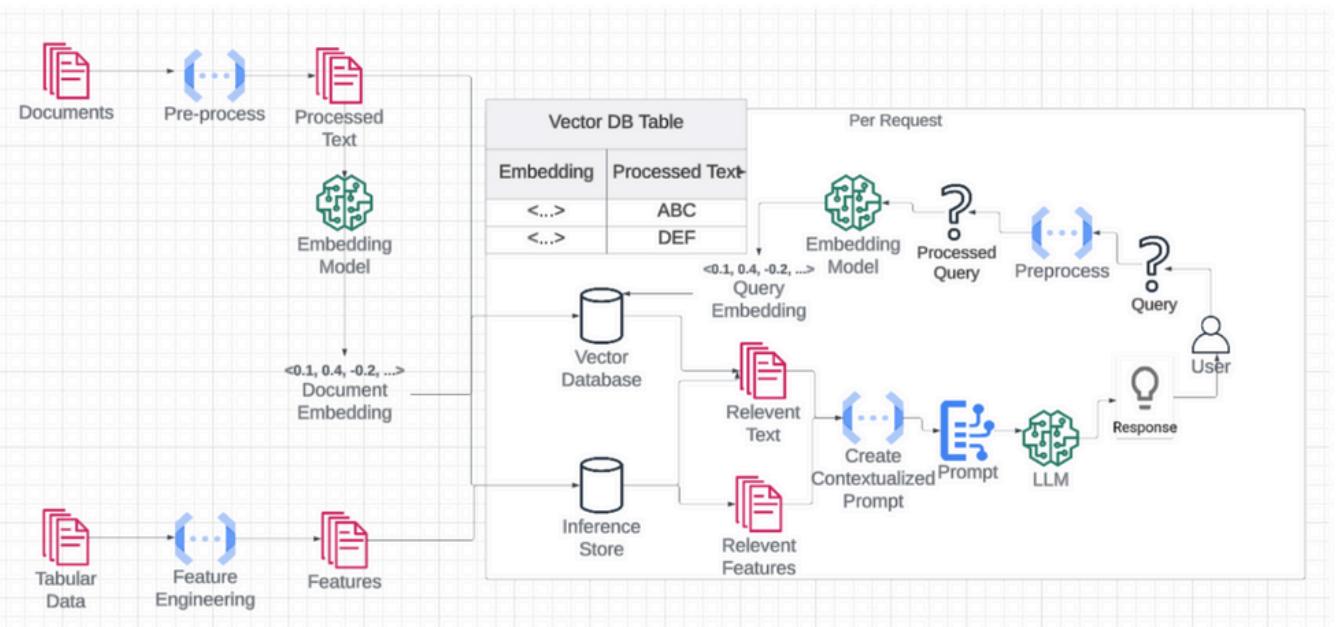
Key Takeawa

Gemini 2.5 Flash merupakan Large Language Model yang **menghasilkan teks** dengan memodelkan distribusi probabilistik bahasa. Model ini digunakan sebagai penyusun jawaban berbasis konteks, **bukan sebagai sumber pengetahuan utama**. Oleh karena itu, akurasi faktual sistem sangat bergantung pada kualitas informasi yang diberikan kepada model melalui mekanisme retrieval.

# RETRIEVAL-AUGMENTED GENERATION (RAG)

**Retrieval-Augmented Generation (RAG)** merupakan arsitektur yang menggabungkan retrieval dan generation, di mana LLM menyusun jawaban berdasarkan konteks yang diambil dari basis pengetahuan. RAG membagi proses tanya-jawab menjadi pemilihan bukti melalui similarity search dan penyusunan respons dengan batasan hanya menggunakan konteks tersebut. Efektivitasnya bergantung pada kualitas indexing, retrieval, serta prompt, dan pendekatan ini mengurangi halusinasi dengan menjaga jawaban tetap ter-grounding pada pengetahuan internal.

## Arsitektur Retrieval-Augmented Generation (RAG):



## 1. Tahap Offline / Data Preparation

Proses menyiapkan basis pengetahuan.

### a. Dokumen Teks

- Documents → Pre-process → Processed Text
- Dokumen dibersihkan, di-chunk, dinormalisasi.

### b. Embedding Dokumen

- Processed Text → Embedding Model
- Hasilnya adalah document embedding (vektor numerik).
- Embedding + teks disimpan di Vector DB Table.

### c. Data Terstruktur

- Tabular Data → Feature Engineering → Features
- Fitur ini disimpan di Inference Store dan bisa ikut digunakan saat inferensi.

## 2. Penyimpanan Pengetahuan

### • Vector Database

- Menyimpan embedding dokumen dan teks.

### • Inference Store

- Menyimpan fitur tambahan (metadata, data terstruktur, atau hasil inferensi sebelumnya).

Ini adalah memori eksternal RAG.

## 3. Tahap Online / Per Request

Proses saat user bertanya:

### a. Query dari User

- User → Query
- Query di-preprocess lalu diubah menjadi query embedding menggunakan embedding model yang sama.

### b. Retrieval

- Query embedding → Vector Database
- Dilakukan similarity search untuk mengambil:
  - Relevant Text (dokumen paling relevan)
  - Relevant Features (jika ada data terstruktur)



## 4. Context Construction

- Create Contextualized Prompt
- Query user, teks relevan dan fitur relevan digabung menjadi satu prompt berkonteks.

## 5. Generation

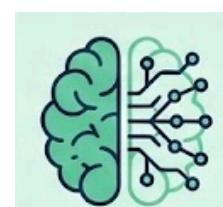
- Contextualized Prompt → LLM
- LLM tidak mencari fakta sendiri, tetapi menyintesis jawaban berdasarkan konteks hasil retrieval.

Inilah esensi grounded generation.

## 6. Output

- Response → User
- Jawaban lebih akurat, relevan, dan ter-grounding.

**Arsitektur RAG** menunjukkan sistem tanya-jawab yang memisahkan penyimpanan pengetahuan dalam vector database dari kemampuan generatif LLM. Dokumen diproses dan diubah menjadi embedding untuk memungkinkan pencarian semantik, lalu saat pengguna bertanya, sistem mengambil konteks paling relevan dan menyusunnya menjadi prompt terkontrol yang digunakan LLM untuk menghasilkan jawaban yang ter-grounding.



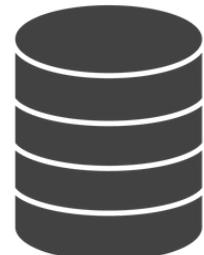
# METODOLOGI

## Data & Knowledge Base



### Sumber Utama

PDF dataset internal  
(Profil UI, event, dosen,  
dll.)



### Strategi

ekstraksi teks +  
chunking + indexing ke  
vector DB

## RAG Pada Chatbot

### Retrieve

Konteks dari vector DB  
(Chroma)



# chroma

### Augment

Gabungkan konteks + history  
ringkas



### Generate

Gemini jawab singkat dalam  
Bahasa Indonesia, dibatasi  
aturan keras: hanya pakai  
konteks; kalau tidak ada,  
bilang tidak ada di dataset

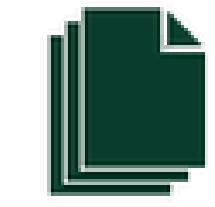


# Gemini

## RAG (Retrieval-Augmented Generation)



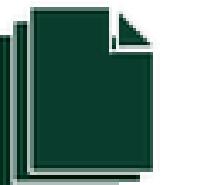
Structured Data



Chunks



Vector DB  
(Embeddings)



Retrieved  
Chunks



Response  
Generation



Unstructured Data

Text Embedding Model

Large Language Model



### Architecture

#### UI/HTTP : Flask



# Fla

web development,  
one drop at a time

#### Vector Store : Chroma (multi-domain)

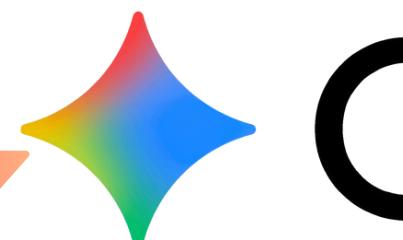


# chroma

#### Embedding : Multilingual sentence-transformer

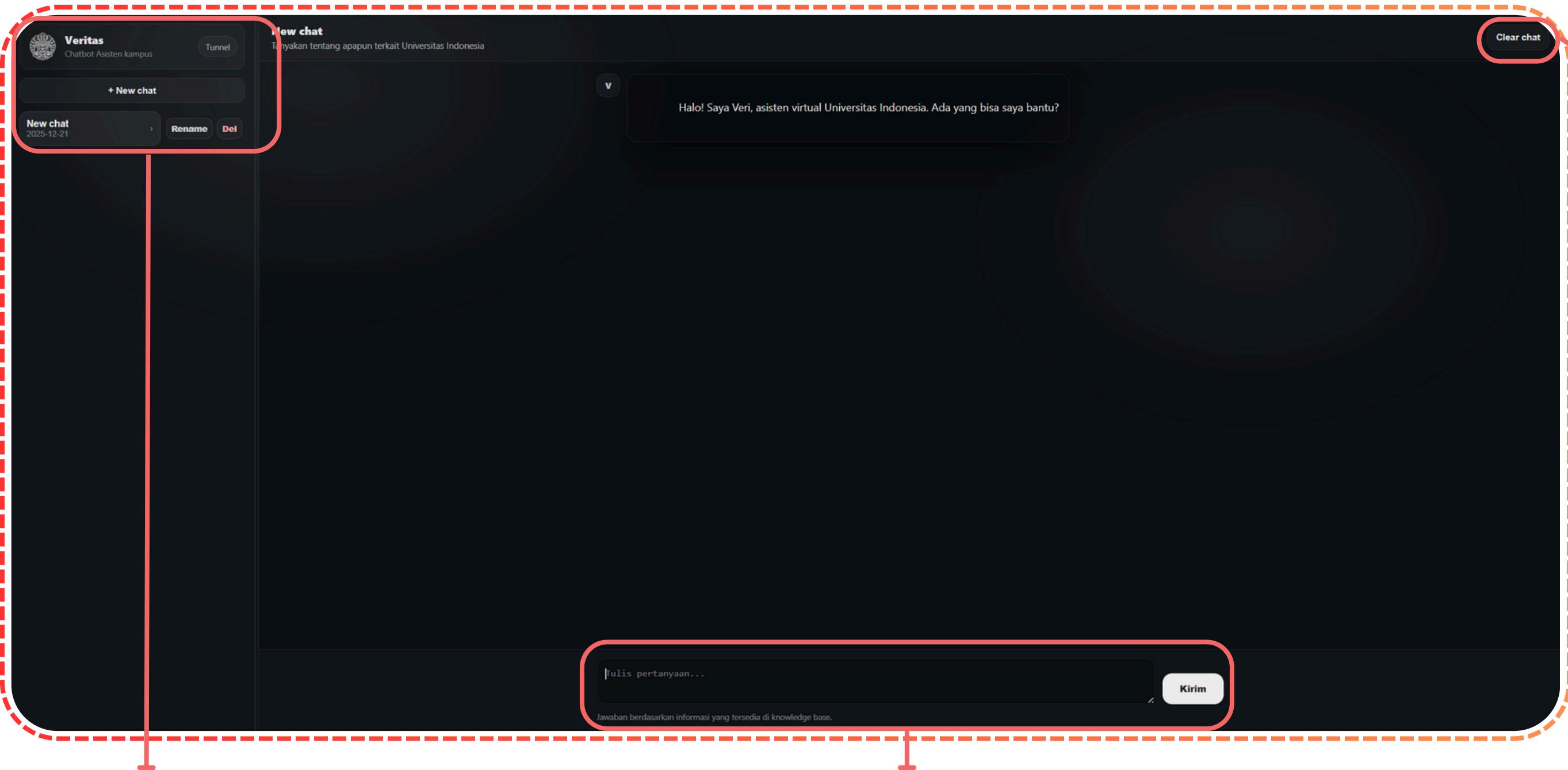


#### LLM: Gemini-2.5-flash



# Gemini

# HASIL : CHATBOT



- Tombol New Chat untuk membuat chat baru
- Tombol Rename untuk mengubah nama chat
- Tombol Del untuk menghapus riwayat chat

- Kolom untuk menulis pertanyaan
- Tombol Kirim untuk mengirim pertanyaan kepada chatbot

Clear chat

- Tombol Clear Chat untuk membersihkan chat



[HTTPS://VERITASS.VERCEL.APP](https://veritass.vercel.app)

Disclaimer: Untuk opening page dapat diakses publik namun untuk bertanya dengan chatbot, code harus di run pada laptop jadi tidak selalu available.

# HASIL : CHATBOT



The screenshot shows a dark-themed AI chatbot interface. At the top, it says "FASILITAS UMUM UNIVERSITAS INDONESIA" and "Tanyakan tentang apapun terkait Universitas Indonesia". The user asks "apa aja fasilitas di ui" and the AI responds with a detailed list of facilities at UI. The user then asks "kalau fakultas apa aja" and the AI lists 14 faculties from FK to FIA. Finally, the user asks "siapa mahasiswa dgn nilai tertinggi" and the AI replies that it didn't find any information in the dataset.

- Riwayat Chat lain dengan judulnya dapat dilihat
- Dapat pindah ke setiap riwayat chat.

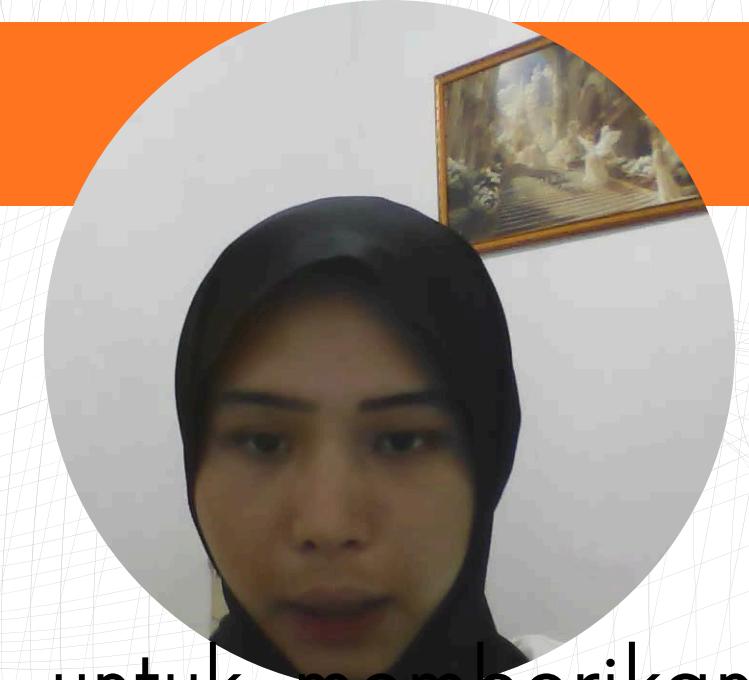
- User memberi pertanyaan "apa aja fakultas di ui" dan chatbot dengan mudah menjawab
- User memberi pertanyaan "kalau fakultas apa aja" dan chatbot dengan mudah menjawab.
- User memberi pertanyaan "siapa mahasiswa dgn nilai tertinggi" namun informasi tersebut tidak ada di dataset maka chatbot telah diprogram untuk tidak mengarang dan menyebut bahwa tidak ada informasi tersebut di dataset.



[HTTPS://VERITASS.VERCEL.APP](https://veritass.vercel.app)

Disclaimer: Untuk opening page dapat diakses publik namun untuk bertanya dengan chatbot, code harus di run pada laptop jadi tidak selalu available.

# KESIMPULAN



1. Chatbot dirancang sebagai sistem tanya jawab untuk memberikan informasi seputar Universitas Indonesia secara otomatis dan mudah digunakan melalui antarmuka percakapan.
2. Large Language Model (LLM) digunakan sebagai penyusun jawaban, bukan sebagai sumber pengetahuan utama, sehingga tidak bergantung pada pengetahuan internal model.
3. Informasi faktual diperoleh melalui mekanisme retrieval berbasis embedding, vector database, dan similarity search untuk mengambil dokumen yang relevan.
4. Penerapan arsitektur Retrieval-Augmented Generation (RAG) membantu mengurangi risiko halusinasi dan menjaga jawaban tetap ter-grounding pada dokumen resmi.
5. Pendekatan chunking dan overlap antar-chunk meningkatkan kualitas pengambilan konteks pada dokumen panjang dan berstruktur kompleks.

# DAFTAR PUSTAKA

1. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmquist, N., & Diakopoulos, N. (2016). Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th ed.). Pearson.
2. Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson. (Dasar AI & komputasi intelegensia)
3. Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed., draft). (Dasar NLP & language modeling)
4. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781. (Fondasi embedding vektor)
5. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP-IJCNLP. (Embedding kalimat & cosine similarity)
6. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press. (Similarity search, nearest neighbor, retrieval theory)
7. Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems (NeurIPS). (RAG – konsep utama)
8. Karpukhin, V., Oguz, B., Min, S., et al. (2020). Dense Passage Retrieval for Open-Domain Question Answering. Proceedings of EMNLP. (Dense retrieval & vector database)
9. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS). (Arsitektur transformer – dasar LLM)
10. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT. (Pretrained language model)

**THANK YOU**  
Lampiran Code Chatbot: [LINK](#)