



# Prompt flow

Build high-quality LLM apps - from prototyping, testing to production deployment and monitoring



## BMRI EDA Workshop

Promptflow session



MITRA INTEGRASI INFORMATIKA

# Agenda

1. Introduction to Promptflow
2. Deep Dive into Promptflow Components
3. Promptflow Demo on Azure
4. Deploying Promptflow on Azure
5. Q&A

# Introduction to Promptflow

- **Azure OpenAI Playground** adalah langkah awal yang cepat untuk melakukan eksplorasi dan eksperimen, **not suitable for production**
- **Azure OpenAI Playground** lebih cocok untuk uji coba prompt sederhana, bukan untuk membangun solusi chatbot yang scalable dan terintegrasi.
- Kekurangan:
  - Terbatas pada template
  - Kurang fleksibel
  - Sulit dikelola

# What is promptflow

- **PromptFlow** adalah tools yang dirancang untuk mempermudah siklus pengembangan end-to-end **aplikasi AI berbasis LLM**
- Proses prompt engineering memerlukan repetisi yang banyak
- Scalable, Prototyping, Evaluation and Monitoring
- Promptflow concept
  - [Flows](#)
  - [Tools](#)
  - [Connections](#)
  - [Design principles](#):

Multi-Round Q&A on Your ... Chat Completed View outputs Runtime Select runtime Deploy Clone Save Evaluate Chat

Flow

LLM Prompt Python More tools Raw file mode Wrap text

Inputs

Name	Type	Value	Chat input
chat_history	list	[ { "inputs": { "chat_input": "considering information from	Chat history
chat_input	string	which page is the information "The minimum amount for top-u	

Add input

Outputs

Name	Value	Chat output
chat_output	\${chat_with_context.output}	

Add output

modify\_query\_with\_history LLM

Connection Default\_AzureOpenAI Api chat deployment\_name squirrelAIGPT4 temperature 0 stop max\_tokens 1000

Advanced

Function calling

Prompt Referring to: modify\_query\_with\_history.jinja2

Inputs Validate and parse input

Activate config

Outputs Duration 7.17s Tokens 201 Completed View full output

embed\_the\_question Embedding

Inputs

Activate config

Outputs Duration 0.31s Tokens 42 Completed View full output

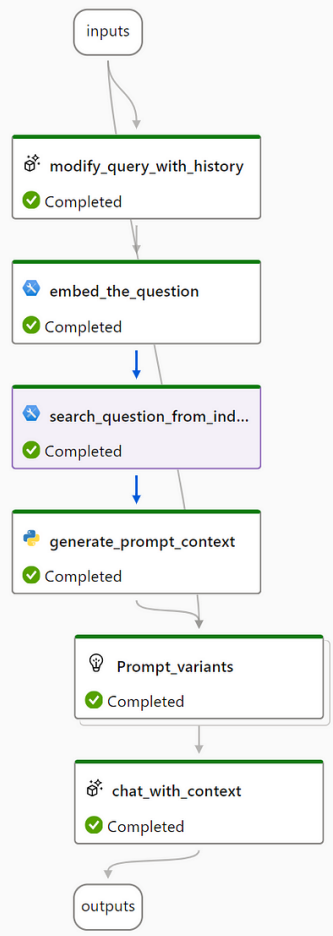
search\_question\_from\_indexed\_docs Vector Index Lookup

Inputs

Activate config

Files

Graph



```
graph TD; inputs([inputs]) --> modify_query_with_history[modify_query_with_history]; modify_query_with_history --> embed_the_question[embed_the_question]; embed_the_question --> search_question_from_indexed_docs[search_question_from_indexed_docs]; search_question_from_indexed_docs --> generate_prompt_context[generate_prompt_context]; generate_prompt_context --> Prompt_variants[Prompt_variants]; Prompt_variants --> chat_with_context[chat_with_context]; chat_with_context --> outputs([outputs]);
```

100%

5

# Types in Promptflow

## **Standard Flow:**

- Used for developing general LLM applications.
- Defines the sequence of operations without chat-specific features.

## **Chat Flow:**

- Tailored for chat applications.
- Supports `chat_history`, `chat_input`, and `chat_output`.
- Provides a chat-like experience during development and includes a sample chat app for deployment.

## **Evaluation Flow:**

- Designed to test and evaluate the quality of your LLM applications.
- Runs on outputs from standard or chat flows.
- Computes metrics to assess performance (e.g., accuracy, fact-based responses).

# Why Promptflow

- With prompt flow, you will be able to:
- Create **flows** that link [LLMs](#), [prompts](#), [Python](#) code and other [tools](#) together in a executable workflow.
- **Debug and iterate your flows**, especially [tracing interaction with LLMs](#) with ease.
- **Evaluate your flows**, calculate quality and performance metrics with larger datasets.
- **Integrate the testing and evaluation into your CI/CD system** to ensure quality of your flow.
- **Deploy your flows** to the serving platform you choose or integrate into your app's code base easily.
- (Optional but highly recommended) **Collaborate with your team** by leveraging the cloud version of [Prompt flow in Azure AI](#).

# Promptflow Flows

---



Data & AI Team

*We help organization make better & faster decision*



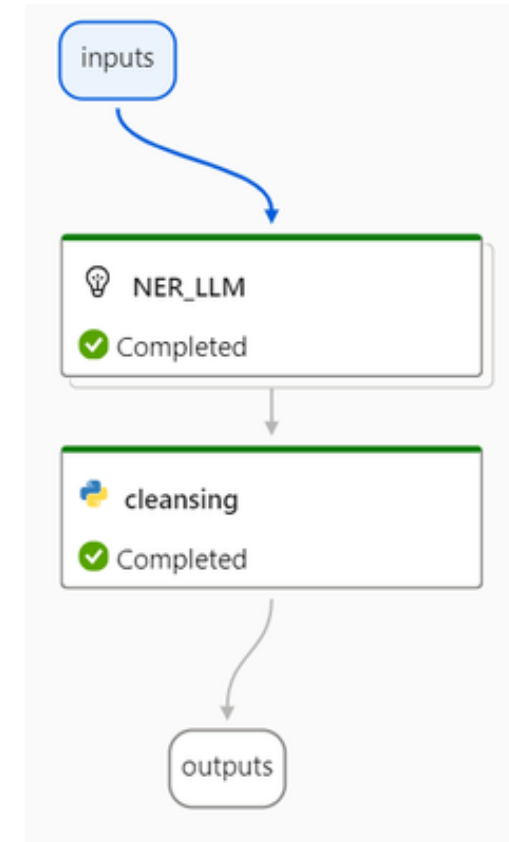
# Promptflow Concept - Flows

## What is a Flow?

- A flow represents the logic and sequence of operations in your LLM application.
- It defines how different functions and services are connected and executed.

## Types of Flows:

- DAG (Directed Acyclic Graph) Flow
- Flex Flow



# Promptflow Concept – Flows – DAG Flows

## Definition and Features:

- Represents your LLM app as a Directed Acyclic Graph of functions, known as tools.
- Functions are connected via input/output dependencies and executed based on the graph topology.

## How to Create a DAG Flow:

- Define your flow in a YAML file (`flow.dag.yaml`).
- Use the PromptFlow VS Code Extension to visualize and edit the DAG.

## ● Benefits of DAG Flow:

- **Low Code Development:** Create apps by dragging and dropping components in a UI.
- **DAG Visualization:** Easily understand the app's logic structure through visual graphs.

# Promptflow Concept – Flows – DAG Flows

- **Basic Example:**
  - A standard flow using a custom Python tool to call Azure OpenAI.
  - Connection details are managed via environment variables.
- **Chat-Basic Example:**
  - Illustrates creating a basic chat flow in a DAG structure.
  - The chatbot can recall previous messages and uses the conversation history for generating replies.
- **Eval-Basic Example:**
  - Provides a template for evaluating basic aspects of your LLM app.
  - Can be extended to compute various performance metrics.

# Promptflow Concept – Flows – Flex Flows

- **Definition and Features:**

- A code-centric approach to building LLM apps.
- Uses a Python function or class as the entry point, encapsulating your app logic.

- **How to Create a Flex Flow:**

- Write your app logic in Python.
- Optionally define a `flow.flex.yaml` file pointing to your Python entries.
- Test and run using pure code or the PromptFlow VS Code Extension.

- **Benefits of Flex Flow:**

- **Customizable:** Customize with Python code.
- **Advanced Orchestration:** Utilize Python control structures and third-party libraries.
- **Easy Integration:** Smooth onboarding from platforms like LangChain or Semantic Kernel.



# Flex Flows - Examples

## Basic Example:

- A simple standard flow using a function entry that calls Azure OpenAI.
- Connection information is stored in environment variables.

## Chat-Basic Example:

- Demonstrates creating a chatbot that remembers previous interactions.
- Uses a class entry to manage conversation history and generate responses.

## Eval-Code-Quality Example:

- Shows how to evaluate the quality of code snippets.
- Uses a function entry to assess code and provide feedback.

# When to use which Flows?

## Flex Flow Use Cases:

- When you prefer coding and need advanced control over the app logic.

## DAG Flow Use Cases:

- When you prefer a visual, low-code approach.
- Suitable for straightforward apps where visualizing the logic is beneficial.

## Comparison:

- **Flex Flow:** More flexibility and control through code.
- **DAG Flow:** Easier to build and understand through visual representation.

# Promptflow Tools

---



**Data & AI Team**

*We help organization make better & faster decision*

# Promptflow Concept - Tools

- **Definition:**
- Tools are the **building blocks** of any flow in PromptFlow. Each tool is essentially a **function** designed to perform specific tasks that drive the logic of your LLM app.
- **Examples of Tasks Performed by Tools:**
- Accessing and interacting with **LLMs** (Large Language Models)
- Querying **databases**
- Extracting information from **search engines**
- Performing **pre-processing** or **post-processing** of data



# Tools - The Three Basic Tools in PromptFlow

- **LLM Tool:**

- This tool allows you to write custom **prompts** and use LLMs for various tasks.
- **Use Cases:** Summarizing articles, generating responses for customer support, and more.

- **Python Tool:**

- Enables you to write custom Python functions for specific tasks.
- **Use Cases:** Fetching web pages, processing data, calling third-party APIs.

- **Prompt Tool:**

- Prepares complex prompts as strings, which can be used with other tools.
- **Use Cases:** Advanced prompt generation, combining multiple prompt tools or Python tools.

# Tools – More Advance Tools

- **Tools From Partners:**

- **Vector DB Lookup**
- **Faiss Index Lookup**
- These tools are contributed by partners, expanding the capabilities of PromptFlow for complex tasks.

- **Custom Tools in PromptFlow**

- **Create Your Own Tools:**

- You can build **custom tools** for specific use cases.
- These tools can be **shared** with your team or with the global community.

- **Why Create Custom Tools?:**

- Tailored to specific application needs
- Can incorporate unique business logic or proprietary algorithms
- **Learn More:**

# Promptflow Connections

---



Data & AI Team

*We help organization make better & faster decision*

# Promptflow Concept - Connections

- Connections in PromptFlow allows to securely manage **credentials** or **secrets** for accessing external services like:
  - Large Language Models (LLMs)
  - Databases
  - APIs
- Connections store key information such as **Endpoints** and **API keys**
- **Managing Connections in Different Environments**
  - **Local Development:**

In local development, connections are stored on your local machine with **keys encrypted** for security.
  - **Azure AI:**
    - In Azure, connections can be shared across an entire workspace.
    - **Secrets** related to connections are securely stored in **Azure Key Vault**, adhering to strict security and compliance standards.

# Connections - Built-in Connections in PromptFlow

- PromptFlow provides a variety of **pre-built connections** for seamless integration with external services.
- These connections enable you to connect and interact with services directly from built-in tools.

Connection	Used In
Azure OpenAI	LLM or Python
OpenAI	LLM or Python
Cognitive Search	Vector DB Lookup or Python
Serp API	Serp API or Python
Serverless	LLM or Python



# Connections - Custom Connections in PromptFlow

- Custom Connection Types:
  - You can also create your own custom connections using key-value pairs.
  - These custom connections provide flexibility to tailor integration with specific services, especially useful for Python tools.
- Benefits:
  - Customization: Easily adapt to your application's specific requirements.
  - Flexibility: Use custom connections with external APIs and services, beyond the built-in options.

# Promptflow Variants

---



Data & AI Team

*We help organization make better & faster decision*

# Promptflow Concept - Variants

- Definition:
  - A variant refers to a specific version of a tool node with distinct settings. Variants allow you to fine-tune your prompt and are currently available for the LLM tool in PromptFlow.
- Purpose:
  - Variants let you experiment with different prompts or settings to improve the quality of the responses generated by the model.



# Variant - Examples

- Scenario:

You want to generate a summary of a news article. By creating variants, you can adjust the prompt and connection settings as shown:

Variant	Prompt	Temperature
Variant 0	Summary: {{input sentences}}	1.0
Variant 1	Summary: {{input sentences}}	0.7
Variant 2	What is the main point of this article? {{input sentences}}	1.0

You can adjust both prompts and settings to explore how the model responds to different instructions and configurations.



# Variants - Summary

- **Variants** allow you to fine-tune LLM prompts by adjusting **settings** and **instructions**.
- **Improve quality, save time, and boost productivity** in prompt engineering.
- Variants simplify the process of testing multiple configurations and make **comparisons easy**, driving better outcomes for your LLM applications.



# Prompt Flow

## Prototyping code-first solution

## DEMO ON AZURE



# Thank You

 **BMRI EDA Workshop**

**Promptflow session**



MITRA INTEGRASI INFORMATIKA