PRAKTIKUM

MMAI1004 – PEMBELAJARAN MESIN

PERTEMUAN V

GENERATIVE AI WITH LARGE LANGUAGE MODELS

DOSEN PENGASUH:

Prof. Dr. Taufik Fuadi Abidin, S.Si., M.Tech

Dr. Rumaisa Kruba, S.Si

Prof. Dr. Ir. Hammam Riza, M.Sc. IPU

Praktisi Komdigi (Dr. Said Mirza Pahlevi)

ASISTEN PRAKTIKUM:

Muhammad Chaidir, S.Kom, M.Kom

Hands on Generative AI with Large Language Models Using Mistral and Hugging Face API

1. Introduction

Large Language Models (LLMs) are deep learning models trained on vast text corpora to generate human-like text, answer questions, and assist in various natural language processing (NLP) tasks. Today's tutorial will focus on:

- ✓ Understanding LLMs and their applications.
- ✓ Practicing Mistral API for text generation.
- ✓ Practicing Hugging Face API for different NLP tasks.
- ✓ Comparing responses from different APIs.

By the end of this session, students will be able to make API calls and integrate LLMs into their own applications.

2. Setting Up the Environment

Step 1: Install Required Libraries

```
pip install requests
```

Step 2: Set Up API Keys

To use Mistral and Hugging Face APIs, you need API keys.

- ✓ Mistral API Key: Sign up at https://mistral.ai
- ✓ Hugging Face API Key: Get an API key from https://huggingface.co/settings/tokens

Store your keys securely and avoid sharing them.

```
import os
os.environ["MISTRAL_API_KEY"] = "your_mistral_api_key"
os.environ["HUGGINGFACE_API_KEY"] = "your_huggingface_api_key"
```

3. Using Mistral API for Text Generation

Step 1: Import Requests and Set Up API Key

```
import requests

API_KEY = "your_mistral_api_key"  # Replace with your actual API key
headers = {"Authorization": f"Bearer {API_KEY}", "Content-Type":
"application/json"}
```

Step 2: Generate Text with Mistral

```
data = {
    "model": "mistral-small",  # Ensure this is a valid model
    "messages": [{"role": "user", "content": "What is the meaning of life?"}]
}

response = requests.post("https://api.mistral.ai/v1/chat/completions", json=data,
headers=headers)

# Print full API response to check for errors
print(response.json())

# Extract response if available
if "choices" in response.json():
    print(response.json()["choices"][0]["message"]["content"])
else:
    print("Error: 'choices' key not found in response")
```

4. Using Hugging Face API for NLP Tasks

Step 1: Import Requests and Set Up API Key

```
import requests

API_URL = "https://router.huggingface.co/hf-inference/models/google/flan-t5-
small"
headers = {"Authorization": "Bearer your_huggingface_api_key"}
```

Step 2: Use Hugging Face for Question Answering

```
def query(payload):
    response = requests.post(API_URL, headers=headers, json=payload)
    return response.json()

output = query({
    "inputs": "What is the capital city of Indonesia?",
})

print(output)

# Extract response if available
if "choices" in response.json():
    print(response.json()["choices"][0]["message"]["content"])
else:
    print("Error: 'choices' key not found in response")
```

5. Comparing Mistral and Hugging Face Responses

To analyze the differences between Mistral and Hugging Face models, students should:

- ✓ Use the same input prompts for both APIs.
- ✓ Compare the fluency, accuracy, and creativity of the generated responses.
- ✓ Document their observations and discuss which API is more suitable for different tasks.

```
def compare_apis(prompt):

    # Mistral API
    API_KEY = "your_api_key"  # Replace with your actual API key
    headers = {"Authorization": f"Bearer {API_KEY}", "Content-Type":
"application/json"}
    mistral_data = {
        "model": "mistral-small",
        "messages": [{"role": "user", "content": prompt}]
    }
    mistral_response =
requests.post("https://api.mistral.ai/v1/chat/completions",
json=mistral_data, headers=headers)
    mistral_output = mistral_response.json().get("choices",
[{}])[0].get("message", {}).get("content", "No response")

    # Hugging Face API
    huggingface_response = query({"inputs": prompt})

    print("Mistral Response:")
    print(mistral_output)
    print("\nHugging Face Response:")
    print(huggingface_response)
```

6. Conclusion

- ✓ Mistral API provides high-quality, human-like responses for various NLP tasks.
- ✓ Hugging Face API offers diverse NLP models for different use cases.
- ✓ By practicing with these APIs, students gain practical experience in integrating LLMs into applications.