

Praktikum Pemrograman

Dosen : Tri Hadiah Muliawati S.ST., M.Kom



Disusun Oleh:

Wahyu Ikbal Maulana (3323600056)

**POLITEKNIK ELEKTRONIKA NEGERI
SURABAYA**

5. Create a simple program which uses min. 2 libraries

The screenshot shows a Jupyter Notebook interface in a web browser. The title bar says "Mood detection using tweets | Kaggle". The notebook cell [9]: contains Python code for loading a BERT model from the transformers library. The code imports AutoModelForSequenceClassification and AutoTokenizer from transformers, loads the 'indobenchmark/indobert-base-p2' model, and sets the device to GPU if available. It then prints information about the loaded model and tokenizer. The output shows the model info and various files being loaded, with progress bars and file sizes. A warning message indicates that some weights were not initialized from the checkpoint and should be trained.

```
from transformers import AutoModelForSequenceClassification, AutoTokenizer
import torch

# Load the Indonesian BERT model and tokenizer
# We use a model fine-tuned for sentiment analysis (if available)
model_name = "indobenchmark/indobert-base-p2"
model = AutoModelForSequenceClassification.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name)

# Check if a GPU is available and set the device accordingly
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = model.to(device)

# Information about the loaded model and tokenizer
model.info = (
    "Model Name": model_name,
    "Device": device.type
)
model.info

config.json: 100% [██████████] 1.53k/1.53k [00:00<00:00, 89.0kB/s]
pytorch_model.bin: 100% [██████████] 498M/498M [00:01<00:00, 378MB/s]
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at indobenchmark/indobert-base-p2 and are newly initialized: ['classifier.weight', 'classifier.bias']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
tokenizer_config.json: 100% [██████████] 2.00/2.00 [00:00<00:00, 141B/s]
vocab.txt: 100% [██████████] 229k/229k [00:00<00:00, 1.37MB/s]
special_tokens_map.json: 100% [██████████] 112/112 [00:00<00:00, 7.31kB/s]
[9]: {'Model Name': 'indobenchmark/indobert-base-p2', 'Device': 'cpu'}
```

Karena saya mager untuk membuat kode dari awal, disini saya memakai kode yang pernah saya pakai pada project sebelumnya. Gambar di atas adalah screenshot dari program pengimportan model transformer, mungkin kurang tepat jika disebut sebagai program

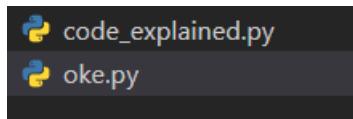
Pada cell kode di atas memakai library transformer dan pytorch yaitu sebuah library python yang mendukung proses NLP, untuk objektif nya saya sedang membuat analisis deteksi karakter seseorang dari tweet twitter, dengan memakai dataset ratusan data twitter lalu dilakukan EDA, preprocessing, dan analisa mendalam lalu kemudian bisa dilakukan prediksi untuk bagaimana emosi seseorang terlihat hanya dari kata/kalimat yang telah diberikan. Pada bagian cell kode ini lebih spesifik ke pengimportan library modul.

Disini saya mengimport library dari huggingface yaitu indobert sebuah llm bahasa indonesia yang masih belum sempurna, namun cukup mampu untuk menganalisa dataset bahasa indonesia. Dengan memanggil function torch dan transformer dan function spesifik seperti AutoTokenizer dan AutoModelSequenceClassification yang merupakan generic model class untuk text classification.

Disini saya membutuhkan kuota internet untuk melakukan pengimportan library, dan outputnya disini menampilkan progres dari proses men “load” pre-trained model yang akan kita gunakan dalam NLP nanti.

6. Create a library module and import it in your program.

Disini saya ingin membuat modul python yang bisa menjawab pertanyaan-pertanyaan siswa/pelajar ketika mengalami kesulitan. Jadi nantinya kode tersebut dapat menjawab pertanyaan baik di pelajaran matematika, ipa, sejarah dan lain-lain. Untuk demo nanti saya akan memakai pelajaran sejarah dengan menanyakan pertanyaan sejarah, lalu untuk penjelasan kode nya ada di bawah ini.



Code program berjudul code_explained.py

```
from dotenv import load_dotenv
import os
import google.generativeai as genai

# Load environment variables
load_dotenv()

# Configure generativeai
genai.configure(api_key="AIzaSyCYyr1lSjWLICK3sHkfnZiV3RLZd_e7qFc")

def generate_gemini_content(tulisan, prompt):
    model = genai.GenerativeModel("gemini-pro")
    response = model.generate_content(prompt + tulisan)
    return response.text

def generate_content(text):
    prompt = """Anda adalah seorang yang tahu segala hal. Berikan jawaban terhadap pertanyaan yang telah diberikan"""
    return generate_gemini_content(text, prompt)
```

Untuk code yang saya buat, disini saya membuat program yang dapat menjawab semua pertanyaan yang kita ajukan, dan menjawab seperti chatgpt. Disini saya memakai model dari gemini pro dengan mengimport library dotenv untuk environtmentnya. Bagaimana kemudian menggunakan model gemini supaya bisa menjawab pertanyaan, disini diperlukan sebuah api yang diambil dari google data studio. Untuk pertanyaan yang spesifik disini saya membuat sebuah prompt tambahan supaya mendapatkan respon yang lebih masuk akal. Dengan dijadikan satu dalam satu fungsi def, yang berisi inputan text dan promptnya, selanjutnya kode yang telah dibuat tinggal di save dan siap untuk dipakai.

Code Program berjudul oke.py

```
from code_explained import generate_content

# Call the generate_content function with a sample text
result = generate_content("Siapa presiden Indonesia yang pertama")
print(result)
```

Setelah membuat modul python tadi, langkah selanjutnya adalah mengimport/memakai modul nya dengan mengimportnya, disini saya memakai spesifik di generate_content, yang kemudian bisa dipakai salah satu functionnya. Dengan menginputkan pertanyaan “Siapa presiden Indonesia yang pertama”, selanjutnya kita print hasil dari functionnya.

The screenshot shows the Visual Studio Code interface. In the top left, there are two tabs: 'code_explained.py' and 'oke.py'. The 'oke.py' tab is active, displaying the following Python code:

```
from code_explained import generate_content
# Call the generate_content function with a sample text
result = generate_content("Siapa presiden Indonesia yang pertama")
#print(result)
```

Below the code editor is a navigation bar with tabs: PROBLEMS, DEBUG CONSOLE, TERMINAL (which is underlined), OUTPUT, ..., and a Python icon followed by '+', a close button, and other icons. The terminal window below shows the command line output:

```
● PS D:\Jupyter lab\modul> & "C:/Users/M S I/AppData/Local/Programs/Python/Python310/python.exe" "d:/Jupyter lab/modul/oke.py"
Soekarno
○ PS D:\Jupyter lab\modul>
```

Disini terlihat bahwa modul yang saya pakai berjalan lancar dan menghasilkan output sesuai yang kita harapkan.

Terima kasih sudah menyimak 😊