INFORMATION RETRIEVAL & INFORMATION EXTRACTION

Chapter 5

TM dan NLP

Sains Data

Information retrieval (IR) atau Sistem Temu Kembali Informasi

PENGERTIAN

- Information Retrieval adalah **sebuah proses** untuk **menemukan kembali informasi** yang dibutuhkan **dari sebuah sistem penyimpanan dan penelusuran informasi**.
- Sistem Temu Kembali Informasi mensyaratkan:
  - Ada kebutuhan informasi dari pengguna
  - Ada dokumen atau *record* yang berisi informasi yang diorganisasikan dalam sebuah sistem
  - Ada proses yang memudahkan temu kembali informasi dan strategi penelusuran yang tepat sehingga dokumen yang sesuai dengan kebutuhan dapat ditemukan kembali.

TUJUAN IR DALAM NLP

1. Mengambil dokumen yang relevan dari korpus besar
2. Memahami maksud pengguna (query understanding)
3. Memberikan hasil yang cepat dan akurat

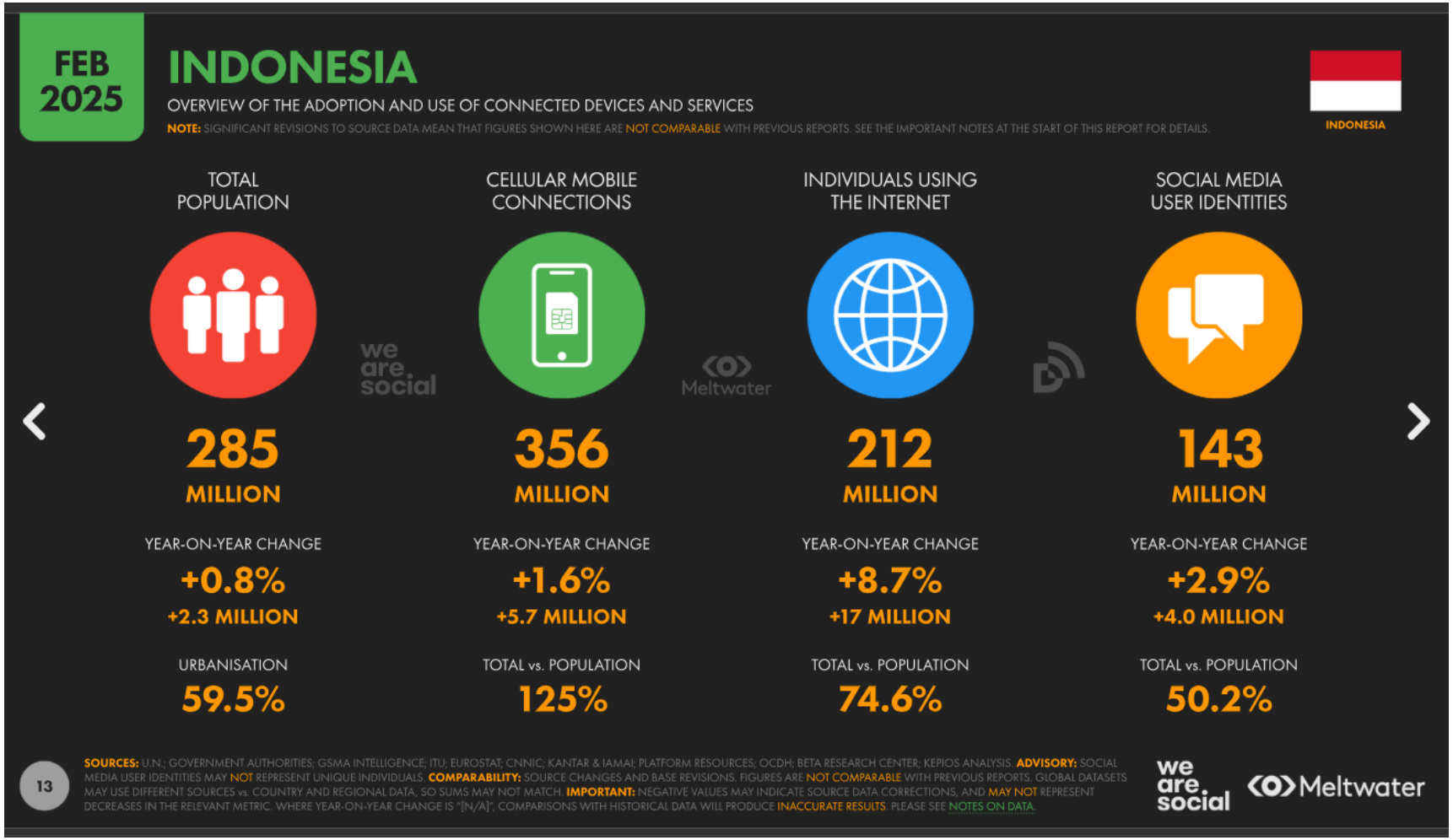
KOMPONEN UTAMA DALAM IR

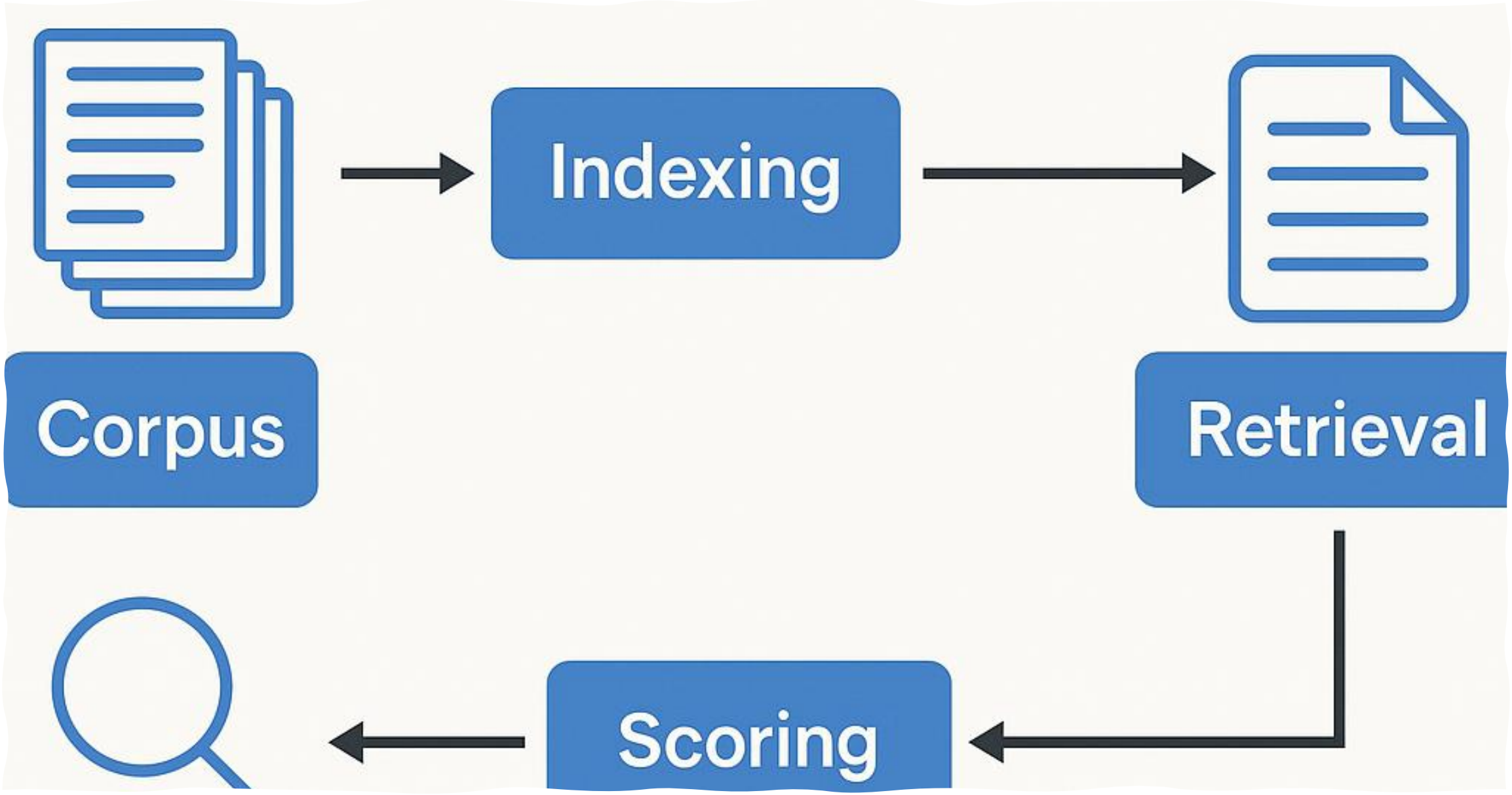
1. Corpus: Kumpulan dokumen teks
2. Indexing: Proses mengubah dokumen menjadi format pencarian
3. Query: Masukan dari pengguna
4. Scoring: Menghitung relevansi dokumen terhadap query
5. Retrieval: Mengambil dokumen relevan berdasarkan skor tertinggi

Ilustrasi

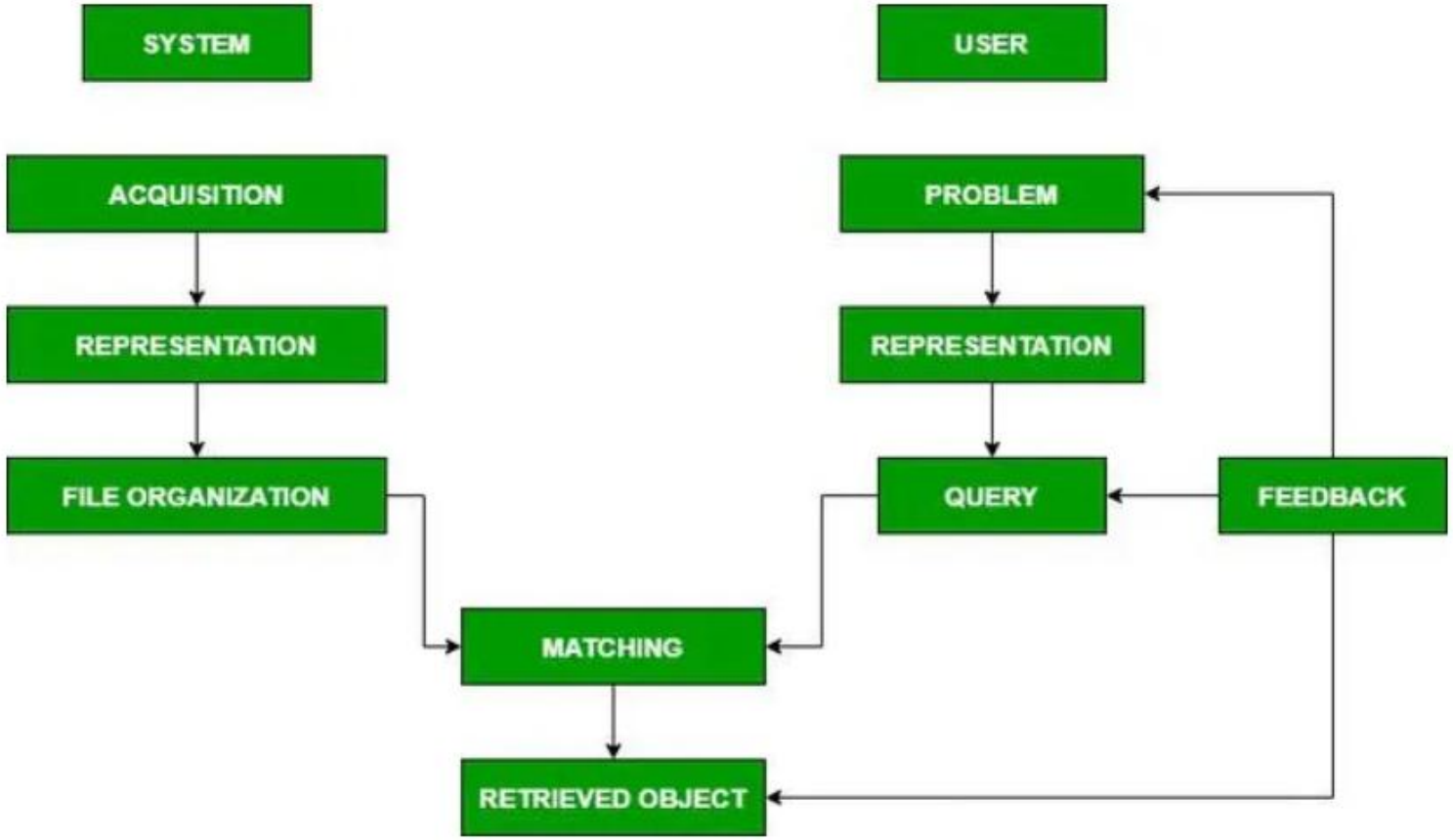
http://boston.lti.cs.cmu.edu/classes/11-744/treclogo-c.gif

KONDISI DI INDONESIA




Corpus
Indexing
Retrieval
Scoring

IR DALAM NLP


SYSTEM
USER
ACQUISITION
PROBLEM
REPRESENTATION
REPRESENTATION
FILE ORGANIZATION
QUERY
FEEDBACK
MATCHING
RETRIEVED OBJECT

SUPERVISED LEARNING

PENGERTIAN

- Salah satu pendekatan *machine learning* di mana model dilatih menggunakan data yang **memiliki** label. Artinya, setiap data pelatihan memiliki input (fitur) dan output (label) yang diketahui, sehingga algoritma dapat mempelajari hubungan di antara keduanya

JENIS SUPERVISED LEARNING

Jenis	Tujuan	Contoh
Klasifikasi	Pengelompokan berdasarkan label/kelas	Klasifikasi camaba diterima atau tidak diterima di suatu universitas
Regresi	Prediksi nilai kontinu	Prediksi harga emas

ALGORITMA SUPERVISED LEARNING

- Decision Tree
- Support Vector Machine (SVM)
- Naïve Bayes Classifier
- Neural Network
- K-Nearest Neighbors
- Random Forest
- Logistic Regresion

CODING PYTHON DENGAN ALGORITMA DECISION TREE

```python
# Import library
from sklearn.tree import DecisionTreeClassifier

# Data pelatihan
X_train = [[1], [2], [3], [4]]
y_train = ['A', 'A', 'B', 'B']

# Membuat dan melatih model
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Prediksi data baru
X_test = [[1.5], [3.5]]
predictions = model.predict(X_test)

print("Prediksi:", predictions)
```

UNSUPERVISED LEARNING

PENGERTIAN

- Pendekatan *machine learning* di mana model dilatih menggunakan dataset yang **tidak memiliki** label.
- *Unsupervised Learning* digunakan untuk menemukan struktur tersembunyi, pola, atau hubungan di dalam data. Model tidak diberi tahu apa yang harus dicari, sehingga belajar dari data itu sendiri tanpa panduan eksplisit.
- Salah satu Algoritma *Unsupervised Learning* adalah : *K-Means Clustering*

KARAKTERISTIK UNSUPERVISED LEARNING

UNLABELED DATA

PENEMUAN MODEL POLA, GROUP, DAN STRUKTUR DALAM DATA

CODING UNSUPERVISED LEARNING MENGGUNAKAN PYTHON

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Data dummy
data = {
    'Pendapatan': [15, 16, 17, 18, 19, 20, 25, 30, 35, 40],
    'Pengeluaran': [39, 81, 6, 77, 40, 76, 80, 30, 20, 60]
}
df = pd.DataFrame(data)

# Model KMeans dengan 2 cluster
kmeans = KMeans(n_clusters=2, random_state=0)
kmeans.fit(df)

# Tambahkan kolom cluster ke data
df['Cluster'] = kmeans.labels_

# Visualisasi
plt.scatter(df['Pendapatan'], df['Pengeluaran'], c=df['Cluster'
plt.xlabel('Pendapatan')
plt.ylabel('Pengeluaran')
plt.title('Segmentasi Pelanggan dengan K-Means')
plt.show()
```

INFORMATION EXTRACTION

PENGERTIAN

- Information Extraction (IE) adalah **proses otomatis** dalam mengidentifikasi, mengekstraksi, dan **menyusun informasi spesifik** yang bernilai dari teks tidak terstruktur (seperti artikel, email, komentar, atau dokumen berita).
- Tujuan utamanya adalah **mengubah teks mentah menjadi informasi yang terstruktur**, seperti dalam bentuk entitas, relasi, atau fakta

KOMPONEN IE

1. Named Entity Recognition (NER): Mengidentifikasi entitas seperti nama orang, lokasi, organisasi, tanggal, dll.
2. Relation Extraction: Mengidentifikasi hubungan antar entitas (misal: siapa melakukan?, Apa?, kepada siapa?).
3. Event Extraction: Menangkap peristiwa penting dari teks.
4. Template Filling: Menyusun hasil ekstraksi ke dalam struktur data tertentu.

CONTOH INFORMATION EXTRACTION

- Misalkan ada kalimat : “Presiden Prabowo Subianto melakukan sidak program makan bergizi gratis di salah satu SD Negeri di Bantul, Yogyakarta pada 9 Mei 2025”
- Hasil IE:
  - Entitas :
    - Person : Prabowo Subianto
    - Location : Bantul, Yogyakarta
    - Organization : SD
  - Waktu : 9 Mei 2025
  - Aksi : Sidak
  - Objek : makan bergizi gratis

TEKNIK IE

- Rule-based extraction: menggunakan pola linguistic atau regular expression
- Machine Learning: menggunakan algoritma klasifikasi
- Deep Learning: menggunakan algoritma LSTM, BERT, Transformer
- Pretrained Models: menggunakan SpaCY, Flair, Stanza, HuggingFace Transformer (Library Python)

CONTOH PENGGUNAAN NER MENGGUNAKAN PYTHON

```python
import spacy
nlp = spacy.load("en_core_web_sm")

text = "President Prabowo Subianto visited a elementary school in Yogyakarta regarding the free nutritious meal program on May 9, 2025"
doc = nlp(text)

print("Named Entities, Phrases, Labels, Time, Location, object")
for ent in doc.ents:
    print(ent.text, ent.label_)
```

```
Named Entities, Phrases, Labels, Time, Location, object
Prabowo Subianto PERSON
Yogyakarta GPE
May 9, 2025 DATE
```

CONTOH PENGGUNAAN RELATION EXTRACTION

```
for token in doc:
    if token.dep_ == "ROOT":
        subject = [w for w in token.lefts if w.dep_ == "nsubj"]
        object_ = [w for w in token.rights if w.dep_ == "dobj"]
        if subject and object_:
            print(f"Subjek: {subject[0]}, Predikat: {token}, Objek: {object_[0]}")
```

```
Subjek: Subianto, Predikat: visited, Objek: school
```

CONTOH PENGGUNAAN REGULAR EXPRESSION

```
import re
text = "Jika ada permasalahan, bisa menghubungi saya melalui 0812-3456-7890 dan email saya adityopw@gmail.com"

phone = re.findall(r"\d{4}-\d{4}-\d{4}", text)
email = re.findall(r"[\w\.-]+@[\w\.-]+", text)

print("Telepon:", phone)
print("Email:", email)
```

```
Telepon: ['0812-3456-7890']
Email: ['adityopw@gmail.com']
```

TERIMAKASIH