

Virtual Internship Experience

**HOME
CREDIT**

**Programming Language
Scripting : Various
Library Used in Python**



**HOME
CREDIT**

Outline

- 1** *Pengertian Library pada Python*
- 2** *Jenis Library Python*
- 3** *Kegunaan Library Python*

Pengertian Library pada Python



HOME
CREDIT

Library python adalah kumpulan modul terkait, berisi kumpulan kode yang dapat digunakan berulang kali dalam program yang berbeda. Adanya library membuat pemrograman python menjadi lebih sederhana dan nyaman bagi programmer karena tidak perlu menulis kode yang sama berulang kali untuk program yang berbeda.



Jenis Library pada Python (1)



HOME
CREDIT



Best Python Libraries for Machine Learning and Deep Learning

Jenis Library pada Python (2)



HOME
CREDIT

Tensorflow - TensorFlow adalah platform end-to-end open-source untuk membuat aplikasi machine learning atau komputasi numerik cepat yang dibuat dan dirilis oleh Google.

NumPy (Numerical Python) - Numpy adalah library python yang digunakan untuk bekerja dengan array dan juga memiliki fungsi yang bekerja dalam domain aljabar linier, transformasi fourier, dan matriks.

Scikit-learn - Scikit-learn adalah library python terkenal yang digunakan untuk data kompleks. Perpustakaan open source ini mendukung machine learning dengan mendukung berbagai algoritma yang diawasi dan tidak diawasi seperti regresi linier, klasifikasi, pengelompokan, dan lain sebagainya. Library ini bekerja sama dengan Numpy dan SciPy.

PyTorch - Pytorch adalah perpustakaan machine learning terbesar yang mengoptimalkan komputasi tensor.

Jenis Library pada Python (2)



HOME
CREDIT

Pandas - Pandas adalah perpustakaan penting bagi para data scientist. Library untuk machine learning yang bersifat open source ini menyediakan struktur data tingkat tinggi yang fleksibel serta berbagai alat analisis.

Keras - Keras adalah API deep learning yang ditulis dengan python dan berjalan di atas platform machine learning TensorFlow.

NLTK - NLTK adalah singkatan dari Natural Language Toolkit dan merupakan pustaka Python untuk bekerja dengan pemrosesan bahasa alami.

Kegunaan Library Python (1)



HOME
CREDIT

TensorFlow

Beberapa area penting dalam ML dan DL tempat **TensorFlow** menonjol adalah:

- Menangani jaringan neural dalam
- Pemrosesan Bahasa Alami
- Persamaan Diferensial Parsial
- Kemampuan abstraksi
- Pengenalan Gambar, Teks, dan Ucapan
- Kolaborasi ide dan kode yang mudah

Kegunaan Library Python (2)



HOME
CREDIT

NumPy

Beberapa poin yang mendukung **NumPy** adalah:

- Dukungan untuk operasi matematika dan logika
- Manipulasi bentuk
- Kemampuan Menyortir dan Memilih
- Transformasi Fourier Diskrit
- Operasi statistik dan aljabar linier dasar
- Simulasi acak
- Dukungan untuk larik n-dimensi

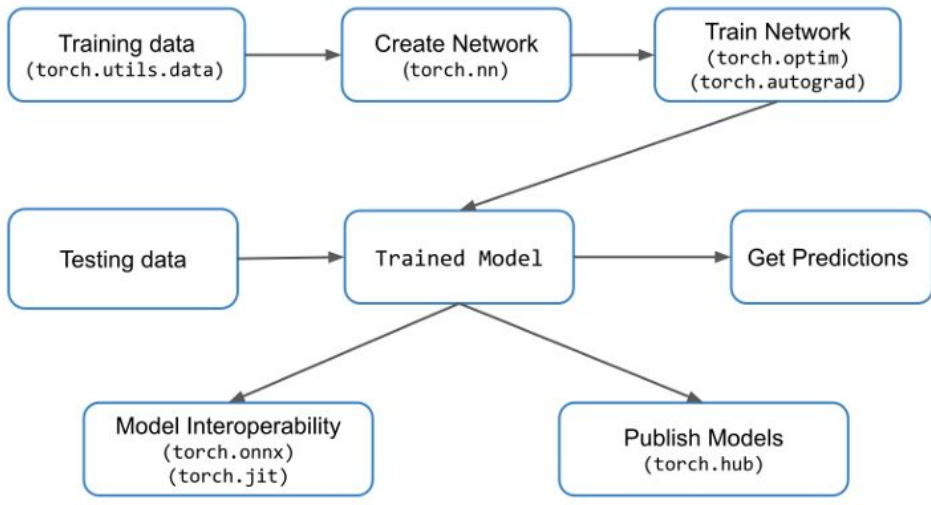
Scikit-learn

Scikit-learn difokuskan pada pemodelan data dan bukan pada tugas-tugas lain seperti memuat, menangani, manipulasi, dan visualisasi data . Ini dianggap cukup untuk digunakan sebagai ML end-to-end, mulai dari tahap penelitian hingga penerapan. Untuk pemahaman yang lebih dalam tentang scikit-learn, Anda dapat melihat tutorial Scikit-learn .

Kegunaan Library Python (3)



Pytorch



Beberapa fitur penting yang membedakan

PyTorch dari TensorFlow adalah:

- Komputasi sensor dengan kemampuan untuk pemrosesan yang dipercepat melalui Unit Pemrosesan Grafik
- Mudah dipelajari, digunakan, dan diintegrasikan dengan ekosistem Python lainnya
- Dukungan untuk jaringan neural yang dibangun pada sistem auto diff berbasis tape

Kegunaan Library Python (4)



Pandas

Pandas

```
In [8]: import pandas as pd

In [10]: pd_series = pd.Series(data=["Val1", "Val2", "Val3"], index=range(0,3), name="Series_object")

In [11]: pd_series
Out[11]: 0    Val1
         1    Val2
         2    Val3
         Name: Series_object, dtype: object

In [14]: df = pd.DataFrame(data=[{"col_1": [1, 2, 3, 4],
                                   "col_2": ["A", "B", "C", "D"]}]

In [15]: df
Out[15]:
```

	col_1	col_2
0	1	A
1	2	B
2	3	C
3	4	D

Beberapa fitur hebat Pandas saat menangani data adalah:

- Pembentukan ulang dan perputaran kumpulan data
- Penggabungan dan penggabungan kumpulan data
- Penanganan data yang hilang dan keselarasan data
- Berbagai opsi pengindeksan seperti pengindeksan sumbu hierarki, pengindeksan Fancy
- Opsi filtrasi data

Kegunaan Library Python (5)



HOME
CREDIT

Keras

Keras

```
In [2]: from keras.models import Sequential
        from keras.layers.core import Dense, Activation
        from keras.utils import np_utils
```

Using TensorFlow backend.

```
In [3]: model = Sequential()
        model.add(Dense(16, input_shape=(4,)))
        model.add(Activation('sigmoid'))
        model.add(Dense(3))
        model.add(Activation('softmax'))
        model.compile(optimizer='adam',
                      loss='categorical_crossentropy', metrics=["accuracy"])
```

```
In [4]: model.summary()
```

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 16)	80
activation_1 (Activation)	(None, 16)	0
dense_2 (Dense)	(None, 3)	51
activation_2 (Activation)	(None, 3)	0
Total params: 131		
Trainable params: 131		
Non-trainable params: 0		

Keras menampilkan beberapa blok penyusun dan alat yang diperlukan untuk membuat jaringan saraf seperti:

- Lapisan saraf
- Aktivasi dan fungsi biaya
- Tujuan
- Normalisasi batch
- Putus sekolah
- Pooling

Kegunaan Library Python (6)



HOME
CREDIT

NLTK

NLTK

```
In [1]: import nltk
sentence = "The quick brown fox jumped over the lazy dog"

#tokenize
tokens = nltk.word_tokenize(sentence)
print("Tokens={}".format(tokens))

# POS Tags
tagged = nltk.pos_tag(tokens)
print("POS Tags={}".format(tagged))

Tokens=['The', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']
POS Tags=[('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox', 'NN'), ('jumped', 'VBD'), ('over', 'IN'), ('the', 'DT'), ('la
zy', 'JJ'), ('dog', 'NN')]
```

Beberapa yang menarik dari NLTK adalah:

- Mencari kata kunci dalam dokumen
- Tokenisasi dan klasifikasi teks
- Pengakuan pada suara dan tulisan tangan
- Lemmatizing dan Stemming of words



**How is it implemented in
Home Credit Indonesia?**



Thank You!



**HOME
CREDIT**