



Library Populer pada Python

Kali ini kita akan membahas Modul dan Library yang sering digunakan pada Python. Artikel ini akan terus di-update sesuai dengan perkembangan dan jika diperlukan akan ditambah. Modul yang sering dipakai secara umum dalam *scripting* sehari-hari antara lain:

String

Dokumentasi: <https://docs.python.org/3/library/string.html>.

String merupakan salah satu modul bawaan Python yang tidak perlu dideklarasikan. Modul yang sudah bawaan dari Python disebut modul *built-in* (<https://docs.python.org/3/library/stdtypes.html>). Termasuk di dalam modul built in antara lain adalah string, list, range, boolean operator, dan sebagainya. Pada modul string terdapat fungsi-fungsi yang dapat dioperasikan pada variabel bertipe string seperti di bawah.

- `upper()`: Ubah setiap huruf dalam string menjadi huruf kapital.
- `lower()`: Ubah setiap huruf dalam string menjadi huruf kecil.
- `split()`: Pisahkan teks berdasarkan delimiter(karakter pemisah).
- `title()`: Jadikan setiap awal kata kapital.
- `zfill()`: Tambahkan nol di awal string sebanyak nilai yang ada pada parameter.

Contoh implementasi kode di atas pada Notebook IBM Watson Studio sebagai berikut:

```
In [3]: tesString = "dicoding"
        tesString.upper()

Out[3]: 'DICODING'

In [4]: tesString.lower()

Out[4]: 'dicoding'

In [5]: tesString.title()

Out[5]: 'Dicoding'

In [6]: tesString.split("i")

Out[6]: ['d', 'cod', 'ng']

In [7]: tesString.zfill(20)

Out[7]: '0000000000000dicoding'
```

OS

Dokumentasi: <https://docs.python.org/3/library/os.html>

Modul OS pada Python merupakan modul untuk fungsi-fungsi yang berkaitan dengan sistem operasi, misalnya `open()`, `path()`, `getcwd()`, dan fungsi lainnya. Modul ini memungkinkan Anda untuk memanfaatkan f



DIBANTU



Contoh kode di bawah menunjukkan fungsi `os.getcwd()`. Fungsi ini akan mengembalikan string representasi dari Current Working Directory yaitu direktori di mana program Python kita berada. Fungsi ini berlaku di semua OS.

```
1. import os
2. print(os.getcwd())
```

```
/Users/gheddi/PycharmProjects/unit_test
Process finished with exit code 0
```

File Input/Output

Dokumentasi: <https://docs.python.org/3/tutorial/inputoutput.html>.

Modul Input/Output adalah modul yang berkaitan untuk fungsi yang mengurus masukan dan keluaran pada Python. Contoh fungsi yang ada pada modul ini adalah fungsi untuk menampilkan hasil dari program Python ke layar dan membaca teks yang kita ketik di keyboard.

Kode di bawah menunjukkan 2 contoh fungsi yang ada pada modul Input/Output yaitu membaca teks yang kita ketik di keyboard dan menampilkan teks tersebut.

```
1. string = input("Masukkan sesuatu: ")
2. print("Input yang masuk adalah : ", string)
```

Pickle

Dokumentasi: <https://docs.python.org/3/library/pickle.html?highlight=pickle>

Jika Anda memiliki sebuah list yang ingin disimpan atau ditransmisikan tanpa khawatir bentuknya akan rusak atau kacau, Anda dapat memanfaatkan fungsi dari library pickle. Pickle merupakan fungsi Object Serialization pada Python. Pickling adalah istilah untuk mengubah objek menjadi byte stream, sedangkan Unpickling adalah perlakuan sebaliknya.

Kode berikut adalah contoh bagaimana melakukan proses pickle pada sebuah object dictionary dan menyimpannya pada sebuah file.

```
1. import pickle
2. contoh_dictionary = {1:"6", 2:"2", 3:"f"}
3. pickle_keluar = open("dict.pickle","wb")
4. pickle.dump(contoh_dictionary, pickle_keluar)
5. pickle_keluar.close()
```



DIBANTU



```
1. pickle_masuk = open("dict.pickle","rb")
2. contoh_dictionary = pickle.load(pickle_masuk)
```

JSON

Dokumentasi: <https://docs.python.org/3/library/json.html>.

Untuk serialization dengan bahasa lain, umumnya kita menggunakan JSON (JavaScript Object Notation) yang memiliki beberapa perbedaan karakteristik dengan pickle, yakni:

- JSON adalah format text-serialization dan umumnya menggunakan Unicode atau UTF-8. Sementara pickle bersifat binary serialization.
- JSON dapat dibaca dengan mudah oleh manusia, sementara pickle tidak.
- JSON dapat dioperasikan dan digunakan di luar ekosistem Python. Pickle adalah Python-specific.
- JSON secara default hanya dapat merepresentasikan subset dari built-in type pada Python.
- Pickle dapat merepresentasikan hampir (jika tidak seluruh) tipe Python dan secara default melakukan kompresi data.

Seperti yang telah disebutkan sebelumnya, JSON adalah format text yang ditujukan untuk serialization. Agar data dapat dengan mudah ditransmisikan antar berbagai sumber tanpa khawatir bentuknya kacau, menggunakan JSON adalah salah satu pilihan yang tepat.

JSON memiliki format yang hampir mirip dengan dictionary di mana data disimpan dengan format key & value pair. Namun tentunya JSON jauh lebih kompleks dari dictionary. Dapat dilihat dari contoh JSON untuk data pembelian di bawah. Dengan JSON kita dapat menyimpan data dengan lebih terorganisir. Sebuah key seperti children di bawah dapat memiliki sebuah dictionary baru yang berisi informasi terkait objek children tersebut.

```
{
  "firstName": "Jane",
  "lastName": "Doe",
  "hobbies": ["running", "sky diving", "singing"],
  "age": 35,
  "children": [
    {
      "firstName": "Alice",
      "age": 6
    },
    {
      "firstName": "Bob",
      "age": 8
    }
  ]
}
```

Untuk membuat JSON sederhana ketik seperti kode di bawah.



DIBANTU



```
2.  
3. # contoh JSON:  
4. x = '{ "nama":"Buchori", "umur":22, "Kota":"New York"}'  
5.  
6. # parse x:  
7. y = json.loads(x)  
8.  
9. print(y["umur"])
```

Scraper

Dokumentasi urllib: <https://docs.python.org/3/library/urllib.html>.

Dokumentasi beautifulsoup4: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.

Web scrapping adalah sebuah proses terotomasi untuk mendapatkan dan parsing data dari web. Disiplin seperti data science, business intelligence, dan lainnya mendapatkan banyak manfaat dari menganalisis data pada situs-situs yang ada di internet. Ada 2 buah library yang terkenal untuk web scraping yaitu *urllib* dan *beautifulsoup*.

- **Urllib** adalah library bawaan dari Python. Kode di bawah adalah contoh untuk memulai proses scraping pada situs dengan domain *python.org* dan menampilkan isi dari tag title dari situs tersebut.

```
1. from urllib.request import urlopen  
2. url = "http://python.org/"  
3. page = urlopen(url)  
4. html = page.read().decode("utf-8")  
5. start_index = html.find("<title>") + len("<title>")  
6. end_index = html.find("</title>")  
7. title = html[start_index:end_index]  
8. title
```

- **Beautifulsoup** adalah library yang penggunaanya lebih sederhana dari urllib. Untuk menggunakan beautifulsoup Anda harus menginstalnya terlebih dahulu. Berikut adalah contoh penulisan kode beautifulsoup.

```
1. from urllib.request import urlopen  
2. from bs4 import BeautifulSoup  
3.  
4. url = "http://python.org/"  
5. page = urlopen(url)  
6. html = page.read().decode("utf-8")  
7. soup = BeautifulSoup(html, "html.parser")  
8. soup.title
```



DIBANTU



Regex

Dokumentasi: <https://docs.python.org/3/library/re.html>.

Regex atau regular expression adalah sebuah cara untuk mencari teks berdasarkan pola tertentu. Ketika kita ingin mencari sebuah kata dalam kamus, misalnya arti dari kata parsing, kita akan mencari kata tersebut di halaman yang memiliki kata dengan awalan p, atau pa. Regex bekerja dengan konsep yang sama. Pada regex, kita mencari sebuah kata atau kumpulan kata dengan memberikan pola yang kita inginkan. Contoh umum regex adalah pada email di mana kita dapat menggunakan regex untuk mengecek apakah karakter @ ada pada email atau tidak.

Contoh di bawah menunjukkan penggunaan regex. Pada variabel pattern di bawah, ^a berarti kita ingin mencari teks dengan awalan 'a', dan s\$ berarti kita ingin mencari string berakhiran 's'.

```
1. import re
2.
3. pola= '^a...s$'
4. string_tes= 'abyss'
5. hasil= re.match(pola, string_tes)
6.
7. if hasil:
8.     print("Pencarian berhasil.")
9. else:
10.    print("Pencarian gagal.")
```

Argument Parser

Dokumentasi Getopt: <https://docs.python.org/3.8/library/getopt.html>.

Dokumentasi ArgParse: <https://docs.python.org/3.8/library/argparse.html>.

Argument parser bermanfaat jika kita ingin membuat program atau skrip kecil yang langsung menerima parameter pada saat pemanggilan program. Hal ini biasa digunakan dalam pemanggilan aplikasi atau skrip di CLI/terminal **nix-based* misalnya linux dan MacOS. Contoh perintah dimaksud adalah:

```
1. python panggildicoding.py -o
```

Contoh dengan menggunakan ArgParse untuk menambahkan Argument yang bersifat opsional / tidak wajib:

file: panggildicoding.py



DIBANTU



```
2.
3. parser = argparse.ArgumentParser()
4. parser.add_argument('-o', '--output', action='store_true', help="tampilkan output")
5. args = parser.parse_args()
6.
7. if args.output:
8.     print("Halo, ini merupakan sebuah output dari panggildicoding.py")
```

Maka pada saat dijalankan, terdapat beberapa hal yaitu:

- Berkas panggildicoding.py dapat menerima parameter -o atau --output.
- Jika kita memanggil berkas tanpa parameter -o maka berkas tidak akan menampilkan apapun.
- Jika kita memanggil dengan -o atau --output maka berkas akan menampilkan **Halo, ini merupakan sebuah output dari panggildicoding.py**.
- Jika kita memanggil --help, maka akan tampil help dengan penjelasan "tampilkan output".

```
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ nano panggildicoding.py
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py -o
Halo, ini merupakan sebuah output dari panggildicoding.py
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --output
Halo, ini merupakan sebuah output dari panggildicoding.py
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --help
usage: panggildicoding.py [-h] [-o]

optional arguments:
  -h, --help  show this help message and exit
  -o, --output  tampilkan output
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$
```

Kita juga bisa membuat argumennya bersifat wajib. Modifikasi berkas panggildicoding.py menjadi seperti berikut:

```
1. import argparse
2.
3. parser = argparse.ArgumentParser()
4. parser.add_argument('-n', '--nama', required=True, help="Masukkan Nama Anda")
5. args = parser.parse_args()
6.
7. print("Terima kasih telah menggunakan panggildicoding.py, "+args.nama)
```

Maka pada saat dijalankan, terdapat beberapa hal yaitu:

- Berkas panggildicoding.py harus dipanggil dengan parameter -n atau --nama.
- Jika kita memanggil berkas tanpa parameter -n maka berkas akan meminta parameter n atau nama.
- Jika kita memanggil dengan -n NAMA KITA atau --nama NAMA KITA maka berkas akan menampilkan **Terima kasih telah menggunakan panggildicoding.py NAMA KITA**.



DIBANTU



```
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py
usage: panggildicoding.py [-h] -n NAMA
panggildicoding.py: error: the following arguments are required: -n/--nama
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --help
usage: panggildicoding.py [-h] -n NAMA

optional arguments:
  -h, --help            show this help message and exit
  -n NAMA, --nama NAMA  Masukkan Nama Anda
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py -n Yoza
Terima kasih telah menggunakan panggildicoding.py, Yoza
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --nama Aprilio
Terima kasih telah menggunakan panggildicoding.py, Aprilio
```

Anda dapat mensuplai lebih dari satu argumen, dengan menambahkan **parser.add_argument** sejumlah yang Anda inginkan.

Ayo berlatih melengkapi script sebelumnya agar mampu menampilkan hasil berikut:

```
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --nama Dicoding --tanggallahir 05-01-2015
Terima kasih telah menggunakan panggildicoding.py pada tahun 2020, kakak Dicoding
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --nama Dicoding --tanggallahir 05-01-1991
Terima kasih telah menggunakan panggildicoding.py pada tahun 2020, kakak Dicoding
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --nama Dicoding --tanggallahir 05-01-1990
Terima kasih telah menggunakan panggildicoding.py pada tahun 2020, bapak Dicoding
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --nama Dicoding
usage: panggildicoding.py [-h] -n NAMA -t TANGGALLAHIR
panggildicoding.py: error: the following arguments are required: -t/--tanggallahir
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$ python3 panggildicoding.py --help
usage: panggildicoding.py [-h] -n NAMA -t TANGGALLAHIR

optional arguments:
  -h, --help            show this help message and exit
  -n NAMA, --nama NAMA  Masukkan Nama Anda
  -t TANGGALLAHIR, --tanggallahir TANGGALLAHIR
                        Masukkan Tanggal Lahir Anda (DD-MM-YYYY)
yoza@DESKTOP-C05E3MM:/mnt/c/Users/yoza1$
```

Ketentuan skrip:

- Berkas panggildicoding.py harus dipanggil dengan parameter n/nama dan t/tanggallahir.
- Format Tanggal lahir adalah dd-mm-yyyy.
- Jika kita memanggil berkas tanpa parameter maka berkas akan menolak.
- Jika usia pemanggil saat ini kurang dari 30 tahun, tambahkan panggilan kakak, selain itu gunakan panggilan bapak sebelum NAMA KITA.

[← Sebelumnya](#)

[Selanjutnya >](#)



Dicoding Space
Jl. Batik Kumeli No.50, Sukaluyu,
Kec. Cibeunying Kaler, Kota Bandung
Jawa Barat 40123



Penghargaan

Decode Ideas
Discover Potential

[Tentang Kami](#)

[Blog](#)

[Reward](#)

[Showcase](#)

[Hubungi Kami](#)

[FAQ](#)



