**Pengenalan Library**

Dalam pengembangan program atau aplikasi, kita tidak akan lepas dari peran **package** atau **library**. Sebagaimana yang sudah dijelaskan dalam materi-materi sebelumnya, package adalah sebuah direktori yang berisi satu atau lebih modul yang terkait dan saling berhubungan, sedangkan library adalah koleksi dari banyaknya modul dan paket yang saling terkait dan dapat digunakan berulang kali.

Pada materi ini, kita akan mempelajari berbagai library yang telah dikembangkan oleh banyak pengembang dan dapat digunakan kembali oleh kita. Jumlah library Python sangat banyak yang terbagi menjadi **Python Standard Library**dan **Python External Library**.

Python Standard Library adalah jenis library yang telah terinstal secara otomatis ketika kita melakukan instalasi Python. Anda tidak perlu melakukan instalasi kembali jika ingin menggunakannya. Beberapa contoh Python Standard Library adalah “os”, “datetime”, “re”, serta lainnya yang dapat Anda baca lebih lengkap pada laman [berikut](https://docs.python.org/3/library/" \t "_blank).

Python External Library adalah kumpulan kode yang telah dikembangkan oleh orang lain atau komunitas dan disediakan dalam bentuk paket atau modul yang dapat diimpor. Jenis library ini mengharuskan Anda untuk melakukan instalasi agar dapat digunakan. External library ini dikembangkan oleh individu atau organisasi di luar tim inti pengembang Python.

Untuk melakukan instalasi library eksternal, Anda dapat melakukan beberapa cara, seperti menggunakan PIP dan conda.

**PIP**

PIP adalah package manager resmi dari Python yang dapat digunakan untuk mengunduh, menginstal, menghapus, dan mengelola package Python dari Python Package Index (PyPI) dan repositori lainnya. [PyPI](https://pypi.org/" \t "_blank) merupakan repositori online yang menyediakan ribuan package dari Python dan siap digunakan oleh para pengembang.

Selain mengelola paket, Anda juga bisa membuat lingkungan virtual dalam Python menggunakan PIP. Hanya saja, pip cenderung difokuskan untuk mengelola instalasi paket dibanding lingkungan virtual. Lingkungan virtual memungkinkan kita untuk membuat sebuah lingkungan terisolasi yang terpisah satu sama lain. Kita akan bahas lebih dalam terkait lingkungan virtual pada materi conda.

Kabar baiknya, pip biasanya telah terpasang secara otomatis jika Anda menggunakan Python 2 untuk versi 2.7.9 ke atas atau Python 3 untuk versi 3.4 ke atas. Maka dari itu, silakan periksa bahwa pip telah terpasang dengan menjalankan perintah berikut.

Anda telah menyiapkan pip sebagai package manager yang mendukung instalasi package dan library Python. Untuk melakukan instalasi, Anda bisa mengikuti perintah berikut.

|  |
| --- |
| pip install <nama-package> |

Ganti <nama-package> dengan nama package atau library yang ingin Anda unduh. Untuk menghapus package, Anda bisa mengikuti perintah berikut.

|  |
| --- |
| pip uninstall <nama-package> |

Silakan ganti <nama-package> dengan package atau library yang ingin Anda hapus.

Contohnya, jika ingin membuat lingkungan virtual menggunakan pip, Anda harus menginstal package pipenv berikut.

|  |
| --- |
| pip install pipenv |

Anda bisa membaca lebih detail terkait pipenv pada laman [ini](https://pypi.org/project/pipenv/" \t "_blank).

**Conda**

Selain pip yang termasuk package manager resmi dari Python, ada juga conda yang merupakan package manager dan environment manager untuk Python. Conda memungkinkan kita untuk membuat dan mengelola lingkungan (environment) terisolasi atau terpisah satu sama lain. Dengan terisolasinya setiap lingkungan tersebut, kita diuntungkan untuk mencegah konflik yang terjadi antar proyek.

Contohnya ketika Anda memiliki proyek machine learning dengan versi Python yang digunakan adalah Python 3.9. Di sisi lain, Anda memiliki proyek web development A dengan versi Python yang digunakan adalah Python 3.8. Ada pula proyek web development B dengan versi Python yang digunakan sama-sama menggunakan Python versi 3.8.



Dengan adanya lingkungan yang terisolasi, Anda bisa menyesuaikan semua library, modul, hingga versi Python sesuai dengan kebutuhan masing-masing proyek. Conda sendiri dapat diinstal melalui Anaconda dan Miniconda. Anaconda merupakan sistem distribusi perangkat lunak yang di dalamnya mencakup Conda. Ketika menginstal Anaconda, Anda pun dapat menggunakan beberapa library dan plugin tambahan melalui Anaconda tersebut. Miniconda adalah versi ringan dari Anaconda. Miniconda hanya berisi conda dan beberapa package dasar yang diperlukan untuk menjalankannya.

Conda sendiri hadir dalam dua bentuk utama: “conda” sebagai package dan environment manager serta “conda-forge”, yakni sebuah repositori berisi ribuan paket yang disediakan oleh komunitas Conda.

Kelebihan Conda adalah sifat tidak terbatasnya pada bahasa pemrograman Python. Conda juga mendukung paket-paket dalam bahasa pemrograman lain, seperti R.

Kali ini, kita akan menginstal conda melalui sistem terdistribusi Anaconda, silakan ikuti langkah-langkah berikut.

1. Kunjungi laman berikut: [https://www.anaconda.com/download](https://www.anaconda.com/download" \t "_blank).
2. Silakan scroll ke bawah dan Anda akan menjumpai “Anaconda Installer” yang tersedia untuk tiga sistem operasi, yakni Windows, Linux, dan Mac. Silakan unduh Anaconda sesuai dengan sistem operasi Anda.
3. Setelah mengunduh, lakukan instalasi dengan menjalankan file installer yang telah Anda unduh.

Terakhir, Anda memeriksa bahwa conda telah terinstal. Silakan buka “Anaconda Prompt” dengan klik windows button atau sistem pencarian sesuai sistem operasi Anda dan ketik “Anaconda Prompt”.

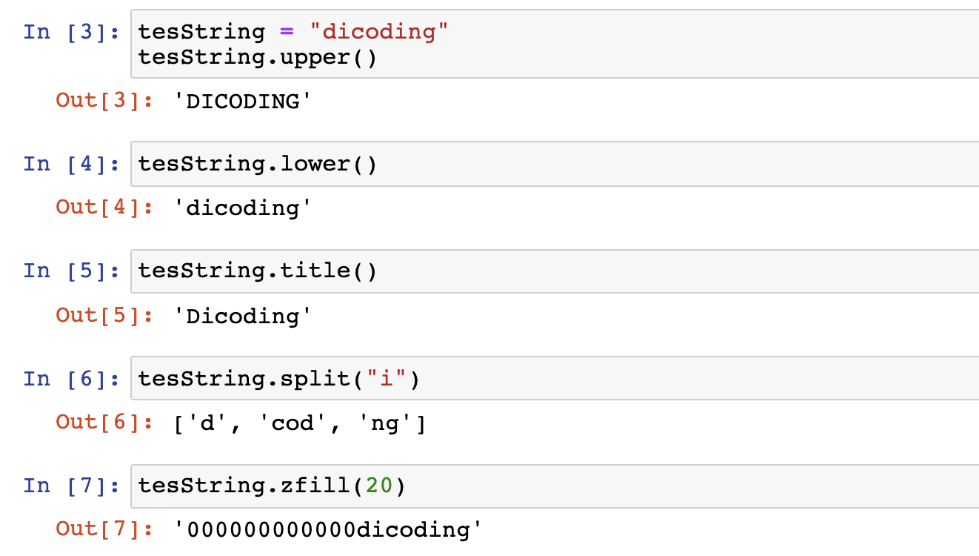
Pada terminal tersebut, jalankan kode berikut.

|  |
| --- |
| conda -V |

**Library Text Processing**

Pertama adalah sekumpulan library yang bertujuan untuk melakukan pemrosesan teks dan menyederhanakan serta mempercepat tugas-tugas pemrosesan teks.

1. String  
   [String](https://docs.python.org/3/library/string.html)adalah salah satu modul bawaan Python yang tidak perlu dideklarasikan. Pada modul string ada fungsi-fungsi yang dapat dioperasikan pada variabel bertipe string seperti di bawah.
   1. upper(): Ubah setiap huruf dalam string menjadi huruf kapital.
   2. lower(): Ubah setiap huruf dalam string menjadi huruf kecil.
   3. split(): Pisahkan teks berdasarkan delimiter (karakter pemisah).
   4. title(): Jadikan setiap awal kata kapital.
   5. zfill(): Tambahkan nol di awal string sebanyak nilai yang ada pada parameter.



1. Regex  
   [Regex](https://docs.python.org/3/library/re.html) atau regular expression adalah sebuah cara untuk mencari teks berdasarkan pola tertentu. Umpamanya, ketika ingin mencari sebuah kata dalam kamus, misalnya arti dari kata parsing, kita akan mencari kata tersebut di halaman yang memiliki kata dengan awalan p, lalu pa. Regex bekerja dengan konsep yang sama.  
     
   Pada regex, kita mencari sebuah kata atau kumpulan kata dengan memberikan pola yang diinginkan. Contoh umum regex adalah pada email. Kita dapat menggunakan regex untuk mengecek bahwa 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'.

|  |
| --- |
| import re # Import modul regex    pola= '^a...s$'  string\_tes= 'abyss'  hasil= re.match(pola, string\_tes)    if hasil:  print("Pencarian berhasil.")  else:  print("Pencarian gagal.")  """  Output:  Pencarian berhasil.  """ |

Perlu diperhatikan bahwa beberapa modul perlu diimpor terlebih dahulu untuk bisa digunakan. Pada contoh di atas, kita melakukan “import re” untuk mengimpor modul regex pada Python

**Library Matematika**

Selanjutnya adalah library [math](https://docs.python.org/3/library/math.html" \t "_blank) yang termasuk salah satu modul bawaan Python dan menyediakan berbagai fungsi dan konstanta matematika. Anda hanya perlu melakukan impor untuk modul math. Berikut contoh penerapannya.

|  |
| --- |
| import math  print(math.sqrt(25))  print(math.pi)  """  Output:  5.0  3.141592653589793  """ |

Pada contoh di atas, hal pertama yang dilakukan adalah melakukan impor modul math untuk menyediakan berbagai fungsi dan konstanta matematika. Di bawahnya, kita mencoba melakukan operasi akar dari bilangan 25 yang hasilnya adalah 5. Kemudian, kita mencoba mendapatkan nilai pi dalam modul math yang bernilai “3.141592653589793”.

**Library Parser**

Library parser pada Python menyediakan fasilitas untuk menguraikan kode Python menjadi struktur data yang dapat diproses dan dianalisis. Anda dapat menggunakan [Getopt](https://docs.python.org/3.8/library/getopt.html" \t "_blank) atau [ArgParse](https://docs.python.org/3.8/library/argparse.html" \t "_blank).

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 berikut.

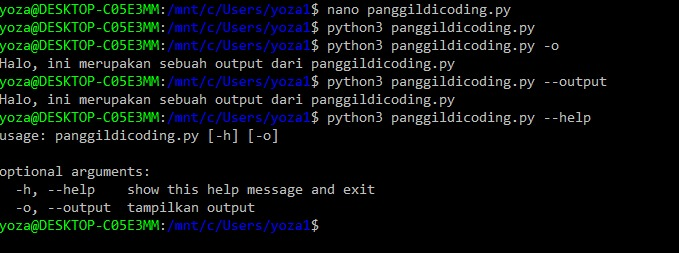
|  |
| --- |
| python panggildicoding.py -o |

Contoh tindakan menambahkan Argument yang bersifat opsional/tidak wajib dengan menggunakan ArgParse adalah berikut.

|  |
| --- |
| import argparse    parser = argparse.ArgumentParser()  parser.add\_argument('-o', '--output', action='store\_true', help="tampilkan output")  args = parser.parse\_args()    if args.output:  print("Halo, ini merupakan sebuah output dari panggildicoding.py") |

Jadi, pada saat dijalankan, ada beberapa hal yang perlu diperhatikan, yaitu berikut.

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

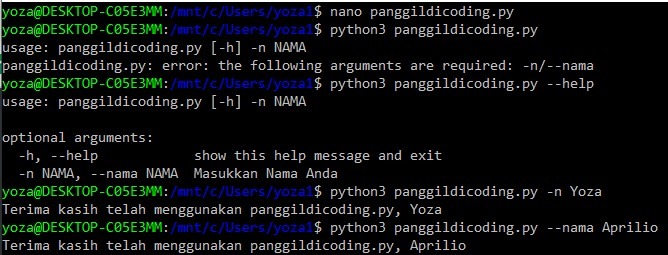


Kita juga bisa membuat argumennya bersifat wajib. Modifikasi berkas panggildicoding.py menjadi seperti berikut.

|  |
| --- |
| import argparse    parser = argparse.ArgumentParser()  parser.add\_argument('-n', '--nama', required=True, help="Masukkan Nama Anda")  args = parser.parse\_args()    print("Terima kasih telah menggunakan panggildicoding.py, "+args.nama) |

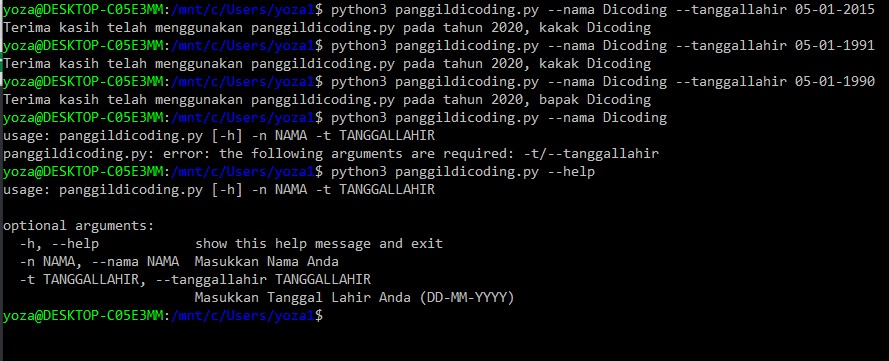
Jadi, pada saat dijalankan, ada beberapa hal yang menjadi perhatian seperti berikut.

* Berkas panggildicoding.py harus dipanggil dengan parameter -n atau --nama.
* Jika kita memanggil berkas tanpa parameter -n, berkas akan meminta parameter n atau nama.
* Jika kita memanggil dengan -n NAMAKITA atau --nama NAMAKITA, berkas akan menampilkan **Terima kasih telah menggunakan panggildicoding.py NAMAKITA.**
* Jika kita memanggil --help, tampil help dengan penjelasan "Masukkan Nama Anda".



Anda dapat menyuplai lebih dari satu argumen dengan menambahkan **parser.add\_argument**sejumlah yang Anda inginkan.

Ayo berlatih melengkapi script sebelumnya agar mampu menampilkan hasil berikut.



Ketentuan skrip sebagai berikut.

* 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, berkas akan menolak.

Jika usia pemanggil saat ini kurang dari 30 tahun, tambahkan panggilan kakak, selain itu gunakan panggilan bapak sebelum NAMAKITA.

**Library Pengolahan Data**

Library pengolahan data bertujuan untuk membantu dalam manipulasi, analisis, dan pemrosesan data. Library ini menyediakan berbagai fungsi dan metode yang memudahkan pengguna untuk melakukan operasi pengolahan data dengan lebih efisien dan cepat.

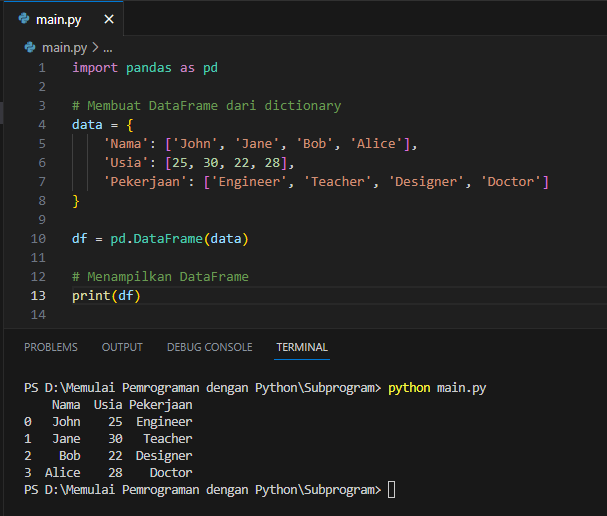
Tujuan dari library ini untuk menyederhanakan tugas-tugas kompleks yang berkaitan dengan pengolahan data sehingga Anda tidak perlu mengimplementasikan semuanya dari awal. Berikut adalah beberapa library populer yang digunakan untuk pengolahan data.

1. **Pandas**  
   [Pandas](https://pandas.pydata.org/) adalah library populer yang digunakan untuk pengelolaan dan analisis data. Library ini menyediakan struktur data dan alat untuk membantu pengguna dalam melakukan manipulasi, pembersihan, transformasi, dan analisis data dengan mudah dan efisien.  
     
   Meskipun pandas sudah terinstal secara otomatis pada beberapa IDE dan versi Python, perlu diingat bahwa pandas bukan merupakan library bawaan Python. Oleh karena itu, Anda harus menginstal library ini terlebih dahulu sebelum dapat menggunakannya. Silakan buka command prompt dan jalankan kode berikut.

|  |
| --- |
| pip install pandas |

Atau jika Anda ingin menggunakan conda, silakan jalankan kode berikut untuk menginstalnya.

|  |
| --- |
| conda install pandas |



Pada contoh di atas, kita membuat DataFrame dari dictionary dan menampilkannya ke layar. DataFrame merupakan struktur data utama dalam pandas yang mirip seperti tabel atau spreadsheet. DataFrame merupakan struktur dua dimensi yang menyimpan data dalam bentuk baris dan kolom.

1. NumPy  
   Library [NumPy](https://numpy.org/) adalah package fundamental yang sering digunakan untuk scientific computing pada Python. Library ini menyediakan objek array multidimensi, berbagai jenis objek lainnya, seperti masked array dan matrix, dan sebagainya.  
     
   NumPy termasuk library eksternal, meskipun NumPy juga sudah terinstal secara otomatis pada beberapa IDE dan versi Python, perlu diingat bahwa NumPy bukan termasuk library bawaan Python. Oleh karena itu, Anda harus menginstal library ini terlebih dahulu sebelum dapat menggunakannya.  
     
   Silakan buka command prompt dan jalankan kode berikut.

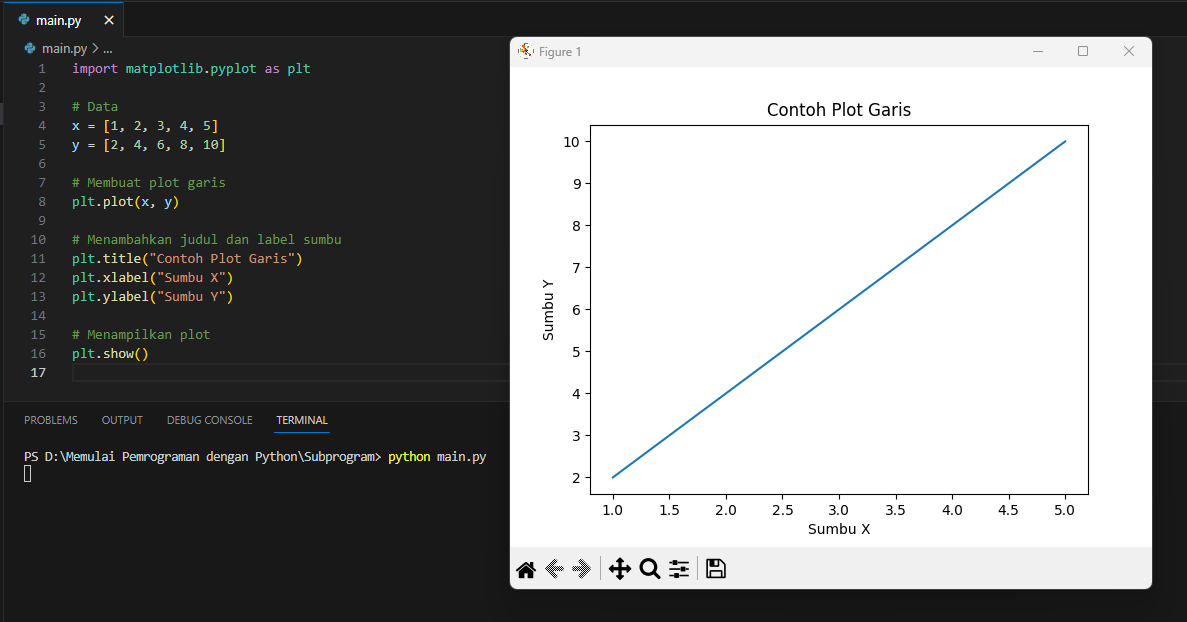
|  |
| --- |
| pip install numpy  conda install numpy  matriks = numpy.array([[1, 2, 3], [4, 5, 6], [7, 8 ,9]])  print(matriks)  """  Output:  [[1 2 3]  [4 5 6]  [7 8 9]]  """ |

Pada kode di atas, kita mengimpor library "numpy" terlebih dahulu untuk mengambil fungsi-fungsi atau kode yang berada pada library tersebut. Selanjutnya, mengubah nested list menjadi array dengan menggunakan fungsi “.array()”.

1. **Matplotlib**  
   Selanjutnya adalah [matplotlib](https://matplotlib.org/" \t "_blank)yang merupakan library untuk melakukan visualisasi data. Matplotlib termasuk jenis library eksternal sehingga Anda perlu melakukan instalasi matplotlib terlebih dahulu. Silakan jalankan kode berikut.

|  |
| --- |
| python -m pip install -U matplotlib  conda install matplotlib  import matplotlib.pyplot as plt    # Data  x = [1, 2, 3, 4, 5]  y = [2, 4, 6, 8, 10]    # Membuat plot garis  plt.plot(x, y)  # Menambahkan judul dan label sumbu  plt.title("Contoh Plot Garis")  plt.xlabel("Sumbu X")  plt.ylabel("Sumbu Y")    # Menampilkan plot  plt.show() |

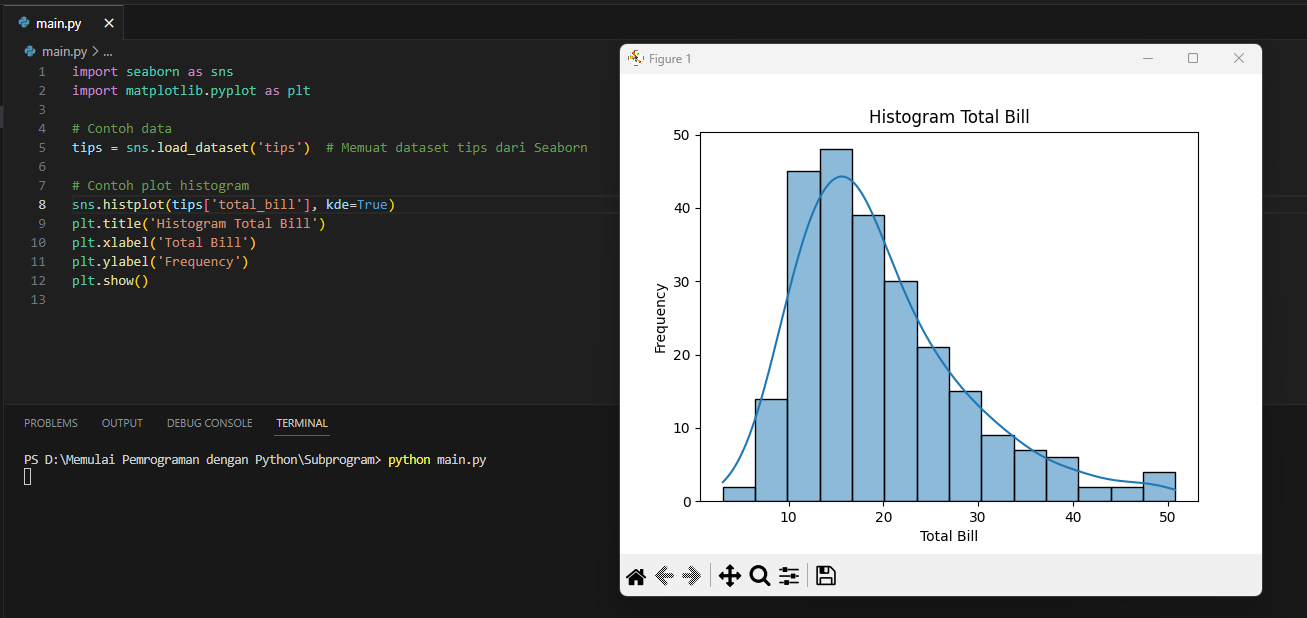
Pada kode di atas, kita akan membuat visualisasi berdasarkan data dari variabel x dan y. Hal pertama yang dilakukan adalah mengimpor library dengan menggunakan sintaks “import matplotlib.pyplot as plt".  
  
Selanjutnya, ini adalah contoh sehingga kita perlu membuat variabel sebagai data yang akan digunakan. Di sini kita membuat variabel x dan y sebagai data yang akan divisualisasi.  
  
Untuk membuat visualisasinya, kita menggunakan sintaks “plt.plot(x, y)” dengan argumennya adalah variabel x dan y. Lalu, kita menambahkan informasi tambahan seperti title, xlabel, dan ylabel. Terakhir, kita menampilkan visualisasi tersebut dengan sintaks “plt.show()”.  
  
Berikut adalah output ketika kode di atas dijalankan.



1. **Seaborn**  
   Terakhir adalah library [seaborn](https://seaborn.pydata.org/) yang termasuk jenis library dengan tujuan untuk visualisasi data sama seperti matplotlib. Bahkan library seaborn dibangun berdasarkan pada library matplotlib.  
     
   Seaborn termasuk library eksternal sehingga Anda perlu menginstalnya terlebih dahulu. Silakan jalankan kode berikut untuk menginstalnya menggunakan pip.

|  |
| --- |
| pip install seaborn  conda install seaborn  # Contoh data  tips = sns.load\_dataset('tips') # Memuat dataset tips dari Seaborn    # Contoh plot histogram  sns.histplot(tips['total\_bill'], kde=True)  plt.title('Histogram Total Bill')  plt.xlabel('Total Bill')  plt.ylabel('Frequency')  plt.show() |

Pada contoh di atas, kita menggunakan seaborn untuk melakukan visualisasi berdasarkan dataset tips. Dataset ini adalah bawaan dari library seaborn yang dapat Anda gunakan.  
  
Hal pertama yang dilakukan adalah mengimpor modul seaborn. Selanjutnya, kita load dataset dan menyimpannya pada variabel tips.  
  
Untuk membuat plot yang baik, di sini kita menggabungkan seaborn dan juga matplotlib. Library matplotlib digunakan untuk membuat title, xlabel, ylabel, dan menampilkannya ke layar.  
  
Untuk membuat plot histogram pada seaborn, Anda dapat menggunakan sintaks “sns.histplot()” dengan sns adalah library seaborn dan histplot merupakan fungsinya. Jangan lupa untuk mengisikan value dalam fungsi tersebut. Pada contoh di atas kita menggunakan kolom total\_bill yang ada dalam dataset tips.  
  
Berikut adalah tampilannya ketika kode tersebut dijalankan.



**Library File Management**

Library file management adalah kumpulan library yang dirancang untuk membantu pengguna dalam mengelola dan berinteraksi dengan berkas dan direktori pada sistem file. Beberapa library file management adalah berikut.

1. **OS**  
   Modul [OS](https://docs.python.org/3/library/os.html) pada Python berguna untuk fungsi-fungsi yang berkaitan dengan sistem operasi, misalnya open(), path(), getcwd(), dan fungsi lainnya. Modul ini memungkinkan Anda untuk memanfaatkan fungsi yang sama dan mengeksekusi fungsi terkait OS yang mungkin berbeda dalam setiap sistem operasi. Ada beberapa fitur yang hanya bekerja pada sistem operasi tertentu.  
     
   Contoh kode di bawah menunjukkan fungsi os.getcwd(). Fungsi ini akan mengembalikan string representasi dari Current Working Directory, yaitu direktori tempat program Python kita berada. Fungsi ini berlaku pada semua OS.

|  |
| --- |
| import os  print(os.getcwd()) |

1. **JSON**  
   Untuk serialization dengan bahasa lain, umumnya kita menggunakan [JSON](https://docs.python.org/3/library/json.html)(JavaScript Object Notation) yang memiliki beberapa perbedaan karakteristik dengan pickle, yakni berikut.

* 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.

Sebagaimana 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, yakni data disimpan dengan format key dan 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 teratur. Sebuah key seperti children di bawah dapat memiliki sebuah dictionary baru yang berisi informasi terkait objek children tersebut.



Untuk membuat JSON sederhana, ketik seperti kode di bawah.

|  |
| --- |
| import json    # contoh JSON:  x = '{ "nama":"Buchori", "umur":22, "Kota":"New York"}'    # parse x:  y = json.loads(x)    print(y["umur"]) |

1. **Pickle**  
   Jika Anda memiliki sebuah list yang ingin disimpan atau ditransmisikan tanpa khawatir bentuknya akan rusak atau kacau, fungsi dari library pickle dapat dimanfaatkan. Pickle termasuk fungsi Object Serialization pada Python. Pickling adalah istilah untuk mengubah objek menjadi byte stream, sedangkan unpickling adalah perlakuan sebaliknya.  
     
   Kode berikut adalah contoh cara melakukan proses pickle pada sebuah object dictionary dan menyimpannya pada sebuah file.

|  |
| --- |
| import pickle  contoh\_dictionary = {1:"6", 2:"2", 3:"f"}  pickle\_keluar = open("dict.pickle","wb")  pickle.dump(contoh\_dictionary, pickle\_keluar)  pickle\_keluar.close() |

Kode berikut adalah contoh untuk mengekstraksi berkas pickle dan menaruhnya pada sebuah variabel.

|  |
| --- |
| import pickle  pickle\_masuk = open("dict.pickle", "rb")  contohDictionary = pickle.load(pickle\_masuk)  pickle\_masuk.close()    print(contohDictionary) |

**Library Web Scraping**

Library web scraping adalah jenis library untuk membantu pengguna mengumpulkan data dari halaman web. Proses ini disebut sebagai “web scraping” atau “web crawling”. Anda bisa menggunakan fungsi dan metode pada library ini untuk mengekstraksi informasi dari situs web dan menyimpannya dalam format yang dapat diakses dan digunakan dalam analisis atau aplikasi lainnya.

Beberapa library untuk melakukan web scraping adalah berikut.

1. Beautifulsoup  
   Beautifulsoupadalah library untuk mengambil data dari halaman web dan mengekstrak informasi yang diperlukan. Untuk menggunakan beautifulsoup, Anda harus menginstalnya terlebih dahulu. Silakan jalankan kode pip berikut untuk menginstalnya.

|  |
| --- |
| pip install beautifulsoup4  from urllib.request import urlopen  from bs4 import BeautifulSoup    # Pengambilan konten  url = "http://python.org/"  page = urlopen(url)  html = page.read().decode("utf-8")    # Membuat objek BeautifulSoup  soup = BeautifulSoup(html, "html.parser")    # Mencetak judul halaman  print(soup.title) |

Pada contoh di atas, kita melakukan web scraping untuk mengambil judul dari laman web “http://python.org/”. Hal pertama yang dilakukan adalah mengimpor Beautifulsoup sebagai library yang akan kita gunakan. Selanjutnya kita mengambil konten dari url dengan menggunakan fungsi dari modul “urlopen”. Setelah konten diambil, kita membuat objek BeautifulSoup dan dari objek ini kita bisa memunculkan beberapa konten berdasarkan tag html. Pada contoh di atas, kita mengambil judul halaman dengan menggunakan method “title”.

1. **Urllib**  
   Urllibadalah library bawaan dari Python yang bertujuan untuk scraping konten dari sebuah website. Penggunaan urllib berbeda dengan beautifulsoup. Bisa dikatakan bahwa cara penggunaan urllib sedikit kompleks dibandingkan beautifulsoup. Kode di bawah adalah contoh untuk memulai proses scraping pada situs dengan domain **python.org**dan menampilkan isi dari tag title dari situs tersebut.

|  |
| --- |
| from urllib.request import urlopen    # Pengambilan konten  url = "http://python.org/"  page = urlopen(url)  html = page.read().decode("utf-8")    # Mencari indeks awal dan akhir  start\_index = html.find("<title>") + len("<title>")  end\_index = html.find("</title>")    # Mengekstrak dan mencetak judul halaman  title = html[start\_index:end\_index]  print(title) |

Pada kode di atas, kita melakukan scraping terhadap url yang sama seperti contoh sebelumnya. Namun, kali ini kita menggunakan urlopen untuk mengambil title dari laman “http://python/org/”

Hal pertama yang dilakukan adalah mengimpor urlopen dengan menggunakan “from urllib.request import urlopen”. Selanjutnya, kita mengambil konten dari url yang telah ditentukan.

Tahapan ketiga adalah kita mencari indeks awal dan akhir. Tujuan kita adalah mengambil title sehingga indeksnya ditentukan dari tag “<title>” dan “</title>”. Terakhir, kita mengekstrak dan mencetak judul halaman tersebut.

**Library Machine Learning**

Selanjutnya adalah library yang digunakan untuk melakukan pemelajaran mesin. Anda dapat menggunakan library berikut untuk membantu Anda menyelesaikan permasalahan machine learning. Berikut adalah beberapa library populer untuk machine learning.

1. **scikit-learn**Pertama adalah scikit-learn yang menyediakan berbagai algoritma pemelajaran mesin siap pakai untuk membantu dalam pengembangan model pemelajaran mesin, pemrosesan data, dan evaluasi kinerja model.  
     
   Scikit-learn termasuk library eksternal sehingga Anda perlu menginstalnya terlebih dahulu untuk bisa menggunakannya. Silakan jalankan kode berikut untuk menginstalnya menggunakan pip.

|  |
| --- |
| pip install -U scikit-learn  conda create -n sklearn-env -c conda-forge scikit-learn  conda activate sklearn-env |

Anda bisa merujuk pada tautan [berikut](https://scikit-learn.org/stable/auto_examples/index.html" \t "_blank) untuk melihat berbagai penerapan machine learning menggunakan scikit-learn.

1. **TensorFlow**  
   Selanjutnya adalah TensorFlow sebagai salah satu library paling populer terkait machine learning. Dengan menggunakan TensorFlow, Anda bisa mengembangkan machine learning hingga tahap deployment.  
     
   Anda bisa menginstal TensorFlow menggunakan perintah pip berikut.

|  |
| --- |
| pip install tensorflow |

Berikut adalah link dokumentasi yang bisa dijadikan rujukan untuk mempelajari machine learning menggunakan [tensorflow](https://www.tensorflow.org/tutorials).

1. **PyTorch**  
   Terakhir ada PyTorch, yakni library machine learning yang dikembangkan oleh Facebook’s AI Research lab (FAIR). PyTorch menyediakan alat dan kerangka kerja yang kuat untuk mengembangkan model pemelajaran mesin, terutama dalam konteks jaringan saraf tiruan (neural networks).  
     
   Anda bisa menggunakan kode berikut untuk menginstal PyTorch versi stabil menggunakan pip di Windows dan Mac.

|  |
| --- |
| pip install torch torchvision torchaudio  #Linux  pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu |

Anda bisa melihat dokumentasi instalasi lebih dalam pada tautan [ini](https://pytorch.org/get-started/locally/" \t "_blank). Selain itu, jika Anda ingin melihat berbagai contoh penerapannya, silakan merujuk pada tautan [berikut](https://pytorch.org/tutorials/" \t "_blank).

**Library Web Development**

Terakhir, ada library yang bertujuan untuk pengembangan aplikasi web. Sebagaimana yang sudah dijelaskan dalam materi-materi sebelumnya, Python dapat digunakan untuk pengembangan aplikasi web pada sisi server. Berikut adalah library yang dapat digunakan untuk membantu Anda mengembangkan web.

1. **Django**  
   Django adalah high-level Python web framework yang mendukung pengembangan secara cepat, bersih, serta pragmatis. Untuk menginstal Django, Anda bisa menggunakan pip berikut.

|  |
| --- |
| Linux/Mac:  python -m pip install Django  Windows:  py -m pip install Django |

Anda juga bisa merujuk pada dokumentasi [berikut](https://docs.djangoproject.com/en/4.2/topics/install/" \t "_blank) untuk melakukan instalasi. Untuk melihat penerapannya, Anda bisa merujuk pada dokumentasi [berikut](https://docs.djangoproject.com/en/4.2/" \t "_blank).

1. **Flask**  
   Flask adalah web framework dalam Python yang ditujukan untuk membangun aplikasi web. Flask dirancang dengan tujuan menjadi ringan, fleksibel, dan sederhana. Dengan Flask, Anda bisa merancang aplikasi web dari yang sederhana hingga kompleks.  
     
   Anda bisa menginstal Flask menggunakan kode pip berikut.

|  |
| --- |
| pip install Flask |

Untuk penerapan Flask, Anda dapat merujuk pada tautan [ini](https://flask.palletsprojects.com/en/2.3.x/tutorial/).

1. **Fast API**  
   FastAPI adalah web framework untuk Python yang tujuannya merancang dan membangun API dengan cepat, efisien, dan aman. FastAPI memberikan kinerja yang tinggi, sintaks yang intuitif, serta dukungan otomatisasi dokumentasi yang kuat. Jadi, ia cocok untuk pengembangan mikroservis, layanan web responsif, dan sebagainya.  
     
   Anda bisa menginstal FastAPI menggunakan kode pip berikut.

|  |
| --- |
| pip install fastapi |

Anda bisa merujuk pada tautan berikut untuk melihat berbagai penerapan FastAPI.