Modul Pelatihan

Tools Proyek Data Science

Thematic Academy

Digital Talent Scholarship

Tahun 2021

Tujuan Pembelajaran

A. Tujuan Umum

Setelah mempelajari modul ini peserta latih diharapkan mampu membuat program untuk menganalisis dan menginterpretasikan data menggunakan Python.

B. Tujuan Khusus

Adapun tujuan mempelajari unit kompetensi melalui modul Pelatihan Python for Data Science adalah untuk memfasilitasi peserta latih sehingga pada akhir pelatihan diharapkan memiliki kemampuan sebagai berikut:

1. Mampu menyelesaikan permasalahan menggunakan Python.
2. Mampu menganalisis dan menginterpretasikan data menggunakan library Python (NumPy, SciPy Pandas, Matplotlib, Seaborn, Scikit-learn).

Latar belakang

Unit kompetensi ini dinilai berdasarkan tingkat kemampuan dalam menyelesaikan permasalahan menggunakan Python. Adapun penilaian dilakukan dengan menggabungkan serangkaian metode untuk menilai kemampuan dan penerapan pengetahuan pendukung penting. Penilaian dilakukan dengan mengacu kepada Kriteria Unjuk Kerja (KUK) dan dilaksanakan di Tempat Uji Kompetensi (TUK), ruang simulasi atau workshop dengan cara:

1.1. Presentasi
1.2. Tes tertulis
1.3. Demonstrasi
1.4. Metode lain yang relevan.

Deskripsi Pelatihan

Tujuan utama dari modul pelatihan ini adalah untuk membahas data science tools dengan menjelaskan seperangkat kakas dan teknik yang berkaitan dengan keterampilan dasar dalam ilmu komputer, matematika, dan statistik untuk melakukan tugas-tugas yang umumnya terkait dengan data science.

Kompetensi Dasar

A. Mampu menyelesaikan permasalahan menggunakan Python.
B. Mampu menganalisis dan menginterpretasikan data menggunakan library Python (NumPy, SciPy Pandas, Matplotlib, Seaborn, Scikit-learn).

Indikator Hasil Belajar

Peserta mampu memahami dan menggunakan dasar tools yang akan dipergunakan untuk menangani data (NumPy, SciPy Pandas, Matplotlib, Seaborn, Scikit-learn).

INFORMASI PELATIHAN

Akademi	Thematic Academy
Mitra Pelatihan	Kementerian Komunikasi dan Informatika
Tema Pelatihan	**Data Scientist: Artificial Intelligence untuk Dosen dan Instruktur**
Sertifikasi	• *Certificate of Attainment;* • Sertifikat Kompetensi Associate Data Scientist
Persyaratan Sarana Peserta/spesifikasi device Tools/media ajar yang akan digunakan	Memiliki laptop/komputer dengan spesifikasi minimal : • RAM minimal 2 GB (disarankan 4 GB) • Laptop dengan 32/64-bit processor • Laptop dengan Operating System Windows 7, 8, 10, MacOS X atau Linux • Laptop dengan konektivitas WiFi dan memiliki Webcam • Akses Internet Dedicated 126 kbps per peserta per perangkat • Memiliki aplikasi Zoom • Memiliki akun Google Colab
Aplikasi yang akan digunakan selama pelatihan	• Google Colab • Jupyter notebook
Tim Penyusun	Rizal Dwi Prayogo, S.Si, M.Si, M.Sc (ITB)

INFORMASI PEMBELAJARAN

Unit Kompetensi	Materi pembelajaran	Kegiatan pembelajaran	Durasi Pelatihan	Rasio Praktek : Teori	Sumber pembelajaran
Mampu menggunakan Tools dalam proyek data science	Python untuk proyek data science	Daring/Online	Live Class 2 JP LMS 4 JP @ 45 menit	70:30	LMS

Materi Pokok
Tools Proyek Data Science

Sub Materi Pokok
• Pengantar Python • Library Python: NumPy, SciPy Pandas, Matplotlib, Seaborn, Scikit-learn • *Integrated Development Environment* (Spyder, PyCharm) • *Web Integrated Development Environment* (Jupyter Notebook, Google Colaboratory)

A. Materi Pelatihan

1. Pengantar Tools Proyek Data Science

Dalam modul ini, kita berkenalan dengan beberapa kakas yang digunakan *data scientist*. Kakas yang digunakan oleh *data scientist* mana pun, seperti halnya para *programmers*, adalah unsur penting untuk keberhasilan dan peningkatan kinerja. Sebagian besar *effort* dalam proyek data science digunakan untuk pemrosesan data. Memilih kakas yang tepat dapat menghemat banyak waktu dan dengan demikian memungkinkan kita untuk fokus lebih banyak pada analisis data. Hal mendasar yang perlu diputuskan adalah memilih bahasa pemrograman yang akan digunakan.

Beberapa orang hanya menggunakan satu bahasa pemrograman dalam tugas-tugas mereka, yang pertama dan satu-satunya yang mereka pelajari. Karena boleh jadi mempelajari bahasa pemrograman baru adalah tugas besar yang jika memungkinkan harus dilakukan hanya sekali. Masalah utama adalah adanya kemungkinan tidak tersedianya kakas tertentu dalam satu bahasa pemrograman. Sehingga pada akhirnya kita harus mengimplementasikannya kembali atau membuat koneksi untuk menggunakan beberapa bahasa lain hanya untuk tugas tertentu.

Jadi, kita harus siap untuk mengubah ke bahasa terbaik untuk setiap tugas dan kemudian mendapatkan hasilnya. Sebagai alternatif, kita bisa memilih bahasa yang sangat fleksibel dengan ekosistem yang kaya (misalnya, tersedianya library yang bersifat *open-source*). Dalam modul ini, kita akan menggunakan Python sebagai bahasa pemrograman.

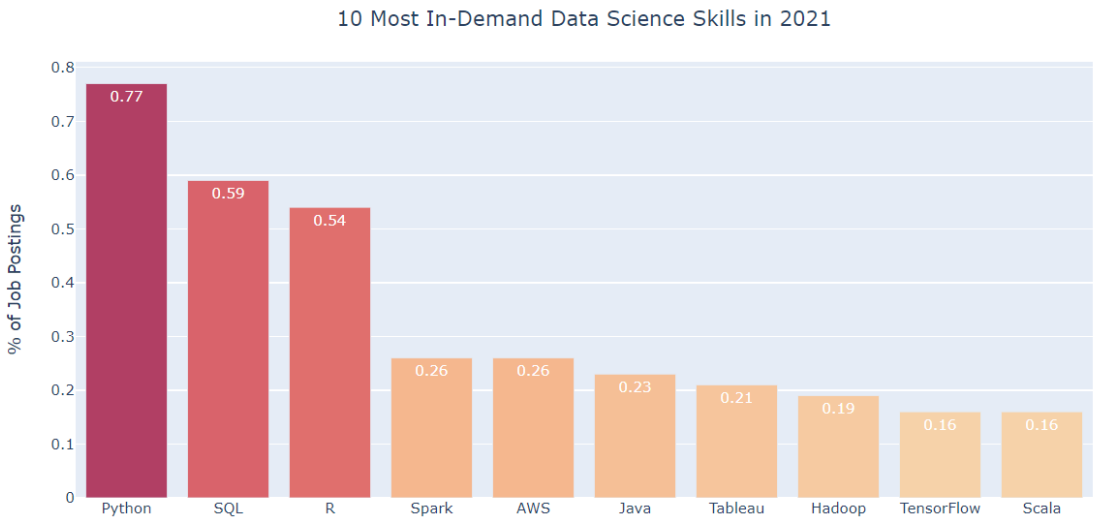
2. Mengapa Memilih Python?

Python[1] didesain oleh Guido van Rossum[2] pada 1991 sebagai *general-purpose language* dan bersifat *open-source*. Python merupakan bahasa pemrograman populer tetapi juga memiliki properti yang sangat baik untuk programmer pemula, sehingga ideal untuk orang yang belum pernah memprogram sebelumnya. Python juga bersifat *cross-platform*, artinya bisa digunakan pada sistem operasi Windows, Mac OS, dan Linux.

Beberapa kelebihan dari properti tersebut adalah kode yang mudah dibaca, pengetikan dan penggunaan memori yang dinamis. Python adalah bahasa interpreter, sehingga kode dieksekusi segera di konsol Python tanpa memerlukan langkah kompilasi ke bahasa mesin. Selain konsol Python (yang disertakan dengan instalasi Python apa pun), Anda

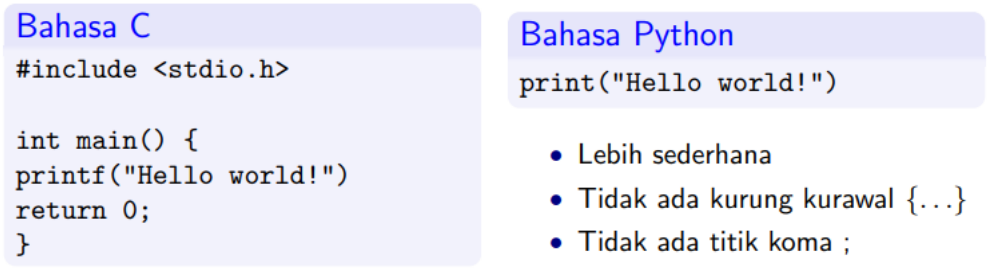
[1] https://www.python.org/downloads/
[2] https://gvanrossum.github.io//

dapat menemukan konsol interaktif lainnya, seperti IPython (*Interactive Python*)[3] dan Google Colab[4] yang memberi Anda lingkungan yang lebih interaktif untuk mengeksekusi kode Python Anda.



Gambar 1 10 Most In-Demand Data Science Skills in 2021 (https://towardsdatascience.com/)

Saat ini, Python adalah salah satu bahasa pemrograman yang paling fleksibel. Salah satu ciri utama yang membuatnya begitu fleksibel adalah dapat dilihat sebagai bahasa multiparadigma. Ini sangat berguna bagi orang yang sudah tahu cara memprogram dengan bahasa lain, karena mereka dapat dengan cepat memulai pemrograman dengan Python dengan cara yang sama. Misalnya, programmer Java akan merasa nyaman menggunakan Python karena mendukung paradigma pemrograman berorientasi objek, atau programmer C dapat mencampur kode Python dan C menggunakan cython.

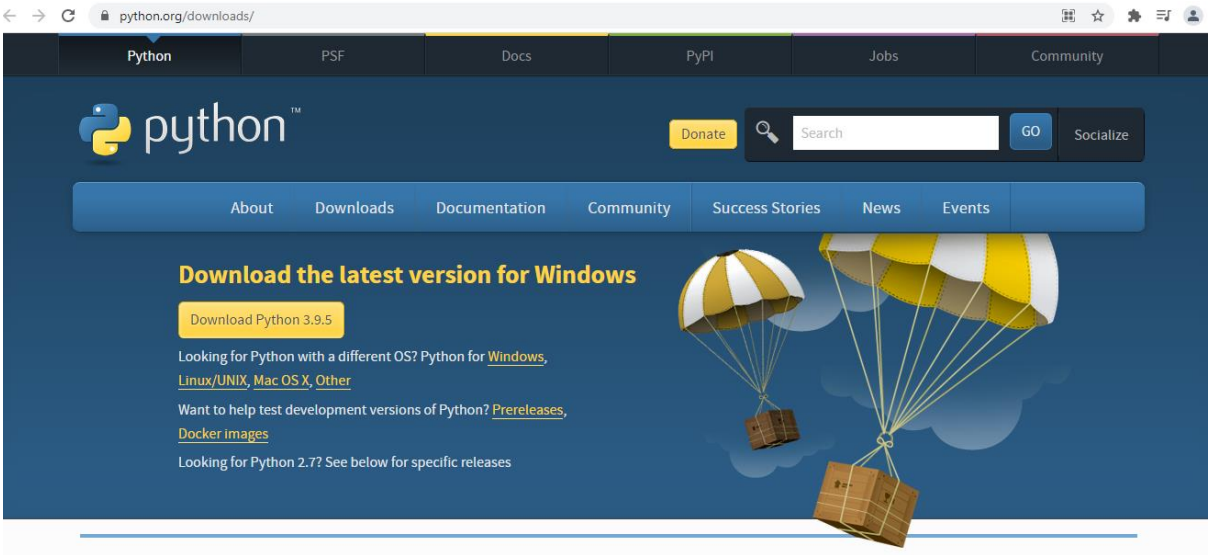


Gambar 2 Perbandingan Bahasa C dan Python

Selanjutnya, bagi siapa saja yang terbiasa memprogram dalam bahasa fungsional seperti Haskell atau Lisp, Python juga memiliki pernyataan dasar untuk pemrograman

[3] http://ipython.org/install.html
[4] https://colab.research.google.com/

fungsional di library intinya sendiri. Dalam modul ini, kami menggunakan bahasa Python karena, seperti yang dijelaskan sebelumnya, ini adalah bahasa pemrograman yang matang, mudah bagi pemula, dan dapat digunakan sebagai platform khusus untuk proyek data science. Hal ini juga didukung dengan ekosistem library yang lengkap dan komunitas pengguna yang besar. Alternatif populer lainnya untuk Python untuk data scientist adalah R dan MATLAB/Octave. Setelah tahun 2000-an, Python memiliki library khusus untuk data science, seperti Numpy, SciPy, Pandas, Scikit-Learn, dan Matplotlib.



Gambar 3 Website untuk mengunduh Python (Open-source dan Cross-platform)

Jika Anda memilih untuk melakukan instalasi secara lengkap, maka *Anaconda Python distribution*[5] adalah pilihan yang baik. Anaconda menyediakan integrasi semua library Python dan aplikasi yang diperlukan untuk proyek data science ke dalam satu direktori. Paket instalasi tersebut tidak hanya menyediakan library dan aplikasi seperti NumPy, Pandas, SciPy, Matplotlib, Scikit-learn, IPython, Spyder, dll, tetapi juga menyediakan library yang lebih spesifik untuk tugas terkait lainnya seperti visualisasi data, optimasi kode, dan pemrosesan data besar (*big data*).

3. Penerapan Python pada Proyek Data Science

Python banyak digunakan pada proyek data science untuk tugas-tugas yang terkait dengan pengolahan data, seperti eksplorasi data, pemrosesan data, pembersihan data, dan pemodelan data.

[5] https://www.anaconda.com/products/individual

3.1. Eksplorasi Data

Data dapat diperoleh dari sumber manapun, baik yang bersifat terstruktur maupun yang tidak terstruktur. Namun, data yang diperoleh tersebut tidak selalu memenuhi kebutuhan, maka perlu dilakukan eksplorasi lebih lanjut untuk bisa mendapatkan data yang diharapkan. Beberapa teknik yang dilakukan diantaranya adalah *scraping* dan *crawling*, yaitu mengeksplorasi data yang terdapat di media sosial atau *website*. Selain itu, eksplorasi data juga dapat dilakukan dengan proses *query* untuk mendapatkan data dari *data warehouse / storage*. Proses eksplorasi data ini dapat disebut juga dengan penambangan data (*data mining*).

3.2. Pemrosesan Data

Pemrosesan data adalah bagian yang paling banyak memakan waktu dalam proyek data science. Sesuai dengan prinsip *Garbage In Garbage Out*, maka ketika data masukannya belum diproses dengan baik, hasilnya pun tidak akan optimal. Data yang diperoleh tidak selalu sesuai dengan ekspektasi dan belum memadai untuk dijadikan sebagai data latih dalam membangun model.

Hal yang perlu dilakukan adalah melihat profil data tersebut, dengan melihat sebaran data (distribusi), statistika deskriptif, dan visualisasi data agar mudah melakukan penanganannya. Tidak semua komponen data bermanfaat untuk membangun model yang diharapkan, maka perlu dilakukan seleksi fitur untuk memilih fitur-fitur yang paling relevan. Visualisasi data dapat membantu untuk melihat profil data.

3.3. Pembersihan Data

Data yang diperoleh tidak selalu dapat langsung diproses, karena mengandung *noise* sehingga perlu dilakukan pembersihan. Data mentah mungkin saja mengandung nilai kosong (*missing values*), baik data numerik maupun data kategorial, sehingga perlu diisi dengan teknik-teknik statistika, seperti mencari nilai rata-rata (*mean*), nilai data tengah (*median*), atau nilai yang paling banyak muncul (*modus*).

Data yang diperoleh pun mungkin mengandung sampel baris yang terduplikasi, artinya ada kesamaan sampel data. Hal ini akan mengganggu proses pemodelan, sehingga data terduplikasi tersebut dapat dihapus. Selain itu, perlu diperiksa kekonsistenan profil dan format data. Apabila ada format data yang tidak sesuai,

maka dapat dilakukan *data formatting* untuk memastikan kekonsistenan data tersebut.

Data riil dapat mengandung pencilan (*outliers*) karena distribusi data yang tidak merata dan memiliki rentang nilai yang jauh. Masalah ini dapat diatasi dengan melakukan normalisasi data, yaitu melakukan penskalaan dengan membatasi rentang data pada nilai tertentu, misal [-1, 1] atau [0, 1].

3.4. **Pemodelan Data**

Data latih yang sudah diproses dengan baik akan diproses untuk membangun model menggunakan algoritma *machine learning*, seperti untuk tugas regresi, klasifikasi, dan klasterisasi. Tugas regresi digunakan untuk memprediksi data-data numerik, seperti memprediksi kenaikan harga barang/komoditas berdasarkan riwayat data-data di periode sebelumnya.

Tugas klasifikasi digunakan untuk memprediksi data-data kategorial, seperti mengklasifikasikan email termasuk *spam* atau bukan, mengklasifikasikan spesies binatang berdasarkan ciri-cirinya, atau mengklasifikasi berita *hoax* atau fakta. Tugas klasterisasi digunakan untuk mengelompokkan objek-objek data berdasarkan kemiripan atau ciri-ciri yang berdekatan, seperti mengklasterisasi kelompok pelanggan atau segmentasi.

4. Library Dasar Python untuk Data Science

Komunitas Python adalah salah satu komunitas pemrograman paling aktif dengan banyaknya pengembang library. Library Python yang paling populer untuk proyek data science adalah NumPy, SciPy, Pandas, Scikit-Learn, dan Matplotlib.

4.1. *NumPy and SciPy: Numeric and Scientifc Computation*

NumPy[6] adalah library untuk tugas *numerical computation* di Python dan termasuk library yang paling dasar untuk melakukan pengolahan data di Python, sehingga diajarkan di bagian paling awal dari modul Python untuk proyek data science. Library NumPy digunakan untuk mengolah data-data dengan karakteristik big data, yaitu *volume, variety, velocity,* dan *veracity*.

Volume menggambarkan ukuran data yang sangat besar, bahkan bisa melebihi kapasitas penyimpan (*storage*) fisik, sehingga harus disimpan pada komputasi awan

[6] https://numpy.org/

(*cloud computing*). *Variety* adalah bentuk data yang bervariasi atau format yang bermacam-macam dari representasi data. Jadi data dapat berupa dokumen, gambar, video, sound clips atau suara, angka-angka atau teks. Sebenarnya ketika data-data tersebut diproses, data tersebut tidak secara mentah-mentah dibaca sebagai video atau dibaca sebagai audio. Tetapi data sudah ditransformasi terlebih dahulu dalam proses *reading* atau pembacaan data menjadi *array* atau *matrix of number*. Array adalah suatu variabel yang dapat menyimpan lebih dari satu nilai dengan tipe data yang sama. Letak atau posisi dari elemen array ditunjukkan oleh suatu index.

Sebagai contoh, kita membuat *list* L = [1,2,3,4,5], ini sebenarnya adalah bentuk sederhana dari array yang berisi angka dari 1 sampai 5. Ketika kita melakukan proses pembacaan data, misalkan gambar, maka gambar berwarna tersebut sebenarnya terdiri dari 3 *plane* atau 3 potong gambar. Pertama adalah R (*Red plane*), G (*Green plane*), dan B (*Blue plane*). *Plane* ini masing-masing menggambarkan intensitas nilai dari seberapa merah atau seberapa hijau atau seberapa biru objek tersebut. Jadi, intensitas ini sebenarnya digambarkan dengan nilai rentang 0 sampai 255, dengan nilai 0 yang merepresentasikan warna hitam dan nilai 255 merepresentasikan warna putih. Nilai rentang tersebut bisa kita konversi ke dalam rentang 0 sampai 1 dengan normalisasi. Jadi, data gambar direpresentasikan ke dalam dua dimensi berisi piksel-piksel yang memiliki angka.

Bagaimana dengan data *audio* atau *speech* atau *sound clips*? Data *audio* direpresentasikan ke dalam satu dimensi yang bersifat *time-series*. Artinya data *audio* akan dicatat sebagai nilai intensitas tertentu mengikuti waktunya. Jadi pada intinya apapun format datanya maka data-data tersebut akan direpresentasikan dalam bentuk Array bilangan sebelum dianalisis atau diproses di Python. NumPy adalah *data operation* yang mencakup *data storage* dan *data manipulation*. NumPy dapat digunakan untuk manipulasi data, seperti transformasi bentuk, operasi matematika atau aljabar, dan membangkitkan bilangan acak. Sementara itu, SciPy[7] menyediakan kumpulan algoritma numerik, termasuk pemrosesan sinyal, optimasi, statistika, dan library Matplotlib untuk visualisasi data.

Untuk bisa menggunakan NumPy, langkah pertama adalah mengimpor library tersebut:

[7] https://www.scipy.org/scipylib/download.html

```
import numpy
```

Untuk bisa mengksplor konten dan dokumentasi dari library NumPy, Anda bisa menggunakan perintah

```
numpy.<TAB>
numpy?
```

4.2. *PANDAS: Python Data Analysis Library*

Pandas[8] menyediakan kakas struktur data berkinerja tinggi dan analisis data. Sama halnya dengan NumPy, Pandas juga bersifat *open-source* dan memiliki komunitas tersendiri. Fitur utama Pandas adalah objek DataFrame yang cepat dan efisien untuk manipulasi data dengan pengindeksan terintegrasi. Struktur DataFrame dapat dilihat sebagai *spreadsheet* yang menawarkan cara bekerja yang sangat fleksibel. Anda dapat dengan mudah mengubah dataset apa pun sesuai keinginan Anda, dengan mengubah bentuk, menambahkan atau menghapus kolom atau baris.

Pandas juga menyediakan fungsi berkinerja tinggi untuk menggabungkan (*aggregating, merging, joining*) kumpulan data. Pandas juga dapat digunakan untuk mengimpor dan mengekspor data dari berbagai format: *comma-separated value* (CSV), file teks, Microsoft Excel, database SQL, dan format HDF5. Dalam banyak kondisi, data yang Anda miliki dalam format-format tersebut ada kemungkinan tidak lengkap atau terstruktur sepenuhnya. Untuk menangani hal tersebut, Pandas menawarkan penanganan data hilang dan penyelarasan data. Selain itu, Pandas menyediakan antarmuka Matplotlib yang nyaman.

Anda juga mungkin bekerja dengan dataset yang memiliki terlalu banyak kolom atau fitur, tapi kolom tersebut tidak terlalu berguna untuk dianalisis. Maka, Pandas dapat digunakan untuk melakukan proses pemisahan data atau *splitting*. Pandas juga dapat digunakan untuk menganalisis data yang bersifat *time-series*, yaitu data yang perlu dianalisis secara berkala.

Untuk bisa menggunakan Pandas, langkah pertama adalah mengimpor library tersebut:

```
import pandas
```

[8] https://pandas.pydata.org/

Untuk bisa mengksplor konten dan dokumentasi dari library Pandas, Anda bisa menggunakan perintah

```
pandas.<TAB>
pandas?
```

4.3. MATPLOTLIB: Data Visualization

Matplotlib[9] digunakan untuk visualisasi data dua dimensi dengan menggunakan berbagai variasi bagan atau diagram, seperti histogram, diagram batang, diagram garis, diagram lingkaran, diagram garis, dan sebagainya. Library Matplotlib sudah dibuat di tahun 2002 oleh John Hunter sebagai bagian dari penelitian postdoctoral.

Matplotlib adalah python 2D *ploting library* yang memiliki banyak function untuk melakukan beberapa jenis plot gambar. Python juga memiliki library sejenis untuk visualisasi data, yaitu seaborn yang memiliki tampilan lebih estetis dari Matplotlib.

Salah satu fitur terpenting Matplotlib adalah kemampuannya yang baik dengan banyak sistem operasi dan *backend* grafis (lintas platform). Matplotlib mendukung beberapa jenis *backend* dan *output*, yang berarti Anda dapat mengandalkannya untuk bekerja, terlepas dari sistem operasi yang Anda gunakan atau format *output* yang Anda inginkan. Sehingga telah menghasilkan basis pengguna yang besar. Setelah itu, Matplotlib berkembang secara mandiri oleh komunitas Matplotlib.

4.4. *SCIKIT-Learn: Machine Learning in Python*

Scikit-Learn[10] adalah library Python yang menyediakan kakas sederhana dan efisien untuk menerapkan tugas analisis data dan pembelajaran mesin, seperti klasifikasi, regresi, klasterisasi, reduksi dimensi, pemilihan model, dan pemrosesan. Algoritma pembelajaran mesin yang dapat diterapkan menggunakan Scikit-Learn diantaranya *Support Vector Machine, Decision Tree, Random Forest, K-Means Clutering,* dan *Neural Network*.

Library ini dapat diakses oleh semua orang karena bersifat *open-source* dan dapat digunakan secara komersial. Scikit-Learn dibangun dari library NumPy, SciPy, dan Matplolib. Scikit-Learn berfokus pada tugas pemodelan data daripada tugas manipulasi dan visualisasi data.

---

[9] https://matplotlib.org/
[10] https://scikit-learn.org/

5. *Integrated Development Environments* (IDE)

Untuk setiap *programmer*, dan *data scientist*, *Integrated Development Environments* (IDE) adalah kakas penting. IDE dirancang untuk memaksimalkan produktivitas *programmer*. Jadi, selama bertahun-tahun perangkat lunak ini telah berkembang untuk membuat tugas pemrograman menjadi tidak terlalu rumit. Memilih IDE yang tepat untuk setiap orang sangat penting dan, sayangnya, tidak ada IDE terbaik untuk semua tugas perograman. Hal ini bisa sangat bervariasi untuk setiap orang dan tugas pemrograman tertentu. Maka, solusi terbaik adalah mencoba beberapa IDE paling populer dan memilih mana yang lebih cocok untuk setiap kasus.

Secara umum, dasar dari setiap IDE ada tiga: *editor, compiler*, (atau *interpreter*) dan *debugger*. Beberapa IDE dapat digunakan dalam beberapa bahasa pemrograman yang disediakan oleh *plug-in* khusus bahasa, seperti Netbeans[11] atau Eclipse[12]. Sementara yang lainnya hanya khusus untuk satu bahasa atau bahkan tugas pemrograman tertentu.

Dalam Python, ada sejumlah besar IDE spesifik, baik komersial (PyCharm[13], WingIDE[14] ...) dan *open-source*. Komunitas *open-source* membantu IDE untuk berkembang, sehingga siapa pun dapat menyesuaikan dan membaginya dengan komunitas lainnya. Misalnya, Spyder[15] (*Scientific Python Development EnviRonment*) adalah IDE yang disesuaikan dengan proyek data science.

6. *Web Integrated Development Environment* (WIDE)

6.1. Jupyter Notebook

Dengan munculnya aplikasi web, generasi baru IDE untuk bahasa interaktif seperti Python telah dikembangkan. Dimulai di komunitas akademisi dan e-learning, IDE berbasis web dikembangkan dengan mempertimbangkan agar kode dan seluruh *environment* Anda dapat disimpan di server. Salah satu aplikasi pertama dari jenis WIDE ini dikembangkan oleh William Stein pada awal 2005 menggunakan Python 2.3 sebagai bagian dari perangkat lunak matematika SageMath. Di SageMath, server dapat diatur di pusat, seperti universitas atau sekolah, dan kemudian siswa dapat mengerjakan pekerjaan rumah mereka baik di kelas maupun di rumah. Selain itu,

[11] https://netbeans.org/downloads/
[12] https://eclipse.org/downloads/
[13] https://www.jetbrains.com/pycharm/
[14] https://wingware.com/
[15] https://www.spyder-ide.org/

siswa dapat menjalankan semua langkah sebelumnya berulang kali, dan kemudian mengubah beberapa sel kode tertentu (segmen dokumen yang mungkin berisi kode sumber yang dapat dieksekusi) dan menjalankan operasi lagi. Instruktur juga dapat memiliki akses ke sesi siswa dan meninjau kemajuan atau hasil mereka.

Saat ini, sesi semacam itu disebut *notebook* dan tidak hanya digunakan di ruang kelas tetapi juga digunakan untuk menunjukkan hasil dalam presentasi atau di dasbor bisnis. Penggunaan *notebook* adalah bagian dari pengembangan IPython. Sejak Desember 2011, IPython telah dirilis sebagai versi browser dari konsol interaktifnya, yang disebut *notebook IPython*, yang menunjukkan hasil eksekusi Python dengan sangat jelas dan ringkas melalui sel baris. Sel dapat berisi konten selain kode. Misalnya, sel *markdown* yang dapat ditambahkan pada sel untuk menjelaskan algoritma yang dibuat. Dimungkinkan juga untuk menyisipkan gambar plot Matplotlib untuk mengilustrasikan contoh atau bahkan halaman web. Belakangan ini, beberapa jurnal ilmiah mulai menerima *notebook* untuk menunjukkan hasil eksperimen, dilengkapi dengan kode dan sumber datanya. Dengan cara ini, eksperimen menjadi lengkap dan dapat direplikasi.

Sejak proyek data science berkembang pesat, *notebook* IPython telah dipisahkan dari perangkat lunak IPython dan sekarang telah menjadi bagian dari proyek yang lebih besar: Jupyter[16]. Jupyter (untuk Julia, Python dan R) bertujuan untuk menggunakan kembali WIDE yang sama untuk semua bahasa interpreter ini dan tidak hanya Python. Semua *notebook* IPython lama diimpor secara otomatis ke versi baru saat dibuka dengan platform Jupyter; tetapi setelah dikonversi ke versi baru, *notebook* tidak dapat digunakan lagi di versi notebook IPython lama.

6.2. Google Colaboratory

Sejalan dengan perkembangan Python, Google adalah salah satu perusahaan yang sudah menerapkan bahasa Python melalui *Google Colaboratory*[17] atau *Google Interactive Notebook* (disingkat Google Colab). Google Colab merupakan satu satu produk berbasis komputasi awan (*cloud*) yang dapat digunakan secara gratis. Google Colab dibuat khusus untuk *programmer* atau peneliti yang membutuhkan akses

[16] https://jupyter.org/
[17] https://colab.research.google.com/

dengan spesifikasi tinggi. Google Colab memiliki *coding environment* Python dengan format yang mirip dengan Jupyter notebook.

Google Colab digunakan untuk berkolaborasi dengan pengguna lainnya karena berbagi *coding* secara daring. Kita bisa lebih mudah bereksperimen secara bersamaan dengan fitur yang fleksibel. Kita dapat dengan mudah menghubungkan Google Colab dengan Jupyter notebook di komputer kita (*local runtime*), menghubungkan dengan *Google Drive*, atau dengan Github.

Google Colab memungkinkan kita menggabungkan kode yang dapat dijalankan dan *rich text* (fitur *markdown*) dalam satu tugas, beserta gambar, HTML, LaTeX, dan lainnya. Saat kita membuat notebook Colab, notebook tersebut akan disimpan di akun *Google Drive* dan dapat dengan mudah membagikan notebook Colab untuk kolaborasi dalam proyek data science.

B. Keterampilan yang diperlukan dalam mengimplementasikan tools proyek data science

1. Mampu mengidentifikasi permasalahan yang ada pada data dan memilih metode yang tepat untuk menyelesaikannya.
2. Mampu melakukan pemrosesan data dan pemodelan data dengan baik.

C. Sikap yang diperlukan dalam mengimplementasikan tools proyek data science

1. Disiplin
2. Rasa ingin tahu yang tinggi
3. Bertanggung jawab
4. Kerja sama dalam tim

Tugas Dan Proyek Pelatihan
1. Mengerjakan *hands-on* pada salah satu *online course – Python for Data Science*

Link Referensi Modul Keempat
1. Python for Data Science: https://cognitiveclass.ai/courses/python-for-data-science 2. Introduction to Python for Data Science: https://www.datacamp.com/courses/intro-to-python-for-data-science 3. Build real-world applications with Python: https://docs.microsoft.com/en-us/learn/paths/python-language/ 4. Introduction to Python for Data Science: https://learning.edx.org/course/course-v1:Microsoft+DAT208x+3T2018/home 5. Data Visualization with Python: https://cognitiveclass.ai/courses/data-visualization-with-python

Link Pertanyaan Modul Keempat

Bahan Tayang
Power Point

Link room Pelatihan dan Jadwal live sesi bersama instruktur
Zoom

Penilaian
Komposisi penilaian Tugas Tools Proyek Data Science: Nilai 100

Target Penyelesaian Modul Keempat
1 hari / sampai 6 JP

KOMINFO

Badan Penelitian dan Pengembangan SDM
Kementerian Komunikasi dan Informatika