

# Mengenal *Machine Learning* Dengan Teknik *Supervised* dan *Unsupervised Learning* Menggunakan *Python*

Endang Retnoningsih <sup>1\*</sup>, Rully Pramudita <sup>2</sup>

<sup>1</sup> Sistem Informasi; Universitas Bina Insani; Jl. Siliwangi No.6 Rawa Panjang Bekasi Bekasi Timur 17114 Indonesia, Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: [endang.retnoningsih@binainsani.ac.id](mailto:endang.retnoningsih@binainsani.ac.id)

<sup>2</sup> Manajemen Informatika; Universitas Bina Insani; Jl. Siliwangi No.6 Rawa Panjang Bekasi Bekasi Timur 17114 Indonesia, Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: [rullypramudita@binainsani.ac.id](mailto:rullypramudita@binainsani.ac.id)

Korespondensi: email: [endang.retnoningsih@binainsani.ac.id](mailto:endang.retnoningsih@binainsani.ac.id)

Diterima: 19 Nopember 2020; Review: 03 Desember 2020; Disetujui: 17 Desember 2020

Cara sitasi: Retnoningsih E, Pramudita R. 2020. Mengenal *Machine Learning* Dengan Teknik *Supervised* dan *Unsupervised Learning* Menggunakan *Python*. Bina Insani ICT Journal. 7 (2): 156-165.

---

**Abstrak:** *Machine learning* merupakan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulang kali diprogram oleh manusia sehingga komputer menjadi semakin cerdas belajar dari pengalaman data yang dimiliki. Berdasarkan teknik pembelajarannya, dapat dibedakan *supervised learning* menggunakan *dataset (data training)* yang sudah berlabel, sedangkan *unsupervised learning* menarik kesimpulan berdasarkan *dataset*. Input berupa *dataset* digunakan pembelajaran mesin untuk menghasilkan analisis yang benar. Permasalahan yang akan diselesaikan bunga iris (*iris tectorum*) yang memiliki bunga bermacam-macam warna dan memiliki sepal dan petal yang menunjukkan spesies bunga, dibutuhkan metode yang tepat untuk pengelompokan bunga-bunga tersebut kedalam spesiesnya iris-setosa, iris-versicolor atau iris-virginica. Penyelesaian digunakan *Python* yang menyediakan algoritma dan *library* yang digunakan membuat *machine learning*. Penyelesaian dengan teknik *supervised learning* dipilih algoritma KNN *Clasiffier* dan teknik *unsupervised learning* dipilih algoritma DBSCAN *Clustering*. Hasil yang diperoleh *Python* menyediakan *library* yang lengkap *numPy*, *Pandas*, *matplotlib*, *sklearn* untuk membuat pemrograman *machine learning* dengan algoritma KNN memanggil `from sklearn import neighbors` termasuk teknik *supervised*, maupun DBSCAN memanggil `from sklearn.cluster import DBSCAN` termasuk teknik *unsupervised learning*. Kemampuan *Python* memberikan hasil output sesuai input dalam *dataset* menghasilkan keputusan berupa klasifikasi maupun klusterisasi.

**Kata kunci:** DBSCAN, KNN, *machine learning*, *python*.

**Abstract:** *Machine learning* is a system that is able to learn on its own to decide something without having to be repeatedly programmed by humans so that computers become smarter in learning from the experience of the data they have. Based on the learning technique, supervised learning can be distinguished using a dataset (training data) that is already labeled, while unsupervised learning draws conclusions based on the dataset. The input in the form of a dataset is used by machine learning to produce the correct analysis. The problem to be solved by iris flowers (*iris tectorum*), which has flowers of various colors and has sepals and petals that indicate the species of flowers, requires an appropriate method for grouping these flowers into iris-setosa, iris-versicolor or iris-virginica species. The solution is used by Python, which provides the algorithms and libraries used to make machine learning. The solution with the supervised learning technique was chosen by the KNN *Clasiffier* algorithm and the unsupervised learning technique was selected by the DBSCAN *Clustering* algorithm. The results obtained by Python provide a complete library of numPy, Pandas, matplotlib, sklearn to

create machine learning programming with KNN algorithms calling from sklearn import neighbors including supervised techniques, and DBSCAN calling from sklearn.cluster import DBSCAN including unsupervised learning techniques. Python's ability to provide output according to the input in the dataset results in decisions in the form of classification and clustering.

**Keywords:** DBSCAN, KNN, machine learning, python.

## 1. Pendahuluan

*Machine learning* merupakan cabang ilmu bagian dari kecerdasan buatan (*artificial intelligence*), dengan pemrograman untuk memungkinkan komputer menjadi cerdas berperilaku seperti manusia, dan dapat meningkatkan pemahamannya melalui pengalaman secara otomatis [1]. *Machine learning* memiliki fokus pada pengembangan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulang kali diprogram oleh manusia. Hal ini menjadikan mesin tidak hanya mampu berperilaku mengambil keputusan, namun juga dapat beradaptasi dengan perubahan yang terjadi. *Machine learning* bekerja apabila tersedia data sebagai input untuk dilakukan analisis terhadap kumpulan data besar (*big data*) sehingga menemukan pola tertentu. Data merupakan bahan input yang akan digunakan untuk melakukan pembelajaran (*training*) sehingga mesin dapat menghasilkan analisis yang benar. Didalam *machine learning* dikenal data training dan data testing, data *training* untuk melatih algoritma dalam *machine learning* dan data *testing* untuk mengetahui performa dari algoritma dalam *machine learning* yang telah dilatih yaitu ketika menemukan data baru yang belum pernah diberikan dalam data training [2].

*Machine learning* menggunakan teknik untuk menangani data besar (*big data*) dengan cara yang cerdas untuk memberikan hasil yang tepat. Berdasarkan teknik pembelajarannya, tipe-tipe *machine learning* dapat dibedakan menjadi *supervised learning*, *unsupervised learning*, *semi supervised learning* dan *reinforcement learning*. *Supervised learning* merupakan salah satu teknik *machine learning* yang menggunakan *dataset* (*data training*) yang sudah berlabel (*labeled data*) untuk melakukan pembelajaran pada mesin, sehingga mesin mampu mengidentifikasi label input dengan menggunakan fitur yang dimiliki untuk selanjutnya melakukan prediksi maupun klasifikasi, sedangkan *unsupervised learning* adalah teknik dengan menarik kesimpulan berdasarkan *dataset* yang merupakan input data *labeled response* [3].

Algoritma yang termasuk kedalam teknik *supervised learning* diantaranya *Decision Tree*, *K-Nearest Neighbor (KNN)*, *Naive Bayes*, *Regresi*, dan *Super Vector Machine* [4]. Klasifikasi menggunakan KNN memiliki karakteristik mempertimbangkan jumlah tetangga terdekat, akan menghasilkan nilai k dengan akurasi tertinggi untuk menggeneralisasi data dilakukan observasi terhadap k sehingga dihasilkan nilai k optimum [5]. Beberapa Algoritma dalam *unsupervised learning* diantaranya *DBSCAN*, *Fuzzy C-Means*, *K-Means*, dan *Self Organizing Map*. *DBSCAN* pengelompokan berdasarkan kepadatan (*density*) data, konsep kepadatan menghasilkan status dari data yaitu *core* (inti), *border* (batas), dan *noise* [6].

Salah satu fungsi dari *machine learning* menjadikan komputer berperilaku lebih cerdas, dengan cara menggeneralisasi secara otomatis berdasarkan pengalaman masa lalu dalam pengklasifikasi yang baik, klasifikasi merupakan model untuk memprediksi suatu nilai yang tidak diketahui [7]. Model klasifikasi dibagi menjadi dua yaitu a) Melatih data kemudian membedakan kedalam *class*; b) Melakukan evaluasi terhadap objek tidak dikenal [8]. Klasifikasi sebagai bagian *unsupervised learning*, memerlukan persiapan data yang baik dan pemilihan bahasa pemrograman [9] yang mampu mendukung *machine learning*.

Bahasa pemrograman *Python* dipilih dalam penelitian, karena memiliki keunggulan seperti readability, efisien, multifungsi, interoperabilitas, dan memiliki dukungan komunitas yang memadai [10]. *Readability*, dengan *Python* memiliki *source code* yang sederhana, sehingga mudah ditulis, mudah diingat dan juga digunakan ulang. Hal ini memudahkan pengembangan aplikasi, mulai dari tahap *coding*, *testing*, perbaikan jika ada kesalahan, *bug* atau *error* lainnya. Efisien, dengan menggunakan *Python* memiliki *library* yang lengkap, *Python code* akan lebih sederhana bila dibandingkan dengan kode yang ditulis dengan bahasa pemrograman lainnya seperti misalnya Java, C, C# maupun C++. Multifungsi, dengan menggunakan *Python* dapat membuat website, aplikasi jaringan, aplikasi bidang robotika, sampai dengan aplikasi kecerdasan buatan. *Python* memiliki banyak modul yang siap digunakan untuk pengembangan aplikasi sesuai kebutuhan. Interoperabilitas, dengan menggunakan *Python* memiliki

interoperabilitas yang tinggi karena mampu berinteraksi dengan bahasa-bahasa pemrograman lain. Program yang dikembangkan *Python* dapat diperasionalisasi pada hampir semua sistem operasi baik *Windows*, *Linux*, *Mac OS*, *Unix*, dan juga sistem operasi pada perangkat lunak berbasis *mobile* seperti *Android* atau *IOS*. Dukungan Komunitas, dengan menggunakan *Python* memiliki dukungan komunitas yang sangat kuat, karena *Python* bersifat *opensource*. Dengan komunitas yang baik, mempermudah pengguna untuk saling berbagi, dan mengembangkan bahasa pemrograman *Python* menjadi bahasa yang handal.

Untuk melakukan instalasi *Python* dapat mengunduh *installer* pada situs resmi <https://www.python.org/>, pada penelitian ini digunakan *Python* versi 3.9. Tabel 1 menunjukkan perkembangan versi *Python* [11]:

Tabel 1. Perkembangan *Python*

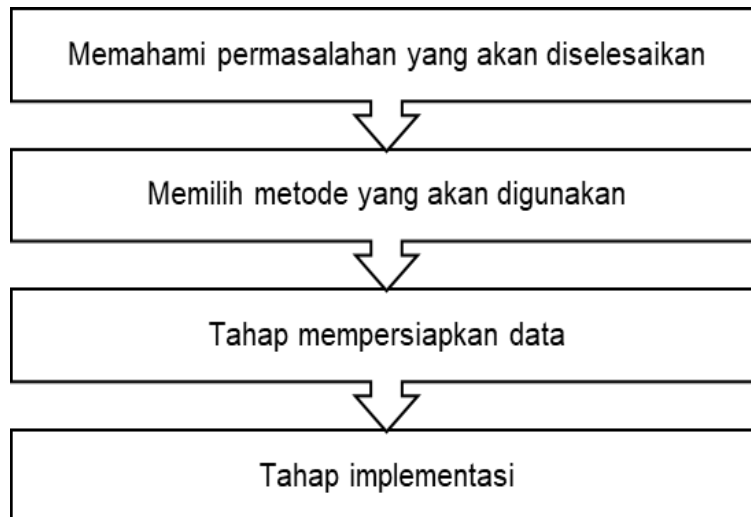
Version	Released
Python 1.4	25 October 1996
Python 1.5	17 February 1998
Python 1.6	5 September 2000
Python 2.0	16 October 2000
Python 2.1	15 April 2001
Python 2.2	21 December 2001
Python 2.3	29 July 2003
Python 2.4	30 November 2004
Python 2.5	19 September 2006
Python 2.6	1 October 2008
Python 2.7	4 July 2010
Python 3.0	3 December 2008
Python 3.1	27 June 2009
Python 3.2	20 February 2011
Python 3.3	29 September 2012
Python 3.4	16 March 2014
Python 3.5	13 September 2015
Python 3.6	23 December 2016
Python 3.7	27 June 2018
Python 3.8	14 October 2019
Python 3.9	5 October 2020

Sumber: <https://www.python.org/> (2020)

Instalasi *Python* dapat juga dengan menggunakan paket distribusi seperti *Anaconda* (*Anaconda Distribution*) dibuat oleh *continuum* dapat digunakan secara gratis, memiliki banyak *package* termasuk *Python* didalamnya. Dengan *Anaconda* akan lebih mudah melakukan instalasi didalamnya sudah berisi berbagai *tools* untuk menjalankan *Python* seperti *Jupyter Notebook*. Unduh installer *Anaconda* dengan mengunjungi website <https://repo.anaconda.com/> pada penelitian ini digunakan installer *Anaconda* 3 versi 4.2.0, yang didalamnya terdapat *Jupyter Notebook* sebagai tool antar muka untuk membuat *machine learning*.

## 2. Metode Penelitian

Langkah-langkah membuat *machine learning* [10] ditunjukkan pada gambar 1, berikut penjelasan untuk tahapan: 1) Memahami permasalahan yang akan diselesaikan, hal-hal yang harus dipahami pada tahap ini antara lain latar belakang masalah dan juga tujuan (goal) dari pemecahan masalah yang akan dilakukan; 2) Memilih metode yang akan digunakan, berdasarkan pada permasalahan yang telah dipahami, selanjutnya dipilih metode untuk memecahkan permasalahan, misal jika masalah klasifikasi maka metode yang dipilih klasifikasi; 3) Tahap mempersiapkan data, memahami data berarti mengetahui informasi tentang suatu data secara lengkap, termasuk variabel dan fitur-fitur yang ada pada data, hal ini akan membantu dalam mempersiapkan data yang akan diolah 4) Tahap Implementasi, melakukan pemrograman untuk mendapatkan output sesuai dengan tujuan yang diinginkan, dari hasil output ini kita juga harus dapat membaca dan memaknai hasilnya, apakah telah sesuai dengan data yang diolah dan sesuai dengan hasil yang diharapkan untuk menyelesaikan masalah yang dihadapi, dalam penelitian ini diimplementasikan menggunakan bahasa pemrograman *Python*.



Sumber: Wahyono (2018)

Gambar 1. Langkah Membuat *Machine Learning*

Mengenal *machine learning* menggunakan *Python* pada penelitian ini dengan teknik *supervised learning* dipilih algoritma *K-Nearest Neighbor (KNN)* yang biasanya untuk menyelesaikan masalah klasifikasi, sedangkan teknik *unsupervised learning* dipilih algoritma *Density Based Spatial Clustering of Application with Noise (DBSCAN)* yang biasanya untuk menyelesaikan masalah *Clusterisasi*. Algoritma *KNN* merupakan salah satu dari teknik *supervised learning*, *KNN* akan menghitung kedekatan antara kasus lama (*data training*) dengan kasus baru (*data testing*). Langkah-langkah dari algoritma *KNN* [13] adalah a) Menentukan parameter *k*, sebagai *nearest neighbors* (banyaknya tetangga terdekat); b) Menghitung kuadrat jarak antara data baru dengan data *training* menggunakan *eucliden*; c) Mengurutkan tetangga terdekat yang mempunyai jarak minimum; d) Mengklasifikasikan tetangga terdekat sesuai nilai *k*; e) Menentukan hasil klasifikasi berdasarkan label mayoritas. Algoritma *DBSCAN* merupakan salah satu dari teknik *unsupervised learning*, proses pengelompokan *DBSCAN* berdasarkan kepadatan (*density*) data [6].

*Dataset* yang digunakan dalam penelitian diambil dari repository *public dataset* oleh *UCI Repository* <http://archive.ics.uci.edu/ml/index.php> [14]. *Dataset* bunga iris dikumpulkan oleh ahli botani Anderson berisi sample acak bunga milik tiga spesies bunga iris yaitu *setosa*, *versicolor*, dan *virginica*. Bunga Iris (*iris tectorum*) salah satu bunga hias yang populer dan berasal dari Jepang. Masing-masing spesies bunga iris memiliki panjang sepal, lebar sepal, panjang petal dan lebar petal dicatat. Sepal dan Petal adalah bagian dari perhiasan bunga iris. Sepal disebut juga daun kelopak, merupakan daun perhiasan bunga yang paling pangkal, berkelompok membentuk kelopak bunga (*calyx*). *Dataset* bunga iris sebagaimana pada tabel 2.

Tabel 2. *Dataset Iris Tectorum*

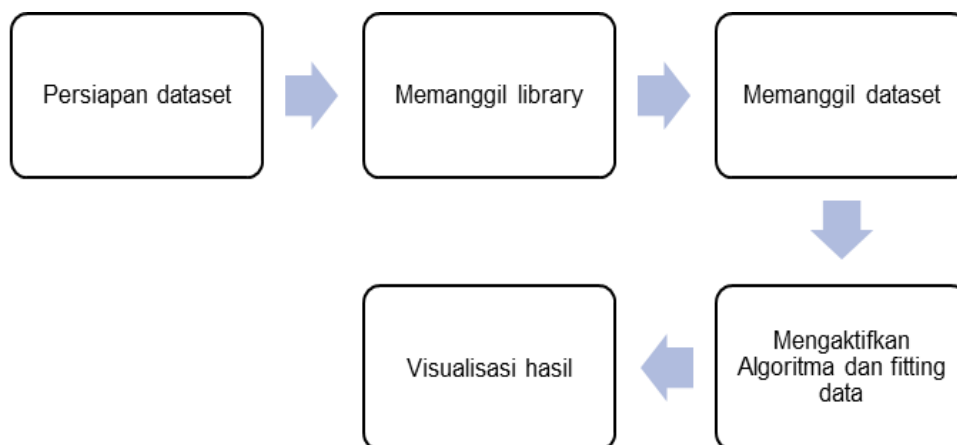
Sepal		Petal		Class
Length	Width	Length	Width	
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
.....	.....	.....	.....	.....
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor

Sepal		Petal		Class
Length	Width	Length	Width	
5.5	2.3	4	1.3	Iris-versicolor
6.5	2.8	4.6	1.5	Iris-versicolor
5.7	2.8	4.5	1.3	Iris-versicolor
6.3	3.3	4.7	1.6	Iris-versicolor
4.9	2.4	3.3	1	Iris-versicolor
6.6	2.9	4.6	1.3	Iris-versicolor
5.2	2.7	3.9	1.4	Iris-versicolor
.....	.....	.....	.....	.....
6.3	3.3	6	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica
7.1	3	5.9	2.1	Iris-virginica
6.3	2.9	5.6	1.8	Iris-virginica
6.5	3	5.8	2.2	Iris-virginica
7.6	3	6.6	2.1	Iris-virginica
4.9	2.5	4.5	1.7	Iris-virginica
7.3	2.9	6.3	1.8	Iris-virginica
6.7	2.5	5.8	1.8	Iris-virginica
7.2	3.6	6.1	2.5	Iris-virginica

Sumber: <http://archive.ics.uci.edu/ml/index.php> (2020)

### 3. Hasil dan Pembahasan

Pembuatan *machine learning* menggunakan *Python*, dilakukan sesuai langkah-langkah pada gambar 1 dimulai dari memahami permasalahan yang akan diselesaikan, permasalahan yang akan diselesaikan adalah Bunga Iris memiliki bunga bermacam-macam warnanya. Ada yang berwarna biru-keunguan, kuning, merah-muda, dan lain sebagainya, maka dibutuhkan metode yang tepat untuk pengelompokan bunga-bunga tersebut sesuai spesiesnya. Tujuan dari penelitian ini adalah bagaimana kemampuan bahasa pemrograman *Python* menyelesaikan permasalahan pengelompokan bunga iris. Memilih metode yang akan digunakan, *Python* menyediakan berbagai metode dan algoritma yang digunakan membuat *machine learning* menggunakan *library sklearn*. Oleh karena hal tersebut, teknik *supervised learning* dipilih algoritma KNN dan teknik *unsupervised learning* dipilih algoritma DBSCAN. Tahap mempersiapkan data, dalam hal ini *dataset* diperoleh dari repository *public dataset* <http://archive.ics.uci.edu/ml/index.php>, yang akan diolah adalah *dataset* bunga iris (*iris.data*), informasi mengenai *dataset* bunga iris (*iris.names*) dapat juga diunduh pada tautan tersebut. Tahap Implementasi, dari *dataset* yang telah dipersiapkan, dan pemilihan algoritma KNN serta DBSCAN selanjutnya diimplementasikan menggunakan bahasa pemrograman *Python*, sebagai *interface* menggunakan *Jupyter Notebook* yang telah terinstall didalamnya. Pada penelitian ini secara umum dalam membuat *machine learning* menggunakan *Python* dalam persiapan data dan implementasi sebagaimana pada gambar 2.

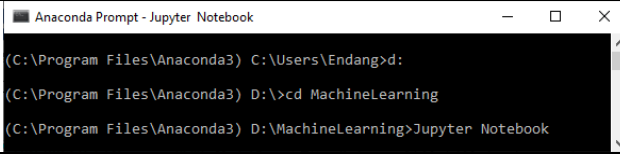
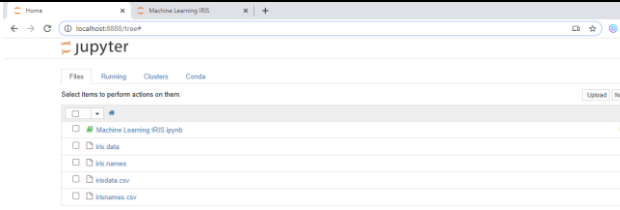
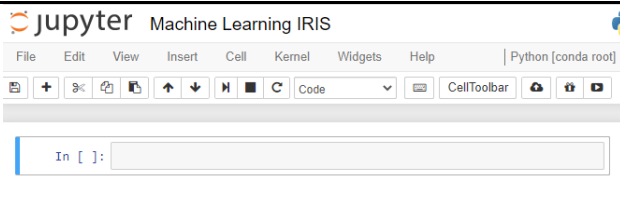


Sumber: Hasil Penelitian (2020)

Gambar 2. Langkah Implementasi *Python*

Untuk memulai pemrograman, diaktifkan terlebih dahulu *Jupyter Notebook* sehingga akan tampil halaman utama *Jupyter Notebook* untuk memulai pemrograman, sebagaimana pada Tabel 3.

Tabel 3. Persiapan *Jupyter Notebook*

Tampilan	Keterangan
	<p>Sebelum memulai pemrograman dengan <i>Jupyter Notebook</i>, terlebih dahulu diaktifkan melalui jendela <i>Anaconda Prompt</i>.</p>
	<p>Setelah <i>Jupyter Notebook</i> aktif, dapat dimulai untuk membuat file baru, dengan klik menu <i>New</i>, kemudian pilih <i>Python [conda root]</i>. Maka akan terbentuk file baru <i>Python</i> dengan ekstensi <i>ipynb</i>.</p>
	<p>Membuat File baru dapat juga dengan klik menu <i>File</i> → <i>New Notebook</i> → <i>Python [conda root]</i>. Selanjutnya dapat dimulai pemrograman untuk membuat <i>machine learning</i>. Setelah program dibuat untuk menjalankan program dengan klik menu <i>Cell</i> → <i>Run Cell</i>.</p>

Sumber: Hasil Penelitian (2020)

### Teknik *supervised learning* menggunakan *K-Nearest Neighbor (KNN)*

Langkah-langkah pemrograman menggunakan *Python*, diawali dengan persiapan *dataset* bunga iris (*iris tectorum*) yang telah diunduh (*iris.data*). Selanjutnya membuat file baru, kemudian memanggil *library-library* yang diperlukan dalam pembuatan program: a) *NumPy*, merupakan *library* yang akan digunakan untuk kebutuhan *scientific* dan matematis; b) *Pandas* merupakan *library* yang digunakan untuk membaca berbagai format data seperti file *.txt*, *.csv*, atau lainnya; c) *Matplotlib*, merupakan *library* yang digunakan untuk membuat grafik plot visualisasi hasil; d) *Sklearn*, merupakan *library* untuk berbagai metode dan algoritma yang digunakan untuk membuat *machine learning*, dalam hal ini memanggil *library KNeighbors Classifier*. *Source Python* untuk pembuatan program sebagaimana pada Tabel 4.

Tabel 4. *Source Code* Tahap Implementasi *Python* Menggunakan *K-Nearest Neighbor (KNN)*

Keterangan	Source Code
Memanggil library	<pre>import numpy as np import pandas as pd import matplotlib.pyplot as plt from matplotlib.colors import ListedColormap from sklearn import neighbors, datasets</pre>
Memanggil dataset.	<pre>iris = datasets.load_iris() X = iris.data[:, :2] y = iris.target</pre>
Mengaktifkan algoritma dan fitting data	<pre>knn = neighbors.KNeighborsClassifier(n_neighbors, weights=weights) knn.fit(X, y)</pre>
Visualisasi hasil	<pre>plt.figure() plt.pcolormesh(xx, yy, Z, cmap=cmap_light)  plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap_bold,             edgecolor='k', s=20) plt.xlim(xx.min(), xx.max()) plt.ylim(yy.min(), yy.max()) plt.title("3-Class classification (k = %i, weights = '%s')           % (n_neighbors, weights))  plt.show()</pre>

Sumber: Hasil Penelitian (2020)

Berdasarkan pada tabel 4, dalam mengenal *machine learning*, *source code* dan hasil visualisasi ketika *Cell* di *Run Cells* selengkapnya sebagaimana pada gambar 3 sebagai berikut:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn import neighbors, datasets

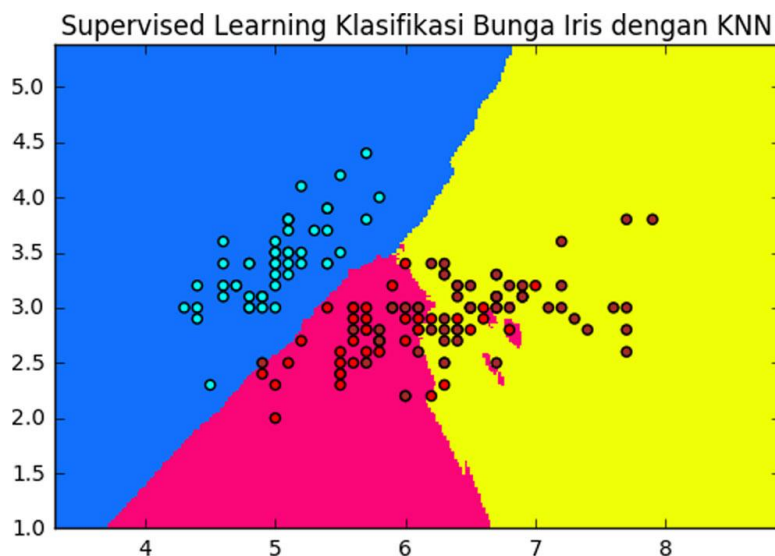
iris = datasets.load_iris()
X = iris.data[:, :2]
y = iris.target
n_neighbors=15
h = .02

cmap_light = ListedColormap(['#1270FC', '#F90677', '#FFFF07'])
cmap_bold = ListedColormap(['cyan', 'red', 'brown'])

for weights in ['uniform']:
    knn=neighbors.KNeighborsClassifier(n_neighbors, weights='uniform')
    knn.fit(X,y)
    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                        np.arange(y_min, y_max, h))

    Z = knn.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    plt.figure()
    plt.pcolormesh(xx, yy, Z, cmap=cmap_light)
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap_bold,
                edgecolor='k', s=20)

    plt.xlim(xx.min(), xx.max())
    plt.ylim(yy.min(), yy.max())
    plt.title('Supervised Learning Klasifikasi Bunga Iris dengan KNN')
plt.show()
```



Sumber: Hasil Penelitian (2020)

Gambar 3. *Source Code* dan Hasil Visualisasi Klasifikasi Bunga Iris

Untuk memulai pembuatan *machine learning* dengan teknik *supervised learning* menggunakan algoritma *KNN*, diperlukan mengimport *library datasets* dan algoritma *neighbors*, pada *KNN* perlu ditentukan jumlah *k* dalam *source* tersebut jumlah *k* ditentukan dengan mengisikan nilai *n\_neighbors* yaitu 15, dengan jarak .02 selanjutnya dilakukan klasifikasi dengan mengaktifkan *source neighbors.KneighborsClassifier* akan diperoleh klasifikasi berdasarkan label mayoritas berdasarkan hasil prediksi terhadap data yang telah dikonversi kedalam array *source knn.predict (np.c\_[xx.ravel(), yy.ravel()])* yang

mana np merupakan numpy library pada Python guna perhitungan scientific dan matematis. Hasil klasifikasi (Classification) yang diperoleh ditampilkan ke bentuk grafik menggunakan library matplotlib dikelompokkan berdasarkan dataset sepal dan petal bunga iris yang terdiri dari tiga Class, node menunjukkan dataset yaitu node warna biru merupakan Iris-setosa, node warna merah merupakan Iris-versicolor dan node warna coklat merupakan Iris-virginica.

**Teknik unsupervised learning menggunakan DBSCAN**

Pemrograman menggunakan Python, dengan menggunakan nama file dan dataset yang sama dapat dilanjutkan dengan insert Cell baru. Selanjutnya memanggil library-library yang diperlukan dalam pembuatan program: a) Pandas merupakan library yang digunakan untuk membaca berbagai format data seperti file .txt, .csv, atau lainnya; b) Matplotlib, merupakan library yang digunakan untuk membuat grafik plot visualisasi hasil; c) Sklearn, merupakan library untuk berbagai metode dan algoritma yang digunakan untuk membuat machine learning, dalam hal ini memanggil library DBSCAN untuk mengaktifkan algoritma yang akan digunakan; d) Principal Component Analysis (PCA) untuk memvisualisasi variabel dalam dataset kedalam grafik 2 dimensi. Source Python untuk pembuatan program sebagaimana pada Tabel 5.

Tabel 5. Source Code Tahap Implementasi Python Menggunakan DBSCAN

Keterangan	Source Code
Memanggil library	<pre>import pandas as pd import matplotlib.pyplot as plt from sklearn import datasets from sklearn.cluster import DBSCAN from sklearn.decomposition import PCA</pre>
Memanggil dataset	<pre>iris=datasets.load_iris() x_axis=iris.data[:, 0], y_axis=iris.data[:, 2],</pre>
Mengaktifkan algoritma dan fitting data	<pre>dbscan=DBSCAN() dbscan.fit(iris.data) pca=PCA(n_components=2).fit(iris.data) pca_2d=pca.transform(iris.data)</pre>
Visualisasi hasil	<pre>for i in range(0, pca_2d.shape[0]):     if dbscan.labels_[i] == 0:         c1=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='blue',marker='+')     elif dbscan.labels_[i] == 1:         c2=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='cyan',marker='o')     elif dbscan.labels_[i] == -1:         c3=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='red',marker='s')  plt.legend([c1, c2, c3], ['Cluster 1', 'Cluster 2','Noise']) plt.title('Unsupervised Learning Clusterisasi Bunga Iris dengan DBSCAN') plt.show()</pre>

Sumber: Hasil Penelitian (2020)

Untuk memulai pembuatan machine learning dengan teknik unsupervised learning menggunakan algoritma DBSCAN, diperlukan mengimport library datasets, algoritma DBSCAN, juga PCA (n\_components=2) dalam hal ini tingkat kerapatan (density) diperoleh dari dataset bunga iris yang memiliki atribut sepal length, sepal width, petal length, dan petal width, selanjutnya dilakukan dekomposisi dari atribut (variable) dengan mengaktifkan source pca.transform(iris.data) akan diperoleh hasil klusterisasi yang divisualisasikan dalam grafik. Hasil klusterisasi (Cluster) ditampilkan menggunakan library matplotlib dikelompokkan berdasarkan dataset sepal dan petal bunga iris yang terdiri dari tiga Cluster, node menunjukkan dataset. Diperoleh Cluster 1 yaitu node warna biru merupakan Cluster memiliki kepadatan dengan data Sepal Length antara -4 sampai -2, Cluster 2 node warna cyan merupakan Cluster memiliki kepadatan dengan data Sepal Length antara 0 sampai 3 dan Cluster Noise node warna merah merupakan Cluster dengan yang datanya tidak berada pada tingkat kepadatan (density).



```

import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import DBSCAN
from sklearn.decomposition import PCA

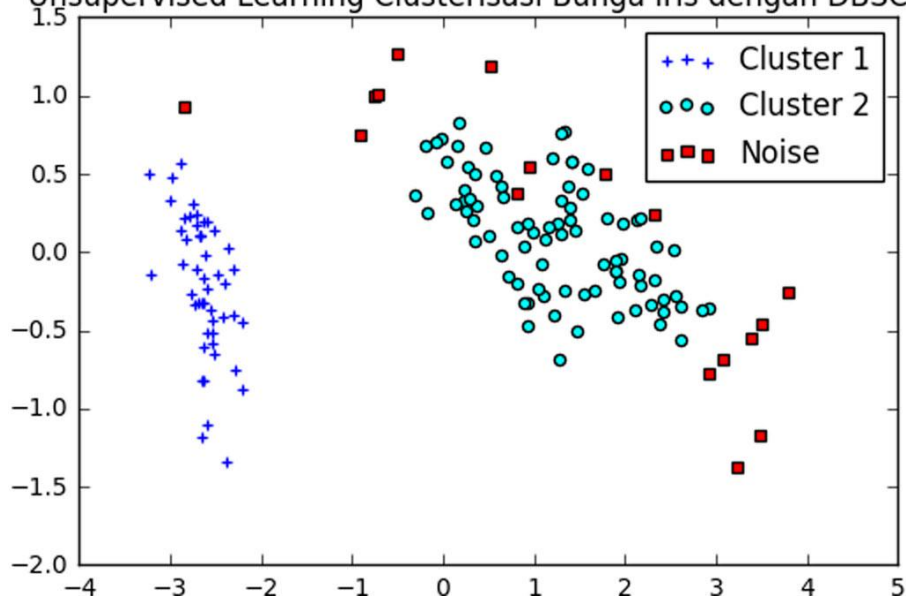
iris=datasets.load_iris()
x_axis=iris.data[:, 0],
y_axis=iris.data[:, 2],
dbscan=DBSCAN()
dbscan.fit(iris.data)
pca=PCA(n_components=2).fit(iris.data)
pca_2d=pca.transform(iris.data)

for i in range(0, pca_2d.shape[0]):
    if dbscan.labels_[i] == 0:
        c1=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='blue',marker='+')
    elif dbscan.labels_[i] == 1:
        c2=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='cyan',marker='o')
    elif dbscan.labels_[i] == -1:
        c3=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='red',marker='s')

plt.legend([c1, c2, c3], ['Cluster 1', 'Cluster 2','Noise'])
plt.title('Unsupervised Learning Clusterisasi Bunga Iris dengan DBSCAN')
plt.show()

```

Unsupervised Learning Clusterisasi Bunga Iris dengan DBSCAN



Sumber: Hasil Penelitian (2020)

Gambar 4. Source Code dan Hasil Visualisasi Clusterisasi Bunga Iris

#### 4. Kesimpulan

Python memiliki *source code* yang sederhana, memudahkan pengembangan aplikasi, mulai dari tahap coding, testing, perbaikan jika ada kesalahan. *Machine learning* menggunakan Python pada penelitian ini dengan teknik *supervised learning* dipilih algoritma *K-Nearest Neighbor (KNN)*, sedangkan teknik *unsupervised learning* dipilih algoritma *Density Based Spatial Clustering of Application with Noise (DBSCAN)*. *Public dataset* yang digunakan diperoleh dari *UCI Repository* berisi sample acak bunga milik tiga spesies bunga iris yaitu setosa, versicolor, dan virginica. Python dengan *interface* menggunakan *Jupyter Notebook* menyediakan berbagai teknik baik *supervised* maupun *unsupervised learning* dan menyediakan berbagai *library* *NumPy*, *Pandas*, *matplotlib*, *sklearn*, dan menyediakan dukungan berbagai

algoritma seperti KNN diaktifkan melalui `from sklearn import neighbors`, DBSCAN diaktifkan melalui `from sklearn.cluster import DBSCAN`. Kemampuan Python memberikan hasil output sesuai input dalam *dataset*.

## Referensi

- [1] P. D. Kusuma, *Machine Learning Teori, Program, dan Studi Kasus*. Yogyakarta: Deepublish, 2020.
- [2] Z. A. Fikriya, M. I. Irawan, and Soetrisno, "Implementasi Extreme Learning Machine untuk Pengenalan Objek Citra Digital," *J. Sains dan Seni ITS*, vol. 6, no. 1, pp. A18–A23, 2017.
- [3] Nurhayati, Busman, and R. P. Iswara, "Pengembangan Algoritma Unsupervised Learning Technique Pada Big Data Analysis di Media Sosial sebagai media promosi Online Bagi Masyarakat," *J. Tek. Inform.*, vol. 12, no. 1, pp. 79–96, 2019.
- [4] F. S. Pamungkas, B. D. Prasetya, and I. Kharisudin, "Perbandingan Metode Klasifikasi Supervised Learning pada Data Bank Customers Menggunakan Python," in *Prosiding Seminar Nasional Matematika*, 2020, vol. 3, pp. 689–694.
- [5] F. A. Bachtiar, I. K. Syahputra, and S. A. Wicaksono, "Perbandingan Algoritme Machine Learning untuk Memprediksi Pengambil Matakuliah," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 5, p. 543, 2019.
- [6] B. S. Ashari, S. C. Otniel, and Rianto, "Perbandingan Kinerja K-Means Dengan DSCAN Untuk Metode Clustering Data Penjualan Online Retail," *J. Siliwangi*, vol. 5, no. 2, pp. 72–77, 2019.
- [7] I. Muhammad and Z. Yan, "Supervised Machine Learning Approaches: a Survey," *ICTACT J. Soft Comput.*, vol. 05, no. 03, pp. 946–952, 2015.
- [8] Fahrizal, F. O. Reynaldi, and N. Hikmah, "Implementasi Machine Learning pada Sistem PETS Identification Menggunakan Python Berbasis Ubuntu," *J. Inf. Syst. Informatics Comput.*, vol. 4, no. 1, pp. 86–91, 2020.
- [9] H. Herlawati and R. T. Handayanto, "Penggunaan Matlab dan Python dalam Klasterisasi Data," *J. Kaji. Ilm.*, vol. 20, no. 1, pp. 103–118, 2020.
- [10] T. Wahyono, *Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan*. Yogyakarta: Gava Media, 2018.
- [11] <http://www.python.org/>, "Python." [Online]. Available: <http://www.python.org/>. [Accessed: 11-Nov-2020].
- [12] <http://repo.anaconda.com/>, "Anaconda." [Online]. Available: <http://repo.anaconda.com/>. [Accessed: 11-Nov-2020].
- [13] A. Wanto et al., *Data Mining : Algoritma dan Implementasi*. Medan: Yayasan Kita Menulis, 2020.
- [14] <http://archive.ics.uci.edu/ml/datasets/Iris>, "UCI Machine Learning Repository: Iris Data Set." [Online]. Available: <http://archive.ics.uci.edu/ml/datasets/Iris>. [Accessed: 11-Nov-2020].