

# BAB 1

---

## CHAPTER 1

---

### 1.1 1174070 - Arrizal Furqona Gifary

#### 1.1.1 Teori

##### 1. Sejarah dan Perkembangan

Kecerdasan Buatan atau dalam Bahasa Inggris sering disebut Artificial Intelligence yang sering disebut juga sebagai AI, pada 10 tahun lalu masyarakat belum terlalu mengetahui hal tersebut dan masih menjadi bahan candaan dikalangan masyarakat. Awal perkembangan AI dimulai pada tahun 1952-1969 yang dimulai dengan kesuksesan Newell dan temannya Simon menggunakan sebuah program yang disebut dengan General Problem Solver. Program ini dibangun untuk tujuan penyelesaian masalah secara manusiawi. Pada tahun 1966-1974 perkembangan kecerdasan buatan mulai melambat. Ada 3 faktor utama yang menyebabkan hal itu terjadi:

- Banyak subjek pada program AI yang bermunculan hanya mengandung sedikit atau bahkan sama sekali tidak mengandung sama sekali pengetahuan (knowledge).

- Kecerdasan buatan harus bisa menyelesaikan banyak masalah.
- Untuk menghasilkan perilkau inteliensia ada beberapa batasan pada struktur yang bisa digunakan.

Definisi kecerdasan buatan itu sendiri adalah suatu system teknologi yang didalamnya ditambahakan kecerdasan oleh manusia, kecerdasan buatan diatur dan dikembangkan dalam konteks ilmiah, dan bentukan dari kecerdasan entitas ilmiah yang ada.

## 2. Definisi

Supervised learning, klasifikasi, regresi, unsupervised learning, dataset, trainingset dan testingset.

- **Supervised Learning**

Supervised Learning merupakan sebuah tipe learning yang mempunyai variable input dan variable output, tipe ini juga menggunakan satu algoritma atau lebih dari satu algoritma yang digunakan untuk mempelajari fungsi pemetaan dari input ke output.

- **Klasifikasi**

Klasifikasi adalah pengelompokan data di mana data yang digunakan memiliki label atau kelas target. Sehingga algoritma untuk menyelesaikan masalah klasifikasi dikategorikan ke dalam pembelajaran terbimbing.

- **Regresi**

regressi metode analisis statistik yang digunakan untuk dapat melihat efek antara dua atau lebih variabel. Hubungan variabel dalam pertanyaan adalah fungsional yang diwujudkan dalam bentuk model matematika. Dalam analisis regresi, variabel dibagi menjadi dua jenis, yaitu variabel respons atau yang biasa disebut variabel dependen dan variabel independen atau dikenal sebagai variabel independen. Ada beberapa jenis analisis regresi, yaitu regresi sederhana yang mencakup linear sederhana dan regresi non-linear sederhana dan regresi berganda yang mencakup banyak linier atau non-linear berganda. Analisis regresi digunakan dalam pembelajaran mesin pembelajaran dengan metode pembelajaran terawasi.

- **Unsupervised learning**

unsupervised learning jenis pembelajaran di mana kita hanya memiliki data input (input data) tetapi tidak ada variabel output yang terkait. Tujuan dari pembelajaran tanpa pengawasan adalah untuk memodelkan struktur dasar atau distribusi data dengan tujuan mempelajari data lebih lanjut, dengan kata lain, itu adalah fungsi simpulan yang menggambarkan atau menjelaskan data.

- **Data set**

Data set objek yang merepresentasikan data dan relasinya di memory.

Strukturnya mirip dengan data di database. Dataset berisi koleksi dari datatable dan datarelation.

- Training Set

Training set adalah bagian dari dataset yang di latih untuk membuat prediksi atau menjalankan fungsi dari algoritma ML lain sesuai dengan masing-masing. Memberikan instruksi melalui algoritma sehingga mesin yang di praktikkan dapat menemukan korelasinya sendiri.

- Testing Set

testing set adalah bagian dari dataset yang kami uji untuk melihat akurasinya, atau dengan kata lain untuk melihat kinerjanya.

### 1.1.2 Praktek

1. Instalasi Library scikit dari ianaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer



**Gambar 1.1** Instalasi Package Scikit Learn

| Name  | Type       | Size        | Value   |
|-------|------------|-------------|---|
| X     | float32    | (150, 2000) | [0.5488125, 0.7151854, 0.00270355, ..., 0.4002078, 0.5438484, 0.361771] |
| X_new | float32    | (20, 2000)  | [0.4088971, 1.3157483, -0.65989507, ..., -0.5368945, 0.49547915]        |
| a     | float64    | (1, 4)      | [1., 2., 3., 4.]  |
| iris  | util.Bunch | 6           | Bunch object of scikit-learn.util module                                |
| x     | float64    | (150, 4)    | [5.1, 3.5, 1.4, 0.2]  |
| y     | int32      | (150,)      | [0 0 ... 2 2 2]   |

**Gambar 1.2** Isi Variabel Explorer

2. Mencoba loading an example dataset

```

1 from sklearn import datasets # Digunakan Untuk Memanggil
    class datasets dari library sklearn
2 iris = datasets.load_iris() # Menggunakan contoh datasets
    iris
3 x = iris.data             # Menyimpan nilai data sets iris
    pada variabel x
4 y = iris.target            # Menyimpan nilai data label iris
    pada variabel y

```

3. Mencoba Learning dan predicting

```

1 from sklearn.neighbors import KNeighborsClassifier #Digunakan
   Untuk Memanggil fungsi KNeighborsClassifier
2                                         # pada
3 class sklearn dan library sklearn
4 import numpy as np # memanggil library numpy dan dibuat alias
   np
5 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
   , dan memanggil fungsi KNeighborsClassifier
6                                         #dan mendefinisikan k
   -nya adalah 1
7 knn.fit(x,y)                         #Perhitungan
   matematika library kkn
8 a=np.array([1.0,2.0,3.0,4.0])        #Membuat Array
9 a = a.reshape(1,-1)                  #Mengubah Bentuk
   Array jadi 1 dimensi
10 hasil = knn.predict(a)              #Memanggil fungsi
   prediksi
11 print(hasil)                      #menampilkan hasil
   prediksi

```

#### 4. Mencoba Model Persistence

```

1 from sklearn import svm # Digunakan untuk memanggil class svm
   dari library sklearn
2 from sklearn import datasets # Diguangkan untuk class datasets
   dari library sklearn
3 clf = svm.SVC()           # membuat variabel clf , dan
   memanggil class svm dan fungsi SVC
4 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
   iris dan mengembalikan nilainya.
5 clf.fit(X, y)             #Perhitungan nilai label
6
7 from joblib import dump, load #memanggil class dump dan load
   pada library joblib
8 dump(clf, '1174070.joblib') #Menyimpan model kedalam 1174027.
   joblib
9 hasil = load('1174070.joblib') #Memanggil model 1174027
10 print(hasil) # Menampilkan Model yang dipanggil sebelumnya

```

#### 5. Mencoba Conventions

```

1 import numpy as np # memanggil library numpy dan dibuat alias
   np
2 from sklearn import random_projection #Memanggil class
   random_projection pada library sklean
3
4 rng = np.random.RandomState(0) #Membuat variabel rng , dan
   mendefinisikan np, fungsi random dan attr RandomState
   kedalam variabel
5 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
   nilai random dari 10 – 2000
6 X = np.array(X, dtype='float32') #menyimpan hasil nilai
   random sebelumnya, kedalam array , dan menentukan
   typedatanya sebagai float32

```

```

7 X.dtype # Mengubah data tipe menjadi float64
8
9 transformer = random_projection.GaussianRandomProjection() #
    membuat variabel transformer , dan mendefinisikan
    classrandom_projection dan memanggil fungsi
    GaussianRandomProjection
10 X_new = transformer.fit_transform(X) # membuat variabel baru
    dan melakukan perhitungan label pada variabel X
11 X_new.dtype # Mengubah data tipe menjadi float64
12 print(X_new) # Menampilkan isi variabel X_new

```

### 1.1.3 Penanganan Error

#### 1. ScreenShoot Error

```

file "D:\Vullah\Semester 6\Tercerdason\Butan\lah\sklearn.py", line 8, in <module>
    from sklearn import datasets
ImportError: cannot import name 'datasets' from 'sklearn' (D:\Vullah\Semester
6\tercerdason\Butan\lah\sklearn.py)

```

**Gambar 1.3** Import Error

```

file "C:\Users\svirul\Anaconda3\lib\site-packages\sklearn\utils\validation.py", line
556, in check_array
    "Expected 2D array, got %r instead"
ValueError: Expected 2D array, got 1D array instead
array[[ 1.23]
      [ 2.34]
      [ 3.45]
      [ 4.56]
      [ 5.67]
      [ 6.78]
      [ 7.89]
      [ 8.90]
      [ 9.01]
      [ 10.12]]
Reshape your data either using array.reshape(-1, 1) if your data has a single feature or
array.reshape(1, -1) if it contains a single sample.

```

**Gambar 1.4** Value Error

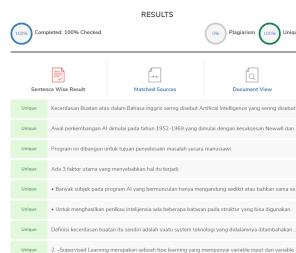
#### 2. Tuliskan Kode Error dan Jenis Error

- Import Error
- Value Error

#### 3. Cara Penangan Error

- Import Error  
Dengan Menginstall Library Yang Tidak Ditemukan
- Value Error  
Mengubah Bentuk Arraynya, Menjadi 1 Dimensi

### 1.1.4 Bukti Tidak Plagiat



Gambar 1.5 Bukti Tidak Melakukan Plagiat 1



Gambar 1.6 Bukti Tidak Melakukan Plagiat 2

## 1.2 Fanny Shafira Damayanti (1174069)

### 1.2.1 Teori

#### 1. Definisi Kecerdasan buatan

Kecerdasan buatan atau Artificial intelligence merupakan kecerdasan yang ditambahkan kedalam suatu sistem yang diatur secara ilmiah. Kecerdasan buatan dibuat untuk menggantikan pekerjaan yang dilakukan oleh manusia menjadi dikerjakan oleh sistem.

#### 2. Sejarah Kecerdasan Buatan

- Abad 17, Rene Descartes berkata bahwa tubuh hewan adalah sekumpulan mesin yang rumit.
- 1642, Blaise Pascal menciptakan mesin penghitung digital mekanis pertama.
- Abad 19, Charles Babbage dan Ada Lovelace bekerja di program penghitung mekanis.
- 1950, John McCarthy membuat istilah “Kecerdasan Buatan”.
- 1960-1970, Joel Moses membuat program yang pertama kali sukses dalam bidang matematika.
- 1980, jaringan saraf digunakan secara meluas dengan algoritme perambatan balik.

- 2004, DARPA membuat kendaraan yang bisa dijalankan sendiri tanpa manusia.

### 3. Perkembangan kecerdasan buatan

- Masa persiapan (1943-1946) Warren McCulloch dan Walter Pitt menge-mukakan tiga hal : pengetahuan fisiologi dasar dan fungsi sel syaraf dalam otak, analisa formal tentang logika proposisi, dan teori komputasi Turing.

Pada tahun 1950, Norbert Wiener membuat penelitian mengenai prinsip-prinsip teori feedback.

Pada tahun 1956, John McCarthy meyakinkan Minsky, Claude Shannon dan Nathaniel Rochester untuk membantunya melakukan penelitian dalam bidang Otomata, Jaringan Syaraf dan pembelajaran intelejensi.

- Awal perkembangan (1952-1969) Pada tahun 1958, McCarthy di MIT AI Lab Memo No.1 mendefinisikan bahasa pemrograman tingkat tinggi yaitu LISP,

Pada tahun 1959, Nathaniel Rochester dari IBM dan mahasiswa-mahasiswanya mengeluarkan program kecerdasan buatan yaitu Geometry Theorem Prover.

Pada tahun 1963, program yang dibuat James Slagle mampu menyelesaikan masalah integral tertutup untuk mata kuliah Kalkulus. Pada tahun 1986, program analogi buatan Tom Evan menyelesaikan masalah analogi geometris yang ada pada tes IQ.

- Perkembangan Kecerdasan Buatan Melambat (1969-1979) Bruce Buchanan dan Joshua Lederberg yang membuat program untuk memecahkan masalah struktur molekul dari informasi yang didapatkan dari spectrometer massa.
- AI Menjadi sebuah industri Industrialisasi kecerdasan buatan diawali dengan ditemukannya sistem pakar yang dinamakan R1 yang mampu mengkonfigurasi system-sistem computer baru.
- Kembalinya Jaringan Syaraf Tiruan (1986-sekarang) Pada tahun 1985-an setidaknya empat kelompok riset menemukan kembali algoritma belajar propagasi balik (Back-Propagation Learning). Algoritma ini berhasil diimplementasikan ke dalam bidang ilmu computer dan psikologi.

### 4. Definisi Supervised Learning

Supervised Learning merupakan cabang dari Artificial Intelligence. supervised learning adalah suatu ilmu yang mempelajari perancangan dan pengembangan algoritma.

### 5. Klasifikasi Supervised Learning

- Logistic regression.

- K-nearest neighbors.
- Support vector machine (SVM)
- Naive Bayes.
- Decision tree classification.
- Random forest classification.

## 6. Regresi dan Unsupervised Learning

Regresi merupakan sebuah metode analisis statistic yang digunakan untuk mengetahui pengaruh antara dua variable atau lebih.

Untuk mempelajari Unsupervised learning kita tidak perlu data training untuk melakukan prediksi maupun klasifikasi.

## 7. Dataset

Dataset merupakan objek yang mempresentasikan data dan relasinya pada memori.

## 8. Training Set

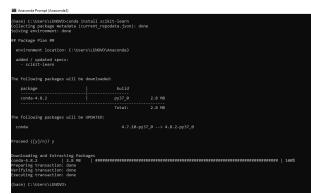
Training Set merupakan bagian dari dataset untuk membuat prediksi atau menjalankan fungsi dari sebuah algoritma Machine Learning.

## 9. Testing Set

Testing set digunakan untuk mengukur apakah classifier berhasil melakukan klasifikasi dengan benar.

### 1.2.2 Instalasi

#### 1. Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer



**Gambar 1.7** Instalasi Package Scikit Learn

| Name   | Type       | Size | Value                                |
|--------|------------|------|--------------------------------------|
| digits | util.Bunch | 5    | Bunch object of sklearn.utils module |
| Iris   | util.Bunch | 6    | Bunch object of sklearn.utils module |

**Gambar 1.8** Isi Variabel Explorer

2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%% Mencoba loading an example dataset
2 from sklearn import datasets # Digunakan Untuk Memanggil
   class datasets dari library sklearn
3 iris = datasets.load_iris() # Menggunakan contoh datasets
   iris
4 x = iris.data           # Menyimpan nilai data sets iris
   pada variabel x
5 y = iris.target         # Menyimpan nilai data label iris
   pada variabel y

```

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Digunakan
   Untuk Memanggil fungsi KNeighborsClassifier
3                                     # pada
   class sklearn dan library sklearn
4 import numpy as np # memanggil library numpy dan dibuat alias
   np
5 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
   , dan memanggil fungsi KNeighborsClassifier
6                                     #dan mendefinisikan k
   -nya adalah 1
7 knn.fit(x,y)                      #Perhitungan
   matematika library kkn
8 a=np.array([1.0,2.0,3.0,4.0])      #Membuat Array
9 a = a.reshape(1,-1)                 #Mengubah Bentuk
   Array jadi 1 dimensi
10 hasil = knn.predict(a)            #Memanggil fungsi
   prediksi

```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm
   dari library sklearn
3 from sklearn import datasets # Diguangkan untuk class datasets
   dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan
   memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
   iris dan mengembalikan nilainya.
6 clf.fit(X, y)             #Perhitungan nilai label
7
8 from joblib import dump, load #memanggil class dump dan load
   pada library joblib

```

```

9 dump(clf, '1174069.joblib') #Menyimpan model kedalam 1174069.
   joblib
10 hasil = load('1174069.joblib') #Memanggil model 1174069

```

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %% Conventions
2 import numpy as np # memanggil library numpy dan dibuat alias
   np
3 from sklearn import random_projection #Memanggil class
   random_projection pada library sklearn
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
   mendefinisikan np, fungsi random dan attr RandomState
   kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
   nilai random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
   random sebelumnya, kedalam array , dan menentukan
   typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
   membuat variabel transformer , dan mendefinisikan
   classrandom_projection dan memanggil fungsi
   GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru
   dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64

```

### 1.2.3 Penanganan Error

1. ScreenShoot Error



```

File "D:\Kuliah\Semester 6\Kecerdasan Buatan\Latihan\1\sklearn.py", line 8, in <module>
    from sklearn import datasets
ImportError: cannot import name 'datasets' from 'sklearn' (D:\Kuliah\Semester 6\Kecerdasan Buatan\Latihan\1\sklearn.py)

```

Gambar 1.9 Import Error

2. Tuliskan Kode Error dan Jenis Error

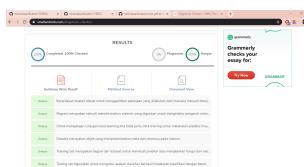
- Import Error

3. Cara Penangan Error

- Import Error

Dengan Menginstall Library Yang Tidak Ditemukan

## 1.2.4 Bukti Tidak Plagiat



**Gambar 1.10** Bukti Tidak Melakukan Plagiat Chapter 1

## 1.2.5 Link Youtube

<https://youtu.be/Ra4Lu-C8OQY>

## 1.3 1174086 - Tia Nur Candida

### 1.3.1 Teori

#### 1. Definisi Kecerdasan Buatan

Kecerdasan Buatan adalah suatu ilmu yang mempelajari bagaimana cara komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Secara sederhana AI adalah teknik dan ilmu untuk membangun atau membuat suatu mesin menjadi cerdas, terutama pada program komputer. Kecerdasan yang dimaksud yaitu seperti yang dimiliki oleh manusia namun pada mesin akan dibuat cepat dan tepat atau akurat.

#### 2. Sejarah Kecerdasan Buatan

Sejarah kecerdasan buatan dimulai pada zaman kuno. Benih kecerdasan buatan modern ditanamkan oleh filosof klasik dengan berusaha menggambarkan proses berpikir manusia. Karya tersebut memuncak pada penemuan komputer digital yang di program pada tahun 1940 an, dimana terdapat sebuah mesin yang didasarkan pada esensi abstrak penalaran matematika. Istilah kecerdasan buatan pertama kali dikemukakan pada tahun 1956 di Konferensi Dartmouth yang kemudian sejak saat itu kecerdasan buatan terus berkembang.

#### 3. Perkembangan kecerdasan buatan

- Masa Persiapan AI (1943-1956) Pada tahun 1943, Warren McCulloch dan Walter Pitt mengemukakan tiga hal : pengetahuan fisiologi dasar dan fungsi sel syaraf dalam otak, analisa formal tentang logika proposisi, dan teori komputasi Turing. Mereka berhasil membuat suatu model sel syaraf tiruan dimana setiap sel syaraf digambarkan sebagai 'on' dan 'off'. Mereka menunjukkan bahwa setiap fungsi dapat dihitung dengan suatu jaringan sel syaraf dan bahwa semua hubungan

logis dapat diimplementasikan dengan struktur jaringan yang sederhana. Pada tahun 1950, Nobert Wiener membuat penelitian mengenai prinsip-prinsip teori feedback. Contoh yang terkenal adalah thermostat. Penemuan ini juga merupakan awal dari perkembangan AI. Pada tahun 1956, John McCarthy meyakinkan Minsky, Claude Shannon dan Nathaniel Rochester untuk membantunya melakukan penelitian dalam bidang Otomata, Jaringan Syaraf dan pembelajaran intelijensia. Mereka mengerjakan proyek ini selama 2 bulan di Dartmouth. Hasilnya adalah program yang mampu berpikir non-numerik dan menyelesaikan masalah pemikiran, yang dinamakan Principia Mathematica. Hal ini menjadikan McCarthy disebut sebagai bapak kecerdasan buatan.

- Awal perkembangan AI (1952-1969) Kecerdasan buatan banyak mengalami kesuksesan pada tahun pertama. Pada tahun 1958, McCarthy di MIT AI Lab Memo No.1 mendefinisikan bahasa pemrograman tingkat tinggi yaitu LISP, yang sekarang mendominasi pembuatan program-program kecerdasan buatan. Kemudian, McCarthy membuat program yang dinamakan Programs with Common Sense. Di dalam program tersebut, dibuat rancangan untuk menggunakan pengetahuan dalam mencari solusi. Pada tahun 1959, Nathaniel Rochester dari IBM dan mahasiswa-mahasiswanya mengeluarkan program kecerdasan buatan yaitu Geometry Theorem Prover. Program ini dapat mengeluarkan suatu teorema menggunakan aksioma-aksioma yang ada. Pada tahun 1963, program yang dibuat James Slagle mampu menyelesaikan masalah integral tertutup untuk mata kuliah Kalkulus. Pada tahun 1986, program analogi buatan Tom Evan menyelesaikan masalah analogi geometris yang ada pada tes IQ.
- Perkembangan kecerdasan buatan melambat (1966-1974) Banyak masalah yang perlu di selesaikan oleh kecerdasan buatan dan baru sedikit program yang keluar menyebabkan melambat.
- Kecerdasan buatan menjadi sebuah industri ( 1980 - 1988 ) Industrialisasi kecerdasan buatan diawali dengan ditemukannya sistem pakar yang dinamakan R1 yang mampu mengkonfigurasi sistem-sistem computer baru. Program tersebut mulai dioperasikan di Digital Equipment Corporation (DEC), McDermott, pada tahun 1982. Pada tahun 1986, R1 telah berhasil menghemat US Dolar 40 juta per tahun. Pada tahun 1988, kelompok kecerdasan buatan di DEC menjalankan 40 sistem pakar. Hampir semua perusahaan besar di USA mempunyai divisi AI. Sehingga perusahaan yang sejak tahun 1982 hanya menghasilkan beberapa juta US dolar per tahun meningkat menjadi 2 miliar US dolar per tahun pada tahun 1988.
- Kembalinya Jaringan Syaraf Tiruan ( 1986 - Sekarang ) Meskipun bidang ilmu computer menolak jaringan syaraf tiruan setelah diterbitkannya buku "Perceptrons" karangan Minsky dan Papert, tetapi

para ilmuwan masih mempelajari bidang ilmu tersebut dari sudut pandang yang lain yaitu fisika. Para ahli fisika seperti Hopfield (1982) menggunakan teknik-teknik mekanika statistika untuk menganalisa sifat-sifat pentimpanan dan optimasi pada jaringan syaraf. Para ahli psikologi, David Rumelhart dan Geoff Hinton, melanjutkan penelitian mengenai model jaringan syaraf tiruan pada memori. Pada tahun 1985-an setidaknya empat kelompok riset menemukan kembali algoritma belajar propagasi balik (Back-Propagation Learning). Algoritma ini berhasil diimplementasikan ke dalam bidang ilmu computer dan psikologi.

#### 4. Definisi Supervised Learning

Merupakan tipe Machine Learning dimana model ini menyediakan training data berlabel. Supervised learning merupakan suatu pembelajaran yang terawasi dimana jika output yang diharapkan telah diketahui sebelumnya. Supervised Learning adalah tipe learning di mana kita mempunyai variable input dan variable output, dan menggunakan satu algoritma atau lebih untuk mempelajari fungsi pemetaan dari input ke output. Goal-nya adalah untuk memperkirakan fungsi pemetaannya, sehingga ketika kita mempunya input baru, kita dapat memprediksi output untuk input tersebut.

#### 5. Klasifikasi

- Logistic regression.
- K-nearest neighbors.
- Support vector machine (SVM).
- Naive Bayes.
- Decision tree classification.
- Random forest classification.

#### 6. Regresi

Regresi adalah suatu metode analisis statistik yang digunakan untuk melihat pengaruh antara dua atau lebih banyak variabel. Hubungan variabel tersebut bersifat fungsional yang diwujudkan dalam suatu model matematis.

#### 7. Unsupervised Learning

Unsupervised Learning adalah tipe learning di mana kita hanya mempunyai data masukan (input data) tetapi tidak ada output variable yang berhubungan.

Goal dari unsupervised learning adalah untuk memodelkan struktur dasar atau distribusi dalam data dengan tujuan untuk mempelajari data lebih

jauh lagi, dengan kata lain, adalah menyimpulkan fungsi yang mendeskripsikan atau menjelaskan data.

#### 8. Dataset

Dataset adalah objek yang merepresentasikan data dan relasinya di memory. Strukturnya mirip dengan data di database. Dataset berisi koleksi dari datatable dan datarelation.

#### 9. Training Set

Training set adalah bagian dataset yang kita latih untuk membuat prediksi atau menjalankan fungsi dari sebuah algoritma ML lainnya sesuai tujuannya masing-masing. Kita memberikan petunjuk melalui algoritma agar mesin yang kita latih bisa mencari korelasinya sendiri.

#### 10. Test Set

Test set adalah bagian dataset yang kita tes untuk melihat keakuratannya, atau dengan kata lain melihat performanya.

### 1.3.2 Praktek

- Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer



**Gambar 1.11** Instalasi Package Scikit Learn

| Name   | Type       | Size | Value                                |
|--------|------------|------|--------------------------------------|
| digits | util.Bunch | 5    | Bunch object of sklearn.utils module |
| Iris   | util.Bunch | 6    | Bunch object of sklearn.utils module |

**Gambar 1.12** Isi Variabel Explorer

- Mencoba loading an example dataset

```

1 %% Mencoba loading an example dataset
2 from sklearn import datasets # Digunakan Untuk Memanggil
# class datasets dari library sklearn
3 iris = datasets.load_iris() # Menggunakan contoh datasets
# iris
4 x = iris.data # Menyimpan nilai data sets iris
# pada variabel x
5 y = iris.target # Menyimpan nilai data label iris
# pada variabel y

```

### 3. Mencoba Learning dan predicting

```

1 #%%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Digunakan
   Untuk Memanggil fungsi KNeighborsClassifier
3                                     # pada
4 class sklearn dan library sklearn
5 import numpy as np # memanggil library numpy dan dibuat alias
   np
6 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
   , dan memanggil fungsi KNeighborsClassifier
7                                     #dan mendefinisikan k
   -nya adalah 1
8 knn.fit(x,y)                      #Perhitungan
   matematika library kkn
9 a=np.array([1.0,2.0,3.0,4.0])      #Membuat Array
10 a=a.reshape(1,-1)                 #Mengubah Bentuk
    Array jadi 1 dimensi
11 hasil = knn.predict(a)            #Memanggil fungsi
    prediksi

```

### 4. Mencoba Model Persistence

```

1 #%% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm
   dari library sklearn
3 from sklearn import datasets # Diguankan untuk class datasets
   dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan
   memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
   iris dan mengembalikan nilainya.
6 clf.fit(X, y)             #Perhitungan nilai label
7
8 from joblib import dump, load #memanggil class dump dan load
   pada library joblib
9 dump(clf, '1174086.joblib') #Menyimpan model kedalam 1174086.
   joblib
10 hasil = load('1174086.joblib') #Memanggil model 1174086

```

### 5. Mencoba Conventions

```

1 #%% Conventions
2 import numpy as np # memanggil library numpy dan dibuat alias
   np
3 from sklearn import random_projection #Memanggil class
   random_projection pada library skean
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
   mendefisikan np, fungsi random dan attr RandomState
   kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
   nilai random dari 10 – 2000

```

```

7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
    random sebelumnya, kedalam array, dan menentukan
    typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
    membuat variabel transformer, dan mendefinisikan
    classrandom_projection dan memanggil fungsi
    GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru
    dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64

```

### 1.3.3 Penanganan Error

#### 1. ScreenShoot Error

`ImportError: cannot import name 'datasets' from 'sklearn'`

Gambar 1.13 Import Error

`ValueError: Expected 2D array, got 1D array instead:
array[1, 2, 3].
Reshape your data either using array.reshape(-1, 1) if your data has a single feature or
array.reshape(1, -1) if it contains a single sample.`

Gambar 1.14 Value Error

#### 2. Tuliskan Kode Error dan Jenis Error

- Import Error
- Value Error

#### 3. Cara Penanganan Error

- Import Error  
Dengan Menginstall Library Yang Tidak Ditemukan
- Value Error  
Mengubah Bentuk Arraynya, Menjadi 1 Dimensi

### 1.3.4 Bukti Tidak Plagiat



Gambar 1.15 Bukti Tidak Melakukan Plagiat Chapter 1

## 1.4 1174054 — Aulyardha Anindita

### 1.4.1 Teori

#### 1. Definisi, Sejarah dan Perkembangan Kecerdasan Buatan

##### ▪ Definisi Kecerdasan Buatan

Kecerdasan buatan adalah suatu kecerdasan yang didalamnya berisi suatu sistem yang biasa diatur dalam sebuah konteks ilmiah. Kecerdasan buatan juga bisa didefinisikan sebagai sebuah kecerdasan yang diciptakan dan dimasukkan kedalam suatu mesin computer agar dapat melakukan pekerjaan seperti yang dapat dilakukan oleh manusia. Ada beberapa macam bidang atau ilmu yang menggunakan kecerdasan buatan diantaranya adalah sistem pakar, permainan computer (game), logika fuzzy, jaringan saraf tiruan dan robotika.

Penelitian dalam AI mencakup pembuatan mesin dan suatu program computer untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku cerdas, seperti : pengendalian, perencanaan dan penjadwalan serta kemampuan untuk menjawab diagnose dan pertanyaan pelanggan serta pengenalan tulisan tangan. Suara dan wajah

##### ▪ Sejarah Kecerdasan Buatan

- Pada tahun 1940 dan 1950 Artificial Intelligence merupakan suatu inovasi baru dalam bidang ilmu pengetahuan dimana pada tahun ini computer modern sudah ada

- Pada tahun 1950 awal, studi tentang “mesin berfikir” mempunyai berbagai nama seperti cybernetics, teori automata, dan pemrosesan informasi

- Pada tahun 1956, para ilmuwan jenius seperti Alan Turing, Norbert Wiener, Claude Shannon dan Warren McCullough bekerja secara independen di bidang cybernetics, matematika, algoritma dan teori jaringan. John McCarthy merupakan orang yang menciptakan istilah tersebut dan mendirikan laboratorium kecerdasan buatan di MIT dan Stanford

- Pada tahun 1956, McCarthy mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Dia merupakan peneliti terkemuka dalam teori kompleksitas, simulasi Bahasa, dan hubungan antara keacakan dan pemikiran kreatif, jaringan saraf diundang. Sehingga Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan.

- Sejak saat itu, Kecerdasan Buatan telah hidup melalui decade kemuliaan dan cemohan yang dikenal dengan luas sebagai musim panas dan musim dingin Ai.

##### ▪ Perkembangan Kecerdasan Buatan

Saat ini, teknologi Artificial Intelligence sangat ramai diperbincangkan

oleh masyarakat. Sudah banyak pekerjaan yang hilang karena adanya AI, seperti pekerjaan kasir, penjaga pintu tol, parkir, dan sebagainya. Hal ini terjadi karena AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Walaupun masih ada beberapa aspek yang memang pekerja manusia masih unggul dibandingkan AI itu sendiri.

Berdasarkan survei yang dilakukan oleh Microsoft, hasilnya adalah 39 responden masih mempertimbangkan untuk menggunakan mobil tanpa pengemudi dan sebanyak 36 responden lainnya setuju bahwa robot atau AI dengan menggunakan software untuk beroperasi mampu meningkatkan produktivitas. Dari survei tersebut, dapat ditarik kesimpulan bahwa pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tidak memiliki efek samping terhadap profuktifitas kerja dan keseharian sebagai pengguna dalam kehidupan sehari-hari.

## 2. Definisi Supervised Learning, Klasifikasi, Regresi, Unsupervised Learning, Data Set, Training Set dan Testing Set

### ▪ Supervised Learning

Supervised learning adalah suatu tugas pengumpulan data yang berfungsi untuk menyimpulkan fungsi dari data pelatihan yang berlabel. Didalam Supervised Learning, setiap contoh merupakan pasangan yang terdiri dari objek input dan nilai output yang diinginkan. Algoritma pembelajaran yang diawasi berupa menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan yang digunakan untuk memetakan contoh baru.

Supervised Learning adalah suatu pendekatan dimana sudah terdapat data yang dilatih selain itu juga sudah memiliki variable yang ditargetkan sehingga tujuan dari pendekatan tersebut adalah mengelompokkan suatu data ke data yang sudah ada. Supervised learning sendiri menyediakan algoritma pembelajaran dengan jumlah yang diketahui untuk mendukung penilaian dimasa depan. Supervised learning sebagian besar memiliki kaitan dengan AI dengan menggunakan model pembelajaran generatif. Data pelatihan untuk pembelajaran yang diawasi mencakup beberapa contoh dengan subjek input yang berpasangan dan output yang diinginkan.

Modul supervised learning mempunyai beberapa keunggulan dibandingkan pendekatan tanpa pengawasan, tapi mereka juga memiliki keterbatasan. System lebih cenderung membuat penilaian bahwa manusia dapat berhubungan, misalnya manusia mempunyai dasar untuk keputusan. Tapi, dalam kasus tersebut yang menggunakan metode berbasis pengambilan, supervised learning mengalami kesulitan dalam menangani suatu informasi baru.

### ▪ Klasifikasi

Klasifikasi merupakan pembagian menurut kelas-kelas. Menurut ilmu

pengetahuan, klasifikasi adalah suatu proses pengelompokan benda berdasarkan ciri-ciri persamaan dan perbedaan. Dalam pembelajaran mesin dan statistic, klasifikasi merupakan suatu pendekatan pembelajaran yang diawasi dimana program computer tersebut belajar dari input data yang diberikan kepadanya lalu menggunakan pembelajaran tersebut untuk mengklasifikasikan pengamatan baru. Kumpulan data tersebut mungkin hanya bersifat dua kelas atau mungkin juga multi-kelas.

- **Regresi**

Regresi adalah suatu metode analisis statistik yang digunakan untuk melihat pengaruh antara dua atau lebih variable. Regresi sendiri membahas masalah ketika variable output yaitu nilai ril atau berkelanjutan seperti gaji atau berat. Banyak model yang dapat digunakan, yang paling sederhana adalah regresi linear.

- **Unsupervised Learning**

Unsupervised learning berbeda dengan supervised learning, perbedaannya yaitu unsupervised learning tidak memiliki data pelatihan, sehingga data dapat dikelompokkan menjadi dua atau 3 begitupun seterusnya. Unsupervised learning adalah suatu pelatihan algoritma kecerdasan buatan (AI) menggunakan beberapa informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut. System AI disini dapat dikelompokkan berdasarkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan. System AI disajikan dengan data yang tidak berlabel, tidak terkategori dan algoritma system bekerja pada data tanpa pelatihan sebelumnya sehingga outputnya tergantung pada algoritma kode.

- **Data Set**

Data set adalah suatu objek yang merepresentasikan data dan memiliki relasi yang ada di dalam memory. Struktur data set mirip dengan data yang ada didatabase, namun bedanya data set berisi koleksi dari data table dan data relation. Untuk mendapatkan data yang tepat, berarti mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin anda prediksi.

- **Training Set**

Training set adalah salah satu set yang biasa digunakan oleh algoritma klasifikasi. Seperti decision tree, bayesian, neural network, dll. Mereka dapat digunakan untuk membentuk model classifier, dalam menjalankan pelatihan yang diatur melalui jaringan saraf yang mengajarkan pada net dengan cara menimbang berbagai fitur, menyeuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan. Kofiesen tersebut juga dikenal sebagai parameter.

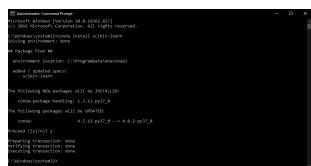
- **Testing Set**

Testing set adalah salah satu set yang digunakan untuk mengukur

sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Hal ini berfungsi sebagai materai persetujuan tapi tak digunakan sampai akhir. Setelah melatih dan mengoptimalkan data, kita dapat melakukan pengujian sarat terhadap pengambilan sampel aca. Dan hasilnya harus memvalidasi bahwa jaring data tersebut secara akurat mengenali gambar atau mengenali setidaknya ( $x$ ) dari jumlah tersebut.

### 1.4.2 Praktek

1. Instalasi library scikit dari anaconda



**Gambar 1.16** Instalasi Package Scikit Learn

| Name   | Type        | Size | Value                                |
|--------|-------------|------|--------------------------------------|
| digits | utils.Bunch | 5    | Bunch object of sklearn.utils module |
| iris   | utils.Bunch | 6    | Bunch object of sklearn.utils module |

**Gambar 1.17** Isi Variabel Explorer

2. Mencoba Loading an example dataset

```
1 #%% Mencoba loading an example dataset
2 from sklearn import datasets # Digunakan Untuk Memanggil
   class datasets dari library sklearn
3 iris = datasets.load_iris() # Menggunakan contoh datasets
   iris
4 x = iris.data           # Menyimpan nilai data sets iris
   pada variabel x
5 y = iris.target         # Menyimpan nilai data label iris
   pada variabel y
```

3. Mencoba Learning and predicting

```
1 #%%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Digunakan
   Untuk Memanggil fungsi KNeighborsClassifier
3                                     # pada
   class sklearn dan library sklearn
4 import numpy as np # memanggil library numpy dan dibuat alias
   np
```

```

5 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
   , dan memanggil fungsi KNeighborsClassifier
6                                         #dan mendefinisikan k
7 knn.fit(x,y)                         #Perhitungan
8 matematika library kkn
9 a=np.array([1.0,2.0,3.0,4.0])
10 a = a.reshape(1,-1)
    Array jadi 1 dimensi
11 hasil = knn.predict(a)                #Memanggil fungsi
    prediksi
12 print(hasil)                        #menampilkan hasil
    prediksi

```

#### 4. Mencoba Model persistence

```

1 #%% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm
   dari library sklearn
3 from sklearn import datasets # Diguangkan untuk class datasets
   dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan
   memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
   iris dan mengembalikan nilainya.
6 clf.fit(X, y)             #Perhitungan nilai label
7
8 from joblib import dump, load #memanggil class dump dan load
   pada library joblib
9 dump(clf, '1174054.joblib') #Menyimpan model kedalam 1174054.
   joblib
10 hasil = load('1174054.joblib') #Memanggil model 1174054
11 print(hasil) # Menampilkan Model yang dipanggil sebelumnya

```

#### 5. Mencoba Conventions

```

1 #%% Conventions
2 import numpy as np # memanggil library numpy dan dibuat alias
   np
3 from sklearn import random_projection #Memanggil class
   random_projection pada library sklean
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
   mendefinisikan np, fungsi random dan attr RandomState
   kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
   nilai random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
   random sebelumnya, kedalam array , dan menentukan
   typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9

```

```

10 transformer = random_projection.GaussianRandomProjection() # membuat variabel transformer, dan mendefinisikan class random_projection dan memanggil fungsi GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64
13 print(X_new) # Menampilkan isi variabel X_new

```

### 1.4.3 Penanganan Error

#### 1. ScreenShoot Error

```

File "D:/Mata Kuliah/Tingkat 3/Semester 6/Kecerdasan Buatan/Chapter 1/1174054.py", line 33, in <module>
    from joblib import dump, load #mengambil class dump dan load
    pada library joblib
ModuleNotFoundError: No module named 'joblib'

```

**Gambar 1.18** Module Not Found Error

```

File "D:/Mata Kuliah/Semester 6/Kecerdasan Buatan/DAIkecilLearn.py", line 8, in <module>
    from sklearn import datasets
ImportError: cannot import name 'datasets' from 'sklearn' (D:\Kuliah\Semester 6\kecerdasan buatan\DAIkecilLearn.py)

```

**Gambar 1.19** Import Error

#### 2. Tuliskan Kode Error dan Jenis Error

- Module Not Found Error
- Import Error

#### 3. Cara Penanganan Error

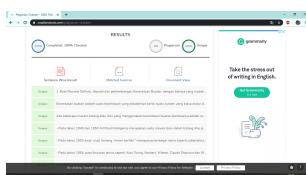
- Module Not Found Error

Dengan memperbaiki penulisan atau kesalahan dalam penulisan kode atau melakukan install package atau modul yang belum terinstal

- Import Error

Dengan Menginstall Library Yang Tidak Ditemukan

### 1.4.4 Bukti Tidak Plagiat



**Gambar 1.20** Bukti Plagiarisme

#### 1.4.5 Link Youtube

<https://youtu.be/gI9Q60DzEfI>

### 1.5 Ainul Filiani 1174073

#### 1.5.1 Pengertian Kecerdasan Buatan

Kecerdasan buatan adalah salah satu cabang ilmu pengetahuan yang berhubungan dengan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal ini biasanya dilakukan dengan mengikuti karakteristik dan analogi berpikir dari kecerdasan atau inteligance manusia, dan menerapkan sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibilitas dan efisien dapat diambil tergantung keperluan yang mempengaruhi bagaimana wujud dari prilaku kecerdasan buatan. AI biasanya dihubungkan dengan ilmu komputer, akan tetapi juga terkait erat dengan bidang lain seperti matematika, Psikologi, Pengamatan, Biolog, filosofi, dan lainnya

#### 1.5.2 Sejarah Kecerdasan Buatan

Kecerdasan buatan merupakan bidang ilmu komputer yang sangat penting di era kini dan masa yang akan datang untuk mewujudkan sistem komputer yang cerdas. Bidang ini telah berkembang sangat pesat di 20 tahun terakhir seiring dengan kebutuhan perangkat cerdas pada industry dan rumah tangga. Kata Intelligence berasal dari bahasa latin "intelligo" yang berarti "saya paham". Berarti dasar dari intelligence adalah kemampuan untuk memahami dan melakukan aksi. nyatanya, bidang Kecerdasan Buatan atau disingkat dengan AI, berawal dari kemunculan komputer sekitar tahun 1940-an, sedangkan perkembangan sejarah dapat ditelusuri sejak zaman Mesir kuno. Pada saat ini, perhatian mendesak diberikan pada kemampuan komputer untuk melakukan hal-hal yang dapat dilakukan manusia. Dalam hal ini, komputer ini dapat meningkatkan kemampuan kecerdasan dan kecerdasan manusia. Pada awal abad ke-17, René berbicara tentang tubuh binatang yang tidak meminta apa pun selain mesin yang rumit. Blaise Pascal membuat mesin hitung digital mekanis pertama pada tahun 1642. Pada 19, Charles Babbage dan Ada Lovelace bekerja pada mesin hitung mekanis yang dapat diprogram. Bertrand Russell dan Alfred Whitehead North menerbitkan Principia Mathematica, yang merombak logistik formal. Warren McCulloch dan Walter Pitts menerbitkan "Kalkulus Logika Gagasan yang Menjaga Aktivitas" pada tahun 1943 yang membentuk dasar bagi jaringan saraf. 1950-an adalah periode upaya aktif dalam AI. program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy menciptakan istilah "kecerdasan buatan" pada konferensi pertama yang menjadi dasar perjanjian itu, pada tahun 1956. Dia

juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan "tes Turing" sebagai cara untuk mengoperasionalkan tes kecerdasan cerdas.

### 1.5.3 Perkembangan dan Penggunaan Kecerdasan

Menurut studi Harvard Business Review dan ICM Unlimited pada tahun 2016, perusahaan besar memberikan kompensasi 10 persen lebih tinggi untuk setiap karyawan, Terrelong melanjutkan pengembangan Artificial Intelligence (AI) tidak hanya untuk membuat gambar atau video palsu lebih mudah, tetapi juga membuatnya sulit untuk membuktikannya. Meskipun pada saat ini, upaya untuk membuat dan mendistribusikan konten hoax, alias hoaks, masih dapat diatasi, tetapi berhasil, tantangan yang dihadapi semakin sulit. Selain itu, AI memungkinkan pembuatan gambar, video, atau audio palsu dari bahan yang relatif minim. Moody's, yang harus disetujui, membuktikan upaya itu akan semakin menantang dan membutuhkan teknik forensik yang lebih canggih. Pada Mei 2019, para peneliti di Samsung AI Center dan Institut Sains dan Teknologi Skolkovo di Moskow, Rusia menunjukkan bahwa mereka dapat membuat tayangan video yang menampilkan masing-masing individu. Video ini sangat realistik tetapi sebenarnya palsu, dibuat menggunakan model pembelajaran tertentu yang disebut Generative Adversarial Network (GAN). Hasil dari proses GAN disebut deepfakes karena mereka menggunakan teknik pembelajaran yang mendalam untuk membuat konten palsu. Untuk jangka pendek, perusahaan diharapkan untuk terus memainkan media sosial dan situs untuk melihat pentingnya disinformasi dan meminta mereka yang bertanggung jawab untuk media sosial dan situs terkait untuk mengunduh konten. Terrelong menambahkan langkah lain yang bisa diam-bil untuk merilis materi resmi untuk melawan konten palsu."Perlawanan terhadap konten palsu membutuhkan kombinasi teknologi dan pendidikan,".

## 1.6 resume mengenai definisi supervised learning, klarifikasi, regresi, dan un-supervised learning. Data Set, training set dan testing set

### 1.6.1 Supervised Learning

Supervised Learning adalah tugas mengumpulkan data untuk melengkapi fungsi data pelatihan yang diberi label. Data pelatihan terdiri dari contoh pelatihan. Dalam pembelajaran terawasi, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output dingin (juga disebut sinyal pengawasan super). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang lengkap, yang dapat digunakan untuk memetakan contoh-contoh baru. Skenario akan memungkinkan algoritma menentukan label kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi

data pelatihan sehingga tidak muncul dengan cara yang "masuk akal". Pembelajaran terawasi semakin dekat di mana ada pelatihan praktis selain dapat bervariasi yang berarti tujuannya adalah di mana mengelompokkan data ke dalam database yang ada. Pembelajaran terawasi menyediakan jumlah pembelajaran yang direkomendasikan untuk mendukung penilaian di masa depan. Obrolan, program mengemudi mandiri, pengenalan wajah, tatap muka dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak diawasi. Pembelajaran terbimbing sebagian besar terkait dengan AI berdasarkan pengambilan mereka juga mungkin diperlukan menggunakan model pembelajaran generatif. Pelatihan data untuk pembelajaran dimulai dengan mendiskusikan contoh-contoh dengan subjek input berpasangan dan output yang diinginkan (juga disebut sebagai sinyal pengawasan). Dalam pembelajaran yang diawasi untuk pemrosesan gambar, misalnya sistem AI dapat lengkap dengan gambar mengemudi yang berlabel dalam kategori mobil dan truk. Setelah jumlah yang memadai, sistem harus dapat membedakan antara dan mengklasifikasikan gambar yang tidak berlabel, di mana waktu pelatihan dapat diselesaikan secara penuh. Model Pembelajaran Terpandu memiliki beberapa keunggulan dibandingkan pengawasan, tetapi mereka juga memiliki keterbatasan. Sistem lebih cenderung membuat penilaian bahwa hak asasi manusia dapat dihubungkan, misalnya karena manusia telah memberikan dasar untuk pengambilan keputusan. Namun, dalam hal metode berbasis pengambilan, Supervised Learning menghilangkan kesulitan dalam menangani informasi baru. Jika sistem dikategorikan untuk mobil dan truk, maka sepeda disediakan, misalnya, harus dikelompokkan dalam satu kategori atau yang lain. Namun. Jika sistem AI generatif, mungkin tidak tahu apa itu sepeda tetapi akan dapat mengenalinya sebagai milik kategori yang terpisah.

### 1.6.2 Klasifikasi

Klasifikasi adalah pembagian hal sesuai dengan kelas (kelas). Menurut Science, klasifikasi adalah proses pengelompokan materi berdasarkan karakteristik dan perbedaan yang sama. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai yang terpisah. Label ( $y$ ) Umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran statistik dan pembelajaran mesin statistik, klasifikasi adalah pembelajaran yang dimulai ketika sebuah program komputer belajar dari input data yang disediakan untuk mendukung dan kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pembelajaran baru. Pengumpulan data ini mungkin hanya dua kelas (seperti mengidentifikasi apakah orang ini laki-laki atau perempuan atau orang itu adalah spam atau bukan-spam) atau mungkin juga multi-kelas. Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, metrik identifikasi, klasifikasi dokumen dll.

### 1.6.3 Regresi

Regresi adalah metode analisis statistik yang digunakan untuk melihat perbedaan antara dua atau lebih variabel. Regresi sedang membahas masalah kompilasi, variabel output adalah nilai nyata atau dipertahankan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan untuk makan, cara paling sederhana adalah linearitas linear. Itu mencoba untuk mencocokkan data dengan pesawat-hyper terbaik yang melewati titik.

### 1.6.4 unsupervised learning

Belajar tanpa pengawasan berbeda dari Belajar dengan Supervisi. Perbedaannya adalah bahwa pembelajaran tanpa pengawasan tidak memiliki data pelatihan, jadi dari data yang tersedia kami mengelompokkan data menjadi 2 atau 3 bagian dan seterusnya. Unsupervised Learning adalah pelatihan dalam algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasi atau diberi label dan menyediakan algoritma untuk memperbaiki informasi yang diberikan tanpa bimbingan. Dalam Unattended Learning, sistem AI dapat mengklasifikasikan informasi yang tidak diurutkan berdasarkan ekuitas dan perbedaan dalam kategori mendadak yang disediakan. Dalam Supervised Learning Learning, sistem AI disajikan dengan sistem wajib yang tidak diberi label, tidak dikategorikan dan algoritma bekerja pada data tanpa pelatihan sebelumnya. Outputnya tergantung pada algoritma kode. Menyerahkan sistem untuk Belajar Tanpa Pengawasan adalah salah satu cara untuk menerima AI. Algoritma Pembelajaran tanpa pengawasan dapat melakukan tugas yang lebih kompleks daripada sistem pembelajaran yang diawasi. Namun, pembelajaran tanpa pengawasan dapat lebih tidak konsisten dengan model alternatif. Sementara Supervised Learning Mungkin, misalnya, mencari sendiri dengan memilih kucing dari anjing, ia juga dapat menambahkan kategori yang tidak diinginkan dari yang tidak diinginkan untuk ditingkatkan menjadi ras yang tidak biasa, membuat pesanan diperlukan.

### 1.6.5 Data Set

Dataset adalah objek yang mewakili data dan hubungan dalam memori. Strukturnya mirip dengan basis data basis data, tetapi hanya kumpulan data yang dikumpulkan dari catatan dan latar belakang yang diaktifkan. dapatkan persetujuan yang tepat untuk mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda hasilkan; yaitu data yang berisi sinyal tentang acara yang Anda sukai. Data harus disinkronkan dengan masalah yang Anda coba selesaikan. Gambar kucing bukan kompilasi yang sangat berguna. Anda sedang membangun sistem identifikasi wajah. Memodifikasi data yang selaras dengan masalah yang ingin Anda selesaikan harus dilakukan oleh para ahli data. Jika Anda tidak memiliki data yang benar, maka upaya Anda untuk membuat solusi AI harus kembali ke in-

stalasi data. Format ujung kanan untuk belajar secara umum adalah array tensor, atau multi-dimensional. Jadi pipa data yang dibangun untuk pembelajaran dibangun secara umum untuk mengubah semua gambar, video, suara, suara, teks atau deret waktu menjadi vektor dan tensor yang dapat digunakan operasi aljabar linier. Data yang diperlukan perlu dinormalisasi, distandarisasi dan dikembalikan untuk meningkatkan kegunaannya, dan semua ini adalah langkah-langkah dalam pembelajaran mesin ETC. Deeplearning4j menawarkan alat ETV Data Vec untuk melakukan tugas memfasilitasi data. Pembelajaran yang mendalam, dan pembelajaran mesin yang lebih umum, membutuhkan pelatihan yang baik agar dapat bekerja dengan baik. Mengumpulkan dan membangun satu set badan pelatihan yang cukup besar dari data yang diketahui membutuhkan waktu dan pengetahuan khusus tentang pengetahuan dan cara untuk mengumpulkan informasi yang relevan. Perangkat pelatihan bertindak sebagai patokan terhadap mana jaring pembelajaran dalam pengeboran. Itulah yang mereka perbarui untuk direkonstruksi sebelum mereka merilis data yang belum pernah dilihat sebelumnya. Pada saat ini, manusia memiliki pengetahuan luas tentang mengidentifikasi instrumen yang tepat dan mengubahnya menjadi representasi numerik yang dapat dipahami oleh algoritma pembelajaran dalam, tensor. Membangun set pelatihan, dalam arti tertentu, pra-pelatihan. Kumpulan pelatihan yang membutuhkan banyak waktu atau keahlian yang dapat membantu dalam dunia data dan pemecahan masalah. Sifat keahlian terbesar Anda dalam memberi tahu algoritma Anda apa yang penting bagi Anda adalah memilih apa yang Anda masukkan dalam kursus pelatihan Anda. Ini melibatkan menceritakan kisah melalui data awal yang Anda pilih untuk memandu proses pembelajaran mendalam Anda dengan mengekstraksi fitur-fitur penting, baik dalam pengaturan pelatihan dan data yang ingin Anda buat untuk dipelajari. Agar pelatihan ini bermanfaat, Anda harus memecahkan masalah yang Anda selesaikan; yaitu, apa yang Anda inginkan agar sesuai dengan pembelajaran Anda, di mana hasil yang ingin Anda prediksi.

### 1.6.6 Training Set

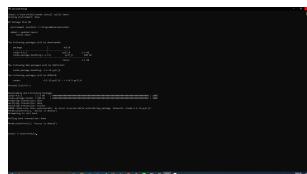
Set Pelatihan adalah set yang digunakan oleh algoritma klasifikasi. Dapat dicontohkan oleh: decisiontree, bayesian, neural network dll. Semuanya dapat digunakan untuk membuat model kelas. Terkait dengan pelatihan yang mengatur melalui jaringan saraf di internet bagaimana menimbang berbagai fitur, sesuaikan koefisien sesuai dengan apa yang mereka tingkatkan dalam hasil Anda. Koefisien, juga dikenal sebagai parameter, akan terkandung dalam sensor dan bersama-sama mereka disebut model, model data karena mereka menyandikan latihan yang mereka praktekkan.

### 1.6.7 testing Set

Tes ini digunakan untuk mengukur sejauh mana classifil berhasil mengklasifikasi dengan benar. Ini digunakan sebagai meterai persetujuan, dan Anda tidak dapat digunakan sampai akhir. Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda untuk mengambil sampel acak akhir ini. Hasilnya harus memvalidasi gambar bersih Anda, atau gambar mengenali [x] dari nomor itu. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan Anda, lihat mitra Anda yang Anda gunakan untuk mengelola jaringan Anda, dan kualitas data Anda dan lihat teknik pra-pemanfaatan yang dapat Anda gunakan.

### 1.6.8 Instalasi

- Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer



**Gambar 1.21** Instalasi Package Scikit Learn

| Name   | Type       | Size | Value                                |
|--------|------------|------|--------------------------------------|
| digits | util.Bunch | 5    | Bunch object of sklearn.utils module |
| iris   | util.Bunch | 6    | Bunch object of sklearn.utils module |

**Gambar 1.22** Isi Variabel Explorer

- Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%% Mencoba loading an example dataset
2 from sklearn import datasets # Digunakan Untuk Memanggil
   class datasets dari library sklearn
3 iris = datasets.load_iris() # Menggunakan contoh datasets
   iris
4 x = iris.data           # Menyimpan nilai data sets iris
   pada variabel x
5 y = iris.target          # Menyimpan nilai data label iris
   pada variabel y

```

- Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Digunakan Untuk Memanggil fungsi KNeighborsClassifier
3                                         # pada
4 class sklearn dan library sklearn
5 import numpy as np # memanggil library numpy dan dibuat alias np
6 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn , dan memanggil fungsi KNeighborsClassifier
7                                         #dan mendefinisikan k -nya adalah 1
8 knn.fit(x,y)                         #Perhitungan matematika library kkn
9 a=np.array([1.0,2.0,3.0,4.0])         #Membuat Array
10 a = a.reshape(1,-1)                  #Mengubah Bentuk
11     Array jadi 1 dimensi
12 hasil = knn.predict(a)               #Memanggil fungsi prediksi

```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm dari library sklearn
3 from sklearn import datasets # Digunakan untuk class datasets dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset iris dan mengembalikan nilainya .
6 clf.fit(X, y)             #Perhitungan nilai label
7
8 from joblib import dump, load #memanggil class dump dan load pada library joblib
9 dump(clf, '1174073.joblib') #Menyimpan model kedalam 1174069. joblib
10 hasil = load('1174073.joblib') #Memanggil model 1174069

```

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %% Conventions
2 import numpy as np # memanggil library numpy dan dibuat alias np
3 from sklearn import random_projection #Memanggil class random_projection pada library sklearn
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan mendefinisikan np, fungsi random dan attr RandomState kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan nilai random dari 10 – 2000

```

```

7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
     random sebelumnya, kedalam array, dan menentukan
     typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
      membuat variabel transformer, dan mendefinisikan
      classrandom_projection dan memanggil fungsi
      GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru
      dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64

```

### 1.6.9 Penanganan Error

1. ScreenShoot Error

```

file "D:\Kuliah\Semester 6\Kecerdasan Buatan\Dataset\sklearn.py", line 8, in <module>
    from sklearn import datasets
ImportError: cannot import name 'datasets' from 'sklearn' (D:\Kuliah\Semester
6\Kecerdasan Buatan\Dataset\sklearn.py)

```

**Gambar 1.23 Import Error**

2. Tuliskan Kode Error dan Jenis Error

- Import Error

3. Cara Penangan Error

- Import Error  
Dengan Menginstall Library Yang Tidak Ditemukan

### 1.6.10 Bukti Tidak Plagiat



**Gambar 1.24**   Bukti Tidak Melakukan Plagiat Chapter 1

### 1.6.11 Link Youtube

## 1.7 Chandra Kirana Poetra (1174079)

### 1.7.1 Teori

#### 1. Definisi Kecerdasan buatan

Dalam bidang komputer, Artificial Intelligence (AI), atau biasa disebut juga sebagai Machine Intelligence merupakan bentuk dari representasi kecerdasan yang dilakukan oleh mesin, hampir mirip seperti bagaimana manusia melakukan kecerdasan. Beberapa sumber mendefinisikan bahwa bidang yang mempelajari suatu agen kecerdasan merupakan suatu alat yang mengenali lingkungan sekitarnya dan mencoba untuk membuat kesimpulan untuk memaksimalkan kemungkinan tingkat keberhasilan dari pencapaian yang ingin dituju.

#### 2. Sejarah dan Perkembangan Kecerdasan Buatan

- Pada tahun 1943, pekerjaan pertama yang dikenal sebagai AI telah dilakukan oleh Warren McCulloch dan juga Walter Pitts yang dinamakan sebagai artificial neurons
- Pada tahun 1955, Allen Newell dan Herbert A. Simon membuat program kecerdasan buatan pertama yang dinamakan Logic Theorist
- Pada tahun 1972, robot pertama dibuat di Jepang dengan nama Wabot-1 dengan kecerdasan buatan
- Pada tahun 1980, muncul bidang baru dari kecerdasan buatan yaitu Expert System yang membantu dalam pemberian keputusan
- Tahun 1997, IBM Deep Blue mengalahkan juara catur dunia Gary Kasparov dan menjadi komputer pertama yang mengalahkannya
- Tahun 2006, perusahaan sudah mulai menerapkan kecerdasan buatan pada produknya seperti Netflix dan Twitter.
- Tahun 2018, Project Debater dari IBM melakukn debat tentang topik yang kompleks dan berakhir dengan hasil memuaskan

#### 3. Definisi Supervised Learning

Supervised Learning adalah proses untuk melatih mesin secara input dan output melalui contoh nyata secara langsung

#### 4. Klasifikasi Supervised Learning

- Support Vector Machines
- linear regression
- logistic regression
- naive Bayes

- linear discriminant analysis
- decision trees
- k-nearest neighbor algorithm
- Neural Networks (Multilayer perceptron)
- Similarity learning

## 5. Regresi dan Unsupervised Learning

Regresi adalah suatu proses statistikal yang mengestimasi hubungan antara variable satu dengan variable yang lainnya.

Unsupervised Learning adalah bentuk dari machine learning yang mencari bentuk atau hubungan dari data set yang tidak mempunyai label dengan bantuan yang minimal dari manusia.

## 6. Dataset

Dataset adalah koleksi suatu data

## 7. Training Set

Training Set merupakan data yang digunakan untuk keperluan pembelajaran yang biasanya digunakan oleh machine learning

## 8. Testing Set

Testing set adalah data yang real yang digunakan untuk melatih machine learning

### 1.7.2 Instalasi

#### 1. Instalasi Library scikit dari a naconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

```

$ conda install -c anaconda scikit-learn
Collecting package metadata (current_repodata.json): done
Solving environment: done
++ Package Plan ++
  environment location: C:\Users\user\Anaconda3
  added / updated specs:
    - scikit-learn

The following packages will be downloaded:
  package          build
  conda-4.8.2           py37_0      2.8 MB
                                         Total:      2.8 MB

The following packages will be UPDATED:
  conda           4.7.12-py37_0 => 4.8.2-py37_0

Proceed ([y]/n)? y

downloading and extracting Packages:
  conda-4.8.2           2.8 MB |████████████████████████████████| 100%
  verifying transaction: done
  writing transaction file: done
  installing transaction file: done
[2020-01-10 11:14:41] CONDA-GLOBAL-MODIFY=1

```

**Gambar 1.25** Instalasi Package Scikit Learn

| Name   | Type        | Size | Value                                |
|--------|-------------|------|--------------------------------------|
| digits | utils.Bunch | 5    | Bunch object of sklearn.utils module |
| iris   | utils.Bunch | 6    | Bunch object of sklearn.utils module |

**Gambar 1.26** Isi Variabel Explorer

2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %%Loading an example dataset
2 from sklearn import datasets # Load library dataset
3 iris = datasets.load_iris()
4 # variable iris diisi dengan contoh data
5 a = iris.data # Menyimpan value data ke variable A
6 b = iris.target # Menyimpan value data ke variable B

```

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan perbaris

```

1 %% Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier
3 #Load library
4 import numpy as np
5 #load library
6
7 knn = KNeighborsClassifier(n_neighbors=1)
8 #mendefinisikan variabel bernama kkn, dan memanggil fungsi
KNeighborsClassifier
9 # dan memberikan value 1
10 knn.fit(a,b) # perhitungan library knn
11
12 x = np.array([1.0 ,2.0 ,3.0 ,4.0])
13 # membuat array
14 x = x.reshape(1,-1)
15 #Convert array menjadi 1 dimensi
16
17 hasil = knn.predict(x)
18 #Memanggil fungsi predict dari KNN
19 print(hasil)
20 #menampilkan value dari variable hasil

```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %% Model Persistense
2 from sklearn import svm
3 # Load library
4 from sklearn import datasets
5 # Load Library
6 clf = svm.SVC()
7 # mendefinisikan variabel clf, dan memanggil fungsi SVC dari
class svm
8 a, b = datasets.load_iris(return_X_y=True)
9 #Variable a dan b diisi dengan dataset iris dan mengembalikan
nilainya.
10 clf.fit(a, b)
11 #memanggil fungsi fit dari clf
12

```

```

13 from joblib import dump, load
14 #Load library
15 dump(clf, '1174079.joblib')
16 #Menyimpan model kedalam 1174079.joblib
17 hasil = load('1174079.joblib')
18 #memuat model 1174079
19 print(hasil) # Menampilkan Hasil

```

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %% Conventions
2 import numpy as np
3 # Load Library
4 from sklearn import random_projection
5 #Load class random_projection dari library sklearn
6
7 rng = np.random.RandomState(0)
8 #Membuat variabel rng, dan mendefinisikan np, fungsi random dan
     attr RandomState kedalam variabel
9 X = rng.rand(10, 2000)
10 # membuat variabel X, dan menentukan nilai random dari 10 –
      2000
11 X = np.array(X, dtype='float32')
12 #menyimpan hasil nilai random sebelumnya, kedalam array , dan
      menentukan typedatanya sebagai float32
13 X.dtype
14 # Mengubah data tipe menjadi float64
15
16 transformer = random_projection.GaussianRandomProjection()
17 #membuat variabel transformer, dan mendefinisikan
      classrandom_projection dan memanggil fungsi
      GaussianRandomProjection
18 X_new = transformer.fit_transform(X)
19 # membuat variabel baru dan melakukan perhitungan label pada
      variabel X
20 X_new.dtype
21 # Mengubah data tipe menjadi float64
22 print(X_new)
23 # Menampilkan isi variabel X_new

```

### 1.7.3 Penanganan Error

1. ScreenShoot Error

```

-----  

exec(compile(f.read(), filename,  

'exec'), namespace)  

File "F:/Poltekpos/D4 TI 3C/Semester 6/  

Kecerdasan Buatan/Github/Upload 1 Naret  

2020/src/src/1174079/1174079.py", line 51, in  

<module>  

    import numpy as np  

ModuleNotFoundError: No module named  

'numpy'  

To run this file in Python 3, you must type:  


```

**Gambar 1.27** No Module Named Numpya

## 2. Tuliskan Kode Error dan Jenis Error

- ModuleNotFoundError

## 3. Cara Penangan Error

- ModuleNotFoundError

Mengecek Typo dan menulis kembali library yang akan diimport

### 1.7.4 Bukti Tidak Plagiat



**Gambar 1.28** Bukti Tidak Melakukan Plagiat Chapter 1

### 1.7.5 Link Youtube

<https://youtu.be/nPua0lRXjO8>

## 1.8 D. Irga B. Naufal Fakhri

### 1.8.1 Teori

**1.8.1.1 Definisi Kecerdasan Buatan** Kecerdasan Buatan atau yang sering disebut AI (Artificial Intelligence) merupakan suatu cabang didalam bisnis sains dan komputer sains yang didalamnya membahas tentang bagaimana caranya untuk membuat sebuah komputer dengan kemampuan atau ke pintaran layaknya atau mirip dengan yang dimiliki manusia. Contohnya, bagaimana komputer bisa berkomunikasi dengan pengguna baik menggunakan kata, suara ataupun yang lainnya. Dengan kemampuan ini, diharapkan komputer dapat

mengambil keputusan dengan sendirinya untuk memecahkan berbagai kasus yang ditemuinya kemudian itulah yang disebut dengan kecerdasan buatan. Kecerdasan buatan adalah kemampuan komputer digital atau robot yang dikendalikan komputer untuk melakukan tugas yang umumnya dikaitkan dengan sesuatu yang cerdas. Istilah ini sering diterapkan pada proyek pengembangan sistem yang diberkahi dengan karakteristik proses intelektual manusia, seperti kemampuan untuk berpikir, menemukan makna, menggeneralisasi, atau belajar dari pengalaman masa lalu.

Kecerdasan Buatan (AI) merupakan salah satu bidang yang sangat berhubungan dengan memanfaatkan mesin (komputer) untuk memecahkan suatu masalah atau persoalan yang rumit dengan cara yang lebih mudah dimengerti oleh manusia. Kecerdasan Buatan (AI) yang semakin canggih yang mampu menambahkan pengetahuannya dengan cara melakukan banyak testing dan perkembangan dari target yang di analisa. Contoh dari kecerdasan buatan yang paling terkenal saat ini adalah Google Assistant, Alexa dan Siri. Google Assistant, Alexa dan Siri sangat dikenal karena penggunaannya yang mudah oleh user untuk menemukan berbagai hal atau membuatnya untuk melakukan sesuatu terhadap smartphone atau smarthome anda dan contohnya masih banyak lagi.

**1.8.1.2 Sejarah Kecerdasan Buatan** Kecerdasan Buatan (Artificial intelligence) mulai dibentuk sejak adanya komputer modern yang diperkirakan terjadi pada tahun 1940 dan 1950. Ilmu pengetahuan kecerdasan buatan ini dikhawasukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer.

Pada awal 50-an, studi tentang “mesin berpikir” memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan informasi. Pada tahun 1956, para ilmuan jenius seperti Alan Turing, Norbert, Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen dibidang cybernetics, matematika, algoritma dan teori jaringan. Namun, seprang ilmuan komputer dan kognitif John McCarthy adalah orang yang dating dengan ide untuk bergabung dengan upaya penelitian terpisah ini kedalam satu bidang yang akan mempelajari topic baru untuk imajinasi manusia yaitu kecerdasan buatan. Dia adalah orang yang menciptakan istilah tersebut dan kemudian mendirikan laboratorium Kecerdasan Buatan di MIT dan Stanford. Pada tahun 1956, McCarthy yang sama mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Peneliti terkemuka dalam teori kompleksitas, simulasi bahasa, hubungan antara keacakan dan pemikiran kreatif, jaringan saraf diundang. Tujuan dari bidang penelitian yang baru dibuat adalah untuk mengembangkan mesin yang dapat mensimulasikan setiap aspek kecerdasan. Itulah sebabnya Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan. Sejak saat itu, Kecerdasan Buatan telah hidup melalui dekade kemuliaan dan cemoohan, yang dikenal luas sebagai musim panas dan musim dingin AI. Musim panasnya ditandai dengan optimisme dan dana be-

sar, sedangkan musim dinginnya dihadapkan dengan pemotongan dana, ketidakpercayaan dan pesimisme.

**1.8.1.3 Perkembangan Kecerdasan Buatan** Teknologi Artificial Intelligence semakin ramai dibahas dalam berbagai diskusi teknologi di seluruh dunia. Menurut kebanyakan orang, pekerjaan seperti kasir, operator telepon, pengendara truk, dan lainnya sangat berpeluang besar untuk tergantikan oleh Artificial Intelligence. Mengapa terjadi hal demikian? dikarenakan memang bahwa AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Namun, dalam beberapa aspek memang pekerja manusia masih unggul dibandingkan AI itu sendiri. Para generasi muda yang ada di dunia terutama di daerah Asia terlihat sudah memahami fungsi dan efek dari AI dalam kehidupan kita sehari-hari. Berdasarkan survei yang dilakukan oleh Microsoft, terdapat 39 persen responden yang mempertimbangkan untuk menggunakan mobil tanpa pengeudi dan 36 persen lainnya setuju bahwa robot masa depan dengan software untuk beroperasi mampu meningkatkan produktivitas. Dari survey tersebut kita sebagai pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tanpa memberikan efek samping terhadap etos kerja dan keseharian kita sebagai pengguna dalam kehidupan sehari-hari.

AI Summer 1 (1956-1973) KOnferensi Dartmounth diikuti oleh 17 tahun kemajuan luar biasa. Proyek penelitian yang dilakukan di MIT, universitas di Edinburgh, Stanford dan Carnegie Mellon menerima dana besar-besaran, yang akhirnya membawa hasil. Selama tahun-tahun itulah komputer pemrograman mulai melakukan masalah aljabar, membuktikan teorema geometris, memahami dan menggunakan sintaks dan tata bahasa Inggris. Terlepas dari ditinggalkannya koneksiisme dan terjemahan mesin yang gagal, yang menunda penelitian Natural Language Processing (NLP) selama bertahun-tahun, banyak prestasi dari masa lalu yang membuat sejarah. Berikut ini beberapa diantaranya : Pelopor pembelajaran mesin, Ray Solomonoff meletakkan dasar-dasar teori metematika AI, memperkenalkan metode Bayesian universal untuk inferensi dan prediksi induktif Thomas Evans menciptakan program ANALOGI heuristik, yang memungkinkan komputer memecahkan masalah geometri-analogi Unimation, perusahaan robotika pertama didunia, menciptakan robot industri Unimate, yang bekerja pada jalur perakitan modil General Motors. Joseph Weizenbaum membangun ELIZA-program interaktif yang dapat membawa percakapan dalam bahasan Inggris tentang topik apapun. Ross Quillian menunjukkan jaring semantik, sedangkan Jaime Carbonell (Sr.) mengembangkan Cendikia-program interaktif untuk instruksi yang dibantu komputer berdasarkan jaring semantik. Edward Feigenbaum dan Julian Feldman menerbitkan Computeks and Thought, kumpulan artikel pertama tentang AI.

**1.8.1.4 Supervised Learning** Supervised Learning adalah tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya

vektor) dan nilai output yang diinginkan(juga disebut sinyal pengawasan super). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan, yang dapat digunakan untuk memetakan contoh-contoh baru. Skenario optimal akan memungkinkan algoritma menentukan label kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi dari data pelatihan untuk situasi yang tidak terlihat dengan cara yang "masuk akal". Supervised Learning adalah pendekatan dimana sudah terdapat data yang dilatih selain itu juga terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini yaitu mengelompokkan suatu data ke dta yang sudah ada. Supervised Learning menyediakan algoritma pembelajaran dengan jumlah yang diketahui untuk mendukung penilaian dimasa depan. Chatbots, mobil self-driving, program pengenalan wajah, sistem pakar dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak diawasi. Supervised Learning sebagian besar terkait dengan AI berbasis pengambilan tetapi mereka juga mungkin mampu menggunakan model pembelajaran generatif. Data pelatihan untuk pembelajaran yang diawasi mencakup serangkaian contoh dengan subjek input berpasangan dan output yang diinginkan (yang juga disebut sebagai sinyal pengawasan).

Dalam pembelajaran yang diawasi untuk pemrosesan gambar, misalnya sistem AI mungkin dilengkapi dengan gambar berlabel kendaraan dalam kategori seperti mobil dan truk. Setelah jumlah pengamatan yang cukup, sistem harus dapat membedakan antara dan mengkategorikan gambar yang tidak berlabel, dimana waktu pelatihan dapat dikatakan lengkap. Model Supervised Learning memiliki beberapa keunggulan dibandingkan pendekatan tanpa pengawasan, tetapi mereka juga memiliki keterbatasan. Sistem lebih cenderung membuat penilaian bahwa manusia dapat berhubungan, misalnya karena manusia telah memberikan dasar untuk keputusan. Namun, dalam kasus metode berbasis pengambilan, Supervised Learning mengalami kesulitan dalam menangani informasi baru. Jika suatu sistem dengan kategori untuk mobil dan truk disajikan dengan sepeda, misalnya ia harus salah dikelompokkan dalam satu kategori ata yang lain. Namun, jika sistem AI bersifat generatif, ia mungkin tidak tahu apa sepeda itu tetapi akan dapat mengenalinya sebagai milik kategori yang terpisah.

**1.8.1.5 Klasifikasi** Klasifikasi adalah pembagian sesuatu menurut kelas-kelas ( class ). Menurut Ilmu Pengetahuan, Klasifikasi merupakan proses pengelompokan benda berdasarkan ciri-ciri persamaan dan juga perbedaan. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah. Label (y) umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran mesin dan statistik, klasifikasi adalah pendekatan pembelajaran yang diawasi di mana program komputer belajar dari input data yang diberikan kepadanya dan kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pengamatan baru. Kumpulan data ini mungkin hanya bersifat dua kelas (seperti mengidentifikasi apakah orang tersebut berjenis ke-

lamin laki-laki atau perempuan atau bahwa surat itu spam atau bukan-spam) atau mungkin juga multi-kelas. Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, identifikasi metrik, klasifikasi dokumen dll.

**1.8.1.6 Regresi** Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel. Regresi adalah membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan makan, yang paling sederhana adalah regresi linier. Ia mencoba untuk menyesuaikan data dengan hyper-plane terbaik yang melewati poin.

**1.8.1.7 Unsupervised Learning** Unsupervised Learning berbeda dengan Supervised Lerning. Perbedaannya ialah unsupervised learning tidak memiliki data latih, sehingga dari data yang ada kita mengelompokan data tersebut menjadi 2 ataupun 3 bagian dan seterusnya. Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Dalam Unsupervised Learning, sistem AI dapat mengelompokkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan.

Dalam Unsupervised Learning, sistem AI disajikan dengan data yang tidak berlabel, tidak terkategorisasi dan algoritma sistem bekerja pada data tanpa pelatihan sebelumnya. Outputnya tergantung pada algoritma kode. Menundukkan suatu sistem pada Unsupervised Learning adalah salah satu cara untuk menguji AI. Algoritma Unsupervised Learning dapat melakukan tugas pemrosesan yang lebih kompleks daripada sistem pembelajaran yang diawasi. Namun, pembelajaran tanpa pengawasan bisa lebih tidak terduga dari pada model alternatif. Sementara Unsupervised Learning mungkin, misalnya, mencari tahu sendiri cara memilah kucing dari anjing, mungkin juga menambahkan kategori yang tidak terduga dan tidak diinginkan untuk menangani breed yang tidak biasa, membuat kekacauan bukannya keteraturan

**1.8.1.8 Data Set** Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory. Strukturnya mirip dengan data di database, namun bedanya dataset berisi koleksi dari data table dan data relation. mendapatkan data yang tepat berarti mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda prediksi; yaitu data yang berisi sinyal tentang peristiwa yang Anda pedulikan. Data harus diselaraskan dengan masalah yang Anda coba selesaikan. Gambar kucing tidak terlalu berguna ketika Anda sedang membangun sistem identifikasi wajah. Memverifikasi bahwa data selaras dengan masalah yang ingin Anda selesaikan harus dilakukan oleh ilmuwan data. Jika Anda tidak memiliki data yang tepat, maka upaya Anda untuk membangun solusi AI harus kembali ke tahap pengumpulan data. Format ujung kanan untuk pembelajaran dalam umumnya adalah

tensor, atau array multi-dimensi. Jadi jalur pipa data yang dibangun untuk pembelajaran mendalam umumnya akan mengkonversi semua data - baik itu gambar, video, suara, suara, teks atau deret waktu menjadi vektor dan tensor yang dapat diterapkan operasi aljabar linier. Data itu seringkali perlu dinormalisasi, distandarisasi dan dibersihkan untuk meningkatkan kegunaannya, dan itu semua adalah langkah dalam ETL pembelajaran mesin. DeepLearning4j menawarkan alat ETV DataVec untuk melakukan tugas-tugas pemrosesan data tersebut.

Pembelajaran yang dalam, dan pembelajaran mesin yang lebih umum, membutuhkan pelatihan yang baik agar bekerja dengan baik. Mengumpulkan dan membangun set pelatihan badan yang cukup besar dari data yang diketahui membutuhkan waktu dan pengetahuan khusus domain tentang di mana dan bagaimana mengumpulkan informasi yang relevan. Perangkat pelatihan bertindak sebagai tolok ukur terhadap mana jaring pembelajaran dalam dilatih. Itulah yang mereka pelajari untuk direkonstruksi sebelum mereka melepaskan data yang belum pernah mereka lihat sebelumnya. Pada tahap ini, manusia yang berpengetahuan luas perlu menemukan data mentah yang tepat dan mengubahnya menjadi representasi numerik yang dapat dipahami oleh algoritma pembelajaran mendalam, tensor. Membangun set pelatihan, dalam arti tertentu, pra-pra pelatihan. Set pelatihan yang membutuhkan banyak waktu atau keahlian dapat berfungsi sebagai keunggulan dalam dunia ilmu data dan pemecahan masalah. Sifat keahlian sebagian besar dalam memberi tahu algoritma Anda apa yang penting bagi Anda dengan memilih apa yang masuk ke dalam set pelatihan. Ini melibatkan menceritakan sebuah kisah melalui data awal yang Anda pilih yang akan memandu jaring pembelajaran mendalam Anda saat mereka mengekstraksi fitur-fitur penting, baik di set pelatihan maupun dalam data mentah yang telah mereka ciptakan untuk dipelajari. Untuk membuat set pelatihan yang bermanfaat, Anda harus memahami masalah yang Anda selesaikan; yaitu apa yang Anda inginkan agar jaring pembelajaran mendalam Anda memperhatikan, di mana hasil yang ingin Anda prediksi.

**1.8.1.9 Training Set** Training Set adalah set digunakan oleh algoritma klasifikasi . Dapat dicontohkan dengan : decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier. Menjalankan pelatihan yang diatur melalui jaringan saraf mengajarkan pada net cara menimbang berbagai fitur, menyesuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan dalam hasil Anda. Koefisien-koefisien tersebut, juga dikenal sebagai parameter, akan terkandung dalam tensor dan bersama-sama mereka disebut model, karena mereka mengkodekan model data yang mereka latih. Mereka adalah takeaways paling penting yang akan Anda dapatkan dari pelatihan jaringan saraf.

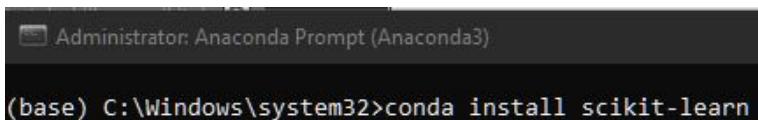
**1.8.1.10 Testing Set** Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Ini berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakaninya

sampai akhir. Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda terhadap pengambilan sampel acak akhir ini. Hasil yang dihasilkannya harus memvalidasi bahwa jaring Anda secara akurat mengenali gambar, atau mengenalinya setidaknya [x] dari jumlah tersebut. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan, lihat hyperparameter yang Anda gunakan untuk menyetel jaringan, serta kualitas data Anda dan lihat teknik pra-pemrosesan Anda.

## 1.8.2 Praktek

### 1.8.2.1 Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

1. Pastikan anda telah menginstall anaconda lalu buka aplikasi Anaconda Prompt
2. Lalu pastikan anda telah menginstall python
3. Pada Anaconda Prompt install scikit dengan cara conda install scikit-learn

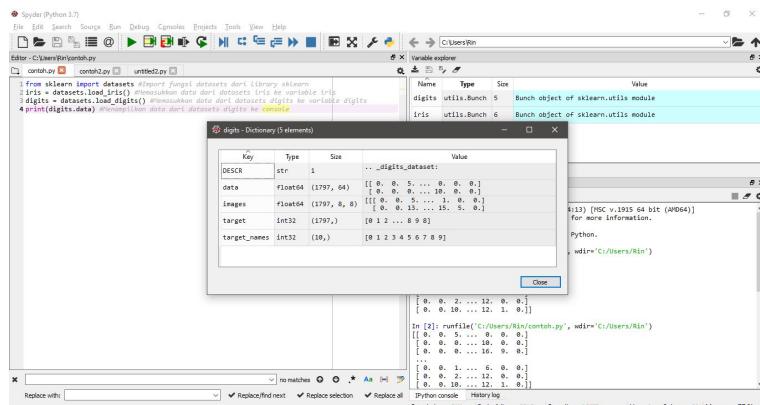


```
(base) C:\Windows\system32>conda install scikit-learn
```

Gambar 1.29 Instalasi Scikit Dari Anaconda Prompt

4. Lalu tulis kode yang ada dibawah ini dan run menggunakan spyder

```
1 from sklearn import datasets #Import fungsi datasets dari
   library sklearn
2 iris = datasets.load_iris() #Memasukkan data dari datasets
   iris ke variable iris
3 digits = datasets.load_digits() #Memasukkan data dari
   datasets digits ke variable digits
4 print(digits.data) #Menampilkan data dari datasets digits ke
   console
```



**Gambar 1.30** Running Kode dari Spyder dan Hasil Variable Explorer

```

In [2]: runfile('C:/Users/Rin/contoh.py', wdir='C:/Users/Rin')
[[ 0.  0.  5. ... 0.  0.  0.]
 [ 0.  0.  0. ... 10. 0.  0.]
 [ 0.  0.  0. ... 16. 9.  0.]
 ...
 [ 0.  0.  1. ... 6.  0.  0.]
 [ 0.  0.  2. ... 12. 0.  0.]
 [ 0.  0.  10. ... 12. 1.  0.]]
    
```

**Gambar 1.31** Running Loading an example dataset dari Spyder

### 1.8.2.2 Loading an example dataset

- Import fungsi datasets dari library sklearn

```

1 from sklearn import datasets #Import fungsi datasets dari
    library sklearn
    
```

- Memasukkan data dari datasets iris ke variable iris

```

1 iris = datasets.load_iris() #Memasukkan data dari datasets
    iris ke variable iris
    
```

- Memasukkan data dari datasets digits ke variable digits

```

1 digits = datasets.load_digits() #Memasukkan data dari
    datasets digits ke variable digits
    
```

- Menampilkan data dari datasets digits ke console

```

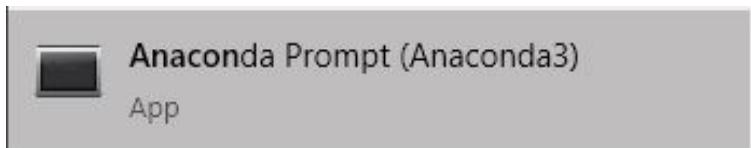
1 print(digits.data) #Menampilkan data dari datasets digits ke
    console
    
```

```
>>> from sklearn import datasets
```

Gambar 1.32 Hasil Running kode loading an example dataset

#### 1.8.2.3 Learning and predicting

- Buka Anaconda Prompt



Gambar 1.33 Anaconda Prompt

- Lalu kita import datasets dari sklearn seperti dibawah ini

```
>>> from sklearn import datasets
```

Gambar 1.34 Menggunakan datasets

- lalu kita mendefinisikan iris dan digits menjadi variable

```
>>> iris = datasets.load_iris()
```

Gambar 1.35 mendefinisikan iris

```
>>> digits = datasets.load_digits()
```

Gambar 1.36 mendefinisikan digits

- Lalu kita import svm dari sklearn yang nantinya digunakan untuk menjadi estimasi angka kita

```
>>> from sklearn import svm
```

**Gambar 1.37** Menggunakan `svm`

- Lalu, kita definisikan `clf` sebagai classifier, disini gamma didefinisikan secara manual

```
>>> clf = svm.SVC(gamma=0.001, C=100.)
```

**Gambar 1.38** Mendefinisikan Classifier

- Estimator `clf` (for classifier) pertama kali dipasang pada model. Ini dilakukan dengan melewati training set ke metode `fit`. Untuk training set, akan menggunakan semua gambar dari set data yang ada, kecuali untuk gambar terakhir, yang dicadangkan untuk prediksi. Pada skrip dibawah memilih training set dengan sintaks Python `[:-1]`, yang menghasilkan array baru yang berisi semua kecuali item terakhir dari digits.

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

**Gambar 1.39** Memanggil Classifier

- Menunjukkan prediksi angka baru

```
>>> clf.predict(digits.data[-1:])
array([8])
```

**Gambar 1.40** Prediksi nilai baru

```
1 lastlinelastline
2 from sklearn import datasets #Import fungsi datasets dari library
   sklearn
3 iris = datasets.load_iris() #Memasukkan data dari datasets iris
   ke variable iris
4 digits = datasets.load_digits() #Memasukkan data dari datasets
   digits ke variable digits
```

```

5 from sklearn import svm #Mengimport sebuah Support Vector Machine
6 # (SVM) yang merupakan algoritma classification yang akan
7 # diambil dari Scikit-Learn.
8 clf = svm.SVC(gamma=0.001, C=100.) #Mendeklarasikan suatu value
9 # yang bernama clf yang berisi gamma.
10 clf.fit(digits.data[:-1], digits.target[:-1]) #Estimator clf (for
11 classifier)
12 hasil = clf.predict(digits.data[-1:]) #Menunjukkan prediksi
13 angka baru
14 print(hasil)

```

#### 1.8.2.4 Model Persistance Model Persistance

```

1 lastlinelastline
2 %%Cara Dump Pertama
3 from sklearn import svm #Mengimport sebuah Support Vector
4 #Machine(SVM) yang merupakan algoritma classification yang
5 #akan diambil dari Scikit-Learn.
6 from sklearn import datasets #Import fungsi datasets dari library
7 #sklearn
8 clf = svm.SVC() #Mendefinisikan clf dengan fungsi svc dari
9 #library svm
10 X, y = datasets.load_iris(return_X_y=True) #Mengisi variable x
11 #dan y dengan data dari datasets
12 clf.fit(X, y) #Estimator clf (for classifier)
13
14
15 %%Cara Dump Kedua
16 from sklearn import svm # Digunakan untuk memanggil class svm
17 # dari library sklearn
18 from sklearn import datasets # Diguankan untuk class datasets
19 # dari library sklearn
20 clf = svm.SVC() # membuat variabel clf , dan
21 # memanggil class svm dan fungsi SVC
22 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
23 # iris dan mengembalikan nilainya.
24 clf.fit(X, y) #Perhitungan nilai label
25
26 from joblib import dump, load #memanggil class dump dan load pada
27 #library joblib
28 dump(clf, '1174066.joblib') #Menyimpan model kedalam 1174066.
29 #joblib
30 hasil = load('1174066.joblib') #Memanggil model 1174066
31 hasil.predict(X[0:1])
32 print(y[0]) # Menampilkan Model yang dipanggil sebelumnya

```

#### 1.8.2.5 Conversion Conversion

```

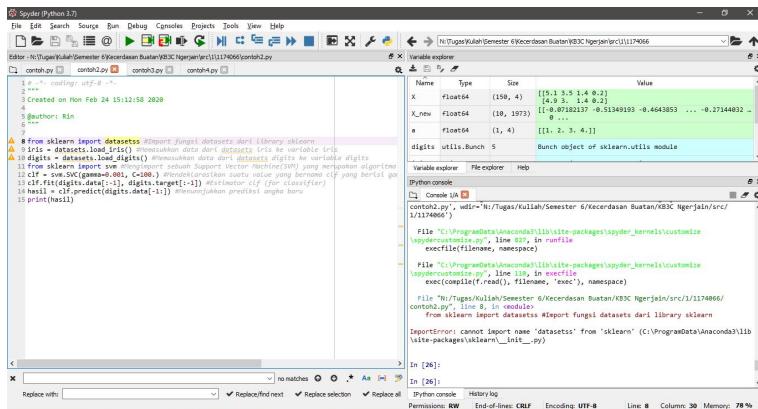
1 lastlinelastline
2 import numpy as np # memanggil library numpy dan dibuat alias np
3 from sklearn import random_projection #Memanggil class
        random_projection pada library sklearn
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
        mendefisikan np, fungsi random dan attr RandomState kedalam
        variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan nilai
        random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai random
        sebelumnya, kedalam array, dan menentukan typedatanya sebagai
        float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
        membuat variabel transformer , dan mendefinisikan
        classrandom_projection dan memanggil fungsi
        GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru dan
        melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64
13 print(X_new) # Menampilkan isi variabel X_new

```

### 1.8.3 Penanganan Error

Dari percobaan yang dilakukan di atas, apabila mendapatkan error maka:

#### 1. Screenshoot Error



**Gambar 1.41** ImportError: cannot import name 'datasetss' from 'sklearn'

#### 2. Tuliskan kode eror dan jenis errornya [hari ke 2](10)

- ImportError

3. Solusi pemecahan masalah error tersebut[hari ke 2](10)

- ImportError  
Cek kembali jika ada yang typo

#### 1.8.4 Bukti Tidak Plagiat



Gambar 1.42 Bukti Tidak Melakukan Plagiat Chapter 1

#### 1.8.5 Link Youtube

[https://youtu.be/S8Sj\\_vZluUs](https://youtu.be/S8Sj_vZluUs)

### 1.9 1174087 - Ilham Muhammad Ariq

#### 1.9.1 Teori

##### 1. Definisi Kecerdasan Buatan

Kecerdasan Buatan adalah suatu ilmu yang mempelajari bagaimana cara komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Secara sederhana AI adalah teknik dan ilmu untuk membangun atau membuat suatu mesin menjadi cerdas, terutama pada program komputer. Kecerdasan yang dimaksud yaitu seperti yang dimiliki oleh manusia namun pada mesin akan dibuat cepat dan tepat atau akurat.

##### 2. Sejarah Kecerdasan Buatan

Artificial intelligence merupakan inovasi baru di bidang ilmu pengetahuan. Mulai terbentuk sejak adanya komputer modern dan kira-kira terjadi sekitaran tahun 1940 dan 1950. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Pada awal 50-an, studi tentang "mesin berpikir" memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan informasi. Pada tahun 1956, para ilmuan jenius seperti Alan Turing, Norbert Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen dibidang cybernetics, matematika, algoritma dan teori jaringan. Namun, seorang ilmuan komputer dan kognitif John McCarthy adalah orang yang dating dengan ide untuk bergabung dengan

upaya penelitian terpisah ini kedalam satu bidang yang akan mempelajari topic baru untuk imajinasi manusia yaitu kecerdasan buatan. Dia adalah orang yang menciptakan istilah tersebut dan kemudian mendirikan laboratorium Kecerdasan Buatan di MIT dan Stan ford.

Pada tahun 1956, McCarthy yang sama mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Peneliti terkemuka dalam teori kompleksitas, simulasi bahasa, hubungan antara keacakan dan pemikiran kreatif, jaringan saraf diundang. Tujuan dari bidang penelitian yang baru dibuat adalah untuk mengembangkan mesin yang dapat mensimulasikan setiap aspek kecerdasan. Itulah sebabnya Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan. Sejak saat itu, Kecerdasan Buatan telah hidup melalui decade kemuliaan dan cemoohan, yang dikenal luas sebagai musim panas dan musim dingin AI. Musim panasnya ditandai dengan optimism dan dana besar, sedangkan musim dinginnya dihadapkan dengan pemotongan dana, ketidakpercayaan dan pesimisme.

### 3. Perkembangan Kecerdasan Buatan

Teknologi Artificial Intelligence semakin ramai dibahas dalam berbagai diskusi teknologi di seluruh dunia. Menurut kebanyakan orang, pekerjaan seperti kasir, operator telepon, pengendara truk, dan lainnya sangat berpeluang besar untuk tergantikan oleh Artificial Intelligence. Mengapa terjadi hal demikian? dikarenakan memang bahwa AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Namun, dalam beberapa aspek memang pekerja manusia masih unggul dibandingkan AI itu sendiri. Para generasi muda yang ada di dunia terutama di daerah Asia terlihat sudah memahami fungsi dan efek dari AI dalam kehidupan kita sehari-hari. Berdasarkan survei yang dilakukan oleh Microsoft, terdapat 39 persen responden yang mempertimbangkan untuk menggunakan mobil tanpa pengemudi dan 36 persen lainnya setuju bahwa robot masa depan dengan software untuk beroperasi mampu meningkatkan produktivitas. Dari survei tersebut kita sebagai pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tanpa memberikan efek samping terhadap etos kerja dan keseharian kita sebagai pengguna dalam kehidupan sehari-hari.

AI Summer 1 (1956-1973) KOnferensi Dartmouth diikuti oleh 17 tahun kemajuan luar biasa. Proyek penelitian yang dilakukan di MIT, universitas di Edinburgh, Stanford dan Carnegie Mellon menerima dana besar-besaran, yang akhirnya membawa hasil. Selama tahun-tahun itulah komputer pemrograman mulai melakukan masalah aljabar, membuktikan teorema geometris, memahami dan menggunakan sintaks dan tata bahasa Inggris. Terlepas dari ditinggalkannya koneksionisme dan terjemahan mesin yang gagal, yang menunda penelitian Natural Language Processing (NLP) selama bertahun-tahun, banyak prestasi dari masa lalu

yang membuat sejarah. Berikut ini beberapa diantaranya : Pelopor pembelajaran mesin, Ray Solomonoff meletakkan dasar-dasar teori metematika AI, memperkenalkan metode Bayesian universal untuk inferensi dan prediksi induktif Thomas Evans menciptakan program ANALOGI heuristik, yang memungkinkan komputer memecahkan masalah geometri-analogi Unimation, perusahaan robotika pertama didunia, menciptakan robot industri Unimate, yang bekerja pada jalur perakitan modil General Motors. Joseph Weizenbaum membangun ELIZA-program interaktif yang dapat membawa percakapan dalam bahasan Inggris tentang topik apapun. Ross Quillian menunjukkan jaring semantik, sedangkan Jaime Carbonell (Sr.) mengembangkan Cendikia-program interaktif untuk instruksi yang dibantu komputer berdasarkan jaring semantik. Edward Feigenbaum dan Julian Feldman menerbitkan Computeks and Thought, kumpulan artikel pertama tentang AI.

4. Definisi supervised learning, klasifikasi, regresi, unsupervised learning, dataset, training set dan testing set.

- **Supervised Learning**

Supervised Learning merupakan sebuah tipe learning yang mempunyai variable input dan variable output, tipe ini juga menggunakan satu algoritma atau lebih dari satu algoritma yang digunakan untuk mempelajari fungsi pemetaan dari input ke output.

- **Klasifikasi**

Klasifikasi adalah pengelompokan data di mana data yang digunakan memiliki label atau kelas target. Sehingga algoritma untuk menyelesaikan masalah klasifikasi dikategorikan ke dalam pembelajaran terbimbing.

- **Regresi**

Regresi metode analisis statistik yang digunakan untuk dapat melihat efek antara dua atau lebih variabel. Hubungan variabel dalam pertanyaan adalah fungsional yang diwujudkan dalam bentuk model matematika. Dalam analisis regresi, variabel dibagi menjadi dua jenis, yaitu variabel respons atau yang biasa disebut variabel dependen dan variabel independen atau dikenal sebagai variabel independen. Ada beberapa jenis analisis regresi, yaitu regresi sederhana yang mencakup linear sederhana dan regresi non-linear sederhana dan regresi berganda yang mencakup banyak linier atau non-linear berganda. Analisis regresi digunakan dalam pembelajaran mesin pembelajaran dengan metode pembelajaran terawasi.

- **Unsupervised learning**

Unsupervised learning jenis pembelajaran di mana kita hanya memiliki data input (input data) tetapi tidak ada variabel output yang terkait. Tujuan dari pembelajaran tanpa pengawasan adalah untuk



```

4 x = iris.data           # Menyimpan nilai data sets iris
  pada variabel x
5 y = iris.target         # Menyimpan nilai data label iris
  pada variabel y

```

### 3. Mencoba Learning dan predicting

```

1 %%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Digunakan
  Untuk Memanggil fungsi KNeighborsClassifier
3                                     # pada
4 class sklearn dan library sklearn
5 import numpy as np # memanggil library numpy dan dibuat alias
  np
6 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
  , dan memanggil fungsi KNeighborsClassifier
7                                     #dan mendefinisikan k
  -nya adalah 1
8 knn.fit(x,y)                      #Perhitungan
  matematika library kkn
9 a=np.array([1.0,2.0,3.0,4.0])      #Membuat Array
10 a = a.reshape(1,-1)                #Mengubah Bentuk
  Array jadi 1 dimensi
11 hasil = knn.predict(a)            #Memanggil fungsi
  prediksi
12 print(hasil)                    #menampilkan hasil
  prediksi

```

### 4. Mencoba Model Persistence

```

1 %% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm
  dari library sklearn
3 from sklearn import datasets # Diguangkan untuk class datasets
  dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan
  memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
  iris dan mengembalikan nilainya.
6 clf.fit(X, y)             #Perhitungan nilai label
7
8 from joblib import dump, load #memanggil class dump dan load
  pada library joblib
9 dump(clf, '1174087.joblib') #Menyimpan model kedalam 1174027.
  joblib
10 hasil = load('1174087.joblib') #Memanggil model 1174027
11 print(hasil) # Menampilkan Model yang dipanggil sebelumnya

```

### 5. Mencoba Conventions

```

1 %% Conventions

```

```

2 import numpy as np # memanggil library numpy dan dibuat alias
    np
3 from sklearn import random_projection #Memanggil class
    random_projection pada library sklean
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
    mendefinisikan np, fungsi random dan attr RandomState
    kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
    nilai random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
    random sebelumnya, kedalam array , dan menentukan
    typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
        membuat variabel transformer , dan mendefinisikan
        classrandom_projection dan memanggil fungsi
        GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru
        dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64
13 print(X_new) # Menampilkan isi variabel X_new

```

### 1.9.3 Penanganan Error

#### 1. ScreenShoot Error

```

ImportError: cannot import name 'dataset' from 'sklearn' (C:\Users\Peneliti\anaconda\lib
\site-packages\sklearn\_init_.py)

```

**Gambar 1.45** Import Error

#### 2. Tuliskan Kode Error dan Jenis Error

- Import Error

#### 3. Cara Penangan Error

- Import Error

Dengan Menginstall Library Yang Tidak Ditemukan atau Memperbaiki Penulisan Library

### 1.9.4 Bukti Tidak Plagiat



**Gambar 1.46**      **Bukti Tidak Melakukan Plagiat**

### 1.9.5 Link Youtube:

<https://www.youtube.com/watch?v=-qw8q7jhEmg>

## 1.10 1174084 - Muhammad Reza Syachrani

### 1.10.1 Teori

#### 1. Definisi Sejarah dan Perkembangan Kecerdasan Buatan

Definisi Kecerdasan Buatan, Kecerdasan Buatan biasa disebut dengan istilah AI (Artificial Intelligence). AI merupakan tentang bagaimana cara untuk melengkapi sebuah komputer dengan kemampuan yang dimiliki oleh manusia. Dengan demikian, Diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemuiinya kemudian itu lah yang disebut dengan kecerdasan buatan. Kecerdasan buatan adalah kemampuan komputer yang dikendalikan komputer untuk melakukan tugas yang umumnya dikaitkan dengan sesuatu yang cerdas.

Sejarah Kecerdasan Buatan, Kecerdasan buatan / Artificial intelligence mulai terbentuk pada tahun 1940 dan 1950. Pada awal 50-an, studi tentang "mesin berpikir" memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan informasi. Pada tahun 1956, para ilmuan jenius seperti Alan Turing, Norbert Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen dibidang cybernetics, matematika, algoritma dan teori jaringan. Namun, seprang ilmuan komputer dan kognitif John McCarthy adalah orang yang dating dengan ide untuk bergabung dengan upaya penelitian terpisah ini kedalam satu bidang yang akan mempelajari topic baru untuk imajinasi manusia yaitu kecerdasan buatan. Pada tahun 1956, McCarthy yang sama mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Tujuan dari bidang penelitian yang baru dibuat adalah untuk mengembangkan mesin yang dapat mensimulasikan setiap aspek kecerdasan. Oleh karena itu Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan.

Perkembangan Kecerdasan Buatan, perkembangan kecerdasan buatan

dapat menggantikan berbagai pekerjaan manusia seperti kasir, operator telepon, pengendara truk, dan lainnya. AI Summer 1 (1956-1973) Konferensi Dartmouth diikuti oleh 17 tahun kemajuan luar biasa. Projek penelitian yang dilakukan di MIT, universitas di Edinburgh, Stanford dan Carnegie Mellon menerima dana besar-besaran, yang akhirnya membawa hasil. Selama tahun-tahun itulah komputer pemrograman mulai melakukan masalah aljabar, membuktikan teorema geometris, memahami dan menggunakan sintaks dan tata bahasa Inggris. Terlepas dari ditinggalkannya koneksiisme dan terjemahan mesin yang gagal, yang menunda penelitian Natural Language Processing (NLP) selama bertahun-tahun, banyak prestasi dari masa lalu yang membuat sejarah. Berikut ini beberapa diantaranya : Pelopor pembelajaran mesin, Ray Solomonoff dasar-dasar teori metematika AI, memperkenalkan metode Bayesian universal untuk inferensi dan prediksi induktif Thomas Evans menciptakan program ANALOGI heuristik, yang memungkinkan komputer memecahkan masalah geometri analogi Unimation, perusahaan robotika pertama didunia, menciptakan robot industri Unimate, yang bekerja pada jalur perakitan modil Genenral Motors. Joseph Weizenbaum membangun ELIZA-program interaktif yang dapat membawa percakapan dalam bahasan Inggris tentang topik apapun. Ross Quillian menunjukkan jaring semantik, sedangkan Jaime Carbonell (Sr.) mengembangkan Cendikia-program interaktif untuk instruksi yang dibantu komputer berdasarkan jaring semantik. Edward Feigenbaum dan Julian Feldman menerbitkan Computeks and Thought, kumpulan artikel pertama tentang AI.

2. Definisi Supervised learning, klasifikasi, regresi, unsupervised learning, dataset, training set dan testing set.

- **Supervised Learning**

Supervised Learning merupakan tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan(juga disebut sinyal pengawasan super).

- **Klasifikasi**

Klasifikasi adalah pembagian sesuatu menurut kelas-kelas. Klasifikasi merupakan proses dari pengelompokan benda berdasarkan ciri-ciri persamaan dan juga perbedaan. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah.

- **Regresi**

Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel. Regresi adalah

membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti "gaji" atau "berat".

- Unsupervised learning

Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Dalam Unsupervised Learning, sistem AI dapat mengelompokkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan.

- Data set

Data set objek yang merepresentasikan data dan relasinya di memory. Strukturnya mirip dengan data di database. Dataset berisi koleksi dari datatable dan datarelation.

- Training Set

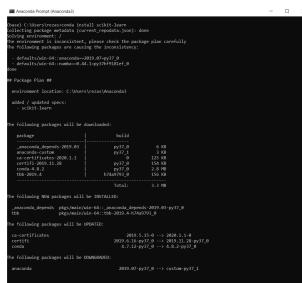
Training Set adalah set digunakan oleh algoritma klassifikasi. Dapat dicontohkan dengan decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier

- Testing Set

Testing Set merupakan set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Dapat berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakan nya sampai akhir.

### 1.10.2 Praktek

- Instalasi Library scikit dari ianaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer



Gambar 1.47 Instalasi Package Scikit Learn

| Name | Type       | Size     | Value                                |
|------|------------|----------|--------------------------------------|
| iris | util.Bunch | 4        | Batch object of sklearn.utils module |
| x    | float64    | (150, 4) | [1, 2, 3, 4, 5, 6, 7, 8, 9]          |
| y    | int32      | (150,)   | [0, 0, 0, ..., 2, 2, 2]              |

Gambar 1.48 Isi Variabel Explorer

## 2. Mencoba loading an example dataset

```

1 #%% Mencoba loading an example dataset
2 from sklearn import datasets # Untuk memanggil class datasets
   dari library sklearn
3 iris = datasets.load_iris() # Menggunakan datasets iris
4 x = iris.data             # Menyimpan nilai data set iris
   pada variabel x
5 y = iris.target           # Menyimpan nilai data label iris
   pada variabel y

```

## 3. Mencoba Learning dan predicting

```

1 #%%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Untuk
   Memanggil fungsi KNeighborsClassifier
3                                     # pada
   class sklearn dan library sklearn
4 import numpy as np # memanggil library numpy dan dibuat alias
   np
5 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
   , dan memanggil fungsi KNeighborsClassifier
6                                     #dan mendefinisikan k
   -nya adalah 1
7 knn.fit(x,y)                      #Perhitungan
   matematika library kkn
8 a=np.array([1.0,2.0,3.0,4.0])      #Membuat Array
9 a = a.reshape(1,-1)                 #Mengubah Bentuk
   Array jadi 1 dimensi
10 hasil = knn.predict(a)            #Memanggil fungsi
   prediksi
11 print(hasil)                    #menampilkan hasil
   prediksi

```

## 4. Mencoba Model Persistence

```

1 #%% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm
   dari library sklearn
3 from sklearn import datasets # Diguankan untuk class datasets
   dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan
   memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
   iris dan mengembalikan nilainya.
6 clf.fit(X, y)             #Perhitungan nilai label

```

```

7
8 from joblib import dump, load #memanggil class dump dan load
   pada library joblib
9 dump(clf, '1174084.joblib') #Menyimpan model kedalam 1174084.
   joblib
10 hasil = load('1174084.joblib') #Memanggil model 1174084
11 print(hasil) # Menampilkan Model yang dipanggil sebelumnya

```

## 5. Mencoba Conventions

```

1 %% Conventions
2 import numpy as np # memanggil library numpy dan dibuat alias
   np
3 from sklearn import random_projection #Memanggil
   random_projection pada library sklean
4
5 rng = np.random.RandomState(0) #Membuat variabel rng, dan
   mendefisikan np, fungsi random dan attr RandomState
   kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
   nilai random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
   random sebelumnya, kedalam array, dan menentukan
   typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
      membuat variabel transformer, dan mendefinisikan
      classrandom_projection dan memanggil fungsi
      GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru
      dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64
13 print(X_new) # Menampilkan isi variabel X_new

```

### 1.10.3 Penanganan Error

#### 1. ScreenShoot Error

```
ModuleNotFoundError: No module named 'sklear'
```

**Gambar 1.49** Module Error

#### 2. Tuliskan Kode Error dan Jenis Error

- Module Error

#### 3. Cara Penangan Error

- Module Error

Dengan memperbaiki penulisan atau kesalahan dalam kode atau melakukan install package atau modul yang belum terinstal

## 1.10.4 Bukti Tidak Plagiat



**Gambar 1.50** Bukti Tidak Melakukan Plagiat

## 1.11 1174083 - Bakti Qilan Mufid

### 1.11.1 Teori

#### 1.11.1.1 Kecerdasan Buatan

##### 1. Definisi Kecerdasan Buatan

Kecerdasan Buatan biasa disebut dengan istilah AI (Artificial Intelligence). AI sendiri merupakan suatu cabang dalam bisnis sains komputer sains dimana mengkaji tentang bagaimana cara untuk men lengkapinya sebuah komputer dengan kemampuan atau kepintaran layaknya atau mirip dengan yang dimiliki manusia. Sebagai contoh, sebagaimana komputer dapat berkomunikasi dengan pengguna baik menggunakan kata, suara maupun lain sebagainya. Dengan kemampuan ini, diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemuiinya kemudian itulah yang disebut dengan kecerdasan buatan. Kecerdasan buatan adalah kemampuan komputer digital atau robot yang dikendalikan komputer untuk melakukan tugas yang umumnya dikaitkan dengan sesuatu yang cerdas. Istilah ini sering diterapkan pada proyek pengembangan sistem yang diberkahi dengan karakteristik proses intelektual manusia, seperti kemampuan untuk berpikir, menemukan makna, menggeneralisasi, atau belajar dari pengalaman masa lalu.

Kecerdasan Buatan adalah salah satu bidang studi yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara lebih manusiawi dan lebih bisa dipahami oleh manusia. Kecerdasan buatan makin canggih dengan kemampuan komputer dalam memperbarui pengetahuannya dengan banyaknya testing dan perkembangan target analisa. Untuk kecerdasan buatan ada banyak contoh dan jenisnya. Salah satu contoh yang paling terkenal dari Artificial Intelligence ialah Google Assistant. Google Assistant digunakan untuk kemudahan user dalam menemukan berbagai hal maupun penyetelan langsung terhadap smartphone yang digunakan dan masih banyak lagi.

##### 2. Sejarah Kecerdasan Buatan

Artificial intelligence merupakan inovasi baru di bidang ilmu pengetahuan. Mulai terbentuk sejak adanya komputer modern dan kira-kira terjadi sekitaran tahun 1940 dan 1950. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Pada awal 50-an, studi tentang “mesin berpikir” memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan informasi. Pada tahun 1956, para ilmuan jenius seperti Alan Turing, Norbert Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen dibidang cybernetics, matematika, algoritma dan teori jaringan. Namun, seprang ilmuan komputer dan kognitif John McCarthy adalah orang yang dating dengan ide untuk bergabung dengan upaya penelitian terpisah ini kedalam satu bidang yang akan mempelajari topic baru untuk imajinasi manusia yaitu kecerdasan buatan. Dia adalah orang yang menciptakan istilah tersebut dan kemudian mendirikan laboratorium Kecerdasan Buatan di MIT dan Stan ford.

Pada tahun 1956, McCarthy yang sama mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Peneliti terkemuka dalam teori kompleksitas, simulasi bahasa, hubungan antara keacakan dan pemikiran kreatif, jaringan saraf diundang. Tujuan dari bidang penelitian yang baru dibuat adalah untuk mengembangkan mesin yang dapat mensimulasikan setiap aspek kecerdasan. Itulah sebabnya Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan. Sejak saat itu, Kecerdasan Buatan telah hidup melalui decade kemuliaan dan cemoohan, yang dikenal luas sebagai musim panas dan musim dingin AI. Musim panasnya ditandai dengan optimism dan dana besar, sedangkan musim dinginnya dihadapkan dengan pemotongan dana, ketidakpercayaan dan pesimisme.

### 3. Perkembangan Kecerdasan Buatan

Teknologi Artificial Intelligence semakin ramai dibahas dalam berbagai diskusi teknologi di seluruh dunia. Menurut kebanyakan orang, pekerjaan seperti kasir, operator telepon, pengendara truk, dan lainnya sangat berpeluang besar untuk tergantikan oleh Artificial Intelligence. Mengapa terjadi hal demikian? dikarenakan memang bahwa AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Namun, dalam beberapa aspek memang pekerja manusia masih unggul dibandingkan AI itu sendiri. Para generasi muda yang ada di dunia terutama di daerah Asia terlihat sudah memahami fungsi dan efek dari AI dalam kehidupan kita sehari-hari. Berdasarkan survei yang dilakukan oleh Microsoft, terdapat 39 persen responden yang mempertimbangkan untuk menggunakan mobil tanpa pengemudi dan 36 persen lainnya setuju bahwa robot masa depan dengan software untuk beroperasi mampu meningkatkan produktivitas. Dari survei tersebut kita sebagai pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tanpa

memberikan efek samping terhadap etos kerja dan keseharian kita sebagai pengguna dalam kehidupan sehari-hari.

AI Summer 1 (1956-1973) KOnferensi Dartmounth diikuti oleh 17 tahun kemajuan luar biasa. Proyek penelitian yang dilakukan di MIT, universitas di Edinburgh, Stanfورد dan Carnegie Mellon menerima dana besar-besaran, yang akhirnya membawa hasil. Selama tahun-tahun itulah komputer pemrograman mulai melakukan masalah aljabar, membuktikan teorema geometris, memahami dan menggunakan sintaks dan tata bahasa Inggris. Terlepas dari ditinggalkannya koneksiisme dan terjemahan mesin yang gagal, yang menunda penelitian Natural Language Processing (NLP) selama bertahun-tahun, banyak prestasi dari masa lalu yang membuat sejarah. Berikut ini beberapa diantaranya : Pelopor pembelajaran mesin, Ray Solomonoff meletakkan dasar-dasar teori metematika AI, memperkenalkan metode Bayesian universal untuk inferensi dan prediksi induktif Thomas Evans menciptakan program ANALOGI heuristik, yang memungkinkan komputer memecahkan masalah geometri-analogi Unimation, perusahaan robotika pertama didunia, menciptakan robot industri Unimate, yang bekerja pada jalur perakitan modil General Motors. Joseph Weizenbaum membangun ELIZA-program interaktif yang dapat membawa percakapan dalam bahasan Inggris tentang topik apapun. Ross Quillian menunjukkan jaring semantik, sedangkan Jaime Carbonell (Sr.) mengembangkan Cendikia-program interaktif untuk instruksi yang dibantu komputer berdasarkan jaring semantik. Edward Feigenbaum dan Julian Feldman menerbitkan Computeks and Thought, kumpulan artikel pertama tentang AI.

#### 1.11.1.2 *Scikit-Learn*

##### 1. Supervised Learning

Supervised Learning adalah tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan(juga disebut sinyal pengawasan super). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan, yang dapat digunakan untuk memetakan contoh-contoh baru. Skenario optimal akan memungkinkan algoritma menentukan label kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi dari data pelatihan untuk situasi yang tidak terlihat dengan cara yang "masuk akal". Supervised Learning adalah pendekatan dimana sudah terdapat data yang dilatih selain itu juga terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini yaitu mengelompokkan suatu data ke dta yang sudah ada. Supervised Learning menyediakan algoritma pembelajaran dengan jumlah yang diketahui untuk men-

dukung penilaian dimasa depan. Chatbots, mobil self-driving, program pengenalan wajah, sistem pakar dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak diawasi. Supervised Learning sebagian besar terkait dengan AI berbasis pengambilan tetapi mereka juga mungkin mampu menggunakan model pembelajaran generatif. Data pelatihan untuk pembelajaran yang diawasi mencakup serangkaian contoh dengan subjek input berpasangan dan output yang diinginkan (yang juga disebut sebagai sinyal pengawasan).

Dalam pembelajaran yang diawasi untuk pemrosesan gambar, misalnya sistem AI mungkin dilengkapi dengan gambar berlabel kendaraan dalam kategori seperti mobil dan truk. Setelah jumlah pengamatan yang cukup, sistem harus dapat membedakan antara dan mengkategorikan gambar yang tidak berlabel, dimana waktu pelatihan dapat dikatakan lengkap. Model Supervised Learning memiliki beberapa keunggulan dibandingkan pendekatan tanpa pengawasan, tetapi mereka juga memiliki keterbatasan. Sistem lebih cenderung membuat penilaian bahwa manusia dapat berhubungan, misalnya karena manusia telah memberikan dasar untuk keputusan. Namun, dalam kasus metode berbasis pengambilan, Supervised Learning mengalami kesulitan dalam menangani informasi baru. Jika suatu sistem dengan kategori untuk mobil dan truk disajikan dengan sepeda, misalnya ia harus salah dikelompokkan dalam satu kategori ata yang lain. Namun, jika sistem AI bersifat generatif, ia mungkin tidak tahu apa sepeda itu tetapi akan dapat mengenalinya sebagai milik kategori yang terpisah.

## 2. Regresi

Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel. Regresi adalah membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan makan, yang paling sederhana adalah regresi linier. Ia mencoba untuk menyesuaikan data dengan hyper-plane terbaik yang melewati poin.

## 3. Klasifikasi

Klasifikasi adalah pembagian sesuatu menurut kelas-kelas ( class ). Menurut Ilmu Pengetahuan, Klasifikasi merupakan proses pengelompokan benda berdasarkan ciri-ciri persamaan dan juga perbedaan. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah. Label ( $y$ ) umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran mesin dan statistik, klasifikasi adalah pendekatan pembelajaran yang diawasi di mana program komputer belajar dari input data yang diberikan kepadanya dan kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pengamatan baru. Kumpulan data ini mungkin hanya bersifat dua kelas (seperti mengidentifikasi apakah orang tersebut berjenis kelamin laki-laki atau perempuan atau bahwa

surat itu spam atau bukan-spam) atau mungkin juga multi-kelas. Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, identifikasi metrik, klasifikasi dokumen dll.

#### 4. Unsupervised Learning

Unsupervised Learning berbeda dengan Supervised Learning. Perbedaannya ialah unsupervised learning tidak memiliki data latih, sehingga dari data yang ada kita mengelompokan data tersebut menjadi 2 ataupun 3 bagian dan seterusnya. Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasi atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Dalam Unsupervised Learning, sistem AI dapat mengelompokkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan.

Dalam Unsupervised Learning, sistem AI disajikan dengan data yang tidak berlabel, tidak terkategorisasi dan algoritma sistem bekerja pada data tanpa pelatihan sebelumnya. Outputnya tergantung pada algoritma kode. Menundukkan suatu sistem pada Unsupervised Learning adalah salah satu cara untuk menguji AI. Algoritma Unsupervised Learning dapat melakukan tugas pemrosesan yang lebih kompleks daripada sistem pembelajaran yang diawasi. Namun, pembelajaran tanpa pengawasan bisa lebih tidak terduga daripada model alternatif. Sementara Unsupervised Learning mungkin, misalnya, mencari tahu sendiri cara memilah kucing dari anjing, mungkin juga menambahkan kategori yang tidak terduga dan tidak diinginkan untuk menangani breed yang tidak biasa, membuat kekacauan bukannya keteraturan

#### 5. Data Set

Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory. Strukturnya mirip dengan data di database, namun bedanya dataset berisi koleksi dari data table dan data relation. mendapatkan data yang tepat berarti mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda prediksi; yaitu data yang berisi sinyal tentang peristiwa yang Anda pedulikan. Data harus diselaraskan dengan masalah yang Anda coba selesaikan. Gambar kucing tidak terlalu berguna ketika Anda sedang membangun sistem identifikasi wajah. Memverifikasi bahwa data selaras dengan masalah yang ingin Anda selesaikan harus dilakukan oleh ilmuwan data. Jika Anda tidak memiliki data yang tepat, maka upaya Anda untuk membangun solusi AI harus kembali ke tahap pengumpulan data. Format ujung kanan untuk pembelajaran dalam umumnya adalah tensor, atau array multi-dimensi. Jadi jalur pipa data yang dibangun untuk pembelajaran mendalam umumnya akan mengkonversi semua data - baik itu gambar, video, suara, suara, teks atau deret waktu menjadi vektor dan tensor yang dapat diterapkan operasi aljabar linier. Data itu seringkali perlu

dinormalisasi, distandarisasi dan dibersihkan untuk meningkatkan kegunaannya, dan itu semua adalah langkah dalam ETL pembelajaran mesin. Deeplearning4j menawarkan alat ETV DataVec untuk melakukan tugas-tugas pemrosesan data tersebut.

Pembelajaran yang dalam, dan pembelajaran mesin yang lebih umum, membutuhkan pelatihan yang baik agar bekerja dengan baik. Mengumpulkan dan membangun set pelatihan badan yang cukup besar dari data yang diketahui membutuhkan waktu dan pengetahuan khusus domain tentang di mana dan bagaimana mengumpulkan informasi yang relevan. Perangkat pelatihan bertindak sebagai tolok ukur terhadap mana jaring pembelajaran dalam dilatih. Itulah yang mereka pelajari untuk direkonstruksi sebelum mereka melepaskan data yang belum pernah mereka lihat sebelumnya. Pada tahap ini, manusia yang berpengetahuan luas perlu menemukan data mentah yang tepat dan mengubahnya menjadi representasi numerik yang dapat dipahami oleh algoritma pembelajaran mendalam, tensor. Membangun set pelatihan, dalam arti tertentu, prapra pelatihan. Set pelatihan yang membutuhkan banyak waktu atau keahlian dapat berfungsi sebagai keunggulan dalam dunia ilmu data dan pemecahan masalah. Sifat keahlian sebagian besar dalam memberi tahu algoritma Anda apa yang penting bagi Anda dengan memilih apa yang masuk ke dalam set pelatihan. Ini melibatkan menceritakan sebuah kisah melalui data awal yang Anda pilih yang akan memandu jaring pembelajaran mendalam Anda saat mereka mengekstraksi fitur-fitur penting, baik di set pelatihan maupun dalam data mentah yang telah mereka ciptakan untuk dipelajari. Untuk membuat set pelatihan yang bermanfaat, Anda harus memahami masalah yang Anda selesaikan; yaitu apa yang Anda inginkan agar jaring pembelajaran mendalam Anda memperhatikan, di mana hasil yang ingin Anda prediksi.

## 6. Training Set

Training Set adalah set digunakan oleh algoritma klassifikasi . Dapat dicontohkan dengan : decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier. Menjalankan pelatihan yang diatur melalui jaringan saraf mengajarkan pada net cara menimbang berbagai fitur, menyesuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan dalam hasil Anda. Koefisien-koefisien tersebut, juga dikenal sebagai parameter, akan terkandung dalam tensor dan bersama-sama mereka disebut model, karena mereka mengkodekan model data yang mereka latih. Mereka adalah takeaways paling penting yang akan Anda dapatkan dari pelatihan jaringan saraf.

## 7. Testing Set

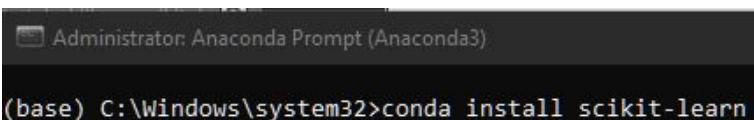
Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Ini berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakan sampai akhir.

Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda terhadap pengambilan sampel acak akhir ini. Hasil yang dihasilkannya harus memvalidasi bahwa jaring Anda secara akurat mengenali gambar, atau mengenalinya setidaknya [x] dari jumlah tersebut. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan, lihat hyperparameter yang Anda gunakan untuk menyetel jaringan, serta kualitas data Anda dan lihat teknik pra-pemrosesan Anda.

## 1.11.2 Praktek

### 1.11.2.1 Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

1. Pastikan anda telah menginstall anaconda lalu buka aplikasi Anaconda Prompt
2. Lalu pastikan anda telah menginstall python
3. Pada Anaconda Prompt install scikit dengan cara conda install scikit-learn

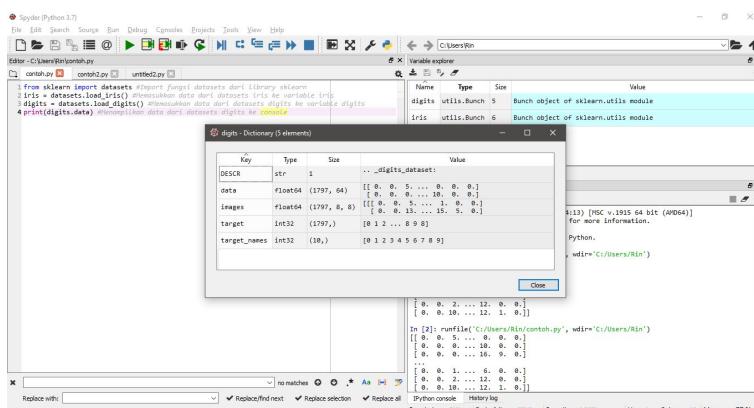


```
(base) C:\Windows\system32>conda install scikit-learn
```

Gambar 1.51 Instalasi Scikit Dari Anaconda Prompt

4. Lalu tulis kode yang ada dibawah ini dan run menggunakan spyder

```
1 from sklearn import datasets #Import fungsi datasets dari
   library sklearn
2 iris = datasets.load_iris() #Memasukkan data dari datasets
   iris ke variable iris
3 digits = datasets.load_digits() #Memasukkan data dari
   datasets digits ke variable digits
4 print(digits.data) #Menampilkan data dari datasets digits ke
   console
```



**Gambar 1.52** Running Kode dari Spyder dan Hasil Variable Explorer

```

In [2]: runfile('C:/Users/Rin/contoh.py', wdir='C:/Users/Rin')
[[ 0.  0.  5. ... 0.  0.  0.]
 [ 0.  0.  0. ... 10. 0.  0.]
 [ 0.  0.  0. ... 16. 9.  0.]
 ...
 [ 0.  0.  1. ... 6.  0.  0.]
 [ 0.  0.  2. ... 12. 0.  0.]
 [ 0.  0.  10. ... 12. 1.  0.]]
    
```

**Gambar 1.53** Running Loading an example dataset dari Spyder

### 1.11.2.2 Loading an example dataset

- Import fungsi datasets dari library sklearn

```

1 from sklearn import datasets #Import fungsi datasets dari
    library sklearn
    
```

- Memasukkan data dari datasets iris ke variable iris

```

1 iris = datasets.load_iris() #Memasukkan data dari datasets
    iris ke variable iris
    
```

- Memasukkan data dari datasets digits ke variable digits

```

1 digits = datasets.load_digits() #Memasukkan data dari
    datasets digits ke variable digits
    
```

- Menampilkan data dari datasets digits ke console

```

1 print(digits.data) #Menampilkan data dari datasets digits ke
    console
    
```

```
>>> from sklearn import datasets
```

Gambar 1.54 Hasil Running kode loading an example dataset

#### 1.11.2.3 Learning and predicting

- Buka Anaconda Prompt



Gambar 1.55 Anaconda Prompt

- Lalu kita import datasets dari sklearn seperti dibawah ini

```
>>> from sklearn import datasets
```

Gambar 1.56 Menggunakan datasets

- lalu kita mendefinisikan iris dan digits menjadi variable

```
>>> iris = datasets.load_iris()
```

Gambar 1.57 mendefinisikan iris

```
>>> digits = datasets.load_digits()
```

Gambar 1.58 mendefinisikan digits

- Lalu kita import svm dari sklearn yang nantinya digunakan untuk menjadi estimasi angka kita

```
>>> from sklearn import svm
```

**Gambar 1.59** Menggunakan `svm`

- Lalu, kita definisikan `clf` sebagai classifier, disini gamma didefinisikan secara manual

```
>>> clf = svm.SVC(gamma=0.001, C=100.)
```

**Gambar 1.60** Mendefinisikan Classifier

- Estimator `clf` (for classifier) pertama kali dipasang pada model. Ini dilakukan dengan melewati training set ke metode `fit`. Untuk training set, akan menggunakan semua gambar dari set data yang ada, kecuali untuk gambar terakhir, yang dicadangkan untuk prediksi. Pada skrip dibawah memilih training set dengan sintaks Python `[:-1]`, yang menghasilkan array baru yang berisi semua kecuali item terakhir dari digits.

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

**Gambar 1.61** Memanggil Classifier

- Menunjukkan prediksi angka baru

```
>>> clf.predict(digits.data[-1:])
array([8])
```

**Gambar 1.62** Prediksi nilai baru

```
1 lastlinelastline
2 from sklearn import datasets #Import fungsi datasets dari library
   sklearn
3 iris = datasets.load_iris() #Memasukkan data dari datasets iris
   ke variable iris
4 digits = datasets.load_digits() #Memasukkan data dari datasets
   digits ke variable digits
```

```

5 from sklearn import svm #Mengimport sebuah Support Vector Machine
6   (SVM) yang merupakan algoritma classification yang akan
7   diambil dari Scikit-Learn.
8 clf = svm.SVC(gamma=0.001, C=100.) #Mendeklarasikan suatu value
9   yang bernama clf yang berisi gamma.
10 clf.fit(digits.data[:-1], digits.target[:-1]) #Estimator clf (for
11 classifier)
12 hasil = clf.predict(digits.data[-1:]) #Menunjukkan prediksi
13 angka baru
14 print(hasil)

```

#### 1.11.2.4 Model Persistance Model Persistance

```

1 lastlinelastline
2 %%Cara Dump Pertama
3 from sklearn import svm #Mengimport sebuah Support Vector
4   Machine(SVM) yang merupakan algoritma classification yang
5   akan diambil dari Scikit-Learn.
6 from sklearn import datasets #Import fungsi datasets dari library
7   sklearn
8 clf = svm.SVC() #Mendefinisikan clf dengan fungsi svc dari
9   library svm
10 X, y = datasets.load_iris(return_X_y=True) #Mengisi variable x
11   dan y dengan data dari datasets
12 clf.fit(X, y) #Estimator clf (for classifier)
13
14
15 %%Cara Dump Kedua
16 from sklearn import svm # Digunakan untuk memanggil class svm
17   dari library sklearn
18 from sklearn import datasets # Diguakan untuk class datasets
19   dari library sklearn
20 clf = svm.SVC()           # membuat variabel clf , dan
21   memanggil class svm dan fungsi SVC
22 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
23   iris dan mengembalikan nilainya.
24 clf.fit(X, y)             #Perhitungan nilai label
25
26 from joblib import dump, load #memanggil class dump dan load pada
27   library joblib
28 dump(clf, '1174083.joblib') #Menyimpan model kedalam 1174066.
29   joblib
30 hasil = load('1174083.joblib') #Memanggil model 1174066
31 hasil.predict(X[0:1])
32 print(y[0]) # Menampilkan Model yang dipanggil sebelumnya

```

#### 1.11.2.5 Conventions Conversion

```

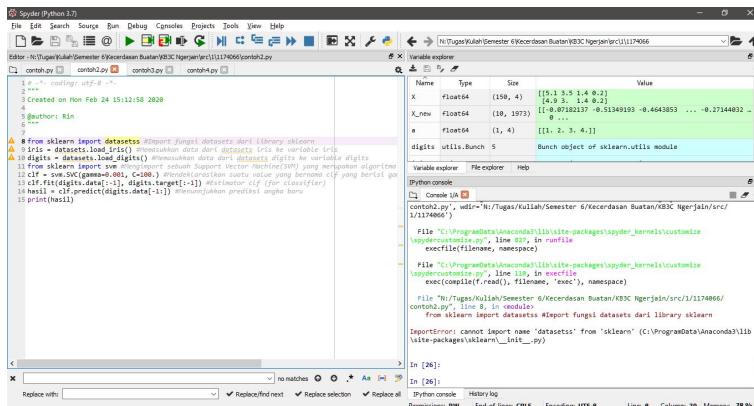
1 lastlinelastline
2 import numpy as np # memanggil library numpy dan dibuat alias np
3 from sklearn import random_projection #Memanggil class
        random_projection pada library sklearn
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
        mendefinisikan np, fungsi random dan attr RandomState kedalam
        variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan nilai
        random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai random
        sebelumnya, kedalam array, dan menentukan typedatanya sebagai
        float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
        membuat variabel transformer , dan mendefinisikan
        classrandom_projection dan memanggil fungsi
        GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru dan
        melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64
13 print(X_new) # Menampilkan isi variabel X_new

```

### 1.11.3 Penanganan Error

Dari percobaan yang dilakukan di atas, apabila mendapatkan error maka:

#### 1. Screenshoot Error



**Gambar 1.63** ImportError: cannot import name 'datasetss' from 'sklearn'

#### 2. Tuliskan kode eror dan jenis errornya [hari ke 2](10)

- ImportError

3. Solusi pemecahan masalah error tersebut [hari ke 2] (10)

- ImportError

Cek kembali jika ada yang typo

#### 1.11.4 Bukti Tidak Plagiat



Gambar 1.64    Bukti Tidak Melakukan Plagiat Chapter 1

#### 1.11.5 Link Youtube

<https://youtu.be/w4lTVoumb1g>

### 1.12 Nurul Izza Hamka - 1174062

#### 1.12.1 Pemahaman Teori

1. Definisi Sejarah Kecerdasan Buatan

Kecerdasan buatan atau yang dikenal dengan Artificial Intelligence (AI) adalah suatu perkembangan teknologi yang muncul untuk membentuk suatu mesin teknologi yang lebih pintar yang mana agar lebih memudahkan setiap pekerjaan manusia. Selain itu AI ini juga untuk memahami kecerdasan dalam artian membuat sebuah mesin yang dapat membantu memahami kecerdasan contohnya dapat memecahkan sebuah masalah dengan lebih cepat.

2. Definisi Kecerdasan Buatan

Kecerdasan buatan atau yang disebut dengan Artificial Intelligence mulai muncul sekitar tahun 1940 dan 1950 sejak adanya komputer. Munculnya AI ini memberikan banyak keuntungan seperti AI ini berisfat permanen, artinya bisa digunakan secara berulang-ulang dimana saja dan kapan saja. Selain itu menawarkan kemudahan dalam artian data yang telah disimpan sebelumnya akan mudah untuk di akses kembali. Kerja AI ini juga lebih cepat jika dibandingkan dengan kerja manusia

3. Definisi Perkembangan Kecerdasan Buatan Tahun 1960 s/d 1970, mulailah berbagai diskusi tentang bagaimana komputer dapat menirukan dengan sedetail mungkin kemampuan otak manusia, saat itu dikategorikan dengan "classical AI". Kemudian pada tahun 1980, saat itu komputer sudah mudah didapatkan dengan harga yang terjangkau yang memudahkan berbagai riset dibidang kecerdasan buatan berkembangan dengan pesat di berbagai universitas dunia.

John McCarthy dari Massachusetts Institute of Technology atau yang dikenal sebagai Bapak AI, pada tahun 1956 McCarthy mengadakan konferensi Dartmouth Workshop yang melahirkan suatu bidang baru dengan nama "Artificial Intelligence". Pada konferensi Dartmouth itu mempertemukan semua para pendiri AI, dimana John McCarthy yang mengusulkan defisi dari AI itu. AI adalah cabang dari ilmu komputer yang berfokus pada pengembangan komputer yang dapat memiliki kemampuan layaknya manusia.

4. Definisi supervised learning Supervised learning mempunyai input dan output yang bisa dibuat menjadi model hubungan matematis, dan juga sebuah pendekatan dimana sudah terdapat data yang dilatih selain itu juga ada sebuah variable yang sudah ditargetkan sebagai tujuan dari pendekatan ini yaitu pengelompokan data ke data yang sudah ada sebelumnya.

Pengertian dalam konteks AI, supervised learning adalah sistem dimana sebuah input dan output data yang kita inginkan sudah tersedia. Input dan output data ini diberi label untuk klasifikasi dasar pembelajaran untuk pemrosesan data yang akan datang. Supervised learning ini menyediakan algoritma untuk pembelajaran dengan jumlah diketahui untuk mendukung sebuah penilaian yang akan datang seperti: Regresi Linear Berganda, Analisis Deret Waktu, Decision tree dan Random Forest, Artificial Neural Network, dan lain sebagainya.

5. Definisi Klasifikasi Klasifikasi merupakan sebuah proses pengelompokan benda berdasarkan ciri-ciri persamaan dan perbedaan. Artinya kita memberitahu mesin tersebut bagaimana cara pengerjaannya berdasarkan kelompok.
6. Definisi Regresi Regresi adalah bagian dari problem Supervised Learning, regresi ini menggunakan metode statistika.
7. Definisi Unsupervised Learning Unsupervised Learning berbeda dengan supervised learning. Unsupervised Learning tidak memiliki data latih, sehingga dari data yang telah ada kita kelompokkan menjadi dua atau tiga bagian begitupun seterusnya. Unsupervised Learning ini merupakan pelatihan algoritma kecerdasan buatan emnggunakan informasi

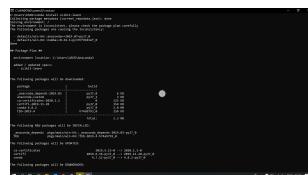
yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa panduan.

Tujuan dari algoritma tersebut adalah untuk mengelompokkan sebuah objek yang hampir mirip atau sama ke dalam area tertentu

8. Definisi Data Set Dataset merupakan objek yang merepresentasikan sebuah data dan relasi yang ada di memory. Struktur data set mirip dengan data yang ada didalam sebuah database. Namun, didalam dataset berisi sebuah koleksi dari data tabel dan data relation.
9. Definisi Training Set Training set merupakan set yang digunakan oleh algoritma klasifikasi. Contohnya adalah decision tree, bayesian, neural network, dan lain sebagainya.
10. Testing Set Testing set merupakan sebuah set yang digunakan untuk mengevaluasi sejauh mana sebuah classifier berhasil melakukan klasifikasi dengan benar. Testing set berfungsi sebagai materai persetujuan, tapi tidak dapat kita gunakan sampai akhir. Setelah data di optimalkan, kita dapat melakukan pengujian jaringan saraf terhadap pengambilan sampel acak. Kemudian hasil yang diperoleh harus valid bahwa jaringan kita akurat dalam menegnali gambar.

### 1.12.2 Instalasi

1. Instalasi Library Scikit Anaconda



**Gambar 1.65** Instalasi Package Scikit Learn

|       | Name     | Type       | Size  | Notes |
|-------|----------|------------|---|-------|
| X     | fitest32 | (1K, 2800) | {[0.5464315, 0.7151834, 0.0077035, ..., 0.4900778, 0.5428466, 0.5817731, ...] |       |
| X_new | fitest34 | (1K, 375)  | {[0.5343848, -0.4807478, 1.1488481, ..., 0.4288133, 0.79515418]               |       |
| a     | fitest34 | (1, 4)     | {[(1, 2, 3, 4)]}  |       |
| ans1  | fitest32 | (1,)       | {[]}  |       |
| ans2  | fitest34 | (1,)       | {[]}  |       |
| ans3  | fitest34 | (1,)       | {[1.5, 2.5, 3.4, 4.2]}  |       |
| x     | fitest34 | (100, 4)   | {[2.9, 3, 1.4, 2]}  |       |
| y     | fitest32 | (100,)     | {[0.9, ..., 2.2]}}  |       |

**Gambar 1.66** Hasil Variabel Explorer

2. Mencoba Loading an Example Dataset

```

1 #%% Mencoba loading an Example Dataset
2 from sklearn import datasets #digunakan untuk memanggil class
   dataset dari library sklearn
3 iris = datasets.load_iris() # artinya kita menggunakan
   dataset iris
4 x = iris.data # artinya kita menyimpan data set iris di
   variable x
5 y = iris.target #artinya kita menyimpan data label iris pada
   variavle y

```

### 3. Mencoba Learning and Predicting

```

1 #%%Mmencoba Learning dan Predicting
2 from sklearn.neighbors import KNeighborsClassifier # kta
   menggunakan fungsi KNighborsClassifer pada kelas sklearn
   dan libarry sklearn
3 knn=KNeighborsClassifier(n_neighbors=1) # kita membuat
   variable knn, dan memanggil function KNighbors
4
5 knn.fit(x,y) #kita membuat perhitungan matematika library knn
6 a=np.array([1.0 ,2.0 ,3.0 ,4.0]) # artinya kita membuat array
7 a = a.reshape(1,-1) # mengubah bentuk array jadi 1
   dimensi
8 hasil = knn.predict(a) #kita memanggil fungsi prediksi
9 print(hasil) #menampilkan hasil prediksi

```

### 4. Mencoba Model Persintence

```

1 #%% Model Persistense
2 from sklearn import svm #untuk memanggil class svm dari
   library sklearn
3 from sklearn import datasets #untuk class dataset dati
   library sklearn
4 clf = svm.SVC() #kita membuat variable clf, dan memanggil
   class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Memanggil dataset
   iris dan mengembalikan nilainya
6 clf.fit(X, y) # untuk menampilkan model yang dipanggil sebelumnya
7
8 from joblib import dump, load #memanggil class dump dan load
   pada library joblib
9 dump(clf, '1174062.joblib') #menyimpan model kedalam 1174062.
   joblib
10 hasil = load('1174062.joblib') #memanggil model 1174062
11 print(hasil) #untuk menampilkan model yang dipanggil
   sebelumnya

```

### 5. Mencoba Conventions

```

1 #%% mencoba Conventions

```

```

2 import numpy as np #digunakan untuk memanggil library numpy
    dan dibuat alias np
3 from sklearn import random_projection #memanggil class
    random_projection pada library sklearn
4
5 rng = np.random.RandomState(0) #untuk membuat variable rng,
    mendefinisikan np, function random dan attr randomstate
    kedalam variable
6 X = rng.rand(10, 2000) #membuat variable X, dan menentukan
    nilai random dari 10–2000
7 X = np.array(X, dtype='float32') #untuk menyimpan hasil nilai
    random senelumnya, kedalam array dan menentukan
    typedatanya sebagai float32
8 X.dtype #untuk mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
        membuat varibale transformer dan mendefinisikan
        classrandom_projection dan memanggil fungsi
        GaussianRandomProjection
11 X_new = transformer.fit_transform(X) #untuk membuat variable
        baru serta melakukan perhitungan label pada variabel X
12 X_new.dtype #mengubah data tipe menjadi float64
13 print(X_new) #menampilkan isi variable X_new

```

### 1.12.3 Penanganan Error

(a) Hasil ScreenShoot Error

```

file "D:\Kuliah\Semester 6\Kecerdasan Buatan\Bahan\1\sklearn.py", line 8, in <module>
    from sklearn import datasets
ImportError: cannot import name 'datasets' from 'sklearn' (D:\Kuliah\Semester
6\Kecerdasan Buatan\Bahan\1\sklearn.py)

```

Gambar 1.67 Import Error

(b) Tuliskan Kode Error dan Jenis Error

- Import Error

(c) Cara Penangan Error

- Import Error  
Dengan Menginstall Library Yang Tidak Ditemukan

### 1.12.4 Bukti Tidak Melakukan Plagiat



**Gambar 1.68**    Bukti Tidak Plagiat

### 1.12.5 Link Youtube

- (d) Link Youtube Chapter 1  
<https://www.youtube.com/watch?v=AdvP3Y18GE4>

## 1.13 Alvan Alvanzah (1174077)

### 1.13.1 Teori

#### 1. Definisi, Sejarah dan Perkembangan Kecerdasan Buatan

Definisi kecerdasan buatan itu sendiri adalah suatu sistem teknologi yang didalamnya ditambahkan kecerdasan oleh manusia, kecerdasan buatan diatur dan dikembangkan dalam konteks ilmiah, dan bentukan dari kecerdasan entitas ilmiah yang ada. Kecerdasan Buatan atau dalam Bahasa Inggris sering disebut Artificial Intelligence yang sering disebut juga sebagai AI, pada 10 tahun lalu masyarakat belum terlalu mengetahui hal tersebut dan masih menjadi bahan candaan dikalangan masyarakat. Awal perkembangan AI dimulai pada tahun 1952-1969 yang dimulai dengan kesuksesan Newell dan temannya Simon menggunakan sebuah program yang disebut dengan General Problem Solver. Program ini dibangun untuk tujuan penyelesaian masalah secara manusiawi. Pada tahun 1966-1974 perkembangan kecerdasan buatan mulai melambat. Ada 3 faktor utama yang menyebabkan hal itu terjadi:

- Banyak subjek pada program AI yang bermunculan hanya mengandung sedikit atau bahkan sama sekali tidak mengandung sama sekali pengetahuan (knowledge).
- Kecerdasan buatan harus bisa menyelesaikan banyak masalah.
- Untuk menghasilkan perlakuan intelijensia ada beberapa batasan pada struktur yang bisa digunakan.

#### 2. Definisi

Supervised learning, klasifikasi, regresi, unsupervised learning, dataset, trainingset dan testingset.

- Supervised Learning

Supervised Learning merupakan sebuah tipe learning yang mempun-

yai variable input dan variable output, tipe ini juga menggunakan satu algoritma atau lebih dari satu algoritma yang digunakan untuk mempelajari fungsi pemetaan dari input ke output.

- **Klasifikasi**

Klasifikasi adalah pengelompokan data di mana data yang digunakan memiliki label atau kelas target. Sehingga algoritma untuk menyelesaikan masalah klasifikasi dikategorikan ke dalam pembelajaran ter-bimbing.

- **Regresi**

Regresi metode analisis statistik yang digunakan untuk dapat melihat efek antara dua atau lebih variabel. Hubungan variabel dalam pertanyaan adalah fungsional yang diwujudkan dalam bentuk model matematika. Dalam analisis regresi, variabel dibagi menjadi dua jenis, yaitu variabel respons atau yang biasa disebut variabel dependen dan variabel independen atau dikenal sebagai variabel independen. Ada beberapa jenis analisis regresi, yaitu regresi sederhana yang mencakup linear sederhana dan regresi non-linear sederhana dan regresi berganda yang mencakup banyak linier atau non-linear berganda. Analisis regresi digunakan dalam pembelajaran mesin pembelajaran dengan metode pembelajaran terawasi.

- **Unsupervised Learning**

Unsupervised Learning jenis pembelajaran di mana kita hanya memiliki data input (input data) tetapi tidak ada variabel output yang terkait. Tujuan dari pembelajaran tanpa pengawasan adalah untuk memodelkan struktur dasar atau distribusi data dengan tujuan mempelajari data lebih lanjut, dengan kata lain, itu adalah fungsi simpulan yang menggambarkan atau menjelaskan data.

- **Data Set**

Data Set objek yang merepresentasikan data dan relasinya di memory. Strukturnya mirip dengan data di database. Dataset berisi koleksi dari datatable dan datarelation.

- **Training Set**

Training Set adalah bagian dari dataset yang di latih untuk membuat prediksi atau menjalankan fungsi dari algoritma ML lain sesuai dengan masing-masing. Memberikan instruksi melalui algoritma sehingga mesin yang di praktikkan dapat menemukan korelasinya sendiri.

- **Testing Set**

testing set adalah bagian dari dataset yang kami uji untuk melihat akurasinya, atau dengan kata lain untuk melihat kinerjanya.

### 1.13.2 Praktek

- Instalasi Library scikit dari ianaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer



**Gambar 1.69** Instalasi Package Scikit Learn

| Name  | Type          | Size       | Value  |
|-------|---------------|------------|--|
| X     | float64       | (16, 2000) | [0.54683359 0.71531034 0.0829339 ... -0.4003976  |
| X_new | float64       | (16, 1973) | [0.64386044 0.3927731 ... 0.00000000 0.00000000 0.00000000 ... -0.79318666 0.31266443 ... -1.06165433 -0.00000000] |
| a     | float64       | (1, 4)     | [1. 1. 1. 1.]  |
| iris  | utilist.Bunch | 6          | Բառապահ օբյեկտ sklearn.utils module  |
| x     | float64       | (150, 4)   | [5.0 3.4 1.5 0.2] [4.6 3.0 1.4 0.2] ...  |
| y     | int32         | (150,)     | [0 0 ... 2 2 2]  |

**Gambar 1.70** Isi Variabel Explorer

- Mencoba loading an example dataset

```

1 %% Mencoba loading an example dataset
2 from sklearn import datasets # Digunakan Untuk Memanggil
   class datasets dari library sklearn
3 iris = datasets.load_iris() # Menggunakan contoh datasets
   iris
4 x = iris.data           # Menyimpan nilai data sets iris
   pada variabel x
5 y = iris.target         # Menyimpan nilai data label iris
   pada variabel y

```

- Mencoba Learning dan predicting

```

1 %%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Digunakan
   Untuk Memanggil fungsi KNeighborsClassifier
3                                     # pada
   class sklearn dan library sklearn
4 import numpy as np # memanggil library numpy dan dibuat alias
   np
5 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
   , dan memanggil fungsi KNeighborsClassifier
6                                     #dan mendefinisikan k
   -nya adalah 1
7 knn.fit(x,y)                      #Perhitungan
   matematika library kkn

```

```

8 a=np.array([1.0 ,2.0 ,3.0 ,4.0])          #Membuat Array
9 a = a.reshape(1,-1)                         #Mengubah Bentuk
    Array jadi 1 dimensi
10 hasil = knn.predict(a)                     #Memanggil fungsi
     prediksi

```

#### 4. Mencoba Model Persistence

```

1 %% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm
    dari library sklearn
3 from sklearn import datasets # Diguankan untuk class datasets
    dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan
    memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
    iris dan mengembalikan nilainya.
6 clf.fit(X, y)             #Perhitungan nilai label
7
8 from joblib import dump, load #memanggil class dump dan load
    pada library joblib
9 dump(clf, '1174077.joblib') #Menyimpan model kedalam 1174027.
    joblib
10 hasil = load('1174077.joblib') #Memanggil model 1174027

```

#### 5. Mencoba Conventions

```

1 %% Conventions
2 import numpy as np # memanggil library numpy dan dibuat alias
    np
3 from sklearn import random_projection #Memanggil class
    random_projection pada library sklean
4
5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
    mendefisikan np, fungsi random dan attr RandomState
    kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
    nilai random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
    random sebelumnya, kedalam array , dan menentukan
    typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
    membuat variabel transformer , dan mendefinisikan
    classrandom_projection dan memanggil fungsi
    GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru
    dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64

```

### 1.13.3 Penanganan Error

#### 1. ScreenShoot Error



**Gambar 1.71** Import Error



**Gambar 1.72** Value Error

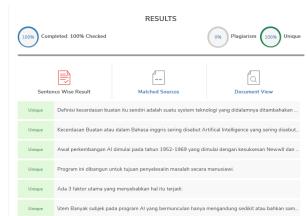
#### 2. Tuliskan Kode Error dan Jenis Error

- Import Error
- Value Error

#### 3. Cara Penanganan Error

- Import Error  
Dengan Menginstall Library Yang Tidak Ditemukan
- Value Error  
Mengubah Bentuk Arraynya, Menjadi 1 Dimensi

### 1.13.4 Bukti Tidak Plagiat



**Gambar 1.73** Bukti Tidak Melakukan Plagiat Chapter 1

## 1.14 Difa Al Fansha

### 1.14.1 Teori

#### 1. Definisi, Sejarah dan Perkembangan Kecerdasan Buatan

##### ▪ Definisi Kecerdasan Buatan

Kecerdasan Buatan biasa disebut dengan istilah AI (Artificial Intelligence). AI sendiri merupakan suatu cabang dalam bisnis sains komputer sains dimana mengkaji tentang bagaimana cara untuk lengkapi sebuah komputer dengan kemampuan atau kepintaran layaknya atau mirip dengan yang dimiliki manusia. Sebagai contoh, sebagaimana komputer dapat berkomunikasi dengan pengguna baik menggunakan kata, suara maupun lain sebagainya. Dengan kemampuan ini, diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemuiinya kemudian itulah yang disebut dengan kecerdasan buatan.

Kecerdasan Buatan adalah salah satu bidang studi yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara lebih manusiawi dan lebih bisa dipahami oleh manusia. Kecerdasan buatan makin canggih dengan kemampuan komputer dalam memperbarui pengetahuannya dengan testing dan perkembangan target analisa.

##### ▪ Sejarah Kecerdasan Buatan

Pada tahun 1956, McCarthy yang sama mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Peneliti terkemuka dalam teori kompleksitas, simulasi bahasa, hubungan antara keacakan dan pemikiran kreatif, jaringan saraf diundang. Tujuan dari bidang penelitian yang baru dibuat adalah untuk mengembangkan mesin yang dapat mensimulasikan setiap aspek kecerdasan. Itulah sebabnya Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan. Sejak saat itu, Kecerdasan Buatan telah hidup melalui decade kemuliaan dan cemoohan, yang dikenal luas sebagai musim panas dan musim dingin AI. Musim panasnya ditandai dengan optimism dan dana besar, sedangkan musim dinginnya dihadapkan dengan pemotongan dana, ketidakpercayaan dan pesimisme.

Artificial intelligence merupakan inovasi baru di bidang ilmu pengetahuan. Mulai terbentuk sejak adanya komputer modern dan kira-kira terjadi sekitaran tahun 1940 dan 1950. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi tingkah

laku cerdas dalam sistem kecerdasan komputer. Pada awal 50-an, studi tentang “mesin berpikir” memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan innformasi. Pada tahun 1956, para ilmuan jenius seperti Alan Turing, Norbert, Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen dibidang cybernetics, matematika, algoritma dan teori jaringan. Namun, seprang ilmuan komputer dan kognitif John McCarthy adalah orang yang dating dengan ide untuk bergabung dengan upaya penelitian terpisah ini kedalam satu bidang yang akan mempelajari topic baru untuk imajinasi manusia yaitu kecerdasan buatan. Dia adalah orang yang menciptakan istilah tersebut dan kemudian mendirikan laboratorium Kecerdasan Buatan di MIT dan Stan ford.

- Perkembangan Kecerdasan Buatan

Teknologi Artificial Intelligence semakin ramai dibahas dalam berbagai diskusi teknologi di seluruh dunia. Menurut kebanyakan orang, pekerjaan seperti kasir, operator telepon, pengendara truk, dan lainnya sangat berpeluang besar untuk tergantikan oleh Artificial Intelligence. Mengapa terjadi hal demikian? dikarenakan memang bahwa AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Namun, dalam beberapa aspek memang pekerja manusia masih unggul dibandingkan AI itu sendiri. Para generasi muda yang ada di dunia terutama di daerah Asia terlihat sudah memahami fungsi dan efek dari AI dalam kehidupan kita sehari-hari. Berdasarkan survei yang dilakukan oleh Microsoft, terdapat 39 persen responden yang mempertimbangkan untuk menggunakan mobil tanpa pengemudi dan 36 persen lainnya setuju bahwa robot masa depan dengan software untuk beroperasi mampu meningkatkan produktivitas. Dari survey tersebut kita sebagai pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tanpa memberikan efek samping terhadap etos kerja dan keseharian kita sebagai pengguna dalam kehidupan sehari-hari.

AI Summer 1 (1956-1973) KOnferensi Dartmounth diikuti oleh 17 tahun kemajuan luar biasa. Proyek penelitian yang dilakukan di MIT, universitas di Edinburgh, Stanford dan Carnegie Mellon menerima dana besar-besaran, yang akhirnya membawa hasil. Selama tahun-tahun itulah komputer pemrograman mulai melakukan masalah aljabar, membuktikan teorema geometris, memahami dan menggunakan sintaks dan tata bahasa Inggris. Terlepas dari ditinggalkannya koneksiisme dan terjemahan mesin yang gagal, yang menunda penelitian Natural Language Processing (NLP) selama bertahun-tahun, banyak prestasi dari masa lalu yang membuat sejarah. Berikut ini beberapa diantaranya : Pelopor pembelajaran mesin, Ray Solomonoff mele-

takkan dasar-dasar teori metematika AI, memperkenalkan metode Bayesian universal untuk inferensi dan preddiksi induktif Thomas Evans menciptakan program ANALOGI heuristik, yang memungkinkan komputer memecahkan masalah geometrianalogi Unimation, perusahaan robotika pertama didunia, menciptakan robot industri Unimate, yang bekerja pada jalur perakitan modil Genenral Motors. Joseph Weizenbaum membangun ELIZA-program interaktif yang dapat membawa percakapan dalam bahasan Inggris tentang topik apapun. Ross Quillian menunjukkan jaring semantik, sedangkan Jaime Carbonell (Sr.) mengembangkan Cendikia-program interaktif untuk instruksi yang dibantu komputer berdasarkan jaring semantik. Edward Feigenbaum dan Julian Feldman menerbitkan Computeks and Thought, kumpulan artikel pertama tentang AI.

## 2. Definisi Supervised Learning, Klasifikasi, Regresi, Unsupervised Learning, Data Set, Training Set dan Testing Set

### ▪ Definisi Supervised Learning

Supervised Learning adalah tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan(juga disebut sinyal pengawasan super). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan, yang dapat digunakan untuk memetakan contoh-contoh baru. Skenario optimal akan memungkinkan algoritma menentukan label kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi dari data pelatihan untuk situasi yang tidak terlihat dengan cara yang "masuk akal". Supervised Learning menyediakan algoritma pembelajaran dengan jumlah yang diketahui untuk mendukung penilaian dimasa depan. Chatbots, mobil self-driving, program pengenalan wajah, sistem pakar dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak diawasi. Supervised Learning sebagian besar terkait dengan AI berbasis pengambilan tetapi mereka juga mungkin mampu menggunakan model pembelajaran generatif. Data pelatihan untuk pembelajaran yang diawasi mencakup serangkaian contoh dengan subjek input berpasangan dan output yang diinginkan (yang juga disebut sebagai sinyal pengawasan).

### ▪ Klasifikasi

Klasifikasi adalah pembagian sesuatu menurut kelas-kelas ( class ). Menurut Ilmu Pengetahuan, Klasifikasi merupakan proses pengelom-

pokok benda berdasarkan ciri-ciri persamaan dan juga perbedaan. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah. Label ( $y$ ) umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran mesin dan statistik, klasifikasi adalah pendekatan pembelajaran yang diawasi di mana program komputer belajar dari input data yang diberikan kepadanya dan kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pengamatan baru. Kumpulan data ini mungkin hanya bersifat dua kelas (seperti mengidentifikasi apakah orang tersebut berjenis kelamin laki-laki atau perempuan atau bahwa surat itu spam atau bukan-spam) atau mungkin juga multi-kelas. Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, identifikasi metrik, klasifikasi dokumen dll.

- Regresi

Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel. Regresi adalah membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan makan, yang paling sederhana adalah regresi linier. Ia mencoba untuk menyesuaikan data dengan hyper-plane terbaik yang melewati poin.

- Unsupervised Learning

Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Dalam Unsupervised Learning, sistem AI dapat mengelompokkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan.

- Data set

Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory. Strukturnya mirip dengan data di database, namun bedanya dataset berisi koleksi dari data table dan data relation. mendapatkan data yang tepat berarti mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda prediksi; yaitu data yang berisi sinyal tentang peristiwa yang Anda pedulikan. Data harus diselaraskan dengan masalah yang Anda coba selesaikan. Gambar kucing tidak terlalu berguna ketika Anda sedang membangun sistem identifikasi wajah. Memverifikasi bahwa data selaras dengan masalah yang ingin Anda selesaikan harus dilakukan oleh ilmuwan data. Jika Anda tidak memiliki data yang tepat, maka upaya Anda

untuk membangun solusi AI harus kembali ke tahap pengumpulan data.

- Training Set

Training Set adalah set digunakan oleh algoritma klasifikasi . Dapat dicontohkan dengan : decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier. Menjalankan pelatihan yang diatur melalui jaringan saraf mengajarkan pada net cara menimbang berbagai fitur, menyesuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan dalam hasil Anda. Koefisien-koefisien tersebut, juga dikenal sebagai parameter, akan terkandung dalam tensor dan bersama-sama mereka disebut model, karena mereka mengkodekan model data yang mereka latih. Mereka adalah takeaways paling penting yang akan Anda dapatkan dari pelatihan jaringan saraf.

- Testing set

Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Ini berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakan sampai akhir. Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda terhadap pengambilan sampel acak akhir ini. Hasil yang dihasilkannya harus memvalidasi bahwa jaringan Anda secara akurat mengenali gambar, atau mengenalinya setidaknya [x] dari jumlah tersebut. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan, lihat hyperparameter yang Anda gunakan untuk menyetel jaringan, serta kualitas data Anda dan lihat teknik pra-pemrosesan Anda.

### 1.14.2 Praktek

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

- Buka prompt anaconda lalu ketikkan conda scikit-learn
- Biarkan proses berjalan hingga selesai

```
(base) C:\Users\DiffaAl21>conda install scikit-learn
```

Gambar 1.74 Install library scikit

```
(base) C:\Users\Diffa121>conda install scikit-learn
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\Diffa121\Anaconda3

added / updated specs:
- scikit-learn

The following packages will be downloaded:

  package          build
ca-certificates-2020.1.1      0         165 KB
conda-4.8.2                   py37_0     3.1 MB
scikit-learn-0.20.3           py37h343c172_0   4.7 MB
                                           Total:    7.9 MB

The following packages will be UPDATED:

  ca-certificates              2019.11.27-0 --> 2020.1.1-0
  conda                         4.7.12-py37_0 --> 4.8.2-py37_0
  scikit-learn                  0.20.1-py37h343c172_0 --> 0.20.3-py37h343c172_0

Proceed ([y]/n)?
```

Downloading and Extracting Packages
ca-certificates-2020 | 165 KB | #####| 100%
conda-4.8.2 | 3.1 MB | #####| 100%
scikit-learn-0.20.3 | 4.7 MB | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

**Gambar 1.75** Proses jika berhasil

2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

- Ketikkan script berikut di spyder, lalu run

```
1 #Import fungsi datasets dari library sklearn
2 from sklearn import datasets
3
4 #Memasukkan data dari datasets iris ke variable iris
5 iris = datasets.load_iris()
6
7 #Memasukkan data dari datasets digits ke variable digits
8 digits = datasets.load_digits()
9
10 #Menampilkan data dari datasets digits ke console
11 print(digits.data)
```

```
In [1]: runfile('E:/Difa/Tugas Kuliah/Tingkat 3/Semester 6/Kecerdasan Buatan/src1/test.py', wdir='E:/Difa/Tugas Kuliah/Tingkat 3/Semester 6/Kecerdasan Buatan/src1')
[[ 0.  0.  5. ... 0.  0.  0.]
 [ 0.  0.  0. ... 10. 0.  0.]
 [ 0.  0.  0. ... 16. 9.  0.]
 ...
 [ 0.  0.  1. ... 6.  0.  0.]
 [ 0.  0.  2. ... 12. 0.  0.]
 [ 0.  0. 10. ... 12. 1.  0.]]
In [2]: |
```

Gambar 1.76 hasil test.py

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan perbaris

```
1 #Import fungsi datasets dari library sklearn
2 from sklearn import datasets
3
4 #Memasukkan data dari datasets iris ke variable iris
5 iris = datasets.load_iris()
6
7 #Memasukkan data dari datasets digits ke variable digits
8 digits = datasets.load_digits()
9
10 #Mengimport sebuah Support Vector Machine(SVM) yang merupakan
11 #algoritma classification yang akan diambil dari Scikit –
12 #Learn.
13 from sklearn import svm
14 #Mendeklarasikan suatu value yang bernama clf yang berisi
15 #gamma.
16 clf = svm.SVC(gamma=0.001, C=100.)
17 #Estimator clf (for classifier)
18 clf.fit(digits.data[:-1], digits.target[:-1])
19
20 #Menunjukkan prediksi angka baru
21 hasil = clf.predict(digits.data[-1:])
22
23 #Menampilkan
24 print(hasil)
```

| Name   | Type        | Size  | Value                                |
|--------|-------------|-------|--------------------------------------|
| digits | utils.Bunch | 5     | Bunch object of sklearn.utils module |
| hasil  | int32       | (1, ) | [8]                                  |
| iris   | utils.Bunch | 6     | Bunch object of sklearn.utils module |

Gambar 1.77 variabel explorer test2.py

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```
1 %%Cara Dump Pertama
2
3
4
5 #Mengimport sebuah Support Vector Machine(SVM) yang merupakan
   algoritma classification yang akan diambil dari Scikit-
   Learn.
6 from sklearn import svm
7 #Import fungsi datasets dari library sklearn
8 from sklearn import datasets
9
10 #Mendefinisikan clf dengan fungsi svc dari library svm
11 clf = svm.SVC()
12
13 #Mengisi variable x dan y dengan data dari datasets
14 X, y = datasets.load_iris(return_X_y=True)
15
16 #Estimator clf (for classifier)
17 clf.fit(X, y)
18
19 #Mengimport Library pickle
20 import pickle
21
22 #Menyimpan hasil dari clf kedalam sebuah dump
23 s = pickle.dumps(clf)
24
25 #Memanggil dump yang dihasilkan pickle lalu memasukkan hasil
   dumpnya ke variable
26 clf2 = pickle.loads(s)
27
28
29 #Memprediksi angka yang akan muncul
30 clf2.predict(X[0:1])
31
32 #Menampilkan data prediksi
33 print(y[0])
34
35 %%Cara Dump Kedua
36 # Digunakan untuk memanggil class svm dari library sklearn
37 from sklearn import svm
38
39 # Digunakan untuk class datasets dari library sklearn
40 from sklearn import datasets
41
42 # membuat variabel clf , dan memanggil class svm dan fungsi
   SVC
43 clf = svm.SVC()
44
45 #Mengambil dataset iris dan mengembalikan nilainya.
46 X, y = datasets.load_iris(return_X_y=True)
47
48 #Perhitungan nilai label
49 clf.fit(X, y)
```

```

51
52 #memanggil class dump dan load pada library joblib
53 from joblib import dump, load
54
55
56 #Menyimpan model kedalam 1174066.joblib
57 dump( clf , '1174083.joblib' )
58
59 #Memanggil model 1174066
60 hasil = load('1174083.joblib')
61 hasil.predict(X[0:1])
62
63 # Menampilkan Model yang dipanggil sebelumnya

```

| Name   | Type        | Size     | Value                                   |
|--------|-------------|----------|---|
| X      | float64     | (150, 4) | [[5.1 3.5 1.4 0.2]<br>[4.9 3. 1.4 0.2]] |
| digits | utils.Bunch | 5        | Bunch object of sklearn.utils module    |
| iris   | utils.Bunch | 6        | Bunch object of sklearn.utils module    |
| y      | int32       | (150,)   | [0 0 0 ... 2 2 2]                       |

Gambar 1.78 variabel explorer test3.py

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1
2 # memanggil library numpy dan dibuat alias np
3 import numpy as np
4
5 #Memanggil class random_projection pada library sklearn
6 from sklearn import random_projection
7
8 #Membuat variabel rng, dan mendefinisikan np, fungsi random dan
9 #attr RandomState kedalam variabel
10 rng = np.random.RandomState(0)
11
12 # membuat variabel X, dan menentukan nilai random dari 10 –
13 # 2000
14 X = rng.rand(10, 2000)
15
16 #menyimpan hasil nilai random sebelumnya, kedalam array, dan
17 #menentukan typedatanya sebagai float32
18 X = np.array(X, dtype='float32')
19
20 # Mengubah data tipe menjadi float64
21 X.dtype
22
23 #membuat variabel transformer, dan mendefinisikan
#classrandom_projection dan memanggil fungsi
#GaussianRandomProjection

```

```

21 transformer = random_projection.GaussianRandomProjection()
22 # membuat variabel baru dan melakukan perhitungan label pada
23 X_new = transformer.fit_transform(X)
24 # Mengubah data tipe menjadi float64
25 X_new.dtype
26 # Menampilkan isi variabel X_new
27 print(X_new)

```

| Name   | Type        | Size       | Value   |
|--------|-------------|------------|---|
| X      | float32     | (10, 2000) | [[ 0.5488135  0.71518934  0.60276335 ... 0.4801078<br>0.64386404  0.5017731 ... |
| X_new  | float64     | (10, 1973) | [ [ 0.16482812  0.90149112  0.16933195 ... -0.83564827 -0...<br>-0 ...          |
| digits | utils.Bunch | 5          | Bunch object of sklearn.utils module  |
| iris   | utils.Bunch | 6          | Bunch object of sklearn.utils module  |
| y      | int32       | (150,)     | [ 0 0 0 ... 2 2 2]  |

Gambar 1.79 variabel explorer test4.py

```

In [2]: runfile('E:/Difa/Tugas Kuliah/Tingkat 3/Semester 6/Kecerdasan Buatan/src1/test4.py', wdir='E:/Difa/Tugas Kuliah/Tingkat 3/Semester 6/Kecerdasan Buatan/src1')
[[ 0.21633181  0.46361945 -0.0893285 ... -0.20120145 -0.89976685
-0.07772901]
[ -0.06005892  0.37376534 -0.15962054 ...  0.2372093 -0.74607398
-0.76761648]
[ 0.40926478  0.43447018  0.3673702 ... -0.2036962 -0.23671443
-0.28807406]
...
[ 0.40032337  0.93748013 -0.38694532 ... -0.28596616 -0.58826772
-1.24867457]
[ 0.46108146  0.4797264 -0.34775959 ...  0.22005661 -0.64094577
-0.73309616]
[ 0.17155332  0.64167509 -0.51863713 ... -0.41102532  0.15211431
-0.28959053]]
```

Gambar 1.80 hasil test4.py

### 1.14.3 Penanganan Error

#### 1. Screenshoots Error



ModuleNotFoundError: No module named 'joblib'

Gambar 1.81 error

#### 2. Jenis Error

- Not Module name "joblib"

### 3. Cara Penanganan Error

- install library joblib

#### **1.14.4 Bukti tidak plagiat**

#### **1.14.5 Link Youtube**

### **1.15 1174074 — Mochamad Arifqi Ramadhan**

#### **1.15.1 Teori**

##### **1. Definisi Sejarah dan Perkembangan Kecerdasan Buatan**

Kecerdasan Buatan biasa disebut dengan istilah AI (Artificial Intelligence). AI sendiri merupakan suatu cabang dalam bisnis sains komputer sains dimana mengkaji tentang bagaimana cara untuk men lengkap se buah komputer dengan kemampuan atau kepintaran layaknya atau mirip dengan yang dimiliki manusia. Sebagai contoh, sebagaimana komputer dapat berkomunikasi dengan pengguna baik menggunakan kata, suara maupun lain sebagainya. Dengan kemampuan ini, diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemui nya kemudian itulah yang disebut dengan kecerdasan buatan. Kecerdasan buatan adalah kemampuan komputer digital atau robot yang dikendalikan komputer untuk melakukan tugas yang umumnya dikaitkan dengan sesuatu yang cerdas. Istilah ini sering diterapkan pada proyek pengembangan sistem yang diberkahi dengan karakteristik proses intelektual manusia, seperti kemampuan untuk berpikir, menemukan makna, menggeneralisasi, atau belajar dari pengalaman masa lalu.

Kecerdasan Buatan adalah salah satu bidang studi yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara lebih manusiawi dan lebih bisa di pahami oleh manusia. Kecerdasan buatan makin canggih dengan kemampuan komputer dalam memperbarui pengetahuannya dengan banyaknya testing dan perkembangan target analisa. Untuk kecerdasan buatan ada banyak contoh dan jenisnya. Salah satu contoh yang paling terkenal dari Artificial Intelligence ialah Google Assistant. Google Assistant digunakan untuk kemudahan user dalam menemukan berbagai hal maupun penyetelan langsung terhadap smartphone yang digunakan dan masih banyak lagi.

Sejarah Kecerdasan Buatan, Artificial intelligence merupakan inovasi baru di bidang ilmu pengetahuan. Mulai terbentuk sejak adanya komputer modern dan kira-kira terjadi sekitaran tahun 1940 dan 1950. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi

tingkah laku cerdas dalam sistem kecerdasan komputer. Pada awal 50-an, studi tentang “mesin berpikir” memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan informasi. Pada tahun 1956, para ilmuwan jenius seperti Alan Turing, Norbert, Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen dibidang cybernetics, matematika, algoritma dan teori jaringan. Namun, seprang ilmuwan komputer dan kognitif John McCarthy adalah orang yang dating dengan ide untuk bergabung dengan upaya penelitian terpisah ini kedalam satu bidang yang akan mempelajari topic baru untuk imajinasi manusia yaitu kecerdasan buatan. Dia adalah orang yang menciptakan istilah tersebut dan kemudian mendirikan laboratorium Kecerdasan Buatan di MIT dan Stan ford.

Perkembangan Kecerdasan Buatan, Teknologi Artificial Intelligence semakin ramai dibahas dalam berbagai diskusi teknologi di seluruh dunia. Menurut kebanyakan orang, pekerjaan seperti kasir, operator telepon, pengendara truk, dan lainnya sangat berpeluang besar untuk tergantikan oleh Artificial Intelligence. Mengapa terjadi hal demikian? dikarenakan memang bahwa AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Namun, dalam beberapa aspek memang pekerja manusia masih unggul dibandingkan AI itu sendiri. Para generasi muda yang ada di dunia terutama di daerah Asia terlihat sudah memahami fungsi dan efek dari AI dalam kehidupan kita sehari-hari. Berdasarkan survei yang dilakukan oleh Microsoft, terdapat 39 persen responden yang mempertimbangkan untuk menggunakan mobil tanpa pengemudi dan 36 persen lainnya setuju bahwa robot masa depan dengan software untuk beroperasi mampu meningkatkan produktivitas. Dari survey tersebut kita sebagai pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tanpa memberikan efek samping terhadap etos kerja dan keseharian kita sebagai pengguna dalam kehidupan sehari-hari.

2. Definisi Supervised learning, klasifikasi, regresi, unsupervised learning, dataset, training set dan testing set.

- **Supervised Learning**

Supervised Learning merupakan tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan(juga disebut sinyal pengawasan super).

- **Klasifikasi**

Klasifikasi adalah pembagian sesuatu menurut kelas-kelas. Klasifikasi merupakan proses dari pengelompokan benda berdasarkan ciri-ciri

persamaan dan juga perbedaan. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah.

- Regresi

Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel. Regresi adalah membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti "gaji" atau "berat".

- Unsupervised learning

Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Dalam Unsupervised Learning, sistem AI dapat mengelompokkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan.

- Data set

Data set objek yang merepresentasikan data dan relasinya di memory. Strukturnya mirip dengan data di database. Dataset berisi koleksi dari datatable dan datarelation.

- Training Set

Training Set adalah set digunakan oleh algoritma klasifikasi. Dapat dicontohkan dengan decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier

- Testing Set

Testing Set merupakan set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Dapat berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakan-nya sampai akhir.

### 1.15.2 Praktek

- Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

```
(base) C:\Users\kaptira\anaconda\envs\scikit_learn
Collecting scikit-learn
  Using cached https://files.pythonhosted.org/packages/1d/e3/5c3355a0a8f39e4138a97a6a6a51a638/scikit-learn-0.20.3.tar.gz
    Preparing metadata (setup.py) ...
    building [██████████] 100%     build
      added / updated specs:
        scikit-learn==0.20.3

The following packages will be downloaded:
  package              | build
  conda-package-handling | py37_0       200 KB
  conda-4.8.2           | py37_0       3.1 MB
  ...
  total:               3.3 MB
```

**Gambar 1.82** Instalasi Package Scikit Learn

**Gambar 1.83** Isi Variabel Explorer

## 2. Mencoba loading an example dataset

```

1 #%% Mencoba loading an example dataset
2 from sklearn import datasets # Digunakan Untuk Memanggil
   class datasets dari library sklearn
3 iris = datasets.load_iris() # Menggunakan contoh datasets
   iris
4 x = iris.data           # Menyimpan nilai data sets iris
   pada variabel x

```

## 3. Mencoba Learning dan predicting

```

1 y = iris.target          # Menyimpan nilai data label iris
   pada variabel y
2
3 #%%Mencoba Learning dan predicting
4 from sklearn.neighbors import KNeighborsClassifier #Digunakan
   Untuk Memanggil fungsi KNeighborsClassifier
5                                         # pada
   class sklearn dan library sklearn
6 import numpy as np # memanggil library numpy dan dibuat alias
   np
7 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn
   , dan memanggil fungsi KNeighborsClassifier
8                                         #dan mendefinisikan k
   -nya adalah 1
9 knn.fit(x,y)                  #Perhitungan
   matematika library kkn
10 a=np.array([1.0,2.0,3.0,4.0]) #Membuat Array
11 a = a.reshape(1,-1)           #Mengubah Bentuk
   Array jadi 1 dimensi

```

## 4. Mencoba Model Persistence

```

1 hasil = knn.predict(a)          #Memanggil fungsi
   prediksi
2 print(hasil)                   #menampilkan hasil
   prediksi
3
4 #%% Model Persistense
5 from sklearn import svm # Digunakan untuk memanggil class svm
   dari library sklearn

```

```

6 from sklearn import datasets # Digunakan untuk class datasets
    dari library sklearn
7 clf = svm.SVC() # membuat variabel clf , dan
    memanggil class svm dan fungsi SVC
8 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset
    iris dan mengembalikan nilainya .
9 clf.fit(X, y) #Perhitungan nilai label
10
11 from joblib import dump, load #memanggil class dump dan load
    pada library joblib

```

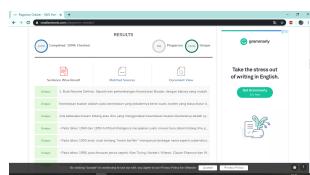
## 5. Mencoba Conventions

```

1 dump(clf, '1174074.joblib') #Menyimpan model kedalam 1174074.
    joblib
2 hasil = load('1174074.joblib') #Memanggil model 1174074
3 print(hasil) # Menampilkan Model yang dipanggil sebelumnya
4
5 #%% Conventions
6 import numpy as np # memanggil library numpy dan dibuat alias
    np
7 from sklearn import random_projection #Memanggil class
    random_projection pada library sklean
8
9 rng = np.random.RandomState(0) #Membuat variabel rng , dan
    mendefinisikan np, fungsi random dan attr RandomState
    kedalam variabel
10 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
    nilai random dari 10 – 2000
11 X = np.array(X, dtype='float32') #menyimpan hasil nilai
    random sebelumnya, kedalam array , dan menentukan
    typedatanya sebagai float32
12 X.dtype # Mengubah data tipe menjadi float64

```

### 1.15.3 Bukti Tidak Plagiat



Gambar 1.84    Bukti Tidak Melakukan Plagiat

## 1.16 Advent Nopele Olansi Damiahan Sihite (1174089)

### 1.16.1 Teori

#### 1. Definisi Kecerdasan buatan

Kecerdasan Buatan biasa disebut dengan istilah AI (Artificial Intelligence). Kecerdasan Buatan adalah salah satu bidang studi yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara lebih manusiawi dan lebih bisa di pahami oleh manusia. Kecerdasan buatan makin canggih dengan kemampuan komputer dalam memperbarui pengetahuannya dengan banyaknya testing dan perkembangan target analisa. Untuk kecerdasan buatan ada banyak contoh dan jenisnya. Salah satu contoh yang paling terkenal dari Artificial Intelligence ialah Google Assistant. Google Assistant digunakan untuk kemudahan user dalam menemukan berbagai hal maupun penyetelan langsung terhadap smartphone yang digunakan dan masih banyak lagi.

#### 2. Sejarah dan Perkembangan Kecerdasan Buatan

- Pada tahun 1943, pekerjaan pertama yang dikenal sebagai AI telah dilakukan oleh Warren McCulloch dan juga Walter Pitts yang dinamakan sebagai artificial neurons
- Pada tahun 1955, Allen Newell dan Herbert A. Simon membuat program kecerdasan buatan pertama yang dinamakan Logic Theorist
- Pada tahun 1972, robot pertama dibuat di jepang dengan nama Wabot-1 dengan kecerdasan buatan
- Pada tahun 1980, muncul bidang baru dari kecerdasan buatan yaitu Expert System yang membantu dalam pemberian keputusan
- Tahun 1997, IBM deep blue mengalahkan juara catur dunia Gary Kasparov dan menjadi komputer pertama yang mengalahkannya
- Tahun 2006, perusahaan sudah mulai menerapkan kecerdasan buatan pada produknya seperti Netflix dan Twitter.
- Tahun 2018, Project Debater dari IBM melakuakn debat tentang topik yang kompleks dan berakhir dengan hasil memuaskan

#### 3. Definisi Supervised Learning

Supervised Learning adalah tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan(juga disebut sinyal pengawasan super).

#### 4. Klasifikasi Supervised Learning

- Klasifikasi adalah pembagian sesuatu menurut kelas-kelas ( class ). Menurut Ilmu Pengetahuan, Klasifikasi merupakan proses pengelompokan benda berdasarkan ciri-ciri persamaan dan juga perbedaan. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah. Label (y) umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran mesin dan statistik, klasifikasi adalah pendekatan pembelajaran yang diawasi di mana program komputer belajar dari input data yang diberikan kepadanya dan kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pengamatan baru. Kumpulan data ini mungkin hanya bersifat dua kelas (seperti mengidentifikasi apakah orang tersebut berjenis kelamin laki-laki atau perempuan atau bahwa surat itu spam atau bukan-spam) atau mungkin juga multi-kelas. Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, identifikasi metrik, klasifikasi dokumen dll.

## 5. Regresi dan Unsupervised Learning

Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel. Regresi adalah membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan makan, yang paling sederhana adalah regresi linier. Ia mencoba untuk menyesuaikan data dengan hyper-plane terbaik yang melewati poin.

Unsupervised Learning berbeda dengan Supervised Learning. Perbedaannya ialah unsupervised learning tidak memiliki data latih, sehingga dari data yang ada kita mengelompokan data tersebut menjadi 2 ataupun 3 bagian dan seterusnya. Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Algoritma Unsupervised Learning dapat melakukan tugas pemrosesan yang lebih kompleks dari pada sistem pembelajaran yang diawasi

## 6. Dataset

Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory. Strukturnya mirip dengan data di database, namun bedanya dataset berisi koleksi dari data table dan data relation. mendapatkan data yang tepat berarti mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda prediksi; yaitu data yang berisi sinyal tentang peristiwa yang Anda pedulikan. Data harus diselaraskan dengan masalah yang Anda coba selesaikan.

## 7. Training Set

Training Set adalah set digunakan oleh algoritma klasifikasi . Dapat di-contohkan dengan : decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier. Menjalankan

pelatihan yang diatur melalui jaringan saraf mengajarkan pada net cara menimbang berbagai fitur, menyesuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan dalam hasil Anda. Koefisien-koefisien tersebut, juga dikenal sebagai parameter, akan terkandung dalam tensor dan bersama-sama mereka disebut model, karena mereka mengkodekan model data yang mereka latih. Mereka adalah takeaways paling penting yang akan Anda dapatkan dari pelatihan jaringan saraf.

## 8. Testing Set

Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Ini berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakan sampai akhir. Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda terhadap pengambilan sampel acak akhir ini. Hasil yang dihasilkannya harus memvalidasi bahwa jaring Anda secara akurat mengenali gambar, atau mengenalinya setidaknya  $[x]$  dari jumlah tersebut. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan, lihat hyperparameter yang Anda gunakan untuk menyetel jaringan, serta kualitas data Anda dan lihat teknik pra-pemrosesan Anda.

### 1.16.2 Instalasi

- Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

```
environment location: C:\Users\yoppele
Added / updated spec(s):
scikit-learn

The following packages will be downloaded:
package          build
scikit-learn      0.23.2           scikit-learn-0.23.2-py37_0
Total:           2.8 MB

The following packages will be UPDATED:
codecs           4.0.2-py37_0 -> 4.0.2-py37_0

Proceed ([y]/n)? y

Unpacking and Extracting Packages
scikit-learn-0.23.2-py37_0
Extracting transaction...[0%]
Extracting transaction...[100%]
Extracting transaction...[0%]
```

**Gambar 1.85** Instalasi Package Scikit Learn

| Name | Type       | Size     | Value                                  |
|------|------------|----------|--|
| a    | float64    | (150, 4) | [5.1 3.5 1.4 0.2]<br>[4.9 3.0 1.4 0.2] |
| b    | int32      | (150,)   | [0 0 0 ... 2 0 2]                      |
| iris | util.Bunch | 6        | Bunch object of sklearn.utils module   |

**Gambar 1.86** Isi Variabel Explorer

- Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```
1 %%Loading an example dataset
```

```

1 from sklearn import datasets # Load library dataset
2 iris = datasets.load_iris()
3 # variable iris diisi dengan contoh data
4 a = iris.data # Menyimpan value data ke variable A
5 b = iris.target # Menyimpan value data ke variable B

```

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan perbaris

```

1 %% Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier
3 #Load library
4 import numpy as np
5 #load library
6
7 knn = KNeighborsClassifier(n_neighbors=1)
8 #mendefinisikan variabel bernama kkn, dan memanggil fungsi
     KNeighborsClassifier
9 # dan memberikan value 1
10 knn.fit(a,b) # perhitungan library knn
11
12 x = np.array([1.0 ,2.0 ,3.0 ,4.0])
13 # membuat array
14 x = x.reshape(1,-1)
15 #Convert array menjadi 1 dimensi
16
17 hasil = knn.predict(x)
18 #Memanggil fungsi predict dari KNN
19 print(hasil)
20 #menampilkan value dari variable hasil

```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %% Model Persistense
2 from sklearn import svm
3 # Load library
4 from sklearn import datasets
5 # Load Library
6 clf = svm.SVC()
7 # mendefinisikan variabel clf, dan memanggil fungsi SVC dari
     class svm
8 a, b = datasets.load_iris(return_X_y=True)
9 #Variable a dan b diisi dengan dataset iris dan mengembalikan
     nilainya.
10 clf.fit(a, b)
11 #memanggil fungsi fit dari clf
12
13 from joblib import dump, load
14 #Load library
15 dump(clf, '1174079.joblib')
16 #Menyimpan model kedalam 1174079.joblib
17 hasil = load('1174079.joblib')

```

```
18 #memuat model 1174079
19 print(hasil) # Menampilkan Hasil
```

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```
1 %% Conventions
2 import numpy as np
3 # Load Library
4 from sklearn import random_projection
5 #Load class random_projection dari library sklearn
6
7 rng = np.random.RandomState(0)
8 #Membuat variabel rng, dan mendefinisikan np, fungsi random dan
9     attr RandomState kedalam variabel
10 X = rng.rand(10, 2000)
11 # membuat variabel X, dan menentukan nilai random dari 10 -
12     2000
13 X = np.array(X, dtype='float32')
14 #menyimpan hasil nilai random sebelumnya, kedalam array, dan
15     menentukan typedatanya sebagai float32
16 X.dtype
17 # Mengubah data tipe menjadi float64
18 transformer = random_projection.GaussianRandomProjection()
19 #membuat variabel transformer, dan mendefinisikan
20     classrandom_projection dan memanggil fungsi
21     GaussianRandomProjection
22 X_new = transformer.fit_transform(X)
23 # membuat variabel baru dan melakukan perhitungan label pada
24     variabel X
25 X_new.dtype
26 # Mengubah data tipe menjadi float64
27 print(X_new)
28 # Menampilkan isi variabel X_new
```

### 1.16.3 Penanganan Error

- ScreenShoot Error

```
In [2]: from sklearn import datasets # Load Library dataset
...: iris = datasets.load_iris()
...: print("Dataset Iris berisi 150 sampel dengan 4 fitur")
...: a = iris.data # Menyimpan value data ke variable A
...: b = iris.target # Menyimpan value data ke variable B
tracesback (most recent call last):
File "<ipython-input-2-55cf92ea2d3>", line 5, in <module>
    b = iris.target # Menyimpan value data ke variable B
NameError: name 'iri' is not defined
```

Gambar 1.87 No Module Named Numpya

- Tuliskan Kode Error dan Jenis Error

- ModuleNotFoundError

### 3. Cara Penangan Error

- ModuleNotFoundError

Mengecek Typo dan menulis kembali library yang akan diimport

#### 1.16.4 Bukti Tidak Plagiat



**Gambar 1.88**    Bukti Tidak Melakukan Plagiat Chapter 1

### 1.17 Handi Hermawan (1174080)

#### 1.17.1 Teori

##### 1. Definisi Kecerdasan buatan

Kecerdasan buatan atau Artificial intelligence merupakan kecerdasan yang ditambahkan kedalam suatu sistem yang diatur secara ilmiah. Kecerdasan buatan dibuat untuk mengantikan pekerjaan yang dilakukan oleh manusia menjadi dikerjakan oleh sistem.

##### 2. Sejarah Kecerdasan Buatan

- Abad 17, Rene Descartes berkata bahwa tubuh hewan adalah sekumpulan mesin yang rumit.
- 1642, Blaise Pascal menciptakan mesin penghitung digital mekanis pertama.
- Abad 19, Charles Babbage dan Ada Lovelace bekerja di program penghitung mekanis.
- 1950, John McCarthy membuat istilah “Kecerdasan Buatan”.
- 1960-1970, Joel Moses membuat program yang pertama kali sukses dalam bidang matematika.
- 1980, jaringan saraf digunakan secara meluas dengan algoritme perambatan balik.
- 2004, DARPA membuat kendaraan yang bisa dijalankan sendiri tanpa manusia.

##### 3. Perkembangan kecerdasan buatan

- Masa persiapan (1943-1946) Warren McCulloch dan Walter Pitt mengemukakan tiga hal : pengetahuan fisiologi dasar dan fungsi sel syaraf

dalam otak, analisa formal tentang logika proposisi, dan teori komputasi Turing.

Pada tahun 1950, Norbert Wiener membuat penelitian mengenai prinsip-prinsip teori feedback.

Pada tahun 1956, John McCarthy meyakinkan Minsky, Claude Shannon dan Nathaniel Rochester untuk membantunya melakukan penelitian dalam bidang Otomata, Jaringan Syaraf dan pembelajaran intelejensi.

- Awal perkembangan (1952-1969) Pada tahun 1958, McCarthy di MIT AI Lab Memo No.1 mendefinisikan bahasa pemrograman tingkat tinggi yaitu LISP,
- Pada tahun 1959, Nathaniel Rochester dari IBM dan mahasiswa-mahasiswanya mengeluarkan program kecerdasan buatan yaitu Geometry Theorem Prover.
- Pada tahun 1963, program yang dibuat James Slagle mampu menyelesaikan masalah integral tertutup untuk mata kuliah Kalkulus. Pada tahun 1986, program analogi buatan Tom Evan menyelesaikan masalah analogi geometris yang ada pada tes IQ.
- Perkembangan Kecerdasan Buatan Melambat (1969-1979) Bruce Buchanan dan Joshua Lederberg yang membuat program untuk memecahkan masalah struktur molekul dari informasi yang didapatkan dari spectrometer massa.
- AI Menjadi sebuah industri Industrialisasi kecerdasan buatan diawali dengan ditemukannya sistem pakar yang dinamakan R1 yang mampu mengkonfigurasi sistem-sistem computer baru.
- Kembalinya Jaringan Syaraf Tiruan (1986-sekarang) Pada tahun 1985-an setidaknya empat kelompok riset menemukan kembali algoritma belajar propagasi balik (Back-Propagation Learning). Algoritma ini berhasil diimplementasikan ke dalam bidang ilmu computer dan psikologi.

#### 4. Definisi Supervised Learning

Supervised Learning merupakan cabang dari Artificial Intelligence. supervised learning adalah suatu ilmu yang mempelajari perancangan dan pengembangan algoritma.

#### 5. Klasifikasi Supervised Learning

- Logistic regression.
- K-nearest neighbors.
- Support vector machine (SVM)
- Naive Bayes.
- Decision tree classification.
- Random forest classification.

## 6. Regresi dan Unsupervised Learning

Regresi merupakan sebuah metode analisis statistic yang digunakan untuk mengetahui pengaruh antara dua variable atau lebih.

Untuk mempelajari Unsupervised learning kita tidak perlu data training untuk melakukan prediksi maupun klasifikasi.

## 7. Dataset

Dataset merupakan objek yang mempresentasikan data dan relasinya pada memori.

## 8. Training Set

Training Set merupakan bagian dari dataset untuk membuat prediksi atau menjalankan fungsi dari sebuah algoritma Machine Learning.

## 9. Testing Set

Testing set digunakan untuk mengukur apakah classifier berhasil melakukan klasifikasi dengan benar.

### 1.17.2 Instalasi

- Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

**Gambar 1.89** Instalasi Package Scikit Learn

| Name  | Type        | Size | Value                                |
|-------|-------------|------|--------------------------------------|
| digit | utils.Bunch | 5    | Bunch object of sklearn.utils module |
| Iris  | utils.Bunch | 6    | Bunch object of sklearn.utils module |

**Gambar 1.90** Isi Variabel Explorer

- Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1
2 #%% Learning dan predicting
3 from sklearn.neighbors import KNeighborsClassifier

```

```

4 #Load library
5 import numpy as np

```

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1
2 knn = KNeighborsClassifier(n_neighbors=1)
3 #mendefinisikan variabel bernama kkn, dan memanggil fungsi
4     KNeighborsClassifier
5 # dan memberikan value 1
6 knn.fit(a,b) # perhitungan library knn
7
8 x = np.array([1.0 ,2.0 ,3.0 ,4.0])
9 # membuat array
10 x = x.reshape(1,-1)
11 #Convert array menjadi 1 dimensi

```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #Memanggil fungsi predict dari KNN
2 print(hasil)
3 #menampilkan value dari variable hasil
4
5 %% Model Persistense
6 from sklearn import svm
7 # Load library
8 from sklearn import datasets
9 # Load Library
10 clf = svm.SVC()

```

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #Variable a dan b diisi dengan dataset iris dan mengembalikan
2     nilainya.
3 clf.fit(a, b)
4 #memanggil fungsi fit dari clf
5
6 from joblib import dump, load
7 #Load library
8 dump(clf, '1174080.joblib')
9 #Menyimpan model kedalam 1174080.joblib
10 hasil = load('1174080.joblib')
11 #memuat model 1174080
12 print(hasil) # Menampilkan Hasil

```

### 1.17.3 Penanganan Error

#### 1. ScreenShoot Error

```
File "D:\Semester 6\Kecerdasan Buatan\KBC-master\src\1174000\1\1174000.py", line 9
    from sklearn.neighbors import *
          ^
SyntaxError: invalid syntax
```

Gambar 1.91 Import Error

#### 2. Tuliskan Kode Error dan Jenis Error

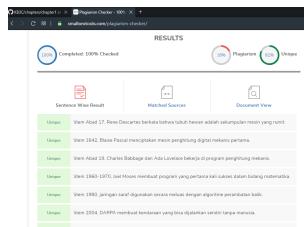
- Import Error

#### 3. Cara Penangan Error

- Import Error

Dengan Menginstall Library Yang Tidak Ditemukan

### 1.17.4 Bukti Tidak Plagiat



Gambar 1.92 Bukti Tidak Melakukan Plagiat

### 1.17.5 Link Youtube

<https://youtu.be/V4NWVnF2bo>

## 1.18 Muhammad Abdul Gani Wijaya (1174071)

### 1.18.1 Teori Soal 1

**1.18.1.1 Definisi Kecerdasan Buatan** Kecerdasan buatan merupakan kecerdasan yang diterapkan pada teknologi, dan diatur serta dikembangkan dalam bidang ilmiah, sebagai bentuk kecerdasan yang dibuat dari kecerdasan ilmu ilmiah yang telah ada.

**1.18.1.2 Sejarah Kecerdasan Buatan** Kecerdasan buatan telah ada pada di zaman kuno dalam cerita dongeng tentang atau benda buatan yang diberkahi dengan kecerdasan atau kesadaran oleh pengrajin dan pembuatnya. Kecerdasan buatan modern menggambarkan proses cara berpikir manusia secara mekanis. Kecerdasan buatan memuncak pada penemuan komputer digital yang sudah dapat diprogram pada tahun 1940-an, sebuah mesin yang didasarkan pada perhitungan dan esensi penalaran matematika. Perangkat ini dan ide-ide nya menginspirasi para ilmuwan untuk mulai serius membahas kemungkinan membangun otak elektronik atau yang sekarang dikenal dengan kecerdasan buatan/artificial intelligence.

#### **1.18.1.3 Perkembangan Kecerdasan Buatan**

##### **1. Awal Mula Kecerdasan Buatan (1943 – 1955)**

- Awal Mula AI dikerjakan oleh McCulloh dan Pitts yang membuat Neuron buatan dengan menirukan cara kerja neuron manusia dengan logika proposisional. Project tersebut bisa menyelesaikan fungsi komputasi dengan struktur neuron network.
- Hebbian learning, memperkenalkan aturan-aturan sederhana untuk meng-update kekuatan antar neuron.
- Minsky dan Edmonds berhasil membangun komputer neural network pertama pada 1950.
- Allan Turing dianggap sebagai orang pertama yang mengeluarkan pikiran mengenai Artificial Intelligence secara utuh pada artikelnya yang berjudul “Computing machinery and Intelligent” pada tahun 1950.

##### **2. Kelahiran Kecerdasan Buatan (1956)**

- McCarthy menginisiasi Dartmouth Workshop pada tahun 1956 dan melahirkan suatu bidang baru yaitu “Artificial Intelligence”.

##### **3. Awal mula AI yang penuh dengan antusias dan harapan besar di masa depan (1952 – 1969)**

- Sebuah tahap pengembangan aplikasi AI yang sukses jika dibandingkan dengan program komputer primitif. Banyak dari aplikasi AI yang berhasil sehingga muncul istilah “evolusi mesin”

##### **4. AI menjadi industry (1980 – sekarang)**

- Aplikasi komersial pertama yang menggunakan sistem pakar bernama R1 yang digunakan oleh perusahaan Amerika (1982).
- Jepang juga membentuk proyek jangka panjang menggunakan komputer cerdas dengan berbasis Prolog.

##### **5. Kecerdasan Buatan menjadi disiplin ilmu (1987 – sekarang)**

##### **6. AI menampakkan diri di semua bidang (1995 – sekarang)**

## 1.18.2 Teori Soal 2

**1.18.2.1 Definisi Supervised Learning** Supervised Learning adalah pembelajaran dengan diiringi oleh supervisornya. Maksud dari supervisornya adalah label pada tiap data nya. Maksud dari label adalah tag oada data yang ditambahkan dalam machine learning model. Pada contoh gambar burung di tag burung” pada setiap masing masing image burung dan gambar ikan di tag “ikan” pada setiap masing masing gambar ikan. Machine learning kategori dapat berupa clasification (“ikan”, “kucing”, “burung”, dsb) dan regression ( berat, tinggi dsb). Supervised learning digunakan untuk memprediksi pola-pola dengan contoh data yang diberikan, jadi pola yang terbentuk adalah hasil pembelajaran dari data lengkap tersebut. Tentunya jika kita mengisi data baru, lalu setelah kita melakukan ETL (Extract Transform Load) maka kita akan mendapat info feature-feature dari sample baru tersebut. Kemudian dari feature-feature tersebut akan di compare dengan pattern clasification dari model yang didapat dari labeled data. Setiap label dicompare sampai selesai, dan yang memiliki percentage lebih banyak diambil sebagai prediksi akhir.

**1.18.2.2 Definisi Klasifikasi** Klasifikasi adalah penggolongan atau pengelompokan. Menurut KBBI klasifikasi adalah penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang ditetapkan. Harrolds Librarians Glossary menjelaskan bahwa klasifikasi adalah pengelompokkan benda secara logis menurut ciri-ciri kesamaannya. Lalu klasifikasi menurut Sulistyo Basuki yang menjelaskan bahwa klasifikasi merupakan proses yang digunakan untuk pengelompokan/pengumpulan benda atau entitas yang sama, serta memisahkan benda atas entitas yang tidak sama. Namun secara umum klasifikasi adalah suatu kegiatan yang mengelompokkan benda-benda yang memiliki beberapa ciri-ciri yang sama dan memisahkan benda yang tidak sama.

**1.18.2.3 Definisi Regresi** Regresi merupakan suatu metode analisis statistik yang digunakan untuk melihat pengaruh antara dua atau lebih banyak variabel. Hubungan variabel-variabel tersebut bersifat fungsional yang diwujudkan dalam suatu model yang matematis. Analisis regresi pada variabel dibagi menjadi dua, yaitu variabel respons (response variable) atau variabel bergantung (dependent variable), dan variabel explanatory atau penduga (predictor variable) atau disebut variabel bebas (independent variable).

**1.18.2.4 Definisi Unsupervised Learning** Unsupervised learning memiliki keunggulan dari unsupervised learning. Jika unsupervised learning memiliki label sebagai dasar prediksi serta membuat clasification dan regression algorithm memungkinkan. Namun pada realitanya, data real itu banyak yang tidak memiliki label. Label data akan masuk ke ERP apapun bentuk ERPnya, sedangkan jika datanya berupa natural input seperti suara, gambar, dan video tidak bisa. Pada unsupervised learning tidak menggunakan label pada saat memprediksi target feauture / variable nya. Namun menggunakan kesamaan

dari attribut yang dimiliki. Jika attribut dan sifat dari data feature yang diekstrak memiliki kemiripan, maka akan dikelompokan (clustering). Sehingga nantinya akan menimbulkan kelompok kelompok (cluster). Jumlah cluster bisa tak terbatas. Dari kelompok kelompok itu model akan melabelkan, dan jika data baru yang mau di prediksi, maka akan dicocokkan dengan kelompok yang mirip featurenya.

**1.18.2.5 Definisi Data Set** Dataset adalah kumpulan dari data. Yang paling umum satu data set sesuai dengan isi tabel pada database tunggal, atau matriks data pada statistik tunggal, di mana setiap kolom tabel mewakili suatu variabel tertentu, dan setiap baris sesuai dengan anggota tertentu dari dataset yang dipertanyakan.

**1.18.2.6 Definisi Training Set** Training Set adalah set digunakan oleh algoritma klasifikasi . Contohnya : decision tree, bayesian, neural network dll. Semuanya biasanya digunakan untuk membentuk model classifier. Menjalankan pelatihan dan diatur melalui jaringan saraf yang mengajarkan pada net cara menimbang berbagai fitur, lalu menyesuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan pada hasil. Koefisien-koefisien tersebut, juga dikenal dengan parameter, akan terkandung dalam tensor dan bersama-sama mereka disebut dengan model, karena mereka mengkodekan model data yang telah mereka latih. Mereka adalah takeaways yang paling penting yang akan didapatkan dari pelatihan jaringan saraf.

**1.18.2.7 Definisi Testing Set** Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Ini berfungsi sebagai meterai persetujuan, dan tidak menggunakan sampai akhir. Setelah melatih dan mengoptimalkan data, dapat menguji jaringan saraf terhadap pengambilan sampel acak akhir ini. Hasil yang dihasilkannya harus memvalidasi bahwa jaring secara akurat mengenali gambar, atau mengekalinya setidaknya  $[x]$  dari jumlah tersebut. Jika tidak mendapatkan prediksi yang akurat, kembali ke set pelatihan, lihat hyperparameter yang digunakan untuk menyetel jaringan, serta kualitas data dan lihat teknik pra-pemrosesan.

### 1.18.3 Instalasi

- Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer
- Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%% Mencoba loading an example dataset
2 from sklearn import datasets
3 # Memanggil class datasets dari library sklearn

```

```

  % Total    Downloaded   Speed     Estimated Left Time
  0%          0B       0B/s      00:00:00
  1%          0B       0B/s      00:00:00
  2%          0B       0B/s      00:00:00
  3%          0B       0B/s      00:00:00
  4%          0B       0B/s      00:00:00
  5%          0B       0B/s      00:00:00
  6%          0B       0B/s      00:00:00
  7%          0B       0B/s      00:00:00
  8%          0B       0B/s      00:00:00
  9%          0B       0B/s      00:00:00
 10%          0B       0B/s      00:00:00
 11%          0B       0B/s      00:00:00
 12%          0B       0B/s      00:00:00
 13%          0B       0B/s      00:00:00
 14%          0B       0B/s      00:00:00
 15%          0B       0B/s      00:00:00
 16%          0B       0B/s      00:00:00
 17%          0B       0B/s      00:00:00
 18%          0B       0B/s      00:00:00
 19%          0B       0B/s      00:00:00
 20%          0B       0B/s      00:00:00
 21%          0B       0B/s      00:00:00
 22%          0B       0B/s      00:00:00
 23%          0B       0B/s      00:00:00
 24%          0B       0B/s      00:00:00
 25%          0B       0B/s      00:00:00
 26%          0B       0B/s      00:00:00
 27%          0B       0B/s      00:00:00
 28%          0B       0B/s      00:00:00
 29%          0B       0B/s      00:00:00
 30%          0B       0B/s      00:00:00
 31%          0B       0B/s      00:00:00
 32%          0B       0B/s      00:00:00
 33%          0B       0B/s      00:00:00
 34%          0B       0B/s      00:00:00
 35%          0B       0B/s      00:00:00
 36%          0B       0B/s      00:00:00
 37%          0B       0B/s      00:00:00
 38%          0B       0B/s      00:00:00
 39%          0B       0B/s      00:00:00
 40%          0B       0B/s      00:00:00
 41%          0B       0B/s      00:00:00
 42%          0B       0B/s      00:00:00
 43%          0B       0B/s      00:00:00
 44%          0B       0B/s      00:00:00
 45%          0B       0B/s      00:00:00
 46%          0B       0B/s      00:00:00
 47%          0B       0B/s      00:00:00
 48%          0B       0B/s      00:00:00
 49%          0B       0B/s      00:00:00
 50%          0B       0B/s      00:00:00
 51%          0B       0B/s      00:00:00
 52%          0B       0B/s      00:00:00
 53%          0B       0B/s      00:00:00
 54%          0B       0B/s      00:00:00
 55%          0B       0B/s      00:00:00
 56%          0B       0B/s      00:00:00
 57%          0B       0B/s      00:00:00
 58%          0B       0B/s      00:00:00
 59%          0B       0B/s      00:00:00
 60%          0B       0B/s      00:00:00
 61%          0B       0B/s      00:00:00
 62%          0B       0B/s      00:00:00
 63%          0B       0B/s      00:00:00
 64%          0B       0B/s      00:00:00
 65%          0B       0B/s      00:00:00
 66%          0B       0B/s      00:00:00
 67%          0B       0B/s      00:00:00
 68%          0B       0B/s      00:00:00
 69%          0B       0B/s      00:00:00
 70%          0B       0B/s      00:00:00
 71%          0B       0B/s      00:00:00
 72%          0B       0B/s      00:00:00
 73%          0B       0B/s      00:00:00
 74%          0B       0B/s      00:00:00
 75%          0B       0B/s      00:00:00
 76%          0B       0B/s      00:00:00
 77%          0B       0B/s      00:00:00
 78%          0B       0B/s      00:00:00
 79%          0B       0B/s      00:00:00
 80%          0B       0B/s      00:00:00
 81%          0B       0B/s      00:00:00
 82%          0B       0B/s      00:00:00
 83%          0B       0B/s      00:00:00
 84%          0B       0B/s      00:00:00
 85%          0B       0B/s      00:00:00
 86%          0B       0B/s      00:00:00
 87%          0B       0B/s      00:00:00
 88%          0B       0B/s      00:00:00
 89%          0B       0B/s      00:00:00
 90%          0B       0B/s      00:00:00
 91%          0B       0B/s      00:00:00
 92%          0B       0B/s      00:00:00
 93%          0B       0B/s      00:00:00
 94%          0B       0B/s      00:00:00
 95%          0B       0B/s      00:00:00
 96%          0B       0B/s      00:00:00
 97%          0B       0B/s      00:00:00
 98%          0B       0B/s      00:00:00
 99%          0B       0B/s      00:00:00
100%          0B       0B/s      00:00:00

```

Gambar 1.93 Instalasi Package Scikit



Gambar 1.94 Isi Variabel Explorer Package Scikit

```

4 iris = datasets.load_iris()
5 # Menggunakan contoh datasets iris
6 x = iris.data
7 # Menyimpan data sets iris pada variabel x
8 y = iris.target
9 # Menyimpan data label iris pada variabel y

```

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan perbaris

```

1 %%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier
3 #Memanggil fungsi KNeighborsClassifier dari library sklearn
4 import numpy as np
5 #Memamnggil library numpy dengan alias np
6 knn=KNeighborsClassifier(n_neighbors=1)
7 #Membuat variabel kkn dan memanggil fungsi
8 #KNeighborsClassifier lalu mendefinisikan k adalah 1
9 knn.fit(x,y)
10 #Perhitungan matematika kkn
11 a=np.array([1.0 ,2.0 ,3.0 ,4.0])
12 #Membuat Array
13 a = a.reshape(1,-1)
14 #Mengubah Bentuk Array jadi 1 dimensi
15 hasil = knn.predict(a)
16 #Memanggil fungsi prediksi
17 print(hasil)
18 #menampilkan hasil prediksi

```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %% Model Persistense
2 from sklearn import svm
3 #Memanggil class svm dari library sklearn

```

```

4 from sklearn import datasets
5 #Memanggil class datasets dari library sklearn
6 clf = svm.SVC()
7 #Membuat variabel dengan nama clf , dan memanggil class svm
    dan fungsi SVC
8 X, y = datasets.load_iris(return_X_y=True)
9 #Mengambil dataset iris dan mengembalikan nilainya .
10 clf.fit(X, y)
11 #Perhitungan nilai label
12
13 from joblib import dump, load
14 #memanggil class dump dan load pada library joblib
15 dump(clf, '1174071.joblib')
16 #Menyimpan model kedalam 1174071.joblib
17 hasil = load('1174071.joblib')
18 #Memanggil model 1174071 dan disimpan pada variable hasil
19 print(hasil)
20 #Menampilkan variable hasil

```

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

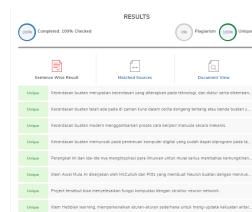
1 %% Conventions
2 import numpy as np
3 #memanggil library numpy dengan alias np
4 from sklearn import random_projection
5 #Memanggil class random_projection dari library sklearn
6
7 rng = np.random.RandomState(0)
8 #Membuat variabel rng, dan mendefinisikan np, memanggil fungsi
    random dan attr RandomState kedalam variabel
9 X = rng.rand(10, 2000)
10 #Membuat variabel X, dan menentukan nilai random dari 10 –
    2000
11 X = np.array(X, dtype='float32')
12 #Menyimpan hasil nilai random sebelumnya , kedalam array , dan
    menentukan typedatanya yaitu float32
13 X.dtype
14 #Mengubah data tipe menjadi float64
15
16 transformer = random_projection.GaussianRandomProjection()
17 #Membuat variabel transformer, dan mendefinisikan
    classrandom_projection dan memanggil fungsi
    GaussianRandomProjection
18 X_new = transformer.fit_transform(X)
19 #Membuat variabel X_new dan melakukan perhitungan label pada
    variabel X
20 X_new.dtype
21 #Mengubah data tipe menjadi float64
22 print(X_new)
23 #Menampilkan isi variabel X_new

```

```

File "c:\users\muham\anaconda3\lib\site-packages\spyder_kernels\customize
\syspathcontext.py", line 110, in _execfile
    exec(compile(f.read(), filename, "exec"), namespace)
File "c:\users\muham\documents\sklearn.py", line 8, in <module>
    from sklearn import datasets
ModuleNotFoundError: No module named 'sklearn'

In [2]:
```

**Gambar 1.95** Import Library Error**Gambar 1.96** Install Library**Gambar 1.97** Bukti Tidak Melakukan Plagiarisme Chapter 1

#### 1.18.4 Penanganan Error

1. Screenshot Error
2. Tuliskan Kode Error dan Jenis Error
  - Import Library Error
3. Cara Penangan Error
  - Impor Library Error  
Menginstall Library yang belum ada

#### 1.18.5 Bukti Tidak Plagiarisme

#### 1.18.6 Link Youtube

<http://bit.ly/AIkem71>

### 1.19 Kaka Kamaludin

#### 1.19.1 Teori

1. Definisi, Sejarah dan Perkembangan Kecerdasan Buatan

- Definisi Kecerdasan Buatan

Kecerdasan Buatan biasa disebut dengan istilah AI (Artificial Intelligence). AI sendiri merupakan suatu cabang dalam bisnis sains komputer sains dimana mengkaji tentang bagaimana cara untuk men lengkapi sebuah komputer dengan kemampuan atau kepintaran layaknya atau mirip dengan yang dimiliki manusia. Sebagai contoh, sebagaimana komputer dapat berkomunikasi dengan pengguna baik menggunakan kata, suara maupun lain sebagainya. Dengan kemampuan ini, diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemuinya kemudian itulah yang disebut dengan kecerdasan buatan.

Kecerdasan Buatan adalah salah satu bidang studi yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara lebih manusiawi dan lebih bisa di pahami oleh manusia. Kecerdasan buatan makin canggih dengan kemampuan komputer dalam memperbarui pengetahuannya dengan testing dan perkembangan target analisa.

- Sejarah Kecerdasan Buatan

Pada tahun 1956, McCarthy yang sama mendirikan Konferensi Dartmouth di Hanover, New Hampshire. Peneliti terkemuka dalam teori kompleksitas, simulasi bahasa, hubungan antara keacakan dan pemikiran kreatif, jaringan saraf diundang. Tujuan dari bidang penelitian yang baru dibuat adalah untuk mengembangkan mesin yang dapat mensimulasikan setiap aspek kecerdasan. Itulah sebabnya Konferensi Dartmouth 1956 dianggap sebagai kelahiran Kecerdasan Buatan. Sejak saat itu, Kecerdasan Buatan telah hidup melalui decade kemuliaan dan cemoohan, yang dikenal luas sebagai musim panas dan musim dingin AI. Musim panasnya ditandai dengan optimism dan dana besar, sedangkan musim dinginnya dihadapkan dengan pemotongan dana, ketidakpercayaan dan pesimisme.

Artificial intelligence merupakan inovasi baru di bidang ilmu pengetahuan. Mulai terbentuk sejak adanya komputer modern dan kira-kira terjadi sekitaran tahun 1940 dan 1950. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Pada awal 50-an, studi tentang “mesin berpikir” memiliki berbagai nama seperti cybernetics, teori automata, dan pemrosesan innformasi. Pada tahun 1956, para ilmuan jenius seperti Alan Turing, Norbert, Wiener, Claude Shannon dan Warren McCullough telah bekerja secara independen di bidang cybernetics, matematika, algoritma dan teori jaringan. Na-

mun, seprang ilmuan komputer dan kognitif John McCarthy adalah orang yang dating dengan ide untuk bergabung dengan upaya penelitian terpisah ini kedalam satu bidang yang akan mempelajari topic baru untuk imajinasi manusia yaitu kecerdasan buatan. Dia adalah orang yang menciptakan istilah tersebut dan kemudian mendirikan laboratorium Kecerdasan Buatan di MIT dan Stan ford.

- Perkembangan Kecerdasan Buatan

Teknologi Artificial Intelligence semakin ramai dibahas dalam berbagai diskusi teknologi di seluruh dunia. Menurut kebanyakan orang, pekerjaan seperti kasir, operator telepon, pengendara truk, dan lainnya sangat berpeluang besar untuk tergantikan oleh Artificial Intelligence. Mengapa terjadi hal demikian? dikarenakan memang bahwa AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Namun, dalam beberapa aspek memang pekerja manusia masih unggul dibandingkan AI itu sendiri. Para generasi muda yang ada di dunia terutama di daerah Asia terlihat sudah memahami fungsi dan efek dari AI dalam kehidupan kita sehari-hari. Berdasarkan survei yang dilakukan oleh Microsoft, terdapat 39 persen responden yang mempertimbangkan untuk menggunakan mobil tanpa pengemudi dan 36 persen lainnya setuju bahwa robot masa depan dengan software untuk beroperasi mampu meningkatkan produktivitas. Dari survey tersebut kita sebagai pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tanpa memberikan efek samping terhadap etos kerja dan keseharian kita sebagai pengguna dalam kehidupan sehari-hari.

AI Summer 1 (1956-1973) KOnferensi Dartmounth diikuti oleh 17 tahun kemajuan luar biasa. Proyek penelitian yang dilakukan di MIT, universitas di Edinburgh, Stanford dan Carnegie Mellon menerima dana besar-besaran, yang akhirnya membuaahkan hasil. Selama tahun-tahun itulah komputer pemrograman mulai melakukan masalah aljabar, membuktikan teorema geometris, memahami dan menggunakan sintaks dan tata bahasa Inggris. Terlepas dari ditinggalkannya koneksiisme dan terjemahan mesin yang gagal, yang menunda penelitian Natural Language Processing (NLP) selama bertahun-tahun, banyak prestasi dari masa lalu yang membuat sejarah. Berikut ini beberapa diantaranya : Pelopor pembelajaran mesin, Ray Solomonoff meletakkan dasar-dasar teori metematika AI, memperkenalkan metode Bayesian universal untuk inferensi dan prediksi induktif Thomas Evans menciptakan program ANALOGI heuristik, yang memungkinkan komputer memecahkan masalah geometrianalogi Unimation, perusahaan robotika pertma didunia, menciptakan robot industri Unimate, yang bekerja pada jalur perakitan modil Genenal Motors. Joseph

Weizenbaum membangun ELIZA-program interaktif yang dapat membawa percakapan dalam bahasan Inggris tentang topik apapun. Ross Quillian menunjukkan jaring semantik, sedangkan Jaime Carbonell (Sr.) mengembangkan Cendikia-program interaktif untuk instruksi yang dibantu komputer berdasarkan jaring semantik. Edward Feigenbaum dan Julian Feldman menerbitkan *Computeks and Thought*, kumpulan artikel pertama tentang AI.

## 2. Definisi Supervised Learning, Klasifikasi, Regresi, Unsupervised Learning, Data Set, Training Set dan Testing Set

### ▪ Definisi Supervised Learning

Supervised Learning adalah tugas pengumpulan data untuk menyimpulkan fungsi dari data pelatihan berlabel. Data pelatihan terdiri dari serangkaian contoh pelatihan. Dalam supervised learning, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan(juga disebut sinyal pengawasan super). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan, yang dapat digunakan untuk memetakan contoh-contoh baru. Skenario optimal akan memungkinkan algoritma menentukan label kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi dari data pelatihan untuk situasi yang tidak terlihat dengan cara yang "masuk akal". Supervised Learning menyediakan algoritma pembelajaran dengan jumlah yang diketahui untuk mendukung penilaian dimasa depan. Chatbots, mobil self-driving, program pengenalan wajah, sistem pakar dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak diawasi. Supervised Learning sebagian besar terkait dengan AI berbasis pengambilan tetapi mereka juga mungkin mampu menggunakan model pembelajaran generatif. Data pelatihan untuk pembelajaran yang diawasi mencakup serangkaian contoh dengan subjek input berpasangan dan output yang diinginkan (yang juga disebut sebagai sinyal pengawasan).

### ▪ Klasifikasi

Klasifikasi adalah pembagian sesuatu menurut kelas-kelas ( class ). Menurut Ilmu Pengetahuan, Klasifikasi merupakan proses pengelompokan benda berdasarkan ciri-ciri persamaan dan juga perbedaan. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai terpisah. Label ( $y$ ) umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran mesin dan statistik, klasifikasi adalah pendekatan pembelajaran yang diawasi di mana program komputer belajar dari input data yang diberikan kepadanya dan

kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pengamatan baru. Kumpulan data ini mungkin hanya bersifat dua kelas (seperti mengidentifikasi apakah orang tersebut berjenis kelamin laki-laki atau perempuan atau bahwa surat itu spam atau bukan-spam) atau mungkin juga multi-kelas. Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, identifikasi metrik, klasifikasi dokumen dll.

- Regresi

Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel. Regresi adalah membahas masalah ketika variabel output adalah nilai riil atau berkelanjutan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan makan, yang paling sederhana adalah regresi linier. Ia mencoba untuk menyesuaikan data dengan hyper-plane terbaik yang melewati poin.

- Unsupervised Learning

Unsupervised Learning adalah pelatihan algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan memungkinkan algoritma untuk bertindak atas informasi tersebut tanpa bimbingan. Dalam Unsupervised Learning, sistem AI dapat mengelompokkan informasi yang tidak disortir berdasarkan persamaan dan perbedaan meskipun tidak ada kategori yang disediakan.

- Data set

Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory. Strukturnya mirip dengan data di database, namun bedanya dataset berisi koleksi dari data table dan data relation. mendapatkan data yang tepat berarti mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda prediksi; yaitu data yang berisi sinyal tentang peristiwa yang Anda pedulikan. Data harus diselaraskan dengan masalah yang Anda coba selesaikan. Gambar kucing tidak terlalu berguna ketika Anda sedang membangun sistem identifikasi wajah. Memverifikasi bahwa data selaras dengan masalah yang ingin Anda selesaikan harus dilakukan oleh ilmuwan data. Jika Anda tidak memiliki data yang tepat, maka upaya Anda untuk membangun solusi AI harus kembali ke tahap pengumpulan data.

- Training Set

Training Set adalah set digunakan oleh algoritma klasifikasi . Dapat dicontohkan dengan : decision tree, bayesian, neural network

dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier. Menjalankan pelatihan yang diatur melalui jaringan saraf mengajarkan pada net cara menimbang berbagai fitur, menyesuaikan koefisien berdasarkan kemungkinan mereka meminimalkan kesalahan dalam hasil Anda. Koefisien-koefisien tersebut, juga dikenal sebagai parameter, akan terkandung dalam tensor dan bersama-sama mereka disebut model, karena mereka mengkodekan model data yang mereka latih. Mereka adalah takeaways paling penting yang akan Anda dapatkan dari pelatihan jaringan saraf.

- Testing set

Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Ini berfungsi sebagai meterai persetujuan, dan Anda tidak menggunakan sampai akhir. Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda terhadap pengambilan sampel acak akhir ini. Hasil yang dihasilkannya harus memvalidasi bahwa jaring Anda secara akurat mengenali gambar, atau mengenalinya setidaknya [x] dari jumlah tersebut. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan, lihat hyperparameter yang Anda gunakan untuk menyetel jaringan, serta kualitas data Anda dan lihat teknik pra-pemrosesan Anda.

### 1.19.2 Praktek

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer

- Buka prompt anaconda lalu ketikkan conda scikit-learn
- Biarkan proses berjalan hingga selesai

```
(base) C:\Users\root>conda install scikit-learn
```

Gambar 1.98 Install library scikit

```
(Base) C:\Users\root>conda install scikit-learn
Collecting package metadata (current_repodata.json): done
Solving environment: done
## Package Plan ##
environment location: C:\Users\root\Anaconda3
added / updated specs:
  scikit-learn

The following packages will be downloaded:
  package                               | build
  conda-4.8.2                           | py37_0    2.8 MB
                                           |
                                           Total:   2.8 MB

The following packages will be UPDATED:
  conda                                4.7.12-py37_0 --> 4.8.2-py37_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
  conda-4.8.2                           | 2.8 MB  | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Gambar 1.99 Proses jika berhasil

2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

- Ketikkan script berikut di spyder, lalu run

```
1 #Import fungsi datasets dari library sklearn
2 from sklearn import datasets
3
4 #Memasukkan data dari datasets iris ke variable iris
5 iris = datasets.load_iris()
6
7 #Memasukkan data dari datasets digits ke variable digits
8 digits = datasets.load_digits()
9
10 #Menampilkan data dari datasets digits ke console
11 print(digits.data)
```

```
In [1]: runfile('E:/MyLatex/src/1174067/src1/test.py', wdir='E:/MyLatex/src/1174067/src1')
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0.  10. ... 12.  1.  0.]]
```

Gambar 1.100 hasil test.py

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan perbaris

```

1 #Import fungsi datasets dari library sklearn
2 from sklearn import datasets
3
4
5 #Memasukkan data dari datasets iris ke variable iris
6 iris = datasets.load_iris()
7
8 #Memasukkan data dari datasets digits ke variable digits
9 digits = datasets.load_digits()
10
11 #Mengimport sebuah Support Vector Machine(SVM) yang merupakan
   algoritma classification yang akan diambil dari Scikit-
   Learn.
12 from sklearn import svm
13
14 #Mendeklarasikan suatu value yang bernama clf yang berisi
   gamma.
15 clf = svm.SVC(gamma=0.001, C=100.)
16
17 #Estimator clf (for classifier)
18 clf.fit(digits.data[:-1], digits.target[:-1])
19
20 #Menunjukkan prediksi angka baru
21 hasil = clf.predict(digits.data[-1:])
22
23 #Menampilkan
24 print(hasil)

```

| Name   | Type        | Size | Value                                |
|--------|-------------|------|--------------------------------------|
| digits | utils.Bunch | 5    | Bunch object of sklearn.utils module |
| hasil  | int32       | (1,) | [8]                                  |
| iris   | utils.Bunch | 6    | Bunch object of sklearn.utils module |

Gambar 1.101 variabel explorer test2.py

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 %%Cara Dump Pertama
2
3
4
5 #Mengimport sebuah Support Vector Machine(SVM) yang merupakan
   algoritma classification yang akan diambil dari Scikit-
   Learn.
6 from sklearn import svm
7 #Import fungsi datasets dari library sklearn
8 from sklearn import datasets
9
10 #Mendefinisikan clf dengan fungsi svc dari library svm
11 clf = svm.SVC()

```

```
12
13 #Mengisi variable x dan y dengan data dari datasets
14 X, y = datasets.load_iris(return_X_y=True)
15
16 #Estimator clf (for classifier)
17 clf.fit(X, y)
18
19 #Mengimport Library pickle
20 import pickle
21
22 #Menyimpan hasil dari clf kedalam sebuah dump
23 s = pickle.dumps(clf)
24
25 #Memanggil dump yang dihasilkan pickle lalu memasukkan hasil
#dumpnya ke variable
26 clf2 = pickle.loads(s)
27
28
29 #Memprediksi angka yang akan muncul
30 clf2.predict(X[0:1])
31
32 #Menampilkan data prediksi
33 print(y[0])
34
35 #%%Cara Dump Kedua
36 # Digunakan untuk memanggil class svm dari library sklearn
37 from sklearn import svm
38
39 # Diguangkan untuk class datasets dari library sklearn
40 from sklearn import datasets
41
42 # membuat variabel clf , dan memanggil class svm dan fungsi
# SVC
43 clf = svm.SVC()
44
45 #Mengambil dataset iris dan mengembalikan nilainya .
46 X, y = datasets.load_iris(return_X_y=True)
47
48 #Perhitungan nilai label
49 clf.fit(X, y)
50
51
52 #memanggil class dump dan load pada library joblib
53 from joblib import dump, load
54
55
56 #Menyimpan model kedalam 1174067.joblib
57 dump(clf, '1174067.joblib')
58
59 #Memanggil model 1174067
60 hasil = load('1174067.joblib')
61 hasil.predict(X[0:1])
62
63 # Menampilkan Model yang dipanggil sebelumnya
```

| Name   | Type        | Size     | Value                                   |
|--------|-------------|----------|---|
| X      | float64     | (150, 4) | [[5.1 3.5 1.4 0.2]<br>[4.9 3. 1.4 0.2]] |
| digits | utils.Bunch | 5        | Bunch object of sklearn.utils module    |
| iris   | utils.Bunch | 6        | Bunch object of sklearn.utils module    |
| y      | int32       | (150,)   | [0 0 0 ... 2 2 2]                       |

Gambar 1.102 variabel explorer test3.py

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 # memanggil library numpy dan dibuat alias np
2 import numpy as np
3
4 #Memanggil class random_projection pada library sklearn
5 from sklearn import random_projection
6
7 #Membuat variabel rng, dan mendefisikan np, fungsi random dan
8 #attr RandomState kedalam variabel
9 rng = np.random.RandomState(0)
10
11 # membuat variabel X, dan menentukan nilai random dari 10 –
12 # 2000
13 X = rng.rand(10, 2000)
14
15 #menyimpan hasil nilai random sebelumnya, kedalam array, dan
16 #menentukan typedatanya sebagai float32
17 X = np.array(X, dtype='float32')
18
19 # Mengubah data tipe menjadi float64
20 X.dtype
21
22 #membuat variabel transformer, dan mendefinisikan
23 #classrandom_projection dan memanggil fungsi
24 # GaussianRandomProjection
25 transformer = random_projection.GaussianRandomProjection()
26 # membuat variabel baru dan melakukan perhitungan label pada
27 # variabel X
28 X_new = transformer.fit_transform(X)
29 # Mengubah data tipe menjadi float64
30 X_new.dtype
31 # Menampilkan isi variabel X_new
32 print(X_new)

```

| Name   | Type        | Size       | Value   |
|--------|-------------|------------|---|
| X      | float32     | (10, 2000) | [[0.5488135 0.71518934 0.60276335 ... 0.4801078 0.64386404 0.5017731 ...  |
| X_new  | float64     | (10, 1973) | [[-0.74506698 0.45457383 0.88209808 ... -0.00255147 -0.19283521<br>-0 ... |
| digits | utils.Bunch | 5          | Bunch object of sklearn.utils module                                      |
| iris   | utils.Bunch | 6          | Bunch object of sklearn.utils module                                      |
| y      | int32       | (150,)     | [0 0 0 ... 2 2 2]   |

Gambar 1.103 variabel explorer test4.py

```
In [5]: runfile('E:/MyLatex/src/1174067/src1/test4.py', wdir='E:/MyLatex/src/1174067/src1')
[[-0.74506698 0.45457383 0.88209808 ... -0.00255147 -0.19283521
-0.0999184 ]
[-0.74474565 -0.07596842 0.75740284 ... 0.92772654 -0.08133772
-0.12007804]
[-0.65903242 0.32393276 1.35294319 ... 0.69425209 -0.01671971
0.49786433]
...
[-0.44103442 0.42938771 0.53121863 ... 0.35585446 0.13729904
0.23401575]
[-0.70748146 -0.29337808 1.12836231 ... 0.81903801 -0.35044182
0.03876353]
[-0.53525128 0.6922521 0.615109 ... 0.75372493 0.10521824
0.2622668 ]]

In [6]: |
```

Gambar 1.104 hasil test4.py

### 1.19.3 Penanganan Error

#### 1. Screenshoots Error



```
ModuleNotFoundError: No module named 'joblib'
```

Gambar 1.105 error

#### 2. Jenis Error

- Not Module name "joblib"

#### 3. Cara Penanganan Error

- install library joblib

### **1.19.4 Bukti tidak plagiat**

### **1.19.5 Link Youtube**

<https://youtu.be/FDQopV2LUIo>

## **1.20 Dini Permata Putri - 1174053**

### **1.20.1 Teori**

### **1.20.2 Pengertian Kecerdasan Buatan**

Kecerdasan buatan adalah salah satu cabang ilmu pengetahuan yang berhubungan dengan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal ini biasanya dilakukan dengan mengikuti karakteristik dan analogi berpikir dari kecerdasan atau inteligance manusia, dan menerapkan sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibilitas dan efisien dapat diambil tergantung keperluan yang mempengaruhi bagaimana wujud dari prilaku kecerdasan buatan. AI biasanya dihubungkan dengan ilmu komputer, akan tetapi juga terkait erat dengan bidang lain seperti matematika, Psikologi, Pengamatan, Biolog, filosofi, dan lainnya

### **1.20.3 Sejarah Kecerdasan Buatan**

Kecerdasan buatan merupakan bidang ilmu komputer yang sangat penting di era kini dan masa yang akan datang untuk mewujudkan sistem komputer yang cerdas. Bidang ini telah berkembang sangat pesat di 20 tahun terakhir seiring dengan kebutuhan perangkat cerdas pada industry dan rumah tangga. Kata Intelligence berasal dari bahasa latin "intelligo" yang berarti "saya paham". Berarti dasar dari intelligence adalah kemampuan untuk memahami dan melakukan aksi. nyatanya, bidang Kecerdasan Buatan atau disingkat dengan AI, berawal dari kemunculan komputer sekitar tahun 1940-an, sedangkan perkembangan sejarah dapat ditelusuri sejak zaman Mesir kuno. Pada saat ini, perhatian mendesak diberikan pada kemampuan komputer untuk melakukan hal-hal yang dapat dilakukan manusia. Dalam hal ini, komputer ini dapat meningkatkan kemampuan kecerdasan dan kecerdasan manusia. Pada awal abad ke-17, René berbicara tentang tubuh binatang yang tidak meminta apa pun selain mesin yang rumit. Blaise Pascal membuat mesin hitung digital mekanis pertama pada tahun 1642. Pada 19, Charles Babbage dan Ada Lovelace bekerja pada mesin hitung mekanis yang dapat diprogram. Bertrand Russell dan Alfred Whitehead North menerbitkan Principia Mathematica, yang merombak logistik formal. Warren McCulloch dan Walter Pitts menerbitkan "Kalkulus Logika Gagasan yang Menjaga Aktivitas" pada tahun 1943 yang membentuk dasar bagi jaringan saraf. 1950-an adalah periode

upaya aktif dalam AI. program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy menciptakan istilah "kecerdasan buatan" pada konferensi pertama yang menjadi dasar perjanjian itu, pada tahun 1956. Dia juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan "tes Turing" sebagai cara untuk mengoperasionalkan tes kecerdasan cerdas.

#### **1.20.4 Perkembangan dan Penggunaan Kecerdasan**

Menurut studi Harvard Business Review dan ICM Unlimited pada tahun 2016, perusahaan besar memberikan kompensasi 10 persen lebih tinggi untuk setiap karyawan, Terrelong melanjutkan pengembangan Artificial Intelligence (AI) tidak hanya untuk membuat gambar atau video palsu lebih mudah, tetapi juga membuatnya sulit untuk membuktikannya. Meskipun pada saat ini, upaya untuk membuat dan mendistribusikan konten hoax, alias hoaks, masih dapat diatasi, tetapi berhasil, tantangan yang dihadapi semakin sulit. Selain itu, AI memungkinkan pembuatan gambar, video, atau audio palsu dari bahan yang relatif minim. Moody's, yang harus disetujui, membuktikan upaya itu akan semakin menantang dan membutuhkan teknik forensik yang lebih canggih. Pada Mei 2019, para peneliti di Samsung AI Center dan Institut Sains dan Teknologi Skolkovo di Moskow, Rusia menunjukkan bahwa mereka dapat membuat tayangan video yang menampilkan masing-masing individu. Video ini sangat realistik tetapi sebenarnya palsu, dibuat menggunakan model pembelajaran tertentu yang disebut Generative Adversarial Network (GAN). Hasil dari proses GAN disebut deepfakes karena mereka menggunakan teknik pembelajaran yang mendalam untuk membuat konten palsu. Untuk jangka pendek, perusahaan diharapkan untuk terus memainkan media sosial dan situs untuk melihat pentingnya disinformasi dan meminta mereka yang bertanggung jawab untuk media sosial dan situs terkait untuk mengunduh konten. Terrelonge menambahkan langkah lain yang bisa diam-bil untuk merilis materi resmi untuk melawan konten palsu."Perlindungan terhadap konten palsu membutuhkan kombinasi teknologi dan pendidikan,".

#### **1.21 resume mengenai definisi supervised learning, klarifikasi, regresi, dan un-supervised learning. Data Set, training set dan testing set**

##### **1.21.1 Supervised Learning**

Supervised Learning adalah tugas mengumpulkan data untuk melengkapi fungsi data pelatihan yang diberi label. Data pelatihan terdiri dari contoh pelatihan. Dalam pembelajaran terawasi, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output dingin (juga disebut sinyal pengawasan super). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang lengkap, yang dapat digu-

nakan untuk memetakan contoh-contoh baru. Skenario akan memungkinkan algoritma menentukan lable kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi data pelatihan sehingga tidak muncul dengan cara yang "masuk akal". Pembelajaran terawasi semakin dekat di mana ada pelatihan praktis selain dapat bervariasi yang berarti tujuannya adalah di mana mengelompokkan data ke dalam database yang ada. Pembelajaran terawasi menyediakan jumlah pembelajaran yang direkomendasikan untuk mendukung penilaian di masa depan. Obrolan, program mengemudi mandiri, pengenalan wajah, tatap muka dan robot adalah beberapa sistem yang dapat menggunakan pembelajaran yang diawasi atau tidak diawasi. Pembelajaran terbimbing sebagian besar terkait dengan AI berdasarkan pengambilan mereka juga mungkin diperlukan menggunakan model pembelajaran generatif. Pelatihan data untuk pembelajaran dimulai dengan mendiskusikan contoh-contoh dengan subjek input berpasangan dan output yang diinginkan (juga disebut sebagai sinyal pengawasan). Dalam pembelajaran yang diawasi untuk pemrosesan gambar, misalnya sistem AI dapat lengkap dengan gambar mengemudi yang berlabel dalam kategori mobil dan truk. Setelah jumlah yang memadai, sistem harus dapat membedakan antara dan mengklasifikasikan gambar yang tidak berlabel, di mana waktu pelatihan dapat diselesaikan secara penuh. Model Pembelajaran Terpandu memiliki beberapa keunggulan dibandingkan pengawasan, tetapi mereka juga memiliki keterbatasan. Sistem lebih cenderung membuat penilaian bahwa hak asasi manusia dapat dihubungkan, misalnya karena manusia telah memberikan dasar untuk pengambilan keputusan. Namun, dalam hal metode berbasis pengambilan, Supervised Learning menghilangkan kesulitan dalam menangani informasi baru. Jika sistem dikategorikan untuk mobil dan truk, maka sepeda disediakan, misalnya, harus dikelompokkan dalam satu kategori atau yang lain. Namun. Jika sistem AI generatif, mungkin tidak tahu apa itu sepeda tetapi akan dapat mengenalinya sebagai milik kategori yang terpisah.

### 1.21.2 Klasifikasi

Klasifikasi adalah pembagian hal sesuai dengan kelas (kelas). Menurut Science, klasifikasi adalah proses pengelompokan materi berdasarkan karakteristik dan perbedaan yang sama. Dalam masalah klasifikasi, kami mencoba memprediksi sejumlah nilai yang terpisah. Label ( $y$ ) Umumnya datang dalam bentuk kategorikal dan mewakili sejumlah kelas. Dalam pembelajaran statistik dan pembelajaran mesin statistik, klasifikasi adalah pembelajaran yang dimulai ketika sebuah program komputer belajar dari input data yang disediakan untuk mendukung dan kemudian menggunakan pembelajaran ini untuk mengklasifikasikan pembelajaran baru. Pengumpulan data ini mungkin hanya dua kelas (seperti mengidentifikasi apakah orang ini laki-laki atau perempuan atau orang itu adalah spam atau bukan-spam) atau mungkin juga multi-kelas.

Beberapa contoh masalah klasifikasi adalah: pengenalan ucapan, pengenalan tulisan tangan, metrik identifikasi, klasifikasi dokumen dll.

### 1.21.3 Regresi

Regresi adalah metode analisis statistik yang digunakan untuk melihat perbedaan antara dua atau lebih variabel. Regresi sedang membahas masalah kompilasi, variabel output adalah nilai nyata atau dipertahankan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan untuk makan, cara paling sederhana adalah linearitas linear. Itu mencoba untuk mencocokkan data dengan pesawat-hyper terbaik yang melewati titik.

### 1.21.4 unsupervised learning

Belajar tanpa pengawasan berbeda dari Belajar dengan Supervisi. Perbedaannya adalah bahwa pembelajaran tanpa pengawasan tidak memiliki data pelatihan, jadi dari data yang tersedia kami mengelompokkan data menjadi 2 atau 3 bagian dan seterusnya. Unsupervised Learning adalah pelatihan dalam algoritma kecerdasan buatan (AI) menggunakan informasi yang tidak diklasifikasikan atau diberi label dan menyediakan algoritma untuk memperbaiki informasi yang diberikan tanpa bimbingan. Dalam Unattended Learning, sistem AI dapat mengklasifikasikan informasi yang tidak diurutkan berdasarkan ekuitas dan perbedaan dalam kategori mendadak yang disediakan. Dalam Supervised Learning Learning, sistem AI disajikan dengan sistem wajib yang tidak diberi label, tidak dikategorikan dan algoritma bekerja pada data tanpa pelatihan sebelumnya. Outputnya tergantung pada algoritma kode. Menyerahkan sistem untuk Belajar Tanpa Pengawasan adalah salah satu cara untuk menerima AI. Algoritma Pembelajaran tanpa pengawasan dapat melakukan tugas yang lebih kompleks daripada sistem pembelajaran yang diawasi. Namun, pembelajaran tanpa pengawasan dapat lebih tidak konsisten dengan model alternatif. Sementara Supervised Learning Mungkin, misalnya, mencari sendiri dengan memilih kucing dari anjing, ia juga dapat menambahkan kategori yang tidak diinginkan dari yang tidak diinginkan untuk ditingkatkan menjadi ras yang tidak biasa, membuat pesanan diperlukan.

### 1.21.5 Data Set

Dataset adalah objek yang mewakili data dan hubungan dalam memori. Strukturnya mirip dengan basis data basis data, tetapi hanya kumpulan data yang dikumpulkan dari catatan dan latar belakang yang diaktifkan. dapatkan persetujuan yang tepat untuk mengumpulkan atau mengidentifikasi data yang berkorelasi dengan hasil yang ingin Anda hasilkan; yaitu data yang berisi sinyal tentang acara yang Anda sukai. Data harus disinkronkan dengan masalah yang Anda coba selesaikan. Gambar kucing bukan kompilasi yang sangat berguna. Anda sedang membangun sistem identifikasi wajah.

Memodifikasi data yang selaras dengan masalah yang ingin Anda selesaikan harus dilakukan oleh para ahli data. Jika Anda tidak memiliki data yang benar, maka upaya Anda untuk membuat solusi AI harus kembali ke instalasi data. Format ujung kanan untuk belajar secara umum adalah array tensor, atau multi-dimensional. Jadi pipa data yang dibangun untuk pembelajaran dibangun secara umum untuk mengubah semua gambar, video, suara, suara, teks atau deret waktu menjadi vektor dan tensor yang dapat digunakan operasi aljabar linier. Data yang diperlukan perlu dinormalisasi, distandarisasi dan dikembalikan untuk meningkatkan kegunaannya, dan semua ini adalah langkah-langkah dalam pembelajaran mesin ETC. Deeplearning4j menawarkan alat ETV Data Vec untuk melakukan tugas memfasilitasi data. Pembelajaran yang mendalam, dan pembelajaran mesin yang lebih umum, membutuhkan pelatihan yang baik agar dapat bekerja dengan baik. Mengumpulkan dan membangun satu set badan pelatihan yang cukup besar dari data yang diketahui membutuhkan waktu dan pengetahuan khusus tentang pengetahuan dan cara untuk mengumpulkan informasi yang relevan. Perangkat pelatihan bertindak sebagai patokan terhadap mana jaring pembelajaran dalam pengeboran. Itulah yang mereka perbarui untuk direkonstruksi sebelum mereka merilis data yang belum pernah dilihat sebelumnya. Pada saat ini, manusia memiliki pengetahuan luas tentang mengidentifikasi instrumen yang tepat dan mengubahnya menjadi representasi numerik yang dapat dipahami oleh algoritma pembelajaran dalam, tensor. Membangun set pelatihan, dalam arti tertentu, pra-pelatihan. Kumpulan pelatihan yang membutuhkan banyak waktu atau keahlian yang dapat membantu dalam dunia data dan pemecahan masalah. Sifat keahlian terbesar Anda dalam memberi tahu algoritma Anda apa yang penting bagi Anda adalah memilih apa yang Anda masukkan dalam kursus pelatihan Anda. Ini melibatkan menceritakan kisah melalui data awal yang Anda pilih untuk memandu proses pembelajaran mendalam Anda dengan mengekstraksi fitur-fitur penting, baik dalam pengaturan pelatihan dan data yang ingin Anda buat untuk dipelajari. Agar pelatihan ini bermanfaat, Anda harus memecahkan masalah yang Anda selesaikan; yaitu, apa yang Anda inginkan agar sesuai dengan pembelajaran Anda, di mana hasil yang ingin Anda prediksi.

### **1.21.6 Training Set**

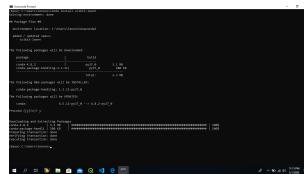
Set Pelatihan adalah set yang digunakan oleh algoritma klasifikasi. Dapat dicontohkan oleh: decisiontree, bayesian, neural network dll. Semuanya dapat digunakan untuk membuat model kelas. Terkait dengan pelatihan yang mengatur melalui jaringan saraf di internet bagaimana menimbang berbagai fitur, sesuaikan koefisien sesuai dengan apa yang mereka tingkatkan dalam hasil Anda. Koefisien, juga dikenal sebagai parameter, akan terkandung dalam sensor dan bersama-sama mereka disebut model, model data karena mereka menyandikan latihan yang mereka praktekkan.

### 1.21.7 testing Set

Tes ini digunakan untuk mengukur sejauh mana classifil berhasil mengklasifikasi dengan benar. Ini digunakan sebagai meterai persetujuan, dan Anda tidak dapat digunakan sampai akhir. Setelah Anda melatih dan mengoptimalkan data Anda, Anda menguji jaringan saraf Anda untuk mengambil sampel acak akhir ini. Hasilnya harus memvalidasi gambar bersih Anda, atau gambar mengenali [x] dari nomor itu. Jika Anda tidak mendapatkan prediksi yang akurat, kembalilah ke set pelatihan Anda, lihat mitra Anda yang Anda gunakan untuk mengelola jaringan Anda, dan kualitas data Anda dan lihat teknik pra-pemanfaatan yang dapat Anda gunakan.

### 1.21.8 Instalasi

- Instalasi Library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer



**Gambar 1.106** Instalasi Package Scikit Learn

```
In [45]: makassar['pass1'] = makassar.apply(lambda row: 1 if (row['G1']>row['G2']) else 0, axis=1)
In [46]: print(makassar.info())
Out[46]:

RangeIndex: 35 entries, 0 to 34
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype  
 0   school  35 non-null    object 
 1   sex     35 non-null    object 
 2   address 35 non-null    object 
 3   famsize 35 non-null    object 
 4   Pstatus  35 non-null    object 
 5   Mjob    35 non-null    object 
 6   Fjob    35 non-null    object 
 7   reason  35 non-null    object 
 8   guardian 35 non-null    object 
 9   traveltime 35 non-null    int64  
 10  studytime 35 non-null    int64  
 11  higher  35 non-null    object 
 12  G1      35 non-null    float64
 13  G2      35 non-null    float64
 14  G3      35 non-null    float64
 15  classes 35 non-null    float64
 16  health  35 non-null    float64
 17  absences 35 non-null    int64  
 18  pass1   35 non-null    int64  
dtypes: float64(6), int64(3), object(6)
memory usage: 2.9 KB
In [47]: makassar.head()
Out[47]:
   school sex address famsize Pstatus Mjob Fjob reason guardian traveltime studytime higher
0  U       G1  Vila       I       D   TSO  TSO  Home  WorkingMother  12        450  0
1  G1     F       U       I       D   TSO  TSO  Home  WorkingMother  17        450  0
2  G1     F       U       I       D   TSO  TSO  Home  WorkingMother  15        450  0
3  G1     F       U       I       D   TSO  TSO  Home  WorkingMother  16        450  0
4  G1     F       U       I       D   TSO  TSO  Home  WorkingMother  16        450  0
   G1      G2      G3  classes  health  absences  pass1
0  0.00  0.00  0.00  0.0000  0.0000  0.0000  0
1  0.00  0.00  0.00  0.0000  0.0000  0.0000  0
2  0.00  0.00  0.00  0.0000  0.0000  0.0000  0
3  0.00  0.00  0.00  0.0000  0.0000  0.0000  0
4  0.00  0.00  0.00  0.0000  0.0000  0.0000  0

```

**Gambar 1.107** Isi Variabel Explorer

- Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```
1 from sklearn import datasets # Digunakan Untuk Memanggil
2   class datasets dari library sklearn
3 iris = datasets.load_iris() # Menggunakan contoh datasets
4   iris
5 x = iris.data # Menyimpan nilai data sets iris
6   pada variabel x
7 y = iris.target # Menyimpan nilai data label iris
8   pada variabel y
```

3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%%Mencoba Learning dan predicting
2 from sklearn.neighbors import KNeighborsClassifier #Digunakan Untuk Memanggil fungsi KNeighborsClassifier
3                                     # pada
4     class sklearn dan library sklearn
5 import numpy as np # memanggil library numpy dan dibuat alias np
6 knn=KNeighborsClassifier(n_neighbors=1) #membuat variabel kkn , dan memanggil fungsi KNeighborsClassifier
7                                     #dan mendefinisikan k -nya adalah 1
8 knn.fit(x,y)                         #Perhitungan matematika library kkn
9 a=np.array([1.0,2.0,3.0,4.0])
10 a = a.reshape(1,-1)                  #Membuat Array #Mengubah Bentuk
11     Array jadi 1 dimensi
12 hasil = knn.predict(a)               #Memanggil fungsi prediksi

```

4. Mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%% Model Persistense
2 from sklearn import svm # Digunakan untuk memanggil class svm dari library sklearn
3 from sklearn import datasets # Digunakan untuk class datasets dari library sklearn
4 clf = svm.SVC()           # membuat variabel clf , dan memanggil class svm dan fungsi SVC
5 X, y = datasets.load_iris(return_X_y=True) #Mengambil dataset iris dan mengembalikan nilainya.
6 clf.fit(X, y)             #Perhitungan nilai label
7
8 from joblib import dump, load #memanggil class dump dan load pada library joblib
9 dump(clf, '1174053.joblib') #Menyimpan model kedalam 1174069. joblib
10 hasil = load('1174053.joblib') #Memanggil model 1174069

```

5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

```

1 #%% Conventions
2 import numpy as np # memanggil library numpy dan dibuat alias np
3 from sklearn import random_projection #Memanggil class random_projection pada library sklearn
4

```

```

5 rng = np.random.RandomState(0) #Membuat variabel rng , dan
     mendefisikan np, fungsi random dan attr RandomState
     kedalam variabel
6 X = rng.rand(10, 2000) # membuat variabel X, dan menentukan
     nilai random dari 10 – 2000
7 X = np.array(X, dtype='float32') #menyimpan hasil nilai
     random sebelumnya, kedalam array, dan menentukan
     typedatanya sebagai float32
8 X.dtype # Mengubah data tipe menjadi float64
9
10 transformer = random_projection.GaussianRandomProjection() #
      membuat variabel transformer, dan mendefinisikan
      classrandom_projection dan memanggil fungsi
      GaussianRandomProjection
11 X_new = transformer.fit_transform(X) # membuat variabel baru
      dan melakukan perhitungan label pada variabel X
12 X_new.dtype # Mengubah data tipe menjadi float64

```

### 1.21.9 Penanganan Error

1. ScreenShoot Error

```

file "D:\Kuliah\Semester 6\Keerdesan Susten\Bahan\livelklearn.py", line 8, in <module>
    from sklearn import datasets
ImportError: cannot import name 'datasets' from 'sklearn' (D:\Kuliah\Semester 6\Keerdesan Susten\Bahan\livelklearn.py)

```

**Gambar 1.108 Import Error**

2. Tuliskan Kode Error dan Jenis Error

- Import Error

3. Cara Penangan Error

- Import Error

Dengan Menginstall Library Yang Tidak Ditemukan

### 1.21.10 Bukti Tidak Plagiat



**Gambar 1.109 Bukti Tidak Melakukan Plagiat Chapter 1**

### 1.21.11 Link Youtube

Buka Link

## BAB 2

---

# CHAPTER 4

---

### 2.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
1 @inproceedings{awangga2017colenak,
2   title={Colenak: GPS tracking model for post-stroke
3   rehabilitation program using AES-CBC URL encryption and QR-
4   Code},
5   author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
6   Hasanudin, Trisna Irmayadi},
7   booktitle={Information Technology, Information Systems and
8   Electrical Engineering (ICITISEE), 2017 2nd International
   conferences on},
9   pages={255--260},
10  year={2017},
11  organization={IEEE}
12 }
```



**Gambar 2.1** Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

#### **2.1.1 Teori**

#### **2.1.2 Praktek**

#### **2.1.3 Penanganan Error**

#### **2.1.4 Bukti Tidak Plagiat**



**Gambar 2.2** Kecerdasan Buatan.

## **2.2 1174069 - Fanny Shafira Damayanti**

#### **2.2.1 Teori**

1. Jelaskan apa itu klasifikasi teks, sertakan gambar ilustrasi buatan sendiri.

Klasifikasi teks merupakan sebuah model yang biasa digunakan untuk untuk mengkategorikan sebuah teks ke dalam kelompok-kelompok yang lebih terorganisir. Jadi untuk setiap kalimat yang di masukan ke dalam mesin, mesin tersebut akan menjadikan setiap kata dari kalimat tersebut menjadi sebuah kolom. Untuk ilustrasinya bisa dilihat pada gambar berikut :



**Gambar 2.3** Klasifikasi teks.

2. Jelaskan mengapa hal ini bisa terjadi, klasifikasi bunga tidak bisa digunakan untuk machine learning, sertakan ilustrasi gambar sendiri.

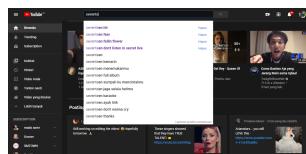
Karena machine learning tidak dapat menampilkan inputan sesuai dengan apa yang kita inputkan. Karena inputan tersebut serupa namun mesin memberikan output yang berbeda, biasanya output atau error ini disebut dengan istilah noise. Untuk contoh sederhananya misalkan kita inputkan salah satu label yang terdapat pada bunga, output yang dihasilkan oleh mesin tersebut ialah label yang lain. Itu dikarenakan bunga banyak jenis yang serupa namun tidak sama. Untuk ilustrasinya bisa dilihat pada gambar berikut :



**Gambar 2.4** Klasifikasi Bunga.

3. Jelaskan bagaimana yang dimaksud dengan teknik pembelajaran mesin pada teks yang digunakan dan sertakan ilustrasi buatan sendiri.

Teknik yang digunakan pada youtube salah satunya ialah keywords. Dengan keywords tersebut mesin dapat memberikan video sesuai dengan keyword yang kita inputkan pada kolom pencarian. Teknik pembelajarannya tergantung user memberikan input teks seperti apa, karena pada youtube itu sendiri akan menyesuaikan dengan apa yang biasa kita inputkan dan akan memfilter video secara otomatis sesuai dengan keyword yang biasa kita inputkan. Contoh ilustrasi sederhananya seperti berikut :



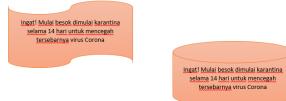
**Gambar 2.5** Klasifikasi teks Youtube.

4. Jelaskan apa yang dimaksud vektorisasi data.

Vektorisasi data ialah suatu pemecahan atau pembagian data berupa teks, sebagai contoh terdapat 5 paragraf, data teks tersebut di pecah menjadi kalimat-kalimat yang lebih sederhana, lalu di pecah lagi menjadi kata untuk setiap kalimatnya.

5. Jelaskan apa yang dimaksud dengan bag of words dengan ilustrasi sendiri.

Representasi penyederhanaan sebuah kalimat atau perhitungan setiap kata pada suatu kalimat dengan presentase berapa kali muncul kata tersebut untuk setiap kalimatnya. Contoh ilustrasi sederhananya seperti berikut :



**Gambar 2.6** Bag of words.

6. Jelaskan apa yang dimaksud dengan TF-IDF.

TF-IDF merupakan metode untuk menghitung bobot setiap kata pada suatu kalimat yang paling sering digunakan. TF-IDF ini akan menghitung nilai Term Frequency dan Inverse Document Frequency pada setiap kata dalam setiap kalimat yang muncul dengan diimbangi dengan jumlah dokumen dalam korpus yang mengandung kata. Contoh ilustrasi sederhananya seperti gambar berikut :

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

|           | Doc 1 | Doc 2 | ... | Doc n |
|-----------|-------|-------|-----|-------|
| Term(s) 1 | 12    | 2     | ... | 1     |
| Term(s) 2 | 0     | 1     | ... | 0     |
| ...       | ...   | ...   | ... | ...   |
| Term(s) n | 0     | 6     | ... | 3     |

**Gambar 2.7** TF-IDF.

## 2.2.2 Praktek Program

1. Soal 1

```

1 #%% Soal 1
2 import pandas as pd #digunakan untuk mengimport library
# pandas dengan alias pd
3 pd = pd.read_csv("F://Semester 6/Artificial Intelligence/
# Tugas 4/src/csv_fanny.csv") #membaca file csv

```

Kode di atas digunakan untuk buat aplikasi sederhana menggunakan pandas dengan format csv sebanyak 500 baris, hasilnya ialah sebagai berikut :

**Gambar 2.8** Hasil Soal 1.

## 2. Soal 2

```

1 #%% Soal 2
2 d_train=pd[:450] #membagi data training menjadi 450
3 d_test=pd[450:] #membagi data menjadi 50 atau sisa dari data
      yang tersedia

```

Kode di atas digunakan untuk memecah dataframe tersebut menjadi dua bagian yaitu 450 row pertama dan 50 row kedua. Hasilnya adalah sebagai berikut :

**Gambar 2.9** Hasil Soal 2.

## 3. Soal 3

```

1 #%% Soal 3
2 import pandas as fanny #untuk import library pandas berguna
      untuk mengelola dataframe
3 fanny = fanny.read_csv("F://Semester 6/Artificial
      Intelligence/Tugas 4/src/Youtube03-LMFAO.csv") #membaca
      file dengan format csv
4
5 spam=fanny.query('CLASS == 1') #membagi tabel spam
6 nospam=fanny.query('CLASS == 0')#membagi tabel no spam
7
8 from sklearn.feature_extraction.text import CountVectorizer #
      untuk import countvectorizer berfungsi untuk memecah data
      tersebut menjadi sebuah kata yang lebih sederhana
9 vectorizer = CountVectorizer () #ntuk menjalankan fungsi
      tersebut , pada code ini tidak ada hasilnya dikarenakan
      spyder tidak mendukung hasil dari instasiasi.
10
11 dvec = vectorizer.fit_transform(fanny['CONTENT']) #untuk
      melakukan pemecahan data pada dataframe yang terdapat pada
      kolom konten

```



## 5. Soal 5

```

1 #%%soal 5
2
3 from sklearn import tree
4 clftree = tree.DecisionTreeClassifier()
5 clftree.fit(d_train_att, d_train_label)

```

Plotlah confusion matrix dari praktik modul ini menggunakan matplotlib.  
Hasilnya adalah sebagai berikut :

```

In [8]: from sklearn import tree
...: clftree = tree.DecisionTreeClassifier()
Out[8]: DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
criterion='gini',
max_depth=None, max_features=None,
max_leaf_nodes=None,
min_impurity_decrease=0.0,
min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0,
presort='deprecated',
random_state=None, splitter='best')

```

**Gambar 2.12** Hasil Soal 5.

## 6. Soal 6

```

1 #%%soal 6/
2
3 from sklearn.metrics import confusion_matrix
4 pred_labels=clftree.predict(d_test)
5 cm=confusion_matrix(d_test_label, pred_labels)
6
7 #%%
8
9 import matplotlib.pyplot as plt
10
11 def plot_confusion_matrix(cm, classes,
12                           normalize=False,
13                           title='Confusion matrix',
14                           cmap=plt.cm.Blues):
15
16     if normalize:
17         cm = cm.astype('float') / cm.sum(axis=1)[:, np.
18         newaxis]
19         print("Normalized confusion matrix")
20     else:
21         print('Confusion matrix, without normalization')
22
23     print(cm)

```

Menjalankan program cross validation. Hasilnya adalah sebagai berikut :

```
In [11]: import matplotlib.pyplot as plt
...
...: def plot_confusion_matrix(cm, classes,
...:                         normalize=False,
...:                         title='Confusion matrix',
...:                         cmap=plt.cm.Blues):
...:     """
...:     If normalize is True, cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
...:     """
...:     if normalize:
...:         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
...:         print("Normalized confusion matrix")
...:     else:
...:         print('Confusion matrix, without normalization')
...:     print(cm)
...:     cm
```

**Gambar 2.13** Hasil Soal 6.**7. Soal 7**

```
1 %%soal 7
2
3 from sklearn.model_selection import cross_val_score
4
5 scores=cross_val_score(clf, d_train_att, d_train_label, cv=5)
6
7 skor_rata2=scores.mean()
8 skoresd=scores.std()
```

Tree.export graphviz merupakan fungsi yang menghasilkan representasi Graphviz dari decision tree, yang kemudian ditulis ke outfile. Disini akan menyimpan classifiernya, akan meng ekspor file student performance jika salah akan mengembalikan nilai fail. Hasilnya adalah sebagai berikut :

```
In [12]: from sklearn.model_selection import cross_val_score
...
...: scores=cross_val_score(clf, d_train_att, d_train_label, cv=5)
...: skor_rata2=scores.mean()
...: skoresd=scores.std()
```

**Gambar 2.14** Hasil Soal 7.**8. Soal 8**

```
1 %%soal 8 /
2
3 max_features_opts = range(5, 50, 5) #max_features_opts
4 n_estimators_opts = range(10, 200, 20) #n_estimators_opts
5 rf_params = fanny.empp((len(max_features_opts)*len(
6     n_estimators_opts),4), float) #rf_params sebagai variabel
7     untuk menjumlahkan yang sudah di tentukan sebelumnya
8 i = 0
9 for max_features in max_features_opts: #pengulangan
10    for n_estimators in n_estimators_opts: #pengulangan
11        clf = RandomForestClassifier(max_features=
12            max_features, n_estimators=n_estimators) #menampilkan
13            variabel csf
14            scores = cross_val_score(clf, df_train_att,
15            df_train_label, cv=5) #scores sebagai variabel training
```

```

11     rf_params[i,0] = max_features #index 0
12     rf_params[i,1] = n_estimators #index 1
13     rf_params[i,2] = scores.mean() #index 2
14     rf_params[i,3] = scores.std() * 2 #index 3
15     i += 1 #dengan ketentuan i += 1
16     print("Max features: %d, num estimators: %d, accuracy
17 : %0.2f (+/- %0.2f)" %(max_features, n_estimators, scores.
mean(), scores.std() * 2))
          #print hasil pengulangan yang sudah ditentukan

```

Buatlah program pengamatan komponen informasi. Jadi disini kita akan memprediksi nilai dari variabel test att dan test pass Hasilnya adalah sebagai berikut :

Gambar 2.15 Hasil Soal 8.

### 2.2.3 Penanganan Error

#### 1. ScreenShoot Error

Gambar 2.16 SyntaxError

#### 2. Cara Penangan Error

- SyntaxError

Dengan menyesuaikan letak file csv pada codingannya, sehingga dapat di baca atau dipanggil file csvnya.

## 2.2.4 Bukti Tidak Plagiat



**Gambar 2.17** Bukti Tidak Melakukan Plagiat Chapter 4

## 2.2.5 Link Youtube

<https://youtu.be/X-xd9Nb78Gs>

## 2.3 1174070 - Arrizal Furqona Gifary

### 2.3.1 Teori

1. Jelaskan apa itu klasifikasi teks, sertakan gambar ilustrasi buatan sendiri.

Klasifikasi teks merupakan sebuah model yang biasa digunakan untuk untuk mengkategorikan sebuah teks ke dalam kelompok-kelompok yang lebih terorganisir. Jadi untuk setiap kalimat yang di masukan ke dalam mesin, mesin tersebut akan menjadikan setiap kata dari kalimat tersebut menjadi sebuah kolom. Untuk ilustrasinya bisa dilihat pada gambar berikut :



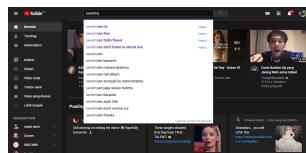
**Gambar 2.18** Klasifikasi teks.

2. Jelaskan mengapa hal ini bisa terjadi, klasifikasi bunga tidak bisa digunakan untuk machine learning, sertakan ilustrasi gambar sendiri. Karena machine learning tidak dapat menampilkan inputan sesuai dengan apa yang kita inputkan. Karena inputan tersebut serupa namun mesin memberikan output yang berbeda, biasanya output atau error ini disebut dengan istilah noise. Untuk contoh sederhananya misalkan kita inputkan salah satu label yang terdapat pada bunga, output yang dihasilkan oleh mesin tersebut ialah label yang lain. Itu dikarenakan bunga banyak jenis yang serupa namun tidak sama. Untuk ilustrasinya bisa dilihat pada gambar berikut :



**Gambar 2.19** Klasifikasi Bunga.

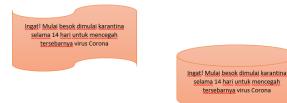
3. Jelaskan bagaimana yang dimaksud dengan teknik pembelajaran mesin pada teks yang digunakan dan sertakan ilustrasi buatan sendiri.  
Teknik yang digunakan pada youtube salah satunya ialah keywords. Dengan keywords tersebut mesin dapat memberikan video sesuai dengan keyword yang kita inputkan pada kolom pencarian. Teknik pembelajarannya tergantung user memberikan input teks seperti apa, karena pada youtube itu sendiri akan menyesuaikan dengan apa yang biasa kita inputkan dan akan memfilter video secara otomatis sesuai dengan keyword yang biasa kita inputkan. Contoh ilustrasi sederhananya seperti berikut :



**Gambar 2.20** Klasifikasi teks Youtube.

4. Jelaskan apa yang dimaksud vektorisasi data.  
Vektorisasi data ialah suatu pemecahan atau pembagian data berupa teks, sebagai contoh terdapat 5 paragraf, data teks tersebut di pecah menjadi kalimat-kalimat yang lebih sederhana, lalu di pecah lagi menjadi kata untuk setiap kalimatnya.
5. Jelaskan apa yang dimaksud dengan bag of words dengan ilustrasi sendiri.

Representasi penyederhanaan sebuah kalimat atau perhitungan setiap kata pada suatu kalimat dengan presentase berapa kali muncul kata tersebut untuk setiap kalimatnya. Contoh ilustrasi sederhananya seperti berikut :



**Gambar 2.21** Bag of words.

6. Jelaskan apa yang dimaksud dengan TF-IDF.  
TF-IDF merupakan metode untuk menghitung bobot setiap kata pada

suatu kalimat yang paling sering digunakan. TF-IDF ini akan menghitung nilai Term Frequency dan Inverse Document Frequency pada setiap kata dalam setiap kalimat yang muncul dengan diimbangi dengan jumlah dokumen dalam korpus yang mengandung kata. Contoh ilustrasi sederhananya seperti gambar berikut :

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

|           | Doc 1 | Doc 2 | ... | Doc n |
|-----------|-------|-------|-----|-------|
| Term(s) 1 | 12    | 2     | ... | 1     |
| Term(s) 2 | 0     | 1     | ... | 0     |
| ...       | ...   | ...   | ... | ...   |
| Term(s) n | 0     | 6     | ... | 3     |

Gambar 2.22 TF-IDF.

### 2.3.2 Praktek Program

#### 1. Soal 1

```

1 %% Soal 1
2 import pandas as pd #digunakan untuk mengimport library
# pandas dengan alias pd
3 pd = pd.read_csv("F:/Semester 6/Artificial Intelligence/
#Tugas 4/src/csv_izal.csv") #membaca file csv

```

Kode di atas digunakan untuk buat aplikasi sederhana menggunakan pandas dengan format csv sebanyak 500 baris, hasilnya ialah sebagai berikut :



Gambar 2.23 Hasil Soal 1.

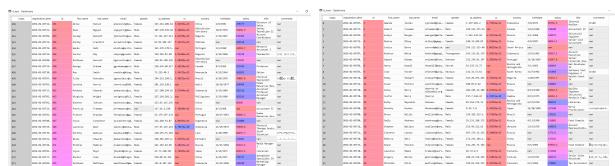
#### 2. Soal 2

```

1 %% Soal 2
2 d_train=pd[:450] #membagi data training menjadi 450
3 d_test=pd[450:] #membagi data menjadi 50 atau sisa dari data
# yang tersedia

```

Kode di atas digunakan untuk memecah dataframe tersebut menjadi dua bagian yaitu 450 row pertama dan 50 row kedua. Hasilnya adalah sebagai berikut :



Gambar 2.24 Hasil Soal 2.

## 3. Soal 3

```

1 %% Soal 3
2 import pandas as izal #untuk import library pandas berguna
# untuk mengelola dataframe
3 izal = izal.read_csv("F:// Semester 6/Artificial Intelligence/
# Tugas 4/src/Youtube03-LMFAO.csv") #membaca file dengan
# format csv
4
5 spam=izal.query('CLASS == 1') #membagi tabel spam
6 nospam=izal.query('CLASS == 0')#membagi tabel no spam
7
8 from sklearn.feature_extraction.text import CountVectorizer #
# untuk import countvectorizer berfungsi untuk memecah data
# tersebut menjadi sebuah kata yang lebih sederhana
9 vectorizer = CountVectorizer () #ntuk menjalankan fungsi
# tersebut , pada code ini tidak ada hasilnya dikarenakan
# spyder tidak mendukung hasil dari instasiasi.
10
11 dvec = vectorizer.fit_transform(izal[ 'CONTENT' ]) #untuk
# melakukan pemecahan data pada dataframe yang terdapat pada
# kolom konten
12 dvec #Untuk menampilkan hasil dari code sebelumnya
13
14 Daptarkata= vectorizer.get_feature_names()
15
16 dshuf = izal.sample(frac=1)
17
18 d_train=dshuf[:300]
19 d_test=dshuf[300:]
20
21 d_train_att = vectorizer.fit_transform(d_train[ 'CONTENT' ])
22 d_train_att
23
24 d_train_label=d_train[ 'CLASS' ]
25 d_test_label=d_test[ 'CLASS' ]

```

Dengan menggunakan  $1174070 \bmod 4$  adalah 1, yang artinya menggunakan dataset LMFAO. Hasilnya adalah sebagai berikut :

Fancy - Dataframe

| Index | COMMENT_ID     | AUTHOR        | DATE            | CONTENT                                      | CLASS |
|-------|----------------|---------------|-----------------|--|-------|
| 0     | i123wnehdip..  | Corey Wilson  | 2015-05-28T22.. | (a   | 0     |
| 1     | i124vcrzsd..   | Toshi Yamada  | 2015-05-28T22.. | ...r...n...g...p...r...i...l...d... o...u... | 0     |
| 2     | i125tjcyjb..   | Lek Music     | 2015-05-28T22.. | Hey guys,...                                 | 1     |
| 3     | i127trqhoqa..  | Cheyell Fox   | 2015-05-28T22.. | Part 1 in the                                | 0     |
| 4     | i128pxivx4nd.. | PATRICK TU    | 2015-05-28T22.. | Rock...lol...                                | 0     |
| 5     | i129xyewu..    | Brian Brad    | 2015-05-28T22.. | Surprise                                     | 0     |
| 6     | i12bedsl1wy..  | Brian Brad    | 2015-05-28T22.. |  | 0     |
| 7     | i1311k06og..   | Alex Borko    | 2015-05-28T22.. | This song is                                 | 0     |
| 8     | i133ghmehob1.. | Glowwood      | 2015-05-28T22.. | Just really =)                               | 0     |
| 9     | i134vmt63l..   | Jesmet        | 2015-05-28T22.. | Answered :)                                  | 0     |
| 10    | i135emzcnn..   | Silvia Basco  | 2015-05-28T22.. | Incredible song,                             | 0     |
| 11    | i136v7x7vnd..  | Unikide       | 2015-05-28T22.. | song so much                                 | 0     |
| 12    | i1374h5mt..    | Hofigged      | 2015-05-28T22.. | 2015 Lifefit                                 | 0     |
| 13    | i138trxtetp..  | Alex Martin   | 2015-05-28T22.. | people dress,                                | 0     |
| 14    | i139uh2ztrv..  | Alex Jensen   | 2015-05-28T22.. | last year of,                                | 0     |
| 15    | i13avf1x3hny.. | Aiden Hill    | 2015-05-28T22.. | Best song                                    | 0     |
| 16    | i13t7tpqz1q..  | [Joe penti..  | 2015-05-28T22.. | super nice,                                  | 0     |
| 17    | i13zgqjxk1w..  | Silvia Kewo   | 2015-05-28T22.. | oooooooooooo                                 | 0     |
| 18    | i1372ng0gl1..  | SoulInTheHaus | 2015-05-27T1..  | PARTY ROCK                                   | 0     |
| 19    | i13dy1gnqmc..  | Phuong Linh   | 2015-05-27T1..  | Thumbs up if                                 | 0     |
| 20    | i13ef7sp9om..  | Goncalo       | 2015-05-27T1..  | this is                                      | 0     |
|       | i13ewqjgv0e..  | atwar qpm     | 2015-05-27T1..  | LIFEA!!!111111                               | 0     |

Gambar 2.25 Hasil Soal 3.

#### 4. Soal 4

```
1 #%% Soal 4
2
3 from sklearn import svm
4 clfsvm = svm.SVR(gamma = 'auto')
5 clfsvm.fit(d_train_att, d_train_label)
```

Klasifikasi dari data vektorisasi menggunakan klasifikasi Decision Tree. Hasilnya adalah sebagai berikut :

```
In [7]: from sklearn import svm
... clfsvm = svm.SVR(gamma = 'auto')
... clfsvm.fit(d_train_att, d_train_label)
Out[7]: SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1,
         gamma='scale', kernel='rbf', max_iter=-1, shrinking=True, tol=0.001,
         verbose=False)
```

Gambar 2.26 Hasil Soal 4.

#### 5. Soal 5

```
1 #%%soal 5
2
3 from sklearn import tree
4 clftree = tree.DecisionTreeClassifier()
5 clftree.fit(d_train_att, d_train_label)
```

Plotlah confusion matrix dari praktik modul ini menggunakan matplotlib. Hasilnya adalah sebagai berikut :

```
In [8]: from sklearn import tree
       clftree = tree.DecisionTreeClassifier()
       ...; clftree.fit(d_train_att, d_train_label)
Out[8]:
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
                       criterion='gini',
                       max_depth=None, max_features=None,
                       max_leaf_nodes=None, min_impurity_decrease=0.0,
                       min_impurity_split=None, min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0,
                       presort='deprecated', random_state=None, splitter='best')
```

**Gambar 2.27** Hasil Soal 5.

## 6. Soal 6

```
1 #%%soal 6/
2
3 from sklearn.metrics import confusion_matrix
4 pred_labels=clftree.predict(d_test)
5 cm=confusion_matrix(d_test_label ,pred_labels)
6
7 #%%
8
9 import matplotlib.pyplot as plt
10
11 def plot_confusion_matrix(cm, classes,
12                           normalize=False,
13                           title='Confusion matrix',
14                           cmap=plt.cm.Blues):
15
16     if normalize:
17         cm = cm.astype('float') / cm.sum(axis=1)[:, np.
newaxis]
18         print("Normalized confusion matrix")
19     else:
20         print('Confusion matrix, without normalization')
21
22     print(cm)
```

Menjalankan program cross validation. Hasilnya adalah sebagai berikut :

```
In [11]: import matplotlib.pyplot as plt
...
...: def plot_confusion_matrix(cm, classes,
...:                         normalize=False,
...:                         title='Confusion matrix',
...:                         cmap=plt.cm.Blues):
...:
...:     if normalize:
...:         cm = cm.astype('float') / cm.sum(axis=1)[:, np.
newaxis]
...:         print("Normalized confusion matrix")
...:     else:
...:         print('Confusion matrix, without normalization')
...:
...:     print(cm)
...:     cm
```

**Gambar 2.28** Hasil Soal 6.

## 7. Soal 7

```
1 #%%soal 7
2
```

```

3 from sklearn.model_selection import cross_val_score
4
5 scores=cross_val_score(clftree,d_train_att,d_train_label, cv
6 =5)
7
8 skor_rata2=scores.mean()
9 skoresd=scores.std()

```

Tree.export graphviz merupakan fungsi yang menghasilkan representasi Graphviz dari decision tree, yang kemudian ditulis ke outfile. Disini akan menyimpan classifiernya, akan meng ekspor file student performance jika salah akan mengembalikan nilai fail. Hasilnya adalah sebagai berikut :

```

In [12]: from sklearn.model_selection import cross_val_score
...:
scores=cross_val_score(clftree,d_train_att,d_train_label, cv=5)
...:
skor_rata2=scores.mean()
...:
skoresd=scores.std()

```

**Gambar 2.29** Hasil Soal 7.

## 8. Soal 8

```

1 %%soal 8 /
2
3 max_features_opts = range(5, 50, 5) #max_features_opts
4 sebagai variabel untuk membuat range 5,50,5
5 n_estimators_opts = range(10, 200, 20) #n_estimators_opts
6 sebagai variabel untuk membuat range 10,200,20
7 rf_params = izal.empty((len(max_features_opts)*len(
8     n_estimators_opts),4), float) #rf_params sebagai variabel
9 untuk menjumlahkan yang sudah di tentukan sebelumnya
10 i = 0
11 for max_features in max_features_opts: #pengulangan
12     for n_estimators in n_estimators_opts: #pengulangan
13         clftree = RandomForestClassifier(max_features=
14             max_features, n_estimators=n_estimators) #menampilkan
15         variabel csf
16         scores = cross_val_score(clf, df_train_att,
17             df_train_label, cv=5) #scores sebagai variabel training
18         rf_params[i,0] = max_features #index 0
19         rf_params[i,1] = n_estimators #index 1
20         rf_params[i,2] = scores.mean() #index 2
21         rf_params[i,3] = scores.std() * 2 #index 3
22         i += 1 #dengan ketentuan i += 1
23         print("Max features: %d, num estimators: %d, accuracy
24 : %0.2f (+/- %0.2f)" %(max_features, n_estimators, scores.
25 mean(), scores.std() * 2))
26         #print hasil pengulangan yang sudah ditentukan

```

Buatlah program pengamatan komponen informasi. Jadi disini kita akan memprediksi nilai dari variabel test att dan test pass Hasilnya adalah sebagai berikut :

```

Max features: 25, num estimators: 100, accuracy: 0.46 (+/- 0.03)
Max features: 30, num estimators: 100, accuracy: 0.33 (+/- 0.02)
Max features: 35, num estimators: 50, accuracy: 0.43 (+/- 0.03)
Max features: 36, num estimators: 70, accuracy: 0.44 (+/- 0.03)
Max features: 36, num estimators: 100, accuracy: 0.44 (+/- 0.03)
Max features: 36, num estimators: 100, accuracy: 0.45 (+/- 0.03)
Max features: 36, num estimators: 130, accuracy: 0.45 (+/- 0.03)
Max features: 36, num estimators: 130, accuracy: 0.46 (+/- 0.03)
Max features: 36, num estimators: 170, accuracy: 0.46 (+/- 0.03)
Max features: 36, num estimators: 170, accuracy: 0.46 (+/- 0.03)
Max features: 36, num estimators: 180, accuracy: 0.34 (+/- 0.03)
Max features: 35, num estimators: 100, accuracy: 0.43 (+/- 0.03)
Max features: 35, num estimators: 100, accuracy: 0.44 (+/- 0.03)
Max features: 35, num estimators: 130, accuracy: 0.45 (+/- 0.04)
Max features: 35, num estimators: 130, accuracy: 0.46 (+/- 0.04)
Max features: 35, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 35, num estimators: 150, accuracy: 0.46 (+/- 0.03)
Max features: 35, num estimators: 170, accuracy: 0.46 (+/- 0.04)
Max features: 35, num estimators: 170, accuracy: 0.46 (+/- 0.04)
Max features: 35, num estimators: 180, accuracy: 0.34 (+/- 0.03)
Max features: 40, num estimators: 100, accuracy: 0.43 (+/- 0.03)
Max features: 40, num estimators: 100, accuracy: 0.44 (+/- 0.03)
Max features: 40, num estimators: 130, accuracy: 0.45 (+/- 0.03)
Max features: 40, num estimators: 130, accuracy: 0.45 (+/- 0.03)
Max features: 40, num estimators: 130, accuracy: 0.45 (+/- 0.04)
Max features: 40, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 40, num estimators: 150, accuracy: 0.46 (+/- 0.03)
Max features: 40, num estimators: 170, accuracy: 0.45 (+/- 0.04)
Max features: 40, num estimators: 170, accuracy: 0.45 (+/- 0.04)
Max features: 40, num estimators: 180, accuracy: 0.34 (+/- 0.03)
Max features: 45, num estimators: 100, accuracy: 0.43 (+/- 0.03)
Max features: 45, num estimators: 100, accuracy: 0.44 (+/- 0.03)
Max features: 45, num estimators: 130, accuracy: 0.45 (+/- 0.04)
Max features: 45, num estimators: 130, accuracy: 0.45 (+/- 0.04)
Max features: 45, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 45, num estimators: 150, accuracy: 0.46 (+/- 0.03)
Max features: 45, num estimators: 170, accuracy: 0.45 (+/- 0.04)
Max features: 45, num estimators: 170, accuracy: 0.46 (+/- 0.04)
Max features: 45, num estimators: 180, accuracy: 0.34 (+/- 0.03)
Max features: 45, num estimators: 180, accuracy: 0.43 (+/- 0.03)
Max features: 45, num estimators: 180, accuracy: 0.44 (+/- 0.03)
Max features: 45, num estimators: 190, accuracy: 0.45 (+/- 0.03)
Max features: 45, num estimators: 190, accuracy: 0.45 (+/- 0.03)
Max features: 45, num estimators: 190, accuracy: 0.46 (+/- 0.03)

```

**Gambar 2.30** Hasil Soal 8.

### 2.3.3 Penanganan Error

#### 1. ScreenShoot Error

```

FileNotFoundError: [Errno 2] File b'F://Semester 6/Artificial Intelligence/Tugas 4/srvcfmany.csv' does not exist: b'F://Semester 6/Artificial Intelligence/Tugas 4/srvcfmany.csv'

```

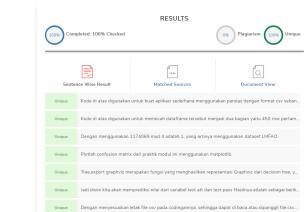
**Gambar 2.31** SyntaxError

#### 2. Cara Penanganan Error

- **SyntaxError**

Dengan menyesuaikan letak file csv pada codingannya, sehingga dapat di baca atau dipanggil file csvnya.

### 2.3.4 Bukti Tidak Plagiat

**Gambar 2.32** Bukti Tidak Melakukan Plagiat Chapter 4

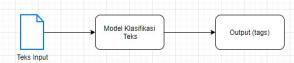
### 2.3.5 Link Youtube

## 2.4 1174066 - D.Irga B. Naufal Fakhri

### 2.4.1 Teori

#### 2.4.1.1 Klasifikasi Teks

Klasifikasi teks merupakan sebuah model yang biasa digunakan untuk mengkategorikan sebuah teks ke dalam kelompok-kelompok yang lebih terorganisir. Jadi untuk setiap kalimat yang di masukan ke dalam mesin, mesin tersebut akan menjadikan setiap kata dari kalimat tersebut menjadi sebuah kolom. Untuk ilustrasinya bisa dilihat pada gambar berikut :



**Gambar 2.33** Klasifikasi Teks.

#### 2.4.1.2 Jelaskan mengapa hal ini bisa terjadi, klasifikasi bunga tidak bisa digunakan untuk machine learning

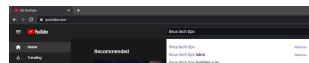
Karena machine learning tidak dapat menampilkan inputan sesuai dengan apa yang kita inputkan. Karena inputan tersebut serupa namun mesin memberikan output yang berbeda, biasanya output atau error ini disebut dengan istilah noise. Untuk contoh sederhananya misalkan kita inputkan salah satu label yang terdapat pada bunga, output yang dihasilkan oleh mesin tersebut ialah label yang lain. Itu dikarenakan bunga banyak jenis yang serupa namun tidak sama. Untuk ilustrasinya bisa dilihat pada gambar berikut:



**Gambar 2.34** Klasifikasi Bunga.

#### 2.4.1.3 Jelaskan bagaimana yang dimaksud dengan teknik pembelajaran mesin pada teks yang digunakan

Teknik yang digunakan pada youtube salah satunya ialah keywords. Dengan keywords tersebut mesin dapat memberikan video sesuai dengan keyword yang kita inputkan pada kolom pencarian. Teknik pembelajarannya tergantung user memberikan input teks seperti apa, karena pada youtube itu sendiri akan menyesuaikan dengan apa yang biasa kita inputkan dan akan memfilter video secara otomatis sesuai dengan keyword yang biasa kita inputkan. Contoh ilustrasi sederhananya seperti berikut:



**Gambar 2.35** Klasifikasi Teks pada Youtube.

#### 2.4.1.4 Vektorisasi Data

Vektorisasi data ialah suatu pemecahan atau pembagian data berupa teks, sebagai contoh terdapat 5 paragraf, data teks tersebut di pecah menjadi kalimat-kalimat yang lebih sederhana, lalu di pecah lagi menjadi kata untuk setiap kalimatnya.

#### 2.4.1.5 Bag of Words

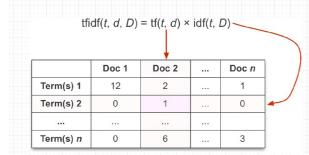
Representasi penyederhanaan sebuah kalimat atau perhitungan setiap kata pada suatu kalimat dengan presentase berapa kali muncul kata tersebut untuk setiap kalimatnya. Contoh ilustrasi sederhananya seperti berikut:



**Gambar 2.36** Bag of Words.

#### 2.4.1.6 TF-IDF

TF-IDF merupakan metode untuk menghitung bobot setiap kata pada suatu kalimat yang paling sering digunakan. TF-IDF ini akan menghitung nilai Term Frequency dan Inverse Document Frequency pada setiap kata dalam setiap kalimat yang muncul dengan diimbangi dengan jumlah dokumen dalam korpus yang mengandung kata. Contoh ilustrasi sederhananya seperti gambar berikut:



**Gambar 2.37** TF-IDF.

#### 2.4.2 Praktek Program

##### 2.4.2.1 Nomor 1

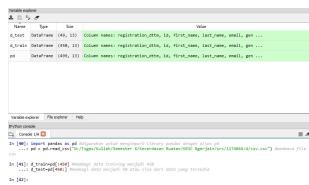
```
1 import pandas as pd #digunakan untuk mengimport library pandas
    dengan alias pd
2 pd = pd.read_csv("N:/Tugas/Kuliah/Semester 6/Kecerdasan Buatan/
    KB3C Ngerjain/src/1174066/4/csv.csv") #membaca file csv
```

```
In [4]: import pandas as pd #digunakan untuk mengimport library pandas dengan alias pd
```

Gambar 2.38 Nomor 1

## 2.4.2.2 Nomor 2

```
1 d_train=pd[:450] #membagi data training menjadi 450
2 d_test=pd[450:] #membagi data menjadi 50 atau sisa dari data yang tersedia
```



Gambar 2.39 Nomor 2

## 2.4.2.3 Nomor 3

```
1 import pandas as pd #digunakan untuk mengimport library pandas dengan alias pd
2 d = pd.read_csv("N:/Tugas/Kuliah/Semester 6/Kecerdasan Buatan/KB3C Ngerjain/src/1174066/4/Youtube04-Eminem.csv") #Membaca file csv
3
4 from sklearn.feature_extraction.text import CountVectorizer # import fungsi countvectorize dari sklearn
5 vectorizer = CountVectorizer() #membuat instansi CountVectorizer
6
7 dvec = vectorizer.fit_transform(d['CONTENT']) #Memasukkan data ke dvec
8 dvec #Melihat data yang dimasukkan ke dvec
9
10 daptarkata = vectorizer.get_feature_names() #Mendapatkan data dan memasukkannya ke daptarkata
11
12 dshuf = d.sample(frac=1) #Memasukkan sample kedalam variable dshuf
13
14 d_train = dshuf[:300] #Membuat data training
15 d_test = dshuf[300:] #Membuat data test
16
17 d_train_att = vectorizer.fit_transform(d_train['CONTENT']) # Memasukkan data training dari vectorizer
18 d_train_att #Melihat data training
19
20 d_test_att = vectorizer.transform(d_test['CONTENT']) #Memasukkan data test dari vectorizer
21 d_test_att #Melihat data training
```

```

22
23 d_train_label = d_train['CLASS'] #Memberi label
24 d_test_label = d_test['CLASS'] #Memberi Label

```

The screenshot shows a Jupyter Notebook interface with two code cells. The first cell contains the code provided above. The second cell's output is displayed in a green box:

```

In [42]: d_train_label = d_train['CLASS']
        d_test_label = d_test['CLASS']

d_train_label
Out[42]: Series[1000, 1] Column names: CLASS_ID, AUTHOR, DATE, CONTEXT, CLASS
          Series object of pandas.core.series module

d_train_label
Out[43]: Series[1000, 1] Column names: CLASS_ID, AUTHOR, DATE, CONTEXT, CLASS
          Series object of pandas.core.series module

d_train_label
Out[44]: Series[1000, 1] Column names: CLASS_ID, AUTHOR, DATE, CONTEXT, CLASS
          Series object of pandas.core.series module

d_train_label
Out[45]: Series[1000, 1] Column names: CLASS_ID, AUTHOR, DATE, CONTEXT, CLASS
          Series object of pandas.core.series module

```

Gambar 2.40 Nomor 3

#### 2.4.2.4 Nomor 4

```

1 from sklearn import svm #Mengimport svm dari sklearn
2 clfsvm = svm.SVC() #Membuat svc kedalam variable svm
3 clfsvm.fit(d_train_att, d_train_label) #Memprediksi data dari
   data training
4 clfsvm.score(d_test_att, d_test_label) #Memunculkan clf sebagai
   testing yang sudah di training tadi

```

```

In [46]: from sklearn import svm
        clfsvm = svm.SVC()
        clfsvm.fit(d_train_att, d_train_label)
        clfsvm.score(d_test_att, d_test_label)
C:\Users\HP\PycharmProjects\Project\venv\lib\site-packages\sklearn\svm\__init__.py:139: FutureWarning: The default value of gamma will change
from 'auto' to 'scale' in version 0.20. Set gamma explicitly to 'auto' or 'scale' to
avoid this warning.
  warnings.warn(_FUTURE_WARNING_MSG, FutureWarning)
Out[46]: 0.9594594594594594

```

Gambar 2.41 Nomor 4

#### 2.4.2.5 Nomor 5

```

1 from sklearn import tree #Mengimport tree dari sklearn
2 clftree = tree.DecisionTreeClassifier() #Membuat decision tree
3 clftree.fit(d_train_att, d_train_label) #Memprediksi data dari
   data training
4 clftree.score(d_test_att, d_test_label) #Memunculkan clf sebagai
   testing yang sudah di training tadi

```

```

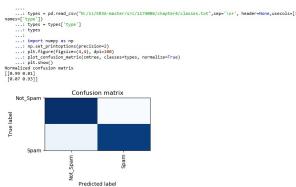
In [45]: from sklearn import tree
        clftree = tree.DecisionTreeClassifier()
        clftree.fit(d_train_att, d_train_label)
        clftree.score(d_test_att, d_test_label)
Out[45]: 0.9594594594594594

```

Gambar 2.42 Nomor 5

#### 2.4.2.6 Nomor 6

```
1 from sklearn.metrics import confusion_matrix
2 pred_labelstree = clftree.predict(d_test_att)
3 cmtree = confusion_matrix(d_test_label, pred_labelstree)
4 cmtree
5
6 import matplotlib.pyplot as plt
7 import itertools
8 def plot_confusion_matrix(cm, classes,
9                           normalize=False,
10                           title='Confusion matrix',
11                           cmap=plt.cm.Blues):
12     """
13     This function prints and plots the confusion matrix.
14     Normalization can be applied by setting 'normalize=True'.
15     """
16     if normalize:
17         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
18         print("Normalized confusion matrix")
19     else:
20         print('Confusion matrix, without normalization')
21
22     print(cm)
23
24     plt.imshow(cm, interpolation='nearest', cmap=cmap)
25     plt.title(title)
26     #plt.colorbar()
27     tick_marks = np.arange(len(classes))
28     plt.xticks(tick_marks, classes, rotation=90)
29     plt.yticks(tick_marks, classes)
30
31     fmt = '.2f' if normalize else 'd'
32     thresh = cm.max() / 2.
33     #for i, j in itertools.product(range(cm.shape[0]), range(cm.
34     #shape[1])):
35     #    plt.text(j, i, format(cm[i, j], fmt),
36     #              horizontalalignment="center",
37     #              color="white" if cm[i, j] > thresh else "black"
38     #)
39
40     plt.tight_layout()
41     plt.ylabel('True label')
42     plt.xlabel('Predicted label')
43 types = pd.read_csv("N:/zz/KB3A-master/src/1174006/chapter4/
44   classes.txt", sep='\s+', header=None, usecols=[1], names=['type
45   '])
46 types = types['type']
47 types
48
49 import numpy as np
50 np.set_printoptions(precision=2)
51 plt.figure(figsize=(4,4), dpi=100)
52 plot_confusion_matrix(cmtree, classes=types, normalize=True)
53 plt.show()
```



Gambar 2.43 Nomor 6

## 2.4.2.7 Nomor 7

```

1 from sklearn.model_selection import cross_val_score
2
3 scorestree = cross_val_score(clftree, d_train_att, d_train_label,
4                             cv=5)
5 print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
6                                         scorestree.std() * 2))
7
8 scoresvm = cross_val_score(clfsvm, d_train_att, d_train_label,
9                             cv=5)
10 print("Accuracy: %0.2f (+/- %0.2f)" % (scoresvm.mean(),
11                                         scoresvm.std() * 2))
12
13 scores = cross_val_score(clf, d_train_att, d_train_label, cv=5)
14 print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std()
15                                         () * 2))

```

```
In [30]: from sklearn.model_selection import cross_val_score
... scorestree = cross_val_score(clftree, d_train_att, d_train_label, cv=5)
... print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
...                                         scorestree.std() * 2))
... scoresvm = cross_val_score(clfsvm, d_train_att, d_train_label, cv=5)
... print("Accuracy: %0.2f (+/- %0.2f)" % (scoresvm.mean(),
...                                         scoresvm.std() * 2))
... scores = cross_val_score(clf, d_train_att, d_train_label, cv=5)
... print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std()
...                                         () * 2))

Accuracy: 0.97 (+/- 0.00)
Accuracy: 0.97 (+/- 0.00)
Accuracy: 0.98 (+/- 0.00)

In [31]: # We can see that all three classifiers perform well. RandomForestClassifier has the best accuracy. Let's use it to predict the test set.
... from sklearn.metrics import classification_report, confusion_matrix
... y_pred = clf.predict(d_test_att)
... print(classification_report(d_test_label, y_pred))
... print(confusion_matrix(d_test_label, y_pred))

Classification report:
              precision    recall  f1-score   support
  ham (0)       0.97      0.97      0.97     459
spam (1)       0.97      0.97      0.97      50
avg / total    0.97      0.97      0.97     509

Confusion matrix:
[[459  10]
 [ 10  50]]
```

Gambar 2.44 Nomor 7

## 2.4.2.8 Nomor 8

```

1 max_features_opts = range(5, 50, 5)
2 n_estimators_opts = range(10, 200, 20)
3 rf_params = np.empty((len(max_features_opts)*len(
4                         n_estimators_opts),4), float)
5 i = 0
6 for max_features in max_features_opts:
7     for n_estimators in n_estimators_opts:
8         clf = RandomForestClassifier(max_features=max_features,
9                                     n_estimators=n_estimators)
10        scores = cross_val_score(clf, d_train_att, d_train_label,
11                                  cv=5)
12        rf_params[i,0] = max_features
13        rf_params[i,1] = n_estimators
14        rf_params[i,2] = scores.mean()
15        rf_params[i,3] = scores.std()
16 i += 1

```



## 2.5 1174083 - Bakti Qilan Mufid

### Chapter-4 Klasifikasi Teks

#### 2.5.1 Teori

##### 2.5.1.1 Jelaskan apa itu klasifikasi teks, sertakan gambar iustrasi buatan sendiri

Klasifikasi teks merupakan salah satu tugas terpenting dalam Pemrosesan Bahasa Alami (Natural Language Processing). Ini adalah proses mengklasifikasikan string teks atau dokumen ke dalam kategori yang berbeda, tergantung pada konten string. Klasifikasi teks memiliki berbagai aplikasi, seperti mendeteksi sentimen pengguna dari tweet, mengklasifikasikan email sebagai spam atau ham, mengklasifikasikan posting blog ke dalam kategori yang berbeda, penandaan otomatis permintaan pelanggan, dan sebagainya. Berikut adalah contoh dari Klasifikasi Teks. Contohnya, misal kita ingin mencari kata dog, is, table, on, the . kemudian jika kata yang dimaksud sesuai maka akan menampilkan bilangan biner 1 dan jika salah 0. Seperti dibawah ini :

the dog is on the table



Gambar 2.48 Klasifikasi Teks

##### 2.5.1.2 Jelaskan mengapa klasifikasi bunga tidak bisa digunakan untuk machine learning, sertakan ilustrasi gambar sendiri

Dikarenakan tidak semua bunga memiliki ciri - ciri yang sama. Atau dalam kata lain terdapat data noise dalam klasifikasi bunga sehingga tidak bisa menggunakan machine learning. Contohnya Anggrek memiliki warna ungu, dengan jumlah kelopak 5. Kemudian ada bunga warna ungu dengan jumlah kelopak yang sama namun ternyata bukan anggrek dan kategorinya banyak sekali. Bahkan ada bunga yang tidak jelas apakah warnanya sesuai atau tidak, sehingga bisa menyebabkan data noise.



**Gambar 2.49** Klasifikasi Bunga

**2.5.1.3** *Jelaskan bagaimana teknik pembelajaran mesin pada teks pada kata-kata yang digunakan di youtube, jelaskan arti per atribut data csv dan sertakan ilustrasi buatan sendiri.*

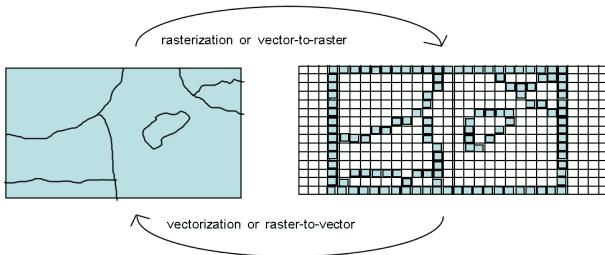
Menggunakan teknik bag-of-words pada klasifikasi berbasis text dan kata untuk mengklasifikasikan komentar yang ada di internet sebagai spam atau bukan. Misalkan pada kolom komentar dapat di cek seberapa sering suatu kata muncul dalam kalimat. Setiap kata dapat dijadikan baris dan kolomnya ini merupakan kategori kata tersebut, apakah masuk kedalam spam atau tidak. dan contoh lainnya yaitu pada Caption. dimana akan muncul subtitle secara otomatis dari youtube menggunakan sensor suara yang disesuaikan dengan kata yang telah ditentukan. Contohnya seperti berikut :

| CONTENT  | CLASS |
|--|-------|
| Huh, anyway check out this you[tube] channel: kobyoshi02   | 1     |
| Hey guys check out my new channel and our first vid THIS IS US THE MONKEYS!!! I'm the monkey in the white shirt,please leave a like comment and please subscribe!!!! | 1     |
| just for test I have to say murdev.com   | 1     |
| me shaking my sexy ass on my channel enjoy ^_^   | 1     |
| watch?v=vtaRGgvGtW0 Check this out .   | 1     |
| Hey, check out my new website!! This site is about kids stuff. kidsmediausa . com  | 1     |
| Subscribe to my channel  | 1     |
| i turned it on mute as soon as i came on i just wanted to check the views...   | 0     |
| You should check my channel for Funny VIDEOS!!   | 1     |
| and u shoud d check my channel and tell me what I should do next!  | 1     |

**Gambar 2.50** Klasifikasi Spam Comment di Youtube

**2.5.1.4** *Jelaskan apa yang dimaksud vektorisasi data.*

Vektorisasi adalah proses konversi data raster(gambar, pindaian) menjadi data vektor yang lebih umum disebut dengan istilah digitalisasi adapun aktifitasnya disebut digitasi. Wujud digitalisasi ini diklasifikasikan secara spesifik dalam tema-tema tertentu yang direpresentasikan oleh bentuk garis, poligon dan titik. Pada akhirnya proses vektorisasi ini menghasilkan suatu wujud topografi yang menggambarkan keadaan permukaan bumi atau bentang alam. Sifat data yang geometris menunjukkan ukuran dimensi yang sesungguhnya.



**Gambar 2.51** Vektorisasi dan Rasterisasi

**2.5.1.5** *Jelaskan apa itu bag of words dengan kata-kata yang sederhana dan ilustrasi sendiri.*

bag-of-words merupakan representasi teks yang menggambarkan kemunculan kata-kata dalam dokumen. Pengelompokan kata kata kedalam perhitungan, berapakah sebanyak berapa kali sebuah kata muncul dalam satu kalimat. Disebut "bag" kata-kata, karena informasi tentang susunan atau struktur kata dalam dokumen dibuang. Model ini hanya berkaitan dengan apakah kata-kata yang diketahui muncul dalam dokumen, bukan di mana dalam dokumen. Contohnya disini akan melihat kemunculan kata dari kalimat :

1. I Love Dogs
2. I hate dogs and knitting
3. Knitting is my hobby and passion.

|       | I | love | dogs | hate | and | knitting | is | my | hobby | passion |
|-------|---|------|------|------|-----|----------|----|----|-------|---------|
| Doc 1 | 1 | 1    | 1    |      |     |          |    |    |       |         |
| Doc 2 | 1 |      | 1    | 1    | 1   | 1        |    |    |       |         |
| Doc 3 |   |      |      |      | 1   | 1        | 1  | 2  | 1     | 1       |

**Gambar 2.52** Bag-Of-Words

**2.5.1.6** *Jelaskan apa itu TF-IDF, ilustrasikan dengan gambar sendiri.*

TF-IDF memberi kita frekuensi kata dalam setiap dokumen dalam korpus atau mengganti data jadi number. Ini adalah rasio berapa kali kata itu muncul dalam dokumen dibandingkan dengan jumlah total kata dalam dokumen itu. Itu meningkat seiring jumlah kemunculan kata itu di dalam dokumen meningkat. Setiap dokumen memiliki tf sendiri. Dalam ilustrasi disini saya akan mengganti contoh Bag of Words menjadi bentuk TF-IDF.

|       |      |             |      |             |      |          |             |             |             |             |
|-------|------|-------------|------|-------------|------|----------|-------------|-------------|-------------|-------------|
|       | I    | love        | dogs | hate        | and  | knitting | is          | my          | hobby       | passion     |
| Doc 1 | 0.18 | <b>0.48</b> | 0.18 |             |      |          |             |             |             |             |
| Doc 2 | 0.18 |             | 0.18 | <b>0.48</b> | 0.18 | 0.18     |             |             |             |             |
| Doc 3 |      |             |      |             | 0.18 | 0.18     | <b>0.48</b> | <b>0.95</b> | <b>0.48</b> | <b>0.48</b> |

Gambar 2.53 Contoh TF-IDF

## 2.5.2 Praktek

2.5.2.1 buat aplikasi klasifikasi sederhana menggunakan pandas, buat data dummy format csv sebanyak 500 baris dan melakukan load ke dataframe panda.jelaskan arti setiap baris kode yang dibuat(harus beda dengan teman sekelas)

```
1 import pandas as pd #import package pandas, lalu dialiaskan
    menjadi pd.
2 marvel = pd.read_csv ('E:/backup/sem 6/Kecerdasan Buatan/KB3C –
    Copy/src/1174083/src4/Marvel.csv', sep=',') #membaca file csv
    dimana data pada file csv dipisahkan oleh koma, lalu
    ditampung di variable marvel.
```

Listing 2.1 kodingan praktek no. 1

Gambar 2.54 hasil praktek soal no. 1

2.5.2.2 dari dataframe tersebut dipecah menjadi dua dataframe yaitu 450 row pertama dan 50 row sisanya(harus beda dengan teman sekelas)

```

1 marvel1 , marvel2 = marvel[:450] , marvel[450:] #membagi data
  menjadi dua bagian , variable marvel1 untuk menampung 450
  baris data pertama , variable marvel2 untuk menampung 50 baris
  data terakhir .

```

**Listing 2.2** kodingan praktek no. 2

| Name    | Type      | Size     | Value  |
|---------|-----------|----------|--|
| marvel  | DataFrame | (500, 7) | Column names: name, ID, ALIGN, EYE, HAIR, SEX, ALIVE |
| marvel1 | DataFrame | (450, 7) | Column names: name, ID, ALIGN, EYE, HAIR, SEX, ALIVE |
| marvel2 | DataFrame | (50, 7)  | Column names: name, ID, ALIGN, EYE, HAIR, SEX, ALIVE |

**Gambar 2.55** hasil praktek soal no. 2

2.5.2.3 praktekkan vektorisasi dan klasifikasi dari data(NPM mod 4, jika 0 maka ketty perry, 1 LMFAO, 2 Eminem, 3 Shakira) dengan Decission Tree. Tunjukkan keluaranya dari komputer sendiri dan artikan maksud luaran yang di dapatkan

```

1 print(1174083%4) #hasilnya 3 , maka Shakira

```

**Listing 2.3** 1174083 mod 4

```

1 # Vektorisasi Data
2 import pandas as pd
3 d = pd.read_csv("Youtube05-Shakira.csv")
4
5 from sklearn.feature_extraction.text import CountVectorizer
6 vectorizer = CountVectorizer()
7
8 dvec = vectorizer.fit_transform(d['CONTENT'])
9 dvec
10
11 daptarkata = vectorizer.get_feature_names()
12
13 dshuf = d.sample(frac=1)
14
15 d_train = dshuf[:300]
16 d_test = dshuf[300:]
17
18 d_train_att = vectorizer.fit_transform(d_train['CONTENT'])
19 d_train_att
20
21 d_test_att = vectorizer.transform(d_test['CONTENT'])
22 d_test_att
23
24 d_train_label = d_train['CLASS']
25 d_test_label = d_test['CLASS']

```

**Listing 2.4** kodingan praktek no. 3

```
In [7]: import pandas as pd
...: d = pd.read_csv("Youtube05-Shakira.csv")
...:
...: from sklearn.feature_extraction.text import CountVectorizer
...: vectorizer = CountVectorizer()
...:
...: dvec = vectorizer.fit_transform(d['CONTENT'])
...: dvec
...:
...: daptarkata = vectorizer.get_feature_names()
...:
...: dshuf = d.sample(frac=1)
...:
...: d_train = dshuf[:300]
...: d_test = dshuf[300:]
...:
...: d_train_att = vectorizer.fit_transform(d_train['CONTENT'])
...: d_train_att
...:
...: d_test_att = vectorizer.transform(d_test['CONTENT'])
...: d_test_att
...:
...: d_train_label = d_train['CLASS']
...: d_test_label = d_test['CLASS']
```

**Gambar 2.56** hasil praktek soal no. 3(1)

Vektorisasi data content dari file Youtube04\_Shakira.CSV

| Name          | Type      | Size     | Value   |
|---------------|-----------|----------|---|
| d             | DataFrame | (370, 5) | Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS                      |
| d_test        | DataFrame | (70, 5)  | Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS                      |
| d_test_label  | Series    | (70,)    | Series object of pandas.core.series module                                  |
| d_train       | DataFrame | (300, 5) | Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS                      |
| d_train_label | Series    | (300,)   | Series object of pandas.core.series module                                  |
| daptarkata    | list      | 1357     | ['00', '000', '0687119038', '08', '10', '100', '101721377578919894134' ...] |
| dshuf         | DataFrame | (370, 5) | Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS                      |

**Gambar 2.57** hasil praktek soal no. 3(2)

**2.5.2.4 Cobalah klarifikasi dari data vektorisasi yang di tentukan di nomor sebelumnya dengan klasifikasi SVM. Tunjukkan keluaranya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan**

```
1 from sklearn import svm
2 clfsvm = svm.SVC()
3 clfsvm.fit(d_train_att, d_train_label)
4 clfsvm.score(d_test_att, d_test_label)
```

**Listing 2.5** kodingan praktek no. 4

```
In [9]: from sklearn import svm
...: clfsvm = svm.SVC()
...: clfsvm.fit(d_train_att, d_train_label)
...: clfsvm.score(d_test_att, d_test_label)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better
for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
Out[9]: 0.6857142857142857
```

**Gambar 2.58** hasil praktek soal no. 4

**2.5.2.5 Cobalah klasifikasikan dari data vektorisasi yang ditentukan dari nomor sebelumnya dengan klasifikasi Decision Tree. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan**

```
1 from sklearn import tree
2 clftree = tree.DecisionTreeClassifier()
3 clftree.fit(d_train_att, d_train_label)
4 clftree.score(d_test_att, d_test_label)
```

**Listing 2.6** kodingan praktek no. 5

```
In [8]: from sklearn import tree
...: clftree = tree.DecisionTreeClassifier()
...: clftree.fit(d_train_att, d_train_label)
...: clftree.score(d_test_att, d_test_label)
Out[8]: 0.9285714285714286
```

**Gambar 2.59** hasil praktek soal no. 5

**2.5.2.6 Plotlah confusion matrix dari praktek modul ini menggunakan matplotlib. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.**

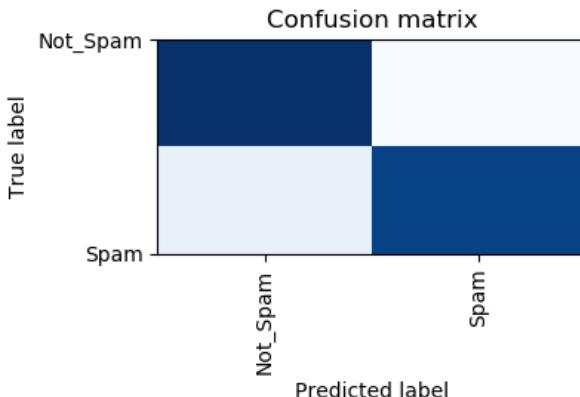
```
1 from sklearn.metrics import confusion_matrix
2 pred_labeltree = clftree.predict(d_test_att)
3 cmmtree = confusion_matrix(d_test_label, pred_labeltree)
4 cmmtree
5
6 import matplotlib.pyplot as plt
7 import itertools
8 def plot_confusion_matrix(cm, classes,
9                           normalize=False,
10                           title='Confusion matrix',
11                           cmap=plt.cm.Blues):
12     """
13     This function prints and plots the confusion matrix.
14     Normalization can be applied by setting `normalize=True`.
15     """
16     if normalize:
17         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
18         print("Normalized confusion matrix")
19     else:
```

```

20     print('Confusion matrix, without normalization')
21
22     print(cm)
23
24     plt.imshow(cm, interpolation='nearest', cmap=cmap)
25     plt.title(title)
26     #plt.colorbar()
27     tick_marks = np.arange(len(classes))
28     plt.xticks(tick_marks, classes, rotation=90)
29     plt.yticks(tick_marks, classes)
30
31     fmt = '.2f' if normalize else 'd'
32     thresh = cm.max() / 2.
33
34     plt.tight_layout()
35     plt.ylabel('True label')
36     plt.xlabel('Predicted label')
37
38 types = pd.read_csv("classes.txt", sep='\s+', header=None, usecols=[1], names=['type'])
39 types = types['type']
40 types
41
42 import numpy as np
43 np.set_printoptions(precision=2)
44 plt.figure(figsize=(4,4), dpi=100)
45 plot_confusion_matrix(cmtree, classes=types, normalize=True)
46 plt.show()

```

**Listing 2.7** kodingan praktek no. 6(1)



**Gambar 2.60** hasil praktek soal no. 6(1)

Plot confusion matrix dari klasifikasi Decission Tree.

```

1 from sklearn.metrics import confusion_matrix
2 pred_labelssvm = clfsvm.predict(d_test_att)

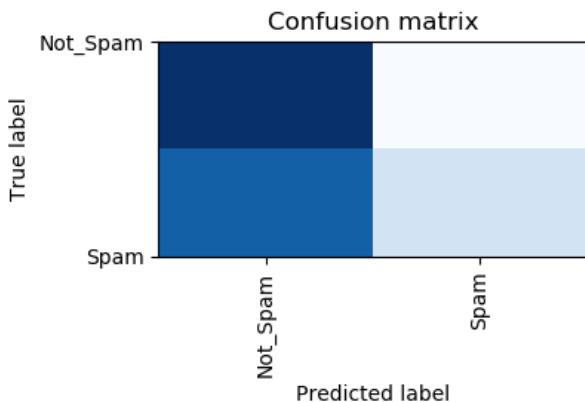
```

```

3 cmsvm = confusion_matrix(d_test_label , pred_labelssvm)
4 cmsvm
5
6 import matplotlib.pyplot as plt
7 import itertools
8 def plot_confusion_matrix(cm, classes,
9                           normalize=False,
10                           title='Confusion matrix',
11                           cmap=plt.cm.Blues):
12     """
13     This function prints and plots the confusion matrix.
14     Normalization can be applied by setting 'normalize=True'.
15     """
16     if normalize:
17         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
18         print("Normalized confusion matrix")
19     else:
20         print('Confusion matrix, without normalization')
21
22     print(cm)
23
24     plt.imshow(cm, interpolation='nearest', cmap=cmap)
25     plt.title(title)
26     #plt.colorbar()
27     tick_marks = np.arange(len(classes))
28     plt.xticks(tick_marks, classes, rotation=90)
29     plt.yticks(tick_marks, classes)
30
31     fmt = '.2f' if normalize else 'd'
32     thresh = cm.max() / 2.
33     #for i, j in itertools.product(range(cm.shape[0]), range(cm.
34     #shape[1])):
35     #    plt.text(j, i, format(cm[i, j], fmt),
36     #              horizontalalignment="center",
37     #              color="white" if cm[i, j] > thresh else "black"
38     )
39
40     plt.tight_layout()
41     plt.ylabel('True label')
42     plt.xlabel('Predicted label')
43
44 types = pd.read_csv("classes.txt", sep='\s+', header=None, usecols
45                     =[1], names=['type'])
46 types = types['type']
47
48 import numpy as np
49 np.set_printoptions(precision=2)
50 plt.figure(figsize=(4,4), dpi=100)
51 plot_confusion_matrix(cmsvm, classes=types, normalize=True)
52 plt.show()

```

**Listing 2.8** kodingan praktik no. 6(2)



**Gambar 2.61** hasil praktek soal no. 6(2)

Plot confusion matrix dari klasifikasi SVM.

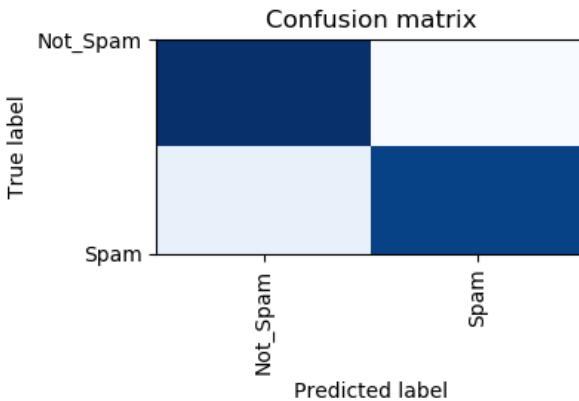
```

1 from sklearn.metrics import confusion_matrix
2 pred_labels = clf.predict(d_test_att)
3 cm = confusion_matrix(d_test_label, pred_labels)
4
5 import matplotlib.pyplot as plt
6 import itertools
7 def plot_confusion_matrix(cm, classes,
8                           normalize=False,
9                           title='Confusion matrix',
10                          cmap=plt.cm.Blues):
11     """
12     This function prints and plots the confusion matrix.
13     Normalization can be applied by setting 'normalize=True'.
14     """
15     if normalize:
16         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
17         print("Normalized confusion matrix")
18     else:
19         print('Confusion matrix, without normalization')
20
21     print(cm)
22
23     plt.imshow(cm, interpolation='nearest', cmap=cmap)
24     plt.title(title)
25     #plt.colorbar()
26     tick_marks = np.arange(len(classes))
27     plt.xticks(tick_marks, classes, rotation=90)
28     plt.yticks(tick_marks, classes)
29
30     fmt = '.2f' if normalize else 'd'
31     thresh = cm.max() / 2.
32     #for i, j in itertools.product(range(cm.shape[0]), range(cm.
33     #shape[1])):
34     #    plt.text(j, i, format(cm[i, j], fmt),
35

```

```
34 # horizontalalignment="center",
35 # color="white" if cm[i, j] > thresh else "black"
36 "
37 plt.tight_layout()
38 plt.ylabel('True label')
39 plt.xlabel('Predicted label')
40
41 types = pd.read_csv("classes.txt", sep='\s+', header=None, usecols=[1], names=['type'])
42 types = types['type']
43 types
44
45 import numpy as np
46 np.set_printoptions(precision=2)
47 plt.figure(figsize=(4,4), dpi=100)
48 plot_confusion_matrix(cmtree, classes=types, normalize=True)
49 plt.show()
```

**Listing 2.9** kodingan praktek no. 6(3)



**Gambar 2.62** hasil praktik soal no. 6(3)

Plot confusion matrix dari klasifikasi Random Forest.

2.5.2.7 jalankan program cross validaiton pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```
1 from sklearn.model_selection import cross_val_score
2
3 scorestree = cross_val_score(clftree, d_train_att, d_train_label,
4 cv=5)
5 print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
6 scorestree.std() * 2))
```

```

5
6 scoreSVM = cross_val_score(clfSVM, d_train_att, d_train_label,
7     cv=5)
7 print("Accuracy: %0.2f (+/- %0.2f)" % (scoreSVM.mean(),
8     scoreSVM.std() * 2))
8
9 scores = cross_val_score(clf, d_train_att, d_train_label, cv=5)
10 print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std()
10     () * 2))

```

**Listing 2.10** kodingan praktek no. 7

**Accuracy: 0.92 (+/- 0.05)**  
**Accuracy: 0.66 (+/- 0.05)**  
**Accuracy: 0.94 (+/- 0.03)**

**Gambar 2.63** hasil praktek soal no. 7

**2.5.2.8** *Buatlah program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.*

```

1 max_features_opts = range(5, 50, 5)
2 n_estimators_opts = range(10, 200, 20)
3 rf_params = np.empty((len(max_features_opts)*len(
4     n_estimators_opts),4), float)
4 i = 0
5 for max_features in max_features_opts:
6     for n_estimators in n_estimators_opts:
7         clf = RandomForestClassifier(max_features=max_features,
8             n_estimators=n_estimators)
9         scores = cross_val_score(clf, d_train_att, d_train_label,
10             cv=5)
11         rf_params[i,0] = max_features
12         rf_params[i,1] = n_estimators
13         rf_params[i,2] = scores.mean()
14         rf_params[i,3] = scores.std() * 2
15         i += 1
16     print("Max features: %d, num estimators: %d, accuracy:
16         %0.2f (+/- %0.2f)" % (max_features,
16             n_estimators, scores.mean(), scores.std() * 2))

```

**Listing 2.11** kodingan praktek no. 8

```
Max features: 5, num estimators: 10, accuracy: 0.89 (+/- 0.09)
Max features: 5, num estimators: 30, accuracy: 0.90 (+/- 0.07)
Max features: 5, num estimators: 50, accuracy: 0.91 (+/- 0.08)
Max features: 5, num estimators: 70, accuracy: 0.92 (+/- 0.07)
Max features: 5, num estimators: 90, accuracy: 0.93 (+/- 0.07)
Max features: 5, num estimators: 110, accuracy: 0.93 (+/- 0.07)
Max features: 5, num estimators: 130, accuracy: 0.94 (+/- 0.04)
Max features: 5, num estimators: 150, accuracy: 0.93 (+/- 0.05)
Max features: 5, num estimators: 170, accuracy: 0.93 (+/- 0.05)
Max features: 5, num estimators: 190, accuracy: 0.92 (+/- 0.06)
Max features: 10, num estimators: 10, accuracy: 0.92 (+/- 0.07)
Max features: 10, num estimators: 30, accuracy: 0.93 (+/- 0.04)
Max features: 10, num estimators: 50, accuracy: 0.94 (+/- 0.03)
Max features: 10, num estimators: 70, accuracy: 0.93 (+/- 0.05)
Max features: 10, num estimators: 90, accuracy: 0.93 (+/- 0.04)
Max features: 10, num estimators: 110, accuracy: 0.92 (+/- 0.06)
Max features: 10, num estimators: 130, accuracy: 0.93 (+/- 0.04)
Max features: 10, num estimators: 150, accuracy: 0.94 (+/- 0.03)
Max features: 10, num estimators: 170, accuracy: 0.93 (+/- 0.04)
Max features: 10, num estimators: 190, accuracy: 0.93 (+/- 0.04)
Max features: 15, num estimators: 10, accuracy: 0.90 (+/- 0.05)
Max features: 15, num estimators: 30, accuracy: 0.93 (+/- 0.04)
Max features: 15, num estimators: 50, accuracy: 0.93 (+/- 0.05)
Max features: 15, num estimators: 70, accuracy: 0.92 (+/- 0.05)
Max features: 15, num estimators: 90, accuracy: 0.94 (+/- 0.04)
Max features: 15, num estimators: 110, accuracy: 0.94 (+/- 0.03)
Max features: 15, num estimators: 130, accuracy: 0.93 (+/- 0.04)
Max features: 15, num estimators: 150, accuracy: 0.94 (+/- 0.03)
Max features: 15, num estimators: 170, accuracy: 0.94 (+/- 0.03)
Max features: 15, num estimators: 190, accuracy: 0.93 (+/- 0.04)
Max features: 20, num estimators: 10, accuracy: 0.91 (+/- 0.05)
Max features: 20, num estimators: 30, accuracy: 0.91 (+/- 0.06)
Max features: 20, num estimators: 50, accuracy: 0.94 (+/- 0.04)
Max features: 20, num estimators: 70, accuracy: 0.92 (+/- 0.04)
Max features: 20, num estimators: 90, accuracy: 0.93 (+/- 0.04)
Max features: 20, num estimators: 110, accuracy: 0.93 (+/- 0.04)
Max features: 20, num estimators: 130, accuracy: 0.94 (+/- 0.03)
Max features: 20, num estimators: 150, accuracy: 0.93 (+/- 0.04)
Max features: 20, num estimators: 170, accuracy: 0.93 (+/- 0.04)
```

**Gambar 2.64** hasil praktek soal no. 8(1)

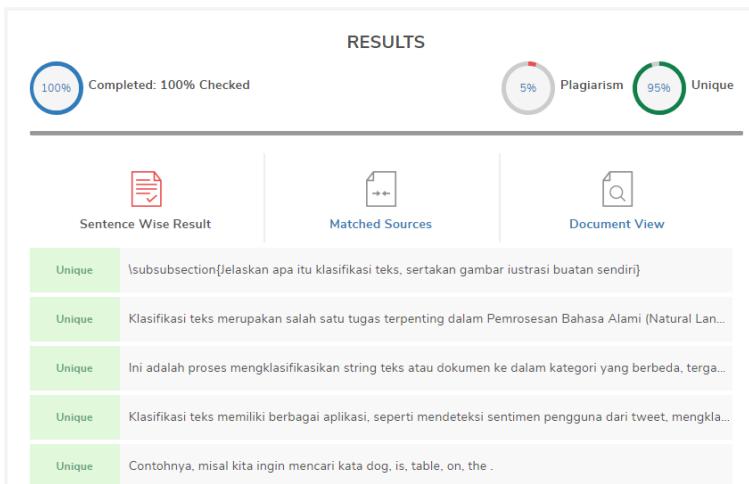
```
Max features: 25, num estimators: 150, accuracy: 0.93 (+/- 0.04)
Max features: 25, num estimators: 170, accuracy: 0.94 (+/- 0.03)
Max features: 25, num estimators: 190, accuracy: 0.94 (+/- 0.03)
Max features: 30, num estimators: 10, accuracy: 0.93 (+/- 0.06)
Max features: 30, num estimators: 30, accuracy: 0.93 (+/- 0.05)
Max features: 30, num estimators: 50, accuracy: 0.95 (+/- 0.04)
Max features: 30, num estimators: 70, accuracy: 0.93 (+/- 0.04)
Max features: 30, num estimators: 90, accuracy: 0.94 (+/- 0.03)
Max features: 30, num estimators: 110, accuracy: 0.94 (+/- 0.03)
Max features: 30, num estimators: 130, accuracy: 0.94 (+/- 0.03)
Max features: 30, num estimators: 150, accuracy: 0.94 (+/- 0.03)
Max features: 30, num estimators: 170, accuracy: 0.93 (+/- 0.04)
Max features: 30, num estimators: 190, accuracy: 0.93 (+/- 0.04)
Max features: 35, num estimators: 10, accuracy: 0.91 (+/- 0.05)
Max features: 35, num estimators: 30, accuracy: 0.94 (+/- 0.05)
Max features: 35, num estimators: 50, accuracy: 0.93 (+/- 0.04)
Max features: 35, num estimators: 70, accuracy: 0.94 (+/- 0.03)
Max features: 35, num estimators: 90, accuracy: 0.94 (+/- 0.03)
Max features: 35, num estimators: 110, accuracy: 0.93 (+/- 0.04)
Max features: 35, num estimators: 130, accuracy: 0.93 (+/- 0.04)
Max features: 35, num estimators: 150, accuracy: 0.94 (+/- 0.03)
Max features: 35, num estimators: 170, accuracy: 0.93 (+/- 0.04)
Max features: 35, num estimators: 190, accuracy: 0.93 (+/- 0.04)
Max features: 40, num estimators: 10, accuracy: 0.91 (+/- 0.06)
Max features: 40, num estimators: 30, accuracy: 0.93 (+/- 0.04)
Max features: 40, num estimators: 50, accuracy: 0.94 (+/- 0.03)
Max features: 40, num estimators: 70, accuracy: 0.94 (+/- 0.03)
Max features: 40, num estimators: 90, accuracy: 0.94 (+/- 0.03)
Max features: 40, num estimators: 110, accuracy: 0.93 (+/- 0.05)
Max features: 40, num estimators: 130, accuracy: 0.93 (+/- 0.04)
Max features: 40, num estimators: 150, accuracy: 0.93 (+/- 0.05)
Max features: 40, num estimators: 170, accuracy: 0.93 (+/- 0.04)
Max features: 40, num estimators: 190, accuracy: 0.93 (+/- 0.04)
Max features: 45, num estimators: 10, accuracy: 0.91 (+/- 0.06)
Max features: 45, num estimators: 30, accuracy: 0.94 (+/- 0.04)
Max features: 45, num estimators: 50, accuracy: 0.93 (+/- 0.04)
Max features: 45, num estimators: 70, accuracy: 0.93 (+/- 0.04)
Max features: 45, num estimators: 90, accuracy: 0.94 (+/- 0.03)
Max features: 45, num estimators: 110, accuracy: 0.93 (+/- 0.04)
Max features: 45, num estimators: 130, accuracy: 0.93 (+/- 0.04)
Max features: 45, num estimators: 150, accuracy: 0.93 (+/- 0.04)
Max features: 45, num estimators: 170, accuracy: 0.94 (+/- 0.03)
Max features: 45, num estimators: 190, accuracy: 0.93 (+/- 0.04)
```

**Gambar 2.65** hasil praktek soal no. 8(2)

### 2.5.3 Penanganan Error

pada chapter 4 ini saya tidak menemukan error.

## 2.5.4 Bukti Tidak Plagiat



**Gambar 2.66** Bukti tidak plagiat

## 2.5.5 Link Youtube

<https://youtu.be/la8Jptbm3Mo>

## 2.6 Muhammad Reza Syachrani - 1174084

### 2.6.1 Teori

1. Jelaskan apa itu klasifikasi teks, dan gambar ilustrasi  
klasifikasi teks adalah cara untuk mengklasifikasikan atau mengelompokkan teks berdasarkan parameter tertentu baik itu jenis teks atau jenis dari dokumen yang terdapat kumpulan teks didalamnya.

**Gambar 2.67** contoh klasifikasi teks

2. Jelaskan mengapa klasifikasi pada bunga tidak bisa menggunakan machine learning, sertakan ilustrasi.

karna melakukan klasifikasi pada bunga tidak dapat menggunakan mesin learning karena terdapat banyak jenis-jenis bunga yang mirip hingga ada yang sama persis tetapi tidak sama. oleh karena itu klasifikasi bunga tida bisa dilakukan oleh mesin learning dikarenakan jika inputan ciri-ciri tidak sesuai maka bunga di inputkan kemungkinan jawaban dari mesin learning itu tidak tepat contoh menginputkan ciri-ciri bunga serupa tetapi pada mesin learning menjawab itu merupakan ciri-ciri bunga teratai.

**Gambar 2.68** contoh klasifikasi bunga

3. Jelaskan bagaimana teknik pembelajaran mesin pada teks pada kata-kata yang digunakan di youtube.

cara pembelajaran teks yang digunakan pada youtube yaitu dengan cara menyimpan inputan teks pada menu pencarian youtube. sehingga pada saat kita akan mencari data yang serupa maka youtube menyediakan rekomendasi-rekomendasi dari pencaharian. contoh pada menu pencarian kita menulis kata sa maka akan muncul banyak rekomendasi yang serupa yang sering di cari oleh orang dalam jangka waktu tertentu.

sa

satu hati sampai mati  
samudra cinta  
sara wijayanto rumah ruben onsu  
sara wijaya  
samudra cinta 22 maret 2020  
saaih halilintar  
sambel trasie happy asmara  
samudra cinta 21 maret 2020  
sapi  
sampek tuwek lirik  
satu nama tetap dihati  
sarah sheila kamalia  
sakit pinggang  
sakit dalam bercinta ipank lirik

Laporan prediksi penulisiran

**Gambar 2.69** contoh teknik pembelajaran mesin

4. Jelaskan apa yang dimaksud vektorisasi data

vektorisasi data merupakan pembagian data menjadi bagian-bagian yang lebih sederhana, contoh pada satu paragraf terdiri dari 200 kata kemudian dilakukan vektorisasi dengan cara membagi-bagi kata dalam paragraf tersebut ke dalam kalimat-kalimat yang terpisah kemudian dipecah lagi menjadi data dalam perkata selanjutnya kata-kata tersebut dijemahkan.

5. Jelaskan apa itu bag of words dan ilustrasi

bag of words merupakan representasi teks yang menggambarkan kemunculan kata-kata dalam dokumen. ePngelompokan kata-kata kedalam perhitungan, berapakah sebuah kata muncul dalam satu kalimat. Disebut "tas" kata-kata, karena informasi tentang susunan atau struktur kata dalam dokumen dibuang. Model ini hanya berkaitan dengan apakah kata-kata yang diketahui muncul dalam dokumen, bukan di mana dalam dokumen. Contohnya disini akan melihat kemunculan kata dari kalimat :

- Ariq suka menonton film. Alvan juga suka film.

- Alvan juga suka menonton anime.

|      | Arig | suka | menonton | film | Alvan | juga | anime |
|------|------|------|----------|------|-------|------|-------|
| Doc1 | 1    | 2    | 1        | 2    | 1     | 1    |       |
| Doc2 |      | 1    | 1        |      | 1     | 1    | 1     |

**Gambar 2.70** contoh bag of words

#### 6. Jelaskan apa itu TF-IDF.

TF-IDF memberi frekuensi kata dalam setiap dokumen dalam mengganti data jadi number. Ini adalah rasio berapa kali kata itu muncul dalam dokumen dibandingkan dengan jumlah total kata dalam dokumen. Itu meningkat sejalan dengan jumlah kemunculan kata itu di dalam dokumen meningkat. Setiap dokumen memiliki tf sendiri. rumus TF-IDF :

$$W_{t,d} = TF_{t,d} * IDF_t \quad (1)$$

Keterangan :

$W_{t,d}$  = bobot dari  $t$  (*term*) dalam satu dokumen

$TF_{t,d}$  = frekuensi kemunculan  $t$  (*term*) dalam dokumen  $d$

$IDF_t$  = *Inverse document frequency*, dimana

$$IDF_t = \log\left(\frac{N}{n_t}\right) \quad (2)$$

Keterangan :

$N$  = jumlah semua dokumen

$n_t$  = jumlah dokumen yang mengandung *term t*

**Gambar 2.71** rumus Tf-IDF

#### 2.6.2 Praktek

##### 1. No. 1

```

1 # In [1]:
2 import pandas as pd #mengimport librari padas
3 us = pd.read_csv("D:/Git Kecerdasan Buatan/KB3C/src
        /1174084/4/us -500.csv") # variabel us untuk membaca file
        csv menggunakan fungsi read csv dari padas
4 print(len(us)) #melihat jumlah dari baris data yang telah di
        import
5 print(us.head()) #melihat lima baris pertama data yang telah
        di import
6 print(us.shape) #mengetahui banyak baris dan kolom dari data

```

```

500
   first_name ... web
0    James ... http://www.bentonjohnbjr.com
1 Josephine ... http://www.chanayjeffreyaesq.com
2      Art ... http://www.chemeljameslcpa.com
3     Lenna ... http://www.feltzprintingservice.com
4   Donette ... http://www.printingdimensions.com

[5 rows x 12 columns]
(500, 12)

```

**Gambar 2.72** aplikasi sederhana pandas

## 2. No. 2

```

1 # In [5]:
2 data_training = us[:450] #membuat data training sebanyak 450
  baris
3 data_testing = us[450:] #membuat data testing dari hasil
  pengurangan 500-450

```

|               |           |           |   |
|---------------|-----------|-----------|---|
| data_testing  | DataFrame | (50, 12)  | Column names: first_name, last_name, company_name,<br>address, city, coun ... |
| data_training | DataFrame | (450, 12) | Column names: first_name, last_name, company_name,<br>address, city, coun ... |

**Gambar 2.73** pecahan dataframe

## 3. No. 3

```

1 # In [1]:
2 import pandas as pd #Mengimport pandas
3 d=pd.read_csv("D:/Git Kecerdasan Buatan/KB3C/src/1174084/4/
  Youtube02-KatyPerry.csv") #Membuat variable d untuk
  membaca file csv dari dataset
4 # In [2]:
5 spam=d.query('CLASS == 1') #mengelompokkan komentar spam
6 nospam=d.query('CLASS == 0') #mengelompokkan komentar bukan
  spam
7 # In [3]: memanggil lib vektorisasi
8 #melakukan fungsi bag of word dengan cara menghitung semua
  kata
9 #yang terdapat dalam file
10 from sklearn.feature_extraction.text import CountVectorizer
11 vectorizer = CountVectorizer()
12 # In [3]:
13 dvec = vectorizer.fit_transform(d['CONTENT']) #melakukan bag
  of word pada dataframe pada colom CONTENT
14 # In [4]:
15 dvec #melihat isi vektorisasi
16 # In [5]:
17 print(d['CONTENT'][342]) #melihat isi data pada baris ke 342
18 # In [6]:

```

```
19 daptarkata=vectorizer.get_feature_names() #feature_names  
    merupakan digunakan untuk mengambil nama kolomnya ada apa  
    saja  
20 # In [7]:  
21 dshuf = d.sample(frac=1) #melakukan randomisasi pada datanya  
    supaya sempurna saat melakukan klasifikasi  
22 # In [8]:  
23 dk_train=dshuf[:300] #Data akan dibagi dari 300 row akhir  
    menjadi data training dan sisanya adalah data testing  
24 dk_test=dshuf[300:] #Data akan dibagi dari 300 row pertama  
    menjadi data training dan sisanya adalah data testing  
25 # In [9]:  
26 dk_train_att=vectorizer.fit_transform(dk_train['CONTENT']) #  
    melakukan training pada data training dan di vektorisasi  
27 print(dk_train_att)  
28 # In [10]:  
29 dk_test_att=vectorizer.transform(dk_test['CONTENT']) #  
    melakukan testing pada data testing dan di vektorisasi  
30 print(dk_test_att)  
31 # In [11]:  
32 dk_train_label=dk_train['CLASS'] #mengambil label spam dan  
    bukan spam  
33 print(dk_train_label)  
34 dk_test_label=dk_test['CLASS'] #mengambil label spam dan  
    bukan spam  
35 print(dk_test_label)
```

|           |   |
|-----------|---|
| (0, 336)  | 1 |
| (0, 242)  | 1 |
| (0, 1561) | 1 |
| (0, 733)  | 1 |
| (0, 1029) | 1 |
| (0, 1178) | 1 |
| (0, 1078) | 1 |
| (0, 811)  | 1 |
| (1, 495)  | 1 |
| (1, 1318) | 1 |
| (1, 816)  | 1 |
| (1, 74)   | 1 |
| (1, 479)  | 1 |
| (1, 111)  | 1 |
| (1, 113)  | 1 |
| (1, 1256) | 1 |
| (1, 425)  | 1 |
| (1, 1285) | 1 |
| (1, 180)  | 6 |
| (1, 741)  | 1 |
| (1, 399)  | 2 |
| (1, 241)  | 2 |

Gambar 2.74 vektorisasi dan klasifikasi

## 4. No. 4

```

1 # In[12]:
2 from sklearn import svm #Mengimport svm
3 clfsvm = svm.SVC() #clfsvm sebagai variabel untuk mengatur
4         fungsi SVC
5 clfsvm.fit(dk_train_att, dk_train_label) #Mengatur data
6         training
7 clfsvm.predict(dk_test_att)
8 clfsvm.score(dk_test_att, dk_test_label) #Mengatur data
9         testing

```

```

In [62]: clfsvm.predict(dk_test_att)
Out[62]:
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0], dtype=int64)

In [63]: clfsvm.score(dk_test_att, dk_test_label) #Mengatur data testing
Out[63]: 0.5

```

**Gambar 2.75** klasifikasi SVM

## 5. No. 5

```

1 # In[13]:
2 from sklearn import tree #Mengimport tree
3 clftree = tree.DecisionTreeClassifier() #clftree sebagai
4         variabel untuk decision tree
5 clftree.fit(dk_train_att, dk_train_label) #Mengatur data
6         training
7 # In[14]:
8 clftree.predict(dk_test_att)
9 # In[15]:
10 clftree.score(dk_test_att, dk_test_label) #Mengatur data
11         testing

```

```

In [60]: clftree.predict(dk_test_att)
Out[60]:
array([0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 1, 1, 1], dtype=int64)

In [61]: clftree.score(dk_test_att, dk_test_label) #Mengatur data testing
Out[61]: 0.92

```

**Gambar 2.76** klasifikasi Decission Tree

## 6. No. 6

```

1 # In[16]:
2 from sklearn.metrics import confusion_matrix #Mengimport
3         Confusion Matrix

```

```

3 pred_labels = clf.predict(dk_test_att) #Membuat variable
    pred_labels dari data testing
4 cm = confusion_matrix(dk_test_label, pred_labels) #cm sebagai
    variabel data label
5 cm

```

**Out[65]:**

```
array([[24,  0],
       [ 3, 23]], dtype=int64)
```

**Gambar 2.77** confusion matrix

## 7. No. 7

```

1 # In[17]:
2 from sklearn.model_selection import cross_val_score #
    Mengimport cross_val_score
3 scores = cross_val_score(clf,dk_train_att,dk_train_label,cv
    =5) #Membuat variable scores sebagai variabel prediksi
        dari data training
4 scorerata2=scores.mean()
5 scorersd=scores.std()
6 # In[18]
7 print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.
    std() * 2)) #menampilkan data scores dengan ketentuan
        akurasi

```

```

In [66]: from sklearn.model_selection import cross_val_score #Mengimport
cross_val_score
...: scores = cross_val_score(clf,dk_train_att,dk_train_label,cv=5) #Membuat
variable scores sebagai variabel prediksi dari data training
...: scorerata2=scores.mean()
...: scorersd=scores.std()

In [67]: print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
#menampilkan data scores dengan ketentuan akurasi
Accuracy: 0.95 (+/- 0.06)

```

**Gambar 2.78** cross validaiton

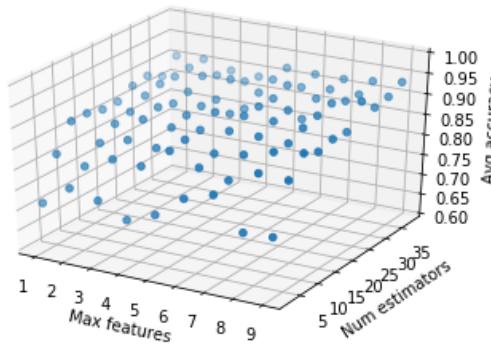
## 8. No. 8

```

1
2 # In[19]:
3 import numpy as np
4 max_features_opts = range(1, 10, 1) #Variable
    max_features_opts sebagai variabel untuk membuat range 1,
        10, 1

```

```
5 n_estimators_opts = range(2, 40, 4) #
   Variabel n_estimators_opts sebagai variabel untuk membuat
   range 2, 40, 4
6 rf_params = np.empty((len(max_features_opts)*len(
   n_estimators_opts),4) , float) #Variable rf_params sebagai
   variabel untuk menjumlahkan yang sudah di tentukan
   sebelumnya
7 i = 0
8 for max_features in max_features_opts: #Perulangan
9     for n_estimators in n_estimators_opts: #Perulangan
10        clf = RandomForestClassifier(max_features=
   max_features, n_estimators=n_estimators) #Menampilkan
   variabel csf
11        scores = cross_val_score(clf , dk_train_att ,
   dk_train_label , cv=5) #Variable scores sebagai variabel
   training
12        rf_params[i,0] = max_features #index 0
13        rf_params[i,1] = n_estimators #index 1
14        rf_params[i,2] = scores.mean() #index 2
15        rf_params[i,3] = scores.std() * 2 #index 3
16        i += 1 #Dengan ketentuan i += 1
17        print("Max features: %d, num estimators: %d, accuracy
   : %0.2f (+/- %0.2f)"%
18 % (max_features , n_estimators , scores.mean() , scores.std() *
   2)) #Print hasil pengulangan yang sudah ditentukan
19
20 # In [20]:
21 import matplotlib.pyplot as plt #Mengimport library
   matplotlib sebagai plt
22 from mpl_toolkits.mplot3d import Axes3D #Mengimport axes3D
   untuk menampilkan plot 3 dimensi
23 from matplotlib import cm #Memanggil data cm yang sudah
   tersedia
24 fig = plt.figure() #Menghasilkan plot sebagai figure
25 fig.clf() #Figure di ambil dari clf
26 ax = fig.gca(projection='3d') #ax sebagai projection 3d
27 x = rf_params[:,0] #x sebagai index 0
28 y = rf_params[:,1] #y sebagai index 1
29 z = rf_params[:,2] #z sebagai index 2
30 ax.scatter(x, y, z) #Membuat plot scatter x y z
31 ax.set_zlim(0.6, 1) #Set zlim dengan ketentuan yang ada
32 ax.set_xlabel('Max features') #Memberikan nama label x
33 ax.set_ylabel('Num estimators') #Memberikan nama label y
34 ax.set_zlabel('Avg accuracy') #Memberikan nama label z
```



**Gambar 2.79** pengamatan komponen informasi

### 2.6.3 Penanganan Error

#### 1. Error

```
FileNotFoundException: [Errno 2] File b'D:/Git Kecerdasan Buatan/KB3C/src/11174084/4/Youtube02-KatyPerry.csv' does not exist: b'D:/Git Kecerdasan Buatan/KB3C/src/11174084/4/Youtube02-KatyPerry.csv'
```

**Gambar 2.80** File Not Found Error

#### 2. Tuliskan Kode Error dan Jenis Error

- FileNotFoundException

#### 3. Cara Penanganan Error

- FileNotFoundException

Error terdapat pada kesalahan saat baca file csv, yang tidak terbaca. Dikarenakan letak file yang dibaca tidak pada direktori yang sama. Seharusnya letakkan file di direktori yang sama.

### 2.6.4 Bukti Tidak Plagiat



**Gambar 2.81** plagiarism

## 2.6.5 Link Video Youtube

[https://youtu.be/v63ayQTw\\_MI](https://youtu.be/v63ayQTw_MI)

## 2.7 1174077 - Alvan Alvanzah

### 2.7.1 Teori

1. Jelaskan apa itu klasifikasi teks, sertakan gambar ilustrasi buatan sendiri.

Klasifikasi teks merupakan sebuah model yang biasa digunakan untuk untuk mengkategorikan sebuah teks ke dalam kelompok-kelompok yang lebih terorganisir. Jadi untuk setiap kalimat yang di masukan ke dalam mesin, mesin tersebut akan menjadikan setiap kata dari kalimat tersebut menjadi sebuah kolom. Untuk ilustrasinya bisa dilihat pada gambar berikut :



**Gambar 2.82** Klasifikasi teks.

2. Jelaskan mengapa hal ini bisa terjadi, klasifikasi bunga tidak bisa digunakan untuk machine learning, sertakan ilustrasi gambar sendiri.

Karena machine learning tidak dapat menampilkan inputan sesuai dengan apa yang kita inputkan. Karena inputan tersebut serupa namun mesin memberikan output yang berbeda, biasanya output atau error ini disebut dengan istilah noise. Untuk contoh sederhananya misalkan kita inputkan salah satu label yang terdapat pada bunga, output yang dihasilkan oleh mesin tersebut ialah label yang lain. Itu dikarenakan bunga banyak jenis yang serupa namun tidak sama. Untuk ilustrasinya bisa dilihat pada gambar berikut :



**Gambar 2.83** Klasifikasi Bunga.

3. Jelaskan bagaimana yang dimaksud dengan teknik pembelajaran mesin pada teks yang digunakan dan sertakan ilustrasi buatan sendiri. Teknik yang digunakan pada youtube salah satunya ialah keywords. Dengan keywords tersebut mesin dapat memberikan video sesuai dengan keyword yang kita inputkan pada kolom pencarian. Teknik pembelajarannya

tergantung user memberikan input teks seperti apa, karena pada youtube itu sendiri akan menyesuaikan dengan apa yang biasa kita inputkan dan akan memfilter video secara otomatis sesuai dengan keyword yang biasa kita inputkan. Contoh ilustrasi sederhananya seperti berikut :



**Gambar 2.84** Klasifikasi teks Youtube.

4. Jelaskan apa yang dimaksud vektorisasi data.

Vektorisasi data ialah suatu pemecahan atau pembagian data berupa teks, sebagai contoh terdapat 5 paragraf, data teks tersebut di pecah menjadi kalimat-kalimat yang lebih sederhana, lalu di pecah lagi menjadi kata untuk setiap kalimatnya.

5. Jelaskan apa yang dimaksud dengan bag of words dengan ilustrasi sendiri.

Representasi penyederhanaan sebuah kalimat atau perhitungan setiap kata pada suatu kalimat dengan presentase berapa kali muncul kata tersebut untuk setiap kalimatnya. Contoh ilustrasi sederhananya seperti berikut :



**Gambar 2.85** Bag of words.

6. Jelaskan apa yang dimaksud dengan TF-IDF.

TF-IDF merupakan metode untuk menghitung bobot setiap kata pada suatu kalimat yang paling sering digunakan. TF-IDF ini akan menghitung nilai Term Frequency dan Inverse Document Frequency pada setiap kata dalam setiap kalimat yang muncul dengan diimbangi dengan jumlah dokumen dalam korpus yang mengandung kata. Contoh ilustrasi sederhananya seperti gambar berikut :

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

|           | Doc 1 | Doc 2 | ... | Doc n |
|-----------|-------|-------|-----|-------|
| Term(s) 1 | 12    | 2     | ... | 1     |
| Term(s) 2 | 0     | 1     | ... | 0     |
| ...       | ...   | ...   | ... | ...   |
| Term(s) n | 0     | 6     | ... | 3     |

Gambar 2.86 TF-IDF.

### 2.7.2 Praktek Program

#### 1. Soal 1

```

1 %% Soal 1
2 import pandas as pd #digunakan untuk mengimport library
#pandas dengan alias pd
3 pd = pd.read_csv("C:/Users/ASUS/Videos/zuplur/src/1174077/4/
alvan.csv") #membaca file csv

```

Kode di atas digunakan untuk buat aplikasi sederhana menggunakan pandas dengan format csv sebanyak 500 baris, hasilnya ialah sebagai berikut :



Gambar 2.87 Hasil Soal 1.

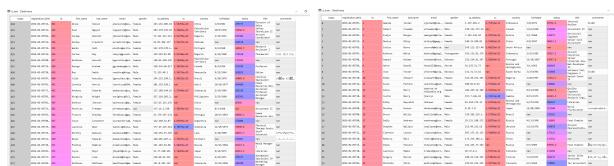
#### 2. Soal 2

```

1 %% Soal 2
2 d_train=pd[:450] #membagi data training menjadi 450
3 d_test=pd[450:] #membagi data menjadi 50 atau sisanya dari data
#yang tersedia

```

Kode di atas digunakan untuk memecah dataframe tersebut menjadi dua bagian yaitu 450 row pertama dan 50 row kedua. Hasilnya adalah sebagai berikut :



Gambar 2.88 Hasil Soal 2.

## 3. Soal 3

```

1 %% Soal 3
2 import pandas as ps #untuk import library pandas berguna
    untuk mengelola dataframe
3 ps = ps.read_csv("C:/Users/ASUS/Videos/zuplur/src/1174077/4/
    Youtube03-LMFAO.csv") #membaca file dengan format csv
4
5 spam=ps.query( 'CLASS == 1') #membagi tabel spam
6 nospam=ps.query( 'CLASS == 0')#membagi tabel no spam
7
8 from sklearn.feature_extraction.text import CountVectorizer #
    untuk import countvectorizer berfungsi untuk memecah data
    tersebut menjadi sebuah kata yang lebih sederhana
9 vectorizer = CountVectorizer () #ntuk menjalankan fungsi
    tersebut, pada code ini tidak ada hasilnya dikarenakan
    spyder tidak mendukung hasil dari instasiasi.
10
11 dvec = vectorizer.fit_transform(ps[ 'CONTENT']) #untuk
    melakukan pemecahan data pada dataframe yang terdapat pada
    kolom konten
12 dvec #Untuk menampilkan hasil dari code sebelumnya
13
14 Daptarkata= vectorizer.get_feature_names()
15
16 dshuf = ps.sample(frac=1)
17
18 d_train=dshuf[:300]
19 d_test=dshuf[300:]
20
21 d_train_att = vectorizer.fit_transform(d_train[ 'CONTENT'])
22 d_train_att
23
24 d_train_label=d_train[ 'CLASS']
25 d_test_label=d_test[ 'CLASS']
```

Dengan menggunakan  $1174077 \bmod 4$  adalah 1, yang artinya menggunakan dataset LMFAO. Hasilnya adalah sebagai berikut :



```
In [8]: from sklearn import tree
...: clftree = tree.DecisionTreeClassifier()
...: clftree.fit(d_train_att, d_train_label)
Out[8]:
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
criterion='gini',
max_depth=None, max_features=None,
max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0,
presort='deprecated', random_state=None, splitter='best')
```

**Gambar 2.91** Hasil Soal 5.

## 6. Soal 6

```
1 #%%soal 6/
2
3 from sklearn.metrics import confusion_matrix
4 pred_labels=clftree.predict(d_test)
5 cm=confusion_matrix(d_test_label ,pred_labels)
6
7 #%%
8
9 import matplotlib.pyplot as plt
10
11 def plot_confusion_matrix(cm, classes,
12                           normalize=False,
13                           title='Confusion matrix',
14                           cmap=plt.cm.Blues):
15
16     if normalize:
17         cm = cm.astype('float') / cm.sum(axis=1)[:, np.
newaxis]
18         print("Normalized confusion matrix")
19     else:
20         print('Confusion matrix, without normalization')
21
22     print(cm)
```

Menjalankan program cross validation. Hasilnya adalah sebagai berikut :

```
In [11]: import matplotlib.pyplot as plt
...:
...: def plot_confusion_matrix(cm, classes,
...:                         normalize=False,
...:                         title='Confusion matrix',
...:                         cmap=plt.cm.Blues):
...:
...:     if normalize:
...:         cm = cm.astype('float') / cm.sum(axis=1)[:, np.
newaxis]
...:         print("Normalized confusion matrix")
...:     else:
...:         print('Confusion matrix, without normalization')
...:
...:     print(cm)
...:     cm
```

**Gambar 2.92** Hasil Soal 6.

## 7. Soal 7

```
1 #%%soal 7
2
```

```

3 from sklearn.model_selection import cross_val_score
4
5 scores=cross_val_score(clftree,d_train_att,d_train_label, cv
6 =5)
7
8 skor_rata2=scores.mean()
9 skoresd=scores.std()

```

Tree.export graphviz merupakan fungsi yang menghasilkan representasi Graphviz dari decision tree, yang kemudian ditulis ke outfile. Disini akan menyimpan classifiernya, akan meng ekspor file student performance jika salah akan mengembalikan nilai fail. Hasilnya adalah sebagai berikut :

```

In [12]: from sklearn.model_selection import cross_val_score
...:
scores=cross_val_score(clftree,d_train_att,d_train_label, cv=5)
...:
skor_rata2=scores.mean()
...:
skoresd=scores.std()

```

**Gambar 2.93** Hasil Soal 7.

## 8. Soal 8

```

1 %%soal 8 /
2
3 max_features_opts = range(5, 50, 5) #max_features_opts
4     sebagai variabel untuk membuat range 5,50,5
5 n_estimators_opts = range(10, 200, 20) #n_estimators_opts
6     sebagai variabel untuk membuat range 10,200,20
7 rf_params = ps.empty((len(max_features_opts)*len(
8     n_estimators_opts),4), float) #rf_params sebagai variabel
9     untuk menjumlahkan yang sudah di tentukan sebelumnya
10 i = 0
11 for max_features in max_features_opts: #pengulangan
12     for n_estimators in n_estimators_opts: #pengulangan
13         clftree = RandomForestClassifier(max_features=
14             max_features, n_estimators=n_estimators) #menampilkan
15             variabel csf
16             scores = cross_val_score(clf, df_train_att,
17             df_train_label, cv=5) #scores sebagai variabel training
18             rf_params[i,0] = max_features #index 0
19             rf_params[i,1] = n_estimators #index 1
20             rf_params[i,2] = scores.mean() #index 2
21             rf_params[i,3] = scores.std() * 2 #index 3
22             i += 1 #dengan ketentuan i += 1
23             print("Max features: %d, num estimators: %d, accuracy
24 : %0.2f (+/- %0.2f)" %(max_features, n_estimators, scores.
25 mean(), scores.std() * 2))
26             #print hasil pengulangan yang sudah ditentukan

```

Buatlah program pengamatan komponen informasi. Jadi disini kita akan memprediksi nilai dari variabel test att dan test pass Hasilnya adalah sebagai berikut :

```
-- Max features: 25, num estimators: 100, accuracy: 0.46 (+/- 0.03)
Max features: 30, num estimators: 100, accuracy: 0.33 (+/- 0.02)
Max features: 35, num estimators: 100, accuracy: 0.43 (+/- 0.03)
Max features: 30, num estimators: 70, accuracy: 0.44 (+/- 0.03)
Max features: 35, num estimators: 70, accuracy: 0.44 (+/- 0.03)
Max features: 30, num estimators: 130, accuracy: 0.45 (+/- 0.03)
Max features: 30, num estimators: 130, accuracy: 0.46 (+/- 0.04)
Max features: 35, num estimators: 130, accuracy: 0.46 (+/- 0.04)
Max features: 30, num estimators: 170, accuracy: 0.46 (+/- 0.03)
Max features: 35, num estimators: 170, accuracy: 0.46 (+/- 0.03)
Max features: 30, num estimators: 50, accuracy: 0.34 (+/- 0.03)
Max features: 35, num estimators: 50, accuracy: 0.44 (+/- 0.03)
Max features: 30, num estimators: 80, accuracy: 0.44 (+/- 0.03)
Max features: 35, num estimators: 80, accuracy: 0.44 (+/- 0.03)
Max features: 30, num estimators: 110, accuracy: 0.46 (+/- 0.04)
Max features: 35, num estimators: 110, accuracy: 0.46 (+/- 0.04)
Max features: 30, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 35, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 30, num estimators: 170, accuracy: 0.46 (+/- 0.04)
Max features: 35, num estimators: 170, accuracy: 0.46 (+/- 0.04)
Max features: 30, num estimators: 100, accuracy: 0.34 (+/- 0.02)
Max features: 35, num estimators: 100, accuracy: 0.43 (+/- 0.03)
Max features: 30, num estimators: 50, accuracy: 0.34 (+/- 0.03)
Max features: 35, num estimators: 50, accuracy: 0.44 (+/- 0.03)
Max features: 30, num estimators: 80, accuracy: 0.44 (+/- 0.03)
Max features: 35, num estimators: 80, accuracy: 0.44 (+/- 0.03)
Max features: 30, num estimators: 110, accuracy: 0.45 (+/- 0.04)
Max features: 35, num estimators: 110, accuracy: 0.45 (+/- 0.04)
Max features: 30, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 35, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 30, num estimators: 170, accuracy: 0.45 (+/- 0.04)
Max features: 35, num estimators: 170, accuracy: 0.45 (+/- 0.04)
Max features: 30, num estimators: 100, accuracy: 0.34 (+/- 0.02)
Max features: 35, num estimators: 100, accuracy: 0.43 (+/- 0.03)
Max features: 30, num estimators: 50, accuracy: 0.34 (+/- 0.03)
Max features: 35, num estimators: 50, accuracy: 0.44 (+/- 0.03)
Max features: 30, num estimators: 80, accuracy: 0.44 (+/- 0.03)
Max features: 35, num estimators: 80, accuracy: 0.44 (+/- 0.03)
Max features: 30, num estimators: 110, accuracy: 0.45 (+/- 0.04)
Max features: 35, num estimators: 110, accuracy: 0.45 (+/- 0.04)
Max features: 30, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 35, num estimators: 150, accuracy: 0.45 (+/- 0.03)
Max features: 30, num estimators: 170, accuracy: 0.45 (+/- 0.04)
Max features: 35, num estimators: 170, accuracy: 0.45 (+/- 0.04)
```

Gambar 2.94 Hasil Soal 8.

### 2.7.3 Penanganan Error

#### 1. ScreenShoot Error

```
FileNotFoundError: [Errno 2] File 'E:\Users\Alv\Documents\tutorial\src\1174077\4\alva.csv' does
not exist: 'E:\Users\Alv\Documents\tutorial\src\1174077\4\alva.csv'
```

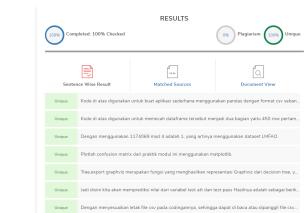
Gambar 2.95 SyntaxError

#### 2. Cara Penanganan Error

- SyntaxError

Dengan menyesuaikan letak file csv pada codingannya, sehingga dapat di baca atau dipanggil file csvnya.

### 2.7.4 Bukti Tidak Plagiat



Gambar 2.96 Bukti Tidak Melakukan Plagiat Chapter 4

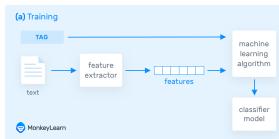
## 2.7.5 Link Youtube

### 2.8 1174079 -Chandra Kirana Poetra

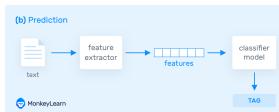
#### 2.8.1 Teori

1. Jelaskan apa itu klasifikasi teks, sertakan gambar ilustrasi buatan sendiri.

Klasifikasi teks merupakan suatu proses dalam memberikan tag atau kategori pada suatu teks berdasarkan konten. Klasifikasi teks merupakan salah satu bagian dari natural language processing yang mencakup analisis, labeling, deteksi spam dan intent detection. Untuk ilustrasinya bisa dilihat pada gambar berikut :



**Gambar 2.97** Klasifikasi teks.



**Gambar 2.98** Klasifikasi teks.

2. Jelaskan mengapa hal ini bisa terjadi, klasifikasi bunga tidak bisa digunakan untuk machine learning, sertakan ilustrasi gambar sendiri. Karena meskipun bunga yang kita coba untuk klasifikasikan memiliki spesies yang sama, kebanyakan bunga sendiri memiliki bentuk ukuran yang berbeda beda sehingga akan membuat machine learning sulit diterapkan karena ukuran tidak bisa dijadikan sebagai salah satu parameter dalam machine learning.



**Gambar 2.99** Ukuran Bunga yang berbeda-beda.

3. Jelaskan bagaimana yang dimaksud dengan teknik pembelajaran mesin pada teks yang digunakan dan sertakan ilustrasi buatan sendiri.  
Youtube menggunakan istilah yang disebut sebagai keyword yang nantinya kita isi didalam form pencarian yang ada kemudian akan muncul video yang terkait



**Gambar 2.100** Klasifikasi teks Youtube.

4. Jelaskan apa yang dimaksud vektorisasi data.  
vektorisasi merupakan suatu proses untuk mengkonversi suatu algoritma yang beroperasi pada satu value dalam satu waktu menjadi kebeberapa value dalam satu waktu
5. Jelaskan apa yang dimaksud dengan bag of words dengan ilustrasi sendiri.

Merupakan model untuk menyederhanakan data dalam bidang natural language processing dan juga pengambilan informasi. di model ini text direpresentasikan sebagai suatu kantong kata, mengabaikan grammer dan susunan kata tapi menyimpan value pengulangan apabila ada kata yang diulang. model ini digunakan pada bidang komputer



**Gambar 2.101** Bag of words.

6. Jelaskan apa yang dimaksud dengan TF-IDF.

TF-IDF merupakan singkatan dari term frequency inverse document frequency yang merupakan statistik angka yang memiliki tujuan untuk memperlihatkan seberapa pentingnya suatu kata dalam suatu dokumen.



Gambar 2.102 TF-IDF.

## 2.8.2 Praktek Program

### 2.8.2.1 Nomor 1

```
1 %% Soal 1
2 import pandas as pd #digunakan untuk mengimport library pandas
3 dengan alias pd
3 pd = pd.read_csv("F:/ Poltekpos/D4 TI 3C/Semester 6/Kecerdasan
4 Buatan/Github/Upload 30 Maret 2020 github tugas 4/src
5 /1174079/4/csv_chandra.csv") #membaca file csv
```

```
In [32]: import pandas as pd #digunakan untuk mengimport
library pandas dengan alias pd
... pd = pd.read_csv("F:/Poltekpos/D4 TI 3C/Semester 6/
Kecerdasan/Buatan/Github/Upload 30 Maret 2020 github tugas 4/
src/1174079/4/csv_chandra.csv") #membaca file csv
```

Gambar 2.103 Nomor 1

### 2.8.2.2 Nomor 2

```
1 %% Soal 2
2 d_train=pd[:450] #membagi data training menjadi 450
3 d_test=pd[450:] #membagi data menjadi 50 atau sisa dari data yang
tersedia
```

```
In [33]: d_train=pd[:450] #membagi data training menjadi 450
... d_test=pd[450:] #membagi data menjadi 50 atau sisa
dari data yang tersedia
```

Gambar 2.104 Nomor 2

### 2.8.2.3 Nomor 3

```
1 %% Soal 3
2 import pandas as pd #digunakan untuk mengimport library pandas
dengan alias pd
```

```

3 d = pd.read_csv("F:/ Poltekpos/D4 TI 3C/Semester 6/Kecerdasan
4 Buatan/Github/Upload 30 Maret 2020 github tugas 4/src
5 /1174079/4/Youtube03-LMFAO.csv") #Membaca file csv
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

```

```

rc(1174079/4/Youtube03-LMFAO.csv") #membaca file csv
.....
.... from sklearn.feature_extraction.text import
.... CountVectorizer #import fungsi countvectorize dari sklearn
.... vectorizer = CountVectorizer() #membuat instansi CountVectorizer
.... countVectorizer
.....
.... dvec = vectorizer.fit_transform(d['CONTENT']) #Memasukkan data ke
.... dvec
.... dvec #Melihat data yang dimasukkan ke dvec
.....
.... daptarkata = vectorizer.get_feature_names() #Mendapatkan data dan
.... memasukkannya ke daptarkata
.....
.... dshuf = d.sample(frac=1) #Memasukkan sample kedalam variable
.... dshuf
.....
.... d_train = dshuf[:300] #Membuat data training
.... d_test = dshuf[300:] #Membuat data test
.....
.... d_train_att = vectorizer.fit_transform(d_train['CONTENT']) #
.... Memasukkan data training dari vectorizer
.... d_train_att #Melihat data training
.....
.... d_test_att = vectorizer.transform(d_test['CONTENT']) #Memasukkan
.... data test dari vectorizer
.... d_test_att #Melihat data training
.....
.... d_train_label = d_train['CLASS'] #Memberi label
.... d_test_label = d_test['CLASS'] #Memberi Label

```

**Gambar 2.105 Nomor 3**

#### 2.8.2.4 Nomor 4

```

1 # SVM
2 from sklearn import svm #Mengimport svm dari sklearn
3 clfsvm = svm.SVC() #Membuat svc kedalam variable svm

```

```

4 clfsvm.fit(d_train_att, d_train_label) #Memprediksi data dari
   data training
5 clfsvm.score(d_test_att, d_test_label) #Memunculkan clf sebagai
   testing yang sudah di training tadi

```

```

In [38]: clfsvm.fit(d_train_att, d_train_label)
          C:\Users\acer\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change
              from 'auto' to 'scale' in version 0.22 to account better for
              unbalanced classes; specify gamma explicitly to 'auto' or 'scale'
              to avoid this warning.
              "avoid this warning.", FutureWarning)
Out[38]: 0.797104492753623

```

Gambar 2.106 Nomor 4

### 2.8.2.5 Nomor 5

```

1 # Decission Tree
2 from sklearn import tree #Mengimport tree dari sklearn
3 clftree = tree.DecisionTreeClassifier() #Membuat decision tree
4 clftree.fit(d_train_att, d_train_label) #Memprediksi data dari
   data training
5 clftree.score(d_test_att, d_test_label) #Memunculkan clf sebagai
   testing yang sudah di training tadi

```

```

In [49]: from sklearn import tree #Mengimport tree dari
          sklearn
          ...
          clftree = tree.DecisionTreeClassifier() #Membuat
          decision tree
          ...
          clftree.fit(d_train_att, d_train_label) #Memprediksi
          data dari data training
          ...
          clftree.score(d_test_att, d_test_label) #Memunculkan
          clf sebagai testing yang sudah di training tadi
Out[49]: 0.9492753623188406

```

Gambar 2.107 Nomor 5

### 2.8.2.6 Nomor 6

```

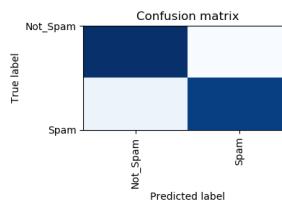
1 # Confusion Matrix – Decission Tree
2 from sklearn.metrics import confusion_matrix
3 pred_labeltree = clftree.predict(d_test_att)
4 cmtree = confusion_matrix(d_test_label, pred_labeltree)
5 cmtree
6
7 import matplotlib.pyplot as plt
8 import itertools
9 def plot_confusion_matrix(cm, classes,
                           normalize=False,
                           title='Confusion matrix',
                           cmap=plt.cm.Blues):
10
11     """
12     This function prints and plots the confusion matrix.
13     Normalization can be applied by setting 'normalize=True'.
14     """
15
16     if normalize:
17         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
18         print("Normalized confusion matrix")
19
20     else:

```

```

21     print('Confusion matrix, without normalization')
22
23     print(cm)
24
25     plt.imshow(cm, interpolation='nearest', cmap=cmap)
26     plt.title(title)
27     #plt.colorbar()
28     tick_marks = np.arange(len(classes))
29     plt.xticks(tick_marks, classes, rotation=90)
30     plt.yticks(tick_marks, classes)
31
32     fmt = '.2f' if normalize else 'd'
33     thresh = cm.max() / 2.
34     #for i, j in itertools.product(range(cm.shape[0]), range(cm.
35     #shape[1])):
36     #    plt.text(j, i, format(cm[i, j], fmt),
37     #              horizontalalignment="center",
38     #              color="white" if cm[i, j] > thresh else "black"
39     #)
40
41     plt.tight_layout()
42     plt.ylabel('True label')
43     plt.xlabel('Predicted label')
44
45 types = pd.read_csv("F:/Poltekpos/D4 TI 3C/Semester 6/Kecerdasan
46 Buatan/Github/Upload 30 Maret 2020 github tugas 4/src
47 /1174079/4/classes.txt", sep='|', header=None, usecols=[1],
48 names=['type'])
49 types = types['type']
50 types
51
52 import numpy as np
53 np.set_printoptions(precision=2)
54 plt.figure(figsize=(4,4), dpi=100)
55 plot_confusion_matrix(cmtree, classes=types, normalize=True)
56 plt.show()

```



**Gambar 2.108** Nomor 6

### 2.8.2.7 Nomor 7

```

1 # Cross Validation
2 from sklearn.model_selection import cross_val_score
3

```

```

4 scorestree = cross_val_score(clftree, d_train_att, d_train_label,
5     cv=5)
6 print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
7     scorestree.std() * 2))
8
9 scoressvm = cross_val_score(clfsvm, d_train_att, d_train_label,
10    cv=5)
11 print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(),
12     scoressvm.std() * 2))
13
14 scores = cross_val_score(clf, d_train_att, d_train_label, cv=5)
15 print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std()
16     () * 2))

```

```
C:\Users\acer\OneDrive\Udemy\stephanie\sklearn\svmbase.py:103: FutureWarning: The default value of gamma will change
from 'auto' to 'scale' in version 0.22 to account better for
unscaled features. Set gamma explicitly to 'auto' or 'scale'
to avoid this warning.
  "Avoid this warning.", FutureWarning)
Accuracy: 0.95 (+/- 0.00)
```

**Gambar 2.109** Nomor 7

### 2.8.2.8 Nomor 8

```

1 # Komponen Informasi
2 max_features_opts = range(5, 50, 5)
3 n_estimators_opts = range(10, 200, 20)
4 rf_params = np.empty((len(max_features_opts)*len(
5     n_estimators_opts),4), float)
5 i = 0
6 for max_features in max_features_opts:
7     for n_estimators in n_estimators_opts:
8         clf = RandomForestClassifier(max_features=max_features,
9             n_estimators=n_estimators)
10        scores = cross_val_score(clf, d_train_att, d_train_label,
11            cv=5)
12        rf_params[i,0] = max_features
13        rf_params[i,1] = n_estimators
14        rf_params[i,2] = scores.mean()
15        rf_params[i,3] = scores.std() * 2
16        i += 1
17        print("Max features: %d, num estimators: %d, accuracy:
18            %0.2f (+/- %0.2f)" % (max_features, n_estimators, scores.mean(
19            ), scores.std() * 2))      #print hasil pengulangan yang
20            sudah ditentukan

```

```

usage: [options] file [file ...]

Max features: 5, num estimators: 10, accuracy: 0.89 (+/- 0.05)
Max features: 5, num estimators: 30, accuracy: 0.92 (+/- 0.07)
Max features: 5, num estimators: 50, accuracy: 0.94 (+/- 0.04)
Max features: 5, num estimators: 70, accuracy: 0.94 (+/- 0.06)
Max features: 5, num estimators: 90, accuracy: 0.95 (+/- 0.06)
Max features: 5, num estimators: 110, accuracy: 0.95 (+/- 0.04)
Max features: 5, num estimators: 130, accuracy: 0.95 (+/- 0.04)
Max features: 5, num estimators: 150, accuracy: 0.94 (+/- 0.08)
Max features: 5, num estimators: 170, accuracy: 0.93 (+/- 0.05)
Max features: 5, num estimators: 190, accuracy: 0.94 (+/- 0.07)
Max features: 10, num estimators: 10, accuracy: 0.92 (+/- 0.03)
Max features: 10, num estimators: 30, accuracy: 0.95 (+/- 0.04)
Max features: 10, num estimators: 50, accuracy: 0.93 (+/- 0.07)
Max features: 10, num estimators: 70, accuracy: 0.93 (+/- 0.07)
Max features: 10, num estimators: 90, accuracy: 0.94 (+/- 0.06)
Max features: 10, num estimators: 110, accuracy: 0.93 (+/- 0.06)

```

**Gambar 2.110** Nomor 8

### 2.8.3 Penanganan Error

#### 1. ScreenShoot Error

```

FileNotFoundException: [Errno 2] File 'b'F:/Poltekps/04 TI 3C/
Semester 6/Kecerdasan Buata/Github/Upload 30 Naret 2020 github
tugas 4/src/1174079/4/csv_chandra.csv' does not exist: b'F:/
Poltekps/04 TI 3C/Semester 6/Kecerdasan Buata/Github/Upload
30 Naret 2020 github tugas 4/src/1174079/4/csv_chandra.csv'

```

**Gambar 2.111** SyntaxError

#### 2. Cara Penangan Error

- **SyntaxError**

Memperbaiki typo dan mengecek nama file serta direktoriinya

### 2.8.4 Bukti Tidak Plagiat

**Gambar 2.112** Bukti Tidak Melakukan Plagiat Chapter 4

### 2.8.5 Link Youtube

<https://youtu.be/8vkMpgUVTQs>



## BAB 3

---

# CHAPTER 5

---

### 3.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

```
1 @inproceedings{awangga2017colenak,
2   title={Colenak: GPS tracking model for post-stroke
3   rehabilitation program using AES-CBC URL encryption and QR-
4   Code},
5   author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
6   Hasanudin, Trisna Irmayadi},
7   booktitle={Information Technology, Information Systems and
8   Electrical Engineering (ICITISEE), 2017 2nd International
   conferences on},
9   pages={255--260},
10  year={2017},
11  organization={IEEE}
12 }
```



**Gambar 3.1** Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### 3.1.1 Teori

### 3.1.2 Praktek

### 3.1.3 Penanganan Error

### 3.1.4 Bukti Tidak Plagiat



**Gambar 3.2** Kecerdasan Buatan.

## BAB 4

---

# CHAPTER 6

---

### 4.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

```
1 @inproceedings{awangga2017colenak,
2   title={Colenak: GPS tracking model for post-stroke
3   rehabilitation program using AES-CBC URL encryption and QR-
4   Code},
5   author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
6   Hasanudin, Trisna Irmayadi},
7   booktitle={Information Technology, Information Systems and
8   Electrical Engineering (ICITISEE), 2017 2nd International
   conferences on},
9   pages={255--260},
10  year={2017},
11  organization={IEEE}
12 }
```



**Gambar 4.1** Kecerdasan Buatan.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

#### 4.1.1 Teori

#### 4.1.2 Praktek

#### 4.1.3 Penanganan Error

#### 4.1.4 Bukti Tidak Plagiat



**Gambar 4.2** Kecerdasan Buatan.

## BAB 5

---

# CHAPTER 7

---

### 5.1 1174006 - Kadek Diva Krishna Murti

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

```
1 @inproceedings{awangga2017colenak,
2   title={Colenak: GPS tracking model for post-stroke
3   rehabilitation program using AES-CBC URL encryption and QR-
4   Code},
5   author={Awangga, Rolly Maulana and Fathonah, Nuraini Siti and
6   Hasanudin, Trisna Irmayadi},
7   booktitle={Information Technology, Information Systems and
8   Electrical Engineering (ICITISEE), 2017 2nd International
   conferences on},
9   pages={255--260},
10  year={2017},
11  organization={IEEE}
12 }
```



**Gambar 5.1** Kecerdasan Buatan.

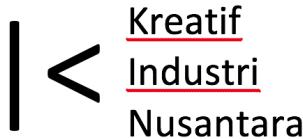
1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

#### **5.1.1 Teori**

#### **5.1.2 Praktek**

#### **5.1.3 Penanganan Error**

#### **5.1.4 Bukti Tidak Plagiat**



**Gambar 5.2** Kecerdasan Buatan.