


PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

Image

a. Percobaan Praktikum

a.	<p>Membaca dan menampilkan gambar Lena</p> <pre>img = cv2.imread("lena.png") #digunakan untuk membaca gambar pada path/folder gambar tersebut. plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB)) #digunakan untuk menampilkan gambar tersebut dengan format color RGB plt.title("Citra Asli") #memberikan judul pada output plt.axis("off") #untuk menghilangkan koordinat x dan y plt.show() # untuk menampilkan gambar</pre> <p>Citra Asli</p> 
b.	<p>Mencari ukuran dan matriks dari gambar Lena</p> <pre>print("Ukuran citra warna:", img.shape) #digunakan untuk menunjukan ukuran gambar(pixel) print("Matriks dari citra warna pada baris 0 dan kolom 0:", img[0,0]) #digunakan untuk menunjukan ukuran matriks pada gambar</pre> <p>Ukuran citra warna: (512, 512, 3) Matriks dari citra warna pada baris 0 dan kolom 0: [125 137 226]</p>
c.	<p>Memisahkan channel dan menampilkan warna yang dipilih</p> <pre># Memisahkan channel warna (B, G, R) = cv2.split(img) #memisahkan 3 warna primer Blue, Green, dan Red # Tampilkan channel Blue plt.imshow(B, cmap='gray') plt.title("Komponen Biru(grayscale)") plt.axis("off") plt.show() menampilkan gambar Lena dengan channel biru dalam bentuk grayscale</pre>

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

Komponen Biru(grayscale)



```
zeromatrik = np.zeros(img.shape[:2], img.dtype) #membuat
matriks kosong dengan ukuran yang sama dengan gambar Lena
m = zeromatrik
blueonly = cv2.merge([B, m, m])
plt.imshow(cv2.cvtColor(blueonly, cv2.COLOR_BGR2RGB))
plt.title("Channel Blue (warna)")
plt.axis("off")
plt.show()
```

menampilkan gambar jika seluruh matriks terisi dengan 0 dan hanya biru yang tetap

Channel Blue (warna)


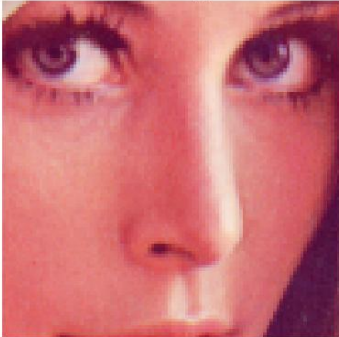


b. Tugas Praktikum

a.	<p>Modifikasi kode bagian (a) agar dapat menampilkan citra ‘Lenna.png’ dalam grayscale</p> <pre>gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #untuk setting warna pada gambar menjadi 'gray' plt.imshow(gray, cmap="gray") plt.title("Lenna - Grayscale") plt.axis("off") plt.show()</pre>
----	--




PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

	<p>Lenna - Grayscale</p> 
b.	<p>Modifikasi kode bagian (a) agar bisa melakukan crop pada citra ‘Lenna.png’</p> <pre>crop = img[250:350, 250:350] #untuk menentukan pada bagian mana yang akan ditampilkan(crop) plt.imshow(cv2.cvtColor(crop, cv2.COLOR_BGR2RGB)) plt.title("Lenna - Cropped") plt.axis("off") plt.show()</pre> <p>Lenna - Cropped</p> 
c.	<p>Modifikasi kode bagian (b) agar dapat menampilkan ukuran citra grayscale dari ‘Lenna.png’ dan nilai matriks dari citra grayscale ‘Lenna.pg’ pada baris ke-0 dan kolom ke-0. Apakah hasilnya berbeda dengan bagian (b), jelaskan alasannya.</p> <pre>print("Ukuran citra grayscale:", gray.shape) print("Nilai pixel grayscale (0,0):", gray[0, 0]) # Bandingkan dengan citra warna asli print("Ukuran citra warna:", img.shape) print("Nilai pixel warna (0,0):", img[0, 0]) # B, G, R</pre> <p>Ukuran citra grayscale: (512, 512) Nilai pixel grayscale (0,0): 162 Ukuran citra warna: (512, 512, 3) Nilai pixel warna (0,0): [125 137 226]</p>

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

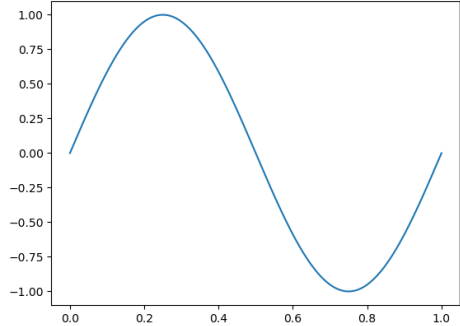
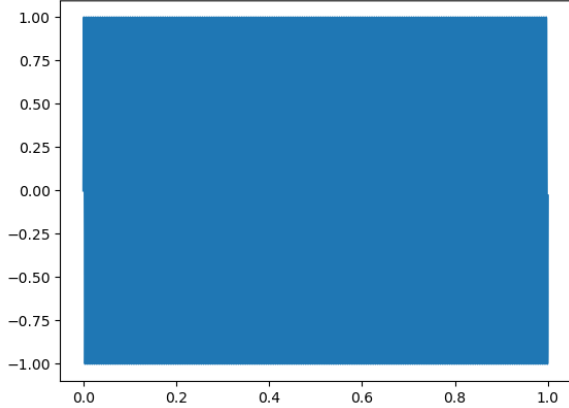
d.	<p>Modifikasi kode bagian (c) untuk menampilkan channel Green dan Red.</p> <pre>zeros = np.zeros(img.shape[:2], img.dtype) greenonly = cv2.merge([m, G, m]) #setting hanya warna hijau redonly = cv2.merge([m, m, R]) #setting hanya warna merah redgreen = cv2.merge([m, G, R]) #setting warna merah & hijau plt.subplot(1, 3, 1) # menampilkan 3 gambar dalam 1 baris dan gambar pertama plt.imshow(cv2.cvtColor(greenonly, cv2.COLOR_BGR2RGB)) plt.title("Green Channel") plt.axis("off") plt.subplot(1, 3, 2) # menampilkan gambar kedua plt.imshow(cv2.cvtColor(redonly, cv2.COLOR_BGR2RGB)) plt.title("Red Channel") plt.axis("off") plt.subplot(1, 3, 3) # menampilkan gambar ketiga plt.imshow(cv2.cvtColor(redgreen, cv2.COLOR_BGR2RGB)) plt.title("Red dan Green Channel") plt.axis("off") plt.show()</pre> <div><div>Green Channel</div><div>Red Channel</div><div>Red dan Green Channel</div></div> <div></div>
e.	<p>Simpanlah file ‘Lenna.png’ menjadi format JPEG dengan menggunakan method imwrite pada OpenCV, apakah terdapat perbedaan nilai array pada file citra asli dan file dengan format JPEG? Jelaskan alasannya.</p> <pre>cv2.imwrite("Lenna_output.jpeg", img) # Simpan sebagai JPEG img_jpeg = cv2.imread("Lenna_output.jpeg") #Memanggil gambar lagi beda = cv2.absdiff(img, img_jpeg) # Hitung selisih absolut print("Nilai selisih minimum:", beda.min()) print("Nilai selisih maksimum:", beda.max())</pre> <p>Nilai selisih minimum: 0 Nilai selisih maksimum: 86</p>

Audio

a.	<p>Percobaan Praktikum</p> <table><tr><td>a.</td><td><p>Membuat sinyal gelombang</p><pre>sr = 44100 #menentukan sampling rate freq = 1 #menentukan frekuensi (Hz)</pre></td></tr></table>	a.	<p>Membuat sinyal gelombang</p> <pre>sr = 44100 #menentukan sampling rate freq = 1 #menentukan frekuensi (Hz)</pre>
a.	<p>Membuat sinyal gelombang</p> <pre>sr = 44100 #menentukan sampling rate freq = 1 #menentukan frekuensi (Hz)</pre>		

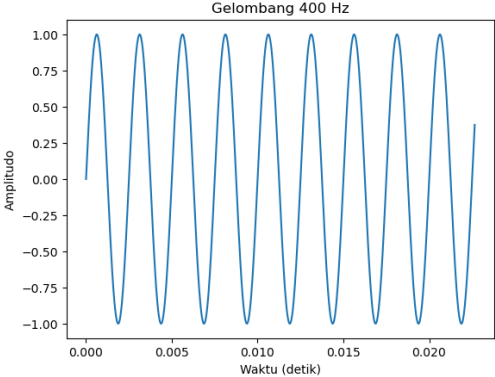
PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

	<pre>length = 1 # Panjang gelombang (Amplitudo) t = np.arange(0, length, 1/sr) #membuat fungsi linear dari 0 s/d time dengan titik berjumlah sr, 1/sr signal = np.sin(2 * np.pi * freq * t) #membuat wave dengan sin function wave = A*sin(2*pi*freq*t) plt.plot(t, signal) #membuat plot dari t dan signal plt.show #menampilkan diagram</pre>  <pre># Simpan sebagai file wav write("file.wav" , sr, signal)</pre>
b.	<p>Mengganti frekuensi agar bisa didengar manusia</p> <pre>freq2 = 200 #setting frekuensi 200 (20-20KHz batas yang bisa didengar manusia) signal2 = np.sin(2 * np.pi * freq2 * t) plt.plot(t, signal2) plt.show</pre> 
b.	<p>Tugas Praktikum</p> <p>a.</p> <p>Modifikasi kode bagian (a) agar membuat gelombang suara dengan frekuensi 400. Berapa panjang gelombangnya sekarang? Apakah file suara yang dihasilkan frekuensi 400 berbeda dengan file suara yang dihasilkan kode bagian (b)? jelaskan alasannya</p> <pre>freq1 = 400 wave1 = np.sin(2 * np.pi * freq1 * t) plt.plot(t[:1000], wave1[:1000]) #menampilkan gelombang 1000 pertama plt.title("Gelombang 400 Hz")</pre>

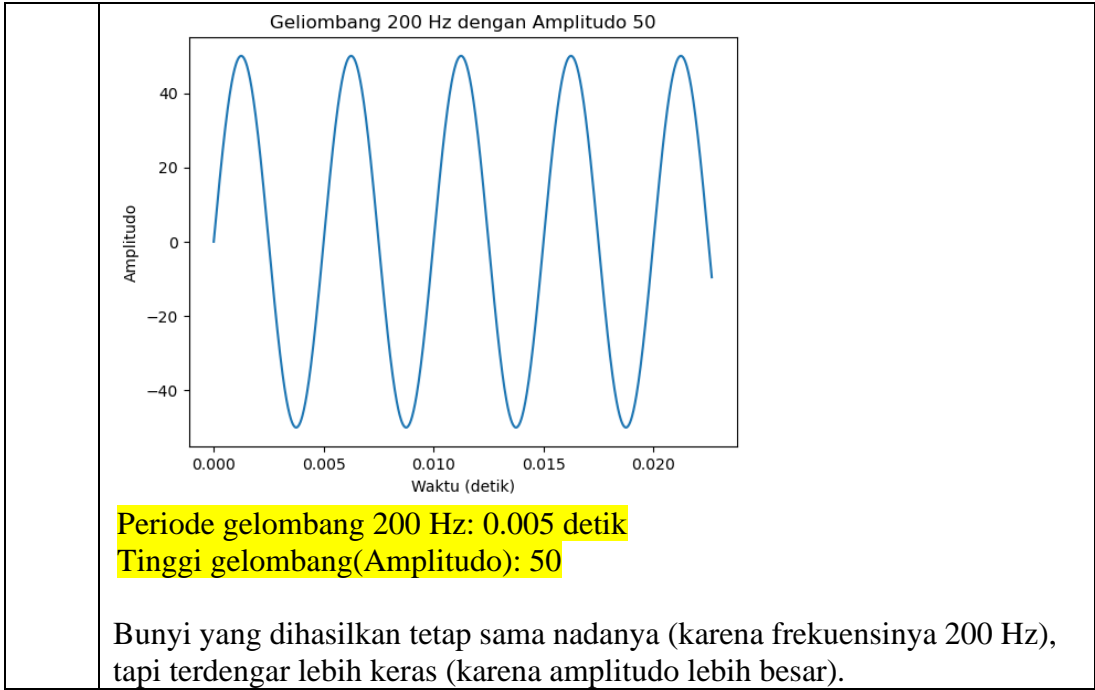
PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

	<pre>plt.xlabel("Waktu (detik)") plt.ylabel("Amplitudo") plt.show() write("wave_400Hz.wav", sr, wave1) period_400 = 1/freq1 print("Periode gelombang 400 Hz:", period_400, "detik")</pre>  <p>Periode gelombang 400 Hz: 0.0025 detik</p> <p>Suara yang dihasilkan lebih nyaring dibanding 200Hz, namun tidak jauh berbeda. Dan Panjang gelombang tetap 1 karena tidak ada perubahan.</p>
b.	<p>Modifikasi kode bagian (b) dengan mengganti nilai amplitudo menjadi 50. Pada output gelombang, berapakah panjang dan tinggi gelombang sekarang? Apakah terdapat perbedaan bunyi dengan hasil suara dari kode bagian (b)? jelaskan alasannya</p> <pre>freq2 = 200 amp = 50 wave2 = amp * np.sin(2 * np.pi * freq2 * t) plt.plot(t[:1000], wave2[:1000]) plt.title("Gelombang 200 Hz dengan Amplitudo 50") plt.xlabel("Waktu (detik)") plt.ylabel("Amplitudo") plt.show() write("wave_200_Hz_amp50.wav", sr, wave2) period_200 = 1/freq2 print("Periode gelombang 200 Hz:", period_200, "detik") print("Tinggi gelombang (Amplitudo):", amp)</pre>

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025



Text

a. Percobaan Praktikum

a.	<p>ASCII</p> <pre>print("ASCII 'A':", ord('A')) # 65</pre> <p>digunakan untuk mengubah karakter menjadi ASCII code</p> <pre>print("Char dari 65:", chr(65)) # 'A'</pre> <p>digunakan untuk mengubah ASCII code menjadi karakter</p> <p>ASCII 'A': 65 Char dari 65: A</p>
b.	<p>One-hot Encoding</p> <pre>docs = "I ate an apple" # Memecah kalimat menjadi list kata split_docs = docs.split(" ") # ['I', 'ate', 'an', 'apple'] # Membuat nested list (setiap kata dianggap sebagai list 1 elemen) data = [doc.split(" ") for doc in split_docs] # Hasil: [['I'], ['ate'], ['an'], ['apple']] # Mengubah nested list menjadi array 1 dimensi (ravel = flatten array) values = array(data).ravel() # Hasil: ['I', 'ate', 'an', 'apple'] # LabelEncoder digunakan untuk mengubah kata menjadi angka unik label_encoder = LabelEncoder() integer_encoded = label_encoder.fit_transform(values)</pre>

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

	<pre># Contoh hasil: [1, 2, 0, 3] (tiap kata diberi ID unik) print("Integer encoded:", integer_encoded) # Inisialisasi OneHotEncoder (sparse_output=False agar hasil berupa array biasa, bukan sparse matrix) onehot_encoder = OneHotEncoder(sparse_output=False) # Reshape agar data menjadi bentuk (n,1) sesuai input yang dibutuhkan OneHotEncoder integer_encoded = integer_encoded.reshape(len(integer_encoded), 1) # Transform angka (label) ke bentuk one-hot encoding onehot_encoded = onehot_encoder.fit_transform(integer_encoded) print("One-hot encoded:\n", onehot_encoded)</pre> <p>Integer encoded: [0 3 1 2] One-hot encoded: [[1. 0. 0. 0.] [0. 0. 0. 1.] [0. 1. 0. 0.] [0. 0. 1. 0.]]</p>
c.	<p>CountVectorizer</p> <pre>text = ["everybody love nlp", "nlp is cool", "nlp is all about helping machines process language", "this tutorial is o basic nlp technique"] # Inisialisasi CountVectorizer # CountVectorizer berfungsi untuk mengubah teks menjadi representasi numerik (bag-of-words) vectorizer = CountVectorizer() # Melatih vectorizer untuk membangun kosakata dari teks vectorizer.fit(text) # Menampilkan kosakata yang berhasil diekstrak # Kosakata adalah semua kata unik dari seluruh dokumen print("Vocabulary:", vectorizer.vocabulary_) # Contoh hasil: {'everybody': 4, 'love': 8, 'nlp': 10, 'is': 7, ...} # Mengubah teks menjadi vektor angka berdasarkan frekuensi kata (bag-of-words) vector = vectorizer.transform(text) # Menampilkan ukuran matriks (jumlah dokumen, jumlah kosakata unik) print("Shape of the vector:", vector.shape) # Contoh: (4, 17) → 4 dokumen, 17 kata unik # Menampilkan hasil transformasi dalam bentuk array print("Vector:\n", vector.toarray()) # Setiap baris mewakili dokumen, setiap kolom mewakili kata unik, nilainya = jumlah kemunculan kata</pre> <p>Vocabulary: {'everybody': 4, 'love': 8, 'nlp': 10, 'is': 6, 'cool': 3, 'all': 1, 'about': 0, 'helping': 5, 'machines': 9, 'process': 11, 'language': 7, 'this': 13, 'tutorial': 14, 'basic': 2, 'technique': 12}</p>

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

	<div>Shape of the vector: (4, 15)</div> <div>Vector:</div> <div>[[0 0 0 1 0 0 0 1 0 1 0 0 0 0]</div> <div>[0 0 0 1 0 0 1 0 0 0 1 0 0 0 0]</div> <div>[1 1 0 0 0 1 1 1 0 1 1 1 0 0 0]</div> <div>[0 0 1 0 0 0 1 0 0 0 1 0 1 1 1]]</div>
d.	<div>TF-IDF</div> <div># Dataset berupa list kalimat</div> <div>text1 = ["everybody love nlp",</div> <div>"nlp is cool",</div> <div>"nlp is all about helping machines process</div> <div>language",</div> <div>"this tutorial is o basic nlp technique"]</div> <div># Inisialisasi TF-IDF Vectorizer</div> <div># TF-IDF = Term Frequency - Inverse Document Frequency</div> <div># Dipakai untuk memberi bobot kata: semakin sering muncul di</div> <div>satu dokumen tapi jarang muncul di dokumen lain → bobot</div> <div>tinggi</div> <div>tf = TfidfVectorizer()</div> <div># Melatih vectorizer dengan data text1 (membangun kosakata</div> <div>dan menghitung IDF)</div> <div>txt_fitted = tf.fit(text1)</div> <div># Transformasi text1 menjadi matriks TF-IDF</div> <div>txt_transformed = tf.transform(text1)</div> <div># Mendapatkan nilai IDF untuk setiap kata di kosakata</div> <div>idf = tf.idf_</div> <div># Menampilkan kosakata beserta nilai IDF-nya dalam bentuk</div> <div>dictionary</div> <div>print(dict(zip(tf.get_feature_names_out(), idf)))</div> <div>{'about': 1.916290731874155, 'all': 1.916290731874155, 'basic':</div> <div>1.916290731874155, 'cool': 1.916290731874155, 'everybody':</div> <div>1.916290731874155, 'helping': 1.916290731874155, 'is':</div> <div>1.2231435513142097, 'language': 1.916290731874155, 'love':</div> <div>1.916290731874155, 'machines': 1.916290731874155, 'nlp': 1.0, 'process':</div> <div>1.916290731874155, 'technique': 1.916290731874155, 'this':</div> <div>1.916290731874155, 'tutorial': 1.916290731874155}</div>

b. Tugas Praktikum

a.	<div>Modifikasi kode bagian (a) agar bisa menampilkan ASCII code untuk kata</div> <div>‘datA mining’</div> <div>text = "datA mining"</div> <div>ascii = {} #membuat dictionary kosong untuk tempat ascii</div> <div>code</div> <div>for char in text:</div> <div>ascii[char] = ord(char) #mengubah kata menjadi ASCII code</div> <div>print(ascii)</div> <div>{'d': 100, 'a': 97, 't': 116, 'A': 65, ' ': 32, 'm': 109, 'i': 105, 'n': 110, 'g': 103}</div>
----	---

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

b.	<p>Tambahkan kode bagian (b) agar bisa menampilkan kembali kata pertama yang di lakukan one-hot encoding</p> <pre>sentence = "white cloud in the blue sky" # Split kalimat jadi list kata words = sentence.split(" ") # Bungkus tiap kata jadi list baru data = [word.split(" ") for word in words] # Ubah nested list menjadi array 1 dimensi values = array(data).ravel() # LabelEncoder untuk ubah kata → angka unik label_encoder = LabelEncoder() integer_encoded = label_encoder.fit_transform(values) # Inisialisasi OneHotEncoder onehot_encoder = OneHotEncoder(sparse_output=False) # Reshape agar jadi (n,1), sesuai input OneHotEncoder integer_encoded = integer_encoded.reshape(len(integer_encoded), 1) # Transform angka label → one-hot encoding onehot_encoded = onehot_encoder.fit_transform(integer_encoded) print("One-hot encoded:\n", onehot_encoded) print(onehot_encoded) # Mengubah kembali one-hot → integer index (posisi 1 pada vektor) integer_onehot = np.argmax(onehot_encoded, axis=1) # Decode angka index → kata asli decoded_words = label_encoder.inverse_transform(integer_onehot) print("\nReverse One-hot encoding:\n", decoded_words)</pre> <p>One-hot encoded: [[0. 0. 0. 0. 0. 1.] [0. 1. 0. 0. 0. 0.] [0. 0. 1. 0. 0. 0.] [0. 0. 0. 0. 1. 0.] [1. 0. 0. 0. 0. 0.] [0. 0. 0. 1. 0. 0.]] [[0. 0. 0. 0. 0. 1.] [0. 1. 0. 0. 0. 0.] [0. 0. 1. 0. 0. 0.] [0. 0. 0. 0. 1. 0.] [1. 0. 0. 0. 0. 0.] [0. 0. 0. 1. 0. 0.]]</p> <p>Reverse One-hot encoding: ['white' 'cloud' 'in' 'the' 'blue' 'sky']</p>
c.	<p>Download file bebas, kemudian lakukan CountVectorizer dan TF-IDF pada korpus tersebut. Jelaskan hasil yang didapatkan.</p> <p>Sumber text: https://id.wikipedia.org/wiki/Revolusi_Prancis</p>

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

	<pre># Membuka file teks yang berisi dokumen with open("wiki_content.txt", "r") as file: documents = file.readlines() # Membaca setiap baris sebagai 1 dokumen cv = CountVectorizer() # Inisialisasi CountVectorizer (bag- of-words) vectorizer = cv # Alias agar konsisten vectorizer.fit(documents) # Membangun kosakata dari seluruh dokumen print("Vocabulary:", vectorizer.vocabulary_) # Menampilkan kosakata unik beserta indeksnya vector = vectorizer.transform(documents) # Mengubah dokumen jadi vektor frekuensi kata print("Shape of the Count Vector:", vector.shape) # Bentuk matriks (jumlah dokumen, jumlah kata unik) print("\nCount Vector:\n", vector.toarray()) # Menampilkan matriks hasil bag-of-words tf = TfidfVectorizer() # Inisialisasi TF-IDF Vectorizer txt_fitted = tf.fit(documents) # Membaca seluruh dokumen dan membangun kosakata + IDF txt_transformed = txt_fitted.transform(documents) # Mengubah dokumen menjadi matriks TF-IDF idf = tf.idf_ # Mengambil nilai IDF untuk setiap kata idf_scores = dict(zip(tf.get_feature_names_out(), idf)) # Membuat dictionary: kata -> IDF score sorted_idf = dict(sorted(idf_scores.items(), key=lambda item: item[1])) # Urutkan kata berdasarkan nilai IDF (dari kecil ke besar) print("Sorted IDF Scores:", sorted_idf) # Menampilkan daftar kata dengan IDF terurut Vocabulary: {'revolusi': 1522, 'perancis': 1384, 'prancis': 1468, 'bahasa': 147, 'révolution': 1542, 'française': 541, '1789': 19, '1799': 26,...} Shape of the Count Vector: (251, 1861) Count Vector: [[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]] Sorted IDF Scores: {'dan': 1.965080896043587, 'yang': 2.110588479714825, 'di': 2.355041817615786, 'prancis': 2.4689860769650043, 'dengan': 2.486377819676873,...}</pre>
d.	<p>Modifikasi kode bagian (d) agar bisa menampilkan grafik dari tiap kata</p> <pre>plt.figure(figsize=(15,8)) # Ukuran figure</pre>

PRAKTIKUM DATA MINING II

BAB : DATA ACQUISITION, REPRESENTATION, AND STORAGE
PRAKTIKUM KE : SATU (1)
NAMA : ARWEN SUTANTO PUTRA
NIM : 164231041
TGL PRAKTIKUM : 29 AGUSTUS 2025

```
plt.bar(list(sorted_idf.keys())[:25],  
list(sorted_idf.values())[:25]) # Plot 25 kata dengan IDF  
terendah  
plt.xlabel("Kata") # Label sumbu X  
plt.ylabel("TF-IDF Score") # Label sumbu Y  
plt.title("TF-IDF Score per Kata (25 Kata Terkecil)") #  
Judul grafik  
plt.tight_layout() # Atur layout agar tidak bertabrakan  
plt.show() # Tampilkan grafik
```

