

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Tinjauan Pustaka bertujuan untuk memperdalam pemahaman mengenai konsep, metode serta penelitian terdahulu. Melalui referensi dari berbagai literatur sebagai dasar ilmiah yang kuat untuk menganalisis dan menyelesaikan permasalahan utama penelitian.

##### **2.1.1 Kandungan Kopi**

Kandungan kopi mencakup analisis kafein yang merupakan salah satu senyawa utama dalam kopi. Kafein adalah senyawa alkaloid yang memiliki efek stimulan pada sistem saraf pusat. Beberapa penelitian sebelumnya menunjukkan variasi kadar kafein dalam kopi, tergantung pada jenis kopi, metode penyeduhan, dan faktor lingkungan. Misalnya, penelitian oleh Maramis dkk. (2013) menemukan rata-rata kandungan kafein dalam kopi kemasan bubuk di Manado sebesar 34,76 mg per sajian.

Sedangkan Suryani dkk. (2016) melaporkan rata-rata kandungan kafein sebesar 45,87 mg per sajian pada kopi kemasan di Banten. Penelitian lain oleh Crismaaji (2018) menunjukkan kadar kafein sebesar 23,45 mg per sajian pada kopi bubuk robusta di Yogyakarta. Kandungan kafein dalam kopi juga dipengaruhi oleh faktor-faktor seperti suhu air seduhan, proses roasting, dan tempat tumbuh kopi. Penelitian ini bertujuan untuk menganalisis kandungan kafein pada kopi untuk

menentukan apakah kadar kafein tersebut memenuhi standar kalori yang bagus atau tidak (Silviana & Santika, 2020).

### **2.1.2 Data Mining**

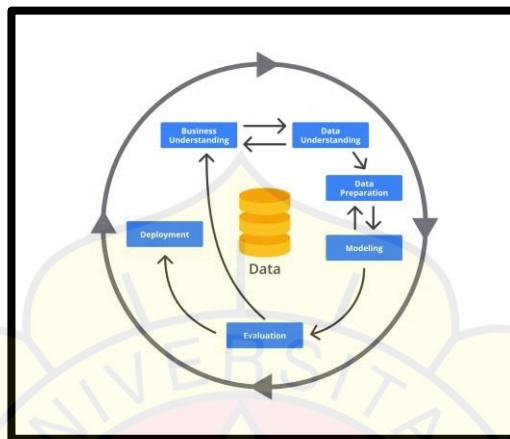
Data mining, atau penambangan data, merupakan bagian dari *Knowledge Discovery in Databases* (KDD) yang berfokus pada pencarian pola tersembunyi dan informasi penting dari kumpulan data besar. Proses ini dapat dilakukan secara otomatis maupun manual, serta melibatkan interaksi antara manusia dan komputer dalam tahapan yang bersifat iteratif (Mai et al., 2022) Tujuan utama data mining adalah menggali pengetahuan baru melalui pengenalan pola dan relasi dalam data. Dalam kajian literatur, batas antara data mining dan *machine learning* tidak selalu jelas.

Beberapa sumber menyatakan bahwa data mining lebih menitikberatkan pada identifikasi pola dan hubungan data, sedangkan *machine learning* berorientasi pada kemampuan memprediksi (Barnabas, 2021, p. 25) Sementara itu, analisis prediktif berfokus pada pemanfaatan data untuk memperkirakan kejadian di masa depan. Oleh karena itu, meskipun saling berkaitan, data mining lebih diarahkan untuk menjelaskan fenomena melalui pola dalam data, sedangkan analisis prediktif digunakan untuk membuat prediksi berdasarkan informasi tersebut (Barnabas, 2021).

### **2.1.3 CRIPS-DM sebagai Tahap Merancang Sistem Berbasis Machine Learning**

Metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*) dengan menerapkan *library Lazypredict* 0.2.12 digunakan untuk memberikan

gambaran klasifikasi dalam proses pembelajaran mesin terhadap data yang ada dan Framework CRISP-DM ini adalah standar untuk proses data mining yang terdiri dari enam fase utama (Dzulhijjah et al., n.d.).



Gambar 2. 1 Metode Crips-dm (sumber: Dicoding)

### 1. Fase Pemahaman Bisnis

Pada tahap ini, beberapa langkah yang dilakukan antara lain memahami kebutuhan dan tujuan dari perspektif bisnis, kemudian mengonversi pengetahuan tersebut menjadi bentuk pendefinisian masalah dalam data mining, serta merumuskan rencana dan strategi untuk mencapai tujuan yang diinginkan dalam data mining. (Hasanah et al., 2021)

### 2. Fase Pemahaman Data

Mengumpulkan data, jika data berasal dari lebih dari satu database maka dilakukan proses integrasi data atau Integration. Data Mengembangkan analisis penyelidikan data untuk mengenali lebih lanjut data dan pencarian pengetahuan awal. Mengevaluasi kualitas data, memeriksa data dan membersihkan data yang

tidak valid atau proses *Data Cleaning*. Jika diinginkan, pilih sebagian kecil grup data yang mungkin mengandung pola dari permasalahan. (Purnama et al., n.d.)

### 3. Fase Persiapan Data

Dalam tahapan ini yaitu membangun dataset akhir dari berupa data mentah. Ada beberapa hal yang akan dilakukan mencakup melakukan pembersihan data (*Data Cleaning*), melakukan pemilihan data (*Data Selection*), *record* dan atribut-atribut, dan juga melakukan transformasi terhadap data (*Data Transformation*) untuk dijadikan masukan dalam tahap pemodelan. (Suhanda et al., 2020)

### 3. Fase Pemodelan

Tahap Modeling dalam metodologi CRISP-DM adalah fase penting dalam proses analisis data, di mana berbagai teknik pemodelan dan algoritma digunakan untuk diterapkan pada data yang telah dipersiapkan dengan cermat. Tujuan dari tahap ini adalah untuk mengembangkan model yang dapat memberikan wawasan atau prediksi sesuai dengan masalah yang ingin diselesaikan. Pada tahap ini, data yang telah diproses dan dibersihkan digunakan sebagai dasar untuk membangun model yang dapat memecahkan permasalahan bisnis atau penelitian yang telah ditentukan sebelumnya. Berbagai teknik pemodelan, seperti klasifikasi, regresi, clustering, atau lainnya, dipilih dan diterapkan untuk menguji kecocokan model dalam mencapai tujuan yang diinginkan. (Apriori et al., 2024)

### 4. Fase Evaluasi

Evaluasi adalah tahap di mana hasil dari proses data mining dianalisis dan diinterpretasikan. Proses evaluasi dilakukan secara menyeluruh dengan tujuan untuk memastikan bahwa model yang dihasilkan sudah memenuhi tujuan yang

telah ditetapkan sebelumnya. Pada tahap ini, dilakukan penilaian terhadap kinerja model untuk memastikan bahwa model tersebut efektif dan akurat dalam memberikan hasil yang diinginkan. (Khumaidi, 2020)

## 5. Fase Penyebaran

Pada fase ini dilakukan pengukuran performa model menggunakan teknik *Confussion Matrix*, serta *10-fold Cross Validation* untuk memvalidasi model setelah melalui fase evaluasi, algoritma yang telah dikembangkan dapat diimplementasikan, baik dalam bentuk laporan atau aplikasi pendukung keputusan berbasis sumber terbuka. (Wurijanto et al., 2022)

### 2.1.4 Algoritma *Principal Component Analysis* (PCA)

PCA adalah metode statistik yang digunakan untuk mengurangi dimensi data dengan mempertahankan bagian-bagian informasi yang paling signifikan. Dalam penelitian ini, Metode ini membantu peneliti dalam mengidentifikasi variabel-variabel kunci yang paling mempengaruhi proses penyelesaian skripsi, sehingga dapat memberikan rekomendasi yang lebih terfokus pada peningkatan kualitas bimbingan akademik dan pengelolaan waktu bagi mahasiswa (Wangge, 2021).

Langkah-langkah penggunaan algoritma PCA:

#### 1. Persiapan dataset

Langkah awal dalam *Principal Component Analysis* (PCA) adalah menyiapkan data numerik dalam bentuk matriks, di mana setiap baris mewakili satu sampel dan setiap kolom merepresentasikan fitur. Dikarenakan setiap fitur dapat memiliki satuan yang berbeda-beda (seperti gram, mililiter, atau kalori), maka

penting untuk menyeragamkan skala agar fitur-fitur tersebut tidak memberikan pengaruh yang tidak proporsional. Tanpa penyesuaian skala, fitur dengan nilai yang lebih besar akan memiliki bobot lebih tinggi dalam pembentukan komponen utama.

## 2. Normalisasi Data *Z-Score*

Untuk memastikan bahwa semua fitur memberikan kontribusi yang seimbang, maka dilakukan proses standarisasi menggunakan metode *Z-Score*, dengan rumus sebagai berikut:

$$Z = \frac{x - \mu}{\sigma} \quad (2.1)$$

Keterangan:

- a. X : merupakan nilai asli suatu fitur
- b. Mu : adalah rata-rata (*mean*) fitur
- c. Sigma : simpangan baku (standar deviasi) fitur

Hasil dari standarisasi ini adalah data dengan mean 0 dan standar deviasi

1. Langkah ini penting karena PCA mengandalkan variansi dalam menentukan komponen utamanya. Jika tidak dinormalisasi, fitur yang memiliki nilai besar akan mendominasi hasil PCA, meskipun mungkin tidak relevan secara substansi. Sebagai ilustrasi, jika fitur A memiliki nilai antara 1–10, sedangkan fitur B antara 100–10.000, maka fitur B akan memiliki variansi lebih tinggi, sehingga bisa memengaruhi hasil secara tidak adil.

## 3. Membuat Matriks Kovarians

Setelah data distandarisasi, tahap selanjutnya adalah membuat matriks kovarians untuk mengetahui hubungan linier antar fitur. Kovarians sendiri

mengukur sejauh mana dua fitur berubah secara bersamaan. Kovarians bernilai positif menunjukkan bahwa kedua fitur meningkat bersama, sedangkan nilai negatif menunjukkan arah perubahan yang berlawanan. Rumus umum kovarians adalah:

$$Cov = \frac{1}{n-1} X^t \cdot X \quad (2.2)$$

Keterangan:

- a. X: adalah matriks data hasil normalisasi
- b. X<sup>t</sup>: adalah transpose dari x
- c. N: adalah jumlah observasi

Ukuran dari matriks kovarians adalah jumlah fitur x jumlah fitur, Dimana:

- a. Elemen diagonal menunjukkan variansi masing-masing fitur
- b. Elemen diluar diagonal menunjukkan kovarians antar fitur

Apabila data telah melalui proses *Z-Score*, maka nilai diagonal mendekati 1 dan nilai lain menggambarkan korelasi antar fitur. Dalam hal ini, matriks kovarians sering dianggap sebagai matriks korelasi.

#### 4. Menghitung *Eigenvalues* dan *Eigenvectors*

Langkah berikutnya adalah menghitung *eigenvalue* dan *eigenvector* dari matriks kovarians. Kedua komponen ini penting karena menentukan arah serta seberapa besar informasi yang dikandung oleh komponen utama. *Eigenvector* menggambarkan arah dari sumbu baru (komponen utama), sementara *Eigenvalue* menunjukkan besar kontribusi (varian) dari setiap komponen utama. Secara matematis, nilai *eigen* diperoleh dengan menyelesaikan:

$$\det(C - \lambda I) = 0 \quad (2.3)$$

Keterangan:

- a. C: adalah matrik kovarians
- b. Lambda : adalah nilai *eigen*
- c. I: adalah matriks identitas

*Eigenvector* dengan nilai *eigen* (*eigenvalue*) terbesar menjadi komponen utama pertama (PC1) karena memuat informasi paling besar. Komponen berikutnya (PC2, PC3, dst) memiliki kontribusi yang lebih kecil dan saling ortogonal terhadap satu sama lain.

#### 5. Proyeksi Data ke Komponen Utama

Setelah menentukan komponen utama, data yang telah dinormalisasi diproyeksikan ke arah vektor-vektor *eigen* tersebut. Proses ini dilakukan dengan mengalikan data dengan *eigenvector* yang terpilih:

$$Y = X_{normalized} \cdot v \quad (2.4)$$

Keterangan:

- a. Y: adalah hasil proyeksi ke ruang baru
- b.  $X_{normalized}$ : adalah data yang telah dinormalisasi
- c. V: adalah *eigenvector*

Dengan langkah ini, dimensi data dapat direduksi. Misalnya, data awal memiliki 4 fitur, dan kita hanya mengambil 2 komponen utama dengan informasi terbanyak, maka data dapat direpresentasikan dalam ruang dua dimensi dengan informasi yang tetap relevan.

## 6. Transformasi Data ke Ruang Dimensi Baru

Proyeksikan data asli ke ruang baru dengan mengalikan data yang sudah dinormalisasi dengan matriks proyeksi:

$$X_{pca} = X_{norm} \cdot W \quad (2.5)$$

Hasil dari langkah ini adalah data yang telah direduksi ke dalam dimensi baru dengan varians yang lebih besar pada komponen utama.

### 2.1.5 Algoritma *Support Vector Machine* (SVM)

*Support Vector Machine* (SVM) merupakan salah satu algoritma *machine learning* yang digunakan dalam penyelesaian permasalahan klasifikasi maupun regresi. Inti dari metode ini adalah memisahkan data menjadi dua kelompok dengan menggunakan garis pemisah terbaik yang disebut *hyperplane*. SVM bertujuan untuk menemukan *hyperplane* yang memberikan jarak maksimum (margin) antara dua kelompok data yang berbeda, sehingga meningkatkan kemampuan model dalam melakukan generalisasi terhadap data baru.

SVM dapat diaplikasikan pada data yang bersifat linier maupun *non-linier*. Pada kasus *non-linier*, SVM memanfaatkan *kernel trick* untuk mentransformasikan data ke dalam ruang berdimensi lebih tinggi agar dapat dipisahkan secara *linier*. Sementara itu, pada data linier, pemisahan antar kelas dapat dilakukan langsung tanpa bantuan fungsi kernel.

Dalam penelitian ini, algoritma SVM digunakan untuk melakukan klasifikasi kategori minuman kopi berdasarkan nilai kalorinya ke dalam dua kelompok, yaitu Kalori Rendah dan Kalori Tinggi, setelah sebelumnya data

direduksi dimensinya menggunakan metode *Principal Component Analysis* (PCA).  
(Lukman Priyambodo et al., 2022).

Langkah-langkah SVM:

### 1. Menyiapkan Dataset

Dataset harus memiliki fitur  $x$  dan label kelas  $y$ . Setiap sampel data dinotasikan sebagai  $(x_i, y_i)$  di mana  $i = 1, 2, \dots, n$ , dengan  $y_i \in \{-1, 1\}$  yang menunjukkan kelas data.

### 2. Menentukan *Hyperplane* dan Margin

Setelah optimasi selesai, diperoleh *hyperplane* yang digunakan untuk memisahkan dua kelas. *Hyperplane* dipresentasikan oleh:

$$f(x) = w \cdot x + b \quad (2.6)$$

*Hyperplane* akan memisahkan data menjadi dua kelas, dan margin adalah jarak antara *hyperplane* dan titik data terdekat dari masing-masing kelas.

### 3. Menggunakan Parameter Regularisasi $C$ (*Soft Margin*)

Dalam kasus di mana data tidak dapat sepenuhnya dipisahkan, parameter regularisasi  $C$  digunakan untuk menyeimbangkan antara margin maksimum dan kesalahan klasifikasi. Fungsi tujuan menjadi:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \varepsilon_i \quad (2.7)$$

Dengan kendala:

$$y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0 \quad (2.8)$$

Disini  $\varepsilon_i$  adalah variabel slack yang memungkinkan beberapa kesalahan klasifikasi.

#### 4. Evaluasi Model *Support Vector Machine* (SVM)

Setelah model *Support Vector Machine* (SVM) dilatih, evaluasi dilakukan menggunakan *Confusion Matrix*, dengan 4 komponen utama:

- a. *True Positive (TP)*
- b. *True Negative (TN)*
- c. *False Positive (FP)*
- d. *False Negative (FN)*

Dari *matrix* ini, dihitung *matrix* evaluasi seperti:

- a. *Accuracy*

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.9)$$

- b. *Precision*

$$precision = \frac{TP}{TP + FP} \quad (2.10)$$

- c. *Recall*

$$recall = \frac{TP}{TP + FN} \quad (2.11)$$

Evaluasi ini penting untuk mengukur ketepatan dan kemampuan model dalam mengenali data kategori tertentu, terutama dalam klasifikasi dua kelas seperti pada penelitian ini.

#### 2.1.6 Pemodelan UML





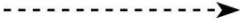
*Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan

oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.

### 2.1.6.1 Use Case Diagram

*Use Case* bekerja dengan menggunakan scenario yang merupakan deskripsi dari urutan atau langkah –langkah yang menjelaskan apa yang dilakukan oleh user terhadap sistem maupun sebaliknya. *Use Case* mengidentifikasi fungsionalitas yang dipunya sistem, interaksi user dengan sistem dan keterhubungan antara user dengan fungsionalitas sistem. (Setiyani, 2021).




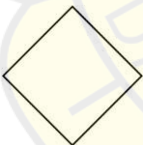

Tabel 2. 1 *Usecase Diagram*

No.	Notasi	Keterangan
1	Aktor 	Aktor adalah setiap individu atau entitas yang berinteraksi dengan sistem yang dibangun, serta semua entitas yang berada di luar batas sistem.
2	<i>Usecase</i> 	<i>Usecase</i> merupakan komponen yang mencerminkan fungsionalitas tingkat tinggi dari sebuah sistem; dengan kata lain, <i>Usecase</i> menjelaskan bagaimana seseorang berinteraksi dan menggunakan sistem tersebut.
3	Relasi Generalisasi 	Istilah "relasi generalisasi" digunakan untuk menunjukkan bahwa beberapa aktor atau kasus memiliki kesamaan tertentu.
4	Relasi <i>Include</i> <<include>> 	Sebuah <i>Usecase</i> dapat memanfaatkan fungsionalitas yang disediakan oleh <i>Usecase</i> lain karena adanya hubungan antara keduanya.
5	Relasi <i>Extend</i> <<extend>> 	Hubungan antara <i>Usecase</i> tambahan dan sebuah <i>Usecase</i> , di mana <i>Usecase</i> tambahan dapat berfungsi secara independen tanpa memerlukan <i>Usecase</i> utama.

### 2.1.6.2 Activity Diagram

*Activity Diagram* menggambarkan sebuah workflow (aliran kerja) atau juga aktivitas dari sebuah sistem atau juga proses bisnis. (Sopriani & Purwanto, n.d.).

Tabel 2. 2 *Activity Diagram*

No.	Notasi	Keterangan
1	<i>Start State</i> 	Notasi <i>start state</i> adalah notasi penanda keadaan awal di mana alur kerja dimulai.
2	<i>State Activity</i> 	Notasi disamping menunjukkan sebuah kegiatan dalam alir diagram
3	Transisi 	Menunjukkan bagaimana alur kerja mengalir dari satu tugas ke tugas lainnya
4	<i>Decision</i> 	Diagram keputusan menunjukkan bahwa keputusan harus dibuat dalam alur kerja
5	<i>State End</i> 	Notasi <i>State End</i> adalah notasi penanda keadaan akhir di mana alur kerja selesai.

### **2.1.7 Software dan Pemrograman Terkait**

Software dan pemrograman terkait dapat mempengaruhi efisiensi, fungsionalitas serta implementasi sistem yang akan dikembangkan. Pada konteks ini, *software* dan pemrograman yang digunakan mencakup editor pengolah data beserta Bahasa pemrograman yang relevan dengan penelitian.

#### **2.1.7.1 Editor Notebook Jupyter**

*Jupyter Notebook* adalah aplikasi analisis data yang diluncurkan pada 2015, dirancang untuk membuat dan berbagi kode, hasil analisis, visualisasi data, serta narasi komputasi. Nama *Jupyter* berasal dari Julia (Ju), *Python* (Py), dan R, bahasa pemrograman yang didukungnya. Aplikasi gratis ini mendukung *open science*, mempermudah analisis, kolaborasi, dan penyajian data. Dengan fitur integrasi, seperti *Google Drive*, *Jupyter* membantu menghasilkan kesimpulan yang akurat untuk mendukung pengambilan keputusan dan strategi perusahaan. (Asyrofi & Asyrofi, 2023).

#### **2.1.7.2 Pemrograman Python**

Pemrograman berbasis *python* adalah bersifat *open-source* yang berarti bersifat umum dan dapat dikembangkan oleh siapa pun. *Python* diinterpretasikan sebagai bahasa tingkat tinggi yang justru menyederhanakan pemrograman yang berorientasi objek melalui pendekatannya. *Python* merupakan bahasa pemrograman yang paling relevan digunakan untuk para ilmuwan data untuk berbagai aplikasi ilmu data, dan juga memiliki fungsionalitas yang sangat baik untuk menangani matematika, statistik, dan fungsi ilmiah (Muhammad & Yulianto, n.d.). *library* yang digunakan untuk pengolahan data pada penelitian ini yaitu:

### A. *Library Pandas*

*Pandas* adalah pustaka sumber terbuka yang digunakan dalam *Python* yang menyediakan metrik kinerja yang ditingkatkan, struktur data yang mudah digunakan, serta paket analisis data, alat, dan pustaka untuk Bahasa Pemrograman *Python*. Penggunaan *pandas* dengan *Python* mencakup berbagai bidang keahlian termasuk ilmu data, pemodelan komputasional, keuangan, ekonomi, analisis statistik, pembelajaran mesin, dan lain-lain (“*Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python*,” 2021). Berikut contoh file CSV ke dalam *DataFrame Pandas*:

```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.to_string())
```

Gambar 2. 2 *Library Pandas* (sumber: w3schools)

### B. *Library Matplotlib*

*Matplotlib* adalah salah satu pustaka visualisasi data yang paling populer digunakan dalam *Python*. Pustaka ini dibuat oleh *John Hunter* bersama dengan beberapa kontributor lainnya, dan mereka telah menghabiskan banyak waktu untuk mempromosikan perangkat lunak ini yang digunakan oleh setiap ilmuwan dan filsuf di seluruh dunia. *Matplotlib* adalah pustaka grafis untuk paket visualisasi data dalam *Python* yang merupakan bagian integral dari tumpukan ilmu data *Python* dan mudah didukung dengan *NumPy*, *Pandas*, dan pustaka relevan lainnya.

(“Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python,” 2021). Berikut contoh Gambar garis dalam diagram dari posisi (0,0) ke posisi (6,250):

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([0, 6])
ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)
plt.show()
```

Gambar 2. 3 Library *matplotlib* (sumber: w3schools)

### C. Library *Seaborn*

*Seaborn* adalah pustaka visualisasi grafis yang dibangun di atas konfigurasi utama *Matplotlib*. Pustaka ini menyediakan aksesibilitas kepada pengguna dengan beberapa proses visualisasi data yang paling umum, serta kebutuhan visualisasi data tertentu seperti pemetaan warna ke variabel atau penggunaan persyaratan *faceting* di seluruh dunia. *Seaborn* lebih terintegrasi untuk bekerja dengan *Pandas DataFrames*. (“Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python,” 2021). Berikut contoh *code Seaborn* :

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot([0, 1, 2, 3, 4, 5])

plt.show()
```

Gambar 2. 4 Library *seaborn* (sumber: w3schools)

#### D. Library Sklearn

*Scikit-Learn* adalah *library Python* yang dikembangkan untuk memberikan kemudahan dalam pengkodean *machine learning* dalam *Python*, melalui “API” yang dikembangkan oleh beberapa kontributor dari berbagai negara yang mana biasanya digunakan pada dunia industri dan dunia akademisi bisa dikatakan juga bahwa *library scikit-learn* ini sudah termasuk performa yang *optimized* karena *library* ini dirancang di atas modul *NumPy (Numerical Python)* dan *SciPy (Scientific Python)*, sehingga perhitungan di dalamnya menjadi lebih efisien. Namun terdapat kelemahan pada *library* ini yaitu *scikit-learn* tidak dianjurkan untuk dipergunakan pada desain data yang lebih besar. (Fahmi, 2023). Berikut contoh code *SKlearn*:

```
from sklearn import datasets
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import KFold, cross_val_score
X, y = datasets.load_iris(return_X_y=True)
clf = DecisionTreeClassifier(random_state=42)
k_folds = KFold(n_splits = 5)
scores = cross_val_score(clf, X, y, cv = k_folds)
print("Cross Validation Scores: ", scores)
print("Average CV Score: ", scores.mean())
print("Number of CV Scores used in Average: ", len(scores))
```

Gambar 2. 5 *Library scikit-learn* (sumber: w3schools)

#### 2.1.7.3 Pemrograman *Streamlit*

Dengan kemajuan teknologi, *framework Streamlit* semakin populer untuk membuat aplikasi web interaktif. Pengembang dapat menggunakan *Streamlit* untuk membuat antarmuka pengguna yang sederhana, intuitif, dan fungsional tanpa harus tahu banyak tentang pengembangan *front-end*. Misalnya, *Streamlit* menawarkan

komponen interaktif seperti slider, tombol, dan input teks yang memudahkan pengguna memasukkan data dan memprediksi hasil. Hal ini membuat Streamlit menjadi pilihan yang sempurna untuk berbagai proyek data *science* dan *machine learning* yang memerlukan interaksi langsung antara model dan pengguna. (Ayu Syafarina & Kalimantan Muhammad Arsyad Al Banjari Banjarmasin, n.d.).

## **2.2. Tinjauan Literatur**

### **2.2.1 Paper 1**

Judul: ANALISIS PENGARUH PENERAPAN GAYA HIDUP SEHAT TERHADAP MINAT MENGGUNAKAN MAKANAN SEHAT PADA *YELLOW FIT KITCHEN*

Author: Amram Marcell Fernandito, Roozana Maria Ritonga

Publikasi: Green Publisher Indonesia

Tahun: 2023

Klasifikasi Journal: Sinta 5

#### **2.2.1.1 Tujuan Penelitian**

Adanya minat dalam suatu gaya hidup juga memungkinkan seseorang untuk memilih suatu produk yang ingin dikonsumsi. Pasca pandemi Covid-19, masyarakat Indonesia banyak yang memulai gaya hidup sehat demi menjaga daya tahan tubuh yang lebih baik. Gaya hidup sehat juga diterapkan untuk menjaga kesehatan tubuh serta menjaga bentuk tubuh lebih proporsional. Berbagai faktor untuk menjaga kesehatan tubuh tersebut dilakukan dengan banyak cara dan salah satunya dengan mengonsumsi makanan sehat dan mengatur jumlah kalori yang

dapat di konsumsi setiap harinya. Kalori yang harus di perhatikan bukan hanya bersumber dari makanan tetapi juga dari minuman.

### **2.2.1.2 Metodologi yang Digunakan**

Metodologi yang digunakan dalam penelitian ini adalah metode kuantitatif, dengan pengolahan data terkait variabel yang akan diteliti. Penelitian ini menetapkan masyarakat yang menjadi pelanggan *Yellow Fit Kitchen* sebagai populasi dan menggunakan rumus Slovin dalam pengambilan sampel, dengan hasil sampel sebanyak 100 responden dan standar akurasi hasil 90%.

### **2.2.1.3 Temuan Utama**

Temuan utama dari penelitian ini adalah bahwa gaya hidup sehat (variabel X) memiliki pengaruh signifikan terhadap minat konsumsi makanan sehat (variabel Y) di *Yellow Fit Kitchen*. Setiap kenaikan satu unit gaya hidup sehat meningkatkan minat konsumsi makanan sehat sebesar 0.480. Koefisien determinasi ( $R^2$ ) menunjukkan bahwa 69.7% variasi minat konsumsi makanan sehat dipengaruhi oleh gaya hidup sehat, dengan tingkat signifikansi 0.000 ( $p < 0.05$ ). Peneliti merekomendasikan *Yellow Fit Kitchen* untuk melakukan kampanye dan edukasi mengenai pola makan sehat melalui media sosial.

### **2.2.1.4 Kesimpulan Penelitian**

Kesimpulan penelitian ini adalah bahwa gaya hidup sehat memiliki pengaruh signifikan terhadap minat konsumsi makanan sehat di *Yellow Fit Kitchen*. Setiap kenaikan satu unit gaya hidup sehat meningkatkan minat konsumsi makanan sehat sebesar 0.480. Koefisien determinasi ( $R^2$ ) menunjukkan bahwa 69.7% variasi

minat konsumsi makanan sehat dipengaruhi oleh gaya hidup sehat, dengan tingkat signifikansi 0.000 ( $p < 0.05$ ). Peneliti merekomendasikan agar *Yellow Fit Kitchen* melakukan kampanye dan edukasi mengenai pola makan sehat melalui media sosial untuk meningkatkan kesadaran Masyarakat (Fernandito & Ritonga, n.d.).

### **2.2.2 Paper 2**

Judul: ANALISIS KANDUNGAN KAFEIN PADA KOPI SEDUHAN WARUNG KOPI DI KOTA BANDA ACEH

Author: Elfariyanti, Ernita Silviana, Mela Santika

Publikasi: PROGRAM STUDI PENDIDIKAN KIMIA, FTK UIN AR-RANIRY BANDA ACEH

Tahun: 2020

Klasifikasi Journal: Sinta 4

#### **2.2.2.1 Tujuan Penelitian**

Banyak coffee shop modern kini menyediakan layanan informasi kandungan kalori pada setiap varian minuman kopinya. Layanan ini disediakan untuk memenuhi kebutuhan konsumen yang semakin peduli dengan asupan nutrisi dan kesehatan. Konsumen dapat dengan mudah mengetahui jumlah kalori dari setiap minuman kopi yang mereka pilih, sehingga mereka bisa mengambil keputusan yang lebih baik sesuai dengan kebutuhan nutrisi, gaya hidup, atau preferensi diet tertentu, seperti rendah gula atau rendah kalori.

### **2.2.2.2 Metodologi yang Digunakan**

Metodologi yang digunakan dalam penelitian ini adalah analisis kandungan kafein dengan uji laboratorium menggunakan alat spektrofotometer UV-Vis. Proses analisis diawali dengan isolasi kafein dari kopi seduhan, dilanjutkan dengan pembuatan kurva baku standar dan penetapan kadar kafein. Isolasi kafein dilakukan dengan metode ekstraksi menggunakan kloroform, dan analisis kadar kafein dilakukan dengan mengukur absorbansi pada panjang gelombang 281 nm.

### **2.2.2.3 Temuan Utama**

Temuan utama dari penelitian ini menunjukkan bahwa isolasi kafein dari kopi seduhan dilakukan menggunakan metode ekstraksi cair-cair dengan pelarut kloroform, di mana panjang gelombang maksimum kafein adalah 281 nm dengan absorbansi 2,651. Kadar kafein pada kopi Gayo lebih tinggi dibandingkan kopi Sareng, dengan dua dari sepuluh sampel kopi Gayo tidak memenuhi syarat SNI, sedangkan semua sampel kopi Sareng memenuhi syarat. Kadar kafein pada kopi Gayo berkisar antara 51 mg hingga 197.1 mg per sajian, dengan dua sampel melebihi batas SNI 150 mg/hari, sementara kadar kafein pada kopi Sareng berkisar antara 35.3 mg hingga 138.3 mg per sajian. Selain itu, faktor-faktor seperti suhu air seduhan, roasting, dan tempat tumbuh kopi mempengaruhi kadar kafein, dan rekomendasi diberikan untuk membatasi konsumsi kopi Gayo karena kadar kafeinnya yang tinggi.

### **2.2.2.4 Kesimpulan Penelitian**

Kesimpulan penelitian ini menunjukkan bahwa isolasi kafein dari kopi seduhan dapat dilakukan dengan metode ekstraksi cair-cair menggunakan pelarut

kloroform, di mana hasil analisis menunjukkan panjang gelombang maksimum kafein adalah 281 nm dengan absorbansi 2,651; kadar kafein pada kopi Gayo lebih tinggi dibandingkan kopi Sareng, dengan dua dari sepuluh sampel kopi Gayo tidak memenuhi syarat SNI, sedangkan semua sampel kopi Sareng memenuhi syarat; kadar kafein pada kopi Gayo berkisar antara 51 mg hingga 197.1 mg per sajian, dengan dua sampel melebihi batas SNI 150 mg/hari, sementara kadar kafein pada kopi Sareng berkisar antara 35.3 mg hingga 138.3 mg per sajian; faktor-faktor seperti suhu air seduhan, roasting, dan tempat tumbuh kopi mempengaruhi kadar kafein; dan rekomendasi diberikan untuk membatasi konsumsi kopi Gayo karena kadar kafeinnya yang tinggi (Silviana & Santika, 2020).

### **2.2.3 Paper 3**

Judul: Klasifikasi Kematangan Tanaman Hidroponik Pakcoy Menggunakan Metode SVM

Author: Hanin Latif Fuadi, Lukman Priyambodo, Tasya Enjelika Saputri, Naura Nazhifah, Angga Bagus Prawira, Ibrohim Huzaimi, Mas Aly Afandi, Eka Setia Nugraha, Agung Wicaksono, Petrus Kerowe Goran

Publikasi: Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)

Tahun: 2022

Klasifikasi Journal: Sinta 2

#### **2.2.3.1 Tujuan Penelitian**

Berdasarkan permasalahan yang dihadapi, diperlukan suatu metode pendekatan untuk mengelompokkan menu berdasarkan komposisi bahan minuman guna menentukan kadar kalori dalam setiap minuman. Penggunaan algoritma

*Support Vector Machine* (SVM) dapat menjadi solusi yang efektif untuk mengelompokkan minuman berdasarkan bahan dan karakteristiknya.

### **2.2.3.2 Metodologi yang Digunakan**

Berdasarkan permasalahan yang dihadapi, diperlukan suatu metode pendekatan untuk mengelompokkan menu berdasarkan komposisi bahan minuman guna menentukan kadar kalori dalam setiap minuman. Penggunaan algoritma *Support Vector Machine* (SVM) dapat menjadi solusi yang efektif untuk mengelompokkan minuman berdasarkan bahan dan karakteristiknya.

### **2.2.3.3 Temuan Utama**

Temuan utama dari penelitian ini adalah bahwa algoritma *Support Vector Machine* (SVM) dapat digunakan untuk mengklasifikasikan kematangan tanaman hidroponik Pakcoy dengan akurasi 100% pada data pelatihan dan akurasi di atas 79% pada data pengujian. Model SVM berhasil membedakan tanaman Pakcoy menjadi dua kategori: "Kecil" untuk yang belum matang dan "Besar" untuk yang sudah matang. Penelitian ini juga menunjukkan potensi penerapan model dalam sistem tertanam menggunakan *Raspberry Pi* untuk membantu petani hidroponik dalam menentukan waktu panen.

### **2.2.3.4 Kesimpulan Penelitian**

Kesimpulan penelitian ini adalah bahwa algoritma *Support Vector Machine* (SVM) dapat digunakan dengan baik untuk mengklasifikasikan kematangan tanaman hidroponik Pakcoy, dengan mencapai akurasi 100% pada data pelatihan dan akurasi di atas 79% pada data pengujian. Penelitian ini juga menunjukkan

potensi penerapan model dalam sistem tertanam menggunakan Raspberry Pi untuk membantu petani hidroponik dalam menentukan waktu panen (Lukman Priyambodo et al., 2022)

