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 Bencana Banjir

Beberapa ahli mendefinisikan banjir sebagai peristiwa di mana suatu area tergenang air akibat meluapnya air yang melebihi kapasitas sistem pembuangan di suatu wilayah, sehingga menyebabkan kerugian fisik, sosial, dan ekonomi. Banjir merupakan ancaman musiman yang terjadi ketika air dari badan air meluap dari salurannya dan menggenangi wilayah sekitarnya (Permana, 2023).

Banjir juga dikenal sebagai salah satu bencana alam yang paling sering terjadi dan memberikan dampak kerugian terbesar, baik dari segi kemanusiaan maupun ekonomi. Penyebab utamanya adalah curah hujan yang tinggi dan kondisi topografi wilayah yang berupa dataran rendah atau cekungan (Muhamad Taslim et al., 2024).

2.1.2 Dampak Dari Bencana Banjir

Dampak bencana banjir terbagi menjadi dua jenis, yaitu dampak langsung dan tidak langsung. Dampak langsung mencakup kerugian fisik yang terjadi segera setelah bencana, seperti kehancuran atau kerusakan infrastruktur dan properti. Sementara itu, dampak tidak langsung adalah kerugian yang muncul akibat

kerusakan langsung tersebut, mencakup beragam konsekuensi lanjutan seperti gangguan mata pencaharian, kerugian ekonomi, serta kerusakan pada lingkungan (Wismana Putra et al., 2020).

2.1.3 *Data Mining*

Data mining, atau penambangan data, merupakan proses yang bertujuan untuk mengidentifikasi pola dalam data. Dalam literatur, tidak terdapat perbedaan yang jelas antara *data mining* dan machine learning. Beberapa publikasi menyebutkan bahwa *data mining* lebih fokus pada penggalian pola dan hubungan antar data, sementara *machine learning* lebih berorientasi pada pembuatan prediksi. Penambangan data berfungsi untuk menemukan pola dan tren yang berguna dalam kumpulan data yang besar (Barnabas, 2021, p.25).

Di sisi lain, analisis prediktif adalah proses untuk mengekstraksi informasi dari kumpulan data besar guna memprediksi dan memperkirakan hasil di masa mendatang. Dengan demikian, perbedaan antara keduanya terletak pada tujuannya. *Data mining* bertujuan untuk menginterpretasikan data dan menemukan pola yang dapat menjelaskan suatu fenomena (Barnabas, 2021, p. 25).

Contoh pemanfaatan *data mining* yang telah dilakukan seperti pemanfaatan data mining dalam mengoptimalkan layanan transportasi umum telah menjadi fokus penting untuk meningkatkan efisiensi dan kualitas layanan publik. Dengan menganalisis data yang dihasilkan dari sistem tiket elektronik, GPS, dan platform reservasi online, data mining tidak hanya membantu meningkatkan kinerja operasional transportasi umum, tetapi juga mendukung pengambilan keputusan yang lebih berbasis informasi dan tepat waktu (Kurniawan, 2024).

2.1.4 *Clustering*

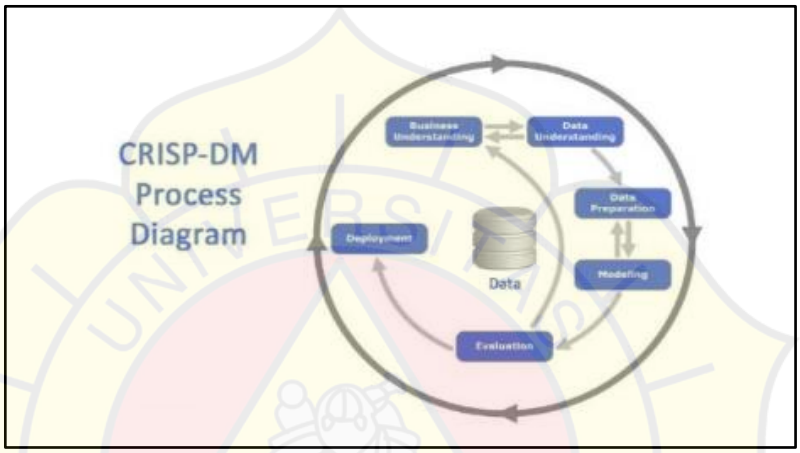
Clustering merupakan metode pengelompokan data ke beberapa cluster atau kelompok, dimana data dalam satu cluster memiliki tingkat kemiripan yang maksimum dan sebaliknya data antar cluster satu dengan yang lain memiliki kemiripan yang minimum, oleh karena itu metode *clustering* sangat berguna untuk menemukan grup atau kelompok yang tidak dikenal dalam data(Teguh, 2020, p. 140).

Clustering adalah metode penganalisisan data, yang sering dimasukkan sebagai salah satu metode data mining, yang tujuannya adalah untuk mengelompokkan data dengan karakteristik yang berbeda ke "wilayah" yang lain (Warmansyah, 2022). Pengertian lainnya mengenai *clustering* merupakan sekumpulan objek data yang memiliki kemiripan satu sama lain dalam satu cluster, tetapi berbeda dengan objek di cluster lain. Objek-objek tersebut dikelompokkan ke dalam satu atau lebih cluster sehingga setiap objek dalam satu cluster memiliki karakteristik yang serupa (Andriani et al., 2024).

Melalui *clustering*, kita dapat mengklasifikasikan area yang padat, mengidentifikasi pola distribusi secara menyeluruh, dan menemukan hubungan menarik antar atribut data. Dalam data mining, fokus utama adalah mengembangkan metode untuk menemukan cluster secara efektif dan efisien pada basis data berukuran besar. Selain itu, terdapat beberapa kebutuhan penting dalam *clustering*, seperti skalabilitas dan kemampuan untuk menangani berbagai jenis atribut data.

2.1.5 CRISP-DM sebagai Tahap Merancang Pengembangan Sistem

Metode implementasi data mining menggunakan kerangka kerja CRISP-DM (*Cross-Industry Standard Process for Data Mining*), yang merupakan metode standar dalam penerapan *data mining* untuk industri (Larose, 2006), terdiri dari beberapa fase, yaitu: (Firmansyah & Yulianto, 2021)



Gambar 2. 1 Metode Crisp-dm (Sumber: (Wisnu Murti, 2024))

1. Fase Pemahaman Bisnis

   Pada fase ini, tujuan yang ingin dicapai oleh organisasi didefinisikan, lalu dikonversi menjadi formula data mining dengan strategi yang dirancang untuk mencapai tujuan tersebut (Kandias et al., 2024).

2. Fase Pemahaman Data

   Aktivitas pada fase ini melibatkan pengumpulan data dari aktivitas organisasi. Data yang relevan dikumpulkan dan dipastikan siap untuk dianalisis (Fadil Danu Rahman et al., 2024).

3. Fase Persiapan Data

   Tahap ini berfokus pada persiapan data, termasuk analisis dan pembersihan data dari redundansi atau data kosong, serta penghapusan kolom dan baris

yang tidak diperlukan, sehingga data yang akan dimasukkan ke tahap pemodelan sesuai kebutuhan (Rossa Amelia Manik & Atik Ariesta, 2023).

4. Fase Pemodelan

   Pada fase ini, algoritma yang sesuai dipilih dan diterapkan pada data. Eksperimen dengan beberapa model algoritma mungkin diperlukan untuk mendapatkan hasil optimal. Metode yang paling sesuai dengan tujuan bisnis dari fase pertama akan dipilih (Iswavigra et al., 2023).

5. Fase Evaluasi

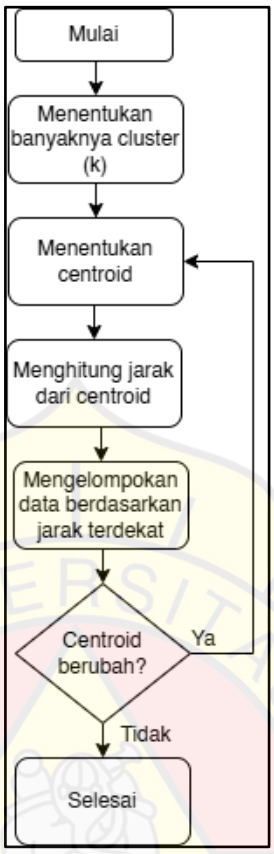
   Hasil dari fase pemodelan kemudian dievaluasi untuk memastikan kesesuaiannya dengan tujuan bisnis yang ditetapkan pada fase awal, guna melihat apakah tujuan tersebut telah tercapai (Sarimole & Hakim, 2024).

6. Fase Penerapan

   Setelah melalui fase evaluasi, algoritma yang telah dikembangkan dapat diimplementasikan, baik dalam bentuk laporan atau aplikasi pendukung keputusan berbasis sumber terbuka (Adelia Irawan et al., 2024).

2.1.6 Algoritma K-Means

Algoritma K-Means adalah metode pengelompokan tanpa supervisi yang berfungsi untuk mengelompokkan sejumlah data yang tidak berlabel ke dalam beberapa "k" cluster. Huruf "k" pada K-Means menunjukkan jumlah cluster yang diinginkan sebagai hasil akhir dari proses pengelompokan (Barnabas, 2021, pp.68-69). Berikut adalah langkah-langkah yang akan digunakan dalam metode K-Means untuk menentukan jumlah cluster dan penempatan data dalam cluster (Kaligis & Yulianto, 2022) yang ada pada gambar 2.2 di bawah ini:



Gambar 2. 2 Algoritma K-Means (Sumber: (Chandra et al., 2021))

1. Langkah pertama dilakukan dengan memilih k data secara acak sebagai pusat cluster.
2. Untuk menentukan jarak antara data dan pusat cluster, Perhitungan jarak antara setiap data ke masing-masing pusat cluster. Dengan menggunakan beberapa metode perhitungan jarak yaitu:

   Euclidean Distance

   Euclidean Distance digunakan untuk mengukur seberapa mirip atau dekat jarak antara data, dengan menggunakan rumus Euclidean (Pangestu & Fitriani, 2022).

$$d(x,y) = |x - y|\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Keterangan:

i : index dari atribut

n : jumlah data

x_i : atribut dari data ke-i (i = 1, 2, 3, ..., …, n)

y_i : atribut dari pusat Cluster ke-i (i = 1, 2, 3, ..., …, n)

Manhattan Distance

Manhattan Distance digunakan sebagai metode untuk mengukur jarak yang diperoleh dari jumlah perbedaan nilai antara dua objek data.

$$d(x, y) = \sum_{k=1}^{n} |x_i - y_i| \quad (2)$$

Keterangan:

d : jarak antara x dan y

x_i : atribut dari data ke-i (i = 1, 2, 3, ..., …, n)

y_i : atribut dari pusat Cluster ke-i (i = 1, 2, 3, ..., …, n)

Minkowski Distance

Minkowski Distance adalah sebuah metrik dalam ruang vektor yang memiliki norma (*normed vector space*) dan dianggap sebagai generalisasi dari *Euclidean Distance* dan *Manhattan Distance* (Sinaga et al., 2023).

$$d(x, y) = (\sum_{k=1}^{n} |xi - yi|p)\frac{1}{p} \quad (3)$$

Keterangan:

d : jarak antara x dan y

x : data pusat cluster

y : data atribut

i : setiap data

n : jumlah data

x_i : atribut dari data ke-i (i = 1, 2, 3, ..., …, n)

y_i : atribut dari pusat Cluster ke-i (i = 1, 2, 3, ..., …, n)

p : power

3. Data yang paling dekat dengan suatu pusat cluster akan dimasukkan ke dalam cluster tersebut, dan posisi pusat cluster akan diperbarui berdasarkan rata-rata posisi data dalam cluster. Proses ini akan mengatur data dalam cluster baru setelah semua data telah dialokasikan ke cluster terdekat.
4. Pada langkah terakhir, proses penentuan pusat cluster akan diulang hingga posisi centroid tidak mengalami perubahan.

2.1.6 Permodelan Sistem UML

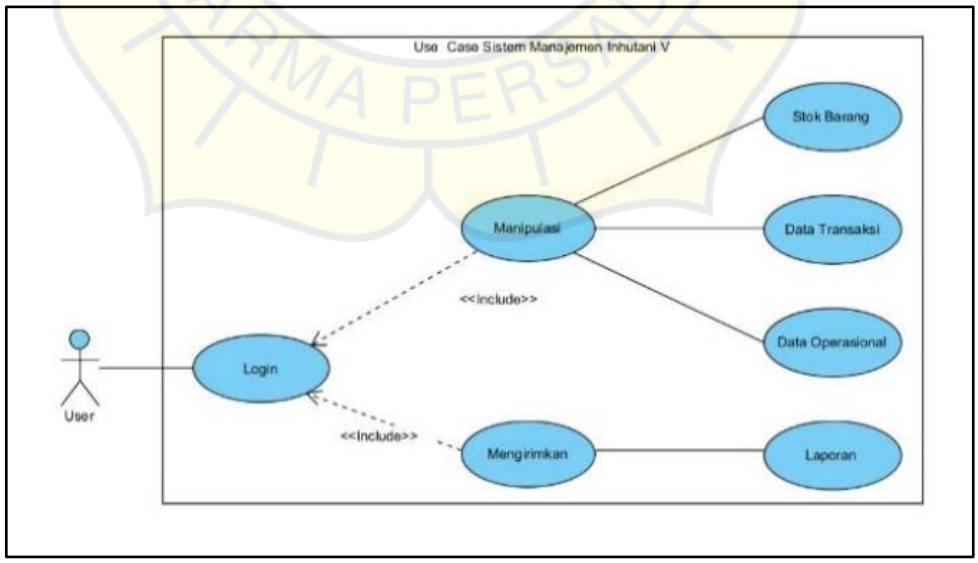
Permodelan sistem menggunakan *Unified Modeling Language* sebagai perancangan sistem perangkat lunak yang membantu dalam memvisualisasikan struktur dan alur kerja sistem. Pada sub bab ini akan menjelaskan mengenai pengertian dari UML dan diagram-diagram seperti *use case* dan *activity*.

2.1.6.1 Pengertian UML

UML, singkatan dari *Unified Modeling Language*, merujuk pada seperangkat alat yang sering digunakan untuk membuat representasi abstrak dari sistem atau perangkat lunak berbasis objek. Selain sebagai bahasa standar untuk menggambarkan model sistem, UML juga berperan penting dalam mempermudah pengembangan aplikasi yang terus berlanjut. Selain itu, UML juga berfungsi sebagai sarana untuk mentransfer pengetahuan tentang sistem atau aplikasi dari satu pengembang ke pengembang lainnya

2.1.6.2 *Use Case Diagram*

Diagram *Use Case* menggambarkan siapa yang terlibat dalam sistem (Aktor) dan apa yang dilakukan oleh sistem (*Use Case*) (Wijayanti et al., 2022). *Use case* diagram adalah representasi visual dari beberapa komponen, seperti aktor, *use case*, dan hubungan antar komponen. Diagram ini menggunakan berbagai simbol atau notasi untuk menggambarkan fungsionalitas suatu sistem. Dengan *use case* diagram, analis dapat lebih mudah merancang kebutuhan (requirement) untuk pengembangan sistem. Selain itu, *use case* diagram juga digunakan untuk menjelaskan desain sistem kepada pengguna serta merancang semua fitur yang akan diimplementasikan dalam sistem tersebut (Siska Narulita et al., 2024). Contoh *use case* diagram dapat dilihat pada gambar di bawah ini yang menjelaskan tentang aktivitas login antara pengguna dengan sistem manajemen melibatkan berbagai aktivitas, seperti melakukan login, memasukkan data transaksi, mencatat data operasional, mengelola stok barang, serta menyusun laporan (Hafsari et al., 2023).

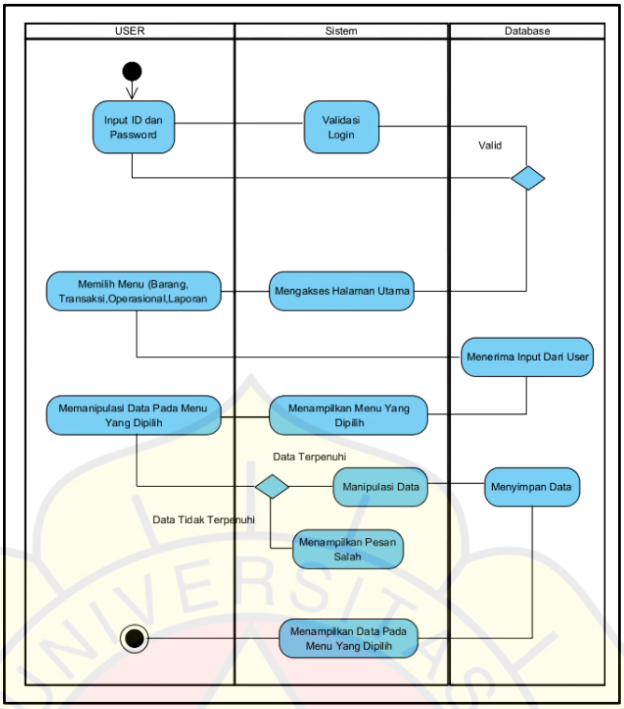


Gambar 2. 3 *Use Case* Diagram (Sumber: (Hafsari et al., 2023))

2.1.6.3 *Activity Diagram*

Diagram aktivitas memberikan representasi visual tentang berbagai aktivitas yang terjadi dalam sistem yang sedang dikembangkan, termasuk bagaimana setiap aktivitas dimulai dari keputusan yang mungkin diambil, serta bagaimana setiap aktivitas berakhir (Wijayanti et al., 2022). Diagram ini juga memvisualisasikan proses-proses paralel yang terjadi selama sistem dijalankan. Setiap tahapan atau langkah dalam sistem direpresentasikan melalui activity diagram. Umumnya, setiap *use case* memiliki setidaknya satu *activity* diagram yang dirancang berdasarkan satu atau beberapa *use case* pada *use case* diagram. Jika use case menunjukkan bagaimana aktor menggunakan sistem untuk melakukan suatu aktivitas, *activity* diagram merepresentasikan detail alur proses yang terjadi di dalam sistem tersebut (Siska Narulita et al., 2024).

Contoh *activity* diagram dapat dilihat pada gambar di bawah ini yang menjelaskan pengguna memulai dengan memasukkan ID dan password, yang akan divalidasi oleh sistem. Jika data salah, pengguna diarahkan kembali ke halaman login hingga data benar. Setelah login berhasil, pengguna dapat memilih menu di halaman utama, dan sistem akan menampilkan halaman sesuai pilihan. Saat pengguna memanipulasi data, sistem memeriksa kelengkapan data tersebut. Jika valid, data disimpan ke database dan halaman diperbarui. Jika tidak valid, sistem menampilkan pesan kesalahan (Hafsari et al., 2023).



Gambar 2. 4 *Activity Diagram* (Sumber: (Hafsari et al., 2023))

2.1.7 *Software* dan Pemrogaman Terkait

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

2.1.7.1 Editor Google Colabs

Google Colab memungkinkan pengguna menulis dan menjalankan kode Python langsung di browser, sehingga sangat cocok untuk kebutuhan *machine learning* dan analisis data. Platform ini mempermudah akses ke sumber daya komputasi yang kuat, serta mendukung penyimpanan dan berbagi notebook secara online dengan praktis (Fadil Danu Rahman et al., 2024).

2.1.7.2 Pemrogaman Python

Python adalah bahasa pemrograman berorientasi objek yang mendukung penggunaan interaktif dan menawarkan struktur data tingkat tinggi, sekaligus berfungsi sebagai bahasa interpretatif dengan beragam fitur. Bahasa ini dirancang untuk memprioritaskan kejelasan dan kemudahan pembacaan kode, sehingga ideal bagi para developer yang mengutamakan sintaks yang ringkas dan jelas (Triono et al., 2023). Python juga mendukung library seperti *Pandas*, yang digunakan untuk manipulasi dan analisis data. *NumPy* untuk operasi matematika dan manipulasi *array*, *Matplotlib* dan *Seaborn* untuk membuat visualisasi data yang menarik dan informatif, serta *Scikit-learn* yang menyediakan berbagai algoritma *machine learning* (Farhanuddin et al., 2024).

2.1.7.3 *Library Pandas*

Python *Panda*s adalah pustaka sumber terbuka yang sangat populer untuk analisis dan manipulasi data, menyediakan alat yang efisien dan kuat untuk bekerja dengan data terstruktur. Pustaka ini memungkinkan pengguna untuk melakukan pembersihan, prapemrosesan, dan transformasi data dengan mudah, menjadikannya sangat membantu dalam penelitian di berbagai bidang. Dalam analisis data, *Pandas* memudahkan peneliti untuk menangani data dalam jumlah besar, termasuk proses memfilter, menggabungkan, dan meringkas data. Hal ini sangat berguna di bidang seperti keuangan, kesehatan, dan ilmu sosial, di mana kumpulan data yang besar sering digunakan untuk mengidentifikasi pola, tren, dan hubungan (Lavanya et al., 2023).

Selain itu, *Pandas* menawarkan kemampuan visualisasi data melalui berbagai jenis plot, bagan, dan grafik, yang membantu peneliti memahami data mereka secara lebih mendalam dan menyampaikan temuan dengan lebih efektif. Untuk analisis deret waktu, Pandas menyediakan seperangkat alat yang kuat, termasuk pengindeksan berbasis waktu, resampling, dan operasi *rolling window*, yang bermanfaat dalam penelitian di bidang keuangan dan ekonomi (Lavanya et al., 2023).

Dalam hal pembersihan dan prapemrosesan data, *Pandas* menawarkan berbagai fungsi untuk menangani data yang hilang, menghapus duplikat, dan mengubah data, membantu peneliti mempersiapkan data mereka dengan akurat dan berkualitas tinggi. *Pandas* juga terintegrasi dengan baik dengan pustaka analisis data dan pembelajaran mesin lainnya seperti *NumPy*, *Scikit-learn*, dan *Matplotlib*, sehingga mudah digunakan dalam pipeline penelitian yang lebih besar. Kombinasi alat yang lengkap ini menjadikan *Pandas* pilihan yang berharga dan fleksibel untuk peneliti di berbagai bidang ilmiah (Lavanya et al., 2023). Contoh penerapan *library Pandas* dapat dilihat pada gambar 2.5 dibawah ini:

```
import pandas as pd

mydataset = {
  'cars': ["BMW", "Volvo", "Ford"],
  'passings': [3, 7, 2]
}

myvar = pd.DataFrame(mydataset)

print(myvar)
```

Gambar 2. 5 *Library Pandas* (Sumber: w3schools)

2.1.7.4 *Library NumPy*

NumPy (*Numerical Python*) adalah *library* dalam Python yang dirancang untuk komputasi numerik dan menyediakan objek larik N-dimensi yang sangat efisien. Sebagai pustaka pemrosesan larik untuk keperluan umum, *NumPy* memungkinkan pengguna bekerja dengan larik multidimensi berkinerja tinggi melalui berbagai fungsi dan operator yang dioptimalkan untuk kecepatan, mengatasi masalah kelambatan dalam pemrosesan data (Salah & Din, 2020).

Fitur utama *NumPy* mencakup penyediaan fungsi yang cepat dan terkompilasi untuk rutinitas numerik, komputasi berbasis larik yang efisien, dukungan terhadap pendekatan berorientasi objek, dan kemampuan vektorisasi untuk komputasi yang lebih ringkas dan cepat. *NumPy* juga sangat bermanfaat dalam analisis data, memungkinkan pembuatan larik N-dimensi yang kuat dan menjadi fondasi pustaka lain, seperti *SciPy* dan *scikit-learn*. Dengan dukungan pustaka tambahan seperti *SciPy* dan *Matplotlib*, *NumPy* bahkan sering digunakan sebagai pengganti MATLAB dalam berbagai aplikasi ilmiah dan teknis (Salah & Din, 2020). Contoh penerapan *library NumPy* dapat dilihat pada 2.6 gambar dibawah ini:

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)
```

Gambar 2. 6 *Library NumPy* (Sumber: w3schools)

2.1.7.5 *Library Matplotlib*

Matplotlib adalah salah satu *library* visualisasi data paling populer dalam Python, dikembangkan oleh John Hunter bersama dengan kontribusi dari banyak rekan lainnya. *Library* ini dirancang untuk memfasilitasi penggunaan di kalangan ilmuwan dan peneliti di seluruh dunia. Sebagai library grafis yang berperan penting dalam ekosistem ilmu data Python, *Matplotlib* mendukung integrasi dengan pustaka terkait seperti *NumPy* dan *Pandas*, menjadikannya alat yang penting dalam proses analisis data. *Matplotli*b memungkinkan pengguna membuat visualisasi interaktif yang dapat dimanipulasi secara *real-time*, sehingga memudahkan eksplorasi data serta identifikasi pola dan tren. Selain itu, *Matplotlib* mempermudah pembuatan visualisasi berkualitas publikasi yang dapat direproduksi, menjadikannya sangat bermanfaat untuk penyajian dalam makalah penelitian, presentasi, dan materi lainnya. Dengan integrasinya yang mudah dengan pustaka lain seperti *NumPy*, *Pandas*, dan *SciPy*, *Matplotlib* berperan penting dalam membantu peneliti menganalisis serta mengomunikasikan temuan mereka secara efektif melalui visualisasi data yang jelas dan informatif (Hafeez & Sial, 2021). Contoh penerapan *library Matplotlib* dapat dilihat pada gambar 2.7 dibawah ini:

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

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

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

Gambar 2. 7 *Library Matplotlib* (Sumber: w3schools)

2.1.7.6 *Library Scikit-learn*

Scikit-learn adalah *library* yang populer di Python, banyak digunakan untuk membuat model prediktif, serta menyediakan alat untuk prapemrosesan data, pemilihan model, dan evaluasi model. Library ini mendukung berbagai metode penggalian data dan analisis yang memungkinkan pengguna untuk menangani masalah kompleks seperti klasifikasi gambar dan teks, regresi, dan pengelompokan (Fahmi, 2023).

Dikembangkan melalui kontribusi dari banyak peneliti di seluruh dunia, *Scikit-learn* menawarkan kemudahan dalam implementasi machine learning di Python melalui API yang sederhana dan intuitif, menjadikannya pilihan populer di kalangan profesional industri dan akademisi. Library ini dirancang di atas modul *NumPy* dan *SciPy*, sehingga memiliki performa yang optimal dan efisien dalam perhitungan numerik. Meski demikian, *Scikit-learn* memiliki keterbatasan dalam menangani data yang sangat besar, sehingga kurang disarankan untuk digunakan pada proyek dengan ukuran data yang masif (Fahmi, 2023). Contoh penerapan *library Scikit-Learn* dapat dilihat pada gambar 2.8 dibawah ini:

```
from sklearn.cluster import KMeans

data = list(zip(x, y))
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```

Gambar 2. 8 *Library Scikit-Learn* (Sumber: w3schools)

2.1.7.7 *Library Seaborn*

Seaborn adalah *library* dalam Python yang berfungsi untuk membuat grafik statistik, dirancang sebagai antarmuka tingkat tinggi untuk *Matplotlib* dan terintegrasi dengan baik dengan struktur data *Pandas*. *Seaborn* memudahkan pengguna menerjemahkan pertanyaan mengenai data menjadi visualisasi yang informatif, dengan API deklaratif dan berbasis dataset. Saat diberikan dataset beserta spesifikasi plot, *Seaborn* secara otomatis memetakan nilai data ke atribut visual seperti warna, ukuran, dan gaya, serta menghitung transformasi statistik yang diperlukan. Selain itu, Seaborn menghias plot dengan label sumbu dan legenda yang informatif. Banyak fungsi dalam Seaborn mampu menghasilkan grafik dengan beberapa panel, memungkinkan perbandingan antara subset data atau antar-pasangan variabel yang berbeda. Seaborn sangat berguna sepanjang siklus proyek ilmiah—mulai dari tahap pembuatan prototipe cepat hingga analisis data

eksploratif, berkat kemampuannya menghasilkan grafik yang lengkap hanya dengan sedikit argumen. Dengan berbagai opsi kustomisasi dan akses ke objek *Matplotlib* yang mendasarinya, *Seaborn* juga memungkinkan pembuatan visualisasi berkualitas publikasi yang professional (Waskom, 2021). Contoh penerapan *library Seaborn* dapat dilihat pada gambar 2.9 dibawah ini:

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.normal(size=1000), hist=False)

plt.show()
```

Gambar 2. 9 *Library Seaborn* (Sumber: w3schools)

2.1.7.3 *Streamlit*

Streamlit adalah *framework* yang digunakan untuk mengembangkan aplikasi web untuk analisis data. Dengan memanfaatkan bahasa pemrograman Python, *Streamlit* memungkinkan transformasi kode menjadi aplikasi web secara mudah. Framework ini membantu para *developer* dalam membangun aplikasi berbasis web di bidang *data science* dan *machine learning*, membuat proses pengembangan menjadi lebih efisien (Bagdja et al., 2024).

2.2 Tinjauan Literatur/Kajian Penelitian Terdahulu

Tinjauan literatur atau kajian penelitian sebelumnya dilakukan untuk membangun dasar teori yang kokoh dan memastikan bahwa penelitian yang dilakukan memberikan kontribusi baru bagi pengembangan ilmu pengetahuan. Kajian ini melibatkan berbagai penelitian yang berkaitan dengan pengelompokan

wilayah rentan bencana, penerapan algoritma K-Means, serta penggunaan data mining dalam upaya mitigasi bencana.

2.2.1 Paper 1

Judul: Pengelompokan Desa Menggunakan K-Means untuk Penyelenggaraan Penanggulangan Bencana Banjir

Author: Shelladita Fitriyani Susilo, Asep Jamaludin, Intan Purnamasari

Publikasi: *Journal of Information System* (JOINS)

Tahun: 2020

Klasifikasi Journal: Sinta 4

2.2.1.1 Tujuan Penelitian

Penelitian ini bertujuan untuk mengelompokkan desa rentan banjir di Kabupaten Tegal menggunakan metode K-Means, dengan hasil optimal 7 cluster menurut metode *elbow*. Hasilnya diharapkan membantu BPBD dalam strategi penanggulangan, meningkatkan respons bencana, dan memberikan rekomendasi untuk strategi manajemen banjir yang lebih efektif.

2.2.1.2 Metodelogi Yang Digunakan

Penelitian ini menggunakan algoritma K-Means untuk mengelompokkan desa di Kabupaten Tegal berdasarkan data banjir. Metodologi mencakup pengumpulan data sekunder, pra-pemrosesan, penentuan jumlah cluster optimal (7 cluster) dengan metode elbow, penerapan K-Means, dan analisis hasil untuk meningkatkan perencanaan serta respons banjir di wilayah tersebut.

2.2.1.3 Temuan Utama

Penelitian ini menunjukkan bahwa algoritma K-Means berhasil mengelompokkan 74 desa di Kabupaten Tegal menjadi 7 cluster terkait banjir, dengan metode *elbow* yang terbukti lebih efektif dibandingkan *silhouette*. Setiap cluster memiliki karakteristik unik, seperti kepadatan penduduk dan lahan resapan, yang memengaruhi risiko banjir. Temuan ini memberikan dasar bagi BPBD untuk merencanakan strategi penanggulangan banjir yang lebih terfokus dan efektif.

2.2.1.4 Kesimpulan Penelitian

Metode K-Means efektif untuk mengelompokkan data laporan kejadian banjir, dengan pemanfaatan metode *elbow* yang menghasilkan tujuh cluster, masing-masing mencerminkan karakteristik risiko dan kepadatan penduduk yang berbeda. Hal ini menunjukkan bahwa pemetaan yang akurat dapat mendukung upaya penanggulangan bencana di daerah yang terkena dampak (Susilo et al., 2020).

2.2.2 Paper 2

Judul: Analysis Prediksi Wilayah Rawan Banjir dengan Algoritma K-Means

Author: Muhammad Makmum Effendi, Inka, Arif Siswandi

Publikasi: *Journal of Information System Research* (JOSH)

Tahun: 2024

Klasifikasi Journal: Sinta 4

2.2.2.1 Tujuan Penelitian

Penelitian ini bertujuan untuk menerapkan algoritma K-Means dalam pengelompokan data banjir di Kota Bekasi berdasarkan faktor-faktor penyebab

seperti curah hujan dan jumlah titik banjir. Penelitian ini ditargetkan menghasilkan tiga cluster yang mengkategorikan tingkat banjir sebagai tinggi, sedang, dan rendah, serta menganalisis kualitas cluster tersebut menggunakan *Davies-Bouldin Index.*

2.2.2.2 Metodelogi Yang Digunakan

Metodologi penelitian ini mencakup pengumpulan data curah hujan dan titik banjir di 50 kecamatan di Kota Bekasi (Januari–Oktober 2022), pra-pemrosesan data, dan pengelompokan menggunakan algoritma K-Means ke dalam tiga kategori banjir: tinggi, sedang, dan rendah. Kualitas hasil dievaluasi dengan *Davies-Bouldin Index*, dan hasil analisis disajikan untuk memahami pola banjir dan mendukung mitigasi di Bekasi.

2.2.2.3 Temuan Utama

Penelitian ini mengidentifikasi tiga cluster risiko banjir di Bekasi tinggi, sedang, dan rendah berdasarkan data banjir 2022, dengan pengaruh *La Nina* yang meningkatkan risiko. Algoritma K-Means menghasilkan pengelompokan berkualitas baik, dengan nilai *Davies-Bouldin Index* -0,452. Temuan ini menekankan pentingnya solusi teknologi untuk menyampaikan informasi risiko banjir, membantu kesiapsiagaan masyarakat, dan mendukung pengelolaan risiko banjir di Bekasi.

2.2.2.4 Kesimpulan Penelitian

Penerapan algoritma K-Means dalam pengelompokan dataset banjir tahun 2022 menghasilkan tiga klaster, yaitu kategori banjir tinggi, sedang, dan rendah. Hasil pengujian menunjukkan nilai *Davies Bouldin Index* sebesar -0,452, yang

menandakan bahwa semakin kecil nilai tersebut, semakin baik kualitas klaster yang dihasilkan (Effendi & Siswandi, 2024).

2.2.3 Paper 3

Judul: Penentuan Titik Lokasi Daerah Rawan Banjir Di Kabupaten Malaka Menggunakan Metode K-Means Clustering

Author: Fransiskus Xavierus Moruk, Vito Daniel Boboy, Wilhelmina Johana Tahuk, Yota Putra Kamirsa, Yampi R Kaesmetan

Publikasi: Jurnal Sistem Informasi dan Informatika (SIMPATIK)

Tahun: 2023

Klasifikasi Journal: Sinta 5

2.2.3.1 Tujuan Penelitian

Penelitian ini bertujuan untuk memetakan daerah rawan banjir di Kabupaten Malaka menggunakan metode K-Means Clustering dan perangkat lunak Quantum GIS, dengan mengidentifikasi serta mengelompokkan wilayah berdasarkan risiko banjir melalui analisis data seperti curah hujan, jenis tanah, kemiringan, dan peta administrasi.

2.2.3.2 Metodelogi Yang Digunakan

Metodologi penelitian ini mencakup pengumpulan data peta administrasi, curah hujan, jenis tanah, dan lereng di Kabupaten Malaka, pra-pemrosesan data dengan Quantum GIS, serta penerapan K-Means Clustering untuk mengelompokkan wilayah berdasarkan risiko banjir. Hasilnya berupa peta daerah rawan banjir yang bermanfaat bagi masyarakat dan mendukung manajemen bencana.

2.2.3.3 Temuan Utama

Penelitian ini menemukan bahwa K-Means Clustering efektif memetakan daerah rawan banjir di Kabupaten Malaka dalam tiga tingkat risiko. Pemetaan ini membantu mengidentifikasi lokasi berisiko tinggi, mendukung manajemen bencana, dan meningkatkan kesadaran masyarakat akan risiko banjir.

2.2.3.4 Kesimpulan Penelitian

Penerapan metode K-Means Clustering dapat digunakan untuk memetakan daerah rawan banjir di Kabupaten Malaka. Hasil pengelompokan menunjukkan adanya klaster yang membantu dalam mengidentifikasi lokasi-lokasi berisiko tinggi terhadap banjir. Hal ini penting untuk pengembangan sistem informasi geografis (SIG) yang dapat menyediakan informasi akurat kepada masyarakat dan mendukung pengambilan keputusan terkait penanganan risiko banjir di wilayah tersebut (Moruk et al., 2024).

2.2.4 Paper 4

Judul: Pengelompokkan Data Bencana Alam Berdasarkan Wilayah, Waktu, Jumlah Korban dan Kerusakan Fasilitas Dengan Algoritma K-Means

Author: Murdiaty, Angela, Chatrine Sylvia

Publikasi: Jurnal Media Informatika Budidarma

Tahun: 2020

Klasifikasi Journal: Sinta 3

2.2.4.1 Tujuan Penelitian

Penelitian ini bertujuan untuk mengumpulkan dan menganalisis data bencana alam di Indonesia dari tahun 2014 hingga 2018, serta menerapkan teknik

data mining, khususnya algoritma K-Means, untuk mengelompokkan data bencana berdasarkan karakteristik tertentu, guna mengurangi dampak bencana dan memprediksi kejadian bencana di masa mendatang.

2.2.4.2 Metodelogi Yang Digunakan

Peneliti mengumpulkan dan menganalisis data bencana alam di Indonesia dari tahun 2014 hingga 2018 dengan menggunakan algoritma K-Means dalam kerangka kerja CRISP-DM. Penelitian ini mengelompokkan provinsi berdasarkan jumlah kejadian dan korban bencana, mengidentifikasi dua kluster utama yang mencerminkan tingkat kerentanan terhadap bencana, dan memberikan rekomendasi untuk penelitian mendatang guna memperkuat strategi manajemen bencana dan memprediksi kejadian di masa depan.

2.2.4.3 Temuan Utama

Penelitian ini menemukan bahwa pengelompokan data bencana alam di Indonesia dari tahun 2014 hingga 2018 dengan algoritma K-Means menghasilkan dua kluster utama: satu dengan tingkat korban tinggi akibat bencana seperti banjir dan tanah longsor, dan kluster lainnya dengan korban gempa bumi yang lebih dominan. Temuan ini memberikan wawasan untuk memperkuat strategi manajemen bencana dan prediksi kejadian di masa mendatang.

2.2.4.4 Kesimpulan Penelitian

Penelitian ini menganalisis data bencana alam di Indonesia dari tahun 2014 hingga 2018 menggunakan algoritma K-Means untuk mengelompokkan provinsi berdasarkan frekuensi kejadian dan jumlah korban bencana, menghasilkan dua kluster utama yang mencerminkan tingkat kerentanan terhadap bencana. Studi ini

juga memberikan rekomendasi untuk penelitian selanjutnya guna meningkatkan strategi manajemen bencana dan prediksi kejadian di masa depan (Murdiaty et al., 2020).

2.2.5 Paper 5

Judul: Prediksi Banjir Di Dki Jakarta Dengan Menggunakan Algoritma K-Means Dan Random Forest

Author: Ruby Haris, Wasis Haryo, Eka Wahyu Pujiharto, Adela Yuza, Kusrini, Kusnawi

Publikasi: Jurnal Informatika Dan Teknologi Komputer (J-ICOM)

Tahun: 2024

Klasifikasi Journal: Sinta 4

2.2.5.1 Tujuan Penelitian

Penelitian ini bertujuan mengembangkan metode prediksi banjir di DKI Jakarta dengan menggabungkan algoritma K-Means dan Random Forest, untuk meningkatkan efektivitas pencegahan dan mitigasi banjir menggunakan data historis pintu air dan tinggi air. Selain itu, penelitian ini diharapkan dapat berkontribusi pada pengelolaan banjir, perencanaan kota tahan banjir, dan pengembangan teknologi prediksi banjir yang lebih canggih.

2.2.5.2 Metodelogi Yang Digunakan

Penelitian ini menggunakan metode kuantitatif. Data historis pintu air dan tinggi air dari Open Data Jakarta untuk periode Januari-Desember 2020 dikumpulkan, dibersihkan, dan variabel penting dipilih. Data kemudian dinormalisasi dan diubah untuk analisis *machine learning*. Dataset dibagi 80%

untuk pelatihan dan 20% untuk pengujian. Algoritma K-Means digunakan untuk clustering dan Random Forest untuk prediksi banjir, guna menghasilkan model prediksi banjir yang efektif untuk mitigasi di DKI Jakarta.

2.2.5.3 Temuan Utama

Penelitian ini menunjukkan bahwa kombinasi K-Means dan Random Forest menghasilkan model prediksi banjir dengan akurasi 95% dan f-1 score 90% pada k=14. Clusterisasi tinggi air dengan K-Means memperbaiki kualitas data, sementara Random Forest efektif memprediksi banjir. Penelitian ini berkontribusi pada pengelolaan banjir di DKI Jakarta dan menyarankan penambahan variabel untuk peningkatan akurasi model.

2.2.5.4 Kesimpulan Penelitian

Penelitian ini menyimpulkan bahwa kombinasi algoritma K-Means dan Random Forest secara efektif meningkatkan akurasi prediksi banjir hingga 95% di DKI Jakarta, memberikan clusterisasi optimal dan wawasan pola data, serta menawarkan kontribusi signifikan untuk pengelolaan banjir dan perencanaan kota yang lebih tanggap, disarankan untuk menambahkan variabel lain di penelitian selanjutnya guna meningkatkan performa model (Sasoko Wasis Haryo et al., 2024).