BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Pada penelitian Ahmad Efendi dan rekan-rekan lainnya yang berjudul "Klasifikasi Kebakaran Hutan Riau Menggunakan Random Forest dan Visualisasi Citra Sentinel-2" menjelaskan pada bulan September 2019, Riau mengalami dampak serius dari kabut asap yang membahayakan kesehatan jiwa dan menghambat kegiatan sehari-hari kurang lebih 6,5 juta warganya. Keadaan ini mendesak perlunya respons cepat dan tepat dalam upaya mitigasi dan pencegahan bencana kebakaran hutan serta lahan. Ahmad Efendi dan rekan-rekan lainnya menggunakan *Data Mining* menggunakan algoritma *Random Forest* dengan visualisasi menggunakan citra Sentinel-2 dengan indeks *Normalized Burn Ratio* (NBR). Dalam proses pengumpulan data, Ahmad Efendi dan rekan-rekan lainnya mengakses data citra melalui website resmi *EO Browser*. Dalam implementasinya, peneliti menggunakan bahasa pemrograman *python*, kemudian dibantu dengan aplikasi Microsoft Excel dalam proses *Data Integration* dan QGIS dalam pengolahan data visualisasi yang sudah dihasilkan [3].

Penelitian lainnya adalah penelitian yang dilakukan oleh Anggita Ghozali dan rekan-rekannya yang berjudul "Implementasi *Data Mining* Menggunakan Metode *Random Forest* dan *Support Vector Machine* Dalam Klasifikasi Penyakit Diabetes". Pada penelitian ini, peneliti bertujuan untuk mengklasifikasi apakah seorang pasien terkena diabetes atau tidak berdasarkan variabel-variabel yang diproses dengan *Data Mining* dan untuk mengetahui perbandingan nilai *accuracy*,

presicion, recall, specificity, dan *F1-score* antara penggunaan metode *Random Forest* dan *Support Vector Machine* sehingga mengetahui mana metode yang lebih baik dari keduanya [4]. Kedua penelitian ini sama sama menggunakan metode *Random Forest* dalam penyelesaian masalahnya, pada penelitian yang dilakukan Ahmad Efendi dan rekan-rekannya mengaplikasikan *Random Forest* untuk mendapatkan hasil klasifikasi dan prediksi dan QGIS untuk menvisualisasikan hasilnya yang diharapkan dapat membantu dalam mengatasi kebakaran hutan di Riau. Pada penelitian yang dilakukan Anggita Ghozali dan rekan-rekannya pun sama sama menggunakan *Random Forest*, hanya saja memiliki tujuan yang berbeda yaitu untuk mengklasifikasi pasien terjangkit penyakit diabetes atau tidak, dan juga untuk melakukan perbandingan dengan menggunakan metode *Support Vector Machine* sebagai pembanding, sehingga menghasilkan informasi metode mana yang lebih baik dalam mengklasifikasi data berdasarkan perhitungan menggunakan *Confusion Matrix.* Dalam penelitian yang dilakukan oleh Ronny Susetyoko dan rekan-rekannya, Widodo, Handoyo, dan Faruk menyatakan bahwa SVM memiliki performa klasifikasi yang lebih unggul dibandingkan dengan Regresi Logistik (Widodo & Handoyo, 2017). Namun, dalam penelitian yang dilakukan oleh Faruk dan rekan-rekannya (2018), model Regresi Logistik Biner menunjukkan kinerja yang baik dalam prediksi, tetapi kurang optimal dalam klasifikasi. Sebaliknya, Random Forest menunjukkan performa yang sangat baik untuk prediksi maupun klasifikasi [5].

2.2. Teori Pendukung

Teori pendukung merupakan kumpulan teori-teori atau definisi-definisi apa saja yang digunakan pada penelitian ini.

2.2.1. Data Mining

Data Mining merupakan metode pengolahan kumpulan data besar untuk mengungkap pola, koneksi, dan wawasan berharga. Melalui penerapan teknik statistik dan algoritma, proses ini bertujuan untuk menyingkap tren dan hubungan dalam data yang belum terdeteksi sebelumnya. Dengan demikian, memungkinkan transformasi data mentah menjadi kecerdasan bisnis yang signifikan dan dapat diaplikasikan [6]. Menurut J. Hutagalung dan F. Sonata pada jurnal Fiqal Kana yang berjudul "Implementasi Data Mining Menganalisa Pola Penjualan Rempah-Rempah Menggunakan Metode Fp-Growth", Data mining merupakan proses analisis data yang bertujuan untuk mengidentifikasi pola-pola tertentu dalam kumpulan data yang luas. Tujuannya adalah untuk menghasilkan informasi yang relevan, yang kemudian dapat dimanfaatkan dan dikembangkan lebih jauh [7].

Menurut Dito Putro Utomo, Data mining adalah analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dari sebelumnya sehingga pemilik data dapat memahaminya dan menggunakannya [8]. Adapun tahapan yang ada dalam data mining, di antaranya: *data cleaning, data integration, data selection, and data* transformation [9]. Dapat disimpulkan bahwa *data mining* merupakan metode pengolahan data yang bertugas memilah dan mengevaluasi kumpulan data besar untuk menemukan pola, koneksi, dan pemahaman yang belum pernah diungkap

sebelumnya. Menggunakan algoritma dan metodologi statistik, inti dari data mining adalah membuka wawasan mengenai tren dan hubungan tersembunyi di dalam data, serta merubah informasi dasar menjadi intelijen bisnis yang siap diaplikasi. Lebih dari sekedar pencarian pola, proses ini diarahkan untuk menciptakan informasi yang bermanfaat dan mendukung perkembangan lebih jauh.

2.2.2. Klasifikasi

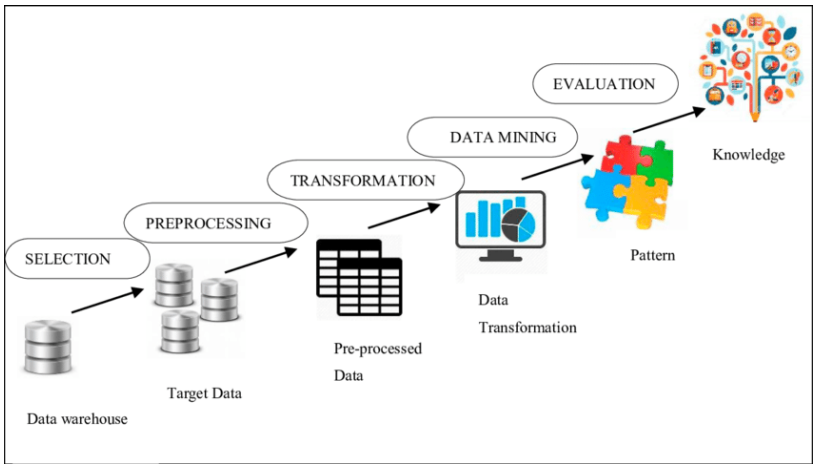
Dalam data mining, klasifikasi adalah metode pembelajaran dengan pengawasan di mana set data latih yang sudah dilabeli kelasnya digunakan sebagai referensi dalam memprediksi label kelas tepat untuk data baru yang belum terlabeli. Secara sederhana, model klasifikasi yang dikembangkan dari set data latih akan digunakan untuk menentukan kelas dari set data uji yang belum memiliki label [10].

2.2.3. Knowledge Discovery in Database

Dalam penelitian ini, metode yang diaplikasikan adalah KDD (Knowledge Discovery in Database). KDD merupakan metode yang diterapkan dalam riset ini dan berfungsi sebagai teknik untuk mengekstrak informasi berbentuk data dari database yang telah tersedia. Database ini terdiri dari tabel-tabel yang saling terkoneksi. Informasi yang diperoleh melalui proses KDD ini dapat dijadikan sebagai basis data yang mendukung proses pengambilan keputusan [11].

Knowledge Discovery in Database (KDD) dan *data mining* adalah kata-kata yang sering digunakan secara bergantian untuk menggambarkan proses pengambilan informasi dari basis data yang besar. Istilah KDD dan data mining memiliki keterkaitan satu sama lain, tetapi memiliki konsep yang berbeda. *Data*

mining adalah satu fase atau tahapan di dalam proses KDD [12]. Tahapan-tahapan yang ada pada KDD dapat divisualisasikan pada gambar berikut.



Gambar 2. 1 Tahapan Dalam KDD

(Sumber: resarchgate.net [13])

1. *Data Selection*

Langkah pertama sebelum penggalian informasi di Knowledge Discovery in Database (KDD) yaitu, data harus diseleksi atau dipilih dari data operasional. Data yang digunakan hanya data yang relevan untuk dianalisis [14].

2. *Pre-processing/Cleaning*

Sebelum melaksanakan fase data mining, langkah krusial yang harus dilakukan adalah pembersihan data, yang berperan sebagai fondasi dalam Knowledge Discovery in Database (KDD). Langkah pembersihan ini mencakup eliminasi data yang duplikat, verifikasi kekonsistenan data, dan koreksi masalah terkait data, termasuk kesalahan pengetikan. Proses ini juga melibatkan penyempurnaan data yang ada dengan menambahkan data atau informasi tambahan yang relevan dan diperlukan untuk proses Knowledge Discovery in Database (KDD), termasuk juga sumber data atau informasi eksternal [14].

3. *Transformation*

Transformasi adalah langkah mengkonversi data terpilih agar menjadi kompatibel untuk keperluan data mining. Proses ini sangat ditentukan oleh jenis database, model informasi, atau spesifikasi pencarian yang diinginkan. Dengan kata lain, transformasi melibatkan penentuan fitur yang efektif untuk merepresentasikan data, sesuai dengan objektif yang hendak dicapai [14].

4. *Data Mining*

Data mining adalah proses identifikasi pola atau informasi penting dalam data terpilih melalui penerapan alat atau metodologi khusus. Variasi teknik, metode, dan algoritma pada data mining sangatlah luas. Pemilihan pendekatan atau algoritma yang sesuai bergantung pada tujuan akhir dan konteks proses Knowledge Discovery in Database (KDD) secara keseluruhan [14].

5. *Interpretation/Evaluation*

Tahap ini termasuk dalam bagian interpretasi proses Knowledge Discovery in Database (KDD), di mana pola atau informasi yang dihasilkan melalui penambangan data perlu diinterpretasikan agar mudah dipahami oleh stakeholder. Tahap ini penting untuk menilai kesesuaian pola atau informasi yang diungkap dengan fakta atau teori yang telah ada sebelumnya. Selain itu, pada tahap ini juga dihasilkan pengetahuan yang akan sangat bermanfaat dalam proses pengambilan keputusan nantinya [14].

2.2.4. *Random Forest*

Random Forest adalah algoritme pembelajaran mesin yang fleksibel dan mudah digunakan. Algoritma Random Forest merupakan salah satu algoritme yang

paling banyak digunakan, karena kesederhanaan dan keragamannya (dapat digunakan untuk tugas klasifikasi dan regresi) [15]. Metode Random Forest berisi gabungan dari Decision Tree untuk melakukan klasifikasi, Random Forest merupakan algoritma ensemble dimana untuk mendapatkan keputusan akhir dilakukan voting mayority dari semua model Decision Tree [16].

Berikut adalah rumus yang digunakan dalam perhitungan manual untuk model *Random Forest*:

Rumus Entropy:

$$Entropy\ (S) = \sum_{i=1}^{n} - p_i\ x\ log_2 p_i$$ ..... persamaan (2.1)

- *S*: Himpunan Kasus
- *n*: Jumlah Partisi S
- *pi:* Proporsi ari *Si* terhadap *S*

Rumus Gain:

$$Gain\ (S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|}\ Entropy\ (S_1)$$ ..... persamaan (2.2)

- *S*: Himpunan Kasus
- *A*: Atribut
- *n*: Jumlah Partisi Atribut *A*
- |*Si*|: Jumlah kasus pada Partisi ke -*i*
- |S|: Jumlah Kasus dalam *S*

2.2.4.1. Penggunaan *Random Forest*

Hasil prediksi model *random forest* akan dipilih berdasarkan suara terbanyak, yaitu kategori atau kelas yang paling sering muncul sebagai hasil prediksi dari

pohon klasifikasi. Pembuatan pohon (*tree*) pada *random forest* sampai dengan mencapai ukuran maksimum dari pohon data atau dengan kata lain pembuatan pohon *random forest* tidak melakukan pemangkasan, hal ini disebabkan karena pembangunan dilakukan dengan cara menggunakan *random feature selection* untuk meminimalisir kesalahan.

2.2.5. Python

Gambar 2. 2 Logo Python
(Sumber : python.org [17])

```
age = 18
if age >= 18:
    print("You are an adult.")
else:
    print("You are a minor.")
```

Gambar 2. 3 Contoh Sintaks Python

Python merupakan bahasa pemrograman interpretatif serbaguna yang dirancang dengan penekanan pada keterbacaan kode. *Python* dianggap mampu menggabungkan kapabilitas kuat dengan sintaksis yang sangat jelas serta pustaka standar yang besar dan komprehensif. *Python* dikenal sebagai bahasa pemrograman tujuan umum yang dikembangkan untuk memastikan *source code* mudah dibaca. Selain itu, *Python* memiliki koleksi *library* yang lengkap, memungkinkan

programmer untuk membuat aplikasi canggih dengan *source code* yang terlihat sederhana [18].

Pada penelitian ini, ada beberapa *library* yang digunakan, yaitu:

1. *Pandas*

Pandas merupakan sebuah *library* yang sangat berguna untuk manipulasi dan analisis data. *Library* ini menyediakan struktur data canggih seperti *dataframe*, yang mempermudah proses pengolahan, penyaringan, dan transformasi data dalam bentuk tabel.

2. *Numpy*

NumPy adalah *library* penting untuk komputasi numerik dalam *Python*. *NumPy* menyediakan fungsi-fungsi dan struktur data untuk mengelola array multidimensi, yang sering kali digunakan dalam berbagai proses transformasi data.

3. *Seaborn*

Seaborn adalah *library* visualisasi data berbasis *Python* yang menawarkan antarmuka tingkat tinggi untuk membuat grafik statistik yang informatif. *Seaborn* dikembangkan di atas *Matplotlib*, salah satu pustaka visualisasi data paling mendasar dan populer di *Python*. Ini berarti *Seaborn* menggunakan fungsionalitas inti *Matplotlib*, sambil memberikan lapisan abstraksi tambahan untuk mempermudah analis dalam membuat visualisasi yang lebih kompleks.

4. *Matplotlib*

Matplotlib adalah *library* populer di lingkungan *Python* yang digunakan untuk visualisasi data. Meskipun *Matplotlib* tidak dirancang khusus untuk

transformasi data dalam *data mining*, pustaka ini dapat digunakan untuk memvisualisasikan data setelah transformasi dilakukan. Visualisasi data yang baik dapat membantu dalam memahami data dan mengidentifikasi pola.

5. *Scikit-Learn*

Scikit-learn adalah *library* populer untuk *machine learning* dalam *Python*. Selain menyediakan metode *machine learning*, *Scikit-learn* juga menawarkan fungsi untuk *pre-processing* data, termasuk pemisahan data, normalisasi, encoding kategori, dan penanganan *missing value*.

6. *Pickle*

Pickle adalah *library* dalam *Python* yang digunakan untuk serialisasi dan deserialisasi objek *Python*. Meskipun tidak dirancang khusus untuk transformasi data dalam data mining, *Pickle* dapat digunakan untuk menyimpan dan memuat data yang telah ditransformasi dalam bentuk objek *Python*.

2.2.6. Jupyter

Gambar 2. 4 Logo Jupyter Notebook
(Sumber: jupyter.org [19])

Menurut S. Junaidi, I. A. Ashari dan rekan-rekannya pada jurnal yang ditulis oleh Arfian Haris P dan Apriade Voutama, Jupyter merupakan aplikasi web gratis

yang memungkinkan Anda membuat dan membagikan dokumen yang berisi teks, kode, hasil perhitungan, dan visualisasi. Nama Jupyter adalah singkatan dari tiga bahasa pemrograman penting bagi ilmuwan data, yaitu Julia (Ju), Python (Py), dan R. Jupyter memfasilitasi pembuatan narasi komputasi yang menjelaskan arti data dan memberikan insight [20].

2.2.7. HTML (Hyper Text Markup Language)

Gambar 2. 5 Logo HTML
(Sumber: wikipedia.org [21])

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Hello, World!</title>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

Gambar 2. 6 Contoh Sintaks HTML

Menurut Winarno dan Utomo, yang dikutip oleh Agus Prayitno dan Yulia Safitri, "HTML adalah singkatan dari Hypertext Markup Language dan digunakan untuk menampilkan halaman web." Kode HTML berfungsi sebagai bahan untuk merender halaman web. Karena berbasis teks murni (plain text), HTML memiliki

ukuran yang kecil dan tidak menghabiskan banyak bandwidth saat ditransfer melalui jaringan internet. Dalam pemrograman HTML, terdapat istilah yang dikenal sebagai Tag. Tag adalah sintaks HTML yang ditulis di antara dua tanda lebih kecil dan lebih besar "< >" [22].

2.2.8. CSS (Cascading Style Sheet)

Gambar 2. 7 Logo CSS
(Sumber: wikipedia.org [23])

```
body {
    font-family: Arial, sans-serif;
}

h1 {
    color: blue;
    text-align: center;
}

p {
    font-size: 20px;
    color: green;
}
```

Gambar 2. 8 Contoh Sintaks CSS

Menurut Adhi Prasetio yang disitasi oleh Raden Shafira Annisa Ridmadhani, dkk menjelaskan bahwa CSS *(Cascading Style Sheet)* adalah salah satu bahasa desain *web* yang digunakan untuk mengatur tampilan elemen yang tertulis dalam

bahasa *markup*, seperti HTML. CSS berfungsi untuk memisahkan konten utama dengan tampilan visualnya. CSS menciptakan fleksibilitas dalam mengontrol spesifikasi tampilan suatu halaman *web*. CSS dibuat dan dikembangkan oleh W3C *(World Wide Web Consortium)* pada tahun 1996 [24].

2.2.9. PHP

Gambar 2. 9 Logo PHP
(Sumber: wikipedia.org [25])

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Hello, World!</title>
</head>
<body>
    <?php
        echo "Hello, World!";
    ?>
</body>
</html>
```

Gambar 2. 10 Contoh Sintaks PHP

Menurut Alexander F. K. Sibero, seperti yang dikutip oleh Harri Hidayat dan lainnya, PHP (Hypertext Preprocessor) adalah bahasa pemrograman interpreter di mana proses penerjemahan kode sumber menjadi kode mesin yang dimengerti komputer dilakukan secara langsung saat kode tersebut dijalankan. PHP dikenal sebagai Server Side Programming karena seluruh prosesnya dijalankan di server,

bukan di client. PHP merupakan bahasa pemrograman dengan lisensi terbuka atau Open Source, memungkinkan pengguna untuk mengembangkan kode PHP sesuai kebutuhan mereka [26].