

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Nyamuk *Aedes Aegypti* dan Penyakit Demam Berdarah (DBD)

Nyamuk *Aedes Aegypti* merupakan jenis nyamuk yang membawa virus dengue penyebab penyakit demam berdarah. Nyamuk DBD membawa virus di dalam tubuhnya. Penyakit DBD ditularkan ke manusia melalui gigitan nyamuk *Aedes Aegypti*. Gejala penyakit muncul 4-7 hari setelah digigit nyamuk. Demam berdarah sendiri menjadi penyakit yang umum di Indonesia terutama di musim penghujan karena Hujan yang menggenang menciptakan habitat ideal bagi nyamuk *Aedes aegypti*, pembawa virus Dengue penyebab demam berdarah. Telur nyamuk ini dapat bertahan lama di tempat yang kering dan akan menetas ketika terendam air hujan. Berdasarkan penelitian Nurul Inayah, Irwan, dan Adnan Sauddin (2024), "Penyakit ini telah menjadi masalah kesehatan masyarakat di Indonesia"

Kementerian Kesehatan mencatat hampir 100.000 orang terinfeksi setiap tahunnya, dimana dalam jangka waktu perbulan, hampir setiap daerah Kabupaten/Kota terdapat kasus penularan penyakit DBD". Maka dari itu ini menjadi penyakit yang menyerang setiap tahun di setiap daerah di indonesia.

2.1.1.1 Karakteristik Nyamuk *Aedes*

Menurut Soegijanto (2006) Nyamuk ini memiliki ciri khusus ditandai dengan pita atau garis-garis putih keperakan di atas dasar hitam, ukuran nyamuk

A.aegypti berkisar sekitar 3-4 mm dengan ring putih pada bagian kakinya. Nyamuk ini memiliki ukuran yang kecil dan memiliki tubuh berwarna hitam dengan belang putih di sekujur tubuhnya. Nyamuk *Ae. aegypti* lebih menyukai darah manusia daripada binatang (antropofilik) dan bersifat menggigit pada beberapa orang sebelum merasa kenyang. Nyamuk *Ae.aegypti* L. ini hidup dan berkembang dengan baik di daerah tropis yaitu pada garis isotherm 20⁰ yang terletak diantara 45⁰ LU dan 35⁰LS dengan ketinggian kurang dari 1000 meter di atas permukaan laut (Sugito, 1989).

2.1.1.2 Penyebaran dan Pencegahan Penyakit DBD

Penyakit DBD disebabkan oleh virus dengue yang dibawa oleh nyamuk aedes aegypti melalui gigitan nyamuk, selain itu Penyebaran nyamuk *Ae. aegypti* L. sangat dipengaruhi oleh lingkungan fisik, biologik dan kebiasaan dari masyarakat yang mendukung untuk perkembangannya. Perilaku masyarakat juga berpengaruh besar karena perilaku masyarakat dapat memberikan daya dukung lingkungan bagi perkembangan nyamuk (Sukesi, 2012). Lingkungan dengan genangan air, seperti di tempat penyimpanan air, wadah terbuka, dan sampah, dapat menjadi lokasi ideal bagi nyamuk *Aedes* untuk berkembang biak. Kelembapan udara yang tinggi, curah hujan yang meningkat, dan suhu udara yang panas juga mendukung perkembangbiakan nyamuk ini. Selain perubahan iklim, musim hujan juga menciptakan lebih banyak genangan air, yang meningkatkan potensi perkembangbiakan nyamuk pembawa virus dengue, sehingga kasus DBD cenderung meningkat selama musim hujan. Pencegahan yang efektif yang dilakukan Masyarakat ialah 3M plus yaitu menutup, menguras dan menimbun, Dimana 3M yakni

1. Menguras tempat penampungan air
2. Menutup tempat-tempat penampungan air
3. Mendaur ulang berbagai barang yang memiliki potensi untuk dijadikan tempat berkembang biak nyamuk *Aedes aegypti* yang membawa virus DBD pada manusia.

Selain 3 M diatas yang dimaksud pada poin Plus antara lain

1. Menanam tanaman yang dapat menangkal nyamuk
2. Memeriksa tempat-tempat yang digunakan untuk penampungan air
3. Memelihara ikan pemakan jentik nyamuk
4. Menggunakan obat anti nyamuk, dll.

2.1.2 Metode Prediksi Kasus DBD

2.1.2.1 Pendekatan Tradisional dalam Prediksi DBD

Dalam pendekatan tradisional memprediksi penyakit demam berdarah di suatu daerah biasanya didasarkan pada statistik deskriptif, pemantauan lapangan, pemantauan dan analisis manual terhadap faktor lingkungan seperti suhu, kelembaban, dan curah hujan. Metode ini tidak memerlukan pemodelan yang rumit dan didasarkan pada hubungan langsung yang mudah diamati, seperti peningkatan kasus DBD pada musim hujan. Hal ini tentu saja masih banyak kekurangannya untuk beradaptasi dengan data baru serta keakuratan lebih rendah dibandingkan

dengan menggunakan model seperti algoritma yang mampu menangani data yang besar dan dengan hasil prediksi yang akurat.

2.1.2.2 Metode Modern dalam Prediksi DBD (ARIMA dan Transformer)

Penggunaan algoritma dalam memprediksi penyakit seperti dbd yang dapat mengenali pola kompleks dan data yang lebih besar dan bervariasi juga dapat menganalisis hubungan jangka panjang dalam data dengan lebih akurat. Selain itu dengan melibatkan data cuaca seperti kelembapan dan curah hujan menunjukkan faktor cuaca mempunyai hubungan linear dengan kasus DBD yang bisa meningkatkan akurasi berdasarkan variable cuaca yang digunakan. Metode berbasis algoritma modern seperti ARIMA dan Transformer memberikan prediksi yang jauh lebih kuat dan adaptif untuk data deret waktu, terutama dalam menghadapi perubahan tren dan hubungan data yang kompleks, seperti halnya DBD.

2.1.3 Time Series dalam Prediksi DBD

2.1.3.1 Pengertian dan Cakupan Time Series

Analisis deret berkala atau *time series* dilakukan untuk menyusun pola hubungan antara variabel yang dicari dengan variabel waktu yang mempengaruhinya (Muawanah, 2018). data deret waktu sering digunakan dalam analisis tren atau pola yang terjadi dari waktu ke waktu dan digunakan di berbagai bidang untuk memprediksi kejadian masa depan dengan menggunakan data

historis. *Time series* pada umumnya digunakan dalam berbagai bidang yakni keuangan, ekonomi, cuaca atau meteorologi dan lain banyak hal yang menggunakan waktu sebagai variabel utamanya. cakupan time series memiliki cakupan yang luas dan biasanya mencakup aspek penting seperti dekripsi pola waktu yang menjelaskan fenomena yang terjadi pada interval waktu tertentu. adapun peramalan seperti Memprediksi nilai di masa depan berdasarkan data historis, seperti perkiraan cuaca, prediksi ekonomi, atau estimasi permintaan produk, tentu ini menjadi alat yang sangat berguna di masa mendatang karena dapat mengatasi ketidakpastian dan membantu pengambilan keputusan yang lebih akurat dan efektif.

2.1.3.2 *Machine Learning* dan *Deep Learning*

Machine Learning adalah cabang ilmu komputer yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk belajar dan membuat prediksi berdasarkan data secara eksplisit untuk setiap tugas spesifik, yang membantu tugas secara mandiri untuk mengenali pola atau identifikasi hubungan. Adapun proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui dua tahap antara lain latihan (*training*) dan pengujian (*testing*) (Huang, Zhu, & Siew, 2006). sedangkan *Deep learning* adalah salah satu bidang machine learning yang memanfaatkan banyak layer pengolahan informasi nonlinier untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi (Deng dan Yu, 2014). Dimana *Deep learning* adalah cabang dari *Neural Network* yang berfokus pada pembelajaran mendalam untuk memungkinkan

komputer menyelesaikan berbagai tugas secara lebih efektif. Keduanya merupakan cabang ilmu *Artificial intelligence* (AI) yang menjadi dasar bagi inovasi modern, digunakan secara luas ini berbagai bidang, yang digunakan *machine learning* dan *deep learning*, pemrosesan bahasa alami, sistem rekomendasi, kendaraan otonom, dan analisis prediktif.

2.1.3.3 Jenis-jenis Deep Learning untuk Time Series

1. *Recurrent Neural Networks* (RNN)

Jaringan saraf berulang atau *Recurrent Neural Network* (RNN) adalah jenis arsitektur jaringan syaraf tiruan dengan masukan yang diproses secara berulang-ulang (Zhang et al., 2018). Jaringan saraf berulang atau *Recurrent Neural Network* (RNN) adalah jenis artificial intelligence yang membutuhkan banyak lapisan (*layer*) untuk memproses data masukan, sehingga RNN dimasukkan dalam kategori *deep learning*. dalam konteks *time series*, RNN sangat berguna karena arsitekturnya memungkinkan pemodelan data sekuensial atau deret waktu dengan mempertimbangkan ketergantungan antar data pada titik waktu sebelumnya. RNN banyak digunakan dalam berbagai aplikasi deret waktu seperti perkiraan cuaca, perkiraan penjualan, dan analisis keuangan karena dapat menangani data yang bersifat temporal atau kontinu.

2. *Long Short-Term Memory* (LSTM)

Salah satu algoritma *deep learning* yang terbukti berhasil digunakan untuk memprediksi data time series adalah *Long Short-Term Memory* (LSTM), yang merupakan turunan dari *Recurrent Neural Network* (RNN) (Hsu, 2018). *Long Short-Term Memory* (LSTM) adalah jenis jaringan saraf tiruan yang dikembangkan khusus untuk menangani data urutan atau *sequential* data, seperti teks, video, atau data deret waktu (*time series*). LSTM memiliki struktur memori yang memungkinkan jaringan untuk “mengingat” informasi penting dan “melupakan” informasi yang tidak relevan dalam jangka waktu yang lama. Karena sifat ini, LSTM sangat efektif untuk tugas-tugas yang membutuhkan pemahaman konteks waktu atau urutan, seperti prediksi cuaca, dan yang lainnya.

3. Transformer

Transformer adalah model arsitektur *deep learning* yang dirancang untuk menangani data sekuensial dengan memanfaatkan mekanisme *self-attention*. *Transformer* merupakan Arsitektur model yang menggabungkan *self-attention* dan *layer-layer neural network* untuk pemrosesan sekuensial telah dikembangkan untuk meningkatkan efisiensi dalam pemrosesan data yang berbasis urutan (Vaswani et al., 2017). Dalam kasus *time series* adapun *informer*, *Informer Transformer* adalah model pembelajaran mendalam yang dirancang untuk memproses rangkaian data panjang dengan lebih efisien, terutama untuk tugas prediksi deret waktu.

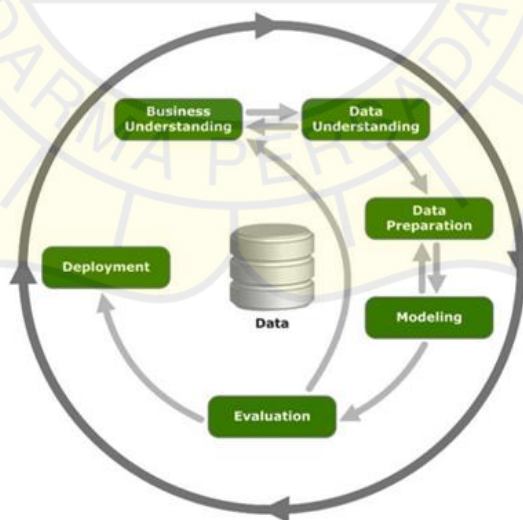
Model ini memperkenalkan mekanisme perhatian mandiri *Prob Sparse*, yang mengurangi biaya komputasi lapisan perhatian mandiri dengan hanya memilih

kueri utama, dan mengatasi tantangan kompleksitas komputasi yang biasanya berbentuk kuadrat pada *transformer* tradisional.

2.1.4 CRISP-DM sebagai Tahap Merancang Sistem Berbasis Machine Learning

2.1.4.1 Proses dan Tahapan CRISP-DM

CRISP-DM adalah kerangka kerja data mining yang menerapkan enam fase dalam metodenya untuk mengidentifikasi input dan output dalam suatu proses. Keenam fase tersebut adalah *business understanding*, *data understanding*, *data preparation*, *modeling*, *deployment*, dan evaluasi. Dalam 6 tahap CRISP-DM digambarkan pada gambar 2.1 sebagai berikut ini:



Gambar 2. 1 Tahapan CRISP-DM

tahapan dalam CRISP-DM yakni sebagai berikut:

- a. *Business Understanding* (Pemahaman Bisnis) Beberapa hal yang dilakukan pada tahap ini seperti memahami kebutuhan serta tujuan dari sudut pandang bisnis selanjutnya mengartikan pengetahuan ke dalam bentuk pendefinisian masalah pada data mining dan kemudian menentukan rencana serta strategi untuk mencapai tujuan data mining.
- b. *Data Understanding* (Pemahaman Data) Tahapan ini diawali dengan mengumpulkan data, mendeskripsikan data, serta mengevaluasi kualitas data.
- c. *Data Preparation* (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.
- d. *Modelling* (Pemodelan) Pada tahapan ini secara langsung melibatkan Machine Learning untuk penentuan teknik *data mining*, alat bantu *data mining* beserta algoritma *data mining*.
- e. *Evaluation* (Pengujian) Tahap ini dilakukan dengan melihat tingkat performa dari pola yang dihasilkan oleh algoritma. Parameter yang digunakan untuk evaluasi komparasi algoritma adalah *Confusion Matrix* dengan aturan nilai akurasi, presisi dan *recall*.
- f. *Deployment* (Penyebaran) Tahapan ini dilakukan dengan pembuatan laporan dan artikel jurnal menggunakan model yang dihasilkan.

2.1.4.2 Implementasi CRISP-DM dalam Prediksi Kasus DBD

1. Business understanding

Fase ini berfokus pada pemahaman mendalam terhadap permasalahan bisnis, dalam hal ini peramalan kasus demam berdarah, serta dampak yang diharapkan dari hasil perkiraan outcome. Tujuannya adalah untuk mengurangi atau mengendalikan jumlah kasus demam berdarah dengan memperkirakan wilayah, periode dan kondisi lingkungan yang paling mungkin meningkatkan jumlah kasus. Memahami pola penyebaran demam berdarah dan faktor risiko terkait juga menjadi perhatian utama.

2. Pemahaman Data

Langkah ini melibatkan pengumpulan dan eksplorasi data terkait kasus demam berdarah. Data yang umum digunakan antara lain:

- a. Data epidemiologi: jumlah kasus demam berdarah berdasarkan lokasi dan waktu.
- b. Data cuaca dan iklim: suhu, curah hujan, kelembapan, dan indikator iklim lainnya dapat mempengaruhi penyebaran penyakit.
- c. Data lingkungan: data kepadatan penduduk, kebersihan lingkungan dan parameter lainnya.
- d. Data kesehatan masyarakat: program pencegahan dan intervensi dilaksanakan di daerah tertentu.

Dengan analisis data awal ini, kita dapat memahami pola awal dan korelasi antar variabel, mengidentifikasi outlier, dan memperoleh informasi yang relevan untuk langkah analisis selanjutnya.

3. Persiapan data

Pada tahap ini data yang diperoleh akan diolah agar sesuai untuk dianalisis lebih lanjut. Beberapa proses yang dilakukan antara lain:

- a. Pembersihan data: menangani nilai yang hilang, mengoreksi data yang tidak akurat atau tidak konsisten, dan menangani outlier yang dapat mempengaruhi hasil.
- b. Transformasi Data: Menormalkan, mengubah ukuran, atau mengubah data untuk memastikan konsistensi analisis.
- c. Rekayasa Fitur: Membuat variabel baru atau menggabungkan variabel yang ada berdasarkan pengetahuan domain. Misalnya menggabungkan data cuaca ke dalam indeks risiko demam berdarah.
- d. Pemisahan Data: Membagi kumpulan data menjadi kumpulan pelatihan dan pengujian untuk memastikan bahwa model yang dikembangkan dapat diuji keakuratan dan performa generalisasinya.

4. Pemodelan

Langkah ini meliputi pengembangan model prediksi kasus DBD dengan menggunakan algoritma data mining yang sesuai. Algoritma yang biasa digunakan untuk memprediksi kasus DBD antara lain:

- a. Regresi linier: untuk mengetahui korelasi kasus DBD dengan variabel cuaca atau lingkungan.

Decision Tree : untuk mengklasifikasikan area risiko berdasarkan atribut tertentu.

b. *Random forest* dan *gradient boosting* : Untuk meningkatkan akurasi prediksi dengan menggabungkan hasil dari beberapa pohon keputusan.

c. Algoritma pembelajaran mendalam: seperti jaringan saraf tiruan yang mampu menangkap pola kompleks antar variabel.

Model dipilih berdasarkan performa prediktifnya dan sesuai dengan data yang tersedia.

5. Evaluasi

Pada tahap evaluasi, model yang dikembangkan diuji dengan data eksperimen untuk mengukur keakuratan prediksinya. Beberapa metrik evaluasi yang dapat digunakan antara lain:

a. Akurasi: Persentase prediksi yang benar.

b. *Recall* dan presisi: untuk melihat kemampuan model dalam mendeteksi kasus positif (daerah dengan jumlah kasus DBD tinggi) dan keakuratan prediksi kasus positif.

c. ROC-AUC: Untuk mengukur kemampuan model dalam membedakan area berisiko tinggi dan rendah.

d. Evaluasi ini memungkinkan kita untuk menentukan apakah model tersebut cukup efektif untuk diterapkan atau perlu perbaikan lebih lanjut.

6. Deployment

Langkah terakhir adalah menerapkan model prediksi kasus demam berdarah ke dalam sistem yang digunakan oleh pemerintah atau dinas kesehatan setempat.





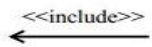
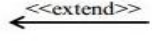
Hal ini dapat dicapai dengan membuat aplikasi *web* atau *dashboard* interaktif untuk memantau dan memprediksi risiko demam berdarah di berbagai wilayah.

2.1.5 Pemodelan Sistem UML

2.1.5.1 Diagram *Use Case*

Diagram *Use Case* adalah bagian dari pemodelan *Unified Modeling Language* (UML) yang menggambarkan interaksi antara pengguna (aktor) dengan sistem. *Use case* adalah seperangkat skenario yang diikat bersama oleh user untuk mencapai tujuan (Fowler & Scott, 1993). Setiap *use case* mewakili fungsi atau layanan yang disediakan oleh sistem, menggambarkan langkah-langkah atau interaksi yang diperlukan untuk mencapai hasil yang diinginkan. Untuk lambang dan pengertian setiap *case* dijelaskan di tabel 2.1 sebagai berikut:



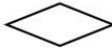



Tabel 2. 1 Penjelasan simbol Use case diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2.1.5.2 Diagram Aktivitas

Diagram *activity* menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity* diagram juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan. “Diagram *activity* adalah aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas” (Haviluddin, 2011). *Activity* diagram sendiri adalah cabang dari *Unified Modeling Language* (UML) yang menggambarkan perilaku dinamis sistem yang melibatkan interaksi antar objek, transisi antar status (*state*), dan kejadian yang mempengaruhi jalannya aktivitas. Dalam penggambaran ini, setiap aktivitas dapat memiliki kondisi awal dan akhir, serta keputusan yang harus diambil berdasarkan kondisi tertentu, yang memungkinkan adanya percabangan dan pengulangan proses dalam alur kerja tersebut. lambang dan pengertian dari *activity* diagram di jelaskan di tabel 2.2 sebagai berikut:

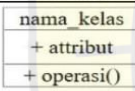



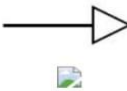


Tabel 2. 2 Penjelasan simbol diagram aktivitas

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.1.5.3 Diagram Kelas

Class diagram merupakan arsitektur dari sebuah sistem yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang terdapat dalam sistem. *Class* diagram memberikan gambaran mengenai relasi-relasi antar tabel dalam sistem yang dibangun (Hidayat, 2019). Diagram kelas merupakan komponen utama dari *Unified Modeling Language* (UML) yang digunakan untuk merepresentasikan struktur sistem berorientasi objek. Diagram ini menunjukkan kelas-kelas sistem, atribut-atributnya, metode-metodenya, dan hubungan-hubungan di antara kelas-kelas tersebut. berikut ini lambang dan pengertian dari masing-masing lambang di tabel 2.3 berikut ini :

Tabel 2. 3 penjelasan simbol *class* diagram

Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem.
	Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
	Association	Relasi antarclass dengan arti umum, asosiasi biasanya juga disertai dengan Multiplicity.
	Directed Association	Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
	Dependency	Relasi antarkelas dengan makna kebergantungan antarkelas'
	Aggregation	Relasi antarkelas dengan makna semua-bagian (whole-part)

2.1.6 Software dan Pemrograman Terkait

2.1.6.1 Editor Jupyter

Jupyter Editor adalah platform yang memungkinkan pengguna menulis dan menjalankan kode interaktif dalam format notebook. Platform ini sangat berguna untuk analisis data, pembelajaran mesin, dan pengembangan perangkat lunak karena memungkinkan eksekusi kode langkah demi langkah, visualisasi data, dan perekaman hasil secara langsung dalam satu tampilan. Menurut Zhang (2021), *Jupyter* menyediakan lingkungan yang ramah pengguna untuk mengintegrasikan kode *Python* dengan visualisasi, teks, dan gambar, memfasilitasi analisis data interaktif. Hal ini menjadikan *Jupyter* pilihan utama bagi banyak ilmuwan data dan pengembang di bidang pemrosesan data dan eksperimen komputasi.

2.1.6.2 Bahasa Pemrograman Python

Python adalah bahasa pemrograman yang sangat populer dalam analisis data dan pembelajaran mesin karena sintaksisnya yang sederhana dan ketersediaan perpustakaan yang kuat. Menurut Suryana (2020), *Python* memungkinkan pemrograman dengan cepat melakukan tugas analitis yang kompleks melalui berbagai perpustakaan seperti *Pandas*, *Numpy*, dan *Matplotlib*. *Python* mendukung paradigma pemrograman fungsional dan berorientasi objek serta memiliki komunitas yang besar, menjadikannya pilihan utama dalam berbagai aplikasi, mulai dari pemrosesan data hingga pengembangan kecerdasan buatan. Penggunaannya

dalam analisis deret waktu menjadi sangat relevan karena kemampuan pemrosesan data deret waktu yang mendalam dari perpustakaan.

2.1.6.3 Library terkait: scikit-learn, TensorFlow, PyTorch, OpenCV

Pustaka *time series* dalam *Python* digunakan untuk menganalisis, memodelkan, dan memprediksi data deret waktu, di mana data diurutkan berdasarkan waktu, seperti data cuaca, penjualan, atau fluktuasi harga saham. Pustaka utama dalam analisis *time series* antara lain:

1. Pandas

Pandas adalah pustaka yang sangat penting untuk manipulasi data deret waktu. *Pandas* memungkinkan pengguna untuk mengakses, memanipulasi, dan menganalisis data besar dalam bentuk *DataFrame* yang fleksibel, terutama yang berkaitan dengan komponen waktu. Pustaka ini dilengkapi dengan fungsi untuk menggabungkan data indeks waktu, *resampling*, dan mengisi nilai yang hilang (McKinney, 2018). *Pandas* menyediakan alat efisien untuk perhitungan statistik, agregasi, dan visualisasi data deret waktu.

2. Statsmodels

Statsmodels adalah pustaka *Python* untuk analisis *time series* yang mencakup model ARIMA, SARIMA, dan regresi lainnya. Menurut Seabold dan Perktold (2010), pustaka ini mendukung analisis prediktif, peramalan, identifikasi

pola musiman, dan uji hipotesis pada data deret waktu. Pustaka ini menawarkan fungsi untuk analisis regresi linear, analisis varians, dan tes statistik.

3. Prophet

Prophet adalah pustaka yang dikembangkan oleh *Facebook* untuk menganalisis data *time series* dengan komponen musiman kuat dan tren non-linier. Keunggulannya terletak pada kemampuannya menangani banyak komponen musiman serta hari libur yang mempengaruhi peramalan. *Prophet*, yang dikenal mudah digunakan, memungkinkan pengguna membuat model peramalan akurat tanpa pengalaman mendalam dalam statistik atau analisis *time series* (Taylor & Letham, 2018). Pustaka ini ideal untuk aplikasi bisnis dan ekonomi, di mana data historis dipengaruhi faktor eksternal.

4. TensorFlow dan Keras (*Deep Learning* untuk *Time Series*)

Penggunaan *deep learning* untuk analisis *time series* telah meningkat pesat dalam beberapa tahun terakhir. *TensorFlow* dan *Keras* menawarkan alat untuk membangun dan melatih model jaringan syaraf tiruan yang efektif untuk data *time series*. Model seperti *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Units* (GRU) di *Keras* cocok untuk memodelkan data deret waktu dengan ketergantungan jangka panjang. Penggunaan teknik *deep learning* memungkinkan model menangkap pola kompleks dalam data deret waktu yang sulit diidentifikasi oleh model tradisional (2014).

5. Scikit-learn

Scikit-learn adalah pustaka *Python* untuk algoritma pembelajaran mesin yang dapat digunakan pada data time series, termasuk regresi, klasifikasi, dan *clustering*. *Scikit-learn*, meski lebih umum digunakan untuk data numerik, juga menawarkan alat untuk mempersiapkan data deret waktu, termasuk *train-test split* berdasarkan waktu dan ekstraksi fitur untuk analisisnya. Algoritma regresi seperti Ridge dan Lasso dapat digunakan untuk memprediksi nilai masa depan berdasarkan data historis. Teks tersebut dapat dipersingkat menjadi: "Histori (Pedregosa et al., 2011)

2.2 Kajian Penelitian Terdahulu

2.2.1 Paper 1: (judul, author, publikasi di, tahun, klasifikasi journal: akreditasi atau sinta/Q berapa)

Judul: Prediksi Penyebaran Demam Berdarah Dengue dengan Algoritma *Hybrid Autoregressive Integrated Moving Average* dan *Artificial Neural Network*: Studi Kasus di Kabupaten Bandung.

Author: Ichwanul Muslim Karo

Publikasi di: *Journal of Software Engineering, Information and Communication Technology* (SEICT)

Tahun: 2021

Klasifikasi Jurnal: Sinta 2 (Akreditasi)

2.2.1.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengembangkan model prediksi penyebaran Demam Berdarah Dengue (DBD) menggunakan pendekatan *Hybrid* ARIMA dan *Artificial Neural Network* (ANN), khususnya di Kabupaten Bandung, dengan menganalisis variabel cuaca yang mempengaruhi penyebaran DBD.

2.2.1.2 Metodologi Yang Digunakan

Metodologi yang digunakan adalah penggabungan model ARIMA dengan ANN (*Hybrid ARIMA-ANN*). Data yang digunakan berasal dari periode Januari 2009 hingga November 2016, yang mencakup variabel cuaca seperti suhu udara, kelembapan, curah hujan, dan penyinaran matahari. Model ini dibangun untuk memprediksi kejadian DBD berdasarkan data historis.

2.2.1.3 Temuan Utama

Model hybrid ARIMA-ANN memberikan hasil prediksi yang lebih akurat dibandingkan model ARIMA tunggal, dengan nilai RMSE sebesar 0.0087. Variabel cuaca, terutama suhu, kelembapan, dan curah hujan, memiliki dampak signifikan terhadap tingkat kejadian DBD, sedangkan kecepatan angin tidak berpengaruh signifikan.

2.2.1.4 Kesimpulan Penelitian

Penelitian ini menyimpulkan bahwa model Hybrid ARIMA-ANN lebih efektif untuk memprediksi penyebaran DBD dibandingkan dengan model ARIMA

tunggal. Cuaca memainkan peran penting dalam menentukan tingkat kejadian DBD, yang dapat dimanfaatkan dalam perencanaan dan pencegahan penyakit.

2.2.2 Paper 2: (judul, author, publikasi di, tahun, klasifikasi journal: akreditasi atau sinta/Q berapa)

Judul: *Time Series Forecasting with Transformer Models and Application to Asset Management*

Author: Jiali XU, Edmond LEZMI

Publikasi di: Tidak disebutkan secara spesifik

Tahun: 2023

Klasifikasi Jurnal: Tidak disebutkan

2.2.2.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengeksplorasi penggunaan model Transformer dalam peramalan deret waktu di bidang keuangan, dengan fokus pada prediksi pergerakan harga aset dan peramalan volatilitas untuk manajemen aset.

2.2.2.2 Metodologi yang Digunakan

Penelitian ini menggunakan model Transformer yang memanfaatkan mekanisme perhatian (*attention mechanism*) dan arsitektur *seq2seq* untuk memodelkan ketergantungan jangka panjang antar aset. Aplikasi keuangan mencakup strategi *trend-following* dan peramalan volatilitas.

2.2.2.3 Temuan Utama

Model *Transformer* berhasil diterapkan dalam peramalan deret waktu untuk prediksi harga aset dan volatilitas. Penelitian ini juga membahas tantangan penggunaan machine learning di bidang keuangan, terutama terkait rasio sinyal-ke-noise yang rendah dalam data keuangan.

2.2.2.4 Kesimpulan Penelitian

Penelitian ini menyimpulkan bahwa model *Transformer* dapat digunakan secara efektif dalam peramalan deret waktu di bidang keuangan. Penelitian masa depan harus fokus pada penanganan data keuangan yang bising dan eksplorasi model gabungan seperti *Transformer* dan *Graph Neural Networks* (GNN).

2.2.3 Paper 3: (judul, author, publikasi di, tahun, klasifikasi journal: akreditasi atau sinta/Q berapa)

Judul: *Informer Model with Season-Aware Block for Efficient Power Series Forecasting*

Author: Yunlong Cui, Zhao Li, Peng Zhang

Publikasi di: Tidak disebutkan secara spesifik

Tahun: 2023

Klasifikasi Jurnal: Tidak disebutkan

2.2.3.1 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan model Informer yang lebih efisien dalam peramalan deret waktu konsumsi daya, dengan penekanan pada pengenalan Blok yang Sadar Musim untuk menangani pola musiman dalam data daya.

2.2.3.2 Metodologi yang Digunakan

Model *Informer* dengan Blok yang Sadar Musim digunakan untuk menganalisis data daya. Model ini memanfaatkan mekanisme perhatian mandiri *Prob Sparse* untuk mengurangi kompleksitas memori dan meningkatkan ketergantungan jangka panjang dalam peramalan deret waktu.

2.2.3.3 Temuan Utama

Eksperimen dengan data daya di Provinsi Zhejiang menunjukkan peningkatan 19% dalam MSE. Uji coba ablasi membuktikan bahwa Blok yang Sadar Musim berhasil mengelola faktor buatan seperti acara dan hari libur, yang berpengaruh pada pola konsumsi daya.

2.2.2.4 Kesimpulan Penelitian

Penelitian ini menyimpulkan bahwa penggunaan Blok yang Sadar Musim dalam model Informer meningkatkan akurasi peramalan konsumsi daya dengan mengidentifikasi karakteristik musiman. Model ini juga menunjukkan efisiensi dalam mengelola faktor buatan dalam data deret waktu.