BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian ini dilakukan dengan mengambil beberapa referensi dan informasi dari penelitian-penelitian yang sudah dilakukan sebelumnya terkait metode *hybrid clustering*, prediksi menggunakan *LSTM*, topik pencemaran udara, dan lainnya yang berkaitan dengan penelitian ini. Berikut merupakan tabel penelitian yang sudah pernah dilakukan sebelumnya:

Tabel 2. 1 Penelitian Terdahulu

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
1	Data Mining Implementasi Algoritma K-Means Menggunakan Aplikasi Orange dalam Clustering Pencemaran Udara di DKI Jakarta Tahun 2021. [17]	*Journal of Informatics and Advanced Computing (JIAC)*, *3*(2), 161-164.	Michael Sitorus, Depriansa Fitron, Carolus Agung Segara Wisesa / (2022)	*K-Means*, *Logistic Regression*	Tingkat akurasi dari algoritma *K-Means* berdasarkan pengujian *Logistic Regression* dengan *cluster* kualitas udara sedang dan *cluster* kualitas udara tidak sehat mencapai 0,9622 atau sebesar 96,22%.

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
2	*Prediction of PM10 Concentration in Malaysia Using K-Means Clustering and LSTM Hybrid Model*. [16]	*Atmosphere*, *14*(5), 853.	Noratiqah Mohd Ariff, Mohd Aftar Abu Bakar, Han Ying Lim / (2023)	*Hybrid K-Means* & *Long Short-Term Memory* (LSTM)	Dari 60 Stasiun Pemantau Kualitas Udara dijadikan 2 *cluster* dengan *cluster* pertama terdiri dari 19 stasiun yang ada di daerah berkembang, serta *cluster* kedua terdiri dari 41 stasiun yang berada di daerah pinggiran kota. Untuk hasil prediksi menggunakan *hybrid model* dapat mengetahui pola konsentrasi PM10 rata-rata harian, walaupun memberikan hasil yang lebih buruk daripada model LSTM univariat karena beberapa faktor seperti kondisi meteorologi yang berbeda-beda di setiap lokasi stasiun pemantau. Namun, model *hybrid* tersebut memiliki waktu pelatihan yang lebih singkat. Oleh karena itu, model *hybrid* dapat lebih kompetitif dan cocok untuk diaplikasikan secara langsung dalam meramalkan kualitas udara.

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
3	Pengujian Algoritma *Long Short Term Memory* untuk Prediksi Kualitas Udara dan Suhu Kota Bandung. [18]	*Jurnal Telematika*, *15*(1), 13-18.	Ali Khumaidi, Ridwan Raafi'udin, Indra Permana Solihin / (2020)	*Long Short-Term Memory* (LSTM)	Model LSTM menghasilkan performa yang baik dalam memprediksi 3 parameter, yaitu suhu, kelembapan, dan ISPU. Nilai RMSE prediksi lebih kecil dari nilai standar deviasi uji *dataset*. Untuk hasil prediksi menggunakan 4 parameter, parameter yang paling baik diuji adalah kelembapan, suhu, ISPU, dan PM10.
4	*Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering*. [19]	*Expert Systems with Applications*, *169*, 114513.	Rui Yan, Jiaqiang Liao, Jie Yang, Wei Sun, Mingyue Nong, Feipeng Li / (2021)	CNN, *Long Short-Term Memory*, *Back-Propagation Neural Network* (BPNN), *Hybrid CNN - LSTM*	Peramalan yang dilakukan secara satu jam ke depan menggunakan BPNN, CNN, LSTM, dan model *hybrid* CNN – LSTM menunjukkan bahwa penggunaan LSTM sebagai model dikatakan optimal untuk melakukan *forecasting* beberapa jam ke depan. Secara garis besar, model LSTM dan *hybrid* CNN – LSTM memiliki performa yang lebih baik dibandingkan CNN ataupun BPNN.

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
5	*Hybrid of K-Means and partitioning around medoids for predicting COVID-19 cases: Iraq case study*. [15]	*Periodicals of Engineering and Natural Sciences*, *9*(4), 569-579.	Nidaa Ghalib Ali, Saba Dhey Abed, Faris Ali Jasim Shaban, Korakod Tongkachok, Samrat Ray, Refed Adnan Jaleel / (2021)	*Hybrid K-Means* & *Partitioning Around Medoids* / *K-Medoids*	Menggunakan model *hybrid* dari *K-Means* dan juga PAM atau disebut sebagai *K-Medoids* untuk memprediksi status pasien COVID-19 berdasarkan data 400 pasien klinik di Iraq menggunakan kuisioner. Hasil akhir adalah model *hybrid K-Means* dan PAM atau K-MP lebih efisien dan efektif dalam menemukan status pasien dengan *K-Means* ataupun PAM.
6	Aplikasi Pengelompokan Data Runtun Waktu dengan Algoritma K-Medoids. [20]	*Inferensi*, *6*(2), 117-123.	Muhammad Aldani Zen, Sri Wahyuningsih, Andrea Tri Rian Dani / (2023)	*Time Series K-Medoids*	Menggunakan algoritma *K-Medoids* untuk menerapkan *time series clustering* pada data harga minyak goreng pada 34 provinsi di Indonesia dari bulan Oktober 2017 – Oktober 2022. Dihasilkan nilai *K* optimal sebanyak 2 klaster berdasarkan koefisien *silhouette* sebesar 0,19 dengan jarak DTW terdapat 19 provinsi pada klaster 1 yang merupakan harga minyak goreng dibawah klaster 2, serta terdapat 15 provinsi pada klaster 2 yang merupakan harga minyak goreng tertinggi.

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
7	*Long Short-Term Memory Approach for Predicting Air Temperature In Indonesia.* [21]	Jurnal Online Informatika, 161-168.	Putu Harry Gunawan, Devi Munandar, Anis Zainia Farabiba / (2020)	*Long Short-Term Memory*	Menggunakan algoritma *Long Short-Term Memory* untuk memprediksi suhu udara di Indonesia. Terdapat 2 *optimizer* yang digunakan, yaitu *Adam* dan *SGD*. Nilai evaluasi yang dihasilkan pada *optimizer Adam* adalah sebesar 32 % pada akurasi R^2, pada MAE sebesar 0.0068 dan untuk *RMSE* sendiri sebesar 0.99. Kesimpulan yang didapatkan adalah bahwa *optimizer Adam* merupakan *optimizer* yang lebih baik dalam memprediksi suhu udara dibandingkan *SGD*.

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
8	Analisis Data *Time Series* Menggunakan *LSTM (Long Short Term Memory)* Dan *ARIMA (Autocorrelation Integrated Moving Average)* Dalam Bahasa Python. [22]	*Ultima InfoSys: Jurnal Ilmu Sistem Informasi*, *11*(1), 1-7.	Adhitio Satyo Bayangkari Karno / (2020)	*Long Short-Term Memory*, *ARIMA*	Menggunakan algoritma *Long Short-Term Memory* dan juga *ARIMA* dalam memprediksi data saham Telkom. Kemudian, membuat masing-masing 7 *model LSTM* dan juga *ARIMA*. Hasil prediksi yang didapatkan adalah nilai *RMSE* terkecil ada pada penggunaan *LSTM* dengan nilai sebesar 1%, sedangkan jika menggunakan *ARIMA* mencapai 2%. Penggunaan *LSTM* juga menunjukkan bahwa akurasi yang didapatkan lebih tinggi dibandingkan dengan *ARIMA*.

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
9	*Cluster-based LSTM network for short-term passenger flow forecasting in urban rail transit.* [23]	*IEEE Access*, *7*, 147653-147671.	Jinlei Zhang, Feng Chen, Qing Shen / (2019)	*Long Short-Term Memory*, *two-step K-Means*	Menggunakan metode *two-step K-Means* untuk melihat tren variasi dari arus penumpang dan karakteristik volume penumpang. Kemudian, menggunakan *Long Short-Term Memory* dengan pendekatan *cluster* untuk memprediksi arus penumpang dalam jangka pendek berdasarkan hasil *clustering* sebelumnya. Hasil yang ditemukan adalah gabungan *model cluster* dan juga prediksi dapat mengurangi kompleksitas jumlah *model* yang harus dibuat, serta meningkatkan performa prediksi.

No	Nama Penelitian	Detil Jurnal	Penulis / Tahun	Metodologi / Algoritma	Hasil dan Simpulan
10	Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU. [24]	*JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(1) , 671-684.	Yadi Karyadi, Handri Santoso / (2022)	*Long Short-Term Memory*, *Bidirectional LSTM*, *GRU*	Memprediksi data kualitas udara dengan menggunakan algoritma *Long Short-Term Memory*, *Bidirectional Long-Short Term Memory*, dan *Gated Recurrent Unit* (*GRU*). Hasil yang didapatkan adalah penggunaan *model LSTM* dan *Bidirectional LSTM* menunjukkan hasil yang bagus jika diterapkan pada data bersifat *time series* dengan nilai *RMSE* yang lebih kecil dibandingkan dengan standar deviasi pada *dataset test*.

Berdasarkan Tabel 2.4 dengan menggunakan *hybrid model K-Means* dengan LSTM dalam memprediksi [16] konsentrasi PM10 di Malaysia sebagai referensi konfigurasi *model LSTM* dan penerapan *clustering*. Selanjutnya penelitian [18] melakukan prediksi menggunakan LSTM mengenai kualitas udara dan suhu di Kota Bandung. Pada penelitian [19] melakukan perbandingan penggunaan *CNN*, *LSTM*, *BPNN* dan *hybrid model CNN – LSTM* untuk memprediksi kualitas udara di Kota Beijing. Penelitian yang menjadi acuan *hybrid clustering* adalah [15] dimana menggunakan *hybrid model K-Means* dengan *PAM* atau disebut *K-Medoids* untuk memprediksi status pasien COVID-19 pada klinik di Iraq. Penelitian ini sendiri akan menggunakan beberapa metode seperti *hybrid* antara algoritma *clustering* dengan LSTM seperti pada [16] tetapi tidak hanya menggunakan *K-Means* melainkan akan menggunakan *hybrid model* yang terdapat pada [15] menggunakan *K-Means* dan *K-Medoids*. Pembeda penelitian ini dengan penelitian terdahulu lainnya adalah pada metode dan juga data. Oleh karena itu, penelitian ini akan menggunakan model *hybrid K-Means* dan *K-Medoids* dengan *Long Short-Term*

Memory (LSTM) dalam memprediksi data Indeks Standar Pencemar Udara (ISPU) DKI Jakarta tahun 2010 – 2023 ataupun tahun 2021 – 2023. Penelitian terdahulu lainnya pada Tabel 2.4 akan menjadi referensi mengenai metode *clustering* atau prediksi. Selain itu, penggunaan 2 *optimizer* yang berbeda mengacu pada penelitian [21], yaitu dengan menggunakan *Adam* dan juga *SGD*.

2.2 Tinjauan Teori

2.2.1 Lingkungan Hidup

2.2.1.1 Kualitas Udara

Kualitas udara atau *air quality* (*AQ*) ditentukan berdasarkan partikel dan komponen gas yang ada di atmosfer sekitar seperti PM2.5, PM10, O3, SO2, NO2, dan CO karena partikel-partikel tersebut menjadi perhatian utama untuk kesehatan manusia hingga perubahan iklim [25]. Pengukuran kualitas udara tersebut dilakukan dengan menggunakan *air quality index* (AQI) yang merupakan metode efektif untuk mengukur level polusi udara sehingga menjadi acuan untuk penanganan risiko atau mengontrol tingkat polusi bagi pemerintah [26]. Pengukuran *air quality index* (*AQI*) harian tersebut dapat dilakukan dengan cara kalkulasi rata-rata konsentrasi polutan udara PM2.5, PM10, SO2, NO2, dan CO selama 24 jam dan konsentrasi harian maksimum rata-rata O3 selama 8 jam [27].

Tabel 2. 2 Pengukuran Indeks Kualitas Udara

KATEGORI	NILAI INDEKS	DESKRIPSI
Baik	0 - 50	Kualitas udara baik dan polusi udara hanya sedikit atau tidak ada risiko
Moderat	51- 100	Kualitas udara masih dapat diterima, tetapi terdapat risiko pada beberapa individu terutama yang sensitif terhadap polusi
Tidak Sehat Bagi Kelompok Sensitif	101 - 150	Terdapat pengaruh kesehatan pada anggota keompok sensitif
Tidak Sehat	151 - 200	Beberapa anggota masyarakat mengalami efek kesehatan dan anggota kelompok sensitif mengalami efek yang lebih serius
Sangat Tidak Sehat	201 - 300	Risiko kesehatan meningkat untuk semua anggota masyarakat
Berbahaya	≥ 301	Semua anggota masyarakat kemungkinan besar akan terpengaruh

Tabel 2.1 [28] merupakan nilai indeks kualitas udara polutan udara dan ozon (O3), serta kategorinya berdasarkan dampak pada masyarakat berdasarkan pengukuran Badan Perlindungan Lingkungan Amerika Serikat (US EPA). Untuk nilai indeks pada kategori baik dimulai dari 0 – 50 dan yang tertingginya adalah kategori berbahaya di angka ≥ 301. Oleh karena itu, level atau kategori udara yang baik memiliki tingkat polutan udara yang sedikit.

2.2.1.2 Pencemaran Udara

Pencemaran udara merupakan kondisi dimana jumlah konsentrasi polutan udara mempengaruhi kondisi lingkungan dan kesehatan masyarakat yang diakibatkan dari meningkatnya industrialisasi sehingga membuat udara semakin kotor [29]. Misalnya, di Indonesia sendiri pertumbuhan penduduk, ekonomi, hingga urbanisasi dan industrialisasi meningkatkan konsentrasi polutan udara seperti PM2.5 ataupun PM10 [30]. Salah satu daerah di Indonesia, yaitu DKI Jakarta memiliki salah satu penyebab dari tingginya angka pencemaran udara yang diakibatkan dari penggunaan kendaraan pribadi karena jumlahnya lebih banyak dari kendaraan umum [31]. Tingginya tingkat pencemaran udara akan menurunkan angka dari indeks kualitas udara dan dapat menimbulkan banyak permasalahan seperti perubahan iklim hingga kesehatan masyarakat banyak.

2.2.1.3 Indeks Standar Pencemar Udara (ISPU)

Indeks Standar Pencemar Udara (ISPU) adalah angka yang tidak mempunyai satuan dan menggambarkan kondisi mutu udara ambien pada lokasi tertentu yang didasarkan kepada dampak terhadap kesehatan manusia, nilai estetika dan juga makhluk hidup lainnya [10]. Penjelasan dari ISPU tersebut tercantum pada Pasal 1 dari Peraturan Menteri Lingkungan Hidup dan Kehutanan nomor 14 tahun 2020 dimana ISPU sendiri merupakan indeks atau angka dari kualitas udara sebagaimana *air quality index* (*AQI*).

Tabel 2. 3 Penggolongan Kategori Indeks Standar Pencemar Udara (ISPU)

RENTANG ANGKA	KATEGORI	PENJELASAN
1 - 50	Baik	Tingkat mutu udara sangat baik dan tidak memberikan efek negatif pada makhluk hidup
51 - 100	Sedang	Tingkat mutu udara masih dapat diterima oleh makhluk hidup
101 - 200	Tidak Sehat	Tingkat mutu udara bersifat merugikan makhluk hidup
201 - 300	Sangat Tidak Sehat	Tingkat mutu udara dapat meningkatkan risiko kesehatan pada kelompok masyarakat yang terpapar
≥ 301	Berbahaya	Tingkat mutu udara dapat merugikan kesehatan secara serius pada masyarakat dan perlu penanganan cepat

Tabel 2.2 [32] merupakan tabel untuk kategori dari masing-masing rentang angka pada Indeks Standar Pencemar Udara (ISPU) yang juga merupakan angka kualitas udara dan ditetapkan oleh Kementerian Lingkungan Hidup dan Kehutanan (KLHK) Indonesia. Kemudian juga terdapat konversi nilai konsentrasi parameter ISPU berdasarkan peraturan yang sudah ditetapkan pada tabel berikut:

Tabel 2. 4 Konversi Nilai Konsentrasi Parameter ISPU

ISPU	24 Jam PM10 (µg/m^3)	24 Jam PM2.5 (µg/m^3)	24 Jam SO_2 (µg/m^3)	24 Jam CO (µg/m^3)	24 Jam O_3 (µg/m^3)	24 Jam NO_2 (µg/m^3)	24 Jam HC (µg/m^3)
0 – 50	50	15,5	52	4000	120	80	45
51 – 100	150	55,4	180	8000	235	200	100
101 – 200	350	150,4	400	15000	400	1130	215
201 – 300	420	250,4	800	30000	800	2260	432
>300	500	500	1200	45000	1000	3000	648

Tabel 2.3 merupakan konversi nilai konsentrasi pada parameter yang ada di data ISPU. Konversi nilai tersebut akan menjadi acuan untuk

perhitungan ISPU itu sendiri. Perhitungannya sendiri adalah berdasarkan nilai ISPU batas atas, ISPU batas bawah, ambien batas atas, ambien batas bawah, dan yang terakhir adalah konsentrasi ambien hasil pengukuran [32]. Tata cara perhitungan ISPU tersebut adalah seperti pada persamaan 2.1 berikut:

$$I = \frac{(Ia - Ib)}{(Xa - Xb)} (Xx - Xb) + Ib \quad (2.1)$$

Keterangan:

I = ISPU terhitung

Ia = ISPU batas atas

Ib = ISPU batas bawah

Xa = Konsentrasi ambien batas atas (µg/m3)

Xb = Konsentrasi ambien batas bawah (µg/m3)

Xx = Konsentrasi ambien nyata hasil pengukuran (µg/m3)

2.3 *Framework* dan Algorima

2.3.1 *Data Mining*

Data mining adalah metode yang sering digunakan oleh perusahaan atau organisasi untuk mendapatkan informasi bermakna dari data mentah yang diambil sehingga dapat membantu bisnis perusahaan tersebut untuk mempelajari mengenai *customers*, memprediksi perilaku *customers* dan meningkatkan strategi marketing bisnis [33]. Oleh karena itu, pada penelitian ini diperlukan proses *data mining* dengan menggunakan algoritma *clustering* untuk menemukan suatu *pattern* pada sebuah *dataset*.

2.3.1.1 *CRISP-DM*

CRISP-DM (*Cross Industry Standard for Data Mining*) adalah sebuah metode yang populer dan merupakan proses model untuk melakukan *data*

mining dimana metode tersebut berupa siklus dengan terdiri dari enam fase berulang dimulai dari *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment* [34]. Tahapan tersebut ada pada Tabel 2.4 seperti berikut:

Tabel 2. 5 Tahapan *CRISP-DM*

Tahapan	Penjelasan
Business Understanding	Pada tahapan pertama ini, dilakukan penilaian situasi bisnis untuk melihat apa saja sumber daya yang tersedia dan dibutuhkan. Kemudian, tujuan utama dari tahapan ini adalah menentukan capaian dari proses *data mining* dengan meninjau tipe *data mining* yang akan digunakan, serta melakukan *planning* terhadap proses selanjutnya.
Data Understanding	Tahapan kedua ini adalah melakukan pengambilan data, melakukan eksplorasi terhadap data, dan melihat kualitas data tersebut dari sumber data. Langkah ini biasanya dilakukan dengan menjelaskan dan mendeskripsikan mengenai data tersebut menggunakan analisis statistik.
Data Preparation	Setelah tahapan pengambilan data dilakukan, maka pada tahapan ini adalah melakukan pembersihan data (*cleaning data*) terhadap data yang tidak cocok atau *bad data*. Selain itu, tahapan ini juga melakukan pemilihan data apa saja yang akan digunakan pada pembuatan model di tahapan selanjutnya.
Modeling	Tahapan pemodelan data ini adalah melakukan pemilihan teknik, dan membuat model serta menguji model tersebut. Pemilihan teknik yang akan digunakan dalam pembuatan model ini harus sesuai dengan tujuan bisnis yang sudah ditentukan pada tahapan awal dari *CRISP-DM*. Selain itu, untuk membangun model, maka diperlukan suatu parameter yang harus ditetapkan, serta melakukan evaluasi model terhadap kriteria yang ada dan memilih model terbaik.
Evaluation	Pada tahapan ini merupakan tahapan evaluasi dari yang dihasilkan model tersebut. Dengan demikian, hasil tersebut harus dijelaskan dan harus ada suatu *action* tertentu, serta tahapan dan proses yang sudah dijalankan sebelumnya ditinjau kembali.
Deployment	Tahapan *deployment* ini merupakan tahapan untuk memilih bagaimana model yang sudah dievaluasi sebelumnya diterapkan secara langsung untuk membantu tujuan bisnis yang sudah ditetapkan. Penerapan tersebut dapat berupa pembuatan *report* bisnis atau dapat berupa suatu *software* yang ditempatkan pada proses bisnis.

2.3.2 Algoritma *Clustering*

Algoritma *clustering* merupakan pembelajaran tanpa pengawasan (*unsupervised learning*) yang berasal dari proses *clustering*, yaitu membagi data

ke dalam grup dengan tujuan untuk menangkap kemungkinan pengelompokan alami dalam data dengan cara memisahkan populasi ke dalam kelompok yang berbeda [35]. Jadi, *clustering* dapat didefinisikan juga sebagai proses dari teknik yang ada pada *machine learning* dalam mengelompokkan data yang tanpa label ke dalam suatu *cluster*. Teknik *clustering* sendiri dalam bisnis sering digunakan untuk melakukan analisis segementasi pasar untuk kebutuhan *marketing* suatu perusahaan.

Teknik *clustering* tersebut bertujuan untuk menemukan pola distribusi pada sebuah kumpulan data dengan melihat kesamaan objek berdasarkan nilai-nilai atribut pada data yang saling berdekatan dan biasanya memiliki simbol titik dalam *plot* diagram [36]. Pola distribusi yang dihasilkan akan dilakukan segmentasi berdasarkan objek yang memiliki kesamaan atribut sehingga akan terlihat titik-titik data yang dikelompokkan sesuai kelompok *cluster*.

2.3.2.1 *Time Series Clustering*

Time Series Clustering merupakan salah satu cara dari algoritma *clustering* untuk dapat menganalisis data deret waktu (*time series*). Teknik *time series clustering* sangat cocok dalam menangani data yang bersifat dinamis seperti data deret waktu (*time series*)[37]. Salah satu pengukuran jarak pada analisis *time series clustering* adalah menggunakan metode *Dynamic Time Warping* (DTW) dengan tujuan untuk mengukur jarak serta mencari jalur optimum antara dua data deret waktu [38].

2.3.2.2 *K-Means*

Salah satu algoritma *clustering* adalah *K-Means*. Algoritma *K-Means* melakukan *clustering* secara *partitioning* dengan memisahkan data pada sebuah dataset ke dalam kelompok yang berbeda-beda. *K-Means* juga melakukan *clustering* dengan cara meminimalisir jarak antara titik data dengan *cluster* data tersebut. Algoritma *K-Means* memilih sebagian data pada dataset untuk menjadikannya sebagai pusat *cluster* (*centroid*) secara *random* dari populasi data. Kemudian, *K-Means* akan menghitung jarak *centroid* sampai semua titik data dikelompokkan ke dalam tiap-tiap *cluster*

yang akan membentuk *cluster* baru [39]. Berikut merupakan proses yang terdapat pada *K-Means* [40] :

1. Menentukan jumlah *cluster* yang ingin dibuat berdasarkan nilai n, yang dimana nilai n tersebut menjadi jumlah cluster.
2. Menetapkan nilai secara *random* untuk *centroid* (pusat *cluster*) sebanyak nilai n, dengan menggunakan rumus *euclidean distance* dalam menentukan jarak setiap data terhadap masing-masing pusat *cluster* pada persamaan 2.2 berikut:

$$d(xi, \mu j) = \sqrt{\sum(xi - \mu j)^2} \quad (2.2)$$

Keterangan:
xi = data kriteria
μj = pusta *cluster* pada *cluster* ke-j

3. Mengelompokkan masing-masing data berdasarkan jarak yang terkecil atau dekat dengan pusat *cluster*.
4. Mencari nilai *centroid* baru yang diperoleh menggunakan rumus rata-rata dari *cluster* pada persamaan 2.3 berikut:

$$\mu(t+1) = \frac{1}{Nsj} \sum_{j \in sj} xj \quad (2.3)$$

Keterangan:
μj(t+1) = pusat *cluster* (*centroid*) baru pada perulangan (t+1)
N*sj* = Data pada cluster *sj*

5. Melakukan perulangan dari langkah ke-2 hingga ke-5 jika data setiap *cluster* belum berhenti sampai anggota setiap *cluster* tidak ada yang berubah atau tetap.

2.3.2.3 *K-Medoids*

Algoritma *clustering* yang lainnya adalah *K-Medoids*. Algoritma *K-Medoids* melakukan proses *clustering* dengan mencari titik yang paling

representatif dari dataset. Titik tersebut dihitung berdasarkan jarak dalam suatu kelompok dari semua kemungkinan titik yang membuat jarak titik antar *cluster* lebih besar. *K-Medoids* sangat cocok untuk mengatasi *noise* dan *outlier* pada dataset yang menyebabkan penyimpangan dalam distribusi data [41]. Tahapan yang ada pada *K-Medoids* adalah sebagai berikut [42] :

1. Menetapkan pusat *cluster* (*centroid*) sebanyak nilai n atau jumlah *cluster*.
2. Menetapkan setiap objek ke dalam *cluster* paling dekat dengan menggunakan rumus *euclidean distance* yang sama seperti pada K-Means.
3. Menetapkan objek secara *random* di setiap *cluster* untuk menjadi pusat *cluster* pada titik data terpilih (*medoid*) yang baru.
4. Menghitung jarak yang ada di setiap objek pada setiap *cluster* dengan pusat *cluster* pada titik data terpilih (*medoid*) yang baru.
5. Menghitung total simpangan (S) yang didapatkan dari nilai total jarak baru dan dikurangi oleh nilai total jarak lama. Jika nilai S kurang dari 0 (S < 0), maka objek ditukar dengan data *cluster* sehingga membentuk sejumlah *cluster* objek baru sebagai pusat *cluster* pada titik data terpilih (*medoid*).
6. Melakukan perulangan tahap ke-3 hingga ke-5 jika *medoid* masih berubah, jika tidak ada yang berubah maka diperoleh jumlah *cluster* dan anggota *cluster* pada masing-masing *cluster*.

2.3.3 *Davies Bouldin Index*

Davies Bouldin Index atau yang disebut sebagai *DBI* adalah sebuah metode untuk melakukan validasi atau pengecekan terhadap hasil algoritma *clustering*. Pengujian pada DBI sendiri berdasarkan jumlah dari kemiripan data terhadap pusat *cluster* terhadap *cluster* tersebut (nilai kohesi) dan jarak antara pusat *cluster* dari *cluster* (nilai separasi) tersebut. Penentuan *cluster* optimal pada DBI adalah berdasarkan nilai separasi yang tinggi tetapi memiliki nilai kohesi yang rendah. Jika nilai DBI semakin mendekati nilai 0, maka semakin baik *cluster*

yang diperoleh dari pengelompokan *clustering* yang digunakan, dengan begitu, semakin rendah nilai DBI, maka *cluster* yang dihasilkan juga optimal [43].

2.3.4 *Silhouette Score*

Selain *DBI*, terdapat metode validasi untuk menentukan jumlah *cluster* optimal pada algoritma atau metode *clustering*, yaitu dengan menggunakan *Silhouette Score*. Pengujian yang dilakukan pada *Silhouette Score* adalah berdasarkan jarak rata-rata di dalam *cluster* dan jarak rata-rata *cluster* terdekat untuk setiap titik data [44]. Penentuan *cluster* yang optimal juga berdasarkan nilai *Silhouette Score* yang tinggi. Jika nilai mendekati angka 1 secara positif, maka *cluster* tersebut dikatakan baik, sedangkan jika nilai angka mendekati angka 1 secara negatif, maka *cluster* tersebut kurang baik.

2.3.5 *Root Mean Square Error* (*RMSE*)

Root Mean Square Error atau disebut dengan RMSE merupakan salah satu bentuk evaluasi pada sebuah model prediksi dengan cara melakukan penjumlahan dari kuadrat *error* atau selisih antara nilai aktual dengan nilai prediksi yang dibagi dengan banyaknya waktu data peramalan dan kemudian menarik akarnya [45]. Persamaan 2.4 berikut adalah rumus dari *RMSE*:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(Y_i - \hat{Y}_i)^2}{n}} \quad (2.4)$$

Keterangan:

n = Jumlah data

Y_i = Nilai aktual

$\hat{Y}_i$ = Nilai prediksi

Akurasi dari nilai RMSE sendiri dikatakan memiliki akurasi yang tinggi jika nilai yang dihasilkan rendah, begitu juga sebaliknya jika nilai yang dihasilkan tinggi maka tingkat keakuratan semakin rendah [46].

2.3.6 *Pearson Correlation*

Korelasi *pearson* adalah uji statistik untuk menyatakan kuat atau lemahnya hubungan antara satu variabel dengan variabel lainnya [47]. Uji korelasi tersebut dilakukan untuk melihat bagaimana hubungan atau korelasi yang dimiliki antara variabel-variabel sehingga hal tersebut dapat memberikan informasi yang berguna. Nilai korelasi tersebut dapat bernilai positif yang berarti jika salah satu variabel mengalami kenaikan, maka variabel satunya lagi mengikuti kenaikan tersebut. Nilai korelasi yang bernilai negatif sendiri menunjukkan bahwa jika salah satu variabel mengalami kenaikan, maka variabel lainnya yang diuji mengalami penurunan, begitu juga sebaliknya. Kemudian pada penelitian ini, korelasi variabel yang kuat didapatkan jika nilai korelasi lebih besar dari 0.6, sedangkan jika di bawah 0.6 menunjukkan korelasi yang lemah.

2.3.7 *Augmented Dickey-Fuller Test*

Tes statistik *Augmented Dickey-Fuller* (ADF) merupakan salah satu cara untuk menguji stasioneritas pada data *time series* yang cederung tidak stasioner [48]. Hasil pengujian menghasilkan *p-value* dengan ketentuan tingkat signifikansi sebesar 5 %. Jika *p-value* yang dihasilkan setelah uji ADF lebih besar dari 5 %, maka data yang diuji tersebut tidak stasioner, sebaliknya jika *p-value* lebih kecil dari 5 %, maka data tersebut dikatakan stasioner.

2.3.8 *Neural Network, Reccurent Neural Network* (*RNN*), dan *Long Short-Term Memory* (*LSTM*)

2.3.8.1 *Neural Network*

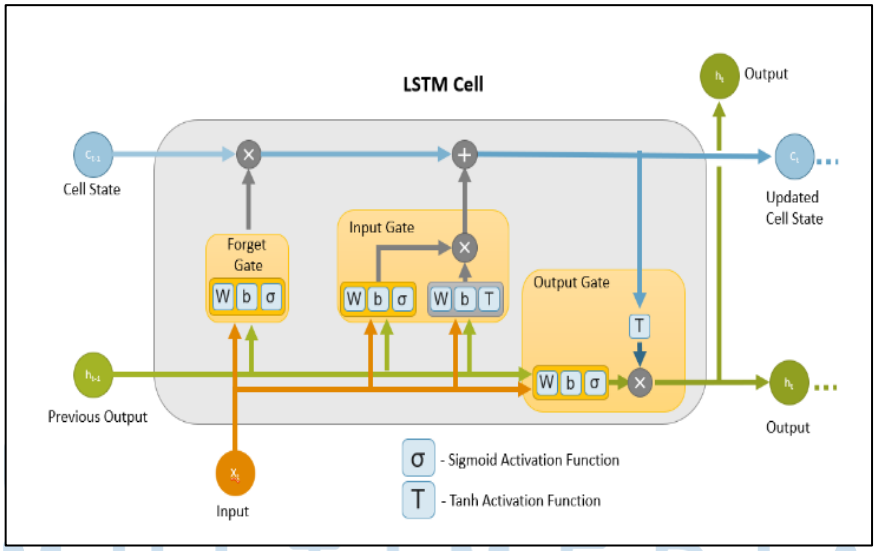
Neural Network merupakan salah satu cara yang sering digunakan untuk mengimplementasikan kecerdasan mesin dimana metode tersebut mengadopsi cara kerja *neuron* dalam otak manusia sehingga dapat disebut sebagai "*true machine learning*" [49]. Kemampuan dari *Neural Network* sendiri adalah dengan membuat komputer dapat berpikir seperti otak manusia sehingga juga mampu mengolah informasi dan mempelajarinya secara cepat [50]. Selain itu, adanya *networks* atau jaringan yang dapat mengetahui hubungan dan dependensi pada suatu data, serta responsif terhadap variabel input [51].

2.3.8.2 *Recurrent Neural Network* (*RNN*)

Recurrent Neural Network (RNN) adalah salah satu kategori dari *Artificial Neural Network* (ANN) yang dikembangkan untuk memproses dan prediksi data yang berbentuk *sequential* atau berurutan [52]. RNN juga sangat cocok digunakan untuk melakukan *time-series* dimana informasi yang sudah ada sangat berguna dalam memprediksi di masa depan, dengan kata lain RNN mampu memecahkan permasalahan antara informasi masa lalu untuk digunakan di masa depan [53]. Salah satu implementasi RNN ada pada *Long Short-Term Memory* (LSTM) yang mampu mengatasi kekurangan dari RNN itu sendiri [54].

2.3.8.3 *Long Short-Term Memory* (*LSTM*)

Long Short-Term Memory atau disingkat LSTM adalah tipe dari *Recurrent Neural Network* (RNN) yang dikembangkan untuk mengatasi kekurangan dari RNN seperti permasalahan gradien yang menghilang dan meledak [55]. LSTM juga dikenal memiliki kemampuan yang lebih unggul dalam membuat model prediksi [56]. Berikut ini adalah arsitektur dari LSTM:



Gambar 2. 1 Arsitektur *LSTM*

Pada Gambar 2.1 [57] merupakan arsitektur dari *Long Short-Term Memory* (*LSTM*) yang memiliki tiga lapisan, yaitu *input*, *hidden* dan *output*.

Untuk bagian *hidden layer* atau lapisan tersembunyi, terdapat tiga gerbang, yaitu *Forget Gate*, *Input Gate*, dan juga *Output Gate* [56]. Ketiga gerbang memiliki tugasnya masing-masing, misalnya untuk *Forget Gate* memiliki kegunaan dalam menentukan informasi dan jumlah informasi yang akan dihapus pada *cell state*. Berikut ini adalah rumus dari *Forget Gate* tersebut[57]:

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \quad (2.5)$$

$$c_f = c_{t-1} * f_t \quad (2.6)$$

Kemudian *Input Gate* berguna dalam memperbanyak hasil antara dua *functional units* dan menambah hasil tersebut pada *cell state*. Berikut ini adalah persamaan 2.7, 2.8, dan 2.9 yang merupakan rumus dari *Input Gate*[57]:

$$\tilde{c}_t = \tanh(W_c . [h_{t-1}, x_t] + b_c) \quad (2.7)$$

$$m_t = \sigma(W_m . [h_{t-i}, x_t] + b_m) \quad (2.8)$$

$$c_t = c_f + \tilde{c}_t * m_t \quad (2.9)$$

Selanjutnya, *Output Gate* akan mengambil informasi relevan dari input yang sekarang dan juga input sebelumnya sampai menghasilkan prediksi [57]. Berikut ini adalah persamaan 2.10 dan 2.11 yang merupakan rumus dari *Output Gate*[57]:

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o) \quad (2.10)$$

$$h_t = o_t * \tanh(c_t) \quad (2.11)$$

2.4 *Tools*

2.4.1 *Python*

Python merupakan bahasa pemrograman tingkat tinggi dengan berorientasi objek yang dimana *syntax* pada *Python* mudah dipelajari serta tersedia banyak *standard library* dan dapat digunakan tanpa biaya untuk semua *platform* umum [58]. Dengan kata lain, *Python* menjadi salah satu bahasa program yang mudah dipelajari bagi pengguna awam, serta ada banyak *library* yang disediakan dalam

Python dan beberapa *library* tersebut bersifat *open sources* sehingga pengguna lainnya dapat berkontribusi untuk mengembangkan salah satu *library* tersebut.

2.4.2 *Jupyter Notebooks*

Pada dasarnya, *Jupyter Notebooks* suatu dokumen atau *file* yang berisikan kode-kode program dengan format *.ipynb* yang ditulis pengguna untuk menjalankan suatu program tertentu yang dimana pada *file* tersebut kode dapat dijalankan secara terpisah berdasarkan *cell* [59]. Bahasa pemrograman yang ditulis pada *Jupyter Notebooks* adalah *Python* yang biasanya digunakan untuk keperluan data analisis, terutama implementasi *machine learning*, baik secara *unsupervised* maupun *supervised*. Salah satu fitur yang terdapat pada *Jupyter Notebooks* adalah *checkpoint* ketika menulis kode program yang tujuannya adalah menyimpan kode tersebut pada file secara berkala agar kode tersebut tetap tersimpan ketika penulis kode program tidak melakukan penyimpanan sebelumnya.