BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

2.1.1 Model *Machine Learning* untuk Peramalan Kualitas Udara

Beberapa studi terbaru pada Tabel 2.1 menunjukkan keefektifan algoritma machine learning yang berfokus pada *boosting* dan *ensemble* dalam meningkatkan akurasi dalam memprediksi kualitas udara. Penelitian yang dilakukan oleh Toharudin dkk. (2023) mengusulkan metode XGBoost dan *SHapley Additive Explanations* (SHAP) untuk menangani klasifikasi tingkat konsentrasi $PM_{2.5}$ dengan data yang tidak seimbang di DKI Jakarta dengan akurasi ~54%.

Di sisi lain, penelitian oleh Elbir dkk. (2025) menyoroti estimasi $PM_{2.5}$ yang didasarkan pada data resolusi tinggi dan terintegrasi dengan faktor lingkungan seperti NDVI, emisi, dan parameter meteorologi dengan LightGBM menunjukkan tingkat akurasi tertinggi dengan R = 0.88, RMSE = 6.42 μg/m³, dan MAPE = 24.64%. Penelitian ini juga menegaskan bahwa faktor vegetasi (NDVI) berkontribusi positif terhadap akurasi prediksi, sedangkan tinggi lapisan batas atmosfer, radiasi matahari, dan suhu udara muncul sebagai variabel yang paling dominan.

Secara keseluruhan, kedua studi tersebut menunjukkan bahwa algoritma yang berbasis boosting seperti LightGBM dan XGBoost efektif dalam mengatasi kompleksitas interaksi nonlinier antara polutan dan variabel meteorologi, serta memberikan dasar yang kuat bagi pengembangan model prediksi kualitas udara yang lebih adaptif dan akurat di masa depan.

Tabel 2.1. *State of The Art* Model Prediksi Kualitas Udara

Judul	Jurnal/ Prosiding	Metode yang Digunakan	Hasil Penelitian
Boosting Algorithm to Handle Unbalanced Classification of $PM_{2.5}$ Concentration Levels by Observing Meteorological Parameters in Jakarta-Indonesia (Toharudin dkk., 2023)	IEEE Access (2023)	AdaBoost XGBoost,CatBoost, LightGBM	Menerapkan AdaBoost, XGBoost, CatBoost, dan LightGBM dan *SHapley Additive Explanations* (SHAP) untuk klasifikasi level $PM_{2.5}$ di Jakarta menggunakan data BMKG 2015–2022 dengan akurasi ~54% untuk prediksi polusi udara di Jakarta dengan rata-rata konsentrasi $PM_{2.5}$ 148 µg/m³.
Rainfall Prediction Model in Semarang City Using Machine Learning (Usman dkk, 2023)	Indonesian Journal of Electrical Engineering and Computer Science, Vol. 30, No. 2, pp. 1224–1231 (2023)	*Multiple Linear Regression, Random Forest Regression,* dan *Artificial Neural Network*	Hasil menunjukkan bahwa *Artificial Neural Network* (ANN) memiliki performa terbaik dengan RMSE = 13.055 dan MAE = 6.621, diikuti oleh *Random Forest Regression* (RMSE = 13.403, MAE = 6.643) dan *Multiple Linear Regression* (RMSE = 13.475, MAE = 7.843). Model ANN dinilai paling akurat dalam memprediksi curah hujan di Kota Semarang.
Evaluating Machine Learning-Based $PM_{2.5}$ Estimation Using Integrated High-Resolution Datasets Across NDVI Levels in an Urban-Industrialized Region (Elbir dkk., 2025)	Environmental Pollution (2025)	*Random Forest,* XGBoost, CatBoost, LightGBM	Mengintegrasikan data resolusi tinggi (NDVI, meteorologi, atmosferik) untuk estimasi $PM_{2.5}$ dengan RF, XGBoost, CatBoost, dan LightGBM. LightGBM memberikan hasil terbaik (R=0.88; RMSE=6.42 µg/m³; MAPE=24.64%), menyoroti pengaruh vegetasi dan suhu terhadap $PM_{2.5}$.
AiCareBreath: IoT-Enabled Location-Invariant Novel Unified Model for Predicting Air Pollutants to Avoid Related Respiratory Disease (Borah dkk., 2024)	IEEE Internet of Things Journal, Vol. 11, No. 8 (2024)	LightGBM-*Random Forest*, XGBoost, LSTM, GRU, Transformer, dan CNN.	Model LightGBM-*Random Forest* berhasil memprediksi NO_2, O_3, CO, SO_2, $PM_{2.5}$, dan PM_{10} secara akurat di Malaysia, India, Filipina dengan R^2 = 0.96–0.99 dan RMSE sangat rendah (1.27–4.83 µg/m³).

Tabel 2.1. *State of The Art* Model Prediksi Kualitas Udara (lanjutan)

Judul	Jurnal/ Prosiding	Metode yang Digunakan	Hasil Penelitian
Analyzing Hybrid Machine Learning Performance Through Air Quality Prediction (Rajhans dkk., 2025)	International Conference on Computing Technologies & Data Communication (ICCTDC)	Hybrid Model : • **Model 1:** *RFR +ExtraTreesRegressor* • **Model 2:** *RFR +AdaBoostRegressor + XGBR + ExtraTreesRegressor (Stacking + Averaging)*	Model 1 (*Random Forest + Extra Trees*) dan Model 2 (*Random Forest + AdaBoost + XGBoost + Extra Trees*) menunjukkan kinerja terbaik dengan R^2 = 0.8517 dan 0.8484 serta MSE = 2200.807 dan 2249.7899. Model 2 memiliki kemampuan generalisasi yang lebih tinggi melalui pendekatan *stacking* dan *averaging*.
Efficient AQI Prediction: A Comparative Study of Artificial Neural Networks, LSTM, Random Forest, and Gradient Boosting Techniques (Pawanekar dkk., 2024)	Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)	ANN, LSTM, *Random Forest, Gradient Boosting*	*Random Forest* menghasilkan prediksi AQI paling akurat (R^2 = 0.959; RMSE = 0.205), mengungguli ANN, LSTM, dan *Gradient Boosting*.
Unveiling Hidden Air Quality Patterns: A Data-Driven Predictive Regression Analysis for Sustainable Air Management (Cyrus dkk., 2024)	2nd International Conference on Networking and Communications (ICNWC)	*Linear Regression, Ridge, Lasso, Elastic Net, Decision Tree, Random Forest, Gradient Boost, AdaBoost, XGBoost*	XGBoost menunjukkan performa terbaik dengan R^2 = 0.878, RMSE = 21.08, MAE = 12.87 (Manali) dan R^2 = 0.838, RMSE = 14.91, MAE = 7.98 (Velachery).
Enhancing Air Quality Index Forecasting through Machine Learning Models (Reddy dkk., 2025)	3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)	*LR*, SVM, *Decision Tree, RF, AdaBoost*	*Decision Tree Regressor* menunjukkan performa terbaik MAE = 1.4455, RMSE = 6.0109, R^2 = 0.9981.

2.1.2 Peran *Attention Mechanism* dalam Prediksi Kualitas Udara

Attention Mechanism berfungsi sebagai elemen krusial dalam model prediksi kualitas udara, karena attention dinilai dapat meningkatkan performa model dalam mendeteksi faktor yang paling berpengaruh dan meningkatkan akurasi model seperti disampaikan pada Tabel 2.2.

Seperti pendekatan yang dikembangkan oleh Li dkk. (2024), model STARes-SaLSTM mengombinasikan tiga komponen. Pertama, *Spatiotemporal Attention* yang terdiri atas *Spatial Attention* dan *Temporal Attention* untuk memberikan pembobotan adaptif pada kota yang paling berpengaruh dan urutan waktu yang paling relevan. Kedua, *Residual Blocks* (ResNet) yang menjaga stabilitas aliran informasi dan mencegah vanishing gradient sehingga kemampuan ekstraksi fitur mendalam tetap terjaga. Ketiga, SaLSTM, yaitu mekanisme *self-attention* yang menggantikan perhitungan Q, K, dan V dengan tiga sel LSTM agar pola temporal tidak hilang selama proses encoding.

Data yang digunakan mencakup polutan dan variabel meteorologi dari 10 kota di China selama periode 2020–2021, terdiri dari 24 fitur ($PM_{2.5}$, PM_{10}, SO_2, NO_2, CO, O_3, *temperature*, *relative humidity*, *air pressure*, *wind speed*, *wind direction*, *precipitation*, *visibility*, *sky condition*, *dew point temperature*, *solar radiation*, *ground temperature*, *maximum temperature*, *minimum temperature*, *atmospheric water content*, *vapor pressure*, *barometric tendency*, *sunshine duration*, dan *surface wind pressure*. Berdasarkan hasil eksperimen, pada dua skenario, yaitu *single-step prediction* untuk prediksi satu jam ke depan dan *multi-step prediction* hingga 24 jam. Selain itu, dilakukan *ablation study* untuk mengevaluasi kontribusi masing-masing modul, ResNet, *Spatiotemporal Attention*, dan SaLSTM menunjukkan bahwa seluruh komponen memberikan peningkatan signifikan terhadap akurasi prediksi. STARes-SaLSTM menunjukkan performa terbaik dengan RMSE 6.716, MAE 4.466, Corr 0.981, dan IA 0.988 untuk prediksi satu jam ke depan, mengungguli CNN-LSTM, ConvLSTM, RCL-Learning, GCNTAG, dan ST-Transformer.

Tabel 2.2 Peran *Attention Mechanism* dalam Pemodelan Prediksi Kualitas Udara

Judul	Jurnal/ Prosiding	Metode yang Digunakan	Hasil Penelitian
Residual Neural Network with Spatiotemporal Attention Integrated with Temporal Self-Attention Based on Long Short-Term Memory Network for Air Pollutant Concentration Prediction (Li dkk., 2024)	Atmospheric Environment	STAResNet-SaLSTM CNN-LSTM, ConvLSTM, *RCL-Learning*, GCNTAG, ST-Transformer	Dari dua skenario *single-step prediction* untuk prediksi satu jam ke depan dan *multi-step prediction* hingga 24 jam, STARes-SaLSTM (Spatiotemporal Attention Residual Blocks–Self-Attention Long Short-Term Memory) menunjukkan performa terbaik dengan RMSE 6.716, MAE 4.466, Corr 0.981, dan IA 0.988 untuk prediksi satu jam ke depan, mengungguli CNN-LSTM, ConvLSTM, *RCL-Learning*, GCNTAG, dan ST-Transformer.
MGAtt-LSTM: A Multi-Scale Spatial Correlation Prediction Model of $PM_{2.5}$ Concentration Based on Multi-Graph Attention (Zhang dkk., 2024)	Environmental Modelling & Software	MGAtt-LSTM, GAT-LSTM, GC-LSTM, CNN-LSTM	Model MGAtt-LSTM *(Multi-Graph Attention-Long Short Term Memory)* menghasilkan prediksi interval per jam (1, 12, 24, 48) paling akurat dengan MAE = 9,72 µg/m³ dan RMSE = 18,93 µg/m³ pada jendela prediksi 1 jam.
Predicting the Complex Variation Characteristics of Equipment Heat Dissipation in Office Buildings via CNN-LSTM-ATT, Multiple Regression, and Similar Day Models (Li dkk., 2025)	Energy and Buildings	CNN-LSTM-ATT, *Multiple Regression*	Model CNN-LSTM-ATT menunjukkan hasil paling akurat dengan MAPE = 4,661% dan CV = 0,014, mampu menangkap variasi pengeluaran energi panas pada peralatan elektronik, listrik, maupun mekanis (*Equipment Heat Dissipation*) di gedung perkantoran untun menjaga kestabilan suhu ruangan.

2.1.3 Kesenjangan dengan Penelitian Sebelumnya

Penelitian-penelitian terdahulu mengenai prediksi kualitas udara telah banyak memanfaatkan model *machine learning* dan *deep learning*, namun tetap terdapat perbedaan metodologi, karakteristik wilayah, serta pendekatan pemodelan sebagaimana disampaikan pada Tabel 2.3.

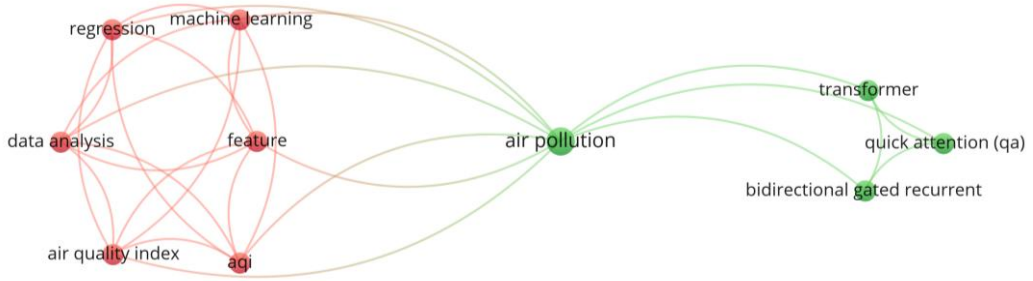
Tabel 2.3 Kesenjangan dengan Penelitian Sebelumnya

No	Judul (Sumber/Tahun)	Kesenjangan dengan Penelitian Sebelumnya
1	Boosting Algorithm to Handle Unbalanced Classification of $PM_{2.5}$ Concentration Levels by Observing Meteorological Parameters in Jakarta-Indonesia (Toharudin dkk., 2023)	Penelitian sebelumnya mengonversi data per jam (65.020 sampel) menjadi data harian (2.358 sampel), peneltian saat ini membuat pola fluktuasi kualitas udara per jam (105.466 sampel)
		Penelitian sebelumnya menggunakan split 80:20 *(train:test)* untuk klasifikasi dan evaluasi dengan F1/MCC, menggunkan split 70:15:15, dan evaluasi dengan RMSE, MAPE, MAE dan uji statistic
2	Evaluating Machine Learning-Based $PM_{2.5}$ Estimation Using Integrated High-Resolution Datasets Across NDVI Levels in an Urban-Industrialized Region. (Elbir dkk., 2025)	Penelitian sebelumnya berhasil memetakan sebaran $PM_{2.5}$ berbasis NDVI dan suhu permukaan, namun tidak menangkap fluktuasi per jam yang umum di Jakarta. Penelitian ini menambah *slicing time sequence* (1, 8, 12, 24, 48 jam)

Tabel 2.3 Kesenjangan dengan Penelitian Sebelumnya (lanjutan)

No	Judul (Sumber/Tahun)	Kesenjangan dengan Penelitian Sebelumnya
3	MGAtt-LSTM: A Multi-Scale Spatial Correlation Prediction Model of $PM_{2.5}$ Concentration Based on Multi-Graph Attention (Zhang dkk., 2024)	Penelitian sebelumnya berhasil menangkap korelasi spasial lewat *multi-layer graph*, namun pendekatan itu kurang relevan untuk DKI Jakarta yang padat, datar, dan sangat dipengaruhi cuaca lokal. Penelitian ini memakai CatBoost dengan *Feature Attention Weighting* yang lebih sesuai untuk karakteristik perkotaan Jakarta.
		Pendekatan sebelumnya membutuhkan komputasi tinggi (NVIDIA GeForce RTX 3090 * 4). Penelitian ini memilih metode yang lebih ringan dan cepat (GPU Tesla T4) dan tetap menghasilkan output akurat.

Pola variasi tersebut turut tergambar pada pemetaan bibliometrik melalui diagram *clustering network* pada Gambar 2.1 *network map* memperlihatkan bahwa *air pollution* berperan sebagai simpul penghubung antara dua *cluster* utama, yakni *cluster* berbasis pendekatan konvensional (misalnya regresi, analisis data, fitur, dan AQI) dengan *machine learning* serta *cluster* berbasis model *deep learning* dengan spasial-temporal modern (seperti Transformer, BiGRU, dan *attention mechanism*). Oleh karena itu, penelitian ini menempati posisi strategis untuk menjembatani kedua *cluster* ini dengan mengkombinasikan model *machine learning* dengan *deep learning* (pembobotan *attention mechanism*).



Gambar 2.1 Peta *Cluster Network Keyword* Penelitian

2.2. Dasar Teori

2.2.1 Polusi Udara

Polusi udara merupakan isu lingkungan yang mendesak karena keberadaan gas berbahaya, partikel ($PM_{2.5}$ dan PM_{10}), serta asap yang menurunkan kualitas udara dan mengancam kesehatan manusia serta ekosistem. Syuhada dkk. (2023) menunjukkan bahwa paparan jangka panjang $PM_{2.5}$ meningkatkan risiko kematian dini, penyakit kardiovaskular, gangguan pernapasan, dan berdampak serius pada kesehatan anak. Di Jakarta—dengan populasi lebih dari 10,5 juta jiwa—polusi udara menyebabkan lebih dari 10.000 kematian per tahun, lebih dari 5.000 kasus rawat inap kardiovaskular, serta lebih dari 7.000 dampak kesehatan pada anak seperti stunting dan kematian bayi. Konsentrasi $PM_{2.5}$ tahunan mencapai 52 $\mu g/m^3$, sementara kadar maksimum harian O_3 hampir dua kali batas NAAQS, memicu peningkatan kasus penyakit jantung, gangguan pernapasan, hingga COPD. Dampak ekonomi polusi juga signifikan, mencapai kerugian sekitar USD 2,94 miliar atau 2,2% PDRB, terutama akibat kematian prematur dan biaya perawatan kesehatan. Temuan ini menegaskan bahwa polusi udara tidak hanya mengancam kesehatan jangka panjang, tetapi juga membebani fasilitas kesehatan dan menurunkan produktivitas. Karena itu, diperlukan kolaborasi pemerintah, industri, dan masyarakat untuk pengawasan kualitas udara yang berkelanjutan. Berikut penjelasan polutan udara menurut WHO (2021):

a) $PM_{2.5}$ (Partikulat dengan Diameter Kurang dari $_{2.5}$ Mikrometer)

$PM_{2.5}$ adalah partikel udara halus dengan diameter lebih kecil dari 2,5 mikrometer yang dapat menembus saluran pernapasan hingga paru-paru dan bahkan aliran darah. Sumber utama $PM_{2.5}$ adalah pembakaran bahan bakar fosil, kendaraan bermotor, dan pembakaran biomassa. Paparan jangka panjang terhadap $PM_{2.5}$ dapat meningkatkan risiko penyakit kardiovaskular, stroke, dan gangguan pernapasan. $PM_{2.5}$ memiliki batas tahunan ≤ 5 µg/m³ dan batas harian ≤ 15 µg/m³ berdasarkan pedoman WHO.

b) PM_{10} (Partikulat dengan Diameter Kurang dari 10 Mikrometer)

PM_{10} adalah partikel yang lebih besar dari $PM_{2.5}$, tetapi masih cukup kecil untuk mempengaruhi saluran pernapasan. Sumber utama PM_{10} termasuk kendaraan bermotor, pembangkit listrik, dan konstruksi. Paparan PM_{10} dapat menyebabkan gangguan pernapasan, penyakit jantung, dan stroke. PM_{10} memiliki batas tahunan ≤ 15 µg/m³ dan batas harian ≤ 45 µg/m³.

c) SO_2 (Sulfur Dioksida)

SO_2 adalah gas yang dihasilkan dari pembakaran bahan bakar yang mengandung sulfur, seperti batubara dan minyak. Paparan SO_2 dapat menyebabkan iritasi saluran pernapasan, sesak napas, serta memperburuk penyakit paru-paru dan asma. WHO merekomendasikan batas harian SO_2 sebesar 40 µg/m³.

d) NO_2 (Nitrogen Dioksida)

NO_2 adalah gas berbahaya yang dihasilkan dari pembakaran bahan bakar, terutama kendaraan dan pembangkit listrik. Paparan NO_2 dapat menyebabkan gangguan pernapasan, asma, serta meningkatkan risiko penyakit kardiovaskular. WHO menetapkan NO_2 memiliki batas tahunan ≤ 10 µg/m³ dan batas harian ≤ 25 µg/m³.

e) CO (Karbon Monoksida)

CO adalah gas beracun yang dihasilkan dari pembakaran bahan bakar yang tidak sempurna, seperti pada kendaraan bermotor dan pembangkit listrik. Paparan CO dapat mengurangi kemampuan darah untuk mengangkut oksigen, menyebabkan pusing, sakit kepala, dan bahkan kematian pada konsentrasi tinggi. WHO menetapkan CO memiliki batas harian ≤ 4 mg/m³.

f) O_3 (Ozon)

Ozon troposferik terbentuk dari reaksi kimia antara nitrogen oksida (NOx) dan senyawa organik volatil (VOCs) yang dipengaruhi sinar matahari. Ozon dapat menyebabkan iritasi saluran pernapasan, batuk, dan memperburuk kondisi pernapasan seperti asma. WHO merekomendasikan O_3 memiliki batas harian ≤ 100 µg/m³ tiap 8 jam.

2.2.2 Klimatologi dan Meteorolgi

Untuk memahami pola polusi udara Jakarta secara temporal dan spasial, penelitian ini memanfaatkan data klimatologi dan meteorologi per jam dari NCEI–NOAA periode 2020–2024. Tabel 2.4 menyajikan dataset serta lokasi stasiun pemantauan AQI dan meteorologi di DKI Jakarta, yang mencakup lima stasiun Indeks Standar Pencemar Udara (ISPU) yang dikelola oleh Kementerian Lingkungan Hidup dan Kehutanan (KLHK) dan empat stasiun meteorologi dari jaringan Global Historical Climatology Network Hourly (GHCN-H) yang dikelola oleh National Oceanic and Atmospheric Administration (NOAA).

Penentuan lokasi dan koordinat stasiun ISPU tersebut telah mengacu pada Peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia Nomor 7 Tahun 2020 tentang Kualitas Udara Ambien, yang menetapkan bahwa titik pemantauan harus ditempatkan pada udara terbuka dengan aliran udara bebas, memiliki ketinggian inlet minimal 2 meter di atas permukaan tanah, serta berjarak minimal sekitar 20 meter dari sumber emisi langsung dan hambatan fisik seperti bangunan atau pepohonan agar hasil pengukuran merepresentasikan kualitas udara

ambien umum, bukan pengaruh lokal sesaat (Kementerian LHK, 2020). Ketentuan ini menegaskan bahwa satu stasiun pemantauan tidak dimaksudkan untuk merepresentasikan satu titik koordinat secara mikro, melainkan kondisi kualitas udara pada wilayah administratif yang lebih luas dengan karakteristik aktivitas manusia dan emisi yang relatif homogen. Sejalan dengan regulasi nasional tersebut, pedoman internasional seperti WHO Ambient Air Quality Monitoring Siting Guidelines menekankan bahwa stasiun pemantauan harus ditempatkan pada lokasi yang representatif secara spasial, memiliki jarak yang memadai dari sumber polusi lokal dan hambatan fisik, serta berada pada lingkungan yang stabil secara penggunaan lahan agar mampu menggambarkan kondisi kualitas udara suatu kawasan perkotaan secara umum (World Health Organization, 2008). Pedoman ini juga menegaskan bahwa dalam konteks perkotaan, satu stasiun pemantauan dapat merepresentasikan area dengan skala ratusan meter hingga beberapa kilometer, tergantung pada homogenitas sumber emisi, topografi, dan dinamika meteorologi setempat.

Untuk stasiun meteorologi, prinsip penentuan lokasi juga mengikuti standar internasional yang serupa. World Meteorological Organization (WMO) dan praktik teknis EPA menyatakan bahwa stasiun meteorologi dirancang untuk merepresentasikan kondisi atmosfer regional, bukan batas administratif mikro, dengan kriteria utama berupa keterbukaan lahan, minim gangguan struktur buatan, serta posisi sensor yang memungkinkan pengukuran suhu, kelembapan, dan angin secara alami (World Meteorological Organization, 2018; United States Environmental Protection Agency, 2016). Oleh karena itu, keberadaan stasiun meteorologi seperti Halim Perdanakusuma, Jakarta Observatory, Tanjung Priok, dan Soekarno–Hatta dapat digunakan untuk mewakili kondisi meteorologi wilayah DKI Jakarta dan sekitarnya dalam analisis spasial-temporal.

Berdasarkan kombinasi ketentuan nasional (Permen LHK) dan pedoman internasional (WHO, WMO, dan EPA) tersebut, maka lokasi stasiun pada Tabel 2.4 dapat dinyatakan telah memenuhi prinsip representativitas spasial, di mana setiap stasiun mewakili kondisi kualitas udara dan meteorologi pada wilayah administratif

tertentu di DKI Jakarta, bukan hanya satu titik koordinat sempit. Pendekatan ini valid dan lazim digunakan dalam studi kualitas udara perkotaan serta mendukung penggunaan data stasiun sebagai dasar analisis spasial-temporal dan pemodelan prediksi AQI dalam penelitian ini.

Tabel 2.4 Dataset dan Lokasi Stasiun AQI dan Meteorologi DKI Jakarta

Nama Dataset	Pengelola	Kode	Lokasi Stasiun	Koordinat (Lat, Long)	Wilayah Administratif
Indeks Standar Pencemar Udara (ISPU)	Kementerian Lingkungan Hidup dan Kehutanan (KLHK)	DKI 1	Bundaran HI	-6.1955, 106.8229	Jakarta Pusat (area Monumen Nasional & Hotel Indonesia)
		DKI 2	Kelapa Gading	-6.1540, 106.9033	Jakarta Utara (Kecamatan Kelapa Gading)
		DKI 3	Jagakarsa	-6.3230, 106.8200	Jakarta Selatan (Kecamatan Jagakarsa)
		DKI 4	Lubang Buaya	-6.3380, 106.9400	Jakarta Timur (Kecamatan Cipayung)
		DKI 5	Kebon Jeruk	-6.1760, 106.7790	Jakarta Barat (Kecamatan Kebon Jeruk)
Global Historical Climatology Network Hourly (GHCN-H)	National Oceanic and Atmospheric Administration (NOAA)	IDI0000WIHH	Halim Perdanakusuma	-6.2666, 106.8911	Jakarta Timur (Kecamatan Makasar, area Bandara Halim)
		IDU096749-1	Soekarno-Hatta	-6.1200, 106.6500	Kabupaten Tangerang, Provinsi Banten (Bandara Internasional Soekarno-Hatta)
		IDU096745-1	Jakarta Observatory	-6.1800, 106.8300	Jakarta Pusat (area Menteng–Gambir)
		IDU096741-1	Tanjung Priok	-6.1000, 106.8700	Jakarta Utara (kawasan Pelabuhan Tanjung Priok)

Berdasarkan penempatan stasiun pemantauan yang memenuhi kriteria teknis dan representativitas spasial sesuai regulasi nasional dan pedoman internasional, data kualitas udara dan meteorologi yang digunakan mampu merepresentasikan dinamika atmosfer di tiap wilayah administratif DKI Jakarta. Dengan resolusi temporal per jam, data ini memungkinkan identifikasi variasi dan perubahan kualitas udara secara lebih rinci, baik dalam skala waktu harian maupun antarwilayah. Hal ini menunjukkan bahwa analisis data cuaca per jam memungkinkan deteksi perubahan kualitas udara dengan akurasi spasial dan temporal yang sangat tinggi. Dalam rangka melakukan pemodelan prediksi, diperlukan beberapa variabel klimatologi, di antaranya:

a) Suhu *(Temperature)*

Suhu merupakan parameter meteorologi yang mengukur energi kinetik rata-rata partikel di atmosfer, biasanya dinyatakan sebagai suhu kering *(dry-bulb)*. Studi terbaru oleh Li dkk. (2024) menunjukkan bahwa anomali suhu permukaan terhadap konsentrasi $PM_{2.5}$. Peningkatan suhu 1°C dapat menaikkan kadar $PM_{2.5}$ hingga 1 µg/m³ pada musim panas. Suhu tinggi juga memicu pembentukan polutan sekunder dan mempercepat reaksi fotokimia, sehingga dapat memperburuk kualitas udara. WHO merekomendasikan kisaran suhu ideal pada rentang 20–25°C untuk menjaga kualitas udara yang lebih stabil.

b) Suhu Titik Embun *(Dew Point Temperature)*

Suhu titik embun adalah suhu saat udara mencapai kejenuhan uap air sehingga embun mulai terbentuk. Parameter ini merepresentasikan kandungan uap air di atmosfer dan berkaitan erat dengan tingkat kelembapan. Konsentrasi parameter ini bergantung dengan konsentrasi suhu udara.

c) Kecepatan Angin *(Wind Speed)*

Kecepatan angin mencerminkan laju pergerakan udara di atmosfer dan berperan penting dalam proses transportasi dan dispersi polutan. Menurut Fathollahi dkk. (2023) menunjukkan bahwa kecepatan angin yang tinggi dapat mengurangi konsentrasi $PM_{2.5}$ melalui proses pencampuran dan pemindahan polutan dari satu

wilayah ke wilayah lain. Sebaliknya, angin lemah dapat menyebabkan akumulasi dan membentuk kantong-kantong polusi berbahaya, terutama di area dengan struktur bangunan padat. WHO merekomendasikan kecepatan angin berada pada rentang 0–15 m/s.

d) Arah Angin *(Wind Direction)*

Arah angin menunjukkan dari mana angin bertiup dan dinyatakan dalam derajat kompas. Arah angin penting dalam menentukan asal-usul polutan, baik dari sumber lokal maupun lintas batas wilayah.

e) Kelembapan Relatif *(Relative Humidity)*

Kelembapan relatif adalah rasio antara kandungan uap air di udara dengan jumlah maksimum yang dapat ditahan udara pada suhu tertentu. Parameter ini menunjukkan sejauh mana udara mendekati kejenuhan. Menurut Matthaios dkk. (2024), kelembapan tinggi dapat memicu pertumbuhan higroskopik partikel dan meningkatkan ukuran serta massa $PM_{2.5}$, serta memperparah kejadian haze. Peningkatan RH sebesar 10% bahkan ditemukan mampu menurunkan efisiensi penyebaran polutan hingga 5% melalui proses kondensasi dan deposisi basah. WHO menyarankan kelembapan relatif berada pada kisaran 40–60%.

f) Curah Hujan 24 Jam *(Precipitation 24 Hour)*

Proses hujan membantu mengurangi konsentrasi polutan melalui mekanisme pencucian atmosfer *(wet deposition)*. Ketika intensitas hujan meningkat, partikel seperti $PM_{2.5}$ dan PM_{10} cenderung terlarut atau tersapu dari atmosfer, sehingga konsentrasinya menurun. Konsentrasi *precipitation* bergantung pada musim.

2.2.3 Statistik Deskriptif

Statistik deskriptif merupakan tahap awal yang penting dalam analisis data, digunakan untuk merangkum dan menyajikan data melalui ukuran seperti rata-rata, median, modus, varians, dan standar deviasi. Pada fase eksplorasi, statistik deskriptif juga mencakup distribusi frekuensi serta visualisasi seperti histogram dan boxplot, yang membantu mengenali pola dan anomali sebelum analisis lanjutan.

Dalam studi, metode ini dianggap fundamental karena memberikan gambaran awal mengenai tren dan penyebaran polusi udara.

a) Rata-rata *(Mean)*

Rata-rata pada persamaan (2.1) merupakan nilai tengah dari sekumpulan angka, dihitung dengan cara menambahkan semua nilai data yang ada (x_i) kemudian membaginya dengan total jumlah data (n).

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.1}$$

b) Median

Median merupakan angka yang terletak di tengah data yang sudah disusun. Apabila jumlah data tidak genap, median adalah angka yang terletak pada posisi tengah. Jika jumlah data genap, median dihitung dengan mengambil rata-rata dari dua angka di posisi tengah.

c) Modus

Modus merujuk pada angka yang paling kerap muncul dalam kumpulan data.

d) Varian

Varian pada persamaan (2.2) menilai sejauh mana data menyebar dari nilai rata-rata $(\bar{x})$ ke nilai data ke-i (x_i).

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{2.2}$$

e) Standar Deviasi

Standar deviasi pada persamaan (2.3) merupakan akar dari varians, yang memberikan indikasi mengenai sebaran data dalam unit yang serupa dengan data aslinya.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.3}$$

Dengan x_i nilai ke-i, $\bar{x}$ rata-rata dari semua x_i dan n adalah jumlah data. Apabila deviasi standar semakin tinggi, maka variasi data akan semakin besar jika dibandingkan dengan nilai rata-ratanya.

f) Rentang *(Range)*

Rentang pada persamaan (2.4) menggambarkan perbedaan antara nilai maksimum (x_{maks}) dan milai minimum (x_{min}) dalam sebuah dataset.

$$Range = x_{maks} - x_{min} \tag{2.4}$$

g) Koefisien Variasi *(Coefficient Variation)*

Koefisien variasi pada persamaan (2.5) menunjukkan seberapa signifikan sebaran data relatif terhadap nilai rata-ratanya. Ini sangat berguna untuk membandingkan variasi antar dataset dengan skala yang berbeda. Dengan x_i adalah nilai ke-i, $\bar{x}$ rata-rata dari semua x_i dan n adalah jumlah data dan $\bar{x}$ adalah rata-rata.

$$CV = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}}{\bar{x}} X\ 100\% \tag{2.5}$$

2.2.4 Deret Waktu *(Time Series)*

Deret waktu atau *time series* merupakan sekumpulan data yang dicatat secara urut pada interval waktu yang konsisten. Analisis data ini biasanya dilakukan untuk mengungkap pola, seperti tren, musiman, dan fluktuasi acak. Menurut Dudek (2023), struktur fundamental dari *time series* dapat digambarkan dalam bentuk aditif pada persamaan (2.6).

$$Y_t = T_t + S_t + R_t \tag{2.6}$$

atau dengan cara multiplikatif pada persamaan (2.7),

$$Y_t = T_t \times S_t \times R_t \tag{2.7}$$

T_t adalah komponen tren, S_t adalah unsur musiman, dan R_t adalah residual/*noise*.

2.2.5 Spatiotemporal Attention

Pendekatan *attention* yang diperkenalkan oleh Li dkk. (2024) menggabungkan dua mekanisme, yaitu *spatial attention* dan *temporal attention*, yang berfungsi menonjolkan informasi paling relevan baik antar lokasi *(spatial)* maupun sepanjang urutan waktu *(temporal)* dengan bobot.

Tahap pertama adalah *Spatial Attention Module* pada Gambar 2.2 (a) yang bertujuan menekankan lokasi geografis yang paling informatif. Pada tahap ini, fitur masukan F diringkas menggunakan *Global Average-Pooling* dan *Global Max-Pooling*, menghasilkan $F^{avg} = AvgPool(F)$ dan $F^{max} = MaxPool(F)$. Keduanya kemudian dikonkatenasi dan diproses melalui konvolusi *3×3* serta fungsi *Sigmoid*, sehingga menghasilkan *spasial attention map*. Peta ini digunakan untuk memperkuat fitur masukan melalui perkalian $M_S \otimes F$, sehingga hanya pola *spasial attention* pada persamaan (2.8) yang diteruskan ke tahap berikutnya.

$$M_S = \sigma(f^{3\times3}([F^{avg}; F^{max}])) \tag{2.8}$$

Tahap kedua adalah *Temporal Attention Module* pada Gambar 2.2 (b), yang menilai kontribusi setiap langkah waktu historis terhadap prediksi saat ini. Fitur spasial $F' = M_S \otimes F$ kembali diringkas menggunakan *Global Pooling*, menghasilkan $F_t^{avg} = AvgPool(F')$ dan $F_t^{max} = MaxPool(F')$. Kedua representasi ini diproses oleh *Shared Neural Network* berbasis *convolution* 1×1, dan pola *temporal attention* yang dirumuskan pada persamaan (2.9).

$$M_T = \sigma(f^{1\times1}(F_t^{avg}) + f^{1\times1}(F_t^{max})) \tag{2.9}$$

Bobot M_T memungkinkan model memfokuskan perhatian pada rentang waktu yang benar-benar berpengaruh.

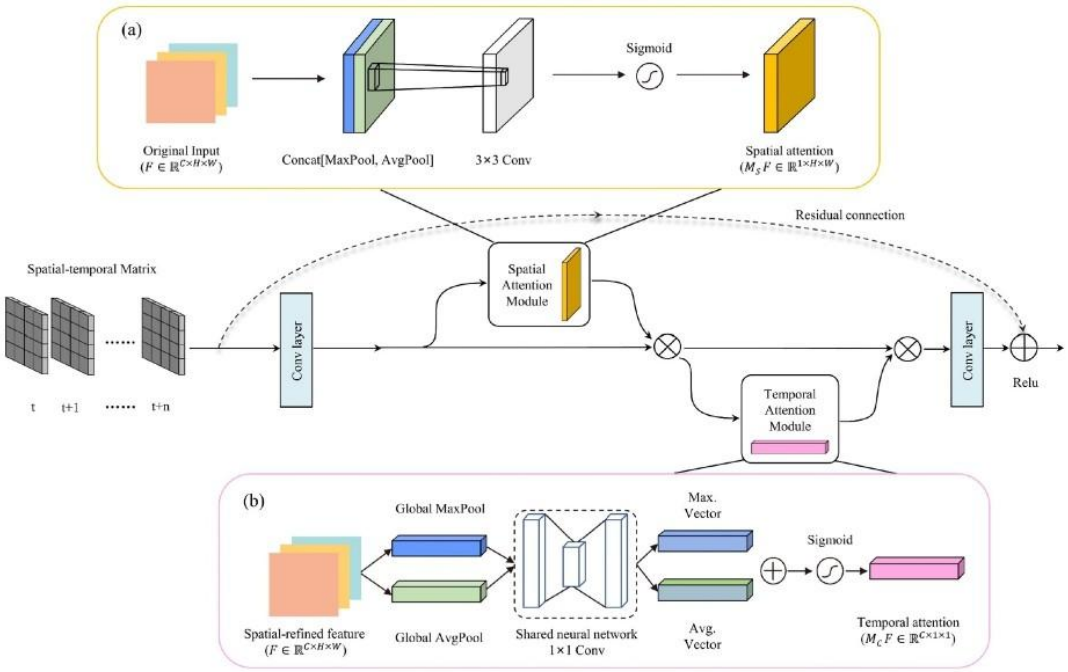
Kemudian, kedua pola ini digabungkan dalam aliran data utama melalui persamaan (2.10).

$$F^{att} = M_T \otimes (M_S \otimes F) \tag{2.10}$$

dan hasilnya dikembalikan ke jalur awal melalui *Residual Connection* pada persamaan (2.11).

$$F^{out} = F^{att} + F \tag{2.11}$$

Koneksi residual menjaga aliran informasi tetap stabil, membantu mencegah *vanishing gradient* (kondisi ketika nilai gradien mengecil hingga hampir hilang selama backpropagation, sehingga *network* kesulitan memperbarui bobot dan proses *learning* menjadi sangat lambat), serta membuat kinerja model lebih efisien tanpa mengulang proses *learning* dari awal.



Gambar 2.2 Arsitektur Spasiotemporal Attention

2.3 Model *Machine Learning*

2.3.1 Linear Regression (LR)

Linear Regression adalah metode pemodelan statistik yang digunakan untuk menunjukkan hubungan secara linier antara variabel dependen, seperti konsentrasi $PM_{2.5}$, dengan satu atau lebih variabel independen, seperti suhu, kelembapan, dan kecepatan angin. Rumus umum LR disampaikan pada persamaan (2.12).

$$Y = \beta_0 + \beta_1 X + \varepsilon \quad (2.12)$$

β_0 adalah *intercept*, β_1 adalah *koefisien regresi*, dan ε adalah *error term*. Model ini memiliki ciri yang interpretatif, sederhana, dan efisien dalam menangani data spasio-temporal yang jumlahnya besar, namun juga cukup sensitif terhadap outlier, sehingga memerlukan data yang sudah dibersihkan. Dalam konteks prediksi kualitas udara, regresi linier dipilih karena kemampuannya memberikan gambaran awal mengenai hubungan antar variabel secara kuantitatif, serta mudah diimplementasikan dan dipahami oleh para pengambil kebijakan.

Berdasarkan studi Aljohani & Aljohani (2025), *Linear Regression* yang dihibrida dengan *Ridge Regression* dan diatur oleh mekanisme *switch* dapat menghasilkan prediksi *Remaining Useful Life (RUL)* baterai dengan *optimals*, setelah fitur dari jalur *correlated* dan *uncorrelated* digabungkan dan diproses melalui lapisan *Dense* (dengan aktivasi *Switch* dan *Dynamic Switch*), outputnya tidak langsung menjadi prediksi tunggal. Sebaliknya, output dari lapisan *Dense* terakhir bercabang ke dua model regresi tradisional, *Ridge Regression* dan *Linear Regression*. *Ridge Regression* adalah bentuk regresi linier yang menambahkan istilah regularisasi L2 ke fungsi *loss*. Tujuannya adalah untuk mengurangi overfitting dengan menyusutkan koefisien regresi, sehingga model menjadi lebih stabil dan mampu menggeneralisasi ke data baru serta kurang peka terhadap perubahan kecil dalam data pelatihan. *Linear Regression* adalah model regresi dasar yang memodelkan hubungan linier antara variabel independen dan variabel dependen. Diagram menunjukkan adanya *"Dense (Units = 1, activation = 'Dynamic Switch')"* yang tampaknya berfungsi sebagai gerbang atau penentu bobot antara hasil output *Ridge Regression* dan *Linear Regression*.

Ini menunjukkan bahwa model tidak hanya memilih satu dari dua metode tersebut, tetapi mungkin melakukan penggabungan adaptif antara prediksi dari kedua metode tersebut. Mekanisme *Dynamic Switch* ini berarti bahwa bobot atau pemilihan model (Ridge atau Linear) bisa berubah secara dinamis, tergantung pada karakteristik data masukan atau bahkan tahap pelatihan model.

Tujuan dari penggabungan kedua metode ini adalah untuk memanfaatkan kelebihan masing-masin, *Linear Regression* mungkin lebih tepat untuk menangkap hubungan linier langsung, sedangkan *Ridge Regression* memberikan stabilitas dan kemampuan generalisasi yang lebih baik dengan mengatasi *multicollinearity* dan *overfitting*. Dengan *Dynamic Switch* model bisa menyesuaikan kontribusi masing-masing regresi secara fleksibel guna menghasilkan prediksi RUL yang optimal dalam berbagai kondisi.

2.3.2 Lasso Regression

Lasso Regression (Least Absolute Shrinkage and Selection Operator Regression) merupakan pengembangan dari regresi linier yang menggunakan pendekatan *regularisasi* untuk meningkatkan akurasi model dan mencegah *overfitting*. Dalam regresi linier konvensional, model hanya berfokus pada meminimalkan RSS (*Residual Sum of Squares)*, yaitu selisih antara nilai aktual (y_i) dan nilai prediksi ($\widehat{y_i}$). Namun, pendekatan tersebut dapat menghasilkan model yang terlalu kompleks, terutama jika jumlah fitur (p) banyak atau terdapat korelasi tinggi antarvariabel. Untuk mengatasi hal ini, *Lasso Regression* menambahkan komponen penalti berupa regularisasi L1 terhadap besar koefisien model (β_j), sehingga fungsi objektif yang diminimalkan menjadi persamaan (2.13).

$$\min_{\beta_0,\beta} \{\frac{1}{2n}\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} | \beta_j | \} \tag{2.13}$$

$y_i$merupakan nilai aktual dari variabel target pada sampel ke-i, x_{ij} adalah nilai fitur ke-j, dan λ merupakan parameter regularisasi yang mengontrol kekuatan penalti. Nilai λ yang besar akan mendorong sebagian koefisien β_j menuju nol, sehingga hanya fitur paling berpengaruh yang dipertahankan. Mekanisme ini membuat Lasso efektif dalam seleksi fitur otomatis sekaligus meningkatkan kemampuan generalisasi model pada data baru. Penelitian oleh Wang dkk. (2022) menunjukkan penerapan berbagai model regresi terpenalti seperti Lasso, Ridge, *Elastic Net,* dan *Adaptive Lasso* pada data RNA-seq kanker payudara yang terdiri dari 230 sampel dan 4606 gen. Hasil penelitian memperlihatkan bahwa Lasso dan

Adaptive Lasso menghasilkan model paling *sparse* dengan akurasi yang kompetitif dibandingkan *Elastic Net*, sementara *Ridge Regression* cenderung mengalami *overfitting* karena tidak melakukan penyusutan koefisien. Selain itu, *Adaptive Lasso* terbukti mampu menjaga keseimbangan antara kompleksitas model dan akurasi prediksi, sehingga direkomendasikan untuk dataset berdimensi tinggi. Temuan tersebut menegaskan bahwa Lasso dan turunannya berperan penting dalam *feature selection*, *prediction*, *network construction*, serta *sample classification* pada berbagai bidang penelitian, termasuk bioinformatika dan analisis spasial.

2.3.3 *Categorical Boosting* (CatBoost) *Regressor*

CatBoost *(Categorical Boosting)* adalah sebuah algoritma GBDT yang dirancang untuk meningkatkan ketepatan, efisiensi, dan konsistensi saat bekerja dengan dataset tabular besar, khususnya yang memiliki fitur kategorikal. Berbeda dengan algoritma boosting lainnya yang menerapkan pertumbuhan pohon berdasarkan level, CatBoost membangun pohon *oblivious* atau *symmetric*, yang memiliki format serupa di setiap tingkat, ini mempermudah optimasi, mempercepat proses prediksi, dan mengurangi kemungkinan terjadinya *overfitting*. Secara matematis, CatBoost mengurangi fungsi kehilangan yang didasarkan pada metode *gradient descent* pada persamaan (2.14).

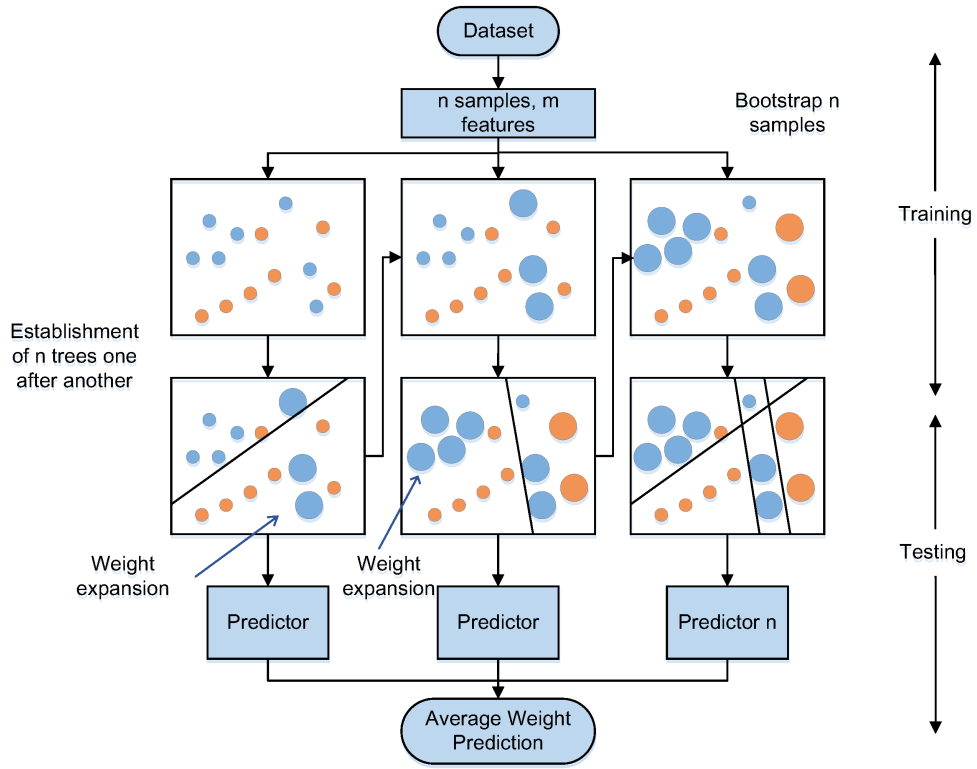
$$L\big(y, f(X)\big) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 \tag{2.14}$$

Pada setiap iterasi ke-t, terdapat pembaruan model dengan menambahkan fungsi pohon regresi baru yang dihasilkan dari gradien negatif kehilangan fungsi sebelumnya pada persamaan (2.15).

$$f_t(x) = f_{t-1} + \eta \,.\, h_t(x) \tag{2.15}$$

Dengan η sebagai tingkat pembelajaran dan $h_t(x)$ sebagai model regresi residual. Kelebihan utama CatBoost terletak pada penggunaannya yang berstrategi ordered boosting, yaitu cara belajar yang mengacak urutan data dan melatih model tanpa memakai informasi target masa depan untuk menghindari kebocoran target. Pendekatan ini membuat CatBoost sangat kokoh baik untuk dataset yang kecil atau

tidak seimbang serta ampuh dalam mengolah fitur kategorikal tanpa perlu *encoding one-hot manual*.



Gambar 2.3 Arsitektur Model Hybrid CatBoost-Bayesian (Yang dkk., 2023)

Mekanisme pembelajaran berlapis CatBoost dapat dipadukan dengan pendekatan Bayesian untuk menciptakan model hibrida yang lebih fleksibel dan akurat. Sebagai ilustrasi, Yang dkk. (2023) merancang model CatBoost–Bayesian Hybrid pada Gambar 2.3 yang bertujuan untuk mengestimasi undrained shear strength pada tanah liat. Model tersebut memanfaatkan prinsip boosting berdasarkan gradien yang dikombinasikan dengan inferensi Bayesian untuk meningkatkan stabilitas dan ketepatan prediksi pada data yang kompleks. Proses ini dimulai dengan pengambilan dataset yang terdiri dari n sampel dan m fitur, dilanjutkan dengan *bootstrap sampling* untuk membangun subset data yang berbeda di setiap iterasi pelatihan. Setiap subset akan digunakan untuk membentuk satu pohon keputusan baru secara berurutan.

Selama tahap pelatihan, dilakukan *weight expansion* yang merupakan pembaruan bobot adaptif untuk setiap observasi berdasarkan tingkat kesalahan prediksi sebelumnya. Observasi dengan kesalahan yang lebih besar akan mendapatkan bobot yang lebih tinggi di iterasi berikutnya, sehingga model lebih fokus dalam memperbaiki prediksi yang kurang tepat. Setiap pohon menghasilkan keluaran parsial yang digabungkan menggunakan mekanisme rata-rata prediksi berbobot hingga menghasilkan model akhir berbasis ansambel dengan bobot rata-rata dari semua pohon. Integrasi konsep pembaruan Bayesian memungkinkan penyesuaian dinamis terhadap ketidakpastian prediksi, menjadikan model lebih tahan terhadap gangguan dan pencilan dalam data. Secara keseluruhan, pendekatan hibrida ini menekankan prinsip dasar CatBoost yang bersifat urutan dan adaptif. Kontribusi setiap pohon diarahkan untuk mengurangi *residual loss function* secara bertahap. Kombinasi antara boosting dan inferensi Bayesian menghasilkan model yang seimbang dalam hal akurasi, kestabilan, dan kemampuan generalisasi model.

Toharudin dkk. (2023) juga mengeksplorasi penggunaan CatBoost untuk memprediksi kualitas udara DKI Jakarta dengan data BMKG Kemayoran 2015–2022. Pada tahap awal, model ini membangun beberapa *weak learners (tree)* yang masih memiliki tingkat kesalahan relatif tinggi. Setiap kali model melakukan prediksi, bobot *(weight)* dari data yang salah diprediksi akan ditingkatkan, sehingga model akan lebih fokus memperbaiki kesalahan sebelumnya. Analisis SHAP *(SHapley Additive Explanations)* menunjukkan bahwa suhu udara, kelembapan relatif, dan konsentrasi $PM_{2.5}$ adalah fitur yang paling dominan dan XGBoost tercatat sebagai model terbaik dengan akurasi ~54% untuk prediksi polusi udara di Jakarta dengan rata-rata konsentrasi $PM_{2.5}$ 148 µg/m³.

2.3.4 *Light Gradient Boosting Machine* (LightGBM) *Regressor*

LightGBM *(Light Gradient Boosting Machine)* adalah algoritma machine learning berbasis *boosting* yang dibuat untuk efisien dan cepat dalam pelatihan, terutama pada dataset besar dan kompleks. LightGBM menggunakan pendekatan pohon yang tumbuh secara *leaf-wise*, berbeda dengan metode *level-wise* yang biasa digunakan, sehingga model ini bisa memperluas cabang pohon dengan cara

menurunkan kesalahan yang paling besar. Secara matematis, model ini membangun fungsi prediksi pada persamaan (2.16).

$$L(y, f(X)) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 \tag{2.16}$$

Pada setiap iterasi, model menambahkan pohon baru yang dibentuk dari gradien negatif fungsi loss sebelumnya, hingga membentuk *ensamble* yang paling baik. LightGBM juga menggunakan teknik *histogram binning* dan pembobotan gradien untuk mempercepat proses pelatihan. Keunggulan utamanya mencakup kemampuan menangani data yang tidak seimbang, efisiensi penggunaan memori, akurasi tinggi, serta performa yang baik pada data tabular dengan banyak fitur. Dalam konteks prediksi kualitas udara spatio-temporal yang didasarkan pada meteorologi dan indeks kualitas udara (AQI), LightGBM dipilih karena kemampuannya menangkap hubungan nonlinier dan kompleks antara polutan seperti $PM_{2.5}$, PM_{10}, CO, NO_2, SO_2, O_3 dengan variabel cuaca seperti suhu, kelembapan, tekanan, dan visibilitas.

Borah dkk., (2024) dalam penelitiannya mengusulkan model LightGBM berbasis IoT untuk memprediksi konsentrasi polutan udara, sehingga bisa mencegah penyakit terkait pernapasan di beberapa lokasi stasiun pemantauan. Model ini menerima data dari berbagai lokasi seperti ID1, ID2, PH3, dan MY3. Data tersebut diproses melalui tahap pra-pemrosesan yang ketat, yaitu pembersihan data menggunakan metode *forward fill*, normalisasi dengan metode *Min-Max scaling*, dekomposisi deret waktu, penambahan fitur melalui *moving average*, serta pembuatan sekuens dengan metode *rolling window*. Setelah itu, model dibagi menjadi dua mekanisme utama untuk memprediksi polutan. Untuk $PM_{2.5}$, PM_{10}, CO, NO_2, SO_2, O_3, mereka menggunakan pendekatan LightGBM secara *cascade*. Hasil prediksi dan sisa dari satu pohon *(tree)* menjadi input untuk pohon berikutnya. Sementara untuk polutan CO, mereka menggunakan model Random Forest untuk prediksi langsung. Prediksi dari kedua mekanisme tersebut kemudian melewati tahap denormalisasi untuk menghasilkan prediksi akhir polutan udara secara komprehensif.

Elbir dkk. (2025) juga menyampaikan bahwa konsentrasi prediksi $PM_{2.5}$ harian dengan LightGBM di wilayah Marmara, Turki, dengan menggabungkan data satelit beresolusi tinggi (MAIAC AOD 1 km), variabel meteorologi dari ERA5, inventaris emisi EDGAR, serta indeks vegetasi NDVI dan kepadatan penduduk memiliki performa superior dibandingkan metode lain seperti Random Forest, XGBoost, dan CatBoost, dengan nilai koefisien korelasi (R) mencapai 0.88 dan Root Mean Square Error (RMSE) sebesar 6.42 $\mu g/m^3$. Penelitian ini membuktikan kemampuannya dalam menangkap hubungan nonlinier yang kompleks antara elemen meteorologi, aktivitas manusia, dan variasi spasial polutan secara efektif, sambil menjaga konsistensi performa dalam berbagai kondisi musim.

Fathollahi dkk. (2023) juga menerapkan LightGBM untuk memperkirakan konsentrasi $PM_{2.5}$ di bagian barat Iran, dengan memanfaatkan kombinasi data AOD dari satelit MODIS dan parameter meteorologi lokal. LightGBM menunjukkan kinerja paling stabil dengan nilai Mean Absolute Error (MAE) sebesar 4.8 $\mu g/m^3$ dan RMSE sebesar 6.1 $\mu g/m^3$, mengungguli model lain seperti Random Forest dan Decision Tree. Keunggulan ini terutama disebabkan oleh strategi pembobotan gradien dan mekanisme pemilihan fitur otomatis, yang membuat model lebih efisien saat berhadapan dengan data besar dan dimensi multivariat.

Secara keseluruhan, hasil penelitian ini menegaskan bahwa LightGBM unggul dalam empat aspek utama, yaitu tingkat akurasi dan kemampuan generalisasi yang tinggi pada data spasiotemporal yang kompleks, efisiensi dalam komputasi berkat teknik *binning* berbasis histogram dan pertumbuhan pohon *leaf-wise*, kemampuan untuk menangani data heterogen dan jumlah fitur yang banyak, serta ketahanan terhadap nilai yang hilang dan distribusi data yang tidak seimbang. Dengan demikian, LightGBM dianggap sangat efektif untuk digunakan dalam riset prediksi indeks kualitas udara (AQI) maupun estimasi konsentrasi polutan seperti $PM_{2.5}$, PM_{10}, SO_2, NO_2, CO, dan O_3 yang didasarkan pada data meteorologi dan satelit.

2.4 Model *Deep Learning*

2.4.1 *Feedforward Neural Network* (FNN)

Feedforward Neural Network (FNN) merupakan salah satu arsitektur dasar dalam bidang deep learning yang digunakan untuk memodelkan hubungan non-linear antar variabel dalam data berskala besar dan kompleks, seperti prediksi kualitas udara yang bersifat spasio-temporal. FNN terdiri dari beberapa lapisan berurutan—lapisan *input, hidden,* dan *output* dengan aliran informasi satu arah tanpa loop, menjadikannya cocok untuk memprediksi variabel kontinu seperti $PM_{2.5}$, PM_{10}, atau indeks kualitas udara (AQI). Setiap neuron dalam FNN melakukan transformasi linier dan non-linear terhadap input yang diterimanya. Secara umum, proses aktivasi dasar pada neuron dirumuskan pada persamaan (2.17).

$$\hat{y} = f\left(\sum_{i=i}^{n} w_i\, x_i + b\right) \tag{2.17}$$

$\hat{y}$ adalah output prediksi, x_i adalah input ke-i, w_i adalah bobot koneksi, b adalah bias, dan f merupakan fungsi aktivasi non-linear, seperti ReLU atau sigmoid.

Implementasi praktis dari FNN umumnya menggunakan bentuk khusus yang disebut Multilayer Perceptron (MLP), yaitu jaringan berlapis ganda, data bergerak secara unidirectional dari input ke output. Proses perhitungan dalam MLP dilakukan dalam beberapa tahap, dimulai dengan perhitungan skor jumlah berbobot pada hidden layer yang dirumuskan pada persamaan (2.18).

$$S_j = \sum_{i=i}^{n} w_{ij} \cdot x_i - \beta_j\,, \quad j = 1, 2, \ldots, h \tag{2.18}$$

dengan S_j adalah skor untuk neuron ke- j pada hidden layer, w_{ij} adalah bobot dari input x_i ke neuron j, dan β_j adalah bias. Aktivasi yang dihasilkan dari hidden layer kemudian dikirim ke output layer untuk menghasilkan prediksi akhir. Proses ini diformulasikan pada persamaan (2.19).

$$\hat{y}_k = \sum_{j=i}^{m} w_{kj} \cdot f_j + b_k \,, \quad k = 1, 2, \ldots, K \tag{2.19}$$

$\hat{y}_k$ adalah output neuron ke- k, w_{kj} adalah bobot yang menghubungkan neuron tersembunyi j ke neuron output k untuk menentukan seberapa besar kontribusi masing-masing aktivasi pada hidden layer terhadap hasil akhir, f_j adalah aktivasi dari neuron *hidden* j, dan b_k adalah bias ke-k yang berfungsi untuk menyesuaikan ambang aktivasi dari neuron output.

Alur kerja FNN dalam penelitian ini dimulai dari tahap pra-pemrosesan data, meliputi interpolasi nilai hilang, normalisasi fitur (seperti *Min-Max Scaling*), dan ekstraksi informasi waktu (jam, hari, musim) untuk membantu model mengenali pola temporal. Data kemudian dimasukkan ke *input layer* sebagai vektor multivariabel, diproses melalui *hidden layer* dengan aktivasi non-linear, dan diteruskan ke *output layer* untuk menghasilkan prediksi nilai AQI atau konsentrasi polutan. Proses pelatihan dilakukan menggunakan algoritma *backpropagation* dan optimasi fungsi loss seperti Mean Squared Error (MSE), dengan pembaruan bobot melalui metode *gradient descent* hingga model mencapai konvergensi.

Untuk meningkatkan representasi sinyal spasio-temporal yang memiliki ciri khas periodik, FNN dikembangkan dengan menambahkan transformasi Fourier ke dalam struktur jaringannya. FNN spektral ini menggunakan gabungan domain waktu dan frekuensi untuk mengekstrak fitur penting dari data yang bersifat siklik atau musiman, seperti pola suhu, kelembapan, atau emisi polutan harian. Secara matematis, prediksi FNN spektral dirumuskan pada persamaan (2.20).

$$\hat{y}(t) = f \sum_{k=i}^{K} \sigma_k \cos(2\pi f_k\, t + \emptyset_k) + \beta_k \, \sin(2\pi f_k\, t + \varphi_k) + Layer\ Fully\ Connected \tag{2.20}$$

f_k adalah frekuensi komponen, σ_k, β_k adalah bobot amplitudo, serta $\emptyset_k, \varphi_k$ adalah fase dari sinyal sinusoidal.

Arsitektur kerja FNN spektral yang terdiri dari beberapa komponen utama, Linear Projection untuk melakukan embedding input, Multi-head Self-Attention untuk menangkap hubungan spasio-temporal, Fast Fourier Transform (FFT) untuk

mengubah sinyal ke domain frekuensi, Matrix Multiplication (MatMul) sebagai operasi penguatan antar dimensi, Inverse FFT (IFFT) untuk mengembalikan sinyal ke domain waktu sebelum masuk ke *layer* prediksi akhir. Pendekatan ini memungkinkan model menangkap informasi frekuensi dominan sekaligus pola temporal yang rumit, sehingga cocok untuk memodelkan data kualitas udara yang menunjukkan dinamika harian maupun musiman.

Pemilihan FNN, baik dalam versi MLP maupun berbasis transformasi Fourier, didasarkan pada karakteristik data kualitas udara yang bersifat spasio-temporal, berulang, dan multivariabel. FNN mampu menggabungkan hubungan non-linear di antara fitur meteorologi dan polusi, sekaligus mengenali pola berulang dalam data. Menurut Chen dkk. (2025), penggabungan CNN dan FNN secara signifikan meningkatkan akurasi prediksi kapasitas baterai dengan menangkap fitur dari domain waktu dan frekuensi secara bersamaan. Pendekatan yang sama diadopsi dalam penelitian ini untuk memprediksi indeks kualitas udara (AQI). Ketersediaan data historis yang besar memungkinkan pelatihan model secara efektif. Model ini juga mampu mengurangi *noise* dan mempercepat proses konvergensi dibandingkan metode berbasis domain waktu murni, serta menunjukkan kemampuan generalisasi yang tinggi terhadap data spasio-temporal yang kompleks.

Studi oleh Doush dkk. (2024) juga menunjukkan bahwa penggabungan FNN dengan model seperti EOL-Jaya (Enhanced Opposition-Based Learning Jaya dengan Archive) adalah algoritma optimasi metaheuristik yang dikembangkan untuk menemukan konfigurasi bobot dan bias optimal pada Multi-Layer Perceptron (MLP) model prediksi kualitas udara. Algoritma ini menggabungkan keunggulan *Jaya* yang bebas parameter, strategi *Opposition-Based Learning* (OBL) untuk eksplorasi ruang solusi, dan *archive mechanism* untuk menyimpan solusi terbaik selama proses *multi-run*.

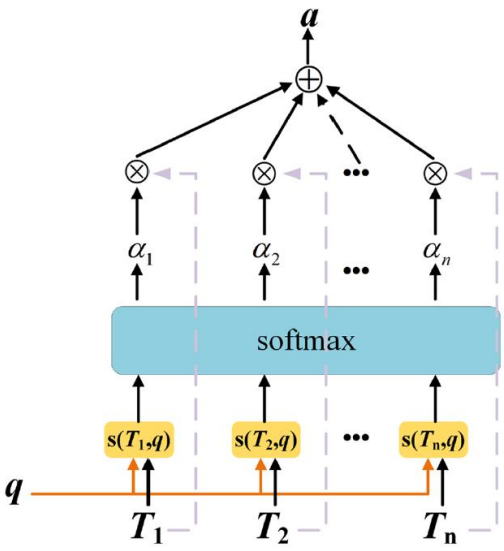
2.4.2 Convolutional Neural Network Long Short-Term Memory

Convolutional Neural Network (CNN) dan Long Short-Term Memory (LSTM) adalah dua jenis arsitektur jaringan saraf dalam deep learning yang memiliki

kekuatan yang saling melengkapi dalam mengekstrak fitur dari data yang kompleks, baik berupa data spasial maupun temporal. CNN menerapkan filter atau kernel pada input untuk menghasilkan feature maps yang menunjukkan pola-pola penting dalam data. Selanjutnya, proses *pooling* dilakukan untuk mengurangi ukuran fitur dan meningkatkan efisiensi komputasi serta mencegah *overfitting*. Setelah beberapa lapisan konvolusi dan *pooling*, lapisan dropout digunakan untuk memastikan model memiliki kemampuan generalisasi yang baik, dengan cara mematikan secara acak sebagian neuron.

Fitur akhir yang telah diekstrak dari CNN kemudian dikirim ke LSTM untuk analisis selanjutnya. X adalah input spasial. Nilai ini dihubungkan dengan hasil pooling yang telah diproses sebelumnya, lalu dikirim ke LSTM untuk dianalisis secara temporal. LSTM menerima output spasial dari CNN sebagai urutan input dan memprosesnya melalui *cell state* dan *gates (input gate, forget gate, dan output gate)*. Setiap gate dikontrol oleh fungsi sigmoid dan $tanh$, yang berperan dalam menyimpan informasi penting dan membuang informasi yang tidak diperlukan. LSTM dirumuskan pada persamaan (2.35), (2.36), (2.37), (2.38), (2.39), dan (2.40).

Gambar 2.4 menggambarkan blok mekanisme attention yang memungkinkan model jaringan saraf, seperti Transformer atau model sekuensial lainnya, untuk secara dinamis memperhatikan bagian input yang paling relevan terhadap suatu pertanyaan tertentu. Dalam proses ini, sejumlah input $T_1, T_2, T_3, \dots, T_n$ (yang merepresentasikan memori atau nilai, seperti kata-kata dalam kalimat) dievaluasi terhadap sebuah pertanyaan q melalui fungsi skor $s(T_i, q)$ yang mengukur tingkat kesamaan atau relevansi antara pasangan tersebut. Hasil dari skor ini kemudian dinormalisasi dengan fungsi $softmax$ untuk menghasilkan bobot perhatian α_i, yang menunjukkan seberapa besar bobot perhatian diberikan kepada setiap T_i. Setiap nilai T_i kemudian dikalikan dengan bobot α_i, dan hasilnya dijumlahkan secara berbobot untuk membentuk vektor konteks akhir α. Proses ini memberikan representasi ringkasan dari input yang menekankan informasi paling relevan terhadap pertanyaan, sehingga memungkinkan model menangkap hubungan penting dalam urutan tanpa kehilangan konteks.



Gambar 2.4 Arsitektur Model Blok Mekanisme Attention (ATT)

Model CNN-LSTM sangat bergantung pada jumlah data yang cukup besar untuk belajar dengan baik representasi *spasiotemporal* yang kompleks. Ini karena proses konvolusi dan pembelajaran berurutan memerlukan banyak parameter yang hanya dapat dioptimalkan dengan stabil jika ada data historis yang panjang dan bervariasi. Dalam penelitian oleh Li dkk. (2025), model CNN-LSTM-ATT terbukti sangat efektif dalam memprediksi variasi kompleks dari panas peralatan bangunan berdasarkan data historis dari berbagai sumber. Integrasi CNN dan LSTM memungkinkan ekstraksi fitur spasial dan temporal secara bersamaan, sementara mekanisme perhatian (attention mechanism) digunakan untuk memberi bobot lebih pada fitur penting. Pendekatan ini menunjukkan bahwa model CNN-LSTM mampu memberikan akurasi prediksi yang tinggi untuk model prediksi.

2.4.3 Gated Recurrent Unit Long Short-Term Memory (GRU-LSTM)

Gated Recurrent Unit (GRU) dan Long Short-Term Memory (LSTM) adalah jenis arsitektur yang lebih canggih dari Recurrent Neural Network (RNN), yang dibuat untuk memproses data berurutan, seperti data yang memiliki ketergantungan jangka panjang atau perubahan dalam waktu. Kedua model ini sering digunakan dalam pemrosesan deret waktu, misalnya dalam memprediksi cuaca, analisis sinyal

spasial, atau mengukur kualitas udara, karena mampu mengatasi kelemahan utama RNN biasa yaitu masalah gradient yang menghilang saat belajar pola dalam jangka waktu lama. GRU dan LSTM bekerja dengan menyimpan informasi dalam sel memori dan menggunakan mekanisme pintu *(gate)* untuk memutuskan apa yang harus dipertahankan, diperbarui, atau diabaikan dalam setiap langkah waktu. Perbedaannya terletak pada jumlah pintu: LSTM memiliki tiga pintu *(input, forget, dan output)*, sedangkan GRU hanya memiliki dua pintu *(reset dan update)*, sehingga GRU lebih ringan secara komputasi namun tetap efektif. Secara matematis, GRU dirumuskan pada persamaan (2.21), (2.22), (2.23) dan (2.34).

$$z_t = \sigma(Wz \cdot [h_{t-1}, x_t]) \quad (2.21)$$

$$r_t = \sigma(Wr \cdot [h_{t-1}, x_t]) \quad (2.22)$$

$$\tilde{h}_t = tanh(Wh \cdot [r_t * h_{t-1}, x_t]) \quad (2.23)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (2.34)$$

z_t adalah *update gate*, r_t adalah *reset gate*, $\tilde{h}_t$ adalah kandidat hidden state, h_t adalah hidden state saat ini, σ adalah fungsi sigmoid, dan x_t adalah input pada waktu t.

LSTM dirumuskan dengan *forget gate* pada persamaan (2.35) untuk menentukan informasi mana yang harus dihapus dari memori yang lama. Apabila tidak relevan, maka akan diabaikan. *Gate* ini memanfaatkan fungsi aktivasi sigmoid untuk menghasilkan nilai di antara 0 dan 1. 0 menunjukkan bahwa informasi sepenuhnya dilupakan, dan 1 menunjukkan bahwa informasi dijaga.

$$f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big) \quad (2.35)$$

f_t adalah nilai yang menentukan seberapa banyak informasi yang akan dilupakan oleh model; σ merupakan fungsi aktivasi sigmoid yang menghasilkan output antara 0 dan 1, W_f adalah bobot yang digunakan untuk *forget gate*, h_{t-1} merupakan keluaran dari waktu sebelumnya, $x_t$adalah input pada waktu t, dan b_f adalah bias yang ditambahkan dalam perhitungan pintu lupa untuk membantu penyesuaian selama proses pelatihan.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2.36)$$

Input gate pada persamaan (2.36) menetapkan data baru mana yang harus disimpan dalam memori. Data yang signifikan akan ditambahkan. *Gate* ini juga memanfaatkan fungsi sigmoid untuk mengatur seberapa banyak data baru yang akan diterima oleh memori sel.

$$C_t = tanh(W_c[h_{t-1}, x_t] + b_c) \quad (2.37)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_{t-1} \quad (2.38)$$

Update Cell State pada persamaan (2.37) dan (2.38) memperbaharui memori sel dengan mengintegrasikan data lama (yang dijaga oleh *forget gate*) dan data baru (yang ditentukan oleh *input gate*). Hal ini memastikan bahwa memori sel hanya menyimpan informasi yang penting.

C_t adalah memori sel pada waktu t yang menyimpan informasi penting untuk jangka panjang; f_t merupakan nilai yang menentukan seberapa banyak informasi lama yang perlu dilupakan; sedangkan $i_t \cdot C_{t-1}$ adalah memori baru yang ditambahkan ke dalam sel, yang dikendalikan oleh pintu input untuk menentukan informasi baru mana yang relevan dan perlu disimpan.

Output menentukan informasi mana dari sel memori yang akan diproduksi sebagai hasil di waktu saat ini. Hasil ini biasanya merupakan versi yang telah diubah dari sel memori yang sudah diperbaharui.

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.39)$$

Output gate (O_t) pada persamaan (2.39) berfungsi untuk menentukan data apa yang akan dikeluarkan dari memori sel pada suatu waktu tertentu. Nilai dari keluaran LSTM pada waktu t dinyatakan dengan h_t, yang didapatkan melalui perkalian antara ot dan hasil aktivasi dari memori sel Ct menggunakan fungsi $tanh$. Dalam konteks ini, W_o adalah bobot yang diterapkan pada *output gate*, sedangkan $tanh\,(C_t)$ menggambarkan versi yang telah diaktivasi dari memori sel yang diperbaharui, yang akhirnya menghasilkan representasi akhir dari keluaran jaringan pada waktu itu. *Output gate* (O_t) berfungsi untuk menentukan data mana yang akan dikeluarkan dari memori sel pada waktu tertentu. Nilai keluaran LSTM pada waktu t, yaitu h_t, diperoleh dari hasil kali antara O_t dan hasil aktivasi memori sel C_t

melalui fungsi $tanh$. Dalam konteks ini, W_o adalah bobot yang digunakan oleh *output gate*, sedangkan $tanh$ (C_t) menggambarkan versi yang teraktivasi dari memori sel yang telah diperbarui, yang dihasilkan untuk menciptakan representasi akhir dari output jaringan pada waktu itu.

Hidden state dalam persamaan (2.40), yang disebut sebagai h_t, adalah keluaran dari LSTM di setiap langkah waktu yang menyimpan informasi dari pemrosesan input yang telah dilakukan sebelumnya, ditentukan oleh kombinasi dari *cell state* C_t dan *output gate* O_t. Nilai h_t akan terus mengalami perubahan di setiap langkah waktu untuk mencerminkan dampak dari input yang baru serta pembaruan memori yang terjadi dalam jaringan.

$$h_t = O_t \cdot tanh(C_t) \tag{2.40}$$

Gabungan antara GRU dan LSTM dalam satu model bertujuan menggabungkan efisiensi komputasi dari GRU dengan kemampuan memori jangka panjang yang dimiliki LSTM. Model ini sangat cocok digunakan untuk data yang bersifat spasio-temporal dan memiliki volume besar, seperti data cuaca (suhu, tekanan, kelembapan, dan angin) serta polutan ($PM_{2.5}$, PM_{10}, NO_2, CO, O_3) yang dicatat setiap jam selama bertahun-tahun.

Beberapa karakteristik utama dari model GRU-LSTM adalah model mampu menangani ketergantungan waktu jangka panjang serta pola kompleks yang ada dalam data time series. Model ini juga fleksibel dalam menghadapi data yang memiliki nilai kosong atau interval pengambilan data yang tidak tetap. Dibandingkan dengan model LSTM murni, GRU lebih efisien dalam hal penggunaan komputasi karena memiliki jumlah parameter yang lebih sedikit.

Penelitian yang dilakukan oleh Shih dkk. (2024) menggunakan model Trans-BiGRU-QA untuk memprediksi konsentrasi merkuri di atmosfer yang memiliki perubahan dinamis secara waktu dan ruang. Hasil penelitian ini menunjukkan bahwa dengan menggabungkan GRU dengan teknik yang lebih canggih, akurasi dan stabilitas prediksi dapat meningkat, terutama pada data lingkungan yang

kompleks. Pendekatan ini menunjukkan bahwa GRU, terutama dalam bentuk bidirectional atau bentuk hibrida dengan LSTM.

Dalam konteks penelitian ini, pendekatan pemodelan berbasis GRU-LSTM sangat cocok digunakan karena beberapa alasan berikut, pertama, data yang digunakan bersifat sekuensial, multivariat, dan memiliki dimensi spasial serta temporal. Kedua, tersedia jumlah data historis yang cukup besar, yang sangat penting untuk proses pelatihan model deep learning. Ketiga, diperlukan kemampuan untuk mengenali pola yang berulang dan hubungan jangka panjang dalam data, seperti pola musiman atau harian. Keempat, model ini dapat dikembangkan lebih lanjut dengan integrasi attention atau layer spasial lainnya.

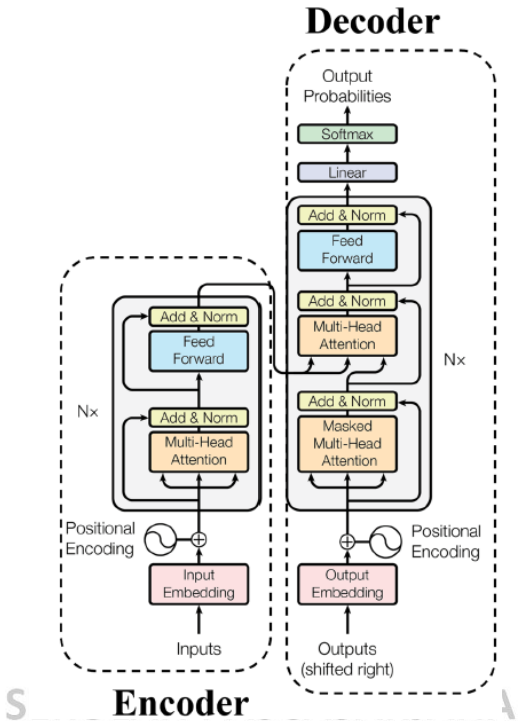
2.4.4 Transformer

Transformer adalah suatu arsitektur yang sepenuhnya didasarkan pada mekanisme perhatian, yang memungkinkan pemrosesan data urutan dilakukan secara simultan tanpa bergantung pada struktur berulang seperti RNN atau konvolusi seperti CNN. Ini menjadikan Transformer sangat efektif dan adaptif dalam menganalisis hubungan antar elemen dalam urutan input, bahkan jika ada jarak yang signifikan di antara mereka. Penelitian oleh Lin dkk. (2022) mencatat bahwa metode ini telah mengubah banyak aplikasi dalam pengolahan bahasa alami, penglihatan komputer, dan deret waktu karena kemampuannya mengidentifikasi ketergantungan jangka panjang dengan efisien. Inti dari Transformer adalah *Scale dot-product attention* yang dirumuskan pada persamaan (2.41). Arsitektur Transformer disajikan pada Gambar 2.5.

$$Attention(Q,K,V) = softmax\left(\frac{QK^t}{\sqrt{d_k}}\right)V \tag{2.41}$$

Fungsi perhatian dot-product pada persamaan (2.41) berfokus pada penilaian relevansi antara elemen-elemen dalam deretan input. Dalam pendekatan ini, Q (matriks Query), K (matriks Key), dan V (matriks value) berperan untuk menggambarkan hubungan antar elemen input. Hasil dari QK^t akan disesuaikan dengan akar kuadrat dari dimensi Key (d_k) untuk menghindari nilai gradien yang

ekstrem, baik yang terlalu besar maupun yang terlalu kecil, kemudian diterapkan fungsi aktivasi $softmax$ untuk menciptakan distribusi probabilitas sebagai bobot dalam menggabungkan nilai-nilai di dalam V secara tepat. Melalui penskalaan ini, model Transformer mampu secara efisien mengidentifikasi konteks global dalam data berurutan tanpa mengorbankan stabilitas numerik selama proses pelatihan (Zhang dkk., 2024).



Gambar 2.5 Arsitektur Model Transformer

Arsitektur Transformer adalah model jaringan saraf yang inovatif digunakan untuk memproses urutan data, terutama dalam bidang pemrosesan bahasa alami (NLP). Model ini terdiri dari dua bagian utama, yaitu Encoder dan Decoder. Encoder menerima input dan memprosesnya melalui sejumlah blok yang sama, yaitu sebanyak N blok. Setiap blok ini terdiri dari dua sub lapisan utama, yaitu *Multi-Head Attention* yang berfungsi untuk menangkap hubungan antar kata dalam kalimat, dan *Feed Forward* yang memproses informasi secara lebih lanjut. Dalam

proses ini, terdapat juga koneksi residual dan normalisasi lapisan untuk memastikan kestabilan dan akurasi model. Sebelum memasuki *Encoder*, input akan diubah menjadi bentuk *Embedding* dan diberi *Positional Encoding* untuk menjaga makna serta urutan kata dalam kalimat. Di bagian *Decoder*, prosesnya mirip dengan *Encoder*, namun terdapat beberapa perbedaan.

Decoder juga terdiri dari N blok dan menggunakan tiga jenis perhatian, yaitu *Masked Multi-Head Attention* untuk mencegah model terlihat ke masa depan dalam prediksi, *Encoder-Decoder Attention* untuk menghubungkan output *Encoder* dengan output *Decoder*, dan *Feed Forward* untuk pemrosesan lanjutan. Adapun, koneksi residual dan normalisasi lapisan juga diterapkan. Selain itu, output dari *Decoder* akan diubah menjadi bentuk *Embedding* yang disesuaikan dan diberi *Positional Encoding* kembali. Setelah selesai dari *Decoder*, model menggunakan lapisan Linear dan $Softmax$ untuk menghasilkan probabilitas masing-masing kata yang menjadi output akhir. Desain model ini memungkinkan proses paralel yang lebih baik dibandingkan model sebelumnya.

Daripada menggunakan satu fokus, Transformer memanfaatkan sejumlah *attention heads* yang bekerja secara simultan untuk menangkap berbagai pola hubungan. Hasil dari proses ini digabungkan dan diproyeksikan kembali untuk menghasilkan output akhir, sehingga memperkaya representasi konteks. Dengan arsitektur yang bersifat paralel, Transformer memasukkan informasi mengenai urutan posisi atau *positional encoding* (PE) dengan menggunakan gelombang sinus pada persamaan (2.42) dan (2.43). Ini memungkinkan Transformer mengenali urutan waktu atau posisi token. (Lin dkk., 2022)

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i}/d_{model}}\right) \tag{2.42}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i}/d_{model}}\right) \tag{2.43}$$

Ye dkk. (2025) mengusulkan model hibrida GNN-Transformer untuk meningkatkan akurasi prediksi konsentrasi $PM_{2.5}$ dengan secara simultan menangkap korelasi spasial dan temporal. Dalam GNN-Transformer, Raw Data

yang diproses oleh IPS Estimator untuk memilih subset Selected Data guna mengurangi bias. Data terpilih ini kemudian diumpankan ke Spatial Impact Modeling Layer, Graph Neural Networks (GNNs) diterapkan pada setiap *time point* untuk menangkap dan memodelkan interaksi spasial antar lokasi. Representasi spasial-temporal yang dihasilkan selanjutnya dimasukkan ke Spatiotemporal Prediction Module, yang merupakan arsitektur Transformer terdiri dari Encoder (memproses fitur spasial dari waktu ke waktu dengan Multi-Head Attention) dan Decoder (menghasilkan prediksi, juga menggunakan Multi-Head Attention termasuk *cross-attention* ke Encoder). Terakhir, FC *(Fully Connected) Layer* memproses output Decoder untuk menghasilkan Output prediksi akhir.

2.5 Evaluasi Performa Model

Matriks penilaian berfungsi sebagai instrumen krusial untuk menilai kinerja model prediksi, seperti:

2.5.1 Mean Absolute Error (MAE)

MAE ditentukan pada persamaan (2.44) dengan menghitung rata-rata dari perbedaan absolut antara nilai yang diprediksi dan nilai sebenarnya.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i | \tag{2.44}$$

2.5.2 Root Mean Square Error (RMSE)

RMSE ditentukan pada persamaan (2.45) merupakan nilai akar dari rata-rata kesalahan kuadrat, yang memiliki tingkat sensitifitas lebih tinggi terhadap data yang menyimpang.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.45}$$

2.5.3 *Mean Absolute Percentage Error* (MAPE)

Mean Absolute Percentage Error (MAPE) ditentukan pada persamaan (2.46) menggambarkan kesalahan dalam bentuk persentase dari nilai yang sebenarnya. MAPE bermanfaat untuk menilai keakuratan model dalam berbagai ukuran karena menampilkan tingkat kesalahan dalam format persentase. Nilai MAPE yang lebih rendah menunjukkan akurasi prediksi yang lebih baik, dan jika MAPE = 0%, berarti prediksi dianggap sempurna (terjadi).

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{2.46}$$

n adalah jumlah sampel, y_i merupakan nilai aktual dari data ke-i, dan $\hat{y}_i$ adalah nilai prediksi dari data ke-i. Untuk MAE, kesalahan dihitung sebagai $|y_i - \hat{y}_i|$ yaitu selisih absolut antara nilai aktual dan prediksi. Untuk RMSE, digunakan $(y_i - \hat{y}_i)^2$ yang merupakan selisih kuadrat antara nilai aktual dan prediksi. Sedangkan pada MAPE, kesalahan dinyatakan dalam bentuk persentase sebagai $\left| \frac{y_i - \hat{y}_i}{y_i} \right|$ yang menunjukkan nilai absolut dari persentase kesalahan antara nilai aktual dan nilai prediksi.

2.5.4 Durasi Pelatihan Model *(Train Time)*

Train Time merujuk pada jumlah waktu yang diperlukan oleh sebuah model untuk menjalani keseluruhan proses pelatihan dari awal hingga akhir. Durasi pelatihan ini dihitung dengan cara mengurangkan waktu dimulainya training dari waktu selesainya training pada persamaan (2.47).

$$T_{train} = T_{end} - T_{start} \tag{2.47}$$

Total waktu pelatihan bisa didapatkan dengan menjumlahkan durasi tiap epoch sebagaimana ditunjukkan pada persamaan (2.48).

$$\int_{i=1}^{E} T_i \tag{2.48}$$

Dalam *train time*, E adalah jumlah epoch dan T_i adalah waktu yang dibutuhkan pada epoch ke-i. *Train Time* adalah metrik penting untuk mengevaluasi seberapa efisien komputasi suatu model, karena model yang memerlukan waktu pelatihan lebih sedikit dianggap memiliki efisiensi lebih baik, terutama jika hasil prediksinya sebanding dengan model lain.

2.5.5 Inference Sample

Inference sample merupakan bagian dari dataset yang digunakan untuk menilai kemampuan generalisasi model terhadap data baru yang tidak dilibatkan pada tahap pelatihan (training). Model sudah belajar dari training sample, dan kemudian diuji pada *inference/test sample* untuk mengukur seberapa baik model memprediksi data baru. Sampel inferensi dinotasikan pada persamaan (2.49).

$$D_{test} = \{(x_i, y_i)\}_{i=1}^{n} \tag{2.49}$$

x_i merepresentasikan fitur input dan $y_i$adalah nilai aktual atau *ground truth*, sedangkan $\hat{y}_i = f(x_i)$ merupakan hasil prediksi model pada sampel ke-i. Evaluasi performa pada *inference sample* dilakukan dengan menghitung selisih antara nilai aktual dan nilai prediksi menggunakan berbagai metrik kuantitatif, seperti *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE).

2.6 Uji Statistik

2.6.1 *Friedman Test*

Uji Friedman adalah metode statistik non-parametrik untuk membandingkan tiga atau lebih perlakuan pada sampel berpasangan, setara dengan *repeated measures* ANOVA namun tanpa asumsi distribusi normal. Uji ini umum digunakan untuk mengevaluasi kinerja beberapa algoritma machine learning pada berbagai dataset atau menilai efektivitas beberapa perlakuan pada subjek yang sama.

Prosesnya dilakukan dengan memberi peringkat pada tiap metode dalam setiap blok atau sampel, lalu menghitung rata-rata peringkat untuk menentukan apakah terdapat perbedaan signifikan antar metode. Hipotesis nol (H_0) dari uji Friedman menyatakan bahwa semua metode menunjukkan kinerja yang serupa, sementara hipotesis alternatif (H_1) mengindikasikan bahwa setidaknya terdapat satu metode yang berbeda secara signifikan. Statistik uji Friedman x_F^2 dihitung dengan menggunakan persamaan (2.50).

$$x_F^2 = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1) \tag{2.50}$$

Dalam uji Friedman, k adalah total metode yang dianalisis, nnn adalah total blok atau sampel, dan R_j^2 merupakan total peringkat untuk metode ke-j. Nilai x_F^2 yang diperoleh kemudian dibandingkan dengan nilai kritis chi-square dengan derajat kebebasan $k+1$ pada tingkat signifikansi tertentu, misalnya $\alpha = 0{,}05$. Jika x_F^2 melebihi nilai kritis, hipotesis nol ditolak sehingga terdapat perbedaan signifikan antar metode; jika tidak, hipotesis nol diterima.

2.6.2 Pairwise Test

Uji Nemenyi digunakan untuk membandingkan peringkat rata-rata tiap metode dengan mempertimbangkan jumlah metode dan jumlah blok sampel. Evaluasi ini didasarkan pada nilai *Critical Difference* (CD) yang dihitung menggunakan persamaan (2.51).

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}} \tag{2.51}$$

k adalah jumlah metode yang dibandingkan, n adalah jumlah blok atau sampel, dan q_α adalah nilai *studentized range statistic* pada tingkat signifikansi α untuk k perlakuan. Setelah nilai CD diperoleh, perbandingan dilakukan dengan menghitung selisih peringkat rata-rata antar metode. Jika selisih tersebut lebih besar daripada CD, perbedaan dianggap signifikan; jika lebih kecil atau sama dengan CD, maka perbedaan tidak signifikan.